# Homework M2: Advanced Containerization Concepts

This is one possible solution for the tasks included in the homework

All steps that follow assume that we decided to base our solution on a **Debian 12** box

Of course, we can use any other box

## The Environment

Prepare the environment by creating a **Vagrantfile** with the following content

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  config.vm.define "docker" do |docker|

    docker.vm.box="shekeriev/debian-12.11"

    docker.vm.hostname = "docker.do1.lab"

    docker.vm.network "private\_network", ip: "192.168.89.100"

    docker.vm.synced\_folder "vagrant/", "/vagrant"

    docker.vm.provision "shell", path: "docker.sh"

  end

end

\* the **docker.vm.synced\_folder** can be skipped (removed) or you can put there the **Dockerfile** (either of the three) and **index.html** files

Create a **docker.sh** file to install and configure the necessary packages

#!/bin/bash

export DEBIAN\_FRONTEND=noninteractive

echo "\* Add hosts ..."

echo "192.168.89.100 docker.do1.lab docker" >> /etc/hosts

echo "\* Add any prerequisites ..."

apt-get update

apt-get install -y ca-certificates curl

echo "\* Add Docker key ..."

install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc

chmod a+r /etc/apt/keyrings/docker.asc

echo "\* Add Docker repository ..."

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian $(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

echo "\* Install Docker ..."

apt-get update

apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

echo "\* Add vagrant user to docker group ..."

usermod -aG docker vagrant

Start the environment with

**vagrant up**

## The Image

Enter the **docker** machine with

**vagrant ssh**

Create **index.html** file with the following content

<h1>Hello from my first container!</h1>

### AlmaLinux Based Image

Create a **Dockerfile.centos** with the following content

FROM almalinux:9.6

RUN dnf install -y httpd

ADD index.html /var/www/html/

EXPOSE 80

CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]

Build the image with

**docker build -t hw2 -f Dockerfile.almalinux .**

### openSUSE Based Image

Create a **Dockerfile.opensuse** with the following content

FROM opensuse/leap:15.6

RUN zypper install -y apache2

ADD index.html /srv/www/htdocs/

EXPOSE 80

CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]

Build the image with

**docker build -t hw2 -f Dockerfile.opensuse .**

### Ubuntu Based Image

Create a **Dockerfile.ubuntu** with the following content

FROM ubuntu:24.04

RUN apt-get update -y

RUN env DEBIAN\_FRONTEND=noninteractive apt-get install -y apache2

ADD index.html /var/www/html/

RUN chown -R www-data:www-data /var/www

ENV APACHE\_RUN\_USER=www-data

ENV APACHE\_RUN\_GROUP=www-data

ENV APACHE\_PID\_FILE=/var/run/apache2/apache2.pid

ENV APACHE\_RUN\_DIR=/var/run/apache2

ENV APACHE\_LOCK\_DIR=/var/lock/apache2

ENV APACHE\_LOG\_DIR=/var/log/apache2

RUN mkdir -p $APACHE\_RUN\_DIR $APACHE\_LOCK\_DIR $APACHE\_LOG\_DIR

EXPOSE 80

CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]

Build the image with

**docker build -t hw2 -f Dockerfile.ubuntu .**

## The Container

Run the container with

**docker container run -d -p 80:80 hw2**

Open browser on the host and navigate to [**http://192.168.89.100**](http://192.168.89.100)

There should be a working web application

## The Cleaning

Destroy the machine

**vagrant destroy --force**