# Solution M2: Introduction to Docker

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

All steps that follow assume that we decided to base our solution on **Debian 11** 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.ssh.insert\_key = false

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

    docker.vm.box="shekeriev/debian-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

echo "\* Update packages ..."

apt-get update -y && apt-get upgrade -y

echo "\* Install additional software ..."

apt-get install -y apt-transport-https ca-certificates curl \

gnupg-agent software-properties-common

echo "\* Install the repository key ..."

curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -

echo "\* Add the repository ..."

add-apt-repository \

   "deb [arch=amd64] https://download.docker.com/linux/debian \

   $(lsb\_release -cs) \

   stable"

echo "\* Install the software ..."

apt-get update -y

apt-get install -y docker-ce docker-ce-cli containerd.io

echo "\* Add the vagrant user to the 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>

### CentOS Based Image

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

FROM centos

RUN sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-\*

RUN sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-\*

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.centos .**

### openSUSE Based Image

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

FROM opensuse/leap:15.3

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:20.04

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 a working web application

## The Cleaning

Destroy the machine

**vagrant destroy --force**