**Note: These labs are prepared for Centos/Fedora OS, and some of the commands will fail on another Linux distribution**

# Lab1: Docker basics

**Exercise 1:** Install docker

1. Log in to your VM.
2. Start terminal and elevate your privileges to root.
3. Run **yum install docker.**

* I have Docker installed previously.

1. After installation is finished, start docker by running this command **systemctl start docker.**
2. Also enable docker service automatic start with command **systemctl enable docker.**
3. Run **docker version** to see installed version.

A screenshot of a computer

Description automatically generated

1. Run **docker help** to see list of available commands.

A picture containing text, screenshot, font

Description automatically generated

1. Search for a command (switch) that will show system-wide information for your instance of docker.

* **docker system info** displays system-wide information.

1. Test it by running **docker <command you have discovered>.**

A screenshot of a computer program

Description automatically generated with medium confidence

1. From the output try to find where the information of number of containers and images is.
2. Also try to find whether this docker is part of a swarm.

A screenshot of a computer

Description automatically generated

**Hint:** Use Linux grep filtering if the output of this command is too verbose for you.

# Lab2: Creating images

**Excercise1:** Build a simple image

1. Create a Docker container that executes a simple bash script. Go to your home directory and run **mkdir test.**

Run **cd test.**

1. Create a simple script. Run **vi test.sh.**
2. Write the following in your script file: #!/bin/bash

sleep 30

exit 1

1. Save the file. In vi editor press **:wq.**
2. Create a docker file. Run **vi Dockerfile.**
3. Write the following in our Dockerfile: FROM alpine

ADD test.sh /

CMD /bin/bash /test.sh

1. Save your Dockerfile.

A screenshot of a computer program

Description automatically generated with medium confidence

1. Build your image. Run **docker build –t my-image1 ./**

A screenshot of a computer

Description automatically generated with medium confidence

1. Now spawn a container. Run **docker run - -name my-test1 my-image1**.
2. Do a **docker ps –a.** Do you see your container running? no
3. Do a **docker logs my-test1**. What is the output of the log?

A screenshot of a computer

Description automatically generated with medium confidence

Note: Because alpine is very light Image it does not have bash binaries.

1. Delete my-test. Run **docker rm –f my-test1**.
2. Delete my–image. Run **docker rmi –f my-image1**.

A screenshot of a computer screen

Description automatically generated with low confidence

Now correct your Dockerfile. In the last line replace **CMD /bin/bash /test.sh** with **CMD**

**/bin/sh /test.sh.**

1. Build your image. Run **docker build –t my-image1 ./**
2. Now spawn a container again. Run **docker run - -name my-test1 my-image1**.
3. Do a **docker ps –a.** Do you see your container running?
4. Delete my-test. Run **docker rm –f my-test1**.
5. Delete my–image. Run **docker rmi –f my-image1.**