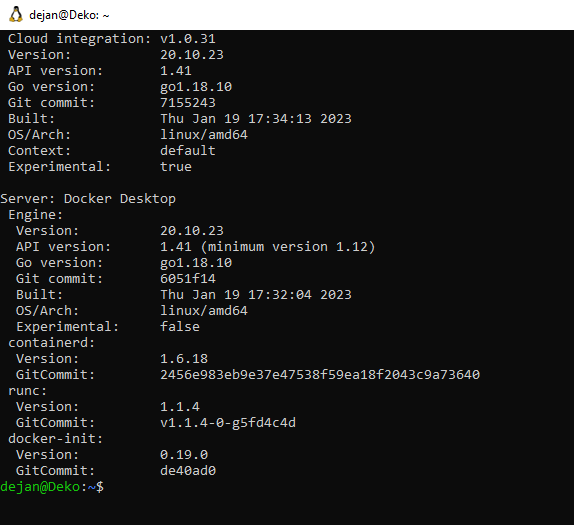
**Note: These labs are prepared for Centos/Fedora OS, and some of the commands will fail on another Linux distribution**

# Lab1: Docker basics

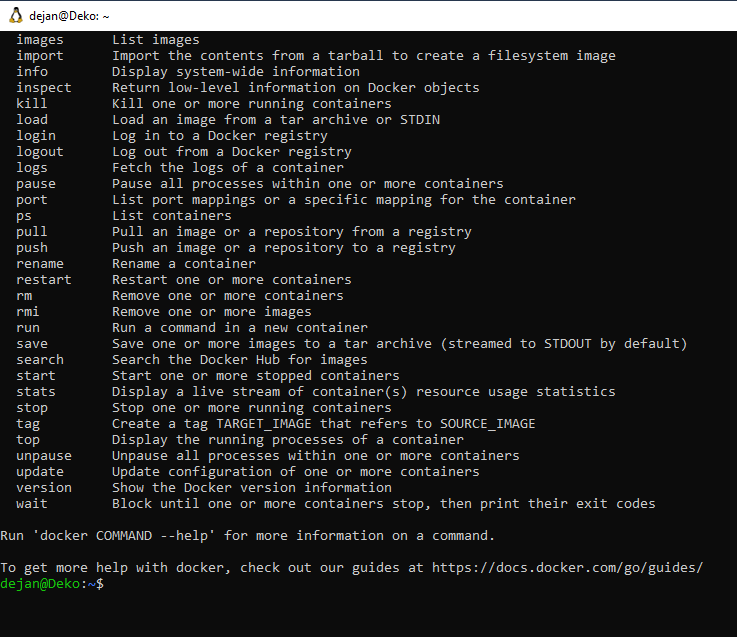
**Exercise 1:** Install docker

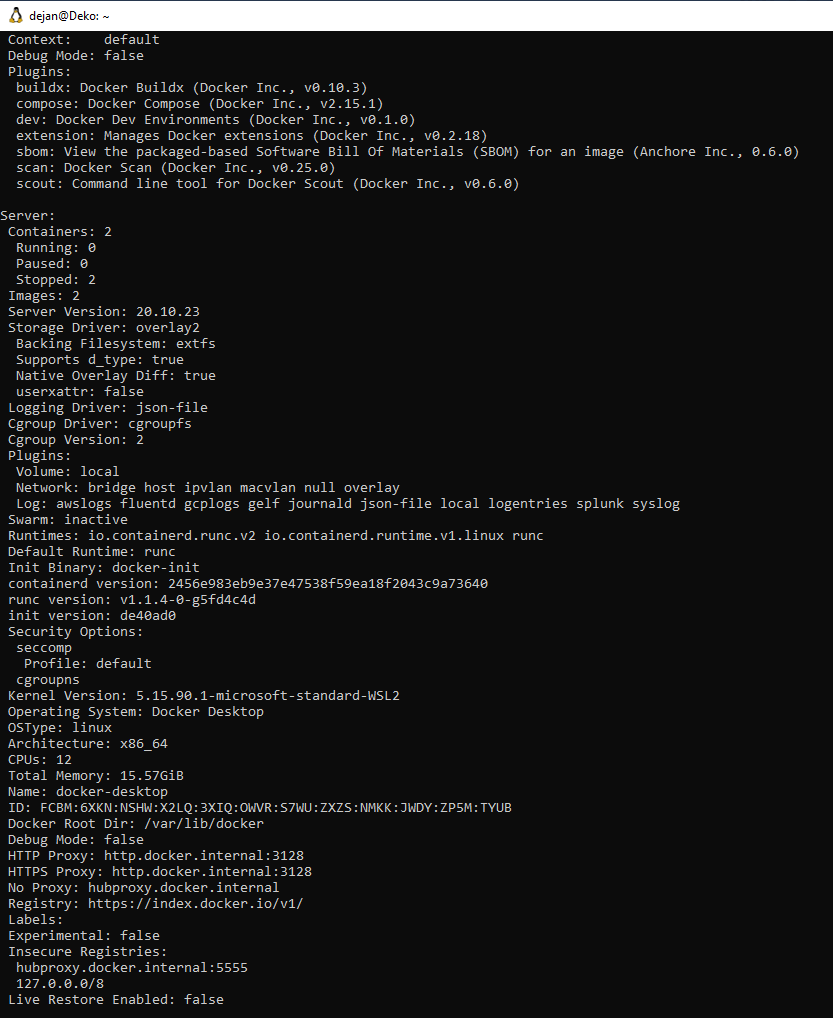
1. Log in to your VM.
2. Start terminal and elevate your privileges to root.
3. Run **yum install docker.**
4. After installation is finished, start docker by running this command **systemctl start docker.**
5. Also enable docker service automatic start with command **systemctl enable docker.**
6. Run **docker version** to see installed version.

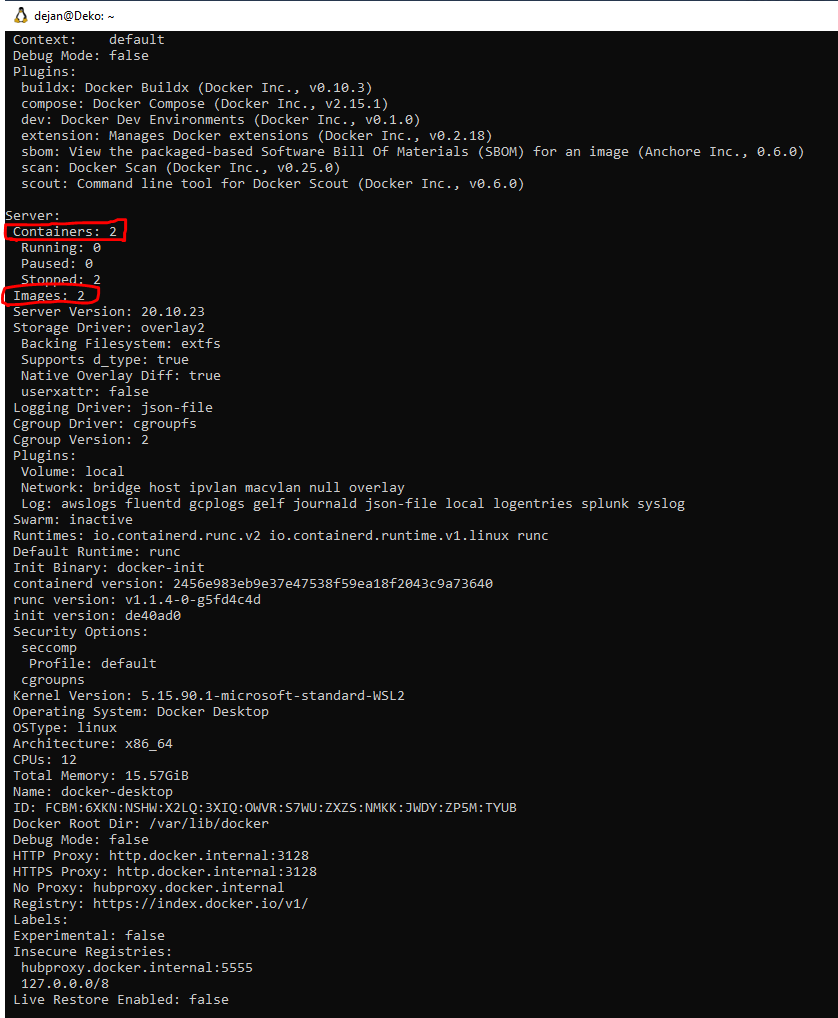
**I have docker desktop installed so I skipped those steps**



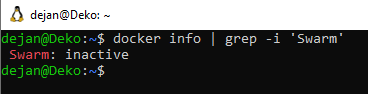
1. Run **docker help** to see list of available commands.



1. Search for a command (switch) that will show system-wide information for your instance of docker.
2. Test it by running **docker <command you have discovered>.** 
3. From the output try to find where the information of number of containers and images is.



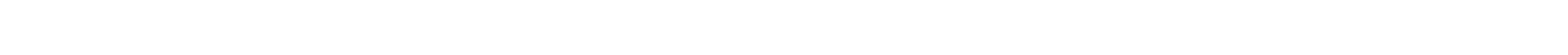
1. Also try to find whether this docker is part of a swarm. **Hint:** Use Linux grep filtering if the output of this command is too verbose for you.



# 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.**



69

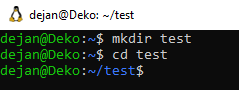
Bulgaria Blvd., Infinity Tower B, fl. 8, Triaditza

district, 1404 Sofia, Bulgaria | +359 2424 6484 |

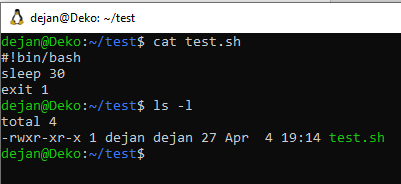
sales@scalefocus.com

| www.scalefocus.com





1. Create a simple script. Run **vi test.sh.**
2. Write the following in your script file: #!/bin/bash sleep 30 exit 1
3. Save the file. In vi editor press **:wq.**



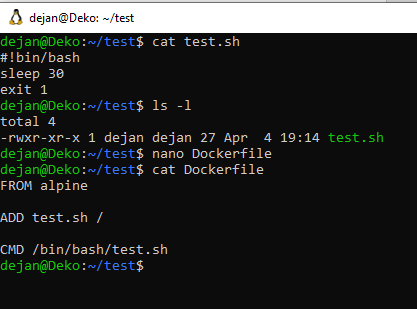
1. Create a docker file. Run **vi Dockerfile.**
2. Write the following in our Dockerfile:

FROM alpine

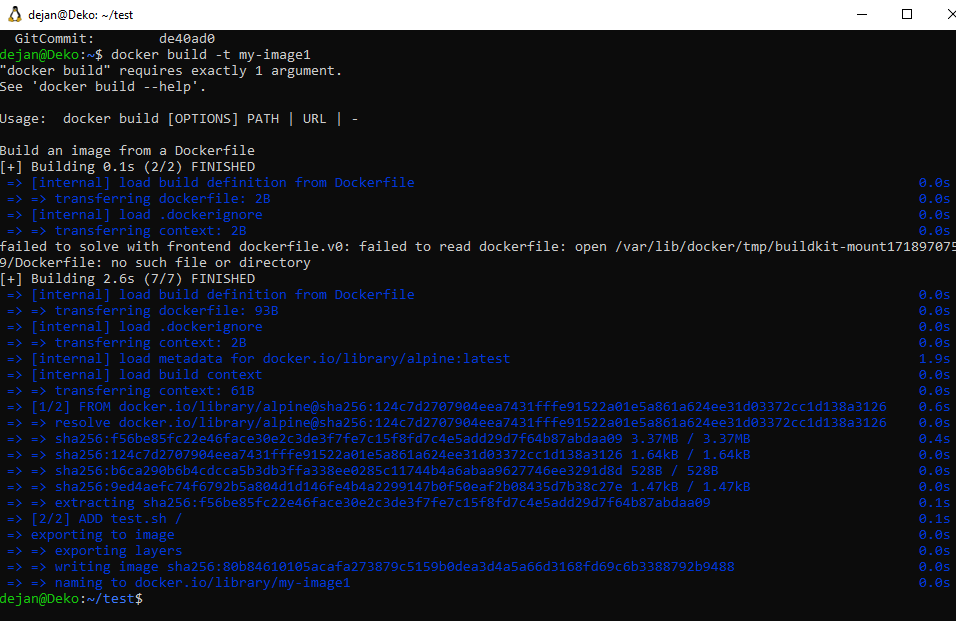
ADD test.sh /

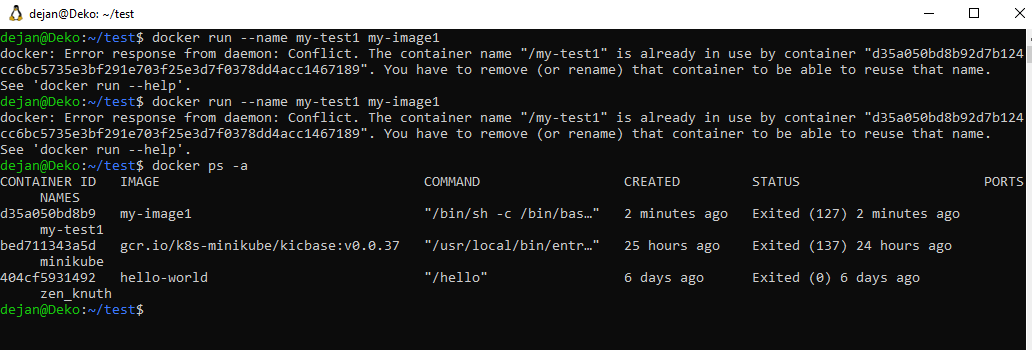
CMD /bin/bash /test.sh

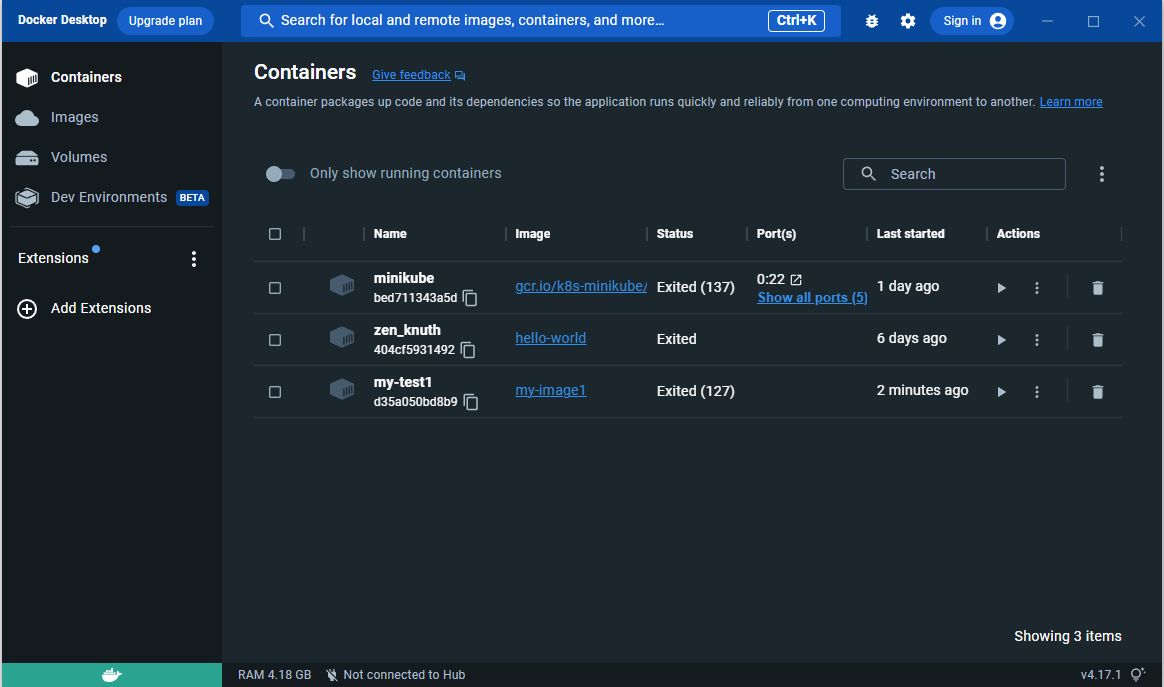
1. Save your Dockerfile.



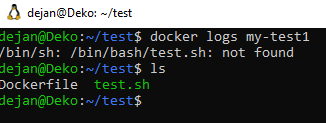
1. Build your image. Run **docker build –t my-image1 ./**



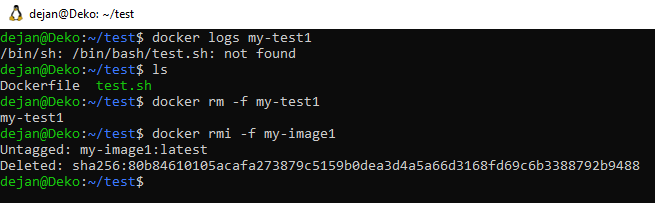
1. Now spawn a container. Run **docker run - -name my-test1 my-image1**. 10. Do a **docker ps –a.** Do you see your container running? \_\_\_\_\_\_\_ 

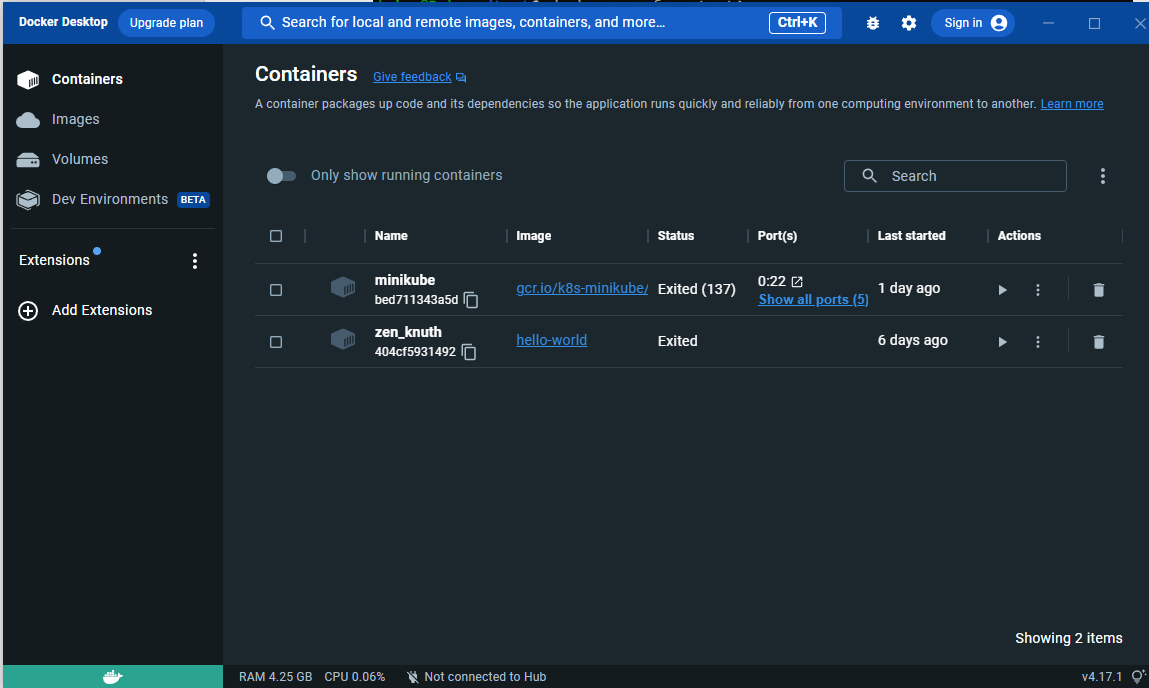


1. Do a **docker logs my-test1**. What is the output of the log? Note: Because alpine is very light Image it does not have bash binaries.



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

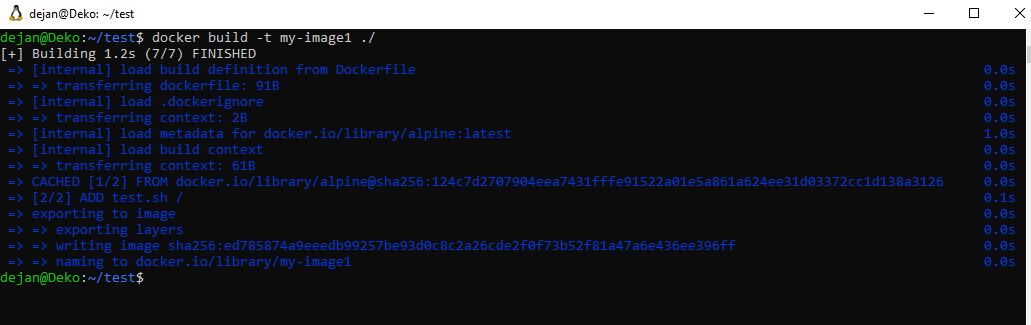




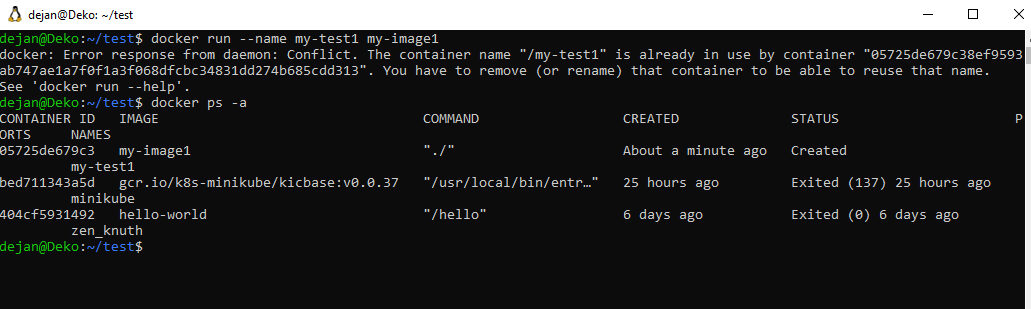
**As we can see it is deleted**

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

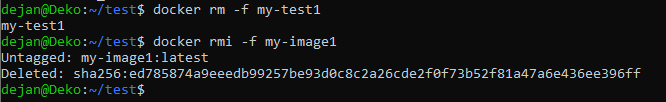
1. Build your image. Run **docker build –t my-image1 ./**

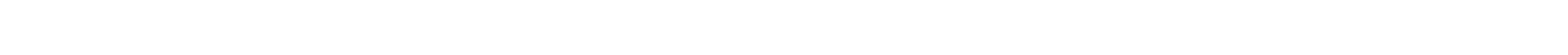


1. Now spawn a container again. Run **docker run - -name my-test1 my-image1**. 16. Do a **docker ps –a.** Do you see your container running?\_\_\_\_\_\_



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





69

Bulgaria Blvd., Infinity Tower B, fl. 8, Triaditza

district, 1404 Sofia, Bulgaria | +359 2424 6484 |

sales@scalefocus.com

| www.scalefocus.com

