**Lab1: Docker basics**

**Exercise 1: Install docker:**

1. Log in to your VM.

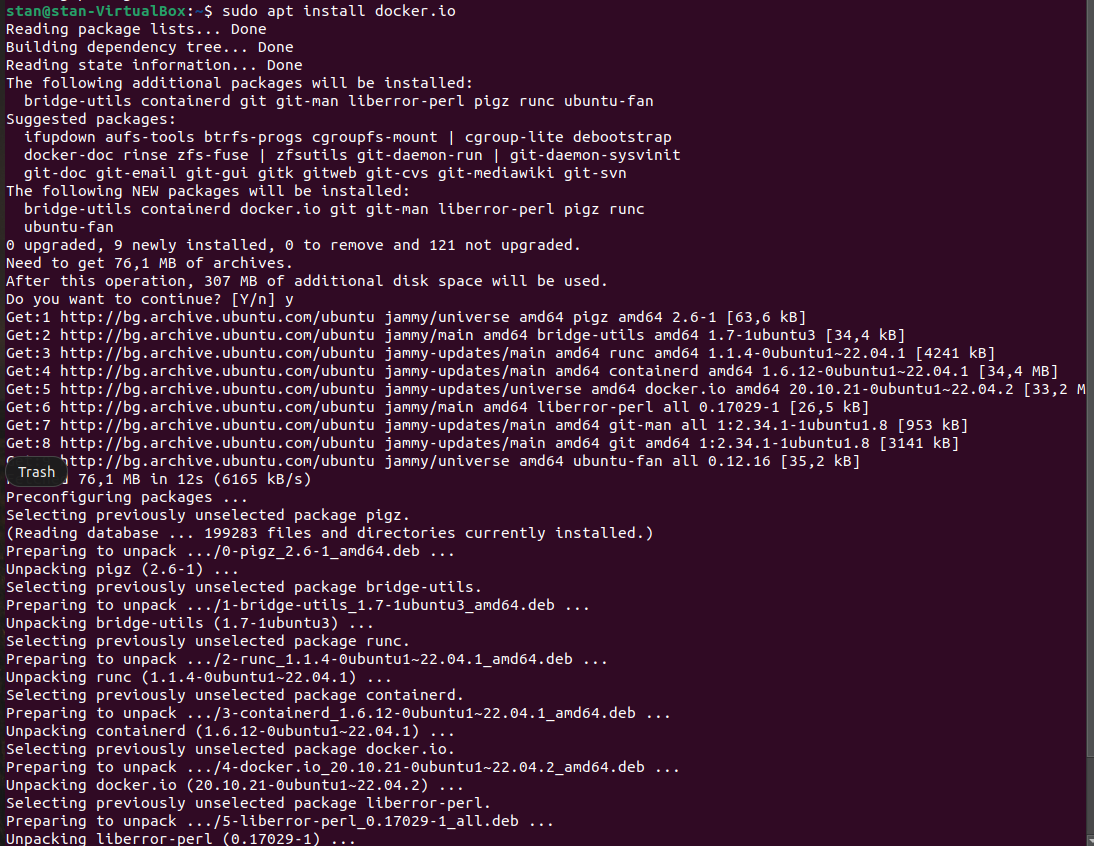
2. Start terminal and elevate your privileges to root.

3. Run yum install docker.

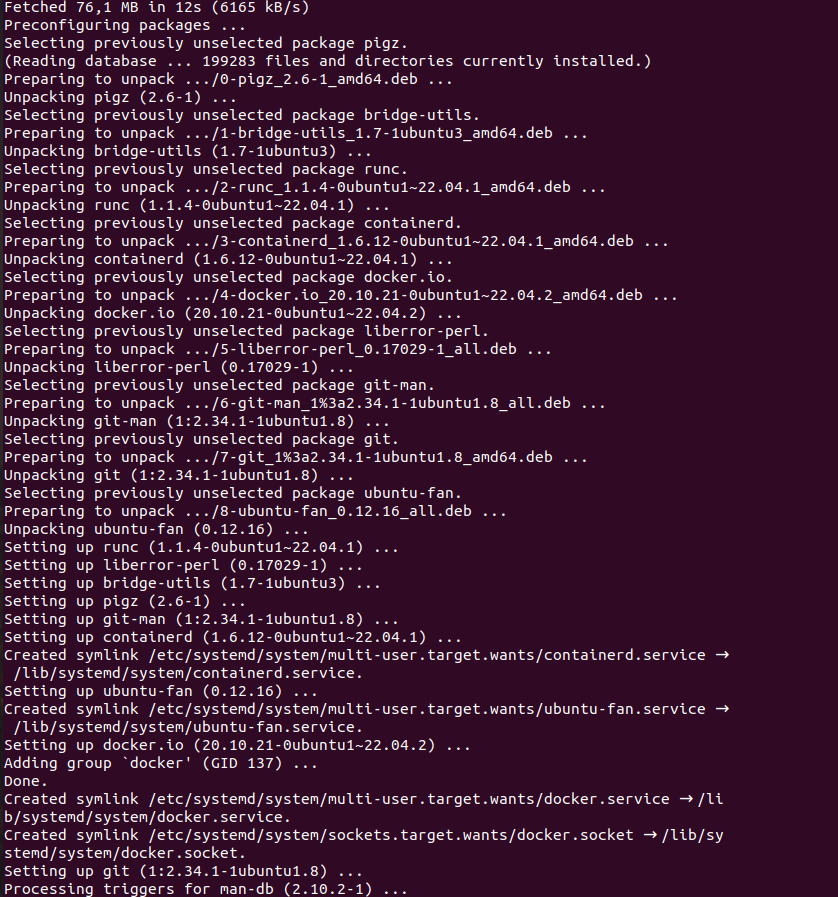
4. After installation is finished, start docker by running this command systemctl start docker.

5. Also enable docker service automatic start with command systemctl enable docker.

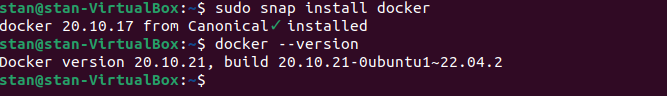
6. Run docker version to see installed version.



* downloading docker IO.

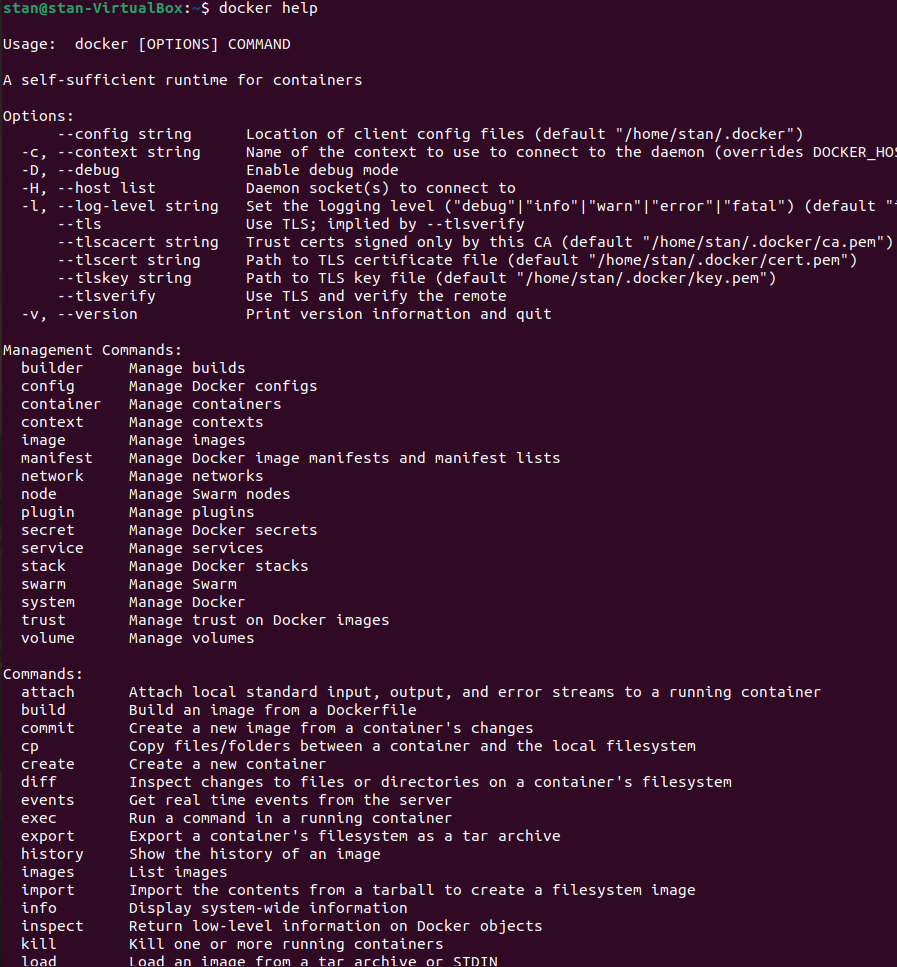


* downloading docker IO part2.

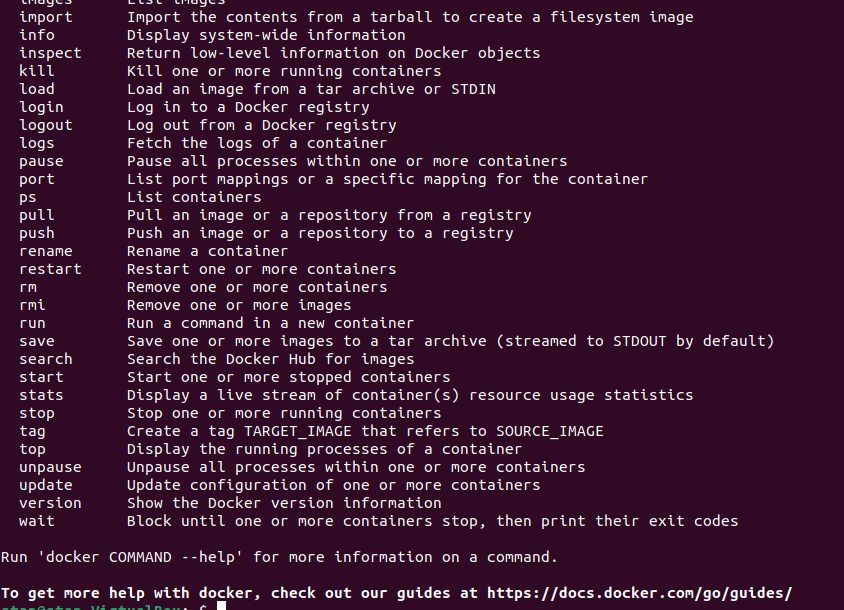


* docker version and installing docker.

7. Run docker help to see list of available commands.



* Docker help command



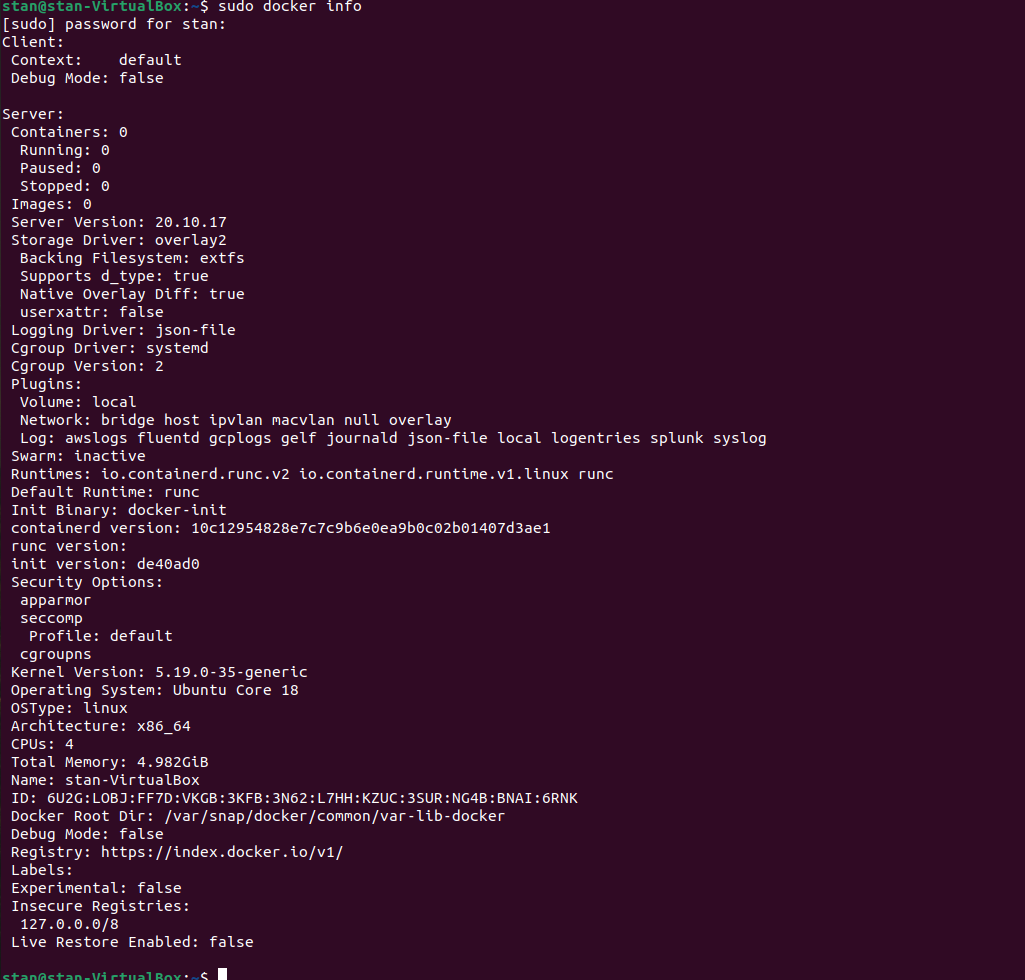
- Docker help command part 2.

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

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

10. From the output try to find where the information of number of containers and images is.

11. Also try to find whether this docker is part of a swarm

 - Swarm:inactive.

- docker info command

**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.

2. Create a simple script. Run vi test.sh.

3. Write the following in your script file:

#!/bin/bash

sleep 30

exit 1

4. Save the file. In vi editor press :wq.

5. Create a docker file. Run vi Dockerfile.

6. Write the following in our Dockerfile:



* Creating directory and the 2 files and editing them, running the image

FROM alpine

ADD test.sh /

CMD /bin/bash /test.sh

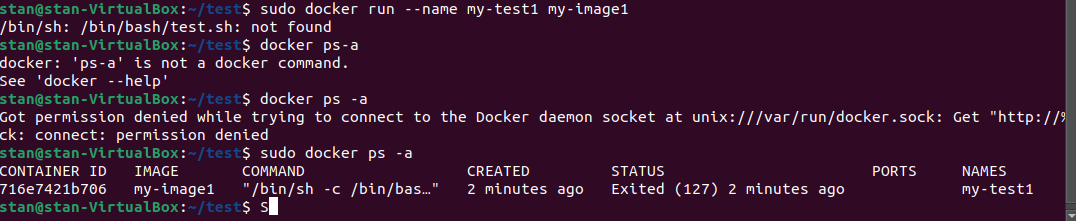
7. Save your Dockerfile.

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

9. Now spawn a container. Run docker run - -name my-test1 my-image1.

10. Do a docker ps –a. Do you see your container running?

11. Do a docker logs my-test1. What is the output of the log?

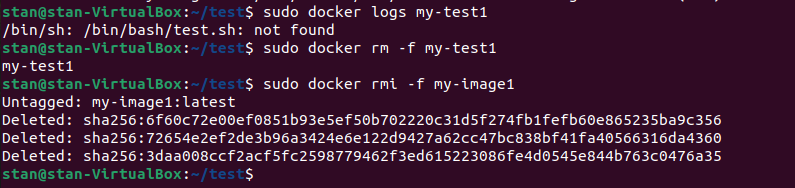


* It created the container, but the image does not have bash binaries

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

12. Delete my-test. Run docker rm –f my-test1.

13. Delete my–image. Run docker rmi –f my-image1.



* deleting the image and the container.

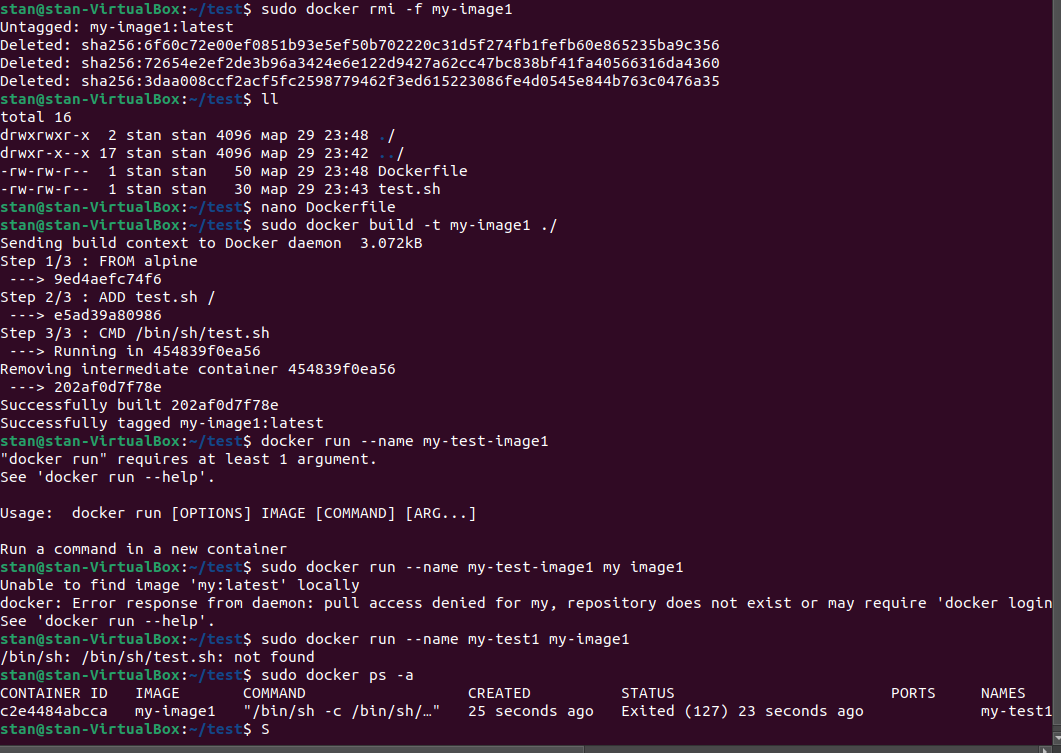
Now correct your Dockerfile. In the last line replace CMD /bin/bash /test.sh with CMD

/bin/sh /test.sh.

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

15. Now spawn a container again. Run docker run - -name my-test1 my-image1.

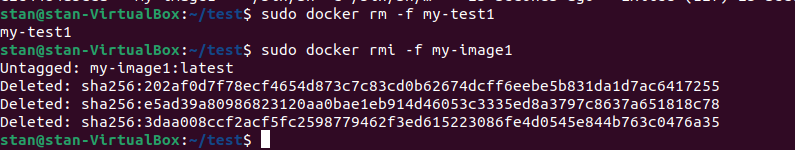
16. Do a docker ps –a. Do you see your container running?\_\_\_\_\_\_



- Changing the Dockerfile and creating the image and conainer again.

17. Delete my-test. Run docker rm –f my-test1.

18. Delete my–image. Run docker rmi –f my-image1



* Deleting the test and image(2-nd time).