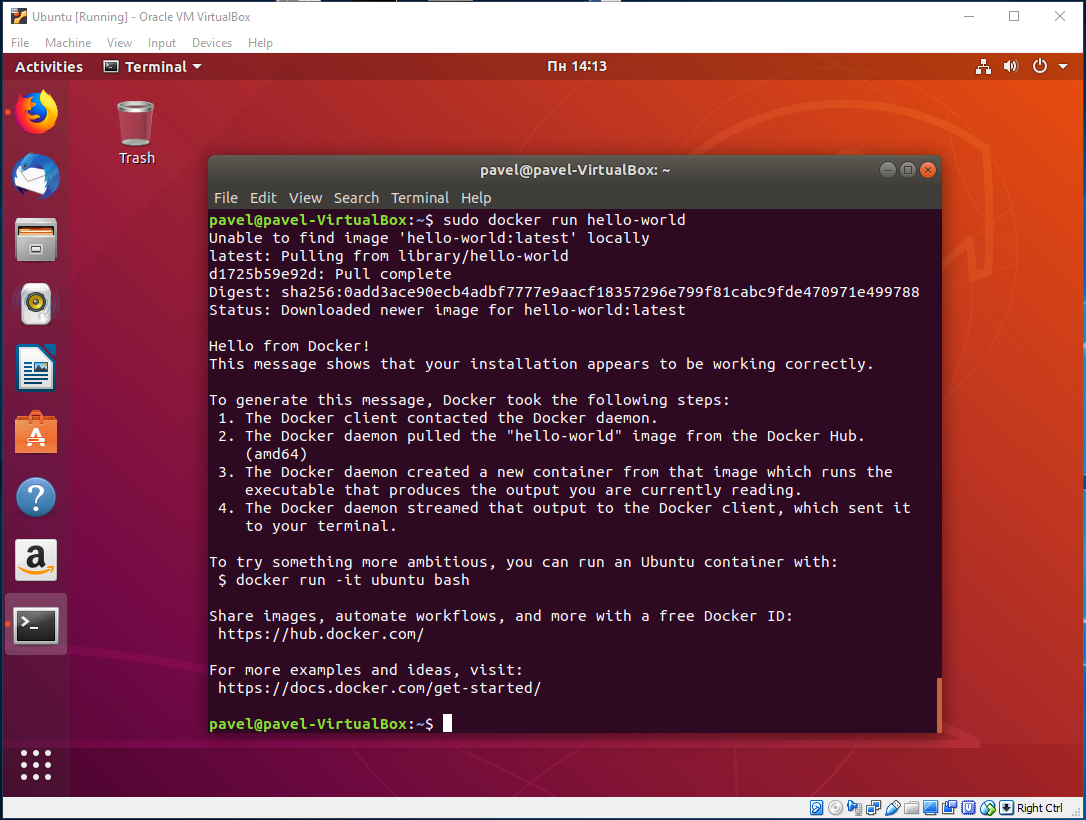
**Docker homework.**

1. **Install docker on your machine**

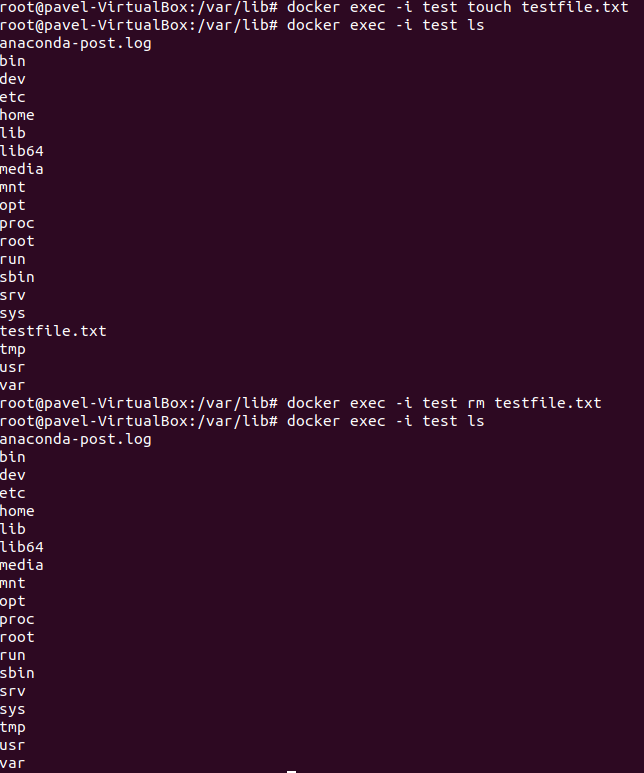
Here is docker running hello-world on my machine:



1. **Create, edit, delete files within a docker container.**

In order to do all of this, we need to use the command docker exec -i container\_name our\_command

This is how you can create/delete a text file in a docker container:



Or you can instead log into the interactive mode by doing docker exec -it container\_name bash and execute commands from there.

In order to change a file, you can log into the interactive mode and do vim text\_file.txt

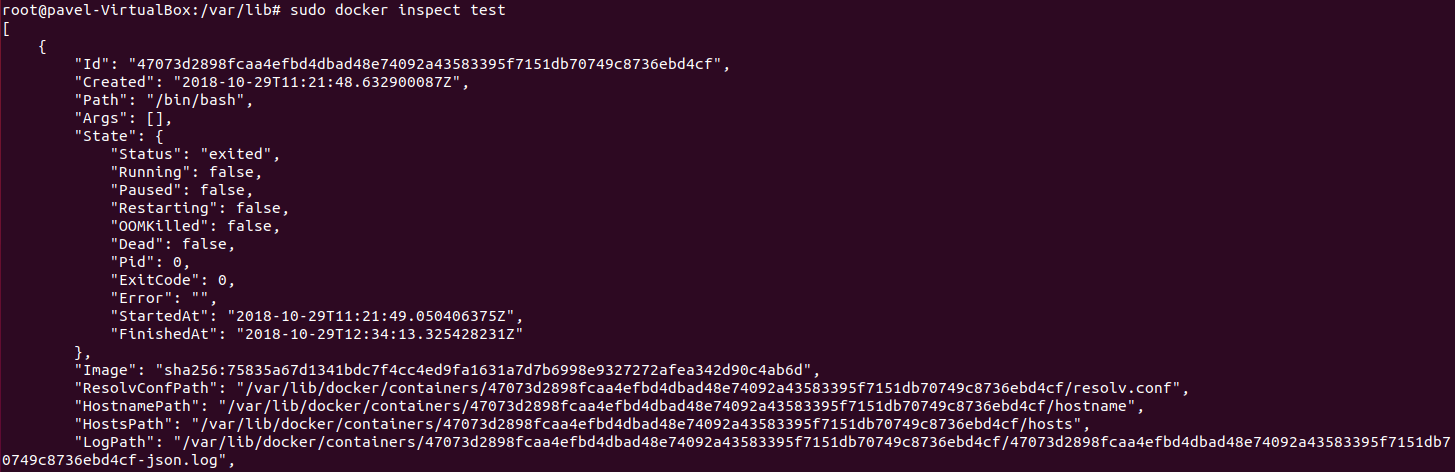
1. **Find running docker container log file location by reading configuration (read log file and make screenshot).**

All the docker container logs are located at:

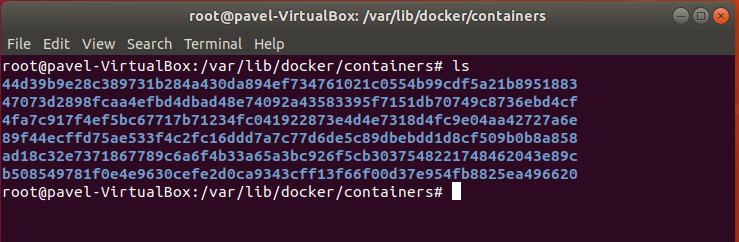
/var/lib/docker/containers/<container-id>/<container-id>-json.log

This can be verified by executing the following command:

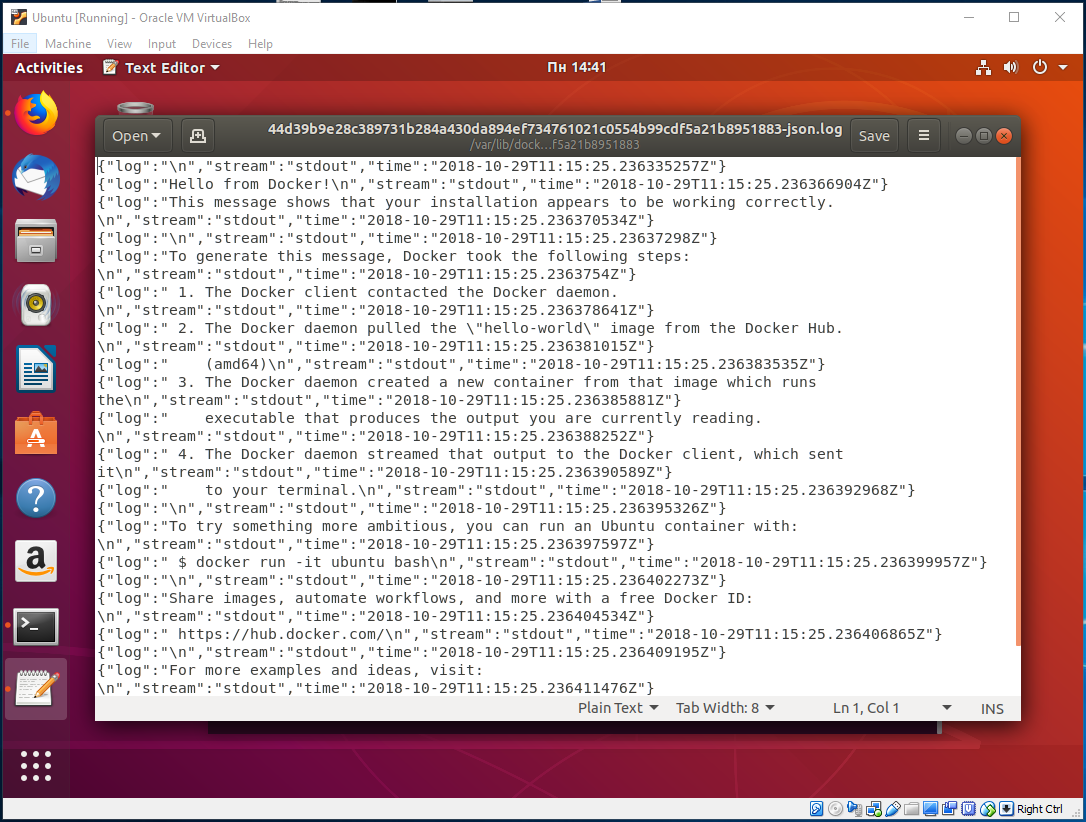
sudo docker inspect test



Let’s navigate there and we will see:

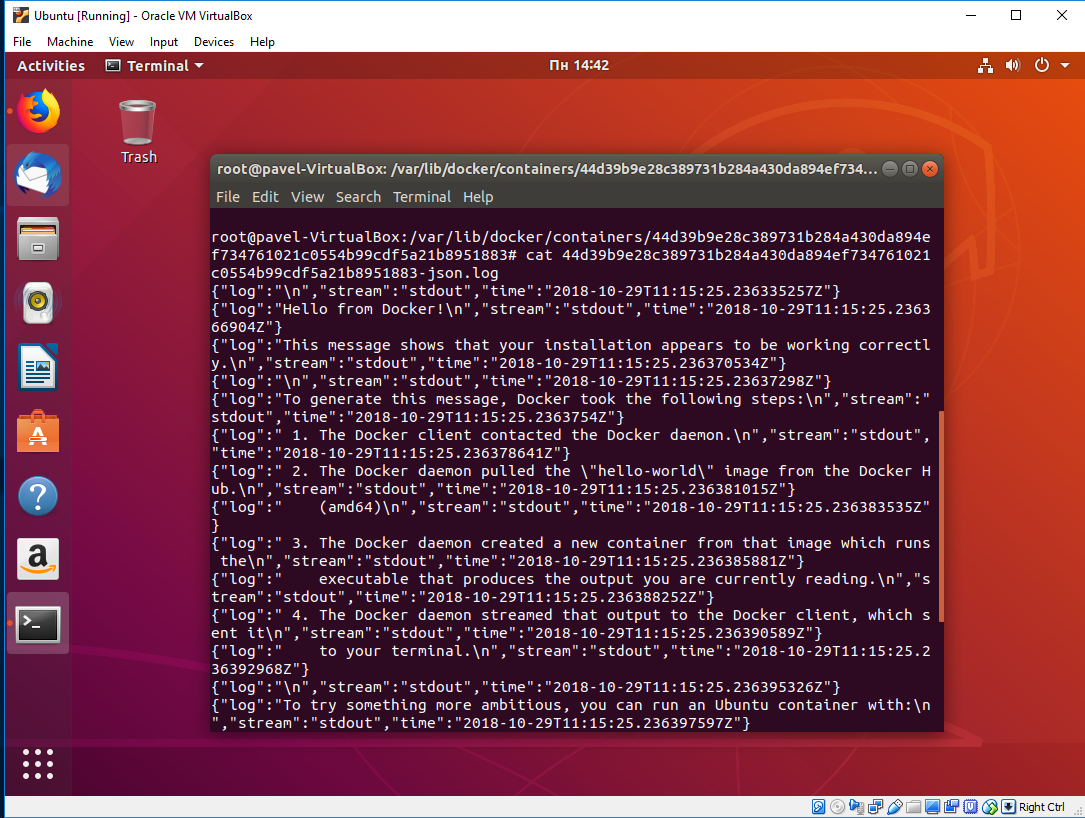


Let’s view the log file for 44d3… which is a container of the hello-world image:



1. **Read running docker container log file using cat command (make screenshot).**

Executing the cat command, we get the same result:

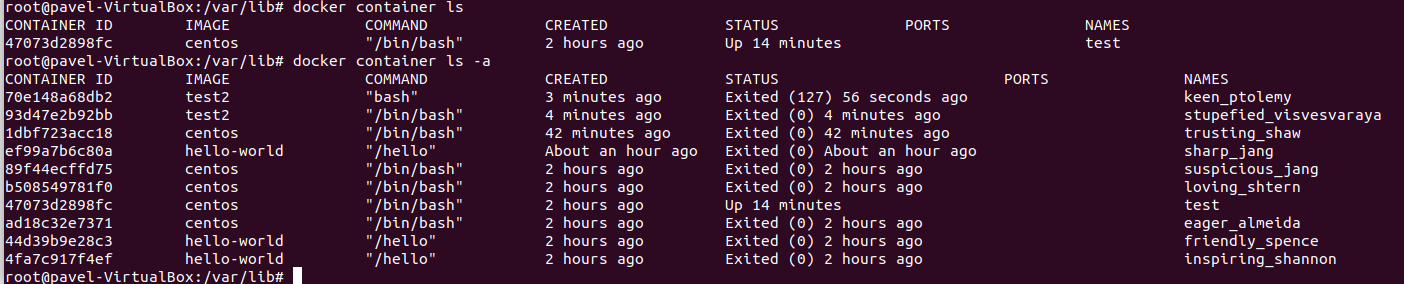


1. **Add file to running docker container and save container. Next time docker container runs it should have file present. (make several screenshots).**

In order to create a new image from a container you have to execute this command:

docker commit container\_name new\_image\_name

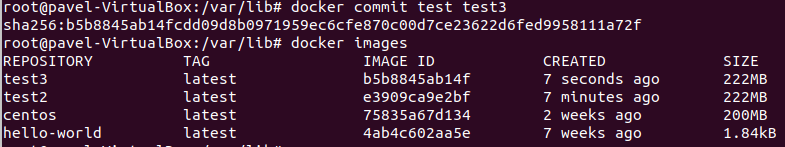
Let’s list all of my containers and their images:



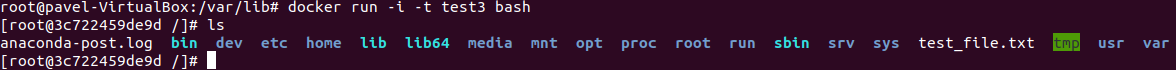
As you can see I have a container called “test”, corresponding to the “centos” image. Let’s look at it’s file system:



As you can see, I have a file, called “test\_file.txt” in there. Let’s now create an image “test3” from this container:



Now, “test3” is among all the other images, we can now run it as a container and look at it’s file system:



We can see that “test\_file.txt” is present in the copy of the image.

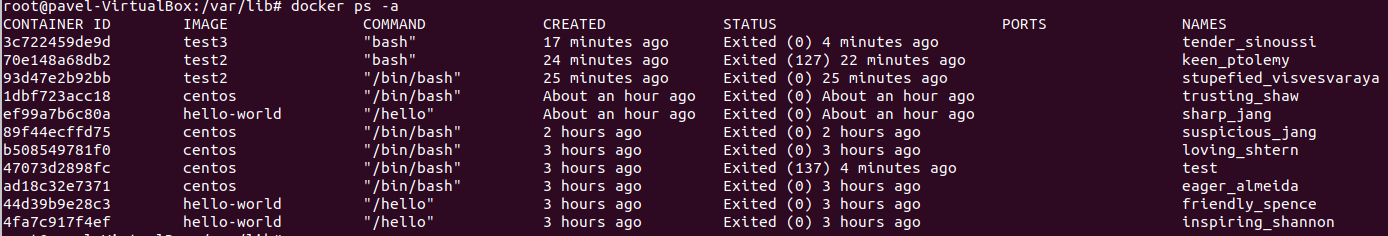
1. **Stop docker container and clean up resources.**

In order to stop a docker container you need to execute this command: docker stop container\_name

To clean up, or remove all stopped containers, you can execute this command: docker container prune

Also, as a means of cleanup, you can remove all dangling images (these are images that don’t have containers) by executing: docker image prune -a

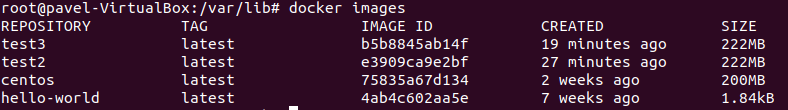
Let’s list our containers before we remove them:



And after:



Let’s list our images before we remove them:



And after:

