Experiment 10

Aim: To study and implement Containerization using Docker.

Requirements: Docker

Theory:

What is Docker?

Docker is an open-source tool designed to create, deploy and run applications with ease by using containers. Docker fits in the deployment phase of the DevOps pipeline. DevOps can be defined as a culture that primarily focuses on improved collaboration, communication and integration between Development and Operations teams.

DevOps improves collaboration and productivity by:

- Automating infrastructure provision
- Automating workflows for building, testing and deploying applications
- Continuously measuring application performances

What is a Container?

A Container is a package which has everything except the Operating System to run the software application.

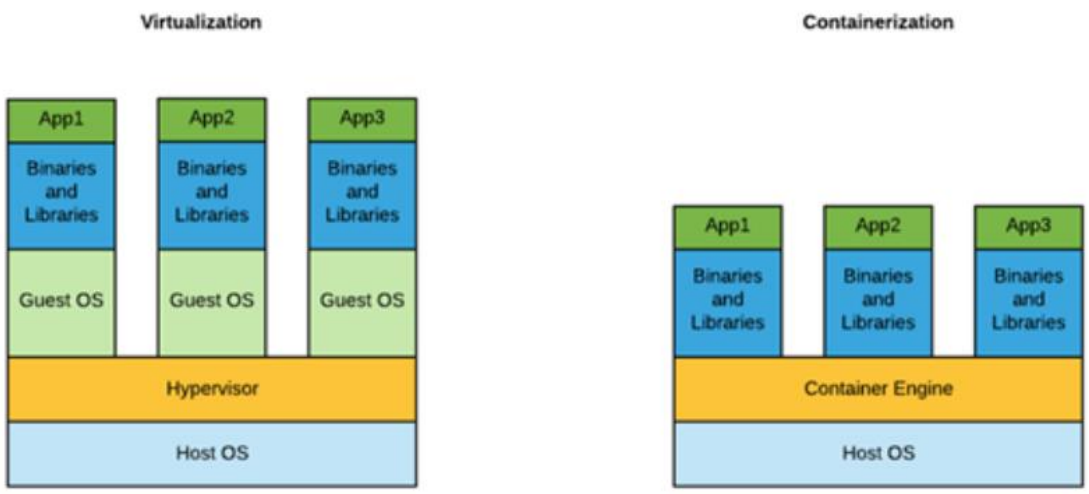
Containers versus Virtual Machines (VM):

Every Virtual Machine has its own Operating System which is the reason why the boot up process takes a longer time. Virtual Machines share the host's hardware with other VMs on the same host.

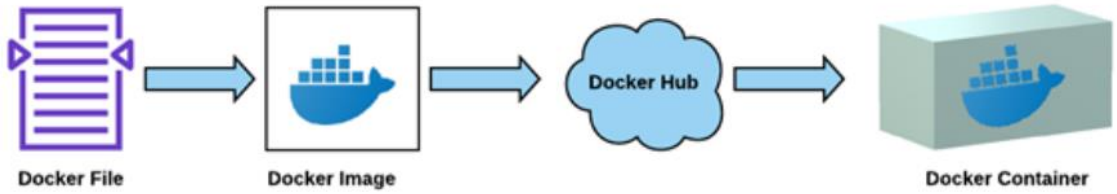
Containers, virtualize the Operating System – every container has its own CPU, memory, block I/O, network stack and uses the host's Operating System.

Containers have a short boot up process. They offer increased efficiency, better utilization and are portable.

Containers versus Virtual Machines (VM):



Docker Architecture:



Docker File:

A Docker file is a text document that contains all the commands a user could call on the command line to assemble an image. Every time, we are going to pick up a base image and build on top of that image.

For example, in the below Docker File, we are taking the base image “tomcat” and adding our web application war file.

From tomcat

ADD LeaveManagementApp.war /usr/local/tomcat/webapps

CMD “catalina.sh” “run”

EXPOSE 8080

Docker Image:

Docker Image is built from the Docker File. Docker Images are made up of multiple layers which are a read only file system A layer is created for each instruction in the Docker File and placed on top of previous layer

docker build –t LeaveManagementImage:1.0

Using the Docker build command, a Docker Image can be created.

Docker Hub:

Once the Docker Image is build, it can be stored or shared through Docker Hub. Just like GitHub, we can create an account in Docker Hub, create public or private repositories and maintain the Docker Images. Using the command below, we can set the Docker Hub configuration to our image:

docker tag LeaveManagementImage:1.0 myrepo/LeaveManagementImage:1.0

And then finally push the image to Docker Hub:

docker push myrepo/LeaveManagementImage:1.0

Now that the image is available on Docker Hub, you’ll be able to run it anywhere. If you try to use it on a new machine that doesn’t have it yet, the Docker client will automatically try and download it from Docker Hub.

Docker Containers: Docker Containers are sort of encapsulated environments in which you run applications. A Container is defined by the image and only have access to resources that are defined in the image.

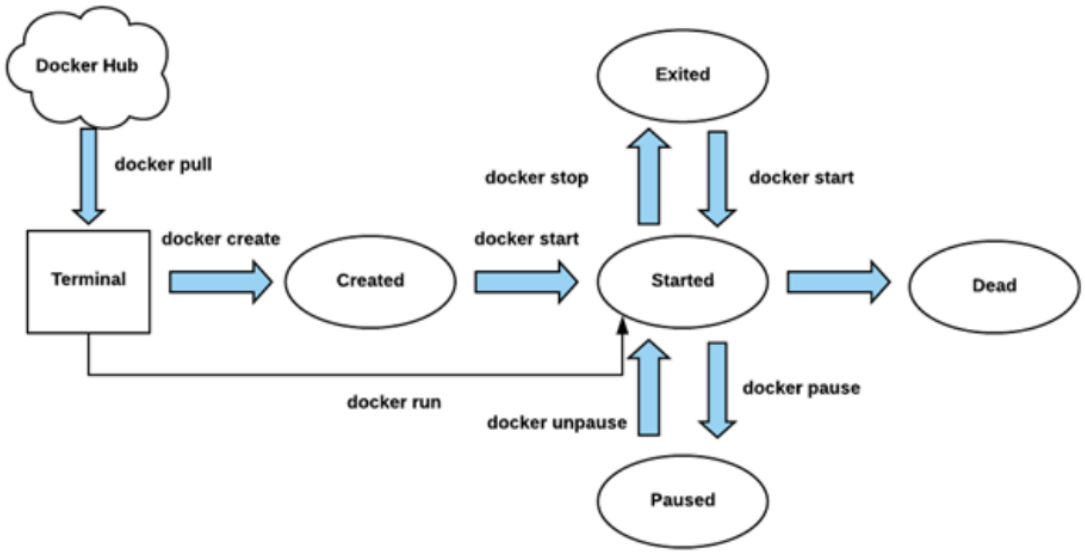
The machine where the container has to run should have a Docker client installed so that Docker commands can be executed. You can use the Docker pull command to get the image from Docker hub to local machine

docker pull LeaveManagementImage

Use Docker run command to fetch the image as well as create a new container from that image

docker run -itd LeaveManagementImage

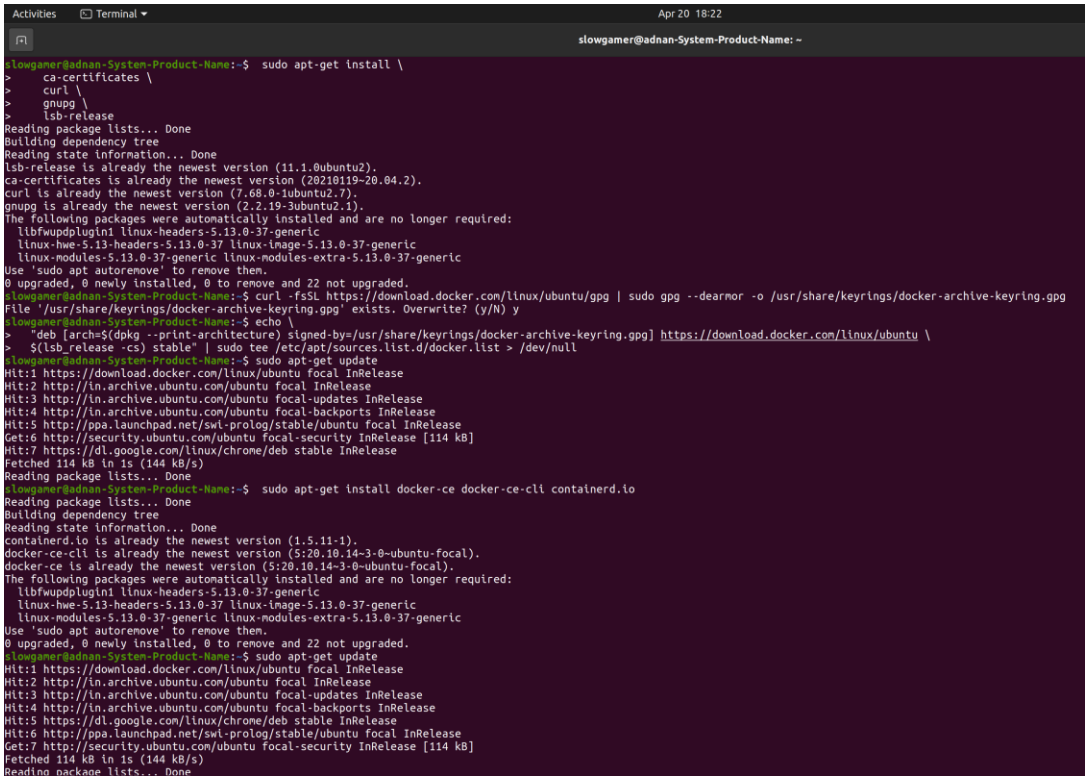
Once the container is created or has started using the run command, the container can be stopped, paused or started based on the requirement.

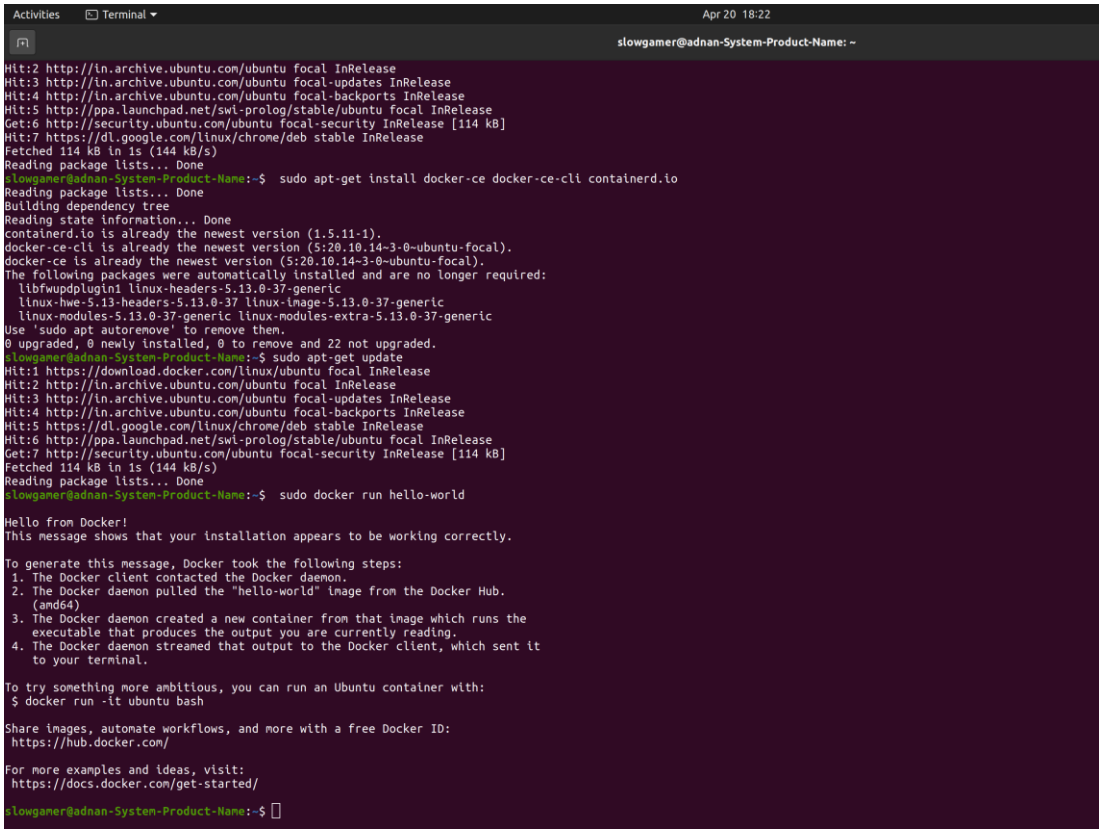


Another good use case of Docker is when you want to experiment with different database servers in your development environment. Instead of installing multiple database servers on your computer, simply use Docker containers to run each database server.

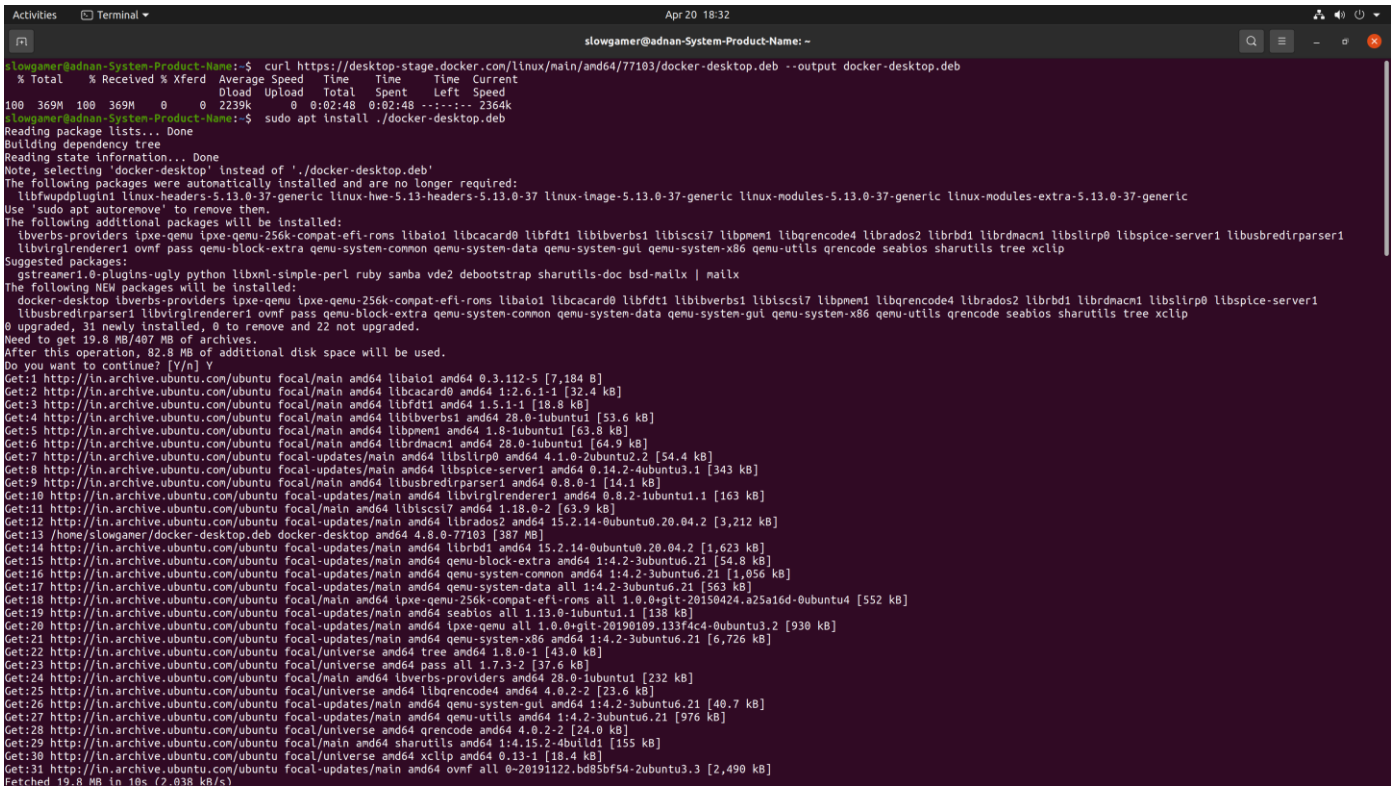
Output:

Installing Docker:

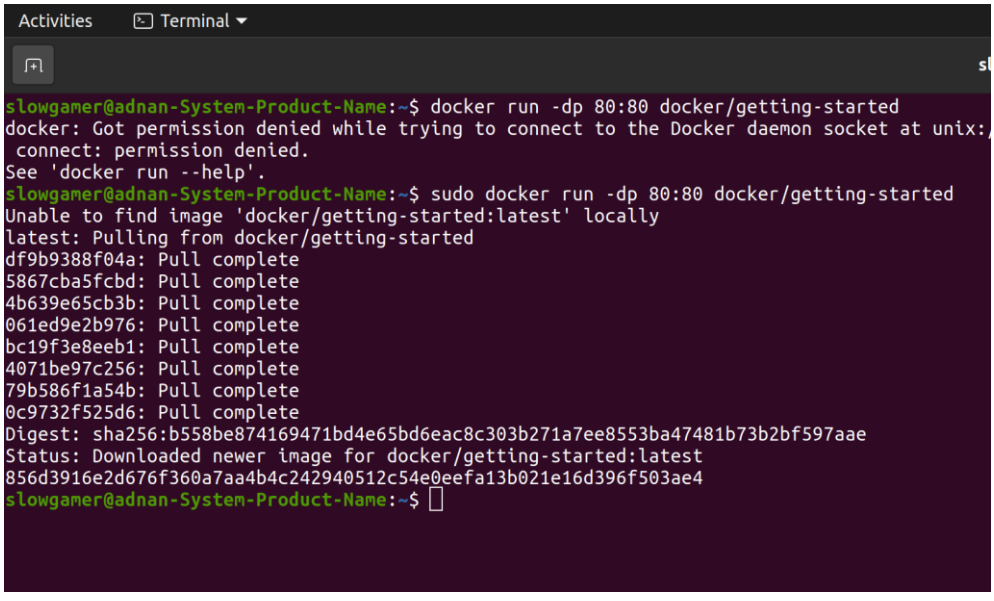




Installing Docker GUI:

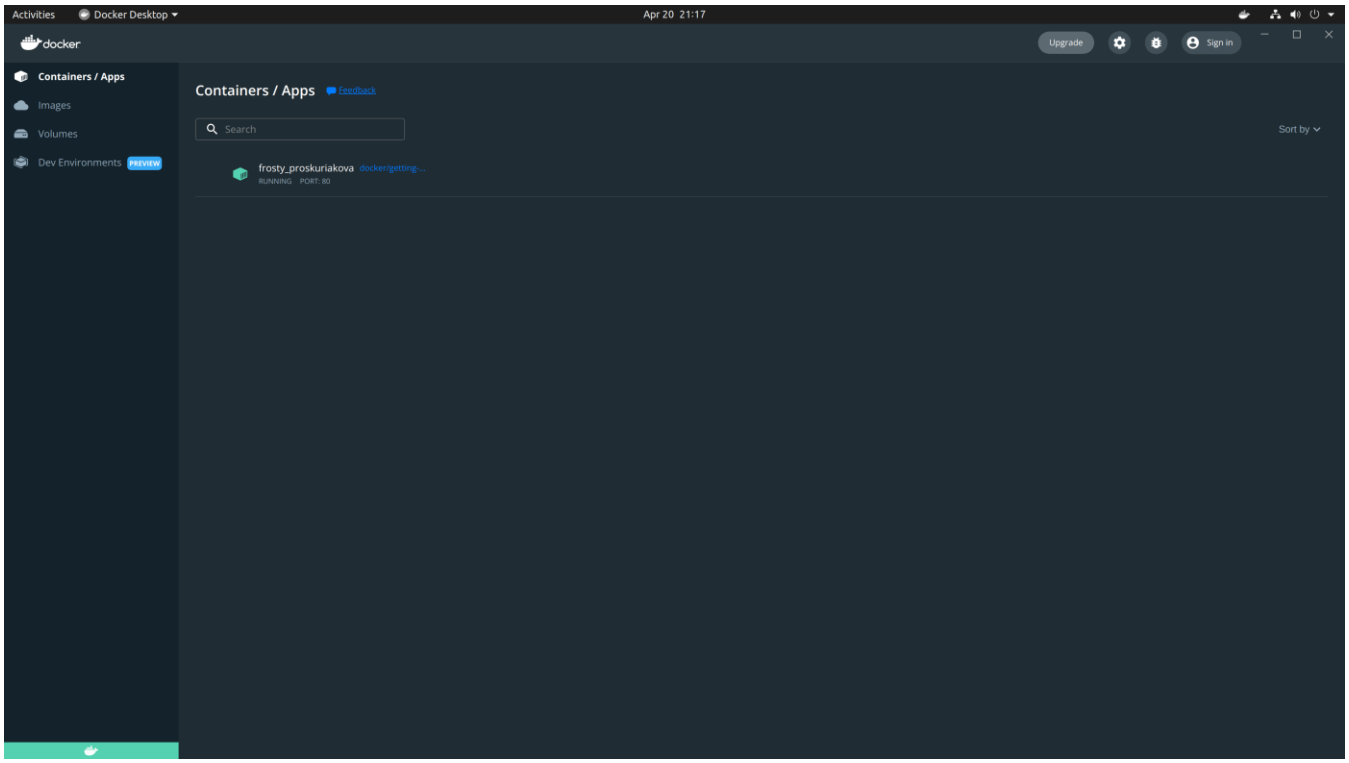


Running first container:

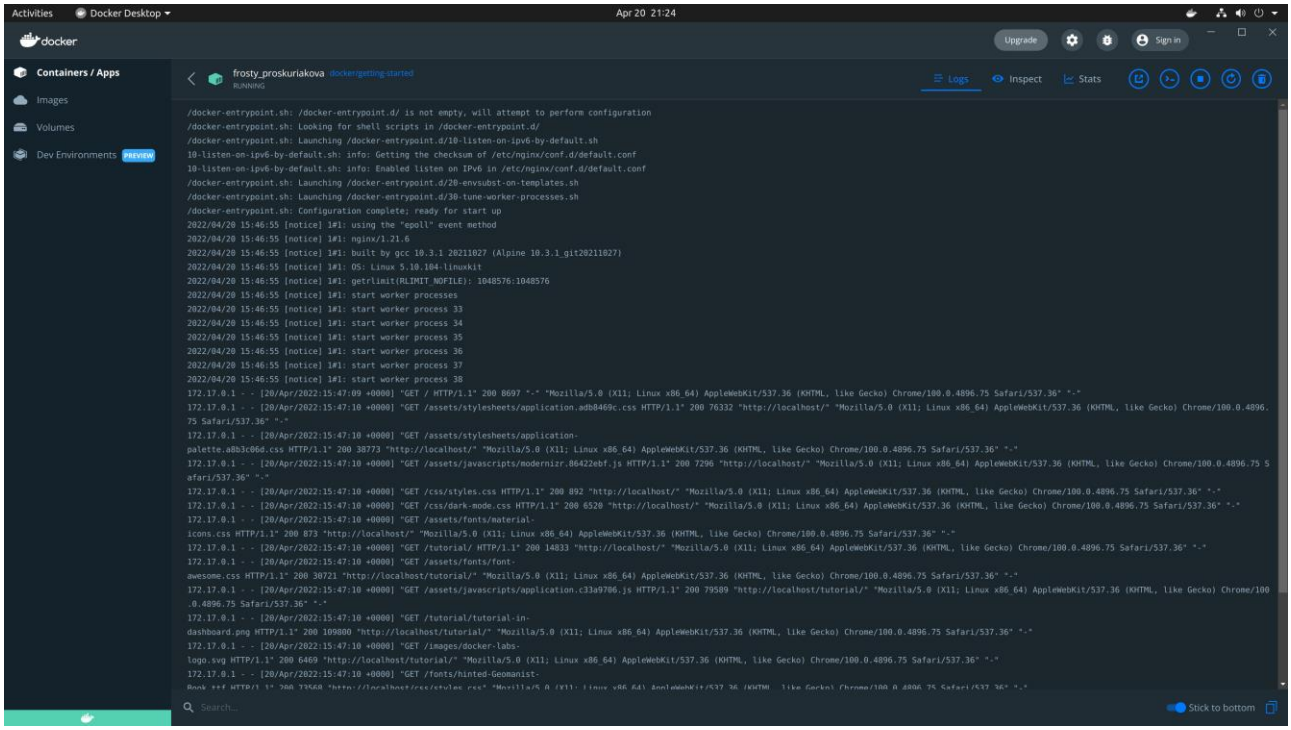


76_Adnan Shaikh

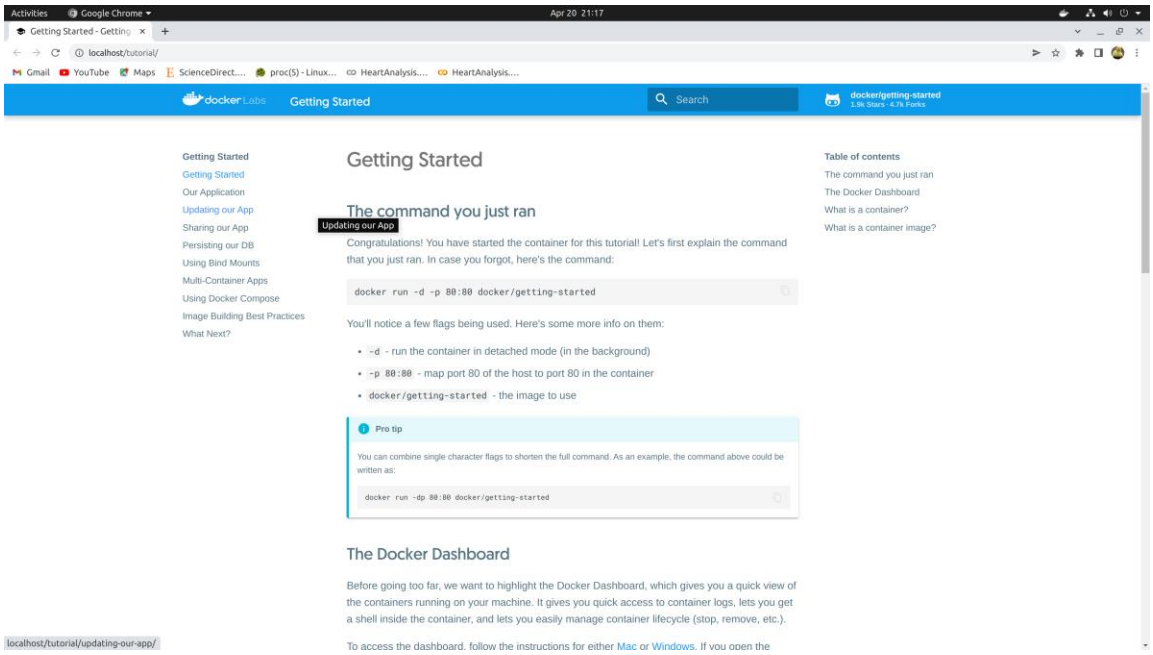
Container in Docker GUI:



Container log:

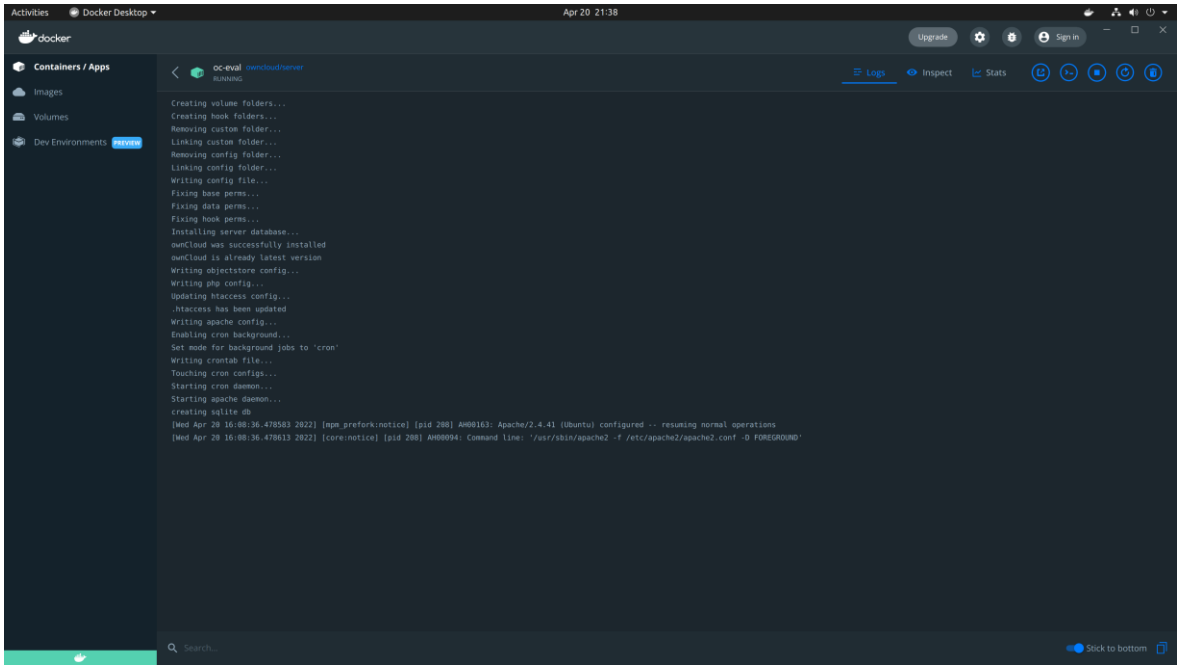


Container app running on port 80:



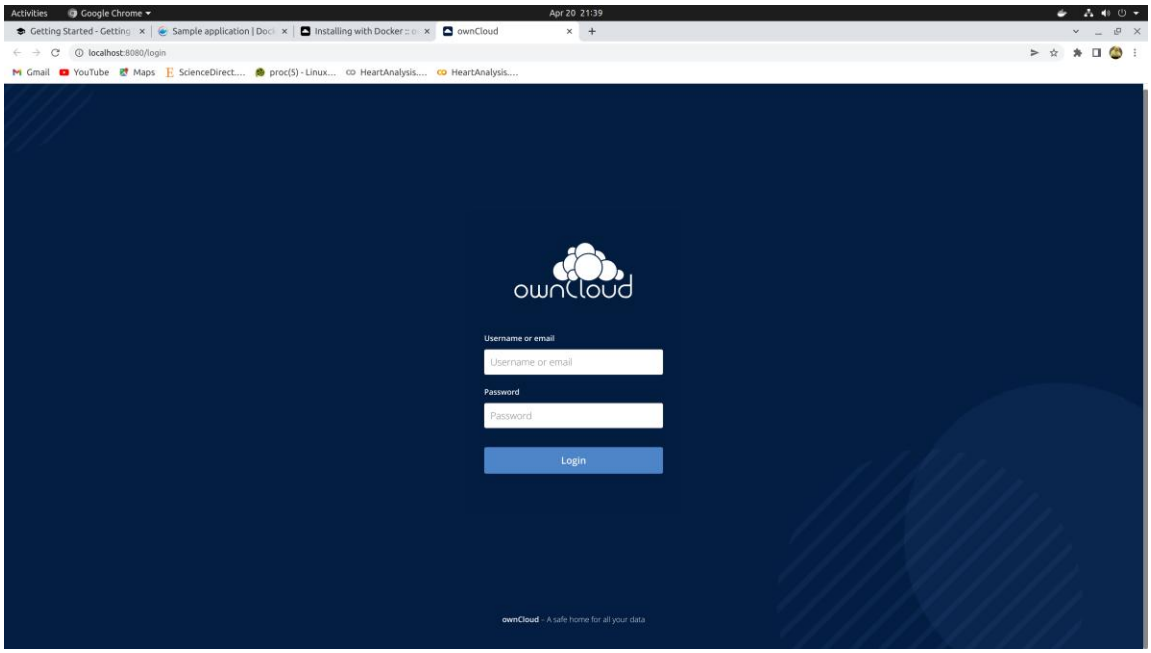
Installing and running container:

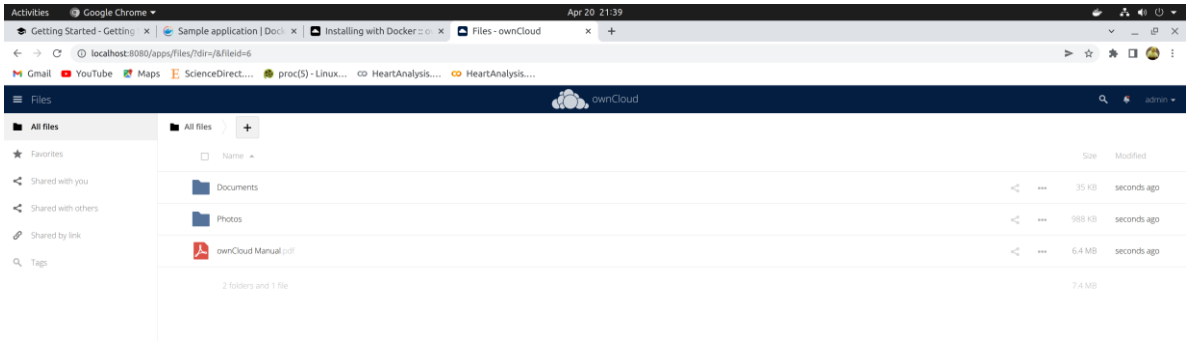
Container log in GUI:

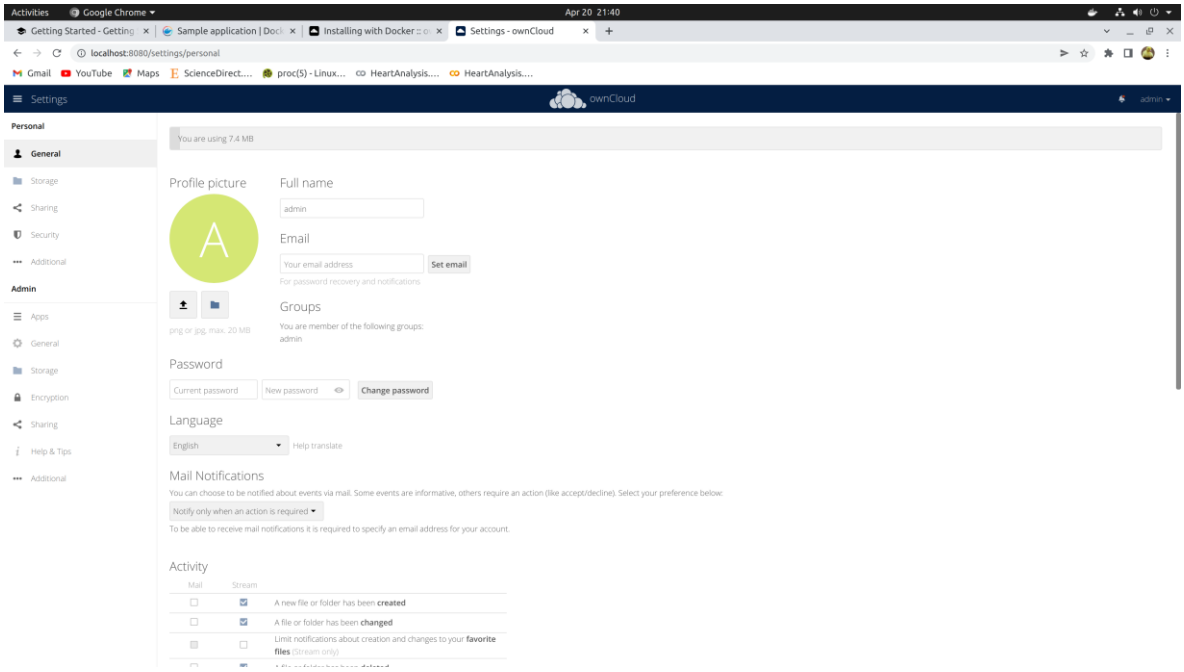


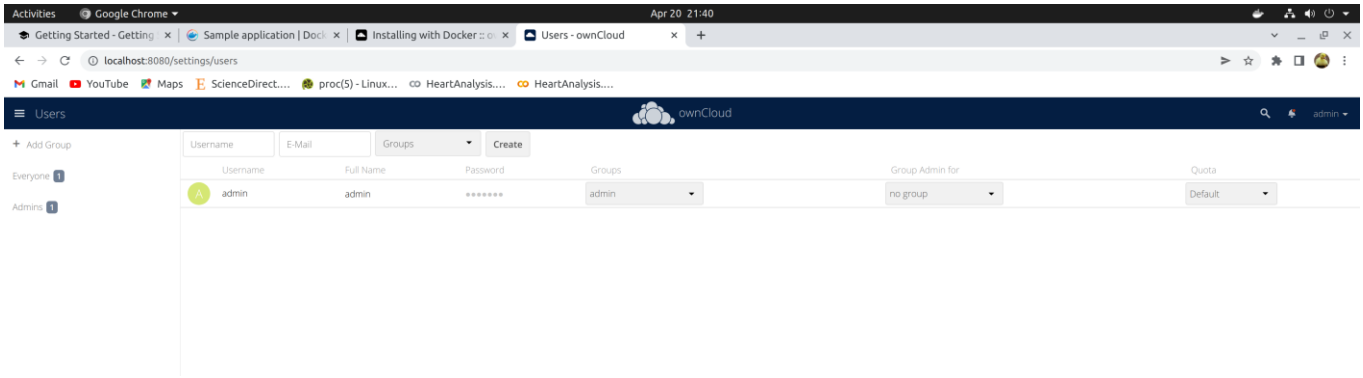
76_Adnan Shaikh

Container app running on port 8080:

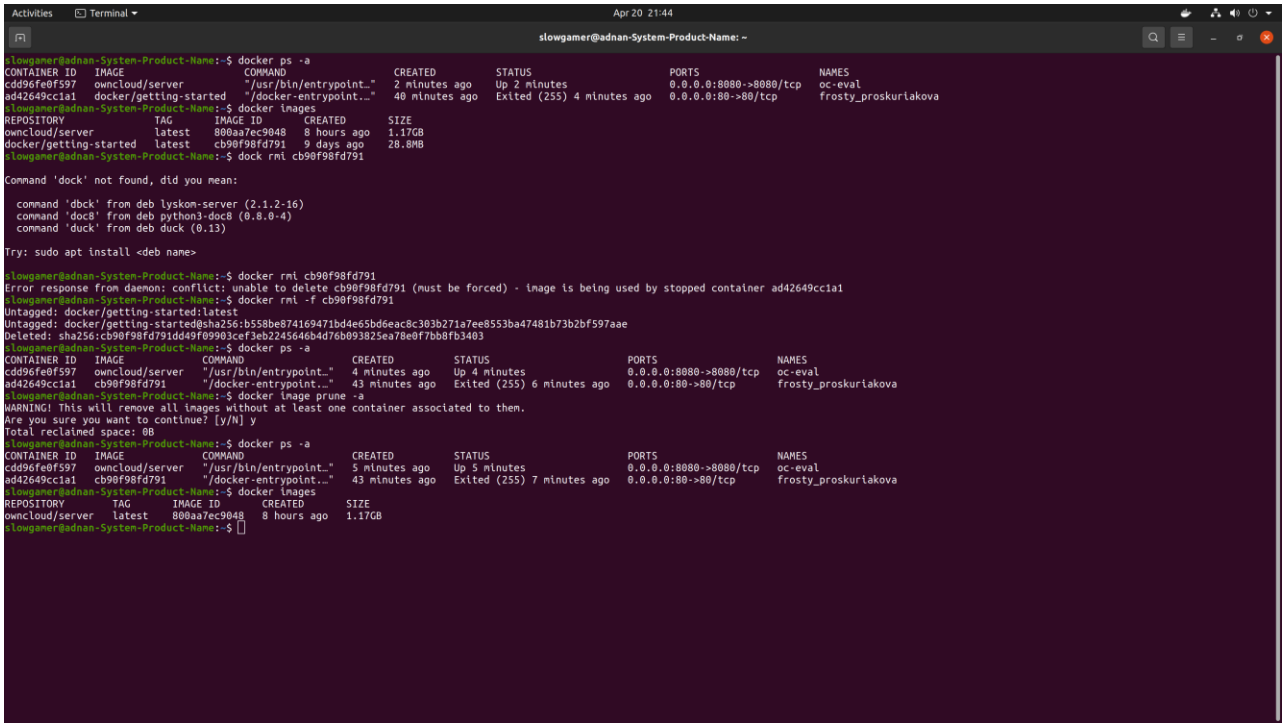








Some Docker commands to list container and to list and remove Images:



Conclusion: We have successfully implemented Containerization using Docker.