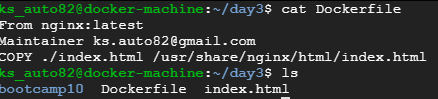
DevOps Bootcamp – Assignment 3

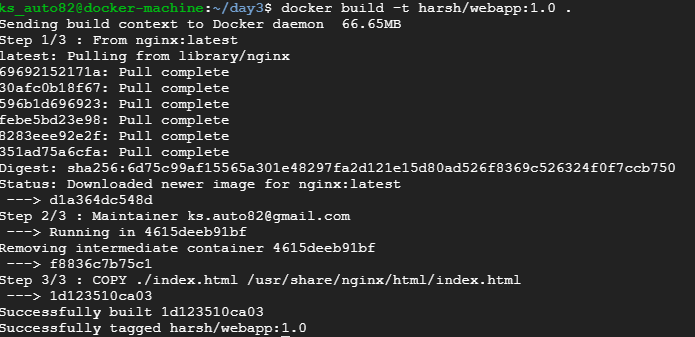
1. Write a docker file to create a docker image for this application using nginx as the base image.
   1. Installed Docker Image



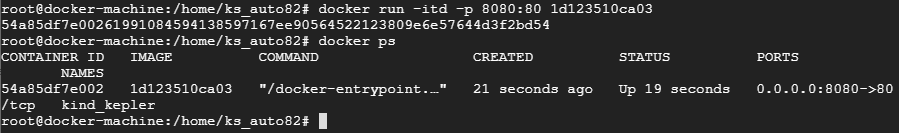
* 1. Creation of Docker file



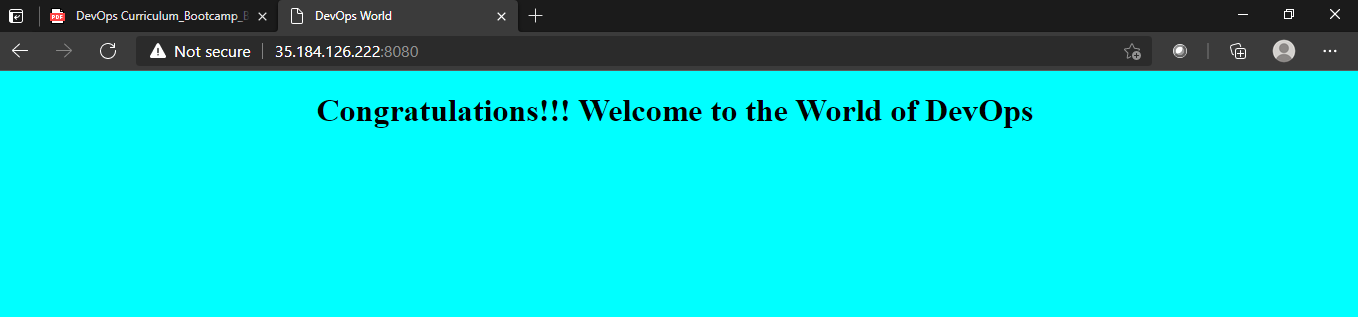
* 1. Building the Image



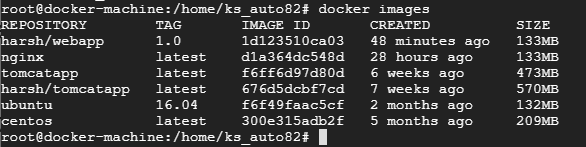
* 1. Run the built docker image



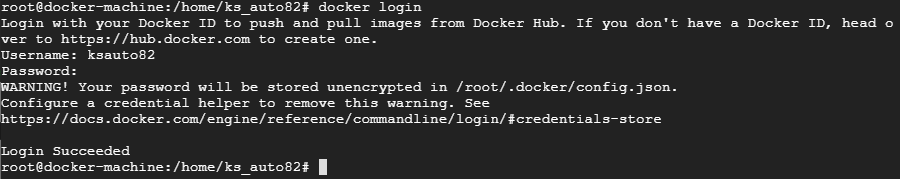
* 1. Output



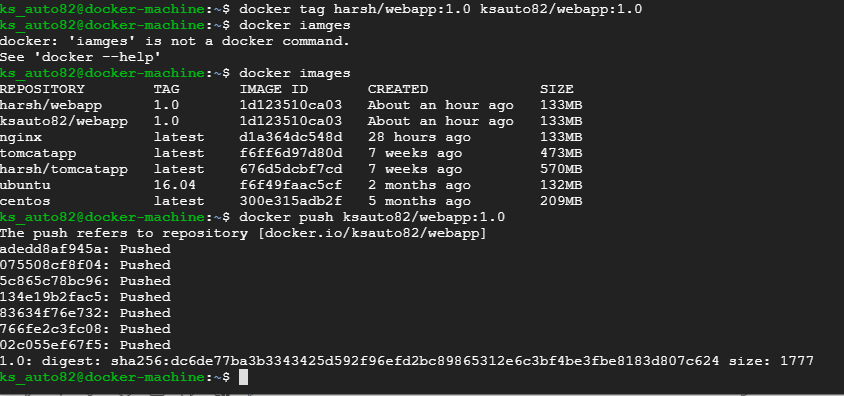
1. Push the docker image to the Docker Hub.
   1. List of images



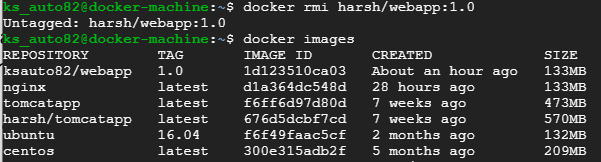
* 1. Login into the Docker

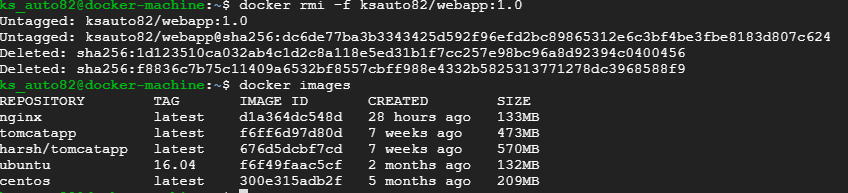


* 1. Push the image into the remote repository

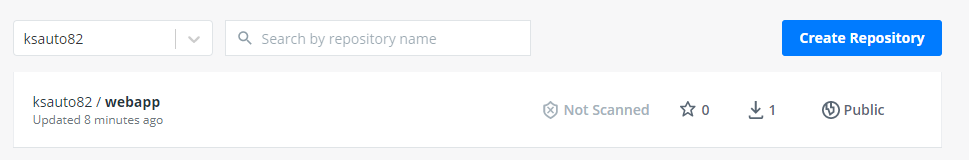


* 1. Delete the image from local

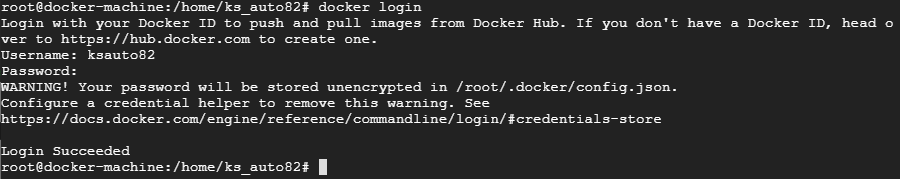




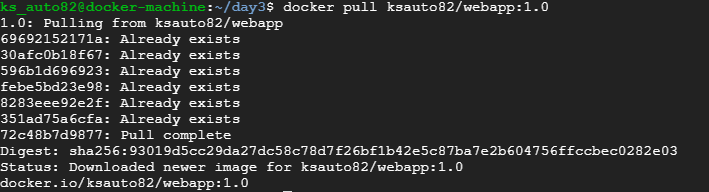
* + - From Docker Repository



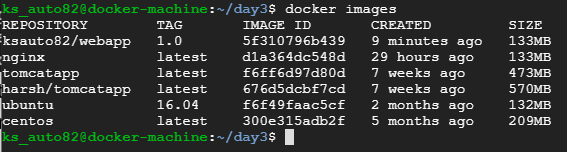
1. Pull and Run the docker containers on any remote instance.
   1. Login to Docker Hub



* 1. Pulling the image from Docker Hub

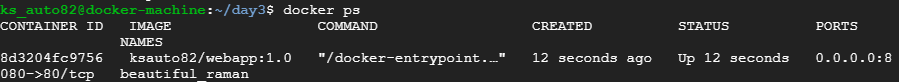


* 1. List the images

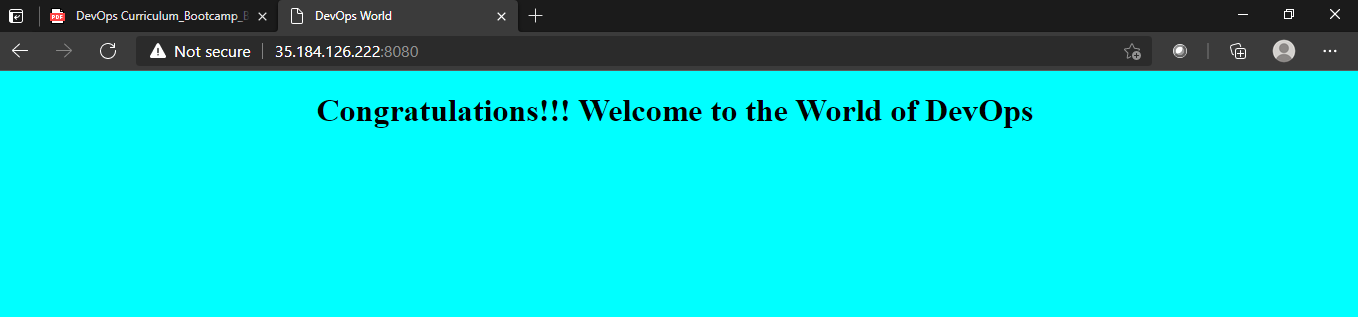


* 1. Running image as container

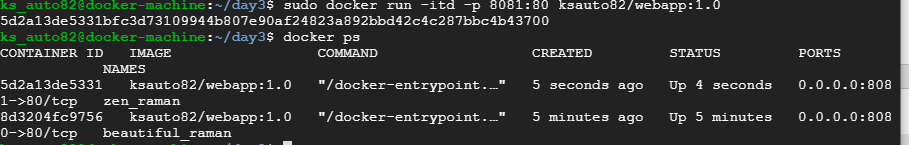




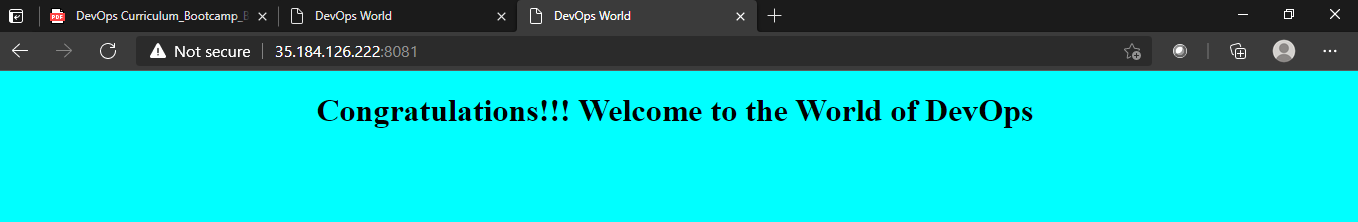
* 1. Display the running containers



1. Access the application from the browser using appropriate url and port.
   1. Run the Image with port 8081

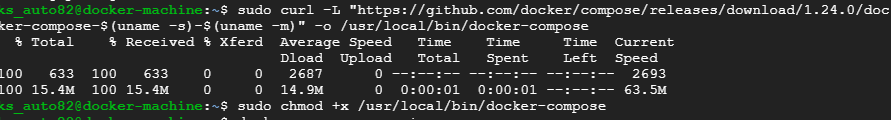


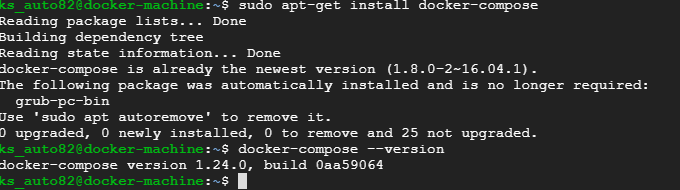
* 1. Docker container is running and able to see the webpage.



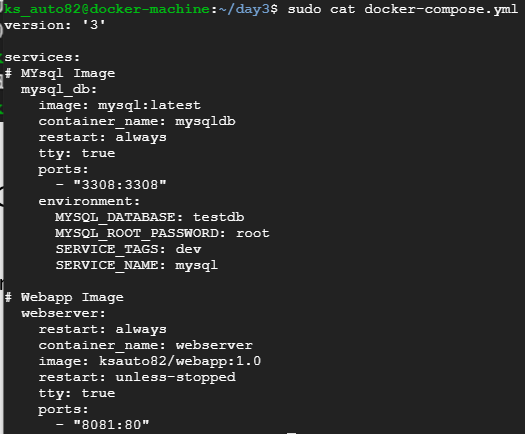
Docker-Compose:

1. Write a docker-compose.yml file to run the above application container along with a mysql database container together.
   1. Install docker-compose

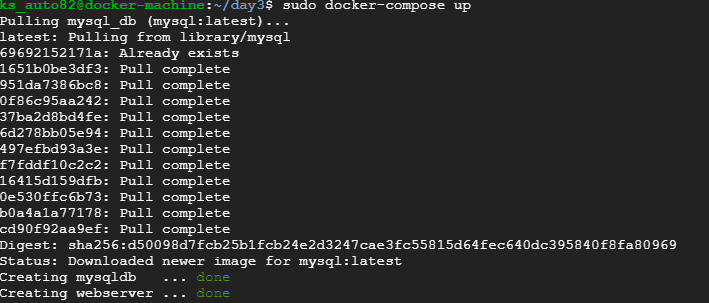


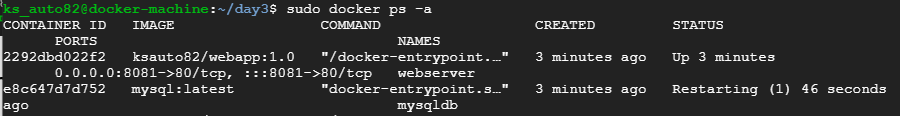


* 1. Create a docker-compact.yml file

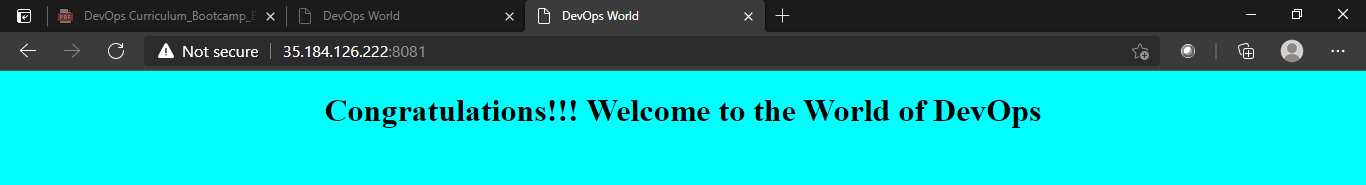


* 1. Execution of Docker compose file

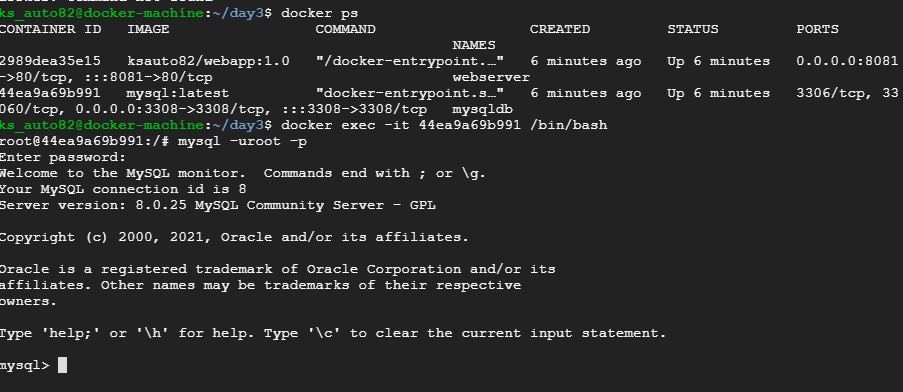




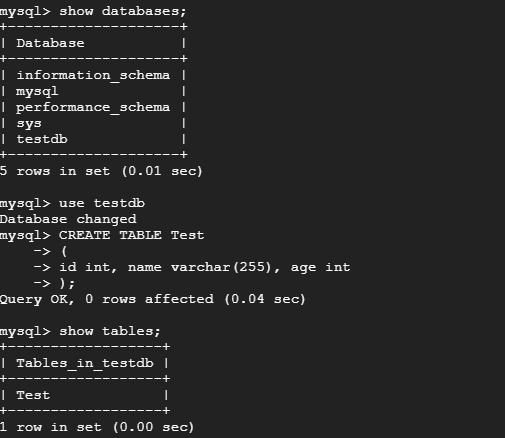
* + - Changed the port of web server to 8081 and created from compose file to access from browser



1. Connect to the mysql database container in the -it mode and create a table named test having id, name and age as attributes.
   1. Connect my sql



* 1. Creating table as per requirement



* 1. Insert and select the data from Test in mysql

