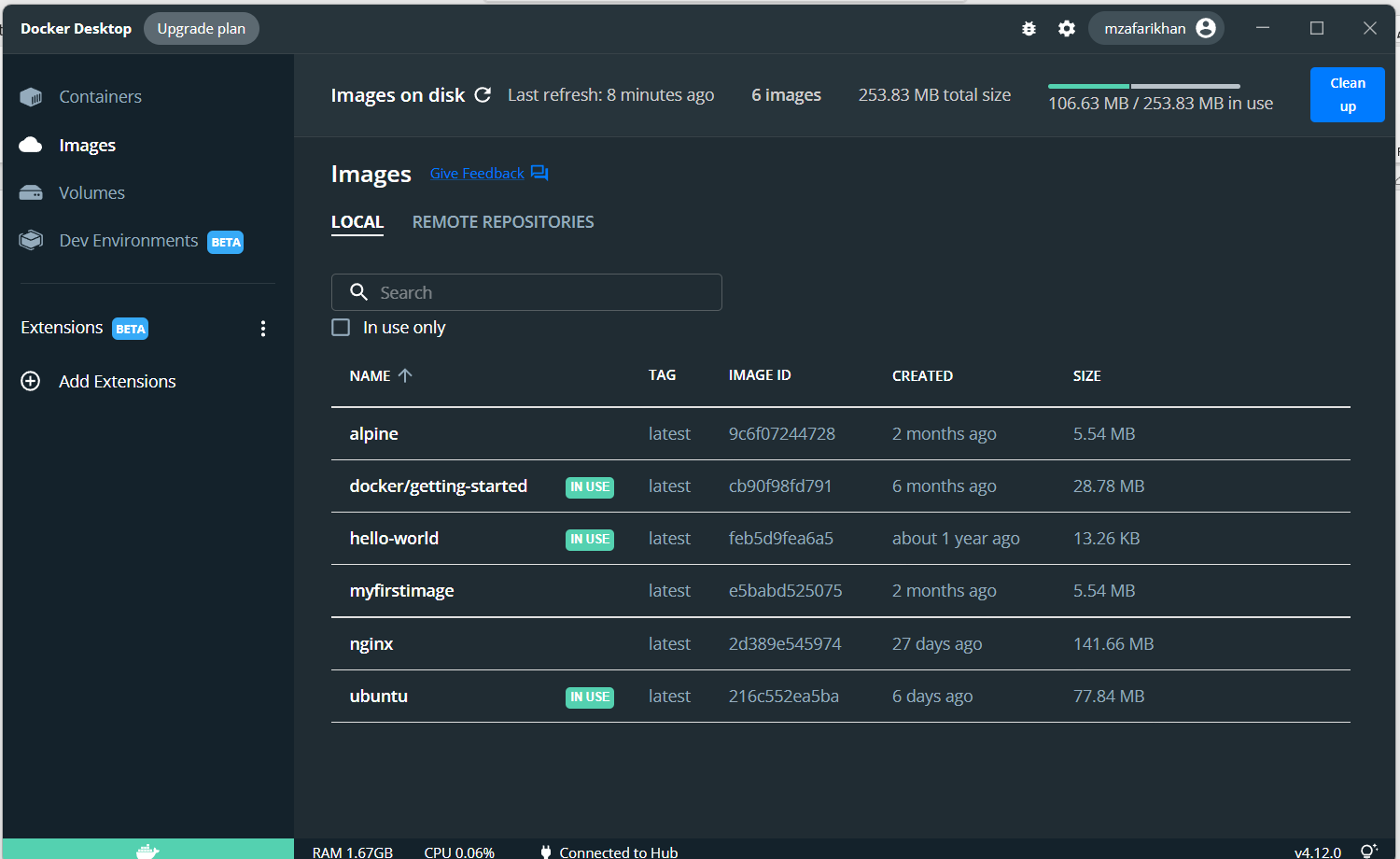
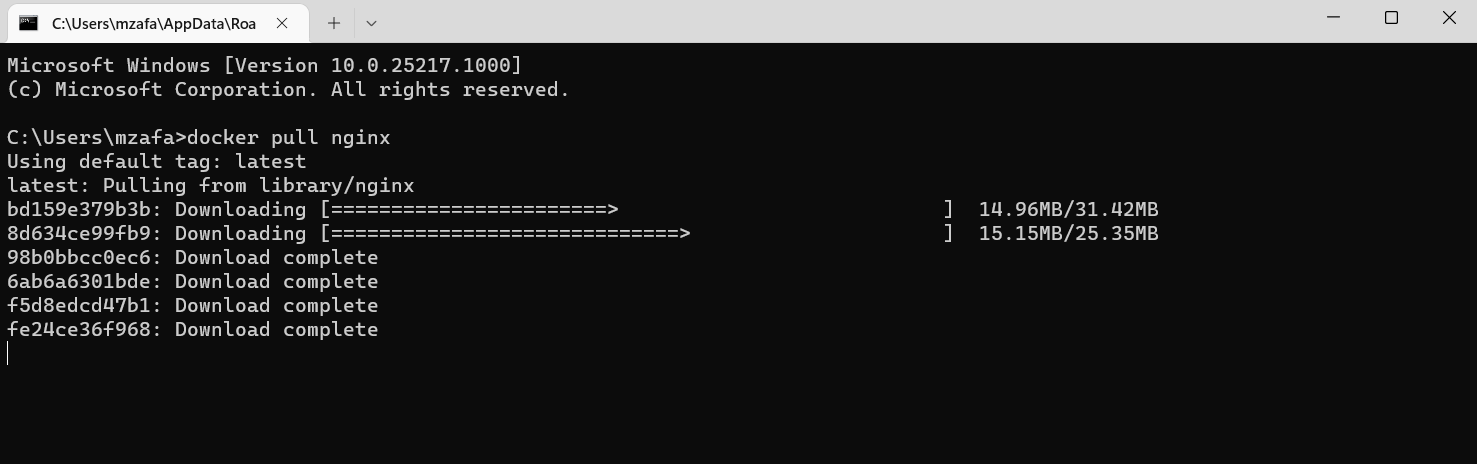
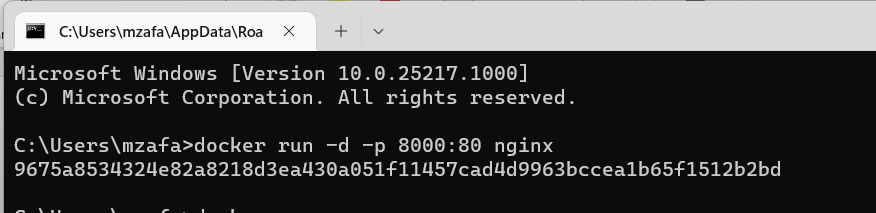
SE489 DevOps Engineering

Lab 7

# Lab 7: Docker Commands and custom images

Objectives: In this lab, students will learn how to use various Docker commands, customize Docker images, build custom images and deploy it. Student will learn famous lightweight webserver Nginx and will learn famous Unix editor Vi/Vim.

1. Start Docker, double click on Docker icon to run the docker, on the docker dashboard, all the images you have downloaded to your local system will be listed. Right side shows username, below username, size of docker images collectively is shown against the total size. Bottom shows the status of connection to the Docker Hub.  
   
2. Let’s pull nginx\*\*\* server from the docker hub and run it.  
     
   it should complete with in few minutes based upon your internet speed  
   Text

   Description automatically generated
3. Now start the nginx, with following command  
   **docker run -d -p 8000:80 nginx**  
   

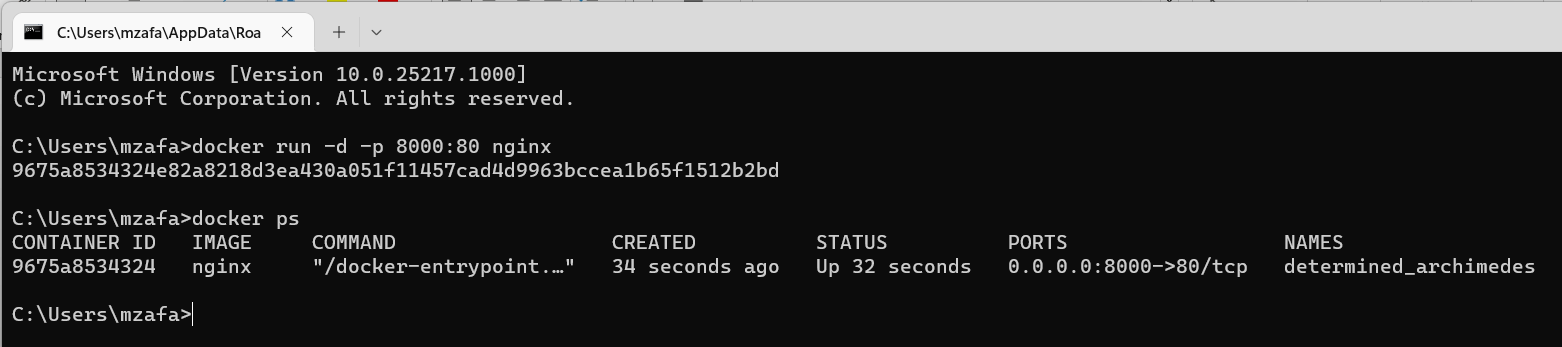
-p 8000:80 is used to map the tcp port 80 to the Nginx port of 8000

-d is used to detach the command window from the nginx daemon process

The long hash in the next line is the process id attached to the nginx container by the docker.

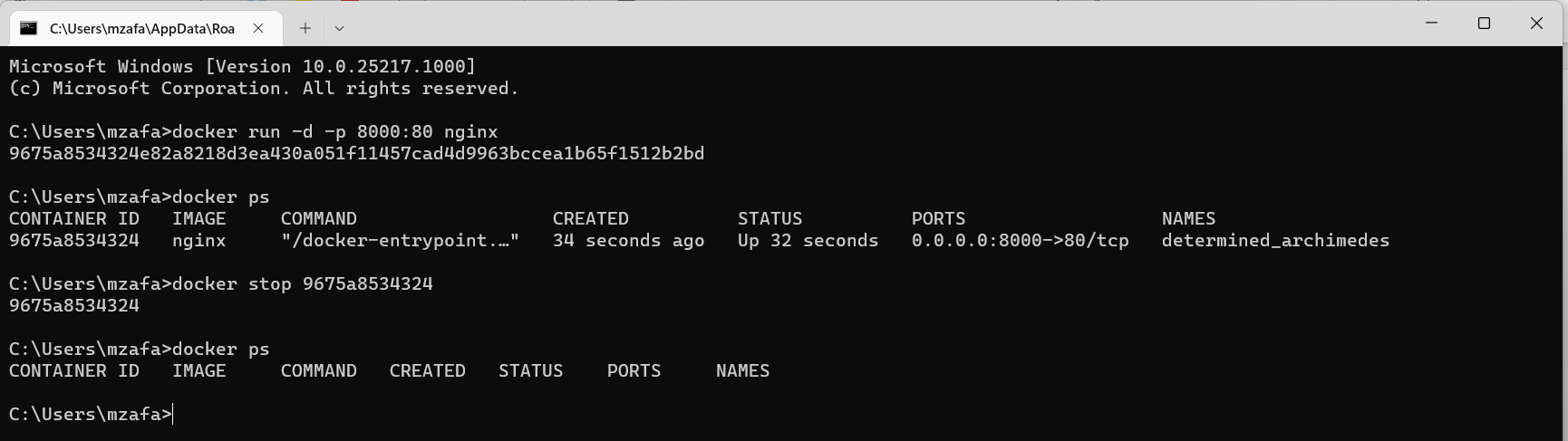
Verify these parameters by issuing following command

**docker ps**

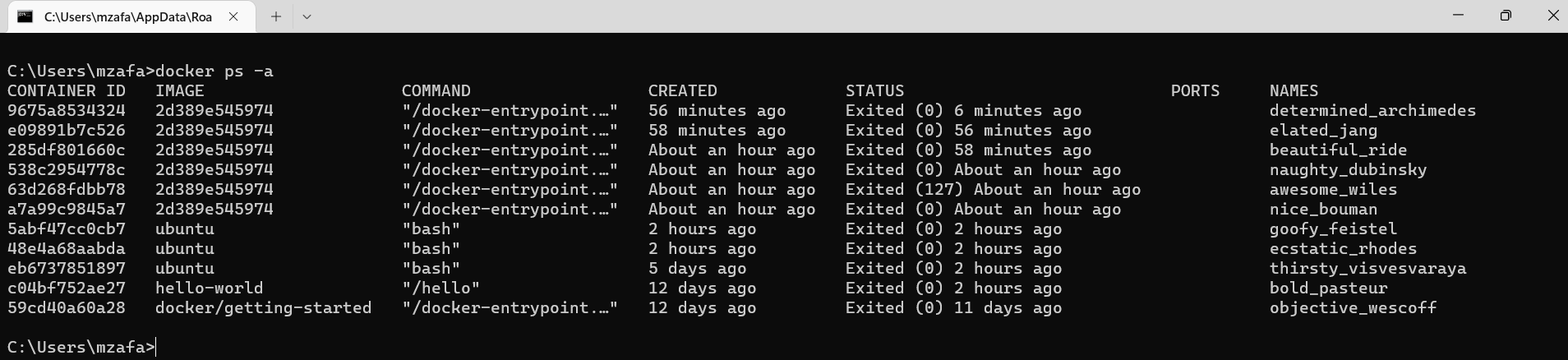
****

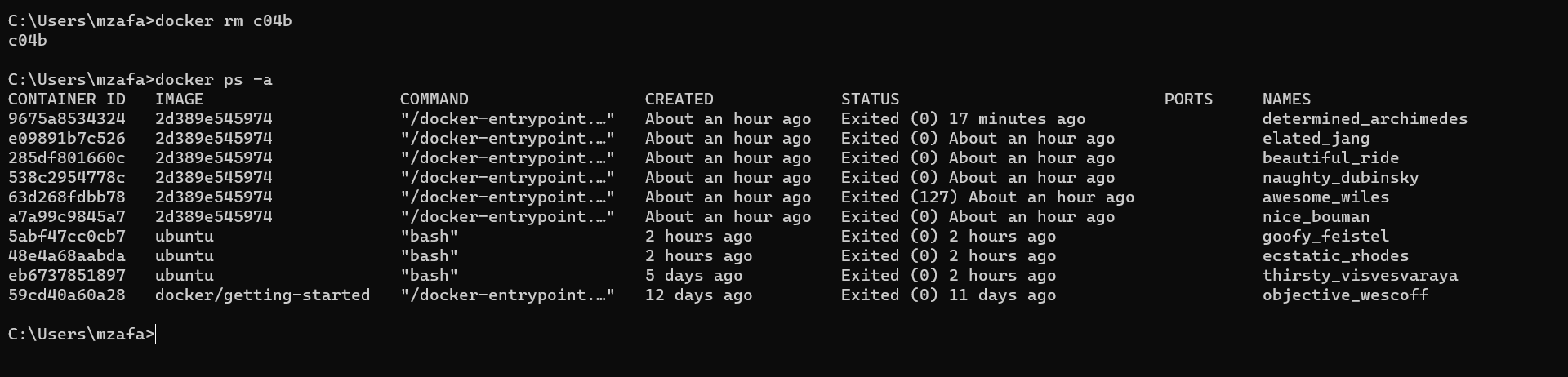
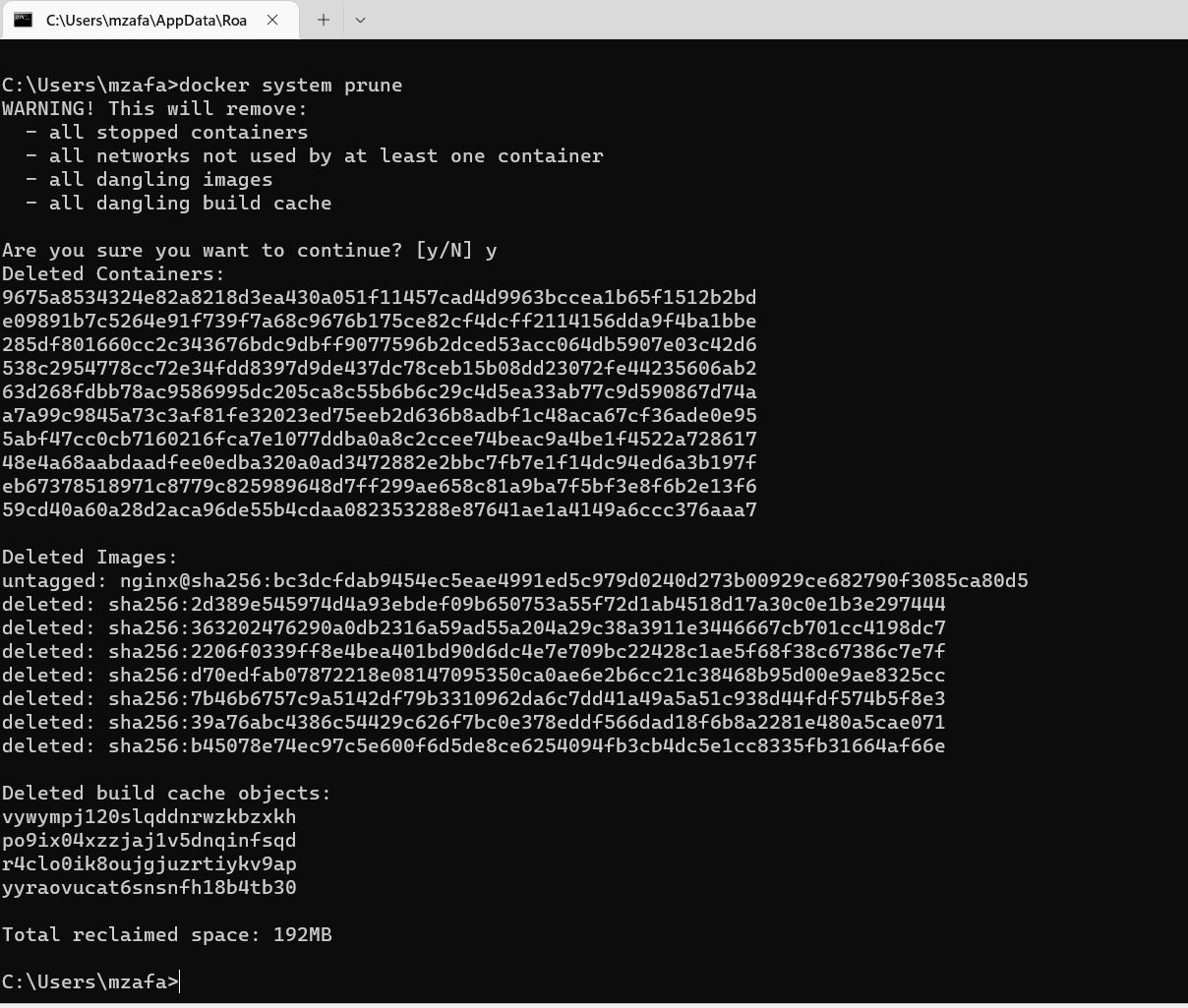
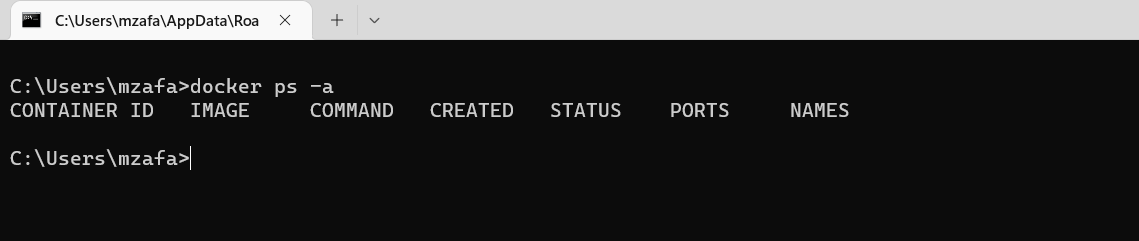
1. Open any web browser of your choice, in the address bar write 127.0.0.1:8000, it will now show homepage of the Nginx server  
   Graphical user interface, application

   Description automatically generated
2. To stop this container, run following command from the command prompt, and then again run the ps command to show the running containers  
   **docker stop <container id>**

**docker ps  
  
  
\*it is not necessary to write complete container ID, only few characters which may identify container uniquely will suffice.**

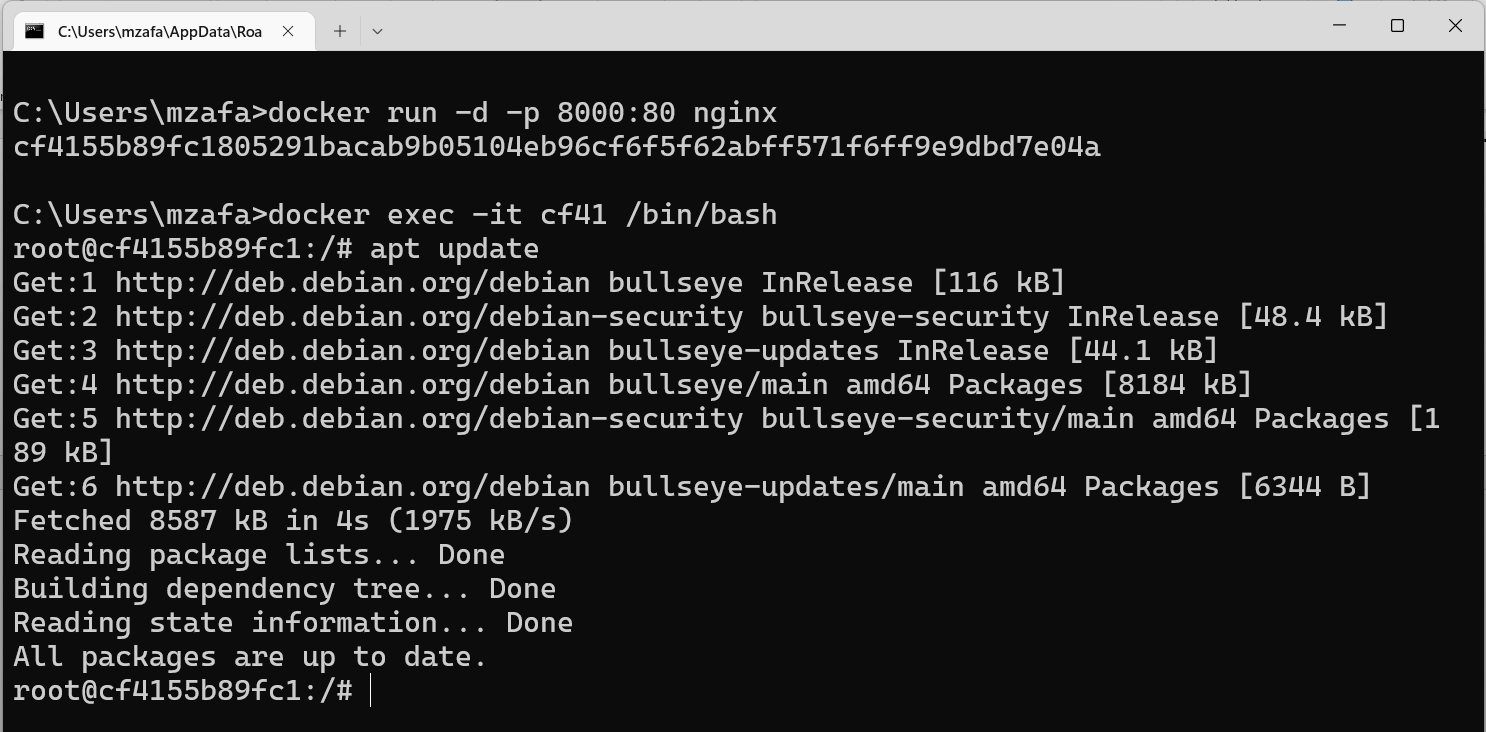
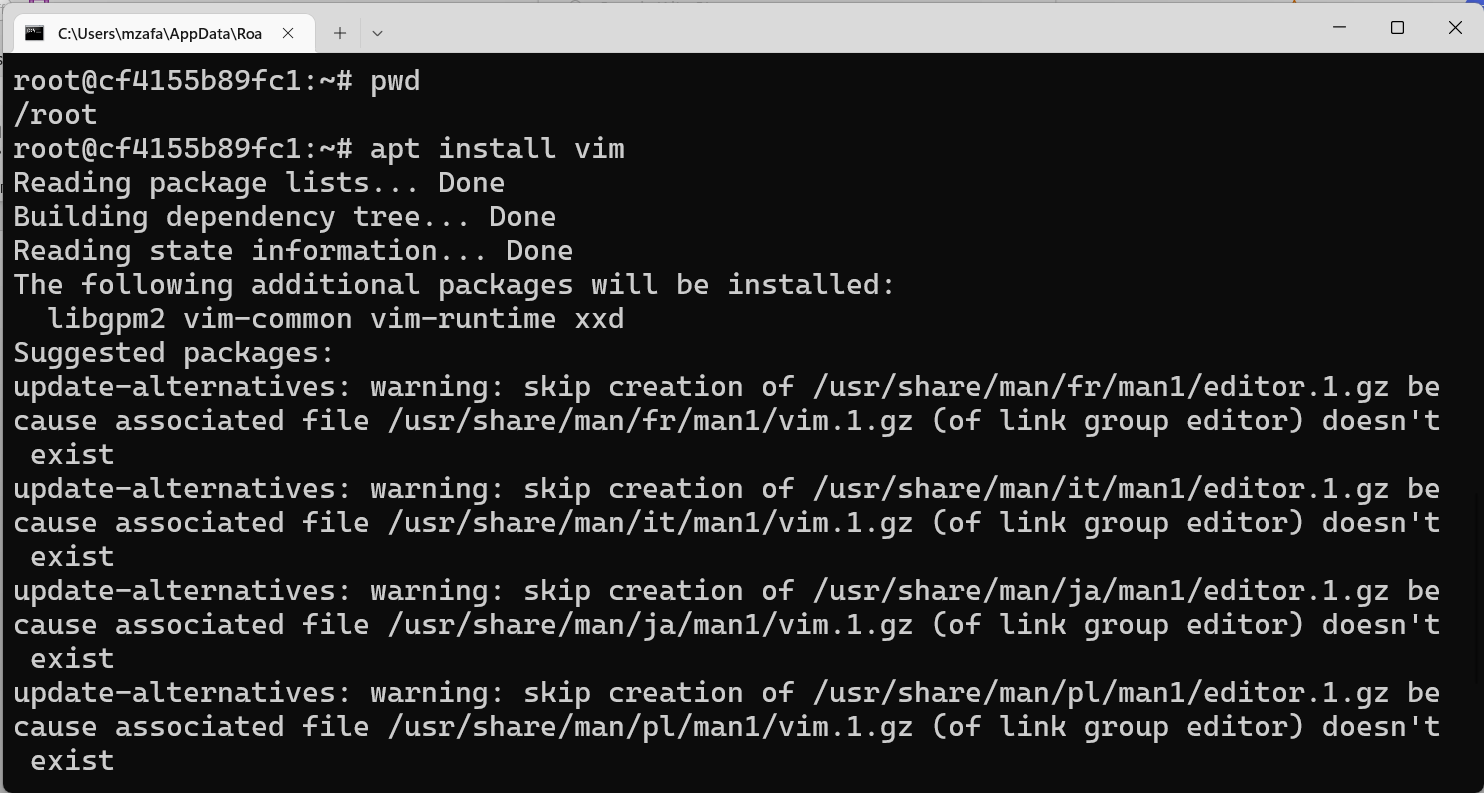
1. To list all the containers created in the system, use -a switch with docker ps command

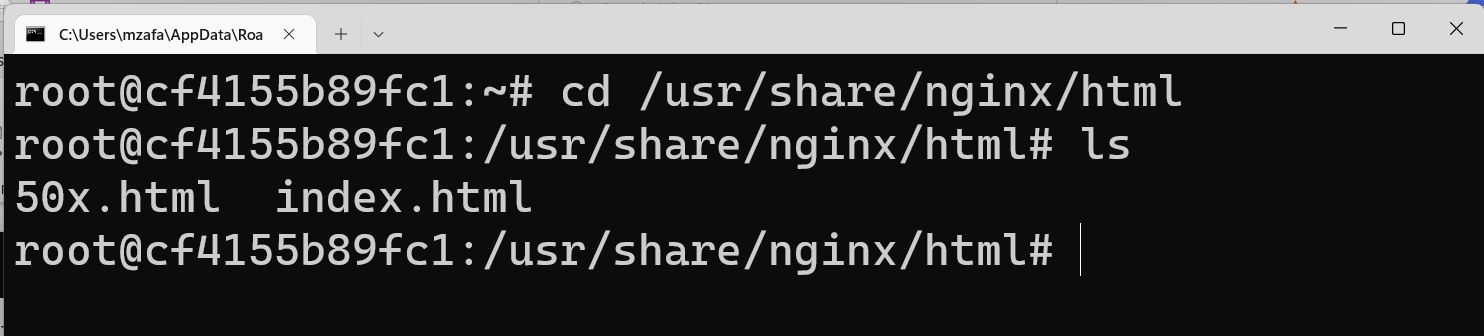
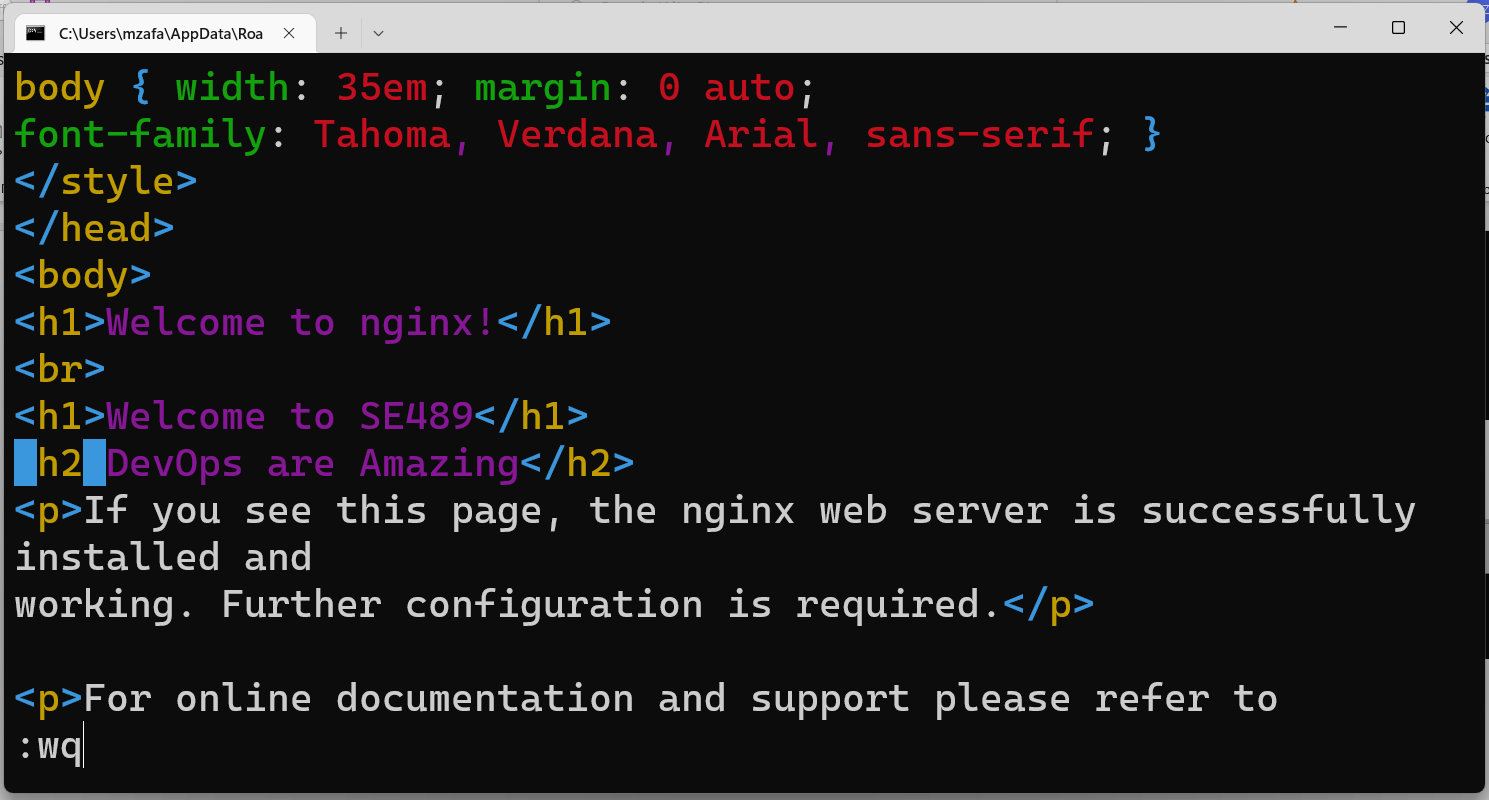
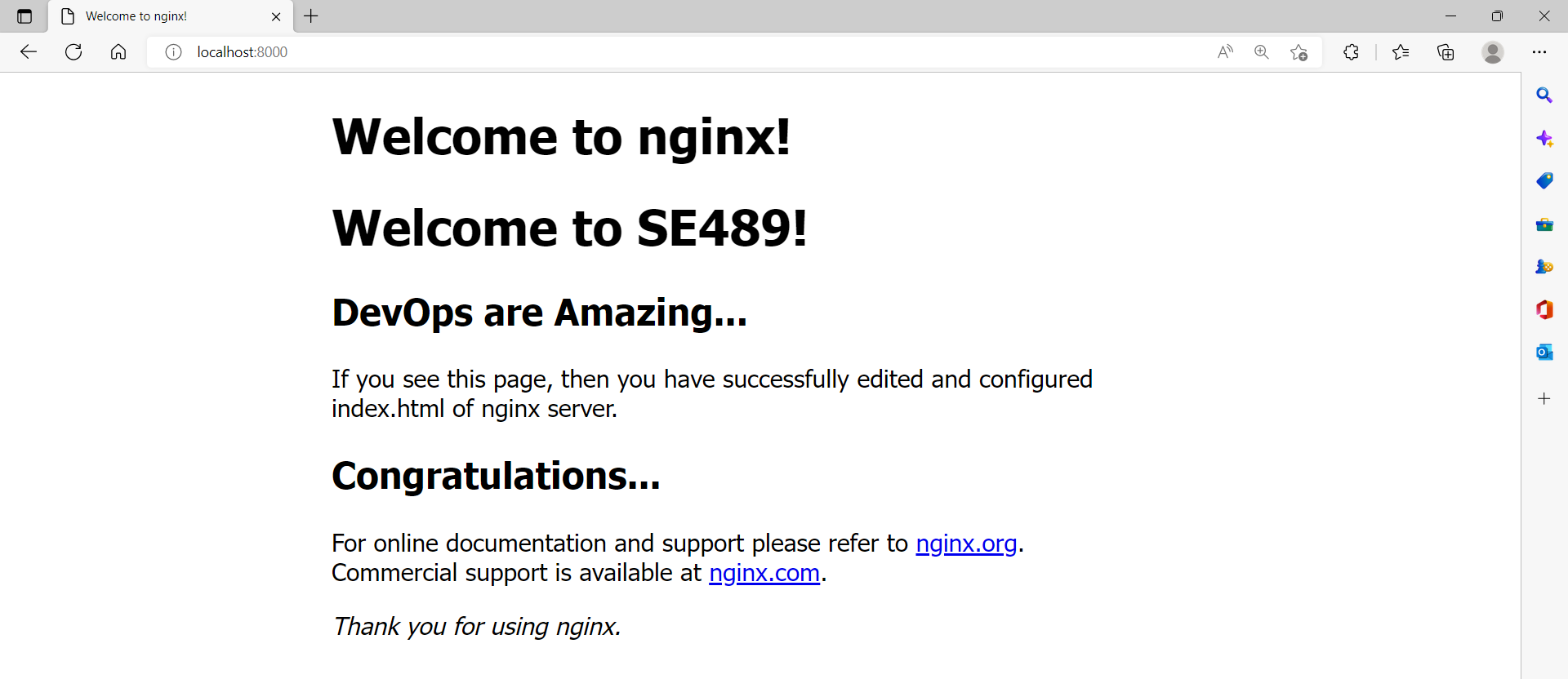
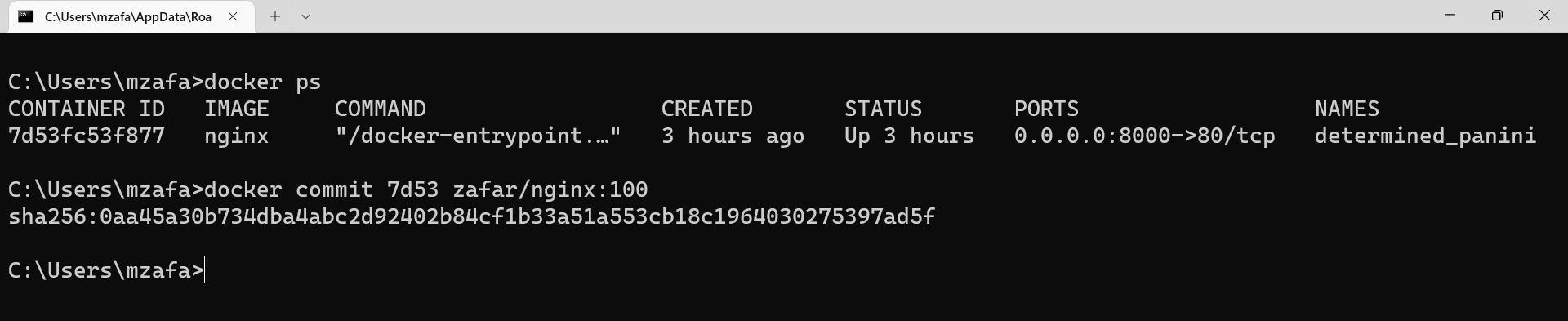
**docker ps -a   
**

1. To remove a container reference, run following command   
   **docker rm <container id>**and then again list all the containers with  
   **docker ps**  
   e.g. let’s remove the *hello-world* container, starting with c04b  
   
2. To remove all these containers, use **docker system prune** command  
     
   Now check again with **docker ps -a** command  
     
   
3. Now let’s again, start the nginx server with docker run command  
   Graphical user interface, text

   Description automatically generated
4. Let’s enter into the nginx shell, with following command  
   A black screen with white text

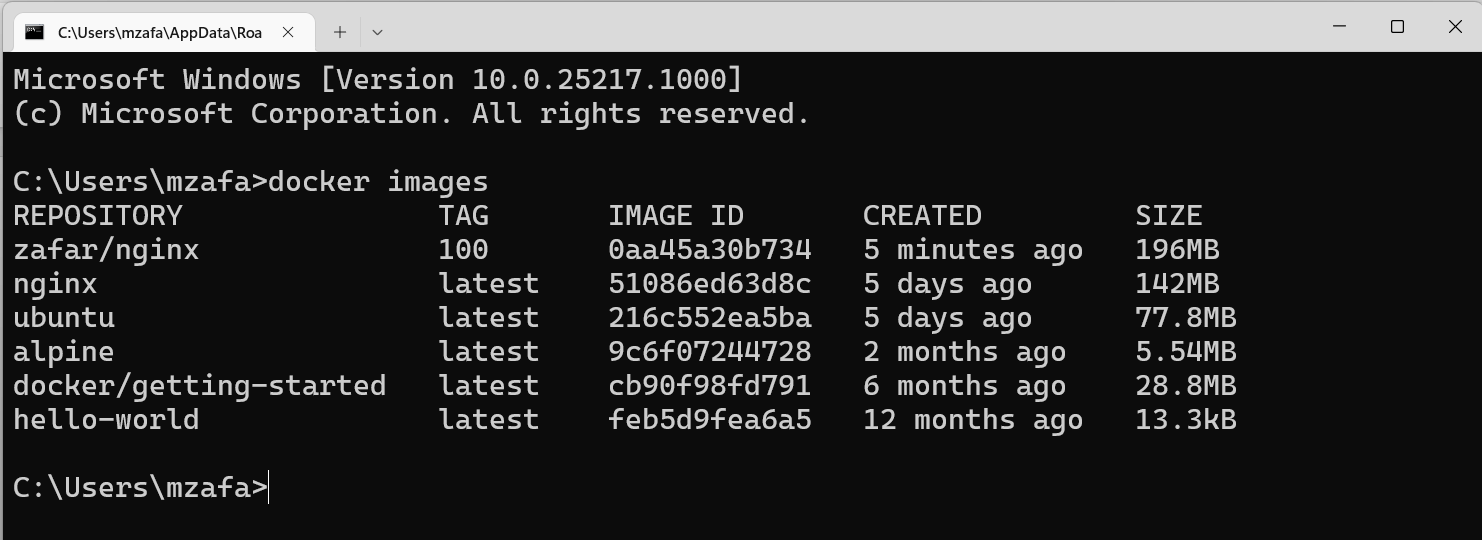
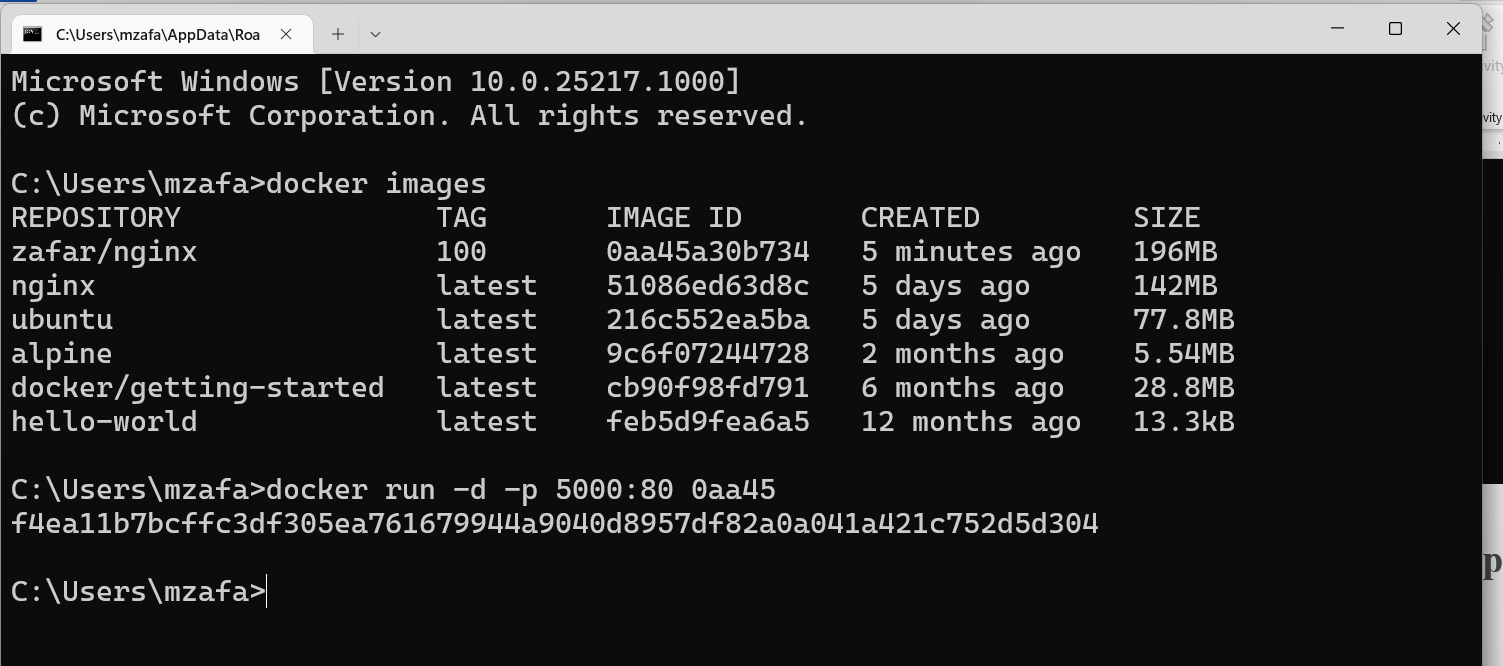
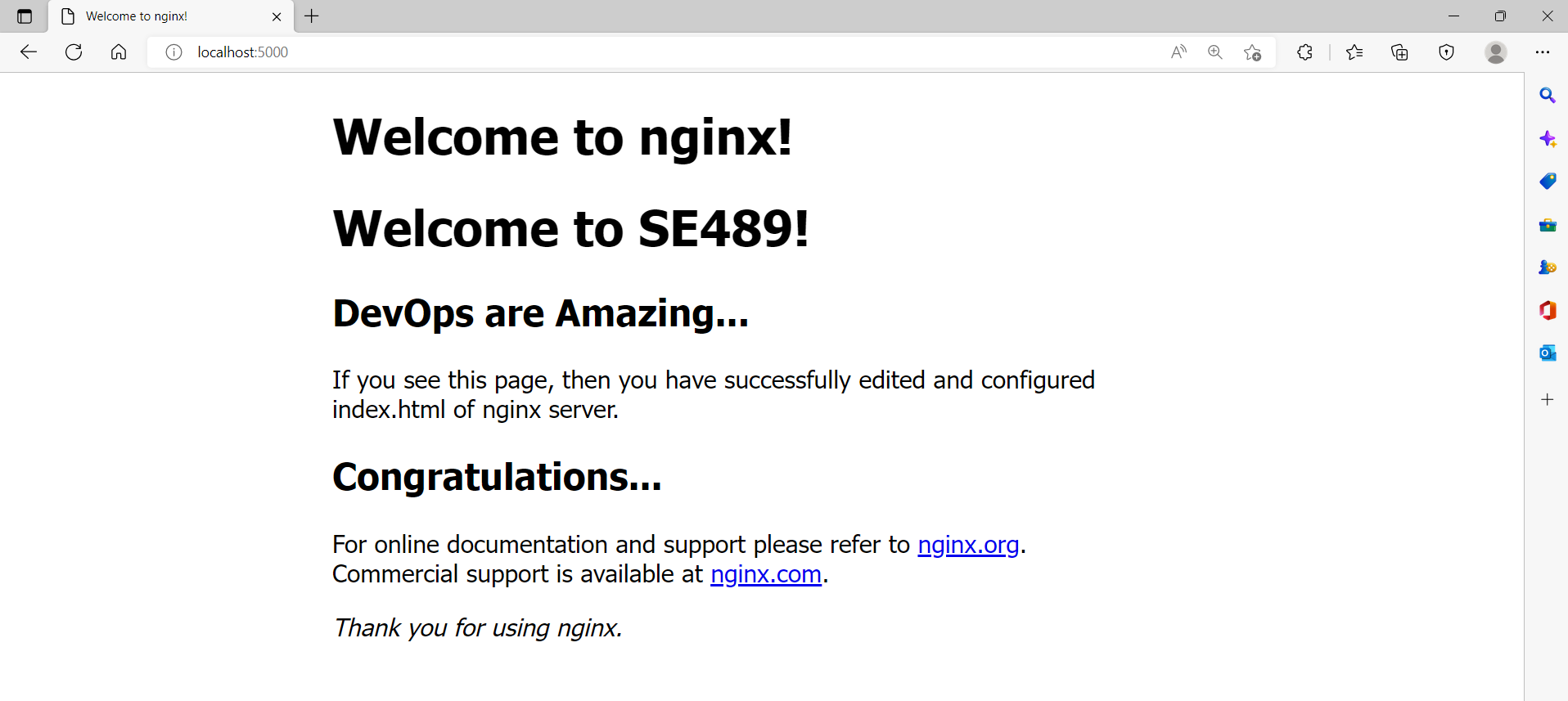
   Description automatically generated with medium confidence  
   clearly now we are in nginx shell, now lets make some changes into this
5. Run the apt update command to update the packages in the nginx  
   Text

   Description automatically generated  
   \*cf41 is the container id, copied from the command output just above the apt update  
   
6. Let’s install vim editor in this with apt install vim command  
     
   After successful installation bash prompt will return on the terminal

1. By default “index.html” file exist in **/usr/share/nginx/html** directory of nginx, let’s browse to this directory and check it’s presence with **ls** command  
   
2. Use Vim editor to edit this file,   
     
   a new vim window will open, to make changes into the file, press i to start insert (edit) mode of the vim editor
3. Make few simple changes in the index.html file, press **Esc** to exit from the insert mode, press **:wq** to save and exit from  
   
4. Now open browser of your choice and, on address bar enter url <http://localhost:8000> , a customized index.html page is displayed, meaning we have successfully edited and configured index.html file on nginx server on docker.  
   
5. Now if we want to pack this customized nginx container into an image that can be shared, posted etc, do following, first commit this container to an image  
   **docker commit <container id> <author> <tag>**  
     
   \*\*to know the container id, we can use **docker ps** command

7d53 is first 4 characters from the container hash  
zafar is the Author name

100 is the tag assigned to this image  
docker has assign a sha256 hash for our customized image of nginx

1. Now run **docker images** command to verify the parameters  
     
   docker has created a customized nginx image for us with size 196 MB  
   \*\*this extra-large size is partially due to the fact that we have updated the nginx with **apt update** command and mainly due to installation of **VIM package** into it.
2. Let’s run this customized image on port 5000  
   
3. Open any web browser of your choice and enter url <http://localhost:5000>   
     
   Our customized home page appears, Congratulations!!

This customized image can be uploaded to docker hub, shared among desired user group.