**Docker assignments**

Assignment1: You have been asked to:

● Pull ubuntu container

● Run this container, and map port 80 on the local

● Install apache2 on this container

● Check if you are able to access the apache page on your browser

Assignment1 solution:

First launch an ec2 Ubuntu instance and connect to it

sudo apt update

sudo apt install docker.io –y

docker –version

sudo docker pull Ubuntu

sudo docker images

sudo docker run –it –d –name assignment1 –p 80:80 ubuntu

sudo docker ps

sudo docker exe –it assignment1 bash

apt install apache2

apt update

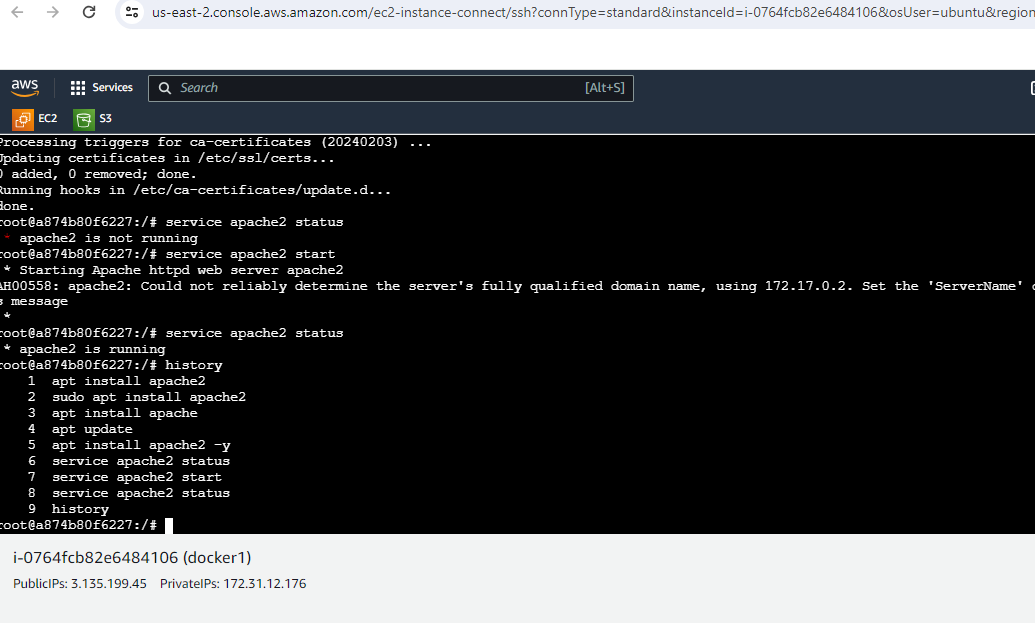
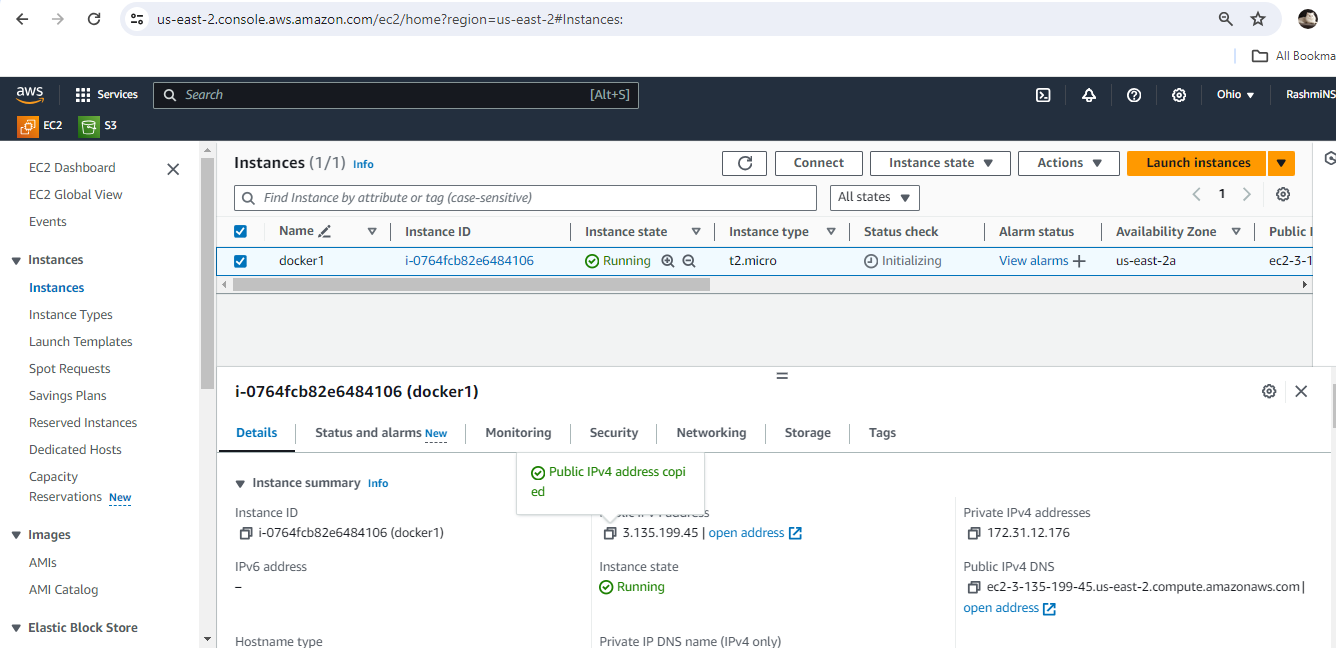
apt install apache2 –y

\*\* Check the apache website

service apache2 status

service apache2 start

service apache2 status



Assignment2: You have been asked to:

● Save the image created in Assignment 1 as a Docker image

● Launch container from this new image and map the port to 81

● Go inside the container and start the apache2 service

● Check if you are able to access it on the browser

Assignment2 solution:

sudo docker ps

sudo docker commit <container\_id> <docker\_image> <name\_of\_image>

sudo docker images

sudo docker run –itd –name assignment2 –p 81:80 <docker\_image\_name>

sudo docker images

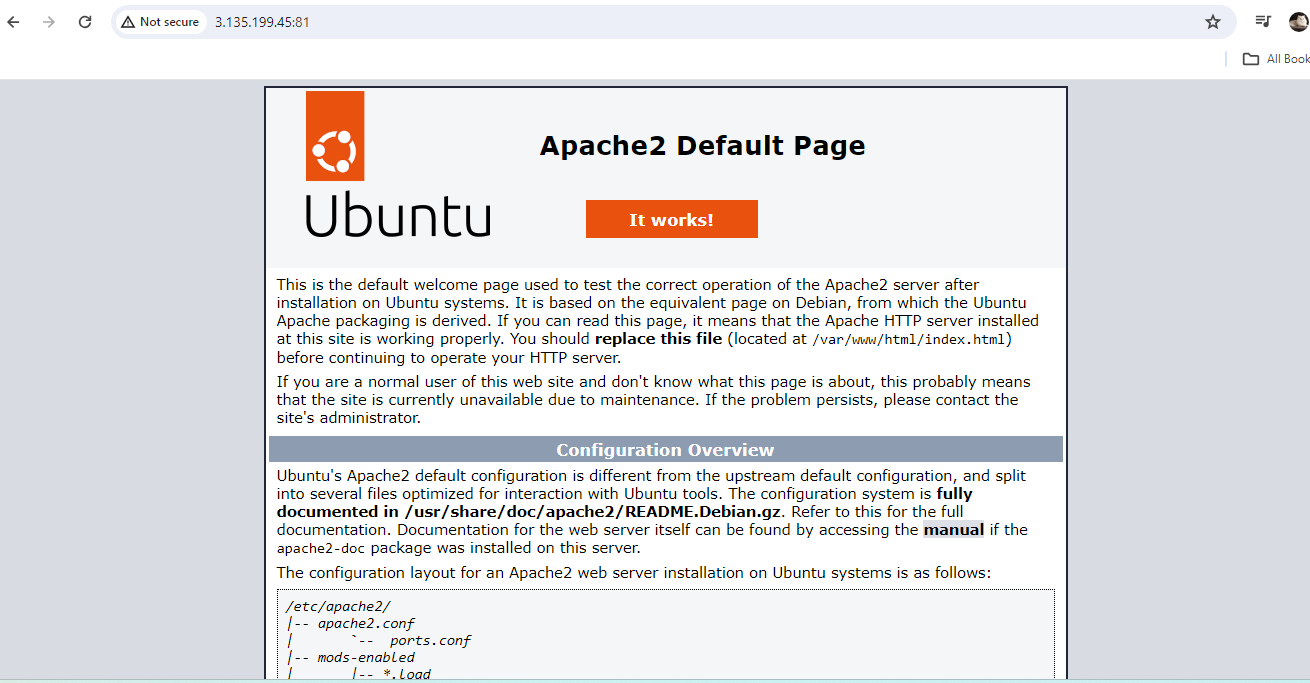
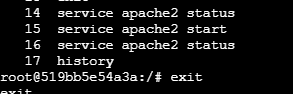
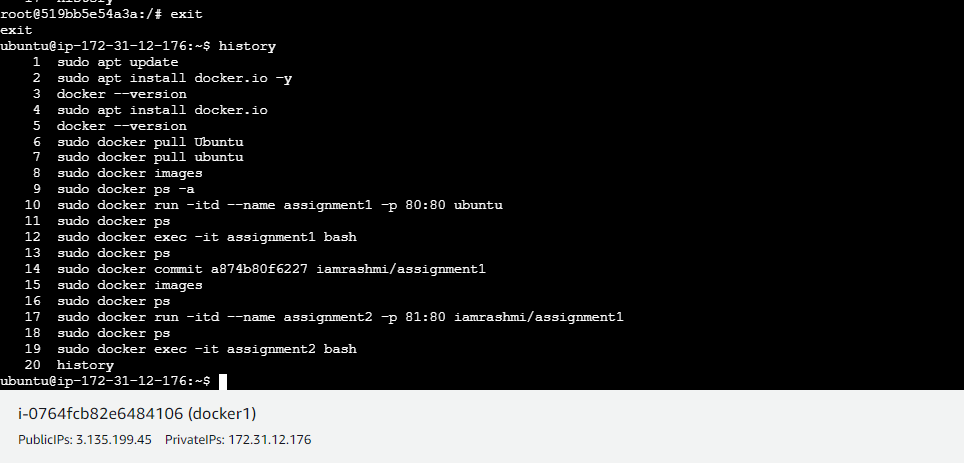
sudo docker ps

sudo docker exe –it assignment2 bash

service apache2 status

service apache2 start

\*\*Check Apache2 website with new port 81



Assignment3: You have been asked to:

● Use the saved image in the previous assignment

● Upload this image on Dockerhub

● On a separate machine pull this Dockerhub image, and launch it on port 80

● Start the apache2 service

● Verify if you are able to see the apache2 service

Assignment3 solution:

sudo docker images

\*\* Login to DockerHub

sudo docker login

sudo docker push <docker\_image\_name>

\*\*Create a new ec2 machine and connect

In new ec2,

sudo apt update

sudo apt install docker.io –y

sudo docker pull <docker\_image\_name>

sudo docker images

sudo docker run –itd –name assignment3 –p 82:80 <docker\_image\_name>

sudo docker ps

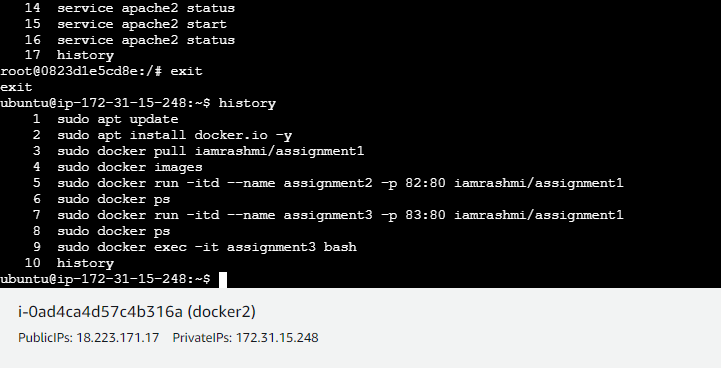
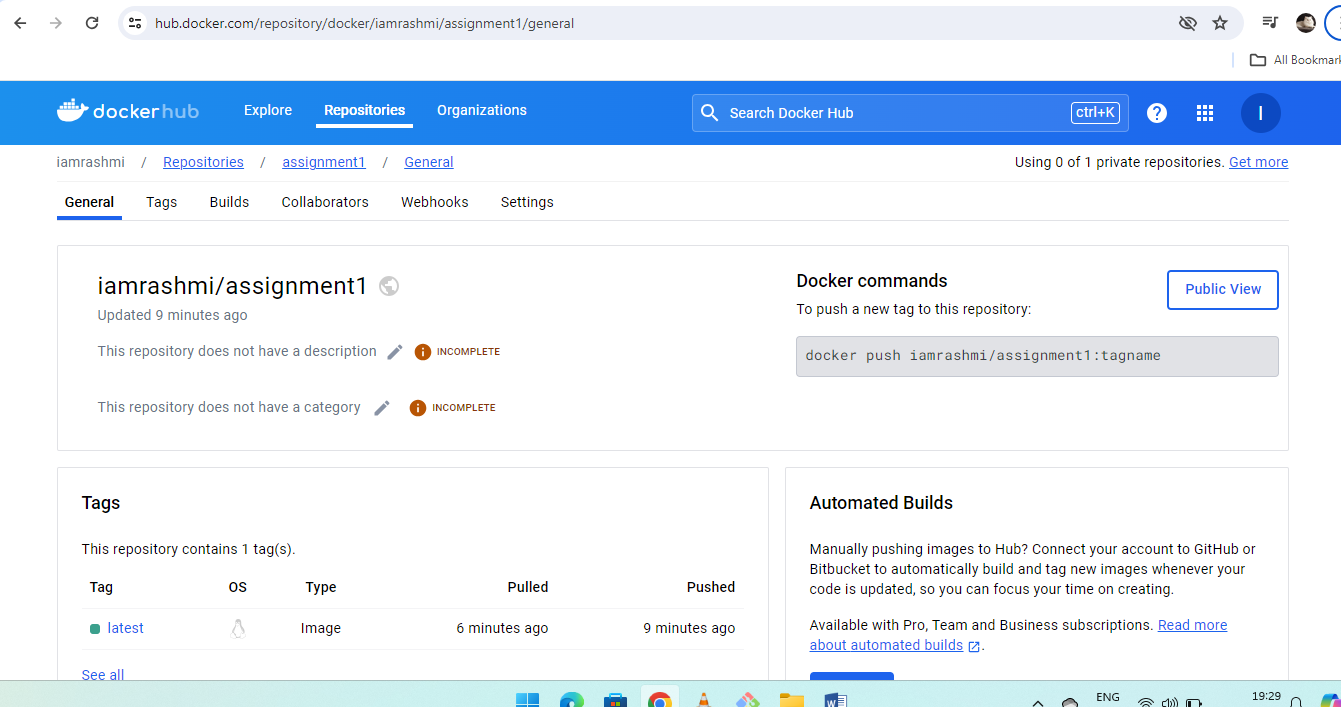
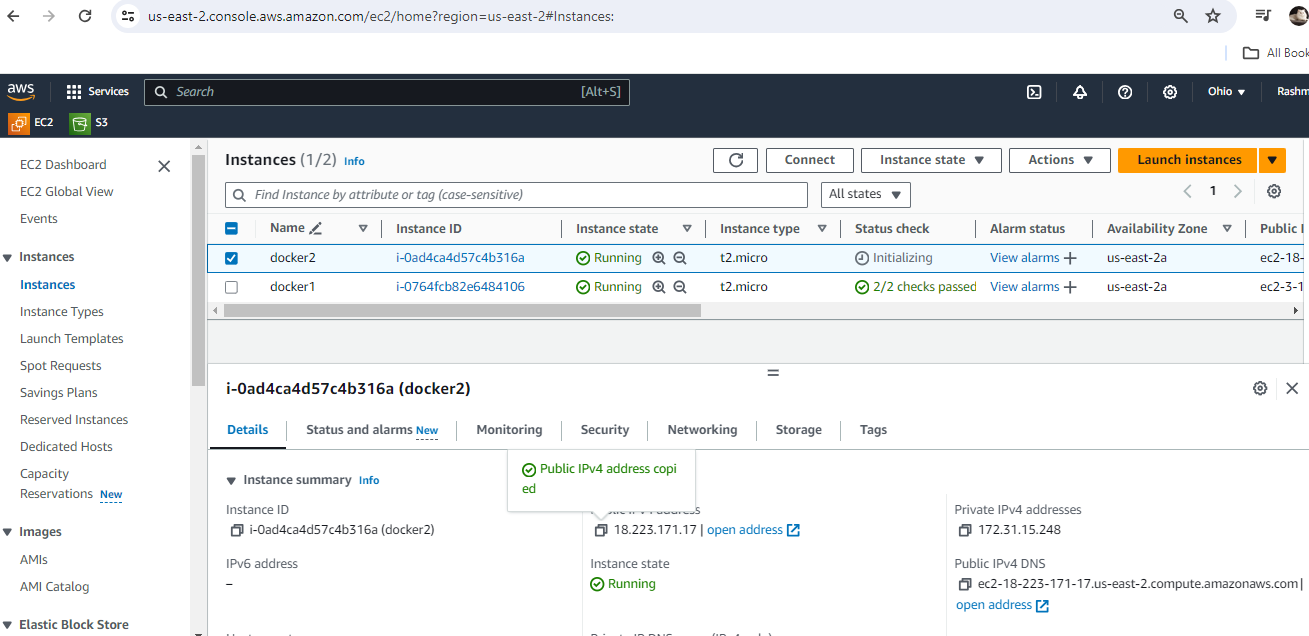
sudo docker exec –it assignment3 bash

service apache2 status

service apache2 start

service apache2 status

\*\*Check Apache2 webite with new port 82



Assignment4: You have been asked to:

Create a dockerfile with the following specs:

● Ubuntu container

● Apache2 installed

● Apache2 should automatically run once the container starts

Assignment4 solution:

In any existing ec2 machine,

ls

sudo docker images

sudo nano Dockerfile

FROM Ubuntu

RUN apt update

RUN apt install apache2 –y

ENTRYPOINT apachectl –d FOREGROUND

ls

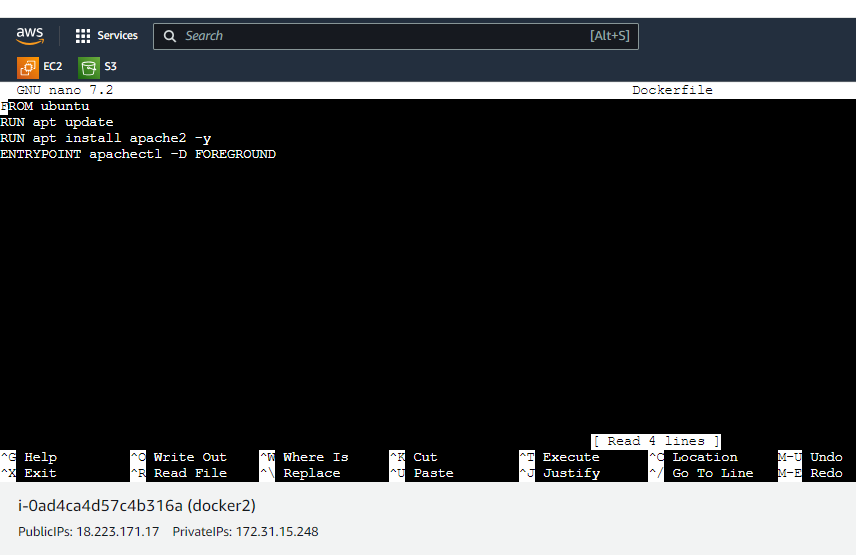
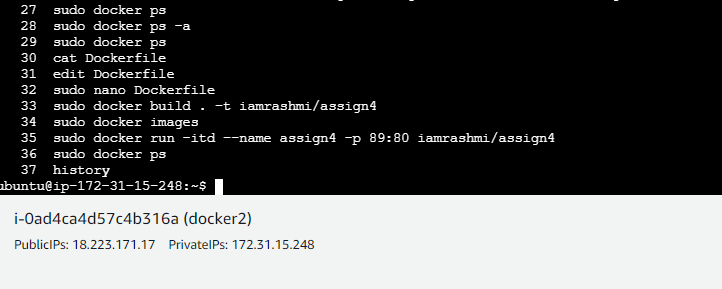
sudo docker build . –t <docker\_image\_name>

sudo docker images

sudo docker run –itd – name assignment4 – p 87:80 <docker\_image\_name>

sudo docker ps

sudo docker ps –a



Assignment5: You have been asked to:

● Create a sample HTML file

● Use the Dockerfile from the previous task

● Replace this sample HTML file inside the docker container with the default page

Assignment5 solution:

Use same ec2 assignment4 machine instance

ls

nano index.html

git clone <https://github.com/hshar/website.git>

ls

cd website/

ls

sudo nano Dockerfile

FROM Ubuntu

RUN apt update

RUN apt install apache2 –y

ADD . /var/www/html

ENTRYPOINT apachectl –D FOREGROUND

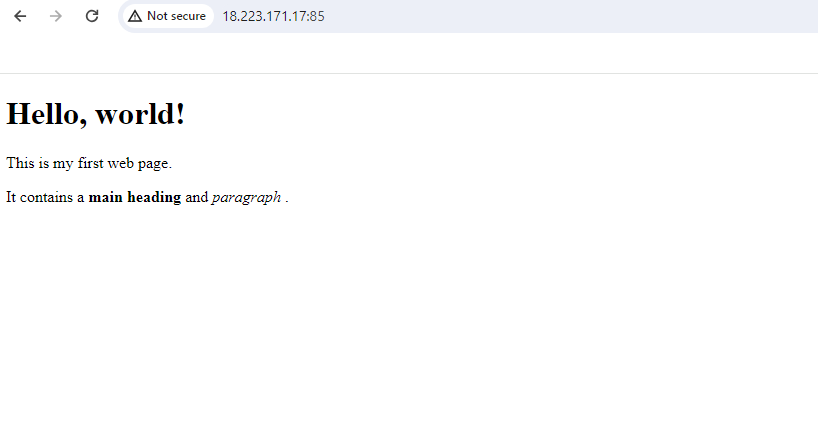
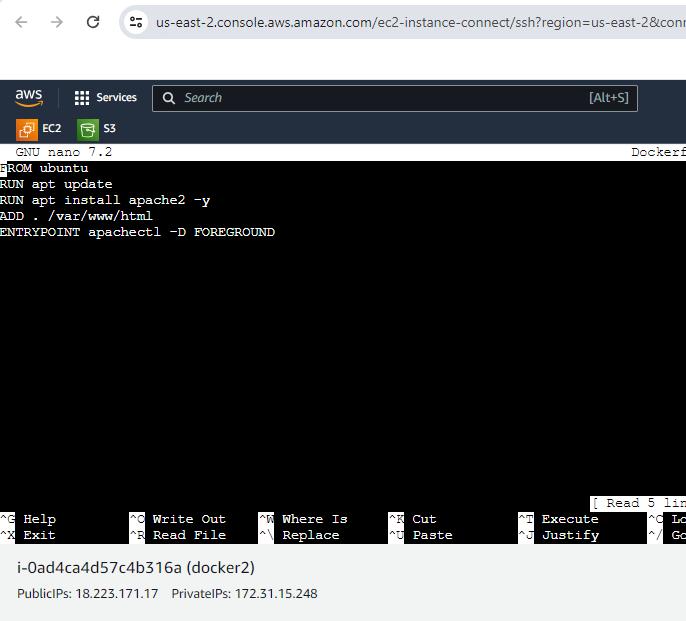
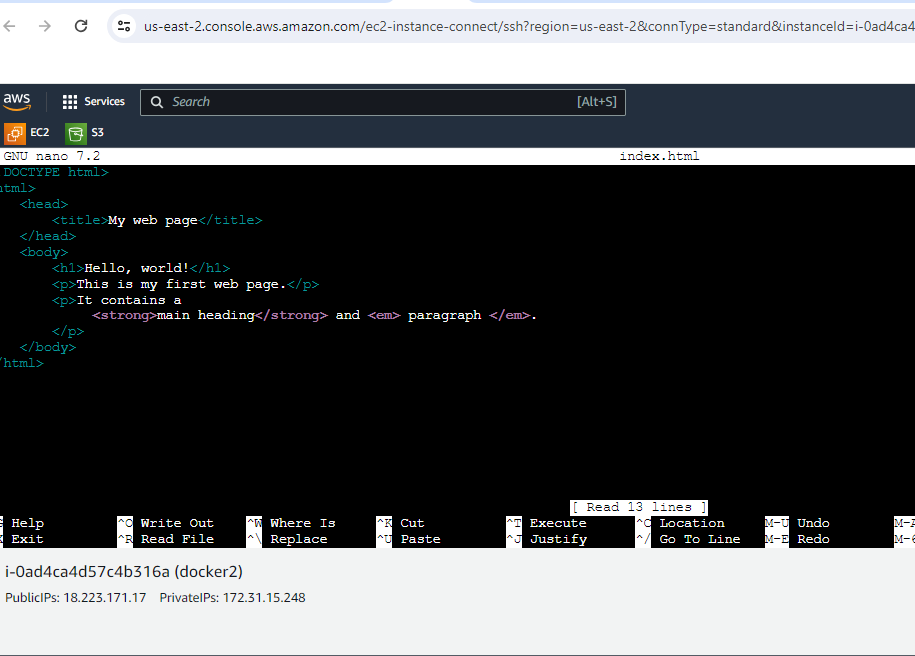
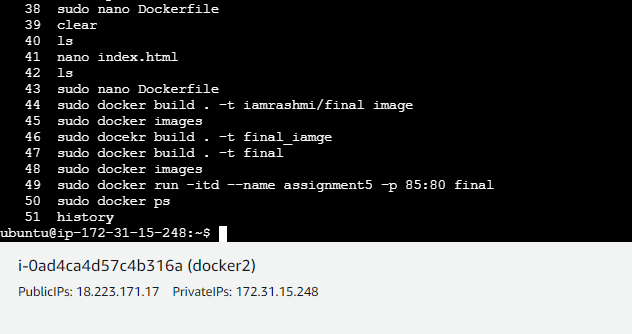
sudo docker build . –t final image

sudo docker images

sudo docker run –itd –name assignment5 –p 84:80 final image

sudo docker ps

\*\*Check Apche2 website on new port 84



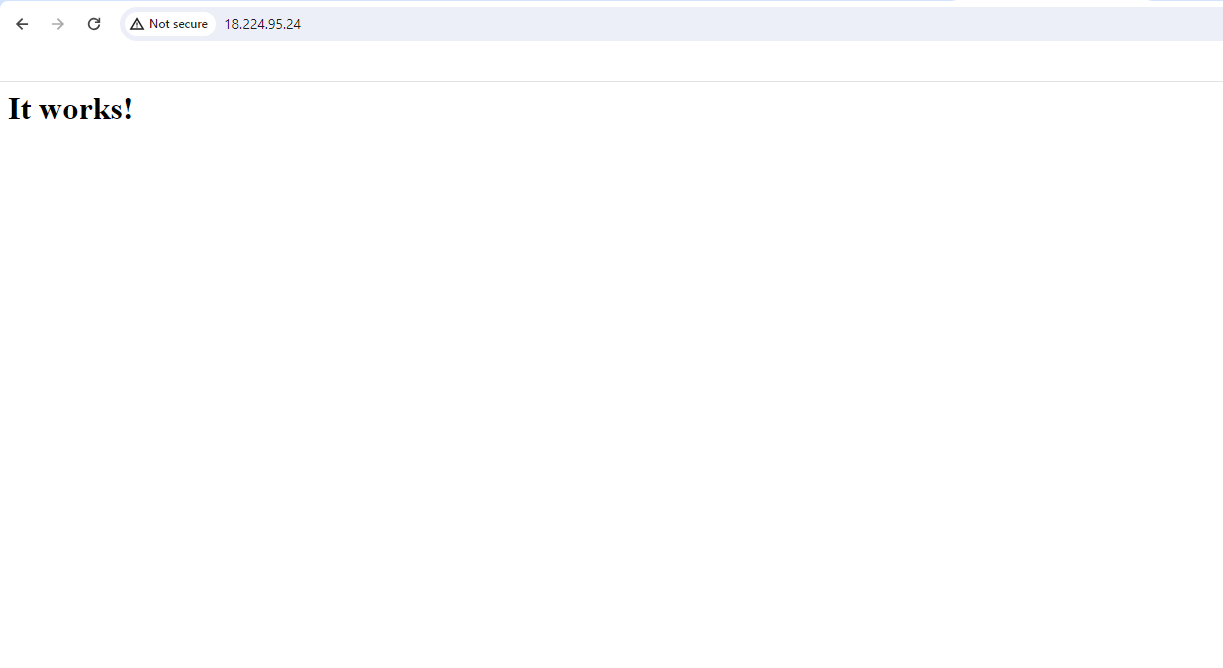
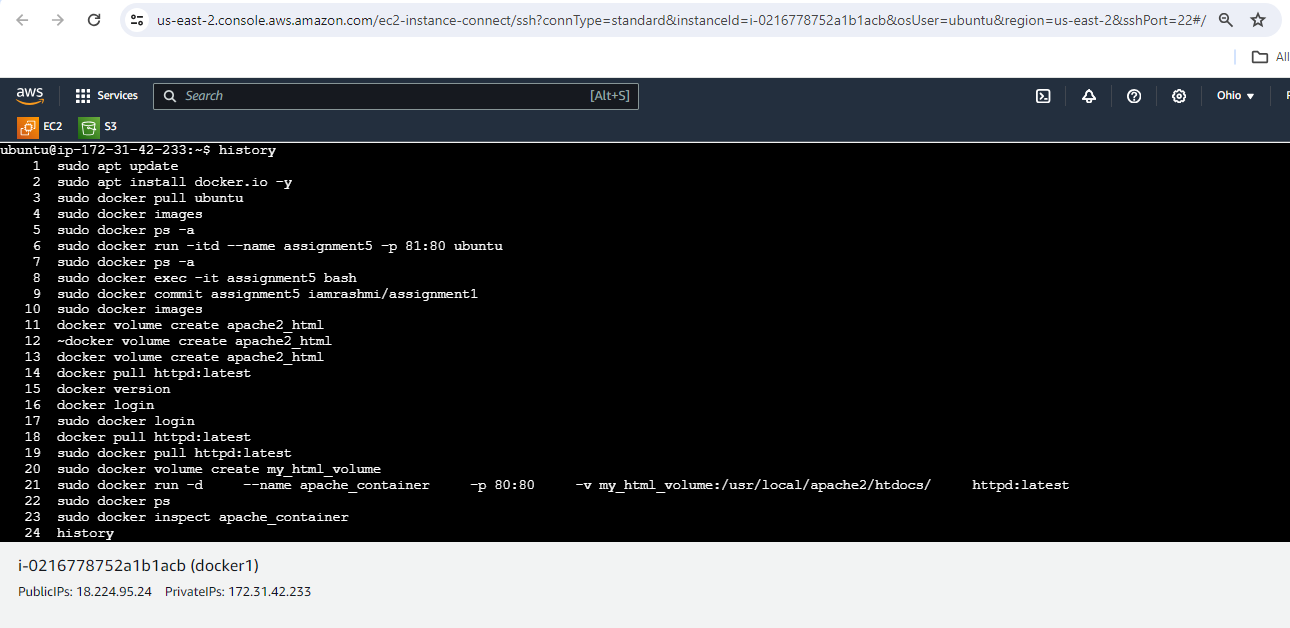
Assignment6: You have been asked to:

● Launch the Apache2 container created in previous module

● Create a Docker volume on /var/www/html

Assignment6 Solution:

From previous module Apache2 container, new Docker volume is created.



Assignment7: You have been asked to:

● Use the apache2 container created in previous module

● Create a bind mount on /var/www/html to replace html files dynamically

Assignment7 solution:

Creating docker bind mount

