**Experiment No: 09**

**Aim:** To learn dockerfile instructions, build an image for a sample web application using dockerfile.

**Lab Outcome**: LO5 - Explain concept of containerization and Analyze the Containerization of OS images and deployment of applications over Docker

**Theory:**

Dockerfile:

* A Dockerfile is a text document that contains commands that are used to assemble an image. We can use any command that call on the command line. Docker builds images automatically by reading the instructions from the Dockerfile.
* The docker build command is used to build an image from the Dockerfile. You can use the -f flag with docker build to point to a Dockerfile anywhere in your file system.

FROM

* This instruction is used to set the Base Image for the subsequent instructions. A valid Dockerfile must have FROM as its first instruction.
* Ex. FROM ubuntu

LABEL

* We can add labels to an image to organize images of our project. We need to use LABEL instruction to set label for the image.
* Ex. LABEL vendorl = "JavaTpoint"

RUN

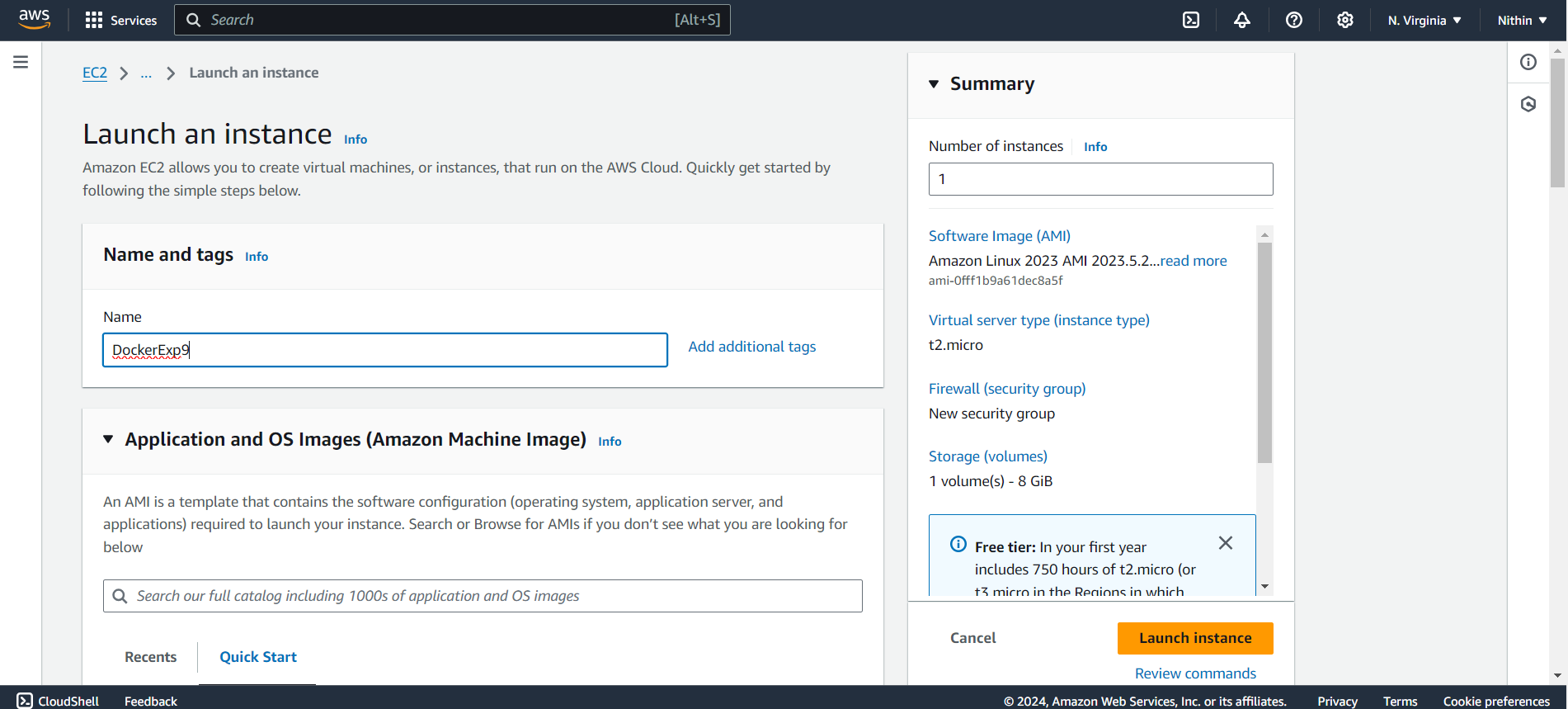
* This instruction is used to execute any command of the current image.
* Ex. RUN /bin/bash -c 'source $HOME/.bashrc; echo $HOME'

CMD

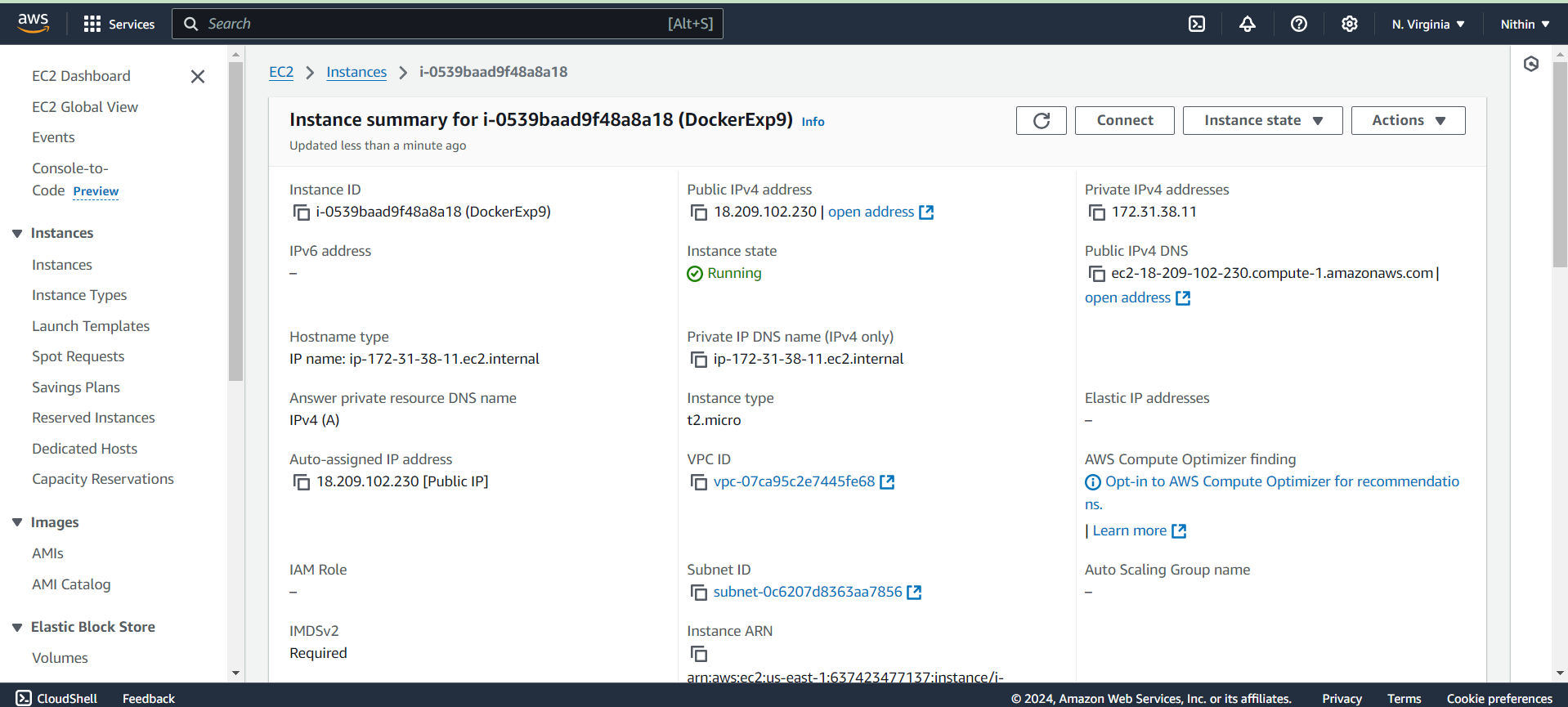
* This is used to execute application by the image. We should use CMD always in the following form CMD ["executable", "param1", "param2"?]
* This is preferred way to use CMD. There can be only one CMD in a Dockerfile. If we use more than one CMD, only last one will execute.

**Output**

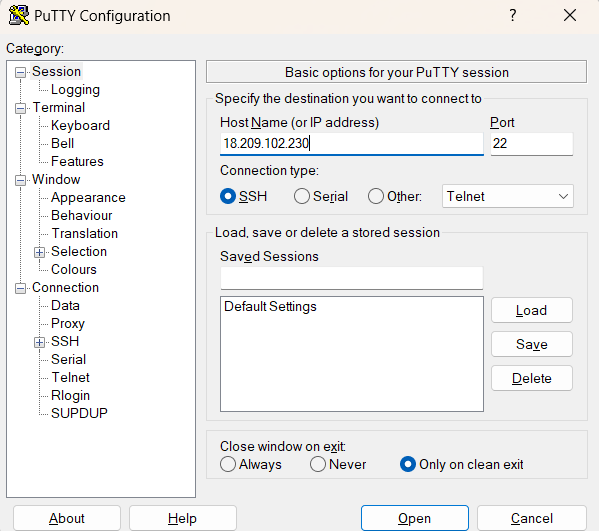
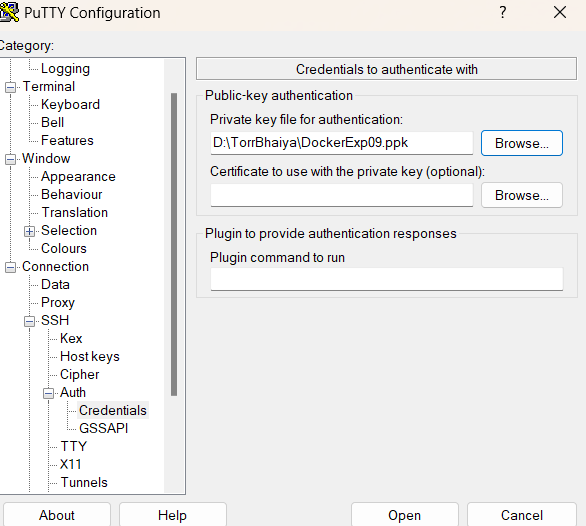
Step 1: login into your AWS account and launch instance.

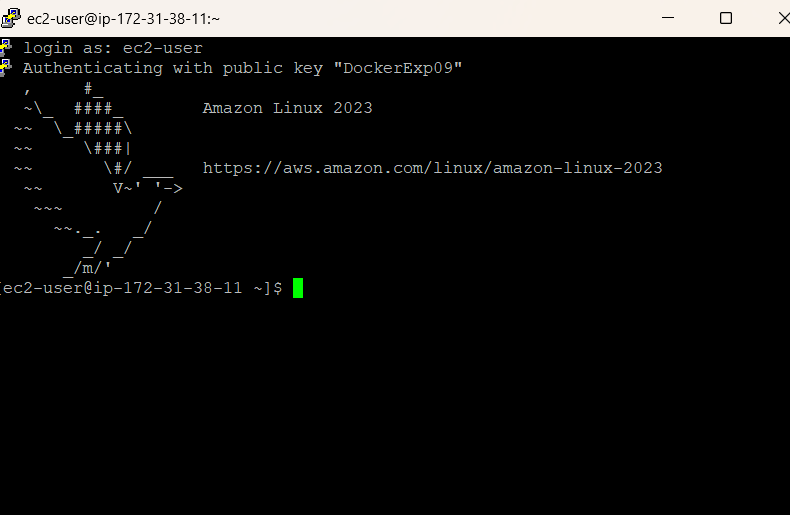


Launch the instance



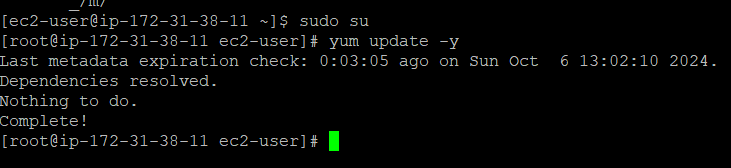
and then login this instance through puTTy.

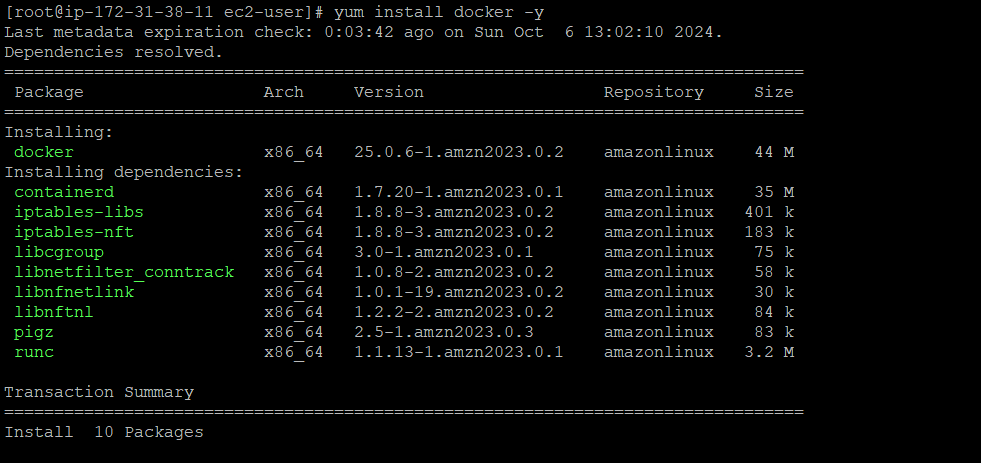


Come to the root and update your machine.

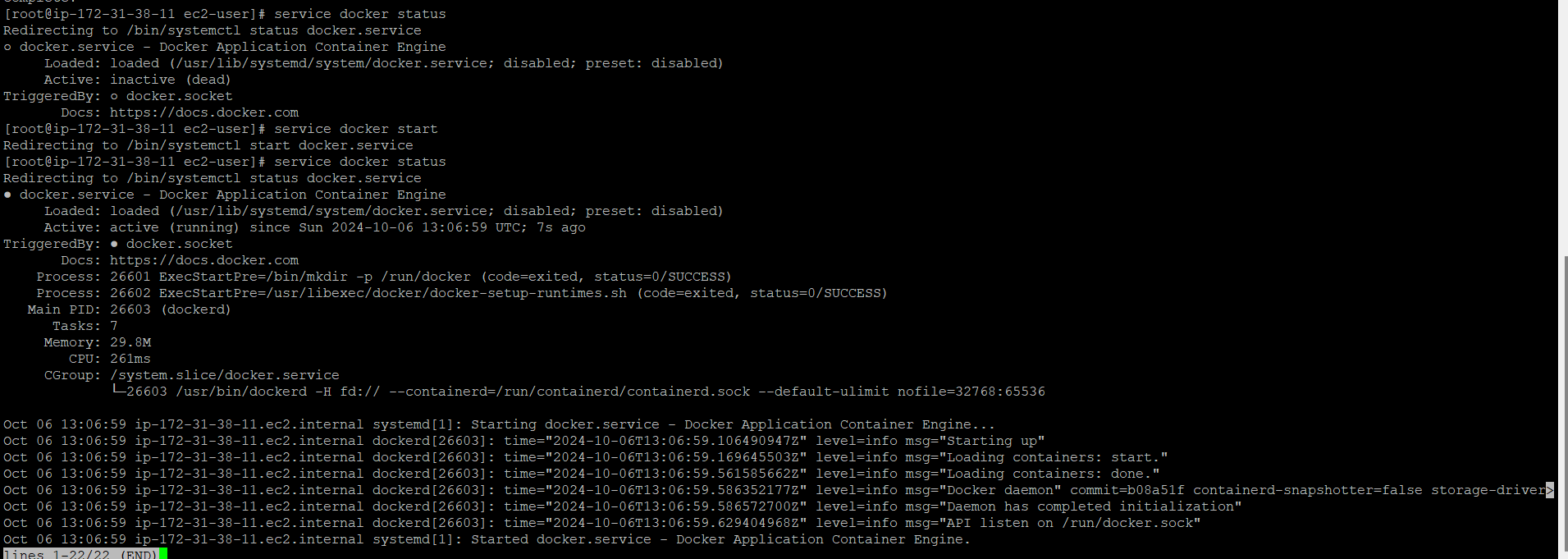
* sudo su
* yum update -y



Step 2: Install docker.



Start the docker services.



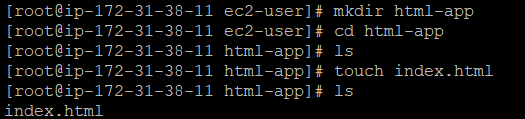
Step 3: Making a directory which contains code related to our application.

mkdir html-app

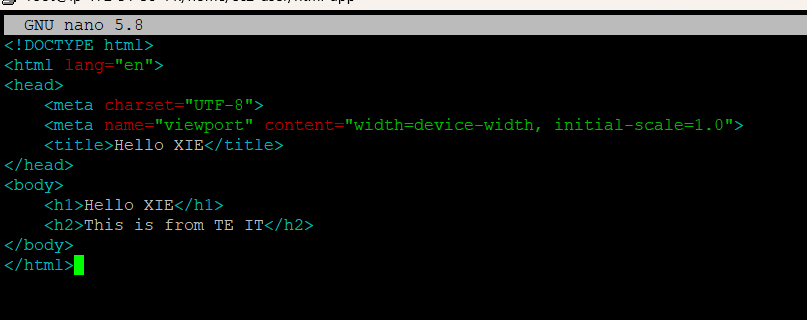
cd html-app

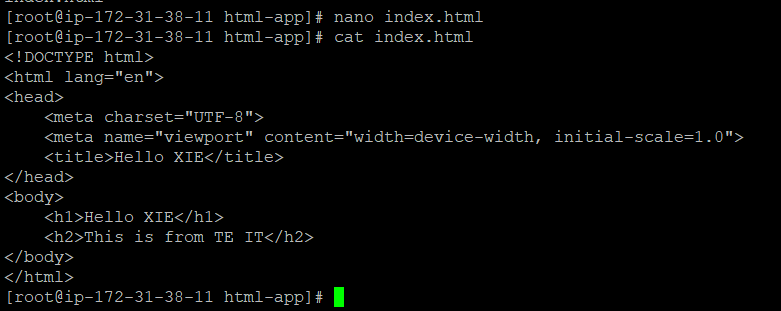
touch index.html

ls



nano index.html

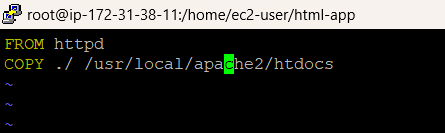




Step 4: Create and configure docker file.

vim Dockerfile



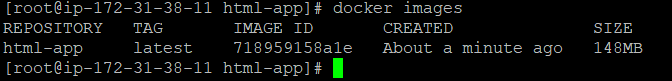


Step 5: Build an Images for the docker file.

docker build -t html-app .



docker images

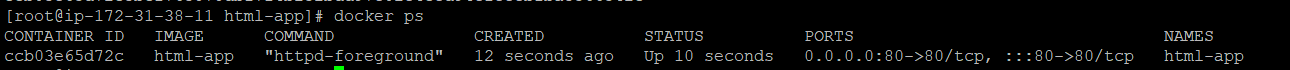


Step 6: Run the image.

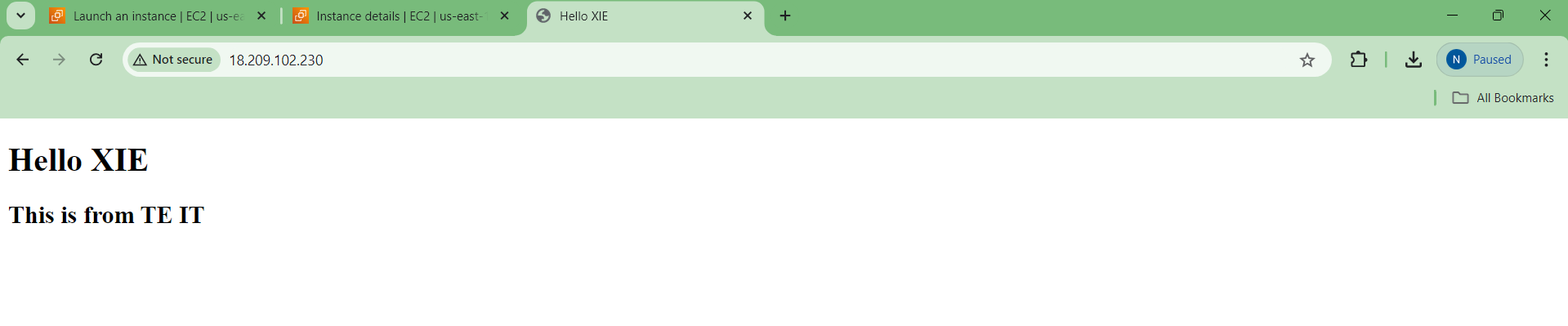
Docker run -itd -p 80:80 –name html-app html-app



docker ps



Step 7: Copy public IP of your instance and paste in the new tab of browser.



**Conclusion:** A Dockerfile is a text document that contains commands that are used to assemble an image. Docker builds images automatically by reading the instructions from the Dockerfile. From this experiment, LO-5 and PO1, PO2, PO3, PO4, PO5, PO8, PO10, PO12 are achieved successfully.