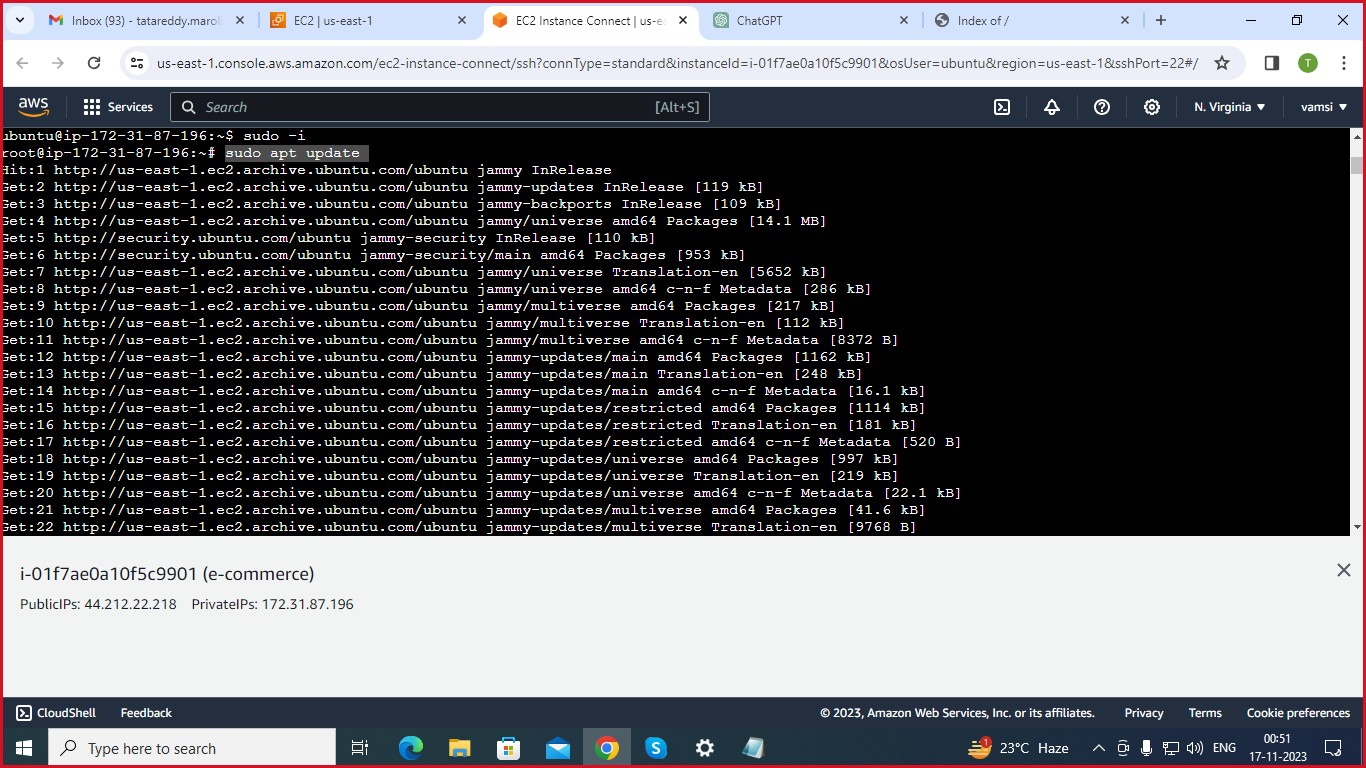
E-COMMERCE PROJECT

**Description:**

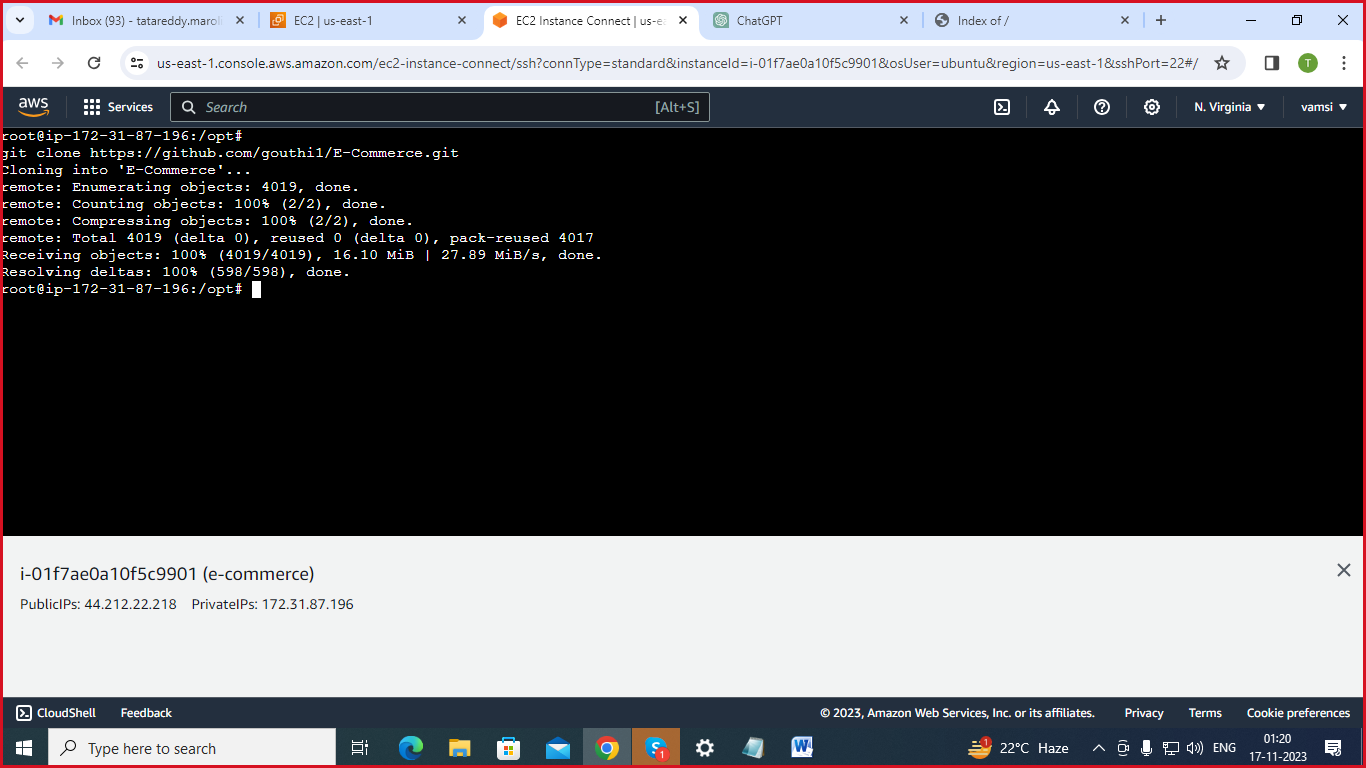
An e-commerce application is a software platform that enables online buying and selling of goods and services over the internet. These applications provide a virtual marketplace where businesses and consumers can connect, browse through products or services, make transactions, and manage their accounts.

**Step by step process for E-Commerce Application:**

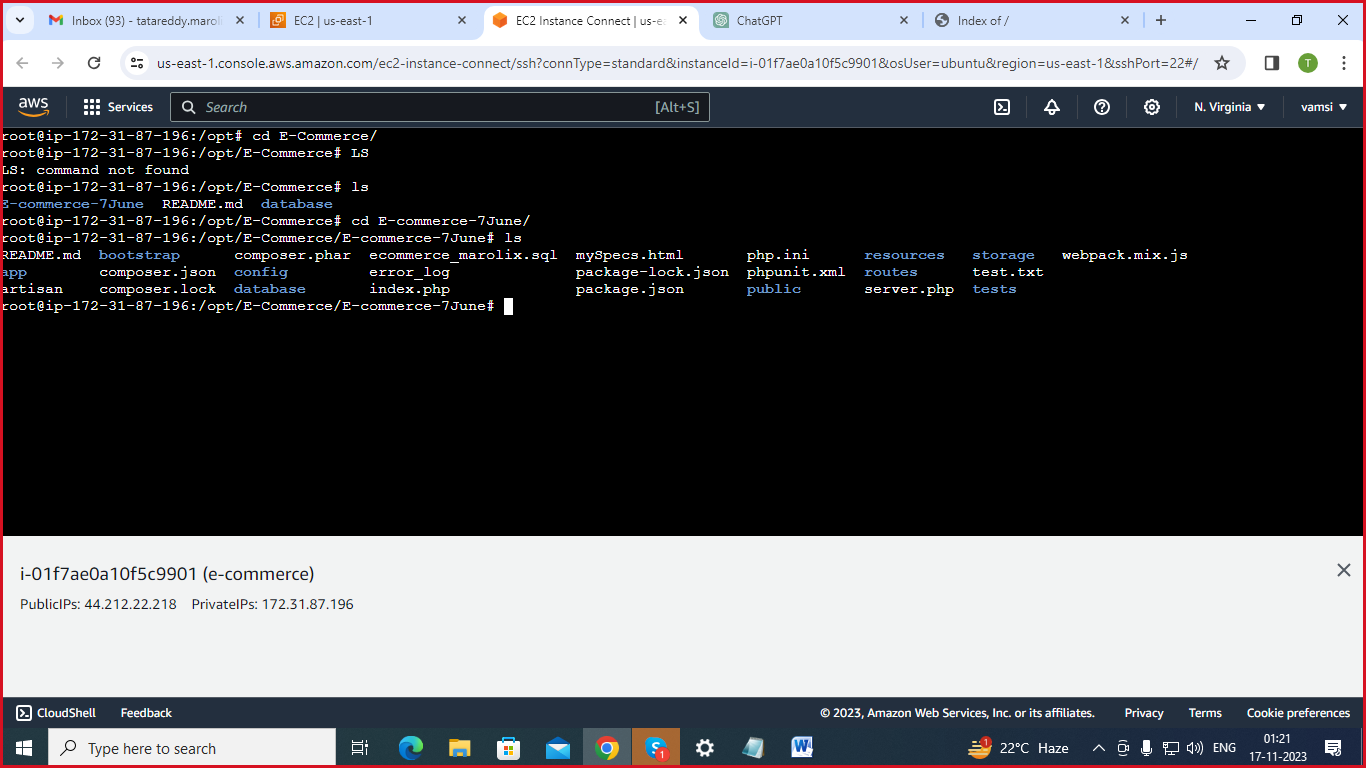
* **sudo apt update**: Updates the local package database on your Linux system



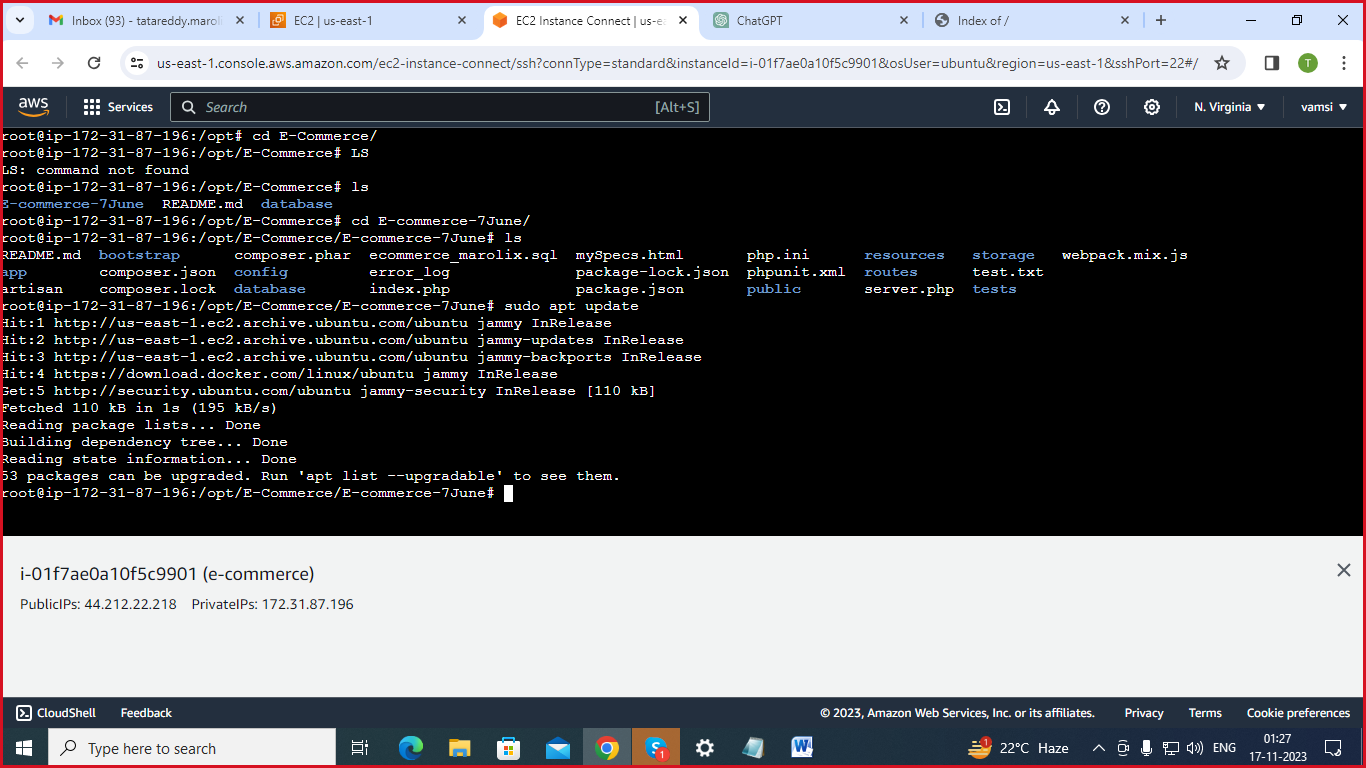
* **git clone https://github.com/gouthi1/E-Commerce.git**: Clones the E-Commerce project from the specified GitHub repository.

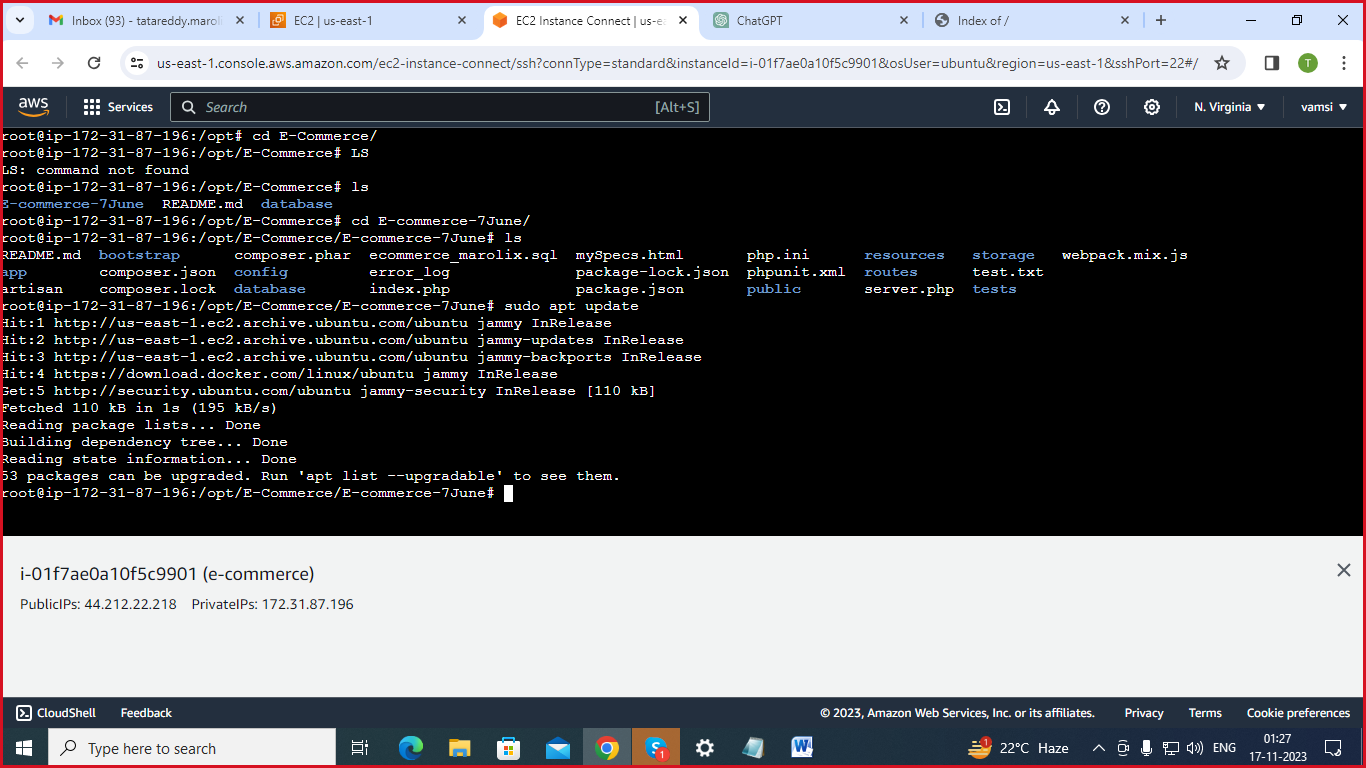


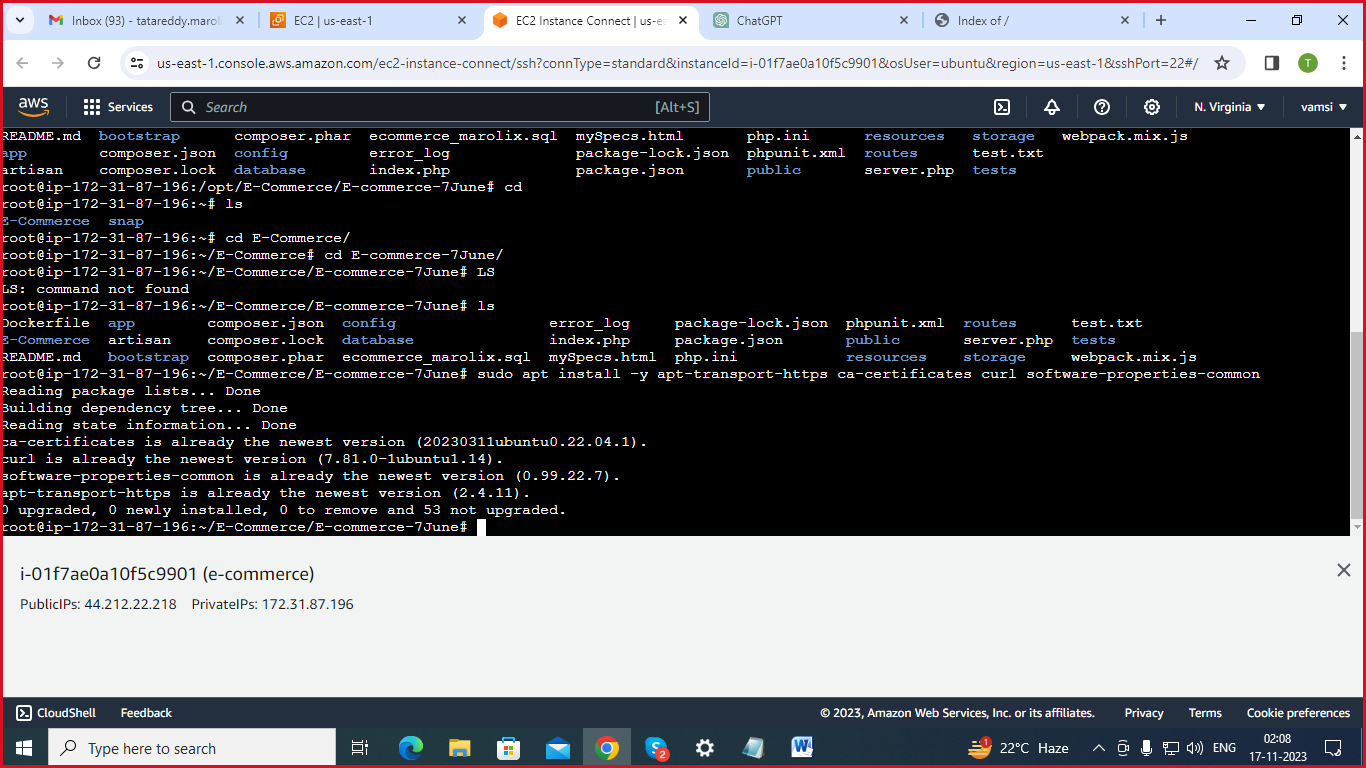
* **ls**: Lists the contents of the current directory.
* **cd E-Commerce/:** Changes the current directory to the "E-Commerce" directory.
* **ls**: This is a typo; it should be ls. Lists the contents of the "E-Commerce" directory.
* **cd E-commerce-7June/:** Changes the current directory to "E-commerce-7June.



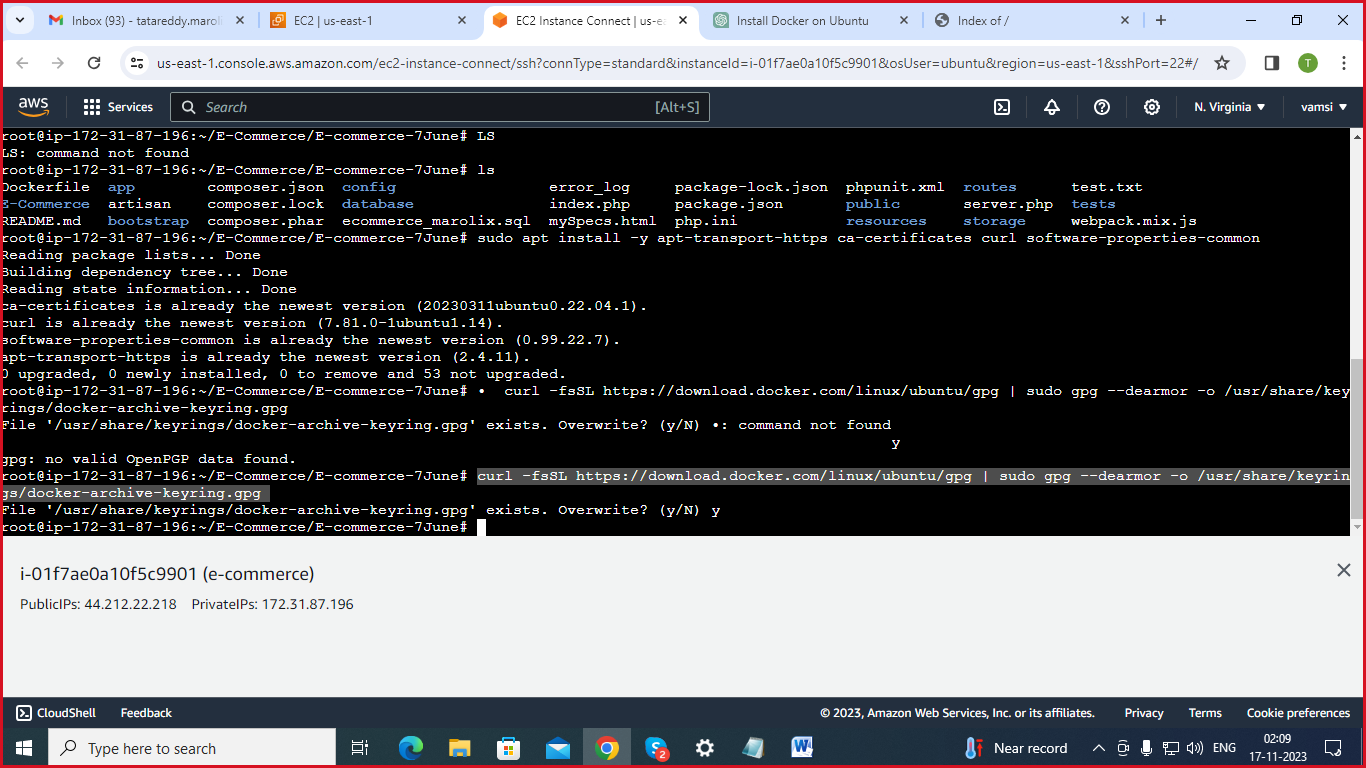
* **sudo apt-get update**: Updates the package lists for upgrades and new package installations.



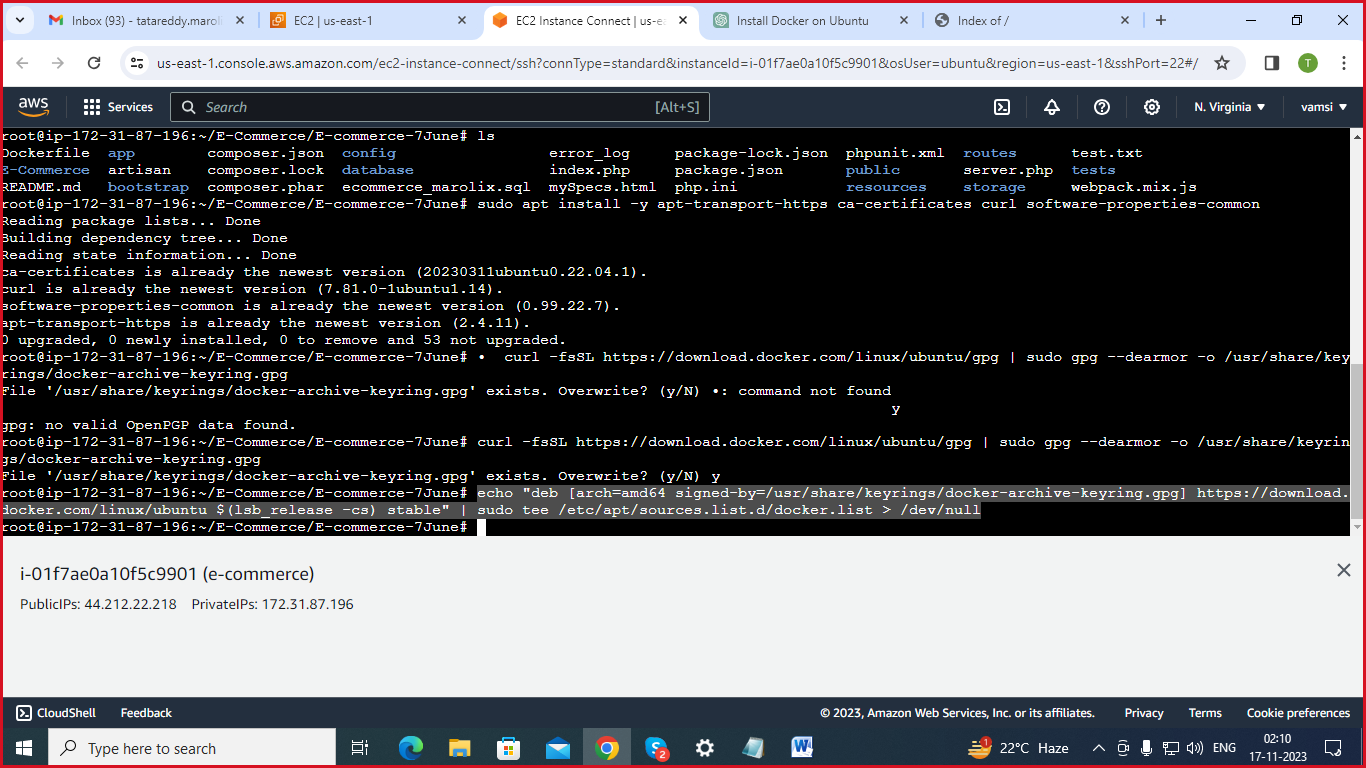
* **DOCKER INSTALLATION STEPS**
* **sudo apt install docker:** Installs Docker**.**
* **docker start service:** Starts the Docker service. However, this is not a standard Docker command, so it might be a specific service related to your project.
* **sudo apt update:** Updates the package lists again.
* ****
* **sudo apt install -y apt-transport-https ca-certificates curl software-properties-common**: Installs required dependencies for adding Docker's GPG key.

****

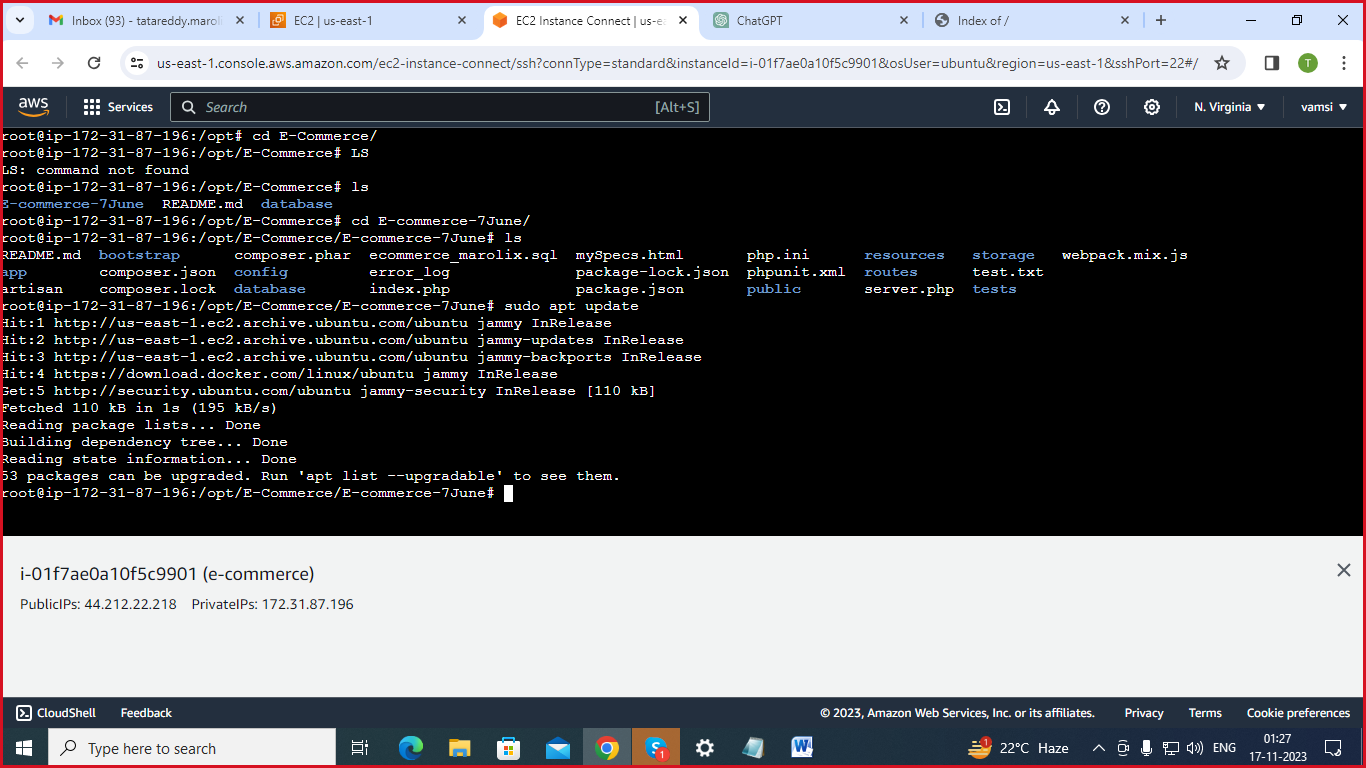
* **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg:** Downloads and adds Docker's GPG key.

****

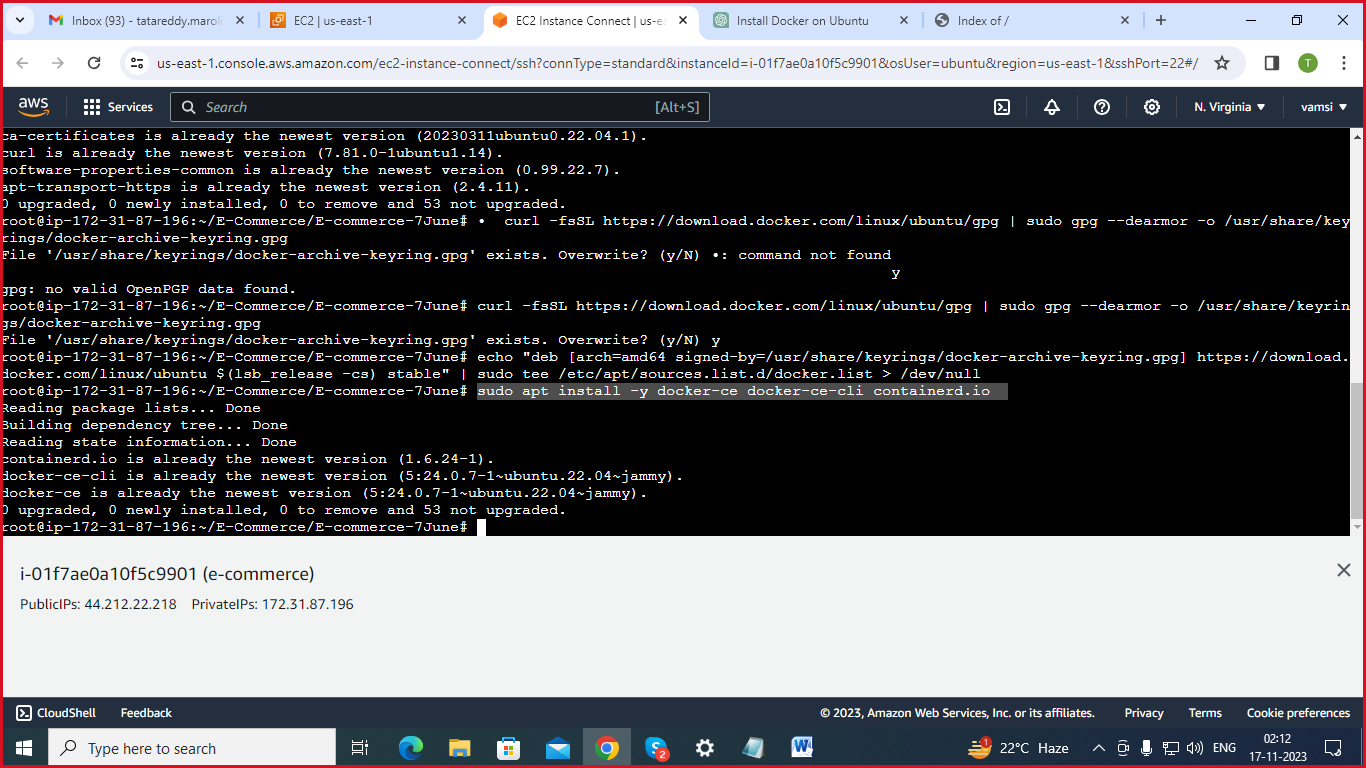
* **echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null:** Adds the Docker repository to the system.

****

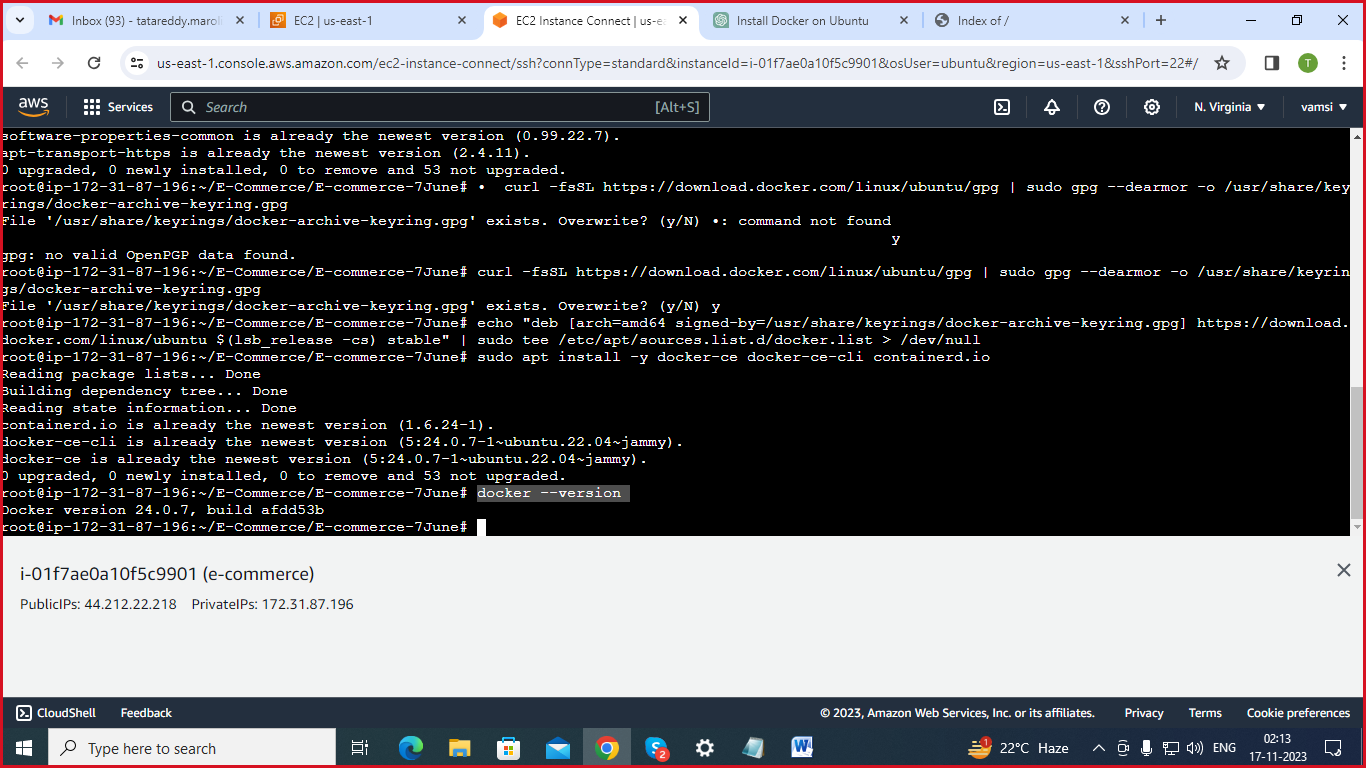
* **sudo apt update:** Updates the package lists again to include the Docker repository.

****

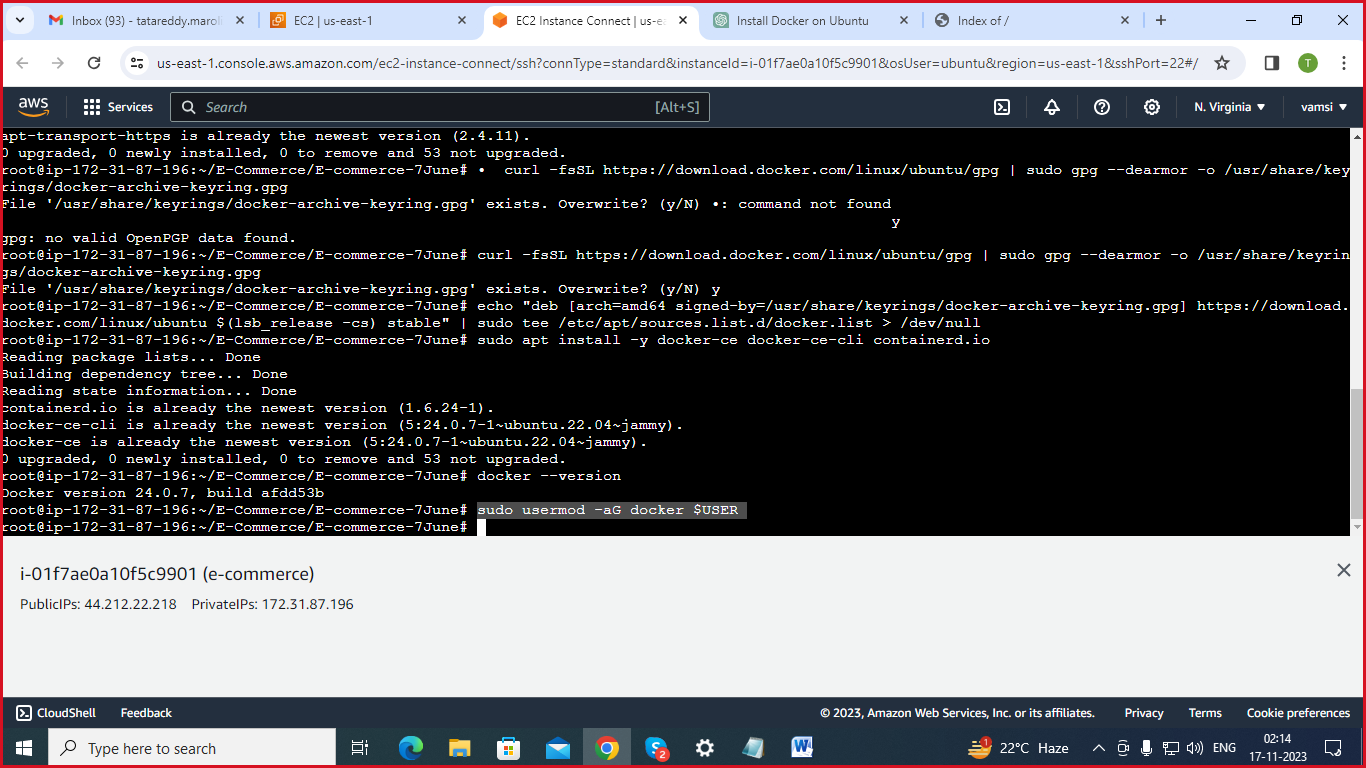
* **sudo apt install -y docker-ce docker-ce-cli containerd.io**: Installs Docker Engine.

****

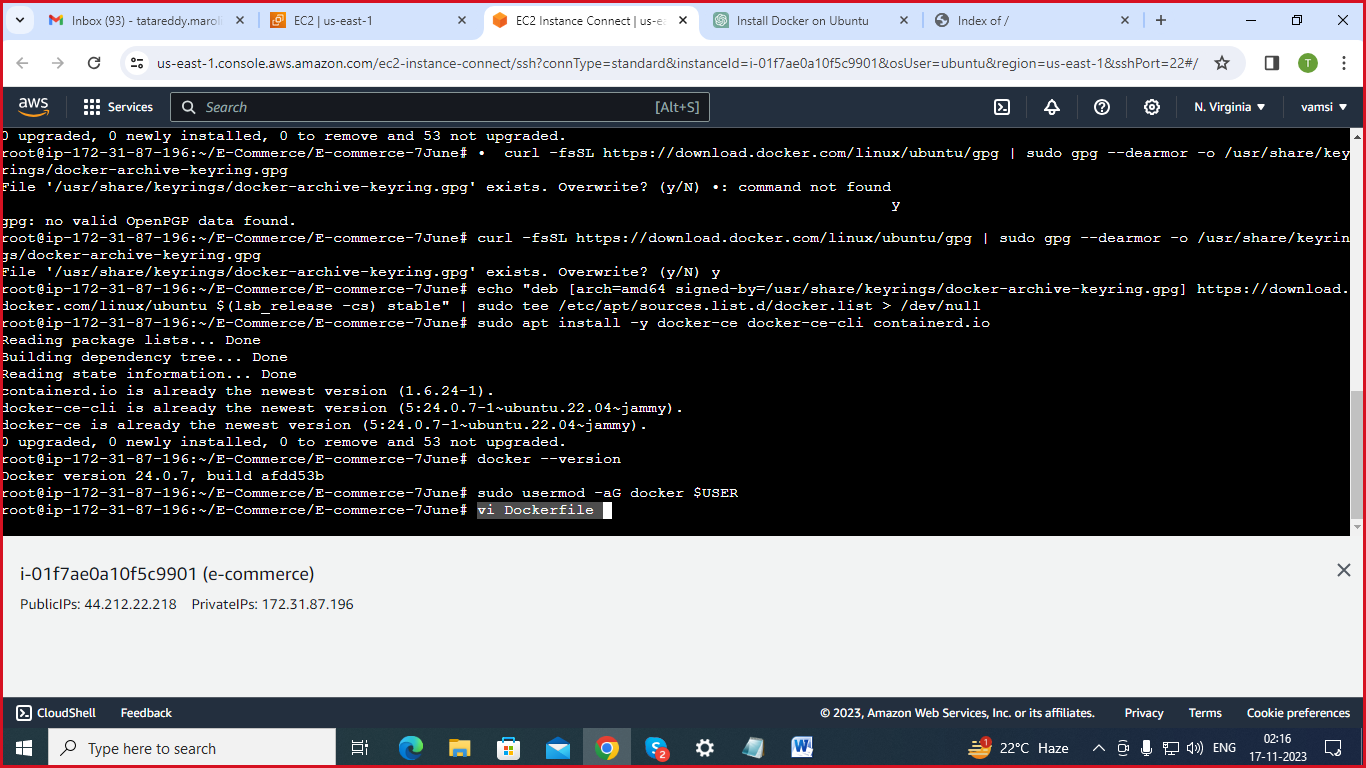
* **sudo docker --version:** Checks the installed Docker version**.**

****

**sudo usermod -aG docker $USER:** Adds the current user to the "docker" group, allowing the user to run Docker commands without sudo.

****

* **create docker file**
* **vi Dockerfile:** Opens the Dockerfile in the vi text editor**.**

****

* **Instructions on dockerfile;**

FROM ubuntu:latest

ENV DEBIAN\_FRONTEND=noninteractive

RUN apt-get update && \

apt-get upgrade -y && \

apt-get install -y \

php php-mysql php-json php-gd php-mbstring php-zip php-curl \

php8.1-xml php8.1-zip php8.1-gd php8.1-intl php8.1-pdo \

apache2 mysql-server git vim composer

RUN rm /var/www/html/index.html

RUN git clone https://github.com/gouthi1/E-Commerce.git /var/www/html/E-Commerce

RUN service mysql start && \

mysql -u root -e "CREATE DATABASE ecommerce\_marolix;" && \

mysql -u root ecommerce\_marolix < /var/www/html/E-Commerce/database/database/ecommerce\_marolix.sql

RUN mv /var/www/html/E-Commerce/E-commerce-7June/.env.example /var/www/html/E-Commerce/E-commerce-7June/.env && \

cd /var/www/html/E-Commerce/E-commerce-7June && \

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');" && \

php composer-setup.php && \

mv composer.phar /usr/local/bin/composer

RUN cd /var/www/html/E-Commerce/E-commerce-7June && \

composer update

RUN apt-get install -y php-curl php-mysqli php-zip php-gd php8.1-xml

RUN a2enmod rewrite && \

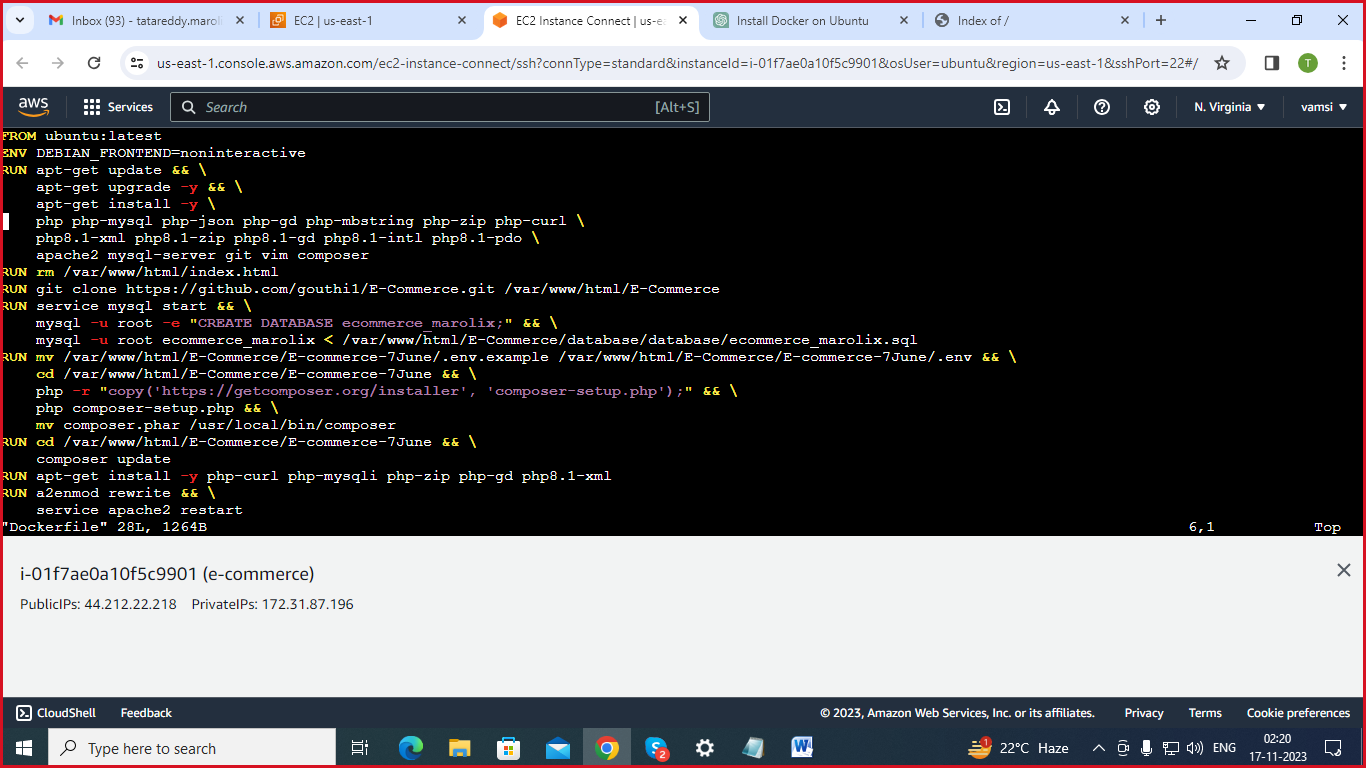
service apache2 restart

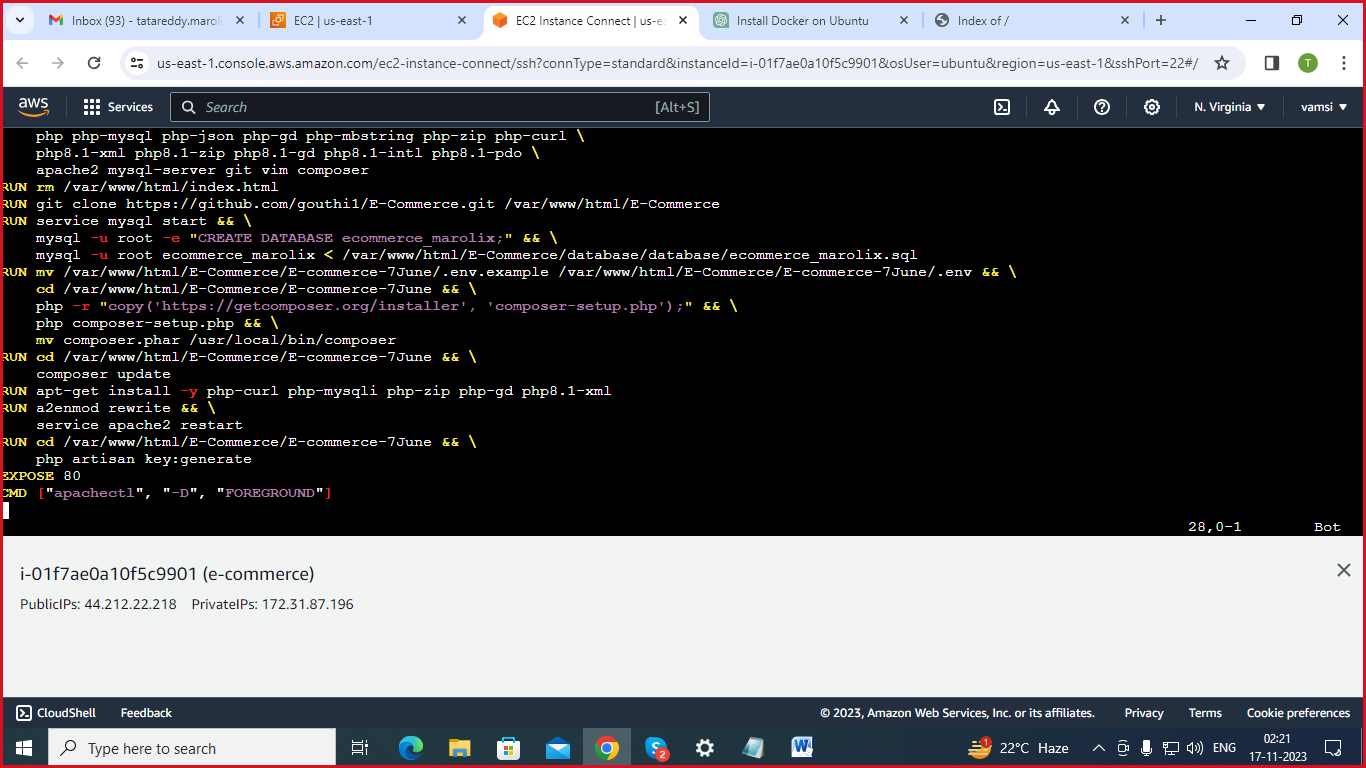
RUN cd /var/www/html/E-Commerce/E-commerce-7June && \

php artisan key:generate

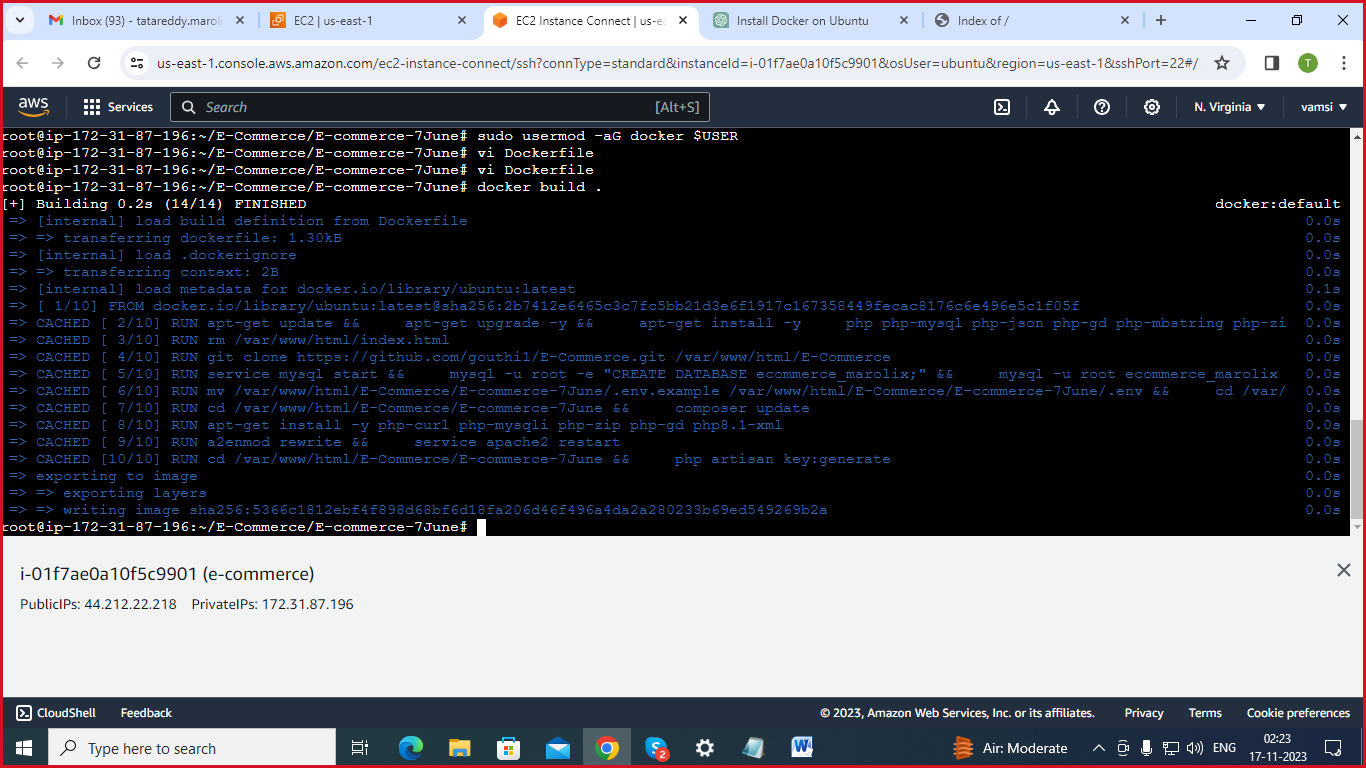
EXPOSE 80

CMD ["apachectl", "-D", "FOREGROUND"]

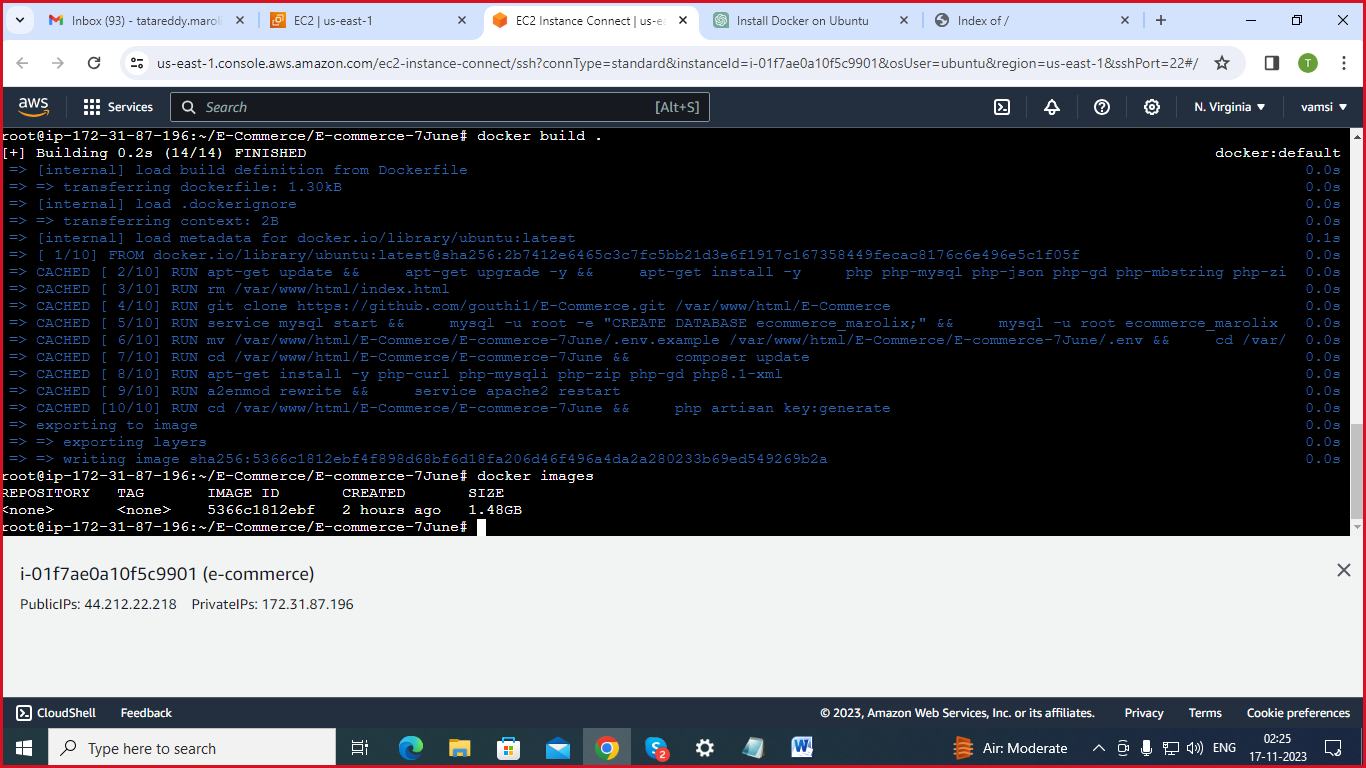
****

****

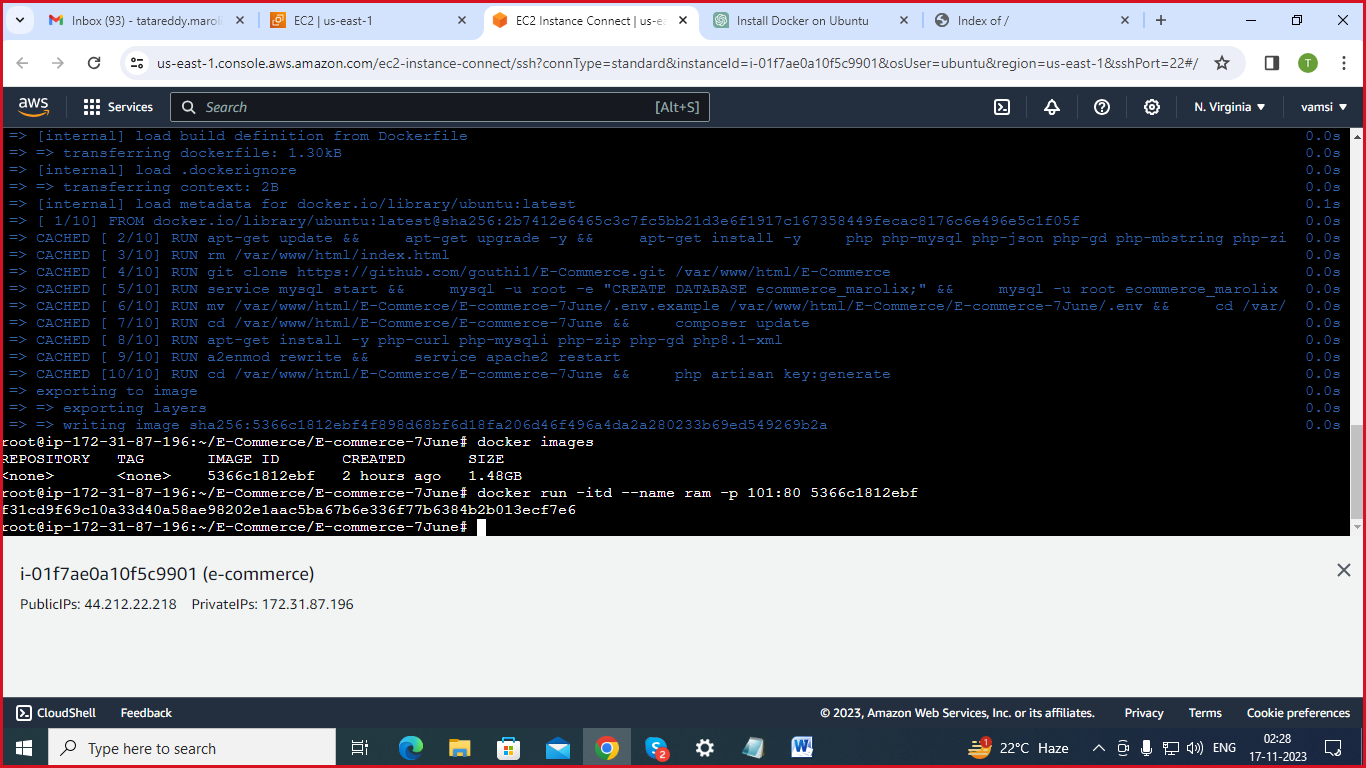
* **docker build .:** Builds a Docker image from the current directory

****

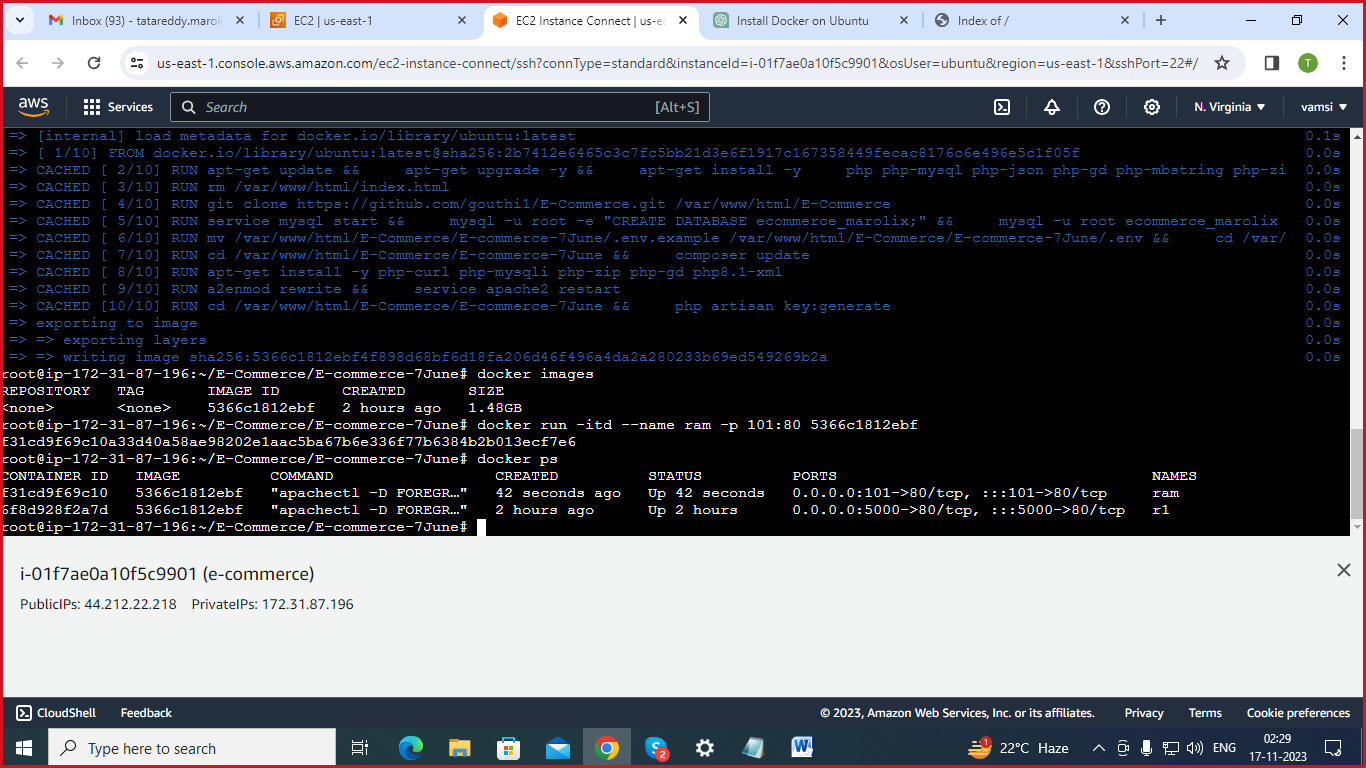
* **docker images:** Lists the Docker images on the system.

****

* **docker run -itd --name ram -p 101:80 5366c1812ebf**: Runs a Docker container named "ram" in detached mode, mapping port 101 on the host to port 80 in the container. The container is based on the image with the specified ID (5366c1812ebf)

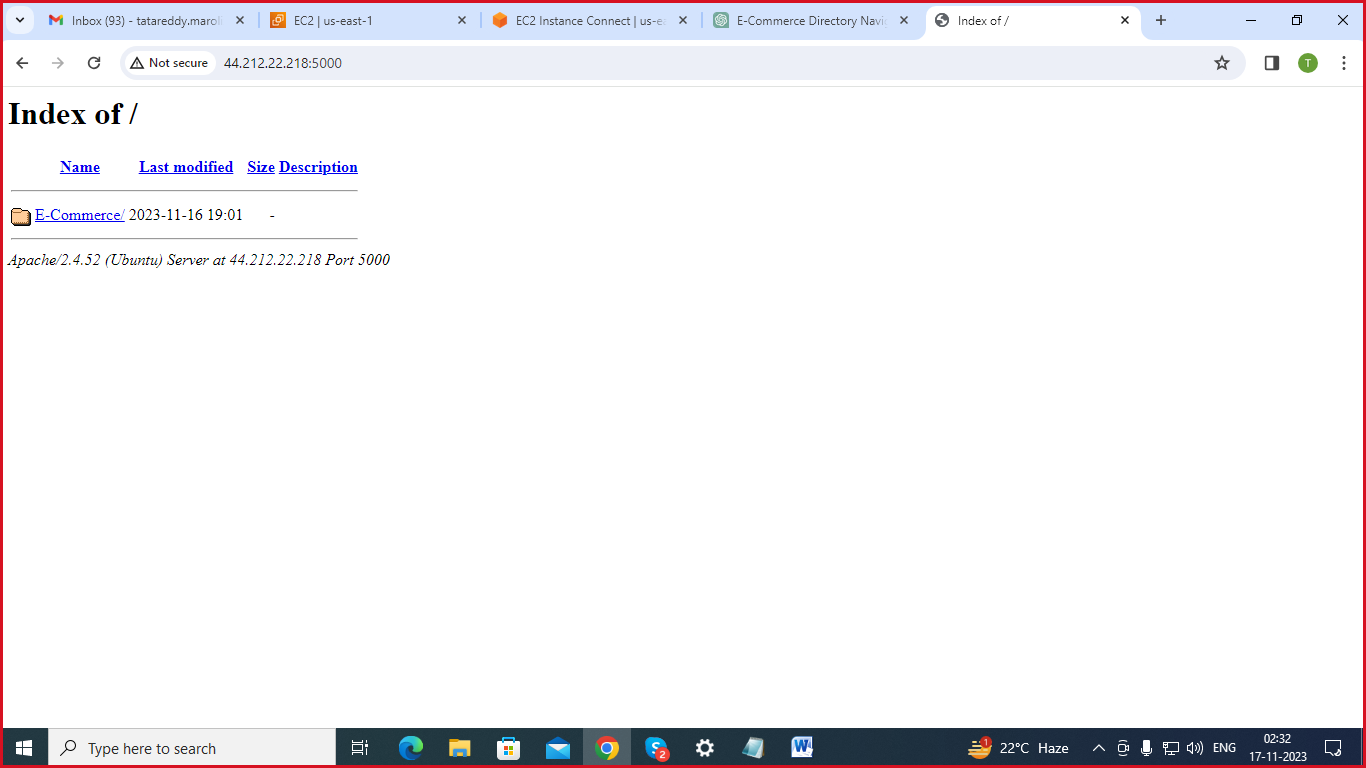
****

* **docker ps:** Lists the Docker containers on the system.

****

* **take the instance public IP:5000**
* **URL:**  **http://44.212.22.218:5000/**

**OUTPUT:**

****