**Experiment 9:Deploying a ReactJS application using Docker**

* **Open Git Bash where you application is present.**
* **Now to push the application into git run the following commands**

git init

git add .

git commit -m “my first commit”

git branch -M main

git remote add origin <https://github.com/Ashrith13/cc-portfolio1.git>

git push -u origin main

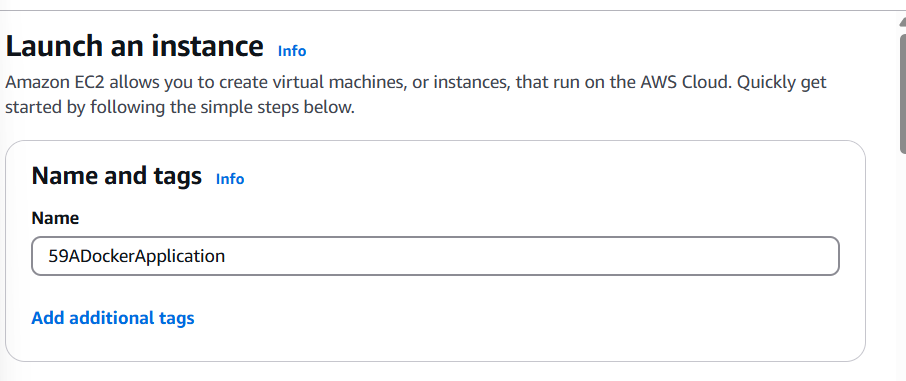


**Launching an EC2 Instance**

* Navigate back to your AWS Console, go to EC2 services and click on Launch Instance

A screenshot of a computer

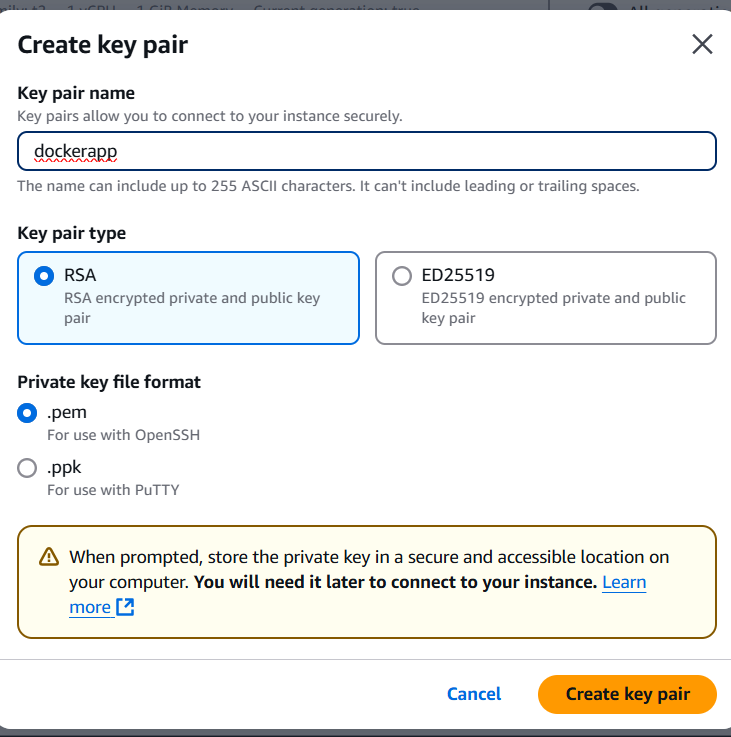
AI-generated content may be incorrect.

* Giving the name of the instance
* 

8adockerapplicatiom

A screenshot of a computer

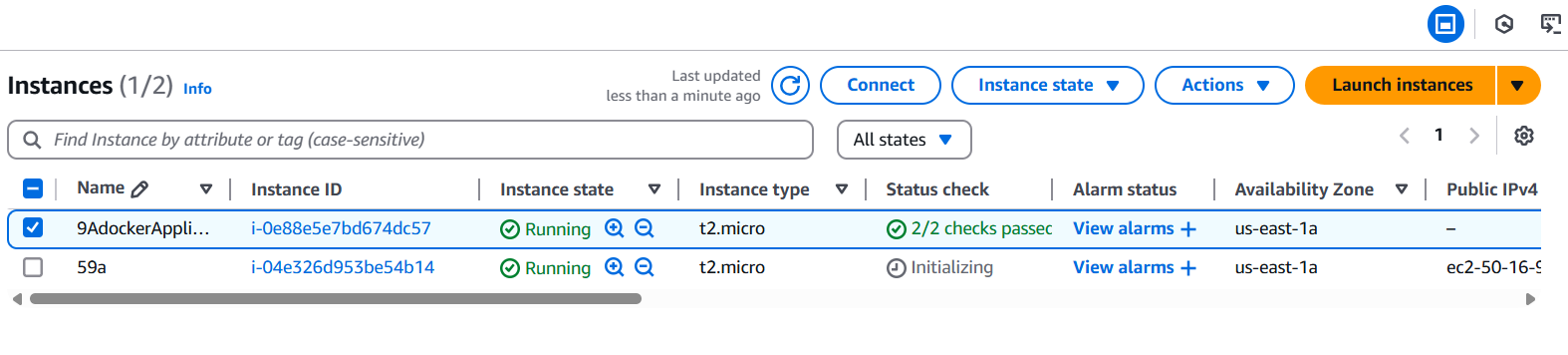
AI-generated content may be incorrect.

* **Creating a pem key pair called dockerapp**
* **Click on “Edit” next to Network Settings, select the VPC created, and select the subnet as web-subnet and Enable the Auto-assign public IP**

A screenshot of a computer

AI-generated content may be incorrect.

Select the instance created and click on Connect



A screenshot of a computer

AI-generated content may be incorrect.

After connecting run the following commands

A black screen with white text

AI-generated content may be incorrect.

* **yum update -y** updates all the packages to their latest versions.

A screen shot of a computer

AI-generated content may be incorrect.

* **yum install git -y** Installs the git package the -y flag automatically confirms the installation without asking for user input.

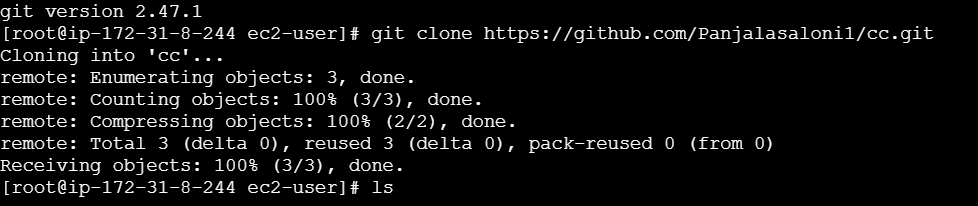
A black screen with white text

AI-generated content may be incorrect.

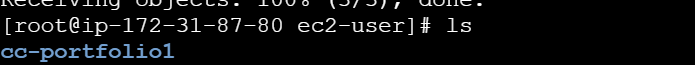
A black background with white text

AI-generated content may be incorrect.

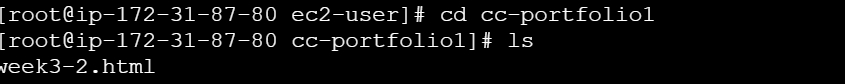
* **git clone https://github.com/Ashrith13/cc.git**Thiscreates a local copy of the remote repository from GitHub onto your computer.



* Run **ls** command to view the files cloned



Get into the directory



* **yum install docker -y** installs Docker and the -y flag automatically confirms the installation without asking for user input.

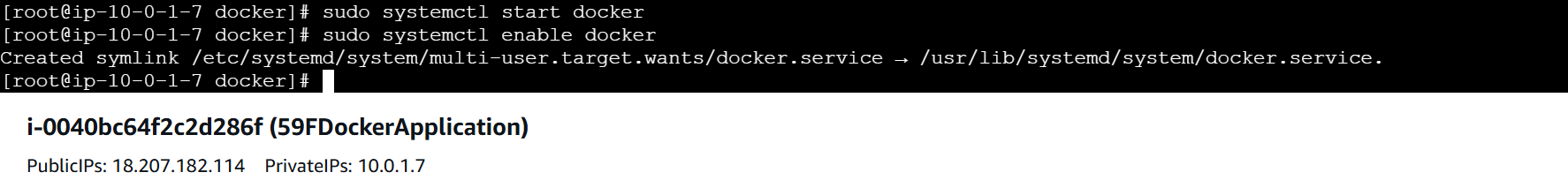
A black screen with white text

AI-generated content may be incorrect.





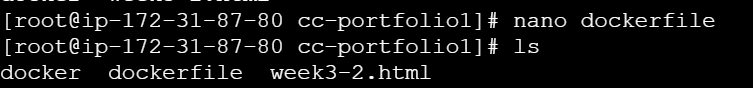
* **sudo systemctl start docker** starts the Docker service on your system, allowing you to run Docker containers and use Docker commands.
* **sudo systemctl enable docker** configures the Docker service to start automatically when your system boots up, ensuring that Docker is running whenever your machine starts.



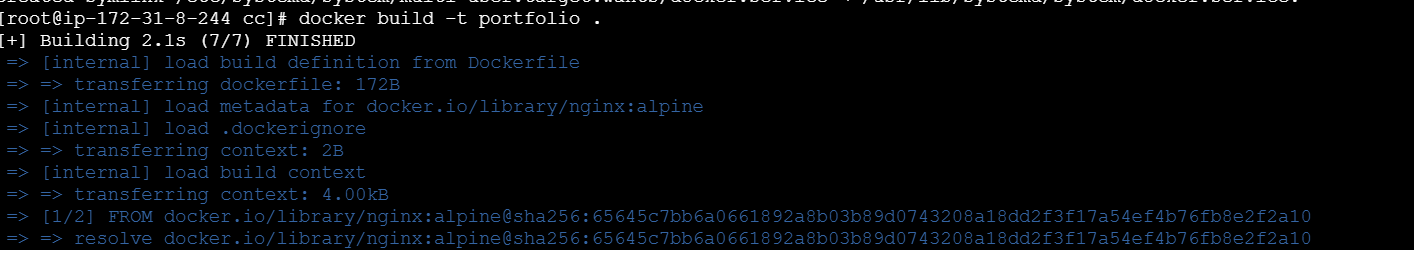
* Run the command **nano dockerfile** and write the following in it:

FROM nginx:alpine

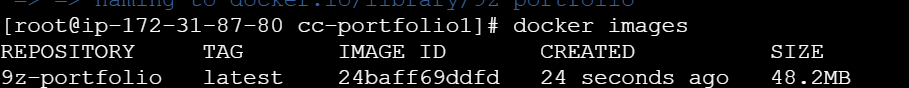
COPY week3-2.html /usr/share/nginx/html/index.html



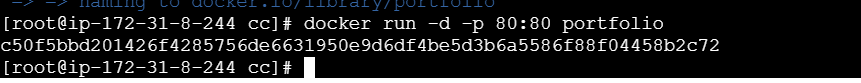
* **docker build -t 8A-portfolio .** This will build a Docker image from the Dockerfile in the current directory (.) and tags it as my-portfolio.



* **docker images** This will lists all the Docker images available on your system.



* **sudo docker run -d -p 80:89 8A-portfolio** runs a Docker container in detached mode (-d), mapping port 80 of the host to port 80 of the container, using the 9f-portfolio image
* **docker ps -a** This command will lists all containers on your system — both running and stopped ones.



* Copy the public IPV4 address of your instance and paste in it another tab.

