Weather App

Project Documentation

Contents

Introduction

Weather App is Angular Based web application which can be used to get latest weather information like temperature, min and max temperature, humidity and wind. Users can search for their city weather by entering on search city text field.

Technologies Used

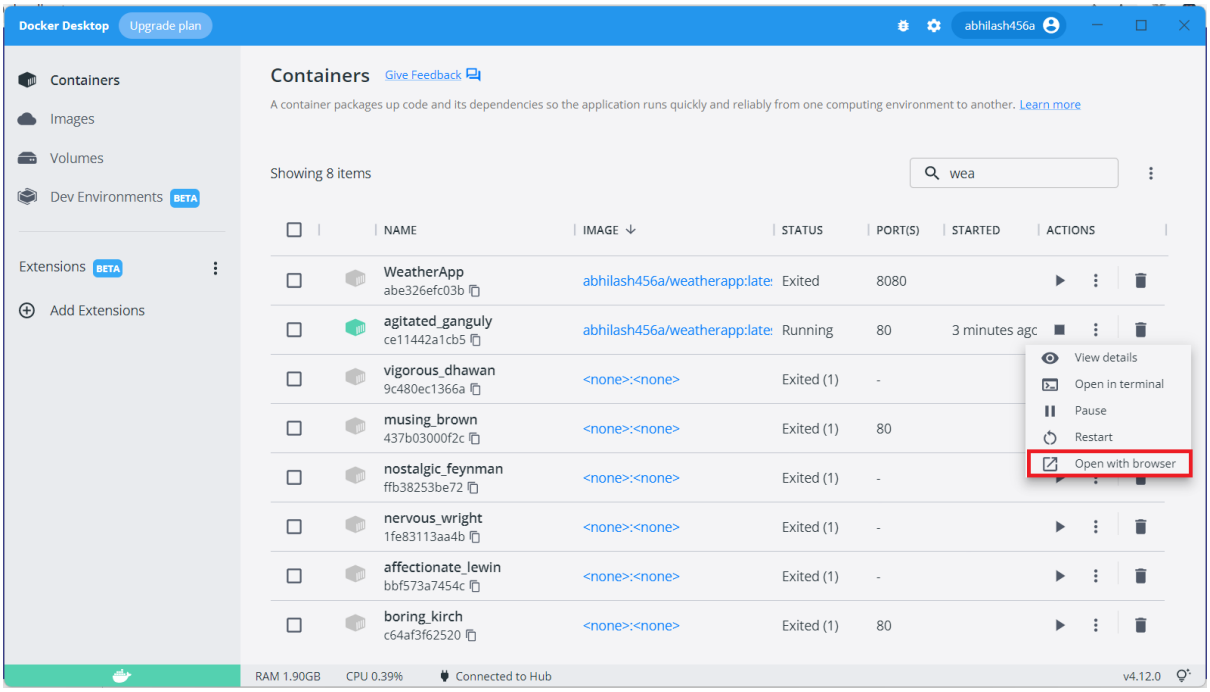
- Visual Studio
- Angular
- AWS EC2 Instance
- Docker
- Node JS
- GitHub

Installation Instructions

Users need to install docker on their machine and run the following command

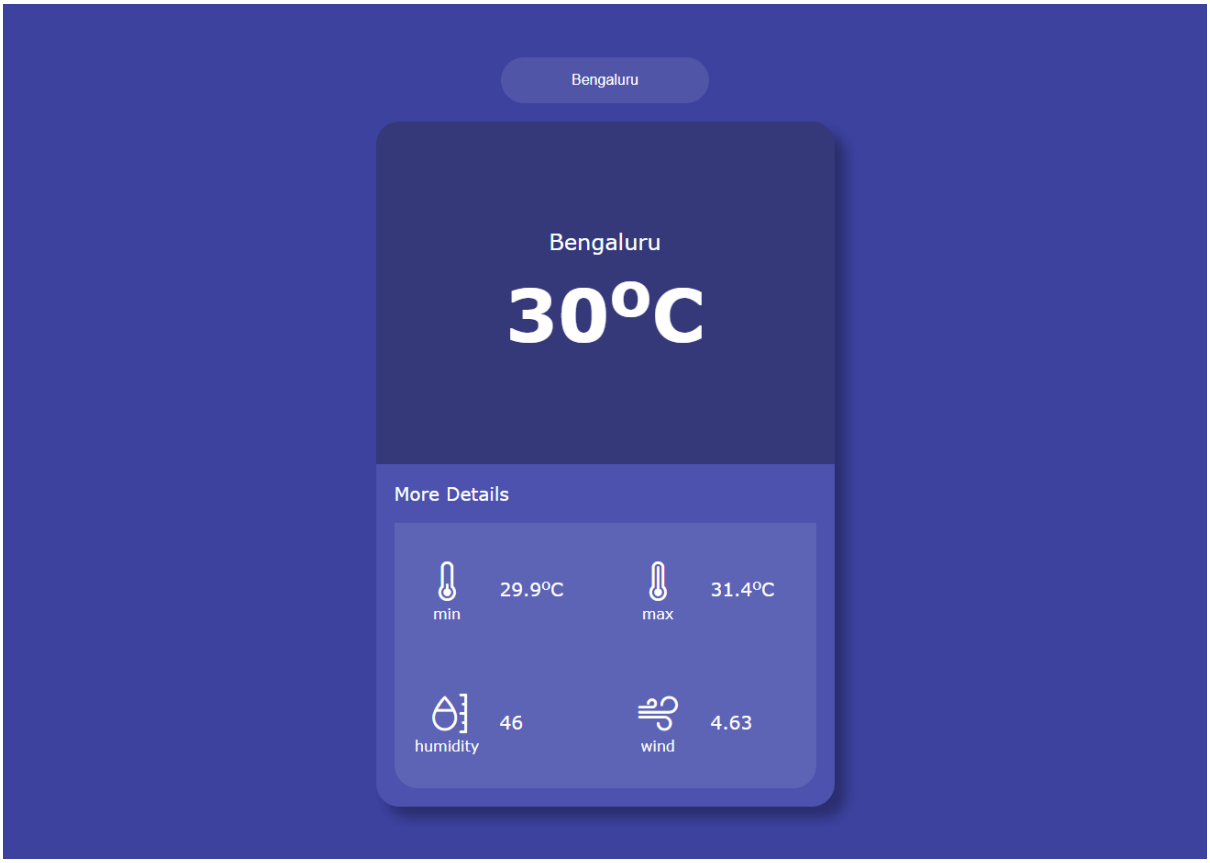
docker run -d -p 80:80 abhilash456a/weatherapp:latest

Users can open the weatherapp by going to docker desktop and selecting the container that's running and click on vertical dot menu and select open in browser option to view inside the browser.



To Download the source users can download from the github repository at https://github.com/abhilashlegend/weatherapp.git . After downloading into your computer. Run npm install to download the packages and dependencies. And Run ng serve to launch on the browser. The angular app is usually hosted at http://localhost:4200/

Home Page



Weather App is a single page application that displays weather details. On the home page you can see the details of default city that is Bengaluru. The main temperature is 30 degrees Celsius. At the bottom there is min and max temperature of the city, humidity and wind. User can change the city by clicking on the top text field and entering.

Dockerizing Angular App

Steps to dockerizing the angular App

1. Install Docker Desktop
2. Create Dockerfile in the project root folder.

```
# Use an official Node.js runtime as the base image
FROM node:latest as build

# Set the working directory in the container
WORKDIR /usr/local/app

# Copy package.json and package-lock.json to the container
COPY ./ /usr/local/app/

# Install application dependencies
RUN npm install

# Generate the build of the application
RUN npm run build

# Stage 2: Serve app with nginx server

# Use official nginx image as the base image
FROM nginx:latest

# Copy the build output to replace the default nginx contents.
COPY --from=build /usr/local/app/dist/weatherapp
/usr/share/nginx/html

# Expose port 80
EXPOSE 80

```

3. Run ng build command to build the angular application
4. Use the following command to generate the Docker image for the Angular application using Dockerfile:
   Example: docker build -t dockerhub_name/image_name:tag dockerfile_location

   docker build -t abhilash456a/weatherapp:latest .

5. Get the list of Docker images using the following command:

   Docker image ls

6. You need to push the Docker image to Docker Hub or any container registry(AWS ECR, Azure CR) if you want to deploy the application on the Cloud server.
   To push the Docker image to the Docker hub you need a Docker hub account. Once you created a Docker hub account, then log in to the Docker hub on your terminal.

   docker login

7. Use the following command to push the Docker image to Docker Hub:

   docker push abhilash456a/weatherapp:latest

8. Run Docker Container
   Run the Angular application using the following command

   docker run -d -p 80:80 abhilash456a/weatherapp:latest

It runs on port number 80. Access the Angular application using the IP address and port number.
http://localhost:80/

9. list the containers by below command->
docker ps

Deploying on AWS EC2 Instance

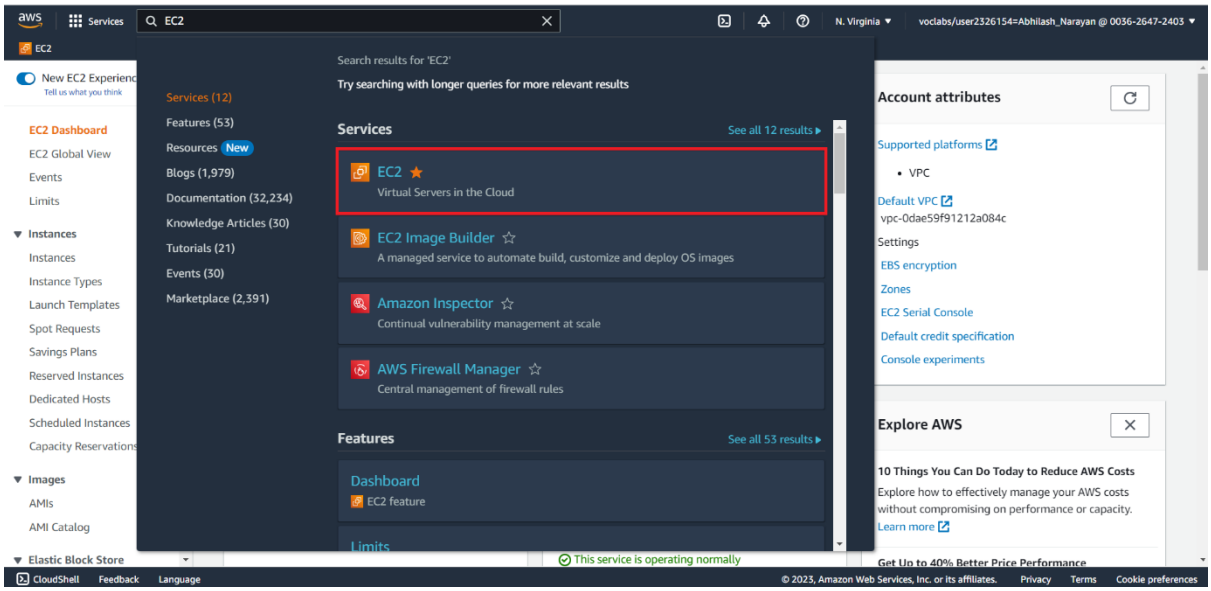
Steps to deploy on AWS EC2 Instance

1. open https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

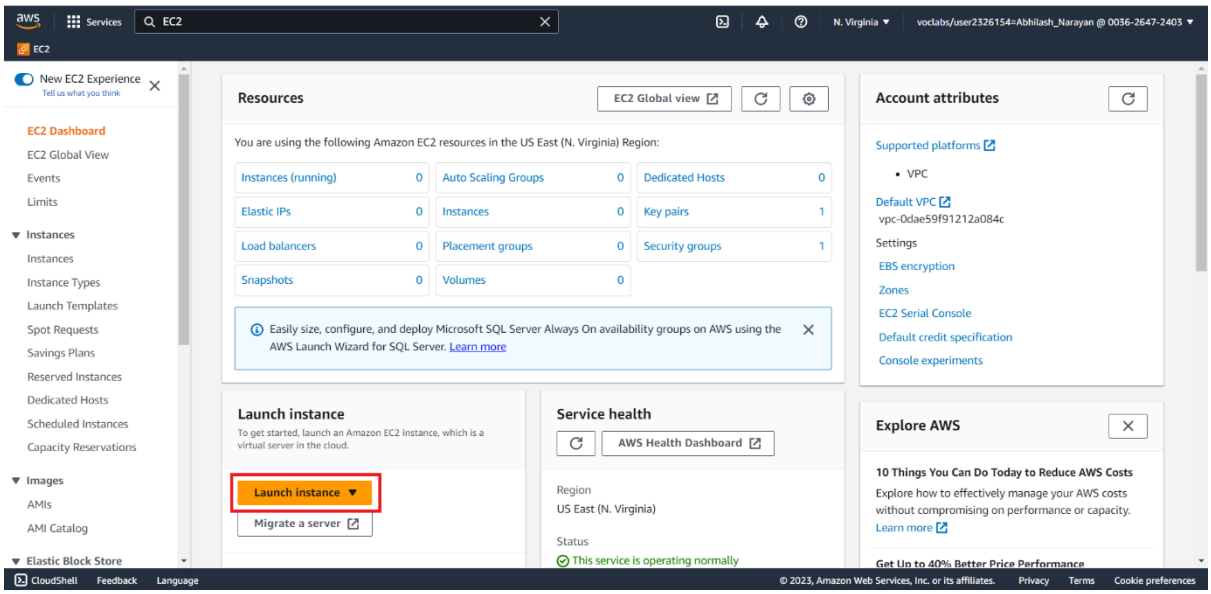
2. Click on Linux/MacOS or Windows as per your Operating system
   For Windows:
   Download and run the AWS CLI MSI installer for Windows (64-bit)

3. To confirm the installation, open the Start menu, search for cmd to open a command prompt window, and at the command prompt use the aws --version command.
   C:\> aws --version
   o/p-> aws-cli/2.10.0 Python/3.11.2 Windows/10 exe/AMD64 prompt/off

4. Go to AWS and login to your account and select EC2 Instance
   Click services -> EC2 (or search for it if unable to find)

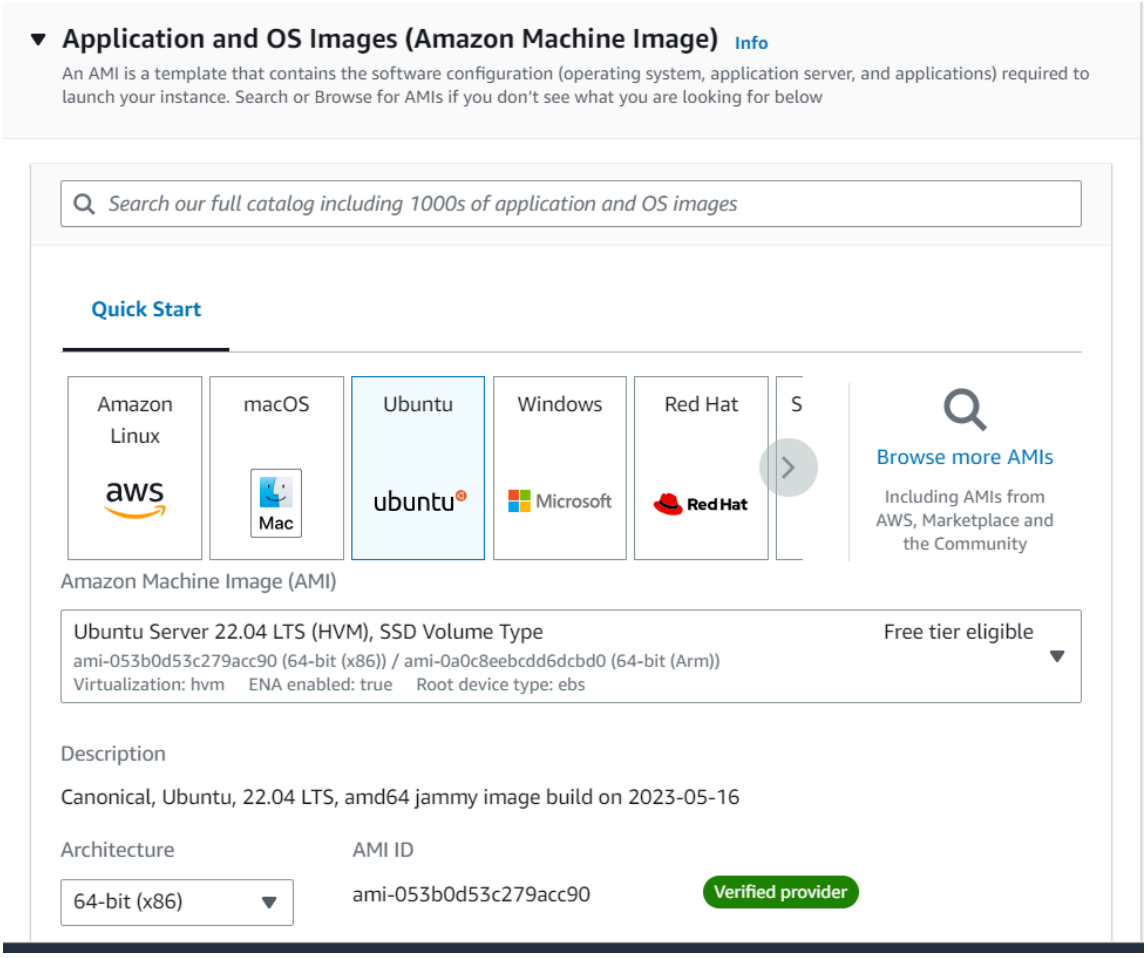


5. Click EC2 dashboard->Launch instance

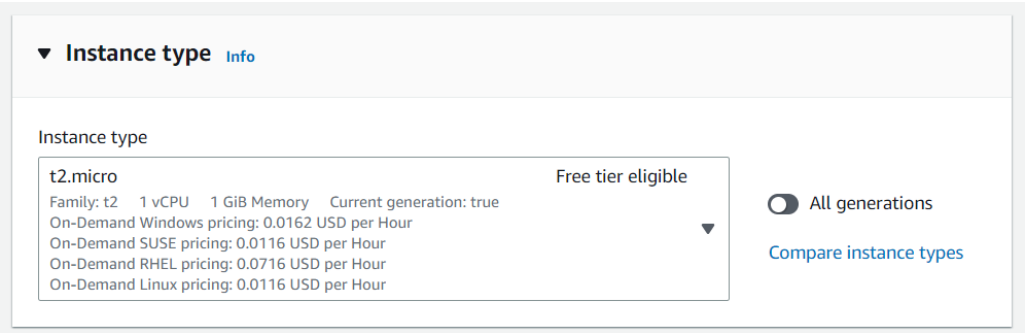


a. Give name as weatherapp

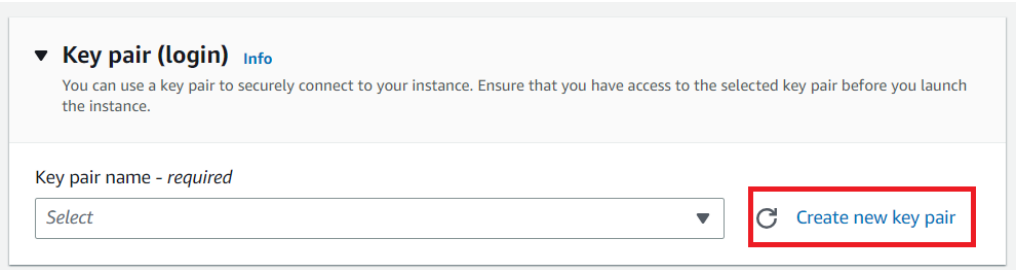
b. In Application and OS Images select Ubuntu
Amazon Machine Image -> Ubuntu Server 22.04 LTS (HVM), SSD Volume Type (Free Tier eligible)



c. Instance type->t2.micro (free tier eligible)

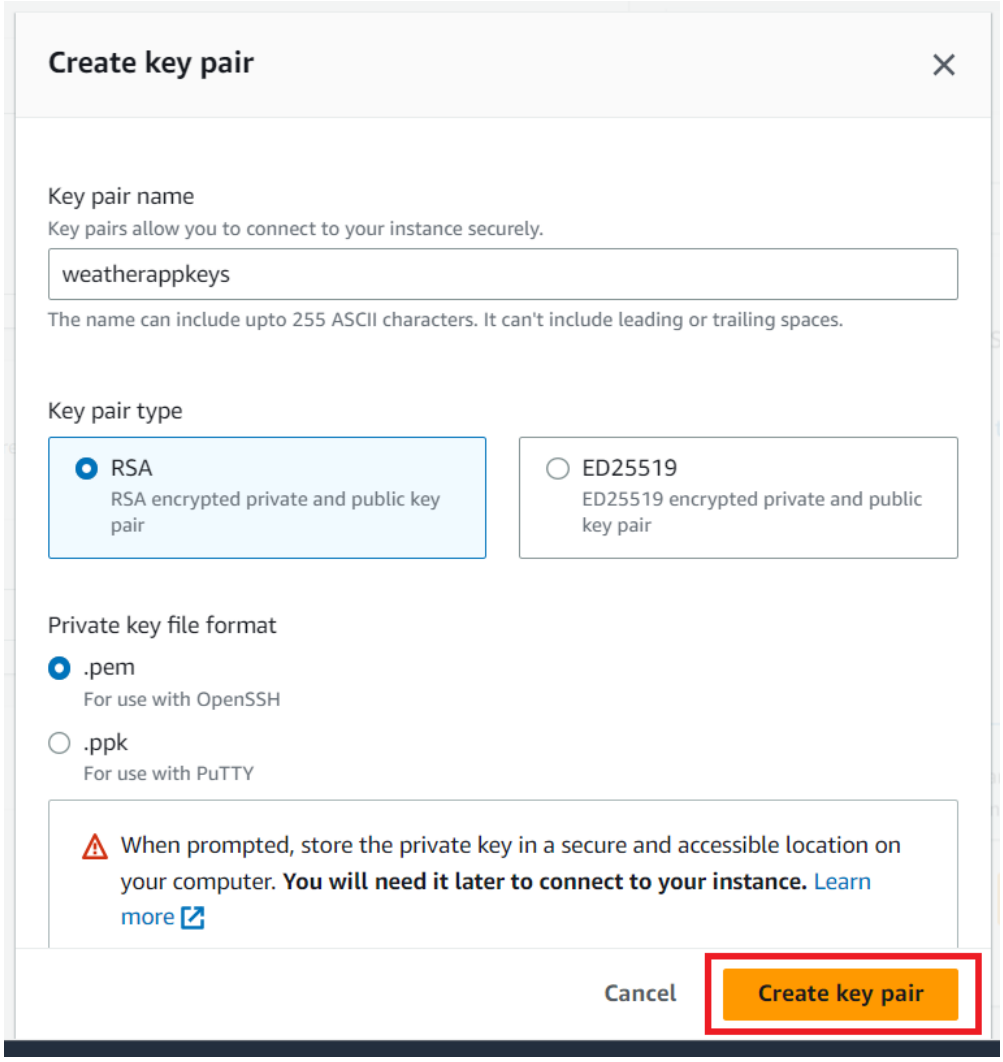


d. Keypair (login)



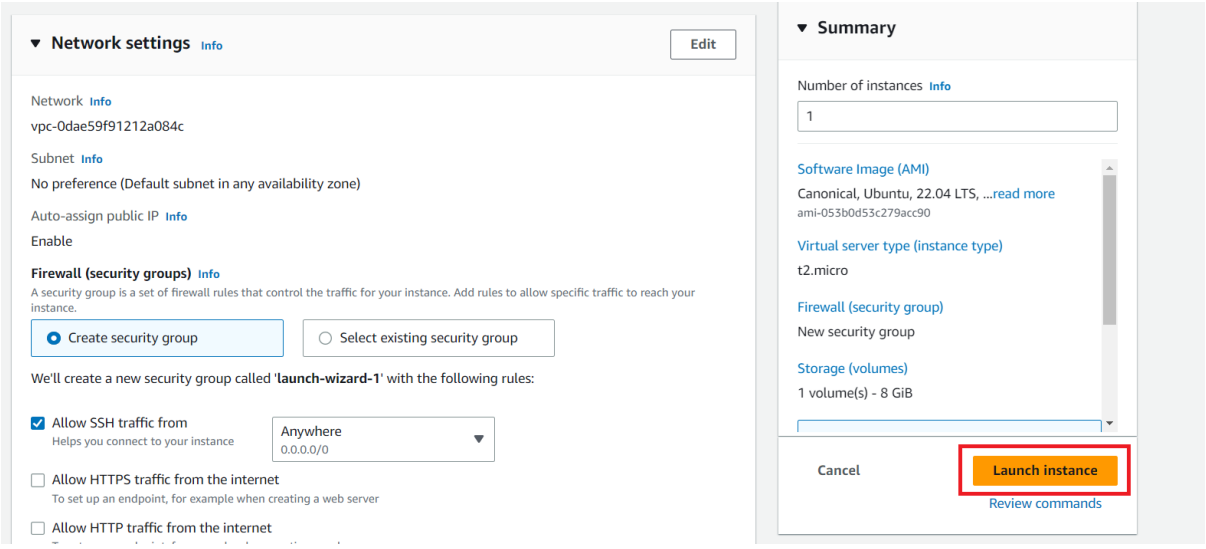
A create key pair window will open.

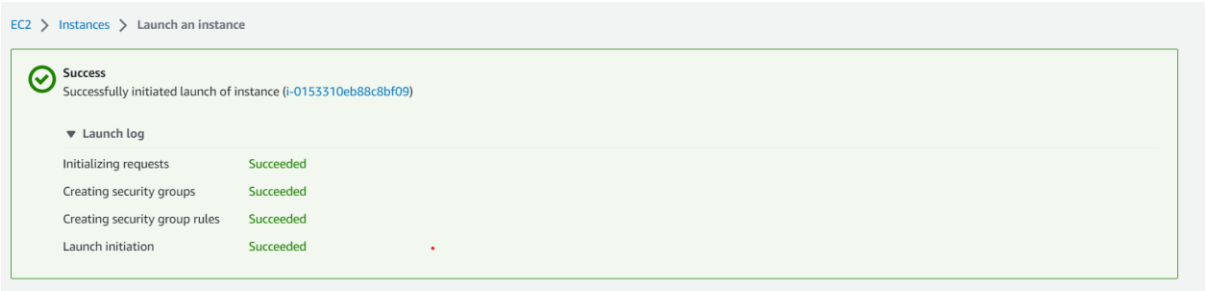
- Enter key pair name -> weatherappkeys.
- Select Key pair type -> RSA
- Private key file format -> .pem
- Click create new Key pair



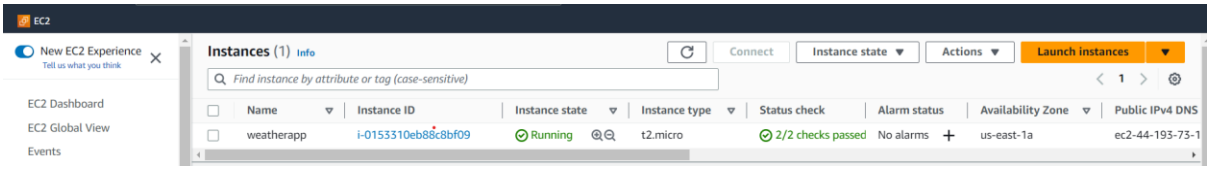
The keys file(weatherappkeys.pem) will get downloaded in your system, you will need them later to connect to EC2 instance.

6. create a folder in c: by name weatherappkeys. Save the weatherappkeys.pem file in c:/weatherappkeys
7. Click Launch Instance(No changes required for other settings)

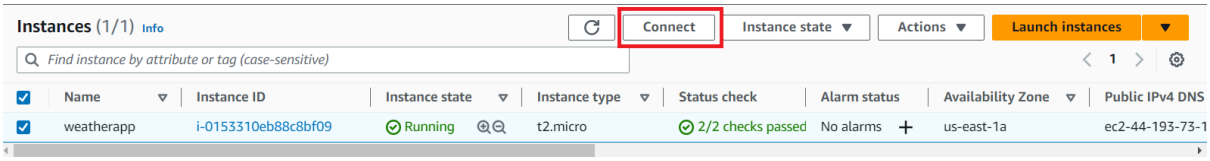


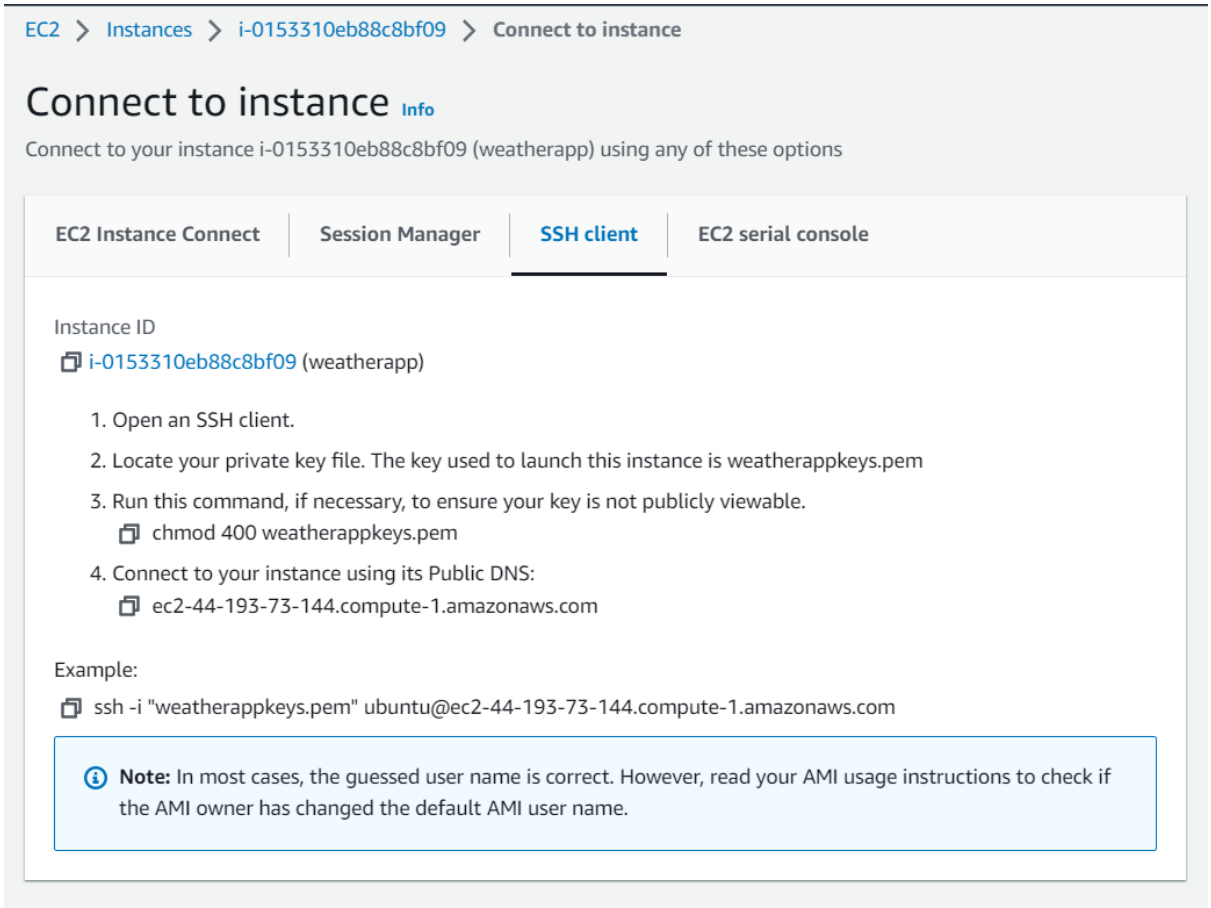


8. Click on instances



9. To connect this instance select checkbox before this row and click on connect





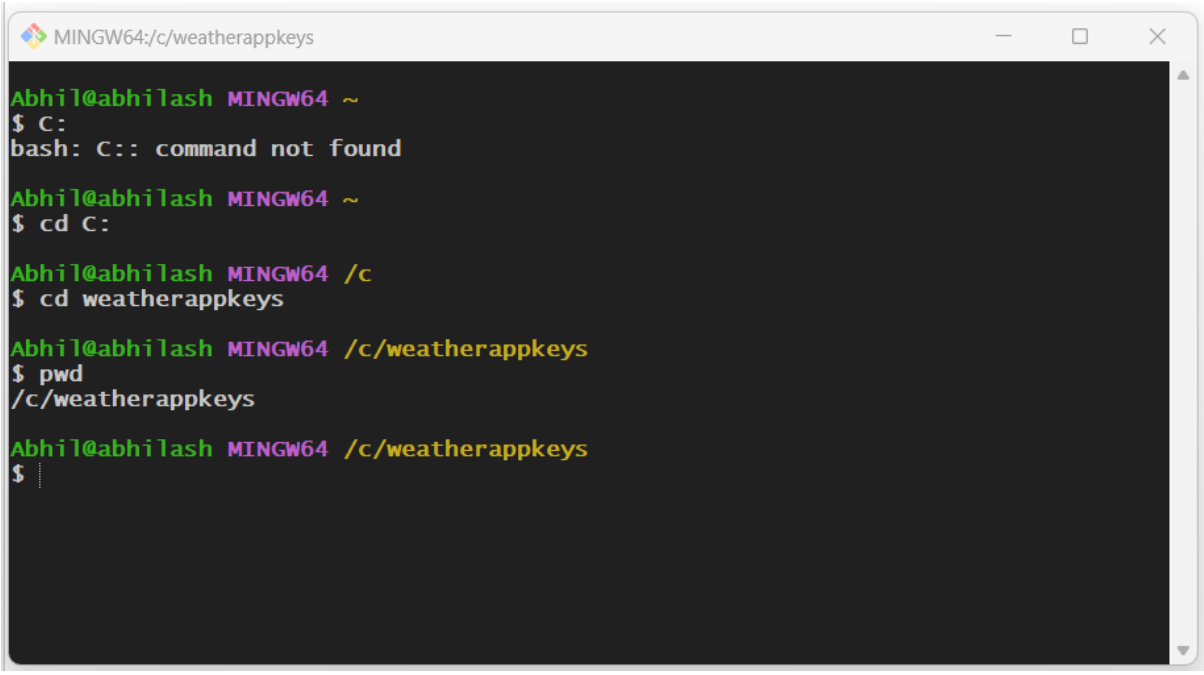
Click on SSH client copy the command
copy the command below Example:

ssh -i "weatherappkeys.pem" ubuntu@ec2-44-193-73-144.compute-1.amazonaws.com

10. Open git bash(you can search for it in search area next to start menu) browse to location where u saved the keys i.e c:/weatherappkeys by

> cd c:
> cd weatherappkeys

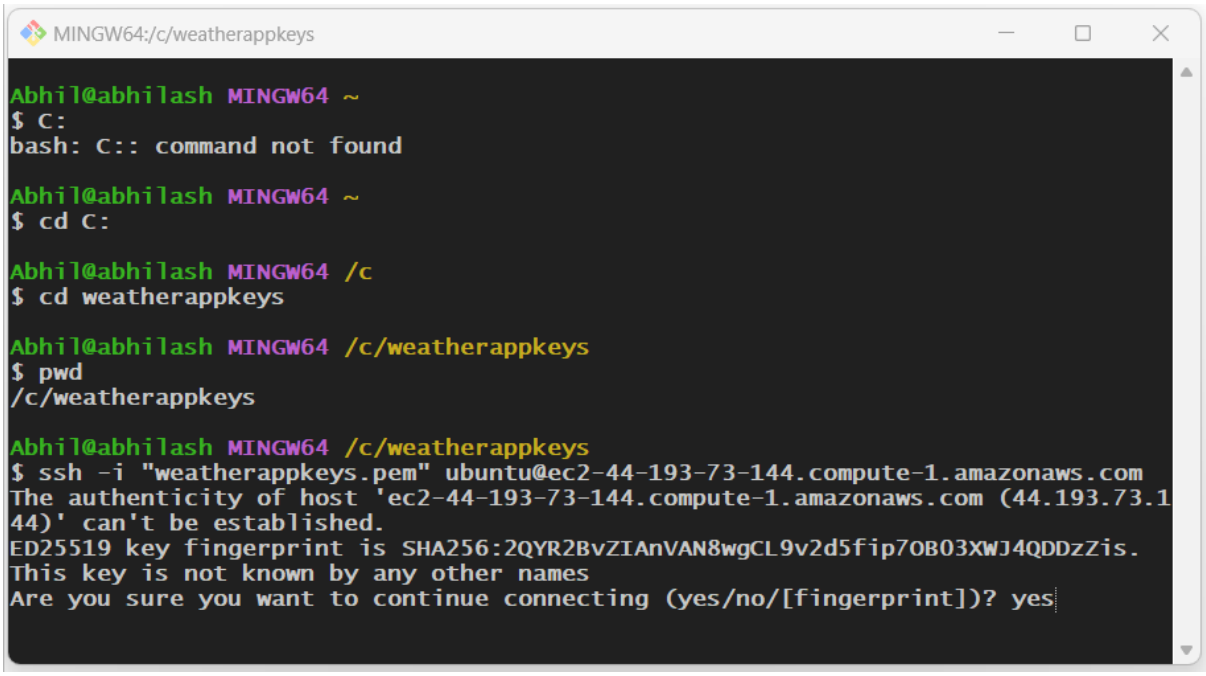
To check ur current location
>pwd



11. Now paste(use right click paste as ctrl v will not work) the command copied earlier

    ssh -i "weatherappkeys.pem" ubuntu@ec2-44-193-73-144.compute-1.amazonaws.com

    Press enter



```
MINGW64:/c/weatherappkeys

Abhil@abhilash MINGW64 ~
$ cd C:

Abhil@abhilash MINGW64 /c
$ cd weatherappkeys

Abhil@abhilash MINGW64 /c/weatherappkeys
$ pwd
/c/weatherappkeys

Abhil@abhilash MINGW64 /c/weatherappkeys
$ ssh -i "weatherappkeys.pem" ubuntu@ec2-44-193-73-144.compute-1.amazonaws.com
The authenticity of host 'ec2-44-193-73-144.compute-1.amazonaws.com (44.193.73.1
44)' can't be established.
ED25519 key fingerprint is SHA256:2QYR2BvZIAnVAN8wgCL9v2d5fip7OBO3XWJ4QDDzZis.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-193-73-144.compute-1.amazonaws.com' (ED25519)
 to the list of known hosts.
Connection closed by 44.193.73.144 port 22

Abhil@abhilash MINGW64 /c/weatherappkeys
$
```

12. Again give same command i.e.

```
Abhil@abhilash MINGW64 /c/weatherappkeys
$ ssh -i "weatherappkeys.pem" ubuntu@ec2-44-193-73-144.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Jun 15 15:16:23 UTC 2023

  System load:  0.0               Processes:             114
  Usage of /:   20.6% of 7.57GB   Users logged in:       0
  Memory usage: 27%               IPv4 address for eth0: 172.31.8.25
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-8-25:~$
```

13.Run the following command to install NGINX

```
sudo -s - for super user
sudo apt update - to update the existing packages
sudo apt install nginx - to install the nginx web server
```

14.Check if git is installed

git --version

To install git in EC2 virtual machine->
sudo apt install git -y

To check if git is successfully installed run the command again
git –version

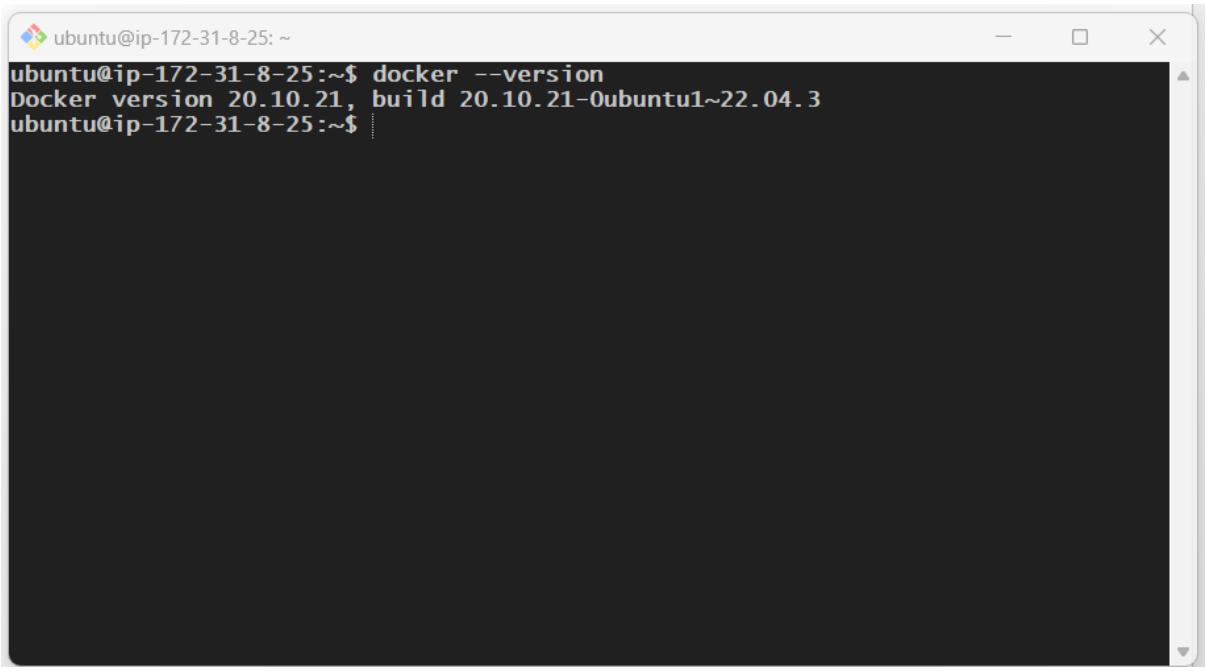
15.Install docker in this virtual EC2 machine

// install most recent package

sudo apt install docker.io

Check if docker was installed by running the command

docker --version



16. Start the service docker
sudo service docker start

17. Pull the docker image
sudo docker pull abhilash456a/weatherapp

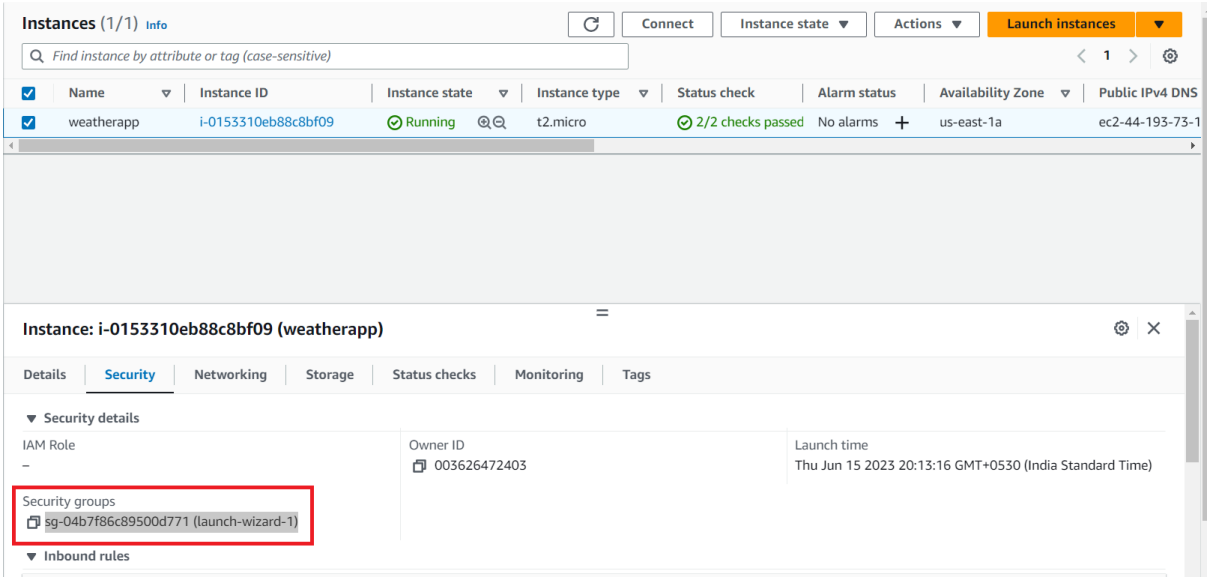
```
ubuntu@ip-172-31-8-25: ~
ubuntu@ip-172-31-8-25:~$ docker --version
Docker version 20.10.21, build 20.10.21-0ubuntu1~22.04.3
ubuntu@ip-172-31-8-25:~$ sudo service docker start
ubuntu@ip-172-31-8-25:~$ sudo docker pull abhilash456a/weatherapp
Using default tag: latest
latest: Pulling from abhilash456a/weatherapp
f03b40093957: Pull complete
eed12bbd6494: Pull complete
fa7eb8c8eee8: Pull complete
7ff3b2b12318: Pull complete
0f67c7de5f2c: Pull complete
831f51541d38: Pull complete
9dd8b0c4050e: Pull complete
Digest: sha256:02176b9900427f81cf97221631d24a2a7dd50acfa01024237316c9906f26a640
Status: Downloaded newer image for abhilash456a/weatherapp:latest
docker.io/abhilash456a/weatherapp:latest
ubuntu@ip-172-31-8-25:~$
```

18. List the images

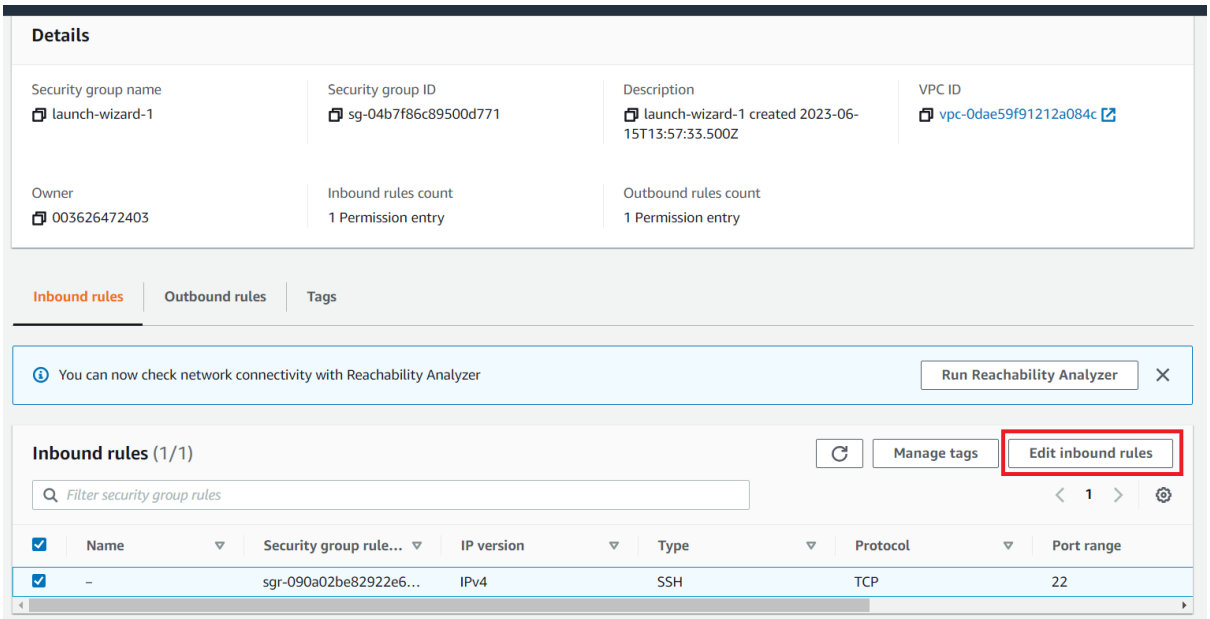
    sudo docker images

```
ubuntu@ip-172-31-8-25: ~
ubuntu@ip-172-31-8-25:~$ docker --version
Docker version 20.10.21, build 20.10.21-0ubuntu1~22.04.3
ubuntu@ip-172-31-8-25:~$ sudo service docker start
ubuntu@ip-172-31-8-25:~$ sudo docker pull abhilash456a/weatherapp
Using default tag: latest
latest: Pulling from abhilash456a/weatherapp
f03b40093957: Pull complete
eed12bbd6494: Pull complete
fa7eb8c8eee8: Pull complete
7ff3b2b12318: Pull complete
0f67c7de5f2c: Pull complete
831f51541d38: Pull complete
9dd8b0c4050e: Pull complete
Digest: sha256:02176b9900427f81cf97221631d24a2a7dd50acfa01024237316c9906f26a640
Status: Downloaded newer image for abhilash456a/weatherapp:latest
docker.io/abhilash456a/weatherapp:latest
ubuntu@ip-172-31-8-25:~$ sudo docker images
REPOSITORY                TAG       IMAGE ID       CREATED      SIZE
abhilash456a/weatherapp   latest    1ea90565cc11   2 days ago   143MB
ubuntu@ip-172-31-8-25:~$
```

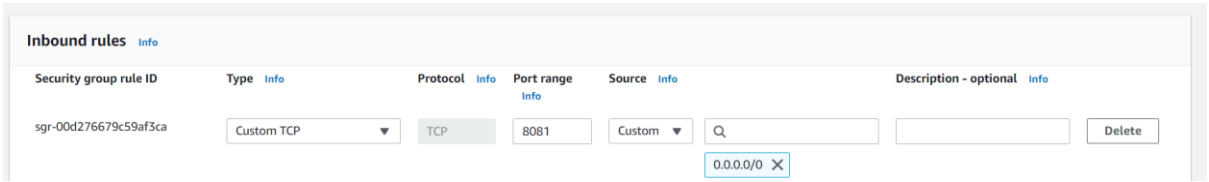
- click instance->security tab-> click security groups link sg-04b7f86c89500d771 (launch-wizard-1)



Go to inbound rules tab and click on edit inbound rules



Click Add Rule



Select Custom TCP, Port Range 8081, change custom to Anywhere-IPv4
And save changes

19. In gitbash run the container using the command
    sudo docker run -d -p 8081:80 abhilash456a/weatherapp

20. Check your public ip address and go to port 8081 Eg:
    http://54.196.220.154:8081