MediConnect – Cloud-Native Containerized Healthcare Application

Project Overview

MediConnect is a **modular healthcare web application** deployed using **AWS, Docker, EC2, S3, CloudFront, and Route 53**. The application consists of three independent backend modules:

- Patient Management System
- Doctor Management System
- Billing & Pharmacy System

The **frontend is hosted as a static website on S3**, and the **backend services are containerized and deployed on EC2 using Docker**. Traffic is served securely using **CloudFront and Route 53**.

Designed and deployed a cloud-native healthcare application with a decoupled frontend (S3 + CloudFront) and containerized backend services (Docker on EC2).

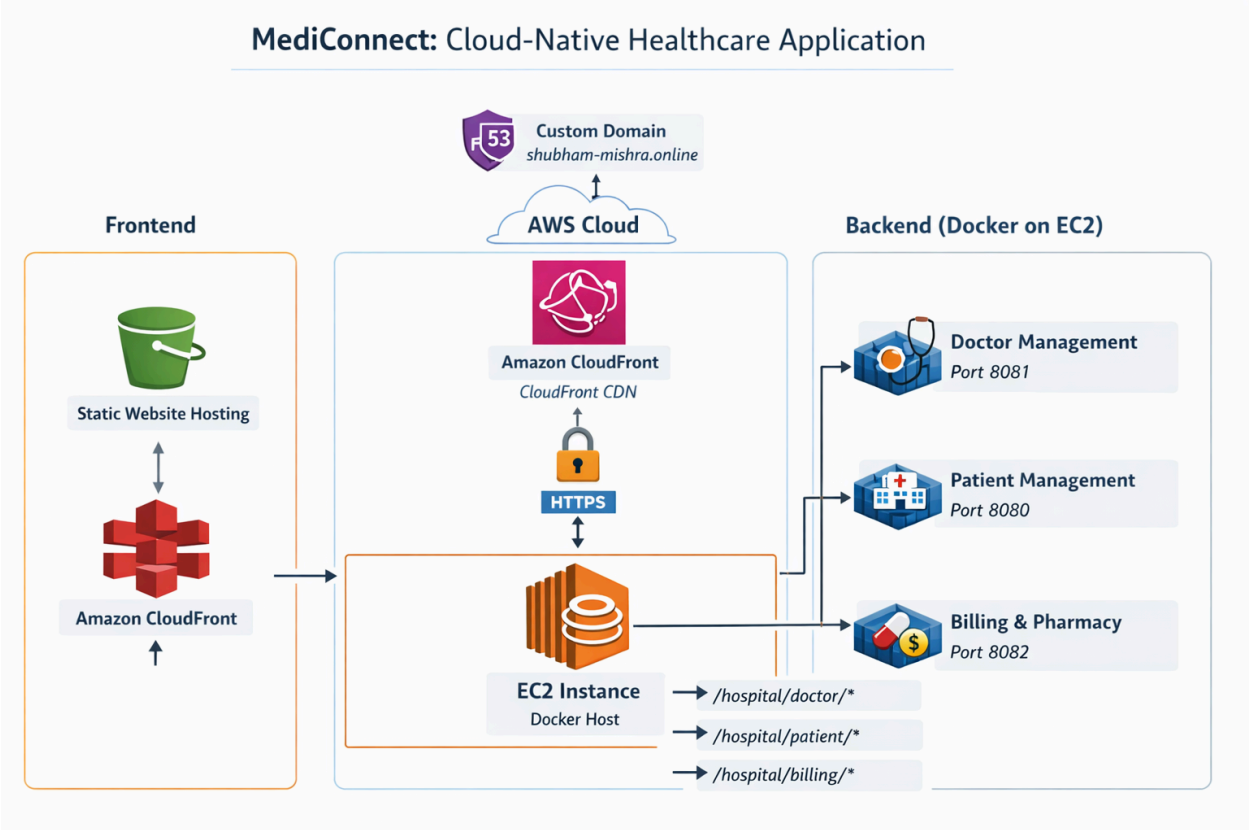
Implemented a modular microservices-style architecture with independent services for patient management, doctor management, and billing systems.

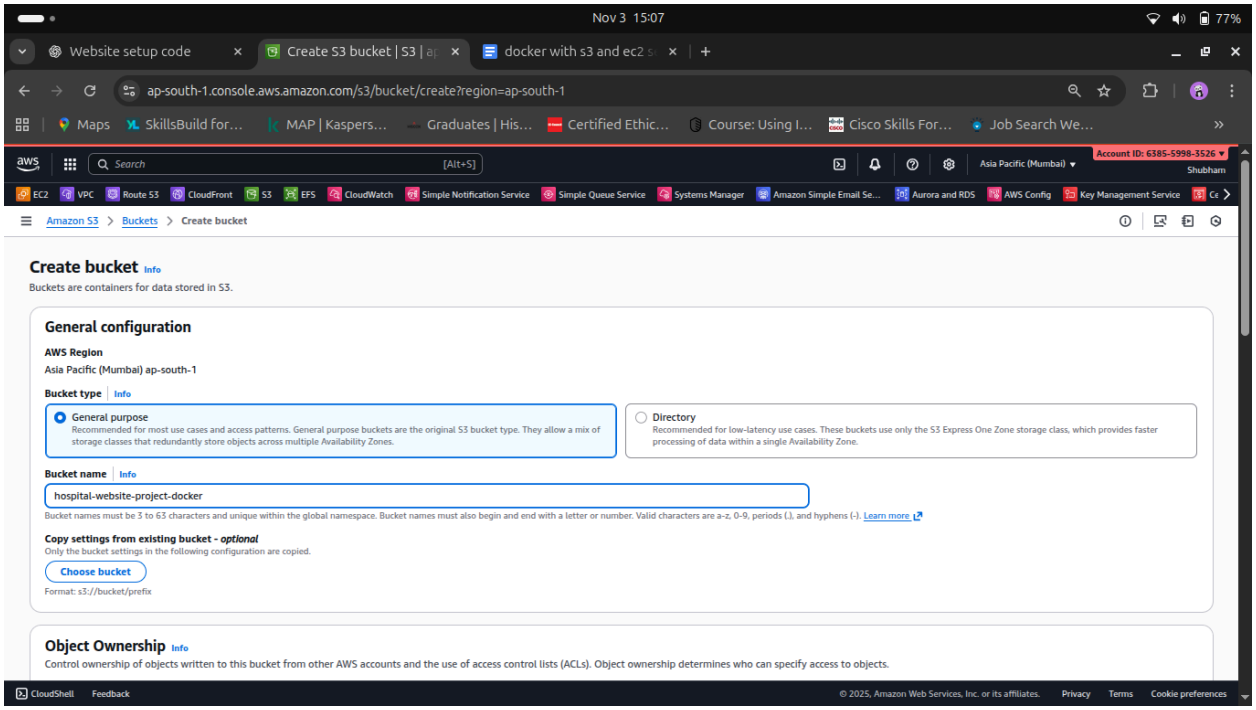
Secured static content delivery using CloudFront with Origin Access Control (OAC), keeping S3 buckets private and enforcing HTTPS.

Integrated Route 53 with CloudFront to deliver the application via a custom domain.

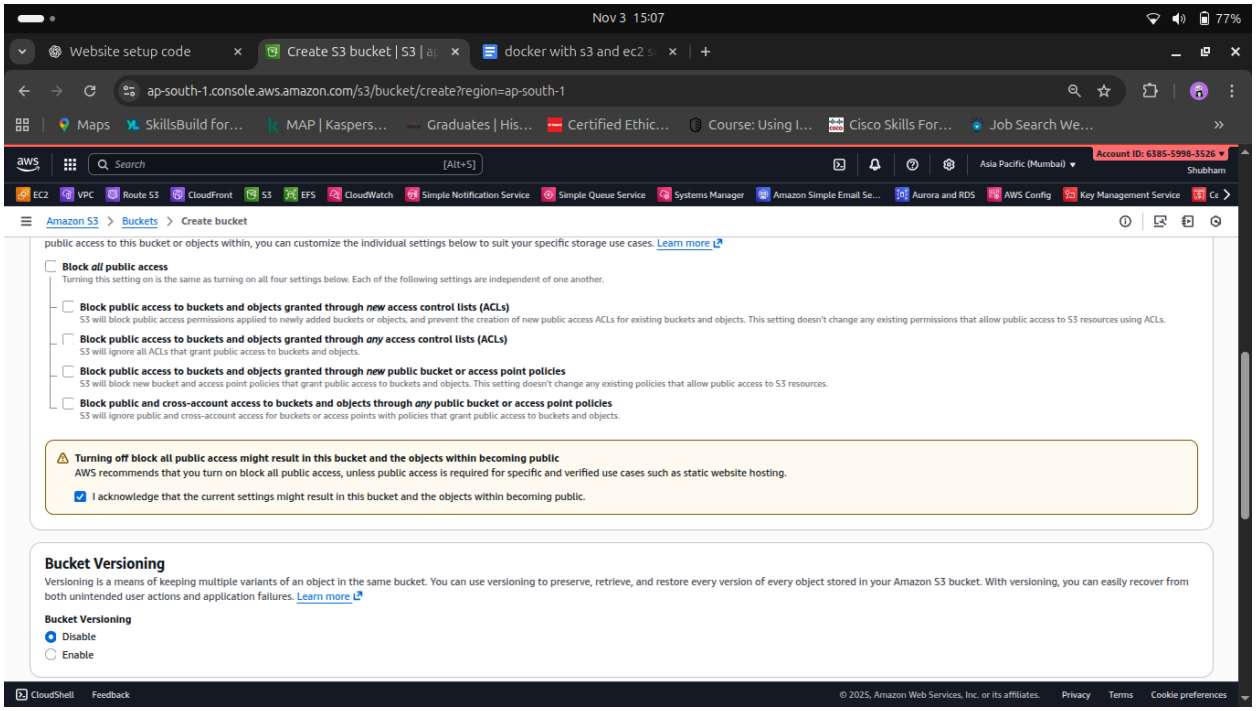
Deployed and managed multiple Docker containers on EC2, exposing services via dedicated ports and integrating them with the frontend.

Built a production-style end-to-end request flow from CDN → frontend → backend services.

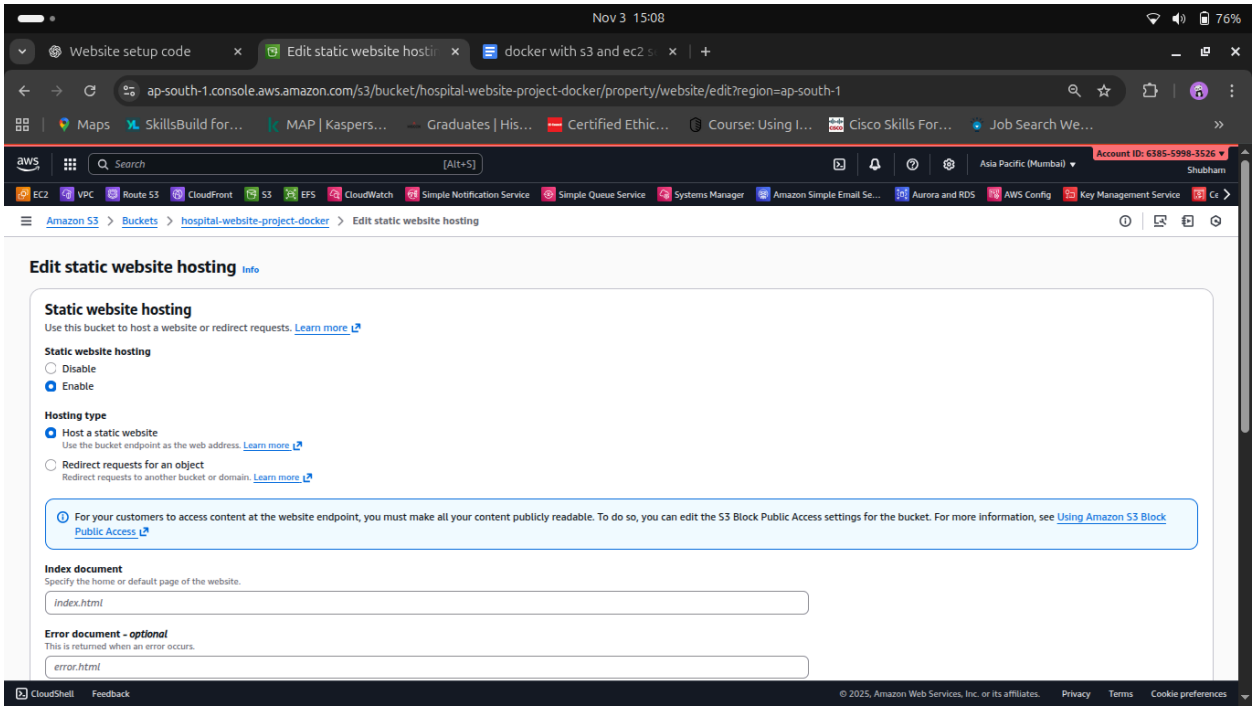




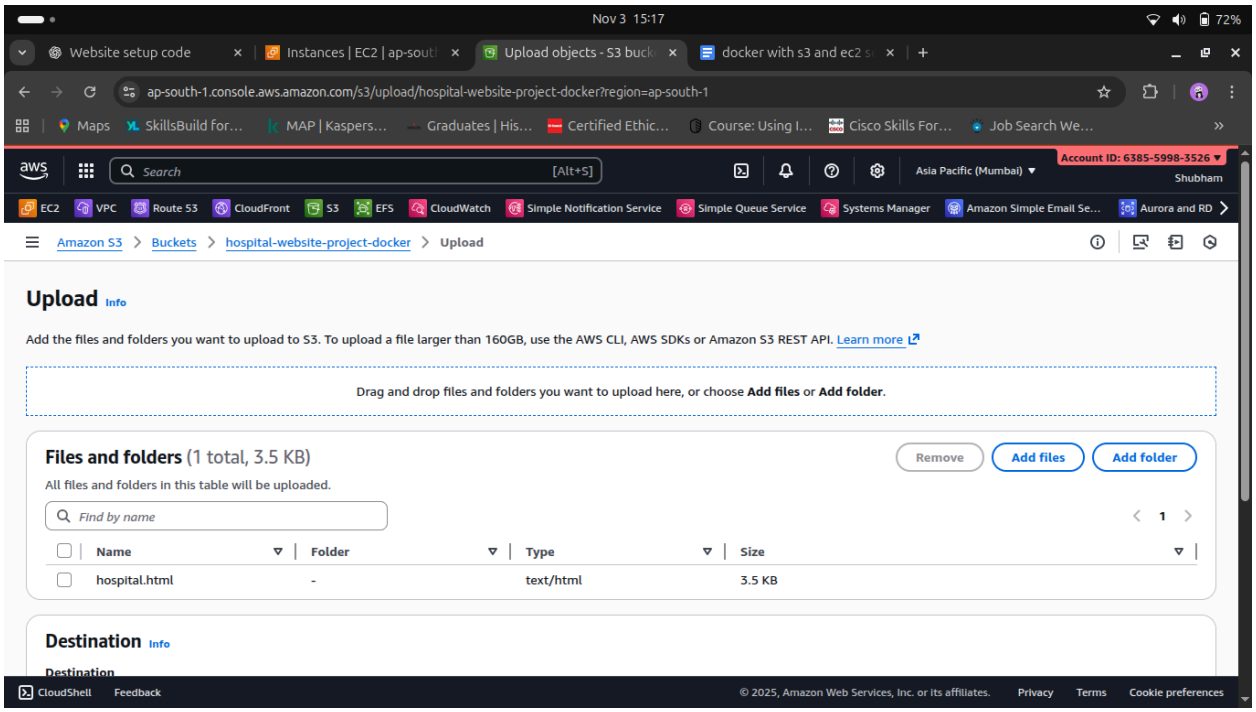
Step 1) We will create S3 bucket for static website hosting



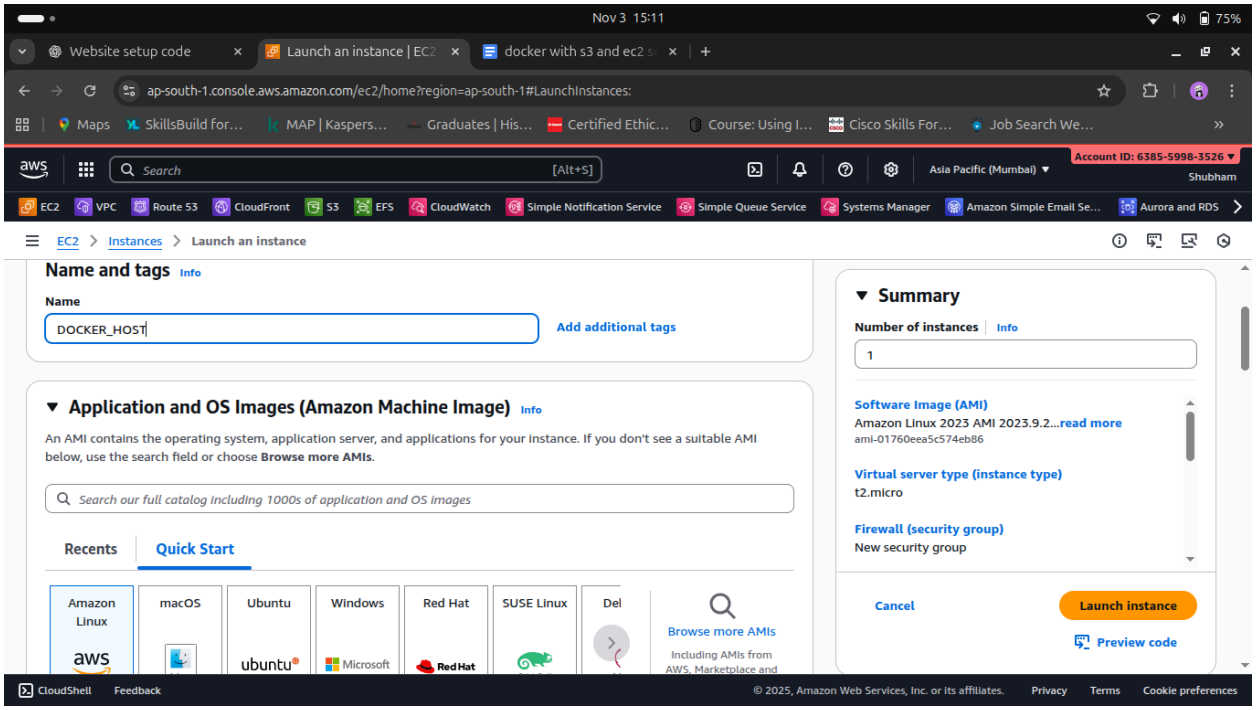
Step 2) Make sure to keep the bucket private



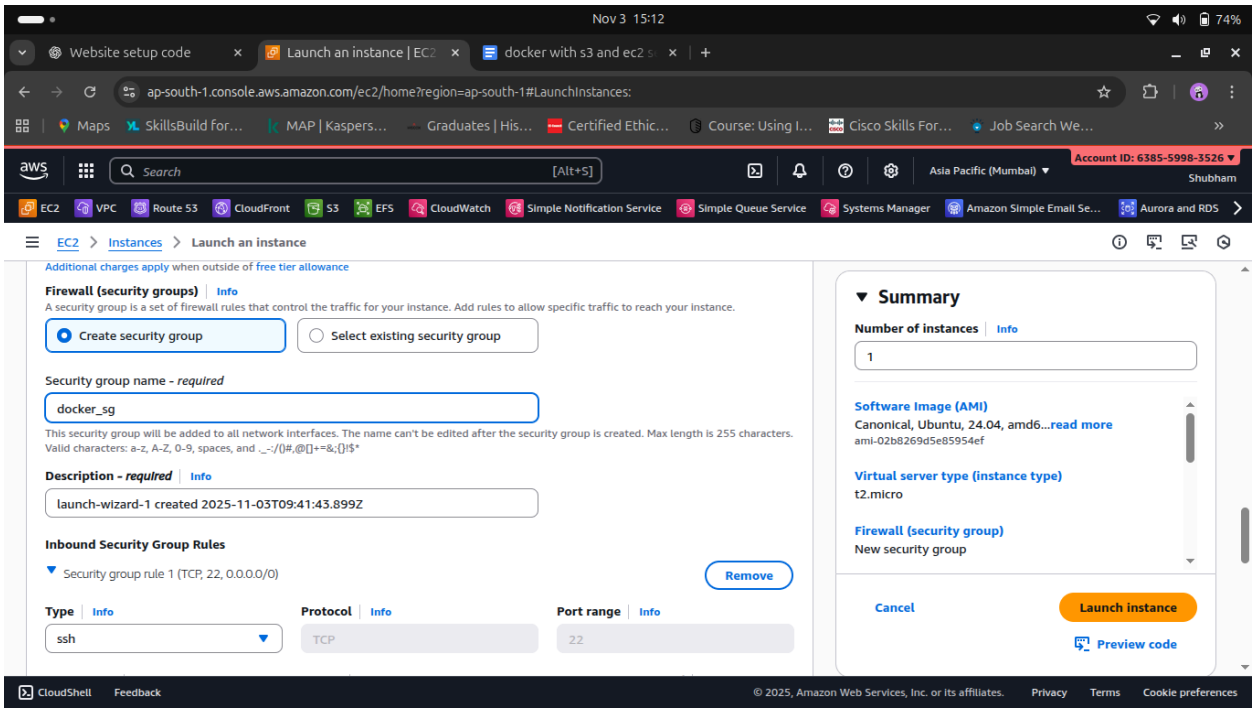
Step 3) We will enable S3 static website hosting enable

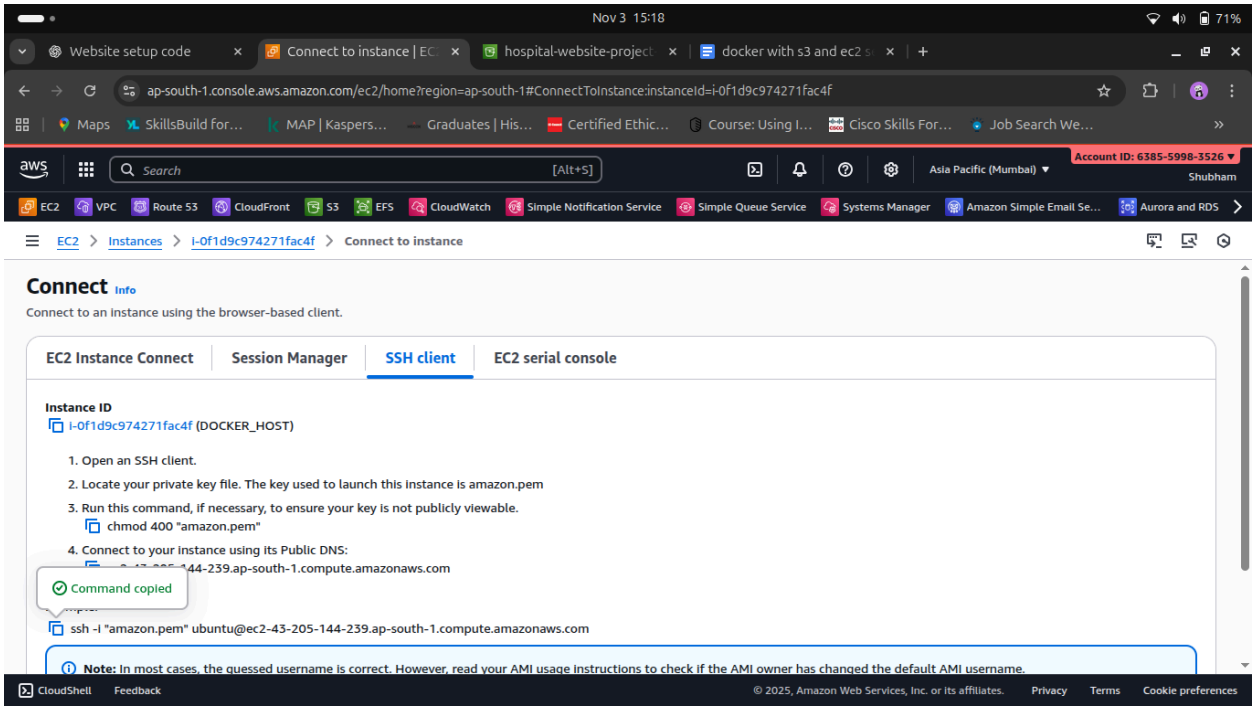


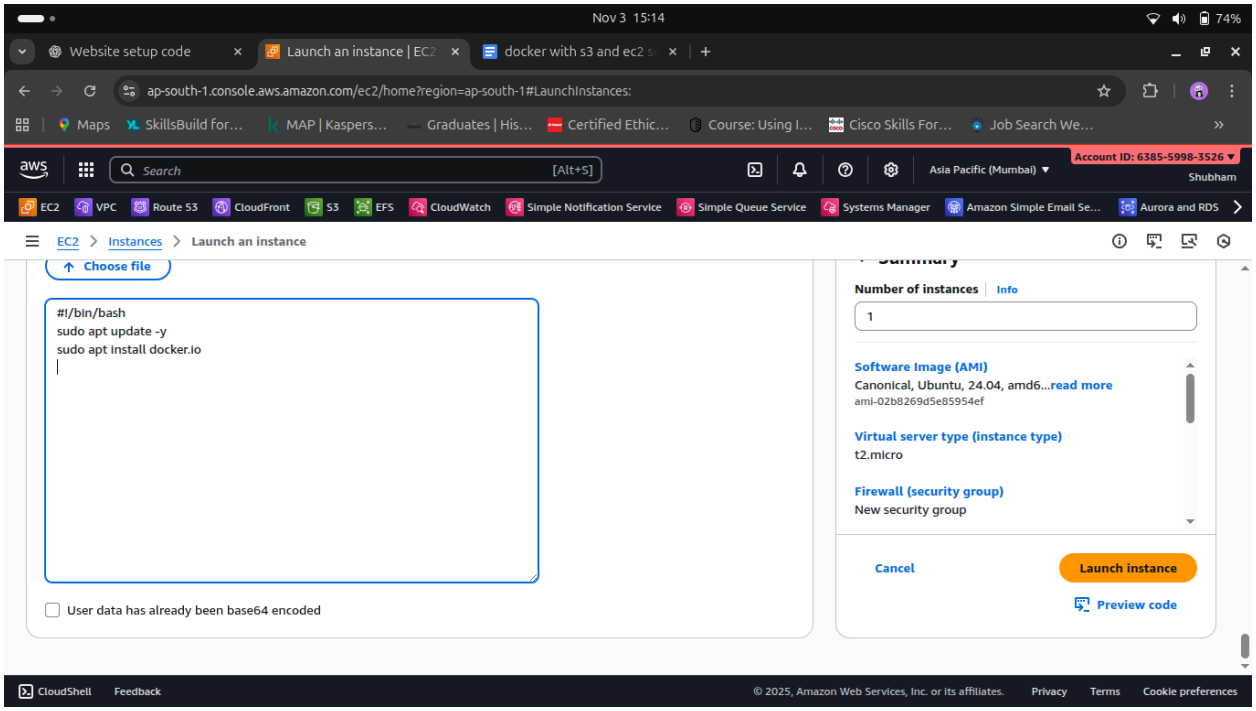
Step 4) We will upload code hospital.html

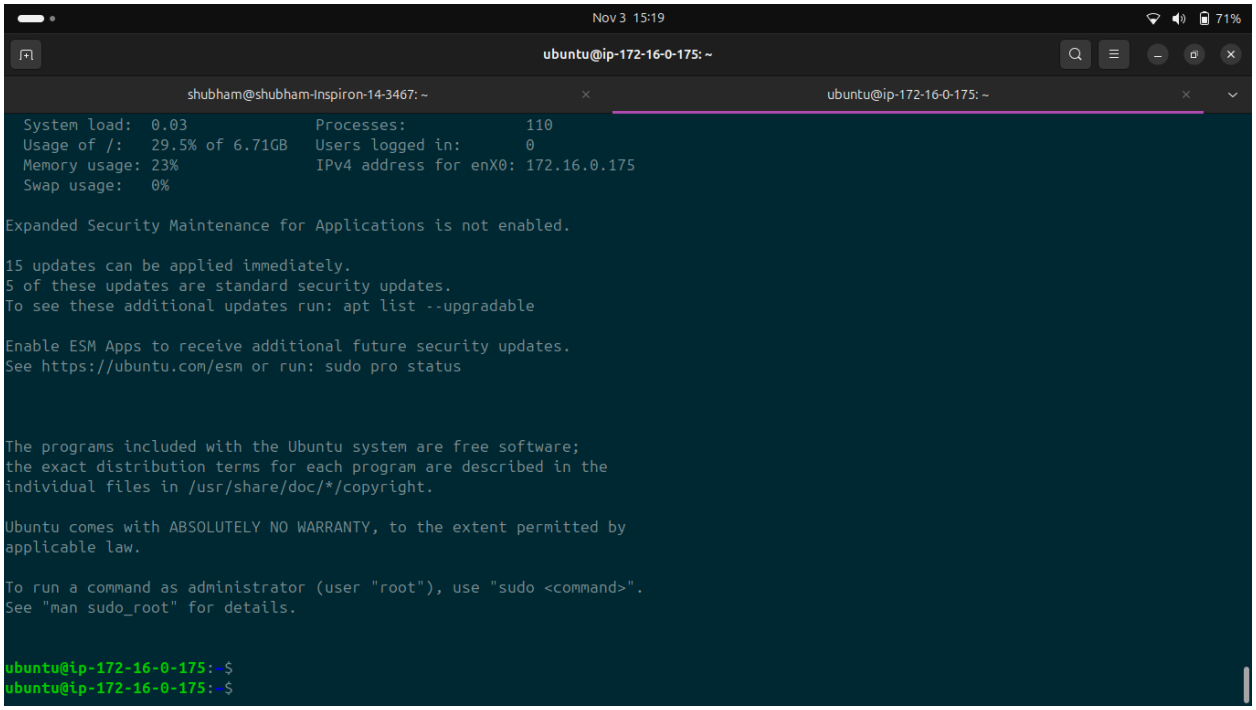


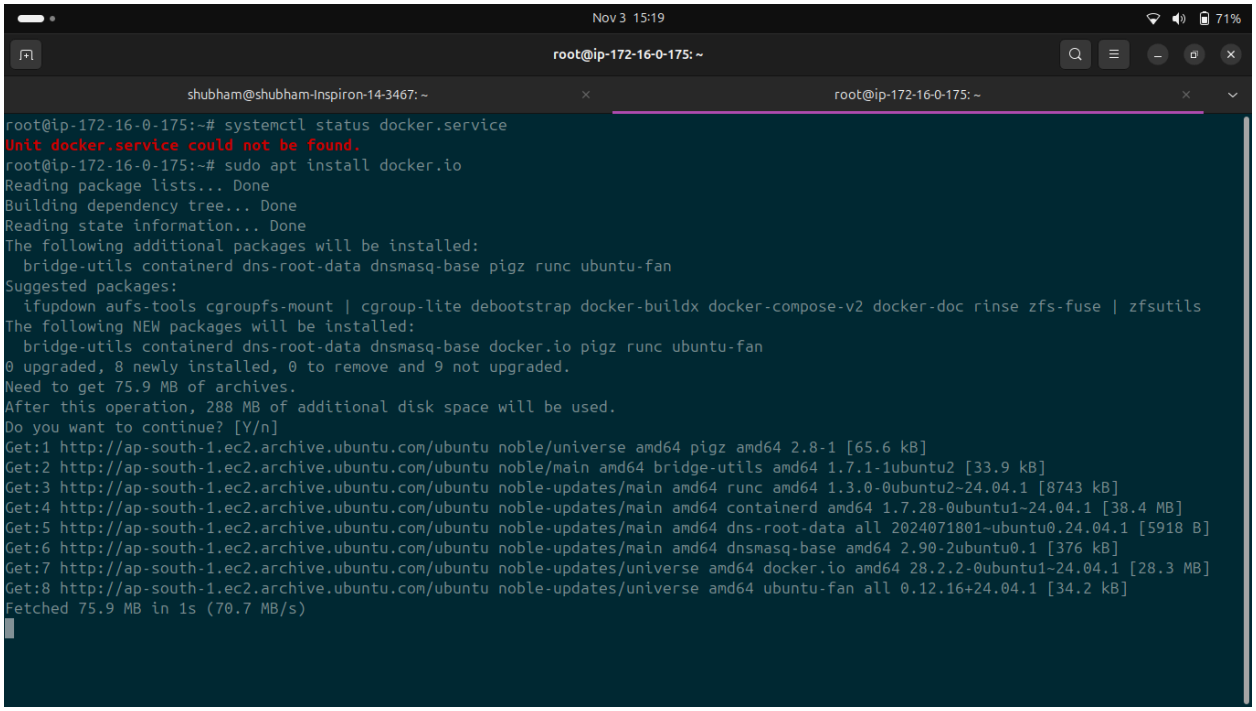
Step 5) Now we will launch ec2 instance with docker installed on it

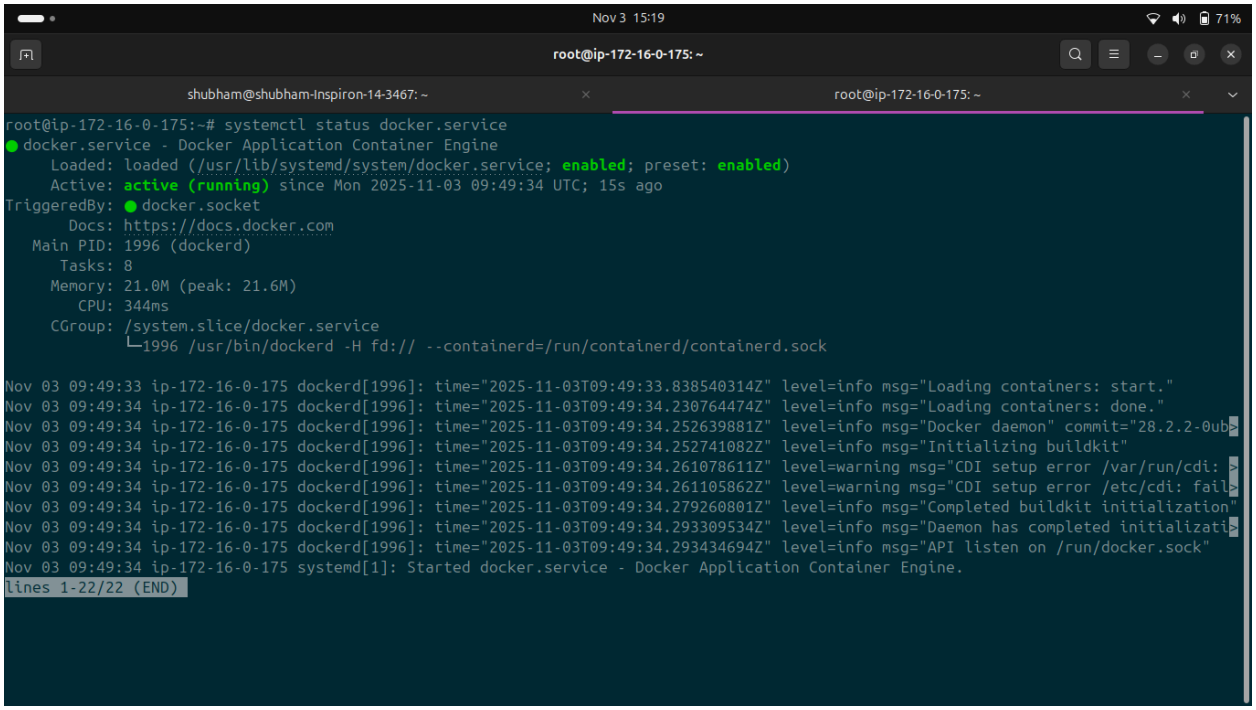



Nov 3 15:18
Website setup code
Connect to instance | EC
hospital-website-project
docker with s3 and ec2
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-0f1d9c974271fac4f
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 6385-5998-3526
Shubham
EC2 > Instances > i-0f1d9c974271fac4f > Connect to instance
Connect Info
Connect to an instance using the browser-based client.
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-0f1d9c974271fac4f (DOCKER_HOST)
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is amazon.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "amazon.pem"
4. Connect to your instance using its Public DNS:
Command copied
ssh -i "amazon.pem" ubuntu@ec2-43-205-144-239.ap-south-1.compute.amazonaws.com
Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

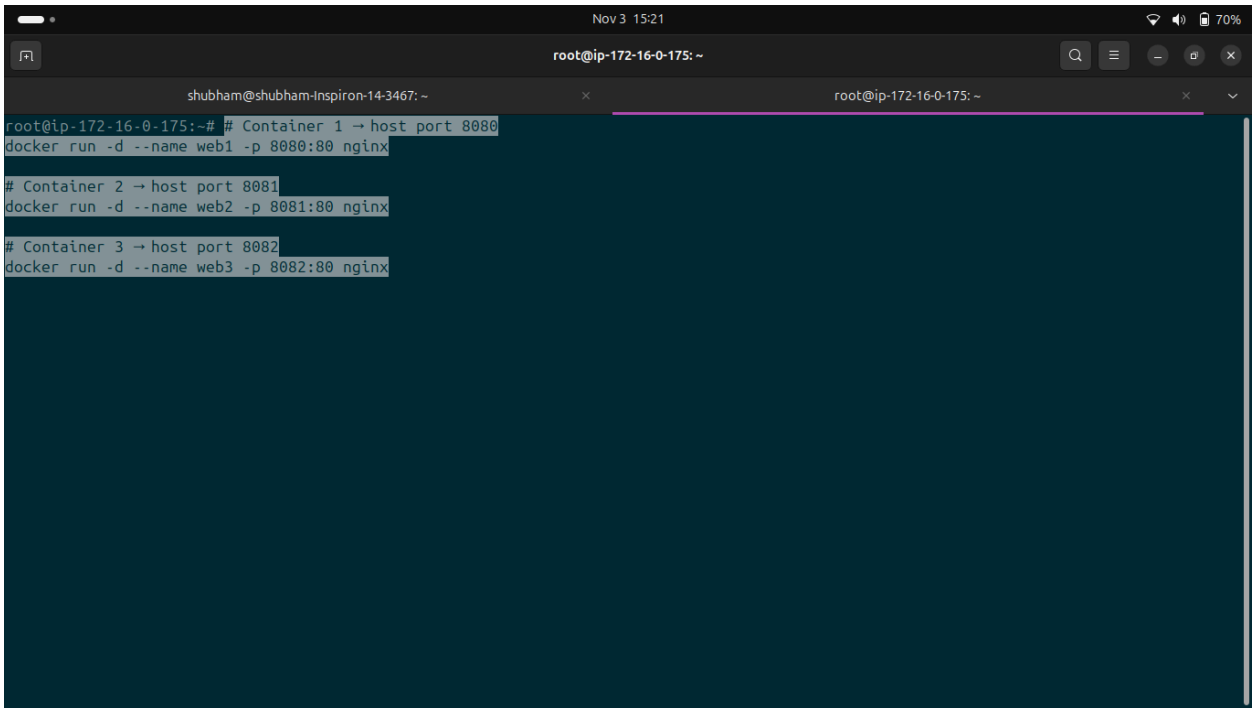

Nov 3 15:14
Website setup code
Launch an instance | EC2
docker with s3 and ec2
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 6385-5998-3526
Shubham
EC2 > Instances > Launch an instance
Choose file
#!/bin/bash
sudo apt update -y
sudo apt install docker.io
User data has already been base64 encoded
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-02b8269d5e85954ef
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Nov 3 15:19
71%
ubuntu@ip-172-16-0-175: ~
shubham@shubham-Inspiron-14-3467: ~
ubuntu@ip-172-16-0-175: ~
System load: 0.03 Processes: 110
Usage of /: 29.5% of 6.71GB Users logged in: 0
Memory usage: 23% IPv4 address for enX0: 172.16.0.175
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
15 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
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-172-16-0-175:~$
ubuntu@ip-172-16-0-175:~$

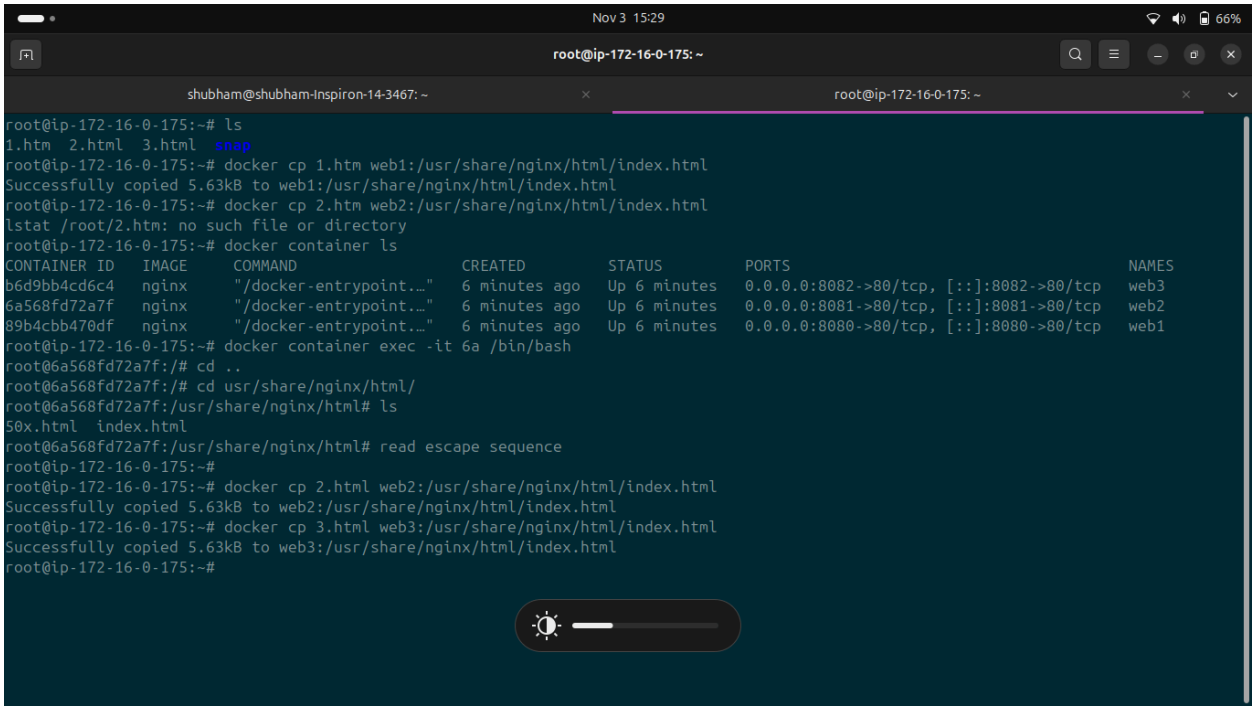

Nov 3 15:19
71%
root@ip-172-16-0-175: ~
shubham@shubham-Inspiron-14-3467: ~
root@ip-172-16-0-175: ~
root@ip-172-16-0-175:~# systemctl status docker.service
Unit docker.service could not be found.
root@ip-172-16-0-175:~# sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 9 not upgraded.
Need to get 75.9 MB of archives.
After this operation, 288 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.3.0-0ubuntu2~24.04.1 [8743 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.28-0ubuntu1~24.04.1 [38.4 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801~ubuntu0.24.04.1 [5918 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1~24.04.1 [28.3 MB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 75.9 MB in 1s (70.7 MB/s)



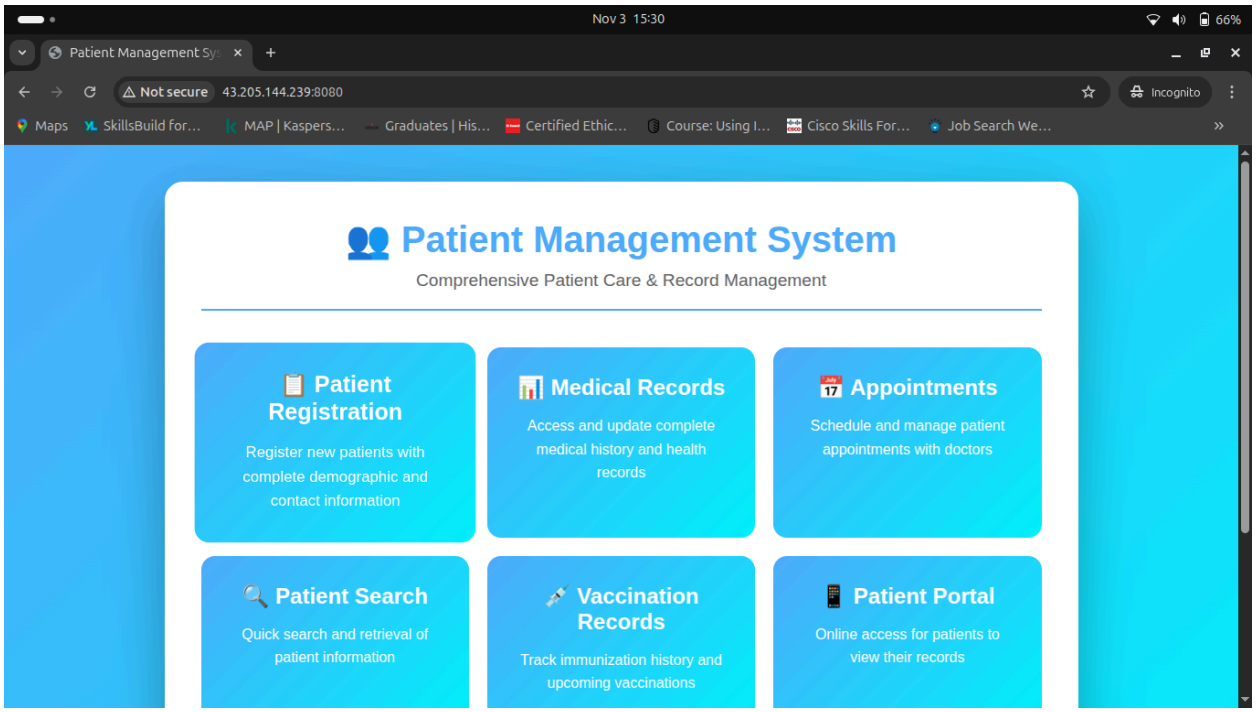
Step 6) We can verify docker is up and in running status



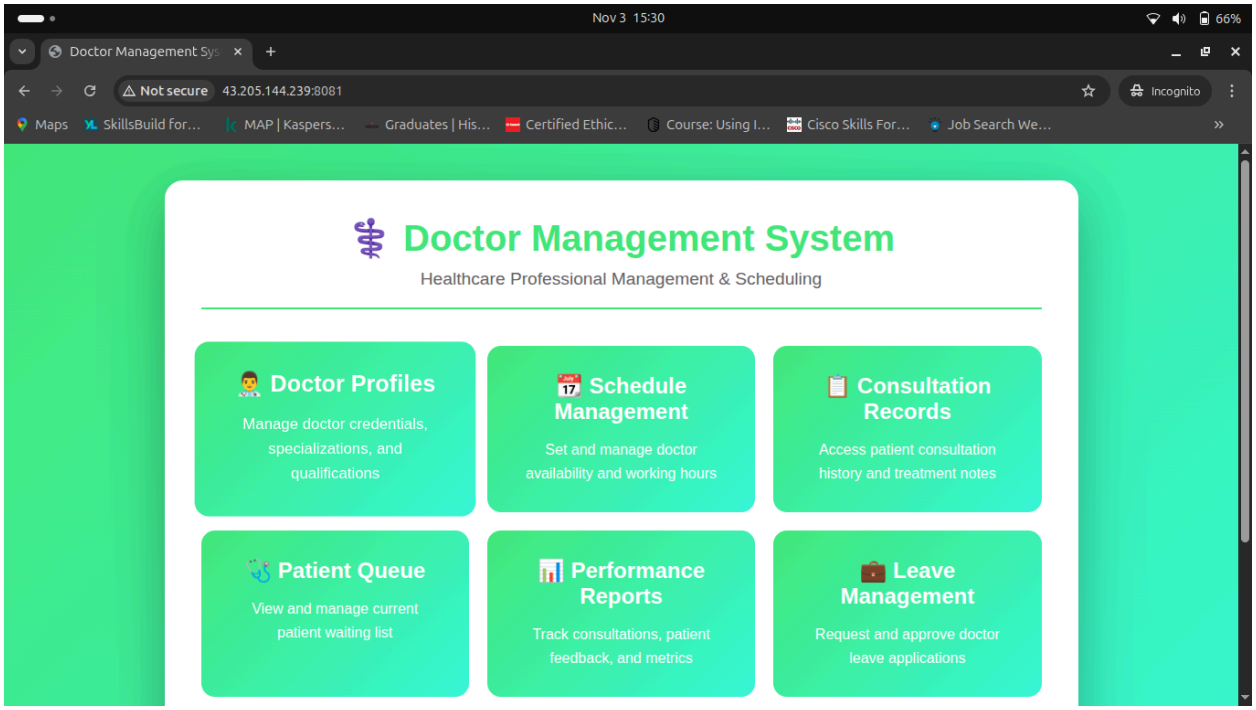
Step 7) Now we will create three containers and port forward then on 8080 8081 and 8082



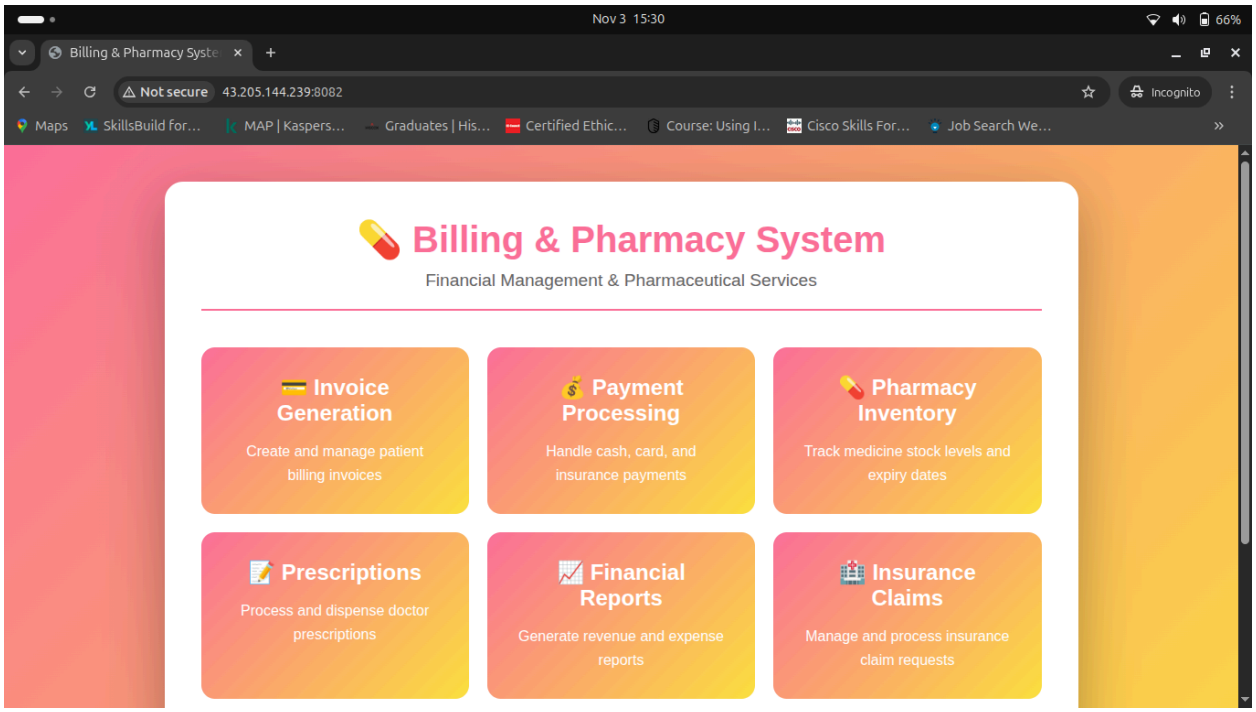
Step 8) We have installed nginx as well and cp files into the folders



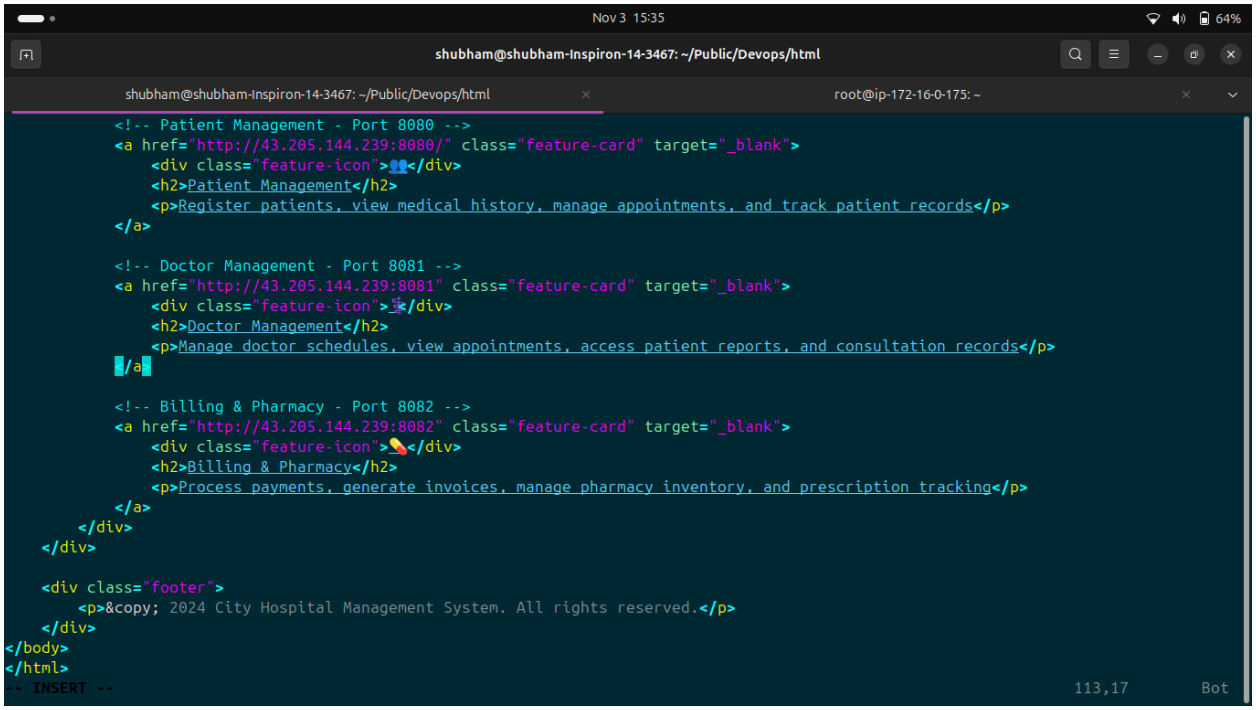
Step 9) Now we can see the public ip and on port 8080 we can see patient management system



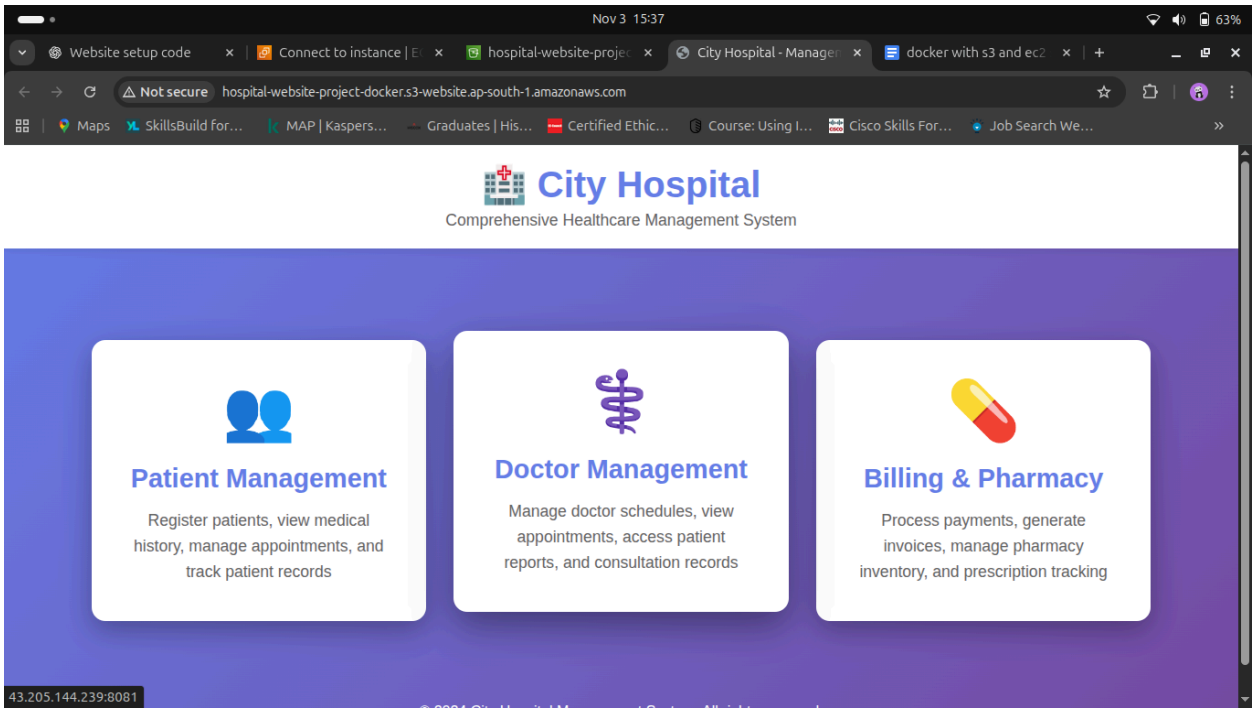
Step 10) Now also on public ip and port 8081 we can see doctor management system



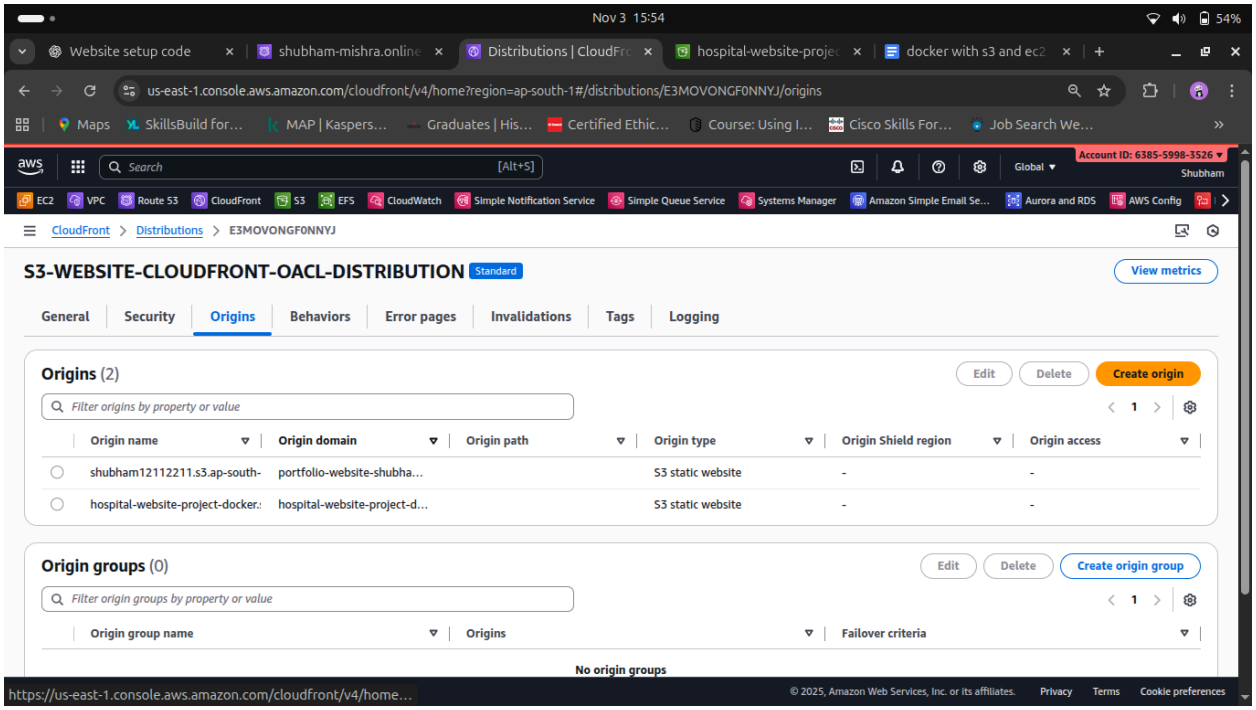
Step 11) Now on public ip and port 8082 we can see the billing and pharmacy system



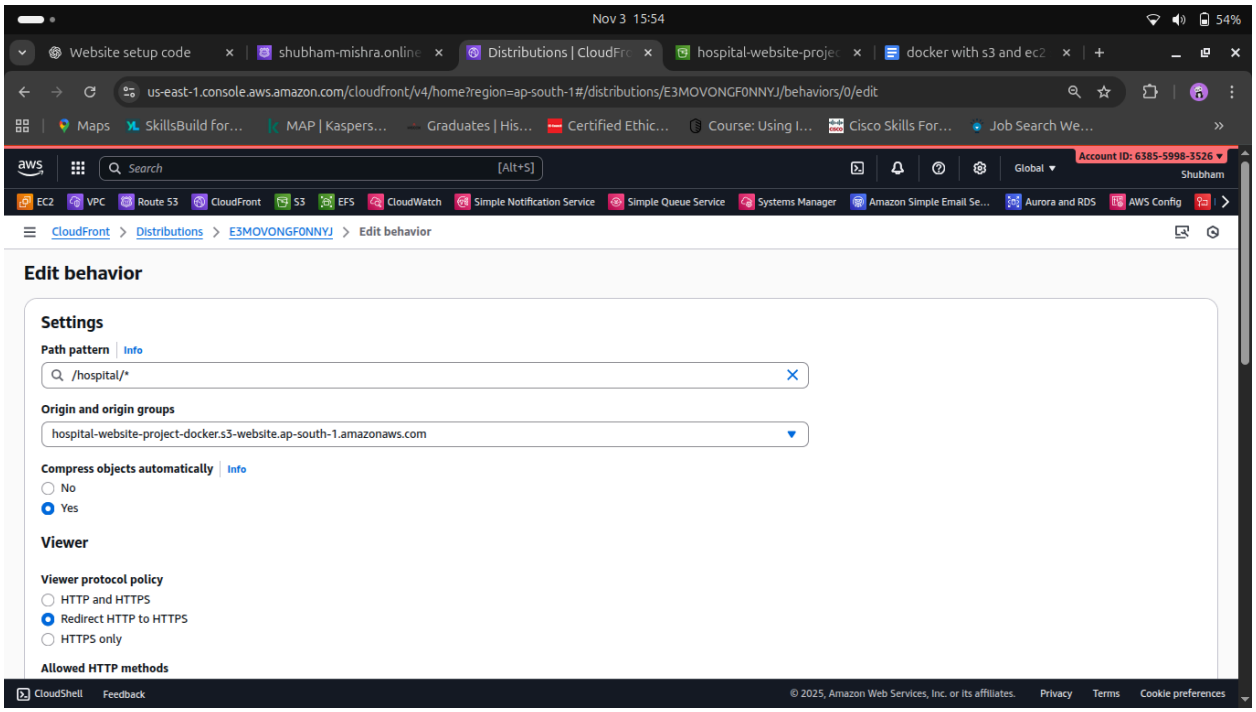
Step 12) Now we need to add the public ip and port on the code of the s3 hospital.html



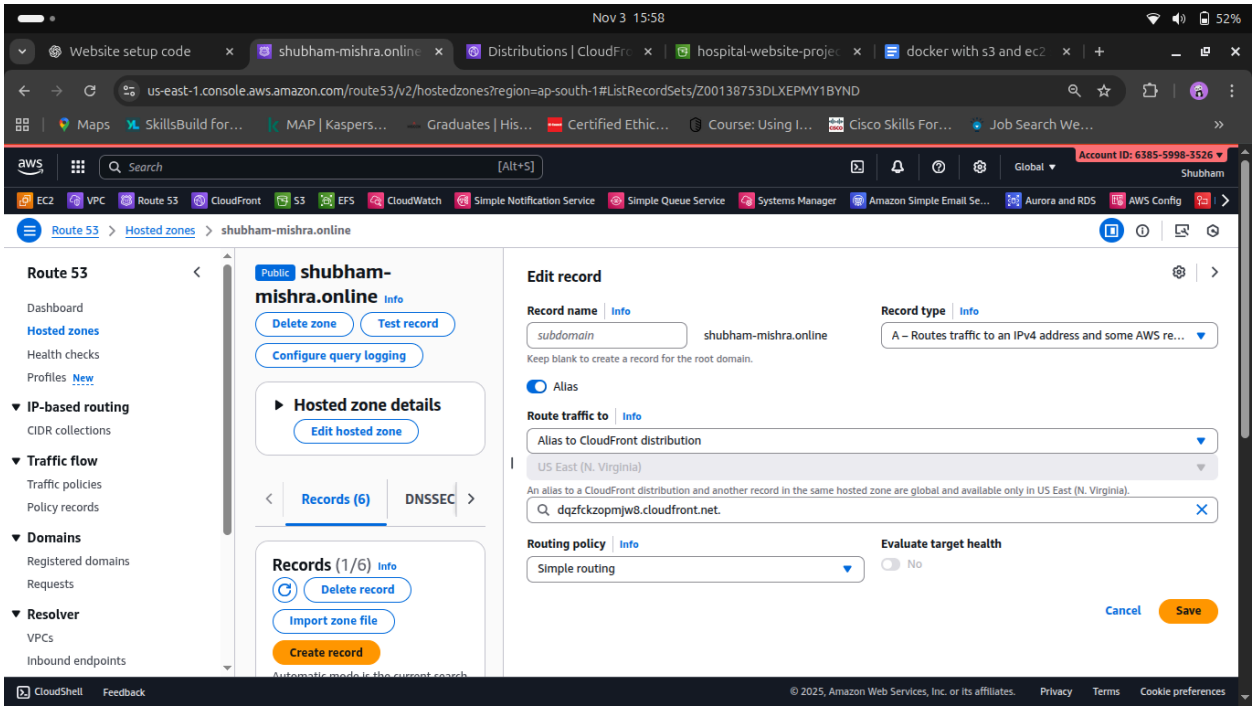
Step 13) Now we can see the s3 bucket website and on clicking each section it will open on different docker ports as we created earlier.



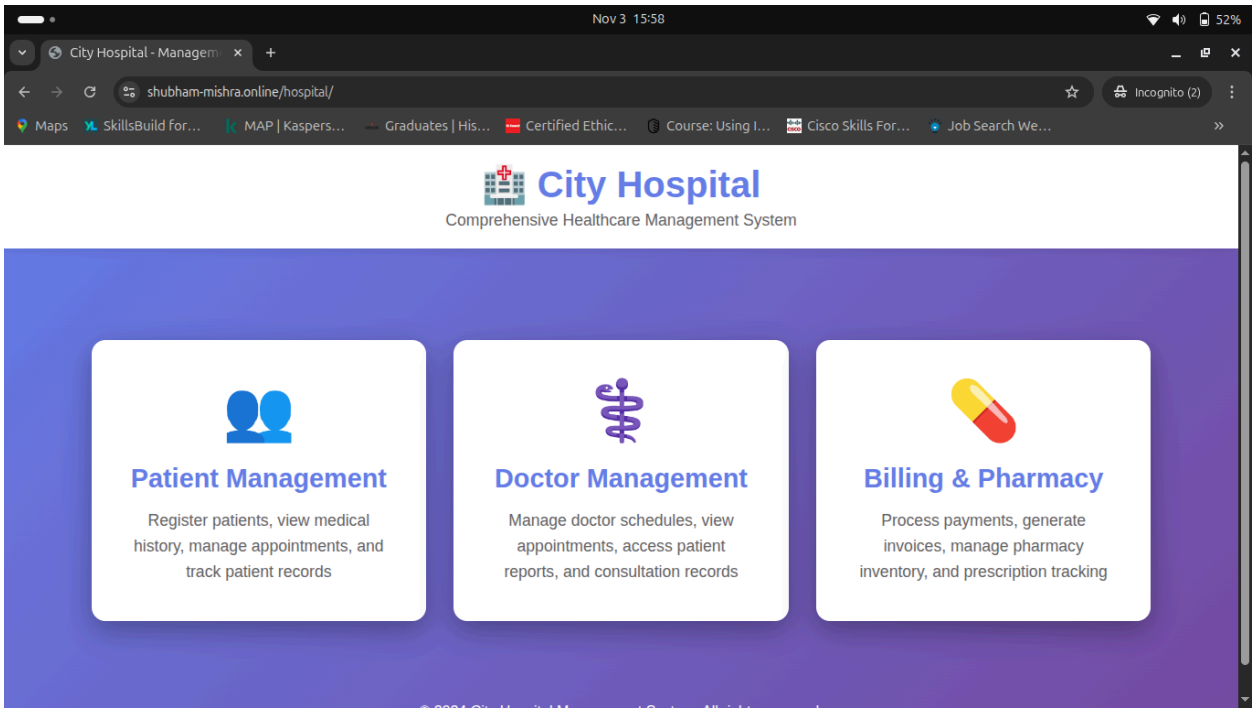
Step 14) Now we will add cloudfront distribution to the s3 and use OACL without making s3 bucket public.



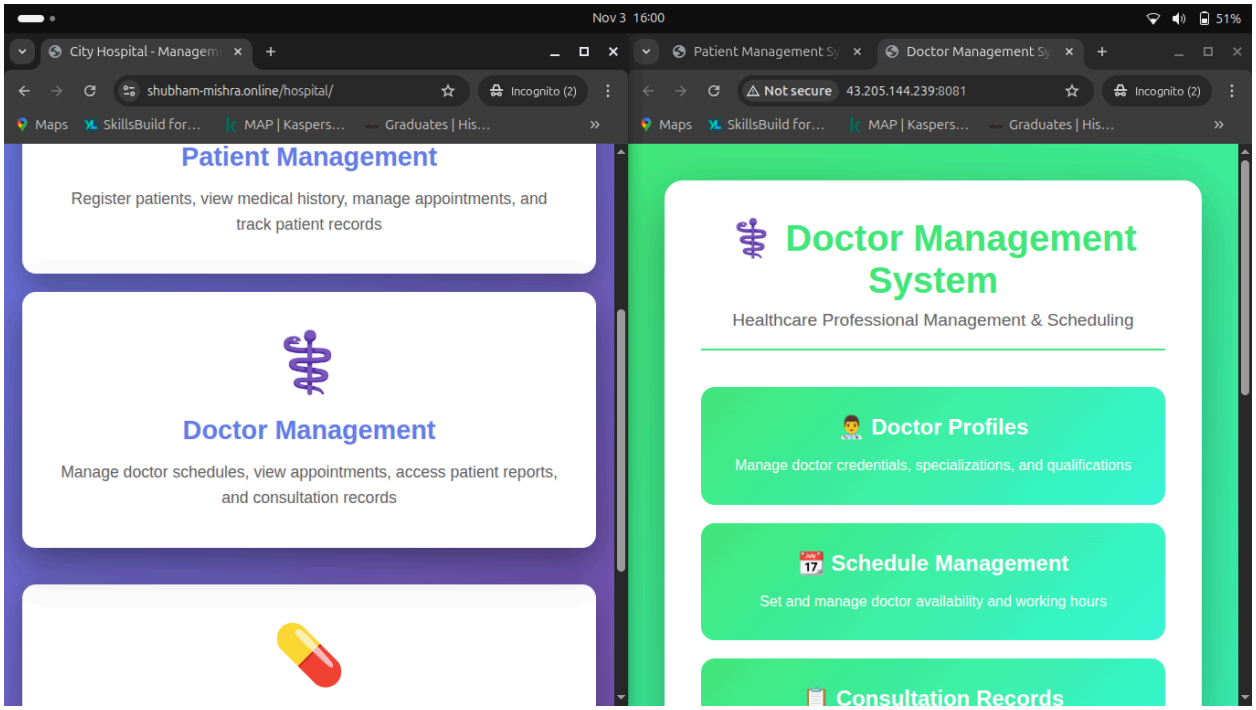
Step 15) Now we will add behaviour as so if /hospital/* then reuqest should goon the hospital s3 bucket



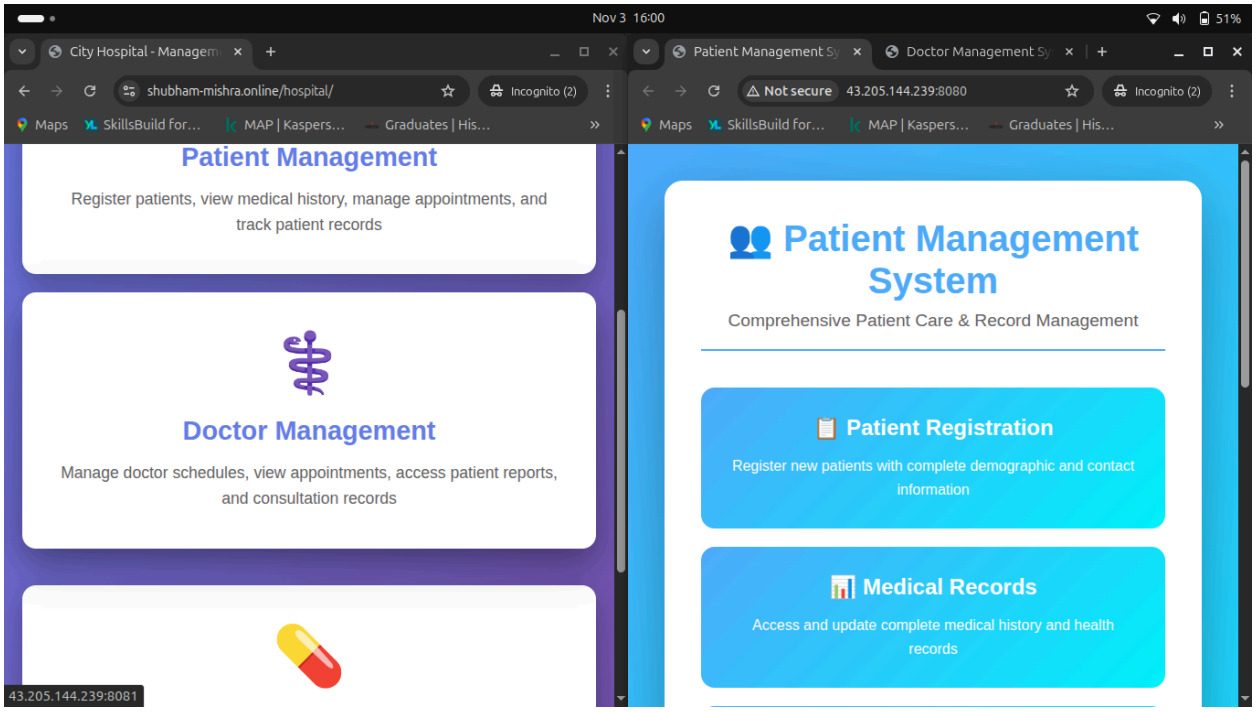
Step 16) Now we will add this clodufront distribution as alias in route53



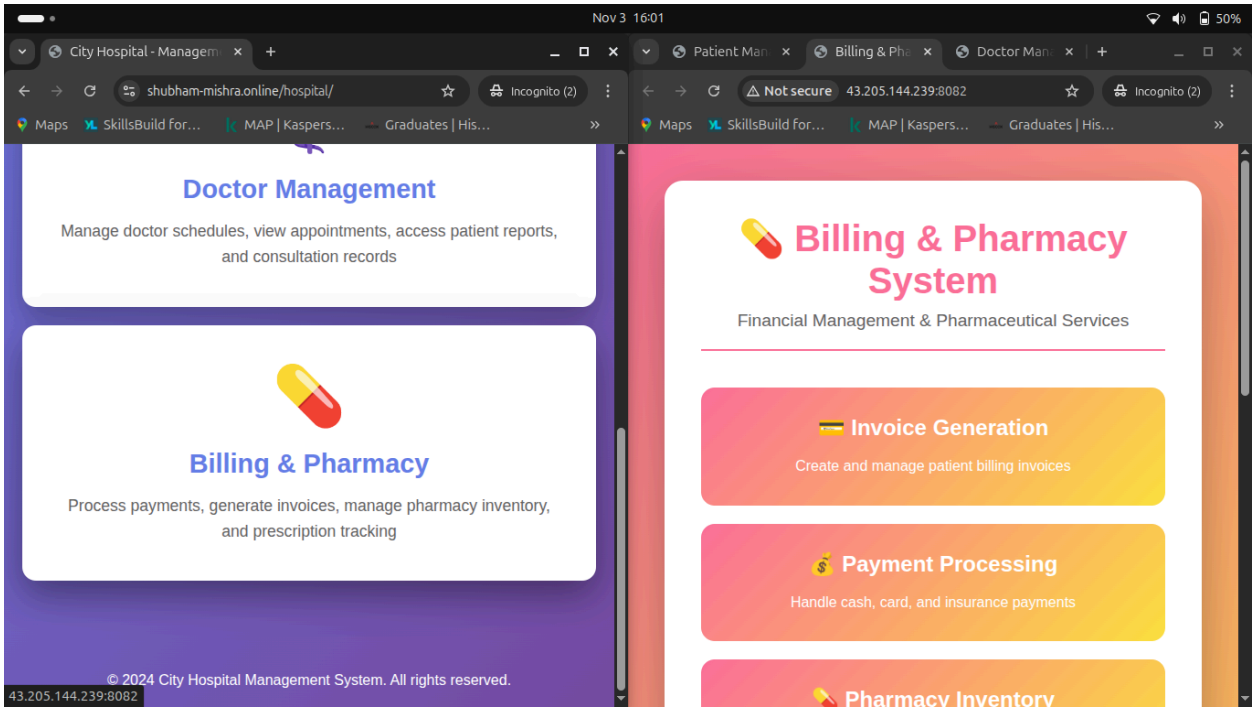
Step 17) Now we can access hospital website on https://shubham-mishra.online/hospital



Step 18) Now on clicking doctor management we can see it opens in ec2 on docker host at port 8081



Step 19) Similarly Now on clicking patient management we can see it opens in ec2 on docker host at port 8080



Step 20) Now on clicking billing management we can see it opens in ec2 on docker host at port 8082