# **Capstone Project**

**Deployment Instruction:**

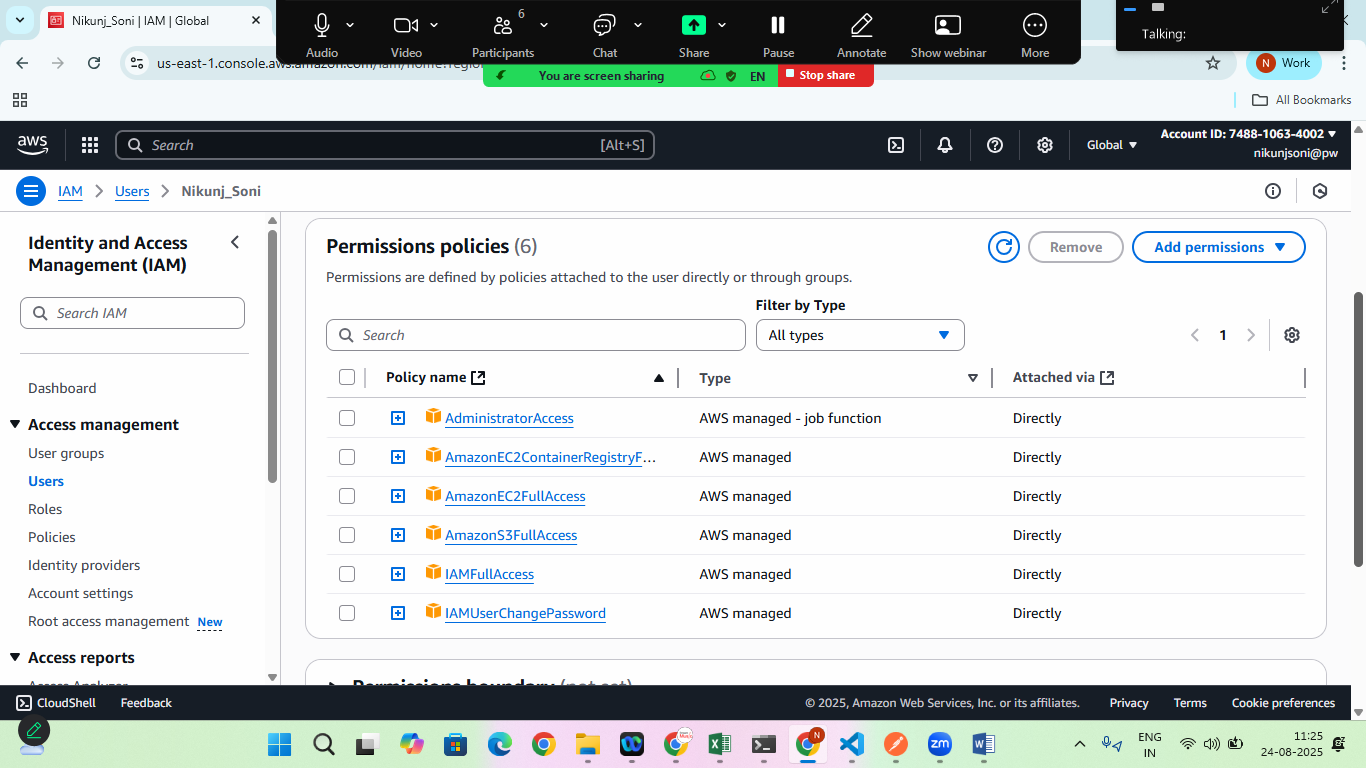
* **Create a Project With Fully Working Backend and Frontend**
* **Upload it on Github and use CI/CD or Jenkins for Automated Build**
* **Link: https://github.com/Nikunj-Java/Devops\_Capstone\_PW.git**

**Prerequisites**

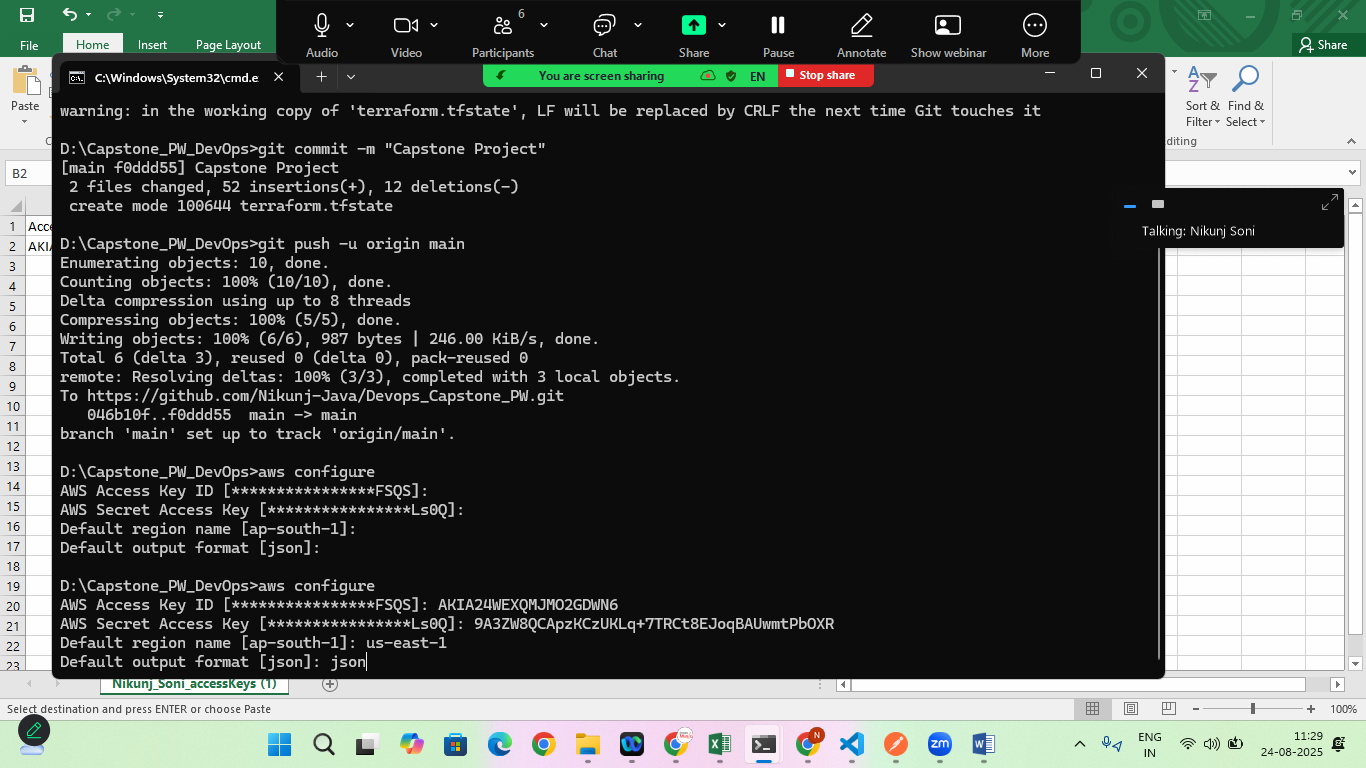
* **AWS Account(With Administartor Access)**
* **AWS CLI Installed (** aws Configure **will be used)**
* **Git and Github Account**
* **Terraform V 1.2 + Above**
* **Node.js (for Future Extensibility)**
* **Python**

**Set up AWS Credentials**

1. **Goto AWS Console> I AM > Create User or Choose Existing USER**
2. **Create New Credentials for AWS CLI**
3. **Add the Below Permission**



1. **Configure aws cli using cmd**



**If its looks empty copy paste the new credentials**

**2. Clone The Devops Accelerator Repository**

**> mkdir CapstoneProject && cd CapstoneProject**

**> git init**

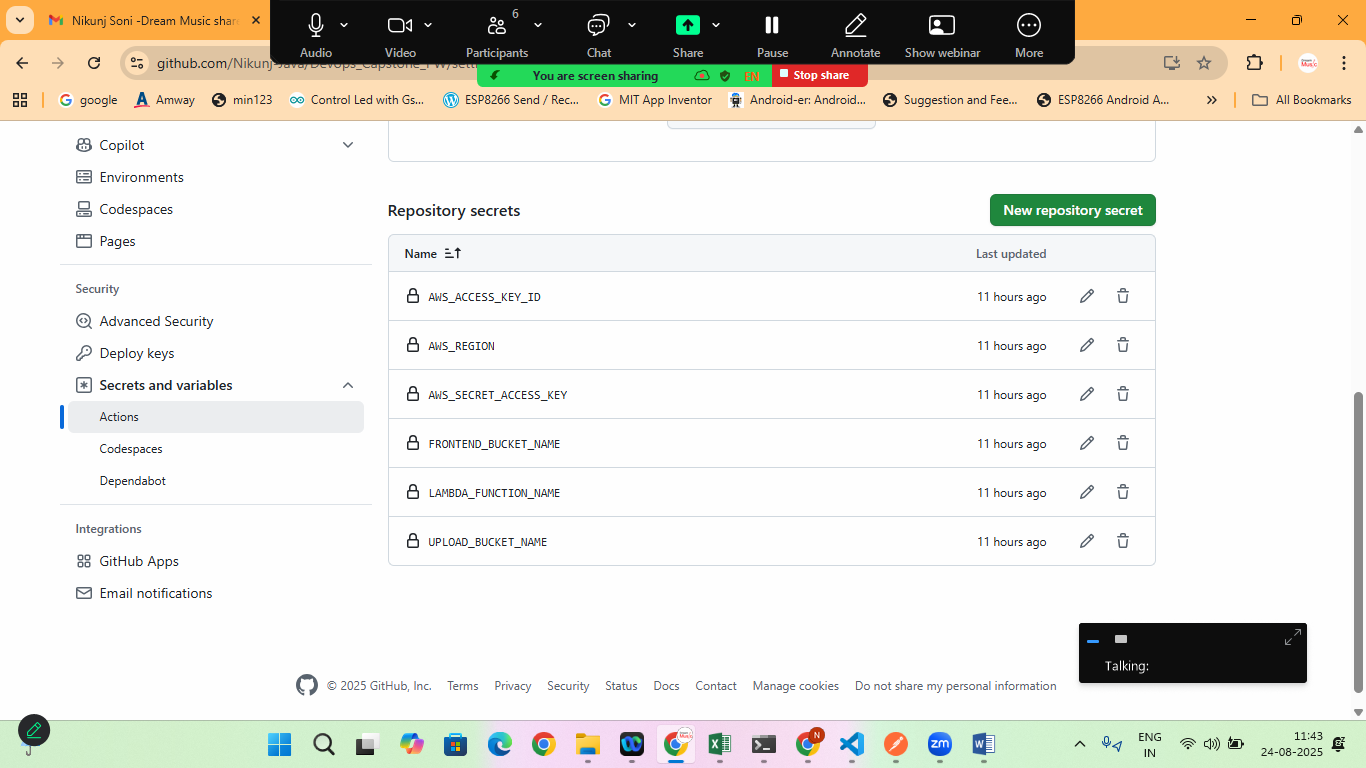
**> git clone** [**https://github.com/Nikunj-Java/Devops\_Capstone\_PW.git**](https://github.com/Nikunj-Java/Devops_Capstone_PW.git)

**Set up Giithub Repository SECRETS**

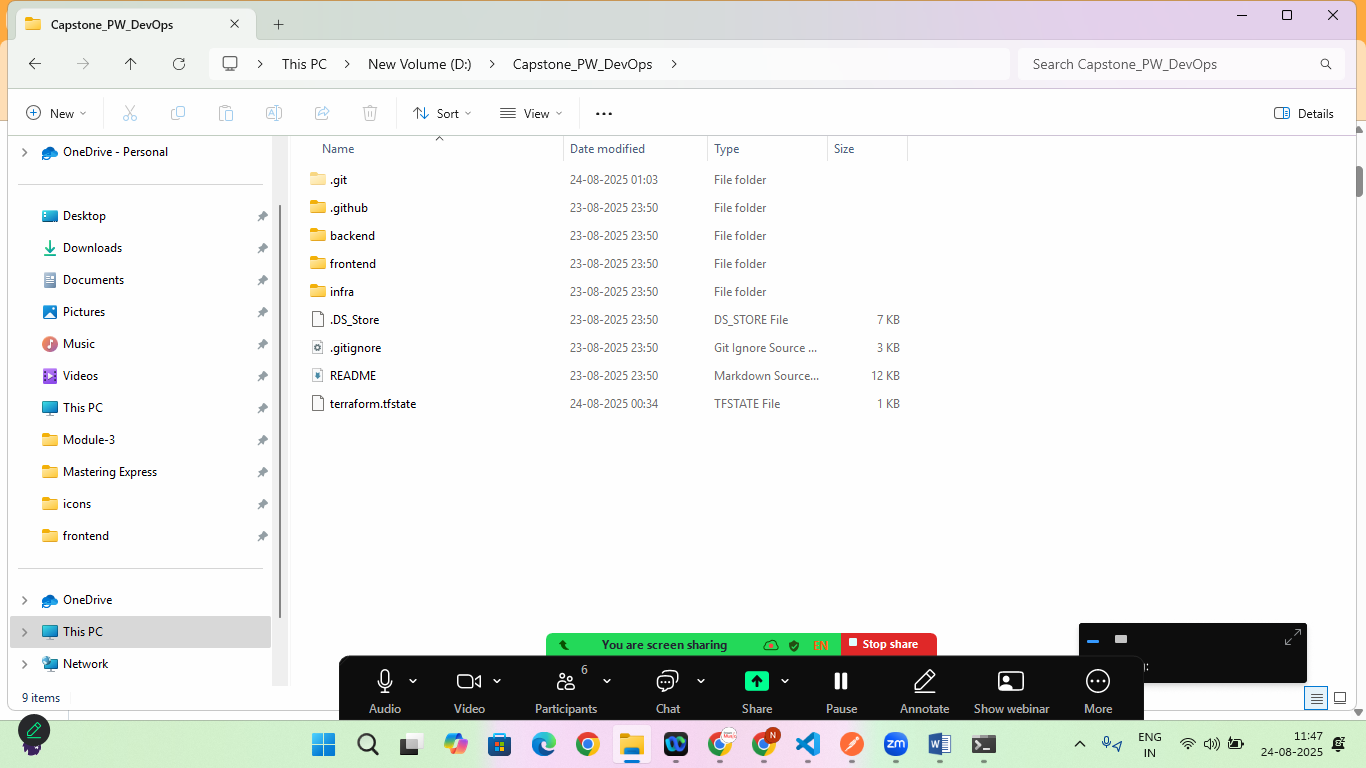
**Navigate To> Github> Repo>Settings>Secrets>Actiions>New repository Secret**

**Add the Below Secrets in Github**

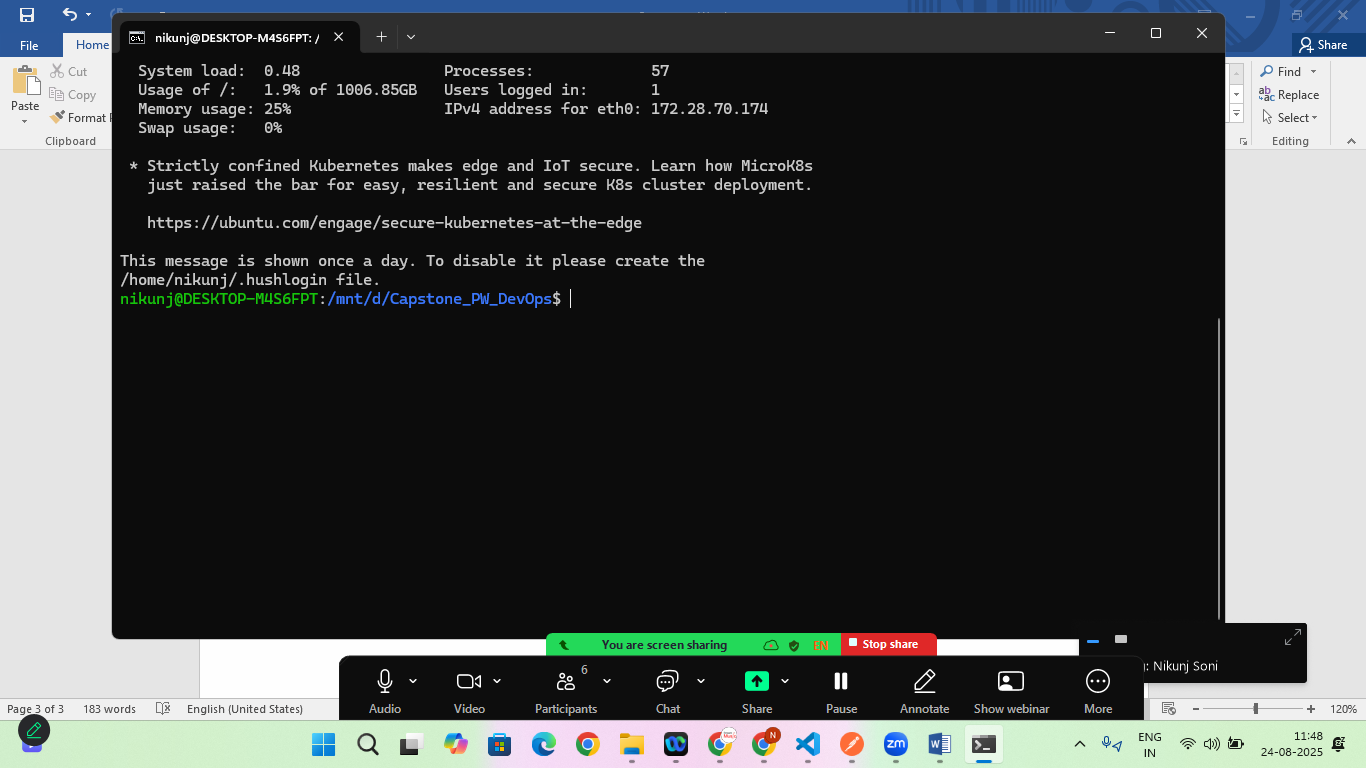
1. **AWS\_ACCESS\_KEY\_ID= YOUR AWS IAM access key**
2. **AWS\_SECRET\_ACCESS\_KEY= Your AWS IAM secret key**
3. **AWS\_REGION= us-east-1**
4. **LAMBDA\_FUNCTION= process-uploaded-file**
5. **FRONTEND\_BUCKET\_NAME= devops-accelerator-frontend-hosting-bucket**
6. **UPLOAD\_BUCKET\_NAME= devops-accelerator-upload-bucket**



**Goto> your local repository> where the Code is Available**



**Open wsl terminal here**



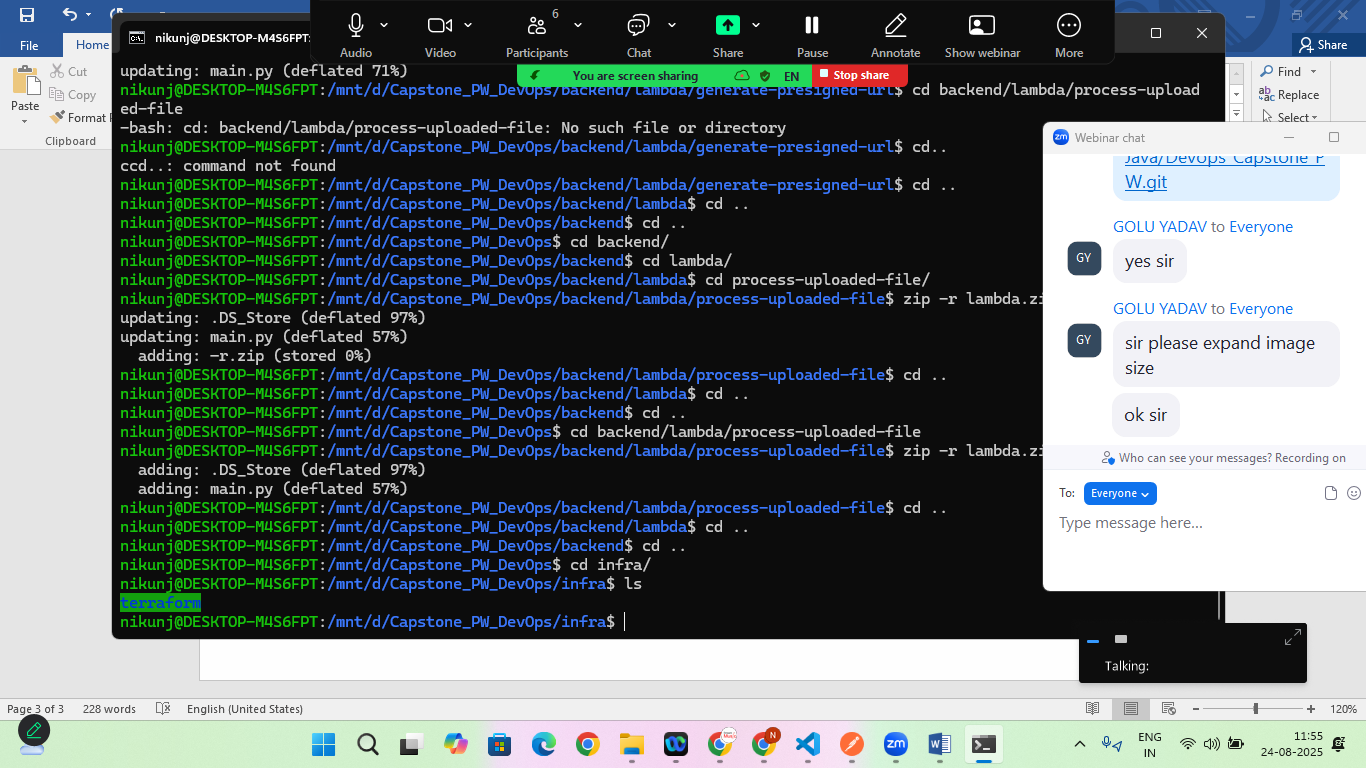
**Step:1 Create Zip File**

* cd backend/lambda/process-uploaded-file
* zip –r lambda.zip .
* cd ../generate-presigned-url
* zip –r lambda.zip .

**check the lambda folder in backend and frontend subdirectory is prepared**

**now come back to the root path and shift to infra folder**

* **cd ../infra**



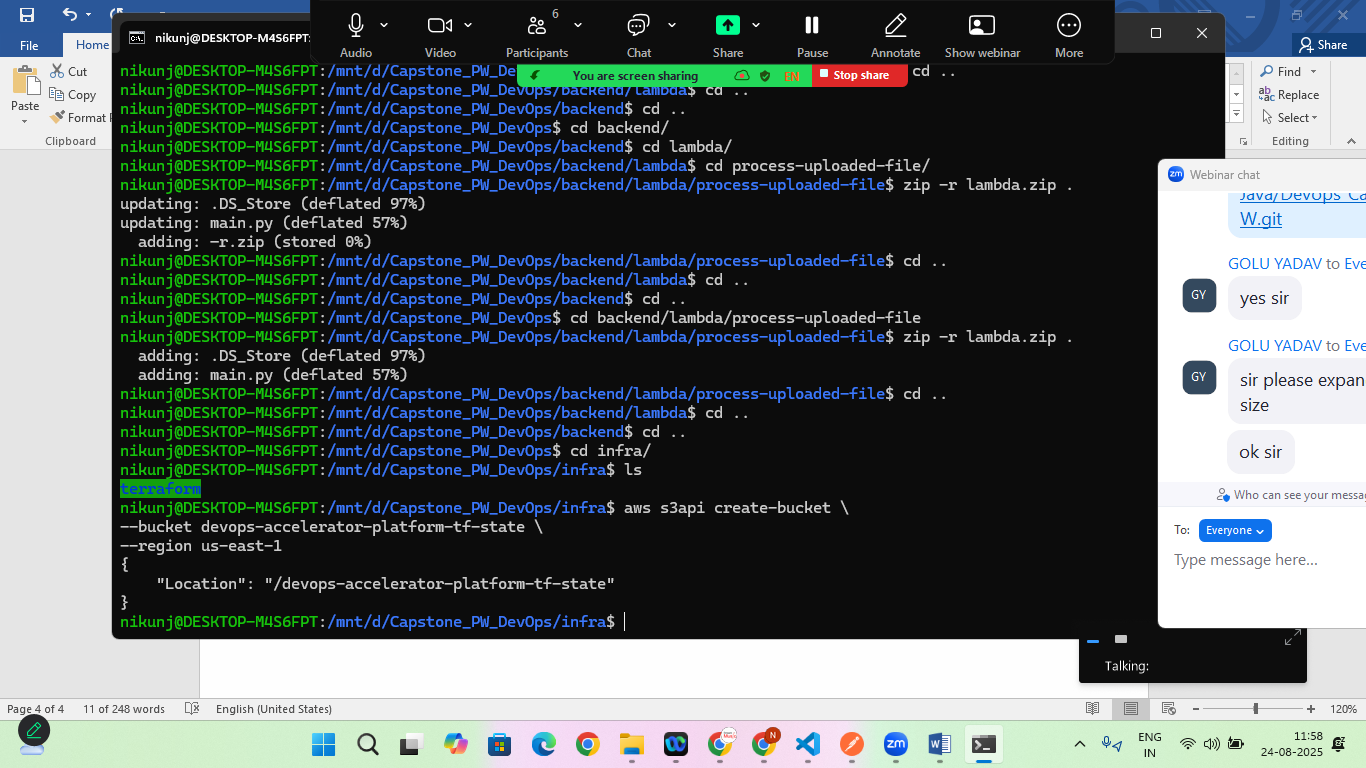
**Step:2 Create required Buckets**

#s3 Bucket for state

aws s3api create-bucket \

--bucket devops-accelerator-platform-tf-state \

--region us-east-1



# Dynamo DB Table for Locking

aws dynamodb create-table \

--table-name devops-accelerator-tf-locker \

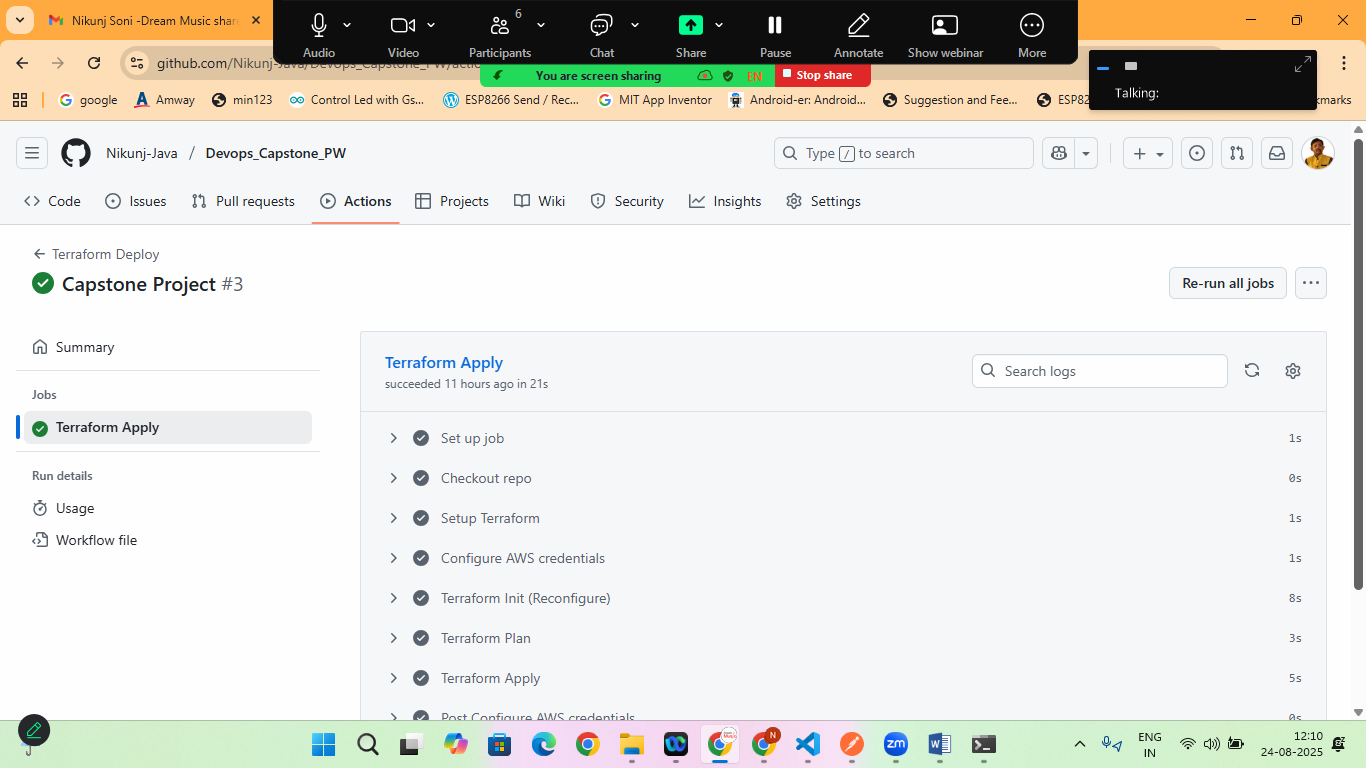
--attribute-definitions AttributeName=LockID,AttributeType=S \

--key-schema AttributeName=LockID,KeyType=HASH \

--billing-mode PAY\_PER\_REQUEST \

--region us-east-1

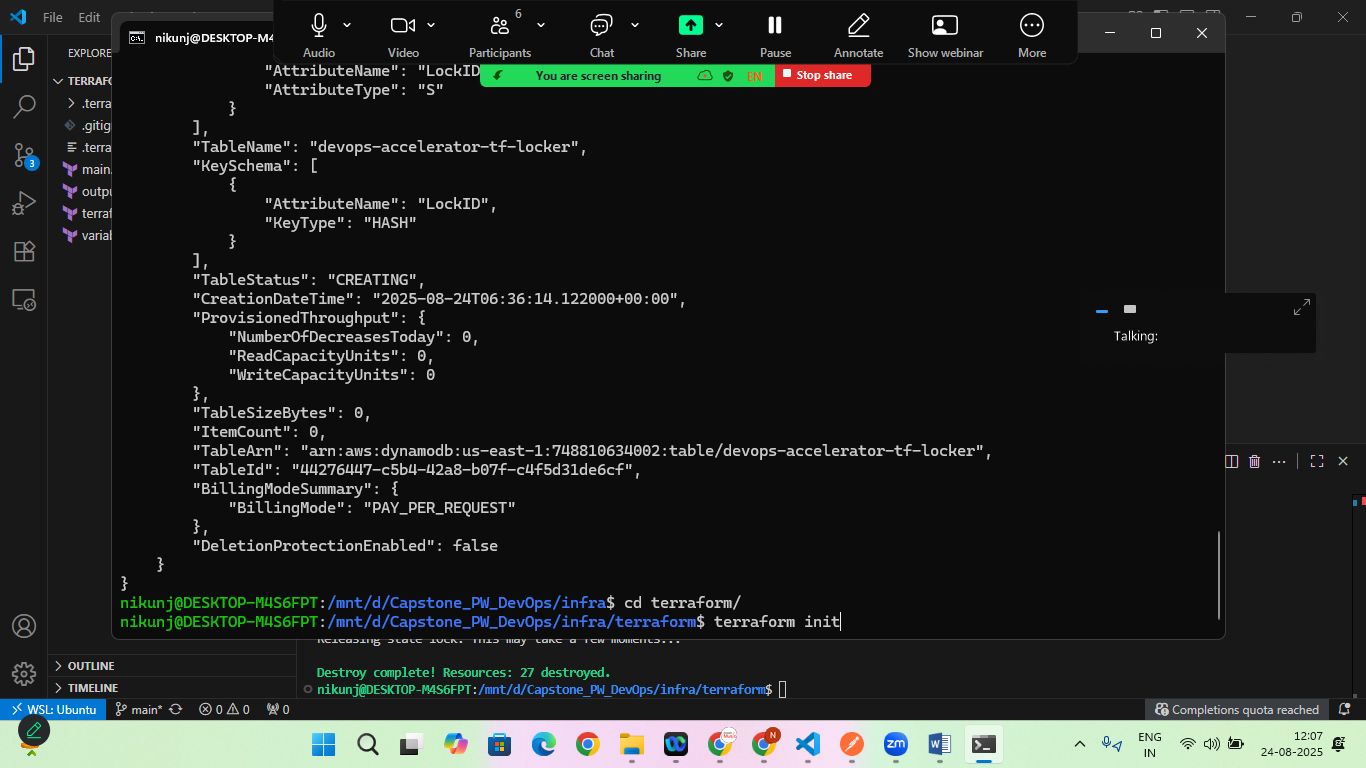
**Step:3 Push the changes to Git hub so that github/action will be performed**



* After successfully build move to the next step

**Step:4 Run Terraform Commands**

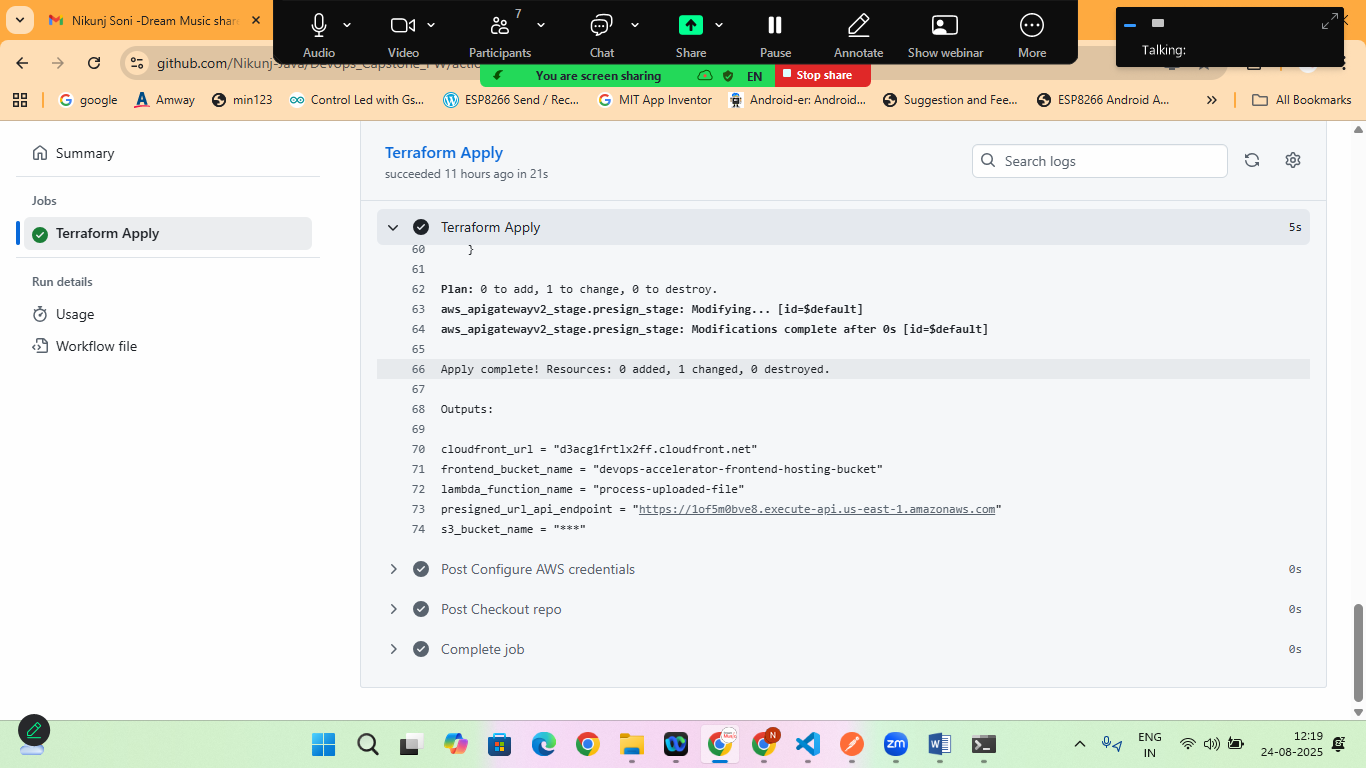
* cd infra/terraform
* terraform init
* terraform validate
* terraform plan



Now Again Push the Code to Github

Check the Github>action>terraform>TerraformApply>

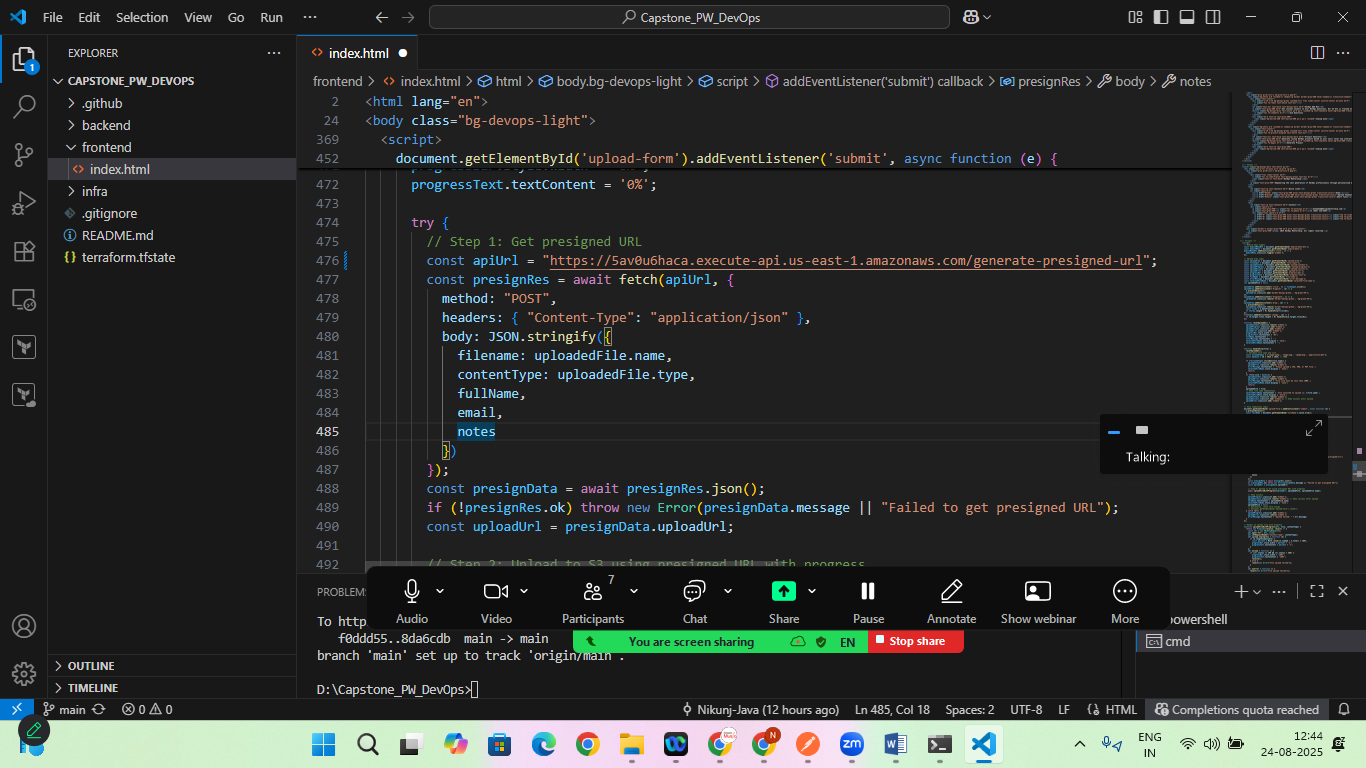
Check the output



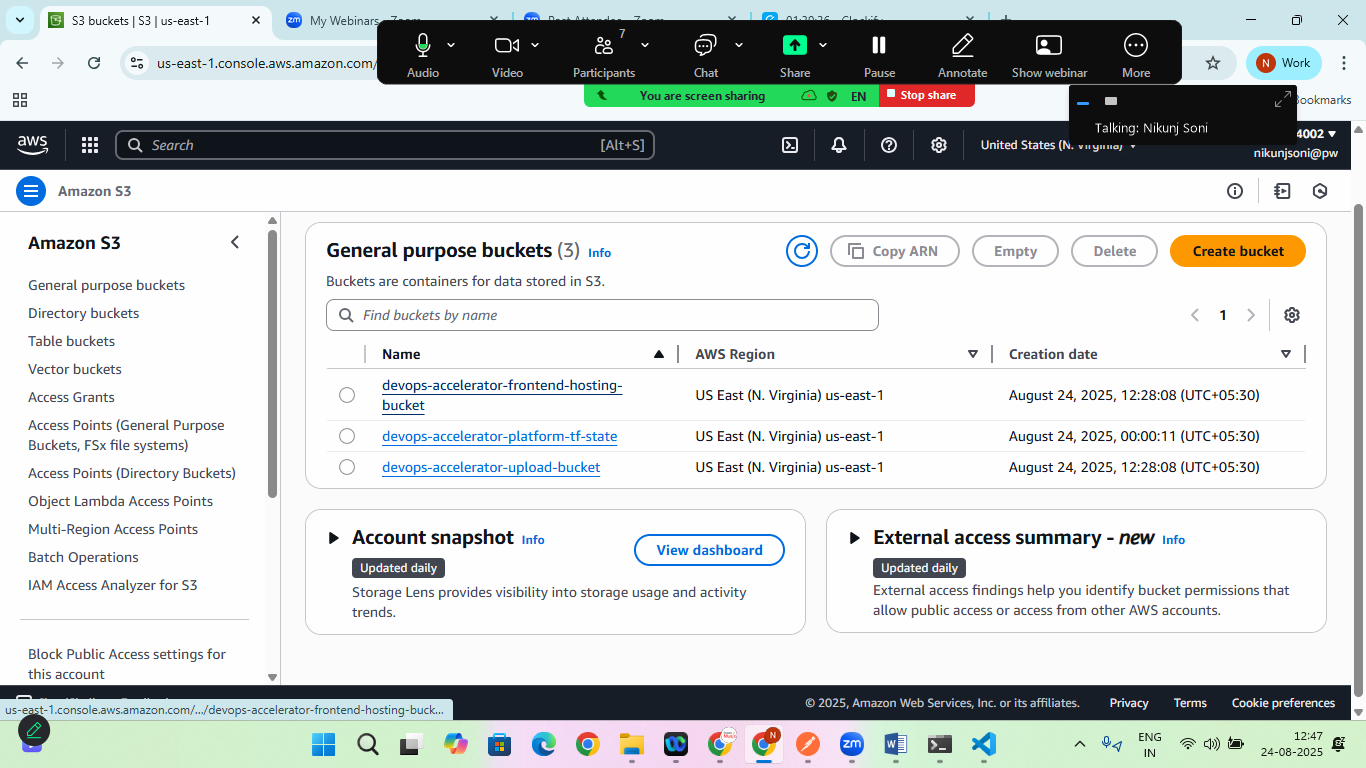
Now once it is ready:

Copy presigned\_url\_api\_endpoint and paste in index.html file

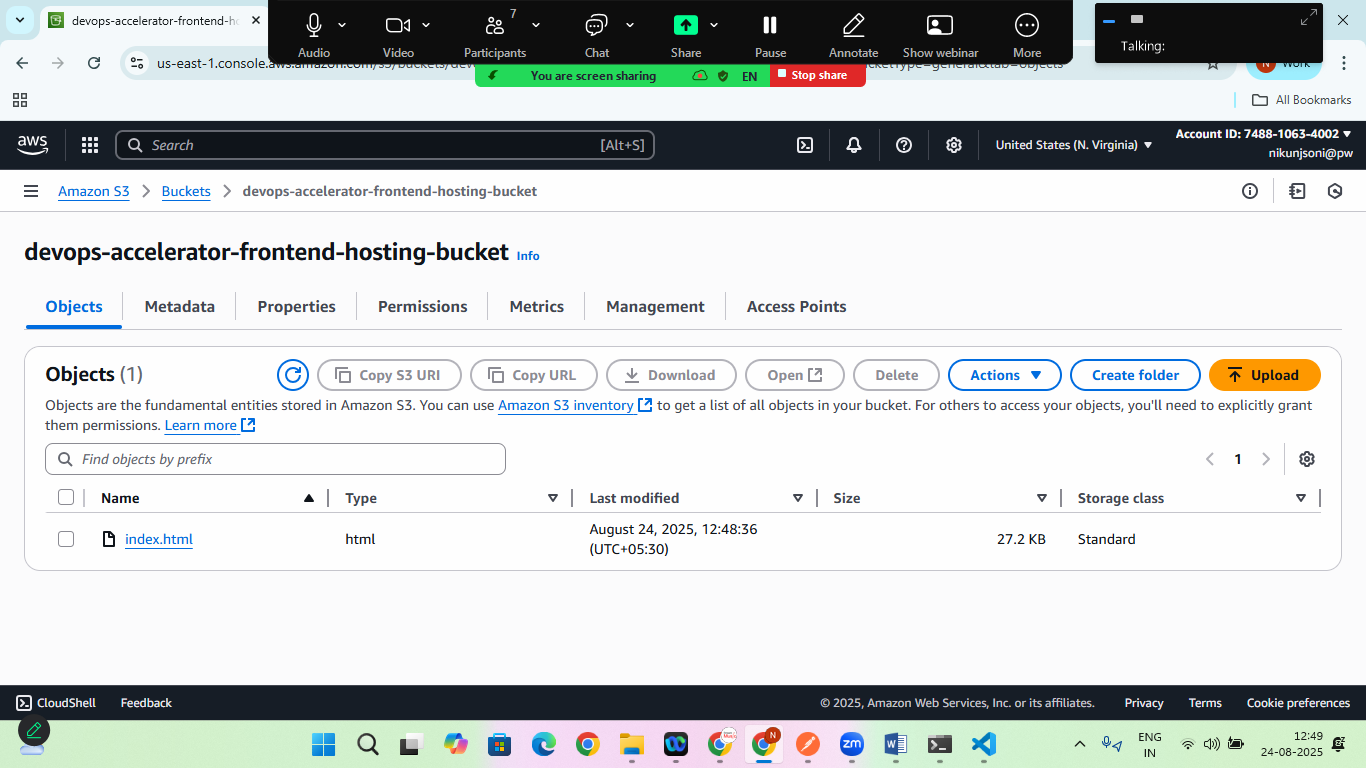
apiURL: <your presignedurl>/generate-presigned-url



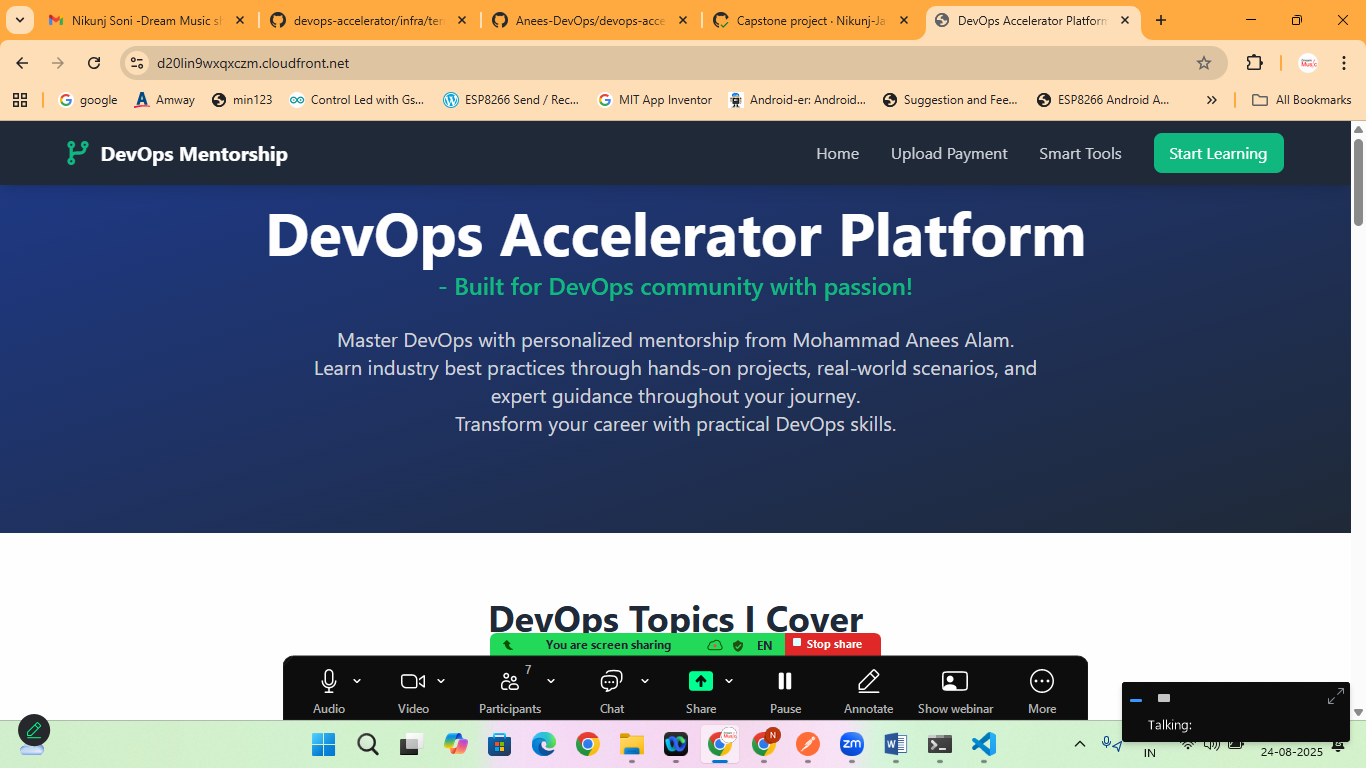
Now upload this index.html file to s3 bucket

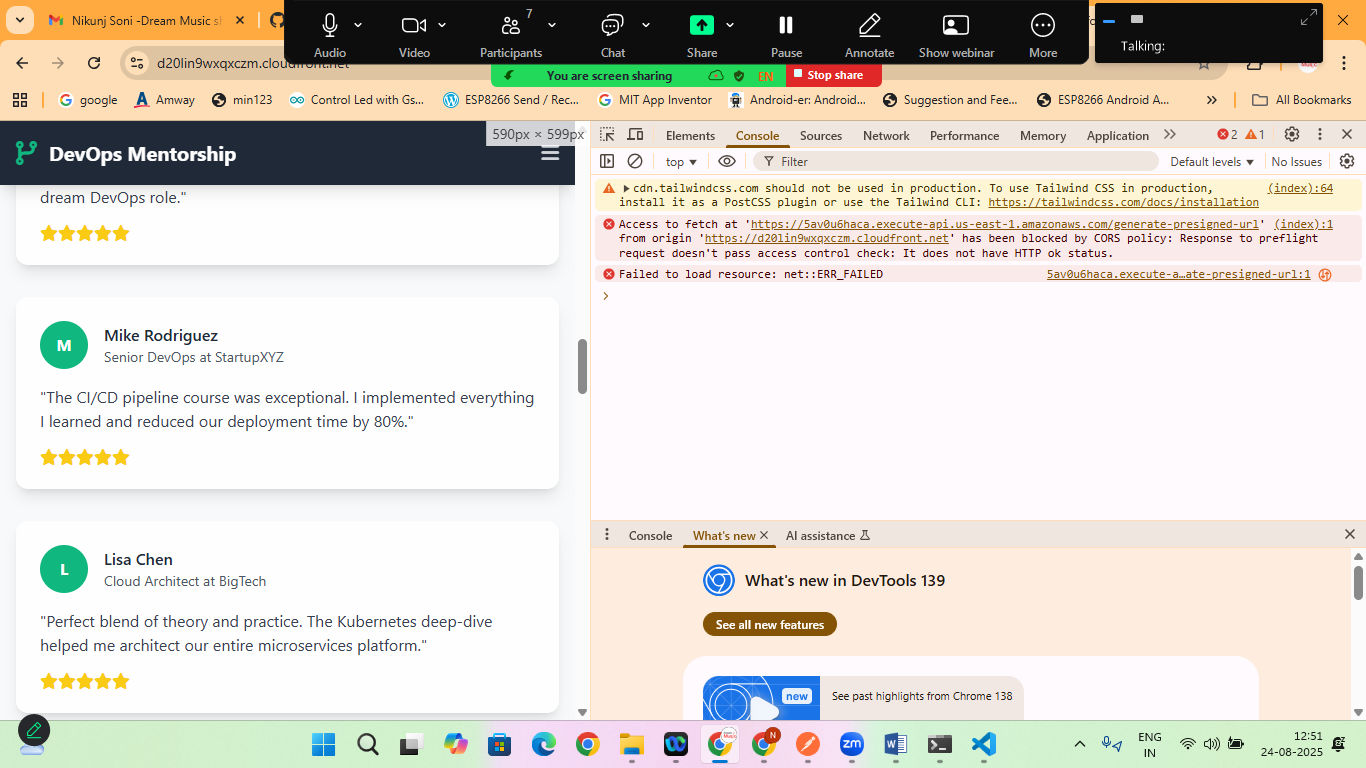


Choose devops-accelerator-frontend-hosting-bucket and upload index.html file here



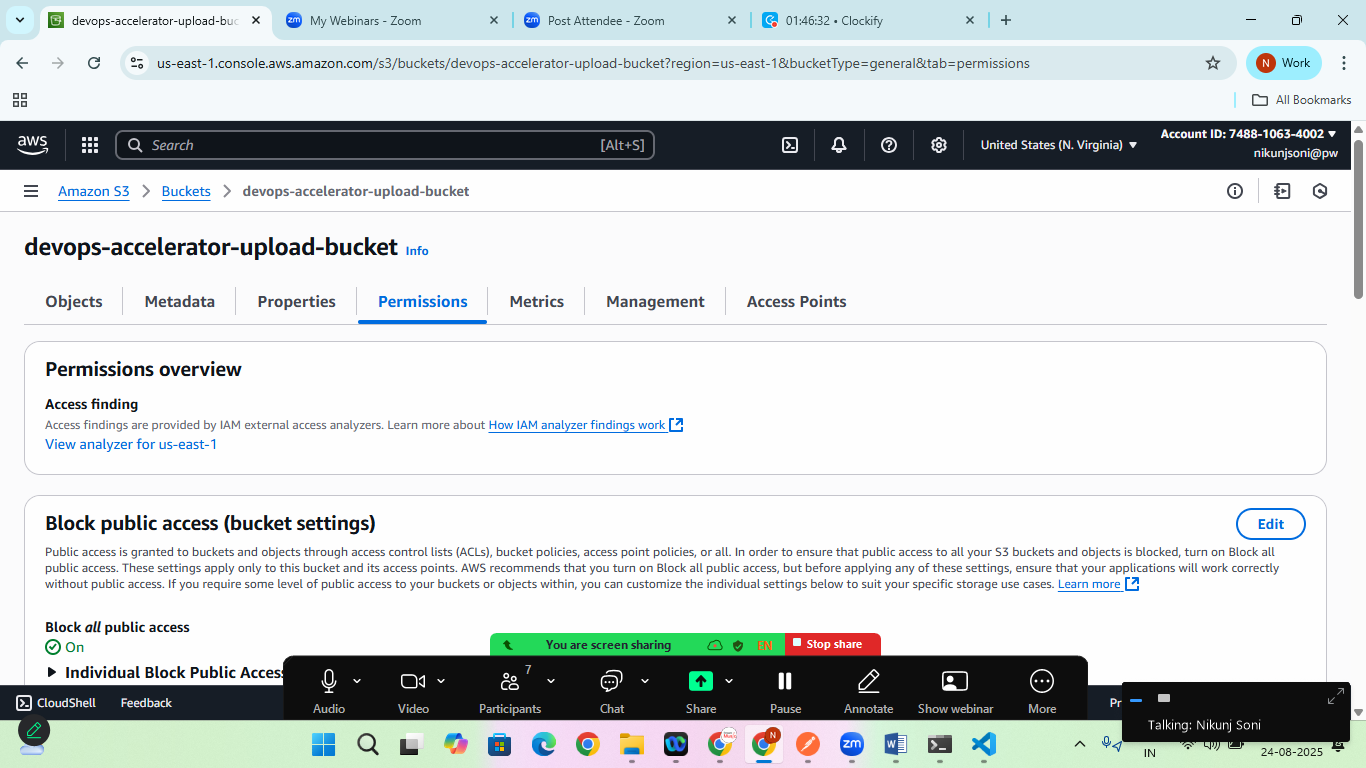
Now copy the cloudfront\_url from github/actions add <https://cloudfront_url> in browser

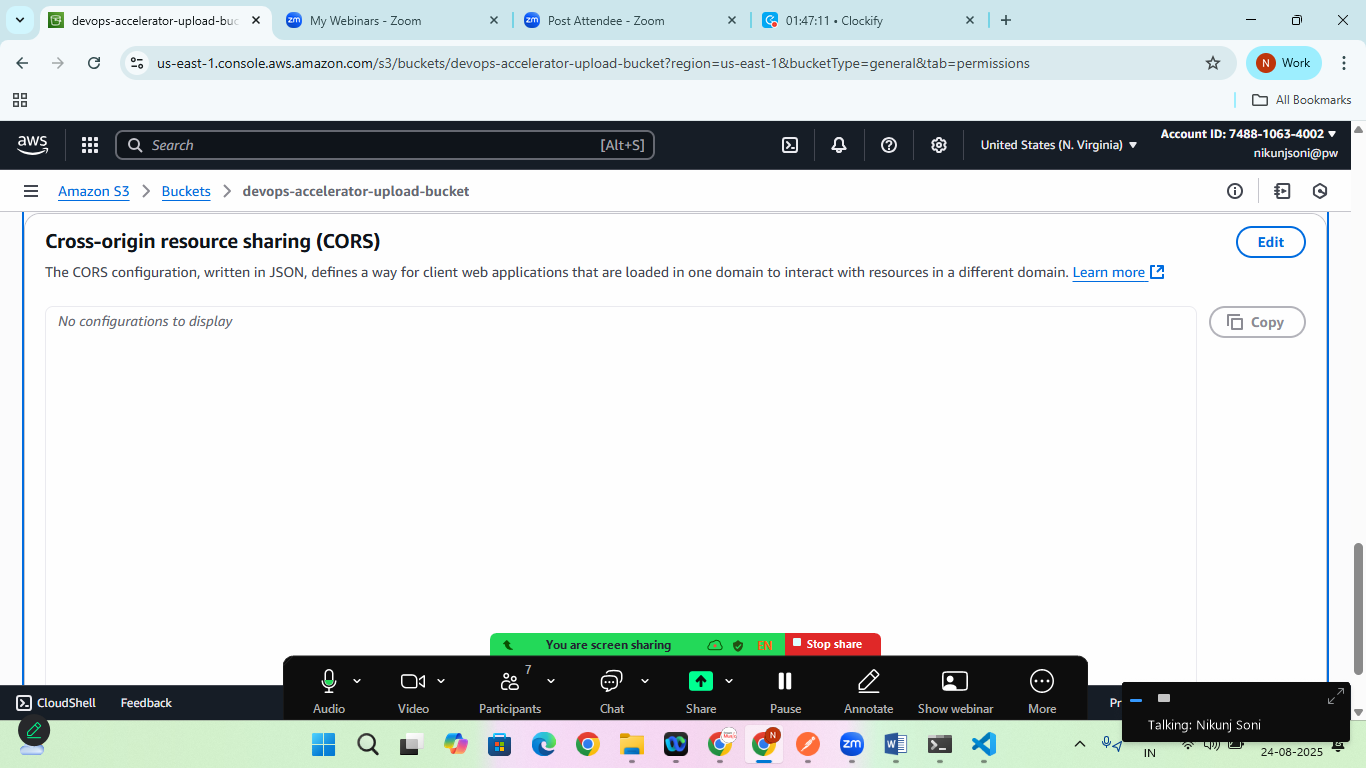




But if you will try to upload something it will cause Cross-Origin Policy Error

So lets goto> s3 bucket>devops-accelerator-upload-bucket> permissions> cross-origin resource sharing





Click on EDIT and

Add the CORS Policy

[

{

"AllowedHeaders": [

"\*"

],

"AllowedMethods": [

"GET",

"PUT",

"POST",

"DELETE",

"HEAD"

],

"AllowedOrigins": [

"\*"

],

"ExposeHeaders": []

}

]

Save it

All Done !