

Capstone Project

Deployment Instruction:

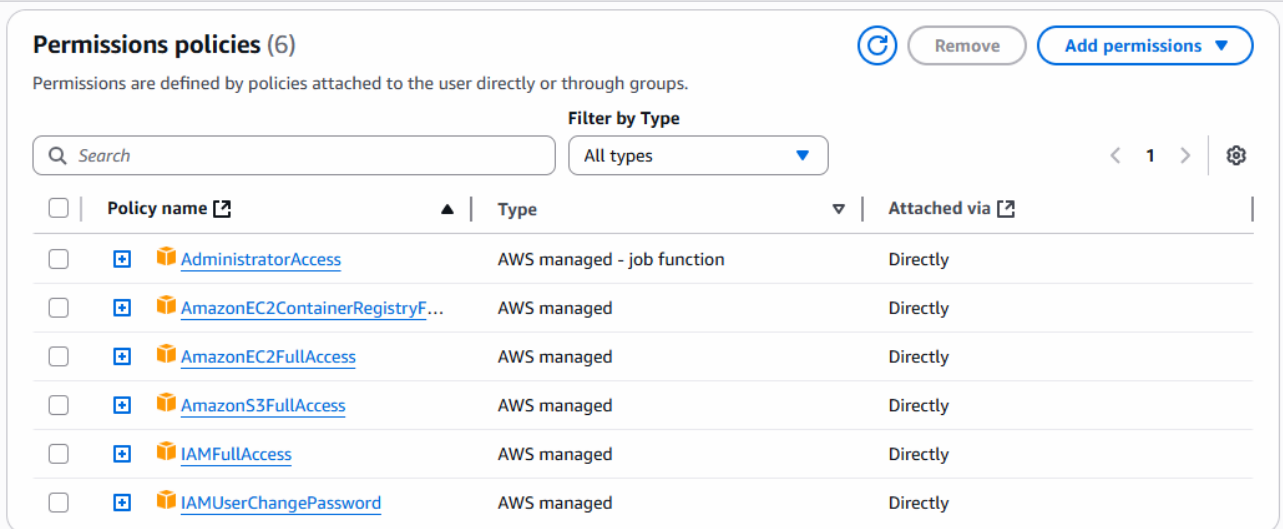
- Create a Project With Fully Working Backend and Frontend
- Upload it on Github and use CI/CD or Jenkins for Automated Build
- Link: https://github.com/Nikunj-Java/Devops_Capstone_PW.git

Prerequisites

- AWS Account(With Administrator 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

- A. Goto AWS Console> IAM > Create User or Choose Existing USER
- B. Create New Credentials for AWS CLI
- C. Add the Below Permission



The screenshot shows the AWS IAM console 'Permissions policies (6)' page. It lists six AWS managed policies attached to a user. The policies are: AdministratorAccess, AmazonEC2ContainerRegistryFullAccess, AmazonEC2FullAccess, AmazonS3FullAccess, IAMFullAccess, and IAMUserChangePassword. Each policy is attached directly to the user.

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly
AmazonEC2ContainerRegistryFullAccess	AWS managed	Directly
AmazonEC2FullAccess	AWS managed	Directly
AmazonS3FullAccess	AWS managed	Directly
IAMFullAccess	AWS managed	Directly
IAMUserChangePassword	AWS managed	Directly

D. Configure aws cli using cmd

```
D:\Capstone_PW_DevOps>aws configure
AWS Access Key ID [*****FSQS]: AKIA24WEXQMJM02GDWN6
AWS Secret Access Key [*****Ls0Q]: 9A3ZW8QCapzKCzUKLq+7TRCt8EJoqBAUwmtPb0XR
Default region name [ap-south-1]: us-east-1
Default output format [json]: json
```

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

Set up Github Repository SECRETS



















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

Add the Below Secrets in Github

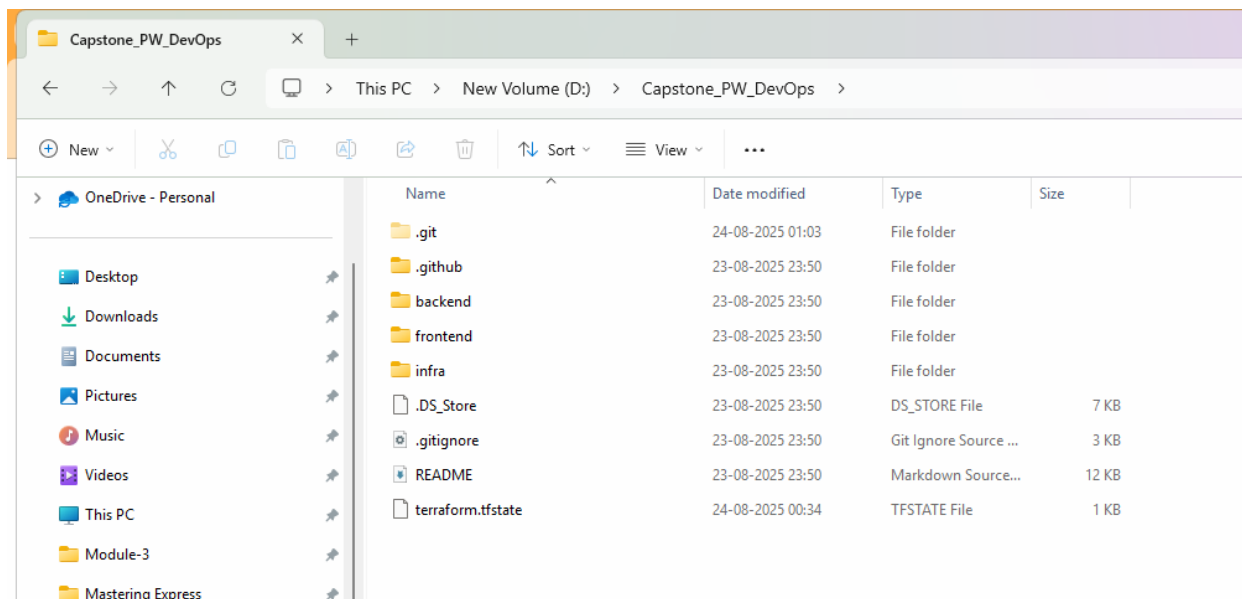
1. AWS_ACCESS_KEY_ID= YOUR AWS IAM access key
2. AWS_SECRET_ACCESS_KEY= Your AWS IAM secret key
3. AWS_REGION= us-east-1
4. LAMBDA_FUNCTION= process-uploaded-file
5. FRONTEND_BUCKET_NAME= devops-accelerator-frontend-hosting-bucket
6. UPLOAD_BUCKET_NAME= devops-accelerator-upload-bucket

Repository secrets

New repository secret

Name ↕	Last updated
 AWS_ACCESS_KEY_ID	11 hours ago  
 AWS_REGION	11 hours ago  
 AWS_SECRET_ACCESS_KEY	11 hours ago  
 FRONTEND_BUCKET_NAME	11 hours ago  
 LAMBDA_FUNCTION_NAME	11 hours ago  
 UPLOAD_BUCKET_NAME	11 hours ago  

Goto> your local repository> where the Code is Available



Open wsl terminal here

```
nikunj@DESKTOP-M4S6FPT: /  
System load: 0.48          Processes: 57  
Usage of /: 1.9% of 1006.85GB Users logged in: 1  
Memory usage: 25%         IPv4 address for eth0: 172.28.70.174  
Swap usage: 0%  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
just raised the bar for easy, resilient and secure K8s cluster deployment.  
  
https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
This message is shown once a day. To disable it please create the  
/home/nikunj/.hushlogin file.  
nikunj@DESKTOP-M4S6FPT: /mnt/d/Capstone_PW_DevOps$ |
```

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

```
terraform  
nikunj@DESKTOP-M4S6FPT: /mnt/d/Capstone_PW_DevOps/infra$ |
```

Step:2 Create required Buckets

#s3 Bucket for state

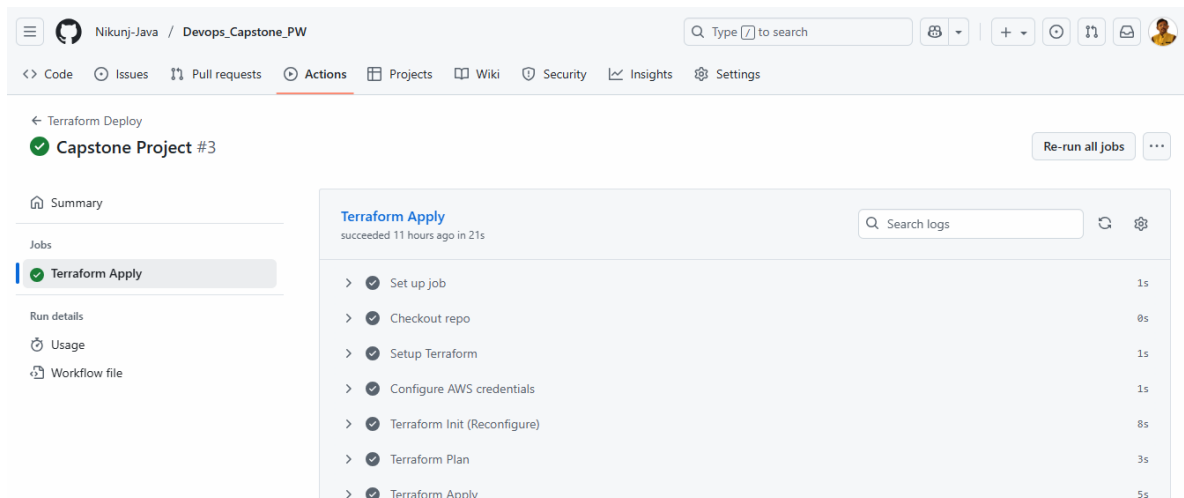
```
aws s3api create-bucket \  
--bucket devops-accelerator-platform-tf-state \  
--region us-east-1
```

```
nikunj@DESKTOP-M4S6FPT:/mnt/d/Capstone_PW_DevOps/infra$ aws s3api create-bucket \  
--bucket devops-accelerator-platform-tf-state \  
--region us-east-1  
{  
  "Location": "/devops-accelerator-platform-tf-state"  
}  
nikunj@DESKTOP-M4S6FPT:/mnt/d/Capstone_PW_DevOps/infra$ |
```

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



The screenshot shows the GitHub Actions interface for a repository named 'Nikunj-Java / Devops_Capstone_PW'. The 'Actions' tab is selected, showing a workflow named 'Terraform Deploy' with a job 'Capstone Project #3'. The job status is 'succeeded 11 hours ago in 21s'. A 'Re-run all jobs' button is visible. On the left, a sidebar lists 'Summary', 'Jobs', 'Run details', 'Usage', and 'Workflow file'. The 'Jobs' section is expanded, showing a list of steps for the 'Terraform Apply' job:

Step	Duration
Set up job	1s
Checkout repo	8s
Setup Terraform	1s
Configure AWS credentials	1s
Terraform Init (Reconfigure)	8s
Terraform Plan	3s
Terraform Apply	5s

- After successfully build move to the next step

Step:4 Run Terraform Commands

- cd infra/terraform
- terraform init
- terraform validate
- terraform plan

```
nikunj@DESKTOP-M4S6FPT:/mnt/d/Capstone_PW_DevOps/infra$ cd terraform/  
nikunj@DESKTOP-M4S6FPT:/mnt/d/Capstone_PW_DevOps/infra/terraform$ terraform init|
```

Now Again Push the Code to Github

Check the Github>action>terraform>TerraformApply>

Check the output

Terraform Apply

succeeded 11 hours ago in 21s

Search logs

✓ Terraform Apply5s

60 }

61

62 Plan: 0 to add, 1 to change, 0 to destroy.

63 aws_apigatewayv2_stage.presign_stage: Modifying... [id=\$default]

64 aws_apigatewayv2_stage.presign_stage: Modifications complete after 0s [id=\$default]

65

66 Apply complete! Resources: 0 added, 1 changed, 0 destroyed.

67

68 Outputs:

69

70 cloudfront_url = "d3acg1frtlx2ff.cloudfront.net"

71 frontend_bucket_name = "devops-accelerator-frontend-hosting-bucket"

72 lambda_function_name = "process-uploaded-file"

73 presigned_url_api_endpoint = "https://1of5m0bve8.execute-api.us-east-1.amazonaws.com"

74 s3_bucket_name = ""

> ✓ Post Configure AWS credentials0s

> ✓ Post Checkout repo0s

> ✓ Complete job0s

Now once it is ready:

Copy presigned_url_api_endpoint and paste in index.html file

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

```
474 try {
475   // Step 1: Get presigned URL
476   const apiUrl = "https://5av0u6haca.execute-api.us-east-1.amazonaws.com/generate-presigned-url";
477   const presignRes = await fetch(apiUrl, {
478     method: "POST",
479     headers: { "Content-Type": "application/json" },
480     body: JSON.stringify({
481       filename: uploadedFile.name,
482       contentType: uploadedFile.type,
483       fullName,
484       email,
```

Now upload this index.html file to s3 bucket

General purpose buckets (3) Info

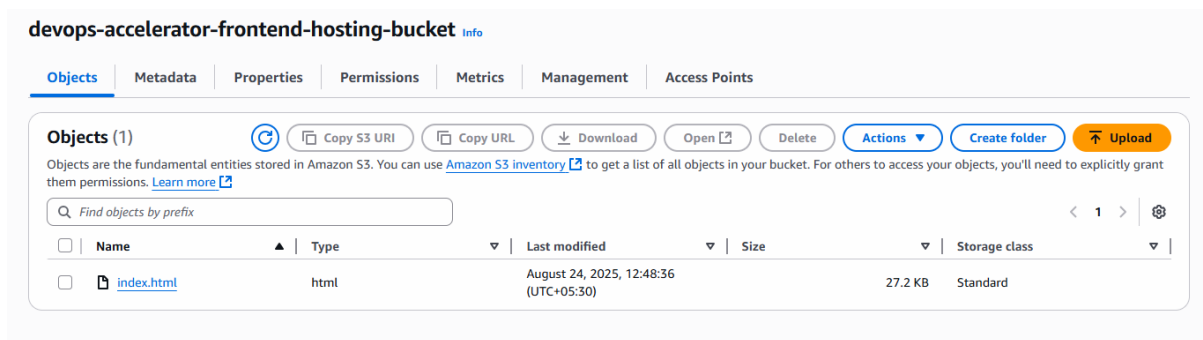
Buckets are containers for data stored in S3.

Find buckets by name

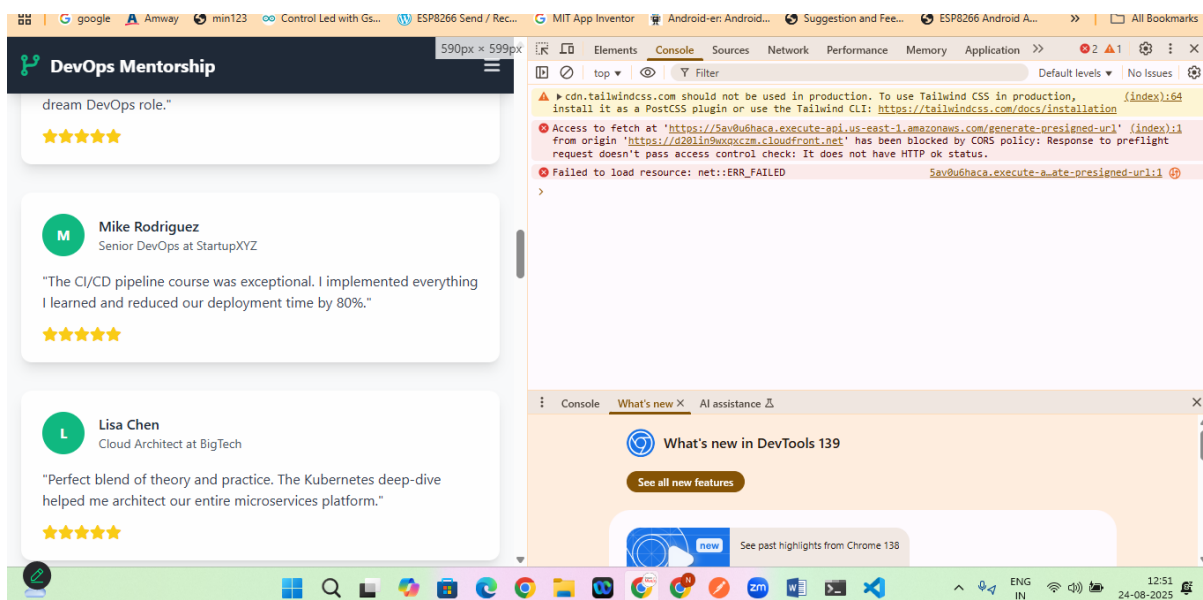
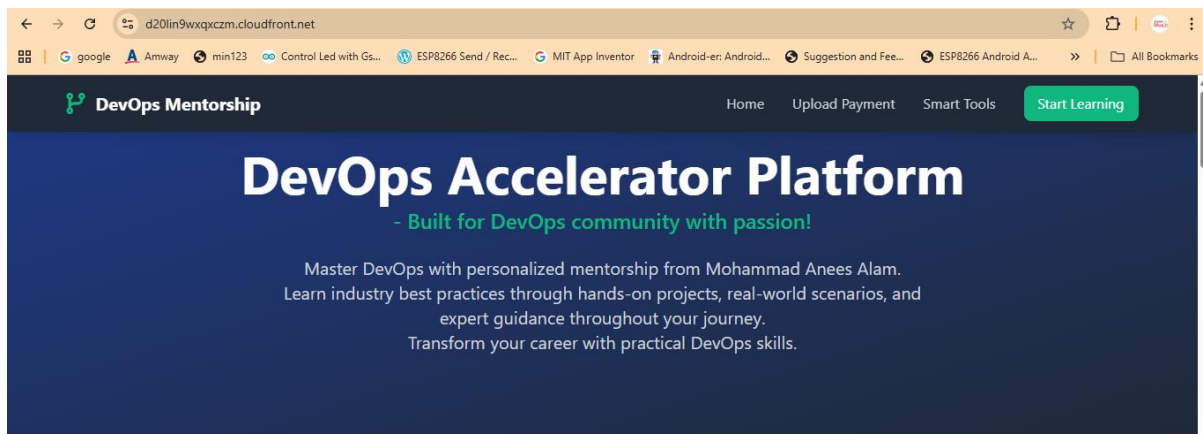
< 1 > ⚙

	Name	▲ AWS Region	▼ Creation date
<input type="radio"/>	devops-accelerator-frontend-hosting-bucket	US East (N. Virginia) us-east-1	August 24, 2025, 12:28:08 (UTC+05:30)
<input type="radio"/>	devops-accelerator-platform-tf-state	US East (N. Virginia) us-east-1	August 24, 2025, 00:00:11 (UTC+05:30)
<input type="radio"/>	devops-accelerator-upload-bucket	US East (N. Virginia) us-east-1	August 24, 2025, 12:28:08 (UTC+05:30)

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

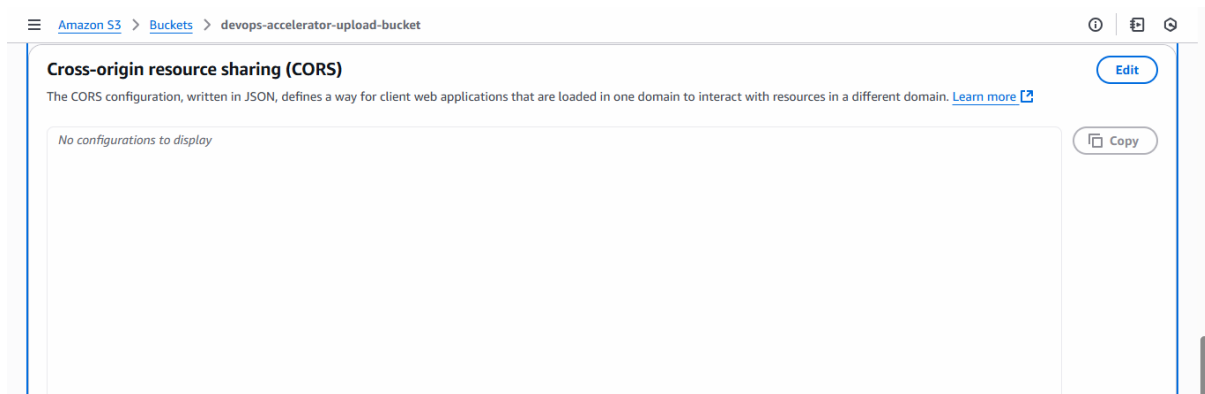
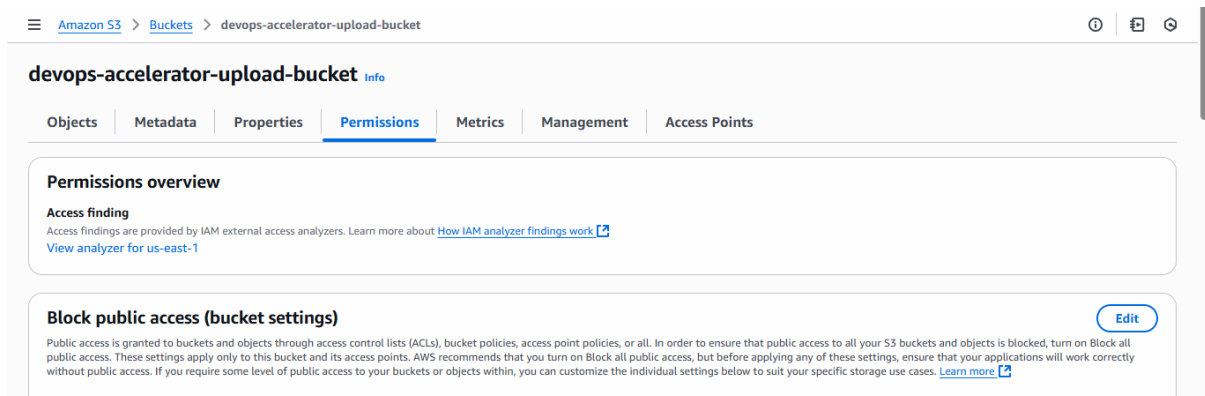


Now copy the cloudfront_url from github/actions add https://cloudfront_url in browser



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": []
}
]

```

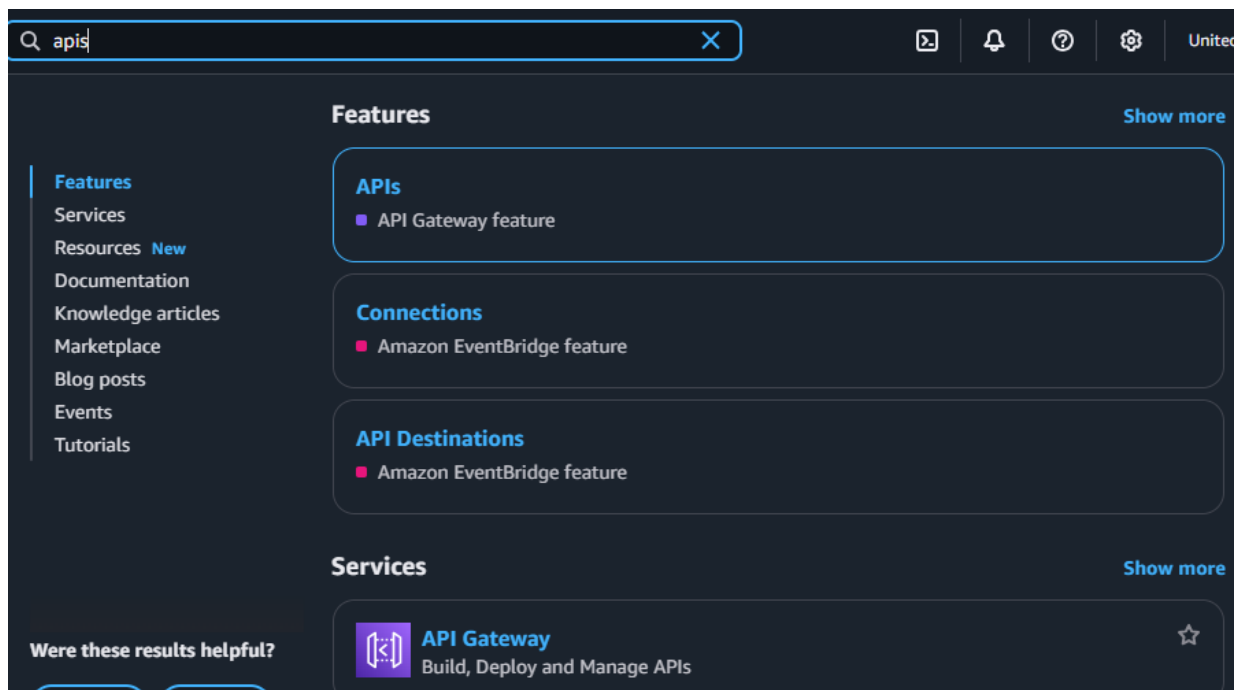
Save it

All Done !

Now check the output and try uploading the image

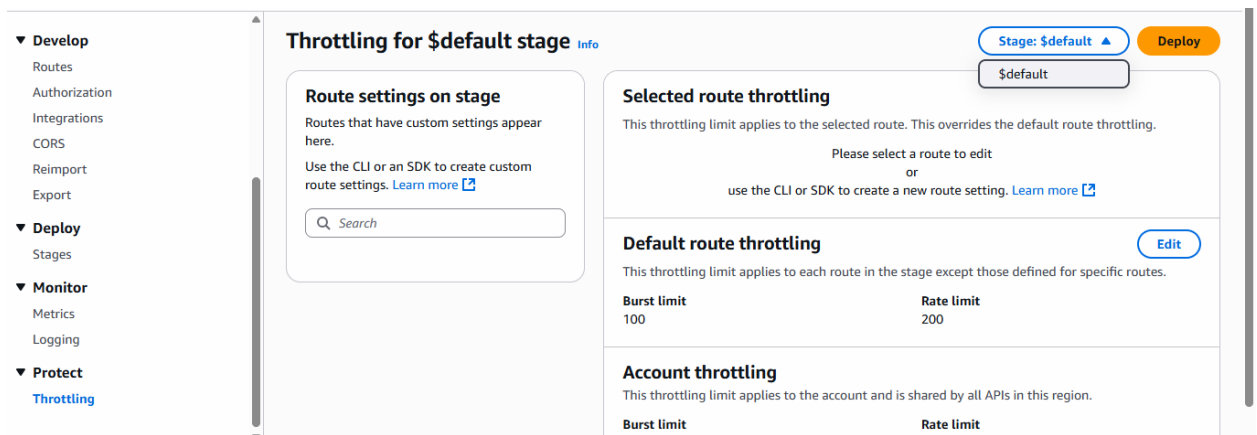
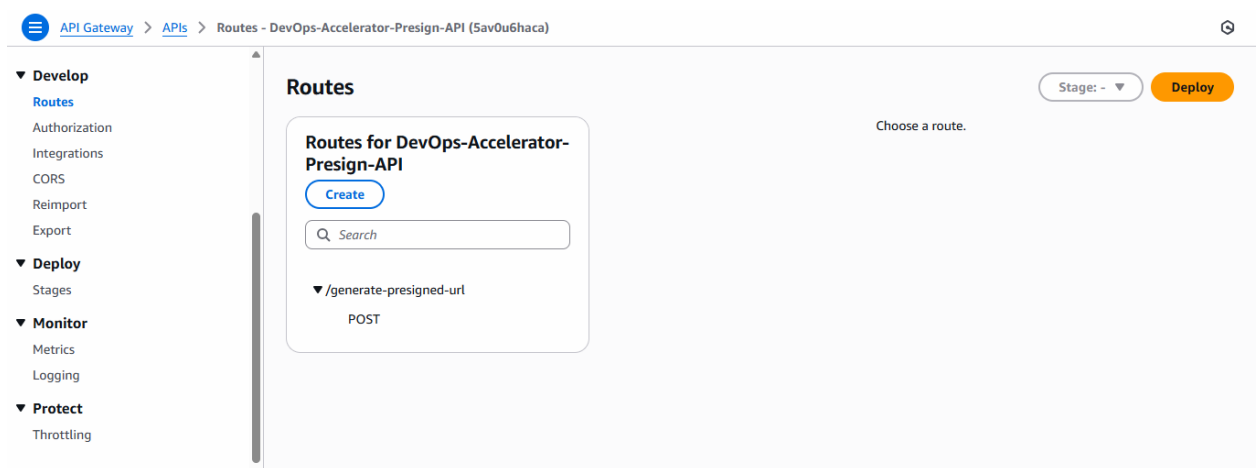
If still error is there

Goto> aws> apis>devops-accelerator-api> click on it



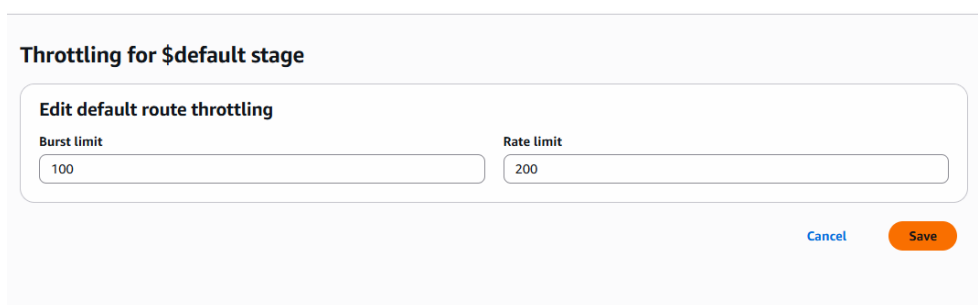
On the left side scroll to bottom you will see **Protect>Throttling**

Select \$default



Choose default throttling >EDIT

And set this values



Save it and all set to go!

Now if you will upload anything it will be Uploaded to the server

Full Name

Nikunj Soni

Email Address

nikunj@testmail.com

Payment Screenshot



Click to upload or drag and drop

JPG, PNG, or PDF (max 10MB)



Upload successful!

307624.png

Additional Notes (Optional)