MINI-PROJECT-1

NAME :-

Upload files into a S3 Bucket using a REST API via API Gateway.

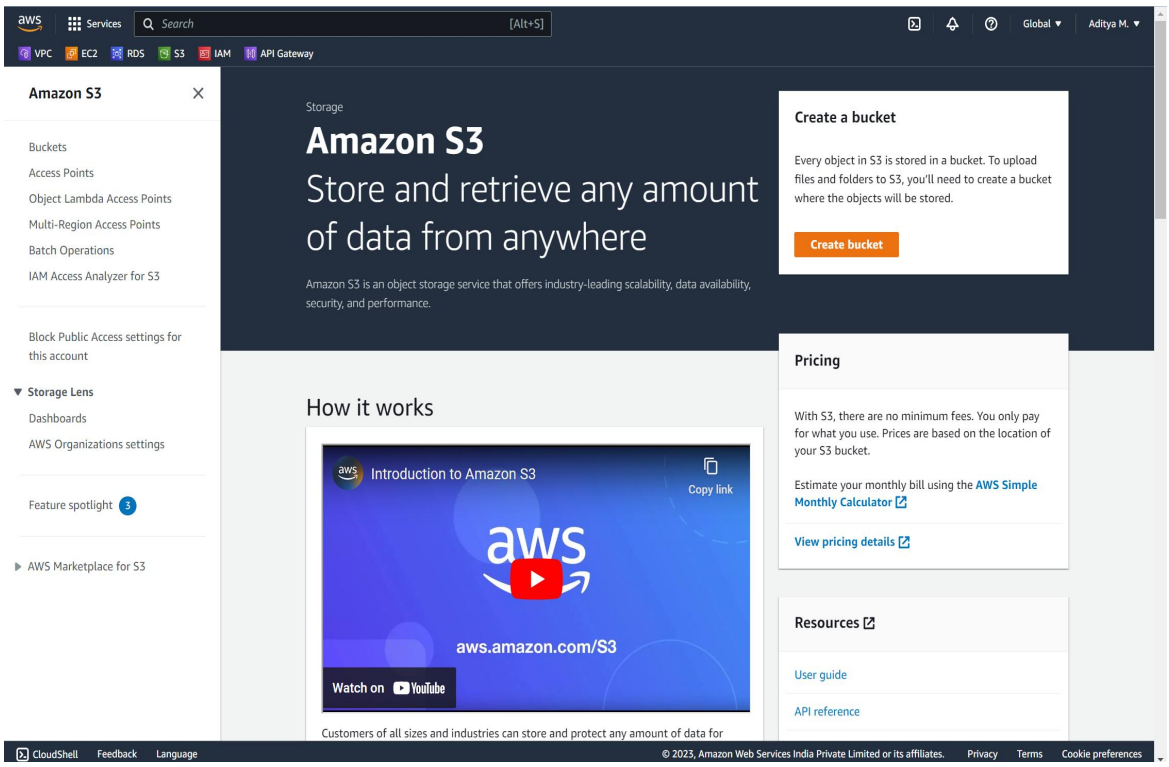
USE CASE :-

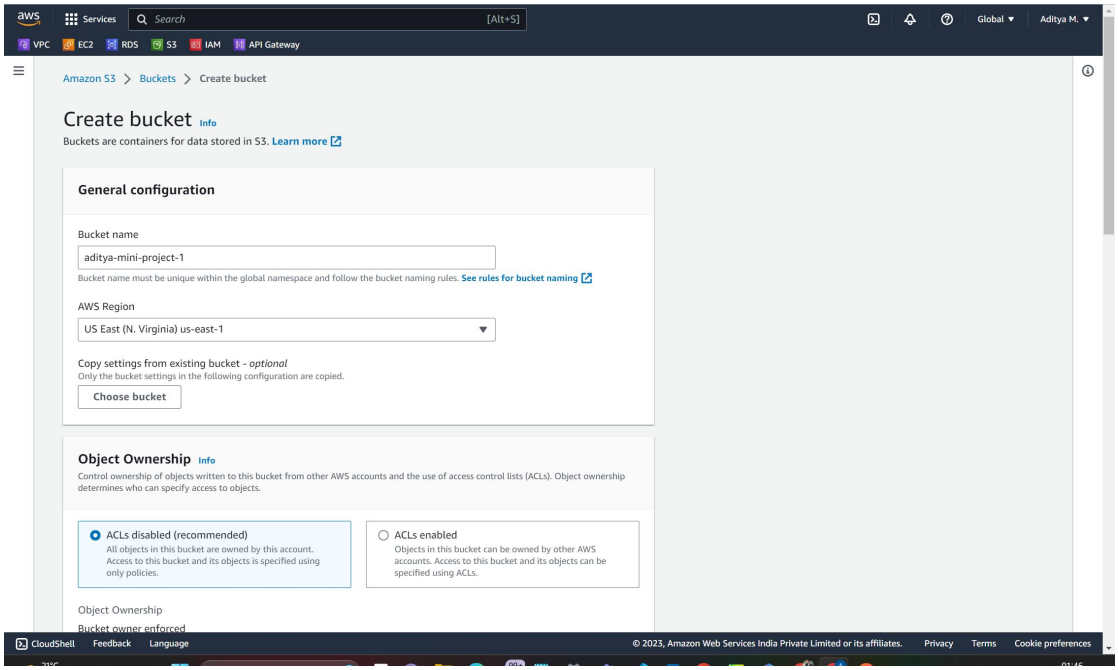
There are times when we need to accept file uploads on our websites. For this we setup the following rest API to upload files in S3 bucket.

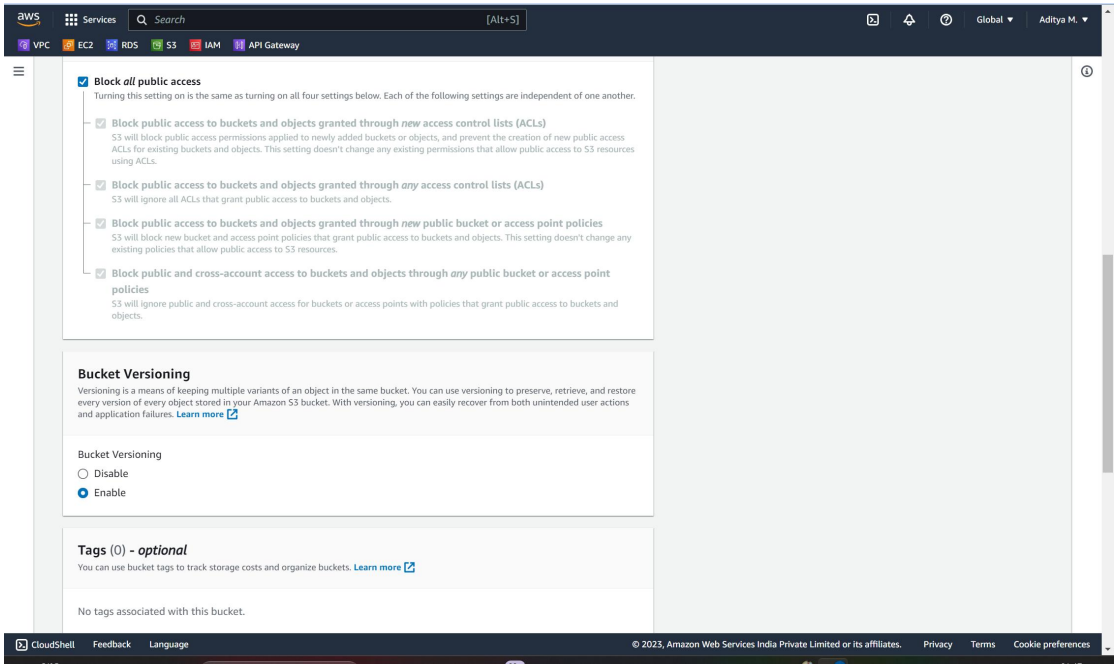
1) S3 Bucket :- Create a S3 bucket to hold our uploaded files.
2) IAM :- Create an IAM role an attach policy to it.
3) API Gateway :- Create a REST API layer to accept PUT requests for file uploads.

STEPS TO FOLLOW:

1) Open S3 service. Click on create bucket. Name the bucket and keep the rest of the settings as it is. Only changes you have to make is enable versioning and create bucket.
(Note:- Bucket name must be globally unique.)







2) In the next step, we have to create an IAM role and attach a policy to it.

3) Open the IAM service. Click on the create roles. Then select the API Gateway service from "use cases for other AWS services" and click next. Give a name and create the role.

4) Now role is created. Now we have to attach a polity to the role.

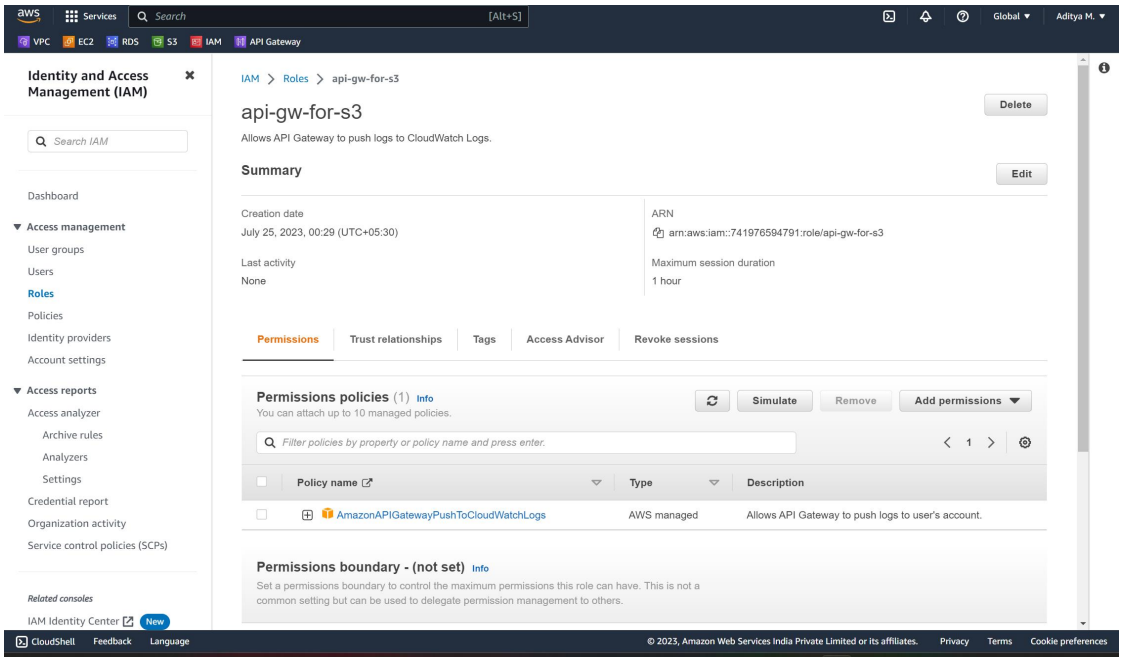
5) In the policy tab on the left hand side of IAM service click on create policy. Then choose the service S3.

6) Then select an action "PUTOBJECT" and in resources add "ARN".

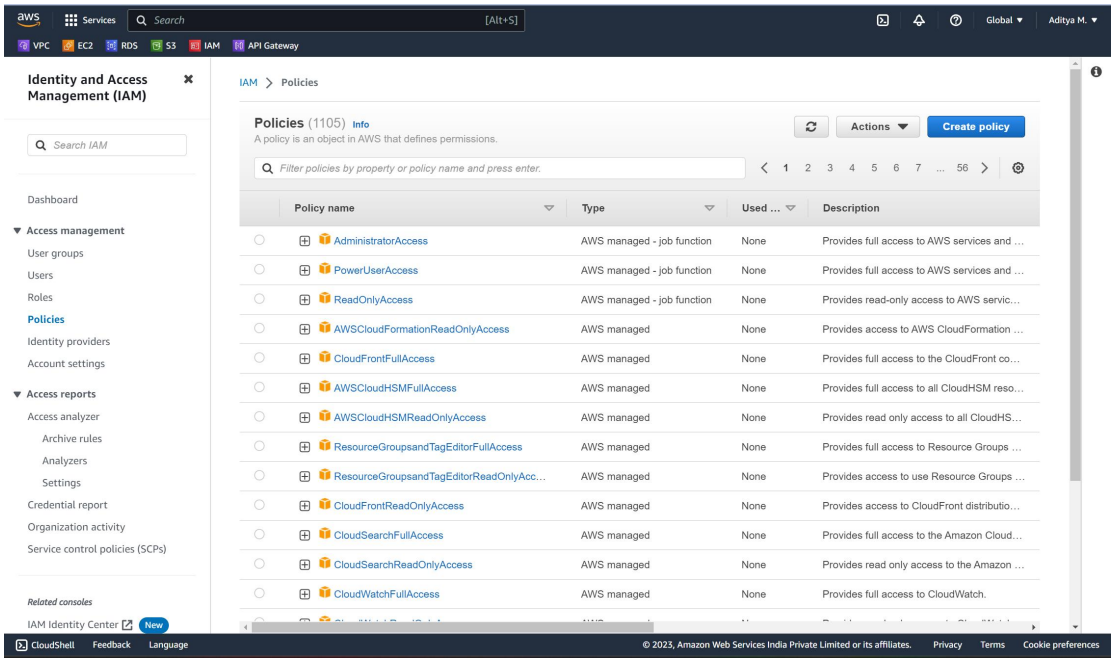
7) In ARN tab put your bucket name and click on any in object name. Give a name to your policy and click on create.

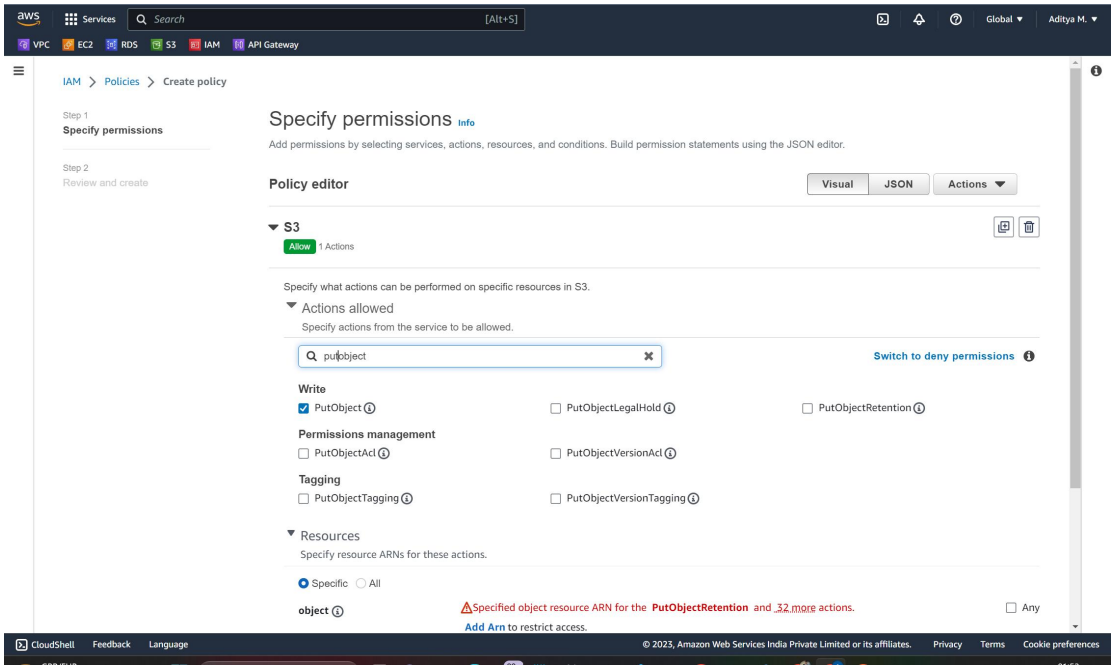
8) After that attach the policy to the role which we have created previously.

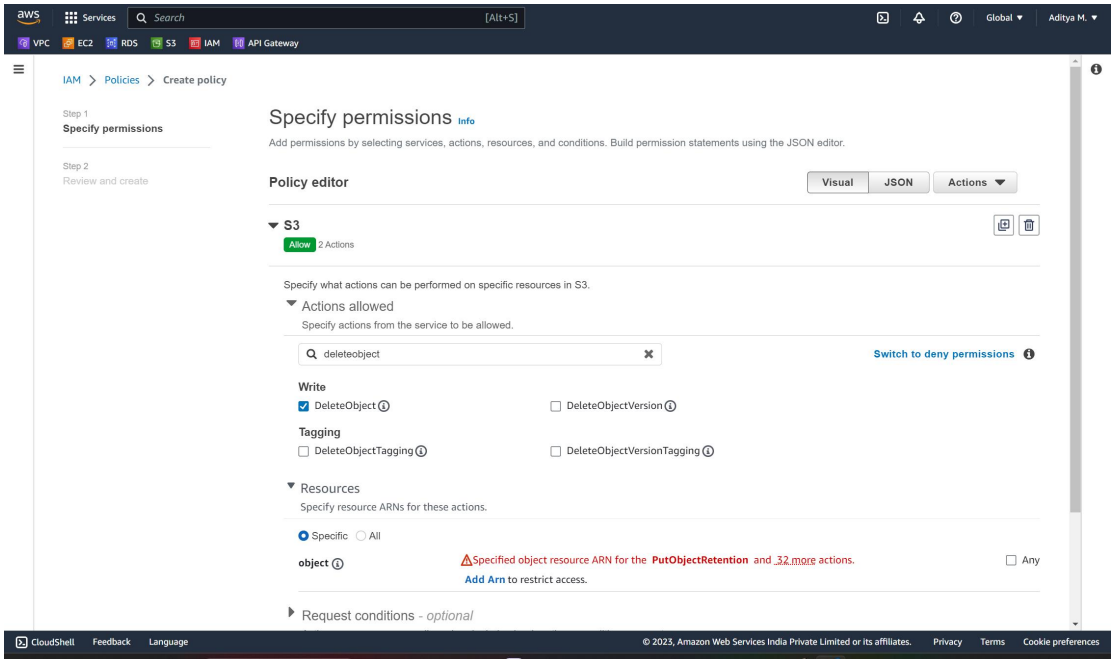
9) In roles, select your created role and click on add permission and then select the policy which we have created recently and save. Now your IAM ROLE is created and working perfectly.
(Note:- Copy the ARN from the roles. We need that ARN to setup API Gateway.)

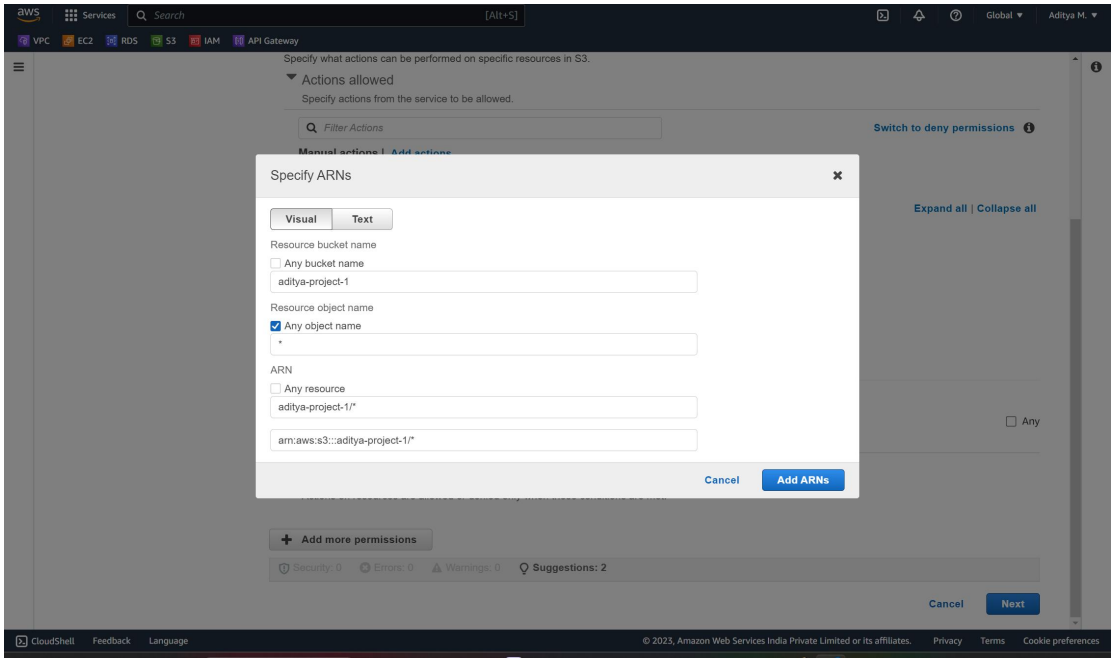


ADITYA MAWALKAR

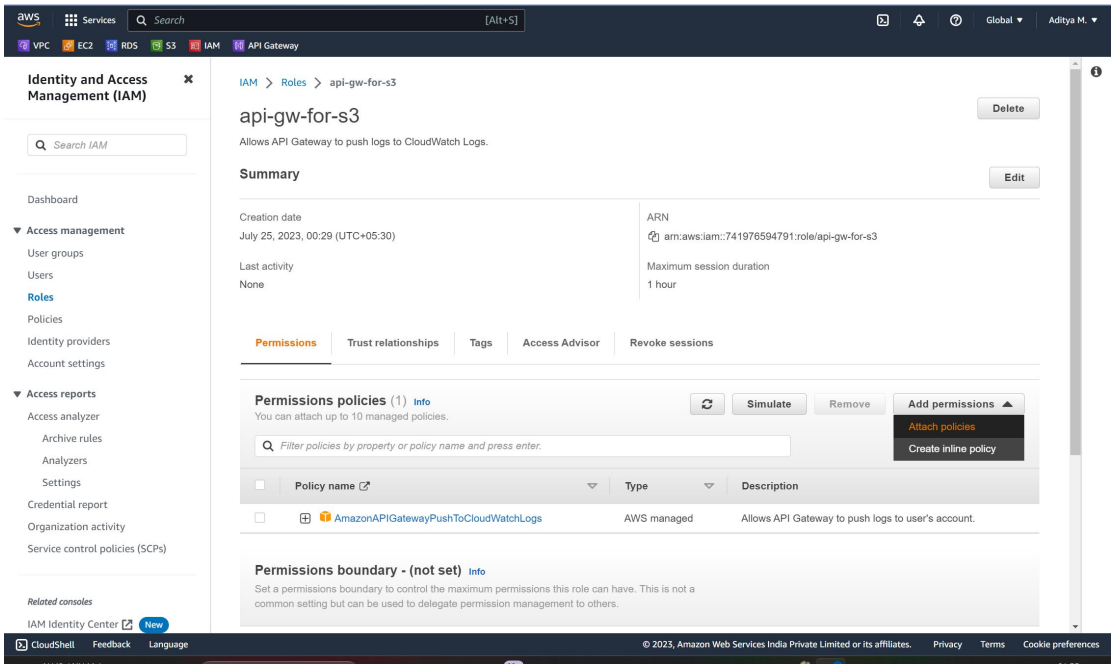


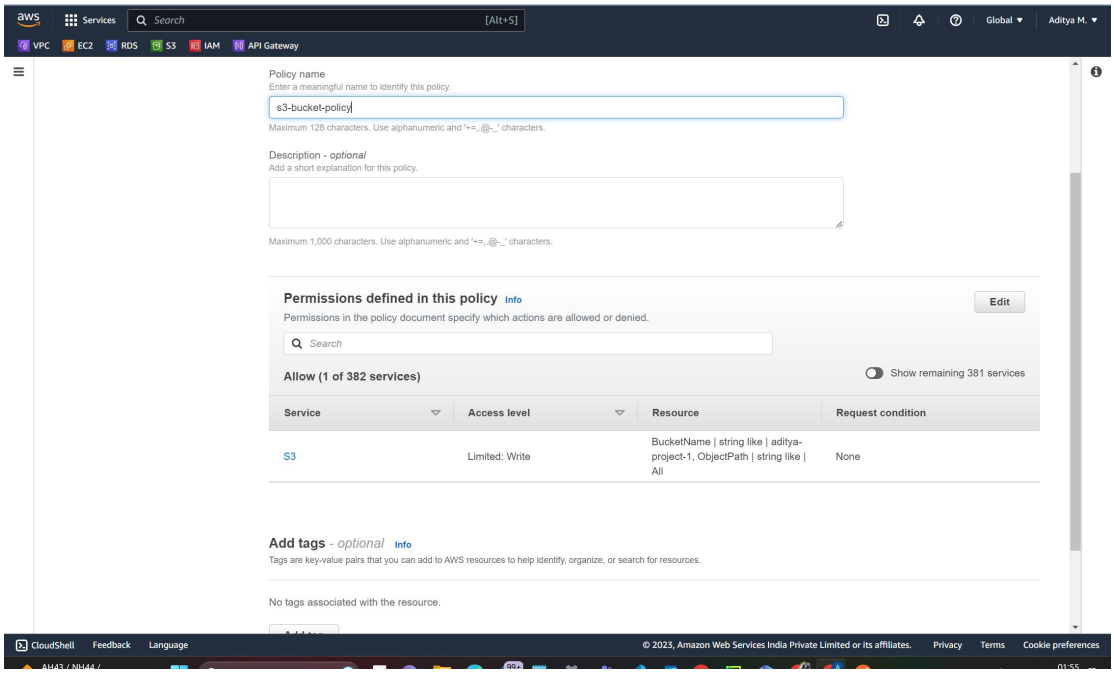


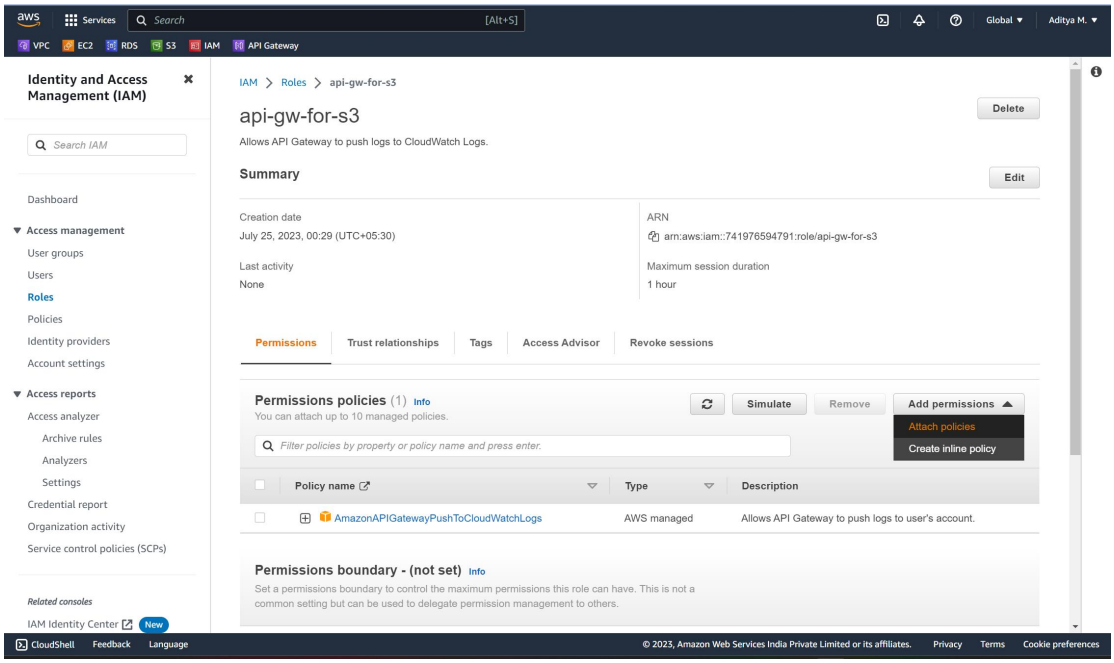

aws
Services
Search
[Alt+S]
Global
Aditya M.
VPC
EC2
RDS
S3
IAM
API Gateway
IAM > Policies > Create policy
Step 1
Specify permissions
Step 2
Review and create
Specify permissions Info
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.
Policy editor
Visual
JSON
Actions
S3
Allow 2 Actions
Specify what actions can be performed on specific resources in S3.
Actions allowed
Specify actions from the service to be allowed.
deleteobject
Switch to deny permissions
Write
DeleteObject
DeleteObjectVersion
Tagging
DeleteObjectTagging
DeleteObjectVersionTagging
Resources
Specify resource ARNs for these actions.
Specific
All
object
Specified object resource ARN for the PutObjectRetention and 32 more actions.
Add Arn to restrict access.
Any
Request conditions - optional
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

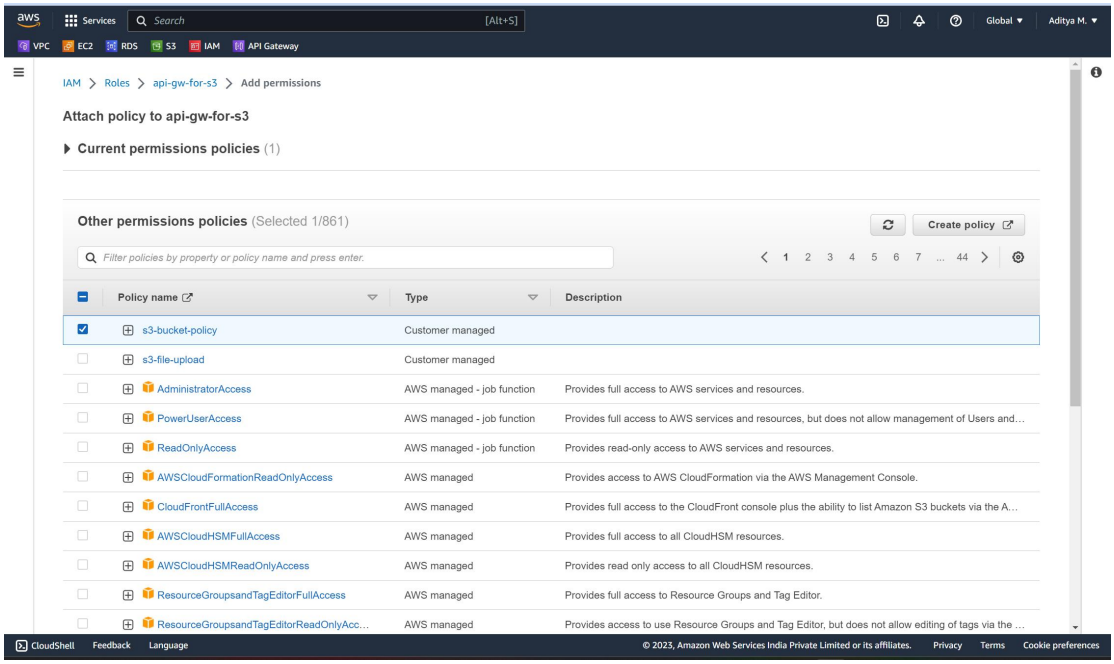

Specify what actions can be performed on specific resources in S3.
Actions allowed
Specify actions from the service to be allowed.
Filter Actions
Switch to deny permissions
Specify ARNs
Visual
Text
Resource bucket name
Any bucket name
aditya-project-1
Resource object name
Any object name
*
ARN
Any resource
aditya-project-1/*
arn:aws:s3:::aditya-project-1/*
Cancel
Add ARNs
Expand all | Collapse all
Any
Add more permissions
Suggestions: 2
Cancel
Next

ADITYA MAWALKAR









10) Now the next step is, go to the API Gateway service. create the "REST API".

11) In this Create a new API.

12) After that click on actions and create resources with the name of bucket and in the resource path "{bucket}" enable CORS and create the resource.

13) Then yo have create another resource under that with the name of filename and in the resource path "{filename}" enable CORS and create the resource.

14) After this step, lets create PUT method under the filename resource .

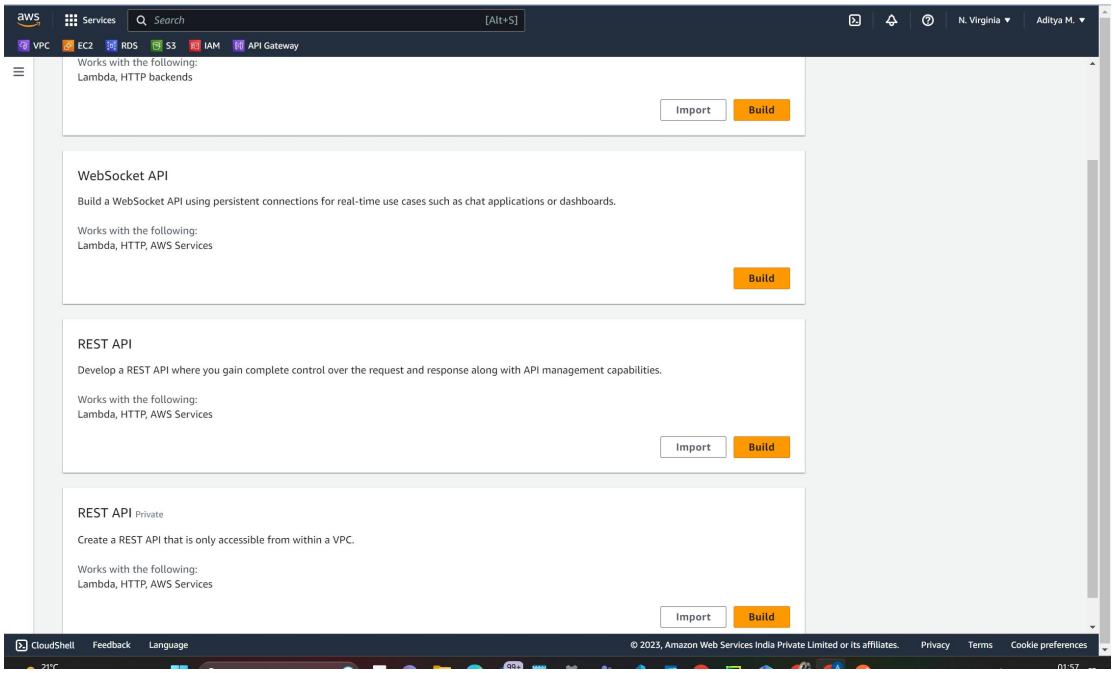
15) Go to the action click create method and select "PUT". In this select "AWS Services" and then select a preffered AWS region, AWS service is "S3", HTTP method is "PUT", action type "use path override", then in path override put a path "{bucket}/{filename}", and then put the ARN which we copied from roles in the execution role and click save.

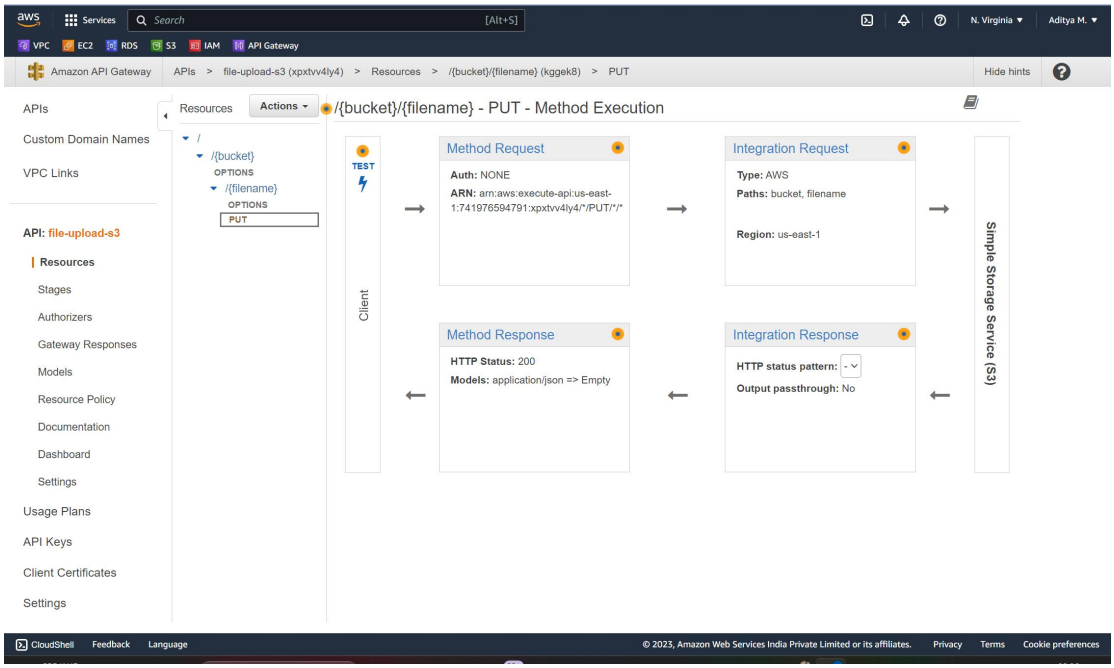
16) Next thing is, in PUT method go to the integration method and add path in the "URL Path Parameters".

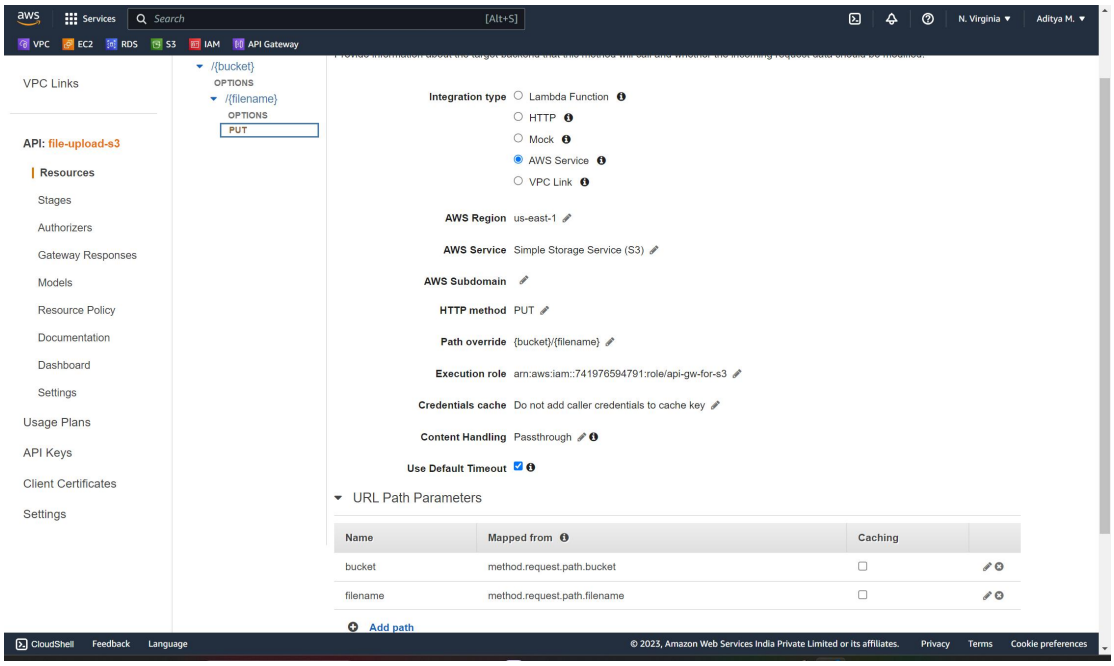
17) Then go to the settings and add "binary media types" and save.

18) Now our API is successfully created. Next is go and deploy the API from the action drop-down menu. Name the deployment stage as "[New Stage]" and deploy.

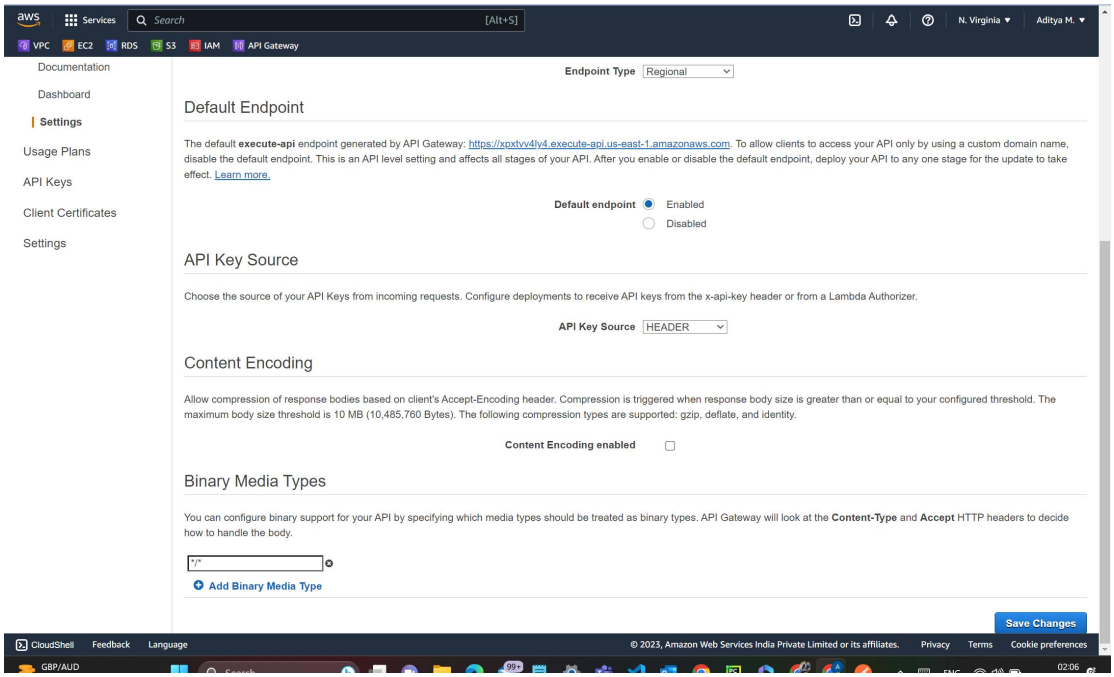
19) Copy the API endpoint and test it in the POSTMAN.

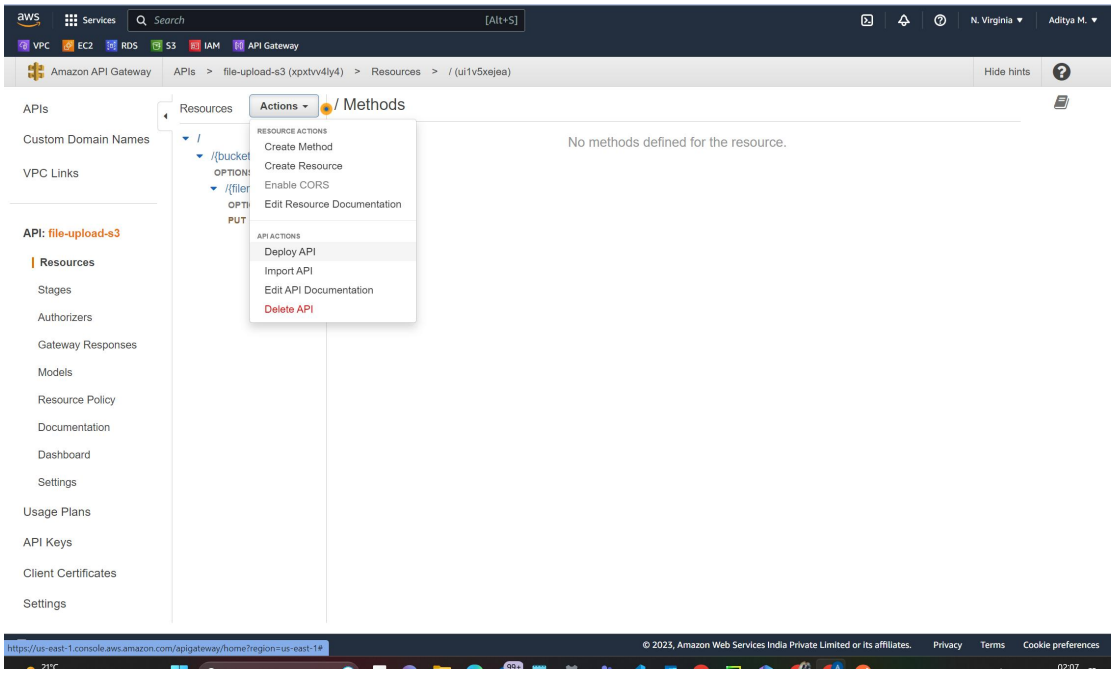


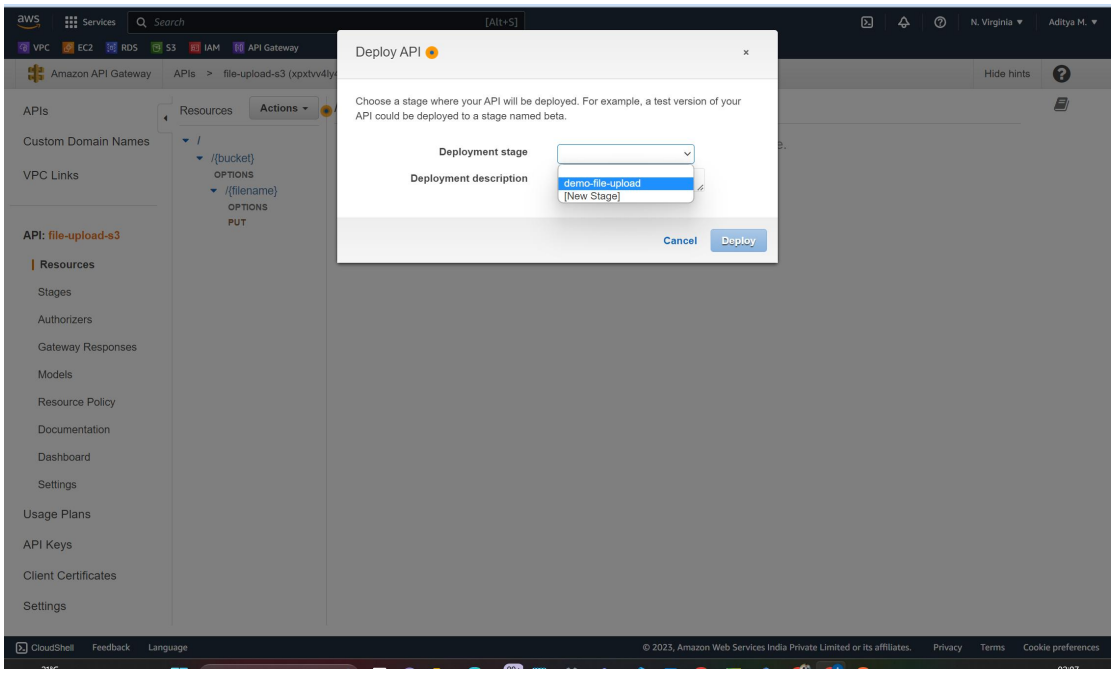

Services
Search
[Alt+S]
N. Virginia
Aditya M.
VPC
EC2
RDS
S3
IAM
API Gateway
Amazon API Gateway
APIs > file-upload-s3 (xpxtvv4ly4) > Resources > /{bucket}/{filename} (kggek8) > PUT
Hide hints
APIs
Custom Domain Names
VPC Links
API: file-upload-s3
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys
Client Certificates
Settings
Resources
Actions
/{bucket}/{filename} - PUT - Method Execution
/
/{bucket}
OPTIONS
/{filename}
OPTIONS
PUT
TEST
Client
Method Request
Auth: NONE
ARN: arn:aws:execute-api:us-east-1:741976594791:xpxtvv4ly4/*/PUT/*/*
Integration Request
Type: AWS
Paths: bucket, filename
Region: us-east-1
Simple Storage Service (S3)
Method Response
HTTP Status: 200
Models: application/json => Empty
Integration Response
HTTP status pattern:
Output passthrough: No
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

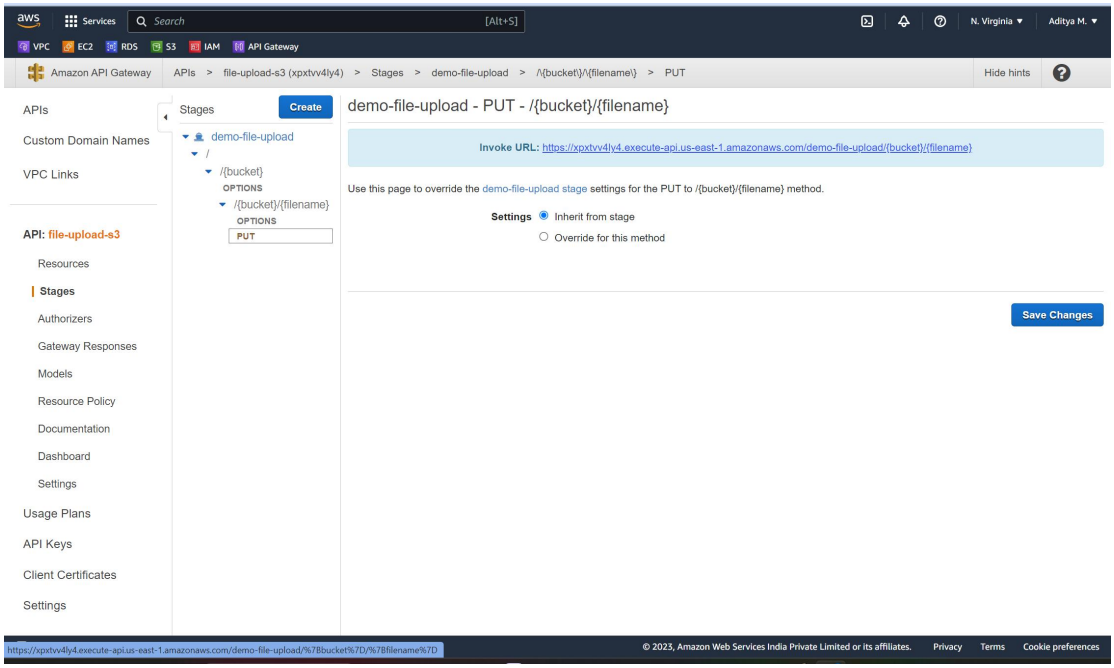

Services
Search
[Alt+S]
N. Virginia
Aditya M.
VPC
EC2
RDS
S3
IAM
API Gateway
VPC Links
API: file-upload-s3
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys
Client Certificates
Settings
/{bucket}
OPTIONS
/{filename}
OPTIONS
PUT
Integration type
Lambda Function
HTTP
Mock
AWS Service
VPC Link
AWS Region us-east-1
AWS Service Simple Storage Service (S3)
AWS Subdomain
HTTP method PUT
Path override {bucket}/{filename}
Execution role arn:aws:iam::741976594791:role/api-gw-for-s3
Credentials cache Do not add caller credentials to cache key
Content Handling Passthrough
Use Default Timeout
URL Path Parameters
Name
Mapped from
Caching
bucket
method.request.path.bucket
filename
method.request.path.filename
Add path
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

ADITYA MAWALKAR







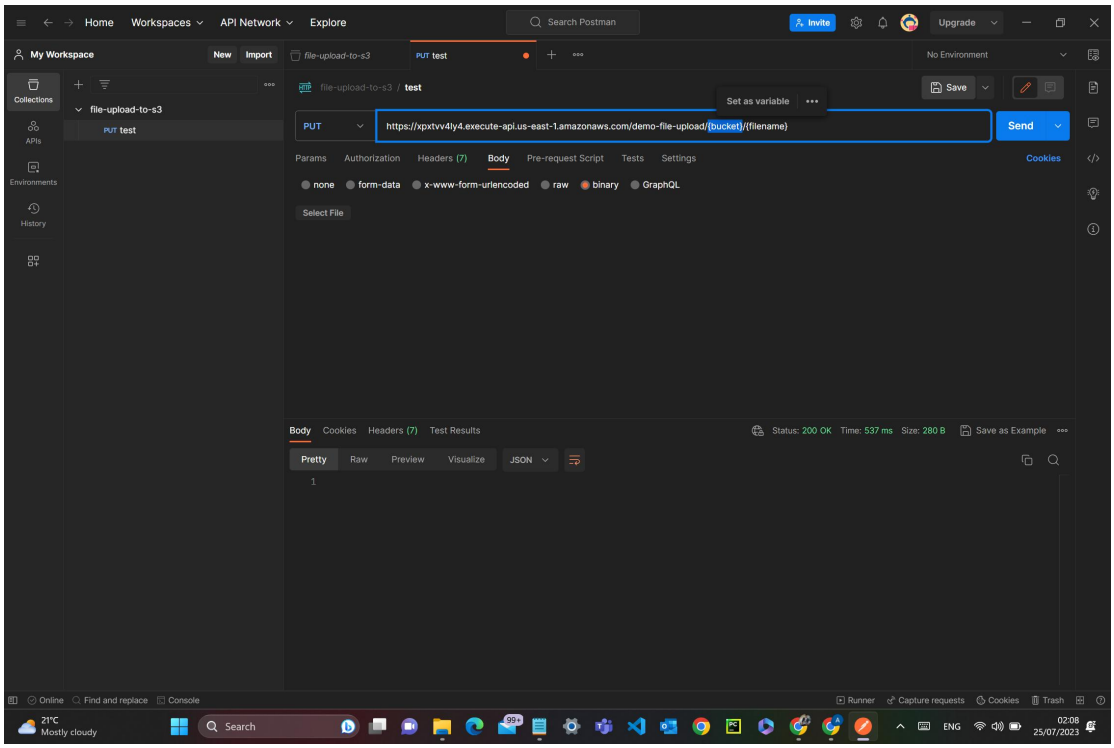


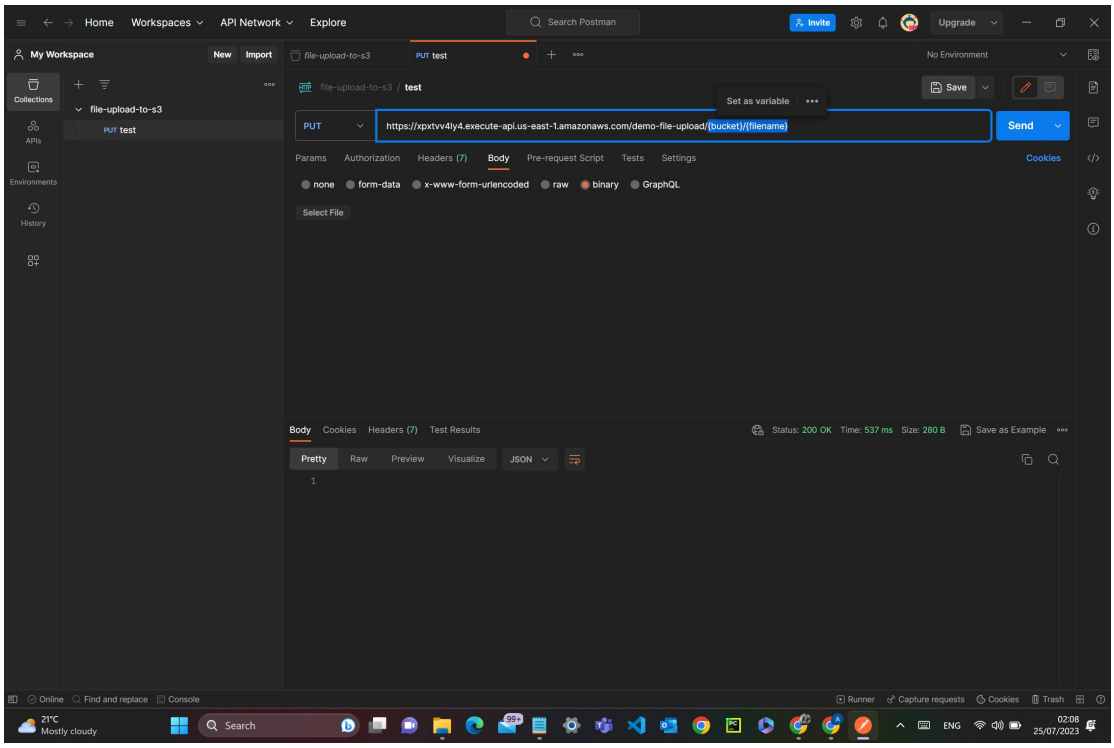
20) Go to the POSTMAN and create a new workspace and in that create a "PUT" request.

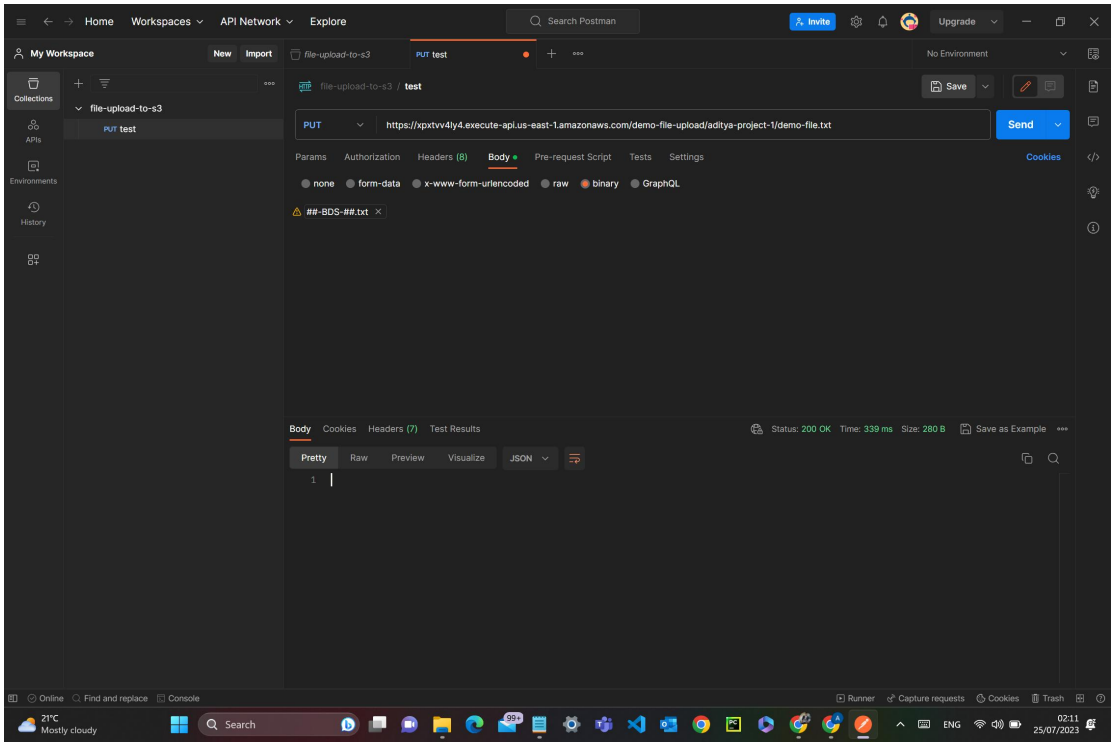
21) Paste the API endpoint and click on body -> binary and select the file and send.

(Note:- Put you bucket name in "{bucket}" and any random filename in "{filename}".)

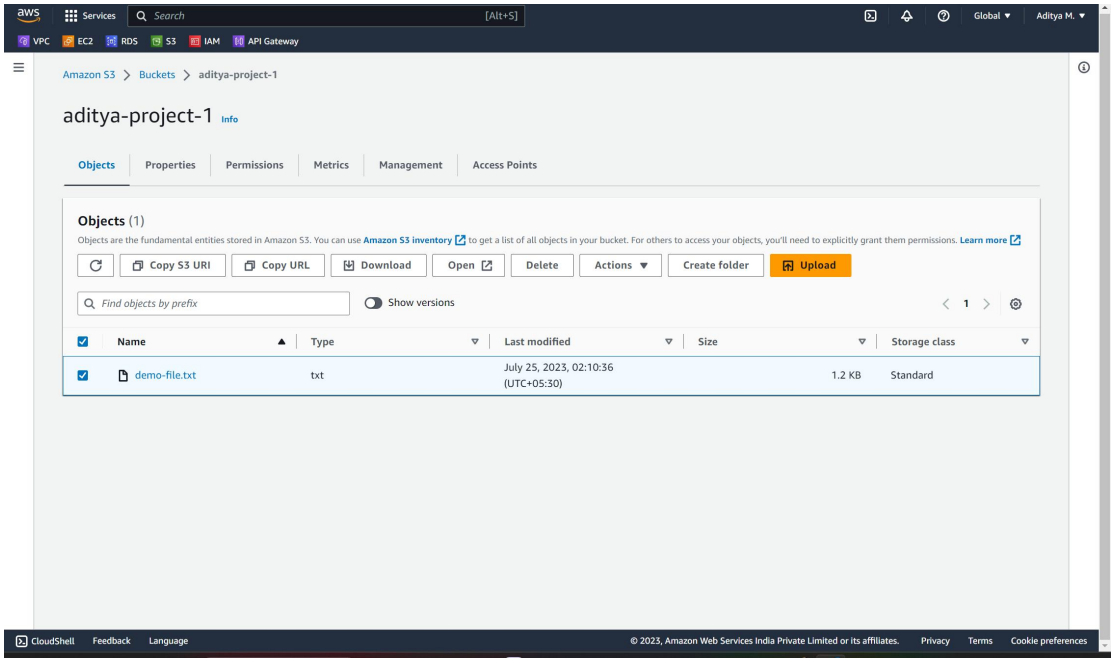
22) If status is 200 Ok and in body "1" then the file is upload to the S3 bucket.







23) To check the output go to the S3 service and open your bucket you can see the file uploaded in the bucket.



END :-

So that's how you set a file upload via API Gateway into S3 bucket.