



Ganpat University

॥ विद्यया समाजोत्कर्षः ॥

Institute of Computer Technology

Name: Tushar Panchal

En.No: 21162101014

Sub: CCE(Cloud Computing Essentials)

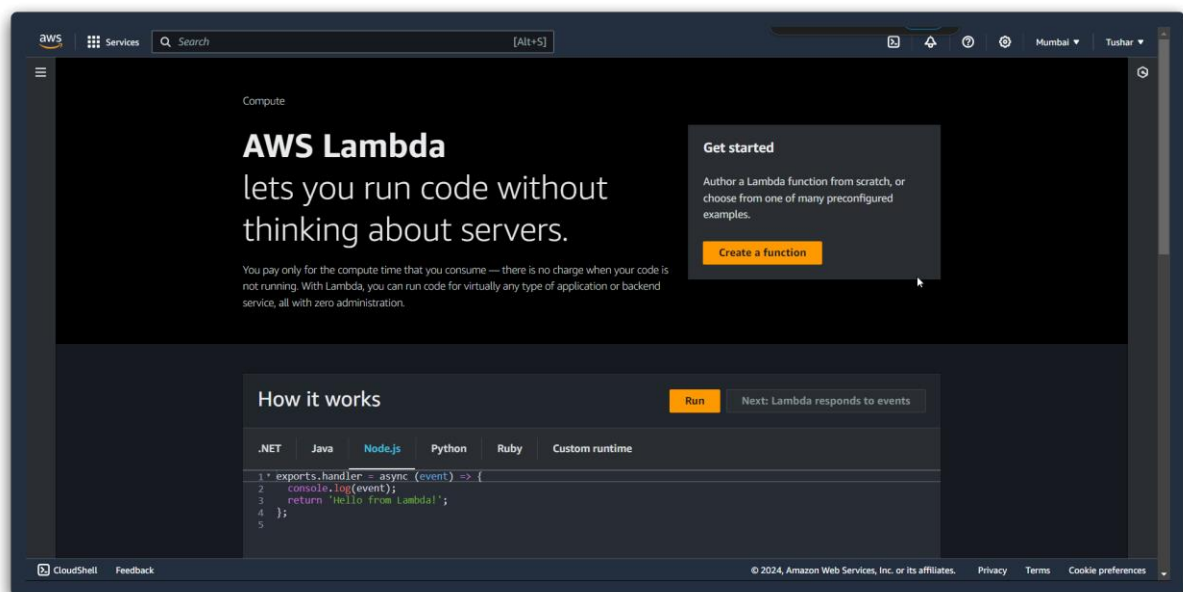
Branch: CBA

Batch:61

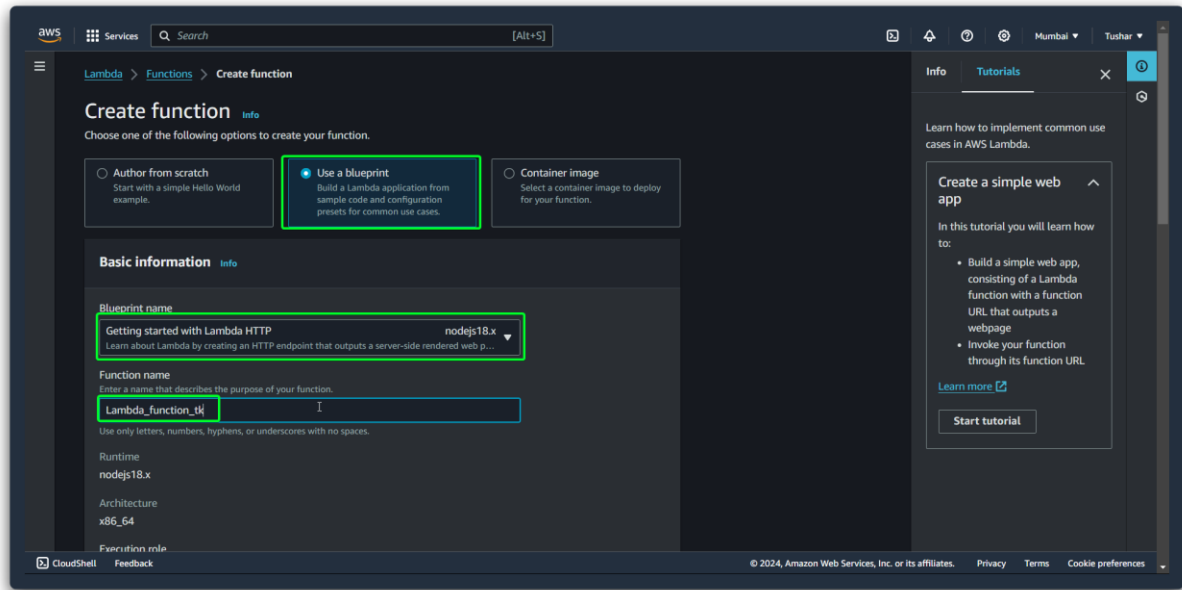
-----PRACTICAL 08-----

Demonstration of Serverless Lambda Service in AWS. Also create a function.

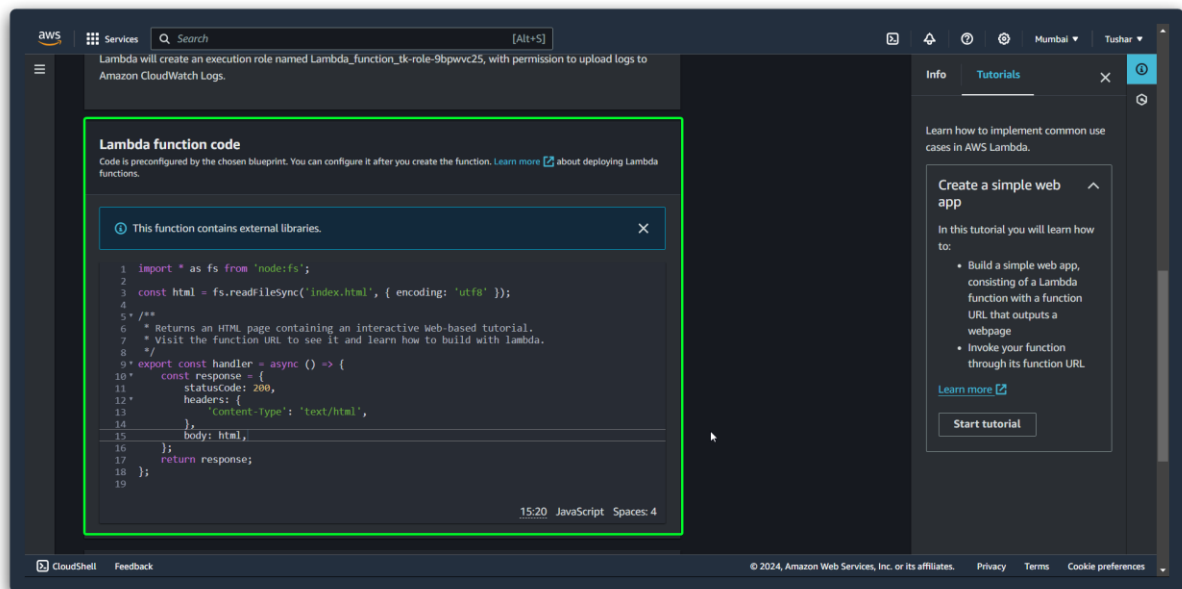
» **Navigate to Lambda service in catalogue**



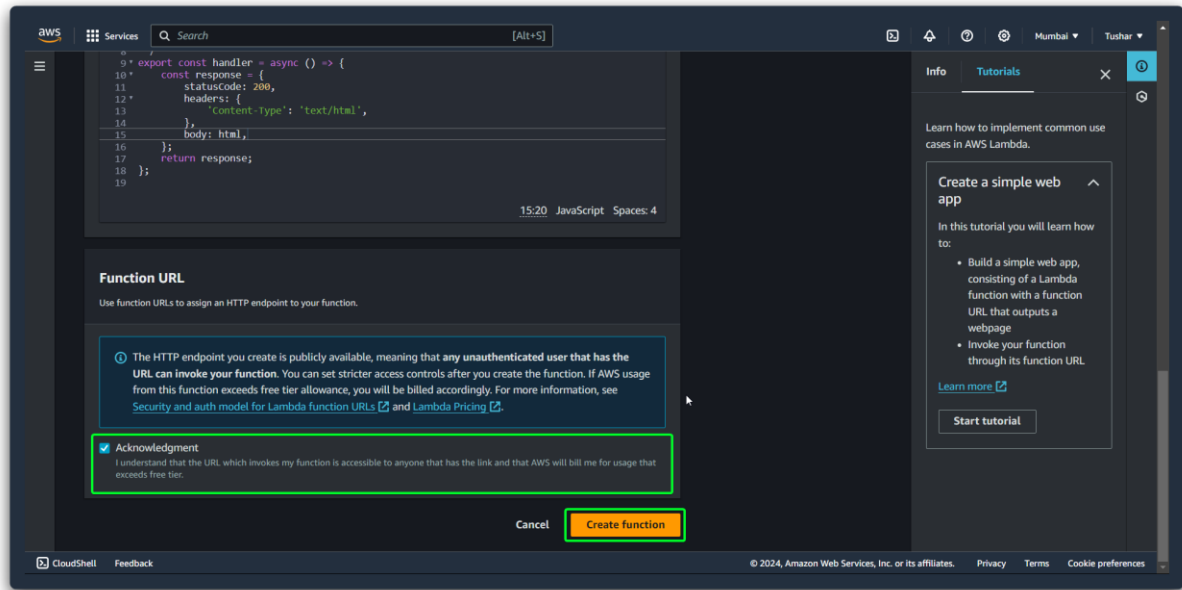
» **Select blueprint set default and set function name**



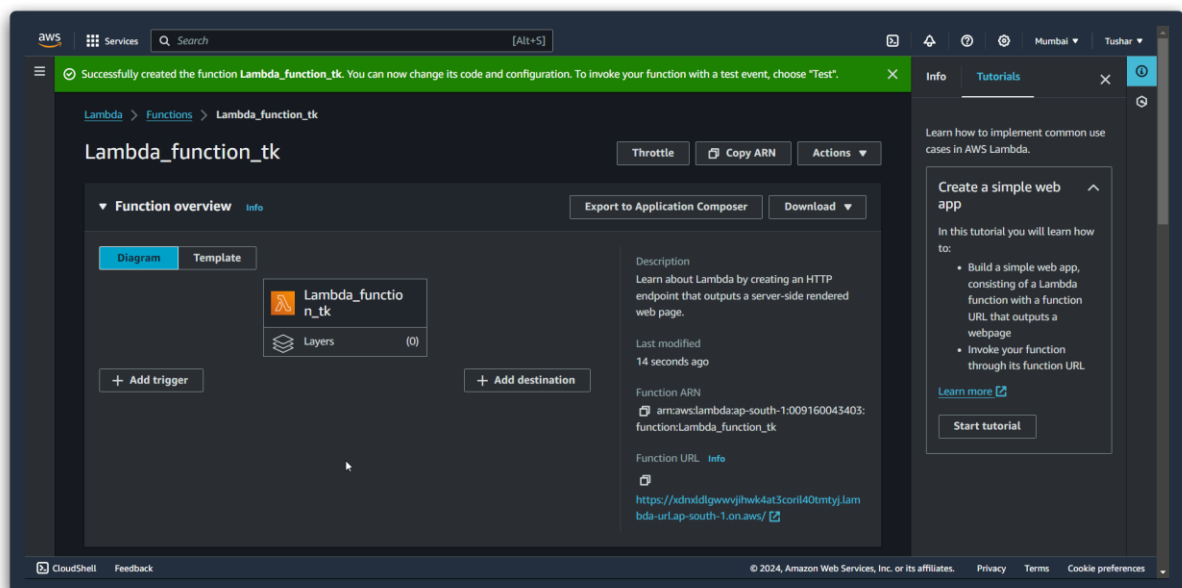
Create Function



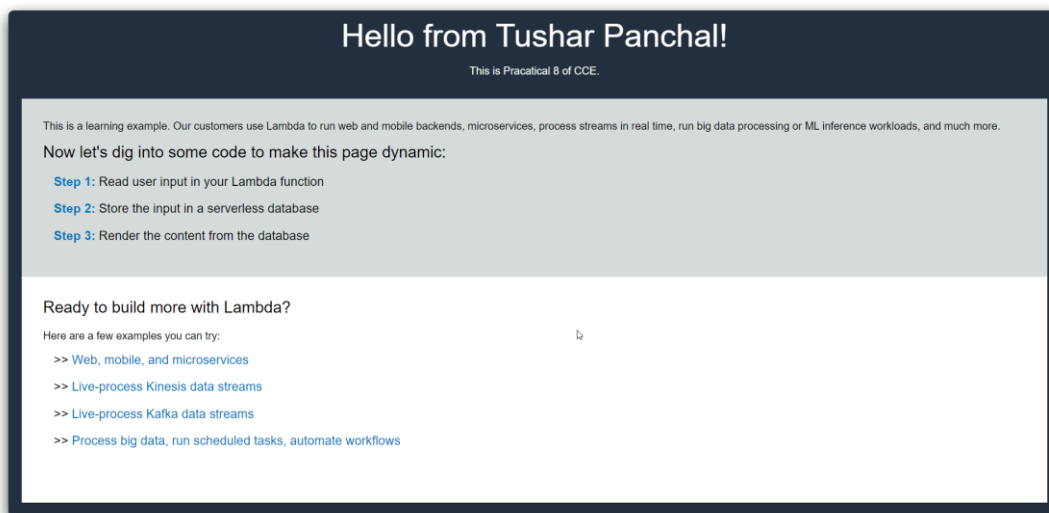
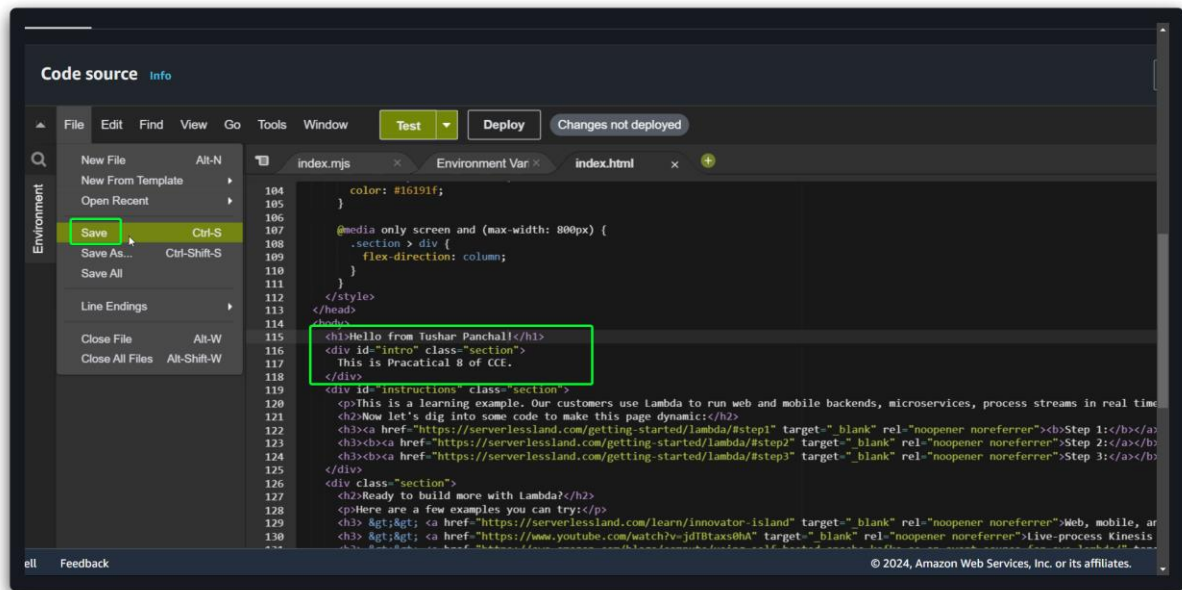
Then after check right on acknowledgment and hit create function



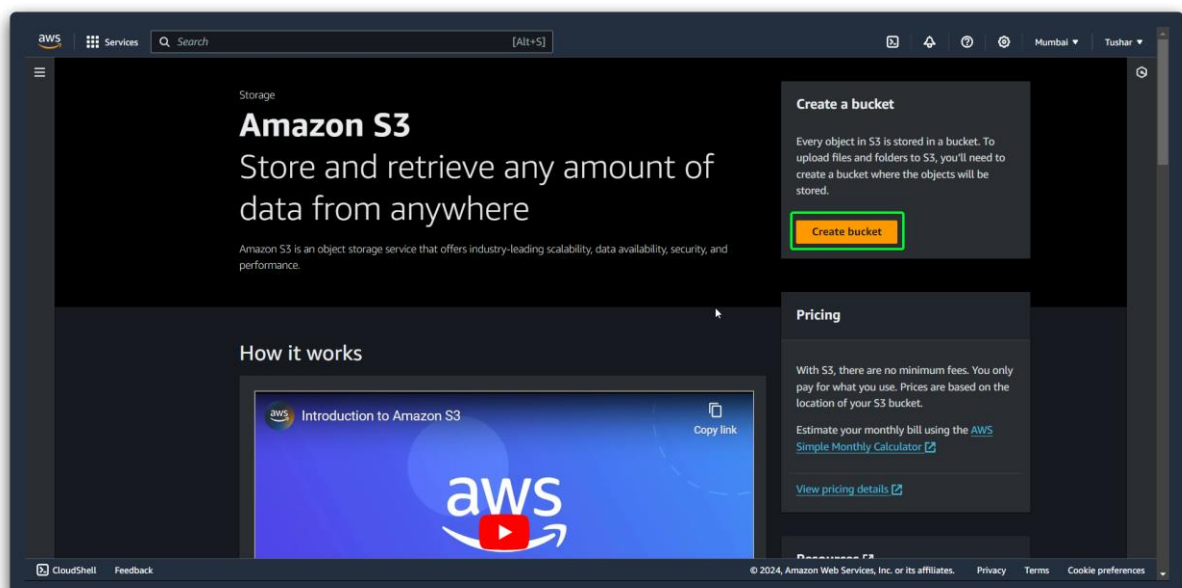
As you can see below our function is created successfully



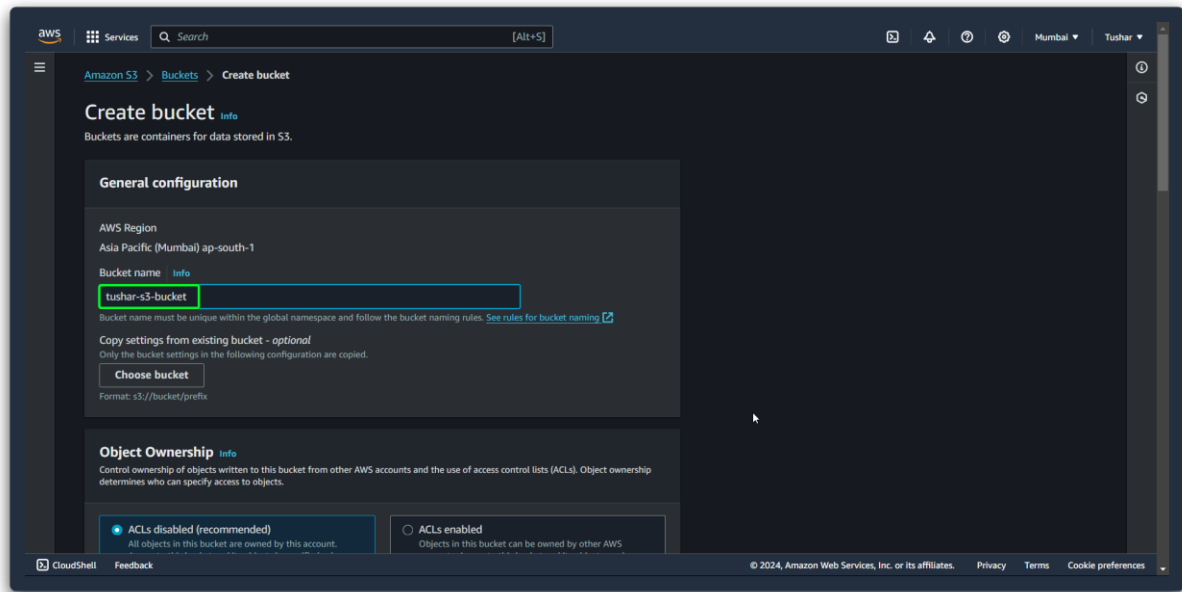
Now in code source let's make some changes in index.html then save file deploy again to make changes



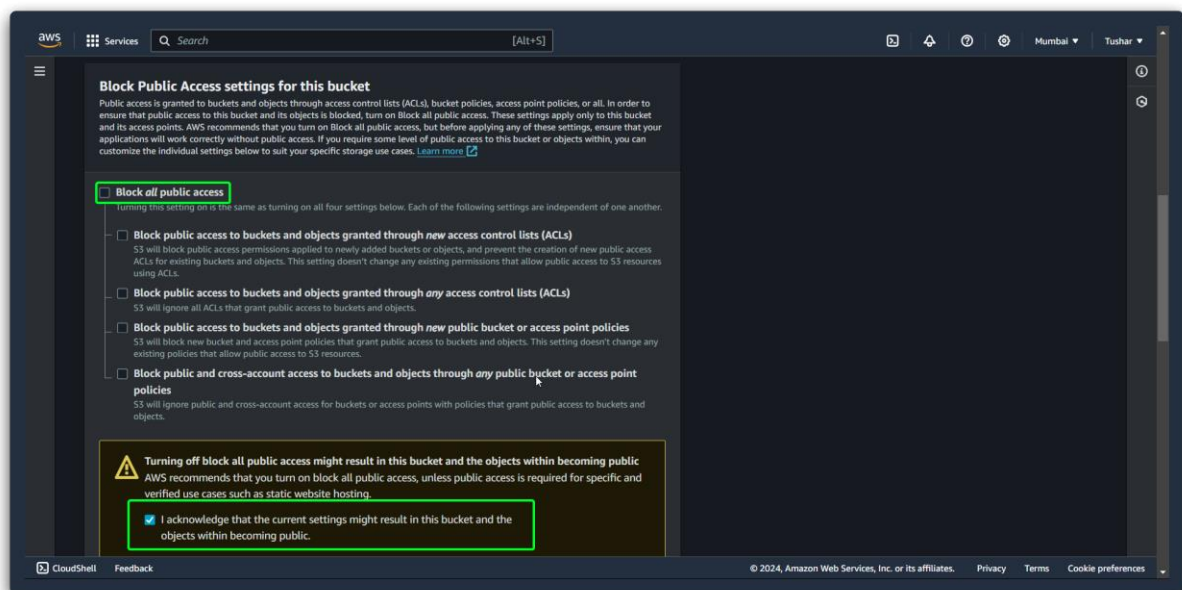
Now navigate to S3 bucket service and hit create button



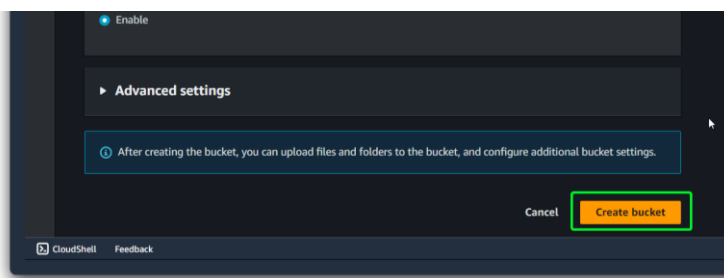
Give it a name



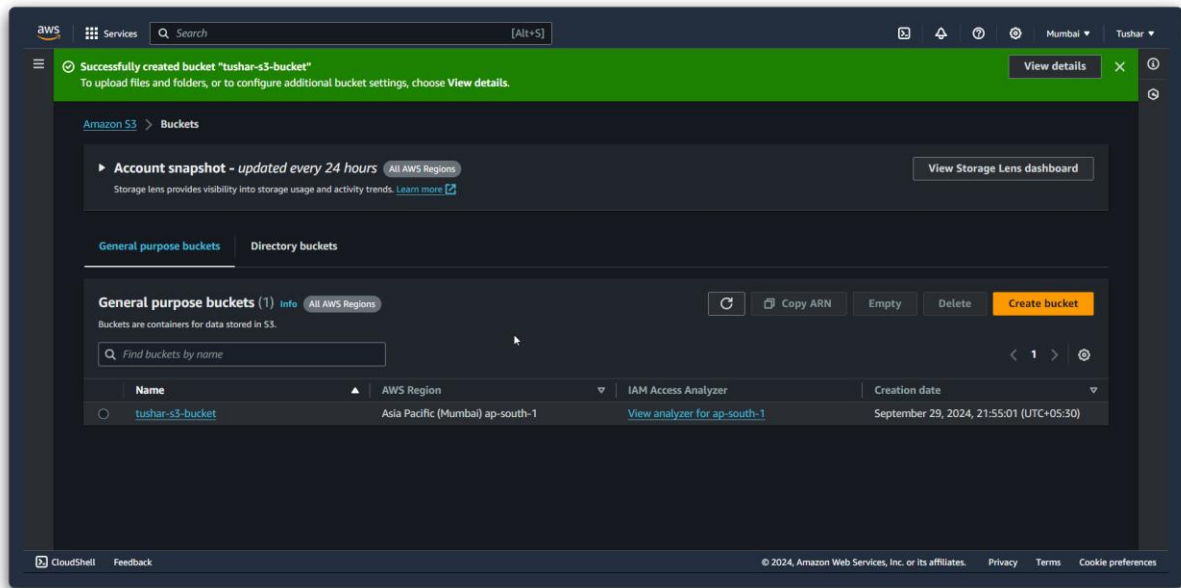
Uncheck the Block *all* public access selection



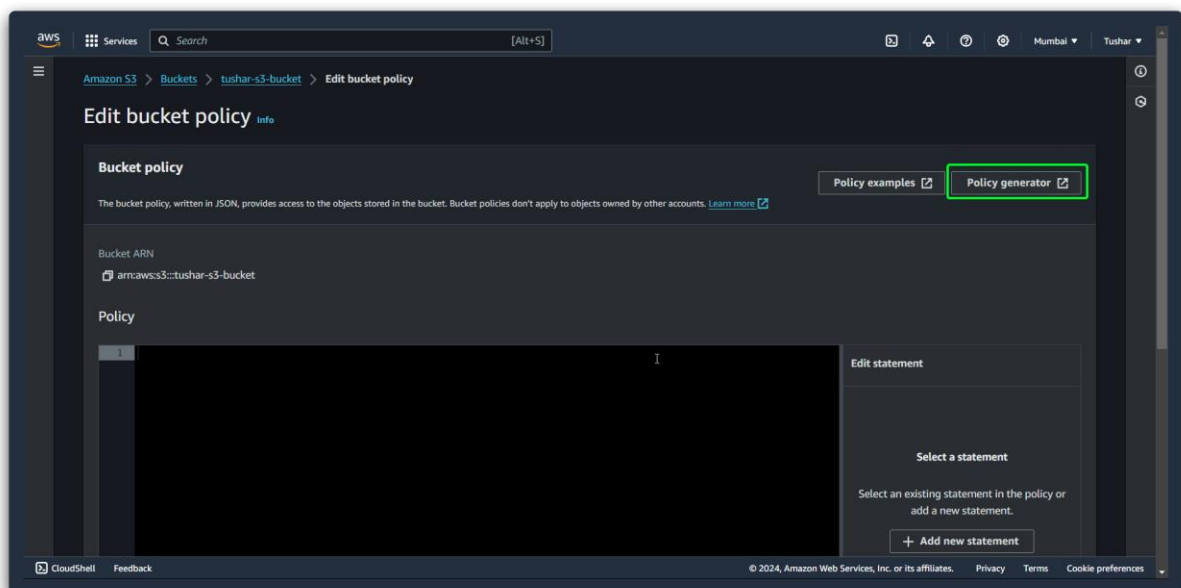
And then don't do any other changes just hit create bucket button



As you can see in below my bucket has been created successfully



Now edit the bucket policy from policy generator section:



amazon web services

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are sample policies.

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy: **S3 Bucket Policy**

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect: **Allow** ☐ Deny **this is " * "**

Principal: *****

Use a comma to separate multiple values.

AWS Service: **Amazon S3**

☐ All Services ("*")

Use multiple statements to add permissions for more than one service.

Actions: **1 Action(s) Selected** ☐ All Actions ("*")

GetObject

☐ GetObjectAcl ☐ GetObjectAttributes ☐ GetObjectLegalHold

Amazon Resource Name (ARN): **[BucketName]/[Key]**

Then go back to edit bucket policy page and copy Bucket ARN

Amazon S3 > **Buckets** > **tushar-s3-bucket** > **Edit bucket policy**

Edit bucket policy Info

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

Policy examples [Policy generator](#)

Bucket ARN copied

arn:aws:s3:::tushar-s3-bucket

Policy

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement

Then paste it into policy generator page like below and hit add statement

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

Principal
Use a comma to separate multiple values.

AWS Service
Amazon S3
☐ All Services ("*")

Use multiple statements to add permissions for more than one service.

Actions 1 Action(s) Selected ☐ All Actions ("*")

Amazon Resource Name (ARN)
ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${Keyname}.
Use a comma to separate multiple values.

Add Conditions (Optional)

Step 3: Generate Policy

A policy is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

Add one or more statements above to generate a policy.

This AWS Policy Generator is provided for informational purposes only; you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

©2010, Amazon Web Services LLC or its affiliates. All rights reserved.

An [amazon.com](#) company

Now hit generate policy button

Amazon S3
☐ All Services ("*")

Use multiple statements to add permissions for more than one service.

Actions -- Select Actions -- ☐ All Actions ("*")

Amazon Resource Name (ARN)
ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${Keyname}.
Use a comma to separate multiple values.

Add Conditions (Optional)

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
* *	Allow	s3:GetObject	arn:aws:s3:::tushar-s3-bucket	None

Step 3: Generate Policy

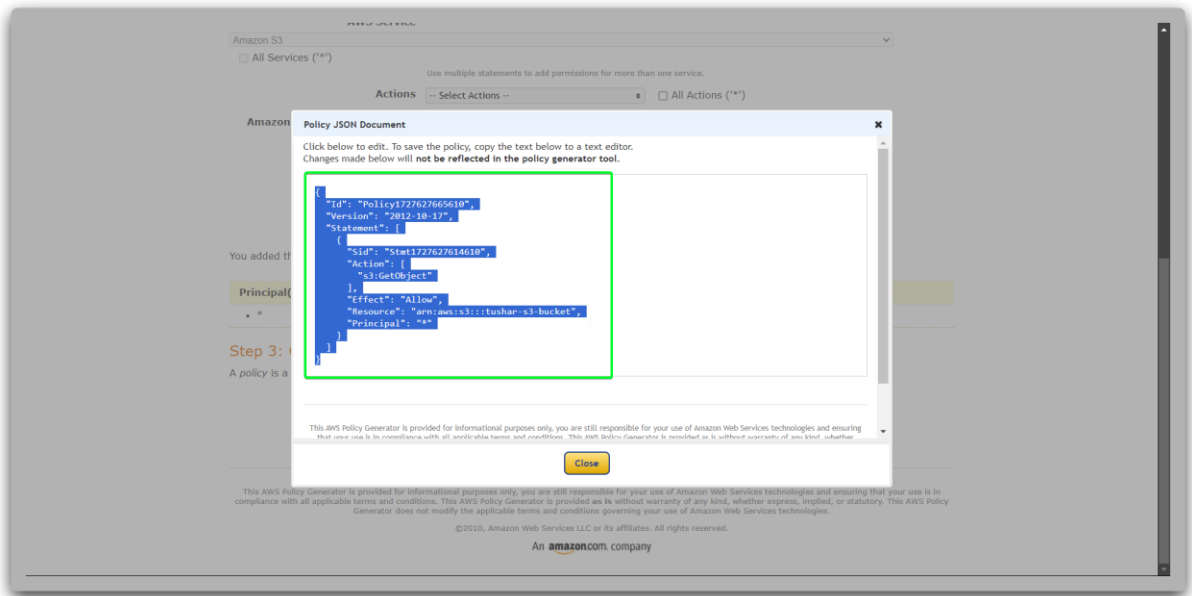
A policy is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

This AWS Policy Generator is provided for informational purposes only; you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

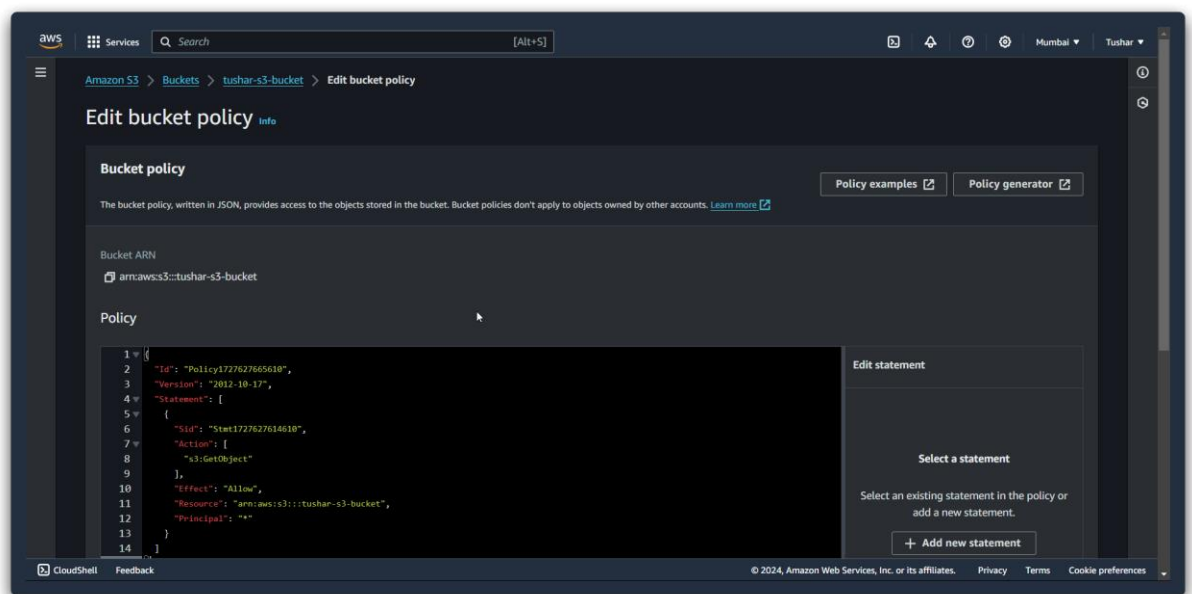
©2010, Amazon Web Services LLC or its affiliates. All rights reserved.

An [amazon.com](#) company

Now copy this generated policy json code



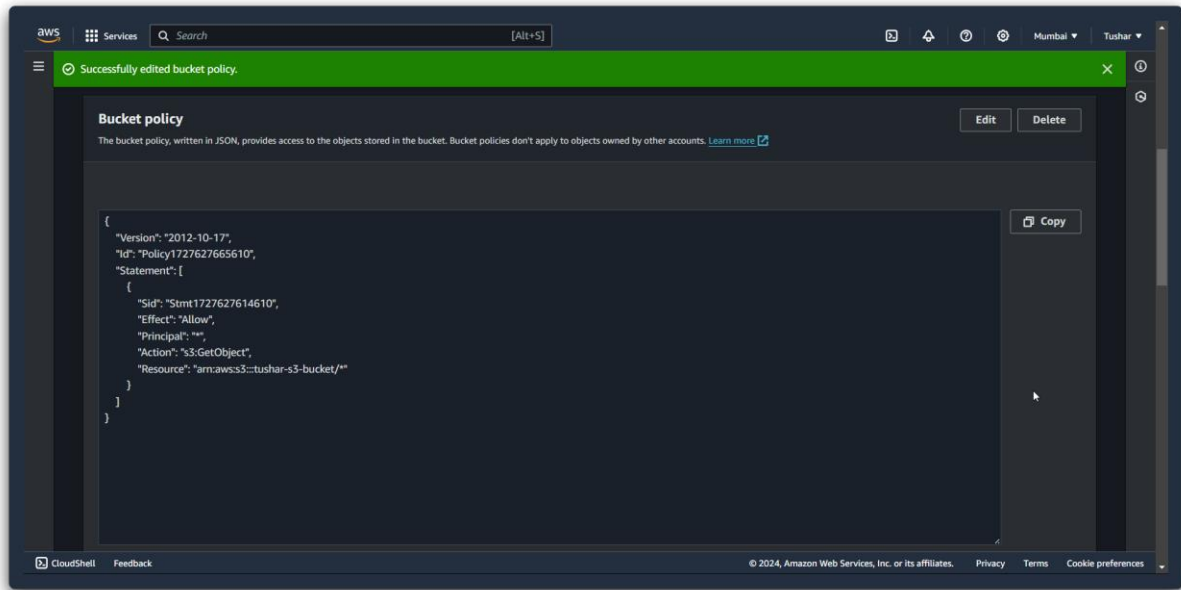
And paste it into our edit bucket policy page



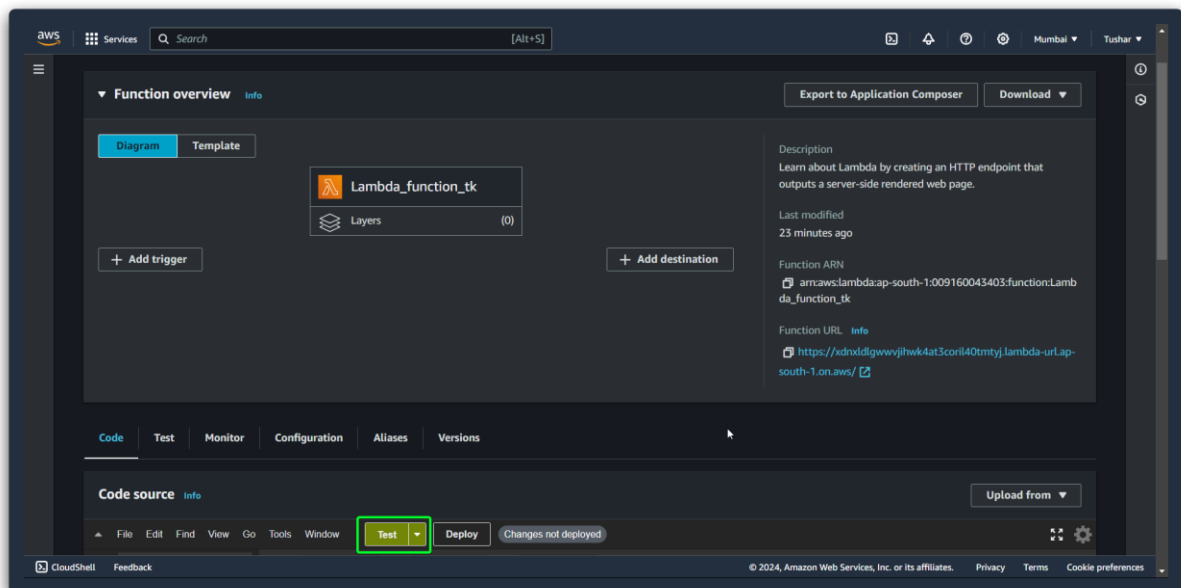
Don't forget to add **/*** at the end of resource

This change specifies that the policy applies to all objects in the bucket by adding **/*** after the bucket name.

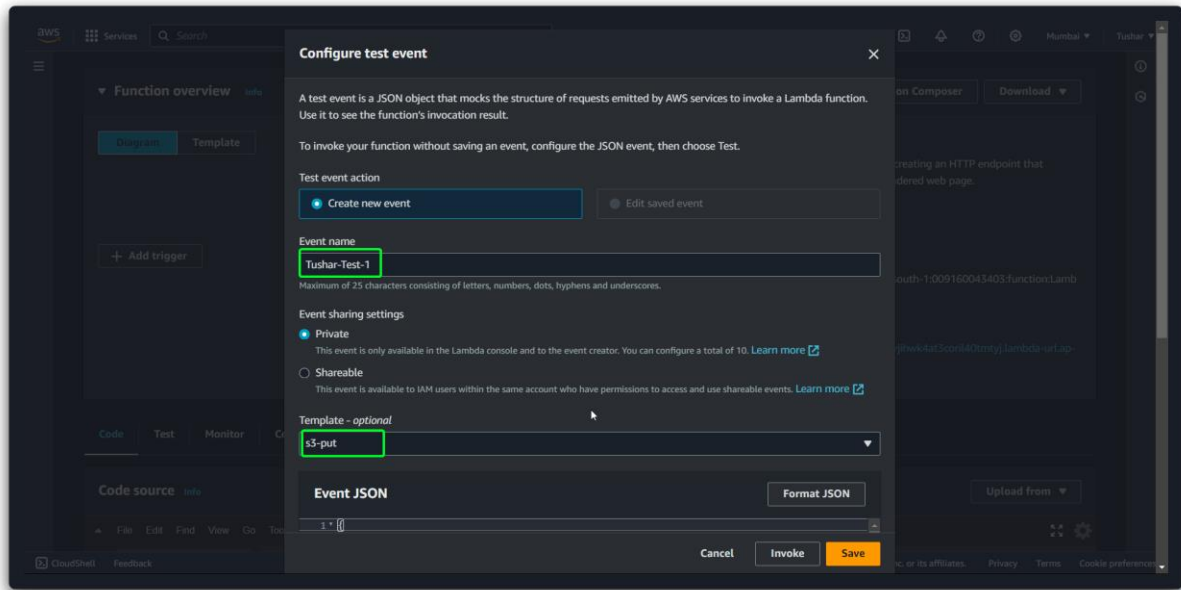
And then hit save statement button as you can see below bucket policy created successfully



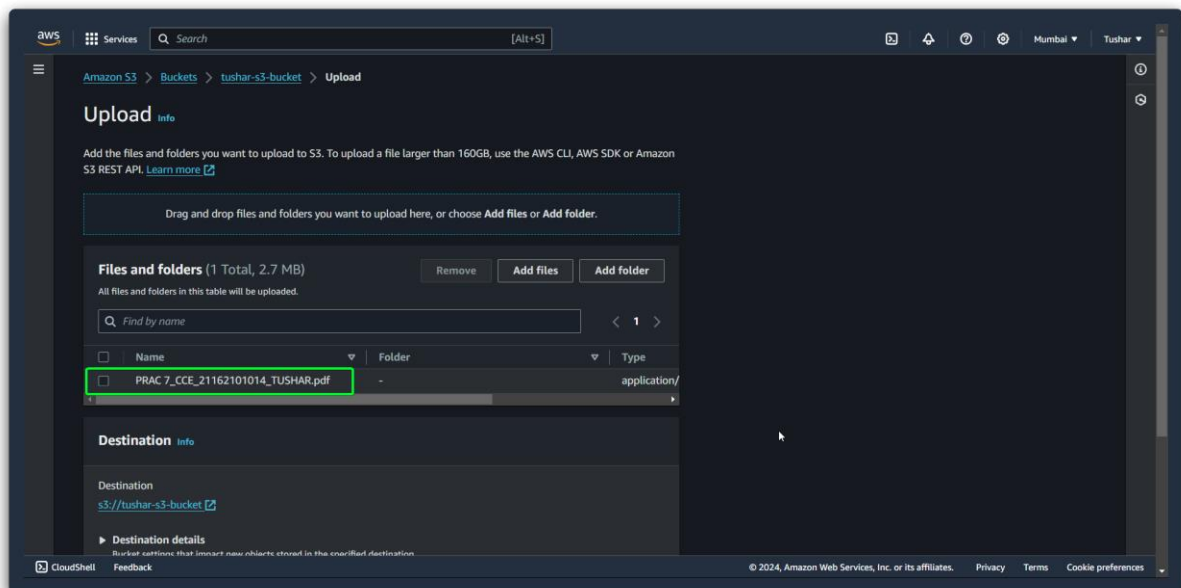
Now go back to lambda service and hit test button



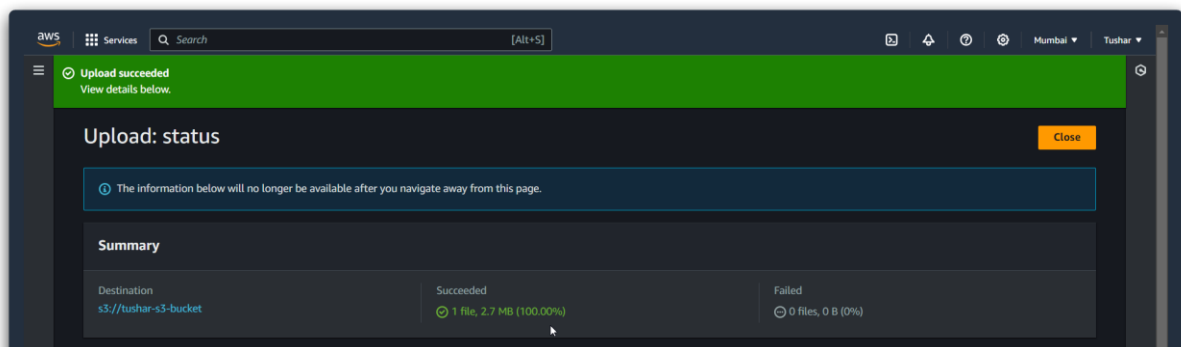
Give it a name and select template as S3-put



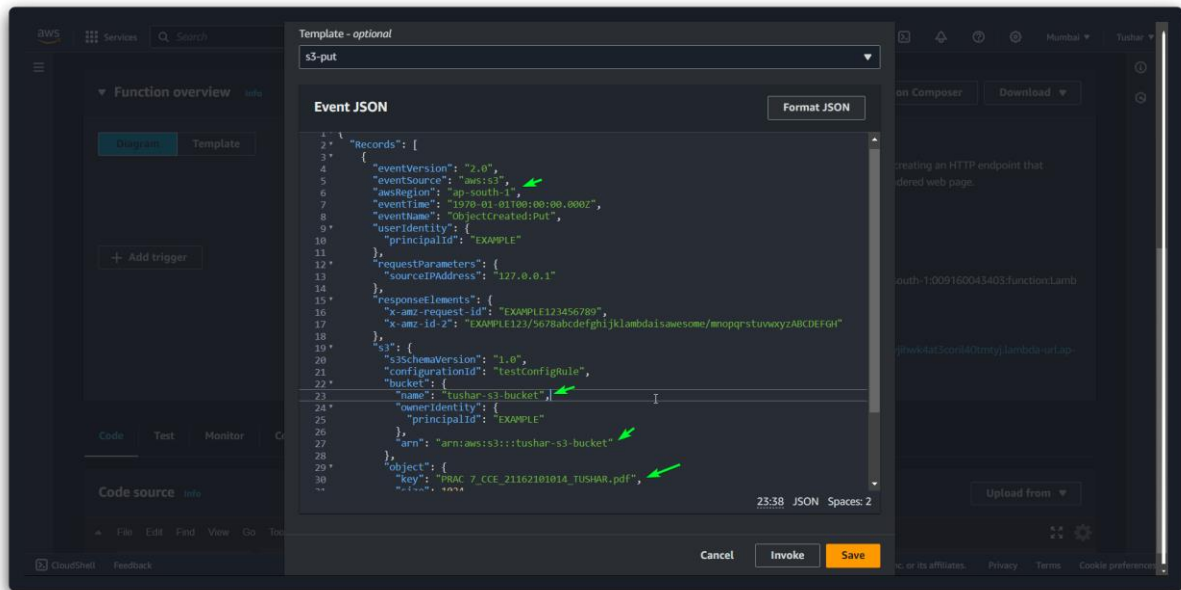
In S3-bucket upload any file of yours



As you can see below file is uploaded successfully



Change region, bucket name, ARN, object name



Tested successfully

