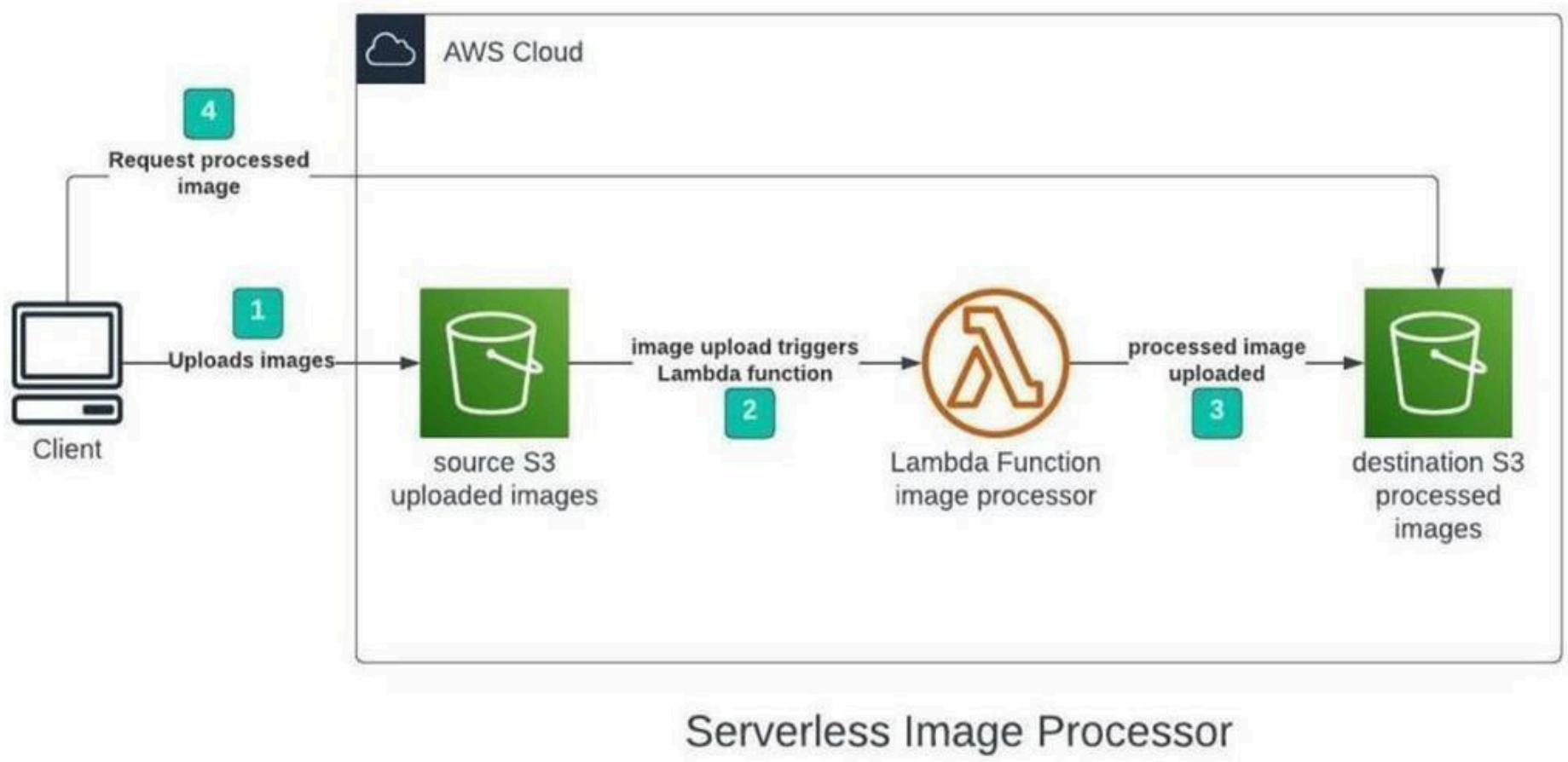
Project-1

Serverless Image Processing

Create a serverless image processing application that automatically resizes and optimizes images uploaded to an Amazon S3 bucket.

- The Serverless Image Handler solution helps you embed images on your websites and mobile applications to drive user engagement. It uses the SHARP Node.js library to provide high- speed image processing without sacrificing image quality.
- To minimize your costs of image optimization, manipulation, and processing, this solution automates version control and provides flexible storage and compute options for file reprocessing.



- **LAB STEPS:-**

Step 1: Sign in to AWS Management Console

1. Click on the Open Console button, and you will get redirected to AWS Console in a new browser tab.

2.On the AWS sign-in page,

- Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.

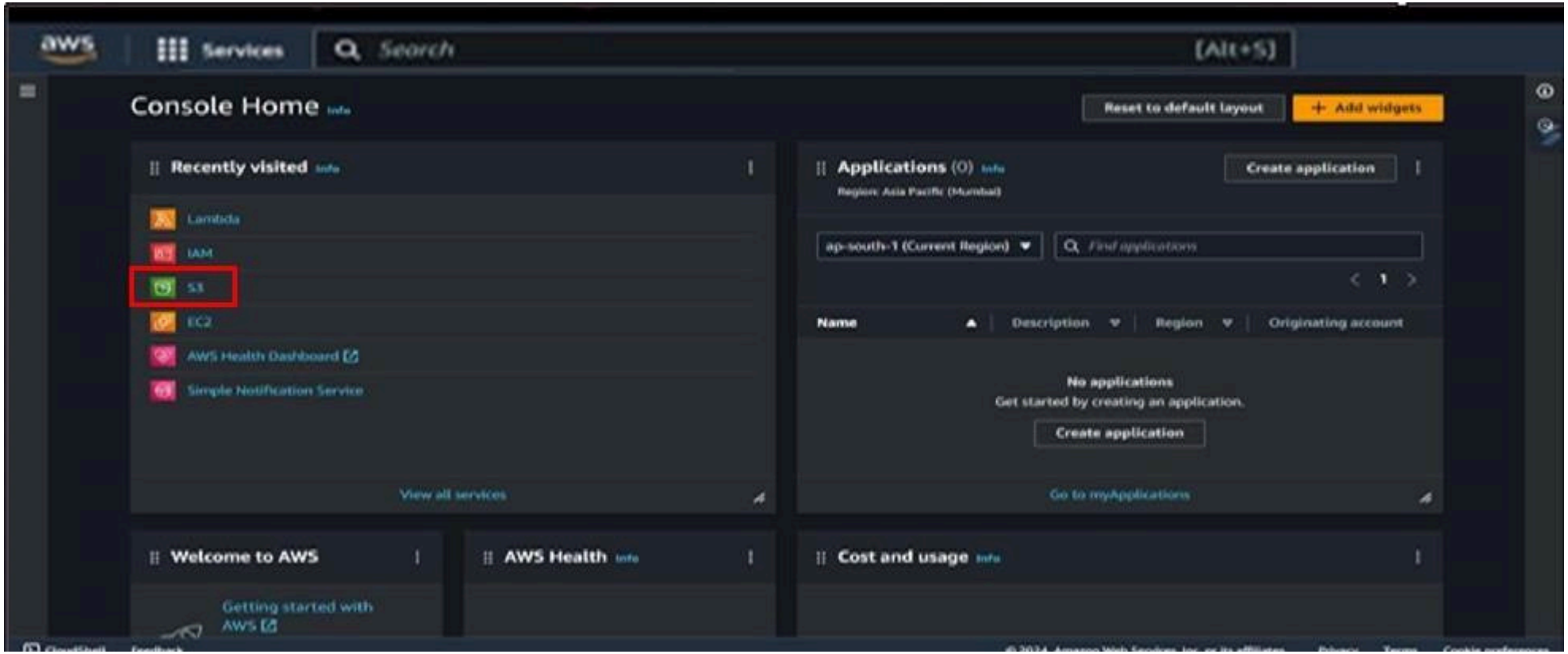
- Now copy your User Name and Password in the Lab Console to the IAM Username and Password in AWS Console and click on the Sign in button.

3. Once Signed In to the AWS Management Console, Make the default AWS Region as US East (N. Virginia) us-east-1.

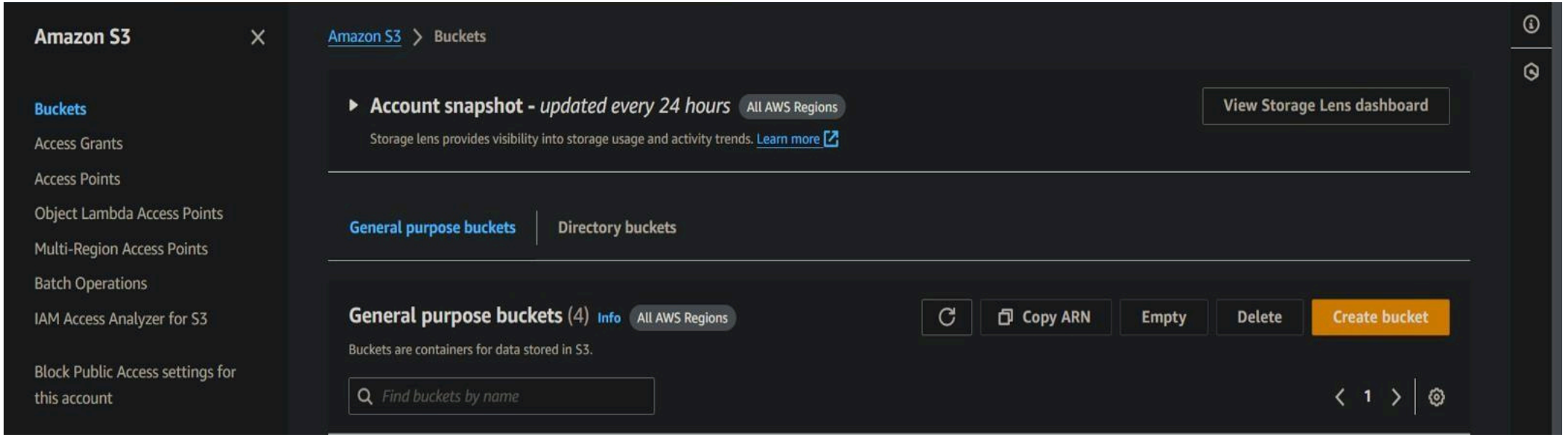
Create Two Amazon S3 Buckets

In this task, we will create two AWS S3 buckets i.e the source bucket and the destination bucket by providing the required configurations like name, region etc.

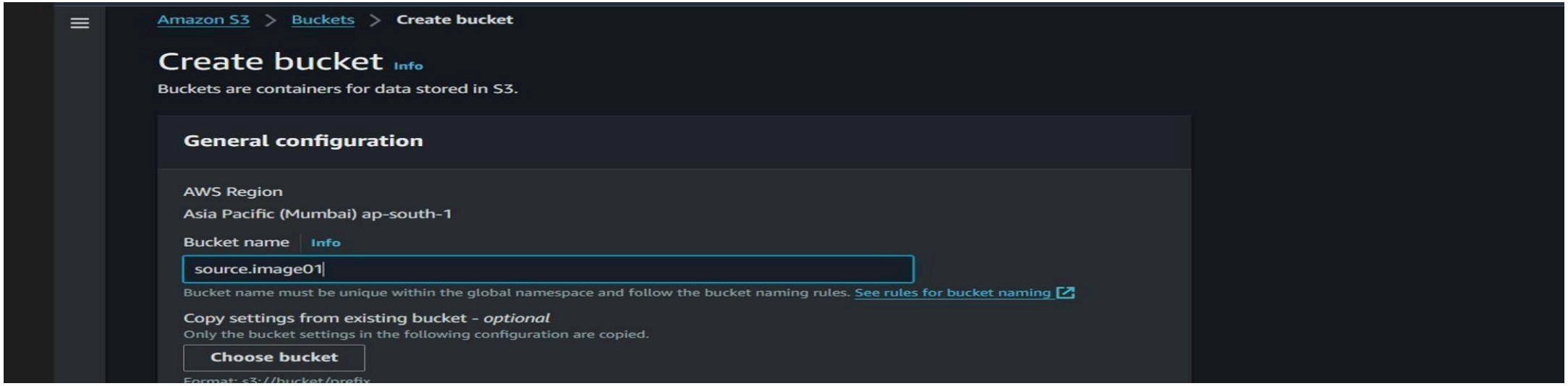
1.Navigate to the **Services** menu in the Top, then click on **S3** in the storage section.



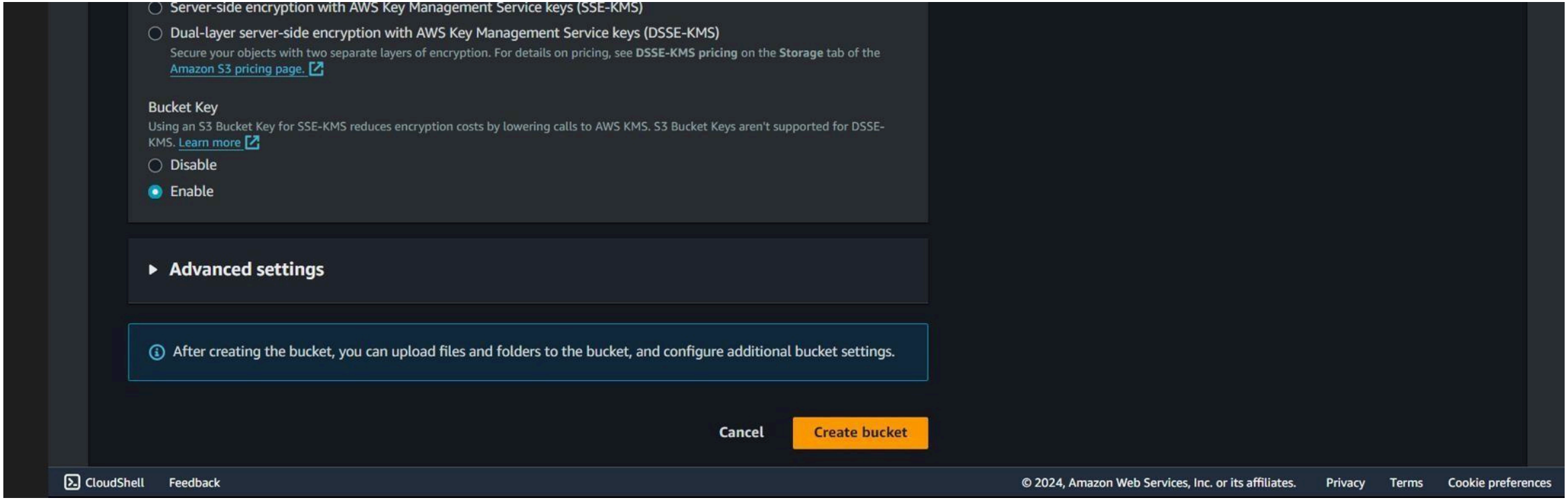
2. Click on Create Bucket button.



3.Create Source Bucket



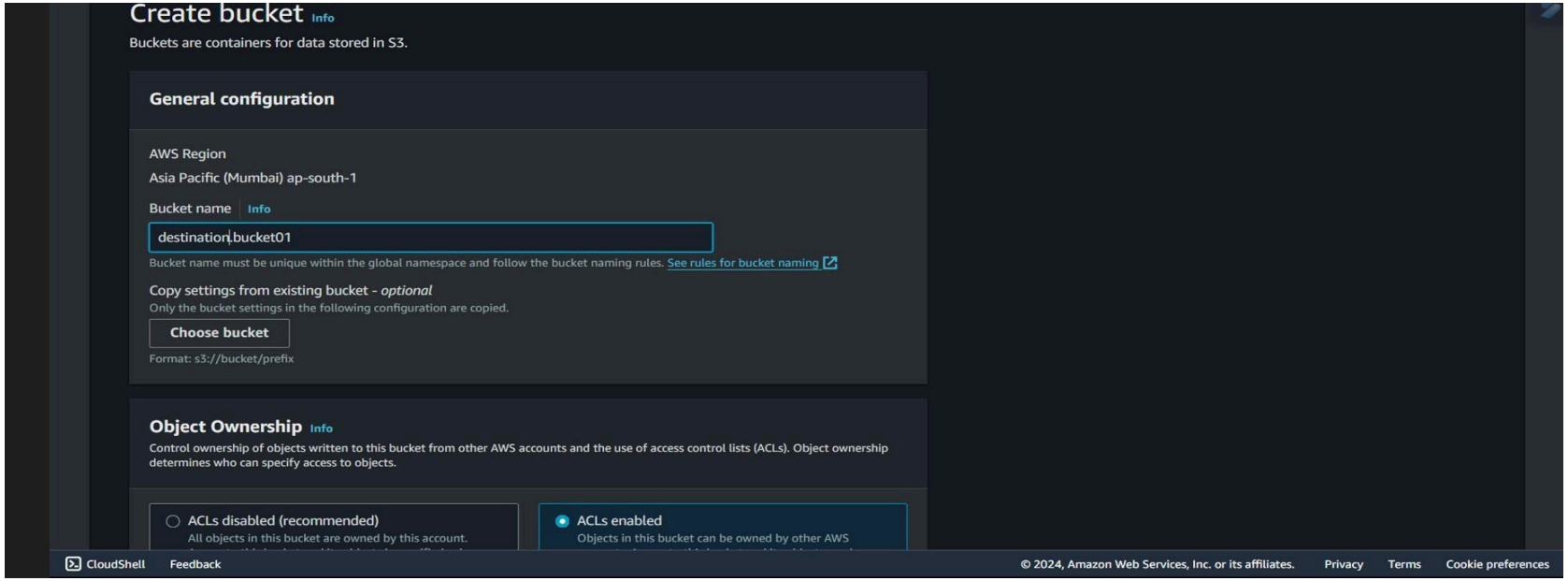
4.Leave Other settings as Default and click on the **Create Bucket** button



5.Once the Bucket is created successfully, Select your S3 bucket.

- Click on the Copy ARN button to copy the ARN.
- Save the source bucket ARN in a text file for later use.
- arn:aws:s3:::source.bucket01

6. Create Destination Bucket



7.Leave Other settings as Default and click on the **Create Bucket** button

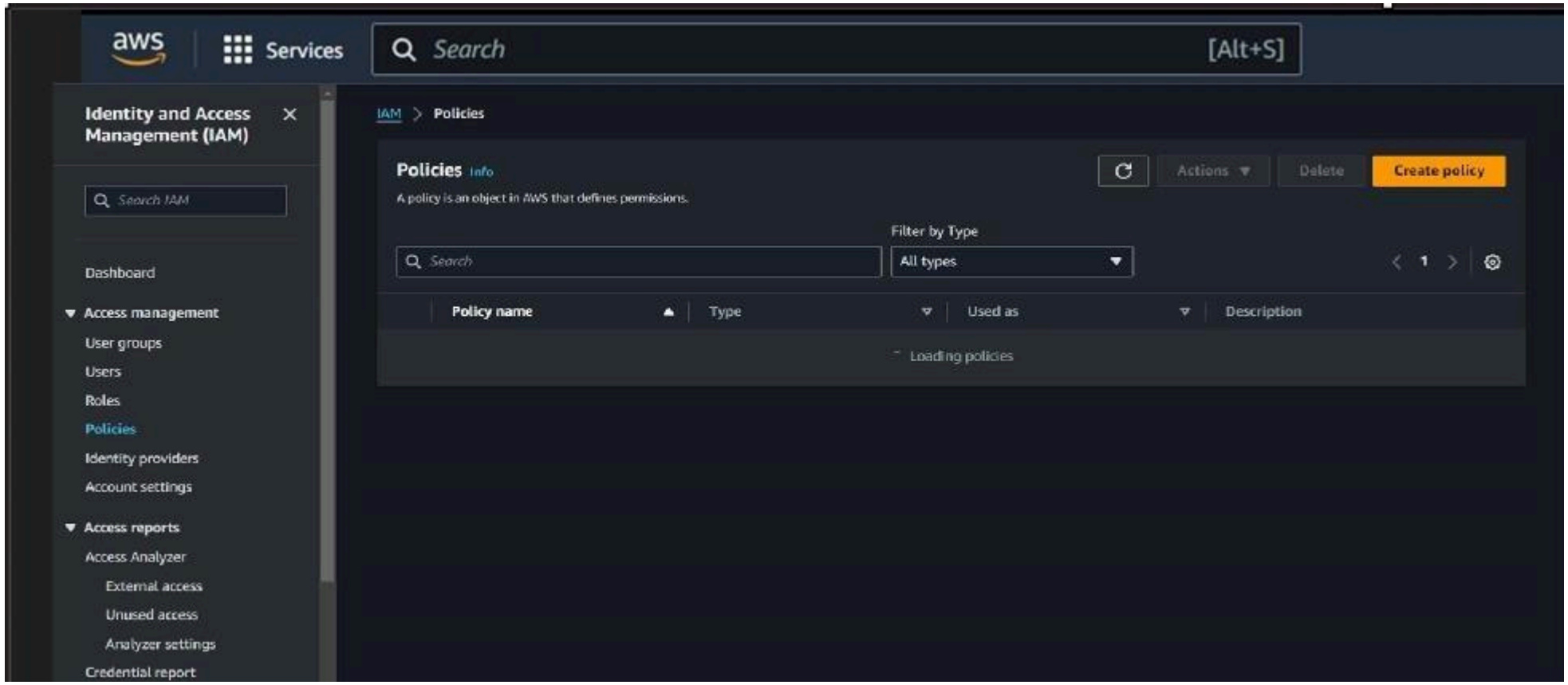
8.Once the Bucket is created successfully, Select your S3 bucket.

- Click on the Copy ARN button to copy the ARN.
- Save the source bucket ARN in a text file for later use.
- arn:aws:s3:::destinationbucket01

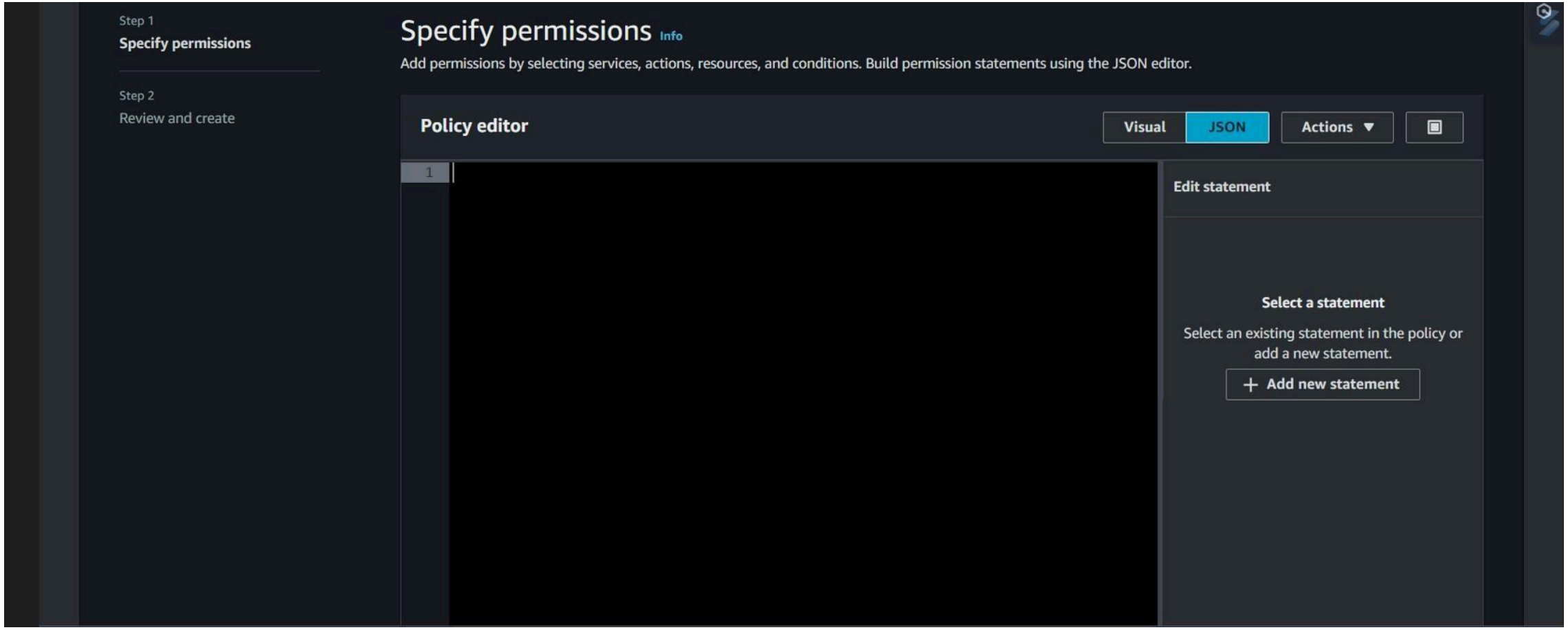
Step 3: Create an IAM Policy

1. Go to **Services** and Select **IAM** under **Security, Identity and Compliance.**

2. Click on **Policies** in the left navigation bar and click on the **Create policy** button.



3. Click on the **JSON** tab, Remove the existing code and copy-paste the below policy statement into the editor:



- **Policy JSON:**

```
{
"Version": "2012-10-17",
 "Statement": [
  {
```

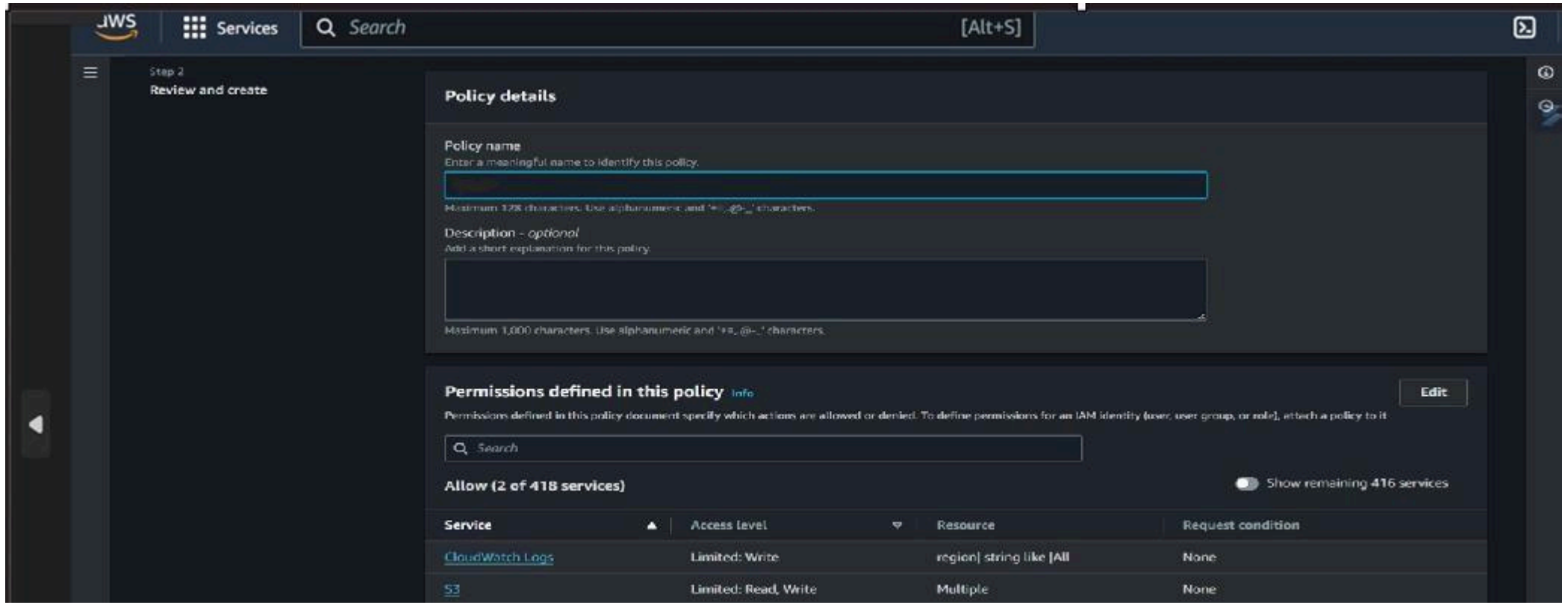
```
  "Effect": "Allow", "Action": [ "logs:PutLogEvents",
    "logs:CreateLogGroup",
    "logs:CreateLogStream"
  ],
  "Resource": "arn:aws:logs:*:*:*"
 },
 {
  "Effect": "Allow",
  "Action": ["s3:GetObject"],
  "Resource": "arn:aws:s3:::source.bucket01"
 },
{
 "Effect": "Allow", "Action": ["s3:PutObject"],
 "Resource": "arn:aws:s3:::destinationbucket01"
}
```

```
    ]
  }
```

4. Leave Everything as default and click on **Next** button.

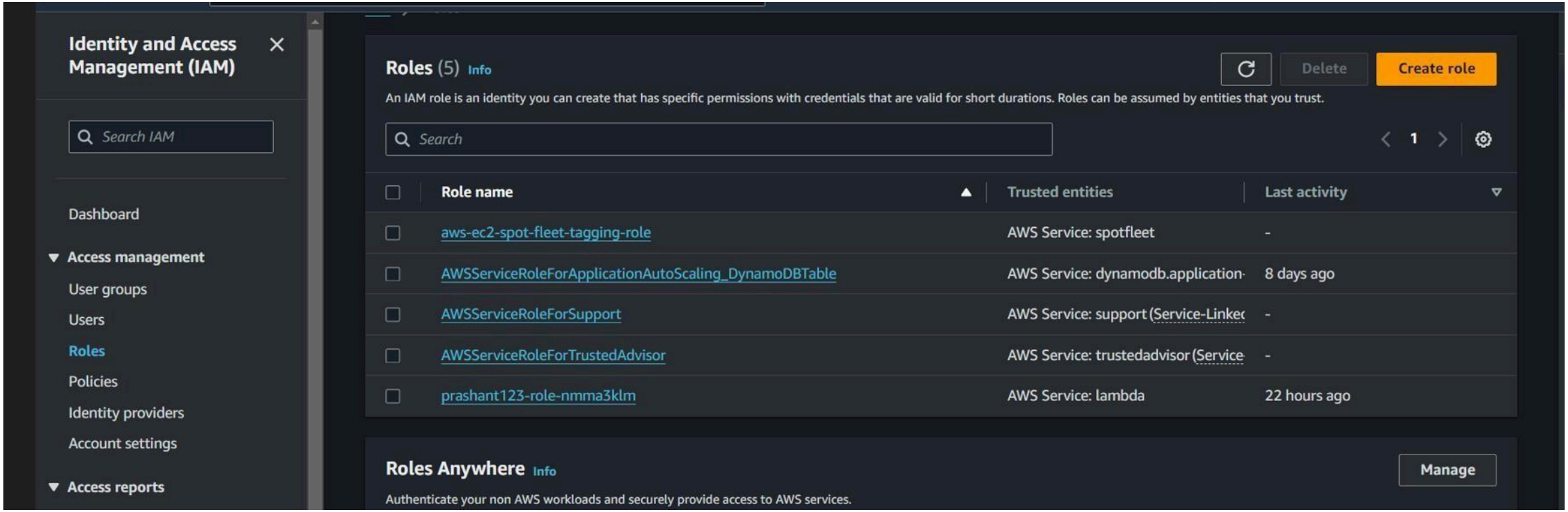
5.On the Review Policy page:

6.Enter **Policy Name and** Click on the **Create policy** button



Step 4: Create an IAM Role

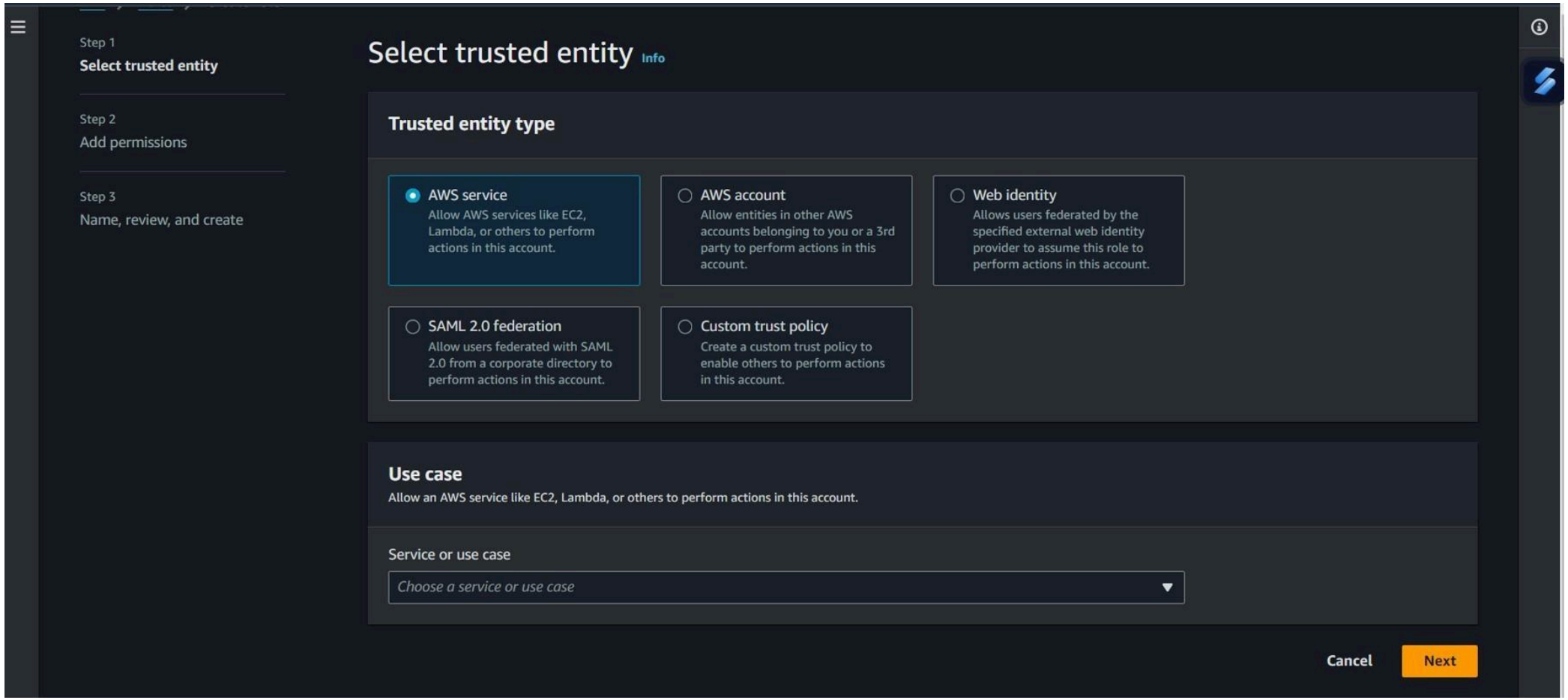
1.In the left menu, click on **Roles** and click on the **Create Role** button.

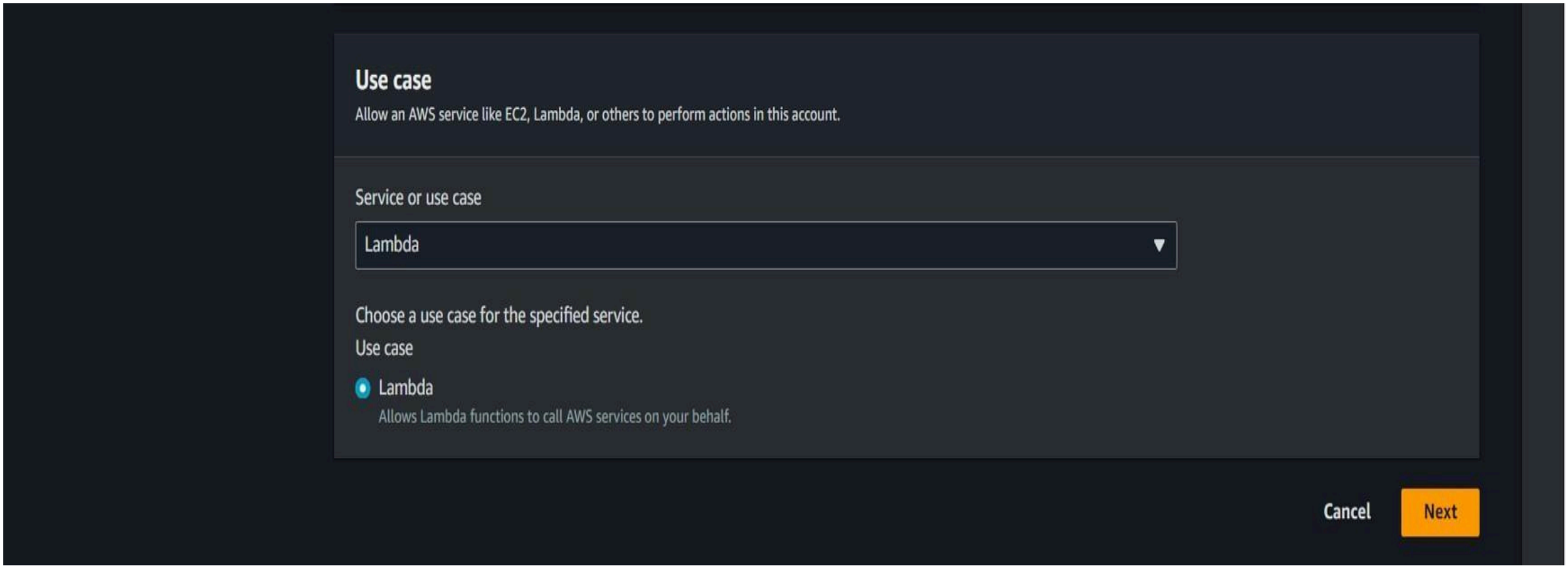


2.Select Lambda from AWS Services list.

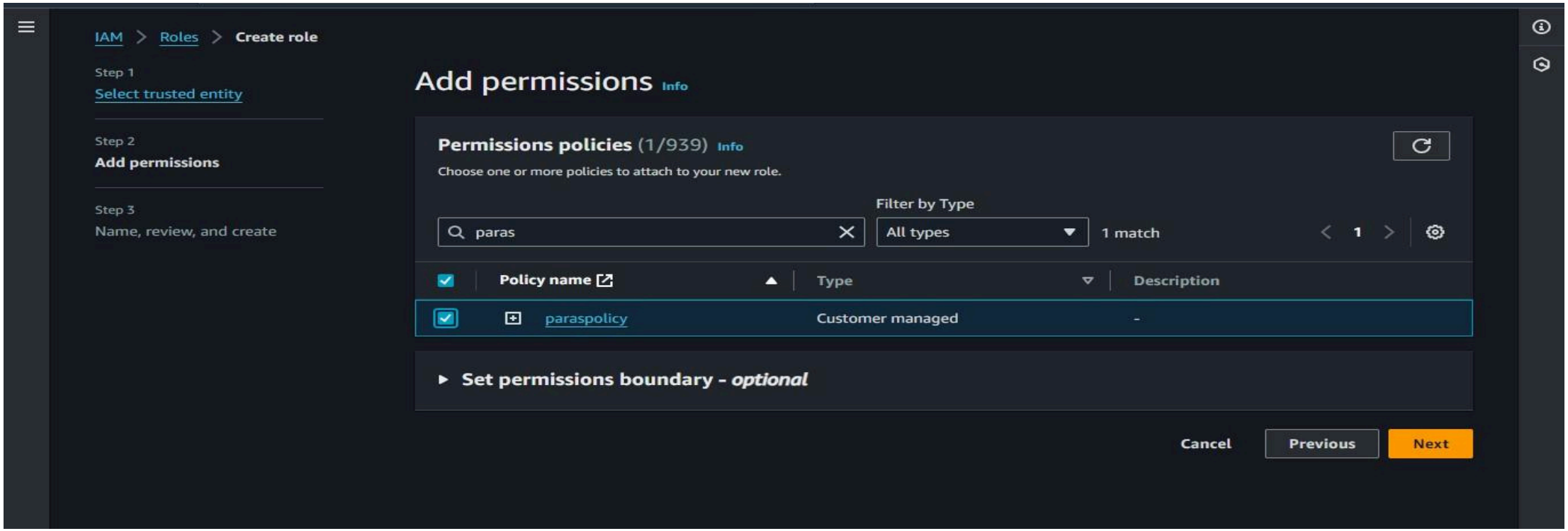
- From Trusted Entity Type: Select AWS Service
- From Use case: Select Lambda

- Click on Next button.





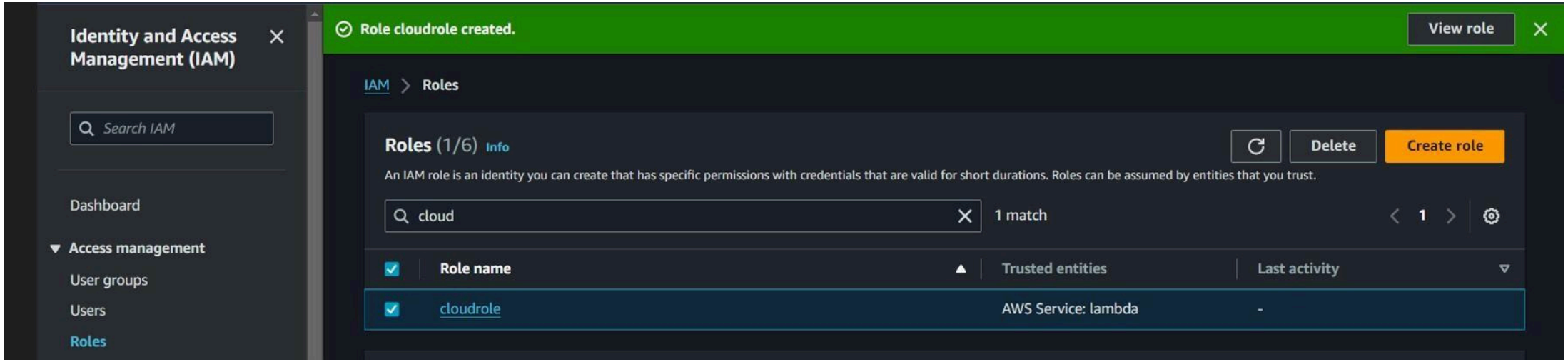
3. Select your **policy** and click on the **Next** button.



4.Role Name: Enter **cloudrole**

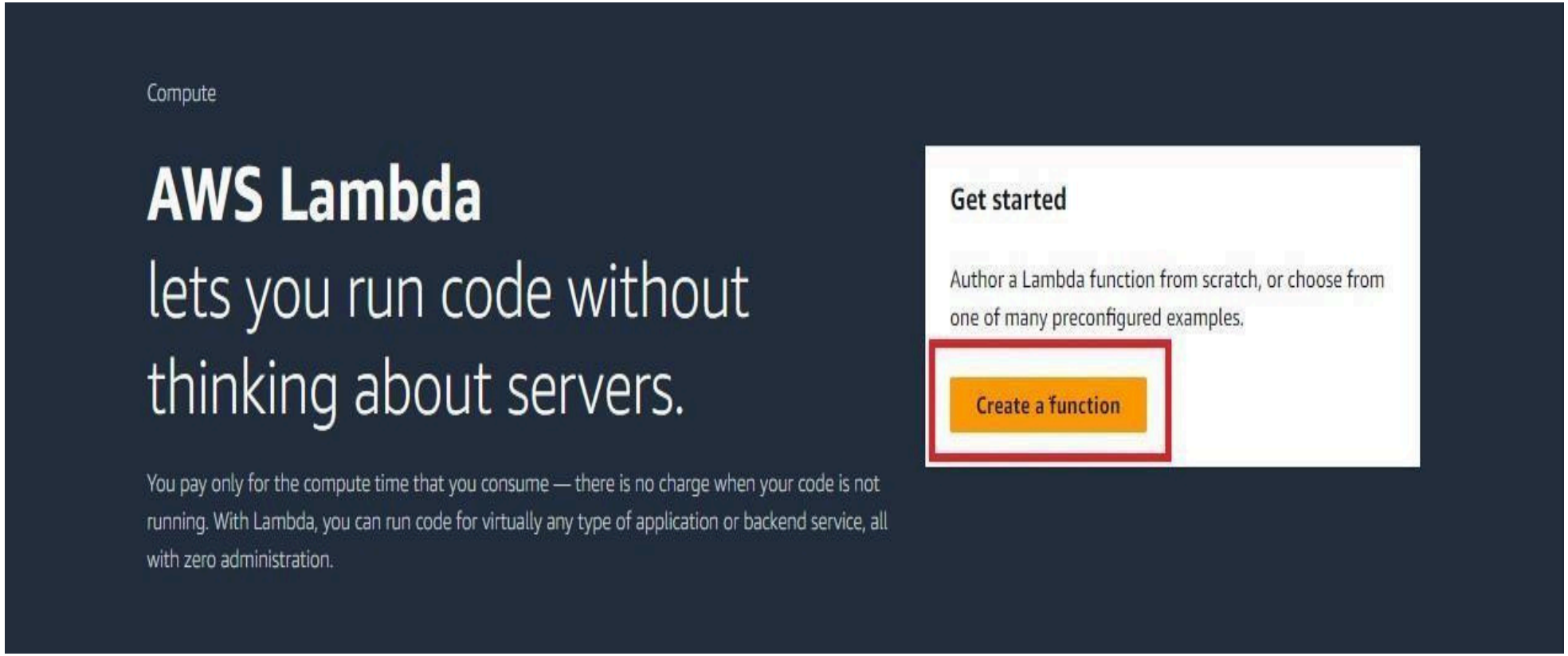
5.Click on the **Create Role** button.

- You have successfully created an IAM role by name cloudrole.

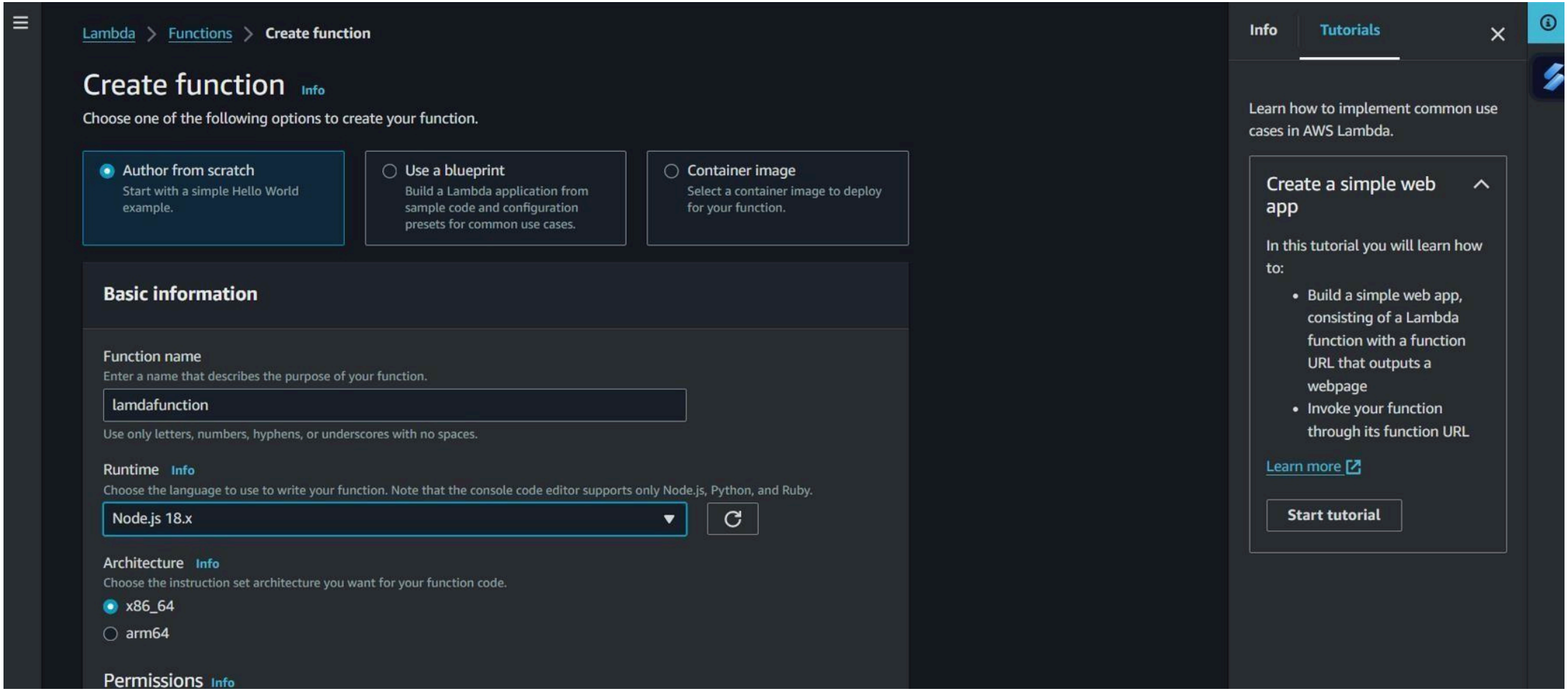


Task 5: Creating Lambda function

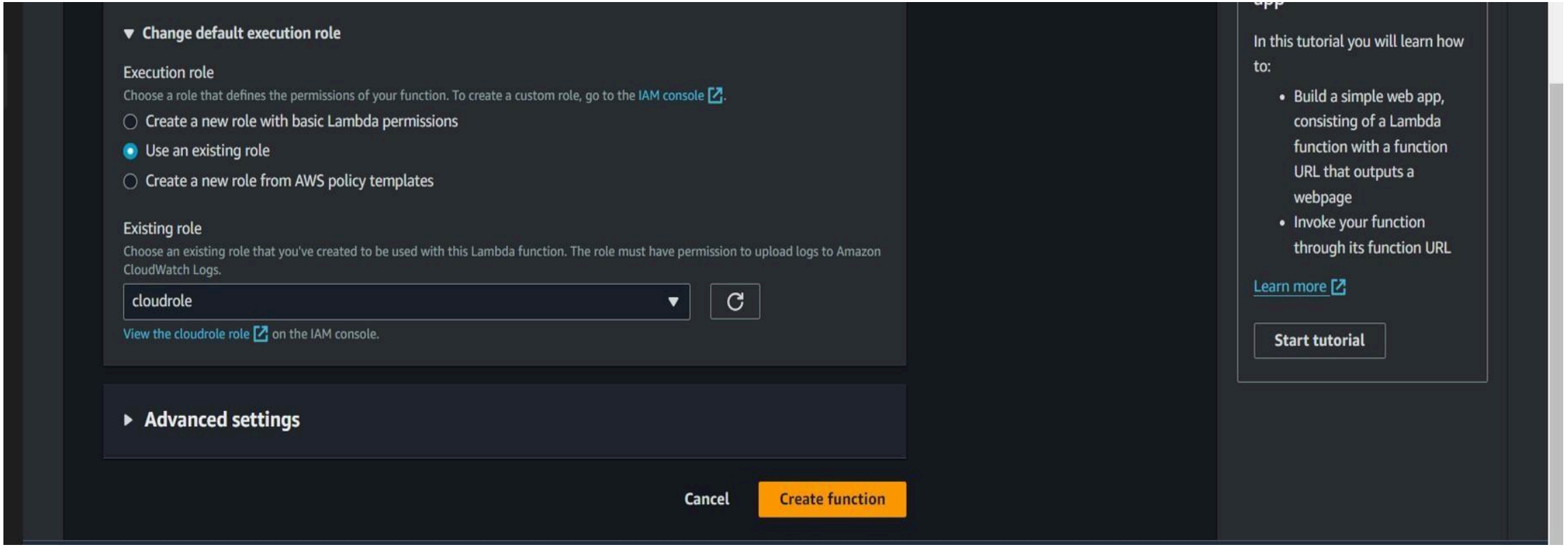
1. Go to AWS Lambda Console, Navigate to functions section . Click **Create function**



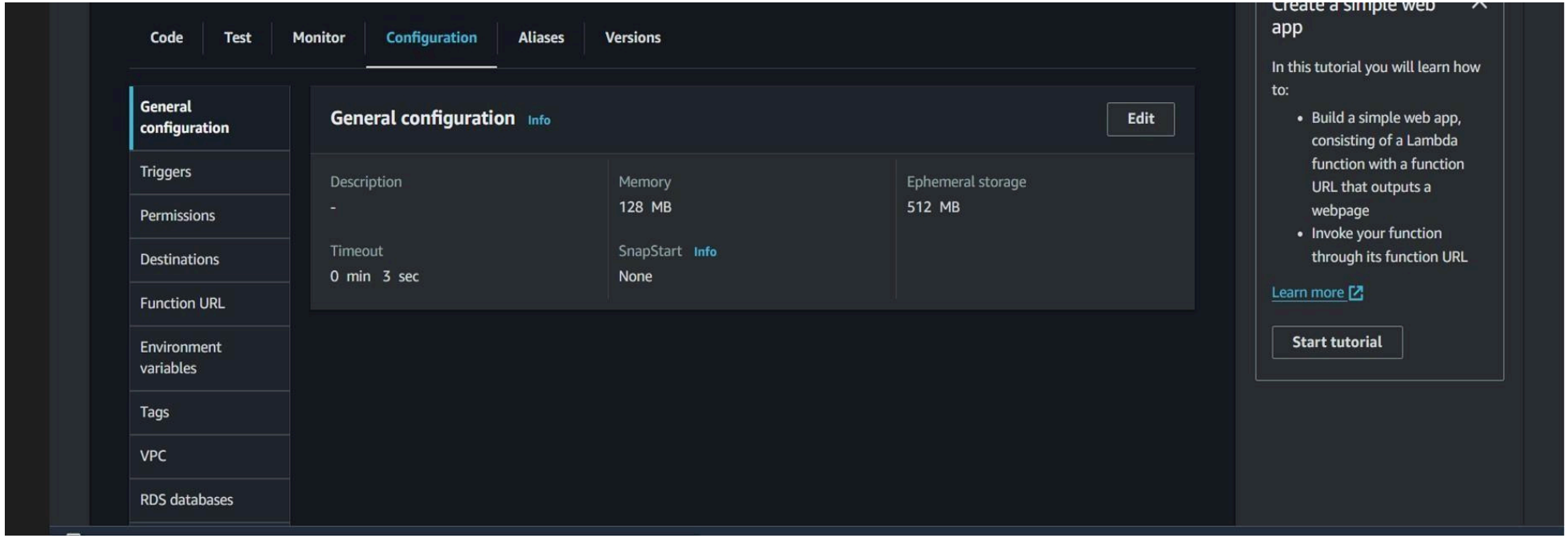
2. Name it and select runtime and Leave all other settings as default.

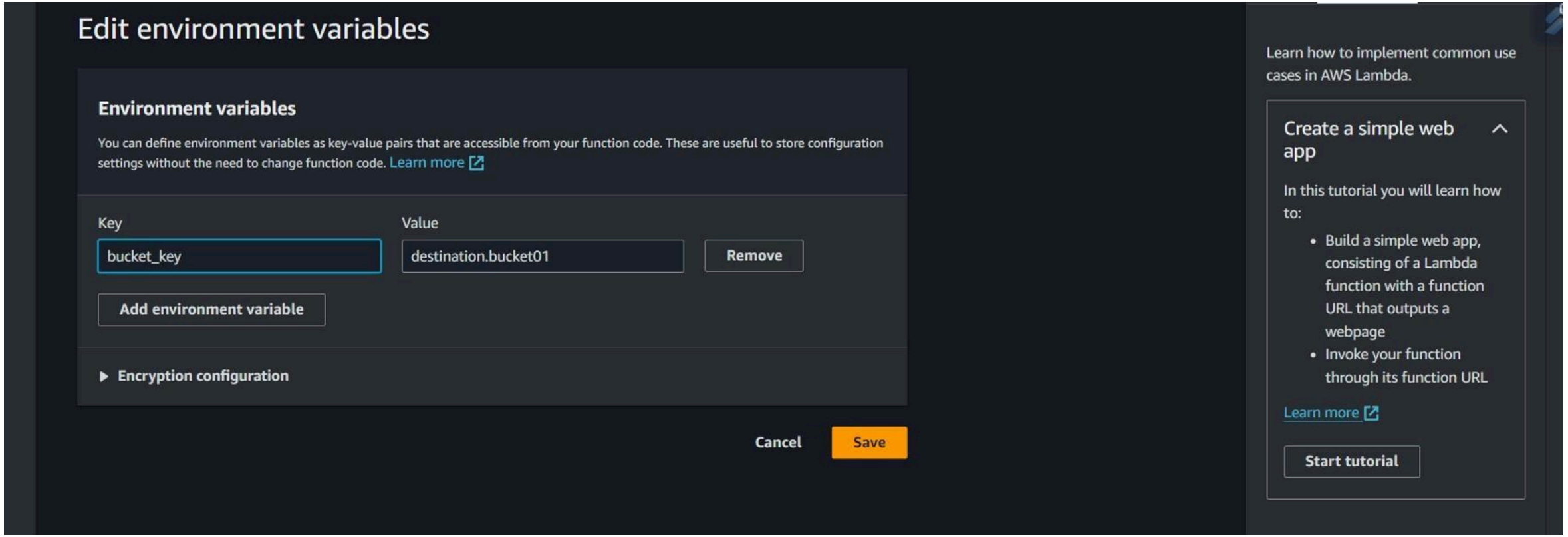


3. Change Default execution role and create function



4. Edit Environment Variables


Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC
RDS databases
General configuration Info
Edit
Description
-
Memory
128 MB
Ephemeral storage
512 MB
Timeout
0 min 3 sec
SnapStart Info
None
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial



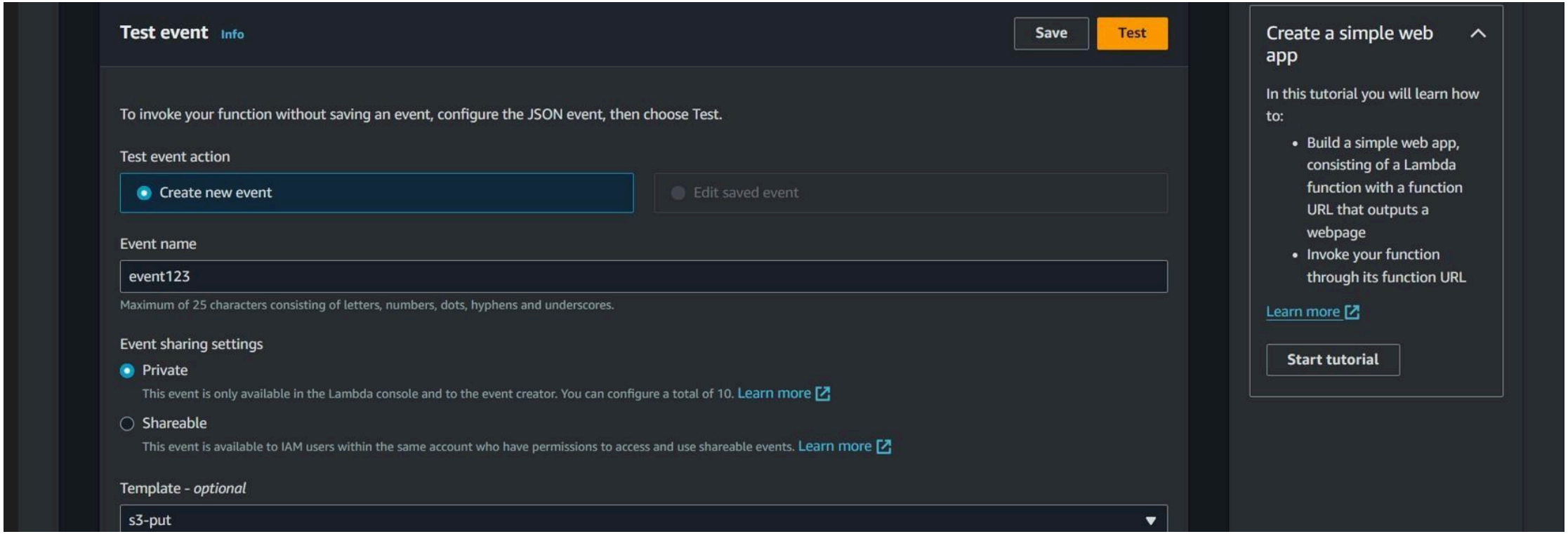
Step 6: Test Lambda Function

*Go to AWS Lambda console. Navigate to Functions section.

*open function then will be created

*open test console

*template=s3-put



EVENT JSON:

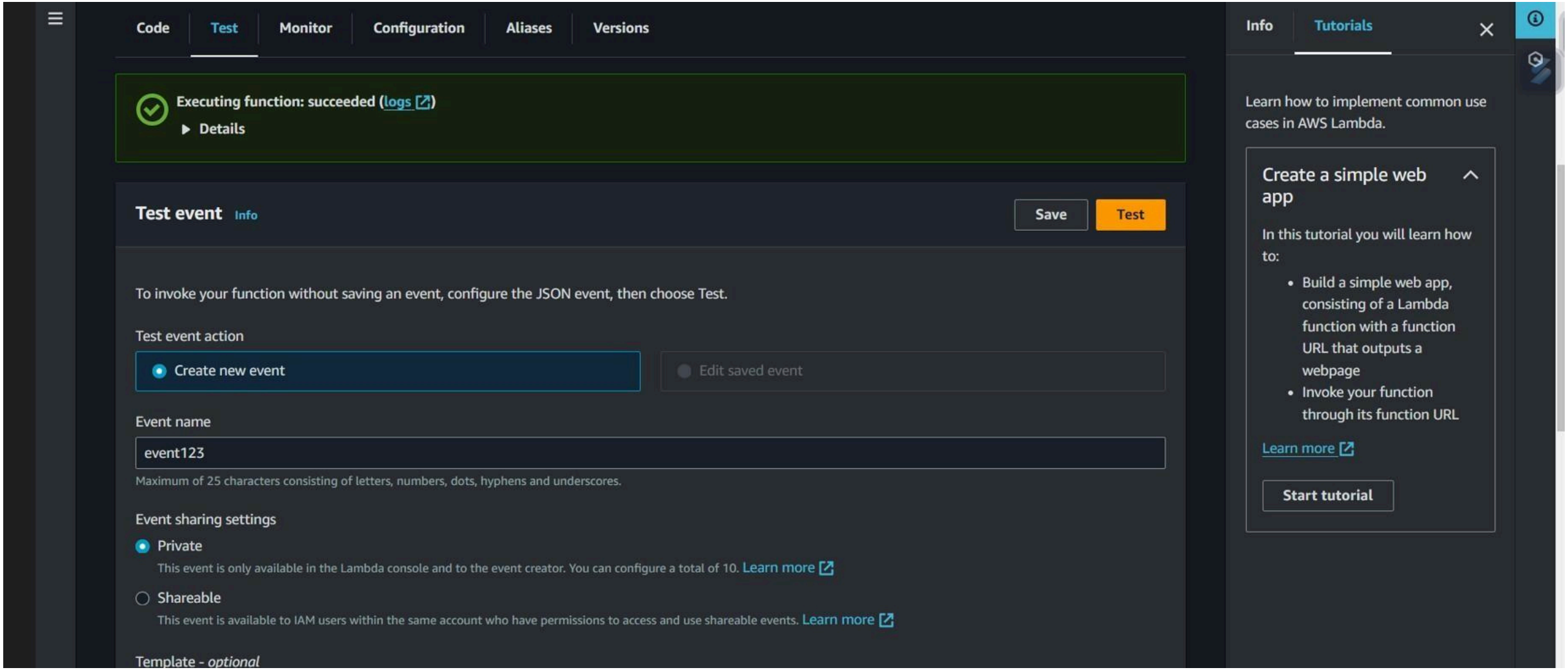

```
{
  "Records": [
    {
      "eventVersion": "2.0",
```

```
    "eventSource": "aws:s3",
    "awsRegion": "us-east-1",
    "eventTime": "1970-01-01T00:00:00.000Z",
    "eventName": "ObjectCreated:Put", "userIdentity": {
     "principalId": "EXAMPLE"
    },
    "requestParameters": { "sourceIPAddress": "127.0.0.1" "sourceIPAddress": "127.0.0.1"
},
"responseElements": {
"x-amz-request-id": "EXAMPLE123456789", "x-amz-id-2":
"EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzAB CDEFGH"
}, "s3": {"s3SchemaVersion": "1.0",
"configurationId": "testConfigRule", "bucket": {
"name": " source.bucket01", "ownerIdentity": {
```

"principalId": "EXAMPLE"
},
"arn": "arn:aws:s3:::source.bucket01"
},
"object": {
"key": "18981044.jpg",
"size": 1024,
"eTag": "0123456789abcdef0123456789abcdef", "sequencer": "0A1B2C3D4E5F678901" } } }
]

}

Now We can Test:



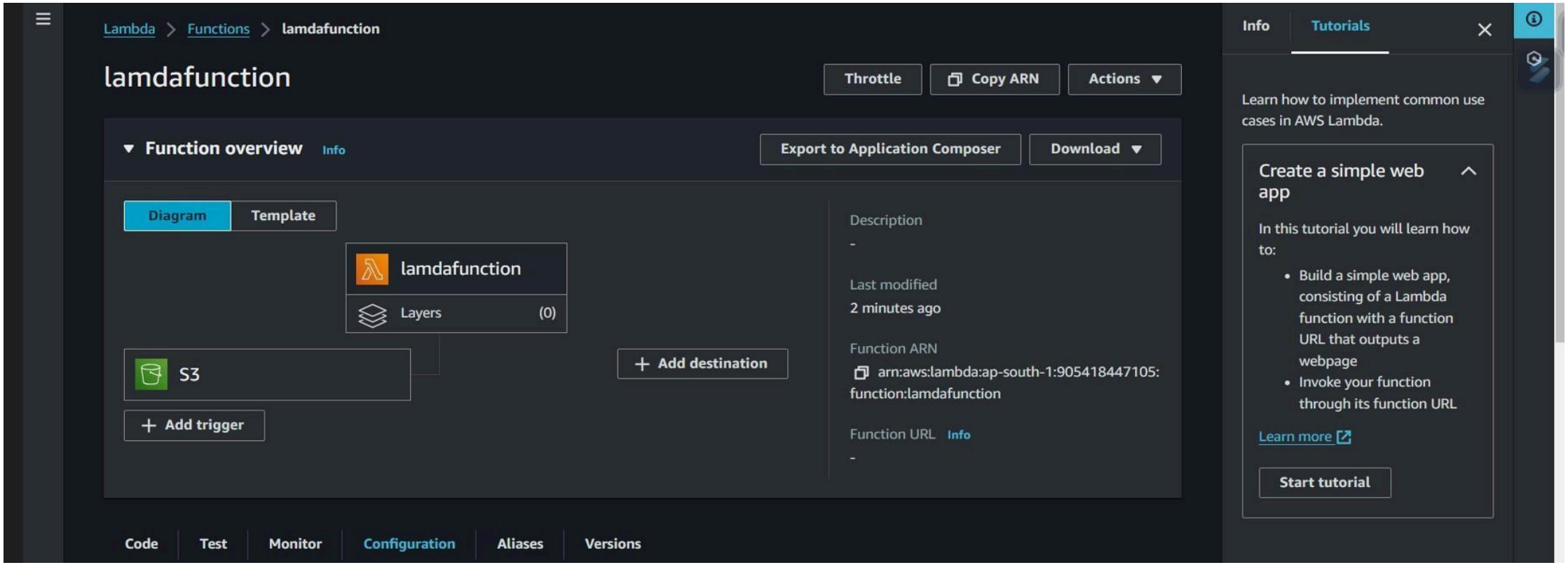
Step 7: Creating S3 Trigger

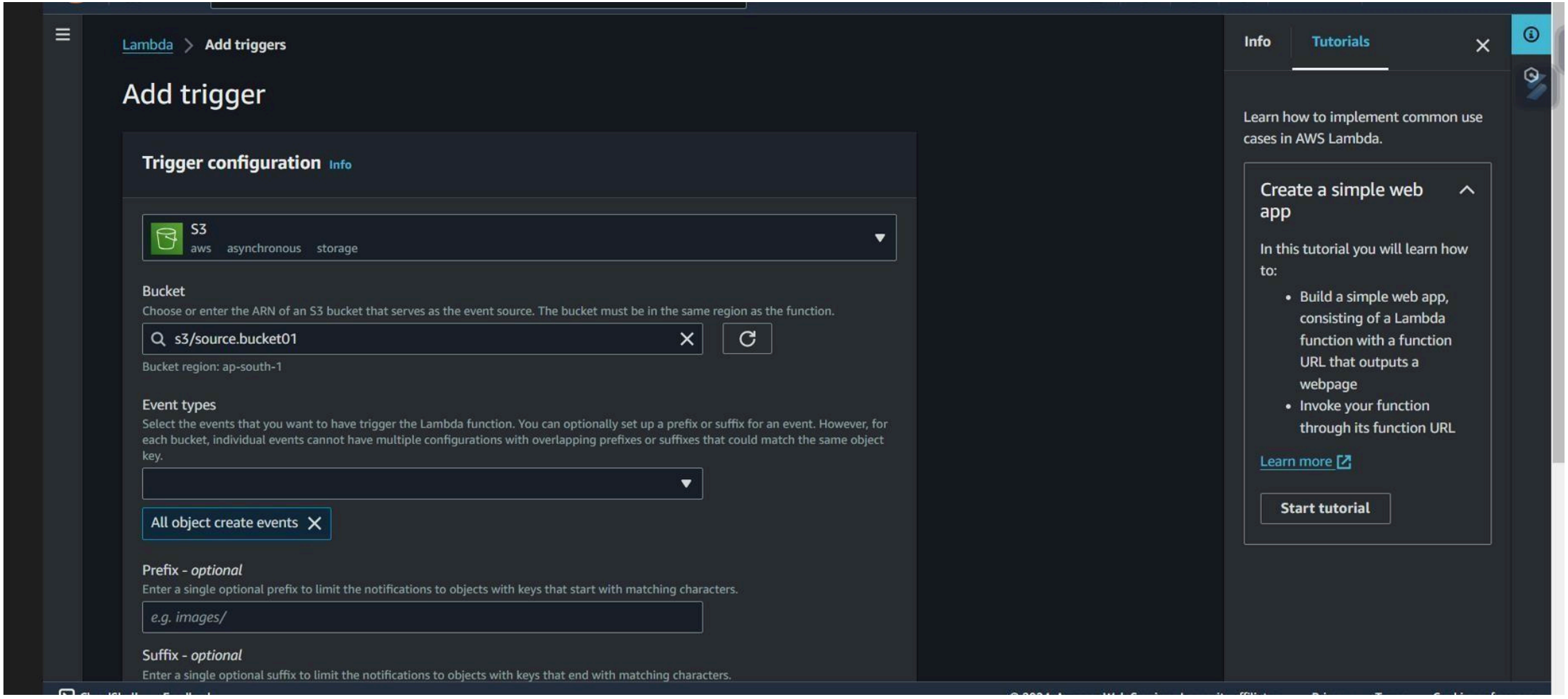
*Add trigger

*Select s3

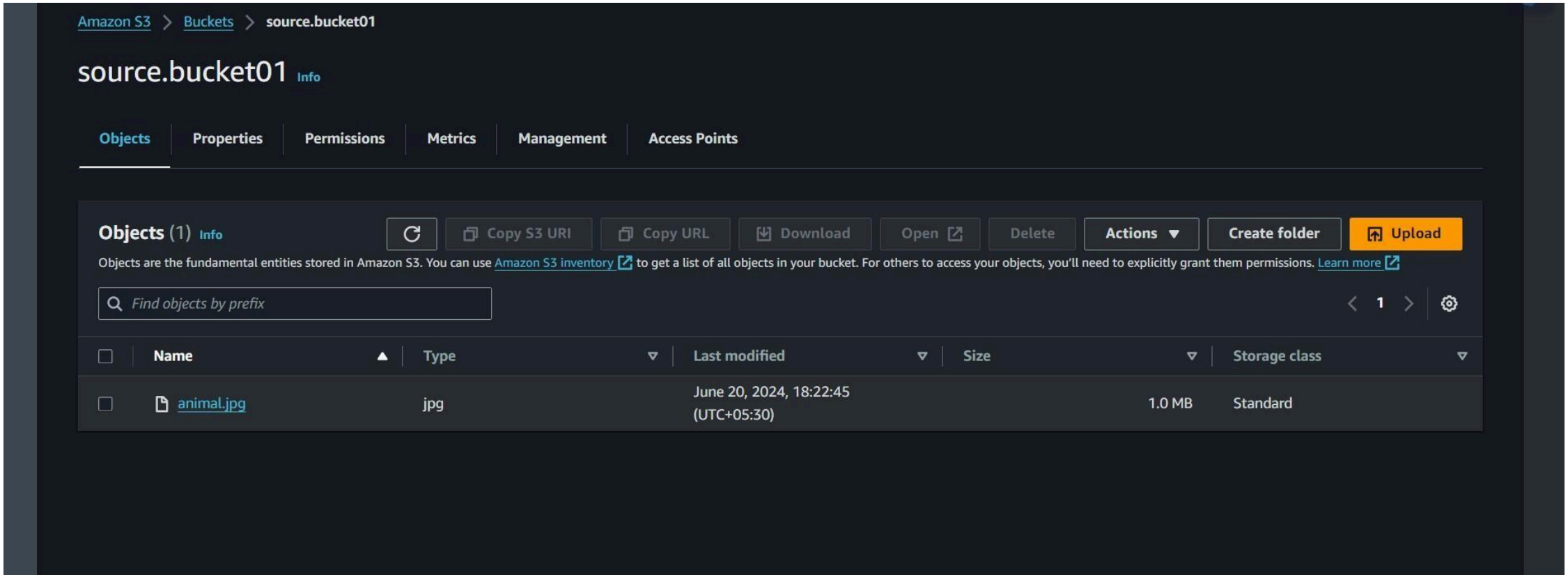
*choose source Bucket name

*Now Add

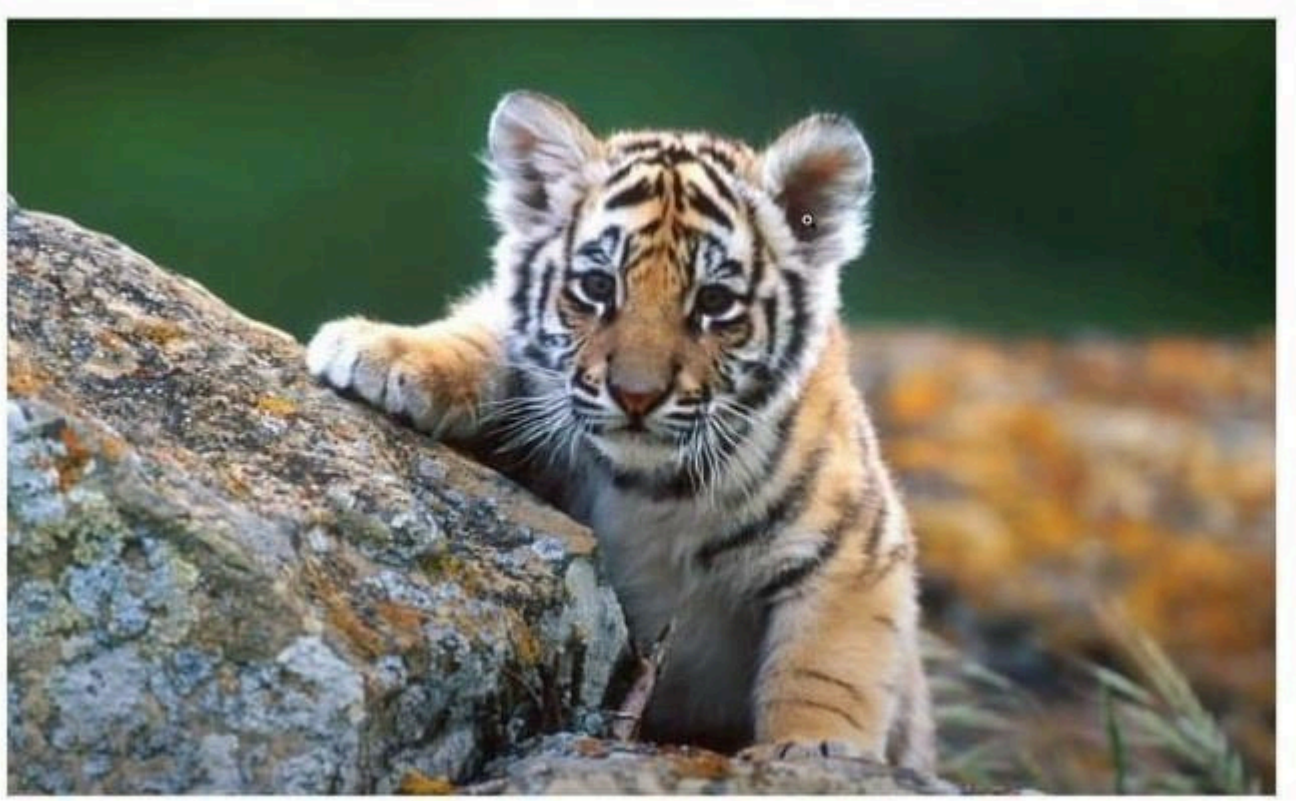



Lambda > Add triggers
Add trigger
Trigger configuration Info
S3
aws asynchronous storage
Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/source.bucket01
Bucket region: ap-south-1
Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
All object create events
Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/
Suffix - optional
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial

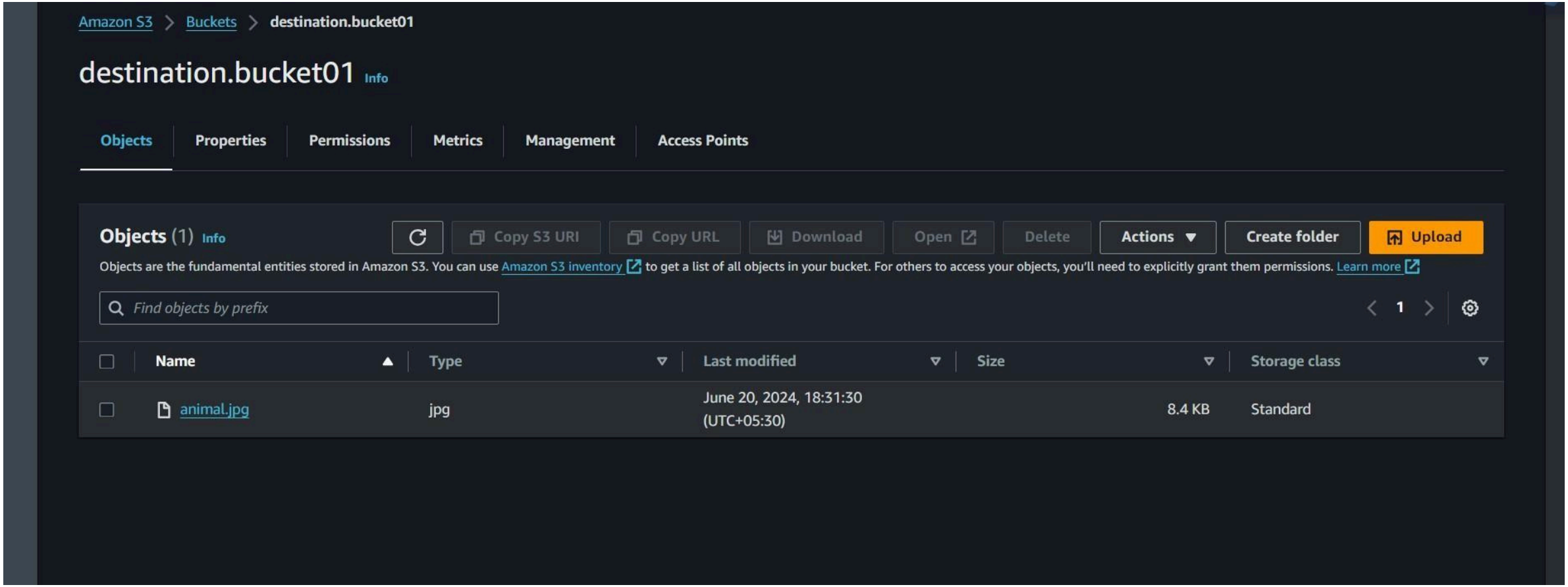
Step 8: Upload image in Source Bucket



Original Image



Destination Bucket


Amazon S3 > Buckets > destination.bucket01
destination.bucket01 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
animal.jpg
jpg
June 20, 2024, 18:31:30 (UTC+05:30)
8.4 KB
Standard

Resize Image