

Deep Chavan

T11-15

## Assignment No. 9

**Aim:** To learn how to use Lambda in order to find the ContentType of Object uploaded in S3 Bucket.

**Theory:**

### Create bucket:

Amazon S3 > Buckets > Create bucket

### Create bucket info

Buckets are containers for data stored in S3. [Learn more](#)

#### General configuration

Bucket name

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

AWS Region

Asia Pacific (Mumbai) ap-south-1

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

#### Object Ownership info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

#### Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all, in order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

create a new policy from iam dashboard;

while creating policy select json tab and paste the following code:

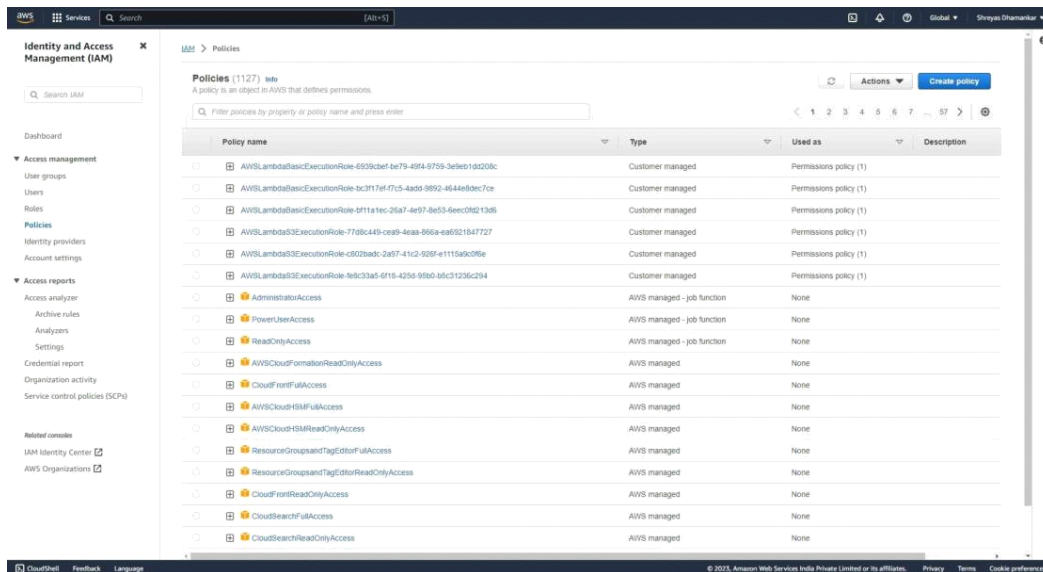
```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "s3:*",  
      "Resource": "*" } ]  
}
```

```

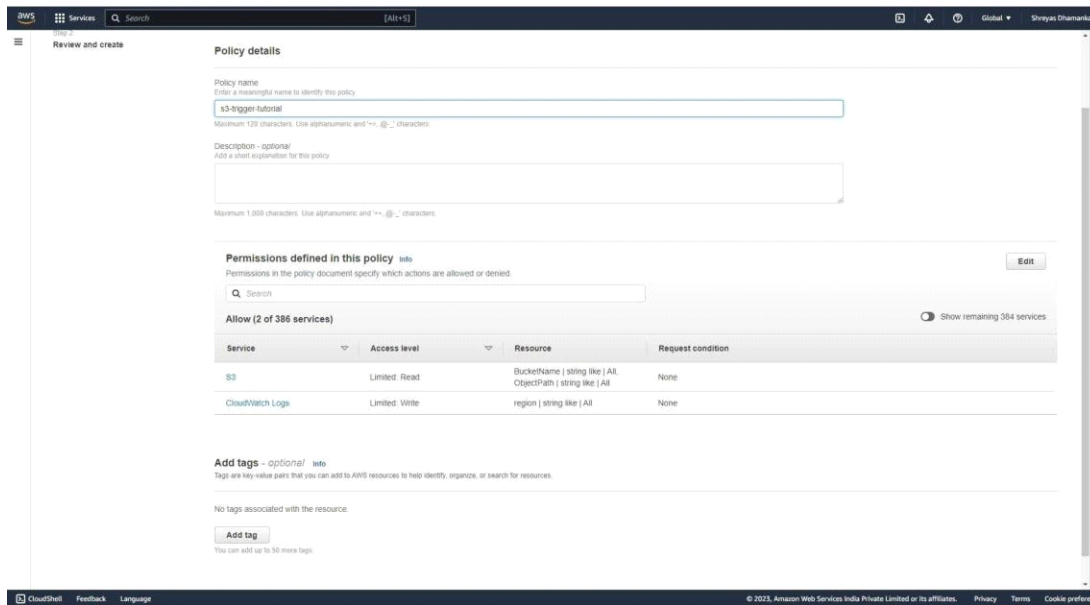
"Effect": "Allow",
"Action": [
    "logs:PutLogEvents",
    "logs:CreateLogGroup",
    "logs:CreateLogStream"
],

"Resource": "arn:aws:logs:*:*:*"
},
{
    "Effect":
    "Allow",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": "arn:aws:s3:::*/*"
}
]
}

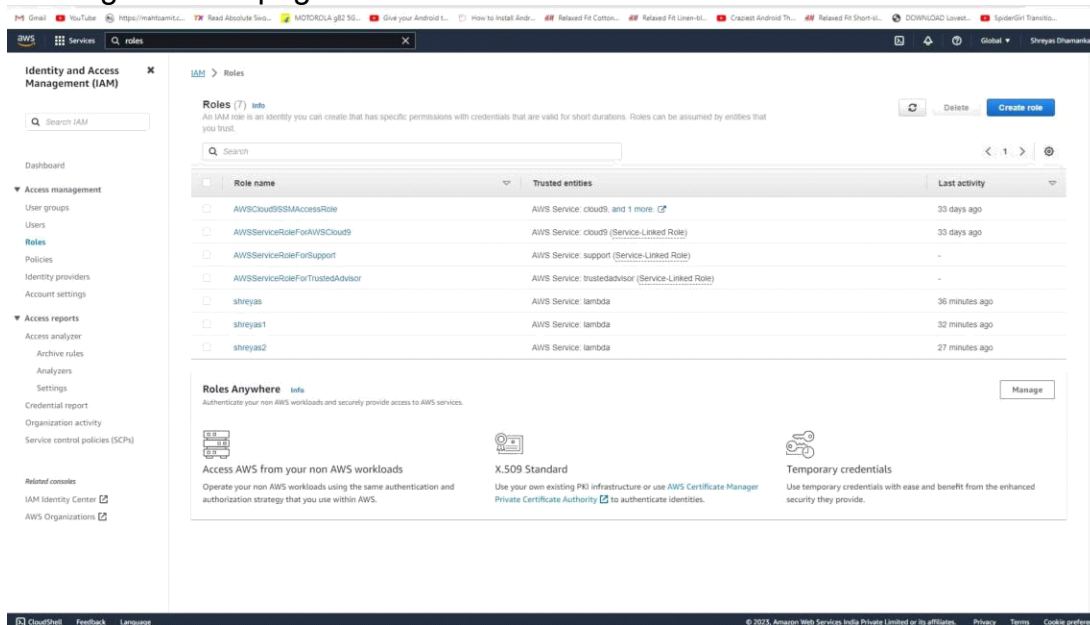
```



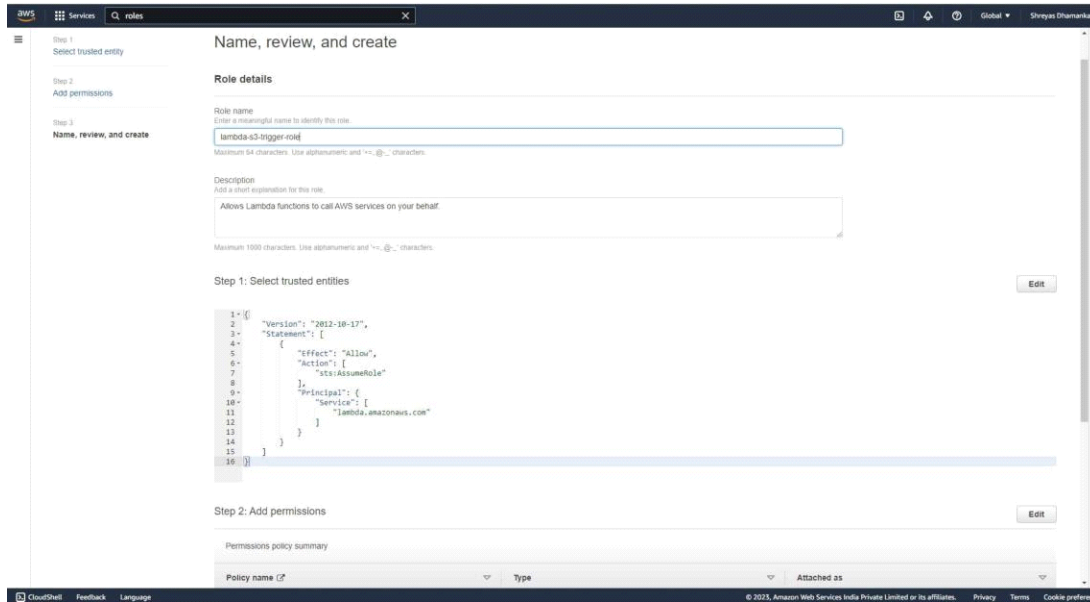
create policy and name it:



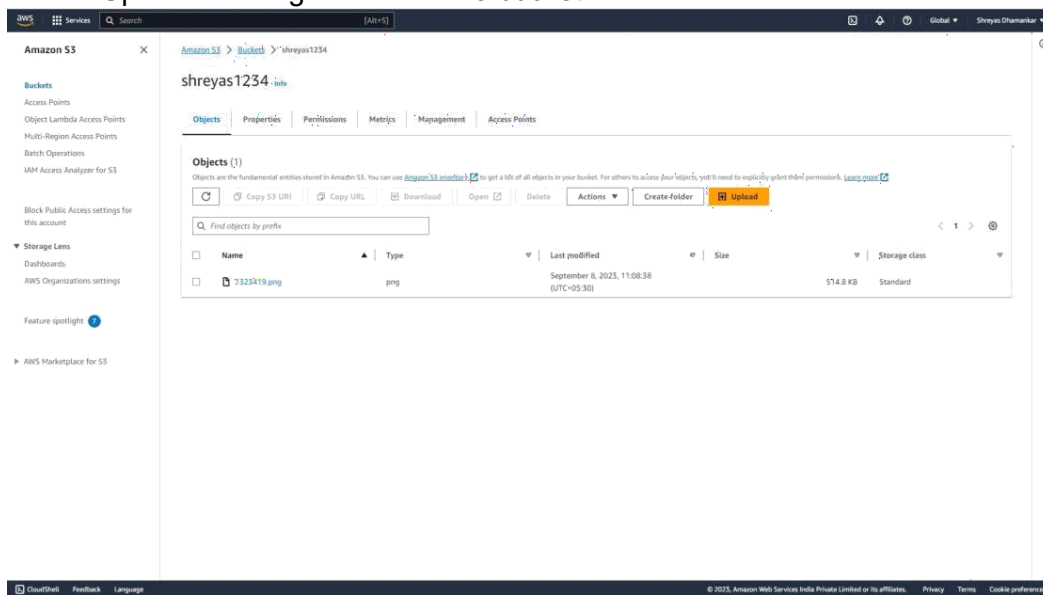
go to roles page and select create a new role



under policies for role select the policy that you have created and click next.  
Then name the role as follows:



Upload an image file in the S3 bucket.



Go to lambda dashboard in aws and create a new function named s3-trigger tutorial. select change execution code and choose the option use existing role. Select lambda- s3-trigger-role.

Once function is created go to code panel and paste the following

code. import json

import

urllib.parse

import boto3

print('Loadin

```

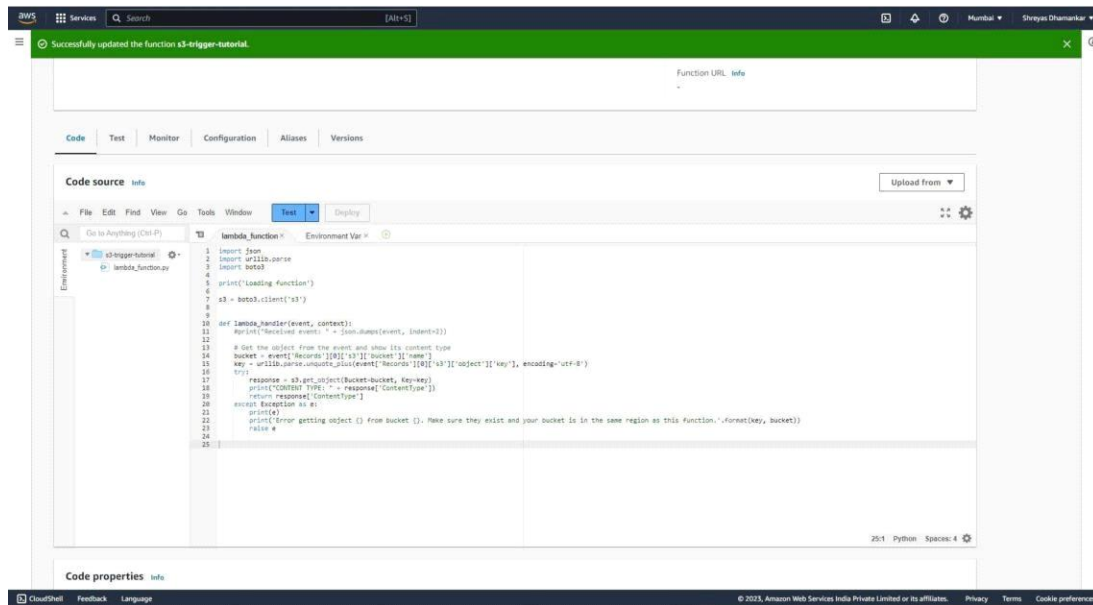
function') s3 =
boto3.client('s3')
def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event,
    indent=2)) # Get the object from the event and show
    its content

    type bucket = event['Records'][0]['s3']['bucket']['name']

    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
    encoding='utf- 8') try:

        response = s3.get_object(Bucket=bucket, Key=key)
    print("An object : "+response['ContentType']+" has been added to S3 Bucket")
    print("CONTENT TYPE: " +
    response['ContentType']) return
    response['ContentType']
except Exception as e:
    print(e)
    print('Error getting object {} from bucket {}. Make sure they exist and your bucket is
    in the same region as this function.'.format(key, bucket))
    raise e
then select deploy changes.

```



then click on test and create a new custom event named MyTestEvent.

For Template, choose S3 Put In the Event JSON, replace the following values:

Replace us-east-1 with the region you created your Amazon S3 bucket in.

Replace both instances of example-bucket with the name of your own

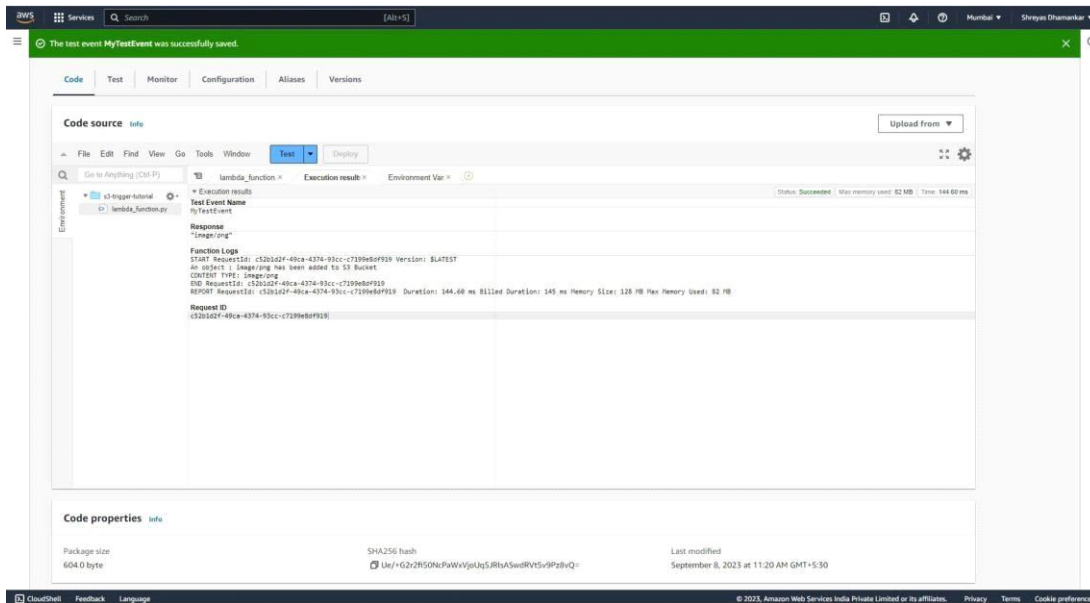
Amazon S3 bucket. Replace test%2FKey with the name of the test object you

uploaded to your bucket earlier

(for example, factorial.py).

Click on save and press Test. You will see the results in Execution tab.

Check execution Tab for the CONTENT TYPE. If you have uploaded the file properly you will get the file type that you have uploaded in the S3 Bucket. Check Function logs where the line has been printed that "An Image has been Uploaded to S3 Bucket".



## Conclusion:

In this assignment we learnt how to use Lambda function to log what type of object has been uploaded to S3 Bucket.