

7. Laboratory Exercise

A. Procedure:

a. Perform following steps.

- Create Execution role using IAM

Select trusted entity [Info](#)

Trusted entity type



AWS service

Allow AWS services like EC2, Lambda, or others to perform actions in this account.



AWS account

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.



Web identity

Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.



SAML 2.0 federation

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.



Custom trust policy

Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases



EC2

Allows EC2 instances to call AWS services on your behalf.



Lambda

Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case



AmazonS3FullAccess

CloudWatchLogsFullAccess

AWSLambdaBasicExecutionRole

Step 1: Select trusted entities

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "sts:AssumeRole"  
8       ],  
9       "Principal": {  
10        "Service": [  
11          "lambda.amazonaws.com"  
12        ]  
13      }  
14    }  
15  ]  
16 }
```

Step 2: Add permissions

[Edit](#)

Permissions policy summary

Policy name 🔗	Type	Attached as
AWSLambdaBasicExecutionRole	AWS managed	Permissions policy
CloudWatchLogsFullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

• empty s3 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

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

- create Lambda function from console

[Lambda](#) > [Functions](#) > Create function

Create function [Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ **Author from scratch**
Start with a simple Hello World example.

☐ **Use a blueprint**
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

[↻](#)

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64
☐ arm64

- Invoke Lambda function and verify results.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

[↻](#)

[View the exp12role role](#) on the IAM console.


✔ Successfully created the function exp12. You can now change its code and configuration. To invoke your function with a test event, choose "Test". [✕](#)


[Lambda](#) > [Functions](#) > exp12

exp12

Throttle [Copy ARN](#) Actions ▼

▼ Function overview [Info](#)


 exp12

 Layers (0)

[+ Add trigger](#) [+ Add destination](#)

Description
-

Last modified
14 seconds ago

Function ARN
 arn:aws:lambda:us-east-1:928565427545:function:exp12

Function URL [Info](#)
-

- lambda function script

```
1 import json
2
3 import boto3
4
5
6 s3=boto3.client('s3')
7
8
9 def lambda_handler(event,context):
10
11     bucket='srushiti12bucket'
12
13     dataToUpload = {}
14
15     dataToUpload['PID'] = '211103'
16
17     dataToUpload['DEPT'] = 'INFT'
18
19     dataToUpload['NAME'] = 'SRUSHTI'
20
21     dataToUpload['FILE'] = 'ABC_111'
22
23     fileName = 'ABC_111' + '.json'
24
25     uploadByteStream= bytes(json.dumps(dataToUpload).encode('UTF-8'))
26
27     s3.put_object(Bucket=bucket,Key=fileName,Body=uploadByteStream)
28
29     print('object has been uploaded')
```

- Create new event for lambda function

Configure test event ×

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event

☐ Edit saved event

Event name

event2

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☐ Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☒ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world ▼

Event JSON Format JSON

```
1 {
2   "key1": "value1",
3   "key2": "value2",
```

Execution results

Status: **Succeeded** Max memory used: 78 MB Time: 232.31 ms

Test Event Name
event2

Response
null

Function Logs
START RequestId: d337ace6-f55b-462e-9889-4516fd78ebd9 Version: \$LATEST
object has been uploaded
END RequestId: d337ace6-f55b-462e-9889-4516fd78ebd9
REPORT RequestId: d337ace6-f55b-462e-9889-4516fd78ebd9 Duration: 232.31 ms Billed Duration: 233 ms Memory Size: 128 MB Max Memory Used: 78 MB Init Duration: 0.00 ms

Request ID
d337ace6-f55b-462e-9889-4516fd78ebd9

Amazon S3 > Buckets > srushti12bucket

srushti12bucket [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (1)
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](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#)

[Upload](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	ABC_111.json	json	August 31, 2023, 16:16:13 (UTC+05:30)	71.0 B	Standard

Amazon S3 > Buckets > srushti12bucket > ABC_111.json

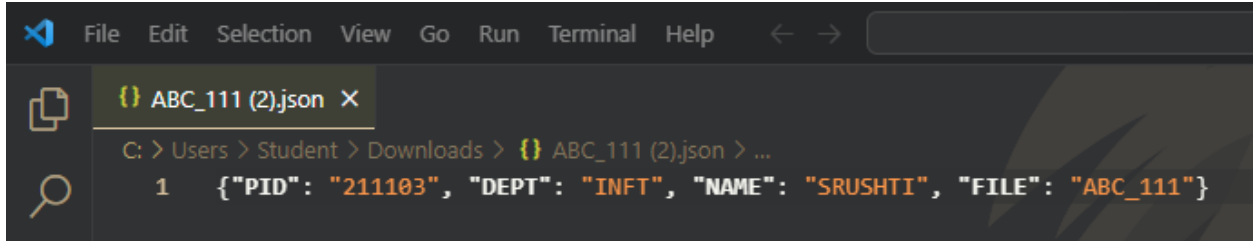
ABC_111.json [Info](#)

[Copy S3 URI](#) [Download](#) [Open](#) [Object actions](#)

[Properties](#) | [Permissions](#) | [Versions](#)

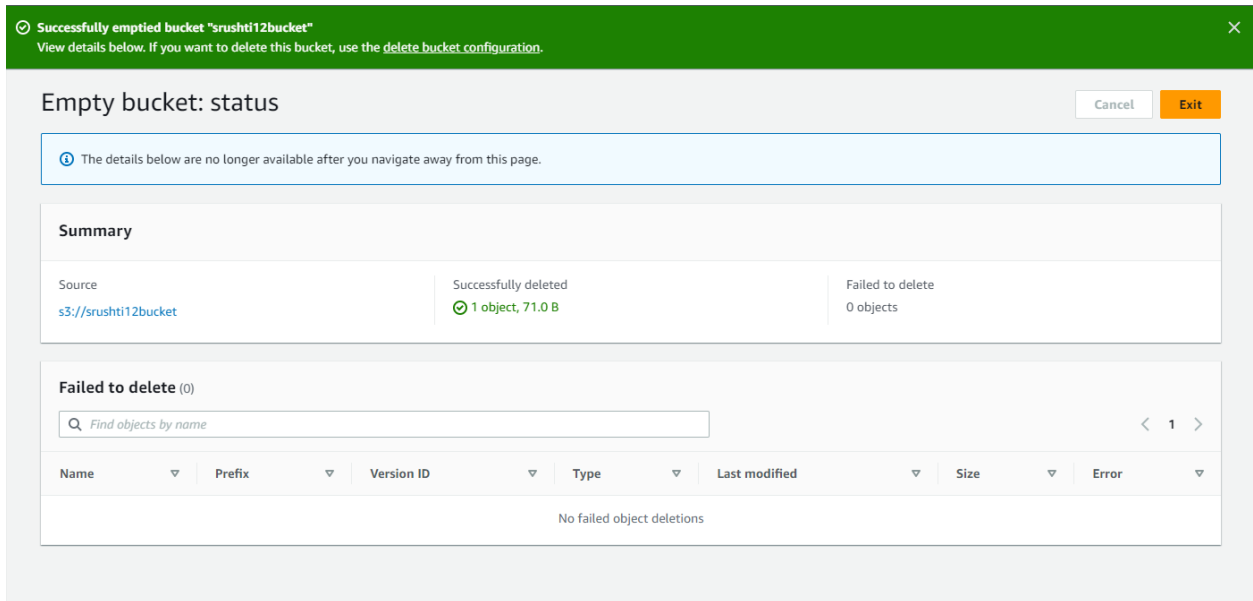
Object overview

Owner srushtiss1305	S3 URI s3://srushti12bucket/ABC_111.json
AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) arn:aws:s3:::srushti12bucket/ABC_111.json
Last modified August 31, 2023, 16:16:13 (UTC+05:30)	Entity tag (Etag) fc2515f71cee56d9a1fb5c2619c03283
Size 71.0 B	Object URL https://srushti12bucket.s3.amazonaws.com/ABC_111.json
Type json	
Key ABC_111.json	



```
{ "PID": "211103", "DEPT": "INFT", "NAME": "SRUSHTI", "FILE": "ABC_111" }
```

- Clean up resources.



Successfully emptied bucket "srushti12bucket"
View details below. If you want to delete this bucket, use the [delete bucket configuration](#).

Empty bucket: status Cancel Exit

ⓘ The details below are no longer available after you navigate away from this page.

Summary

Source s3://srushti12bucket	Successfully deleted ✔ 1 object, 71.0 B	Failed to delete 0 objects
--	--	-------------------------------

Failed to delete (0)

< 1 >

Name	Prefix	Version ID	Type	Last modified	Size	Error
No failed object deletions						