

AWS ASSIGNMENT

TAS-206 Megha Singh

- Login to AWS CLI using the access key.

Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

Command Line Interface (CLI)

You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code

You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service

You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Third-party service

You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

Application running outside AWS

You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

Other

Your use case is not listed here.

Set description tag - optional Info

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

[Cancel](#)[Previous](#)[Create access key](#)

//Configuring AWS Credentials

```
Last login: Fri May  3 11:48:34 on console
[meghasingh@Megha-ka-MacBook-Air ~ % aws configure
AWS Access Key ID [*****]: AKIA47CR2KENINVKLTE7
AWS Secret Access Key [*****]: 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
Default region name [us-east-1]: us-east-1
Default output format [json]: json
```

a) Create a S3 bucket in aws cli

1. Create a IAM role with s3 full-access policy

```
meghasingh@Megha-ka-MacBook-Air ~ % aws iam create-role --role-name meghas-s3-full --assume-role-policy-document file:///Users/meghasingh/Desktop/AWS2/trust-p
olicy.json
{
  "Role": {
    "Path": "/",
    "RoleName": "meghas-s3-full",
    "RoleId": "AROA47CR2KENQN5Y3GKM",
    "Arn": "arn:aws:iam:891377307930:role/meghas-s3-full",
    "CreateDate": "2024-05-03T06:48:22+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
  }
}
meghasingh@Megha-ka-MacBook-Air ~ % aws sts get-caller-identity
{
  "UserId": "891377307930",
  "Account": "891377307930",
  "Arn": "arn:aws:iam::891377307930:root"
}
```

- Policy created
- Attach the policy to the role

b) // Create an EC2 instance with above role

- Create Instance profile
- Create a EC2 instance that will be used throughout the project
- Pass the appropriate Amazon Machine Image, keyName and other required credentials.
- EC2 instance (MyEC2) successfully created and running.

```
}
```

```
meghasingh@Megha-MacBook-Air ~ % aws ec2 run-instances --image-id ami-0af6548a402f12af1 --count 1 --instance-type t2.micro --iam-instance-profile Name=rolemegha
```

```
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0af6548a402f12af1",
      "InstanceId": "i-0cf70cd911433f01f",
      "InstanceType": "t2.micro",
      "LaunchTime": "2024-05-03T07:19:37+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1c",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-28-232.ec2.internal",
      "PrivateIpAddress": "172.31.28.232",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-026718ad580d2bd8c",
      "VpcId": "vpc-06fd3d74157b7210a",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "cc253de3-239f-4758-a05f-992f19bb8d79",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "xen",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::891377307930:instance-profile/rolemegha",
        "Id": "AIPMA7CR2K6NLB4YTXLIG"
      },
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2024-05-03T07:19:37+00:00",
            "AttachmentId": "eni-attach-073155de475142fce",
            "DeviceIndex": 0,
            "Status": "attached"
          }
        }
      ]
    }
  ]
}
```

Instances (1/1) Info		C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
<input type="text"/> Find Instance by attribute or tag (case-sensitive)		All states ▾					
Instance state = running X		Clear filters					
<input checked="" type="checkbox"/>	Name ▼	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾
<input checked="" type="checkbox"/>	myec2	i-0a56b38d686a97b46	Running ? S	t3.micro	2/2 checks passed View alarms +	us-east-1a	ec2-...

c) //Create a bucket from AWS CLI

- Create bucket using CLI command
- Bucket Name - meghasinghbuckie

```
meghasingh@Megha-ka-MacBook-Air ~ % aws s3api create-bucket --bucket meghasinghbuckie --region us-east-1
{
    "Location": "/meghasinghbuckie"
}
```

- Bucket successfully created.

2) Put files in S3 bucket from lambda

- Create custom role for AWS lambda which will have only put object access

```
meghasingh@Megha-ka-MacBook-Air AWS2 % aws iam create-role --role-name lambda-s3-put-role --assume-role-policy-document '{"Version":"2012-10-17", "Statement":[{"Effect":"Allow", "Principal":["Service":"lambda.amazonaws.com"], "Action":"sts:AssumeRole"}]}'
aws iam attach-role-policy --role-name lambda-s3-put-role --policy-arn arn:aws:iam::891377307930:role/meghas-s3-full
{
    "Role": {
        "Path": "/",
        "RoleName": "lambda-s3-put-role",
        "RoleId": "AROA47CR2KENF3E3SJN2",
        "Arn": "arn:aws:iam::891377307930:role/lambda-s3-put-role",
        "CreateDate": "2024-05-03T07:48:05+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "lambda.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

//creating the policy



b) // Add role to generate and access Cloudwatch logs

```
meghasingh@Megha-ka-MacBook-Air aws % aws events put-rule --name "RunLambdaEveryMinute" --schedule-expression "rate(1 minute)"  
lambda_function.zip generate_json.py  
{  
  "RuleArn": "arn:aws:events:us-east-1:891377307930:rule/RunLambdaEveryMinute"  
}
```

```
{ } lambda-create-function-policy.json > ...
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "lambda>CreateFunction",
7              "Resource": "*"
8          }
9      ]
10 }
11
```

2.c In Python Script ,generate json in given format and save.json file in bucket created.

- Create a aws lambda function using python file called (lambda_function.py).

```
meghasingh@Megha-ka-MacBook-Air aws % aws lambda create-function \
    --function-name my-lambda-function \
    --runtime python3.8 \
    --role arn:aws:iam::891377307930:role/LambdaS3PutRole \
    --handler lambda_function.lambda_handler \
    --zip-file fileb:///Users/meghasingh/Desktop/aws/lambda_function.zip
```

- Invoke the lambda expression.
- The lambda function code.
- Added str(datetime.now()) in timestamp field so that each log will have its own custom timestamp.

```
{ } cloudwatch-events-put-targets-policy.json > ...
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "events:PutTargets",
7              "Resource": "arn:aws:events:us-east-1:891377307930:rule/RunLambdaEveryMinute"
8          }
9      ]
10 }
11 }
```

```
meghasingh@Megha-ka-MacBook-Air aws % aws iam attach-user-policy --user-name user2 --policy-arn arn:aws:iam::891377307930:policy/CloudWatchEventsPutTargetsPolicy

meghasingh@Megha-ka-MacBook-Air aws % export POLICY_ARN=arn:aws:iam::891377307930:policy/CloudWatchEventsPutTargetsPolicy
```

```
{ } cloudwatch-events-describe-rule-policy.json > ...
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "events:DescribeRule",
7              "Resource": "arn:aws:events:us-east-1:891377307930:rule/RunLambdaEveryMinute"
8          }
9      ]
10 }
11 }
```

```
names are not allowed.

meghasingh@Megha-ka-MacBook-Air aws % aws iam attach-user-policy --user-name user2 --policy-arn arn:aws:iam::891377307930:policy/CloudWatchEventsDescribeRulePolicy

meghasingh@Megha-ka-MacBook-Air aws % export POLICY_ARN=arn:aws:iam::891377307930:policy/CloudWatchEventsDescribeRulePolicy

meghasingh@Megha-ka-MacBook-Air aws %
```

```
meghasingh@Megha-ka-MacBook-Air aws % aws events describe-rule --name "RunLambdaEveryMinute"

{
    "Name": "RunLambdaEveryMinute",
    "Arn": "arn:aws:events:us-east-1:891377307930:rule/RunLambdaEveryMinute",
    "ScheduleExpression": "rate(1 minute)",
    "State": "ENABLED",
    "EventBusName": "default",
    "CreatedBy": "891377307930"
}
meghasingh@Megha-ka-MacBook-Air aws %
```

2.d Schedule the job to run every three minutes and stop execution after 3 times.

```
{} my-lambda-update-policy.json > ...
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "lambda:UpdateFunctionCode",
7              "Resource": "arn:aws:lambda:us-east-1:891377307930:function:my-lambda-function"
8          }
9      ]
10 }
11
```

```
meghasingh@Megha-ka-MacBook-Air aws % aws lambda update-function-code \
    --function-name my-lambda-function \
    --zip-file fileb://generate_json.zip

{
    "FunctionName": "my-lambda-function",
    "FunctionArn": "arn:aws:lambda:us-east-1:891377307930:function:my-lambda-function",
    "Runtime": "python3.8",
    "Role": "arn:aws:iam::891377307930:role/LambdaS3PutRole",
    "Handler": "generate_json.lambda_handler",
    "CodeSize": 442,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2024-05-03T09:05:21.000+0000",
    "CodeSha256": "/2N89IeK7XI9Im0rJ0W/6a97dDq5+S3Zm0cdS+94rls=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "6a172ef8-f82f-482e-8183-ba68ecc8d60f",
    "State": "Active",
    "LastUpdateStatus": "InProgress",
    "LastUpdateStatusReason": "The function is being created.",
    "LastUpdateStatusReasonCode": "Creating",
    "PackageType": "Zip",
    "Architectures": [
        "x86_64"
    ],
    "EphemeralStorage": {
        "Size": 512
    },
    "SnapStart": {
        "ApplyOn": "None",
        "OptimizationStatus": "Off"
    },
    "RuntimeVersionConfig": {
        "RuntimeVersionArn": "arn:aws:lambda:us-east-1::runtime:830ab1165f1cdc47b8ae1e516f3936d8a176aec50d0699efd25a8bfeabb797b4"
    },
    "LoggingConfig": {
        "LogFormat": "Text",
        "LogGroup": "/aws/lambda/my-lambda-function"
    }
}
```

```
{ } update-function-policy.json > ...
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "lambda:UpdateFunctionConfiguration",
7              "Resource": "arn:aws:lambda:us-east-1:891377307930:function:my-lambda-function"
8          }
9      ]
10 }
11
```

```
{ } invoke-policy.json > ...
1   {
2     "Version": "2012-10-17",
3     "Statement": [
4       {
5         "Effect": "Allow",
6         "Action": "lambda:InvokeFunction",
7         "Resource": "arn:aws:lambda:us-east-1:891377307930:function:my-lambda-function"
8       }
9     ]
10  }
11
```

```
{ } payload.json > ...
1   {
2     "name": "John",
3     "age": 30
4   }
5
```

d) // Schedule the job to run every minute. Stop execution after 3 runs

```
// my-lambda-function
import json
import boto3
import os
from datetime import datetime

def lambda_handler(event, context):
    # Get current execution count from environment variable
    execution_count = int(os.environ.get('EXECUTION_COUNT', 0))

    # Increment execution count
    execution_count += 1

    # Update execution count in environment variable
    os.environ['EXECUTION_COUNT'] = str(execution_count)

    # Stop execution after 3 runs
    if execution_count > 3:
```

```

        return {
            'statusCode': 200,
            'body': 'Execution stopped after 3 runs'
        }

# Generate JSON data
json_data = {
    "transaction_id": 12345,
    "payment_mode": "card/netbanking/upi",
    "amount": 200.0,
    "customer_id": 101,
    "timestamp": str(datetime.now())
}

# Save JSON data to S3
bucket_name = 'kamalmeghubuckie'
file_name =
f"transaction_{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}.json"

s3 = boto3.client('s3')
s3.put_object(Bucket=bucket_name, Key=file_name,
Body=json.dumps(json_data))

return {
    'statusCode': 200,
    'body': json.dumps('JSON file saved successfully')
}

```

```

meghasingh@Megha-ka-MacBook-Air aws % aws lambda invoke --function-name my-lambda-function --payload '{}' output.json
{
    "StatusCode": 200,
    "ExecutedVersion": "$LATEST"
}

```

Function overview [Info](#)

[Export to Application Composer](#) [Download ▾](#)

Diagram [Template](#)

my-lambda-function

Description

Last modified **6 minutes ago**

Function ARN

arn:aws:lambda:us-east-1:891377307930:function:my-lambda-function

Function URL [Info](#)

Code [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

Code source [Info](#)

[Upload from ▾](#)

[File](#) [Edit](#) [Find](#) [View](#) [Go](#) [Tools](#) [Window](#) [Test](#) [Deploy](#) [⚙️](#)

e) // cloudwatch logs are generated

Log groups (4)

By default, we only load up to 10000 log groups.

Log group	Log class	Anomaly d...	Data ...	Sensi...	Retenti...
/aws/lambda/func2	Standard	Configure	-	-	Never expire
/aws/lambda/meghafunction	Standard	Configure	-	-	Never expire
/aws/lambda/my-lambda-function	Standard	Configure	-	-	Never expire
RDSOSMetrics	Standard	Configure	-	-	1 month

3) API gateway - Lambda integration

3.a Modify lambda function to accept parameters and return fileName

a) //modifying lamda function

```
import json  
import boto3  
from datetime import datetime
```

```
def lambda_handler(event, context):  
  
    # Get parameters from the request body  
    transaction_id = event.get('transaction_id')  
  
    payment_mode = event.get('payment_mode')  
  
    amount = event.get('amount')  
  
    customer_id = event.get('customer_id')
```

```
# Check if required parameters are present  
  
if not all([transaction_id, payment_mode, amount, customer_id]):  
  
    return {  
  
        'statusCode': 400,  
  
        'body': json.dumps({'error': 'Missing required parameters'})  
  
    }
```

```
# Generate JSON data with current timestamp
```

```
timestamp = datetime.now().isoformat()

json_data = {

    "transaction_id": transaction_id,
    "payment_mode": payment_mode,
    "amount": amount,
    "customer_id": customer_id,
    "timestamp": timestamp
}

# Save JSON data to S3

bucket_name = 'kamalmeghubuckie'
file_name = f"transaction_{timestamp}.json"

s3 = boto3.client('s3')
s3.put_object(Bucket=bucket_name, Key=file_name, Body=json.dumps(json_data))

return {
    'statusCode': 200,
    'body': json.dumps({'file_name': file_name})
}

b) Lambda code changed .
c) Lambda function tested.
```

```

meghasingh@Megha-ka-MacBook-Air aws % aws lambda invoke --function-name my-lambda-function --cli-binary-format raw-in-base64-out --payload '{"transaction_id": 12345, "payment_mode": "card/netbanking/upi", "amount": 200.0,"customer_id": 101}' output.json
{
    "StatusCode": 200,
    "ExecutedVersion": "$LATEST"
}
meghasingh@Megha-ka-MacBook-Air aws % cat output.json
{"statusCode": 200, "body": "{\"file_name\": \"transaction_2024-05-03T10:54:44.415164.json\""}}

meghasingh@Megha-ka-MacBook-Air aws % aws lambda invoke --function-name my-lambda-function --cli-binary-format raw-in-base64-out --payload file://payload.json
output.json
{
    "StatusCode": 200,
    "ExecutedVersion": "$LATEST"
}
meghasingh@Megha-ka-MacBook-Air aws %

```

Objects (5) [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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 transaction_2024-05-03_09-36-59.json	json	May 3, 2024, 15:07:03 (UTC+05:30)	144.0 B	Standard
<input type="checkbox"/>	 transaction_2024-05-03_09-38-12.json	json	May 3, 2024, 15:08:13 (UTC+05:30)	144.0 B	Standard
<input type="checkbox"/>	 transaction_2024-05-03_09-39-10.json	json	May 3, 2024, 15:09:11 (UTC+05:30)	144.0 B	Standard
<input type="checkbox"/>	 transaction_2024-05-03T10:54:44.415164.json	json	May 3, 2024, 16:24:47 (UTC+05:30)	144.0 B	Standard
<input type="checkbox"/>	 transaction_2024-05-03T10:56:27.229624.json	json	May 3, 2024, 16:26:28 (UTC+05:30)	129.0 B	Standard

Log events

Actions ▾

Start tailing

Create metric filter

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#) Filter events - press enter to search

1m

1h



Local timezone ▾

Display ▾



▶ | Timestamp

Message

No older events at this moment. [Retry](#)

▶ 2024-05-03T16:21:43.209+05:30	INIT_START Runtime Version: python:3.8.v48 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:830ab...
▶ 2024-05-03T16:21:43.501+05:30	START RequestId: 1a2db8f7-fb36-4438-8ed1-cde36b3625aa Version: \$LATEST
▶ 2024-05-03T16:21:43.519+05:30	END RequestId: 1a2db8f7-fb36-4438-8ed1-cde36b3625aa
▶ 2024-05-03T16:21:43.519+05:30	REPORT RequestId: 1a2db8f7-fb36-4438-8ed1-cde36b3625aa Duration: 2.10 ms Billed Duration: 3 ms Memory S...
▶ 2024-05-03T16:24:44.414+05:30	START RequestId: bc5327b2-10a8-4ab5-be48-9cd77246e06a Version: \$LATEST
▶ 2024-05-03T16:24:46.959+05:30	END RequestId: bc5327b2-10a8-4ab5-be48-9cd77246e06a
▶ 2024-05-03T16:24:46.959+05:30	REPORT RequestId: bc5327b2-10a8-4ab5-be48-9cd77246e06a Duration: 2545.07 ms Billed Duration: 2546 ms Me...
▶ 2024-05-03T16:26:27.229+05:30	START RequestId: 5ef55734-3e54-4d9c-948a-e8ee0ed080d7 Version: \$LATEST
▶ 2024-05-03T16:26:27.500+05:30	END RequestId: 5ef55734-3e54-4d9c-948a-e8ee0ed080d7
▶ 2024-05-03T16:26:27.500+05:30	REPORT RequestId: 5ef55734-3e54-4d9c-948a-e8ee0ed080d7 Duration: 271.31 ms Billed Duration: 272 ms Memo...

No newer events at this moment. Auto retry paused. [Resume](#)

S | Services | Search | [Option+S]

Lambda > Functions > my-lambda-function

my-lambda-function

Throttle | Copy ARN | Actions ▾

▼ Function overview [Info](#)

[Export to Application Composer](#) | [Download ▾](#)

[Diagram](#) | [Template](#)

 my-lambda-function

Layers (0)

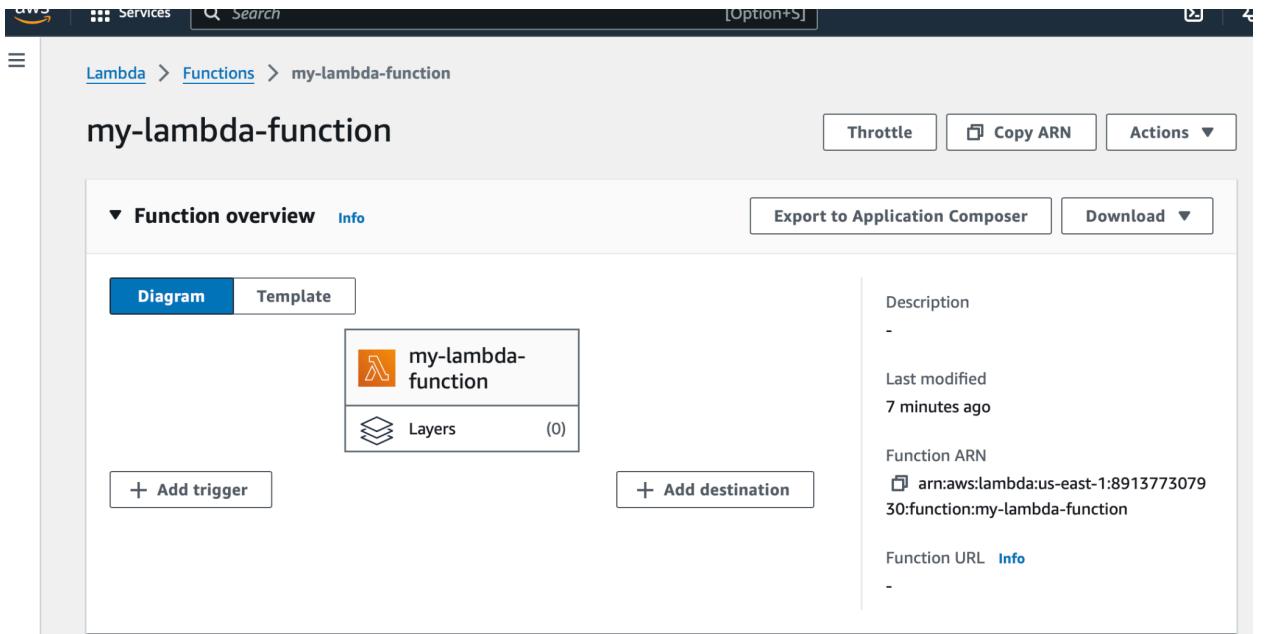
+ Add trigger | + Add destination

Description
-

Last modified
7 minutes ago

Function ARN
 arn:aws:lambda:us-east-1:891377307930:function:my-lambda-function

Function URL [Info](#)
-



Amazon S3 > Buckets > kamalmeghubuckie

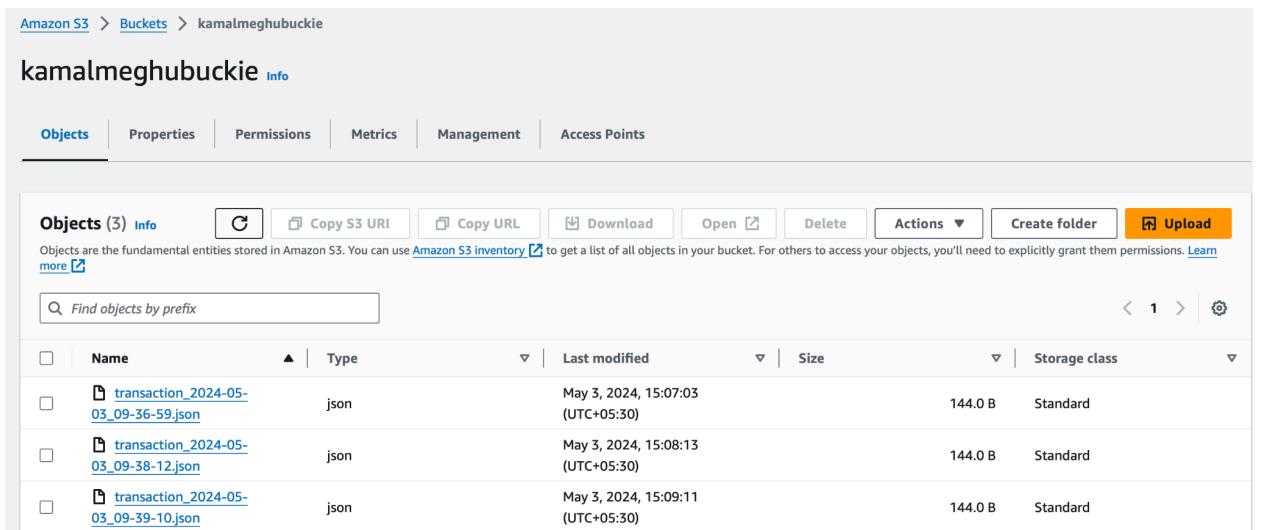
kamalmeghubuckie [Info](#)

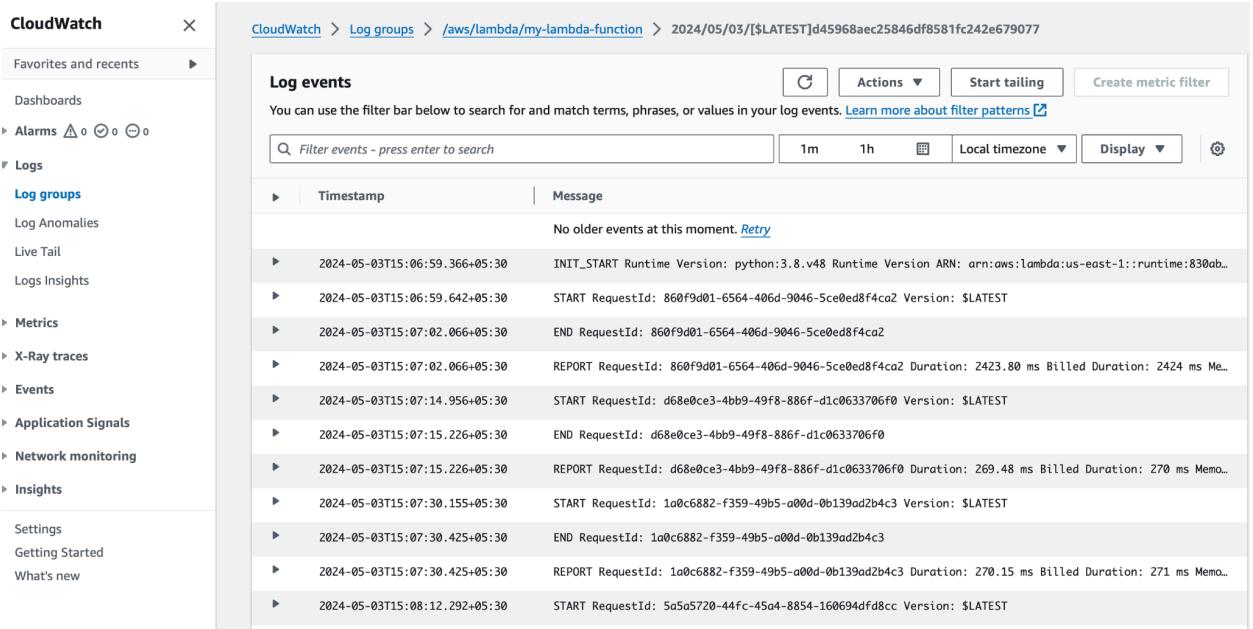
Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (3) [Info](#) | [C](#) | [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](#)

Name	Type	Last modified	Size	Storage class
 transaction_2024-05-03_09-36-59.json	json	May 3, 2024, 15:07:03 (UTC+05:30)	144.0 B	Standard
 transaction_2024-05-03_09-38-12.json	json	May 3, 2024, 15:08:13 (UTC+05:30)	144.0 B	Standard
 transaction_2024-05-03_09-39-10.json	json	May 3, 2024, 15:09:11 (UTC+05:30)	144.0 B	Standard





//cloudwatch logs are being generated

- b) Create a POST API from API Gateway, pass parameters as request body to Lambda and Consuming API from local machine and pass unique data to lambda.

```
meghasingh@Megha-ka-MacBook-Air aws % aws apigateway create-rest-api --name meghuapi --region us-east-1
{
  "id": "4duzbbq320",
  "name": "meghuapi",
  "createdDate": "2024-05-04T21:06:39+05:30",
  "apiKeySource": "HEADER",
  "endpointConfiguration": {
    "types": [
      "EDGE"
    ]
  },
  "disableExecuteApiEndpoint": false,
  "rootResourceId": "wqrckur1f1"
}
meghasingh@Megha-ka-MacBook-Air aws % aws apigateway get-resources --rest-api-id 4duzbbq320 --region us-east-1
{
  "items": [
    {
      "id": "wqrckur1f1",
      "path": "/"
    }
  ]
}
meghasingh@Megha-ka-MacBook-Air aws % aws apigateway create-resource --rest-api-id 4duzbbq320 --parent-id wqrckur1f1 --path-part files --region us-east-1
{
  "id": "ylrigt",
  "parentId": "wqrckur1f1",
  "pathPart": "files",
  "path": "/files"
}
```

- Created the API , connect the API with lambda .
- 3.c Consume the API from local machine and pass unique data to lambda.

```

meghasingh@Megha-MacBook-Air aws % aws apigateway put-method --rest-api-id 4duzbbq320 --resource-id wqrckur1f1 --http-method POST --authorization-type NONE --region us-east-1
{
    "httpMethod": "POST",
    "authorizationType": "NONE",
    "apiKeyRequired": false
}
meghasingh@Megha-MacBook-Air aws % aws apigateway put-integration --rest-api-id 4duzbbq320 --resource-id wqrckur1f1 --http-method POST --type AWS_PROXY --integration-http-method POST --uri 'arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/arn:aws:lambda:us-east-1:891377307930:function:MyLambdaFunction/invocations' --region us-east-1
{
    "type": "AWS_PROXY",
    "httpMethod": "POST",
    "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/arn:aws:lambda:us-east-1:891377307930:function:MyLambdaFunction/invocations",
    "passthroughBehavior": "WHEN_NO_MATCH",
    "timeoutInMillis": 29000,
    "cacheNamespace": "wqrckur1f1",
    "cacheKeyParameters": []
}
meghasingh@Megha-MacBook-Air aws % aws apigateway create-deployment --rest-api-id 4duzbbq320 --stage-name prod --region us-east-1
{
    "id": "jrt0f7",
    "createdDate": "2024-05-04T21:13:14+05:30"
}
meghasingh@Megha-MacBook-Air aws % curl -X POST 'https://4duzbbq320.execute-api.us-east-1.amazonaws.com/prod/files' \
-H 'Content-Type: application/json' \
-d '{"transaction_id": 12345, "payment_mode": "card/netbanking/upi", "amount": 200.0, "customer_id": 101}'

```

3. d Check if cloud watch logs are generated.

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	U5-U5_10-09-52.json		(UTC+05:30)		
<input type="checkbox"/>	transaction_2024-05-03_10-10-41.json	json	May 3, 2024, 15:40:42 (UTC+05:30)	144.0 B	Standard
<input type="checkbox"/>	transaction_2024-05-03_10-11-44.json	json	May 3, 2024, 15:41:45 (UTC+05:30)	144.0 B	Standard
<input type="checkbox"/>	transaction_2024-05-03T11:01:28.269653.json	json	May 3, 2024, 16:31:31 (UTC+05:30)	144.0 B	Standard
<input type="checkbox"/>	transaction_2024-05-03T11:56:20.043686.json	json	May 3, 2024, 17:26:23 (UTC+05:30)	144.0 B	Standard

Log events			Actions ▾	Start tailing	Create metric filter
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns					
<input type="text"/> Filter events - press enter to search		1m	1h	Local timezone ▾	Display ▾
▶	Timestamp	Message			
		No older events at this moment. Retry			
▶	2024-05-03T17:20:10.261+05:30	INIT_START Runtime Version: python:3.8.v48 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:830ab...			
▶	2024-05-03T17:20:10.569+05:30	START RequestId: fe1eb085-1f4c-4fb1-ae12-e3fef2662123 Version: \$LATEST			
▶	2024-05-03T17:20:10.578+05:30	END RequestId: fe1eb085-1f4c-4fb1-ae12-e3fef2662123			
▶	2024-05-03T17:20:10.578+05:30	REPORT RequestId: fe1eb085-1f4c-4fb1-ae12-e3fef2662123 Duration: 8.42 ms Billed Duration: 9 ms Memory S...			
▶	2024-05-03T17:21:12.627+05:30	START RequestId: d48280d1-49ce-406f-ba71-05bdf260c702 Version: \$LATEST			
▶	2024-05-03T17:21:12.639+05:30	END RequestId: d48280d1-49ce-406f-ba71-05bdf260c702			
▶	2024-05-03T17:21:12.639+05:30	REPORT RequestId: d48280d1-49ce-406f-ba71-05bdf260c702 Duration: 12.49 ms Billed Duration: 13 ms Memory...			
▶	2024-05-03T17:21:27.065+05:30	START RequestId: 10a6647d-af64-420f-bfaa-6698791e2355 Version: \$LATEST			
▶	2024-05-03T17:21:27.067+05:30	END RequestId: 10a6647d-af64-420f-bfaa-6698791e2355			

// e) cloudwatch logs are generated

Log streams (5)			Delete	Create log stream	Search all log streams
<input type="text"/> Filter log streams or try prefix search					
<input type="checkbox"/>	Log stream	Last event time			
<input type="checkbox"/>	2024/05/03[\$LATEST]dd9cf13e1344ccf82e7399f636163a2	2024-05-03 17:31:22 (UTC+05:30)			
<input type="checkbox"/>	2024/05/03[\$LATEST]13e6f9d7e09c4525b2f42d7e42c3871b	2024-05-03 16:31:30 (UTC+05:30)			
<input type="checkbox"/>	2024/05/03[\$LATEST]3c0a23ea57c740649212d7764fc4dc61	2024-05-03 15:39:08 (UTC+05:30)			
<input type="checkbox"/>	2024/05/03[\$LATEST]11f6798e1b0345fa81f704b9eef53e85	2024-05-03 15:23:56 (UTC+05:30)			
<input type="checkbox"/>	2024/05/03[\$LATEST]2a7f7ea651804c16a5109fb9c507484c	2024-05-03 15:17:59 (UTC+05:30)			