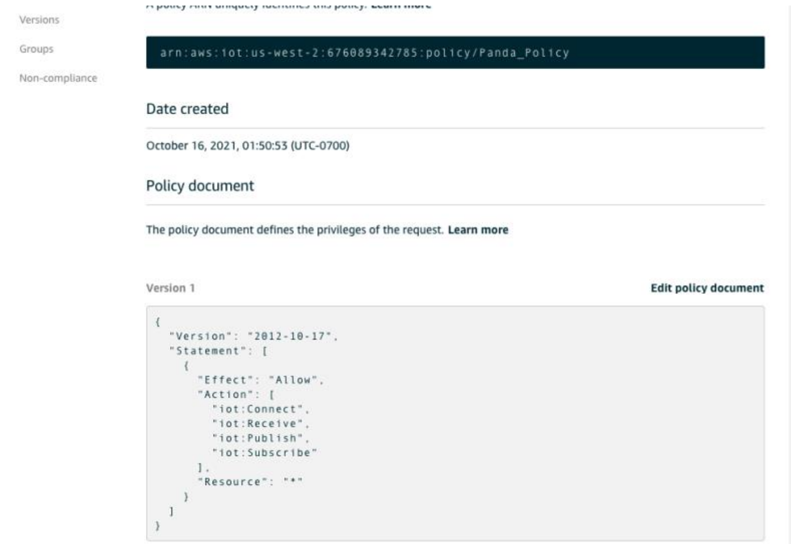
Homework – 1

Sayali Kudale

Part 1 : Configure laptop as IoT Device

1. Created aws account.
2. Created aws IoT policy and Thing object.
   a. While creating the policy in the action tab specified below actions: iot:Connect,iot:Receive,iot:Publish,iot:Subscribe.
   For above mentioned actions device will need permissions.
   b. Resource ARN is specified as *; to allow connection from nay device.
   c. Created a thing object and attached the above created policy to the thing object.
   d. After creating thing, downloaded certificated and keys. 4. Created the certs directory in the root folder and saved all the certificated which are saved while creating thing object and policy in aws iot console.



A policy ARN uniquely identifies this policy. Learn more

arn:aws:iot:us-west-2:676089342785:policy/Panda_Policy

Date created

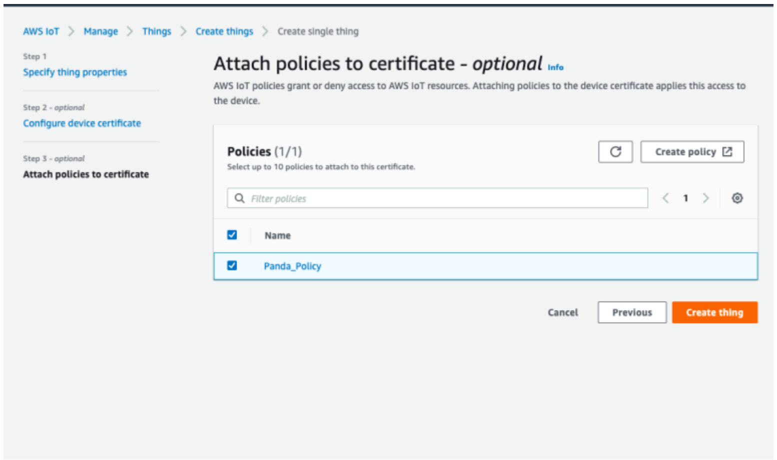
October 16, 2021, 01:50:53 (UTC-0700)

Policy document

The policy document defines the privileges of the request. **Learn more**

Version 1 **Edit policy document**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect",
        "iot:Receive",
        "iot:Publish",
        "iot:Subscribe"
      ],
      "Resource": "*"
    }
  ]
}
```


AWS IoT > Manage > Things > Create things > Create single thing
Step 1
Specify thing properties
Step 2 - optional
Configure device certificate
Step 3 - optional
Attach policies to certificate
Cancel
Previous
Create thing
Create policy

Attach policies to certificate - *optional* Info

AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.

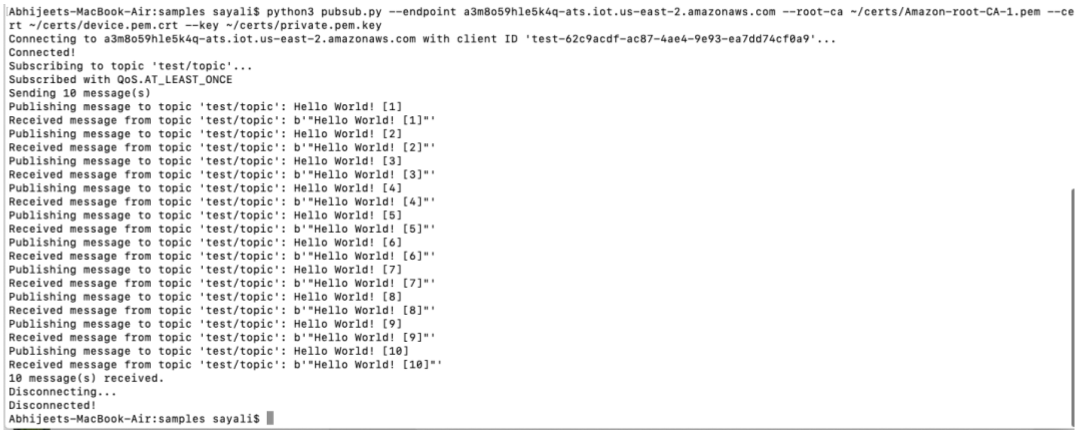
Policies (1/1)

Select up to 10 policies to attach to this certificate.

Filter policies

☑	Name
☑	Panda_Policy

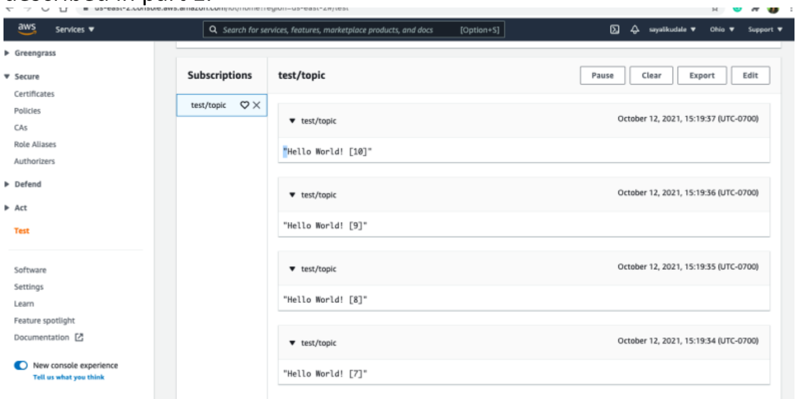
3. Verified git, and python setup and cloned aws-iot-device-sdk-python-v2 repository.
4. If we run the sample application, we are able to publish and receive messages from aws iot console.
5. Set up the policy and run the sample application



```
Abhijeets-MacBook-Air:samples sayali$ python3 pubsub.py --endpoint a3m8o59hle5k4q-ats.iot.us-east-2.amazonaws.com --root-ca ~/certs/Amazon-root-CA-1.pem --ce
rt ~/certs/device.pem.crt --key ~/certs/private.pem.key
Connecting to a3m8o59hle5k4q-ats.iot.us-east-2.amazonaws.com with client ID 'test-62c9acdf-ac87-4ae4-9e93-ea7dd74cf0a9'...
Connected!
Subscribing to topic 'test/topic'...
Subscribed with QoS.AT_LEAST_ONCE
Sending 10 message(s)
Publishing message to topic 'test/topic': Hello World! [1]
Received message from topic 'test/topic': b'"Hello World! [1]"'
Publishing message to topic 'test/topic': Hello World! [2]
Received message from topic 'test/topic': b'"Hello World! [2]"'
Publishing message to topic 'test/topic': Hello World! [3]
Received message from topic 'test/topic': b'"Hello World! [3]"'
Publishing message to topic 'test/topic': Hello World! [4]
Received message from topic 'test/topic': b'"Hello World! [4]"'
Publishing message to topic 'test/topic': Hello World! [5]
Received message from topic 'test/topic': b'"Hello World! [5]"'
Publishing message to topic 'test/topic': Hello World! [6]
Received message from topic 'test/topic': b'"Hello World! [6]"'
Publishing message to topic 'test/topic': Hello World! [7]
Received message from topic 'test/topic': b'"Hello World! [7]"'
Publishing message to topic 'test/topic': Hello World! [8]
Received message from topic 'test/topic': b'"Hello World! [8]"'
Publishing message to topic 'test/topic': Hello World! [9]
Received message from topic 'test/topic': b'"Hello World! [9]"'
Publishing message to topic 'test/topic': Hello World! [10]
Received message from topic 'test/topic': b'"Hello World! [10]"'
10 message(s) received.
Disconnecting...
Disconnected!
Abhijeets-MacBook-Air:samples sayali$ 
```

Part 2 : Configure laptop as IoT Device

In the aws iot console subscribed to the topic and we can see the published messages from sample application described in part 1.



test/topic

test/topic — October 12, 2021, 15:19:37 (UTC-0700)

```
"Hello World! [10]"
```

test/topic — October 12, 2021, 15:19:36 (UTC-0700)

```
"Hello World! [9]"
```

test/topic — October 12, 2021, 15:19:35 (UTC-0700)

```
"Hello World! [8]"
```

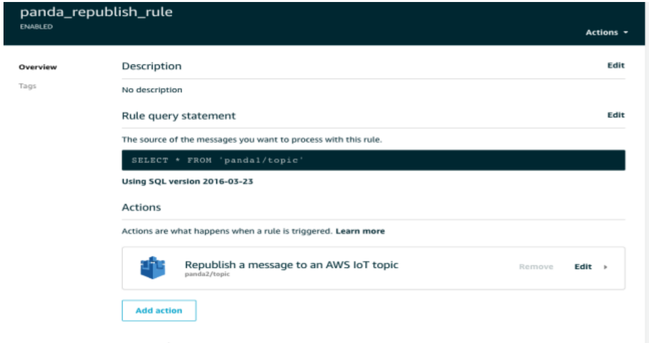
test/topic — October 12, 2021, 15:19:34 (UTC-0700)

```
"Hello World! [7]"
```

Part 3 : Republish feature of AWS IoT

In this part we need to configure AWS IoT to automatically respond to messages sent from the device and display those messages to the device and AWS IoT console also.

1. Create a rule in aws IoT to republish the messages.
   a. In this example we have created a rule aws IoT which will get the messages from one topic (panda1/ topic) and republish those messages to another topic (panda2/ topic).
   b. In the rule query statement, we will select the information from panda1/ topic.
   c. In rule action we will select republish a message to aws iot topic.
   d. In configure action we will select another topic: panda2/topic.
   e. With this we panda2/topic can receive all the messages published by panda1/topic.



panda_republish_rule
ENABLED

Description

No description

Rule query statement

The source of the messages you want to process with this rule.

```
SELECT * FROM 'panda1/topic'
```

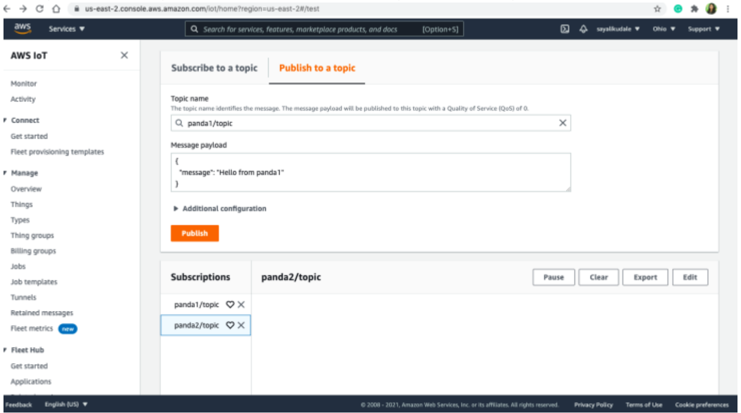
Using SQL version 2016-03-23

Actions

Actions are what happens when a rule is triggered. Learn more

Republish a message to an AWS IoT topic
panda2/topic

2. Test the aws IoT rule with two topics:
   a. Subscribe to the topic panda1/topic and panda2/topic.
   b. If we publish any message from panda1/topic in aws IoT console; those messages will be visible in the panda1/topic console as well as panda2/topic console.



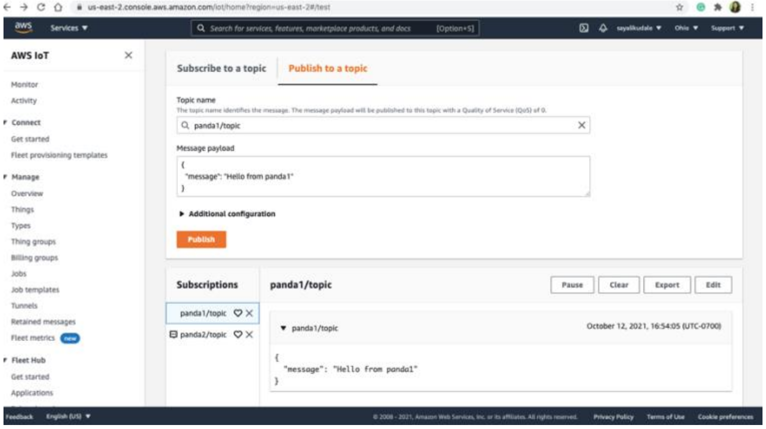
Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.
panda1/topic

Message payload

```
{
  "message": "Hello from panda1"
}
```

Subscriptions: panda1/topic, panda2/topic

panda2/topic



Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.
panda1/topic

Message payload

```
{
  "message": "Hello from panda1"
}
```

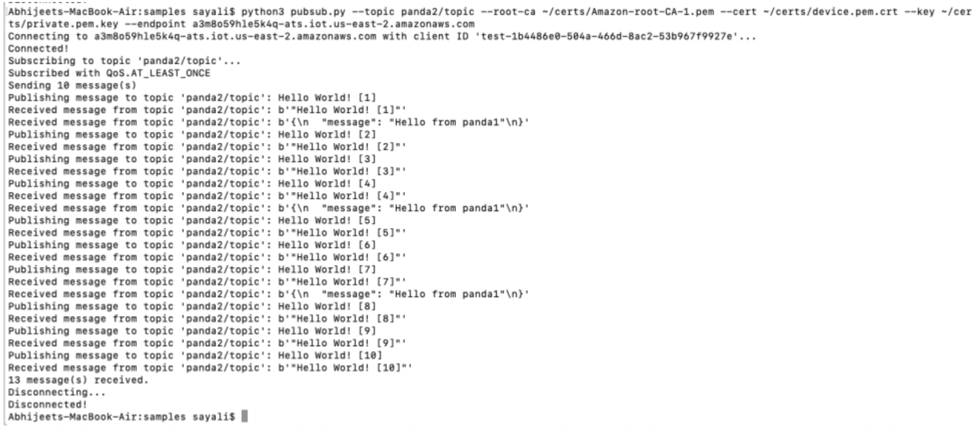
Subscriptions: panda1/topic, panda2/topic

panda1/topic

▼ panda1/topic October 12, 2021, 16:54:05 (UTC-0700)

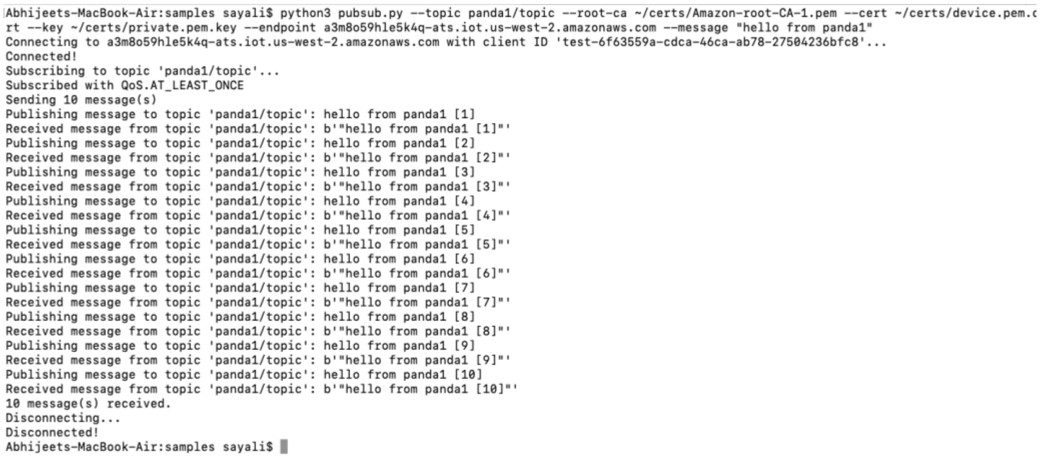
```
{
  "message": "Hello from panda1"
}
```

   c. In laptop subscribe to the panda2/topic by running the sample pubsub.py script. While this script is running if we publish the message from panda1/topic then that message will be visible in our laptop and also in aws console of panda2/topic.

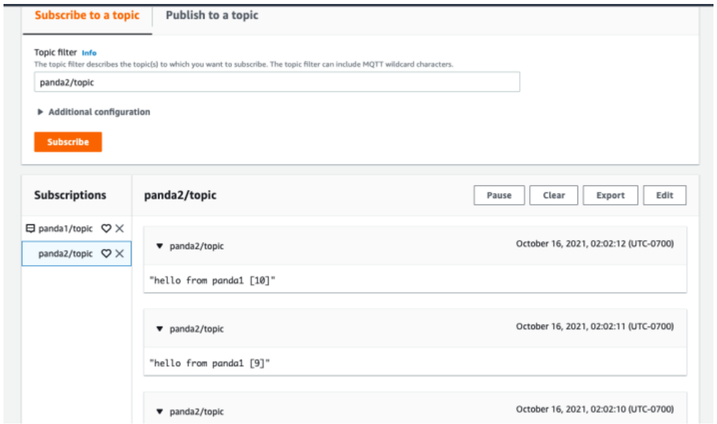


```
Abhijeets-MacBook-Air:samples sayali$ python3 pubsub.py --topic panda2/topic --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/cer
ts/private.pem.key --endpoint a3m8o59hle5k4q-ats.iot.us-east-2.amazonaws.com
Connecting to a3m8o59hle5k4q-ats.iot.us-east-2.amazonaws.com with client ID 'test-1b4486e0-504a-466d-8ac2-53b967f9927e'...
Connected!
Subscribing to topic 'panda2/topic'...
Subscribed with QoS.AT_LEAST_ONCE
Sending 10 message(s)
Publishing message to topic 'panda2/topic': Hello World! [1]
Received message from topic 'panda2/topic': b'"Hello World! [1]"'
Received message from topic 'panda2/topic': b'{\n  "message": "Hello from panda1"\n}'
Publishing message to topic 'panda2/topic': Hello World! [2]
Received message from topic 'panda2/topic': b'"Hello World! [2]"'
Publishing message to topic 'panda2/topic': Hello World! [3]
Received message from topic 'panda2/topic': b'"Hello World! [3]"'
Publishing message to topic 'panda2/topic': Hello World! [4]
Received message from topic 'panda2/topic': b'"Hello World! [4]"'
Received message from topic 'panda2/topic': b'{\n  "message": "Hello from panda1"\n}'
Publishing message to topic 'panda2/topic': Hello World! [5]
Received message from topic 'panda2/topic': b'"Hello World! [5]"'
Publishing message to topic 'panda2/topic': Hello World! [6]
Received message from topic 'panda2/topic': b'"Hello World! [6]"'
Publishing message to topic 'panda2/topic': Hello World! [7]
Received message from topic 'panda2/topic': b'"Hello World! [7]"'
Received message from topic 'panda2/topic': b'{\n  "message": "Hello from panda1"\n}'
Publishing message to topic 'panda2/topic': Hello World! [8]
Received message from topic 'panda2/topic': b'"Hello World! [8]"'
Publishing message to topic 'panda2/topic': Hello World! [9]
Received message from topic 'panda2/topic': b'"Hello World! [9]"'
Publishing message to topic 'panda2/topic': Hello World! [10]
Received message from topic 'panda2/topic': b'"Hello World! [10]"'
13 message(s) received.
Disconnecting...
Disconnected!
Abhijeets-MacBook-Air:samples sayali$ ▌
```

   d. All the messages sent by panda2/topic from laptop are visible in the aws console of panda2/topic.
   e. If we subscribe to the panda1/topic in laptop then all the messages are visible in the aws console of panda1/topic and panda2/topic.



```
Abhijeets-MacBook-Air:samples sayali$ python3 pubsub.py --topic panda1/topic --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.c
rt --key ~/certs/private.pem.key --endpoint a3m8o59hle5k4q-ats.iot.us-west-2.amazonaws.com --message "hello from panda1"
Connecting to a3m8o59hle5k4q-ats.iot.us-west-2.amazonaws.com with client ID 'test-6f63559a-cdca-46ca-ab78-27504236bfc8'...
Connected!
Subscribing to topic 'panda1/topic'...
Subscribed with QoS.AT_LEAST_ONCE
Sending 10 message(s)
Publishing message to topic 'panda1/topic': hello from panda1 [1]
Received message from topic 'panda1/topic': b'"hello from panda1 [1]"'
Publishing message to topic 'panda1/topic': hello from panda1 [2]
Received message from topic 'panda1/topic': b'"hello from panda1 [2]"'
Publishing message to topic 'panda1/topic': hello from panda1 [3]
Received message from topic 'panda1/topic': b'"hello from panda1 [3]"'
Publishing message to topic 'panda1/topic': hello from panda1 [4]
Received message from topic 'panda1/topic': b'"hello from panda1 [4]"'
Publishing message to topic 'panda1/topic': hello from panda1 [5]
Received message from topic 'panda1/topic': b'"hello from panda1 [5]"'
Publishing message to topic 'panda1/topic': hello from panda1 [6]
Received message from topic 'panda1/topic': b'"hello from panda1 [6]"'
Publishing message to topic 'panda1/topic': hello from panda1 [7]
Received message from topic 'panda1/topic': b'"hello from panda1 [7]"'
Publishing message to topic 'panda1/topic': hello from panda1 [8]
Received message from topic 'panda1/topic': b'"hello from panda1 [8]"'
Publishing message to topic 'panda1/topic': hello from panda1 [9]
Received message from topic 'panda1/topic': b'"hello from panda1 [9]"'
Publishing message to topic 'panda1/topic': hello from panda1 [10]
Received message from topic 'panda1/topic': b'"hello from panda1 [10]"'
10 message(s) received.
Disconnecting...
Disconnected!
Abhijeets-MacBook-Air:samples sayali$
```



Topic filter: The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

panda2/topic

Subscriptions: panda1/topic, panda2/topic

panda2/topic

panda2/topic October 16, 2021, 02:02:12 (UTC-0700)

"hello from panda1 [10]"

panda2/topic October 16, 2021, 02:02:11 (UTC-0700)

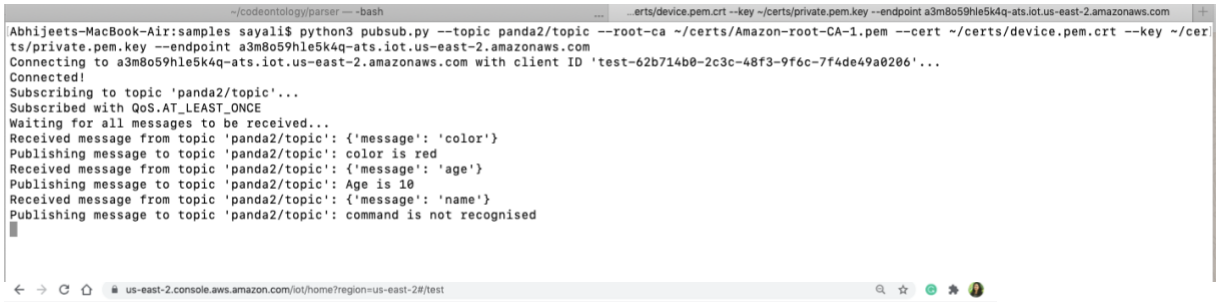
"hello from panda1 [9]"

panda2/topic October 16, 2021, 02:02:10 (UTC-0700)

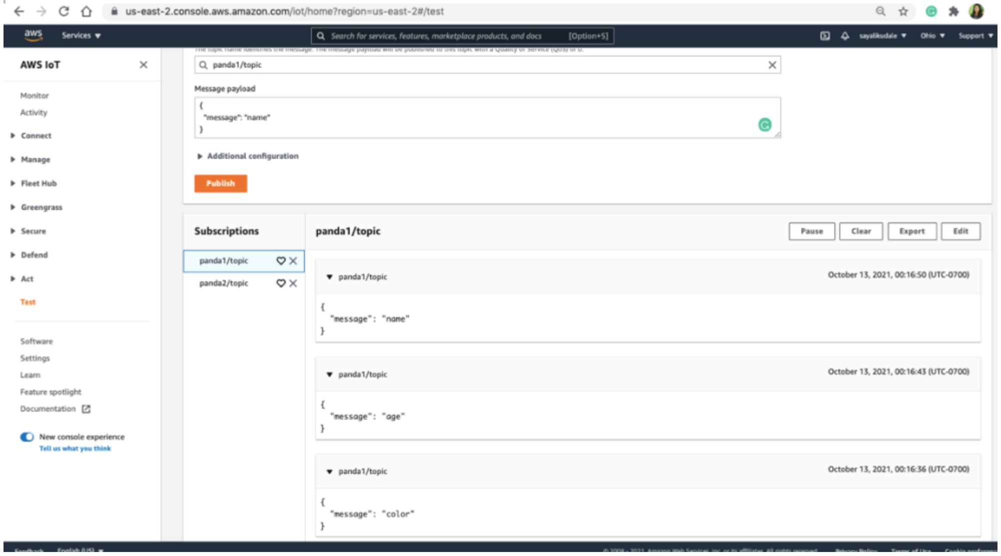
Part 4 : Configure AWS IoT to take user commands

1. In this part, user command will be sent from cloud IoT console from panda1/topic.
2. At device we are subscribing for the panda2/topic. As per the part 3, IoT rule has been created hence, messages sent from panda1/topic will be received by panda2/topic.
3. panda2/topic will receive the messages and parse those messages to determine the commands. Recognized user commands are color, age.
4. Device identifies that user command and process those user commands.
   a. If user command is color, panda2/topic replies as “color is red”
   b. If user command is age, panda2/topic replies as “Age is 10”
   c. If any other text, panda2/topic replies as “command is not recognized.”
5. panda2/topic will form the replies as mentioned above and republish those replies. Those replies and received messages from panda1/topic can be viewed in aws IoT console.
6. To perform above functionalities, we have modified the on_message_received function of sample pubsub.py file. (pubsub.py is part of the samples in aws-iot-device-sdk-python-v2)
   a. This function loads the payload as json and retrieves the message property.
   b. If message property is present, it means that message is from another topic. As panda2/topic does not publish messages in that format.
   c. If message property is not present, then we will not process that message.
   d. In the message property, we are checking is text is ‘color’, if yes then we draft reply with color as red.
   e. In the message property, we are checking is text is ‘age, if yes then we draft reply with age as 10.
   f. In the message property, we are checking is text is not age or color, then we draft reply as ‘not recognized user command.’
   g. After formatting the replies, we publish those using mqtt connection.

7. panda2/topic will only listen to the incoming messages and if valid command then republishes the replies. To do that default message argument is set to blank.
8. Code for above function is attached(pubsub.py).


~/codeontology/parser — -bash
...erts/device.pem.crt --key ~/certs/private.pem.key --endpoint a3m8o59hle5k4q-ats.iot.us-east-2.amazonaws.com

```
Abhijeets-MacBook-Air:samples sayali$ python3 pubsub.py --topic panda2/topic --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/cer
ts/private.pem.key --endpoint a3m8o59hle5k4q-ats.iot.us-east-2.amazonaws.com
Connecting to a3m8o59hle5k4q-ats.iot.us-east-2.amazonaws.com with client ID 'test-62b714b0-2c3c-48f3-9f6c-7f4de49a0206'...
Connected!
Subscribing to topic 'panda2/topic'...
Subscribed with QoS.AT_LEAST_ONCE
Waiting for all messages to be received...
Received message from topic 'panda2/topic': {'message': 'color'}
Publishing message to topic 'panda2/topic': color is red
Received message from topic 'panda2/topic': {'message': 'age'}
Publishing message to topic 'panda2/topic': Age is 10
Received message from topic 'panda2/topic': {'message': 'name'}
Publishing message to topic 'panda2/topic': command is not recognised
```



panda1/topic

Message payload

```
{
 "message": "name"
}
```

Subscriptions: panda1/topic, panda2/topic

panda1/topic

panda1/topic — October 13, 2021, 00:16:50 (UTC-0700)

```
{
  "message": "name"
}
```

panda1/topic — October 13, 2021, 00:16:43 (UTC-0700)

```
{
  "message": "age"
}
```

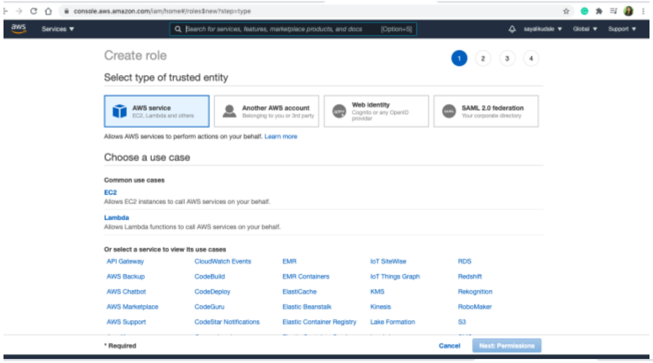
panda1/topic — October 13, 2021, 00:16:36 (UTC-0700)

```
{
  "message": "color"
}
```

Part 5: Connect AWS Lambda to your AWS IoT

1. creating IAM rule and policy
   a. As we need to save data to S3 we should have permissions to put s3 objects and some basic permission to execute lambda function. Hence, we will create a new role from IAM console.
   b. While creating new role, we will create a new policy to add S3 permissions and give a meaningful name as "Put_S3_policy".
   c. We will add Put_S3_policy and AWSLambdaBasicRuleExecution policies to the rule.
   d. S3 policy json data :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "*"
    }
  ]
}
```


console.aws.amazon.com/iam/home#/roles$new?step=type
aws
Services ▼
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale ▼
Global ▼
Support ▼
1 2 3 4
Cancel
Next: Permissions

Create role

Select type of trusted entity

- AWS service — EC2, Lambda and others
- Another AWS account — Belonging to you or 3rd party
- Web identity — Cognito or any OpenID provider
- SAML 2.0 federation — Your corporate directory

Allows AWS services to perform actions on your behalf. Learn more

Choose a use case

Common use cases

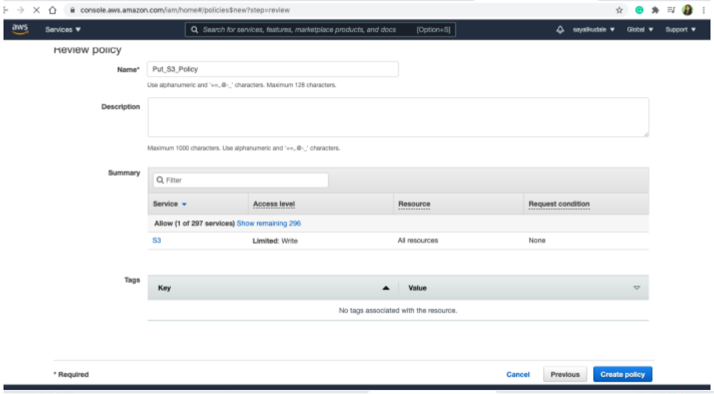
EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CloudWatch Events	EMR	IoT SiteWise	RDS
AWS Backup	CodeBuild	EMR Containers	IoT Things Graph	Redshift
AWS Chatbot	CodeDeploy	ElastiCache	KMS	Rekognition
AWS Marketplace	CodeGuru	Elastic Beanstalk	Kinesis	RoboMaker
AWS Support	CodeStar Notifications	Elastic Container Registry	Lake Formation	S3

* Required


console.aws.amazon.com/iam/home#/policies$new?step=review
aws
Services ▼
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale ▼
Global ▼
Support ▼
Filter
Cancel
Previous
Create policy

Review policy

Name* Put_S3_Policy

Use alphanumeric and '+=,.@-_' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

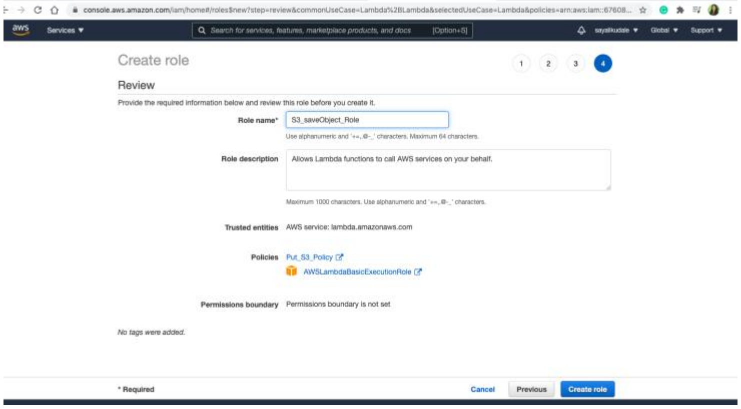
Summary

Service	Access level	Resource	Request condition
Allow (1 of 297 services) Show remaining 296			
S3	Limited: Write	All resources	None

Tags

Key	Value
No tags associated with the resource.	

* Required


console.aws.amazon.com/iam/home#/roles$new?step=review&commonUseCase=Lambda%2BLambda&selectedUseCase=Lambda&policies=arn:aws:iam::67608...
aws
Services ▼
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale ▼
Global ▼
Support ▼
1 2 3 4
Cancel
Previous
Create role

Create role

Review

Provide the required information below and review this role before you create it.

Role name* S3_saveObject_Role

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: lambda.amazonaws.com

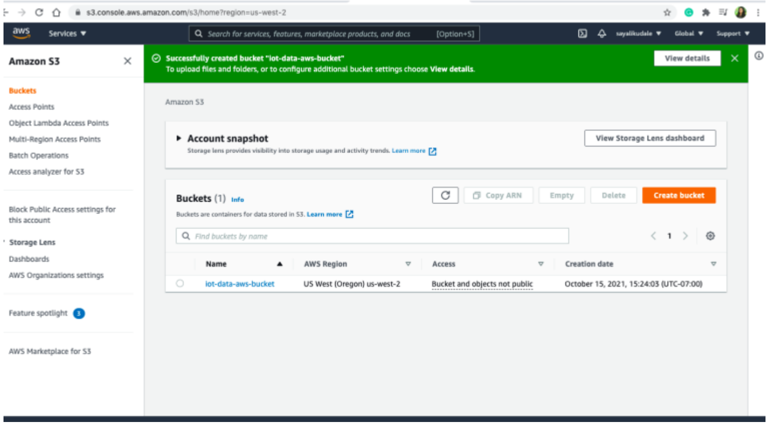
Policies Put_S3_Policy
AWSLambdaBasicExecutionRole

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

2. creating new Bucket:
   a. We will create a new bucket to store data from lambda function. We will name that bucket as ‘iot-data-aws-bucket’


s3.console.aws.amazon.com/s3/home?region=us-west-2
aws
Services ▼
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale ▼
Global ▼
Support ▼
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
View details
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1

Successfully created bucket "iot-data-aws-bucket"
To upload files and folders, or to configure additional bucket settings choose **View details**.

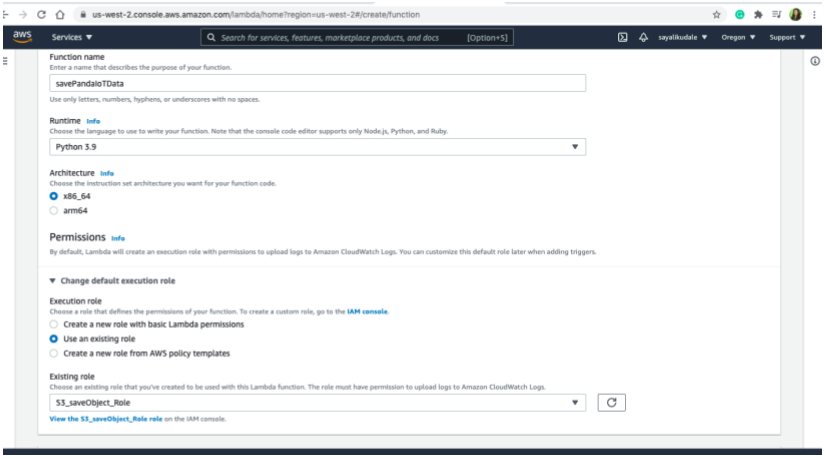
Amazon S3

Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
iot-data-aws-bucket	US West (Oregon) us-west-2	Bucket and objects not public	October 15, 2021, 15:24:03 (UTC-07:00)

3. Create a new lambda function:
   a. In this step we have created new lambda function and attached the new role created in step 1 (S3_saveObject_Role).



us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/create/function

Function name
Enter a name that describes the purpose of your function.
savePandaIoTData
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.
Python 3.9

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

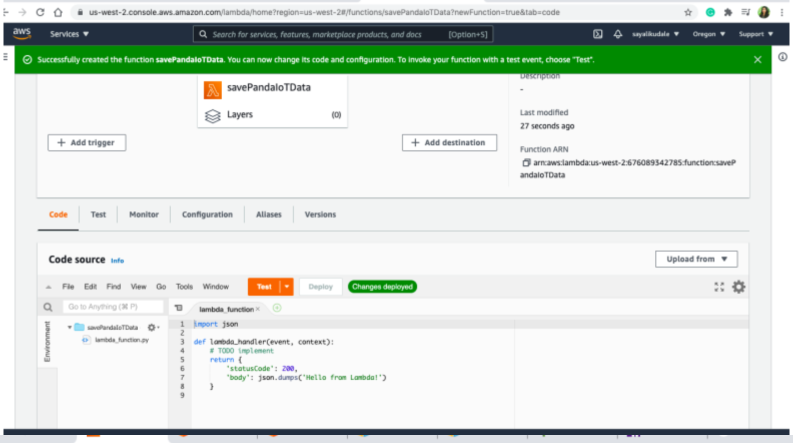
▼ Change default execution role

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.
S3_saveObject_Role
View the S3_saveObject_Role role on the IAM console.

aws
Services ▼
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale ▼
Oregon ▼
Support ▼

   b. Working of the Lambda function:
      i. This function will first save the event data (raw data) into the S3 bucket created in step 2 with file name as "raw_data.json".
      ii. This function will read the event data and processed that data and save into s3 bucket with file name as "processed_data.json"
      iii. While processing the data, this function first reads the age and weight parameter from the event data, and based on the parameter values, it decides whether panda is baby or adult and also categorize it as normal or giant panda.
      iv. This function then reads color, age, panda, babyOrAdult, category convert into readable string format. This readable string format then saved into the S3 bucket.



us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/savePandaIoTData?newFunction=true&tab=code

Successfully created the function savePandaIoTData. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

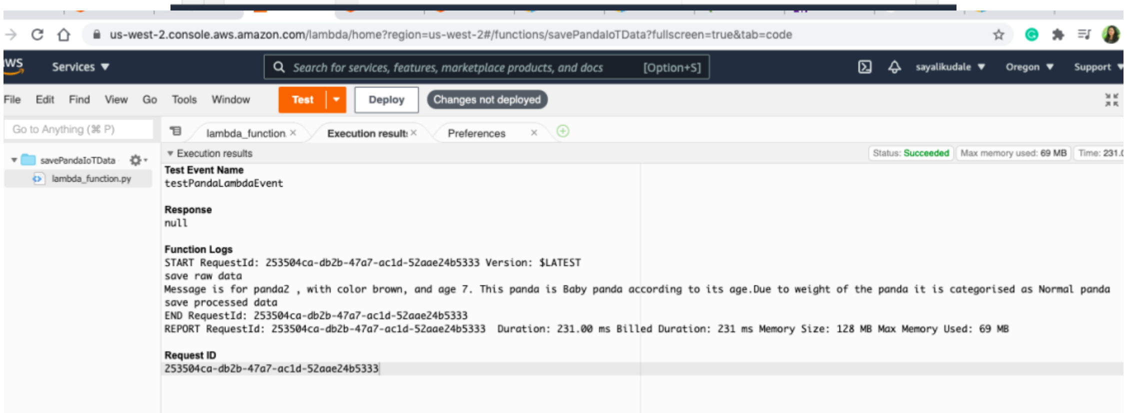
savePandaIoTData
Layers (0)

Description
-
Last modified
27 seconds ago
Function ARN
arn:aws:lambda:us-west-2:676089342785:function:savePandaIoTData

Code source Info

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```



us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/savePandaIoTData?fullscreen=true&tab=code

▼ Execution results

Status: Succeeded | Max memory used: 69 MB | Time: 231.0

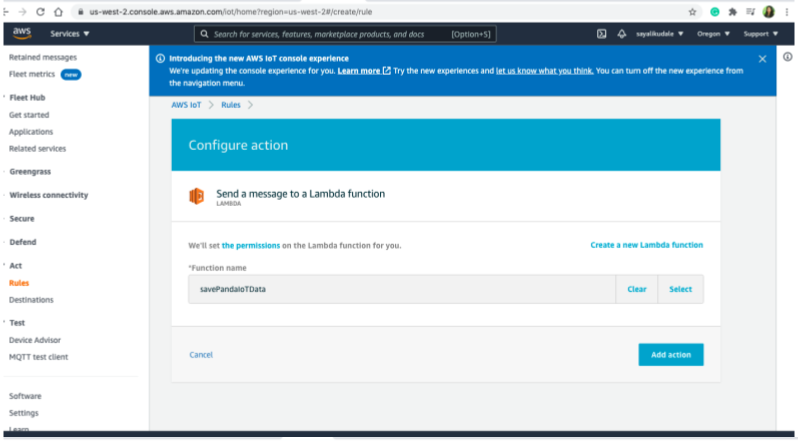
```
Test Event Name
testPandaLambdaEvent

Response
null

Function Logs
START RequestId: 253504ca-db2b-47a7-ac1d-52aae24b5333 Version: $LATEST
save raw data
Message is for panda2 , with color brown, and age 7. This panda is Baby panda according to its age.Due to weight of the panda it is categorised as Normal panda
save processed data
END RequestId: 253504ca-db2b-47a7-ac1d-52aae24b5333
REPORT RequestId: 253504ca-db2b-47a7-ac1d-52aae24b5333  Duration: 231.00 ms Billed Duration: 231 ms Memory Size: 128 MB Max Memory Used: 69 MB

Request ID
253504ca-db2b-47a7-ac1d-52aae24b5333
```

4. Creating IoT rule to send data to lambda function:
    a. In this step, we have created a IoT rule, which will be automatically sends the IoT data to the Lambda Function.
    b. To do that, while creating rule we have configured the Lambda function which is created in step 3 (savePandaIoTData).



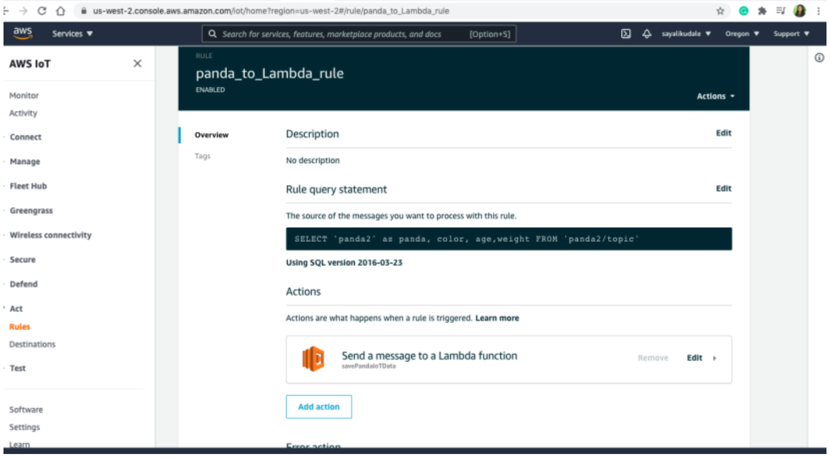
Introducing the new AWS IoT console experience
We're updating the console experience for you. Learn more Try the new experiences and let us know what you think. You can turn off the new experience from the navigation menu.

Configure action

Send a message to a Lambda function
LAMBDA

We'll set the permissions on the Lambda function for you. Create a new Lambda function

*Function name: savePandaIoTData



panda_to_Lambda_rule
ENABLED

Description
No description

Rule query statement
The source of the messages you want to process with this rule.

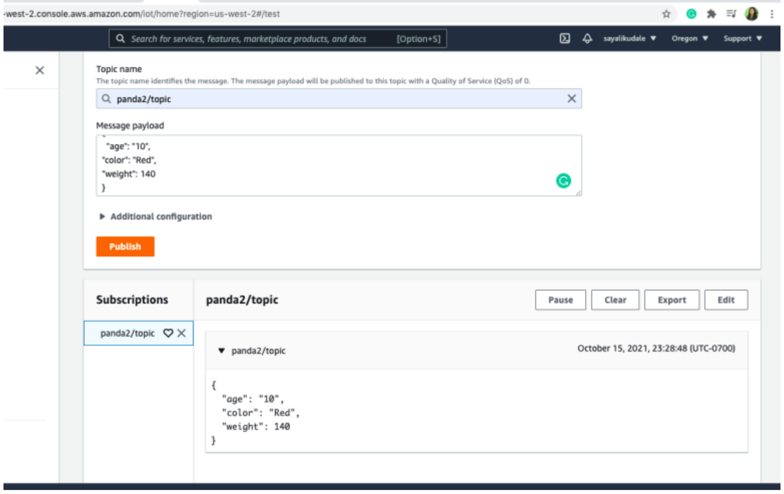
```
SELECT 'panda2' as panda, color, age,weight FROM 'panda2/topic'
```

Using SQL version 2016-03-23

Actions
Actions are what happens when a rule is triggered. Learn more

Send a message to a Lambda function
savePandaIoTData

5. Test the Lambda function
    a. As configuration of aws iot and lambda function is done in earlier steps. We can test the working by publishing data from topic and check the files in S3 bucket.


-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/test
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale
Oregon
Support
Additional configuration
Publish
Subscriptions
Pause
Clear
Export
Edit

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.
panda2/topic

Message payload

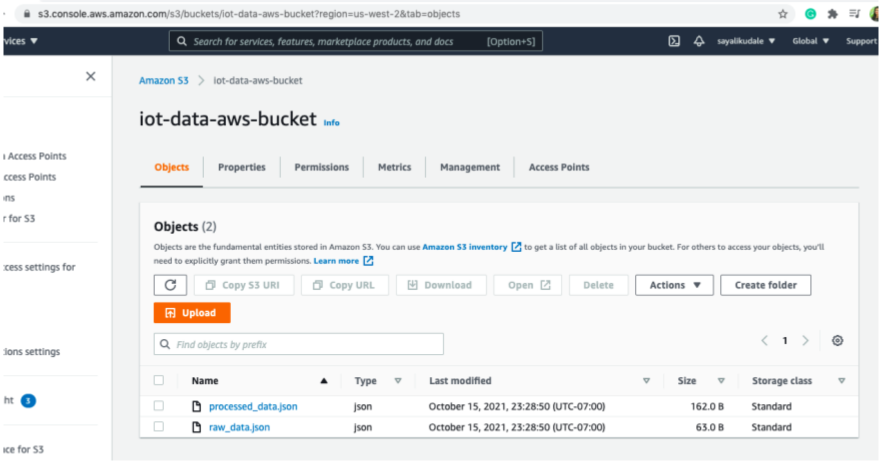
```
 "age": "10",
"color": "Red",
"weight": 140
}
```

panda2/topic

panda2/topic

▼ panda2/topic   October 15, 2021, 23:28:48 (UTC-0700)

```
{
  "age": "10",
  "color": "Red",
  "weight": 140
}
```

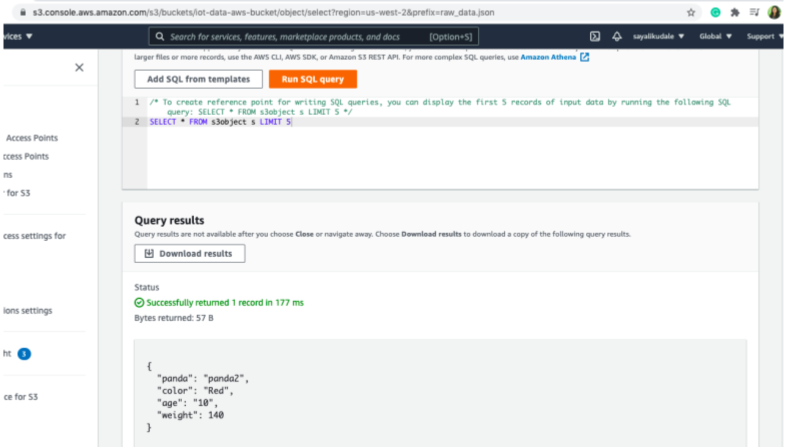

s3.console.aws.amazon.com/s3/buckets/iot-data-aws-bucket?region=us-west-2&tab=objects
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale
Global
Support
Amazon S3 > iot-data-aws-bucket
Objects
Properties
Permissions
Metrics
Management
Access Points
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix

iot-data-aws-bucket

Objects (2)

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
processed_data.json	json	October 15, 2021, 23:28:50 (UTC-07:00)	162.0 B	Standard
raw_data.json	json	October 15, 2021, 23:28:50 (UTC-07:00)	63.0 B	Standard


s3.console.aws.amazon.com/s3/buckets/iot-data-aws-bucket/object/select?region=us-west-2&prefix=raw_data.json
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale
Global
Support
Add SQL from templates
Run SQL query
Download results

larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```
/* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL
    query: SELECT * FROM s3object s LIMIT 5 */
SELECT * FROM s3object s LIMIT 5
```

Query results

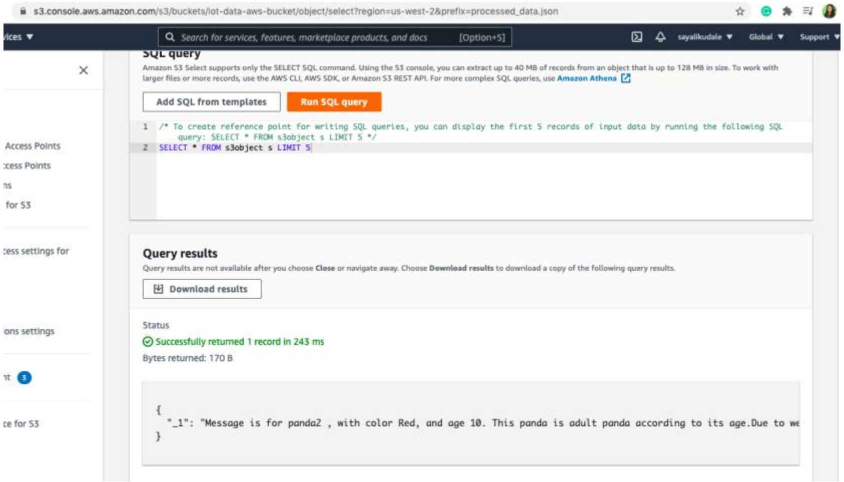
Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Successfully returned 1 record in 177 ms

Bytes returned: 57 B

```
{
  "panda": "panda2",
  "color": "Red",
  "age": "10",
  "weight": 140
}
```


s3.console.aws.amazon.com/s3/buckets/iot-data-aws-bucket/object/select?region=us-west-2&prefix=processed_data.json
Search for services, features, marketplace products, and docs [Option+S]
sayalikudale
Global
Support
Add SQL from templates
Run SQL query
Download results

SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```
/* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL
    query: SELECT * FROM s3object s LIMIT 5 */
SELECT * FROM s3object s LIMIT 5
```

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Successfully returned 1 record in 243 ms

Bytes returned: 170 B

```
{
  "_1": "Message is for panda2 , with color Red, and age 10. This panda is adult panda according to its age.Due to we
}
```

Encountered problems when working on this assignment

1. Tried running the sample pubsub.py script after deleting the thing object; but was getting invalid certificate error; had to create thing and policies again and download new certificates.
2. If changes are done in thing object; had to reattach the policies as well.
3. Due Incorrect policy setup was not able to access the messages.

Time spent on this assignment : 10-12 hrs