

Smart Pet Feeder with Facial Recognition Control Cloud Configuration

1. Configure AWS IoT Core:

- Create an IoT Thing
 - Log in to the AWS Management Console and navigate to AWS IoT Core.
 - Create a new IoT Thing and download the device certificates (root CA, private key, and certificate).
 - Copy the root CA, private key, and certificate files to the Raspberry Pi.
- Set Up a Policy: Attach an IoT policy to allow device communication.

Active version: 5 Info			Builder	JSON
Policy effect	Policy action	Policy resource		
Allow	iot:Connect	*		
Allow	iot:Publish	*		
Allow	iot:Subscribe	*		
Allow	iot:Receive	*		

- Policy for the IAM user as follows: (Added throughout the course of the project)

		Filter by Type	
<input type="text" value="Search"/>		All types	
<input type="checkbox"/>	Policy name ↗		Type
<input type="checkbox"/>	+ AdministratorAccess		AWS managed - job function
<input type="checkbox"/>	+ AmazonEC2FullAccess		AWS managed
<input type="checkbox"/>	+ AmazonS3FullAccess		AWS managed
<input type="checkbox"/>	+ AmazonS3ObjectLambdaExecutionRolePolicy		AWS managed
<input type="checkbox"/>	+ AWSIoTDataAccess		AWS managed
<input type="checkbox"/>	+ AWSIoTFullAccess		AWS managed
<input type="checkbox"/>	+ AWSLambda_FullAccess		AWS managed
<input type="checkbox"/>	+ AWSLambdaBasicExecutionRole-36350f34-596...		Customer managed
<input type="checkbox"/>	+ AWSOrganizationsReadOnlyAccess		AWS managed
<input type="checkbox"/>	+ IAMFullAccess		AWS managed
<input type="checkbox"/>	+ IAMUserChangePassword		AWS managed

2. Configure S3 buckets:

- Create a Bucket

- In the AWS Management Console, go to S3 and create a bucket (*petimagestorage* in this case) in the same region as IoT Core setup.
- Attach the **AmazonS3FullAccess** policy
- **Event notifications**
 - Send a notification when specific events occur in the S3 bucket to the Lambda function

Event notifications (1) Edit Delete Create event notification

Send a notification when specific events occur in your bucket. [Learn more](#)

<input type="checkbox"/>	Name	Event types	Filters	Destination type	Destination
<input type="checkbox"/>	smartpetdispenser	All object create events	, .jpg	Lambda function	smartpetfooddispenser

Amazon EventBridge Edit

For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. [Learn more](#) or [see EventBridge pricing](#)

Send notifications to Amazon EventBridge for all events in this bucket
Off

- **Upload machine learning model**
 - Upload the model into S3 for ease of access from Lambda function

petimagestorage [Info](#)

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

Objects (2) [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

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	a6c58743-0ff3-47ab-b13c-e9b503ab50a6.jpg	jpg	December 3, 2024, 01:48:02 (UTC-08:00)	1.6 MB	Standard
<input type="checkbox"/>	converted_tflite/	Folder	-	-	-

3. Configure AWS DynamoDB

- **Create DynamoDB Table**
 - Go to Amazon DynamoDB in the AWS Console. Create Table. Set a table name (*FeedingTimes* in this case).
 - Set the primary key as PetID (String) and optionally add LastFedTime (Number) to track the feeding times. LastFedTime will be initialized in the table anyway.
 - Attach **AmazonDynamoDBFullAccess** to the IAM policy or you could also attach the policy to the lambda role.

FeedingTimes

▼ Scan or query items

☒ Scan

☐ Query

Select a table or index

Table - FeedingTimes ▼

Select attribute proje

All attributes

► Filters

Run

Reset

✓ Completed. Read capacity units consumed: 2

Items returned (3)

<input type="checkbox"/>	PetID (String) ▼	LastFedTime
<input type="checkbox"/>	Pet2	2024-12-03T08:16:03.816953
<input type="checkbox"/>	Pet1	2024-12-03T04:16:00.445038
<input type="checkbox"/>	NoPet	2024-12-03T04:16:00.478391

4. Configure AWS Lambda


- Navigate to AWS Lambda in the console and create a new Lambda function (*smartpetfooddispenser* in this case). And write the `lambda_handler` code.
- Trigger Lambda from IoT Core: Set up an IoT Rule in AWS IoT Core to trigger the Lambda function based on a message received in the IoT topic.
- An S3 trigger is also added when setting the event notification from the previous step.


Triggers (2) [Info](#)



[Fix errors](#)

☐ Trigger

☐  **AWS IoT: PetDispenserProcessingRule**
arn:aws:iot:us-east-1:241533150659:rule/PetDispenserProcessingRule
[Details](#)

☐  **S3: petimagestorage**
arn:aws:s3:::petimagestorage
[Details](#)

- Add the following layers to resolve dependencies issues.
 - AWSSDKPandas-Python39
 - Add layer → AWS layer → AWSSDKPandas-Python39
 - Note: Make sure the python version is Python 3.9
 - pillow-layer
 - tf-lite-runtime
 - Create a layer from AWS CLI and attach the layer to the lambda function.

Layers [Info](#)

[Edit](#)

[Add a layer](#)




Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	pillow-layer	1	python3.9	-	arn:aws:lambda:us-east-1:241533150659:layer:pillow-layer:1
2	AWSSDKPandas-Python39	26	python3.9	x86_64	arn:aws:lambda:us-east-1:336392948345:layer:AWSSDKPandas-Python39:26
3	tf-lite-runtime-new	1	python3.9	-	arn:aws:lambda:us-east-1:241533150659:layer:tf-lite-runtime-new:1

- Set an appropriate execution role for the Lambda function that grants permissions to interact with DynamoDB, S3, and IoT Core.

Permissions policies (4) [Info](#)

You can attach up to 10 managed policies.

Filter by Type: All types

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	 AmazonDynamoDBFullAccess	AWS managed	1
<input type="checkbox"/>	 AmazonS3FullAccess	AWS managed	2
<input type="checkbox"/>	 AWSLambdaBasicExecutionRole-36350F34-596...	Customer managed	2

- Add the following permissions to the Lambda role

in
ecutionRole-
0-9cab-

Modify permissions in AWSLambdaBasicExecutionRole-36350f34-596a-4ba0-9cab-c1426b96b3c2 [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "logs:CreateLogGroup",
7       "Resource": "arn:aws:logs:us-east-1:241533150659:*"
8     },
9     {
10      "Effect": "Allow",
11      "Action": [
12        "logs:CreateLogStream",
13        "logs:PutLogEvents"
14      ],
15      "Resource": [
16        "arn:aws:logs:us-east-1:241533150659:log-group:/aws/lambda/smartpetfooddispenser:*"
17      ]
18    },
19    {
20      "Effect": "Allow",
21      "Action": "iot:*",
22      "Resource": "*"
23    }
24  ]
25 }
```

5. Configure AWS CloudWatch Monitoring

- Set up CloudWatch to monitor logs for the Lambda functions, MQTT messages, or device status.

Overview of the Cloud Configuration

- IoT Core: Device securely connects using certificates and MQTT, sending data to topics like pet/dispenser/image.
- S3: Device uploads images captured by the Raspberry Pi camera to S3.
- DynamoDB: Stores the recognition data, i.e. Pet ID and last fed time
- Lambda: Processes data from IoT Core and updates DynamoDB and publish messages back to Raspberry pi device
- IAM Roles and Policies: Secure permissions for IoT Core, S3, DynamoDB, and Lambda.