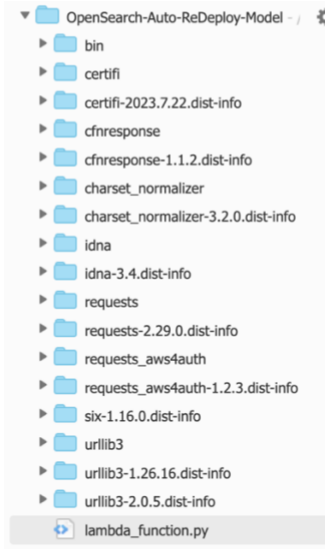
README.pdf

## 1. Purpose:

This is a lambda helpful function to help the customers to conduct auto-deploy model when the models are undeployed in a node, for example, when adding a new node and the model is not deployed to the new node yet. This helpful lambda function can be added with a trigger to run auto deployment in a schedule.

## 2. About the zip file:

In the zip file, please note that the lambda_function.py is the main file to run in the lambda job, the other folders are imported packages. Those are dependencies for the lambda_function.py to run successfully.



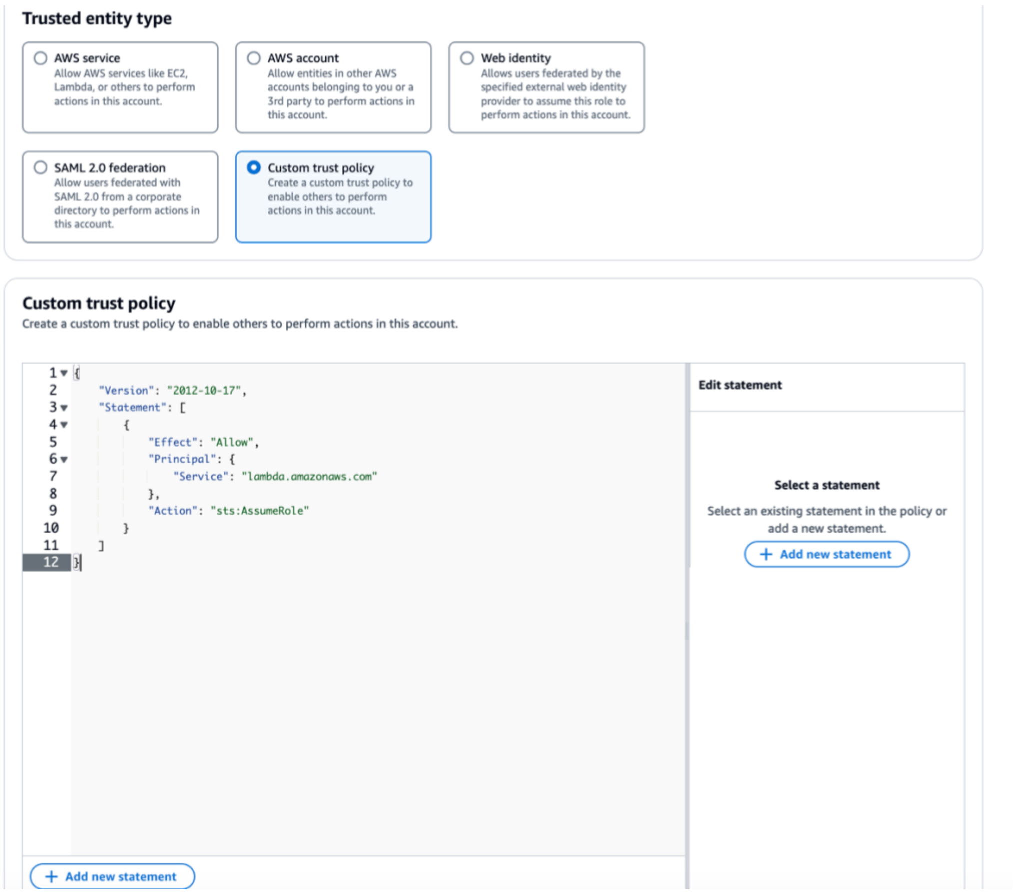
```
OpenSearch-Auto-ReDeploy-Model
  bin
  certifi
  certifi-2023.7.22.dist-info
  cfnresponse
  cfnresponse-1.1.2.dist-info
  charset_normalizer
  charset_normalizer-3.2.0.dist-info
  idna
  idna-3.4.dist-info
  requests
  requests-2.29.0.dist-info
  requests_aws4auth
  requests_aws4auth-1.2.3.dist-info
  six-1.16.0.dist-info
  urllib3
  urllib3-1.26.16.dist-info
  urllib3-2.0.5.dist-info
  lambda_function.py
```

## 3. Set-up Steps by Steps:

3.1 Create IAM role to give lambda access to OpenSearch

3.1.1 Use the following Custom trust policy to create an AWS IAM Role

```
{
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
 }
```


**Trusted entity type**

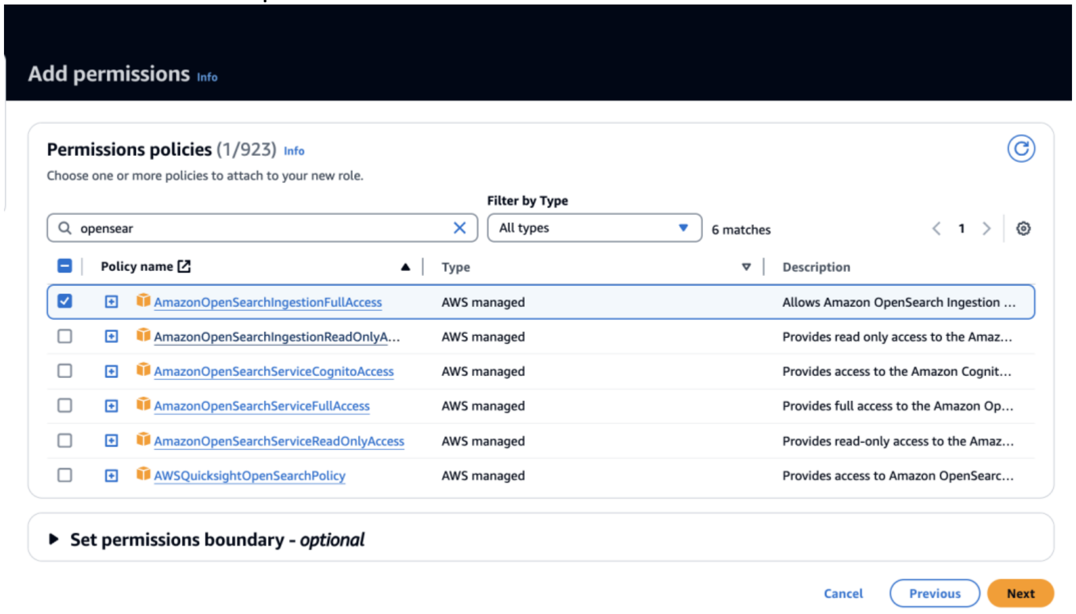
- ○ AWS service — Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ○ AWS account — Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ○ Web identity — Allows users federated by the specified external web identity 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.

**Custom trust policy**

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

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

## 3.1.2 Add AmazonOpenSearchServiceFullAcess

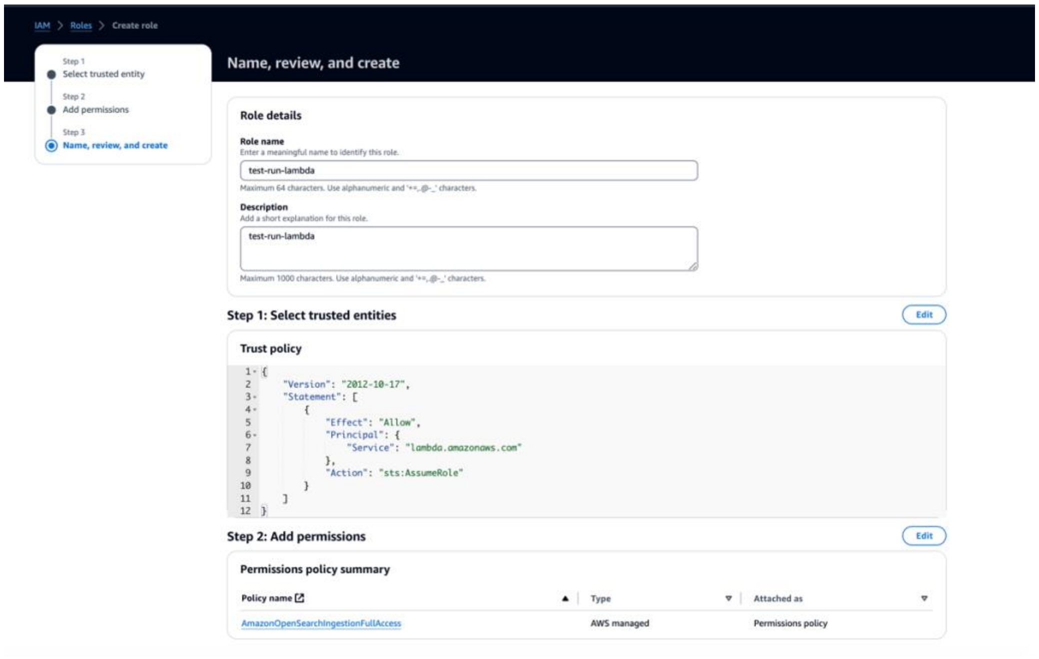

**Add permissions** Info

**Permissions policies** (1/923) Info

Choose one or more policies to attach to your new role.

Search: opensear — Filter by Type: All types — 6 matches — 1

| | Policy name | Type | Description |
|---|---|---|---|
| ☑ | AmazonOpenSearchIngestionFullAccess | AWS managed | Allows Amazon OpenSearch Ingestion ... |
| ☐ | AmazonOpenSearchIngestionReadOnlyA... | AWS managed | Provides read only access to the Amaz... |
| ☐ | AmazonOpenSearchServiceCognitoAccess | AWS managed | Provides access to the Amazon Cognit... |
| ☐ | AmazonOpenSearchServiceFullAccess | AWS managed | Provides full access to the Amazon Op... |
| ☐ | AmazonOpenSearchServiceReadOnlyAccess | AWS managed | Provides read-only access to the Amaz... |
| ☐ | AWSQuicksightOpenSearchPolicy | AWS managed | Provides access to Amazon OpenSearc... |


IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
test-run-lambda
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
test-run-lambda
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

Trust policy

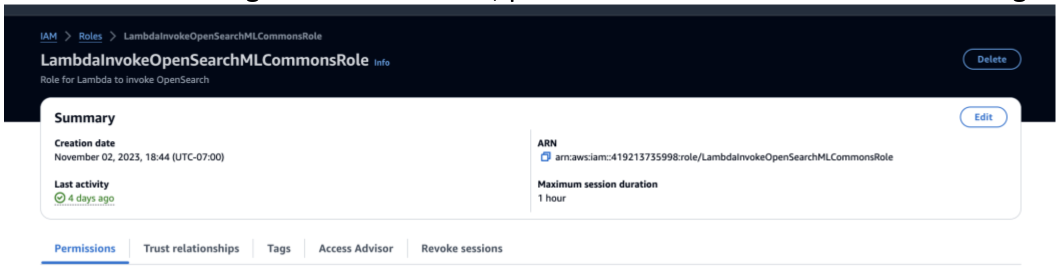
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Step 2: Add permissions

Permissions policy summary

| Policy name | Type | Attached as |
|---|---|---|
| AmazonOpenSearchIngestionFullAccess | AWS managed | Permissions policy |

3.1.3  After creating the new IAM role, please save the role ARN for later config.


IAM > Roles > LambdaInvokeOpenSearchMLCommonsRole

LambdaInvokeOpenSearchMLCommonsRole
Role for Lambda to invoke OpenSearch

Summary

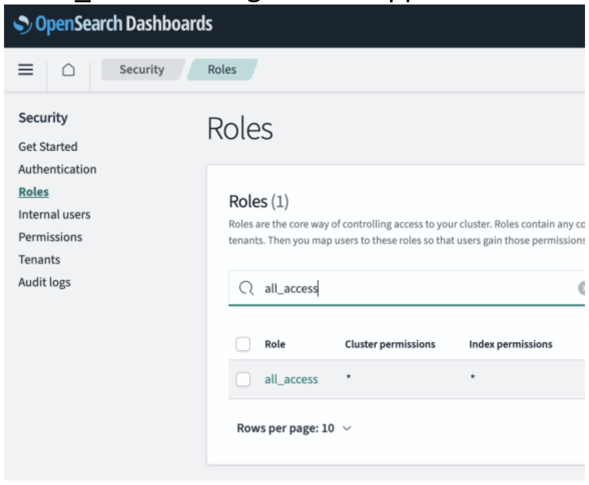
Creation date
November 02, 2023, 18:44 (UTC-07:00)

Last activity
4 days ago

ARN
arn:aws:iam::419213735998:role/LambdaInvokeOpenSearchMLCommonsRole

Maximum session duration
1 hour

3.1.4 Map role to backend role with all_access
Navigate to the OpenSearch Dashboard -> Security -> Roles, find all_access role, click on all_access. Navigate to Mapped users -> Managed Mappings



Roles

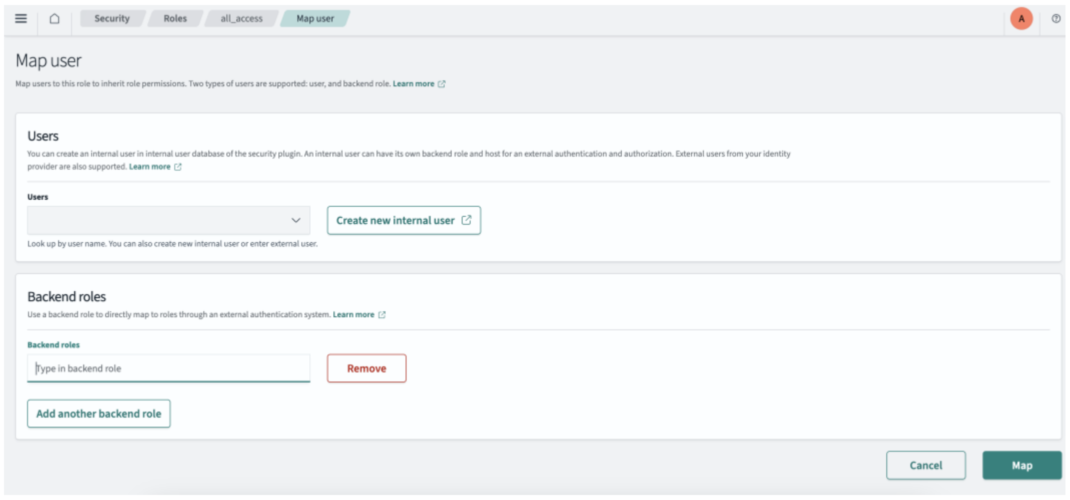
Roles (1)
Roles are the core way of controlling access to your cluster. Roles contain any co
tenants. Then you map users to these roles so that users gain those permissions

all_access

| Role | Cluster permissions | Index permissions |
|---|---|---|
| all_access | * | * |

Rows per page: 10

Map the admin role with the new IAM role created in 3.1.2 step.


Security / Roles / all_access / Map user

Map user

Map users to this role to inherit role permissions. Two types of users are supported: user, and backend role. Learn more

Users

You can create an internal user in internal user database of the security plugin. An internal user can have its own backend role and host for an external authentication and authorization. External users from your identity provider are also supported. Learn more

Users

Create new internal user

Look up by user name. You can also create new internal user or enter external user.

Backend roles

Use a backend role to directly map to roles through an external authentication system. Learn more

Backend roles

Type in backend role

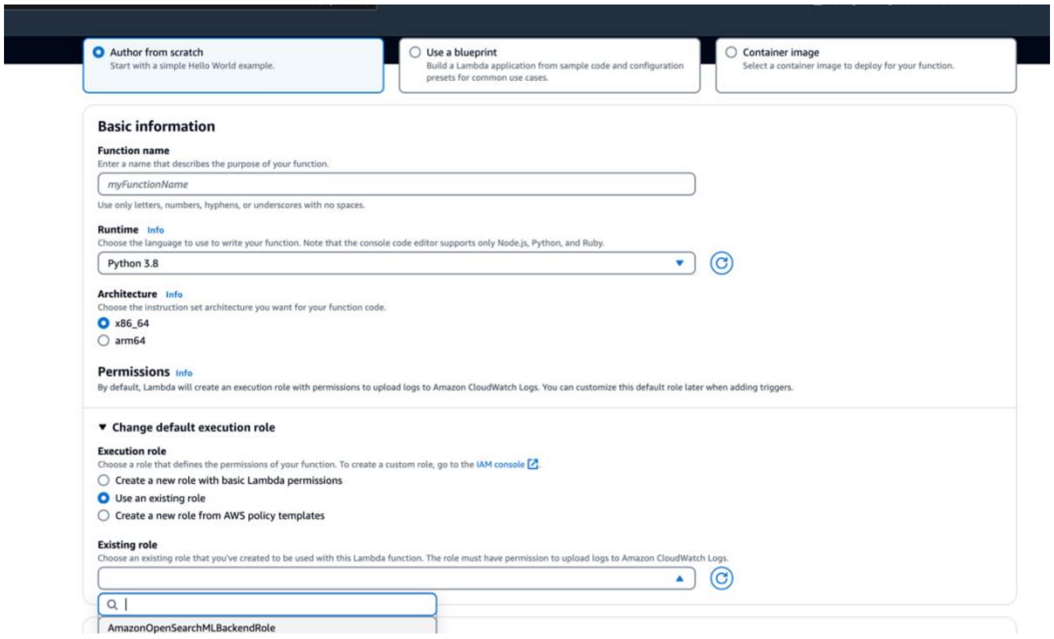
Remove

Add another backend role

Cancel Map

3.2 Create a new AWS [lambda](https://console.aws.amazon.com/lambda) function:

In `create function' config, choose RunTime as Python 3.8 and choose use existing role, click on the role name that you created previous in 3.1.2, leave the rest of the default setting, then click "Create Function"


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.

myFunctionName

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 3.8

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

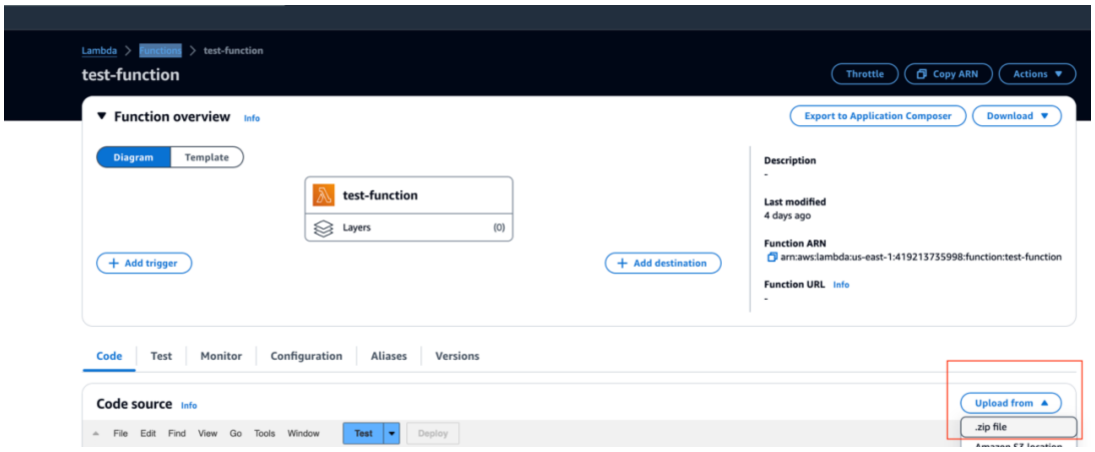
Existing role

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.

AmazonOpenSearchMLBackendRole

3.3 Upload the zipfile

On the new function page, click `Upload from` in the Code Tap, choose the provided zip file.

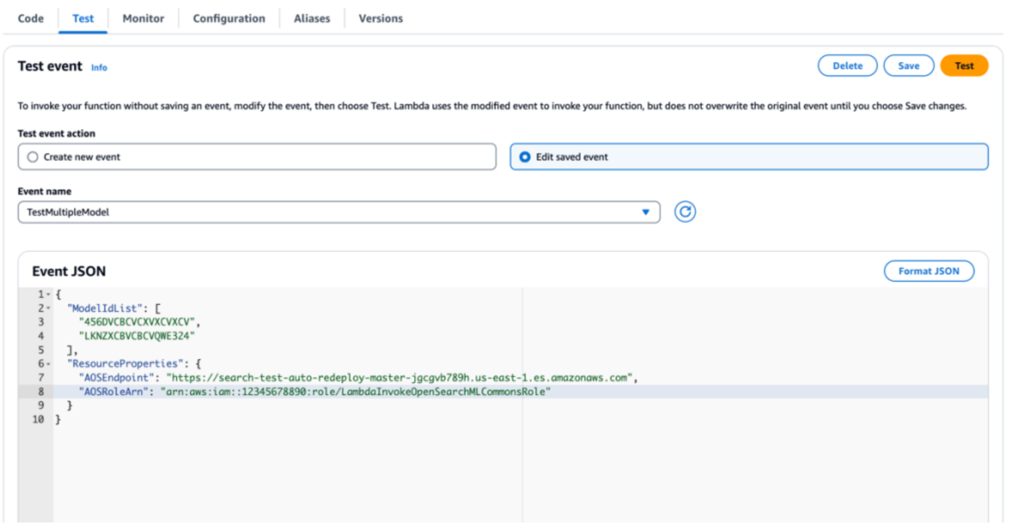


Description
-
Last modified
4 days ago
Function ARN
arn:aws:lambda:us-east-1:419213735998:function:test-function
Function URL Info
-

## 3.4 Testing

Now you can see the that the lambda_function.py in the Code source window. Click on Test Tab. Please put the model_id into ModelIdList that you would like to conduct auto model deployment, input the AOS endpoint which you can find out from AOS domain config and input the lambda role arn that you created in previous 3.1.2. Click "Test" to run auto model deployment.

```
{
  "ModelIdList": [
    "<model_id1>",
    "<model_id2>","
  ],
  "ResourceProperties": {
    "AOSEndpoint": "<AOSEndPoint>",
    "AOSRoleArn": "<RoleARN>"
  }
}
```



To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action
Create new event [✓ Edit saved event]

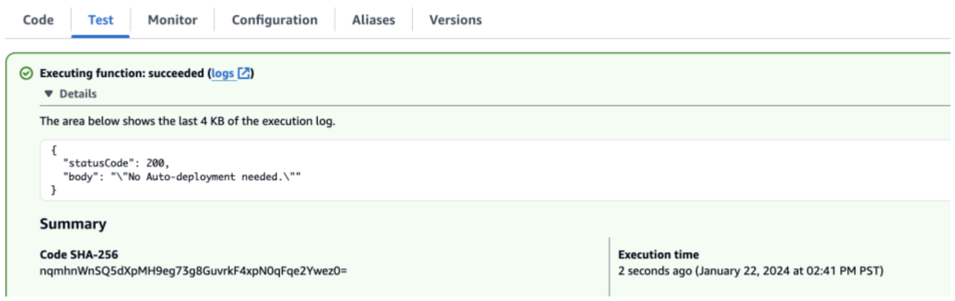
Event name
TestMultipleModel

Event JSON

```
{
  "ModelIdList": [
    "456DVCBCVCXVXCVXCV",
    "LKNZXCBVCBCVQWE324"
  ],
  "ResourceProperties": {
    "AOSEndpoint": "https://search-test-auto-redeploy-master-jgcgvb789h.us-east-1.es.amazonaws.com",
    "AOSRoleArn": "arn:aws:iam::12345678890:role/LambdaInvokeOpenSearchMLCommonsRole"
  }
}
```

3.4.1 Test Success

Please make sure the test success before adding trigger. The sample success outcome is similar to this.



Executing function: succeeded (logs)

Details

The area below shows the last 4 KB of the execution log.

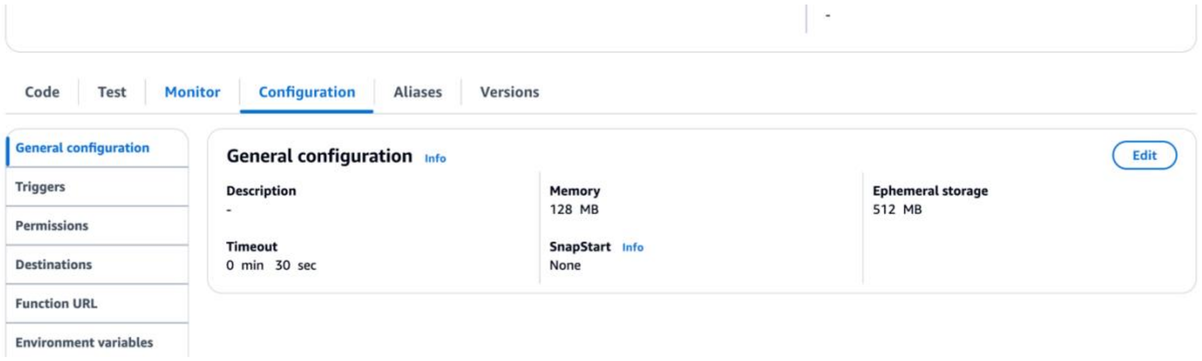
```
{
  "statusCode": 200,
  "body": "\"No Auto-deployment needed.\""
}
```

Summary

Code SHA-256
nqmhnWnSQ5dXpMH9eg73g8GuvrkF4xpN0qFqe2Ywez0=

Execution time
2 seconds ago (January 22, 2024 at 02:41 PM PST)

3.4.2 If lambda timeout, set Timeout to longer timeframe, maximum can be 15 minutes.



General configuration Info

Description
-

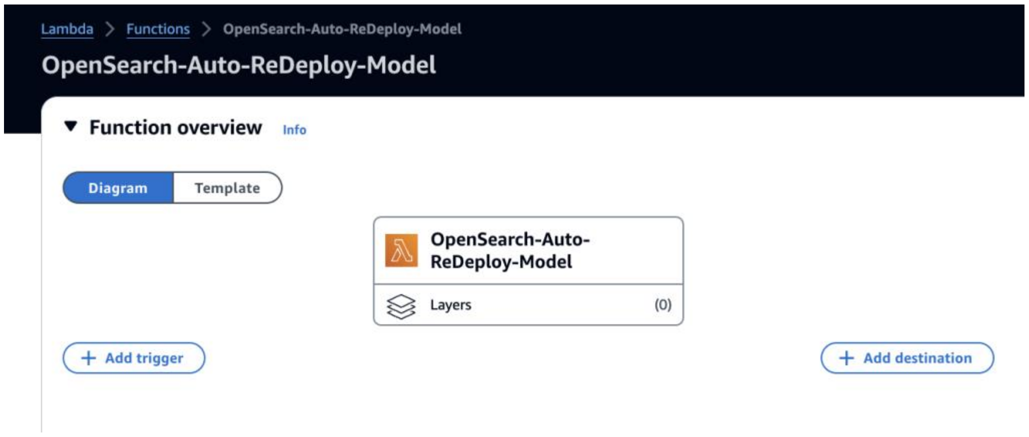
Memory
128 MB

Ephemeral storage
512 MB

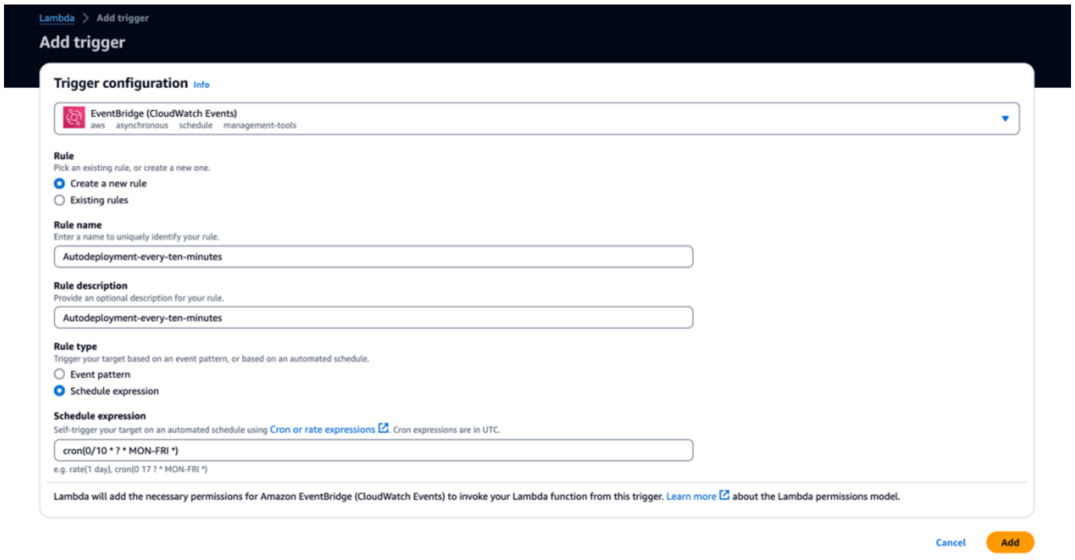
Timeout
0 min 30 sec

SnapStart Info
None

3.5 Add trigger to schedule the auto model deployment schedule


Lambda > Functions > OpenSearch-Auto-ReDeploy-Model
OpenSearch-Auto-ReDeploy-Model
Function overview Info
Diagram
Template
OpenSearch-Auto-ReDeploy-Model
Layers (0)
Add trigger
Add destination

3.5.1 in trigger config page, choose EventBridge and create new rule, choose the rule type to be schedule expression, and put on cron expression, for example, to run every 10 minutes during weekdays, e.g `cron(0/10 * ? * MON-FRI *)`, please refer to the Cron expressions reference to config different schedules.



Add trigger

Trigger configuration Info

EventBridge (CloudWatch Events)
aws asynchronous schedule management-tools

**Rule**
Pick an existing rule, or create a new one.
[✓ Create a new rule]
Existing rules

**Rule name**
Enter a name to uniquely identify your rule.
Autodeployment-every-ten-minutes

**Rule description**
Provide an optional description for your rule.
Autodeployment-every-ten-minutes

**Rule type**
Trigger your target based on an event pattern, or based on an automated schedule.
Event pattern
[✓ Schedule expression]

**Schedule expression**
Self-trigger your target on an automated schedule using Cron or rate expressions. Cron expressions are in UTC.
```
cron(0/10 * ? * MON-FRI *)
```
e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

Now, the auto deployment lambda job is detecting undeployed models from your provided model list and conduct auto-deployment in a schedule.