

Lambda

1. Create a Lambda function which get triggered from EC2 Action and Notify about changes via SNS Topic.

Function name

Enter a name that describes the purpose of your function.

ec2-trigger

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

Runtime [Info](#)

Choose the language to use to write your function.

Python 3.8

Permissions [Info](#)

Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ Choose or create an 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.

service-role

↺

Trigger configuration

 CloudWatch Events/EventBridge

aws events management-tools

▼

Rule

Pick an existing rule, or create a new one.

Create a new rule

Select or create a new rule

Rule name*

Enter a name to uniquely identify your rule.

ec2-Instance

Rule description

Provide an optional description for your rule.

state-change

Rule type

Trigger your target based on an event pattern, or based on an automated schedule.

☒ Event pattern

☐ Schedule expression

Schedule expression

EC2

EC2 Instance state-change notification

☐ Detail

▼ Event pattern preview

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EC2 Instance State-change Notification"
  ]
}
```

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

☒ Enable trigger

Enable the trigger now, or create it in a disabled state for testing (recommended).

Destination configuration

Send invocation records to a destination when your function is invoked asynchronously, or if your function processes records from a stream.

Source

The type of invocation that maps to the destination.

☒ Asynchronous invocation

☐ Stream invocation

Condition

The condition for using the destination.

☐ On failure

☒ On success

Destination type

An SQS queue, SNS topic, Lambda function, or EventBridge event bus.

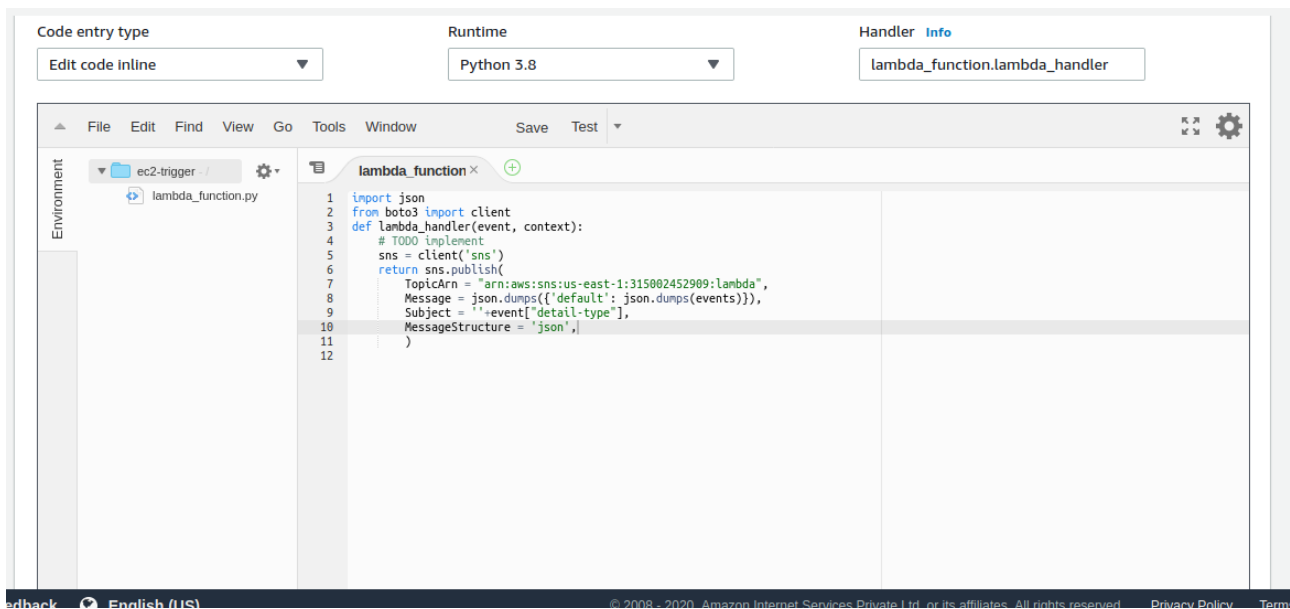
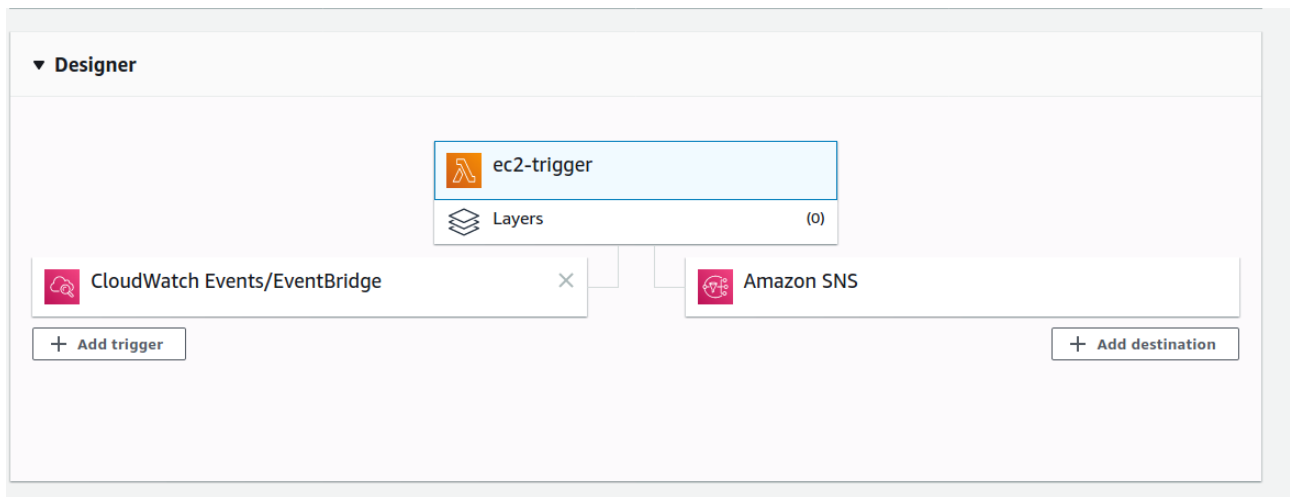
SNS topic

Destination

lambda

Cancel

Save



AWS Notification Message Inbox x



AWS Notifications <no-reply@sns.amazonaws.com>

3:30 AM (0 minutes ago)



to me ▾

{"version":"1.0","timestamp":"2020-05-08T22:00:31.578Z","requestContext":{"requestId":"758e92f1-83eb-41b4-a2d5-03a8c9239131","functionArn":"arn:aws:lambda:us-east-1:315002452909:function:ec2-trigger:\$LATEST","condition":"Success","approximateInvokeCount":1},"requestPayload":{"version":"0","id":"66a0e6c3-09f1-98ee-eac8-fa54afc8e6f","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"315002452909","time":"2020-05-08T22:00:30Z","region":"us-east-1","resources":["arn:aws:ec2:us-east-1:315002452909:instance/i-093fdeb434a448cde"],"detail":{"instance-id":"i-093fdeb434a448cde","state":"pending"}},"responseContext":{"statusCode":200,"executedVersion":"\$LATEST","responsePayload":{"statusCode": 200,"body":"\\\"Hello from Lambda!\\\"\"}}}

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:315002452909:lambda:2f7b4f6f-dbe3-49f2-b397-f99cb97e8079&Endpoint=jay.patel@tothenew.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

2. Create a Lambda function which gets invoked whenever a image is added to a s3 bucket and push the key to SQS.

Create New Queue

Queue Actions

Filter by Prefix:

1 to 1 of 1 items

<input type="checkbox"/>	Name	Queue Type	Content-Based Deduplication	Messages Available	Messages in Flight	Created
<input checked="" type="checkbox"/>	lambda	Standard	N/A	0	0	2020-05-09 03:46:44 GMT+05:30

1 SQS Queue selected

Details	Permissions	Redrive Policy	Monitoring	Tags	Encryption	Lambda Triggers
Name: lambda URL: https://sqs.us-east-1.amazonaws.com/315002452909/lambda ARN: arn:aws:sqs:us-east-1:315002452909:lambda Created: 2020-05-09 03:46:44 GMT+05:30 Last Updated: 2020-05-09 03:46:44 GMT+05:30 Delivery Delay: 0 seconds Queue Type: Standard Content-Based Deduplication: N/A						Default Visibility Timeout: 30 seconds Message Retention Period: 4 days Maximum Message Size: 256 KB Receive Message Wait Time: 0 seconds Messages Available (Visible): 0 Messages in Flight (Not Visible): 0 Messages Delayed: 0

Function name

Enter a name that describes the purpose of your function.

s3get

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

Runtime [Info](#)

Choose the language to use to write your function.

Python 3.8

Permissions [Info](#)

Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ Choose or create an 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.

service-role

Trigger configuration



Bucket

Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

jaypttn

Event type

Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

PUT

Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

e.g. .jpg

Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

s3get

Throttle

Qualifiers ▼

Actions ▼

Select a test event ▼

Test

Save

Code entry type

Edit code inline

Runtime

Python 3.8

Handler

Info
lambda_function.lambda_handler

```
File Edit Find View Go Tools Window Save Test
Environment
  s3get /
    lambda_function.py
lambda_function.py
1 import json
2 import boto3
3 s3 = boto3.client('s3')
4 if event:
5     rec = event["Records"][0]
6     k = str(rec["s3"]["object"]["key"])
7     response = client.send_message(
8         QueueUrl = 'https://sqs.us-east-1.amazonaws.com/315002452909/lambda',
9         MessageBody = k,
10        DelaySeconds = 5,
11        )
12
13
```

Upload

1 Select files

2 Set permissions


3 Set properties

4 Review

1 Files Size: 54.9 KB Target path: jaypttn

To upload a file larger than 160 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. [Learn more](#)

+ Add more files

 sample.png
- 54.9 KB

×

Upload

Next

jaypttn

Overview

Properties

Permissions

Management

Access points

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload


Create folder

Download

Actions

US East (N. Virginia)

Viewing 1 to 1

<input type="checkbox"/> Name	Last modified	Size	Storage class
<input type="checkbox"/>  sample.png	May 9, 2020 3:57:40 AM GMT+0530	54.9 KB	Standard

Viewing 1 to 1

Create New Queue

Queue Actions

Filter by Prefix: Enter Text...

☒ Name

Queue Type

Content-Based Deduplication

Messages Available

Messages in Flight

Created

☒ lambda

Standard

N/A

1

0

2020-05-09 03:46:44 GMT+05:30

3. The SQS should be FIFO and write a Lambda Function which will listen to SQS and compress the image and upload to some other S3 Bucket

Create New Queue

What do you want to name your queue?

Queue Name [Info](#)

jay.fifo

Region [Info](#) US East (N. Virginia)

What type of queue do you need?

Standard Queue

FIFO Queue

Unlimited Throughput: Standard queues support a nearly unlimited number of transactions per second (TPS) per API action.

High Throughput: FIFO queues support up to 300 messages per second (300 send, receive, or delete operations per second). When you batch 10 messages per operation (maximum), FIFO queues can support up to 3,000 messages per second. To request a limit increase, file a support request.

Basic information

Function name

Enter a name that describes the purpose of your function.

s3compress

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

Runtime [Info](#)

Choose the language to use to write your function.

Python 3.8

Permissions [Info](#)

Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ Choose or create an 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

Trigger configuration



SQS

aws

queue



SQS queue

Choose or enter the ARN of an SQS queue.



arn:aws:sqs:us-east-1:315002452909:jay.fifo



Batch size

The maximum number of messages to retrieve in a single batch.

10



In order to read from the SQS trigger, your execution role must have proper permissions.

☒ Enable trigger

Enable the trigger now, or create it in a disabled state for testing (recommended).

Cancel

Add

s3compress

Throttle

Qualifiers ▼

Actions ▼

Select a test event ▼

Test

Save

▼ Designer



s3compress



Layers

(0)



















SQS



+ Add destination

+ Add trigger

```
import json
from PIL import Image
import boto3
def lambda_handler(event, context):
    s3 = boto3.client('s3')
    for record in event["Records"]:
        image=record["body"]
        s3.download_file('jaypttn',image,'/tmp'+image)
        filepath=f'/tmp{image}'
        picture = Image.open(filepath)
        picture.save("/tmp/Compressed_"+image,optimize=True,quality=65)
        compressed = f'/tmp/Compressed_{image}'
        final_image = f'Compressed_{image}'
        res = s3.upload_file(compressed, 'jaypttn', Final_Image)
```


	boto3	1.1 MB	Folder	12 May 2020, 22:07
	boto3-1.12.43.dist-info	16.7 kB	Folder	12 May 2020, 22:07
	botocore	44.7 MB	Folder	12 May 2020, 22:07
	botocore-1.15.43.dist-info	92.2 kB	Folder	12 May 2020, 22:07
	dateutil	623.0 kB	Folder	12 May 2020, 22:07
	docutils	3.1 MB	Folder	12 May 2020, 22:07
	docutils-0.15.2.dist-info	34.8 kB	Folder	12 May 2020, 22:07
	jmespath	111.2 kB	Folder	12 May 2020, 22:07
	jmespath-0.9.5.dist-info	18.8 kB	Folder	12 May 2020, 22:07
	PIL	2.3 MB	Folder	12 May 2020, 22:07
	Pillow.libs	4.4 MB	Folder	12 May 2020, 22:07
	Pillow-7.1.1.dist-info	21.4 kB	Folder	12 May 2020, 22:07
	__pycache__	26.9 kB	Folder	12 May 2020, 22:07
	python_dateutil-2.8.1.dist-info	14.1 kB	Folder	12 May 2020, 22:07
	s3transfer	462.6 kB	Folder	12 May 2020, 22:07
	s3transfer-0.3.3.dist-info	5.8 kB	Folder	12 May 2020, 22:07
	S3	18.1 MB	Folder	12 May 2020, 21:58
	six-1.14.0.dist-info	3.5 kB	Folder	12 May 2020, 22:07
	urllib3	614.1 kB	Folder	12 May 2020, 22:07
	urllib3-1.25.9.dist-info	46.7 kB	Folder	12 May 2020, 22:07
	image-compression.py	531 bytes	Python script	12 May 2020, 22:34
	requirement.txt	126 bytes	plain text d...	12 May 2020, 22:03
	six.py	34.1 kB	Python script	12 May 2020, 22:06

4. Create a Lambda functions which gets triggered daily and takes the AMI of a particular EC2 instance(Filter on the basis of Tag).

Add permissions to service-role

Attach Permissions


Create policy

↺

Filter policies ▾

ec2full

Showing 1 result

	Policy name ▾	Type	Used as
<input checked="" type="checkbox"/>	 AmazonEC2FullAccess	AWS managed	Permissions policy (1)

Cancel

Attach policy

Basic information

Function name

Enter a name that describes the purpose of your function.

ec2-ami

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

Runtime [Info](#)

Choose the language to use to write your function.

Python 3.8

Permissions [Info](#)

Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ Choose or create an 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.

service-role

 CloudWatch Events/EventBridge
aws events management-tools

Rule

Pick an existing rule, or create a new one.

Create a new rule

Select or create a new rule

Rule name*

Enter a name to uniquely identify your rule.

daily

Rule description

Provide an optional description for your rule.

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.

rate(1 day)

lambda_function ×

1 import json

2 import boto3

3

4 def lambda_handler(event, context):

5 client = boto3.client('ec2')

6 response = client.describe_instances(

7 Filters=[

8 {

9 'Name': 'tag:Name',

10 'Values': [

11 'Jay',

12],

13 },

14]

15)

16 print(response)

17 for reservation in (response['Reservations']):

18 for instance in (reservation['Instances']):

19 AMI = client.create_image(

20 Description="This AMI is created by Lambda",

21 InstanceId=instance['InstanceId'],

22 Name="Lambd-Created-AMI",

23 NoReboot=True

24)

25

11:25 Python Spaces: 4

Execution Result ×

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
<input type="checkbox"/>	Jay	i-093fdeb434a448cde	t2.micro	us-east-1c	● running	Initializing

Instance: **i-093fdeb434a448cde (Jay)** Public DNS: ec2-3-235-108-0.compute-1.amazonaws.com

Description Status Checks Monitoring **Tags**

Add/Edit Tags

Key	Value
Name	Jay

[Hide Column](#)

Launch Actions

<input type="checkbox"/>	Name	AMI Name	AMI ID	Source	Owner	Visibility	Status
	Lambda-Created-AMI		ami-00ee36c1c117328d6	315002452909/...	315002452909	Private	pending

Image: ami-00ee36c1c117328d6

Details Permissions Tags

Edit

AMI ID ami-00ee36c1c117328d6

AMI Name Lambda-Created-AMI

5. Create a Lambda function which will login to a EC2 instance and prints all the running services. (Use python's paramiko module to do SSH. Also, launch lambda in a VPC).(Keep Keys in S3 and S3 should be encrypted)

Filter by tags and attributes or search by keyword

1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
Jay	i-093fdeb434a448cde	t2.micro	us-east-1c	running	Initializing

Instance: i-093fdeb434a448cde (Jay)Public DNS: ec2-18-207-180-2.compute-1.amazonaws.com

Description

Status Checks

Monitoring

Tags

Instance ID	i-093fdeb434a448cde	Public DNS (IPv4)	ec2-18-207-180-2.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	18.207.180.2
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-172-31-24-247.ec2.internal	Availability zone	us-east-1c

jaypttn

Overview

Properties

Permissions

Management

Access points

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload

Create folder

Download

Actions

US East (N. Virginia)

Viewing 1 to 6

Name	Last modified	Size	Storage class
<input checked="" type="checkbox"/> Jay.pem	May 12, 2020 10:02:46 PM GMT+0530	1.6 KB	Standard

This property does not affect existing objects in your bucket.

☐ None

☐ AES-256

Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

☒ AWS-KMS

Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

aws/s3



Amazon S3 evaluates and applies bucket policies before applying bucket encryption settings. Even if you enable bucket encryption settings, your PUT requests without encryption information will be rejected if you have bucket policies to reject such PUT requests. Check your bucket policy and modify it if required.

[View bucket policy](#)

☐ Disabled

Cancel

Save

Open

Download

Download as

Make public

Copy path

Owner

jay.patel1325

Last modified

May 12, 2020 10:02:46 PM GMT+0530

Etag

0ea4159a17d186ddf3c4f4d4eeee06b0

Storage class

Standard

Server-side encryption

AWS-KMS

KMS key ID

arn:aws:kms:us-east-1:315002452909:key/a5fdd2d1-b264-46b6-be62-1dfbbd9fc698

Size

1.6 KB

Key

Jay.pem

Object URL

<https://javnftn-s3.amazonaws.com/Jay.pem>

Basic information

Function name

Enter a name that describes the purpose of your function.

ec2-services

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

Runtime [Info](#)

Choose the language to use to write your function.

Python 3.8

Permissions [Info](#)

Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.




















▼ Choose or create an 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

```
import paramiko
import boto3
def ssh(event, context):
    s3 = boto3.client('s3')
    s3.download_file('jaypttn', 'Jay.pem', '/tmp/Jay.pem')
    key = paramiko.RSAKey.from_private_key_file('/tmp/Jay.pem')
    client = paramiko.SSHClient()
    client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    client.connect(hostname='18.207.180.2', username='ubuntu', pkey=key)
    stdin, stdout, stderr = client.exec_command('ps -eaf')
    print(stdout.read())
    client.close()
```


	.libs_cffi_backend	46.6 kB	Folder	23 April 2020, 23:34
	bcrypt	123.0 kB	Folder	23 April 2020, 23:34
	bcrypt-3.1.7.dist-info	19.1 kB	Folder	23 April 2020, 23:34
	cffi	613.9 kB	Folder	23 April 2020, 23:34
	cffi-1.14.0.dist-info	5.7 kB	Folder	23 April 2020, 23:34
	cryptography	8.9 MB	Folder	23 April 2020, 23:34
	cryptography-2.9.2.dist-info	38.7 kB	Folder	23 April 2020, 23:34
	nacl	2.8 MB	Folder	23 April 2020, 23:34
	paramiko	1.2 MB	Folder	23 April 2020, 23:34
	paramiko-2.7.1.dist-info	37.7 kB	Folder	23 April 2020, 23:34
	__pycache__	26.9 kB	Folder	23 April 2020, 23:34
	pycparser	948.7 kB	Folder	23 April 2020, 23:34
	pycparser-2.20.dist-info	5.3 kB	Folder	23 April 2020, 23:34
	PyNaCl-1.3.0.dist-info	20.9 kB	Folder	23 April 2020, 23:34
	six-1.14.0.dist-info	3.5 kB	Folder	23 April 2020, 23:34
	_cffi_backend.cpython-36m-x86_64-linux-gnu.so	853.8 kB	shared library	23 April 2020, 23:34
	my_handler.py	475 bytes	Python script	12 May 2020, 22:06
	requirement.txt	105 bytes	plain text document	23 April 2020, 23:34
	six.py	34.1 kB	Python script	23 April 2020, 23:34

Function code [Info](#)

 The deployment package of your Lambda function "ec2-services" is too large to enable inline code editing. However, you can still invoke your function.

Code entry type

Upload a .zip file

Runtime

Python 3.6

Handler [Info](#)

my_handler.ssh

Function package

 Upload

For files larger than 10 MB, consider uploading using Amazon S3.

Duration

1621.49 ms

Billed duration

1700 ms

Resources configured

128 MB

Max memory used

88 MB

Log output

The section below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. [Click here](#) to view the CloudWatch log group.

```

16:32 ?      00:00:00 [nvme-wq]\nroot      135      2  0 16:32 ?      00:00:00 [nvme-reset-wq]\nroot      136
2  0 16:32 ?      00:00:00 [nvme-delete-wq]\nroot      137      2  0 16:32 ?      00:00:00 [scsi_ah_0]\nroot
138      2  0 16:32 ?      00:00:00 [scsi_tmf_0]\nroot      139      2  0 16:32 ?      00:00:00 [scsi_ah_1]\nroot
140      2  0 16:32 ?      00:00:00 [scsi_tmf_1]\nroot      142      2  0 16:32 ?      00:00:00 [ip6_addrconf]\nroot
150      2  0 16:32 ?      00:00:00 [kworker/0:1H-kb]\nroot      154      2  0 16:32 ?      00:00:00 [kstrp]\nroot
224      2  0 16:32 ?      00:00:00 [cryptd]\nroot      341      2  0 16:32 ?      00:00:00 [raid5wq]\nroot      390
2  0 16:32 ?      00:00:00 [jbd2/xvda1-8]\nroot      391      2  0 16:32 ?      00:00:00 [ext4-rsv-conver]\nroot
461      2  0 16:32 ?      00:00:00 [iscsi_ah]\nroot      464      1  0 16:32 ?      00:00:00 /lib/systemd/systemd-
journald\nroot      474      2  0 16:32 ?      00:00:00 [ib-comp-wq]\nroot      475      2  0 16:32 ?      00:00:00
[ib-comp-unb-wq]\nroot      476      1  0 16:32 ?      00:00:00 /sbin/lvmetad -f\nroot      477      2  0 16:32 ?
00:00:00 [ib_mcast]\nroot      478      2  0 16:32 ?      00:00:00 [ib_nl_saw]\nroot      482      2  0 16:32 ?

```