

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

ques-1

ThrottleQualifiers ▼Actions ▼Select a test event ▼TestSave

ConfigurationPermissionsMonitoring

▼ Designer

ques-1

Layers (0)

+ Add trigger

+ Add destination

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-action

Rule description

Provide an optional description for your rule.

action on ec2

Rule type

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

Event pattern

Schedule expression

EC2

All events

ec2-action-ques1

EditDeletePublish message

Details

Name	ec2-action-ques1	Display name	ec2-action
ARN	arn:aws:sns:us-east-2:835417514122:ec2-action-ques1	Topic owner	835417514122

SubscriptionsAccess policyDelivery retry policy (HTTP/S)Delivery status loggingEncryptionTags

Subscriptions (0)

EditDeleteRequest confirmationConfirm subscriptionCreate subscription

Subscription: faaf0975-ef0a-486f-942a-103b3ee1e6c4

EditDelete

Details

ARN	arn:aws:sns:us-east-2:835417514122:ec2-action-ques1:faaf0975-ef0a-486f-942a-103b3ee1e6c4	Status	⌚ Pending confirmation
Endpoint	ayush.me001@gmail.com	Protocol	EMAIL
Topic	ec2-action-ques1		

Add destination

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

ec2-action-ques1



Cancel

Save

ques-1

Throttle

Qualifiers ▼

Actions ▼

Select a test event ▼

Test

Save

Configuration

Permissions

Monitoring

▼ Designer

ques-1

Layers

(0)

CloudWatch Events/EventBridge



Amazon SNS

+ Add trigger

+ Add destination

```
File Edit Find View Go Tools Window Save Test
ques-2 /
  lambda_function.py
lambda_function
1 from boto3 import client
2 import json
3
4 def lambda_handler(event, context):
5     sns = client('sns')
6     response = sns.publish(
7         TopicArn="arn:aws:sns:us-east-1:835417514122:ec2-action-ques1",
8         Message= json.dumps({'default': json.dumps(event)}),
9         Subject='+event["detail-type"]',
10        MessageStructure='json',
11    )
12    return response
```

EC2 Instance State-change Notification Inbox x



ec2-action <no-reply@sns.amazonaws.com>

5:25 PM (1 minute ago)



to me ▾

{ "version": "0", "id": "09adea73-88c2-ceb5-847d-72670e677041", "detail-type": "EC2 Instance State-change Notification", "source": "aws.ec2", "account": "835417514122", "time": "2020-04-22T11:55:43Z", "region": "us-east-2", "resources": ["arn:aws:ec2:us-east-2:835417514122:instance/i-041a02c161bf670c3"], "detail": { "instance-id": "i-041a02c161bf670c3", "state": "pending" } }

--

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

<https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:835417514122:ec2-action-ques1:faaf0975-ef0a-486f-942a-103b3ee1e6c4&Endpoint=ayush.me001@gmail.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>



ec2-action <no-reply@sns.amazonaws.com>

5:26 PM (0 minutes ago)



to me ▾

{ "version": "0", "id": "6cf567e0-76e6-3ced-7487-15564c0f5a4d", "detail-type": "EC2 Instance State-change Notification", "source": "aws.ec2", "account": "835417514122", "time": "2020-04-22T11:56:05Z", "region": "us-east-2", "resources": ["arn:aws:ec2:us-east-2:835417514122:instance/i-041a02c161bf670c3"], "detail": { "instance-id": "i-041a02c161bf670c3", "state": "running" } }

...

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

ques-2

ThrottleQualifiers ▼Actions ▼Select a test event ▼TestSave

ConfigurationPermissionsMonitoring

▼ Designer

ques-2

Layers (0)

+ Add trigger

+ Add destination

Amazon S3

Buckets (1)

Copy ARNEmptyDeleteCreate bucket

Find bucket by name

	Name ▼	Region ▼	Access ▼	Bucket created
	ques-2-s3	US East (Ohio) us-east-2	Not Public	2020-04-22T12:27:20.000Z

Trigger configuration



S3
aws storage



Bucket

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

ques-2-s3



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.

☒ Enable trigger

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

Create New Queue

What do you want to name your queue?

Queue Name ⓘ

s3-image-ques-2.fifo

Region ⓘ US East (Ohio)

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.

Create New Queue

Queue Actions

Filter by Prefix:

✕

⏪

<input type="checkbox"/>	Name	Queue Type	Content-Based Deduplication	Messages Available	Messages in Flight	Create
<input type="checkbox"/>	s3-image-ques-2.fifo	FIFO	Disabled	0	0	2020-0

1 SQS Queue selected

Details

Permissions

Redrive Policy

Monitoring

Tags

Encryption

Lambda Triggers

Name:

s3-image-ques-2.fifo

URL:

https://sqs.us-east-2.amazonaws.com/835417514122/s3-image-ques-2.fifo

ARN:

arn:aws:sqs:us-east-2:835417514122:s3-image-ques-2.fifo

Created:

2020-04-22 17:35:17 GMT+05:30

Last Updated:

2020-04-22 17:35:17 GMT+05:30

Delivery Delay:

0 seconds

Queue Type:

FIFO

Content-Based Deduplication:

Disabled

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

File

Edit

Find

View

Go

Tools

Window

Save

Test

Environment

ques-2

lambda_function.py

lambda_function

1

import uuid

2

import boto3

3

def lambda_handler(event, context):

4

client = boto3.client('sqs')

5

for record in event["Records"]:

6

dupId = uuid.uuid4()

7

print(dupId)

8

response = client.send_message(

9

QueueUrl='https://sqs.us-east-2.amazonaws.com/835417514122/s3-image-ques-2.fifo ',

10

MessageBody=record["s3"]["object"]["key"],

11

MessageDeduplicationId=f'{dupId}',

12

MessageGroupId='test-ID'

13

)

14

return response

15

Upload

1 Select files

2 Set permissions


3 Set properties

4 Review

1 Files Size: 20.0 KB Target path: ques-2-s3

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

[+ Add more files](#)

 Screenshot from 2020-02-04 09-53-07.png
- 20.0 KB

Upload

Next


View/Delete Messages in s3-image-ques-2.fifo

View up to: 10 messages Poll queue for: 30 seconds

Start Polling for Messages

Stop Now

Polling for new messages once every 2 seconds.

Delete	Body	Message Group ID	Message Deduplication ID	Sequence Number	Size	Sent	Receive Count
	Screenshot+from+2020-02-04 09-53-07.png	test-ID	2a44635f-1050-479c-af22-dfe4f...	18853159348495343616	39 bytes	2020-04-22 18:17:47 GMT+05:30	1

100%

Stopped after polling the queue at 0.5 receives/second for 30.3 seconds. Messages shown above are now available to other consumers.

Close

Delete Messages

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 ⓘ

s3-image.fifo

Region ⓘ

US East (Ohio)

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.


At-Least-Once Delivery: A message is delivered at least once, but occasionally more than one copy of a message is delivered.

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.

First-In-First-out Delivery: The order in which messages are sent and received is strictly preserved.

Add trigger

Trigger configuration

 **SQS**
aws queue

SQS queue

Choose or enter the ARN of an SQS queue.

arn:aws:sqs:us-east-2:835417514122:s3-image.fifo

X

↺

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

ques-3-s3

Overview

Properties

Permissions

Management

Access points

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

📁 Upload

+ Create folder

Download

Actions ▾

☐ Name ▾

Last modified ▾

Size ▾

☐ 🖼️ Selection_006.png

Apr 22, 2020 7:03:21 PM GMT+0530

661.2 KB

ques-3

Throttle

Qualifiers ▾

Actions ▾

Select a test event ▾

Test


Save


Configuration

Permissions

Monitoring

▼ Designer

 ques-3
🔄 Unsaved changes

 Layers (0)

🔗 SQS

×

+ Add destination

+ Add trigger

File Edit Find View Go Tools Window

Save Test ▾

ques-3 - /
Pillow-7.1.1.dist-info
Pillow.libs
PIL
lambda_function.py
requirements.txt

lambda_function x +

```
1 import boto3
2 from PIL import Image
3
4 def lambda_handler(event, context):
5     s3 = boto3.resource('s3')
6     client = boto3.client('s3')
7     for record in event["Records"]:
8         image = record["body"]
9         response = s3.Object('ques-3-s3', f'{image}').download_file(f'/tmp/{image}')
10        filepath = f'/tmp/{image}'
11        picture = Image.open(filepath)
12        picture.save(f'/tmp/Compressed_{image}', optimize=True, quality=65)
13        compressed = f'/tmp/Compressed_{image}'
14        final_image = f'Compressed_{image}'
15        res = client.upload_file(compressed, 'compressed-s3', final_image)
16
```

Send a Message to s3-image.fifo



Message Body

Message Attributes

Enter the text of a message you want to send.

Selection_006.png

Because content-based deduplication is disabled for this queue, the Message Group ID and Message Deduplication ID are both required.

Message Group ID ⓘ 432

Message Deduplication ID ⓘ 232323

Cancel

Send Message

compressed-s3

Overview

Properties

Permissions

Management

Access points

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

📁 Upload

+ Create folder

Download

Actions ▾

☐ Name ▾

Last modified ▾

Size ▾

☐ 🖼️ Compressed_Selection_006.png


Apr 22, 2020 7:19:52 PM GMT+0530

33.9 KB

Image in s3->ques2 lambda-> sqs -> ques3 lambda->new s3 compressed image

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

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.

ami-ec2-ques4

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)|

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

▼ Designer

 ques-4
 Layers (0)

 CloudWatch Events/EventBridge

+ Add trigger

+ Add destination

```
File Edit Find View Go Tools Window Save Test
ques-4 /
lambda_function.py
lambda_function x
1 import boto3
2 def lambda_handler(event, context):
3     client = boto3.client('ec2')
4     response = client.describe_tags(Filters =[
5         {
6             'Name': 'tag:Name',
7             'Values': [
8                 'ayush-ec2',
9             ]
10        },
11    ],
12    DryRun = False
13    )
14    for instanceId in response['Tags']:
15        res = client.create_image(
16            Description=' lambda AMI',
17            DryRun=False,
18            InstanceId=f'{{instanceId["ResourceId"]}}',
19            Name='Lambda AMI',
20            NoReboot=True,
21        )
22    return res
```

search : i-00201159f3f6dd79a Add filter

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
ayush-ec2	i-00201159f3f6dd79a	t2.micro	us-east-2b	running	Initializing	None

Instance: i-00201159f3f6dd79a (ayush-ec2) Public DNS: ec2-13-58-40-51.us-east-2.compute.amazonaws.com

DescriptionStatus ChecksMonitoringTags

Add/Edit Tags

Key	Value
Name	ayush-ec2

▼ Execution results

Response:

```
{
  "ImageId": "ami-03e4cd5190485a397",
  "ResponseMetadata": {
    "RequestId": "e3b65b3b-e88a-49de-89ab-18effd77d937",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "x-amzn-requestid": "e3b65b3b-e88a-49de-89ab-18effd77d937",
      "content-type": "text/xml; charset=UTF-8",
      "content-length": "242",
      "date": "Wed, 22 Apr 2020 15:19:22 GMT",
      "server": "AmazonEC2"
    },
    "RetryAttempts": 0
  }
}
```

Request ID:

"c0b8cbcf-6b28-446a-a570-0074f2e75165"

Launch

Actions ▼

Owned by me ▼



Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name ▼	AMI Name ▲	AMI ID ▼	Source ▼	Owner ▼	Visibility
<input type="checkbox"/>		Lambda AMI	ami-0bbddb45dd775d1e9	835417514122/...	835417514122	Private

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)

ques-5

Overview

Properties

Permissions

Management

Access points

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

Upload

Create folder

Download

Actions

<input type="checkbox"/>	Name	Last modified
<input type="checkbox"/>	ayush.pem	Apr 22, 2020 9:17:54 PM GMT+0530

Default encryption

Automatically encrypt objects when stored in Amazon S3

[Learn more](#)

 AES-256


```

(ques-5) ayush@ayush:~/ques-5$ ls
bcrypt
bcrypt-3.1.7.dist-info
cffi
cffi-1.14.0.dist-info
_cffi_backend.cpython-36m-x86_64-linux-gnu.so
cryptography
cryptography-2.9.1.dist-info
main.py
nacl
paramiko
paramiko-2.7.1.dist-info
__pycache__
pycparser
pycparser-2.20.dist-info
PyNaCl-1.3.0.dist-info
ques-5-s3.zip
six-1.14.0.dist-info
six.py

(ques-5) ayush@ayush:~/ques-5$ cat main.py
import paramiko
import boto3
def ssh(event,context):
    s3 = boto3.client('s3')
    s3.download_file('ques-5','ayush.pem','/tmp/ayush.pem')
    ayush = paramiko.RSAKey.from_private_key_file('/tmp/ayush.pem')
    client = paramiko.SSHClient()
    client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    client.connect(hostname='13.58.40.51',username="ubuntu",pkey=ayush)
    stdin,stdout,stderr = client.exec_command('ps -eaf')
    print(stdout.read())
    client.close()

(ques-5) ayush@ayush:~/ques-5$

```

ques-5-s3
Throttle
Qualifiers
Actions
test
Test
Save

Execution result: succeeded (logs)
Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

null

Summary

Code SHA-256
XrUHJypgQtLxJ6dv+uH2GuFxUDPozg7u9koki16BQpQ=

Duration
2953.49 ms

Resources configured
128 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.

```

d0]\nroot      37      2  0 14:02 ?      00:00:00 [kworker/u31:0]\nroot      38      2  0 14:02 ?      00:00:00 [ecryptfs-kthrea]\nroot      80      2  0 14:02
?      00:00:00 [kthrotld]\nroot      81      2  0 14:02 ?      00:00:00 [nvme-wq]\nroot      82      2  0 14:02 ?      00:00:00 [scsi_
eh_0]\nroot      83
2  0 14:02 ?      00:00:00 [scsi_tmf_0]\nroot      84      2  0 14:02 ?      00:00:00 [scsi_
eh_1]\nroot      85      2  0 14:02 ?      00:00:00
[scsi_tmf_1]\nroot      86      2  0 14:02 ?      00:00:00 [kworker/u30:2]\nroot      90      2  0 14:02 ?      00:00:00 [ip
v6_addrconf]\nroot      100      2
0 14:02 ?      00:00:00 [kstrp]\nroot      160      2  0 14:02 ?      00:00:00 [kworker/0:1H]\nroot      274      2  0 14:03 ?      00:00:00 [raid5wq]\nroot
324      2  0 14:03 ?      00:00:00 [jbd2/xvda1-8]\nroot      325      2  0 14:03 ?      00:00:00 [ext4-rsv-conver]\nroot      403      2  0 14:03 ?
00:00:00 [kworker/0:2]\nroot      404      1  0 14:03 ?      00:00:00 /lib/systemd/systemd-journald\nroot      406      2  0 14:03 ?      00:00:00
[iscsi_
eh]\nroot      408      1  0 14:03 ?      00:00:00 /sbin/lvm
etad -f\nroot      412      1  0 14:03 ?      00:00:00 /lib/systemd/systemd-udevd\nroot
413      2  0 14:03 ?      00:00:00 [ib-comp-wq]\nroot      414      2  0 14:03 ?      00:00:00 [ib-comp-unb-wq]\nroot      415      2  0 14:03 ?      00:00:00
[ib_mcast]\nroot      416      2  0 14:03 ?      00:00:00 [ib_nl_sa_wq]\nroot      426      2  0 14:03 ?      00:00:00 [rdma_cm]\nnsystemd+
450      1  0 14:03 ?

```