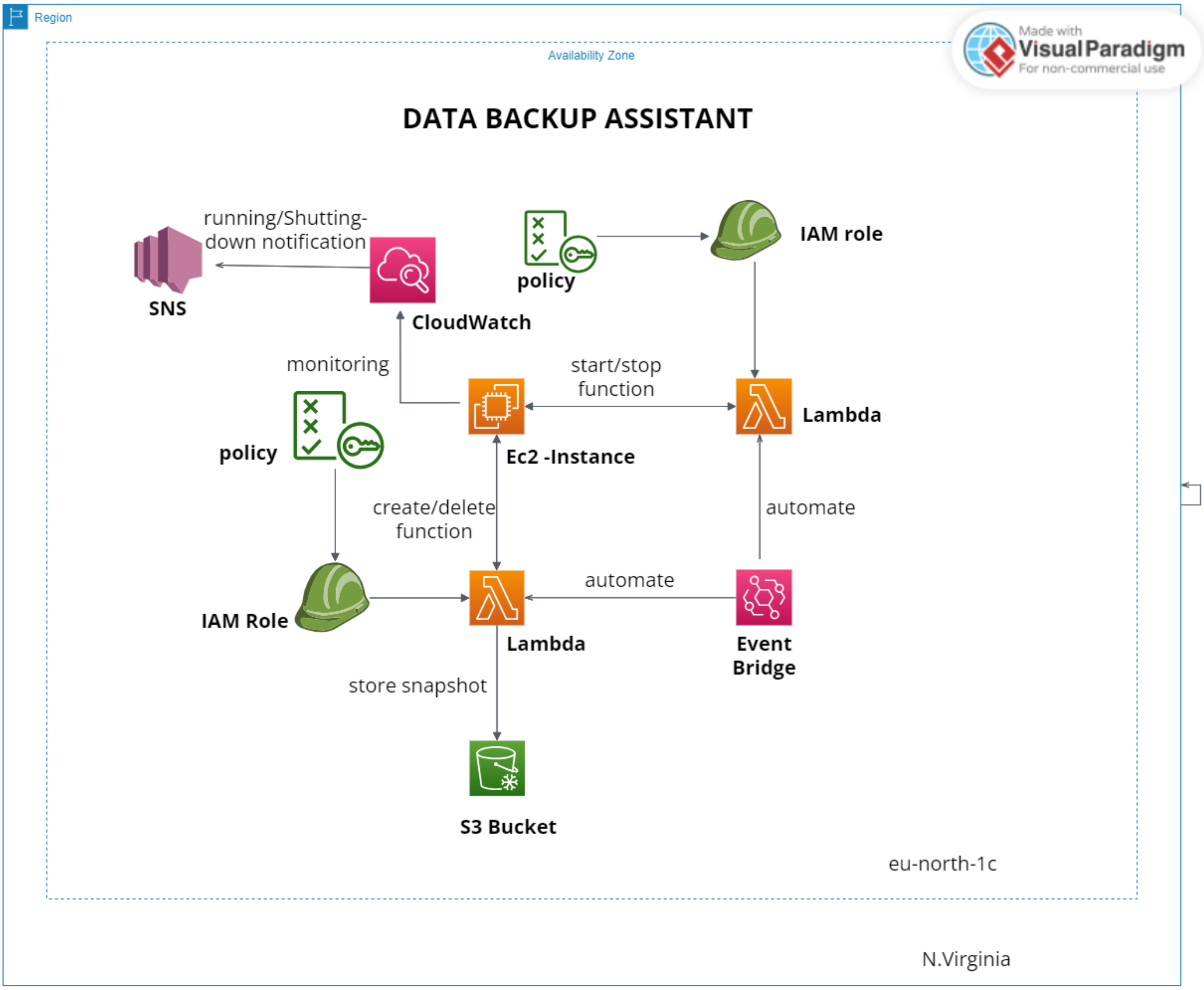
Project title: Data backup assistant

Architecture:

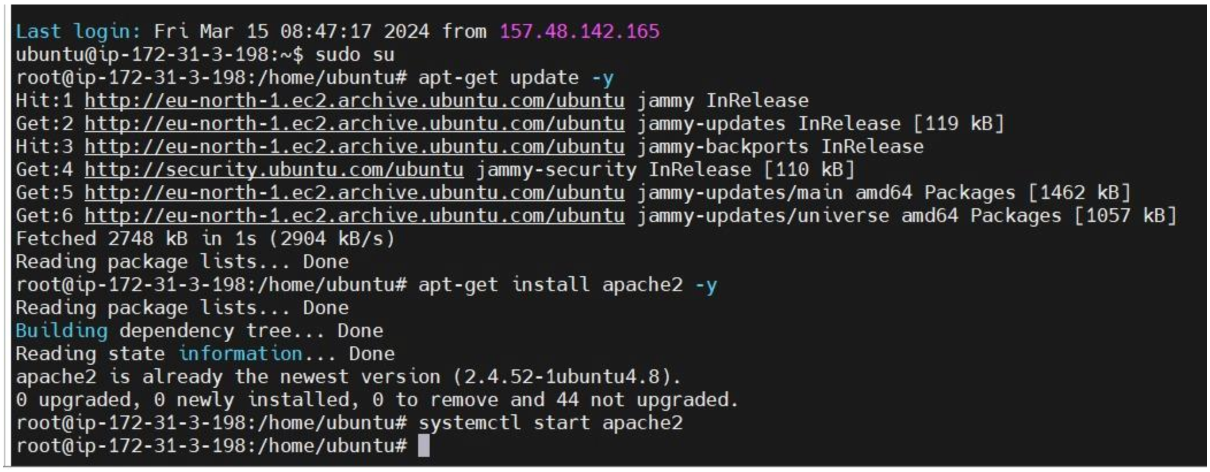


Step 1: Creating a snapshot of one server and retriving in another server

1.launch an ubuntu instance1

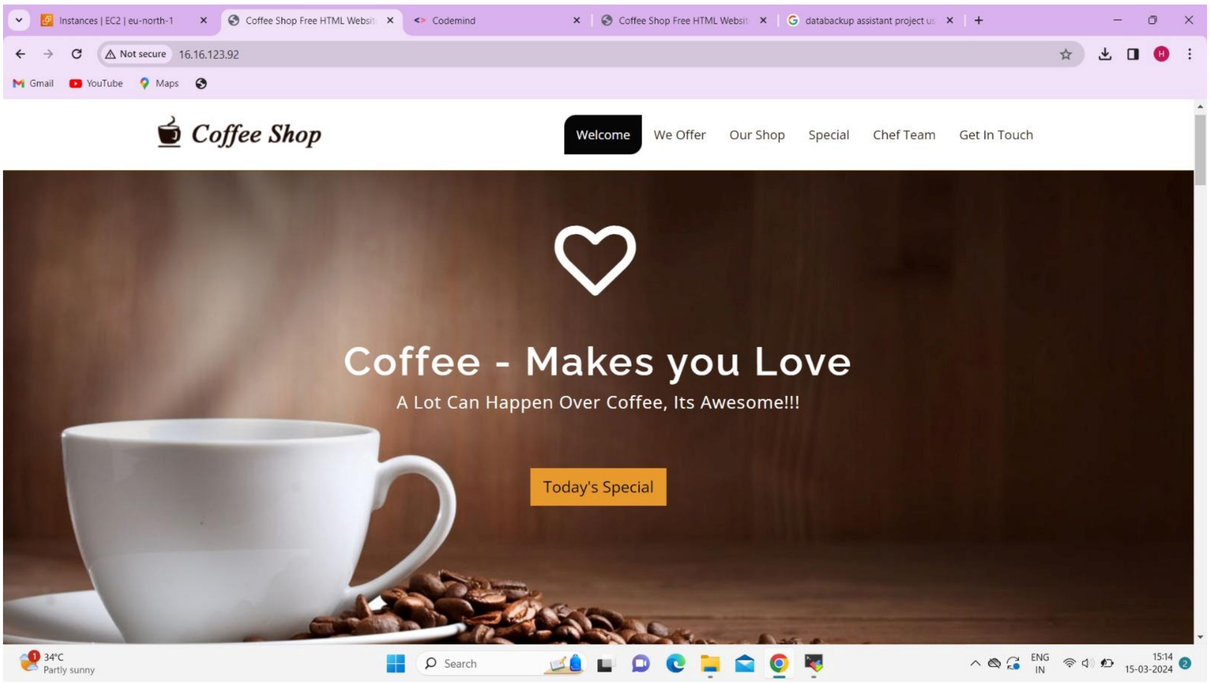
-Connect through mobaxterm by using ip address(public) of instance1

-Commands to launch a static website as follows:



```
Last login: Fri Mar 15 08:47:17 2024 from 157.48.142.165
ubuntu@ip-172-31-3-198:~$ sudo su
root@ip-172-31-3-198:/home/ubuntu# apt-get update -y
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1462 kB]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1057 kB]
Fetched 2748 kB in 1s (2904 kB/s)
Reading package lists... Done
root@ip-172-31-3-198:/home/ubuntu# apt-get install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4.8).
0 upgraded, 0 newly installed, 0 to remove and 44 not upgraded.
root@ip-172-31-3-198:/home/ubuntu# systemctl start apache2
root@ip-172-31-3-198:/home/ubuntu#
```

2.launch a static website in instance1 using ip address



Coffee - Makes you Love

A Lot Can Happen Over Coffee, Its Awesome!!!

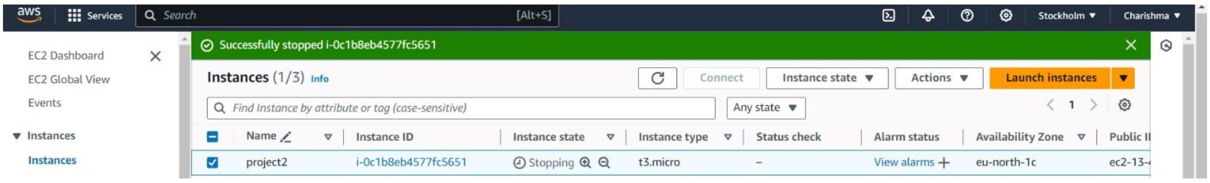
3.Create a snapshot for instance1

- Select instance1 volume for creation of snapshot.

4.launch another ubuntu instance2

-detach the default volume of instance2

- stop the instance2 for a while to detach the volume.



Successfully stopped i-0c1b8eb4577fc5651

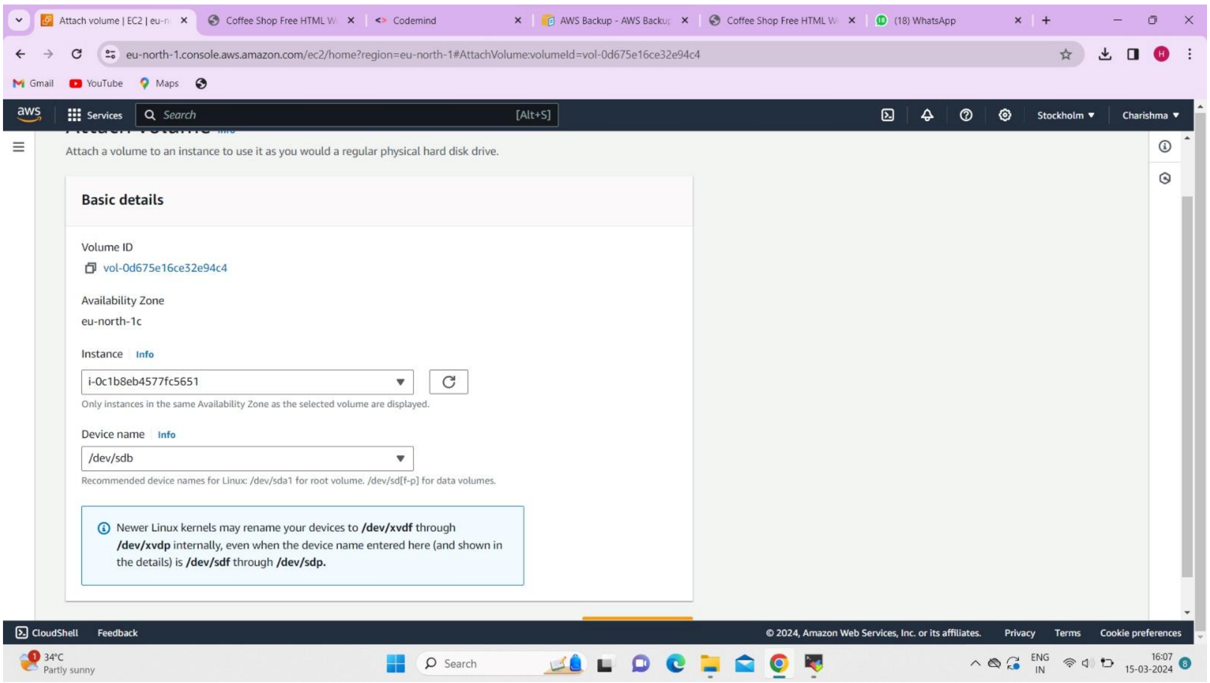
Instances (1/3)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public II
project2	i-0c1b8eb4577fc5651	Stopping	t3.micro	–	View alarms	eu-north-1c	ec2-13-

5.Now make the snapshot as a volume to attach to instance2

-Note: make sure all are in same availability zone

6.Attach the created volume from the instance1 to instance 2



eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#AttachVolume:volumeId=vol-0d675e16ce32e94c4

Attach a volume to an instance to use it as you would a regular physical hard disk drive.

Basic details

Volume ID
vol-0d675e16ce32e94c4

Availability Zone
eu-north-1c

Instance
i-0c1b8eb4577fc5651
Only instances in the same Availability Zone as the selected volume are displayed.

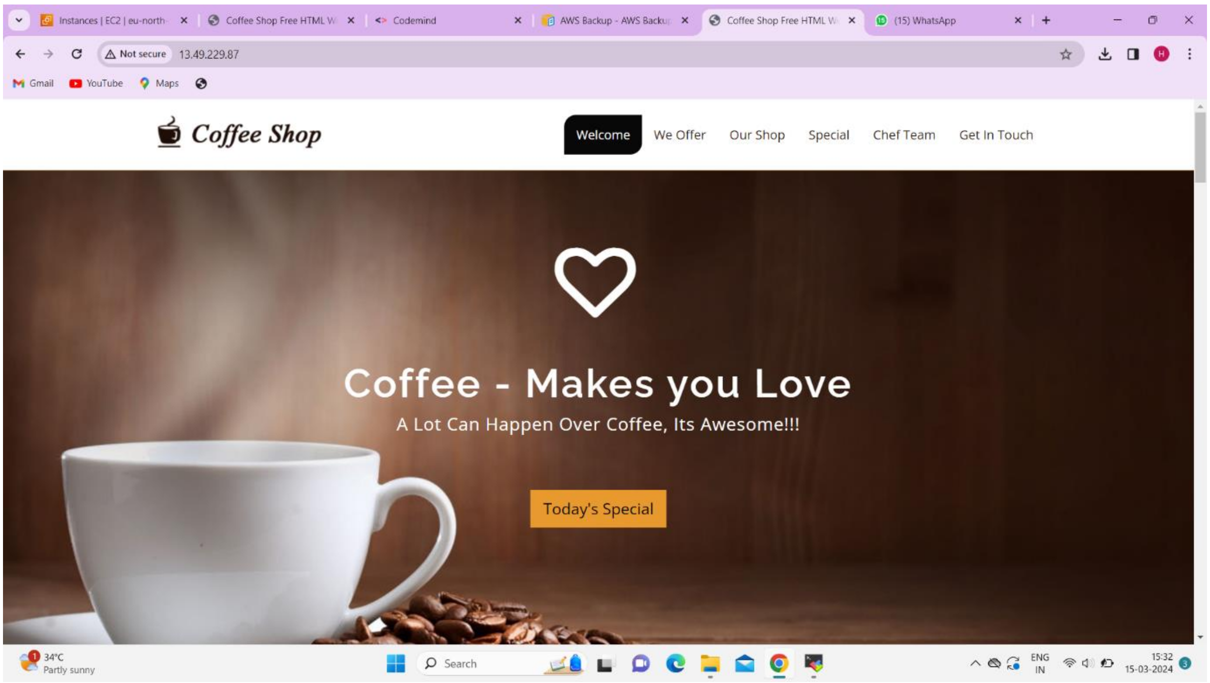
Device name
/dev/sdb
Recommended device names for Linux: /dev/sda1 for root volume. /dev/sd[f-p] for data volumes.

Newer Linux kernels may rename your devices to **/dev/xvdf** through **/dev/xvdp** internally, even when the device name entered here (and shown in the details) is **/dev/sdf** through **/dev/sdp.**

7.Start the instance2 (make it available)

-It gives an error to make a device name as it specified to attach a volume

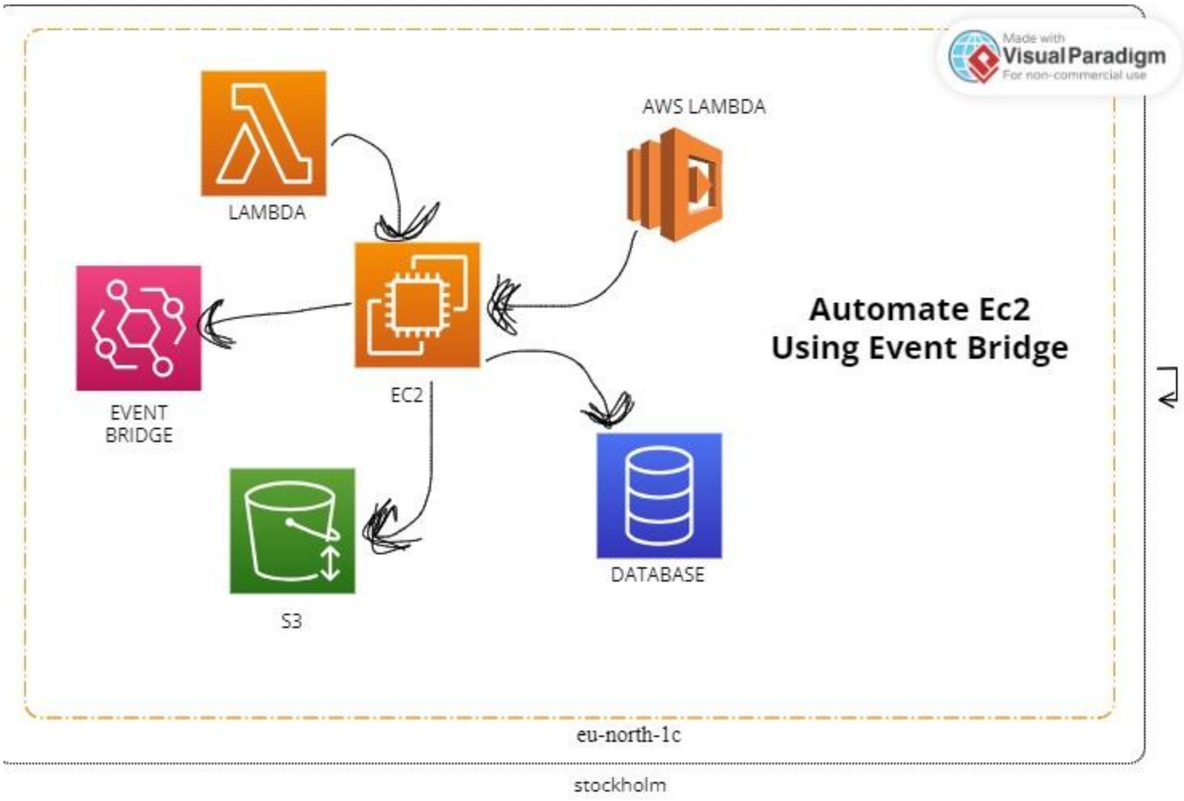
8.Check the instance2 is it successfully launched the website or not.



Coffee - Makes you Love

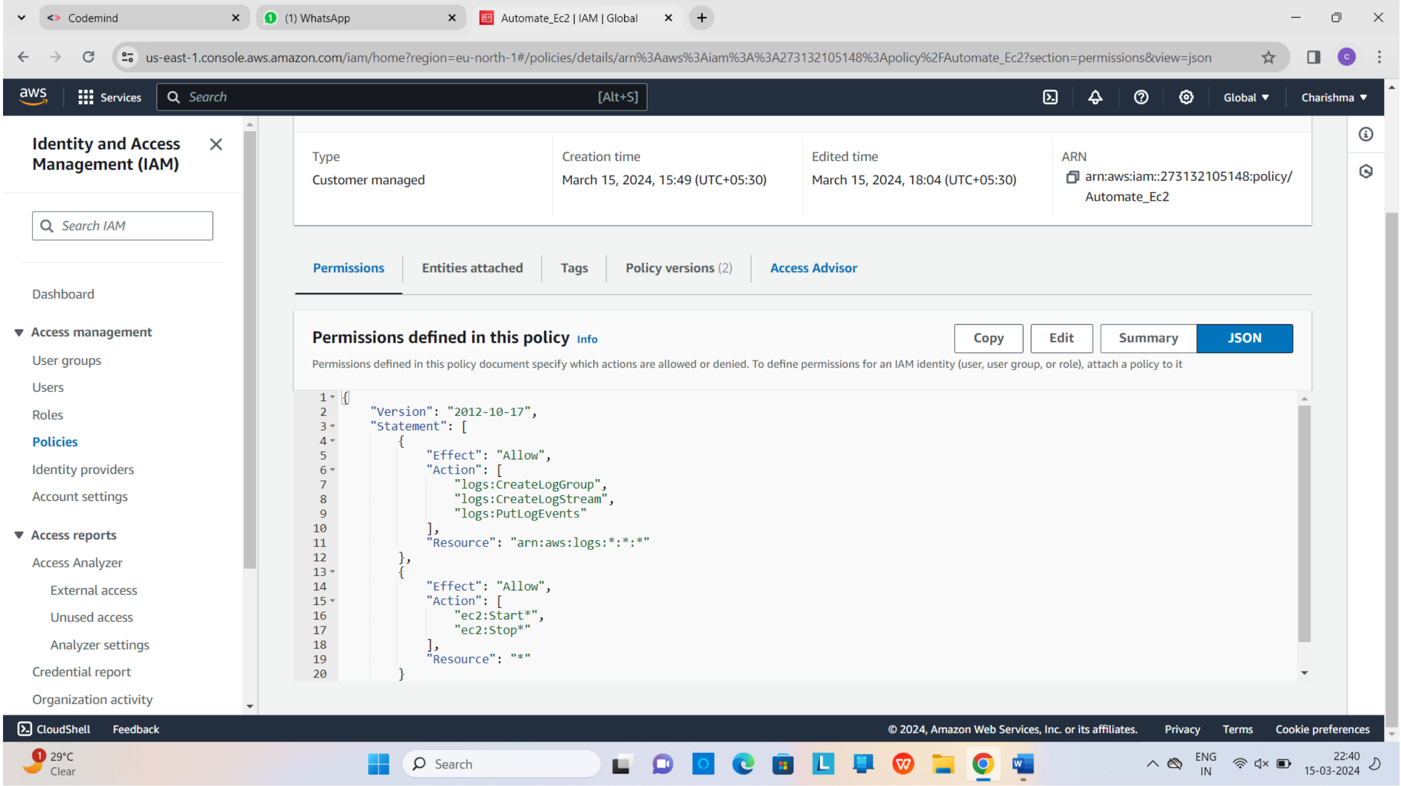
A Lot Can Happen Over Coffee, Its Awesome!!!

Step2: Automate the instances to start and stop at a particular time period


Made with Visual Paradigm For non-commercial use
LAMBDA
AWS LAMBDA
EVENT BRIDGE
EC2
Automate Ec2 Using Event Bridge
S3
DATABASE
eu-north-1c
stockholm

1.create a IAM policy

- json code as follows:



Type	Creation time	Edited time	ARN
Customer managed	March 15, 2024, 15:49 (UTC+05:30)	March 15, 2024, 18:04 (UTC+05:30)	arn:aws:iam::273132105148:policy/ Automate_Ec2

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Start*",
                "ec2:Stop*"
            ],
            "Resource": "*"
        }
```

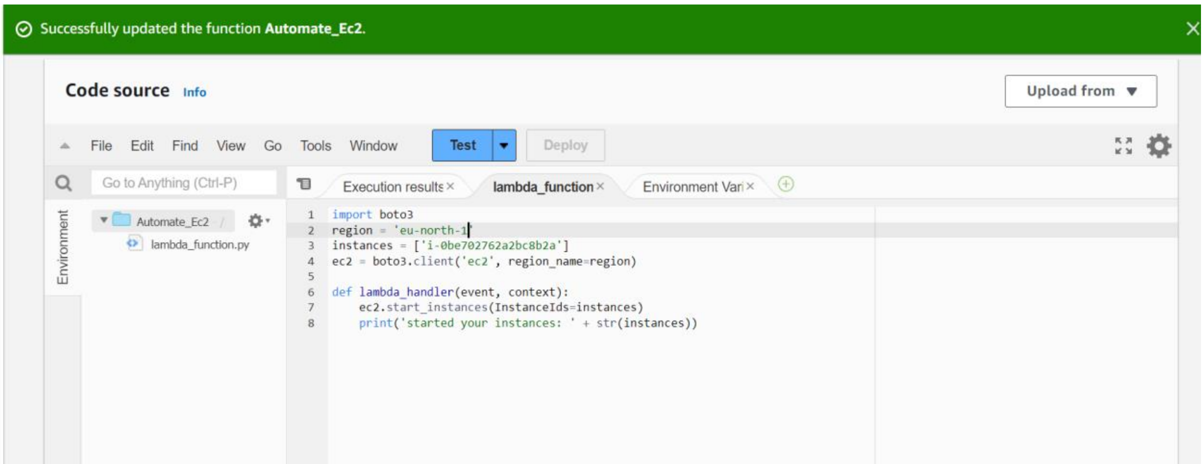
2.create role

-policy attached to our created role

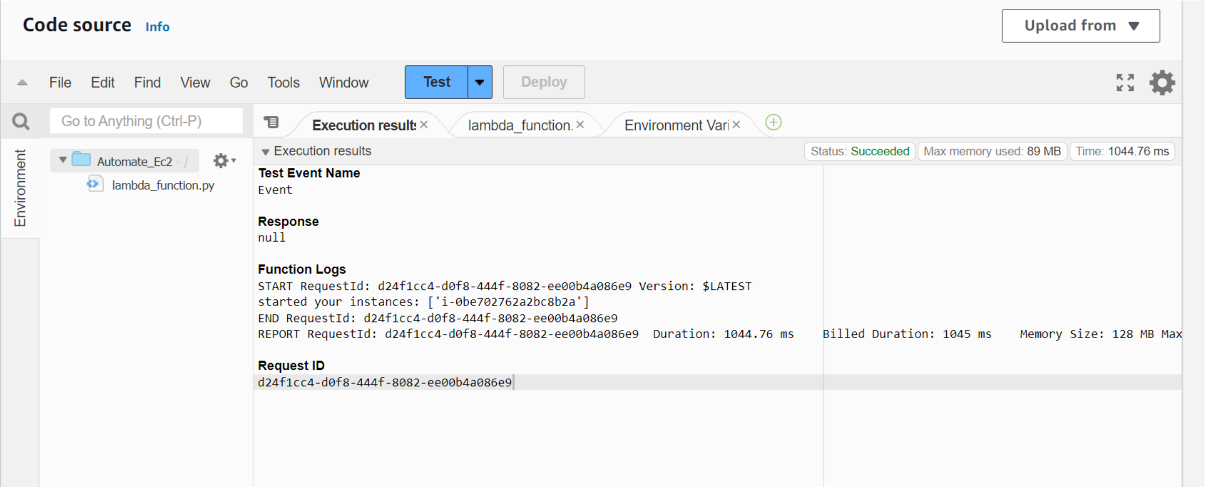
-role is created because we need permissions to access lambda functions

3.We use lambda to create functions as start and stop for the instances

1.start function

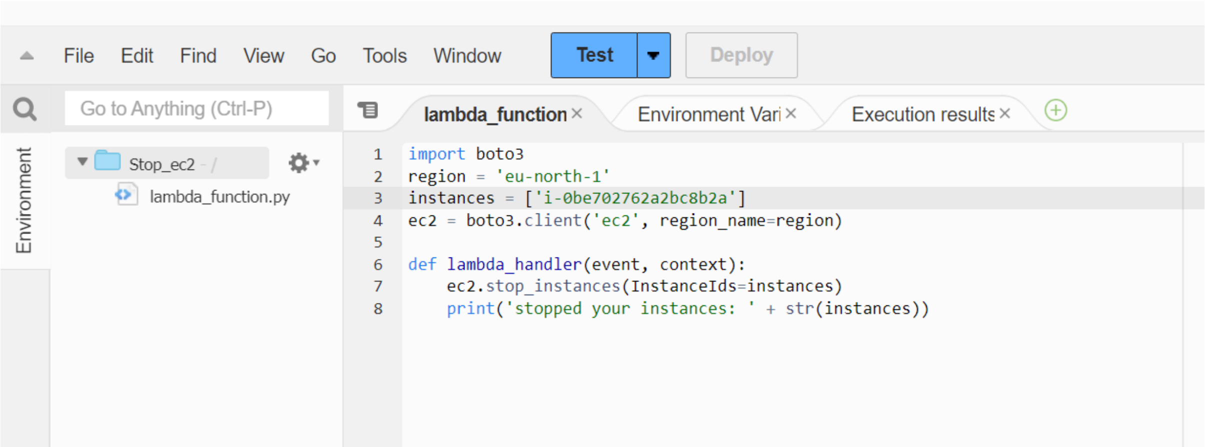


```python
import boto3
region = 'eu-north-1'
instances = ['i-0be702762a2bc8b2a']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

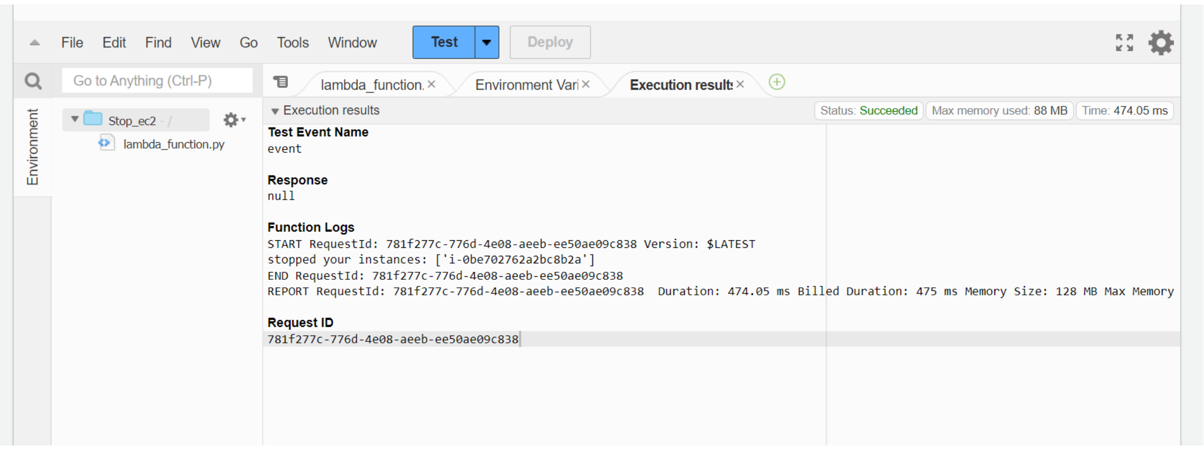


```
Execution results
Status: Succeeded    Max memory used: 89 MB    Time: 1044.76 ms

Test Event Name
Event

Response
null

Function Logs
START RequestId: d24f1cc4-d0f8-444f-8082-ee00b4a086e9 Version: $LATEST
started your instances: ['i-0be702762a2bc8b2a']
END RequestId: d24f1cc4-d0f8-444f-8082-ee00b4a086e9
REPORT RequestId: d24f1cc4-d0f8-444f-8082-ee00b4a086e9  Duration: 1044.76 ms    Billed Duration: 1045 ms    Memory Size: 128 MB Max

Request ID
d24f1cc4-d0f8-444f-8082-ee00b4a086e9
```

2.stop function



```python
import boto3
region = 'eu-north-1'
instances = ['i-0be702762a2bc8b2a']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```



Execution results

Status: Succeeded | Max memory used: 88 MB | Time: 474.05 ms

Test Event Name

```
event
```

Response

```
null
```

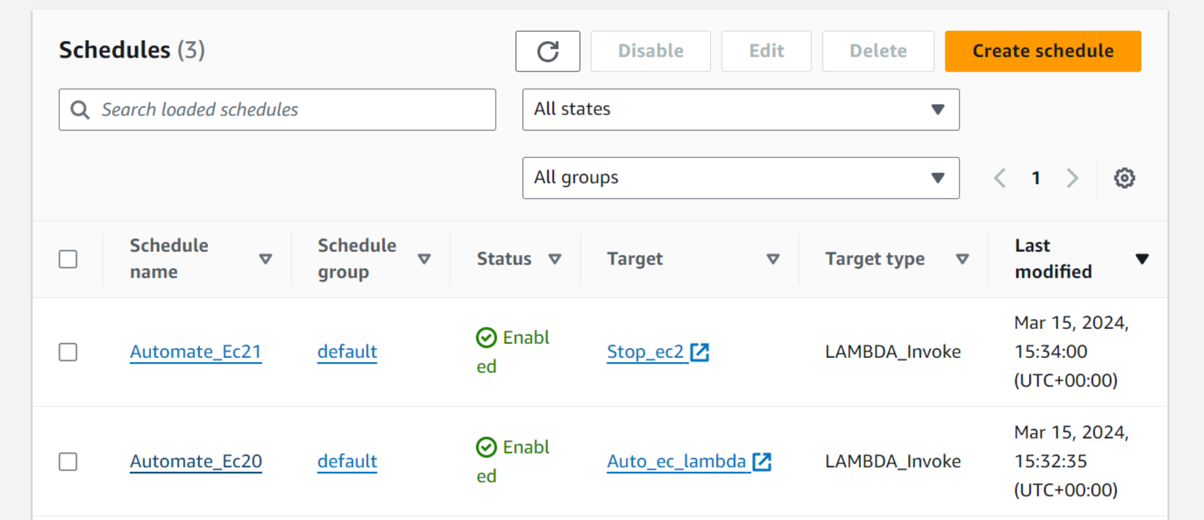
Function Logs

```
START RequestId: 781f277c-776d-4e08-aeeb-ee50ae09c838 Version: $LATEST
stopped your instances: ['i-0be702762a2bc8b2a']
END RequestId: 781f277c-776d-4e08-aeeb-ee50ae09c838
REPORT RequestId: 781f277c-776d-4e08-aeeb-ee50ae09c838  Duration: 474.05 ms Billed Duration: 475 ms Memory Size: 128 MB Max Memory
```

Request ID

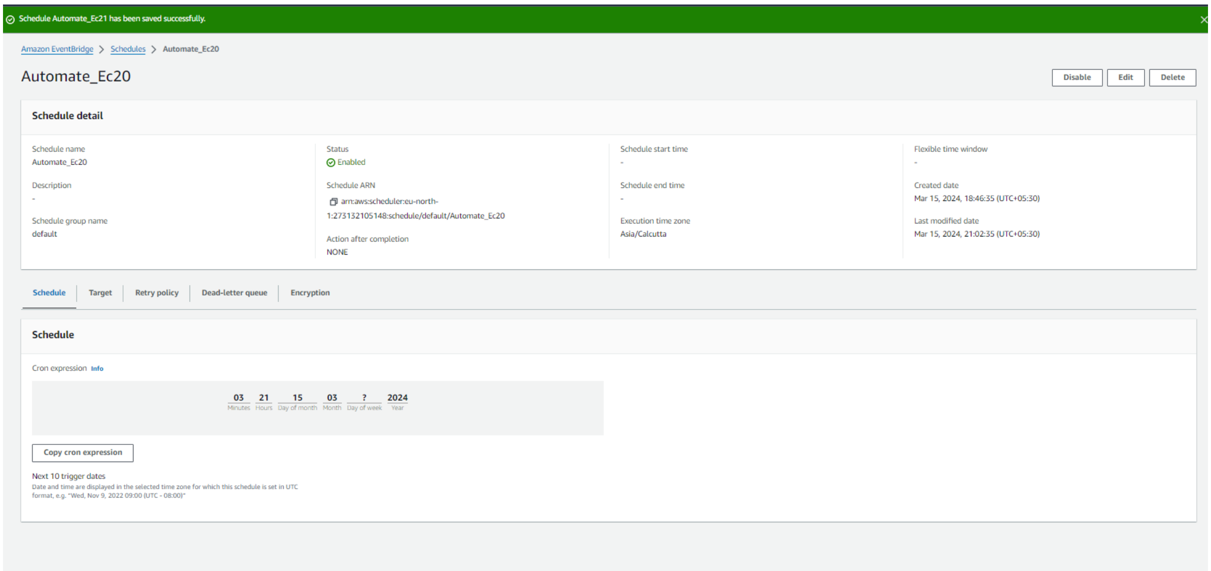
```
781f277c-776d-4e08-aeeb-ee50ae09c838
```

4.create schedules for the functions what we are created (Amazon event bridge)



Schedules (3)

□	Schedule name	Schedule group	Status	Target	Target type	Last modified
□	Automate_Ec21	default	✓ Enabled	Stop_ec2	LAMBDA_Invoke	Mar 15, 2024, 15:34:00 (UTC+00:00)
□	Automate_Ec20	default	✓ Enabled	Auto_ec_lambda	LAMBDA_Invoke	Mar 15, 2024, 15:32:35 (UTC+00:00)



Schedule Automate_Ec21 has been saved successfully.

Amazon EventBridge > Schedules > Automate_Ec20

Automate_Ec20

Schedule detail

Schedule name
Automate_Ec20

Description
-

Schedule group name
default

Status
✓ Enabled

Schedule ARN
arn:aws:scheduler:eu-north-1:273132105148:schedule/default/Automate_Ec20

Action after completion
NONE

Schedule start time
-

Schedule end time
-

Execution time zone
Asia/Calcutta

Flexible time window
-

Created date
Mar 15, 2024, 18:46:35 (UTC+05:30)

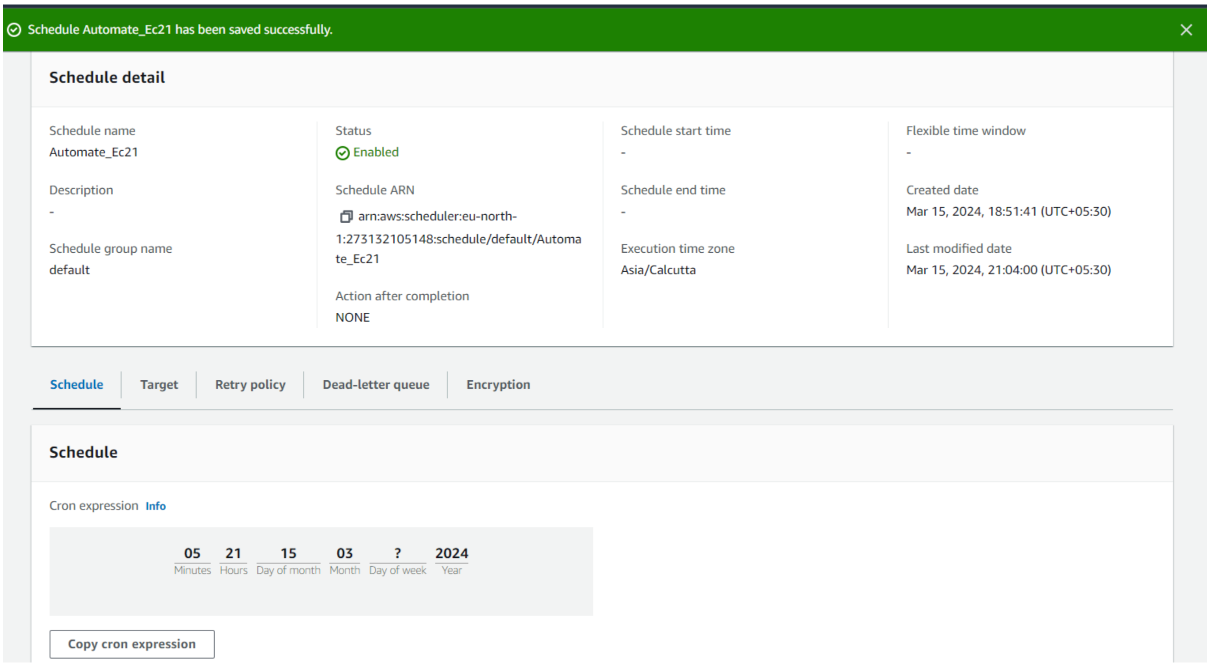
Last modified date
Mar 15, 2024, 21:02:35 (UTC+05:30)

Schedule

Cron expression Info

Minutes	Hours	Day of month	Month	Day of week	Year
03	21	15	03	?	2024

Next 10 trigger dates
Date and time are displayed in the selected time zone for which this schedule is set in UTC format, e.g. "Wed, Nov 9, 2022 09:00 (UTC - 08:00)"



Schedule Automate_Ec21 has been saved successfully.

Schedule detail

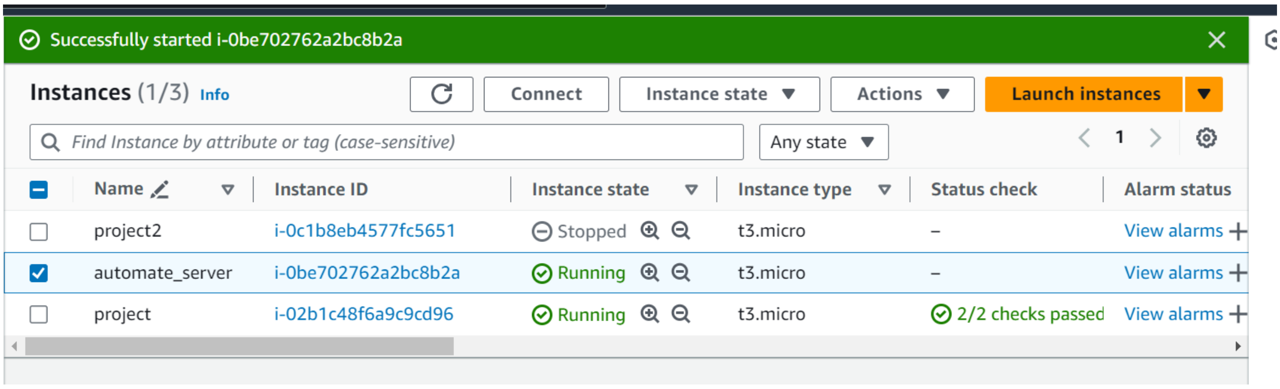
Schedule name Automate_Ec21	Status Enabled	Schedule start time -	Flexible time window -
Description -	Schedule ARN arn:aws:scheduler:eu-north-1:273132105148:schedule/default/Automate_Ec21	Schedule end time -	Created date Mar 15, 2024, 18:51:41 (UTC+05:30)
Schedule group name default		Execution time zone Asia/Calcutta	Last modified date Mar 15, 2024, 21:04:00 (UTC+05:30)
	Action after completion NONE		

Schedule

Cron expression Info

Minutes	Hours	Day of month	Month	Day of week	Year
05	21	15	03	?	2024

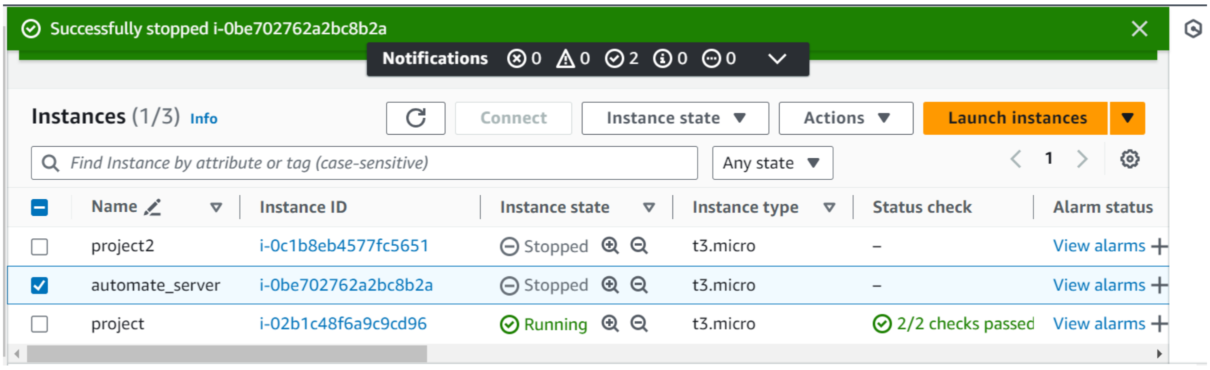
5.The instance are automated as per the schedule



Successfully started i-0be702762a2bc8b2a

Instances (1/3) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	project2	i-0c1b8eb4577fc5651	Stopped	t3.micro	–	View alarms +
☑	automate_server	i-0be702762a2bc8b2a	Running	t3.micro	–	View alarms +
□	project	i-02b1c48f6a9c9cd96	Running	t3.micro	2/2 checks passed	View alarms +



Successfully stopped i-0be702762a2bc8b2a

Notifications ⊗0 ⚠0 ✓2 ⓘ0 ⊖0

Instances (1/3) Info

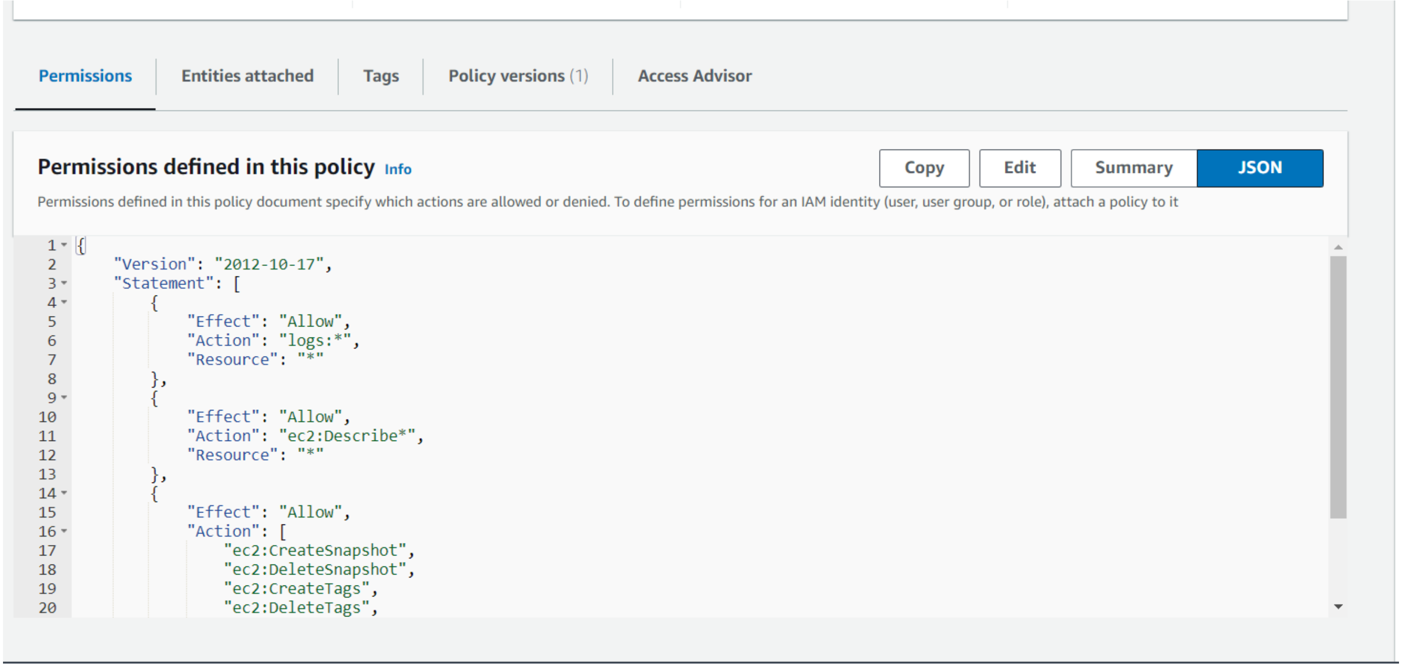
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
1

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	project2	i-0c1b8eb4577fc5651	Stopped	t3.micro	–	View alarms +
☑	automate_server	i-0be702762a2bc8b2a	Stopped	t3.micro	–	View alarms +
□	project	i-02b1c48f6a9c9cd96	Running	t3.micro	2/2 checks passed	View alarms +

Step 3: Automate snapshot creation for instances

1.create a IAM policy

-Json code as follows:



Permissions defined in this policy

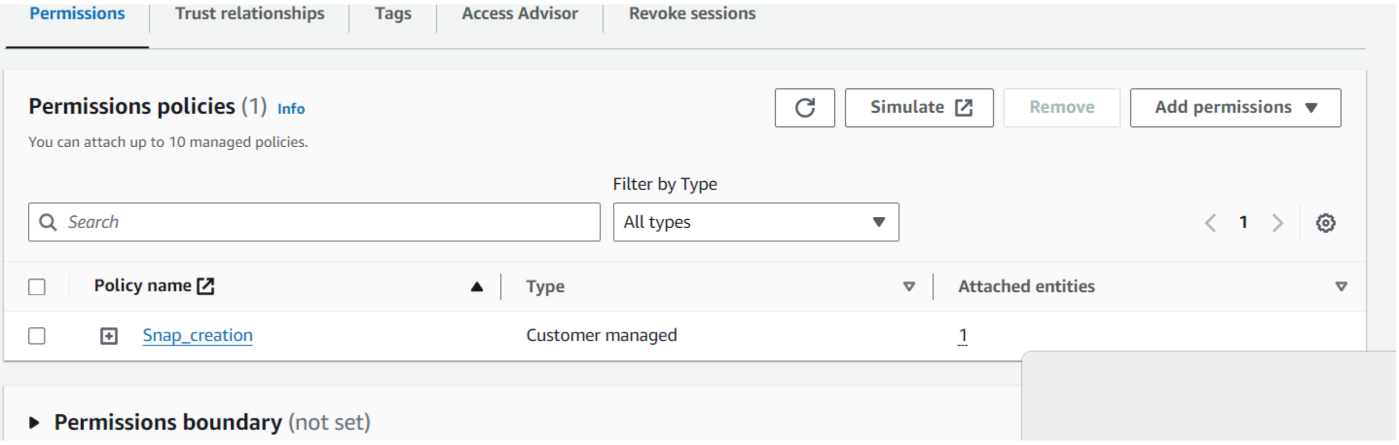
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:DeleteSnapshot",
                "ec2:CreateTags",
                "ec2:DeleteTags",
```

2.create role

-policy attached to our created role

-role is created because we need permissions to access lambda function



Permissions policies (1)

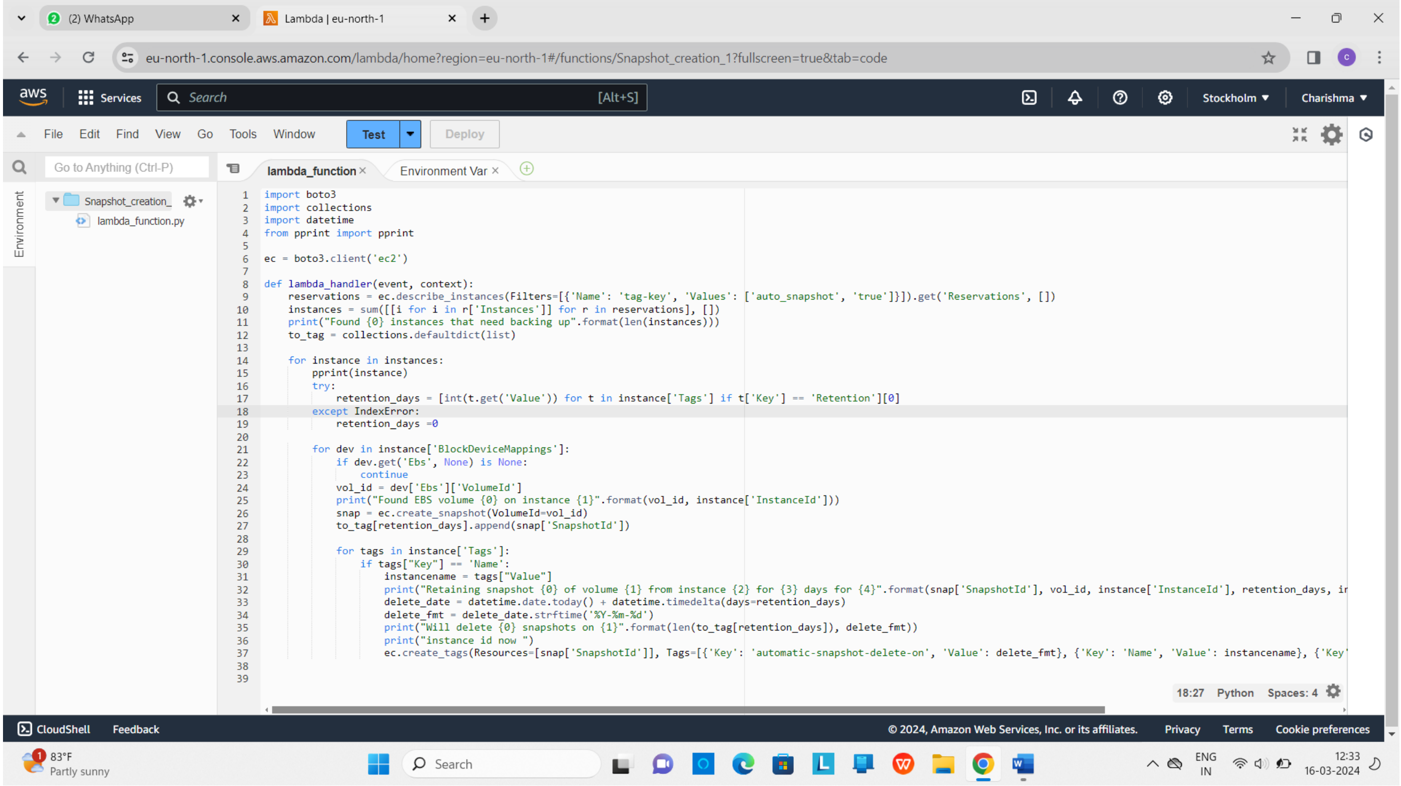
You can attach up to 10 managed policies.

Filter by Type

□	Policy name	Type	Attached entities
□	⊞ Snap_creation	Customer managed	1

▸ Permissions boundary (not set)

3.lambda function for snapshot creation

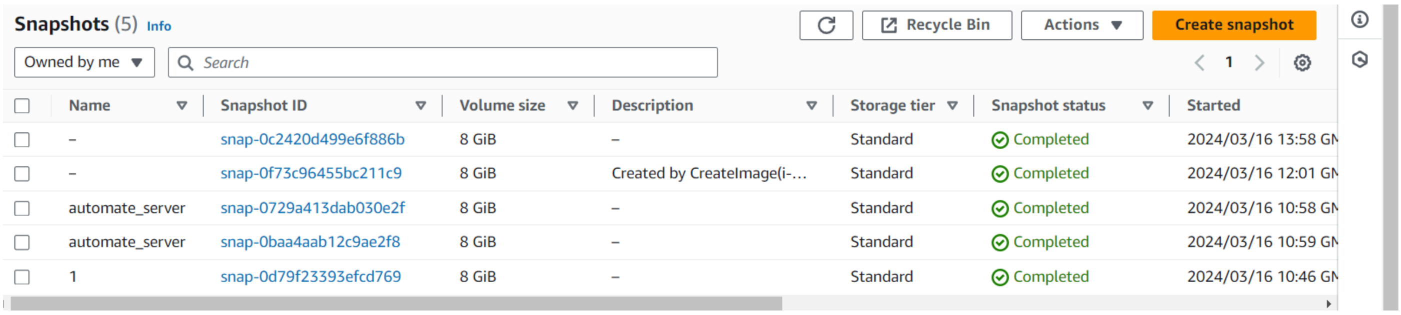


```python
import boto3
import collections
import datetime
from pprint import pprint

ec = boto3.client('ec2')

def lambda_handler(event, context):
    reservations = ec.describe_instances(Filters=[{'Name': 'tag-key', 'Values': ['auto_snapshot', 'true']}]).get('Reservations', [])
    instances = sum([[i for i in r['Instances']] for r in reservations], [])
    print("Found {0} instances that need backing up".format(len(instances)))
    to_tag = collections.defaultdict(list)

    for instance in instances:
        pprint(instance)
        try:
            retention_days = [int(t.get('Value')) for t in instance['Tags'] if t['Key'] == 'Retention'][0]
        except IndexError:
            retention_days =0

        for dev in instance['BlockDeviceMappings']:
            if dev.get('Ebs', None) is None:
                continue
            vol_id = dev['Ebs']['VolumeId']
            print("Found EBS volume {0} on instance {1}".format(vol_id, instance['InstanceId']))
            snap = ec.create_snapshot(VolumeId=vol_id)
            to_tag[retention_days].append(snap['SnapshotId'])

            for tags in instance['Tags']:
                if tags["Key"] == 'Name':
                    instancename = tags["Value"]
                    print("Retaining snapshot {0} of volume {1} from instance {2} for {3} days for {4}".format(snap['SnapshotId'], vol_id, instance['InstanceId'], retention_days, i
                    delete_date = datetime.date.today() + datetime.timedelta(days=retention_days)
                    delete_fmt = delete_date.strftime('%Y-%m-%d')
                    print("Will delete {0} snapshots on {1}".format(len(to_tag[retention_days]), delete_fmt))
                    print("instance id now ")
                    ec.create_tags(Resources=[snap['SnapshotId']], Tags=[{'Key': 'automatic-snapshot-delete-on', 'Value': delete_fmt}, {'Key': 'Name', 'Value': instancename}, {'Key'
```

-give the instance tag in code

4.Schedule to create snapshot for instances (cron expression)

5.As per the schedule it creates snapshot for instances



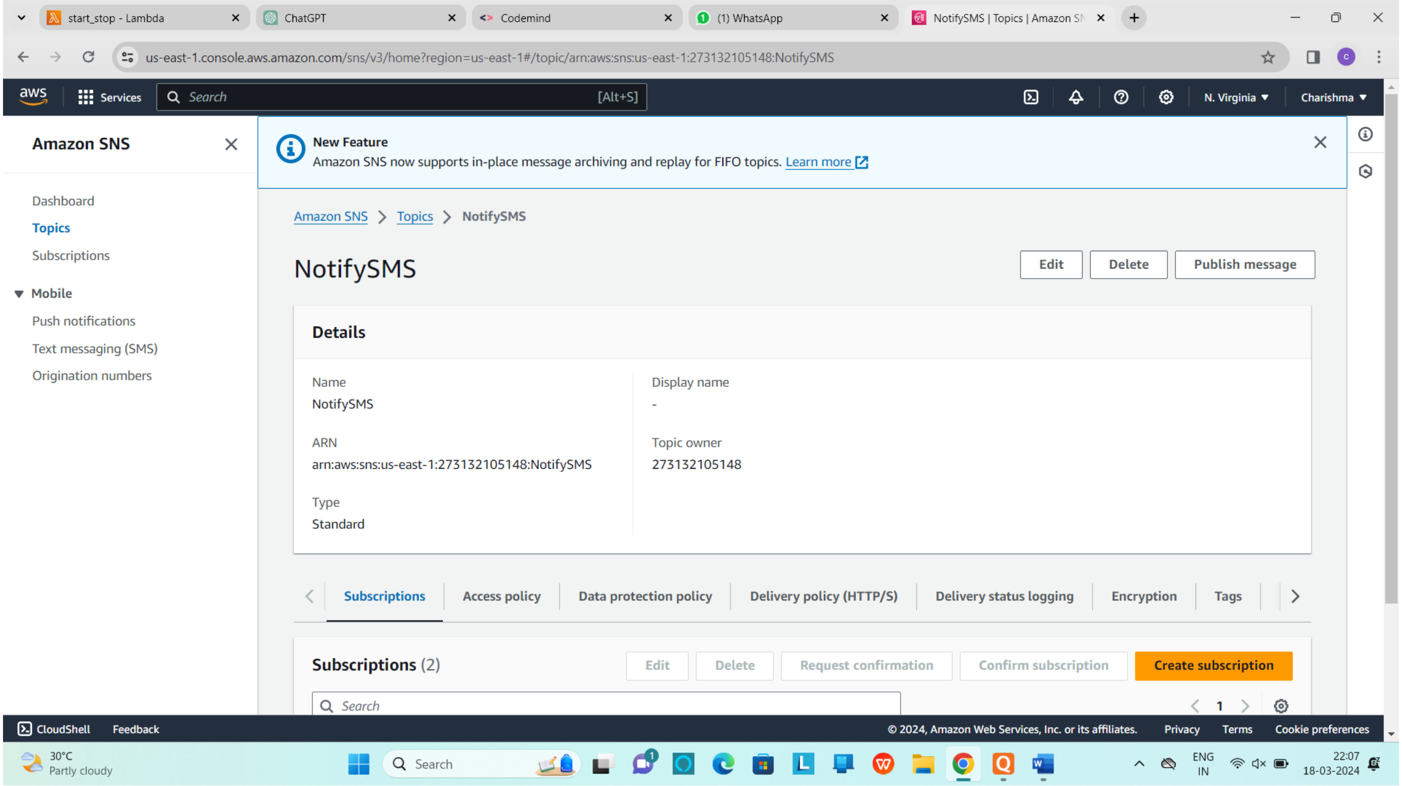
□	Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started
□	–	snap-0c2420d499e6f886b	8 GiB	–	Standard	✓ Completed	2024/03/16 13:58 GN
□	–	snap-0f73c96455bc211c9	8 GiB	Created by CreateImage(i-...	Standard	✓ Completed	2024/03/16 12:01 GN
□	automate_server	snap-0729a413dab030e2f	8 GiB	–	Standard	✓ Completed	2024/03/16 10:58 GN
□	automate_server	snap-0baa4aab12c9ae2f8	8 GiB	–	Standard	✓ Completed	2024/03/16 10:59 GN
□	1	snap-0d79f23393efcd769	8 GiB	–	Standard	✓ Completed	2024/03/16 10:46 GN

Step 4: Send a message notification if any actions trigged in our AWS environment

-message notification for44 ec2 actions

1.create a topic in SNS

-‘Topics’ serves as a communication channel for publishing messages to subscribers



New Feature

Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more

NotifySMS

Details

Name

NotifySMS

Display name

-

ARN

arn:aws:sns:us-east-1:273132105148:NotifySMS

Topic owner

273132105148

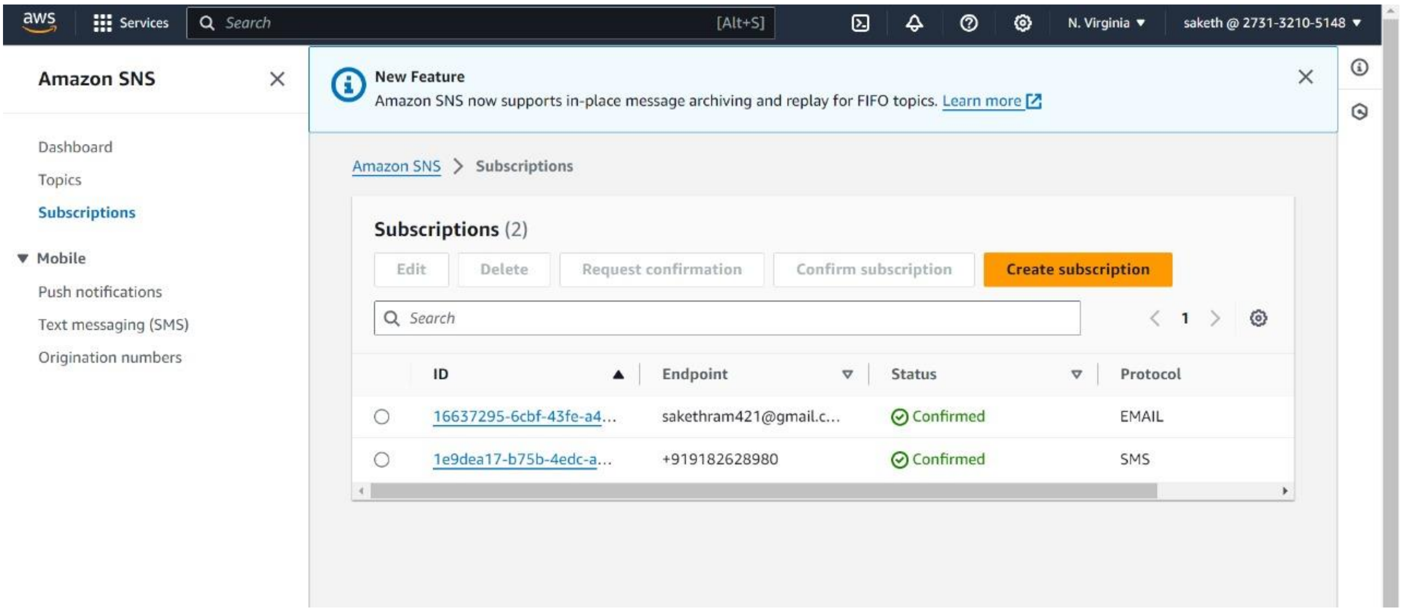
Type

Standard

Subscriptions (2)

2.create subscription

- Creating subscription allows us to specify how and where we want to deliver messages published to a topic.



New Feature

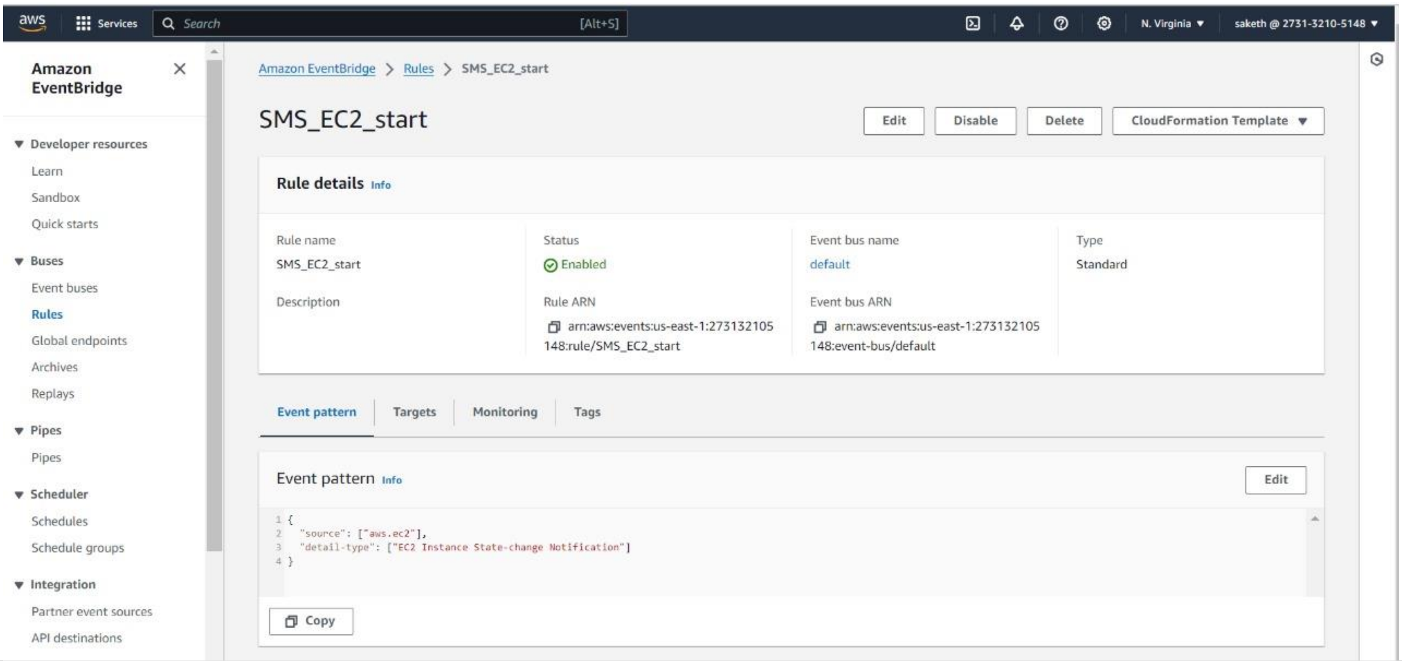
Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more

Subscriptions (2)

	ID	Endpoint	Status	Protocol
○	16637295-6cbf-43fe-a4...	sakethram421@gmail.c...	Confirmed	EMAIL
○	1e9dea17-b75b-4edc-a...	+919182628980	Confirmed	SMS

3.Creating rules in cloudwatch

-Rules enables to define conditions that trigger automated actions in response to events detected in our AWS environment.



SMS_EC2_start

Rule details Info

Rule name: SMS_EC2_start

Description:

Status: Enabled

Rule ARN: arn:aws:events:us-east-1:273132105148:rule/SMS_EC2_start

Event bus name: default

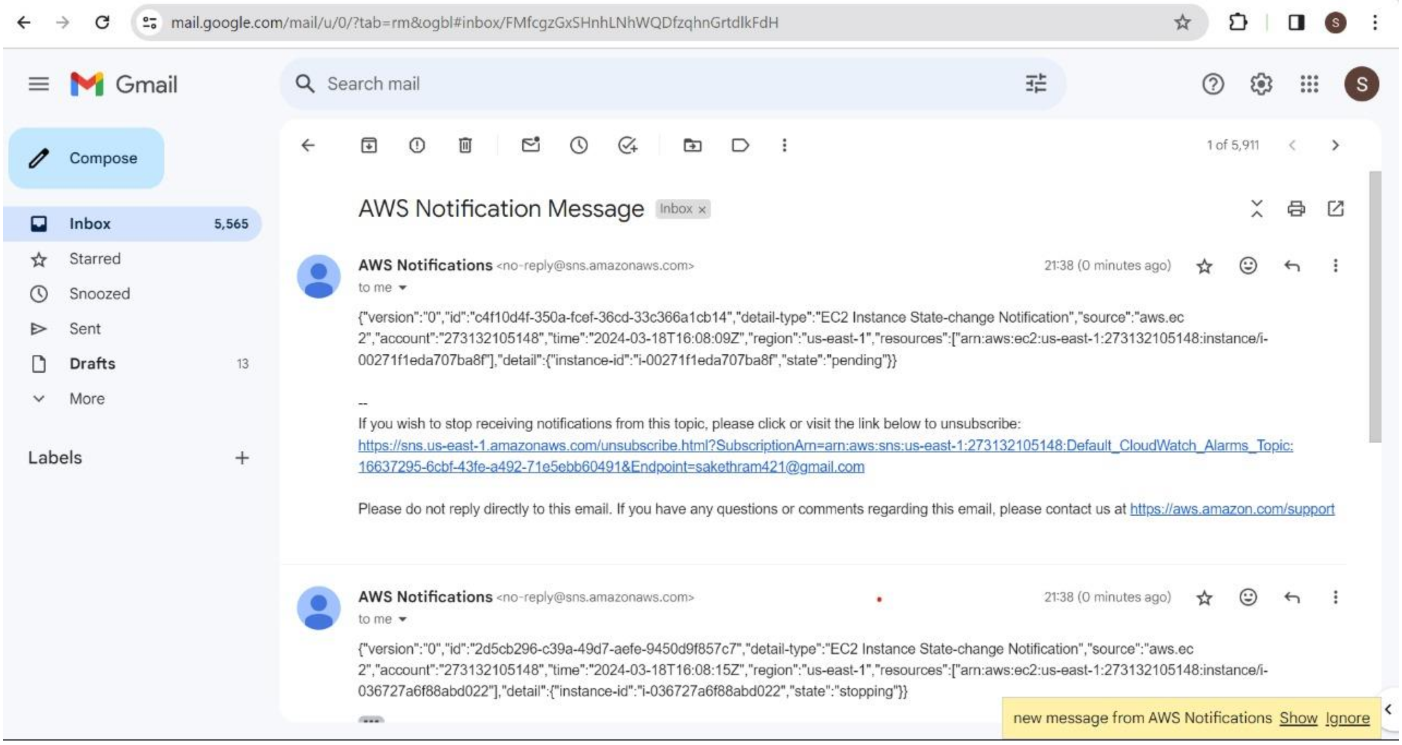
Event bus ARN: arn:aws:events:us-east-1:273132105148:event-bus/default

Type: Standard

Event pattern Info

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

4.If any actions are trigger in our environment a notification message sent to published topic



AWS Notification Message Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com> 21:38 (0 minutes ago)
to me

{"version":"0","id":"c4f10d4f-350a-fcef-36cd-33c366a1cb14","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"273132105148","time":"2024-03-18T16:08:09Z","region":"us-east-1","resources":["arn:aws:ec2:us-east-1:273132105148:instance/i-00271f1eda707ba8f"],"detail":{"instance-id":"i-00271f1eda707ba8f","state":"pending"}}

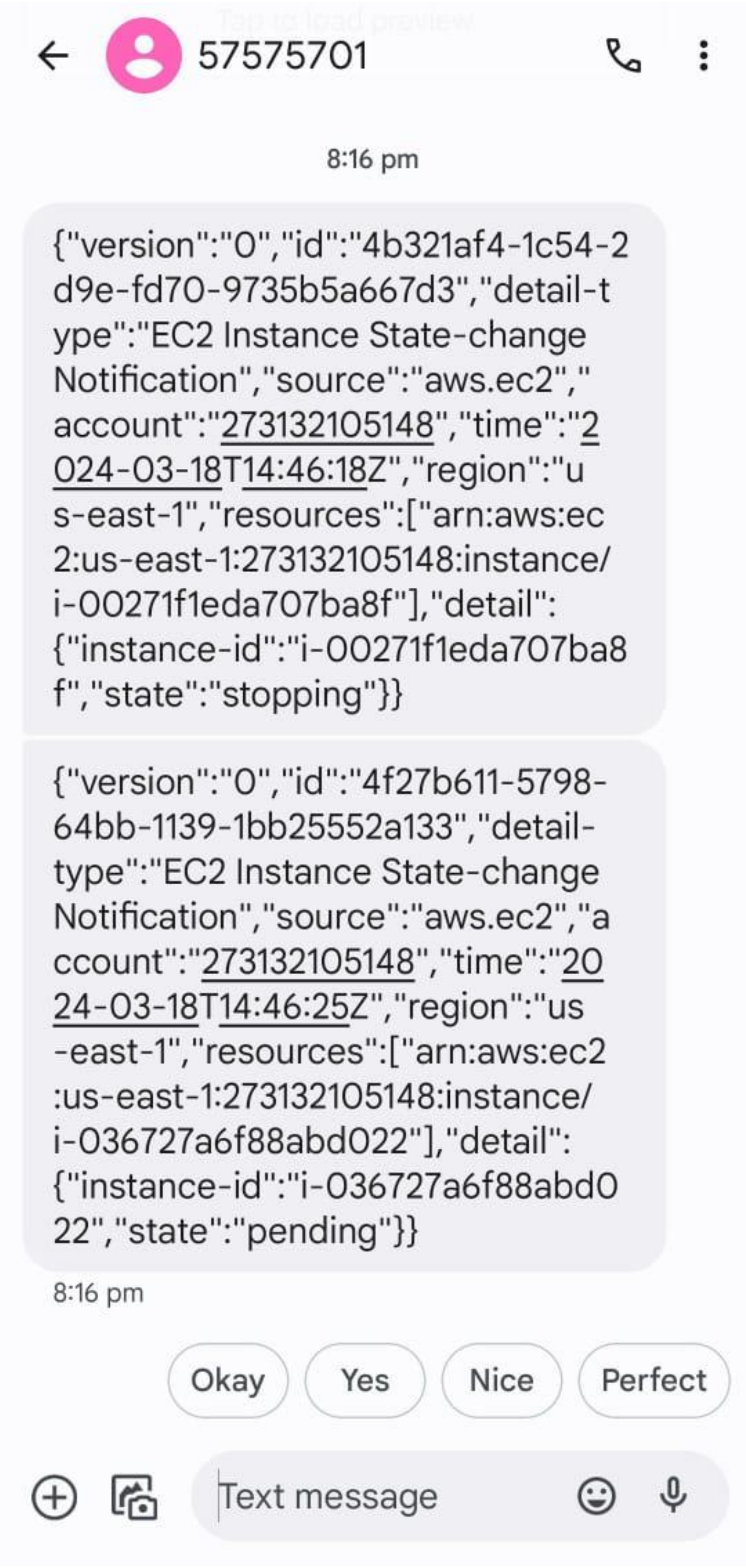
--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:273132105148:Default_CloudWatch_Alarms_Topic:16637295-6cbf-43fe-a492-71e5ebb60491&Endpoint=sakethram421@gmail.com

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

AWS Notifications <no-reply@sns.amazonaws.com> 21:38 (0 minutes ago)
to me

{"version":"0","id":"2d5cb296-c39a-49d7-aefe-9450d9f857c7","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"273132105148","time":"2024-03-18T16:08:15Z","region":"us-east-1","resources":["arn:aws:ec2:us-east-1:273132105148:instance/i-036727a6f88abd022"],"detail":{"instance-id":"i-036727a6f88abd022","state":"stopping"}}

new message from AWS Notifications Show Ignore



57575701

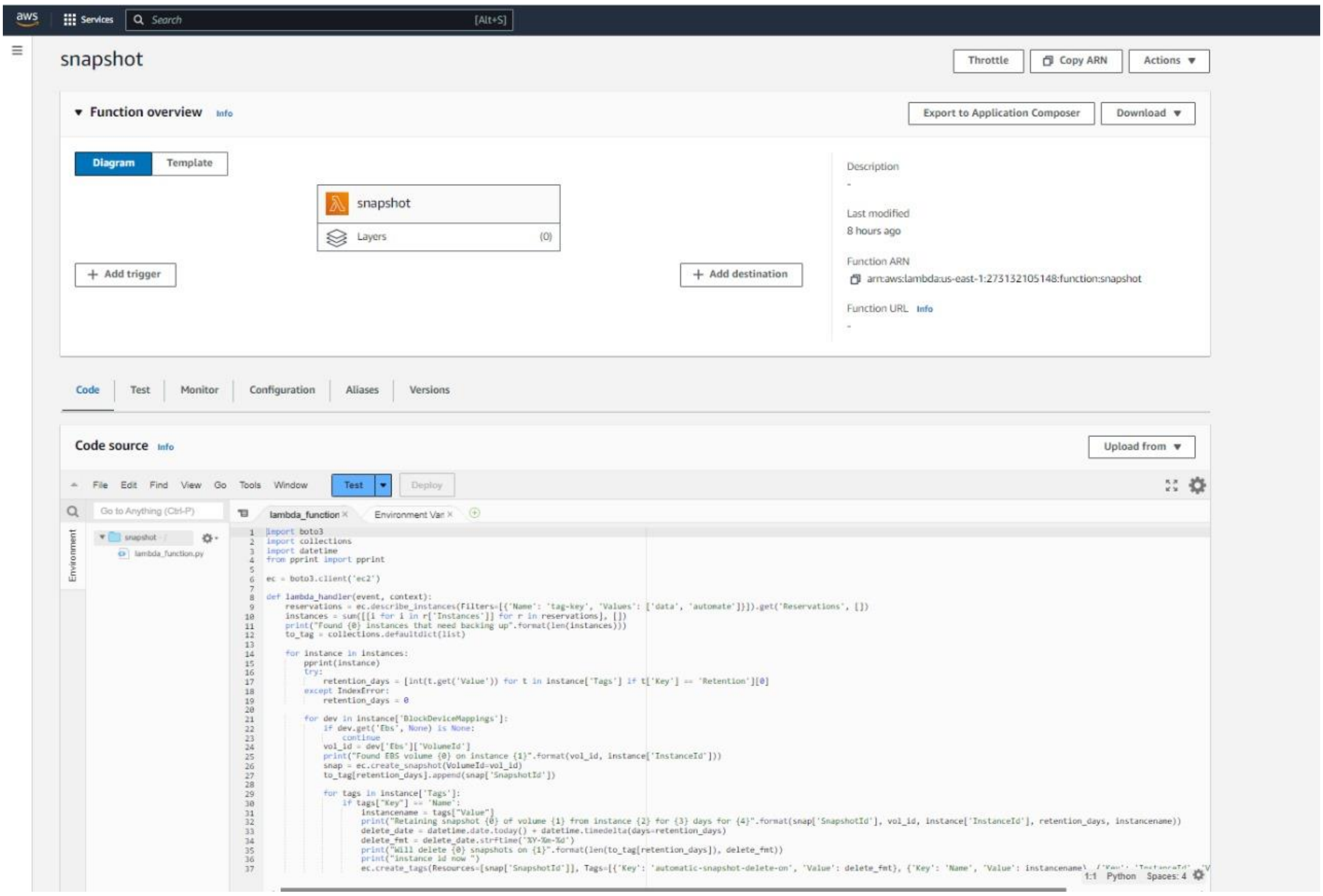
8:16 pm

{"version":"0","id":"4b321af4-1c54-2d9e-fd70-9735b5a667d3","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"273132105148","time":"2024-03-18T14:46:18Z","region":"us-east-1","resources":["arn:aws:ec2:us-east-1:273132105148:instance/i-00271f1eda707ba8f"],"detail":{"instance-id":"i-00271f1eda707ba8f","state":"stopping"}}

{"version":"0","id":"4f27b611-5798-64bb-1139-1bb25552a133","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"273132105148","time":"2024-03-18T14:46:25Z","region":"us-east-1","resources":["arn:aws:ec2:us-east-1:273132105148:instance/i-036727a6f88abd022"],"detail":{"instance-id":"i-036727a6f88abd022","state":"pending"}}

8:16 pm

-message notification for backup action



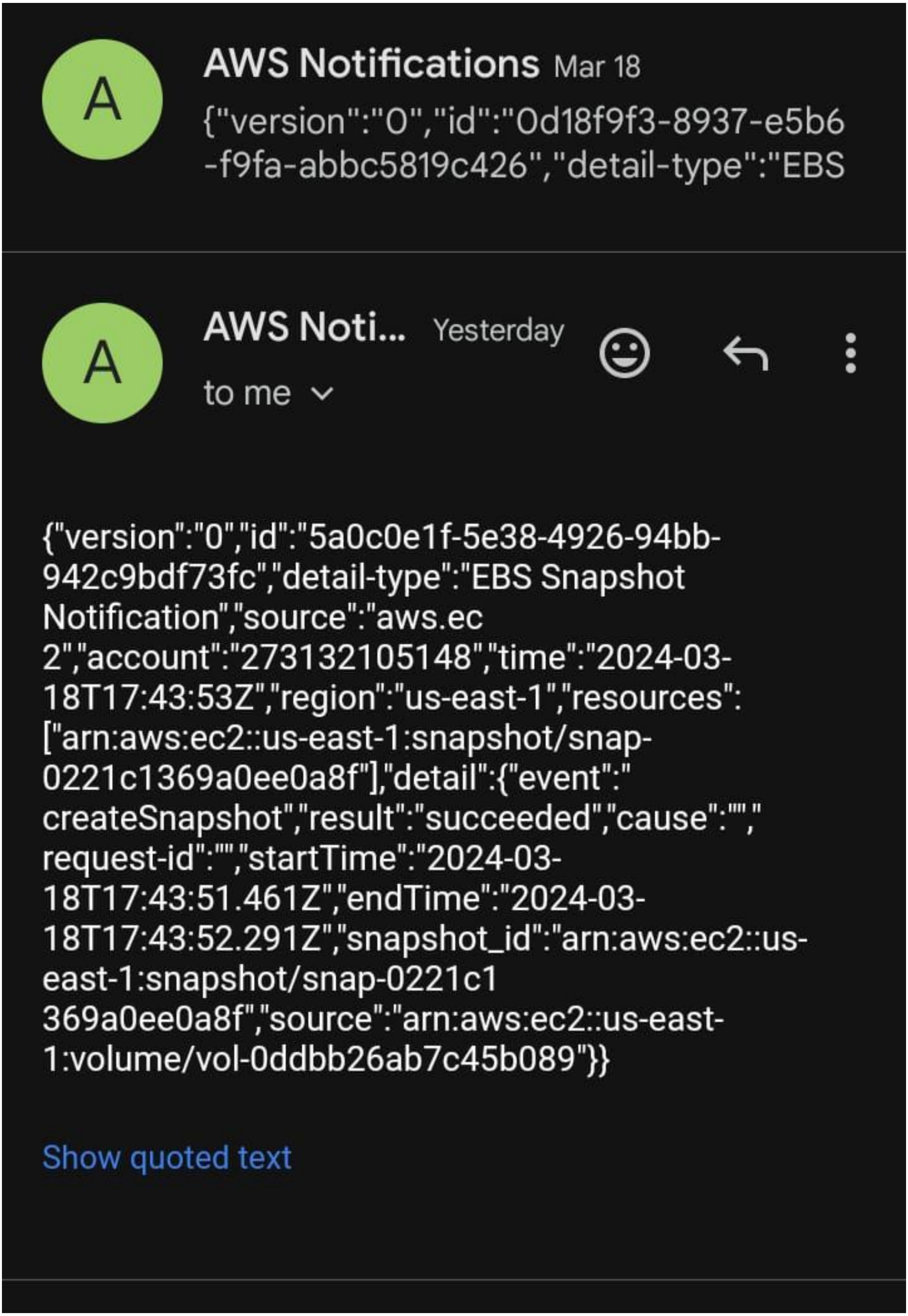
Description
-

Last modified
8 hours ago

Function ARN
arn:aws:lambda:us-east-1:273132105148:function:snapshot

Function URL Info
-

```python
import boto3
import collections
import datetime
from pprint import pprint

ec = boto3.client('ec2')

def lambda_handler(event, context):
    reservations = ec.describe_instances(Filters=[{'Name': 'tag-key', 'Values': ['data', 'automate']}]).get('Reservations', [])
    instances = sum([[i for i in r['Instances']] for r in reservations], [])
    print("Found {0} instances that need backing up".format(len(instances)))
    to_tag = collections.defaultdict(list)

    for instance in instances:
        pprint(instance)
        try:
            retention_days = [int(t.get('Value')) for t in instance['Tags'] if t['Key'] == 'Retention'][0]
        except IndexError:
            retention_days = 0

        for dev in instance['BlockDeviceMappings']:
            if dev.get('Ebs', None) is None:
                continue
            vol_id = dev['Ebs']['VolumeId']
            print("Found EBS volume {0} on instance {1}".format(vol_id, instance['InstanceId']))
            snap = ec.create_snapshot(VolumeId=vol_id)
            to_tag[retention_days].append(snap['SnapshotId'])

            for tags in instance['Tags']:
                if tags["Key"] == 'Name':
                    instancename = tags["Value"]
                    print("Retaining snapshot {0} of volume {1} from instance {2} for {3} days for {4}".format(snap['SnapshotId'], vol_id, instance['InstanceId'], retention_days, instancename))
                    delete_date = datetime.date.today() + datetime.timedelta(days=retention_days)
                    delete_fmt = delete_date.strftime('%Y-%m-%d')
                    print("Will delete {0} snapshots on {1}".format(len(to_tag[retention_days]), delete_fmt))
                    print("instance id now ")
                    ec.create_tags(Resources=[snap['SnapshotId']], Tags=[{'Key': 'automatic-snapshot-delete-on', 'Value': delete_fmt}, {'Key': 'Name', 'Value': instancename}, {'Key': 'InstanceId', 'V
```



A

AWS Notifications Mar 18

{"version":"0","id":"0d18f9f3-8937-e5b6-f9fa-abbc5819c426","detail-type":"EBS

A

AWS Noti... Yesterday

to me

{"version":"0","id":"5a0c0e1f-5e38-4926-94bb-942c9bdf73fc","detail-type":"EBS Snapshot Notification","source":"aws.ec2","account":"273132105148","time":"2024-03-18T17:43:53Z","region":"us-east-1","resources":["arn:aws:ec2::us-east-1:snapshot/snap-0221c1369a0ee0a8f"],"detail":{"event":"createSnapshot","result":"succeeded","cause":"","request-id":"","startTime":"2024-03-18T17:43:51.461Z","endTime":"2024-03-18T17:43:52.291Z","snapshot_id":"arn:aws:ec2::us-east-1:snapshot/snap-0221c1369a0ee0a8f","source":"arn:aws:ec2::us-east-1:volume/vol-0ddbb26ab7c45b089"}}

Show quoted text