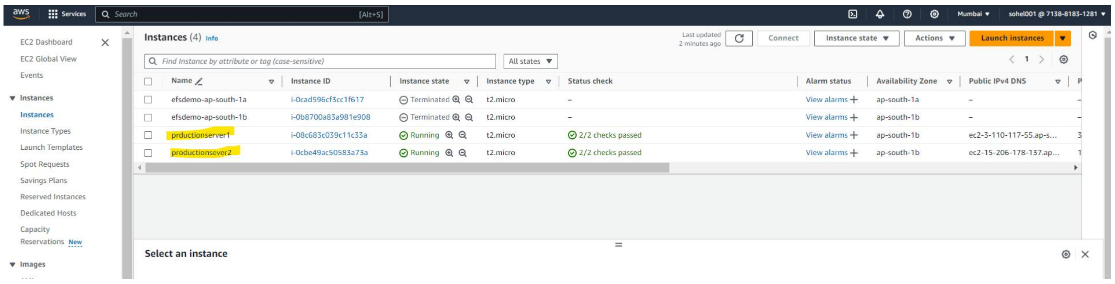
Lambda Practical 29/08/2024

Sohel Pathan

1. Create two instances name prductionserver1 and productionserver2

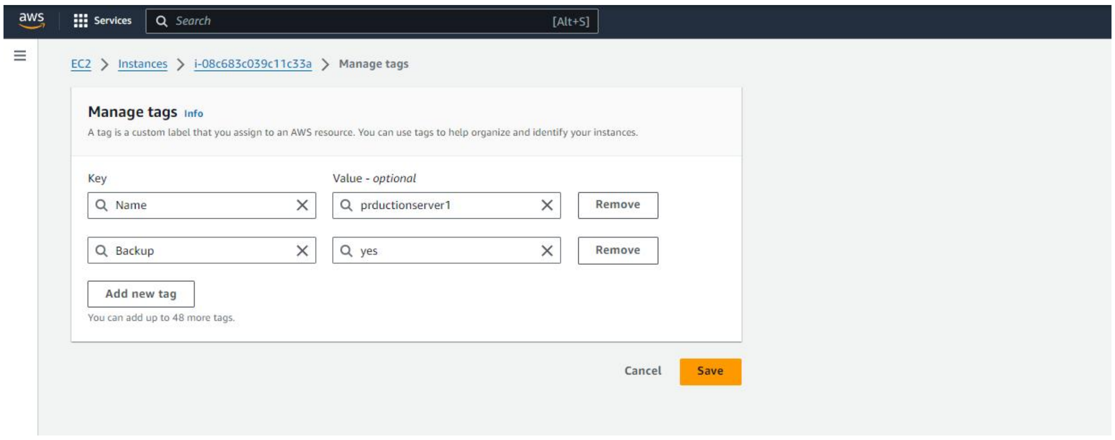


Instances (4)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
efsdemo-ap-south-1a	i-0cad596cf3cc1f617	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
efsdemo-ap-south-1b	i-0b8700a83a981e908	Terminated	t2.micro	–	View alarms +	ap-south-1b	–
prductionserver1	i-08c683c039c11c33a	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-3-110-117-55.ap-s...
productionsever2	i-0cbe49ac50583a73a	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-15-206-178-137.ap...

Select an instance

2. Add tag on one instance (prductionserver1)



EC2 > Instances > i-08c683c039c11c33a > Manage tags

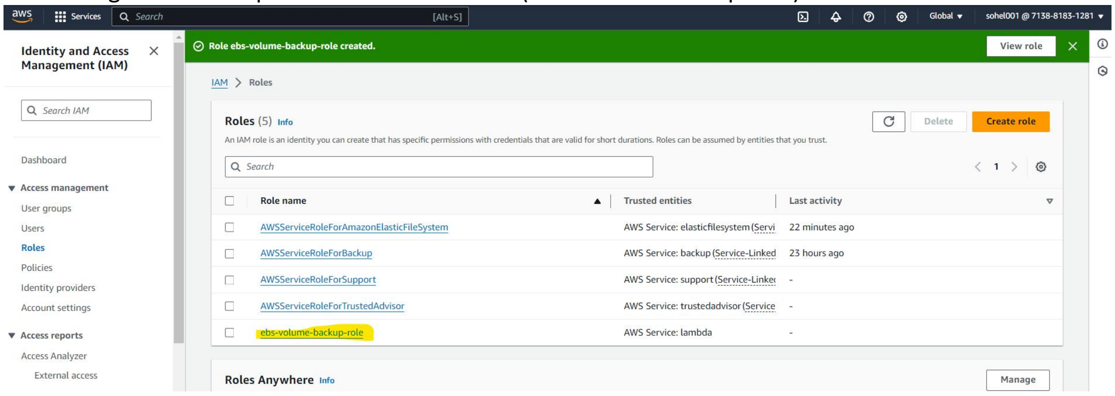
Manage tags

A tag is a custom label that you assign to an AWS resource. You can use tags to help organize and identify your instances.

Key	Value - optional
Name	prductionserver1
Backup	yes

You can add up to 48 more tags.

3. Creating IAM role to perform **Lambda** service(ebs-volume-backup-role)



Role ebs-volume-backup-role created.

IAM > Roles

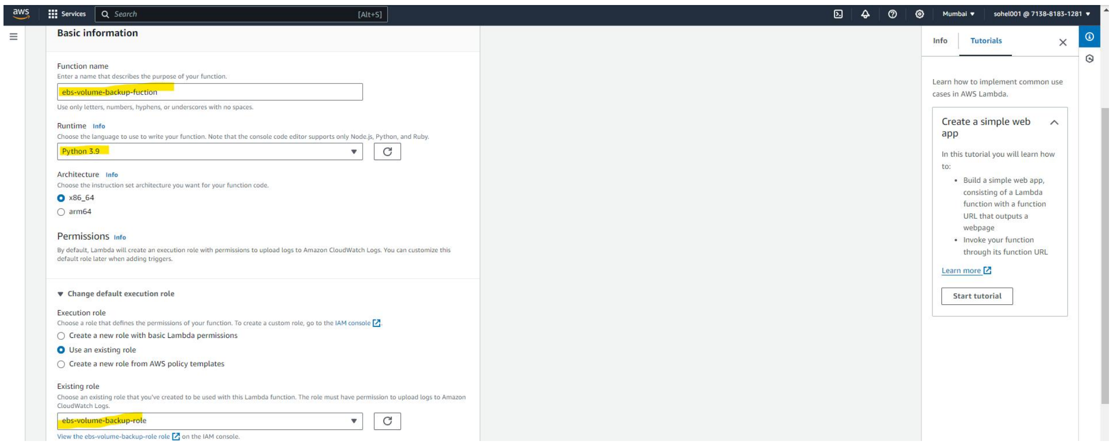
Roles (5)

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Servi	22 minutes ago
AWSServiceRoleForBackup	AWS Service: backup (Service-Linked	23 hours ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
ebs-volume-backup-role	AWS Service: lambda	-

Roles Anywhere

4. While creating Lambda function select any language add our role(ebs-volume-backup-role)



Basic information

Function name
Enter a name that describes the purpose of your function.
ebs-volume-backup-fuction
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 3.9

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ x86_64] arm64

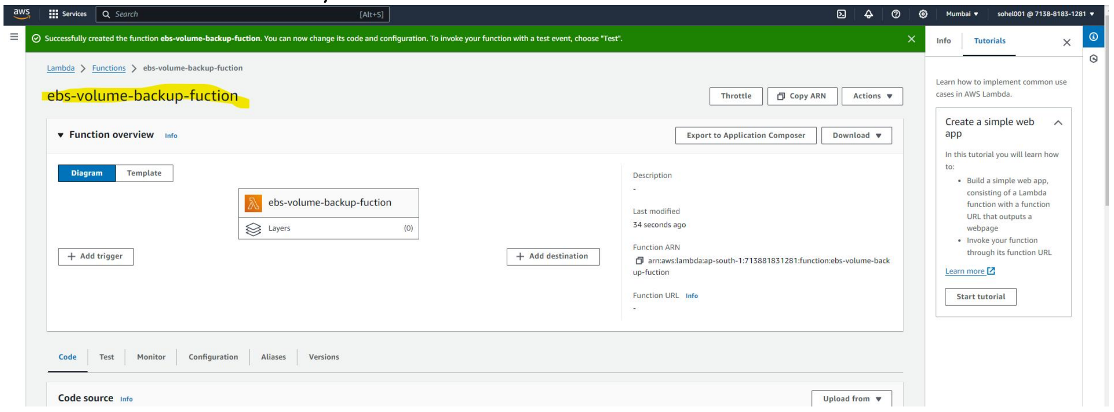
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

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.
ebs-volume-backup-role
View the ebs-volume-backup-role role on the IAM console.

5. Fuction created successfully



Successfully created the function **ebs-volume-backup-fuction**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

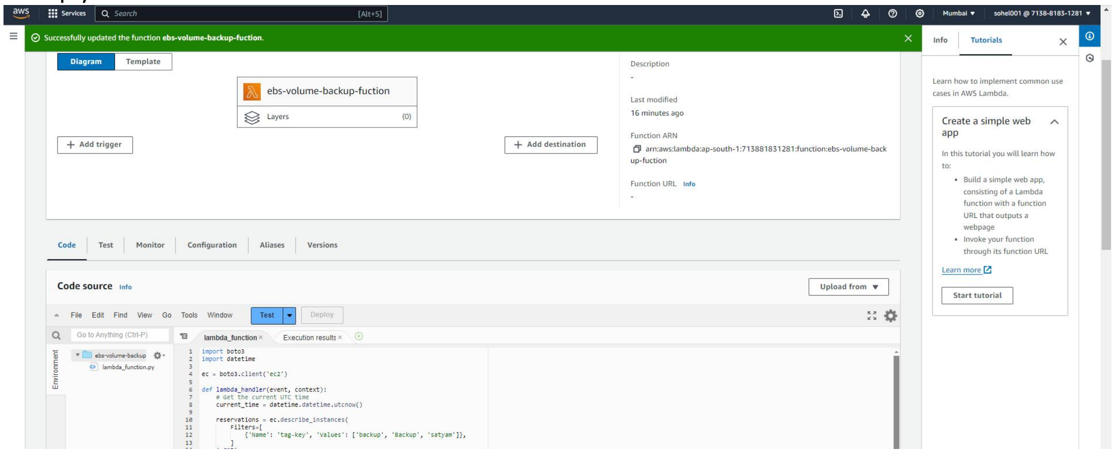
Lambda > Functions > ebs-volume-backup-fuction

ebs-volume-backup-fuction

Function overview

Description
-
Last modified
34 seconds ago
Function ARN
arn:aws:lambda:ap-south-1:713881831281:function:ebs-volume-backup-fuction
Function URL
-

6. Deply the code



Successfully updated the function **ebs-volume-backup-fuction**.

Description
-
Last modified
16 minutes ago
Function ARN
arn:aws:lambda:ap-south-1:713881831281:function:ebs-volume-backup-fuction
Function URL
-

Code source

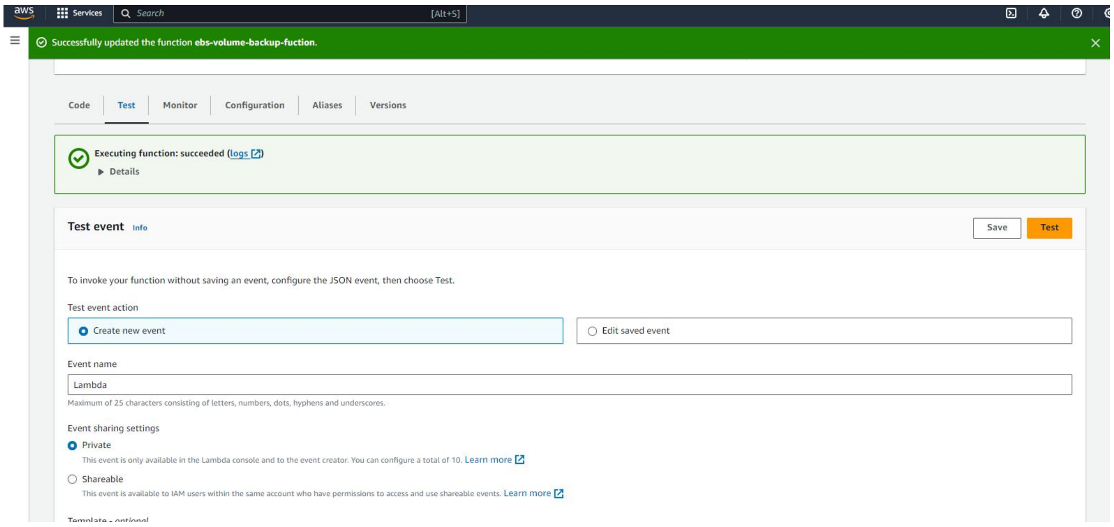
```
import boto3
import datetime

ec = boto3.client('ec2')

def lambda_handler(event, context):
    # Get the current UTC time
    current_time = datetime.datetime.utcnow()

    reservations = ec.describe_instances(
        Filters=[
            {'Name': 'tag-key', 'Values': ['backup', 'Backup', 'satyam']},
        ]
```

7. Configuration for test event to run code


Successfully updated the function **ebs-volume-backup-fuction**.

Executing function: succeeded (logs)

Test event

To invoke your function without saving an event, configure the JSON event, then choose Test.

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

Event name: Lambda

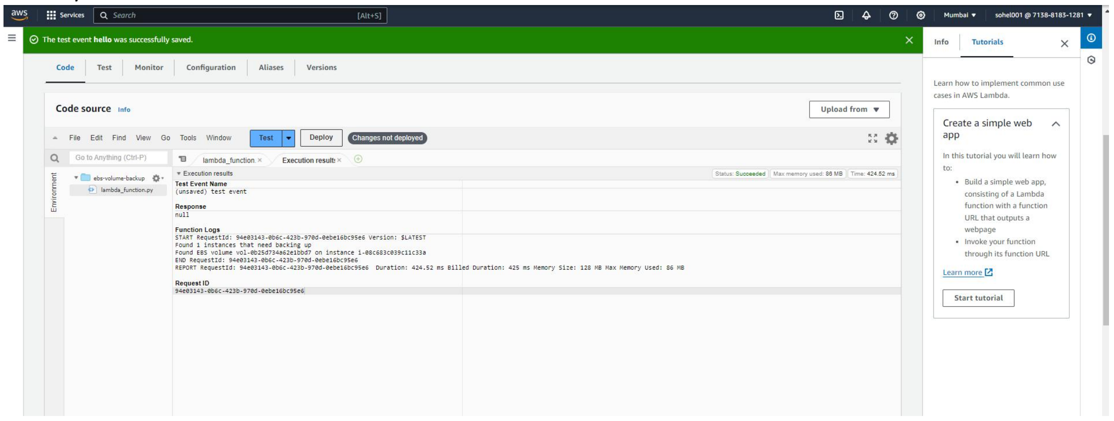
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings: [✓ Private] Shareable

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

8. Test/ Run code


The test event **hello** was successfully saved.

Code source

```
Execution results                Status: Succeeded  Max memory used: 86 MB  Time: 424.52 ms
Test Event Name
(unsaved) test event

Response
null

Function Logs
START RequestId: 94e03143-0b6c-423b-970d-0ebe16bc95e6 Version: $LATEST
Found 1 instances that need backing up
Found EBS volume vol-0b25d734a62e1bbd7 on instance i-08c683c039c11c33a
END RequestId: 94e03143-0b6c-423b-970d-0ebe16bc95e6
REPORT RequestId: 94e03143-0b6c-423b-970d-0ebe16bc95e6  Duration: 424.52 ms Billed Duration: 425 ms Memory Size: 128 MB Max Memory Used: 86 MB

Request ID
94e03143-0b6c-423b-970d-0ebe16bc95e6
```

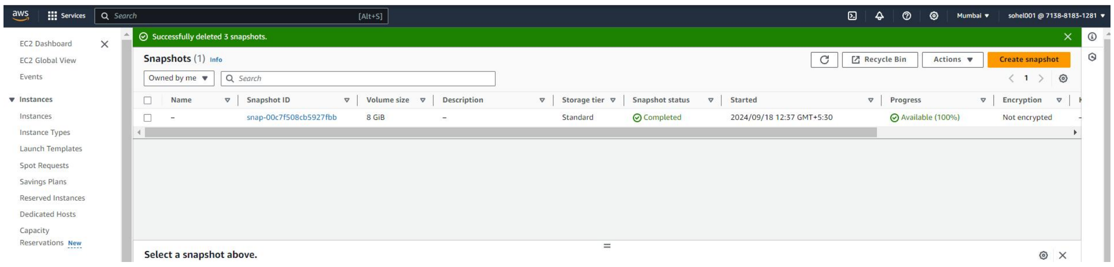
Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

9. Snapshots will be created

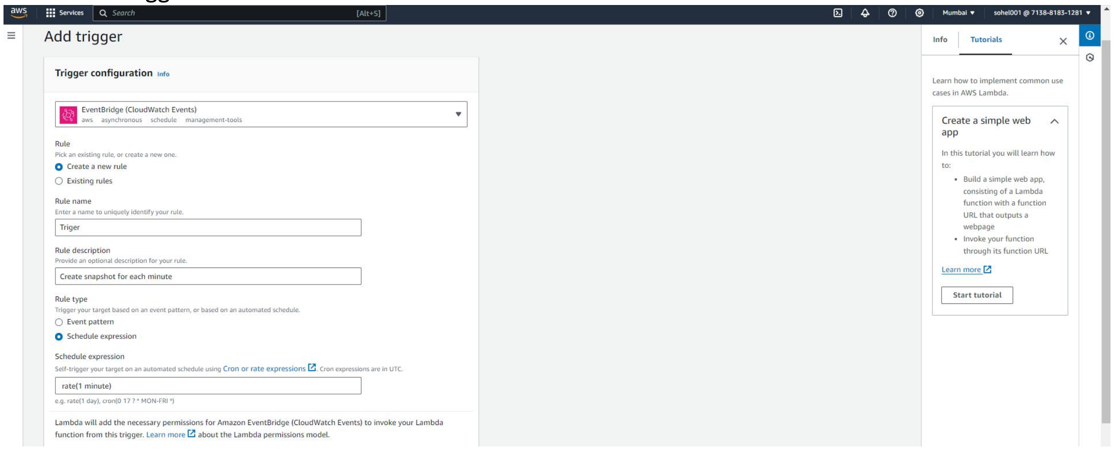

Successfully deleted 3 snapshots.

Snapshots (1)

	Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress	Encryption
☐	–	snap-00c7f508cb5927fbb	8 GiB	–	Standard	Completed	2024/09/18 12:37 GMT+5:30	Available (100%)	Not encrypted

Select a snapshot above.

10. Create trigger for each minute


Add trigger

Trigger configuration

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. Triger

Rule description: Provide an optional description for your rule. Create snapshot for each minute

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 minute)

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.

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

11. Trigger is added