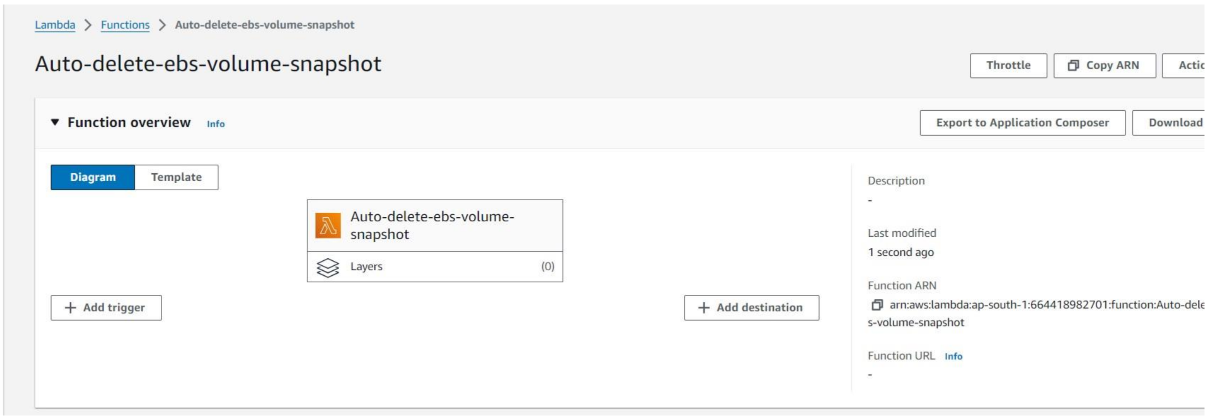
Lambda

- Delete unused EBS Volume Across Regions
- Create lambda function (**Auto-delete-ebs-volume-snapshot**)

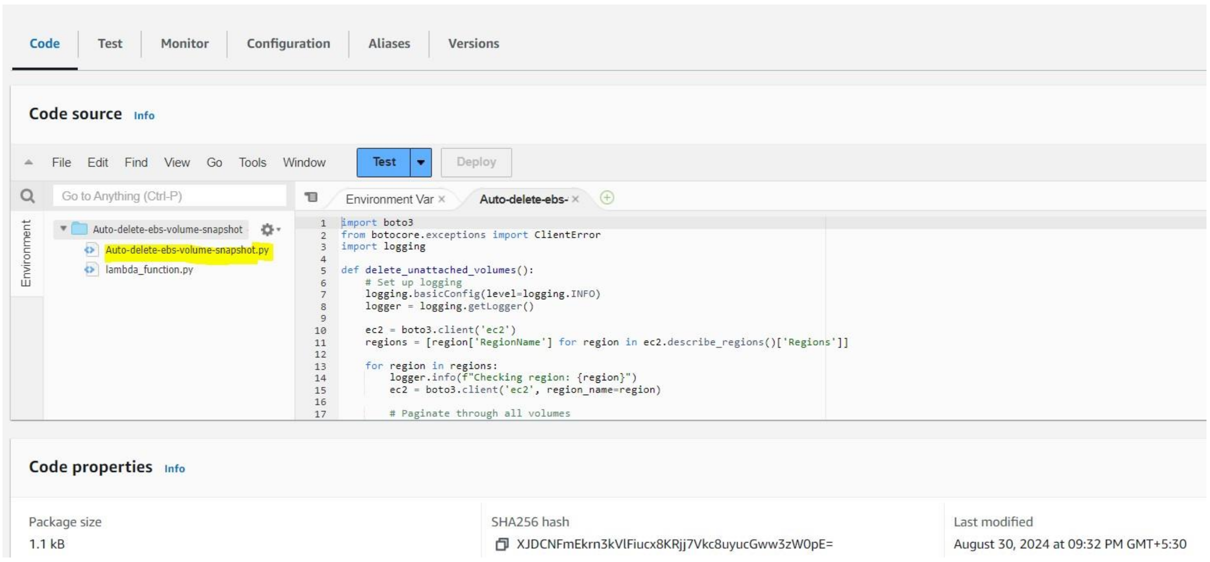


Auto-delete-ebs-volume-snapshot

Auto-delete-ebs-volume-snapshot
Layers (0)

Description
-
Last modified
1 second ago
Function ARN
arn:aws:lambda:ap-south-1:664418982701:function:Auto-dele
s-volume-snapshot
Function URL Info
-

1) Create **Auto-delete-ebs-volume-snapshot.py** file



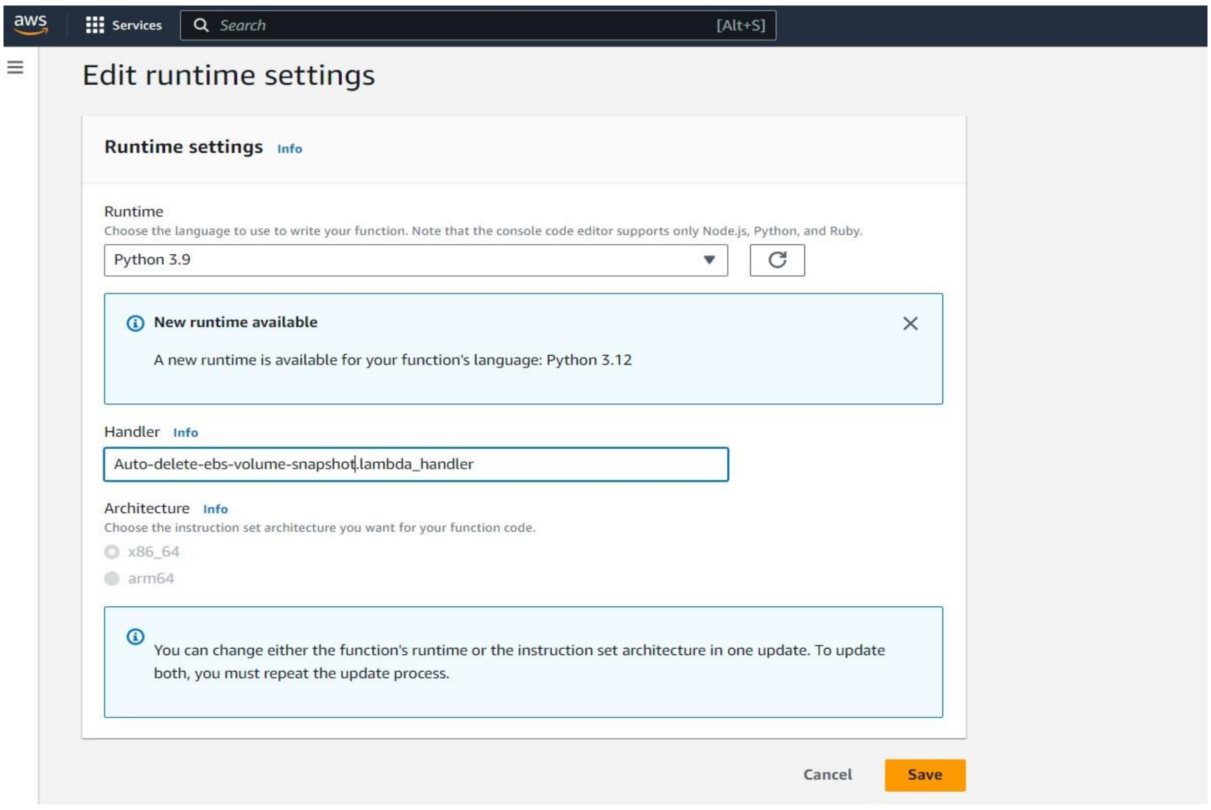
Auto-delete-ebs-volume-snapshot
Auto-delete-ebs-volume-snapshot.py
lambda_function.py

```
import boto3
from botocore.exceptions import ClientError
import logging

def delete_unattached_volumes():
    # Set up logging
    logging.basicConfig(level=logging.INFO)
    logger = logging.getLogger()

    ec2 = boto3.client('ec2')
    regions = [region['RegionName'] for region in ec2.describe_regions()['Regions']]

    for region in regions:
        logger.info(f"Checking region: {region}")
        ec2 = boto3.client('ec2', region_name=region)

        # Paginate through all volumes
```

Code properties Info

Package size	SHA256 hash	Last modified
1.1 kB	XJDCNFmEkrn3kVlFiucx8KRjj7Vkc8uyucGww3zW0pE=	August 30, 2024 at 09:32 PM GMT+5:30

2) Update the Handler name as (**Auto-delete-ebs-volume-snapshot.lambda_handler**)



Edit runtime settings

Runtime settings Info

Runtime
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

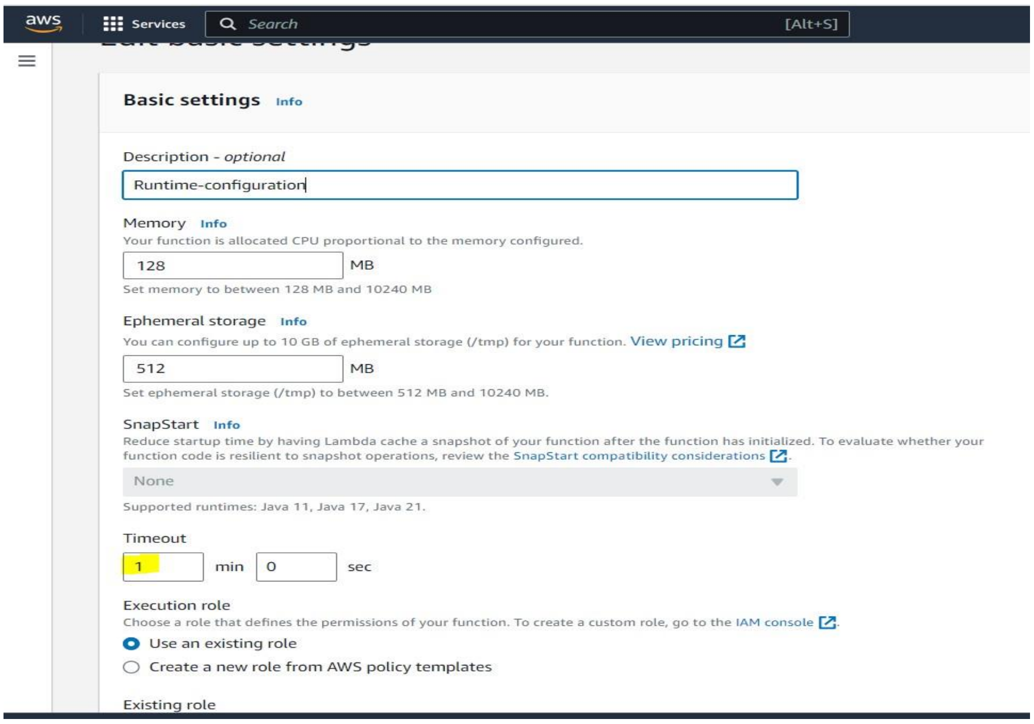
New runtime available
A new runtime is available for your function's language: Python 3.12

Handler Info
Auto-delete-ebs-volume-snapshot.lambda_handler

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

You can change either the function's runtime or the instruction set architecture in one update. To update both, you must repeat the update process.

3) Change the timeout value for 3 seconds to 1 minute in Configuration settings



Basic settings Info

Description - *optional*
Runtime-configuration

Memory Info
Your function is allocated CPU proportional to the memory configured.
128 MB
Set memory to between 128 MB and 10240 MB

Ephemeral storage Info
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing
512 MB
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

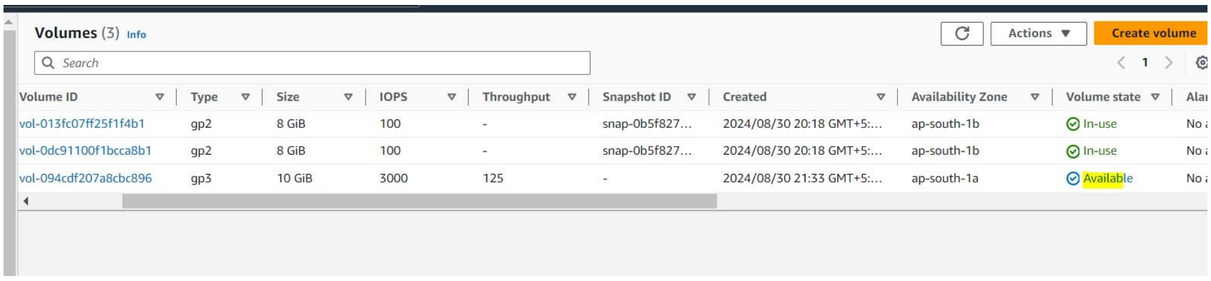
SnapStart Info
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations.
None
Supported runtimes: Java 11, Java 17, Java 21.

Timeout
1 min 0 sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
[✓ Use an existing role]
Create a new role from AWS policy templates

Existing role

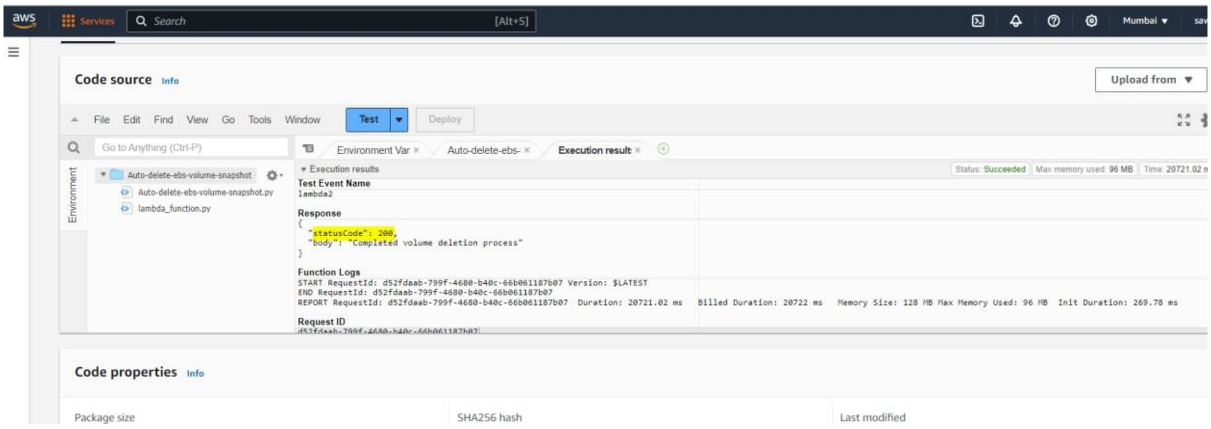
4) Created one volume which is in available state



Volumes (3) Info

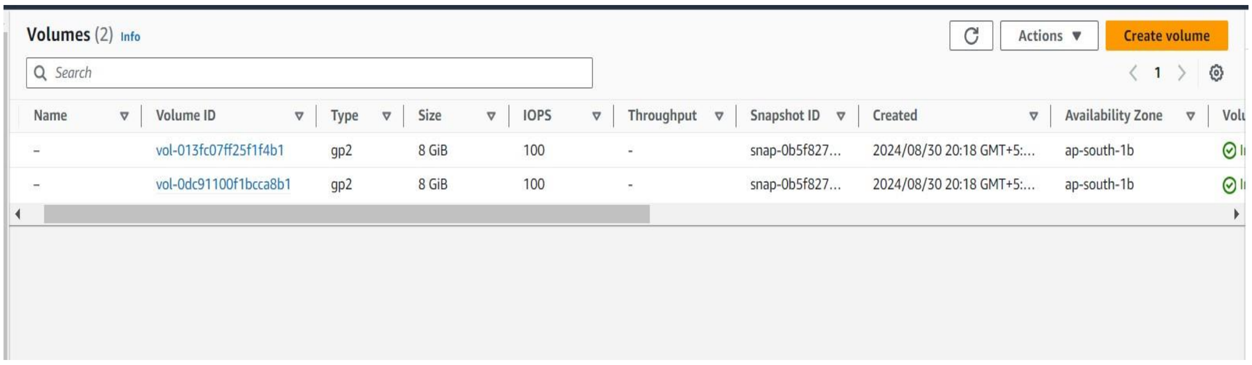
Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created	Availability Zone	Volume state	Ala
vol-013fc07ff25f1f4b1	gp2	8 GiB	100	-	snap-0b5f827...	2024/08/30 20:18 GMT+5:...	ap-south-1b	In-use	No
vol-0dc91100f1bcca8b1	gp2	8 GiB	100	-	snap-0b5f827...	2024/08/30 20:18 GMT+5:...	ap-south-1b	In-use	No
vol-094cdf207a8cbc896	gp3	10 GiB	3000	125	-	2024/08/30 21:33 GMT+5:...	ap-south-1a	Available	No

5) Now, Run the code, It should delete the free volume



```
Execution results                Status: Succeeded   Max memory used: 96 MB   Time: 20721.02 m
Test Event Name
lambda2

Response
{
  "statusCode": 200,
  "body": "Completed volume deletion process"
}

Function Logs
START RequestId: d52fdaab-799f-4680-b40c-66b061187b07 Version: $LATEST
END RequestId: d52fdaab-799f-4680-b40c-66b061187b07
REPORT RequestId: d52fdaab-799f-4680-b40c-66b061187b07  Duration: 20721.02 ms   Billed Duration: 20722 ms   Memory Size: 128 MB Max Memory Used: 96 MB  Init Duration: 269.78 ms

Request ID
d52fdaab-799f-4680-b40c-66b061187b07
```

6) volume is deleted successfully



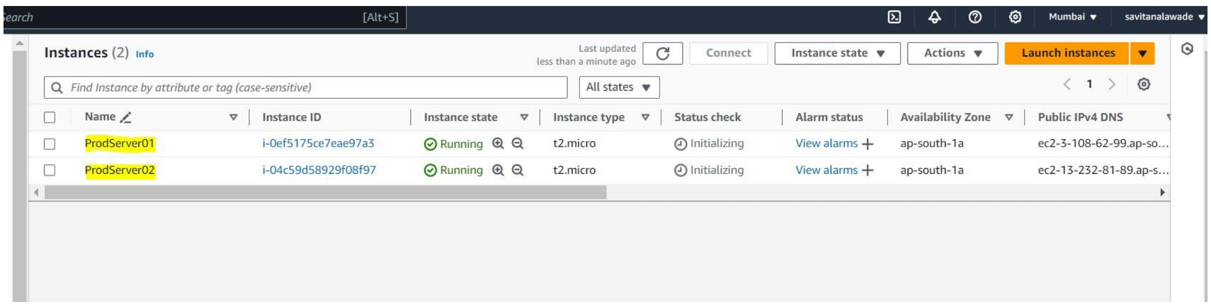
Volumes (2) Info

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created	Availability Zone	Vol
-	vol-013fc07ff25f1f4b1	gp2	8 GiB	100	-	snap-0b5f827...	2024/08/30 20:18 GMT+5:...	ap-south-1b	I
-	vol-0dc91100f1bcca8b1	gp2	8 GiB	100	-	snap-0b5f827...	2024/08/30 20:18 GMT+5:...	ap-south-1b	I

Title: "Automating EBS Volume Snapshots Using AWS Lambda and CloudWatch Triggers"

1) Create two instances (prodserver01/02) on EC2 server

- Open the AWS Management Console.
- Go to the EC2 Dashboard.
- Click Launch Instance.
- Configure the settings as required (AMI, instance type, key pair, etc.).
- Name your instances as `prodserver01` and `prodserver02`.
- Launch the instances.



Instances (2) Info

Last updated less than a minute ago

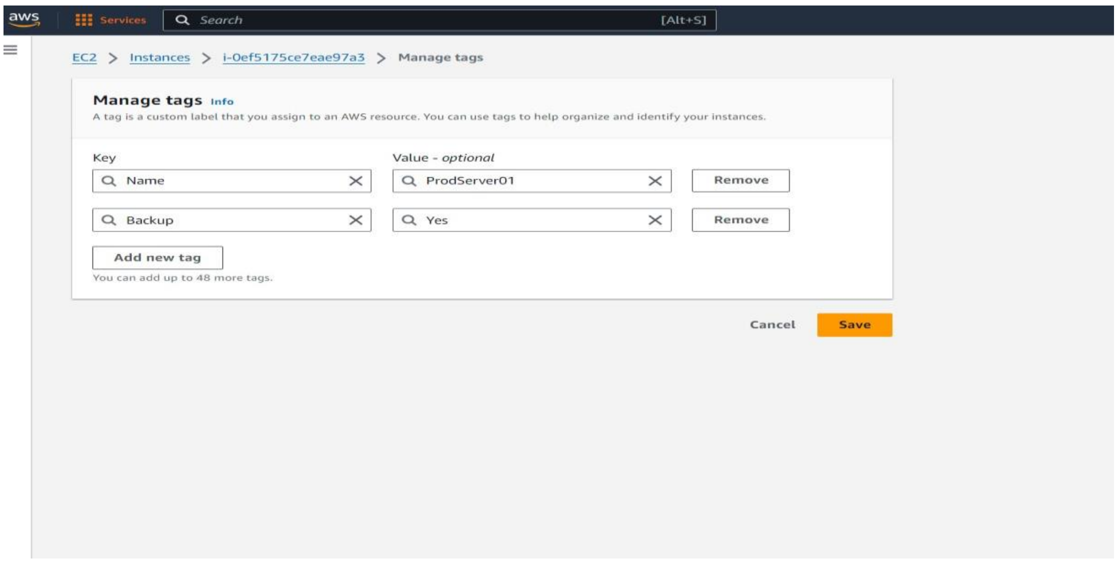
Find Instance by attribute or tag (case-sensitive) | All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	ProdServer01	i-0ef5175ce7eae97a3	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-3-108-62-99.ap-so…
☐	ProdServer02	i-04c59d58929f08f97	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-232-81-89.ap-s…

2) Add Tags for one Instance(ProdServer01)

- In the EC2 Dashboard, select prodserver01.
- Click on the Tags tab.
- Add the necessary tags, such as:

- Key: Name, Value: ProdServer01.
- Key: Environment, Value: Production.



EC2 > Instances > i-0ef5175ce7eae97a3 > Manage tags

Manage tags Info

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	ProdServer01	Remove
Backup	Yes	Remove

Add new tag

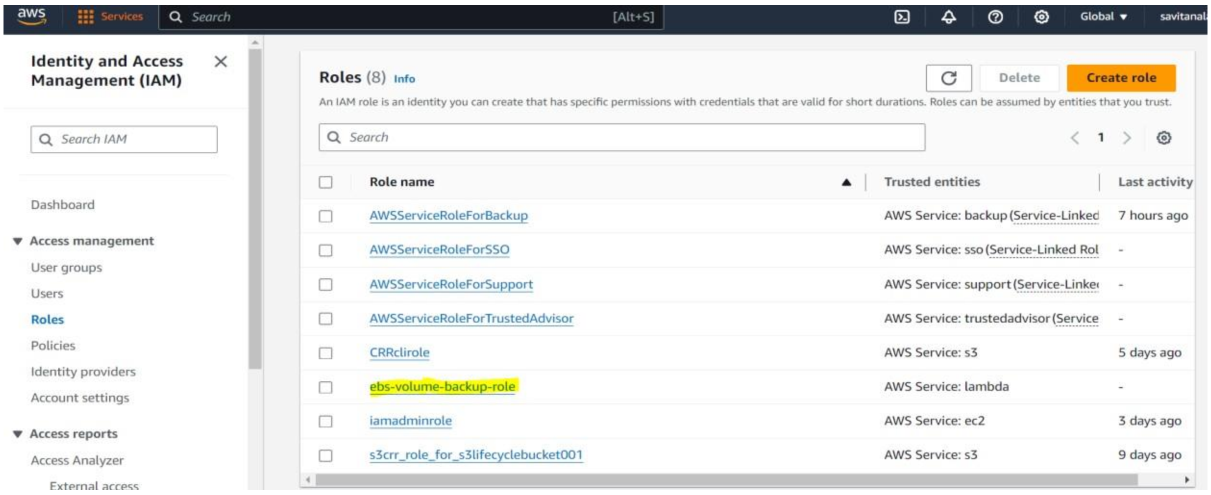
You can add up to 48 more tags.

Cancel Save

3) Create IAM role for Lambda service(ebs-volume-backup-role)

- Go to the IAM Dashboard.
- Select Roles, then click Create Role.
- Select Lambda as the service that will use this role.
    - Attach the following policies:

- AmazonEC2FullAccess (to manage EBS volumes and snapshots).
- AWSLambdaBasicExecutionRole (for Lambda logging to CloudWatch).
- Name the role ebs-volume-backup-role and create it.


Identity and Access Management (IAM)

Dashboard
Access management: User groups, Users, Roles, Policies, Identity providers, Account settings
Access reports: Access Analyzer, External access

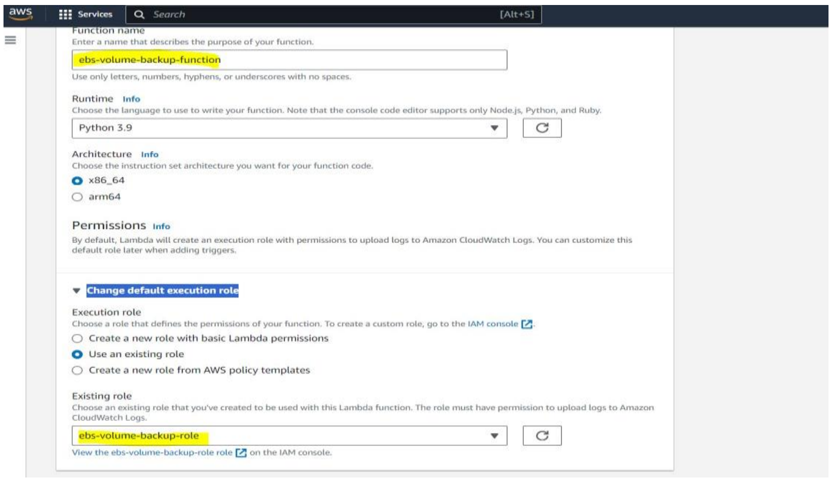
Roles (8) Info

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
AWSServiceRoleForBackup	AWS Service: backup (Service-Linked	7 hours ago
AWSServiceRoleForSSO	AWS Service: sso (Service-Linked Rol	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
CRRclirole	AWS Service: s3	5 days ago
ebs-volume-backup-role	AWS Service: lambda	-
iamadminrole	AWS Service: ec2	3 days ago
s3crr_role_for_s3lifecyclebucket001	AWS Service: s3	9 days ago

4) While creating lambda function select any language and add role

- Go to the Lambda Dashboard.
- Click Create function.
- Select Author from scratch.
- Enter the function name (e.g., `EBSVolumeBackup`).
- Select the runtime (choose any language, e.g., Python or Node.js).
- Under Permissions, choose the IAM role you created (`ebs-volume-backup-role`).
- Click Create function.


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

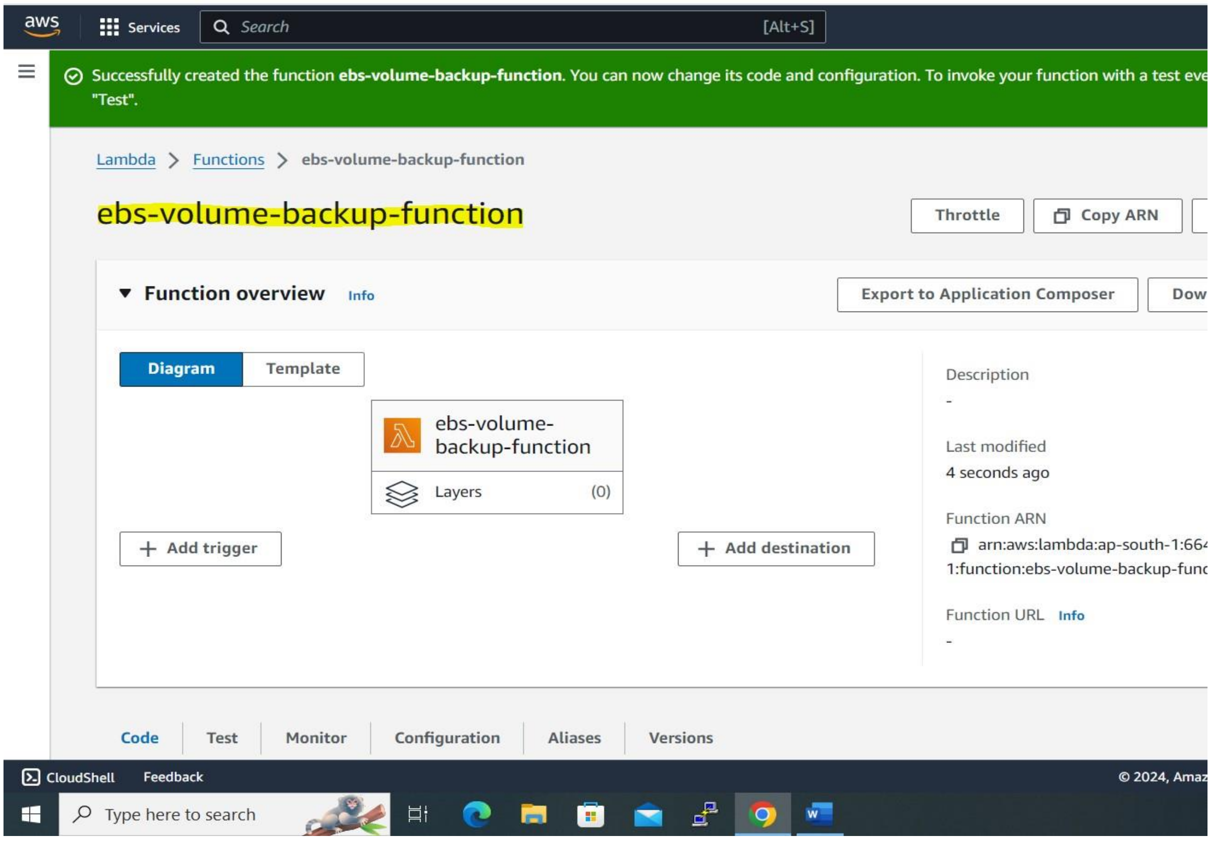
Change default execution role

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) Successfully created function

After creating the function, you will be redirected to the function's configuration page.


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

Lambda > Functions > ebs-volume-backup-function

ebs-volume-backup-function

Function overview Info

Description
-

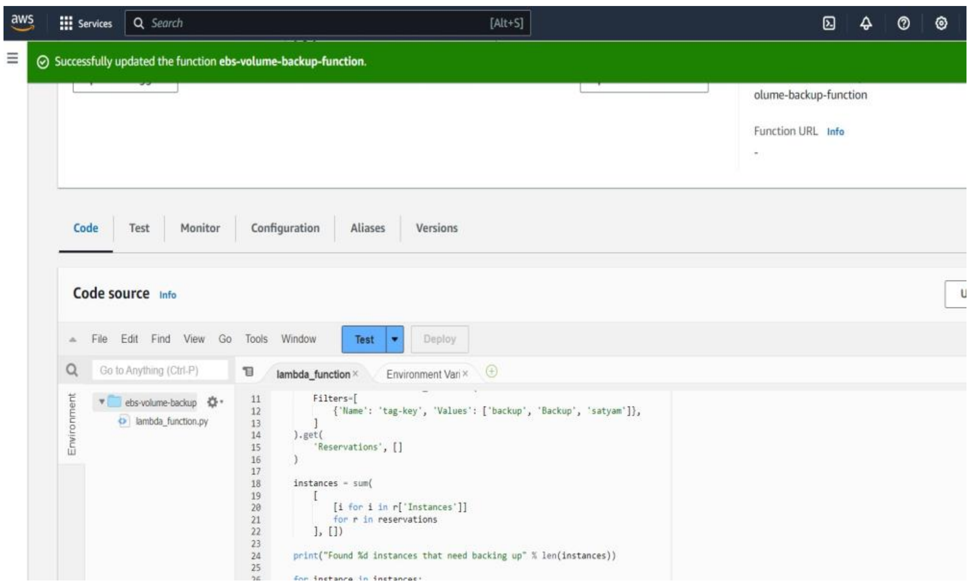
Last modified
4 seconds ago

Function ARN
arn:aws:lambda:ap-south-1:664
1:function:ebs-volume-backup-func

Function URL Info
-

6) Deploy the code

In the Lambda function editor, add code for automating EBS volume snapshots.


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

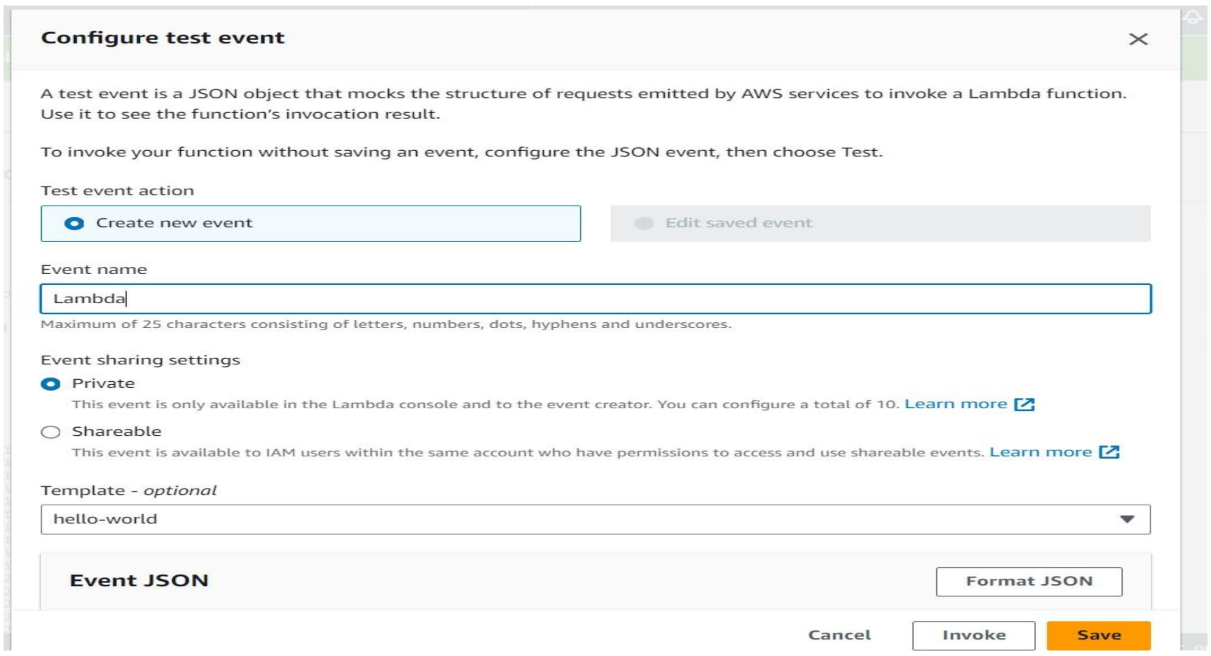
olume-backup-function

Function URL Info
-

```
        Filters=[
            {'Name': 'tag-key', 'Values': ['backup', 'Backup', 'satyam']},
        ]
    ).get(
        'Reservations', []
    )

    instances = sum(
        [
            [i for i in r['Instances']]
            for r in reservations
        ], [])

    print("Found %d instances that need backing up" % len(instances))

    for instance in instances:
```

7) Configure test event to run code

- Click on the Test tab.
- Configure a new test event (you can choose any test template, like "Hello World").
- Save the test event with a name (e.g., `SnapshotTest`).



Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

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

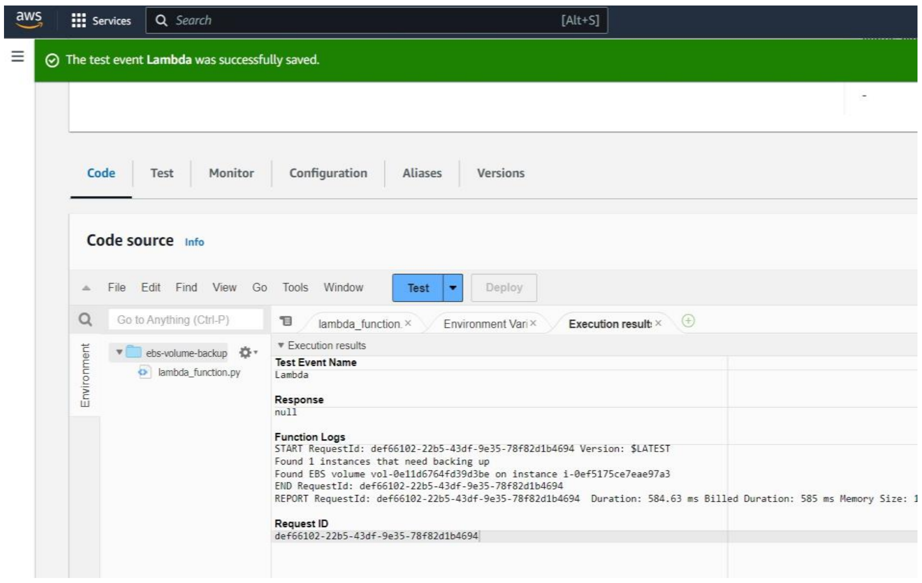
Shareable — This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*: hello-world

Event JSON

8) Test\run the code successfully

- After creating the test event, click Test.
- The Lambda function will run, and you should see logs indicating that snapshots are being created.



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

Execution results

Test Event Name
Lambda

Response
null

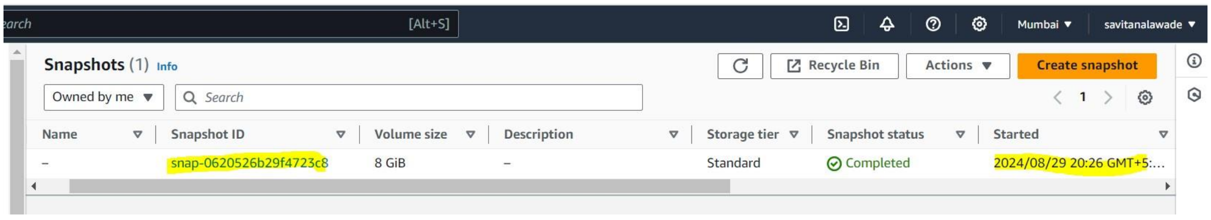
Function Logs

```
START RequestId: def66102-22b5-43df-9e35-78f82d1b4694 Version: $LATEST
Found 1 instances that need backing up
Found EBS volume vol-0e11d6764fd39d3be on instance i-0ef5175ce7eae97a3
END RequestId: def66102-22b5-43df-9e35-78f82d1b4694
REPORT RequestId: def66102-22b5-43df-9e35-78f82d1b4694  Duration: 584.63 ms Billed Duration: 585 ms Memory Size: 1
```

Request ID
def66102-22b5-43df-9e35-78f82d1b4694

9) Snapshot will be created

- Go to the EC2 Dashboard > Snapshots.
- You should see the snapshots being created for the attached EBS volumes.

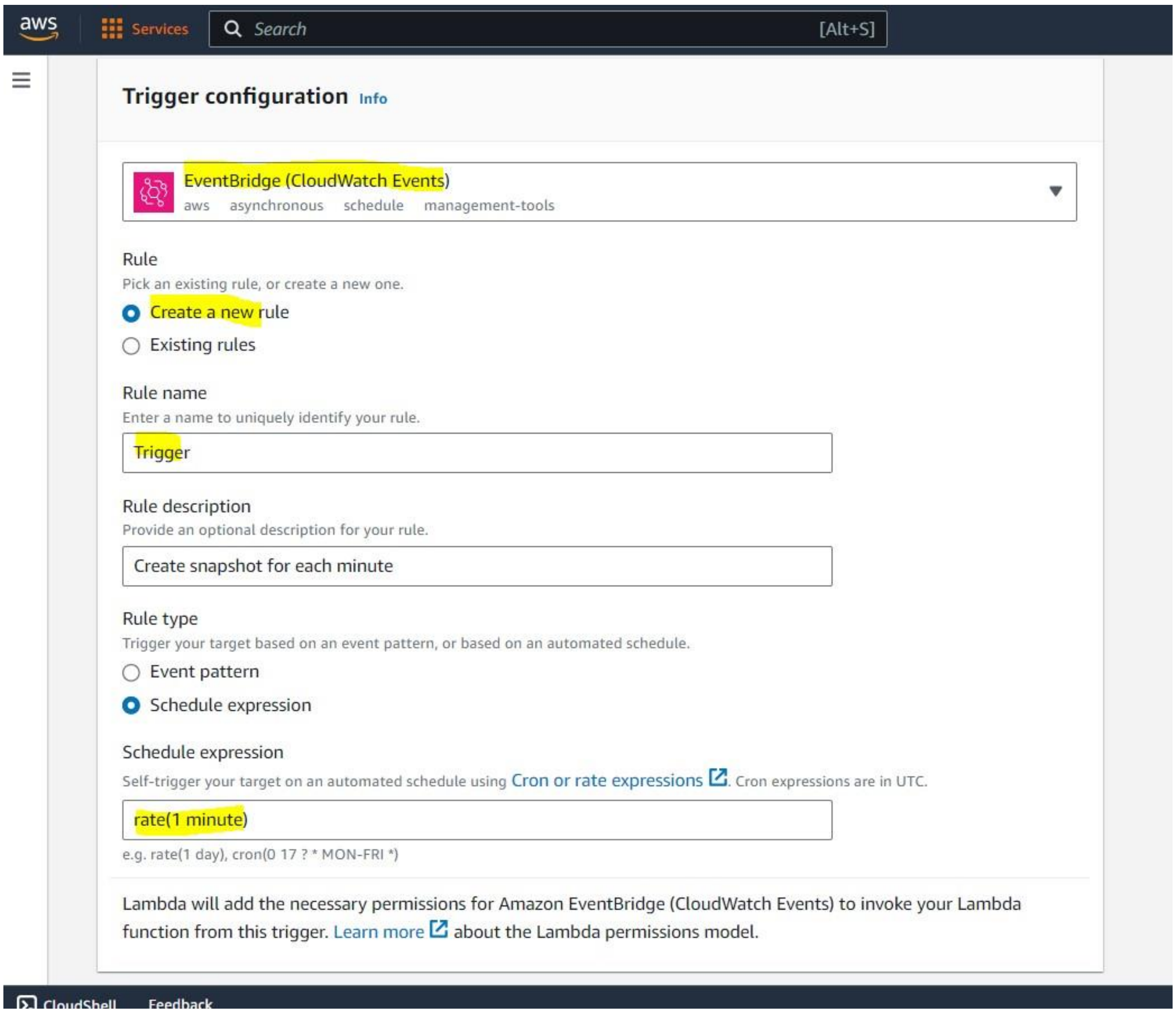


Snapshots (1)

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started
–	snap-0620526b29f4723c8	8 GiB	–	Standard	Completed	2024/08/29 20:26 GMT+5:...

10) Create trigger for each minute

- In the Lambda function, go to the Triggers section.
- Add a new trigger, select CloudWatch Events (EventBridge).



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.
Trigger

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

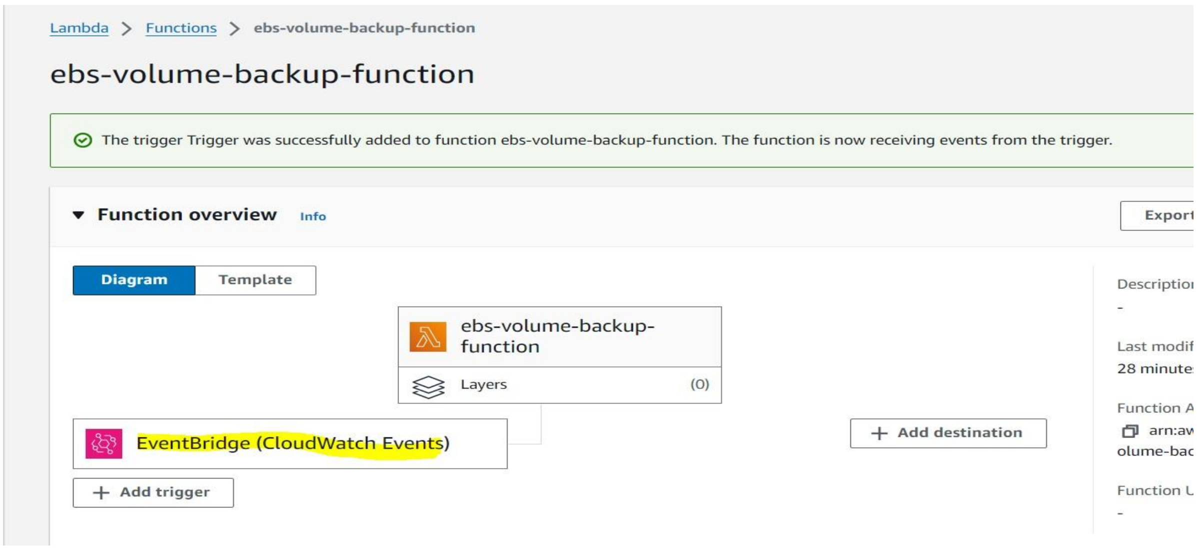
Rule type
Trigger your target based on an event pattern, or based on an automated schedule.
Event pattern
[✓ Schedule expression]

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.

11) Triger is added

- The trigger will now be active and will invoke the Lambda function every minute.

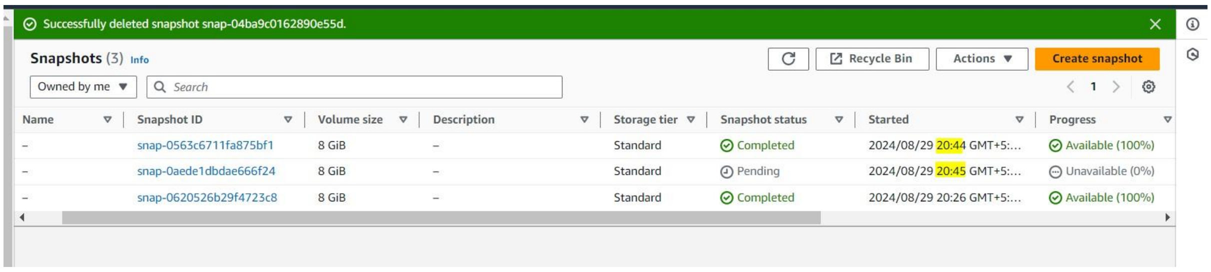

Lambda > Functions > ebs-volume-backup-function

ebs-volume-backup-function

The trigger Trigger was successfully added to function ebs-volume-backup-function. The function is now receiving events from the trigger.

12) We can see Snapshots are creating for each min

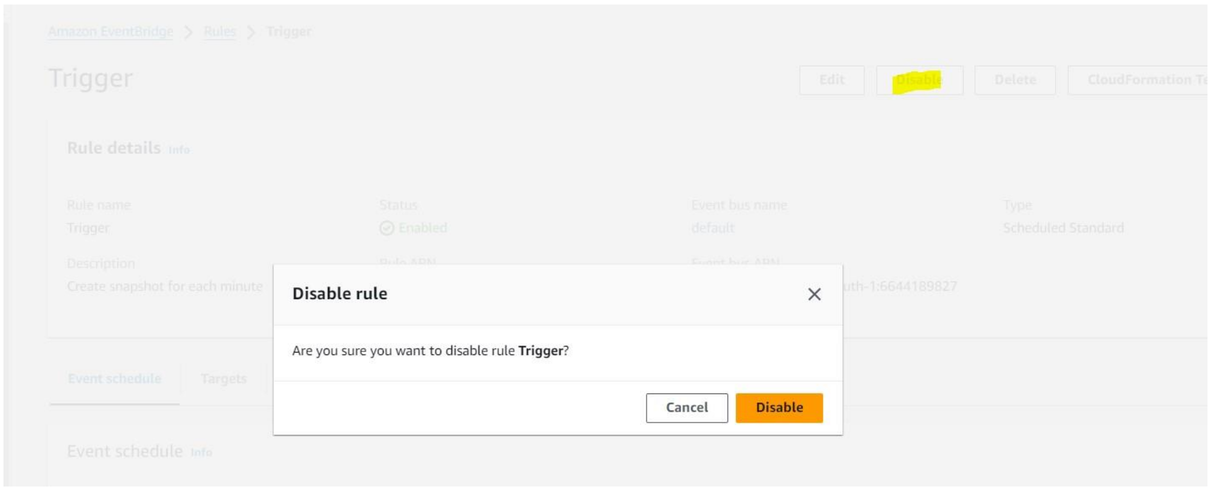
- Go back to the EC2 Dashboard > Snapshots.
- You’ll notice snapshots being created for the EBS volumes at one-minute intervals.


Successfully deleted snapshot snap-04ba9c0162890e55d.

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress
–	snap-0563c6711fa875bf1	8 GiB	–	Standard	Completed	2024/08/29 20:44 GMT+5:...	Available (100%)
–	snap-0aede1dbdae666f24	8 GiB	–	Standard	Pending	2024/08/29 20:45 GMT+5:...	Unavailable (0%)
–	snap-0620526b29f4723c8	8 GiB	–	Standard	Completed	2024/08/29 20:26 GMT+5:...	Available (100%)

13) Now disabled the trigger and deleted

- Go to the Lambda function's Triggers section.
- Disable the CloudWatch Events trigger by editing it.
- After disabling, you can delete the trigger to stop the snapshots from being created..



Rule name
Trigger

Status
Enabled

Event bus name
default

Type
Scheduled Standard

Description
Create snapshot for each minute

Disable rule

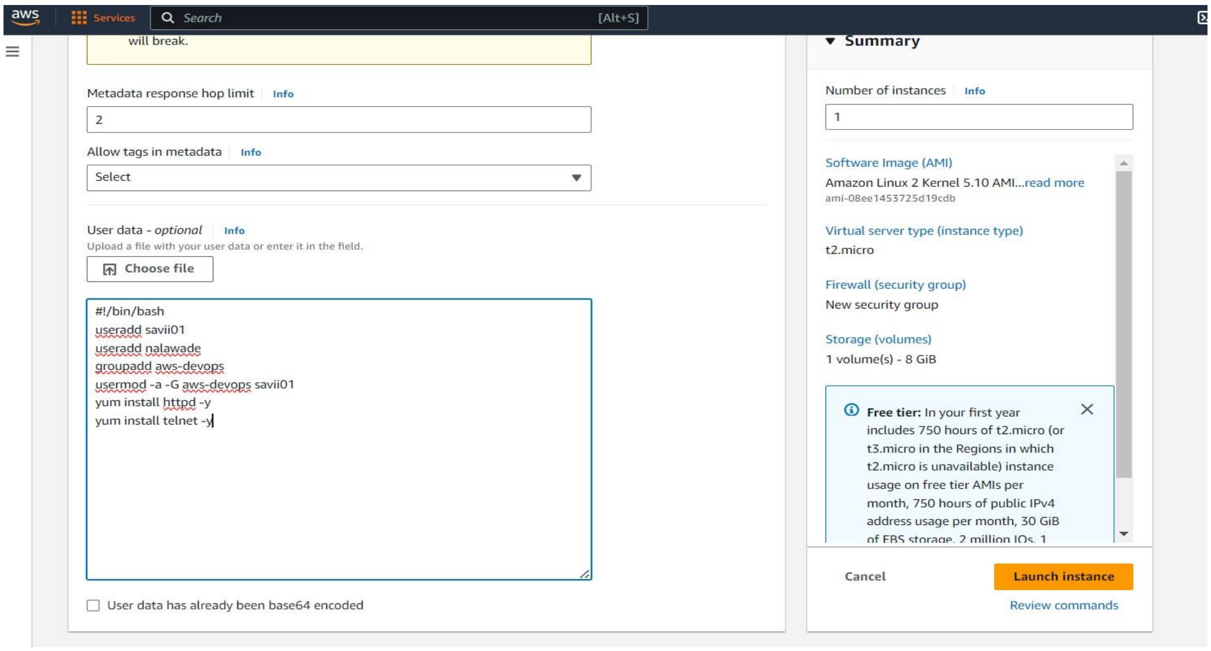
Are you sure you want to disable rule **Trigger**?

Cancel
Disable

This process ensures automated EBS volume backups with scheduled snapshots created and managed by an AWS Lambda function

Bootstrapping

1) Create a EC2 instance with stuffing and installing some package



will break.

Metadata response hop limit Info

2

Allow tags in metadata Info

Select

User data - *optional* Info

Upload a file with your user data or enter it in the field.

Choose file

```
#!/bin/bash
useradd savii01
useradd nalawade
groupadd aws-devops
usermod -a -G aws-devops savii01
yum install httpd -y
yum install telnet -y
```

User data has already been base64 encoded

Summary

Number of instances Info

1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-08ee1453725d19cdb

Virtual server type (instance type)
t2.micro

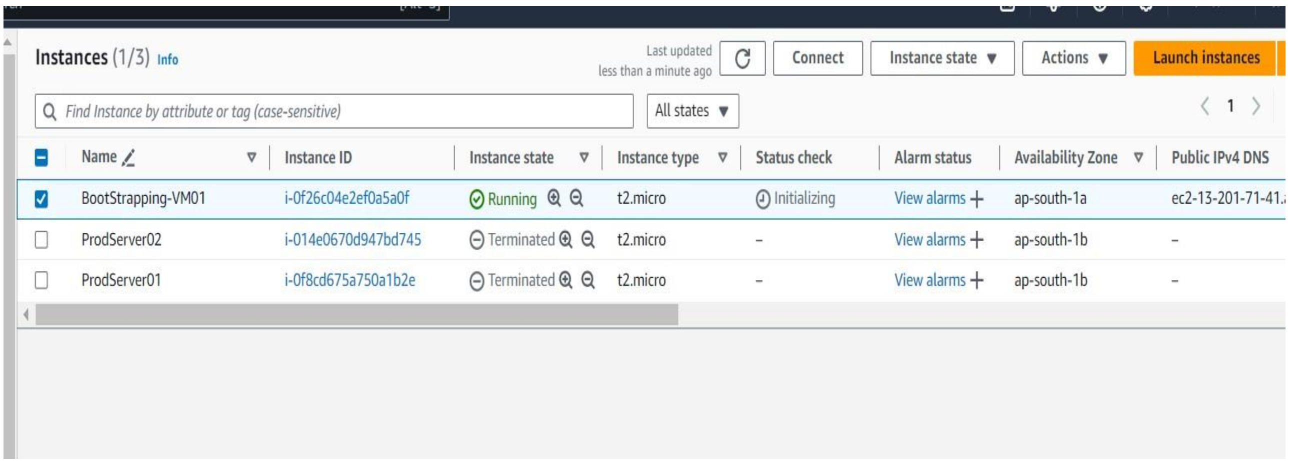
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1

Cancel | Launch instance | Review commands

2) The instance is created successfully



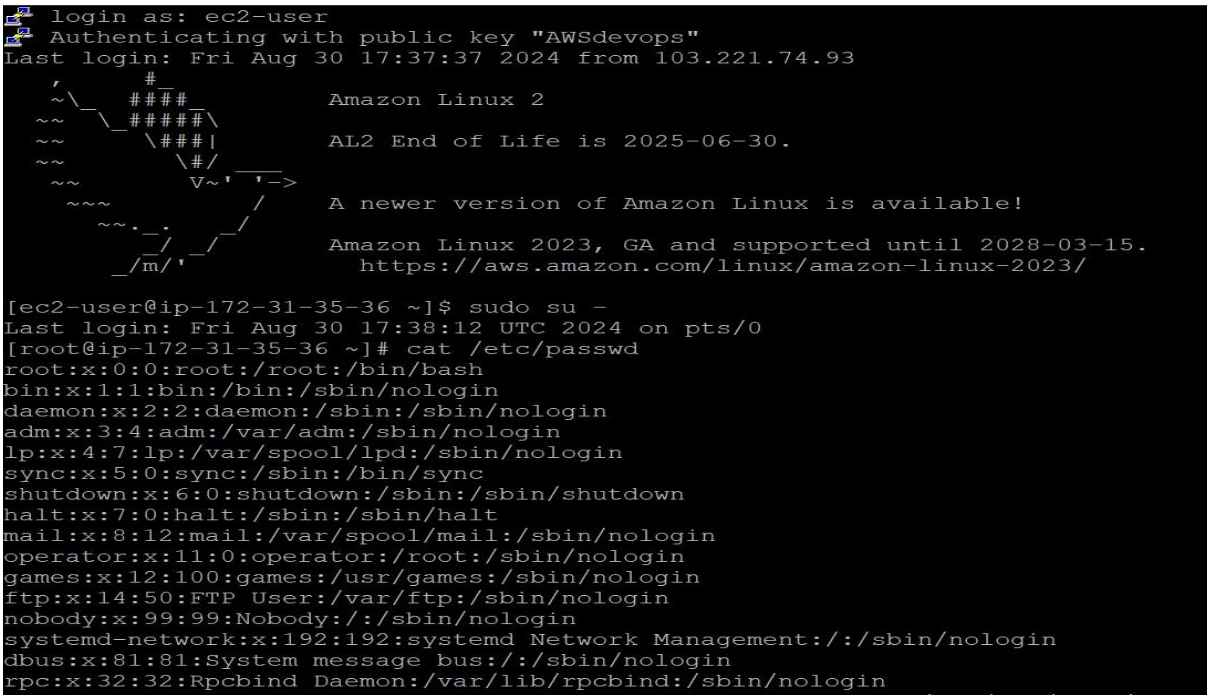
Instances (1/3) Info

Last updated less than a minute ago | Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

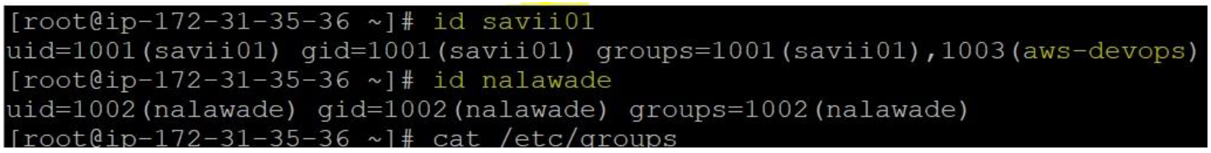
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
BootStrapping-VM01	i-0f26c04e2ef0a5a0f	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-201-71-41.
ProdServer02	i-014e0670d947bd745	Terminated	t2.micro	-	View alarms +	ap-south-1b	-
ProdServer01	i-0f8cd675a750a1b2e	Terminated	t2.micro	-	View alarms +	ap-south-1b	-

3) Connected the BootStarapping-VM01 via putty



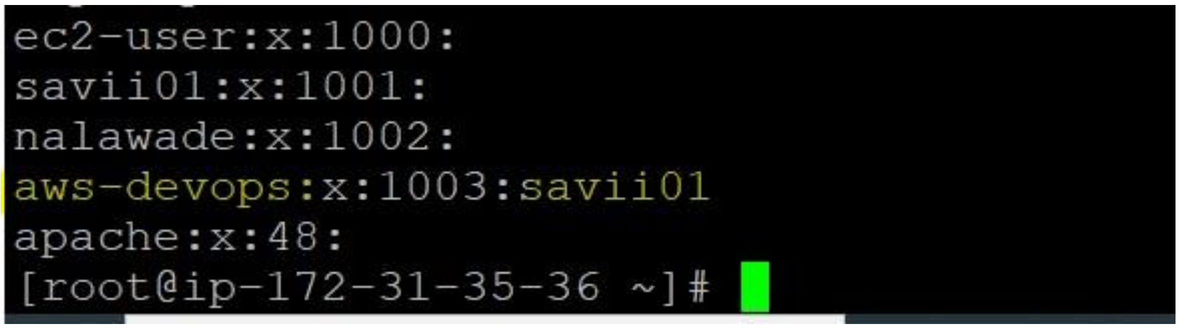
```
 login as: ec2-user
 Authenticating with public key "AWSdevops"
Last login: Fri Aug 30 17:37:37 2024 from 103.221.74.93
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-35-36 ~]$ sudo su -
Last login: Fri Aug 30 17:38:12 UTC 2024 on pts/0
[root@ip-172-31-35-36 ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
```

4) Check the users are created or not



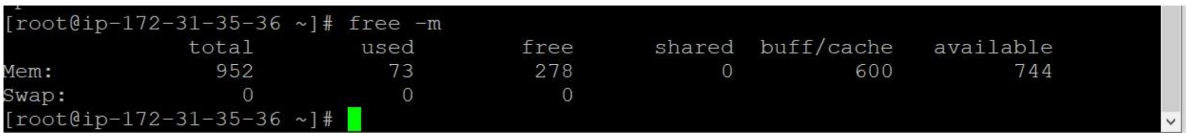
```
[root@ip-172-31-35-36 ~]# id savii01
uid=1001(savii01) gid=1001(savii01) groups=1001(savii01),1003(aws-devops)
[root@ip-172-31-35-36 ~]# id nalawade
uid=1002(nalawade) gid=1002(nalawade) groups=1002(nalawade)
[root@ip-172-31-35-36 ~]# cat /etc/groups
```

5) Check the if user is added or not as per stuffing (cat /etc/group)



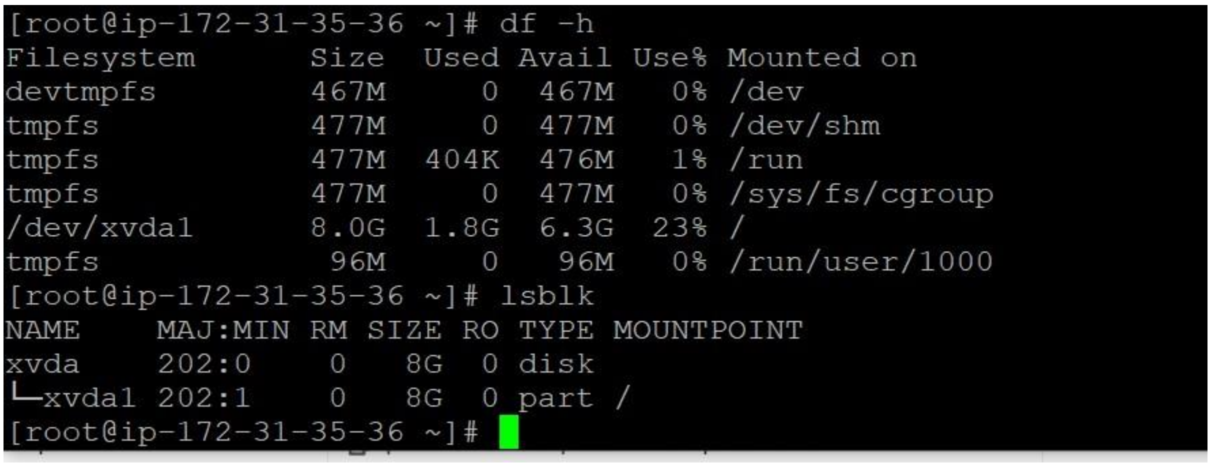
```
ec2-user:x:1000:
savii01:x:1001:
nalawade:x:1002:
aws-devops:x:1003:savii01
apache:x:48:
[root@ip-172-31-35-36 ~]#
```

6) Check the memory



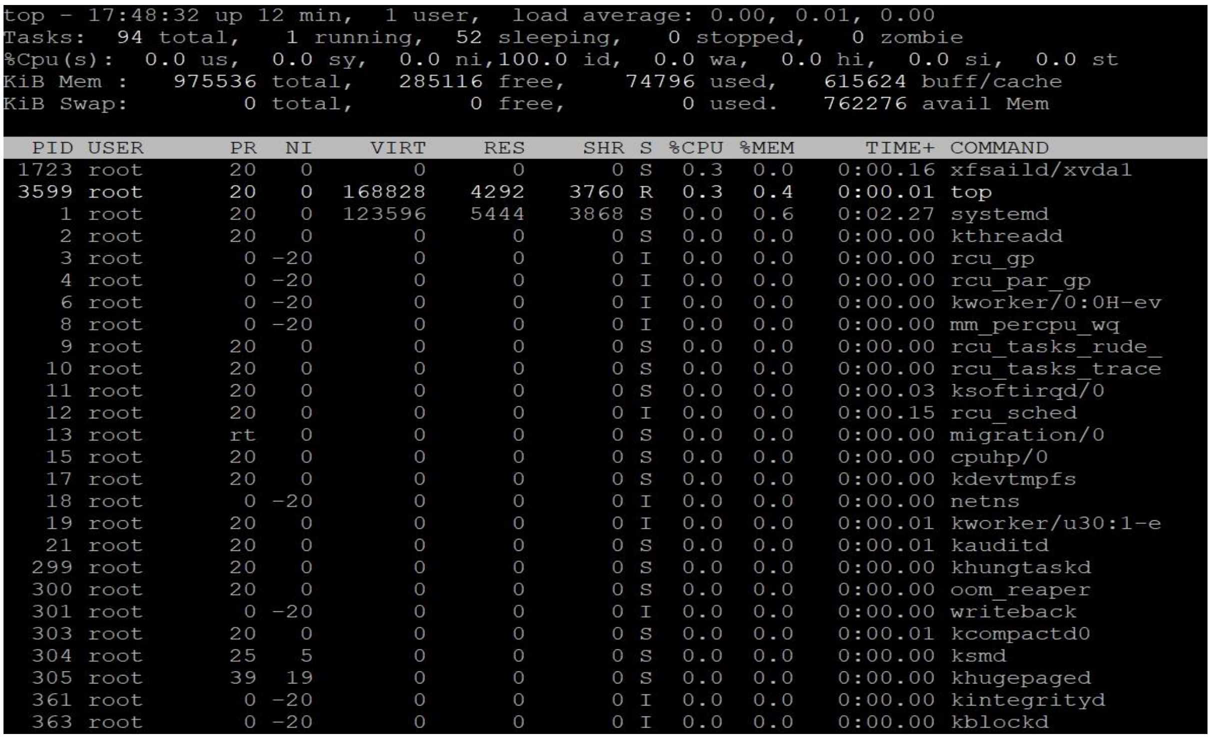
```
[root@ip-172-31-35-36 ~]# free -m
              total        used        free      shared  buff/cache   available
Mem:            952          73         278           0         600         744
Swap:             0           0           0
[root@ip-172-31-35-36 ~]#
```

7) Check the hard disk



```
[root@ip-172-31-35-36 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        467M     0  467M   0% /dev
tmpfs           477M     0  477M   0% /dev/shm
tmpfs           477M  404K  476M   1% /run
tmpfs           477M     0  477M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.8G  6.3G  23% /
tmpfs            96M     0   96M   0% /run/user/1000
[root@ip-172-31-35-36 ~]# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
[root@ip-172-31-35-36 ~]#
```

8) Run Top command to check running process



```
top - 17:48:32 up 12 min,  1 user,  load average: 0.00, 0.01, 0.00
Tasks:  94 total,   1 running,  52 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   975536 total,   285116 free,    74796 used,   615624 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   762276 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1723 root      20   0       0      0      0 S  0.3  0.0   0:00.16 xfsaild/xvda1
 3599 root      20   0  168828   4292   3760 R  0.3  0.4   0:00.01 top
    1 root      20   0  123596   5444   3868 S  0.0  0.6   0:02.27 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/0:0H-ev
    8 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_tasks_rude_
   10 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_tasks_trace
   11 root      20   0       0      0      0 S  0.0  0.0   0:00.03 ksoftirqd/0
   12 root      20   0       0      0      0 I  0.0  0.0   0:00.15 rcu_sched
   13 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
   15 root      20   0       0      0      0 S  0.0  0.0   0:00.00 cpuhp/0
   17 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   18 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 netns
   19 root      20   0       0      0      0 I  0.0  0.0   0:00.01 kworker/u30:1-e
   21 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kauditd
  299 root      20   0       0      0      0 S  0.0  0.0   0:00.00 khungtaskd
  300 root      20   0       0      0      0 S  0.0  0.0   0:00.00 oom_reaper
  301 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 writeback
  303 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kcompactd0
  304 root      25   5       0      0      0 S  0.0  0.0   0:00.00 ksmd
  305 root      39  19       0      0      0 S  0.0  0.0   0:00.00 khugepaged
  361 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kintegrityd
  363 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kblockd
```