Create an Amazon S3 bucket

In this subtask, you will create an Amazon S3 bucket

5. On the **AWS Management Console**, on the **Services** menu, click **S3**.
6. Click **Create bucket**.
7. In the **Create bucket** dialog box, configure:
    - **Bucket name**: Type a bucket name that will be unique across Amazon S3. This value will be referred to as *s3-bucket-name* in subsequent procedures. Make a note of the *s3-bucket-name* for future use.
    - **Region**: Leave as default.
8. Click **Create**.

Attach Instance Profile to Processor

In this section you will attach a pre-created IAM Role as an Instance Profile to the Processor Host, giving it the permissions to interact with your Amazon S3 bucket.

9. On the **Services** menu, click **EC2**.
10. In the navigation pane, click **Instances**.
11. Select the **Processor**.
12. Click on **Actions** then **Security**, followed by **Modify IAM role**.
13. Select the `S3BucketAccess` role under **IAM role**.
14. Click **Apply** and then **Close**.

Task 2: Taking Snapshots of Your Instance

In this section, you will learn how to use the AWS Command Line Interface (CLI) to manage the processing of snapshots of an instance.

Your AWS account is limited in any region to holding 10,000 snapshots. Furthermore, you are charged every month per gigabyte of snapshot data that you store. This charge is minimized by the fact that AWS takes incremental snapshots of your instances after the first snapshot, and also by the fact that snapshot data is compressed. However, to optimize both maintenance and cost, we recommend that you monitor the number of snapshots stored for each instance and routinely delete old snapshots that you no longer need.

Connect to the Command Host using SSH

Taking an Initial Snapshot

In this procedure, you will take an initial snapshot of the Processor instance.

To take a snapshot, you will use the **aws ec2 create-snapshot** command. Because this command takes a volume ID, you will first need to find the volume ID for the Amazon EBS volume attached to your Processor instance. To do this, use the **aws ec2 describe-instances** command.

The **aws ec2 create-snapshot** command will take a snapshot of your disk at the time that the command was issued; subsequent writes to the disk are not included in the snapshot. However, due to application and OS write caching, a snapshot on a running instance might be inconsistent and result in missing or corrupted data. Therefore, before taking the snapshot of the Processor instance, you will shut it down. This ensures a consistent snapshot.

If you are taking a snapshot of a secondary (non-root) Amazon EBS volume, you can also unmount the volume before taking a snapshot to ensure that you get a consistent copy. To back up database systems (e.g., MySQL), you can freeze the file system to suspend write operations or enable replication and take periodic backups of your read replica.

36. To get a full description of the Processor instance, copy the following command and run it from within your instance:

```
aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor'
```

This command uses the **--filter** tag to limit the results description to the new instance that you created in the previous section. The command will respond with a full, JSON-based description of the instance and all of its attributes. You will now modify this command to return just the subset of data—the Amazon EBS volume information—that you are interested in.

37. To narrow down the results of the previous command further, copy the following command and run it from within your instance:

```
aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' --
query
'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:Volu
meId}'
```

This modified command uses the --query attribute to specify a JMESPath query that returns only the volume ID of the only volume (the root volume) attached to the Processor instance. You should receive a response similar to this:

```
{
"VolumeId": "vol-1234abcd"
}
```

This value will be referred to as volume-id in subsequent commands.

38. Before taking a snapshot, you will shut down the Processor instance, which requires its instance ID. To obtain the instance ID, copy the following command and run it from within your instance:

```
aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].InstanceId'
```

This value will be referred to as instance-id in subsequent commands.

39. To shut down the Processor instance, copy the following command, replace *INSTANCE-ID* with your instance id, and run it from within your instance:

```
aws ec2 stop-instances --instance-ids INSTANCE-ID
```

40. Before moving to the next step in this procedure, verify that the Processor instance has stopped by running the following command, replacing *INSTANCE-ID* with your instance id. When the Processor instance has stopped, the command will return to a prompt.

```
aws ec2 wait instance-stopped --instance-id INSTANCE-ID
```

41. To create your first snapshot of the root volume of your Processor instance, copy the following command, replace VOLUME-ID_ with your volume id, and run it in your SSH window:

```
aws ec2 create-snapshot --volume-id VOLUME-ID
```

The command will return a set of information that includes a **SnapshotId** value that uniquely identifies the new snapshot. This value will be referred to as **snapshot-id** in subsequent commands.

42. To check the status of your snapshot, copy the following command, replace *SNAPSHOT-ID* your **snapshot-id**, and run it in your SSH window:

```
aws ec2 wait snapshot-completed --snapshot-id SNAPSHOT-ID
```

Continue with the below procedure when the command completes.

43. To restart the Processor instance, copy the following command, replace the *INSTANCE-ID* to your instance id and run it in your SSH window:

```
aws ec2 start-instances --instance-ids INSTANCE-ID
```

44. To check on the status of the restart operation, copy the following command, replace *INSTANCE-ID* with your instance id, and run it in your SSH window:

```
aws ec2 wait instance-running --instance-id INSTANCE-ID
```

Schedule Creation of Subsequent Snapshots

Using the Linux scheduling system (cron), you can easily set up a recurring snapshot process so that new snapshots of your data are taken automatically.

For the purposes of this lab, you will schedule snapshot creation every minute so that you can verify the results of your work. In the next procedure, you will use automation to manage the number of snapshots that are maintained for a volume.

Note This section of the lab does not stop the instance in order to create a large number of snapshots for the next procedure. If you need to guarantee consistency, you can develop a fuller automation script that shuts down the instance or quiesces the disk first, as discussed in Task 2.

45. To create a cron entry that will schedule a job that runs every minute, copy the following command, replace *VOLUME-ID* with your volume-id and run it from within your instance:

```
echo "* * * * *  aws ec2 create-snapshot --volume-id VOLUME-ID 2>&1 >>
/tmp/cronlog" > cronjob
```

46. To schedule this cron task, copy the following command and run it from within your instance:

```
crontab cronjob
```

Note: This will take 1-2 minutes

47. To verify that subsequent snapshots are being created, copy the following command, replace **<volume-id>** with your volume-id and run it from within your instance:

```
aws ec2 describe-snapshots --filters "Name=volume-id,Values=<volume-
id>"
```

After a few minutes, you should ideally see one or more Snapshots. If this is not working as expected, please request assistance from your instructor.

48. Wait a few minutes so that a few more snapshots will be generated before beginning the next task.

Retaining Only Last Two EBS Volume Snapshots

In this procedure, you will run a Python script that maintains only the last two snapshots for any given Amazon EBS volume associated with your account.

As discussed at the beginning of this section, aggressive snapshot management both limits your costs and simplifies management over time. Using a few lines of code, you can leverage one of the many AWS Software Development Kits (SDKs) to create a program that deletes unnecessary snapshots.

49. Use the following command to stop the cron job that you previously created:

```
crontab -r
```

50. In the home directory of your CommandHost instance is a file named *snapshotter_v2.py*. Examine it with the following command:

```
more snapshotter_v2.py
```

    This command is a simple script written in the Python programming language using Boto (version 3), the Python SDK for AWS. The AWS CLI is also written in Boto, which makes writing Python-powered AWS scripts very convenient because Boto is pre-installed on most Amazon EC2 Linux instances.

    The script finds all Amazon EBS volumes associated with the current user's account and takes snapshots of them. It then examines the number of snapshots associated with the volume, sorts the snapshots by date, and removes all but the two most recent snapshots.

51. Before running snapshotter_v2.py, copy the following command and run it from within your instance (replacing *VOLUME-ID* with your volume-id):

```
aws ec2 describe-snapshots --filters "Name=volume-id, Values=VOLUME-ID"
--query 'Snapshots[*].SnapshotId'
```

You should see multiple snapshot IDs returned for the volume. These are the snapshots that were created by your cron job before you terminated it.

52. Run the snapshotter_v2.py script:

```
python3 snapshotter_v2.py
```

The script should run for a few seconds, and then return a list of all of the snapshots that it deleted:

```
[ec2-user@ip-\*]$ python3 snapshotter_v2.py
Deleting snapshot snap-e8128a20
Deleting snapshot snap-d0d34818
Deleting snapshot snap-ded14a16
Deleting snapshot snap-e8d74c20
Deleting snapshot snap-25d54eed
Deleting snapshot snap-4acb5082
```

53. To examine the new number of snapshots for the current volume, re-run the command from the procedure above:

```
aws ec2 describe-snapshots --filters "Name=volume-id, Values=<volume-
id>" --query 'Snapshots[*].SnapshotId'
```

You should see only two snapshot IDs returned.

54. Quit your SSH connection of **Command Host**.