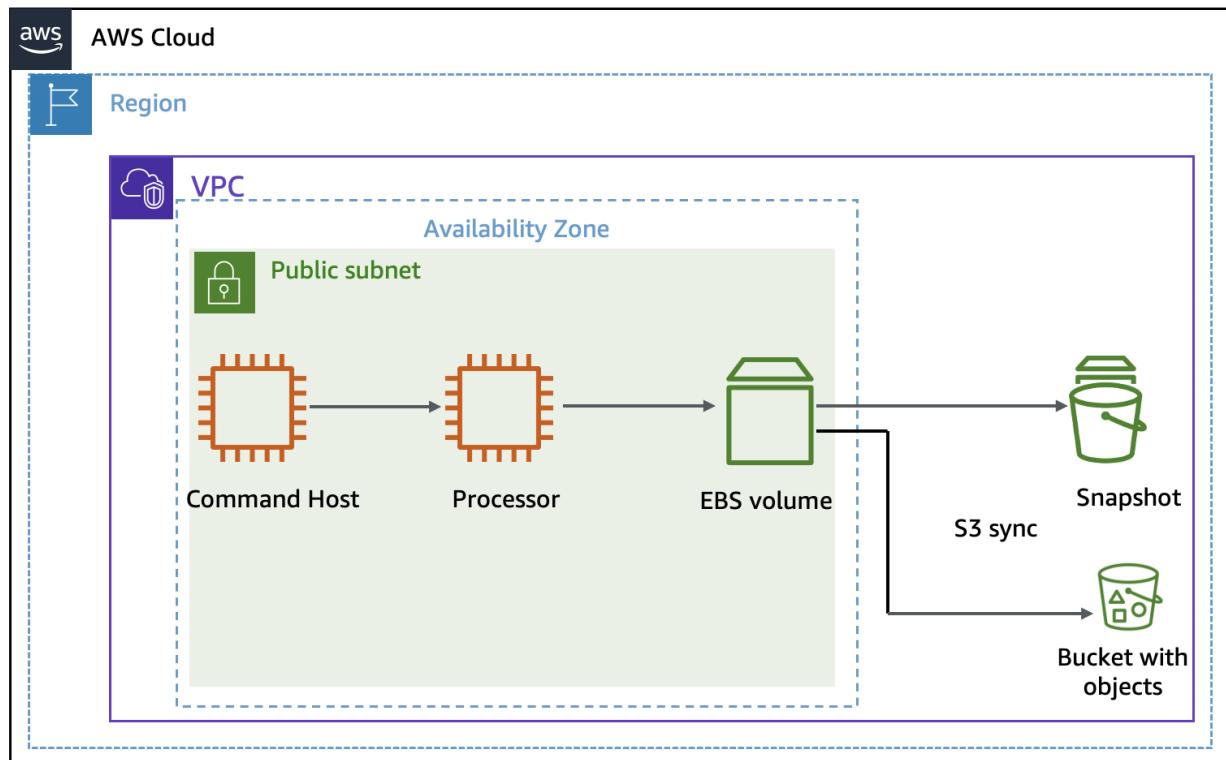


Managing Storage

Lab overview

AWS provides multiple ways to manage data on Amazon Elastic Block Store (Amazon EBS) volumes. In this lab, you use AWS Command Line Interface (AWS CLI) to create snapshots of an EBS volume and configure a scheduler to run Python scripts to delete older snapshots.

In the challenge section of this lab, you are challenged to sync the contents of a directory on an EBS volume to an Amazon Simple Storage Service (Amazon S3) bucket using an Amazon S3 sync command.



Your lab environment consists of a virtual private cloud (VPC) with a public subnet. Amazon Elastic Compute Cloud (Amazon EC2) instances named "Command Host" and "Processor" have already been created in this VPC for you as part of this lab.

The "Command Host" instance will be used to administer AWS resources including the "Processor" instance.

Objectives

By the end of this lab, you will be able to do the following:

- Create and maintain snapshots for Amazon EC2 instances.
- Use Amazon S3 sync to copy files from an EBS volume to an S3 bucket.
- Use Amazon S3 versioning to retrieve deleted files.

Duration

This lab will require approximately **45 minutes** to complete.

Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.
A Start Lab panel opens displaying the lab status.
2. Wait until you see the message "**Lab status: ready**", then choose X to close the Start Lab panel.
3. At the top of these instructions, choose AWS.
This opens the AWS Management Console in a new browser tab. The system automatically logs you in.
Tip: If a new browser tab doesn't open, there's typically a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time to make it easier to follow the lab steps.

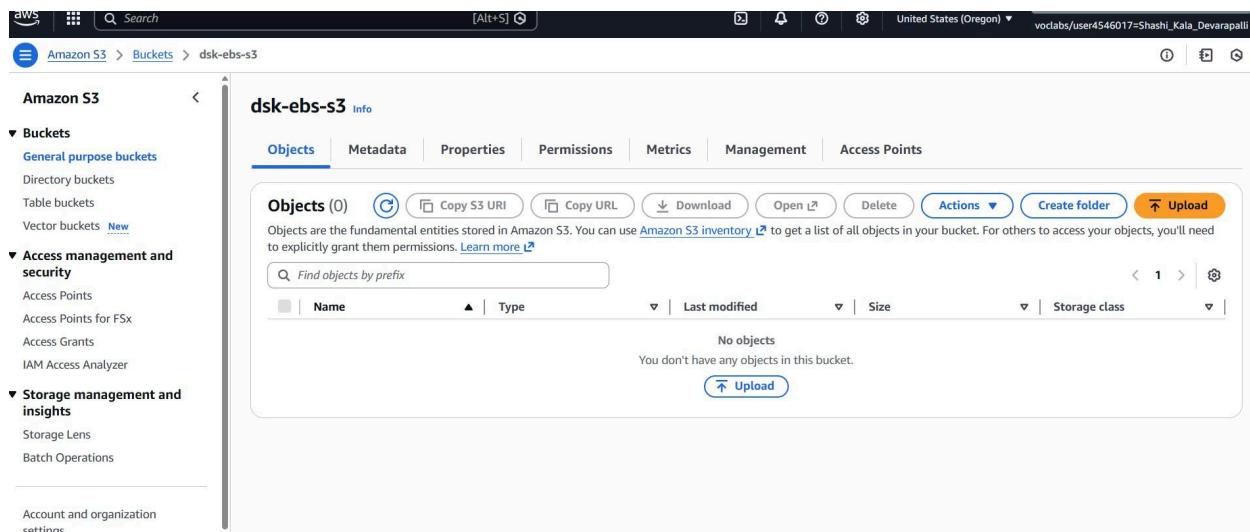
Task 1: Creating and configuring resources

In this task, you create an Amazon S3 bucket and configure the "Command Host" EC2 instance to have secure access to other AWS resources.

Task 1.1: Create an S3 bucket

In this task, you create an S3 bucket to sync files from an EBS volume.

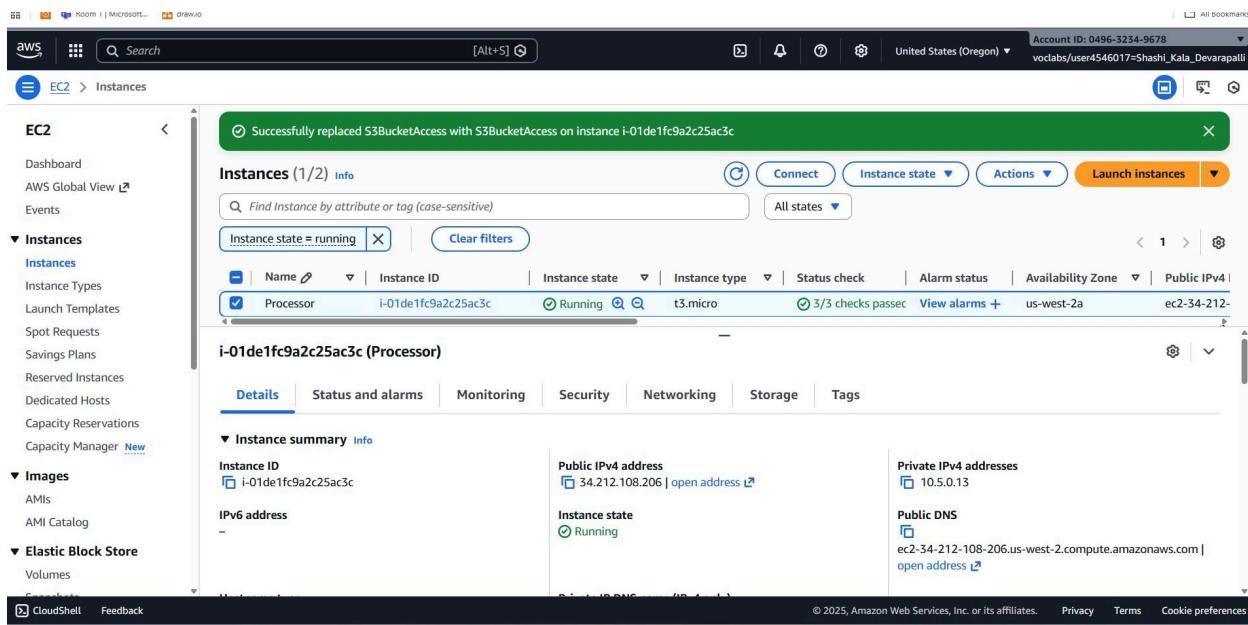
5. On the **AWS Management Console**, in the **Search** bar, enter and choose **S3** to open the **S3 Management Console**.
6. On the console, choose **Create bucket**.
7. In the **Create bucket** section, configure the following:
 - **Bucket name:** Enter a bucket name. Use a combination of characters and numbers to keep it unique.
This will be referred to as "s3-bucket-name" throughout the lab.
 - **Region:** Leave as default.
8. Scroll and choose **Create bucket**.



Task 1.2: Attach instance profile to Processor

In this task, you attach a pre-created IAM role as an instance profile to the EC2 instance "Processor," giving it the permissions to interact with other AWS services such as EBS volumes and S3 buckets.

9. On the **AWS Management Console**, in the **Search** bar, enter and choose **EC2** to open the **EC2 Management Console**.
10. In the navigation pane, choose **Instances**.
11. Choose **Processor** from the list of EC2 instances.
12. Choose **Actions > Security > Modify IAM role**.
13. Choose the **S3BucketAccess** role in the **IAM role** dropdown list.
14. Choose **Update IAM role**.



Task 2: Taking snapshots of your instance

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

Task 2.1: Connecting to the Command Host EC2 instance

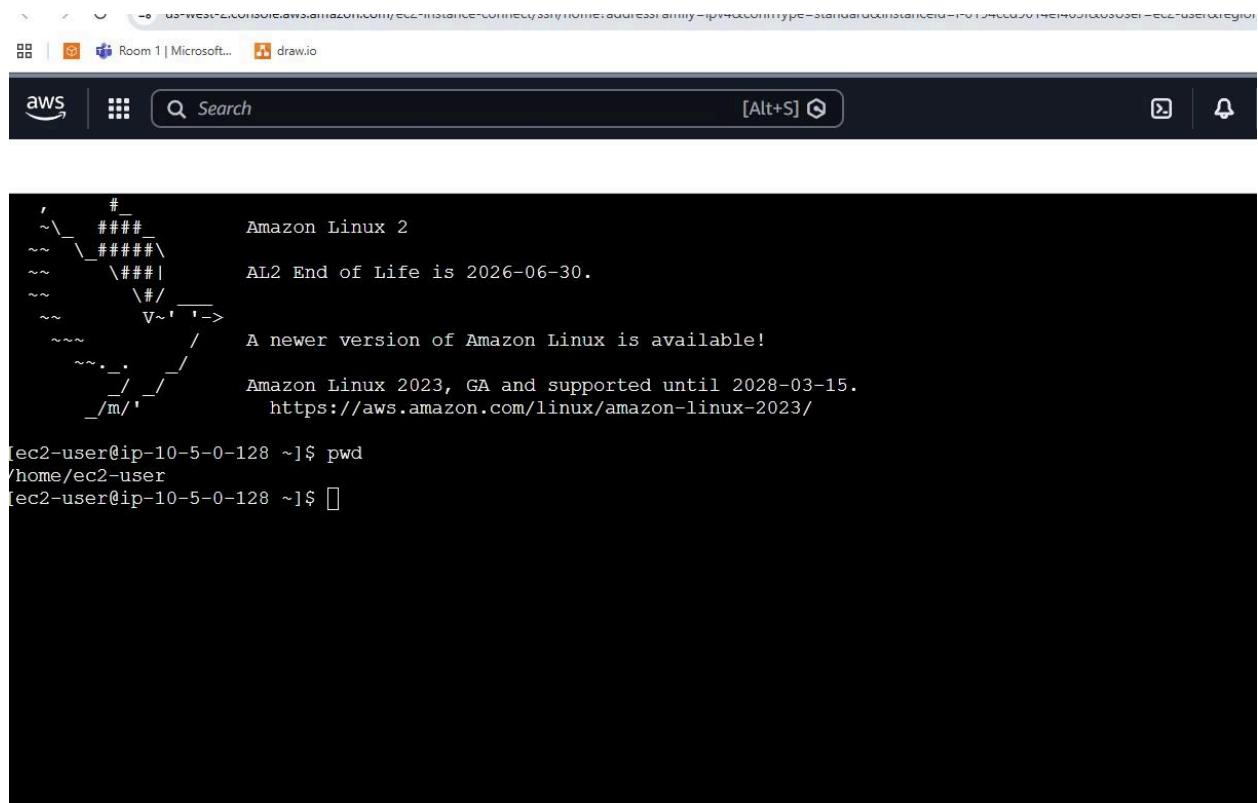
In this task, you use EC2 Instance Connect to connect to the "Command Host" EC2 instance.

15. On the **AWS Management Console**, in the **Search** bar, enter and choose **EC2** to open the **EC2 Management Console**.
16. In the navigation pane, choose **Instances**.
17. From the list of instances, choose **Command Host**.
18. Choose **Connect**.
19. On the **EC2 Instance Connect** tab, choose **Connect**.

This option opens a new browser tab with the **EC2 Instance Connect** terminal window.

Note: If you prefer to use an SSH client to connect to the EC2 instance, see the guidance provided in the additional references section.

You use this terminal window to complete the tasks throughout the lab. If the terminal becomes unresponsive, refresh the browser or use the steps in this task to connect again.



```
#  
~\_\#\#\#_      Amazon Linux 2  
~~\_#\#\#\#\_\_ AL2 End of Life is 2026-06-30.  
~~  \#\#\|  
~~    \|/  
~~     V~'`-->  
~~   /  
~~  / /  
~~  / / A newer version of Amazon Linux is available!  
~~  / /  
~~  / / Amazon Linux 2023, GA and supported until 2028-03-15.  
~~  / / https://aws.amazon.com/linux/amazon-linux-2023/  
  
[ec2-user@ip-10-5-0-128 ~]$ pwd  
/home/ec2-user  
[ec2-user@ip-10-5-0-128 ~]$ 
```

Task 2.2: Taking an initial snapshot

In this task, you identify the EBS volume that's attached to the "Processor" instance and take an initial snapshot. To do so, you run commands in the **EC2 Instance Connect terminal window**. You can copy the command output to a text editor for subsequent use.

20. To display the EBS volume-id, run the following command:

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

Note: The command returns a response similar to this: "VolumeId": "vol-1234abcd".

You use this value for VolumeId throughout the lab steps when prompted.

21. Next, you take snapshot of this volume. Prior to this, you shut down the "Processor" instance, which requires its instance ID. Run the following command to obtain the instance ID:

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

23. The command returns a value for INSTANCE-ID similar to this:

"i-0b06965263c7ac08f"

24. To shut down the "Processor" instance, run the following command and replace "INSTANCE-ID" with the instance-id that you retrieved earlier:

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

26. To verify that the "Processor" instance stopped, run the following command and replace "INSTANCE-ID" with your instance id.

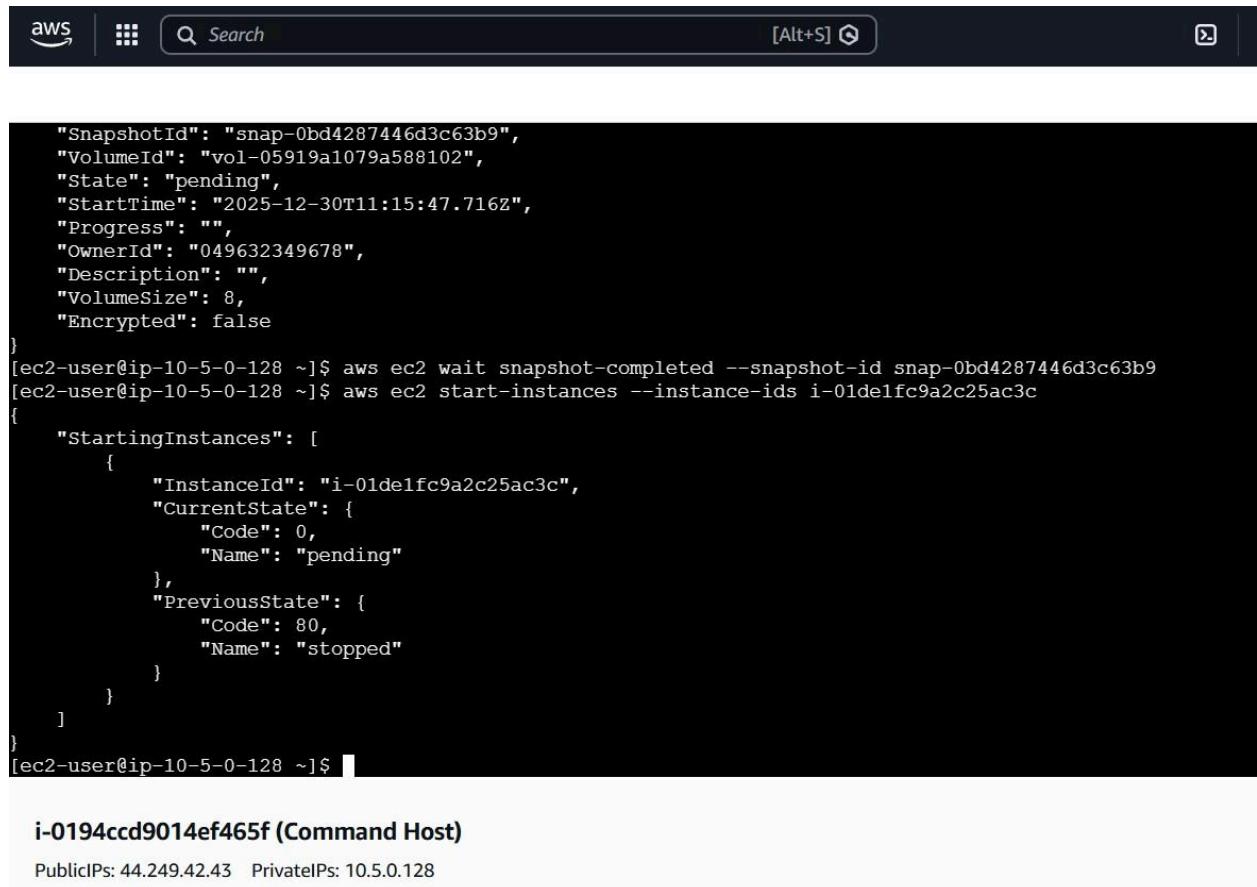
```
27. aws ec2 wait instance-stopped --instance-id INSTANCE-ID
```

28. When the instance stops, the command returns to a prompt.

29. To create your first snapshot of the volume of your "Processor" instance, run the following command and replace "VOLUME-ID" with the VolumeId that you retrieved earlier:

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

31. The command returns a set of information that includes a SnapshotId similar to "snap-0643809e73e6cce13"
32. To check the status of your snapshot, run the following command and replace "SNAPSHOT-ID" with the SnapshotId that you retrieved earlier:
33. aws ec2 wait snapshot-completed --snapshot-id SNAPSHOT-ID
34. Continue to next steps after the command returns to the prompt.
35. To restart the "Processor" instance, run the following command and replace "INSTANCE-ID" with the instance-id that you retrieved earlier:
36. aws ec2 start-instances --instance-ids INSTANCE-ID
37. After a couple minutes, the instance will be in the running state.



The screenshot shows the AWS Cloud9 IDE interface. At the top, there's a navigation bar with the AWS logo, a search bar, and keyboard shortcuts [Alt+S] and [Ctrl+F]. Below the terminal window, there's a status bar showing the instance ID i-0194ccd9014ef465f (Command Host) and its public and private IP addresses.

```

"SnapshotId": "snap-0bd4287446d3c63b9",
"VolumeId": "vol-05919a1079a588102",
"State": "pending",
"StartTime": "2025-12-30T11:15:47.716Z",
"Progress": "",
"OwnerId": "049632349678",
"Description": "",
"VolumeSize": 8,
"Encrypted": false
}
[ec2-user@ip-10-5-0-128 ~]$ aws ec2 wait snapshot-completed --snapshot-id snap-0bd4287446d3c63b9
[ec2-user@ip-10-5-0-128 ~]$ aws ec2 start-instances --instance-ids i-01de1fc9a2c25ac3c
{
  "StartingInstances": [
    {
      "InstanceId": "i-01de1fc9a2c25ac3c",
      "CurrentState": {
        "Code": 0,
        "Name": "pending"
      },
      "PreviousState": {
        "Code": 80,
        "Name": "stopped"
      }
    }
  ]
}
[ec2-user@ip-10-5-0-128 ~]$ █

```

i-0194ccd9014ef465f (Command Host)
Public IPs: 44.249.42.43 Private IPs: 10.5.0.128

Task 2.3 Scheduling the creation of subsequent snapshots

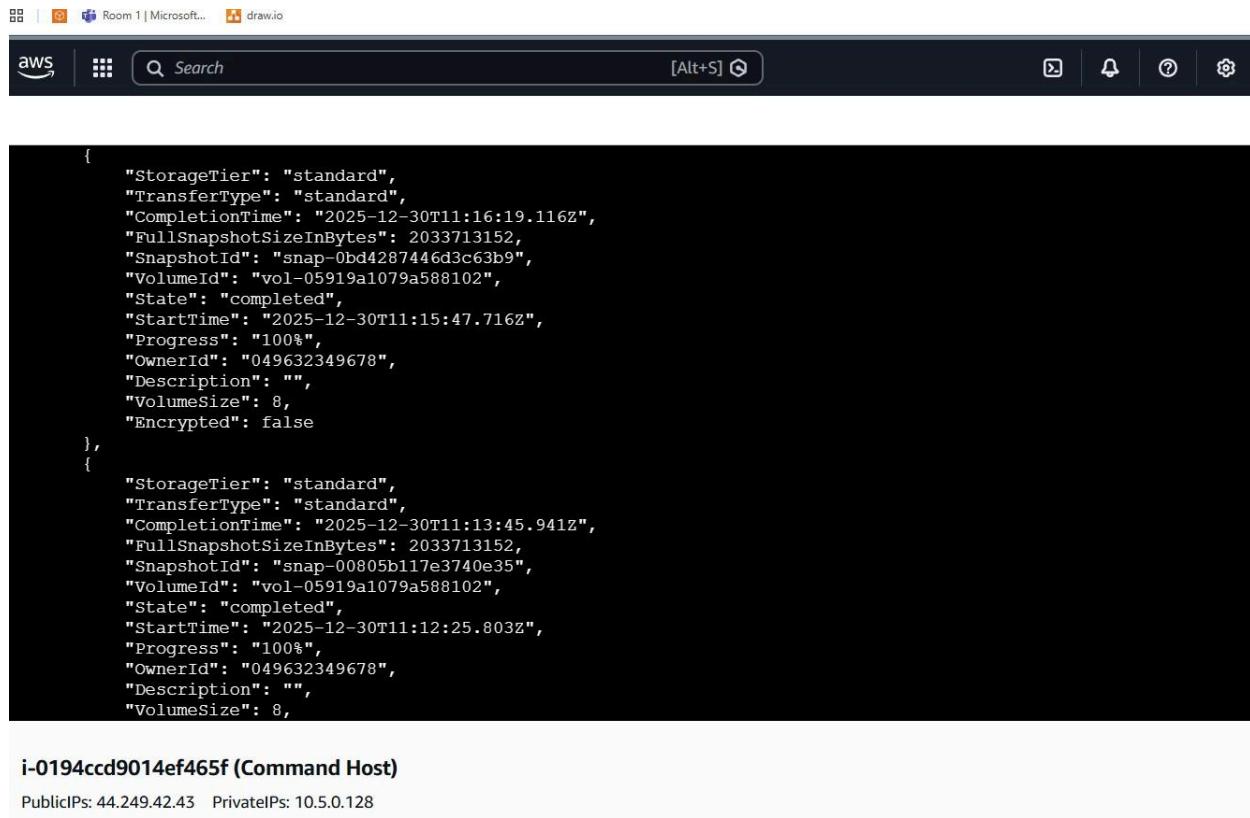
Using the Linux scheduling system (cron), you can set up a recurring snapshot process so that new snapshots of your data are taken automatically.

For the purposes of this lab, you schedule a snapshot creation every minute so that you can verify the results of your work.

In this task, you create a cron job to manage the number of snapshots that are maintained for a volume.

Note: This section of the lab doesn't stop the instance in order to create a large number of snapshots for the next step.

27. To create and schedule a cron entry that runs a job every minute, run the following command and replace "VOLUME-ID" with the Volumeld that you retrieved earlier:
 28. `echo "* * * * * aws ec2 create-snapshot --volume-id VOLUME-ID`
`2>&1 >> /tmp/cronlog" > cronjob`
`crontab cronjob`
 29. **Note:** This will take 1-2 minutes.
 30. To verify that subsequent snapshots are being created, run the following command and replace "VOLUME-ID" with the Volumeld that you retrieved earlier:
 31. `aws ec2 describe-snapshots --filters`
`"Name=volume-id,Values=VOLUME-ID"`
 32. Re-run the command after few minutes to see more snapshots.
 33. Wait a few minutes so that a few more snapshots are generated before beginning the next task.



```
{  
    "StorageTier": "standard",  
    "TransferType": "standard",  
    "CompletionTime": "2025-12-30T11:16:19.116Z",  
    "FullSnapshotSizeInBytes": 2033713152,  
    "SnapshotId": "snap-0bd4287446d3c63b9",  
    "VolumeId": "vol-05919a1079a588102",  
    "State": "completed",  
    "StartTime": "2025-12-30T11:15:47.716Z",  
    "Progress": "100%",  
    "OwnerId": "049632349678",  
    "Description": "",  
    "VolumeSize": 8,  
    "Encrypted": false  
},  
{  
    "StorageTier": "standard",  
    "TransferType": "standard",  
    "CompletionTime": "2025-12-30T11:13:45.941Z",  
    "FullSnapshotSizeInBytes": 2033713152,  
    "SnapshotId": "snap-00805b117e3740e35",  
    "VolumeId": "vol-05919a1079a588102",  
    "State": "completed",  
    "StartTime": "2025-12-30T11:12:25.803Z",  
    "Progress": "100%",  
    "OwnerId": "049632349678",  
    "Description": "",  
    "VolumeSize": 8,  
    "Encrypted": false  
}  
  
i-0194ccd9014ef465f (Command Host)  
PublicIPs: 44.249.42.43 PrivateIPs: 10.5.0.128
```

Task 2.4 Retaining the last two snapshots

In this task, you run a Python script that maintains only the last two snapshots for any given EBS volume.

30. To stop the cron job, run the following command:
31. `crontab -r`
32. To examine the contents of the Python script "snapshotter_v2.py", run the following command:
33. `more /home/ec2-user/snapshotter_v2.py`
34. The script finds all EBS volumes that are associated with the current user's account and takes snapshots. It then examines the number of snapshots that are associated with the volume, sorts the snapshots by date, and removes all but the two most recent snapshots.
35. Before running snapshotter_v2.py, run the following command and replace "VOLUME-ID" with the VolumeId that you retrieved earlier:
36. `aws ec2 describe-snapshots --filters "Name=volume-id, Values=VOLUME-ID" --query 'Snapshots[*].SnapshotId'`

37. The command returns the multiple snapshot IDs that were returned for the volume. These are the snapshots that were created by your cron job before you stopped it.
38. Run the the "snapshotter_v2.py" script using following command:
39. `python3.8 snapshotter_v2.py`

The script runs for a few seconds, and then it returns a list of all of the snapshots that it deleted:

```
[ec2-user@ip-\*]$ python3.8 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
```

40.

41. To examine the new number of snapshots for the current volume, re-run the following command from an earlier step:

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

43. The command returns only **two** snapshot IDs.

```
Last login: Tue Dec 30 10:59:16 2025 from ec2-18-237-140-165.us-west-2.compute.amazonaws.com
'`#_
~\_\ #####
~- \###\
-- \### AL2 End of Life is 2026-06-30.
~~ \#/
~~ V-' --> A newer version of Amazon Linux is available!
~~ / Amazon Linux 2023, GA and supported until 2028-03-15.
~/`/ https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-10-5-0-128 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-5-0-128 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=vol-05919a1079a588102" --query 'Snapshots[*].SnapshotId'
[
    "snap-0bd4287446d3c63b9",
    "snap-00805b117e3740e35"
]
[ec2-user@ip-10-5-0-128 ~]$ python3.8 snapshotter_v2.py
Deleting snapshot snap-00805b117e3740e35
[ec2-user@ip-10-5-0-128 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=vol-05919a1079a588102" --query 'Snapshots[*].SnapshotId'
[
    "snap-0d62ace385c3f7a66",
    "snap-0bd4287446d3c63b9"
]
[ec2-user@ip-10-5-0-128 ~]$ 
```

i-0194ccd9014ef465f (Command Host)
Public IPs: 44.249.42.43 Private IPs: 10.5.0.128

Task 3: Challenge: Synchronize files with Amazon S3

In this task, you are challenged to sync the contents of a directory with the Amazon S3 bucket that you created earlier.

Note: It's recommended that you try this challenge by yourself using the information that's provided in this section **before** reading the detailed solution. When you complete the challenge, check your work by reviewing the detailed solution.

Challenge Description

Run the following command in the terminal to download a sample set of files:

```
wget
https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-RSJAWS-3-1246
27/183-lab-JAWS-managing-storage/s3/files.zip
```

Unzip these files, and then, using the AWS CLI, figure out how to accomplish the following:

- Activate versioning for your Amazon S3 bucket.
- Use a single AWS CLI command to sync the contents of your unzipped folder with your Amazon S3 bucket.

- Modify the command so that it deletes a file from Amazon S3 when the corresponding file is deleted locally on your instance.
- Recover the deleted file from Amazon S3 using versioning.

You can use the solution summary as a guide to complete the challenge yourself. Use the links in additional references section for details on using required AWS CLI commands.

Solution Summary

The solution involves the following steps:

- To activate versioning for the bucket, use the `aws s3api put-bucket-versioning` command.
- To synchronize the local files with Amazon S3, use the `aws s3 sync` command on the local folder.
- Delete a local file.
- To force Amazon S3 to delete any files that aren't present on the local drive but present in Amazon S3, use the `--delete` option with the `aws s3 sync` command.
- Because there's no direct command in Amazon S3 to restore a previous version of a file, to download a previous version of the deleted file from Amazon S3, use the `aws s3api list-object-versions` and `aws s3api get-object` commands. You can then restore the file to Amazon S3 by using `aws s3 sync`.

Task 3.1: Downloading and unzipping sample files

The sample file package contains a folder with three text files: file1.txt, file2.txt, and file3.txt. These are the files that you will sync with your Amazon S3 bucket.

35. Connect to the "Processor" instance using EC2 Instance Connect.

Note: Refer to the earlier steps that you used to connect to the "Command Host" instance.

You run following AWS CLI commands in the EC2 Instance Connect terminal window.

36. To download the sample files on the "Processor" instance, run the following command from within your instance:

37. `wget`

```
https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-1
00-RSJAWS-3-124627/183-lab-JAWS-managing-storage/s3/files.zip
```

38. To unzip the directory, use the following command:

39. `unzip files.zip`

```

[ec2-user@ip-10-5-0-128 ~]$ get https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-RSJAWS-3-124627/183-lab-JAWS-managing-storage/s3/files.zip
-bash: get: command not found
[ec2-user@ip-10-5-0-128 ~]$ wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-RSJAWS-3-124627/183-lab-JAWS-managing-storage/s3/files.zip
--2025-12-30 11:59:04-- https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-RSJAWS-3-124627/183-lab-JAWS-managing-storage/s3/files.zip
Resolving aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)... 52.218.216.81, 3.5.76.126, 52.92.204.74, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)|52.218.216.81|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72110 (70K) [application/zip]
Saving to: 'files.zip'

100%[=====] 72,110      --.-K/s   in 0.001s

2025-12-30 11:59:04 (48.4 MB/s) - 'files.zip' saved [72110/72110]

[ec2-user@ip-10-5-0-128 ~]$ ls
cronjob  files.zip  get-pip.py  snapshotter_v2.py
[ec2-user@ip-10-5-0-128 ~]$ unzip files.zip
Archive: files.zip
  inflating: files/file1.txt
  inflating: files/file2.txt
  inflating: files/file3.txt
[ec2-user@ip-10-5-0-128 ~]$ ls
cronjob  files.zip  get-pip.py  snapshotter_v2.py
[ec2-user@ip-10-5-0-128 ~]$ aws s3api get-bucket-versioning --bucket your-bucket-name
An error occurred (AccessDenied) when calling the GetBucketVersioning operation: Access Denied
[ec2-user@ip-10-5-0-128 ~]$ Use code with caution,
[bash: Use: command not found

i-0194cccd9014ef465f (Command Host)
PublicIPs: 44.249.42.43 PrivateIPs: 10.5.0.128

```

Task 3.2: Syncing files

38. Before syncing content with your Amazon S3 bucket, you need to activate versioning on your bucket.

Run the following command and replace "S3-BUCKET-NAME" with your bucket name:

39. `aws s3api put-bucket-versioning --bucket S3-BUCKET-NAME --versioning-configuration Status=Enabled`

40. To sync the contents of the files folder with your Amazon S3 bucket, run the following command and replace "S3-BUCKET-NAME" with your bucket name:

41. `aws s3 sync files s3://S3-BUCKET-NAME/files/`

42. The command confirms that three files were uploaded to your S3 bucket.

43. To confirm the state of your files, run the following command and replace "S3-BUCKET-NAME" with your bucket name:

44. `aws s3 ls s3://S3-BUCKET-NAME/files/`

45. To delete one of the files on the local drive, run the following command:

46. `rm files/file1.txt`

47. To delete the same file from the S3 bucket, use the `--delete` option with the `aws s3 sync` command.

Run the following command and replace "S3-BUCKET-NAME" with your bucket name:

48. `aws s3 sync files s3://S3-BUCKET-NAME/files/ --delete`

The command returns a message similar to following:

49. `delete: s3://examplebucketname/files/file1.txt`
50. To verify that the file was deleted from the bucket, run the following command and replace "S3-BUCKET-NAME" with your bucket name:
`aws s3 ls s3://S3-BUCKET-NAME/files/`
52. Now, try to recover the old version of file1.txt. To view a list of previous versions of this file, run the following command and replace "S3-BUCKET-NAME" with your bucket name:
`aws s3api list-object-versions --bucket S3-BUCKET-NAME --prefix files/file1.txt`
54. The output contains a `DeleteMarkers` and a `Versions` block. `DeleteMarkers` indicates where the delete marker is. For example, if you perform an `aws s3 rm` operation (or an `aws s3 sync` operation with the `--delete` option), this is the next version that the file will revert to.
The `Versions` block contains a list of all available versions. You should have only a single `Versions` entry. Note the value for `VersionId` for use later.
55. Because there's no direct command to restore an older version of an Amazon S3 object to its own bucket, you need to re-download the old version and sync again to Amazon S3.
To download the previous version of file1.txt, run the following command and replace "S3-BUCKET-NAME" with your bucket name:
`aws s3api get-object --bucket S3-BUCKET-NAME --key files/file1.txt --version-id VERSION-ID files/file1.txt`
57. To verify that the file was restored locally, run the following command:
`ls files`
59. The command shows all three files listed.
60. To re-sync the contents of the files/ folder to Amazon S3, run the following command from within your instance and replace "S3-BUCKET-NAME" with your bucket name:
`aws s3 sync files s3://S3-BUCKET-NAME/files/`
62. Finally, to verify that a new version of file1.txt was pushed to the S3 bucket, run the following command and replace "S3-BUCKET-NAME" with your bucket name:
`aws s3 ls s3://S3-BUCKET-NAME/files/`

Conclusion

Congratulations! You successfully completed the following:

- Created and maintain snapshots for Amazon EC2 instances.
- Used Amazon S3 sync to copy files from an EBS volume to an S3 bucket.
- Used Amazon S3 versioning to retrieve deleted files.

```
        }
    },
    "RequestCharged": null,
    "Prefix": "files/file1.txt"

ec2-user@ip-10-5-0-128 ~]$ aws s3api get-object --bucket dsk-ebs-s3 --key files/file1.txt --version-id VERSION-ID files/file1.txt
An error occurred (InvalidArgument) when calling the GetObject operation: Invalid version id specified
ec2-user@ip-10-5-0-128 ~]$ aws s3api get-object --bucket dsk-ebs-s3 --key files/file1.txt --version-id 6T0dSq56Se1bKCb7wOWJj9xH2r1JEz8S files/file1.txt
An error occurred (MethodNotAllowed) when calling the GetObject operation: The specified method is not allowed against this resource.
ec2-user@ip-10-5-0-128 ~$ ls
ronjob files files.zip get-pip.py snapshotter_v2.py
ec2-user@ip-10-5-0-128 ~$ ls files
file2.txt file3.txt
ec2-user@ip-10-5-0-128 ~]$ aws s3api get-object --bucket dsk-ebs-s3 --key files/file1.txt --version-id 6T0dSq56Se1bKCb7wOWJj9xH2r1JEz8S files/file1.txt
An error occurred (MethodNotAllowed) when calling the GetObject operation: The specified method is not allowed against this resource.
ec2-user@ip-10-5-0-128 ~]$ aws s3api get-object --bucket dsk-ebs-s3 --key files/file1.txt --version-id 6T0dSq56Se1bKCb7wOWJj9xH2r1JEz8S files/file1.txt
An error occurred (MethodNotAllowed) when calling the GetObject operation: The specified method is not allowed against this resource.
ec2-user@ip-10-5-0-128 ~$ aws s3 sync files s3://S3-BUCKET-NAME/files/
fatal error: An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
ec2-user@ip-10-5-0-128 ~$ aws s3 sync files s3://dsk-ebs-s3/files/
ec2-user@ip-10-5-0-128 ~$ aws s3 ls s3://dsk-ebs-s3/files/
025-12-30 12:07:34      43784 file2.txt
025-12-30 12:07:34      96675 file3.txt
ec2-user@ip-10-5-0-128 ~$ 
```

i-0194cccd9014ef465f (Command Host)

Public IPs: 44.249.42.43 Private IPs: 10.5.0.128