

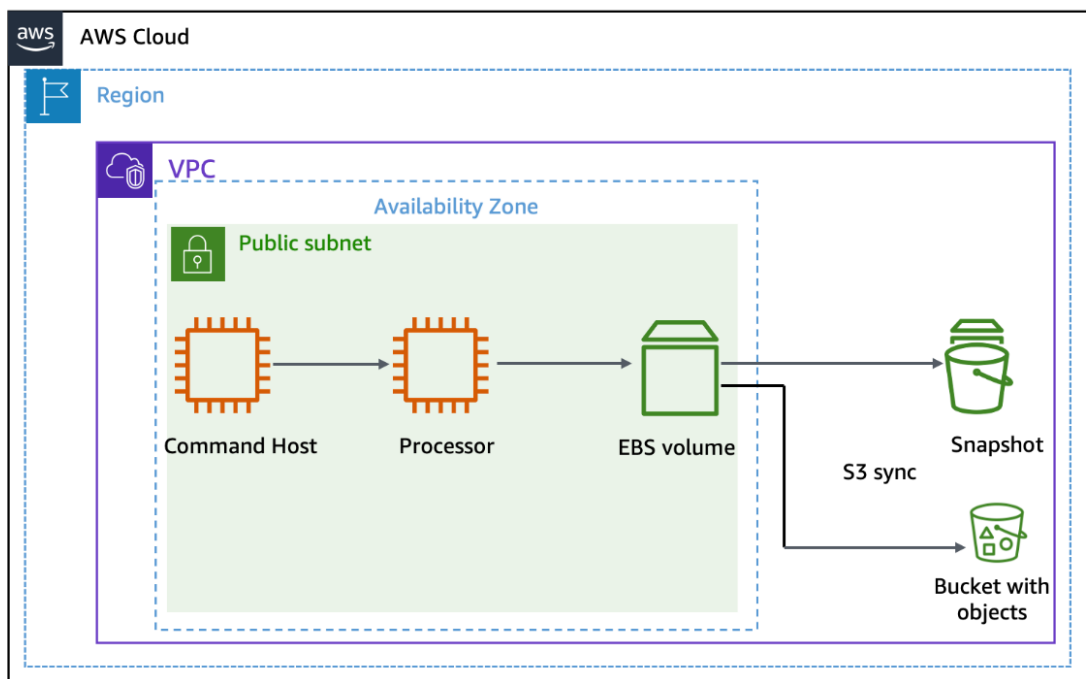
Lab - Managing Storage

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.

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:
 - a. **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.

- b. **Region:** Leave as default.
8. Scroll and choose **Create bucket**.

Amazon S3 > Buckets

Successfully created bucket "s3-bucket-mariana"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

General purpose buckets All AWS Regions Directory buckets

General purpose buckets (1) [Info](#) [Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
s3-bucket-mariana	US West (Oregon) us-west-2	September 15, 2025, 14:16:23 (UTC+01:00)

Account snapshot [Info](#) [View dashboard](#)

Updated daily

Storage Lens provides visibility into storage usage and activity trends.

External access summary - new [Info](#)

Updated daily

External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

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.

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

EC2 > Instances

Dashboard
EC2 Global View
Events

▼ **Instances**

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

▼ **Images**

- AMIs
- AMI Catalog

▼ **Elastic Block Store**

- Volumes
- Snapshots
- Lifecycle Manager

▼ **Network & Security**

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs

Successfully replaced S3BucketAccess with S3BucketAccess on instance i-09202d8b8cd333c7c

Instances (1/2) 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
<input checked="" type="checkbox"/>	Processor	i-09202d8b8cd333c7c	Running	t3.micro
<input type="checkbox"/>	Command Host	i-0e6990af9483112ce	Running	t3.medium

i-09202d8b8cd333c7c (Processor)

< Details Status and alarms Monitoring Security Netv >

▼ **Instance summary** Info

Instance ID i-09202d8b8cd333c7c	Public IPv4 address 34.220.50.75 open address
Private IPv4 addresses 10.5.0.252	IPv6 address -
Instance state Running	Public DNS ec2-34-220-50-75.us-west-2.compute.amazonaws.com open address

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**.

19. On the **EC2 Instance Connect** tab, choose **Connect**.

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

```

      #
    ~\  #####
  ~ ~\  #####\
  ~ ~\  #####|
  ~ ~\  \#/
  ~ ~\  V~' '->
    ~ ~
      ~ ~.
        ~\
      _/m/'

Amazon Linux 2

AL2 End of Life is 2026-06-30.

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-10-5-0-219 ~]$

```

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:


```

~\      #_
~\      ###
~\      #####\
~\      \###|
~\      \#/
~\      V~' '->
~\      /
~\      /
~\      /m/'

Amazon Linux 2
AL2 End of Life is 2026-06-30.
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-10-5-0-219 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:VolumeId}'
{
  "VolumeId": "vol-0d307749e3aa7aee0"
}

[ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].InstanceId'
'i-09202d8b8cd333c7c'
[ec2-user@ip-10-5-0-219 ~]$

```

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

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

```

~\      #_
~\      ###
~\      #####\
~\      \###|
~\      \#/
~\      V~' '->
~\      /
~\      /
~\      /m/'

Amazon Linux 2
AL2 End of Life is 2026-06-30.
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-10-5-0-219 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:VolumeId}'
{
  "VolumeId": "vol-0d307749e3aa7aee0"
}

[ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].InstanceId'
'i-09202d8b8cd333c7c'
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 stop-instances --instance-ids i-09202d8b8cd333c7c
{
  "StoppingInstances": [
    {
      "InstanceId": "i-09202d8b8cd333c7c",
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}

[ec2-user@ip-10-5-0-219 ~]$

```

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

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

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

```

#
##### Amazon Linux 2
#####
##### AL2 End of Life is 2026-06-30.
#####
##### 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-10-5-0-219 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.[VolumeId:VolumeId]'
{
  "VolumeId": "vol-0d307749e3aa7aee0"
}
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].InstanceId'
"i-09202d8b8cd333c7c"
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 stop-instances --instance-ids i-09202d8b8cd333c7c
{
  "StoppingInstances": [
    {
      "InstanceId": "i-09202d8b8cd333c7c",
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 wait instance-stopped --instance-id i-09202d8b8cd333c7c
[ec2-user@ip-10-5-0-219 ~]$
```

24. 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:

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

The command returns a set of information that includes a SnapshotId similar to "snap-0643809e73e6cce13"


```

_/_/_/ Amazon Linux 2023, GA and supported until 2028-03-15.
_/m/'_/_/ https://aws.amazon.com/linux/amazon-linux-2023/

ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor'
'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:VolumeId}'

    "VolumeId": "vol-0d307749e3aa7aee0"

ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor'
y 'Reservations[0].Instances[0].InstanceId'
i-09202d8b8cd333c7c"
ec2-user@ip-10-5-0-219 ~]$ aws ec2 stop-instances --instance-ids i-09202d8b8cd333c7c

    "StoppingInstances": [
      {
        "InstanceId": "i-09202d8b8cd333c7c",
        "CurrentState": {
          "Code": 64,
          "Name": "stopping"
        },
        "PreviousState": {
          "Code": 16,
          "Name": "running"
        }
      }
    ]

ec2-user@ip-10-5-0-219 ~]$ aws ec2 wait instance-stopped --instance-id i-09202d8b8cd333c7c
ec2-user@ip-10-5-0-219 ~]$ aws ec2 create-snapshot --volume-id vol-0d307749e3aa7aee0

    "Tags": [],
    "SnapshotId": "snap-0ef7348c2f887108c",
    "VolumeId": "vol-0d307749e3aa7aee0",
    "State": "pending",
    "StartTime": "2025-09-15T13:26:04.545Z",
    "Progress": "",
    "OwnerId": "080900507702",
    "Description": "",
    "VolumeSize": 8,
    "Encrypted": false

ec2-user@ip-10-5-0-219 ~]$

```

: 0a6000a40487112aa (Command Hash)

25. To check the status of your snapshot, run the following command and replace "SNAPSHOT-ID" with the SnapshotId that you retrieved earlier:

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

Continue to next steps after the command returns to the prompt.

```

ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' --q
y 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:VolumeId}'

    "VolumeId": "vol-0d307749e3aa7aee0"

ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor' --q
y 'Reservations[0].Instances[0].InstanceId'
'i-09202d8b8cd333c7c'
ec2-user@ip-10-5-0-219 ~]$ aws ec2 stop-instances --instance-ids i-09202d8b8cd333c7c

    "StoppingInstances": [
      {
        "InstanceId": "i-09202d8b8cd333c7c",
        "CurrentState": {
          "Code": 64,
          "Name": "stopping"
        },
        "PreviousState": {
          "Code": 16,
          "Name": "running"
        }
      }
    ]

ec2-user@ip-10-5-0-219 ~]$ aws ec2 wait instance-stopped --instance-id i-09202d8b8cd333c7c
ec2-user@ip-10-5-0-219 ~]$ aws ec2 create-snapshot --volume-id vol-0d307749e3aa7aee0

    "Tags": [],
    "SnapshotId": "snap-0ef7348c2f887108c",
    "VolumeId": "vol-0d307749e3aa7aee0",
    "State": "pending",
    "StartTime": "2025-09-15T13:26:04.545Z",
    "Progress": "",
    "OwnerId": "080900507702",
    "Description": "",
    "VolumeSize": 8,
    "Encrypted": false

ec2-user@ip-10-5-0-219 ~]$ aws ec2 wait snapshot-completed --snapshot-id snap-0ef7348c2f887108c
ec2-user@ip-10-5-0-219 ~]$

```

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

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

After a couple minutes, the instance will be in the running state.

```

        "PreviousState": {
            "Code": 16,
            "Name": "running"
        }
    }
}
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 wait instance-stopped --instance-id i-09202d8b8cd333c7c
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 create-snapshot --volume-id vol-0d307749e3aa7aee0
{
    "Tags": [],
    "SnapshotId": "snap-0ef7348c2f887108c",
    "VolumeId": "vol-0d307749e3aa7aee0",
    "State": "pending",
    "StartTime": "2025-09-15T13:26:04.545Z",
    "Progress": "",
    "OwnerId": "080900507702",
    "Description": "",
    "VolumeSize": 8,
    "Encrypted": false
}
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 wait snapshot-completed --snapshot-id snap-0ef7348c2f887108c
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 start-instances --instance-ids i-09202d8b8cd333c7c
{
    "StartingInstances": [
        {
            "InstanceId": "i-09202d8b8cd333c7c",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
[ec2-user@ip-10-5-0-219 ~]$

```

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 VolumeId that you retrieved earlier:

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

Note: This will take 1-2 minutes.

28. To verify that subsequent snapshots are being created, run the following command and replace "VOLUME-ID" with the VolumeId that you retrieved earlier:

```
aws ec2 describe-snapshots --filters "Name=volume-id,Values=VOLUME-ID"
```

Re-run the command after few minutes to see more snapshots.

29. Wait a few minutes so that a few more snapshots are generated before beginning the next task.

```
ec2-user@ip-10-5-0-219 ~]$ echo "* * * * * aws ec2 create-snapshot --volume-id vol-0d307749e3aa7a7aee0 2>&1 >> /tmp/cronlog" > cronjob
ec2-user@ip-10-5-0-219 ~]$ crontab cronjob
ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id,Values=vol-0d307749e3aa7a7aee0"

{
  "Snapshots": [
    {
      "StorageTier": "standard",
      "TransferType": "standard",
      "CompletionTime": "2025-09-15T13:32:38.076Z",
      "FullSnapshotSizeInBytes": 1930952704,
      "SnapshotId": "snap-08813b9cea2d29135",
      "VolumeId": "vol-0d307749e3aa7aee0",
      "State": "completed",
      "StartTime": "2025-09-15T13:32:03.303Z",
      "Progress": "100%",
      "OwnerId": "080900507702",
      "Description": "",
      "VolumeSize": 8,
      "Encrypted": false
    },
    {
      "StorageTier": "standard",
      "TransferType": "standard",
      "CompletionTime": "2025-09-15T13:26:44.716Z",
      "FullSnapshotSizeInBytes": 1930428416,
      "SnapshotId": "snap-0ef7348c2f887108c",
      "VolumeId": "vol-0d307749e3aa7aee0",
      "State": "completed",
      "StartTime": "2025-09-15T13:26:04.545Z",
      "Progress": "100%",
      "OwnerId": "080900507702",
      "Description": "",
      "VolumeSize": 8,
      "Encrypted": false
    }
  ]
}

ec2-user@ip-10-5-0-219 ~]$
```

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:

```
crontab -r
```

31. To examine the contents of the Python script "snapshotter_v2.py", run the following command:

```
more /home/ec2-user/snapshotter_v2.py
```

```
}
[ec2-user@ip-10-5-0-219 ~]$ crontab -r
[ec2-user@ip-10-5-0-219 ~]$ more /home/ec2-user/snapshotter_v2.py
#!/usr/bin/env python

import boto3

MAX_SNAPSHOTS = 2    # Number of snapshots to keep

# Create the EC2 resource
ec2 = boto3.resource('ec2')

# Get a list of all volumes
volume_iterator = ec2.volumes.all()

# Create a snapshot of each volume
for v in volume_iterator:
    v.create_snapshot()

# Too many snapshots?
snapshots = list(v.snapshots.all())
if len(snapshots) > MAX_SNAPSHOTS:

    # Delete oldest snapshots, but keep MAX_SNAPSHOTS available
    snap_sorted = sorted([(s.id, s.start_time, s) for s in snapshots], key=lambda k: k[1])
    for s in snap_sorted[:-MAX_SNAPSHOTS]:
        print("Deleting snapshot", s[0])
        s[2].delete()
[ec2-user@ip-10-5-0-219 ~]$
```

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.

32. Before running snapshotter_v2.py, run the following command and replace "VOLUME-ID" with the VolumeId that you retrieved earlier:

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

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.

```
# Get a list of all volumes
volume_iterator = ec2.volumes.all()

# Create a snapshot of each volume
for v in volume_iterator:
    v.create_snapshot()

# Too many snapshots?
snapshots = list(v.snapshots.all())
if len(snapshots) > MAX_SNAPSHOTS:

    # Delete oldest snapshots, but keep MAX_SNAPSHOTS available
    snap_sorted = sorted([(s.id, s.start_time, s) for s in snapshots], key=lambda k: k[1])
    for s in snap_sorted[: -MAX_SNAPSHOTS]:
        print("Deleting snapshot", s[0])
        s[2].delete()

[ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=vol-0d307aa7aee0" --query 'Snapshots[*].SnapshotId'
[
    "snap-08813b9cea2d29135",
    "snap-0ef7348c2f887108c",
    "snap-049231844049afb33"
]
[ec2-user@ip-10-5-0-219 ~]$
```

33. Run the "snapshotter_v2.py" script using following command:

```
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
```

```

snapshots = list(v.snapshots.all())
if len(snapshots) > MAX_SNAPSHOTS:

    # Delete oldest snapshots, but keep MAX_SNAPSHOTS available
    snap_sorted = sorted([(s.id, s.start_time, s) for s in snapshots], key=lambda k: k[1])
    for s in snap_sorted[:-MAX_SNAPSHOTS]:
        print("Deleting snapshot", s[0])
        s[2].delete()
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=vol-0d307749e8aa7aee0" --query 'Snapshots[*].SnapshotId'
[
    "snap-08813b9cea2d29135",
    "snap-0ef7348c2f887108c",
    "snap-049231844049afb33"
]
[ec2-user@ip-10-5-0-219 ~]$ python3.8 snapshotter_v2.py
Deleting snapshot snap-0ef7348c2f887108c
Deleting snapshot snap-08813b9cea2d29135
[ec2-user@ip-10-5-0-219 ~]$ 

```

i-0a6980a69483112ca (Command Host)

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

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

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

```

ec2 = boto3.resource('ec2')

# Get a list of all volumes
volume_iterator = ec2.volumes.all()

# Create a snapshot of each volume
for v in volume_iterator:
    v.create_snapshot()

# Too many snapshots?
snapshots = list(v.snapshots.all())
if len(snapshots) > MAX_SNAPSHOTS:

    # Delete oldest snapshots, but keep MAX_SNAPSHOTS available
    snap_sorted = sorted([(s.id, s.start_time, s) for s in snapshots], key=lambda k: k[1])
    for s in snap_sorted[:-MAX_SNAPSHOTS]:
        print("Deleting snapshot", s[0])
        s[2].delete()
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=vol-0d307749e8aa7aee0" --query 'Snapshots[*].SnapshotId'
[
    "snap-08813b9cea2d29135",
    "snap-0ef7348c2f887108c",
    "snap-049231844049afb33"
]
[ec2-user@ip-10-5-0-219 ~]$ python3.8 snapshotter_v2.py
Deleting snapshot snap-0ef7348c2f887108c
Deleting snapshot snap-08813b9cea2d29135
[ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=vol-0d307749e8aa7aee0" --query 'Snapshots[*].SnapshotId'
[
    "snap-03d2cef2922ab807a",
    "snap-049231844049afb33"
]
[ec2-user@ip-10-5-0-219 ~]$ 

```

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-3-124627/183-lab-JAWS-managing-storage/s3/files.zip>

```
snapshots = list(v.snapshots.all())
if len(snapshots) > MAX_SNAPSHOTS:

    # Delete oldest snapshots, but keep MAX_SNAPSHOTS available
    snap_sorted = sorted([(s.id, s.start_time, s) for s in snapshots], key=lambda k: k[1])
    for s in snap_sorted[: -MAX_SNAPSHOTS]:
        print("Deleting snapshot", s[0])
        s[2].delete()

ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=vol-0d307749e
a7aee0" --query 'Snapshots[*].SnapshotId'

    "snap-08813b9cea2d29135",
    "snap-0ef7348c2f887108c",
    "snap-049231844049afb33"

ec2-user@ip-10-5-0-219 ~]$ python3.8 snapshotter_v2.py
Deleting snapshot snap-0ef7348c2f887108c
Deleting snapshot snap-08813b9cea2d29135
ec2-user@ip-10-5-0-219 ~]$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=vol-0d307749e
a7aee0" --query 'Snapshots[*].SnapshotId'

    "snap-03d2cef2922ab807a",
    "snap-049231844049afb33"

ec2-user@ip-10-5-0-219 ~]$ wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-RS
JAWS-3-124627/183-lab-JAWS-managing-storage/s3/files.zip
--2025-09-15 13:38:33-- https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-RSJAWS-3-12
4627/183-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.c
om)... 3.5.77.229, 3.5.85.157, 52.92.160.194, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazona
ws.com)|3.5.77.229|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72110 (70K) [application/zip]
Saving to: 'files.zip'

0.00%[=====>] 72,110      --.-K/s   in 0.001s

2025-09-15 13:38:33 (60.9 MB/s) - 'files.zip' saved [72110/72110]

ec2-user@ip-10-5-0-219 ~]$
```

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:

`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`

```
"snap-049231844049afb33"
]
[ec2-user@ip-10-5-0-219 ~]$ 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
--2025-09-15 13:38:33-- 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)... 3.5.77.229, 3.5.85.157, 52.92.160.194, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)|3.5.77.229|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72110 (70K) [application/zip]
Saving to: 'files.zip'

100%[=====>] 72,110      --.-K/s   in 0.00s

2025-09-15 13:38:33 (60.9 MB/s) - 'files.zip' saved [72110/72110]

[ec2-user@ip-10-5-0-219 ~]$ 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
--2025-09-15 13:40:36-- 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)... 52.218.242.185, 52.92.204.138, 52.92.224.58, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)|52.218.242.185|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72110 (70K) [application/zip]
Saving to: 'files.zip.1'

100%[=====>] 72,110      --.-K/s   in 0.00s

2025-09-15 13:40:36 (53.8 MB/s) - 'files.zip.1' saved [72110/72110]

[ec2-user@ip-10-5-0-219 ~]$
```

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

`unzip files.zip`

```
[ec2-user@ip-10-5-0-219 ~]$ wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/WS-3-124627/183-lab-JAWS-managing-storage/s3/files.zip
--2025-09-15 13:40:36-- https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/WS-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)... 52.218.242.185, 52.92.204.138, 52.92.224.58, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)|52.218.242.185|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72110 (70K) [application/zip]
Saving to: 'files.zip.1'

100%[=====] 72.1K 53.8MB/s

2025-09-15 13:40:36 (53.8 MB/s) - 'files.zip.1' saved [72110]

[ec2-user@ip-10-5-0-219 ~]$ unzip files.zip
Archive: files.zip
  inflating: files/file1.txt
  inflating: files/file2.txt
  inflating: files/file3.txt
[ec2-user@ip-10-5-0-219 ~]$
```

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:

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

```
[ec2-user@ip-10-5-0-219 ~]$ aws s3api put-bucket-versioning --bucket s3-bucket-mariana --versioning-configuration Status=Enabled
[ec2-user@ip-10-5-0-219 ~]$
```

39. 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:

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

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

```
[ec2-user@ip-10-5-0-219 ~]$ aws s3api put-bucket-versioning --bucket s3-bucket-mariana --versioning-configuration Status=Enabled
[ec2-user@ip-10-5-0-219 ~]$
```

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

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

```
[ec2-user@ip-10-5-0-219 ~]$ aws s3 ls s3://s3-bucket-mariana/files/
2025-09-15 13:47:29      30318 file1.txt
2025-09-15 13:47:29      43784 file2.txt
2025-09-15 13:47:29      96675 file3.txt
[ec2-user@ip-10-5-0-219 ~]$
```

i-0e6990af9483112ce (Command Host)

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

```
rm files/file1.txt
```

42. 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:

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

The command returns a message similar to following:

```
delete: s3://examplebucketname/files/file1.txt
```

43. 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/
```

```
[ec2-user@ip-10-5-0-219 ~]$ rm files/file1.txt
rm: cannot remove 'files/file1.txt': No such file or directory
[ec2-user@ip-10-5-0-219 ~]$ aws s3 sync files s3://s3-bucket-mariana/files/ --delete
delete: s3://s3-bucket-mariana/files/file1.txt
[ec2-user@ip-10-5-0-219 ~]$ aws s3 ls s3://s3-bucket-mariana/files/
2025-09-15 13:47:29      43784 file2.txt
2025-09-15 13:47:29      96675 file3.txt
[ec2-user@ip-10-5-0-219 ~]$
```

44. 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
```

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.

```

2025-09-15 13:47:29      96675 file3.txt
[ec2-user@ip-10-5-0-219 ~]$ aws s3api list-object-versions --bucket s3-bucket-mariana --prefix file1.txt
{
  "Versions": [
    {
      "ETag": "\"b76b2b775023e60be16bc332496f8409\"",
      "ChecksumAlgorithm": [
        "CRC32"
      ],
      "ChecksumType": "FULL_OBJECT",
      "Size": 30318,
      "StorageClass": "STANDARD",
      "Key": "files/file1.txt",
      "VersionId": "BzQ7qPSSUKqGDlPzkEdm.BPav3k7MuE3",
      "IsLatest": false,
      "LastModified": "2025-09-15T13:47:29.000Z",
      "Owner": {
        "DisplayName": "awslabsc0w5878349t1686387562",
        "ID": "276ed0dbcd509d5d0deb9c4d0631da5cccf60235694483b3990f61bceb62f7"
      }
    },
    {
      "Owner": {
        "DisplayName": "awslabsc0w5878349t1686387562",
        "ID": "276ed0dbcd509d5d0deb9c4d0631da5cccf60235694483b3990f61bceb62f7"
      },
      "Key": "files/file1.txt",
      "VersionId": "DPMYztXIRZxCVJxr2ARlmm.8TJD5S4Vo",
      "IsLatest": true,
      "LastModified": "2025-09-15T13:53:09.000Z"
    }
  ],
  "DeleteMarkers": [
    {
      "Owner": {
        "DisplayName": "awslabsc0w5878349t1686387562",
        "ID": "276ed0dbcd509d5d0deb9c4d0631da5cccf60235694483b3990f61bceb62f7"
      },
      "Key": "files/file1.txt",
      "VersionId": "DPMYztXIRZxCVJxr2ARlmm.8TJD5S4Vo",
      "IsLatest": true,
      "LastModified": "2025-09-15T13:53:09.000Z"
    }
  ],
  "RequestCharged": null,
  "Prefix": "files/file1.txt"
}
[ec2-user@ip-10-5-0-219 ~]$

```

45. 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
```

46. To verify that the file was restored locally, run the following command:

```
ls files
```

The command shows all three files listed.

47. 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/
```

48. 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/
```

```
    "Key": "files/file1.txt",
    "VersionId": "BzQ7qPSsUKqGDlPzkEdm.BPav3k7MuE3",
    "IsLatest": false,
    "LastModified": "2025-09-15T13:47:29.000Z",
    "Owner": {
      "DisplayName": "awslabsc0w5878349t1686387562",
      "ID": "276ed0dbcd509d5d0deb9c4d0631da5cccffc60235694483b3990f61bceb62f7"
    }
  },
  "DeleteMarkers": [
    {
      "Owner": {
        "DisplayName": "awslabsc0w5878349t1686387562",
        "ID": "276ed0dbcd509d5d0deb9c4d0631da5cccffc60235694483b3990f61bceb62f7"
      },
      "Key": "files/file1.txt",
      "VersionId": "DPMYztXiRZxCVJxr2AR1mm.8TJD584Vo",
      "IsLatest": true,
      "LastModified": "2025-09-15T13:53:09.000Z"
    }
  ],
  "RequestCharged": null,
  "Prefix": "files/file1.txt"
}
[ec2-user@ip-10-5-0-219 ~]$ aws s3api get-object --bucket s3-bucket-mariana/files --key files/f
--version-id VERSION-ID files/file1.txt

Parameter validation failed:
Invalid bucket name "s3-bucket-mariana/files": Bucket name must match the regex "^[a-zA-Z0-9.\-
}$" or be an ARN matching the regex "^arn:(aws).*: (s3|s3-object-lambda):[a-z\~0-9]*:[0-9]{12}:a
nt[/:][a-zA-Z0-9\~.]{1,63}$|^arn:(aws).*:s3-outposts:[a-z\~0-9]+:[0-9]{12}:outpost[/:][a-zA-Z0-
3]{/[:]}accesspoint[/:][a-zA-Z0-9\~.]{1,63}$"
[ec2-user@ip-10-5-0-219 ~]$ ls files
file2.txt  file3.txt
[ec2-user@ip-10-5-0-219 ~]$ aws s3 sync files s3://s3-bucket-mariana/files/
[ec2-user@ip-10-5-0-219 ~]$ aws s3 ls s3://s3-bucket-mariana/files/
2025-09-15 13:47:29      43784 file2.txt
2025-09-15 13:47:29     966675 file3.txt
[ec2-user@ip-10-5-0-219 ~]$
```