Amazon EBS

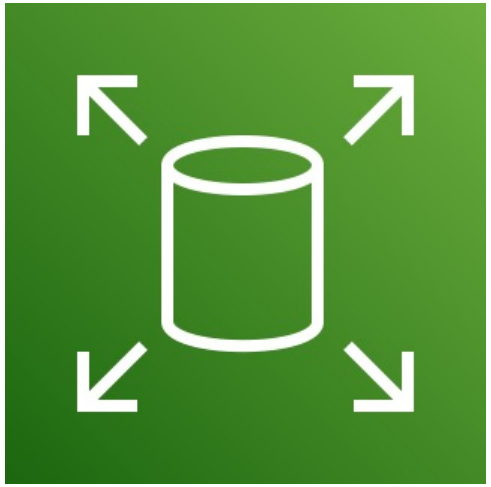
At the core of the lesson

You will learn how to do the following:

- Define Amazon Elastic Block Store (Amazon EBS).
- Identify EBS volume types and examine some use cases.
- Create and manage an EBS volume using the AWS Command Line Interface (AWS CLI).

Amazon EBS overview

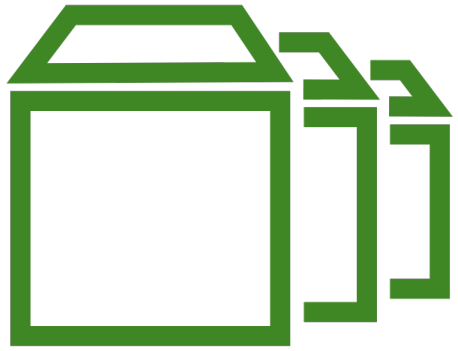
What is Amazon EBS?



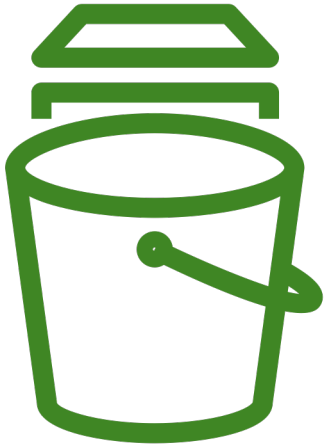
Amazon EBS

- Amazon EBS provides persistent block storage volumes.
- Each EBS volume is automatically replicated within its Availability Zone.
- With Amazon EBS, you can scale your usage up or down within minutes.

Amazon EBS features



Multiple volume types



Snapshots

Encryption

Amazon EBS use cases

EBS volume use cases:

- Boot volumes and primary storage for Amazon Elastic Compute Cloud (Amazon EC2) instances
- Data storage with a file system
- Database hosts

Snapshot use cases:

- Create a backup of critical workloads.
- Recreate EBS volumes.
- Share and copy data.

EBS volume types

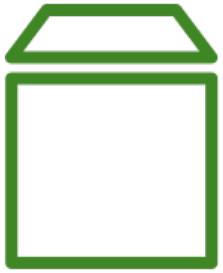
EBS volume types

Solid state drives (SSDs):

- Provisioned IOPS SSD
- General Purpose SSD volumes

Hard disk drives (HDDs):

- Throughput Optimized HDD
- Cold HDD



Volume

Volume type comparison

	Provisioned IOPS SSD (io2 Block Express)	Provisioned IOPS SSD (io2)	Provisioned IOPS SSD (io1)	EBS General Purpose SSD (gp3)	EBS General Purpose SSD (gp2)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Volume size	4 GB–64 TB	4 GB–16 TB	4 GB–16 TB	1 GB–16 TB	1 GB–16 TB	125 GB–16 TB	125 GB–16 TB
Max IOPS/volume	256,000	64,000	64,000	16,000	16,000	500	250
Max throughput/volume	4,000 MBps	1,000 MBps	1,000 MBps	1,000 MBps	250 MBps	500 MBps	250 MBps
Price calculated by	GB per month and Provisioned IOPS per month	GB per month and Provisioned IOPS per month	GB per month and Provisioned IOPS per month	GB per month and Provisioned IOPS per month	GB per month	GB per month	GB per month

Use cases for EBS volume types: SSD

Provisioned IOPS

- I/O-intensive workloads
- Relational databases
- NoSQL databases

General Purpose

- Recommended option for most workloads
- System boot volumes
- Virtual desktops
- Low-latency interactive apps
- Development and test environments

Use cases for EBS volume types: HDD

Throughput-optimized

- Streaming workloads that require consistent, fast throughput at a low price
- Big data
- Data warehouses
- Log processing
- Not a boot volume

Cold

- Throughput-oriented storage for large volumes of data that are infrequently accessed
- Scenarios where the lowest storage cost is important
- Not a boot volume

Demonstration

Use the AWS Management Console to do the following:

- Create an EBS volume.
- View volume information.
- Attach an EBS volume to an instance.

AWS CLI: Creating an EBS volume and attaching it to an EC2 instance

Creating an EBS volume

Command

```
$ aws ec2 create-volume \
    --size 80 \
    --availability-zone us-east-1a \
    --volume-type gp2
```

Result

```
{
    "AvailabilityZone": "us-east-1a",
    "Tags": [],
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "creating",
    "Iops": 240,
    "SnapshotId": "",
    "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "Size": 80
}
```

Describing a new volume

Command

```
$ aws ec2 describe-volumes \
--volume-ids vol-1234567890abcdef0
```

Result


```
{
    "Volumes”          : [
        {
            "Attachments”     : [],
            "AvailabilityZone” : "us-east-1a",
            "CreateTime”: "2023-04-04T21:35Z",
            "Encrypted”       : false,
            "Size”            : 80,
            "SnapshotId”      : "",
            "State”    : "available",
            "VolumeId” : "vol-1234567890abcdef0",
            "Iops”                    : 240,
            "VolumeType”              : "gp2"
        }
    ]
}
```

Attaching an EBS volume

Command

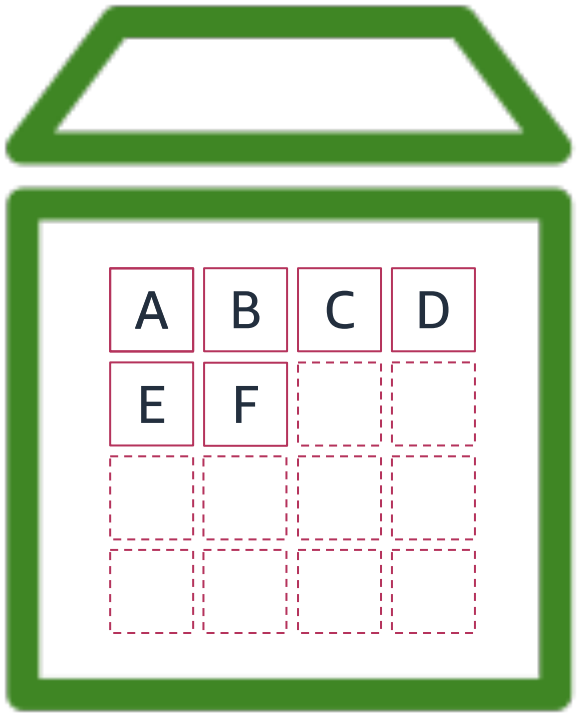
```
$ aws ec2 attach-volume \
--volume-id vol-1234567890abcdef0 \
--instance-id i-01474ef662b89480 \
--device /dev/sdf
```

Result

```
{
    "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "InstanceId": "i-01474ef662b89480",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "attaching",
    "Device": "/dev/sdf"
}
```

AWS CLI: Creating a snapshot of a volume

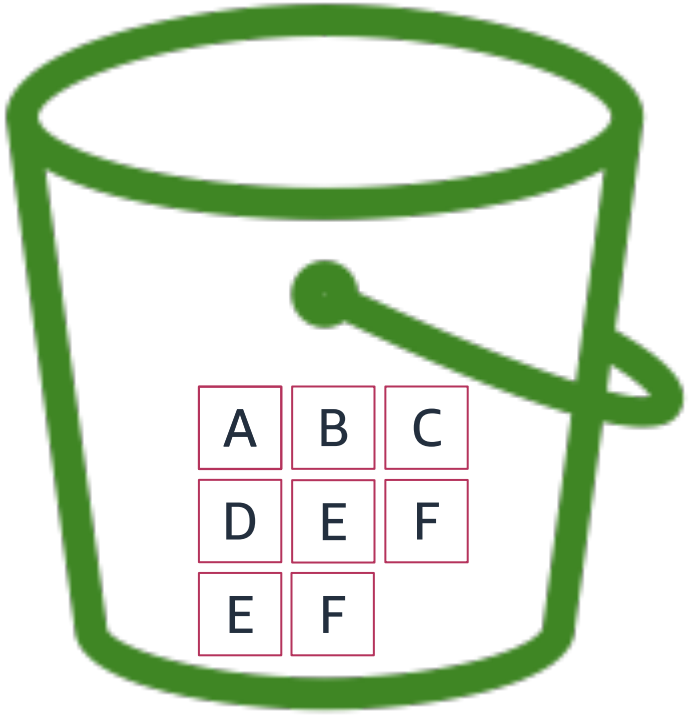
Amazon EBS incremental snapshots



EBS volume data

Snapshot

Snapshot 1 = ABC
Snapshot 2 = ABCDEF
Snapshot 3 = ABCDEF



EBS snapshot data in Amazon S3

Creating a snapshot

Command

```
$ aws ec2 create-snapshot \
--volume-id vol-1234567890abcdef0 \
--description "This is my root volume
snapshot"
```

Result

```
{
    "Description”: "This is my root volume
snapshot",
    "Encrypted”:false,
    "OwnerId:"012345678912",
    "Progress”:"",
    "SnapshotId”:"snap-0fa00f50184685abf",
    "StartTime”:"2023-04-04T21:45:07.000Z",
    "State”:"pending",
    "VolumeId:vol-0065e7a238fbfde9a",
    "VolumeSize”:80,
    "Tags”:[]
}
```

Copying a snapshot

Command

```
$ aws ec2 copy-snapshot \
--region us-east-1 \
--source-region us-west-2 \
--source-snapshot-id snap-
1234567890abcdef0
--description "This is my copied
snapshot"
```

Result

```
{
    "SnapshotId”:"snap-0b3c2a7c2a7e4eec6"
}
```

Proof of a copy

Command

```
$ aws ec2 describe-snapshots \
--snapshot-ids snap-0b3c2a7c2a7e4eec6
\
--region us-west-2
```

Result


```
{
    "Snapshots”:[
        {
            "Description”:"This is my copied
snapshot",
            "Encrypted”:false,
            "OwnerId”:"012345678912",
            "Progress” :"100%",
            "SnapshotId”:"snap-0b3c2a7c2a7e4eec6",
            "StartTime”:"2023-04-04T21:46:53.000Z",
            "State”:"completed",
            "VolumeId”:"vol-ffffffff",
            "VolumeSize”:80
        }
    ]
}
```

AWS CLI: Restoring a snapshot

Restoring a snapshot

Command

```
$ aws ec2 create-volume \
--size 80 \
--availability-zone us-east-1a \
--volume-type gp2 \
--snapshot-id snap-1234567890abcdef0
```

Result

```
{
    "AvailabilityZone”:"us-east-1a",
    "CreateTime”:"2023-04-04T21:51:40.000Z",
    "Encrypted”:false,
    "Size”:80,
    "SnapshotId”:"snap-1234567890abcdef0",
    "State”:"creating",
    "VolumeId”:"vol-0c108c43c627cb26d",
    "Iops”:240,
    "Tags”:[],
    "VolumeType”:"gp2"
}
```

Checking volume status

Command

```
$ aws ec2 describe-volume-status \
--volume-ids vol-0c108c43c627cb26d
```

Result


```
{
    "VolumeStatuses": [
                {
                "VolumeStatus": {
                 "Status": "ok",
                 "Details": [
                                        {

                      "Status": "passed",
                      "Name": "io-enabled"
                        },
                        {
                      "Status": "not-applicable",
                      "Name": "io-performance"
                         }
                    ]
                }
```

Managing EBS volumes with a lifecycle policy and the AWS CLI

What is Amazon Data Lifecycle Manager?

Amazon Data Lifecycle Manager

Amazon Data Lifecycle Manager does the following:

- Automates the creation, retention, and deletion of snapshots
- Uses tags to identify EBS volumes to backup
- Uses a lifecycle policy to define the desired backup and retention actions
- Requires an AWS Identity and Access Management (IAM) role to allow the management actions

Creating the IAM role

Command

```
$ aws dlm create-default-role
```

Result


```
{
    "RolePolicy": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:CreateSnapshot",
                    "ec2:CreateSnapshots",
                    "ec2:DeleteSnapshot",
                    "ec2:DescribeInstances",
                    "ec2:DescribeVolumes",
                    "ec2:DescribeSnapshots",

"ec2:EnableFastSnapshotRestores",

"ec2:DescribeFastSnapshotRestores",

"ec2:DisableFastSnapshotRestores",
```

Creating a policy

Command

```
$ aws dlm create-lifecycle-policy \
--description "My backup policy" \
--state ENABLED \
--execution-role-arn \
arn:aws:iam::197757065544:role/AWSData
LifecycleManagerDefaultRole \
--policy-details
file://policyDetails.json
```

Result

```
{"PolicyId": "policy-0b0ac162276313459"}
```

policyDetails.json

```
{
  "ResourceTypes”: [
     "VOLUME"
  ],
  "TargetTags“        : [
     {
        "Key”:"name",
        "Value”:"production"
     }
  ],
  "Schedules”:[
     {
        "Name”         : "DailySnapshots",
        "TagsToAdd”: [
           {
```

Viewing a policy

Command

```
aws dlm get-lifecycle-policy \
--policy-id policy-0b0ac162276313459
```

Result


```
{
    "Policy": {
        "PolicyId": "policy-0b0ac162276313459",
        "Description": "My backup policy",
        "State": "ENABLED",
        "ExecutionRoleArn":
"arn:aws:iam::012345678912:role/AWSDataLifecycleMan
agerDefaultRole",
        "DateCreated": "2019-04-04T22:45:11Z",
        "DateModified": "2019-04-04T22:45:12Z",
        "PolicyDetails": {
                "ResourceTypes": ["VOLUME"],
                "TargetTags": [
                {
                "Key": "name",
                "Value": "production"
                 }
                ],
                "Schedules": [
```

Checkpoint questions

1. A cloud engineer wants to choose a volume type for additional storage on a virtual desktop. Which volume type should the engineer use?
2. A cloud engineer has restored the EBS volume of an unresponsive instance from a recent snapshot. The engineer is concerned about the initial latency of the restored snapshot. What can the engineer do to prevent this latency?
3. Which IAM role do you need to create before you can use Amazon Data Lifecycle Manager?

Key ideas

- Amazon EBS provides block-level storage volumes for use with EC2 instances.
- EBS volumes are off-instance storage that persists independently from the life of an instance.
- You can choose from HDD and SSD storage types.
- Additional features include replication in the same Availability Zone, transparent encryption, elastic volumes, and backup through snapshots.

Thank you

Corrections, feedback, or other questions?
Contact us at https://support.aws.amazon.com/#/contacts/aws-training.
All trademarks are the property of their owners.