

Storage Refresher

Key Terms:

- **DIRECT (local) attached storage:** Storage on the EC2 Host
- **NETWORK attached storage:** Volumes are created and attached to a device over the network. In AWS, uses EBS (Elastic Block Store). Highly resilient, separate from instance hardware
- **EPHEMERAL storage:** Temporary storage, non-persistent. This is the Instance Store, physical storage attached to an EC2 Host
- **PERSISTENT storage:** Permanent storage that lives past the lifetime of an instance. EXAM: An example of Persistent Storage in AWS is the networked attached storage delivered by EBS. Know which types of storage are Ephemeral and Persistent

Categories of Storage:

- **BLOCK storage:** Create a Volume (Ex. inside EBS) that is presented to the OS as a collection of blocks; NO STRUCTURE beyond that. Mountable. Bootable.
- **FILE storage:** Presented as a file share/file server WITH STRUCTURE. Mountable. NOT bootable.
- **OBJECT Storage:** Collection of objects. NO STRUCTURE; flat. NOT mountable. NOT bootable.

Storage Performance

These items don't exist in isolation. IOPS is like speed of engine a racecar runs at (revolutions per second), IO block size is like the size of the car's wheels, throughput is the top-speed of the racecar.

- **IO BLOCK SIZE:** Size of the data you're writing to disk.
- **IOPS:** Measures the number of I/O operations a storage system supports in a second (how many read/writes in a second)
- **THROUGHPUT:** Amount of data that can be transferred in a given second, MB/s.
- $IO \times IOPS = \text{Throughput}$ ** Ex. 16KB IO block size x 100IOPS = 1.6MB/s throughput

EC2 - EBS Service Architecture

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices. You can mount these volumes as devices on your instances. EBS volumes that are attached to an instance are exposed as storage volumes that persist independently from the life of the instance. You can create a file system on top of these volumes, or use them in any way you would use a block device (such as a hard drive).

- EBS can be encrypted using KMS
- When you attach a volume to an EC2 instance, the instance sees a Block Device, so raw storage.
- EBS volumes appear just like any other storage device to an EC2 instance
- EBS Volumes are attached to ONE instance at a time. However, can be detached and re-attached to a new instance. They are not lifecycle linked to one instance, they're persistent; if an instance moves to a new EC2 host, the volume follows it. EBS is persistent until you delete the volume (separate from the instance)

- Snapshot into S3: You can create a backup of a volume with a Snapshot into S3. This can be used to migrate volumes between AZs.
- EBS can provision volumes based on different physical storage types: SSD, high-performance SSD, volumes based on mechanical disks, different size volumes, different performance profiles
- BILLING: Gigabyte/Month metric (and in some cases, performance)

EXAM: Storage is provisioned in ONE AVAILABILITY ZONE; Resilient in that AZ EXAM: Snapshots are now REGIONALLY RESILIENT

EBS Example - 1 region, 2 AZs (AZA, AZB), 1 S3 bucket.

EBS is AZ-based. So, with two AZs you have two separate EBSs. Same as EC2 hosts--per AZ

- EBS replicated data within an AZ. Failure of an entire AZ means failure of all volumes in that AZ. To fight this, you can Snapshot volumes into S3, creating more resilience, and you can create another volume in a new AZ from this snapshot (or even to a new region)
- For Global Resilience on volumes, copy snapshots over to different regions

EC2 - EBS Volume Types

General Purpose SSD — Provides a balance of price and performance. We recommend these volumes for most workloads.

- Volumes can vary in size from 1GB -> 16TB
- When created, a volume is given an I/O credit allocation

EBS Volume Types - General Purpose (GP):

1. GP2. GP2 is the default general purpose SSD based storage provided by EBS. CREDIT ARCHITECTURE
 - GP2 gets 3 IO credits per second, per GB of volume size
 - "Burst Rate": GP2 volume can burst up to 3,000 IOPS
 - Max IOPS for GP2 is currently 16,000 IO credits per second (baseline performance) - Any volumes > 5.33TB in size achieves this rate constantly
 - Great for: boot volumes, low latency interactive apps, dev/test environments
 - GP2 auto scales on size
2. GP3. SSD based, but removed the credit architecture GP2
 - Every GP3 volume, regardless of size starts with 3,000 IOPS and 125MiB/s
 - ~20% cheaper than GP2
 - Need more performance? Can pay for up to 16,000 IOPS or 1,000 MiB/s
 - GP3 does NOT auto-scale, you have to enable the upgrades

IO Credit. When you create a volume (size ranging from 1GB -> 16TB), it is created with an IO credit allocation.

- An IO is one input/output operation
- IO credit is a 16KB chunk of data

- An IOP is one chunk of 16KB in one second
- Ex. If transferring a 160KB file, that's 10 IO blocks of data. If this is all done in 1 second, that's 10IOPS.
(IO x IOPS = Throughput -> 10 x 1 = 10IOPS)
- 1 IOPS is 1 IO in 1 second
- IF you have NO credits in your IO "bucket" in the volume, you can't perform any IO on the disc
- IO Bucket has capacity of 5.4 million IO credits and "Fills at the rate of Baseline Performance"
- "Fills at the rate of Baseline Performance"... what does this mean? Every volume has baseline performance based on size, with a minimum -- IO Bucket bills with minimum of 100 IO Credits per second REGARDLESS OF VOLUME SIZE

EC2 - EBS Volume Types - Provisioned IOPS (io1, io2, io2 Block Express)

Provisioned IOPS SSD — Provides high performance for mission-critical, low-latency, or high-throughput workloads.

- Three types of provisioned IOPS SSD: io1, io2, io2 Block Express
- IOPS are configurable independent of volume size and designed for super high-performance situations
- Max 64,000 IOPS/volume
- io1/io2, 1,000MB/s throughput
- io2 Block Express: 256,000 IOPS, 4,000MB/s throughput
- Size: 4BG -> 16TB for io1/io2. io2 Block Express 4GB -> 64TB
- Size-to-performance ratio maximums: io1 50IOPS/GB, io2 500IOPS/GB, io2 Block Express 1000IOPS/GB
- Per Instance Performance restriction: the max performance between the EBS svc and an EC2 instance
- NUMBERS (all the specs for Provisioned IOPS): <https://imgur.com/a/zYQk7G1>

Exam:

- EXAM: Provisioned IOPS: Most consistent low latency and jitter.
- EXAM: Prov.IOPS for sub-millisecond latency, consistent latency, and higher performance.

EC2 - EBS Volume Types - HDD-Based*

- Slower, only for specific situations Types:
 1. Throughput Optimized HDD (st1) — A low-cost HDD designed for frequently accessed, throughput-intensive workloads. For sequentially accessed data like streaming -- Big data, data warehouses, log processing
 2. Cold HDD (sc1) — The lowest-cost HDD design for less frequently accessed workloads. -- geared for max economy (lowest EBS cost option) with lowest performance

HDD Based EBS Volume Type METRICS: <https://imgur.com/a/EcGVrx8>

EC2 - Instance Store Volumes - Architecture

An instance store provides TEMPORARY block-level storage devices for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for TEMPORARY

storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store as well as the number of devices available varies by instance type.

The virtual devices for instance store volumes are `ephemeral[0-23]`. Instance types that support one instance store volume have `ephemeral0`. Instance types that support two instance store volumes have `ephemeral0` and `ephemeral1`, and so on.

- ISV = HIGHEST STORAGE PERFORMANCE in AWS
- Physically connected to ONE EC2 host - isolated to host
- ☒ is lost and instance is re-attached to new ephemeral[X]
- Instance Store Volumes should NOT be used for anything that requires persistence
- More IOPS and Throughput VS EBS

Exam

- EXAM: Local to an EC2 Host
- EXAM: Data stored on Instance Store Volumes is lost on instance MOVE, RESIZE, or HARDWARE FAILURE
- EXAM: HIGHEST STORAGE PERFORMANCE storage option in AWS, but can't provide persistence. TEMPORARY, for replaceable data
- EXAM: Instance Store Volumes have to be ATTACHED AT LAUNCH OF INSTANCE, cannot happen after. Instances on this host only can access them

Choosing between the EC2 Instance Store Volumes (ISV) and EBS

Resource: <https://aws.amazon.com/ec2/instance-types/>

Need...

- Persistence? EBS (avoid ISV)
- Resilience? EBS (avoid ISV) Isolated from Instance Life Cycles? (to be able to un-attach and re-attach) EBS (avoid ISV)
- Resilience w/App In-Built Replication... it depends
- High Performance? Depends (ISV is super high performance) Cost is main factor? ISV (it's often included in the price)

REMEMBER ALL OF THIS:

- EXAM: Cost Efficacy? ISV ST1 or ISV SC1. For throughput + streaming? ST1
- EXAM: Boot EC2 instances? NOT st1 or sc1
- EXAM: GP2 / GP3: Has a max performance of 16,000 IOPS
- EXAM: IO1 / IO2: Has max 64,000 IOPS (256,000 IOPS in io2 Block Express) - Max EBS performance with large instance (that is capable of providing this performance)
- EXAM: RAID0 + EBS: Has max 260,000 IOPS
- EXAM: Need anything greater than 260,000 IOPS? use ISV, not EBS. *No persistence

- EXAM: EBS Automated Backups? Amazon Data Lifecycle Manager: To automate the creation, retention, and deletion of Amazon EBS snapshots

TL;DR EBS Volume VS Instance Store

- EBS volume is network attached drive which results in slow performance but data is persistent meaning even if you reboot the instance data will be there.
- Instance store instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. -- Data on an instance store volume persists only during the life of the associated instance; if an instance is stopped or terminated, any data on instance store volumes is lost.

Recap

- SSD:
 - General Purpose (gp2, gp3)
 - Provisioned IOPS SSD volumes (io1, io2, io2 Block Express)
- HDD: st1, sc1

EC2 - EBS Snapshots, Restore & Fast Snapshot Restore (FSR)

- EBS Snapshots are backups of data consumed within EBS Volumes - Stored on S3.
- Snapshots are incremental, the FIRST COPY being a full backup - and any future snapshots being incremental. No massive risk of losing incremental backups (each increment is self-sufficient)
- Snapshots can be used to migrate data to different availability zones in a region, or to different regions of AWS.
- AZ Resilient (vulnerable to issues which impact entire AZ), but with a snapshot the data becomes Region Resilient
- Good for EBS Volume cloning, even moving / copying them across AZs
- BILLING:
 - Gigabyte-month cost; but you're only paying for USED data, not allocated data

Exam

- EXAM: Snapshot doesn't need to be initialized in a new EBS Volume, it's built-in
- EXAM: Snapshots are restored LAZILY; fetched gradually
- EXAM: Requested blocks are fetched immediately, you can do this initially to force rapid initialization so users have quickest access up front
- EXAM: Fast Snapshot Restore (FSR): A setting to Enable that allows immediate restores instead of gradual
- EXAM: Up to 50 FSRs per region. Set on the Snap & AZ, so 50 between AZs
- EXAM: FSR costs extra

DEMO - EC2 - EBS Volumes

Commands for Terminal: https://learn-cantrill-labs.s3.amazonaws.com/awscoursedemos/0004-aws-associate-ec2-ebs-demo/lesson_commands.txt

1. Create an EBS Volume: 1-Click Deploy link > Create Stack > EC2 EBS > Create Volume > select GP2, 10 GiB size, AZ us-east-1a, tag: key "Name" value "EBSTestVolume" > Create

2. Mount it to an EC2 instance: Select new EBS Volume > Actions: Attach Volume > Choose Instance1 > Attach
 3. Create and Mount a file system: Ec2 Instance > select A4L-EBS-INSTANCE1-AZB, Connect > EC2 Instance Connect > `sudo mkfs -t xfs /dev/xvdf`, Enter
 4. Generate a test file: EC2 Connect > `sudo mkdir /ebstest`, Enter > Mount dev/xvdf to /ebstest directory `sudo mount /dev/xvdf /ebstest` > `cd /esbstest` > `sudo nano amazingtestfile.txt` > write text, ctrl+o, enter, ctrl+x > Reboot instance `sudo reboot`
 5. Migrate volume with Snapshot: EC2 > Instances > Stop AZA Instance 1 and 2 > when stopped, Volumes: Detach Volume > right click EBS Volume, Create Snapshot > Right click Snapshot, Create Volume from Snapshot > Zone us-east-1b > Tag key "Name" value "EBSTestVolume-AZB" > Create
 6. Attach Volume to AZB instance: Volume > right click, Attach > ABZ Instance 1 > ATtach
 7. Verify the filesystem and file are intact > EC2 Instance Connet on AZB Instance 1 > `lsblk` > `sudo mkdir /ebstest` > `sudo mount /dev/xvdf /ebstest`
 8. Stop AZB Instance > Detach EBS Volume
 9. Clean up > Detele Snapshopt > Volumes: delete test volumes > CloudFormation: Delete Stack
- The remaining DEMO is not FREE TIER
 - On Instance Store Volumes, Instance reboot does not cause the data to be lost or moving to new host, but a Stop/Start changes hosts and you lose data

EC2 - EBS Encryption

EBS Encryption provides AT REST encryption for volumes and snapshots.

- Uses KMS: AWS/EBS Keys or Customer Managed Keys
- Uses DEK

Exam

- EXAM: AWS Accounts can be set to encrypt by default - Default KMS Key
- EXAM: Each Volume uses ONE unique DEK. Snapshots and future volumes use the SAME DEK. Volume from scratch? New DEK
- EXAM: There is NO WAY To remove an encryption from a volume, unless you clone data to a new volume
- EXAM: O/S is not aware of the encryption (no performance loss)