AWS - EBS

- An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance.
- You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application.
- You can also use them for throughput-intensive applications that perform continuous disk scans.
- EBS volumes persist independently from the running life of an EC2 instance.

- **Here is a list of important information about EBS Volumes:**
    - When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to a failure of any single hardware component.
    - An EBS volume can only be attached to one EC2 instance at a time.
    - After you create a volume, you can attach it to any EC2 instance in the same Availability Zone
    - An EBS volume is off-instance storage that can persist independently from the life of an instance. You can specify not to terminate the EBS volume when you terminate the EC2 instance during instance creation.
    - EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.
    - Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256)
    - EBS Volumes offer 99.999% SLA.

EBS attaching to instance

- EC2 section - Create Volume
- Attach to existing instance
- Attach during EC2 instance creation

EBS

https://aws.amazon.com/ebs/details/

EBS - Cheat sheet

https://tutorialsdojo.com/aws-cheat-sheet-amazon-ebs/

EBS Volume Types

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html

References:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumes.html

https://aws.amazon.com/ebs/features/

Check out this Amazon EBS Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-ebs/

Here is a short video tutorial on EBS:

https://youtu.be/ljYH5lHQdxo

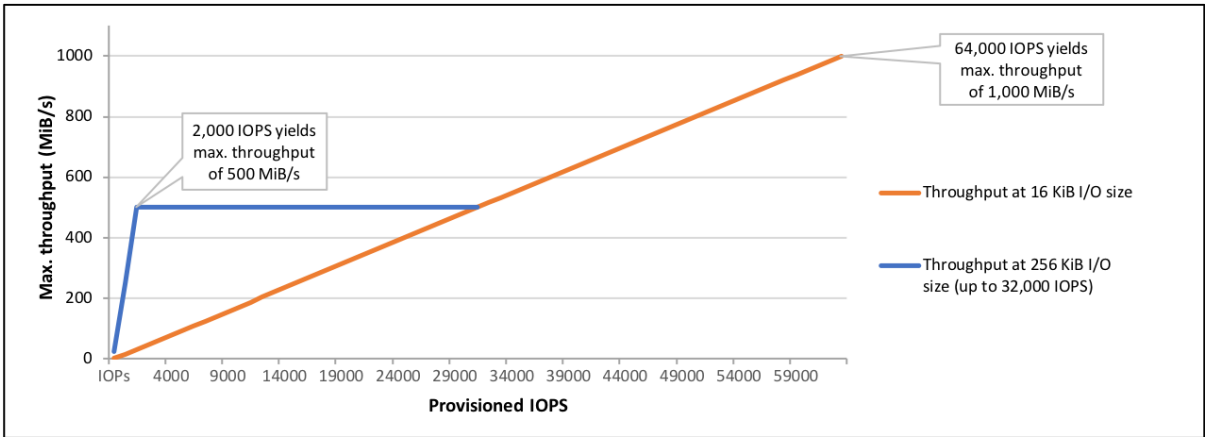
	Solid-state drives (SSD)		Hard disk drives (HDD)	
Volume type	General Purpose SSD (gp2)	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use cases	• Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments	• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: ◦ MongoDB ◦ Cassandra ◦ Microsoft SQL Server ◦ MySQL ◦ PostgreSQL ◦ Oracle	• Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume	• Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume

	Solid-state drives (SSD)		Hard disk drives (HDD)	
		◦ Oracle		
API name	gp2	io1	st1	sc1
Volume size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max IOPS per volume	16,000 (16 KiB I/O) *	64,000 (16 KiB I/O) †	500 (1 MiB I/O)	250 (1 MiB I/O)
Max throughput per volume	250 MiB/s *	1,000 MiB/s †	500 MiB/s	250 MiB/s
Max IOPS per instance ††	80,000	80,000	80,000	80,000
Max throughput per instance ††	2,375 MB/s	2,375 MB/s	2,375 MB/s	2,375 MB/s
Dominant performance attribute	IOPS	IOPS	MiB/s	MiB/s

Provisioned IOPS SSD (io1) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike gp2, which uses a bucket and credit model to calculate performance, an io1 volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

An io1 volume can range in size from 4 GiB to 16 TiB. You can provision from 100 IOPS up to 64,000 IOPS per volume on Nitro system instance families and up to 32,000 on other instance families. The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1.

For example, a 100 GiB volume can be provisioned with up to 5,000 IOPS. On a supported instance type, any volume 1,280 GiB in size or greater allows provisioning up to the 64,000 IOPS maximum (50 × 1,280 GiB = 64,000).



An io1 volume provisioned with up to 32,000 IOPS supports a maximum I/O size of 256 KiB and yields as much as 500 MiB/s of throughput. With the I/O size at the maximum, peak throughput is reached at 2,000 IOPS. A volume provisioned with more than 32,000 IOPS (up

to the cap of 64,000 IOPS) supports a maximum I/O size of 16 KiB and yields as much as 1,000 MiB/s of throughput.

The volume queue length is the number of pending I/O requests for a device. Latency is the true end-to-end client time of an I/O operation, in other words, the time elapsed between sending an I/O to EBS and receiving an acknowledgement from EBS that the I/O read or write is complete. Queue length must be correctly calibrated with I/O size and latency to avoid creating bottlenecks either on the guest operating system or on the network link to EBS.

Optimal queue length varies for each workload, depending on your particular application's sensitivity to IOPS and latency. If your workload is not delivering enough I/O requests to fully use the performance available to your EBS volume then your volume might not deliver the IOPS or throughput that you have provisioned.

Transaction-intensive applications are sensitive to increased I/O latency and are well-suited for SSD-backed io1 and gp2 volumes. You can maintain high IOPS while keeping latency down by maintaining a low queue length and a high number of IOPS available to the volume. Consistently driving more IOPS to a volume than it has available can cause increased I/O latency.

AWS - EBC RAID configuration

With **Amazon EBS**, you can use any of the standard RAID configurations that you can use with a traditional bare metal server, as long as that particular RAID configuration is supported by the operating system for your instance. This is because all RAID is accomplished at the software level.

For greater I/O performance than you can achieve with a single volume, RAID 0 can stripe multiple volumes together;

for on-instance redundancy, RAID 1 can mirror two volumes together.

Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. This replication makes Amazon EBS volumes ten times more reliable than typical commodity disk drives.

Configuration	Use	Advantages	Disadvantages
RAID 0	When I/O performance is more important than fault tolerance; for example, as in a heavily used database (where data replication is already set up separately).	I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput.	Performance of the stripe is limited to the worst performing volume in the set. Loss of a single volume results in a complete data loss for the array.
RAID 1	When fault tolerance is more important than I/O performance; for example, as in a critical application.	Safer from the standpoint of data durability.	Does not provide a write performance improvement; requires more Amazon EC2 to Amazon EBS bandwidth than non-RAID configurations because the data is written to multiple volumes simultaneously.

RAID 5 and RAID 6 are not recommended for Amazon EBS because the parity write operations of these RAID modes consume some of the IOPS available to your volumes. Depending on the configuration of your RAID array, these RAID modes provide 20-30% fewer usable IOPS than a RAID 0 configuration. Increased cost is a factor with these RAID modes as well; when using identical volume sizes and speeds, a 2-volume RAID 0 array can outperform a 4-volume RAID 6 array that costs twice as much.

Reference:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/raid-config.html

Check out this Amazon EC2 Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-elastic-compute-cloud-amazon-ec2/

AWS - EBS Volume Checks

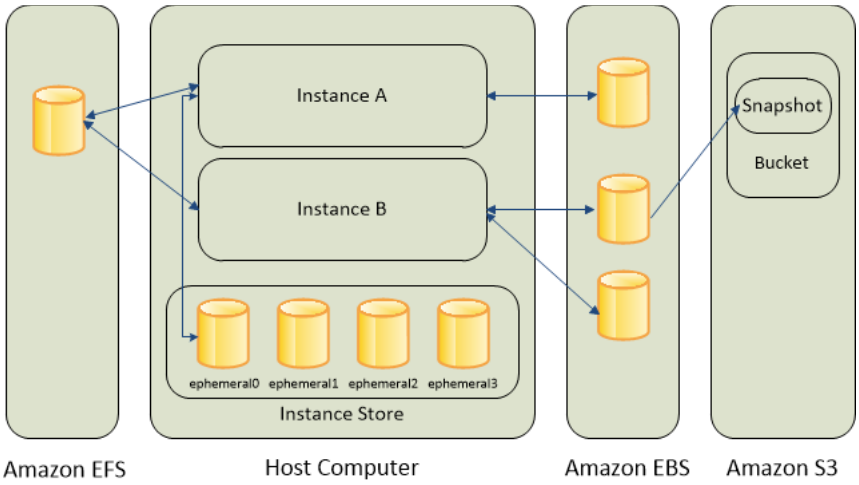
- Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.
  - OK - If all checks pass, the status of the volume is **ok**.
  - IMPAIRED - If a check fails, the status of the volume is **impaired**.
  - WARNING - If the volume is severely degraded or the volume performance is well below expectations, then the status is **warning**,
  - INSUFFICIENT-DATA - If the status is **insufficient-data**, the checks may still be in progress on the volume.

AWS - EBS Encryption

- Amazon EBS encryption offers seamless encryption of EBS data volumes, boot volumes, and snapshots, eliminating the need to build and maintain a secure key management infrastructure.
- EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys, or keys you create and manage using the AWS Key Management Service (KMS).
- The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage.

- **Amazon Elastic Block Store (Amazon EBS)** provides block level storage volumes for use with EC2 instances.

- EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same
  Availability Zone.
- EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance.



When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume

- All data moving between the volume and the instance

- All snapshots created from the volume

- All volumes created from those snapshots

Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

You can encrypt both the boot and data volumes of an EC2 instance.

References:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html

Check out this Amazon EBS Cheat Sheet:

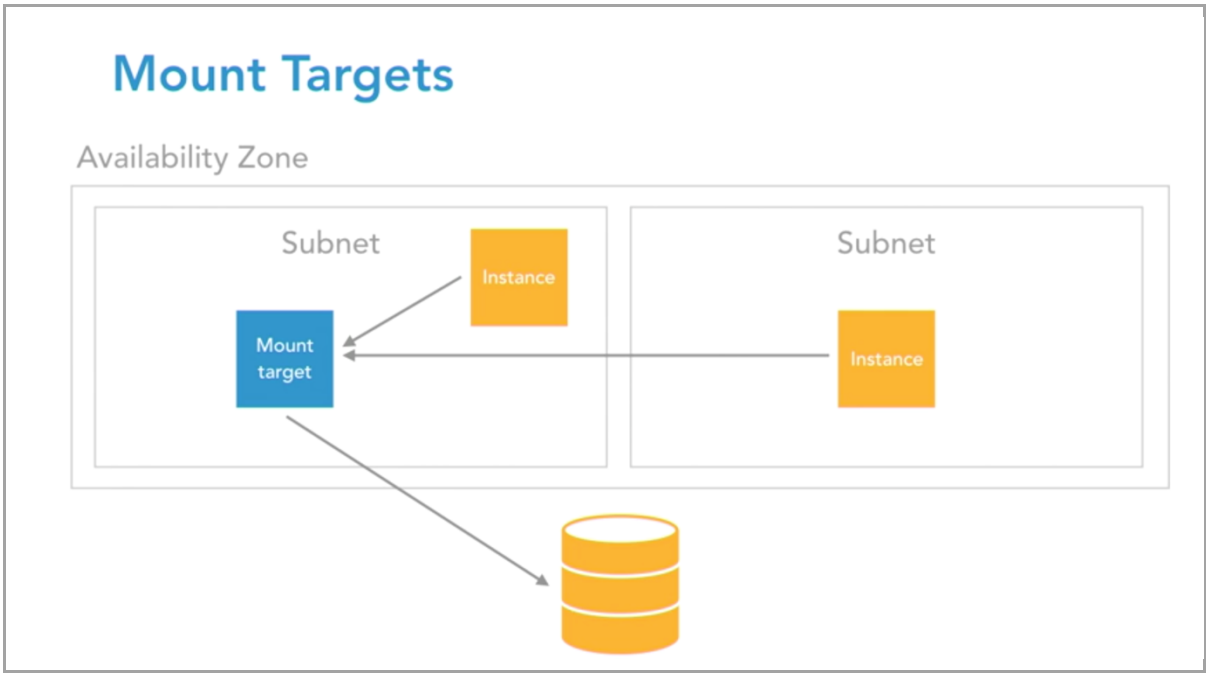
https://tutorialsdojo.com/aws-cheat-sheet-amazon-ebs/

AWS - EFS

Elastic File System (EFS) is like NAS in the cloud and EC2 instances that do not run Windows can access it using NFSv4.

- Create NFS from Terraform/AWS Console
- On EC2 Instance,
    - Install nfs-utils
    - Mount efs directory

/vnkt-efs Sudo mount -t nfs4 -o nfsvers=4.1,resize=1048576,wsize=1048576,hard,timeout=600,retrans=2 <nfs-url-generated-after-creation>: /vnkt-efs
Create a directory




```
https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/
8 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-0-116 ~]$ sudo yum install -y nfs-utils
Loaded plugins: priorities, update-motd, upgrade-helper
Package 1:nfs-utils-1.3.0-0.21.amzn1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-0-116 ~]$ sudo mkdir /efs
[ec2-user@ip-172-31-0-116 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo
=600,retrans=2 fs-5d50fdf4.efs.us-west-2.amazonaws.com: /efs
[ec2-user@ip-172-31-0-116 ~]$ df -h
Filesystem                                Size  Used Avail Use% Mounted on
devtmpfs                                  237M   60K  236M   1% /dev
tmpfs                                     245M     0  245M   0% /dev/shm
/dev/xvda1                                7.8G  980M  6.7G  13% /
fs-5d50fdf4.efs.us-west-2.amazonaws.com:/  8.0E     0  8.0E   0% /efs
[ec2-user@ip-172-31-0-116 ~]$
```

Review and create

Review the configuration below before proceeding to create your file system.

File system access

VPC	Availability Zone	Subnet	IP address	Security groups
vpc-24ac974c (default)	us-east-2a	subnet-9a9ebdf2 (default)	Automatic	sg-cebcaaa3 - default
	us-east-2b	subnet-3ec55d44 (default)	Automatic	sg-cebcaaa3 - default
	us-east-2c	subnet-a329e9ef (default)	Automatic	sg-cebcaaa3 - default

Optional settings

Tags Name: Production
Performance mode General Purpose
Throughput mode Bursting
Encrypted No

Cancel Previous Create File System

To mount your file system:

1. Open an SSH client and connect to your on-premises server.
2. Create a new directory on your on-premises server, such as "efs", for example:
   - `sudo mkdir efs`
3. Mount your file system using one of the mount target IP addresses that AWS Direct Connect or an AWS VPN connection can access. Learn more
   - `sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 MOUNT_TARGET_IP:/efs`

For more details, including about encryption of data in transit, see how to use the EFS mount helper from an on-premises server in the EFS documentation.

Configure file system access

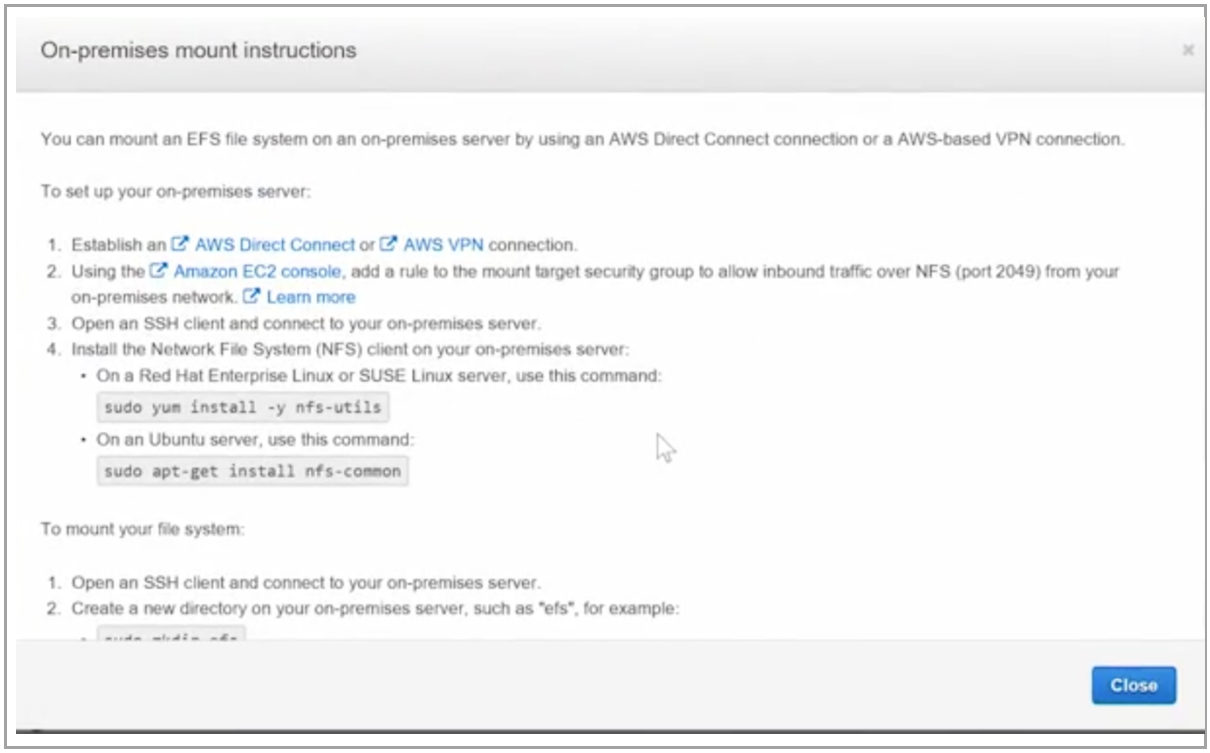
An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system by using a network interface called a mount target. Each mount targ has an IP address, which we assign automatically or you can specify.

VPC vpc-24ac974c (default)

Create mount targets

Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

	Availability Zone	Subnet	IP address	Security groups
☑	us-east-2a	subnet-9a9ebdf2 (default)	Automatic	sg-cebcaaa3 - default
☑	us-east-2b	subnet-3ec55d44 (default)	Automatic	sg-cebcaaa3 - default
☑	us-east-2c	subnet-a329e9ef (default)	Automatic	sg-cebcaaa3 - default


On-premises mount instructions
You can mount an EFS file system on an on-premises server by using an AWS Direct Connect connection or a AWS-based VPN connection.
To set up your on-premises server:
1. Establish an AWS Direct Connect or AWS VPN connection.
2. Using the Amazon EC2 console, add a rule to the mount target security group to allow inbound traffic over NFS (port 2049) from your on-premises network. Learn more
3. Open an SSH client and connect to your on-premises server.
4. Install the Network File System (NFS) client on your on-premises server:
• On a Red Hat Enterprise Linux or SUSE Linux server, use this command:
sudo yum install -y nfs-utils
• On an Ubuntu server, use this command:
sudo apt-get install nfs-common
To mount your file system:
1. Open an SSH client and connect to your on-premises server.
2. Create a new directory on your on-premises server, such as "efs", for example:
Close