

General File System Notes/Highlights

- Amazon Data Lifecycle Manager to create, and manage (retention and deletion) of EBS snapshots and EBS-backed AMIs.
 - Uses tags to work
- Can dynamically increase size of “EBS Elastic Volumes” without restarting the instance or detaching the volume, modify the performance, and change volume type
- Snapshots require initialization to reach full performance (can use `dd` to read all blocks). Exception is fast snapshot restore (FSR) which is fully initialized upon creation (up to 64K IOPS or 1 GB/s throughput, which should be initialized).
 - Initialization will consume volume credits
- When using encryption, AWS keeps the previous key version so it can decrypt data which was encrypted with the specific key version
- Recommend to always use CMK. Amazon managed key policy cannot be updated, hence any volume using this encryption cannot be shared across accounts.

Deciding which storage service

- Instance Store:
 - lowest latency, best performance
 - based on the instance type (so not all instance types)
 - lifecycle attached to the instance
 - storage limited
- Block storage: EBS
 - regular block based storage
 - AZ based
 - Low latency
 - Many types of EBS
 - single client
 - EBS Multi-Attach enables attachment of a single provisioned IOPS SSD (io1 or io2) to multiple instances in same AZ
 - Clustered server use cases
 - [Attach a volume to multiple instances with Amazon EBS Multi-Attach](#)

- Underpins many other AWS Services (EFS, FSx)
- Costing based on allocated storage
- lifecycle not tied to instance, unless option selected (“terminate with instance”)
- Object storage: S3
 - Region based (multiple AZ, based on tier)
 - Eventual read consistency? Not anymore! See <https://aws.amazon.com/blogs/aws/amazon-s3-update-strong-read-after-write-consistency/>
 - Shared storage
 - Usage based
 - Security based on resource policy, Access Control List (ACLs), Access Points, and IAM
 - Lifecycle management (tiered storage layers, Glacier, etc.)
 - Glacier is special callout, for archival storage
 - Many clients
 - Highly durable and available
 - S3 Storage Lens provides reporting capability on usage, tiering.
 - Can be mounted on instance, but no official support, and is slow. Only use for specific cases.
 - s3fs-fuse
 - Underpins many AWS services, such as EMR, Datalake, etc.
- Shared (concurrent) File Storage (network attached storage):
 - Multiple AZ (based on storage service type)
 - Multiple clients
 - EFS (Linux) uses NFS protocol
 - Usage size
 - Concurrent/Shared server access (performance improves with concurrent access). Lambda can also mount!
 - Security based on IAM, security group
 - Tiered storage via EFS Infrequent Access (reduced storage or cost savings at expense of availability and performance)
 - Burst mode, with Max IO mode available (cannot switch once decision made)
 - High performance compute storage:
 - FSx (Windows) uses NFS, SMB protocols based on chosen type
 - Security based on IAM, security group

- Available for NetApp, OpenZFS, Windows File Server, Lustre (HPC use cases)

Using a new volume (Linux)

```
sudo su
# List available blocks devices
lsblk

# Check for file system on disk
file -s /dev/xvdf

# Make/create file system, of type ext4 (other options)
mkfs -t ext4 /dev/xvdf

# Mkdir directory to use for volume point
mkdir /appdata

# Mount volume at directory, and confirm file system
mount /dev/xvdf /appdata
lsblk

# Create sample file in new volume
echo "This is a data file getting created in the volume. Obviously, this is NC
ls -la /appdata
```

Extending a Linux file system

Certain criteria exist for extending volumes, please see [# Requirements when modifying volumes](#)

Procedure differs (slightly) based on type of hypervisor, to check hypervisor:

```
# Check hypervisor type of your instance type
aws ec2 describe-instance-types --instance-type <instance_type> --query "Insta
```

To following are for Xen which is the lab default using **t2.micro**, using an **ext4** file system, and no created partition.

```
# Step 1: List the block devices (and get the partition number)
lsblk
```

```
# Step 2: Confirm the file system type and current size
sudo df -hT
```

```
# Step 3: Unmount the file system
sudo umount /appdata
```

```
# Step 4: Check the file system, and grow the file system
sudo e2fschk -f /dev/xvdf
sudo resize2fs /dev/xvdf
```

```
# Step 3: mount the file system
sudo mount /dev/xdx /appdata
```

```
# Step 2: Confirm the file system size update
sudo df -hT
```

Useful Links:

- [Extend a Linux file system after resizing a volume](#)
- [Choosing an Amazon FSx File System](#)
- [Storage Architecture Selection](#)
- [How incremental snapshots work](#)
- [AWS Storage Gateway](#)
- [How do you select your storage solution?](#)
- [Choosing the Right AWS Cloud Storage for Your Data](#)