**AMI OVERVIEW**

* **AMI:** Amazon Machine Image
* **AMIs** are the customization of EC2 Instances , which means we can add our own software, configuration, OS, monitoring etc. Also, if we create our own AMIs, we get faster boot time, configuration time, because our software is pre-packaged.
* **AMIs** are built for a specific region and can be copied across regions.
* You can launch EC2 instance from:

1. A Public AMI: AWS provided
2. Your own AMI: you make and maintain them yourself
3. An AWS marketplace AMI: An AMI made by someone else and is probably sold.

**AMI PROCESS (from and EC2 Instance)**

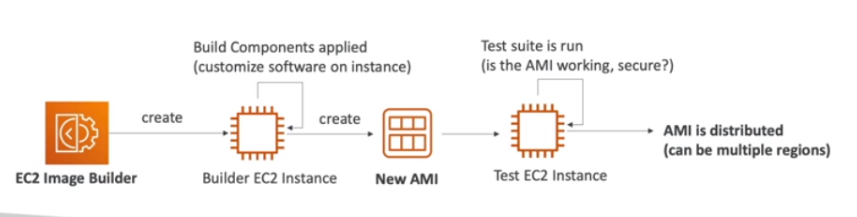
* Start an EC2 instance and customise it
* Stop the instance (for data integrity)
* Build an AMI – this also creates EBS snapshots
* We can launch instances from other AMIs.



**CREATING AN AMI FROM AN EC2 INSTANCE:**

1. Launch an EC2 Instance and provide a user data script (say, the below script to install Apache webserver)  
     
     
     
     
   #/bin/bash  
   # Use this for your user data script  
   # Install httpd (Linux 2 version)  
   yum update -y  
   yum install -y httpd  
   systemctl start httpd  
   systemctl enable httpd
2. Now wait for some time for the instance to be launched. We can now browse the public IP of the instance to see the Apache web server home page.
3. To create an AMI from this instance, right click on the instance -> *Image and Template* -> *Create Image*
4. Give the AMI basic details, such as Image name etc and click on *Create Image*
5. We can see these details under *Images* -> *AMIs.*
6. Once the AMI is up, we can launch instances from this AMI. We can also launch the instances by navigating through *Instances -> Launch Instances ->* Select *My AMIs.*
7. Give the below user Data script:  
   #/bin/bash  
   # Use this for your user data script  
   # Install httpd (Linux 2 version)  
   echo “<h1>Hello World from $(hostname -f)</h1>”> /var/www/html/index.html
8. Here, we don’t need to again install the Apache web server (httpd), since the AMI from which we are launching this instance already contains Apache Web Server. This will save our boot time and it solves the whole purpose of AMIs.

**EC2 IMAGE BUILDER**

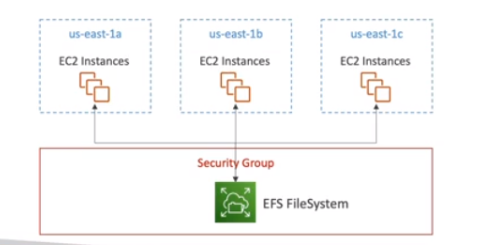
* It is used to automate the creation of VMs or container images
* Automate the creation, maintain, validate and test EC2 AMIs.   
  

**EC2 Instance Store**

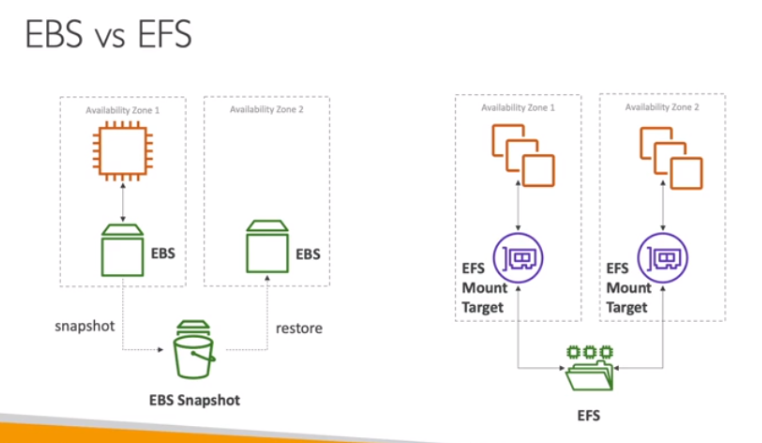
* EBS Volumes are network drives with good but “limited” performance
* If you need a high-performing hardware disk, use EC2 Instance Store
* It has a better I/O performance
* EC2 Instance Store lose their storage if they’re stopped (They last only for a short period of time and can’t be used as durable long-term volumes)
* EC2 Instance Store is good for buffer/cache/scratch data/temporary content
* Risk of data loss if hardware fails.

**EFS – ELASTIC FILE SYSTEM**

* Managed Network File System (NFS) that can be mounted on 100s of EC2.
* EFS works only with Linux EC2 instances in multi-AZ.
* It is highly available, scalable, expensive (3x gp2), pay per use, no capacity planning

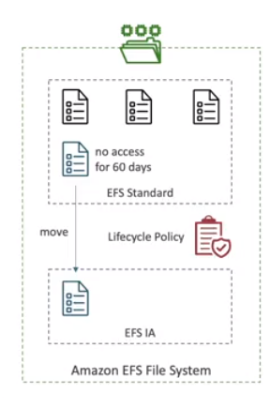


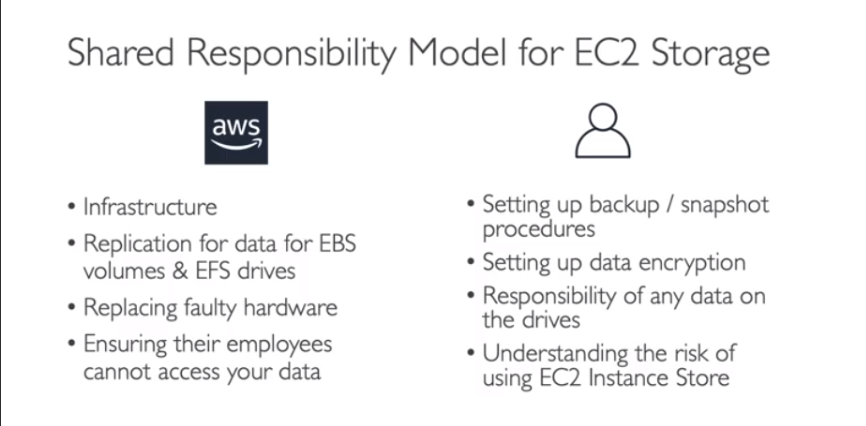
**EBS vs EFS**



**EFS INFREQUENT ACCESS (EFS-IA)**

* Storage class that is cost-optimized for files not accessed everyday
* Up to 92% lower cost compared to EFS standard
* EFS will automatically move your files to EFS-IA based on the last time they were accessed.
* Enable EFS-IA with a Lifecycle policy
* Ex: move files that are not accessed for 60 days to EFS-IA
* Transparent to the applications accessing EFS





**AMAZON FSx: OVERVIEW**

* Launch 3rd party high-performance file systems on AWS
* Fully managed service



1. **Amazon FSx for Windows File Server**

* A fully managed, highly reliable, and scalable Windows native shared file system
* Built on Windows File Server
* Supports SMB protocol and Windows NTFS
* Integrated with Microsoft Active Directory
* Can be accessed from AWS or your on-prem infrastructure

1. **Amazon FSx for Lustre**

* A fully managed, high-performance, scalable file storage for High Performance Computing (HPC)
* Lustre = Linux + Cluster
* ML, Analytics, Video processing, Financial Modelling.
* Scales up to 100 GBps, millions of IOPS, sub-ms latencies

