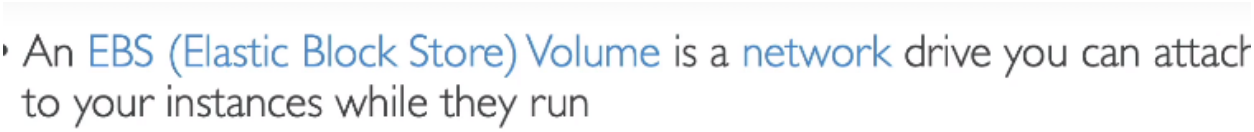
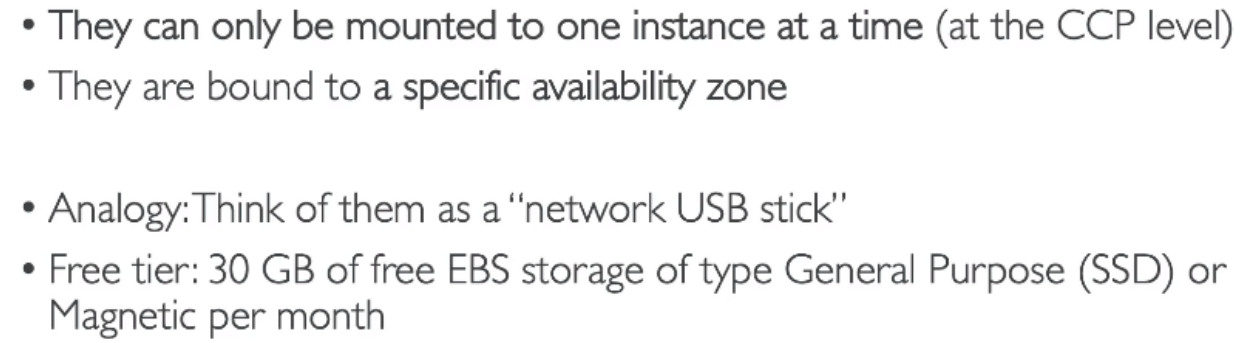
EBS Overview:

1. What is EBS Volume?



2.
3. Even if we terminate our instance ..our data will still exists in EBS

4.
5. So this EBS Volumes allow us to persist data, even after the instance is terminated.
6. And so that's the whole purpose, we can recreate an instance and mount to the same EBS Volume from before and we'll get back our data. That is very helpful.
7. We can think EBS volumes as network USB sticks So, it's a USB stick that you can take from a computer
8. and put it in another computer but you actually don't physically put it in a computer.It's attached through the network.



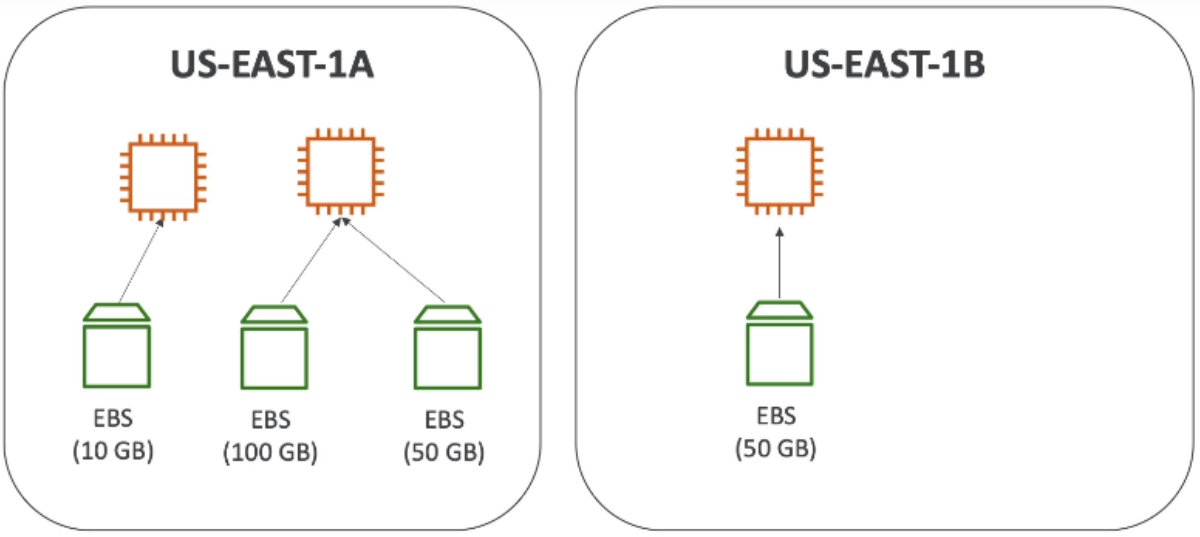
9.
10. EBS volume is a network drive..
11. At first we have specify how much GB we need and how mant IOPS we need (IOPS - Input/output operations perseconds) to define how our EBS volume performs

EBS Volume

- It's a network drive (i.e. not a physical drive)
  - It uses the network to communicate the instance, which means there might be a bit of latency
  - It can be detached from an EC2 instance and attached to another one quickly
- It's locked to an Availability Zone (AZ)
  - An EBS Volume in us-east-1a cannot be attached to us-east-1b
  - To move a volume across, you first need to snapshot it
- Have a provisioned capacity (size in GBs, and IOPS)
  - You get billed for all the provisioned capacity

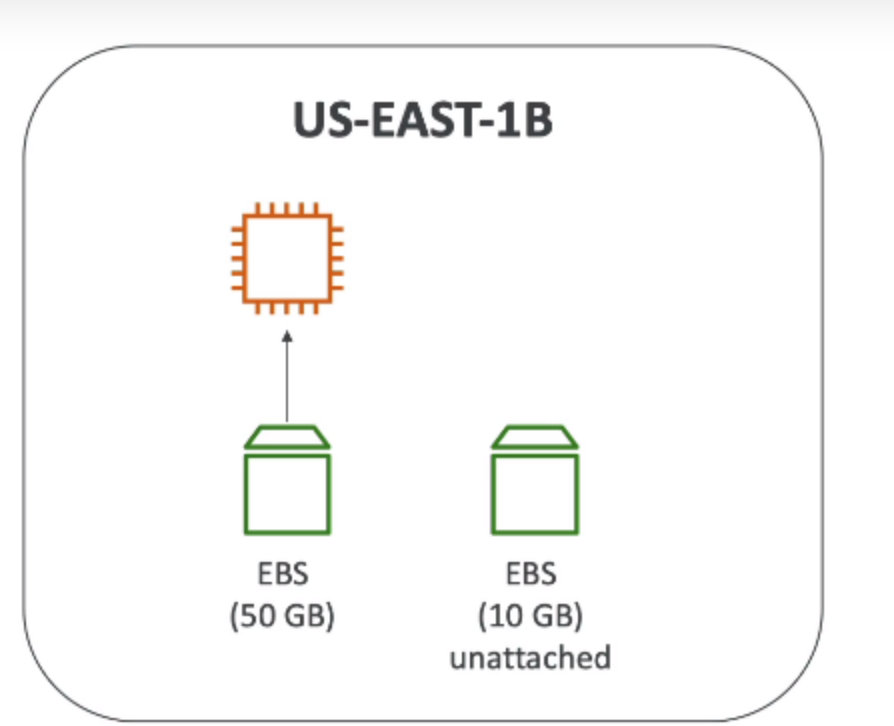
12.

EBS Volume - Example



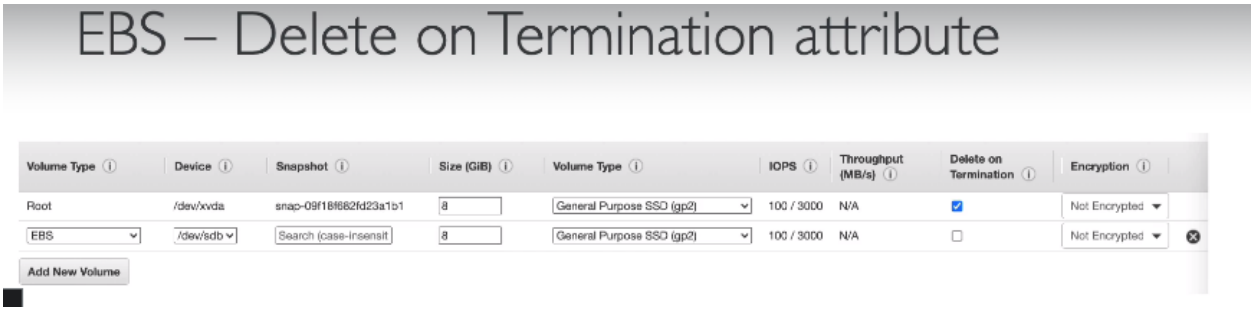
13.

14. Also we can create an EBS volume and left it unattached and later we can attach an instance to it



15.

16. EBS Delete on termination attribute



17. ■

- Controls the EBS behaviour when an EC2 instance terminates
  - By default, the root EBS volume is deleted (attribute enabled)
  - By default, any other attached EBS volume is not deleted (attribute disabled)
- This can be controlled by the AWS console / AWS CLI
- Use case: preserve root volume when instance is terminated

18.

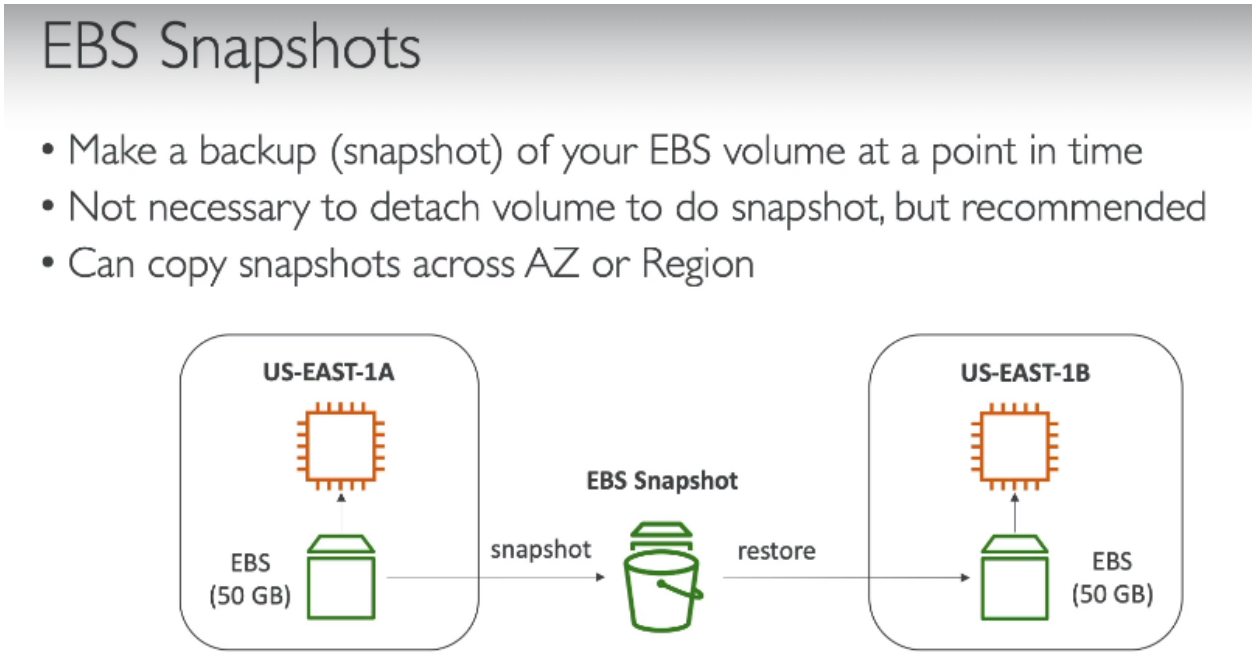
19. If we want to preserver the root volume data..then we have to disable it

EBS HANDS ON:

1. First we have to go our instance then click on the instance
2. After that go to Storage tab of that instance …there we will be having root device name
3. By default we have one volume attached to our instance which is of 8gb..click on that volume
4. We can also directly go to the volume section which is on the left side and access our volumes
5. Lets create a new volume by choosing availability zone same as our ec2 instance AZ…
6. Initially it will be unattached to any instance…Later we can attach this to any instance which is running..by going to actions
7. We can only attach volume to an ec2 instance …if they both are in same AZ(availaility zone)
8. While creating an instance we have an option to choose on_termination attribute…if it is in yes…then if we delete an instance ..the volume which is attached to it will also get deleted

EBS Snapshot Overview:

1. It helps us to make a backup of our volume at a point in time
2. It is recommended to do a snapshot when it is not attached to any instance
3. It helps us to copy snapshots across AZ or region ..so we can transfer our data from one region to other region



4.

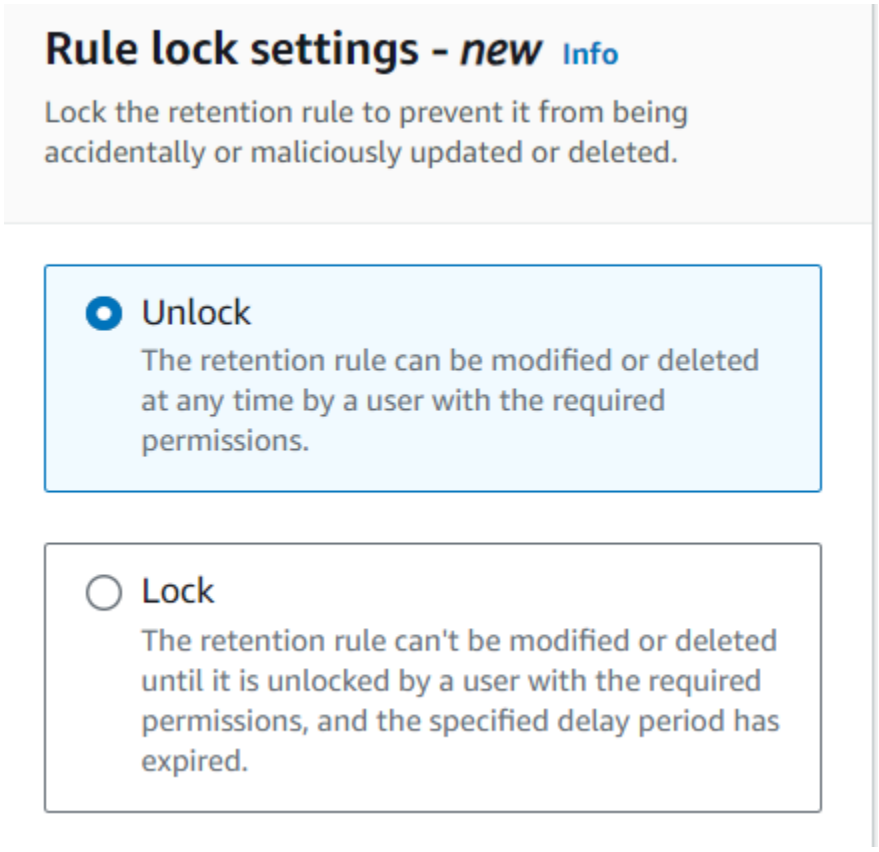
5. First we have to snapchat our ebs volume and we can restore the same thing into other region(B)…and then we can attach to ec2 instance of other region(B)
6. EBS Snapshots Features

7.
8. We can use EBS Snapshot Archive if the data is not very much imp to us…it is very cheap
9. If we delete a snapshot we have 1 year time to retain it back…after that it is gone

EBS Snapshot handson:

1. We can create a snapshot of our ebs volume..
2. By going into ebs volume and click on one value and go to its action and create a snapshot for it
3. We can see all the snapshots in the snapshots section
4. Here if go to actions of the snapshot ..we have a option to copy it to the other AZ
5. We can also create volumes from our snapshot by going into its action

6. 

7. We can create retention rules for our snapshot
8. Rule lock settings
9. 

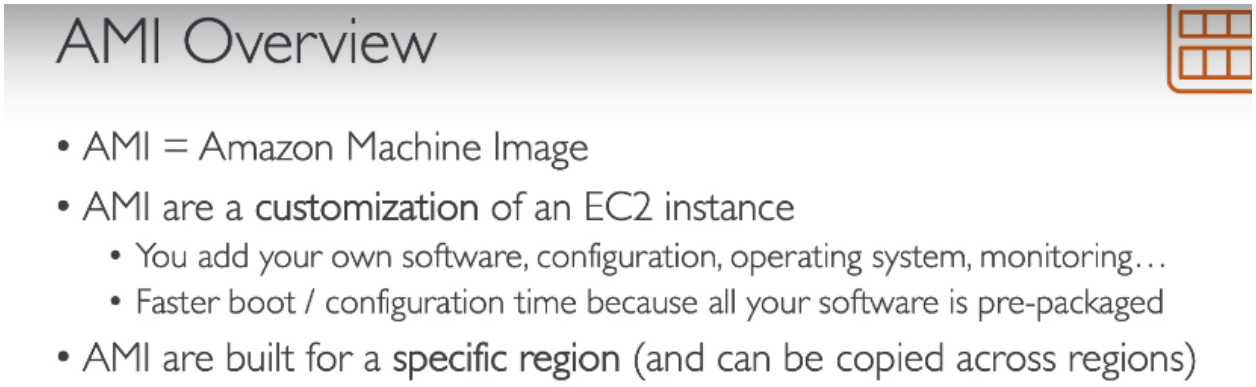
10. Once a rule is locked, it will remain locked for the specified unlock delay period. After the unlock delay period expires, the rule will be unlocked and can be modified or deleted.

    Here are some things to keep in mind about rule lock settings:

    - Only users with the **ModifyRetentionRules** permission can lock or unlock retention rules.
    - You cannot lock a rule that is already locked.
    - If you lock a rule that is currently in the process of being deleted, the deletion will be canceled.
    - If you delete a locked rule, the rule will be deleted immediately and the unlock delay period will not be applied.
11. Deleted snapshots will be in the section resources in the recyclebin
12. There we can again restore it from the resources section

AMI OverView:

1. Amazon Machine Image



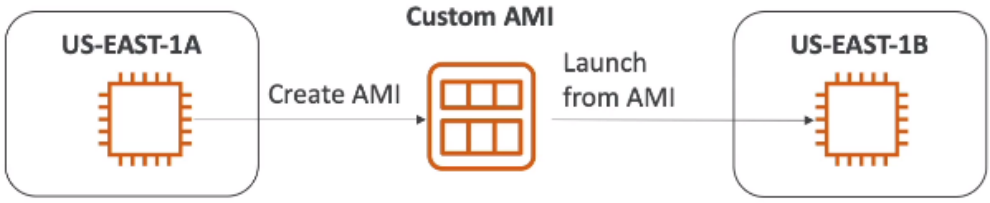
2.
3. If we create our own AMI we get faster boot/ ….all the software will be pre installed in the ec2 instance

- AMI are built for a **specific region** (and can be copied across regions)
- You can launch EC2 instances from:
  - A Public AMI: AWS provided
  - Your own AMI: you make and maintain them yourself
  - An AWS Marketplace AMI: an AMI someone else made (and potentially sells)

4.
5. We can launch EC2 instances from a public AMI (remember an EC2 instance is a virtual machine/computer)
6. People who bought dedicated host…can build an AMI and sell it on the marketplace
7. AMI Process ..

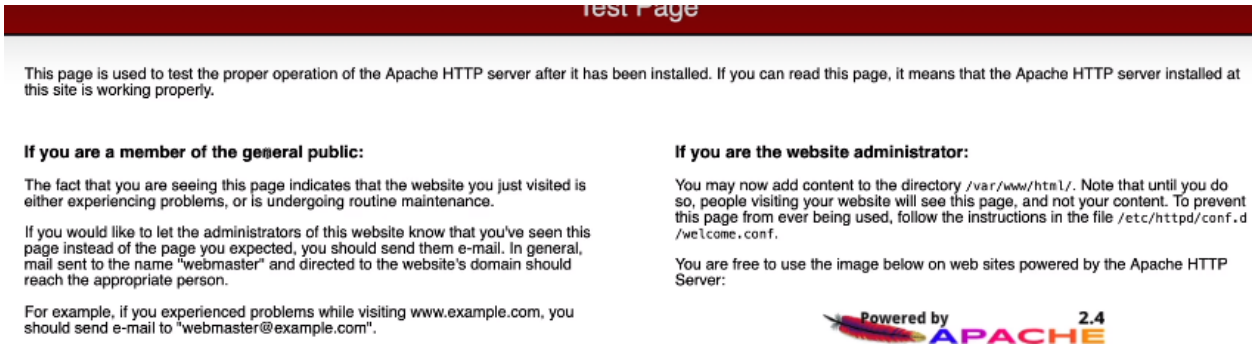
AMI Process (from an EC2 instance)

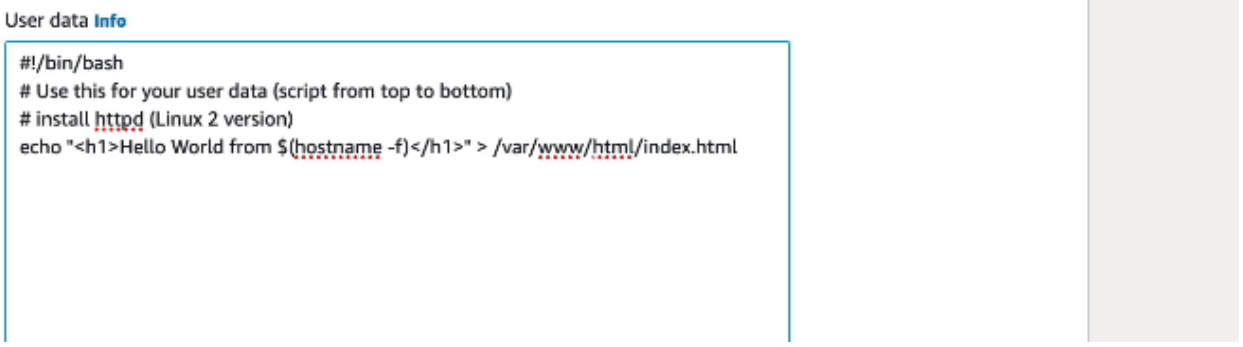
- Start an EC2 instance and customize it
- Stop the instance (for data integrity)
- Build an AMI – this will also create EBS snapshots
- Launch instances from other AMIs



8.
9. Here we create an instance in US-East1a and we will create an AMI for it..while building an AMI ..it will also create EBS snapshots behind the scenes
10. The AMI which we create for 1a will be our custom AMI
11. So we have US-EAST-1A, and we can create the same instance as US-EAST-1B,
12. so the process is we launch the instance in US-EAST-1A, we're going to customize it..then we're going to create an AMI from it
13. this will be our custom AMI. And then in US-EAST-1B ..we will be able to launch from that AMI and we'll effectively create a copy of our EC2 instance.

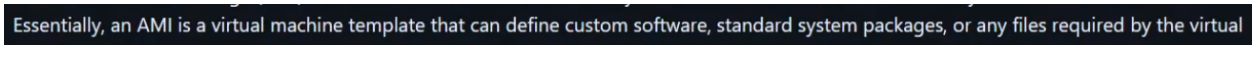
AMI Hands On:

1. First we will launch an instance by specifying all the details like storage details,network details etc ..and in user data we gave 4lines from code..and we have launched an instance
2. After creating if we open its public address on internet we get this
3. 

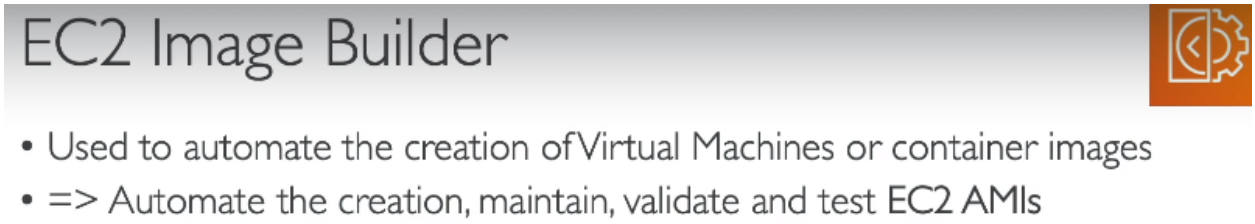
4. Now we will create an AMI for this instance
5. To create an AMI..right click on instance and go to images and templates and click on create image
6. We have created an AMI for this 9gb instance and names the AMI as DemoAMI
7. Now we can launch an instance from this AMI..while creating an Instance from AMI in advanced setting -> in the user data we only give first 3 lines only and index.html line ..as our AMI has 4 lines installed already
8. 

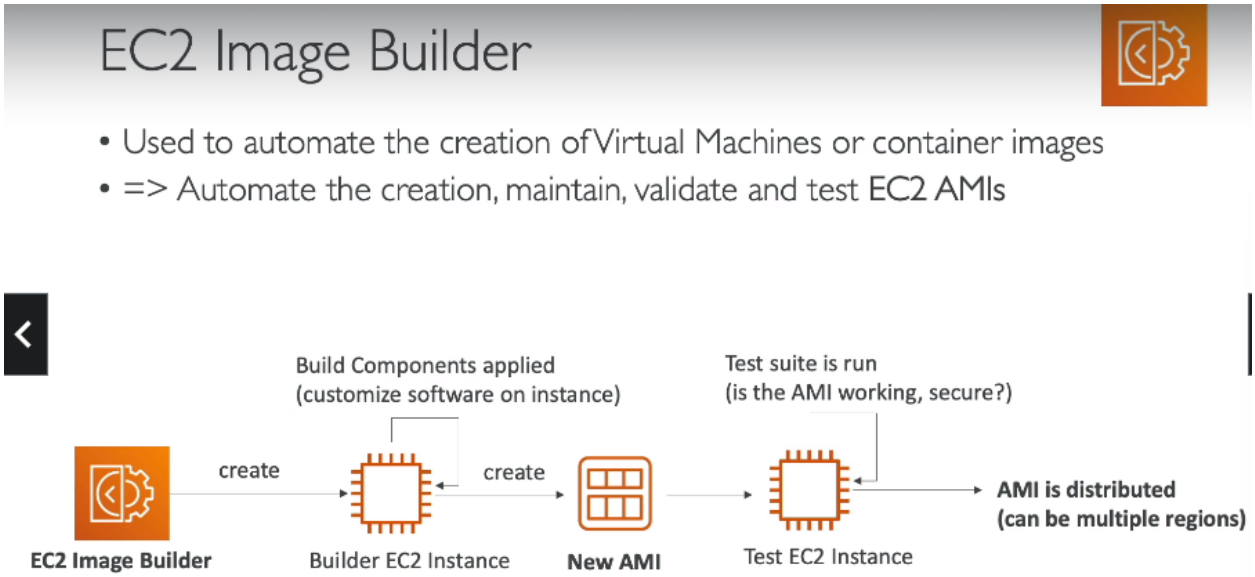
9. We ddidnt write any http for our instance because our AMI has already installed in it
10. This helps in faster bootup time…thats y we create AMI

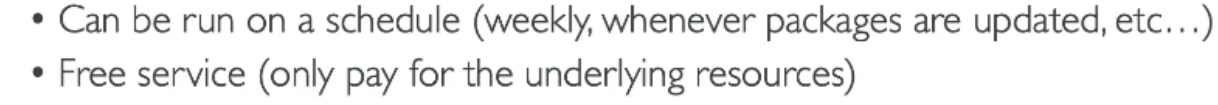
Hello World from ip-172-31-91-39.ec2.internal

11.
12. We get this output from our FromAMI instance
13. Not only http we can also do prerequisite software,security software In AMI and create a instance with it …for faster boot-up
14. 

EC2 Image Builder

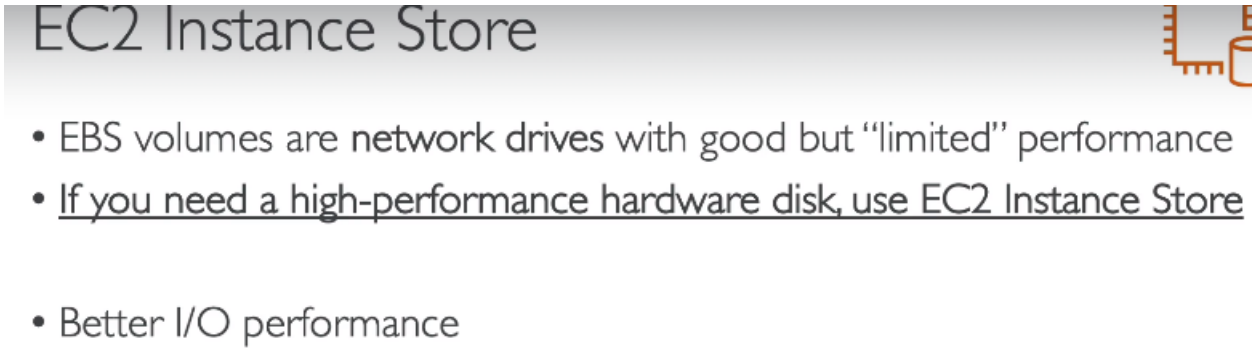
1. It is used to automate the creation of virtual machines
2. 

3. 

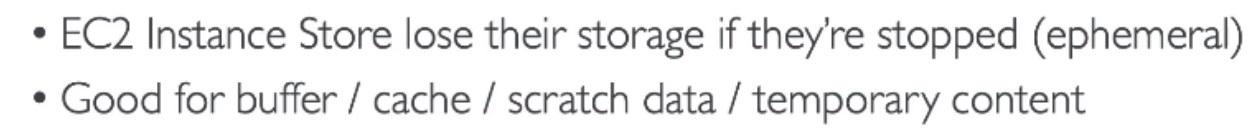
4. So we have the EC2 image builder service and we're going to set it up And it is automatically going when it's going to run, it is going to create an EC2 instance called
5. a builder EC2 instance, and that EC2 instance is going to build components and customize the software. For example, install Java, update the CLI, update the software system, maybe install firewalls,
6. whatever you define to happen on that EC2 instance, maybe install your application and then once this is done, then an Ami is going to be created out of that EC2 instance. But all of this is obviously automated. Then the image is created, but we want to validate it.
7. So EC2 image builder will automatically create a test EC2 instance from that Ami and going to run a
8. bunch of tests that you are defining in advance. And if you don't want to run a test, obviously you can just skip that test.
9. It can also be run on Schedule …like when to updates the softwares,packages etc
10. 

EC2 Instance Store

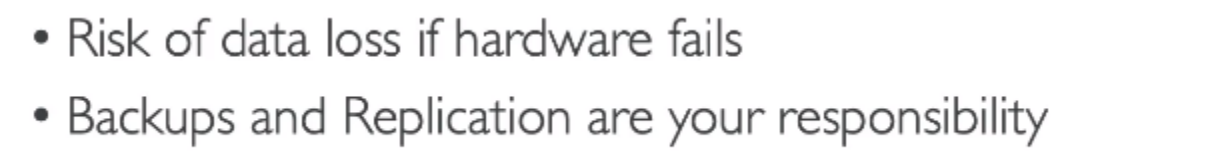
1. Ebs volumes are network drives but they have limited performance
2. If we need high performance it is going to be a hardware disk attached to our ec2 instance
3. Ec2 will be a virtual machine but it is obviously attached to a real hardware server.
4. Some of this servers ..do have the disk space that is attached directly on to server
5. So what we do with EC2 Instance Store, we use them for better I/O performance.
6. We also make sure that they have good through put and so on, so they're a great choice



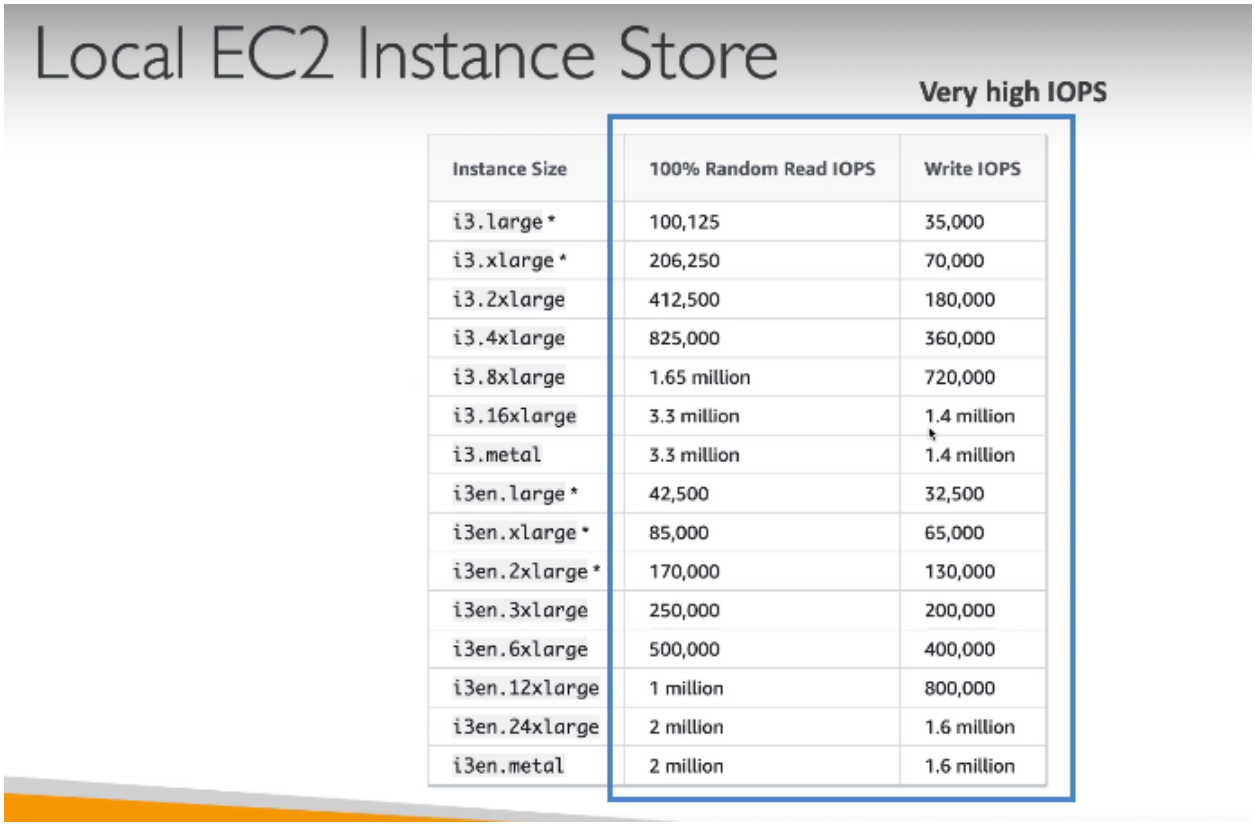
7.



8.
9. It is not suitable for long term storage
10. Like if we stop our ec2 instance that has an instance store…our storage will be lost



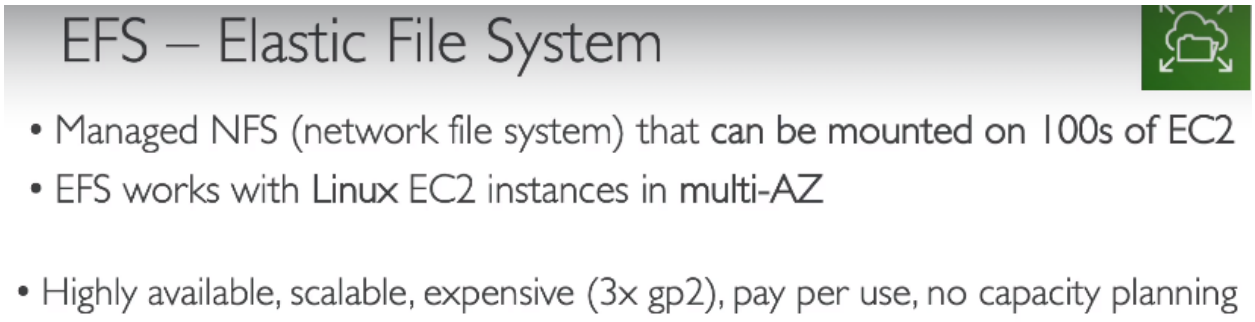
11.



12.
13. Here the highest is 3.3 million RIOPs and 1.3 million WIOPs..the same without instance store would be very less

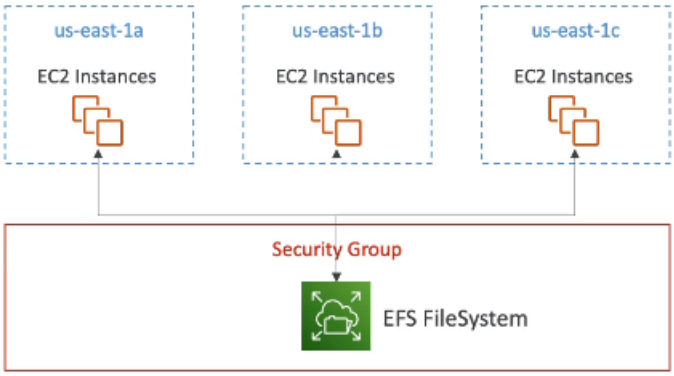
EFS - Elastic File System

1. It is a type of storage we can attache onto ec2 instance
2. EFS is a managed network file system that can be mounter on 100's of ec2's
3. EBS volume can only be attached to one ec2 instance at a time

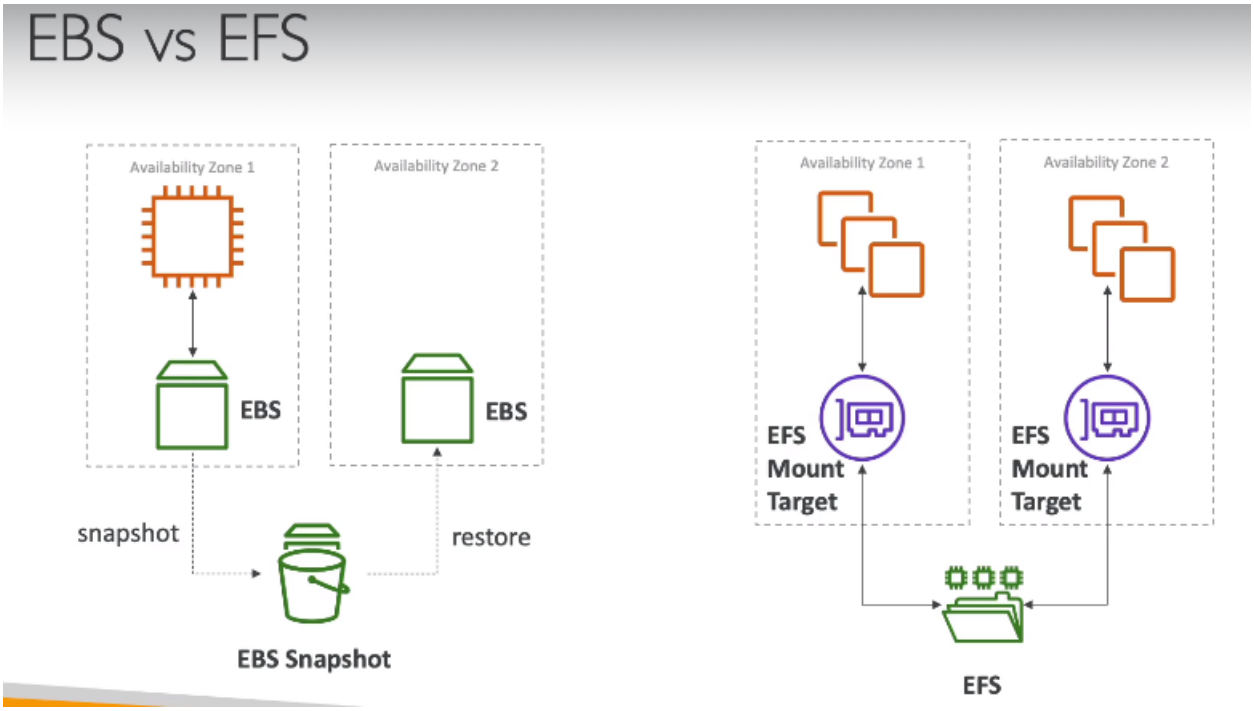


4.

5.
6. We have a EFS file system with the security group on it
7. And we have different ec2 instances in different AZ..they all can be connected on EFS



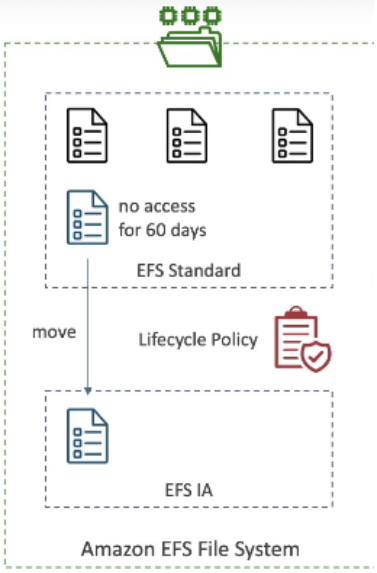
8.
9. EBS vs EFS
10. In EBS if we want to use it into another region ..we have to snapshot and restore it in other region
11. But in EFS ..consider we have many instance in A-Z1 and many instances in A-Z2…they all can be connected to one EFS



12.
13. Say we have many instances in Availability Zone 1 on one or many instances as well on Availability Zone 2. At the same time, all these instances
14. can mount the same EFS drive,using a mount target, and they will all see the same files. So, that makes it a shared file system.
15. EFS-IA

EFS Infrequent Access (EFS-IA)

- **Storage class** that is cost-optimized for files not accessed every day
- 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
- Example: move files that are not accessed for 60 days to EFS-IA
- Transparent to the applications accessing EFS



16.
17. If we enable life cycle policy..then the files which are not used in EFS for more than specified time ..then they will be moved to EFS-IA
18. We cannot see EFS-IA and all..it happens behind the scene

Shared responsibility model for ec2 storage

1. The replication of data in EBS and EFS drives is the responsibility of AWS ..to replicate across many hardware
2. If one day hardware is not working we must not be responsible for it
3. Also, anytime an EBS drive would fail,
4. or one part of it would fail. It is obviously AWS responsibility to replace them

Shared Responsibility Model for EC2 Storage

- Infrastructure
- Replication for data for EBS volumes & EFS drives
- Replacing faulty hardware
- Ensuring their employees cannot access your data

- Setting up backup / snapshot procedures
- Setting up data encryption
- Responsibility of any data on the drives
- Understanding the risk of using EC2 Instance Store

5.

Amazon FSx-Overview

Amazon FSx – Overview

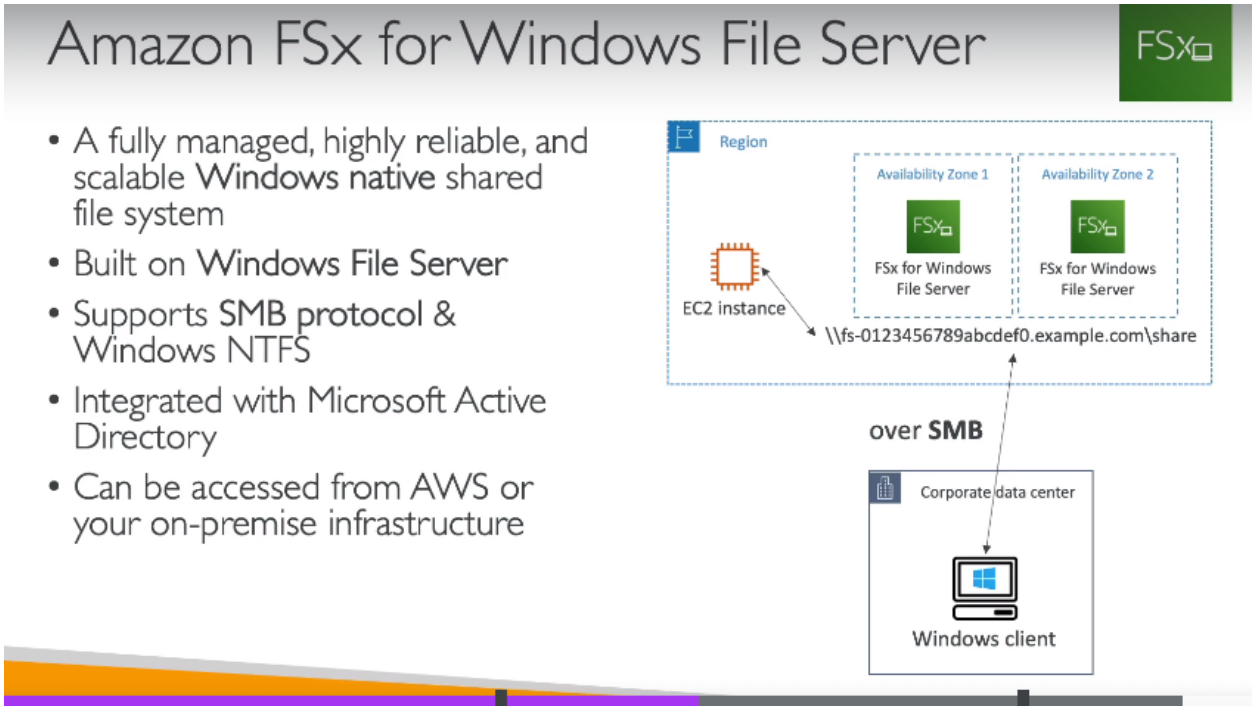
- Launch 3rd party high-performance file systems on AWS
- Fully managed service

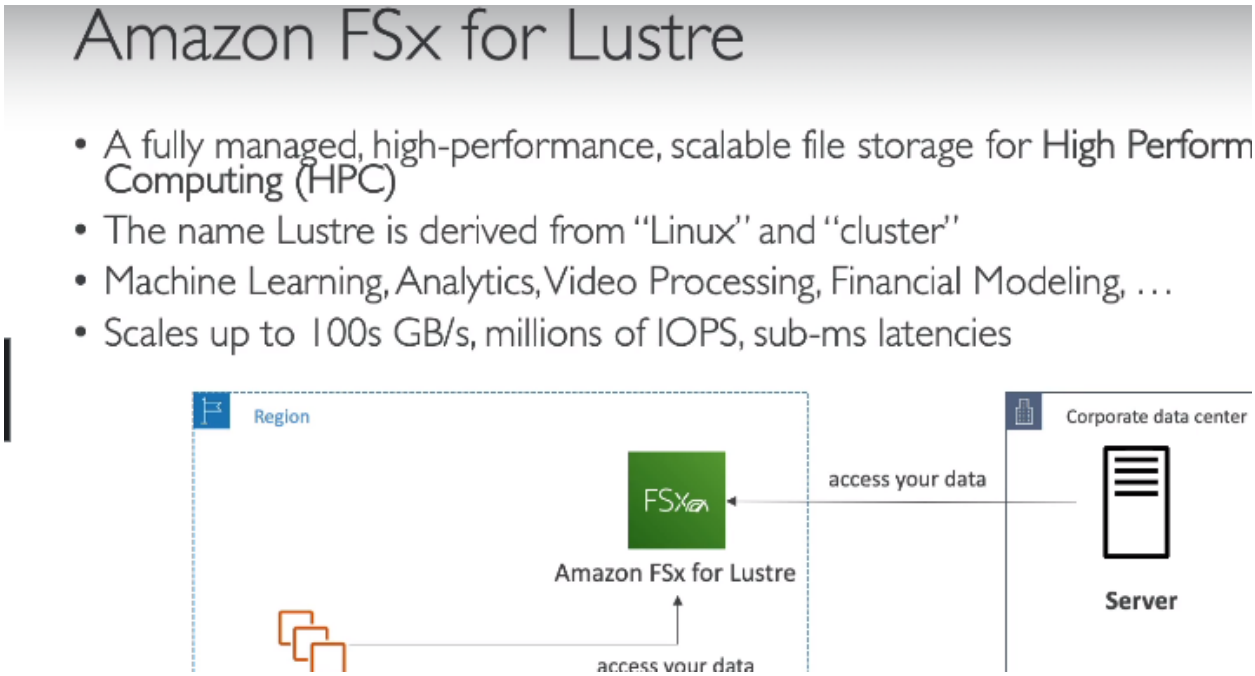
FSx for Lustre

FSx for Windows File Server

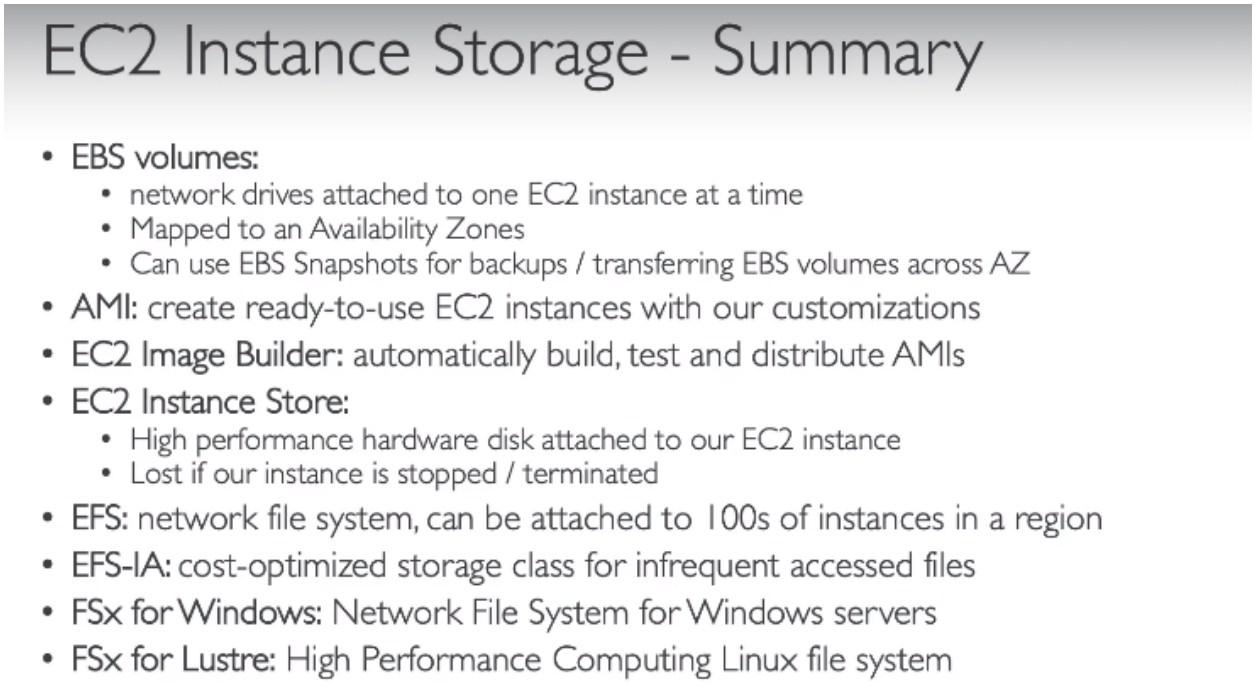
FSx for NetApp ONTAP

1.
2. FSx for windows file server

3. 

4. Amazon Fsx for lustre
5. So the way it works is that Amazon FSx for Lustrecan be connected either to your corporate data center
6. or to your compute instances directly within AWS. And then in the backend, Amazon FSx for Lustre is actually storing your data, possibly onto an Amazon S3 bucket.
7. 

EC2 instance storage summary



1.
2. Have to learn more about FSx