

# EBS & EFS

# What's an EBS Volume?

- An EC2 machine loses its root volume (main drive) when it is manually terminated.
- Unexpected terminations might happen from time to time (AWS would email you)
- Sometimes, you need a way to store your instance data somewhere
- An **EBS (Elastic Block Store) Volume** is a **network** drive you can attach to your instances while they run
- It allows your instances to persist data

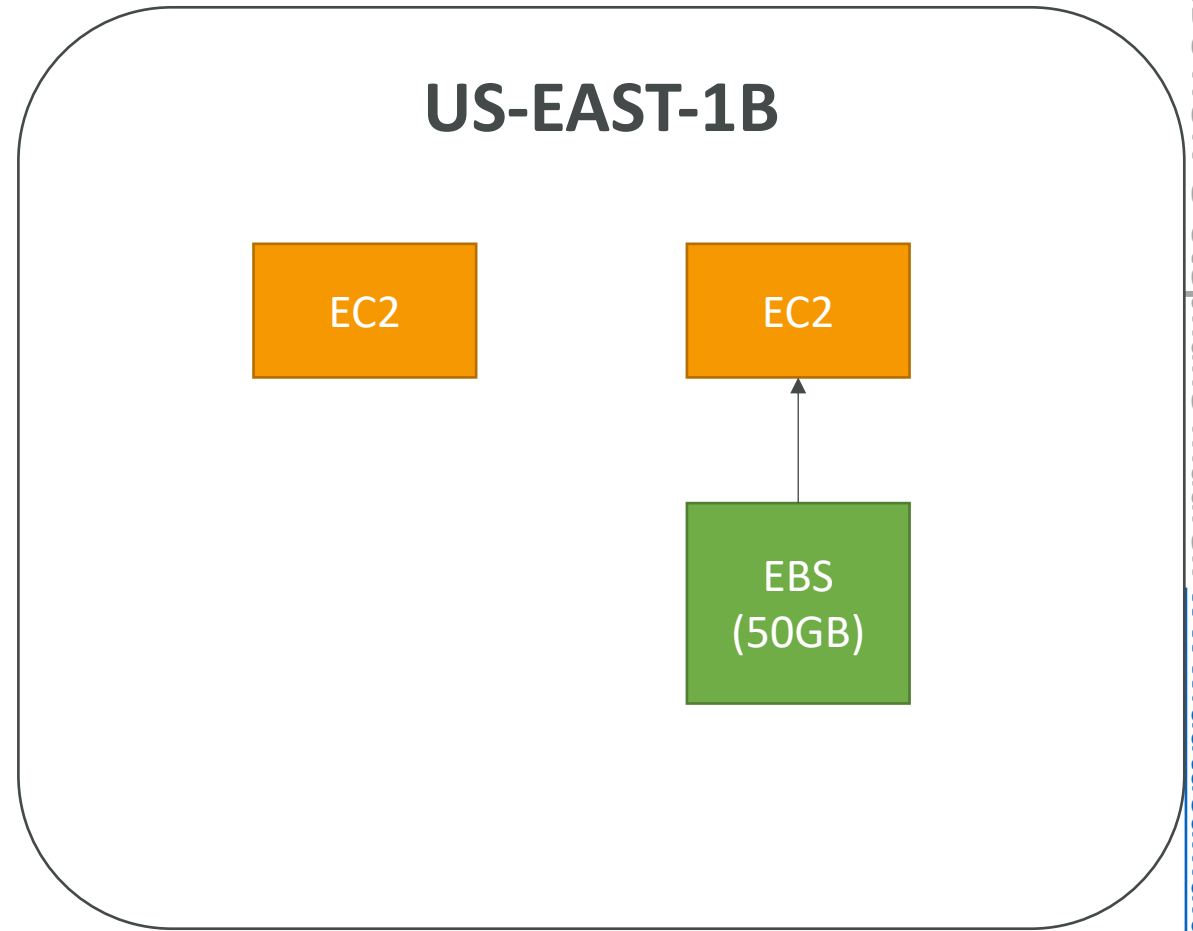
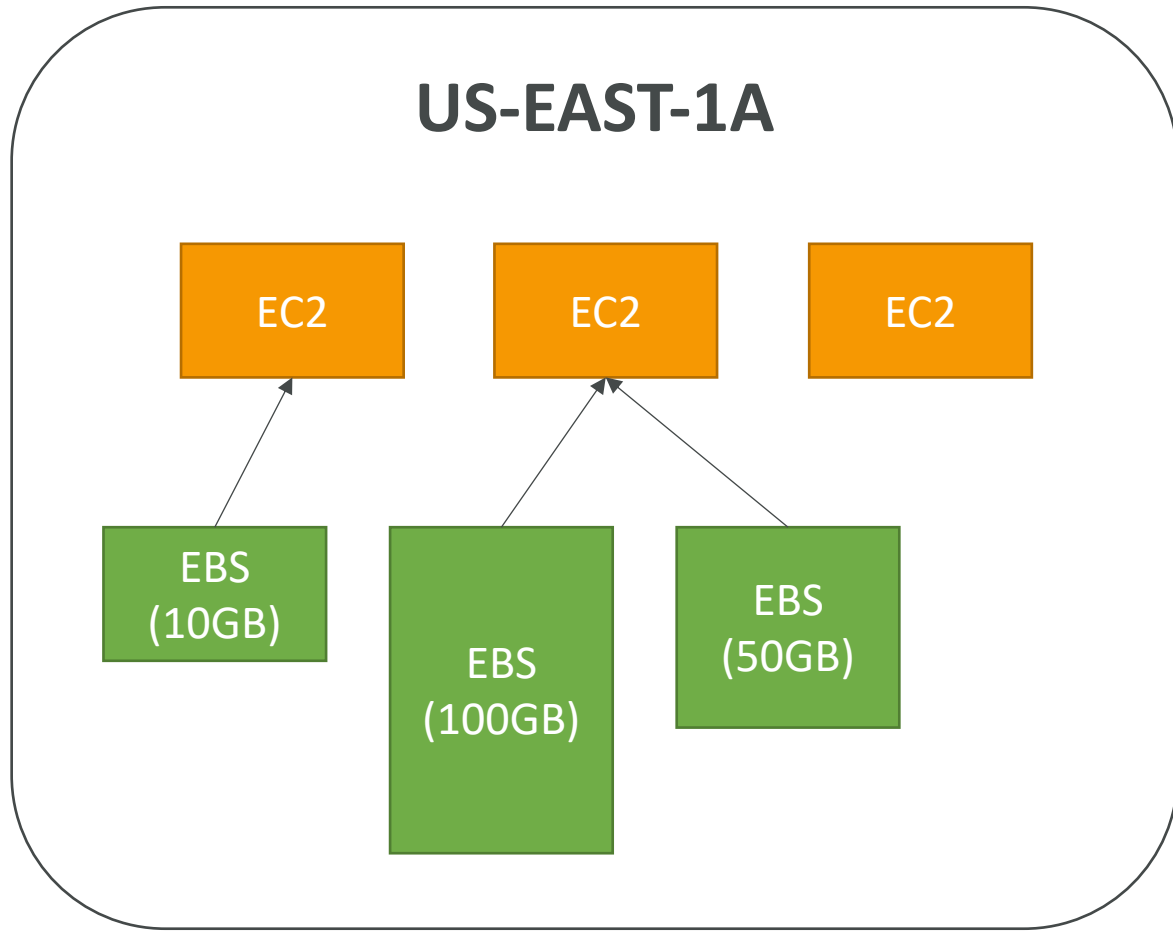


Amazon EBS

# 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
  - You can increase the capacity of the drive over time

# EBS Volume Example



# EBS Volume Types

- EBS Volumes come in 4 types
  - **GP2 (SSD)**: General purpose SSD volume that balances price and performance for a wide variety of workloads
  - **IO1 (SSD)**: Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads
  - **ST1 (HDD)**: Low cost HDD volume designed for frequently accessed, throughput-intensive workloads
  - **SC1 (HDD)**: Lowest cost HDD volume designed for less frequently accessed workloads
- EBS Volumes are characterized in Size | Throughput | IOPS (I/O Ops Per Sec)
- When in doubt always consult the AWS documentation – it's good!
- Only GP2 and IO1 can be used as boot volumes

# EBS Volume Types Use cases

## GP2 (from AWS doc)

- Recommended for most workloads
  - System boot volumes
  - Virtual desktops
  - Low-latency interactive apps
  - Development and test environments
- 
- 1 GiB - 16 TiB
  - Small gp2 volumes can burst IOPS to 3000
  - Max IOPS is 16,000...
  - 3 IOPS per GB, means at 5,334GB we are at the max IOPS

# EBS Volume Types Use cases

## IO1 (from AWS doc)

- Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS per volume (gp2 limit)
  - Large database workloads, such as:
  - MongoDB, Cassandra, Microsoft SQL Server, MySQL, PostgreSQL, Oracle
- 
- 4 GiB - 16 TiB
  - IOPS is provisioned (PIOPS) – MIN 100 - MAX 64,000 (Nitro instances) else MAX 32,000 (other instances)
  - The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1

# EBS Volume Types Use cases

## ST1 (from AWS doc)

- Streaming workloads requiring consistent, fast throughput at a low price.
  - Big data, Data warehouses, Log processing
  - Apache Kafka
  - Cannot be a boot volume
- 
- 500 GiB - 16 TiB
  - Max IOPS is 500
  - Max throughput of 500 MiB/s – can burst



# EBS Volume Types Use cases

## SC1 (from AWS doc)

- 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
- 500 GiB - 16 TiB
- Max IOPS is 250
- Max throughput of 250 MiB/s – can burst

# EBS – Volume Types Summary

- **gp2: General Purpose Volumes (cheap)**
  - 3 IOPS / GiB, minimum 100 IOPS, burst to 3000 IOPS, max 16000 IOPS
  - 1 GiB – 16 TiB , +1 TB = +3000 IOPS
- **io1: Provisioned IOPS (expensive)**
  - Min 100 IOPS, Max 64000 IOPS (Nitro) or 32000 (other)
  - 4 GiB - 16 TiB. Size of volume and IOPS are independent
- **st1: Throughput Optimized HDD**
  - 500 GiB – 16 TiB , 500 MiB /s throughput
- **sc1: Cold HDD, Infrequently accessed data**
  - 500 GiB – 16 TiB , 250 MiB /s throughput

# EBS Snapshots

- Incremental – only backup changed blocks
- EBS backups use IO and you shouldn't run them while your application is handling a lot of traffic
- Snapshots will be stored in S3 (but you won't directly see them)
- Not necessary to detach volume to do snapshot, but recommended
- Max 100,000 snapshots
- Can copy snapshots across AZ or Region
- Can make Image (AMI) from Snapshot
- EBS volumes restored by snapshots need to be pre-warmed (using fio or dd command to read the entire volume)
- Snapshots can be automated using Amazon Data Lifecycle Manager

# EBS Migration

- EBS Volumes are only locked to a specific AZ
- To migrate it to a different AZ (or region):
  - Snapshot the volume
  - (optional) Copy the volume to a different region
  - Create a volume from the snapshot in the AZ of your choice
- Let's practice!

# EBS Encryption

- When you create an encrypted EBS volume, you get the following:
  - Data at rest is encrypted inside the volume
  - All the data in flight moving between the instance and the volume is encrypted
  - All snapshots are encrypted
  - All volumes created from the snapshot
- Encryption and decryption are handled transparently (you have nothing to do)
- Encryption has a minimal impact on latency
- EBS Encryption leverages keys from KMS (AES-256)
- Copying an unencrypted snapshot allows encryption
- Snapshots of encrypted volumes are encrypted

# Encryption: encrypt an unencrypted EBS volume

- Create an EBS snapshot of the volume
- Encrypt the EBS snapshot ( using copy )
- Create new ebs volume from the snapshot ( the volume will also be encrypted )
- Now you can attach the encrypted volume to the original instance

# EBS vs Instance Store

- Some instance do not come with Root EBS volumes
- Instead, they come with “Instance Store” (= ephemeral storage)
- Instance store is physically attached to the machine (EBS is a network drive)
- Pros:
  - Better I/O performance (EBS gp2 has an max IOPS of 16000, io1 of 64000)
  - Good for buffer / cache / scratch data / temporary content
  - Data survives reboots
- Cons:
  - On stop or termination, the instance store is lost
  - You can't resize the instance store
  - Backups must be operated by the user

# Local EC2 Instance Store

- Physical disk attached to the physical server where your EC2 is
- Very High IOPS (because physical)
- Disks up to 7.5 TiB (can change over time), stripped to reach 30 TiB (can change over time...)
- Block Storage (just like EBS)
- Cannot be increased in size
- Risk of data loss if hardware fails

Very high IOPS

Instance Size	100% Random Read IOPS	Write IOPS
i3.large *	100,125	35,000
i3.xlarge *	206,250	70,000
i3.2xlarge	412,500	180,000
i3.4xlarge	825,000	360,000
i3.8xlarge	1.65 million	720,000
i3.16xlarge	3.3 million	1.4 million
i3.metal	3.3 million	1.4 million
i3en.large *	42,500	32,500
i3en.xlarge *	85,000	65,000
i3en.2xlarge *	170,000	130,000
i3en.3xlarge	250,000	200,000
i3en.6xlarge	500,000	400,000
i3en.12xlarge	1 million	800,000
i3en.24xlarge	2 million	1.6 million
i3en.metal	2 million	1.6 million

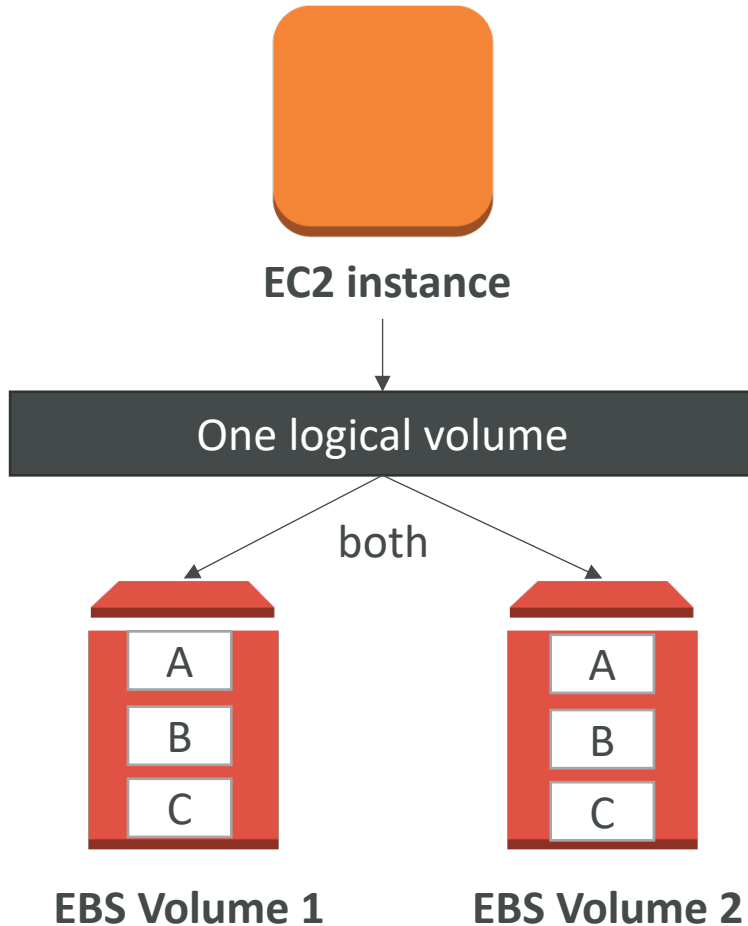


# EBS RAID Options

- EBS is already redundant storage (replicated within an AZ)
- But what if you want to increase IOPS to say 100 000 IOPS?
- What if you want to mirror your EBS volumes?
- You would mount volumes in parallel in RAID settings!
- RAID is possible as long as your OS supports it
- Some RAID options are:
  - RAID 0
  - RAID 1
  - RAID 5 (not recommended for EBS – see documentation)
  - RAID 6 (not recommended for EBS – see documentation)
- We'll explore RAID 0 and RAID 1



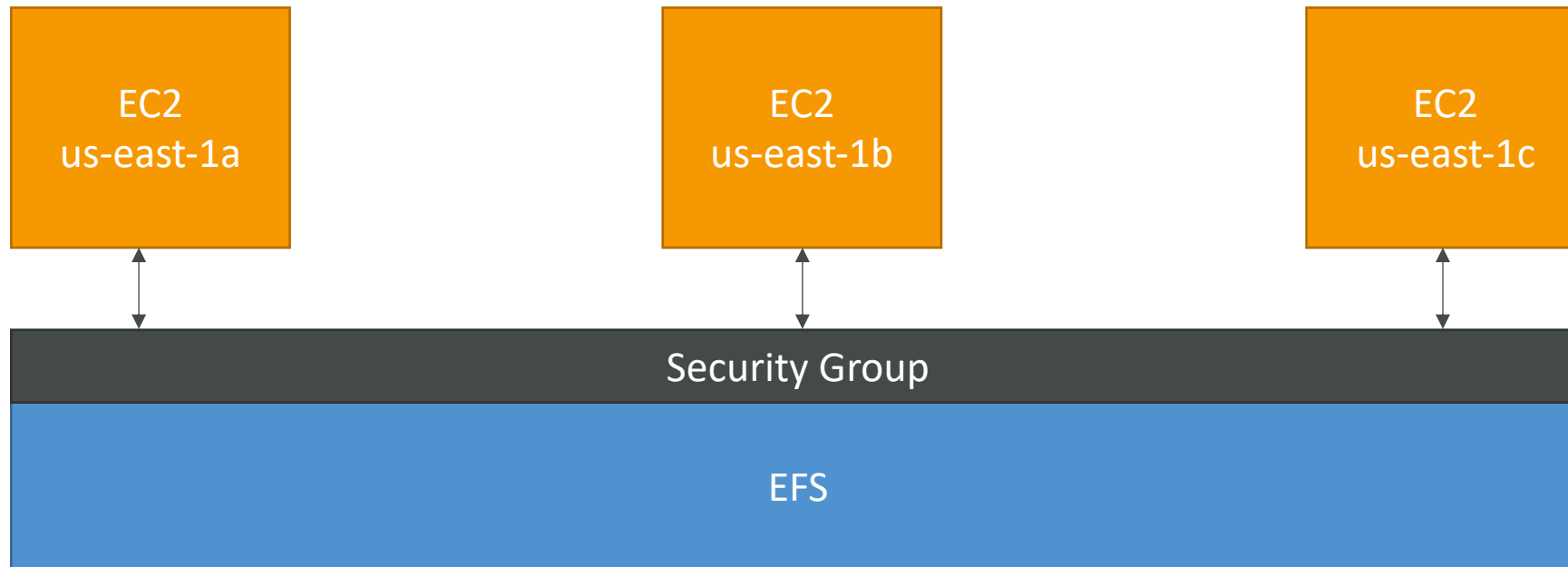
# RAID 1 (increase fault tolerance)



- RAID 1 = Mirroring a volume to another
- If one disk fails, our logical volume is still working
- We have to send the data to two EBS volume at the same time (2x network)
- Use case:
  - Application that need increase volume fault tolerance
  - Application where you need to service disks
- For example:
  - two 500 GiB Amazon EBS io1 volumes with 4,000 provisioned IOPS each will create a...
  - 500 GiB RAID 1 array with an available bandwidth of 4,000 IOPS and 500 MB/s of throughput

# EFS – Elastic File System

- Managed NFS (network file system) that can be mounted on many EC2
- EFS works with EC2 instances in multi-AZ
- Highly available, scalable, expensive (3x gp2), pay per use



# EFS – Elastic File System

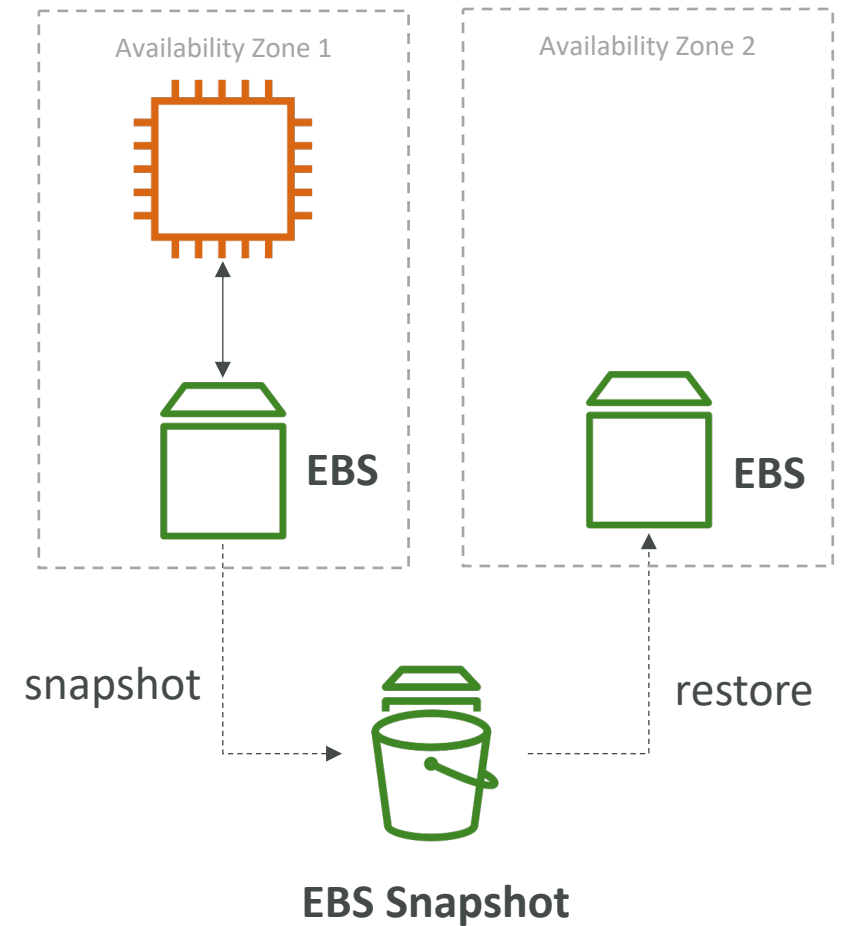
- Use cases: content management, web serving, data sharing, Wordpress
  - Uses NFSv4.1 protocol
  - Uses security group to control access to EFS
  - **Compatible with Linux based AMI (not Windows)**
  - Encryption at rest using KMS
- 
- POSIX file system (~Linux) that has a standard file API
  - File system scales automatically, pay-per-use, no capacity planning!

# EFS – Performance & Storage Classes

- **EFS Scale**
  - 1000s of concurrent NFS clients, 10 GB+ /s throughput
  - Grow to Petabyte-scale network file system, automatically
- **Performance mode (set at EFS creation time)**
  - General purpose (default): latency-sensitive use cases (web server, CMS, etc...)
  - Max I/O – higher latency, throughput, highly parallel (big data, media processing)
- **Storage Tiers (lifecycle management feature – move file after N days)**
  - Standard: for frequently accessed files
  - Infrequent access (EFS-IA): cost to retrieve files, lower price to store

# EBS vs EFS – Elastic Block Storage

- EBS volumes...
  - can be attached to only one instance at a time
  - are locked at the Availability Zone (AZ) level
  - gp2: IO increases if the disk size increases
  - io1: can increase IO independently
- To migrate an EBS volume across AZ
  - Take a snapshot
  - Restore the snapshot to another AZ
  - EBS backups use IO and you shouldn't run them while your application is handling a lot of traffic
- Root EBS Volumes of instances get terminated by default if the EC2 instance gets terminated. (you can disable that)



# EBS vs EFS – Elastic File System

- Mounting 100s of instances across AZ
- EFS share website files (WordPress)
- Only for Linux Instances (POSIX)
- EFS has a higher price point than EBS
- Can leverage EFS-IA for cost savings
- Remember: EFS vs EBS vs Instance Store

