Difference between EBS and EFS

AWS EBS (Elastic Block Store) and AWS EFS (Elastic File System) are both storage services provided by Amazon Web Services (AWS), but they have some key differences in terms of their functionality and use cases. Here's a breakdown of the main differences:

1. **Storage Type**: EBS provides block-level storage, which means it presents storage volumes as raw block devices that can be attached to EC2 instances. On the other hand, EFS offers file-level storage, where it provides a managed file system that can be mounted on multiple EC2 instances concurrently.

2. **Access Method**: EBS volumes are designed to be mounted on a single EC2 instance at a time, offering access to block storage over the network. In contrast, EFS is designed for shared access, allowing multiple EC2 instances to access the file system concurrently using the Network File System (NFS) protocol.

3. **Scalability**: EBS volumes can be scaled up or down in terms of storage capacity and performance independently. However, they are limited to a single EC2 instance at a time. EFS, on the other hand, is highly scalable and can grow or shrink automatically based on the data stored in it. It allows for concurrent access from multiple instances, making it suitable for applications with shared data requirements.

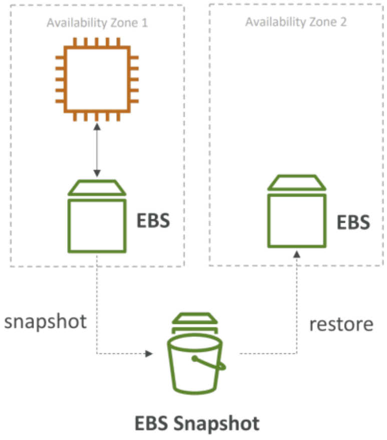
4. **Use Cases**: EBS is commonly used for individual EC2 instances that require persistent block storage, such as database storage, boot volumes, and applications that require low-latency access to data. EFS, with its shared access and scalability, is well-suited for use cases such as content management systems, web serving, data sharing, and other applications that require shared file storage across multiple instances.

5. **Pricing Model**: EBS pricing is based on the provisioned storage capacity and the associated provisioned IOPS (input/output operations per second), while EFS pricing is based on the storage capacity used. EFS also incurs additional costs for data transfer and per-request charges.

In summary, EBS provides block-level storage for single EC2 instances, while EFS offers scalable, shared file storage for multiple EC2 instances. The choice between EBS and EFS depends on your specific requirements, such as the access pattern, scalability needs, and type of storage your applications demand.

EBS

- EBS volumes…
  - one instance (except multi-attach io1/io2)
  - 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)

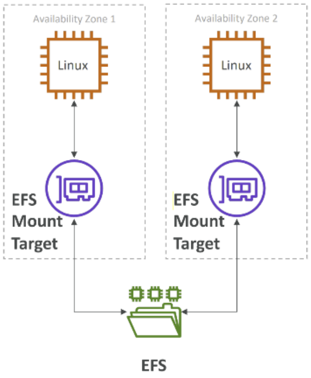


EFS

- Mounting 100s of instances across AZ
- EFS share website files (WordPress)
- Only for Linux Instances (POSIX)

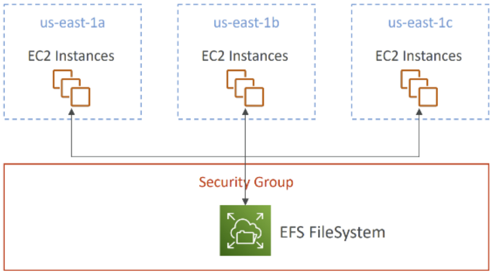
- EFS has a higher price point than EBS
- Can leverage EFS-IA for cost savings

- Remember: EFS vs EBS vs Instance Store



Amazon 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



Some important features of EFS

- 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 and storage class

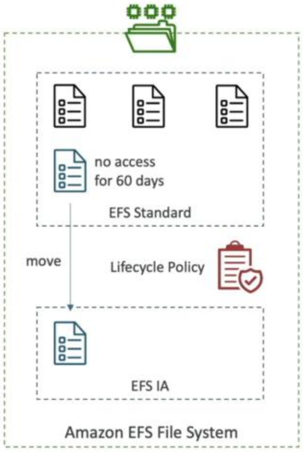
- EFS Scale
  - 1000s of concurrent NFS clients, 10 GB+ /s throughput
  - Grow to Petabyte-scale network file system, automatically
- Performance Mode (set at EFS creation time)
  - General Purpose (default) – latency-sensitive use cases (web server, CMS, etc…)
  - Max I/O – higher latency, throughput, highly parallel (big data, media processing)
- Throughput Mode
  - Bursting – 1 TB = 50MiB/s + burst of up to 100MiB/s
  - Provisioned – set your throughput regardless of storage size, ex: 1 GiB/s for 1 TB storage
  - Elastic – automatically scales throughput up or down based on your workloads
    - Up to 3GiB/s for reads and 1GiB/s for writes
    - Used for unpredictable workloads

EFS- Storage Classses

- 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. Enable EFS-IA with a Lifecycle Policy

- Availability and durability
  - Standard: Multi-AZ, great for prod
  - One Zone: One AZ, great for dev, backup enabled by default, compatible with IA (EFS One Zone-IA)

- Over 90% in cost savings



LAB: EFS

Create 2 instances.

Instance A
instance B

Create a efs

After creating the EFS click on attach then you get the command to run in ec2 instance to use this EFS

Now login to ec2 instance-A and run the command

```
#yum install amazon-efs-utils -y
#sudo mount -t efs -o tls fs-0bf3a140410edc9a5:/ /mnt
#df -hT
#cd /mnt
#sudo vi index.html
```

write something and save it

Now login to ec2 instance-B and run the command

```
#yum install amazon-efs-utils -y
#sudo mount -t efs -o tls fs-0bf3a140410edc9a5:/ /mnt
#df -hT
#cd /mnt
#cat /mnt/index.html
```

we can see same file is there with the same content

now open the file form here and write something and check it from ec2 instance-B and we can see the new added text is there

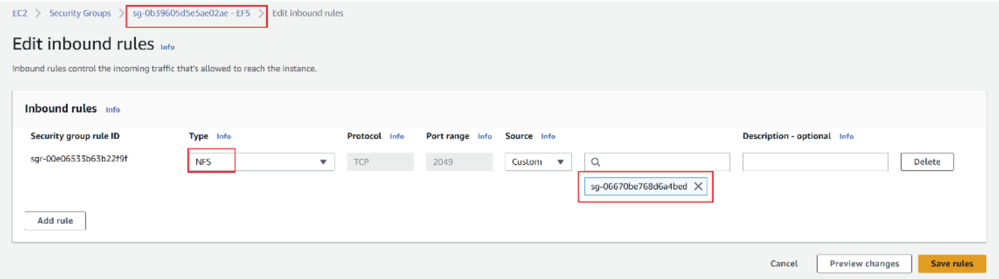
```
#sudo vi index.html
```

write something and save it

If want after restart the efs already mount need to add the following line in /etc/fstab

```
172.31.7.161:/          /var/www/html    nfs      _netdev,defaults        0 0
or
fs-0caa24132718e8ad2:/  /var/www/html    efs      _netdev,defaults        0 0
```

NB: in EFS security group need to permit the NFS for the ec2 instance security group id as show in below


EC2 > Security Groups > sg-0b39605d5e5ae02ae - EFS > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

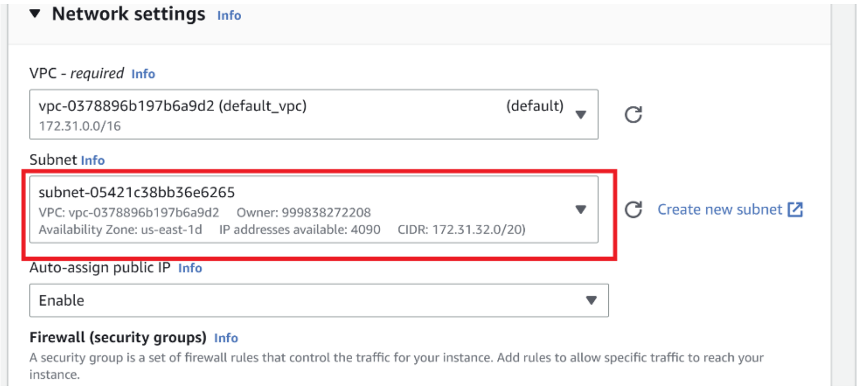
Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info
sgr-00e06533b63b22f9f	NFS	TCP	2049	Custom	sg-06670be768d6a4bed	

Delete
Add rule
Cancel
Preview changes
Save rules

Create an instance with efs when launching

Make sure need to select a subnet from vpc network



▼ Network settings Info

VPC - *required* Info

vpc-0378896b197b6a9d2 (default_vpc) (default)
172.31.0.0/16

Subnet Info

subnet-05421c38bb36e6265
VPC: vpc-0378896b197b6a9d2 Owner: 999838272208
Availability Zone: us-east-1d IP addresses available: 4090 CIDR: 172.31.32.0/20)

Create new subnet

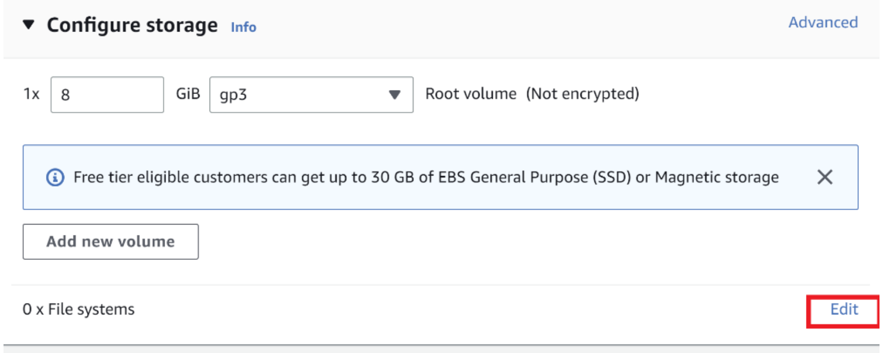
Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

2nd step is



▼ **Configure storage** Info Advanced

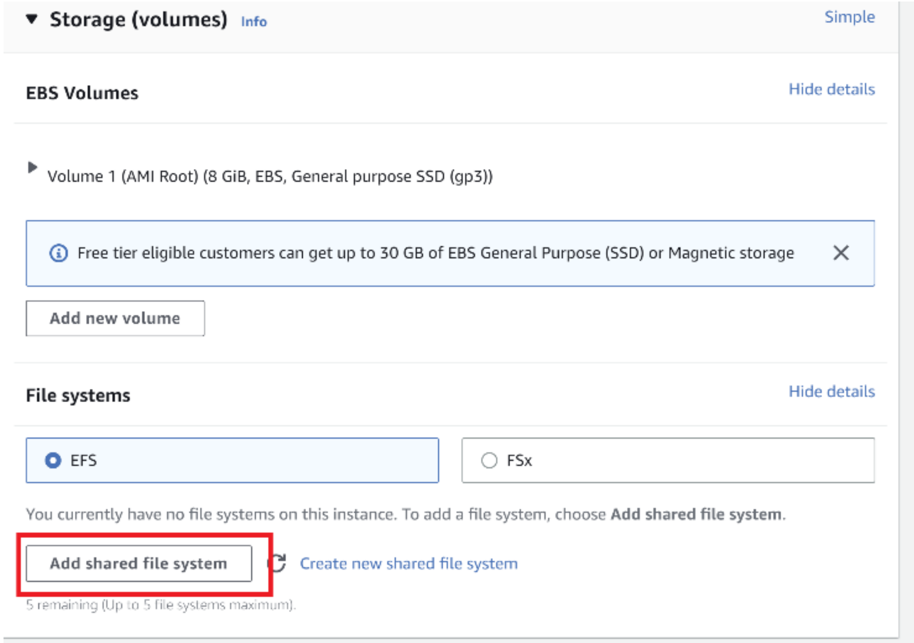
1x 8 GiB gp3 Root volume (Not encrypted)

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

0 x File systems Edit

Then get the following from there need to click the Add shared file system



▼ Storage (volumes) Info Simple

EBS Volumes Hide details

▶ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

Add new volume

File systems Hide details

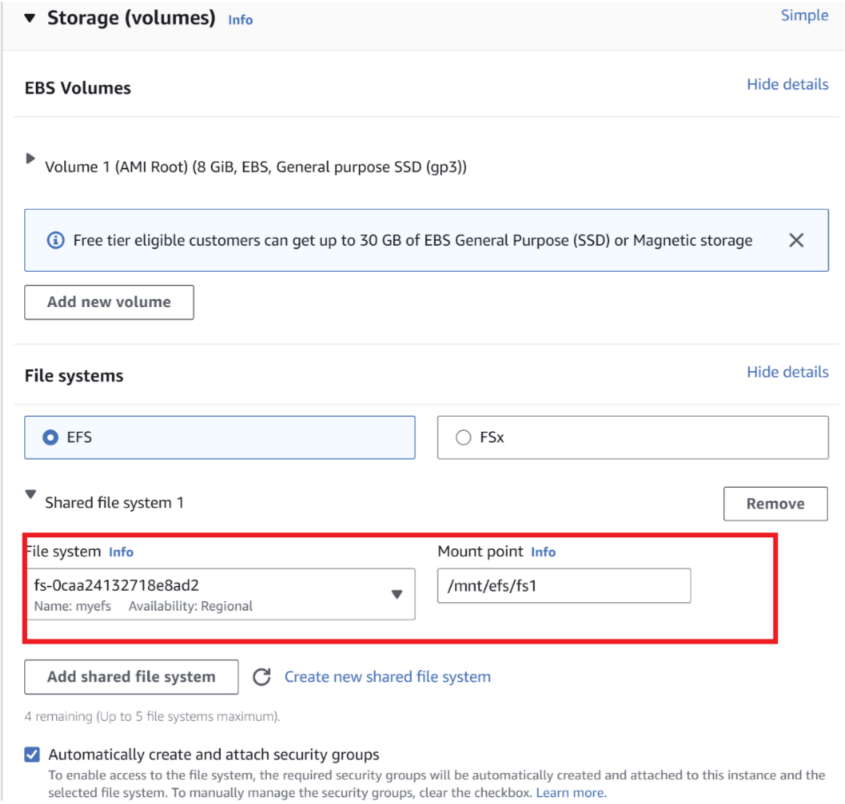
● EFS ○ FSx

You currently have no file systems on this instance. To add a file system, choose **Add shared file system**.

Add shared file system Create new shared file system

5 remaining (Up to 5 file systems maximum).

After clicking te Add shared file system it automatically come show the file seyetm is selected with mount point. So, we need to nothing now just click the launch and check df -hT



▼ Storage (volumes) Info Simple

EBS Volumes Hide details

▶ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

Add new volume

File systems Hide details

● EFS ○ FSx

▼ Shared file system 1 Remove

File system Info: fs-0caa24132718e8ad2 (Name: myefs Availability: Regional)

Mount point Info: /mnt/efs/fs1

Add shared file system Create new shared file system

4 remaining (Up to 5 file systems maximum).

☑ Automatically create and attach security groups

To enable access to the file system, the required security groups will be automatically created and attached to this instance and the selected file system. To manually manage the security groups, clear the checkbox. Learn more.

Now login to the new ec2

```
#ls /mnt/efs/fs1/
#cat /mnt/efs/fs1/a.txt
```

Throughput:

In computer networking, throughput refers to the amount of data that can be transmitted over a network or a communication channel in a given period of time. It is a measure of the network's capacity to carry and deliver data effectively.

Throughput is typically measured in bits per second (bps) or its multiples such as kilobits per second (Kbps), megabits per second (Mbps), or gigabits per second (Gbps). It represents the rate at which data can be successfully transmitted from a source to a destination.

There are several factors that can affect the throughput in a network, including the available bandwidth, network congestion, latency, and packet loss. Higher bandwidth, minimal congestion, low latency, and minimal packet loss generally result in higher throughput.

It's important to note that throughput is different from latency. While throughput focuses on the amount of data that can be transmitted, latency refers to the delay or the time it takes for data to travel from the source to the destination. Both factors are crucial in determining the overall performance and efficiency of a network.

Network File System (NFS): Network File System

NFS stands for, and it is a distributed file system protocol that allows a user on a client computer to access files over a network as if they were located locally. In the context of AWS (Amazon Web Services), AWS provides a managed NFS service called Amazon Elastic File System (EFS).
NFS using 2049 port to communicate with ec2 instances.