EFS and Multiple Access

One of three major storage services offered by AWS, Amazon Elastic File System (EFS) provides us with a fully managed, serverless, scalable, POSIX-compatible file storage service on the cloud. As mentioned in the background, file storage systems follows the same structure and is operated in the same manner as many traditional desktop file systems.

EFS is highly available and durable, making it ideal for workloads that require shared access to file storage across multiple instances. The service is considered moderately priced, generally costing more than S3 but less than EBS (Elastic Block Store). It also has a Infrequent Access (EFS-IA) feature which functions similar to S3 Infrequent Access through its lifecycle policies with regards to IA are much more limited than its S3 counterpart.

Elastic File Systems can be easily mounted onto EC2 instances, AWS Containers, Lambda functions, or on-premise servers and are often done so in order to augment their default storage capabilities.

Finally, EFS offers two different performance and throughput modes for potential adopters:

Performance Modes:

- **General-Purpose** (Most widely used, particularly efficient for latency-sensitive use cases)
- **Max I/O** (For highly parallel applications which necessitate high throughput)

Differences between the two performance modes are discussed below:

Feature/Characteristic	**General Purpose Mode**	**Max I/O Mode**
Use Case	Ideal for latency-sensitive applications (e.g., web serving, content management)	Ideal for applications with highly parallelized workloads (e.g., big data, media processing)
Latency	Low latency	Higher latency compared to General Purpose
Throughput	Up to 7 GB/s with Bursting Throughput	Higher throughput (up to 12 GB/s) with Bursting Throughput
IOPS	Up to 500,000 IOPS	More than 500,000 IOPS
Parallelism	Suitable for moderate to high parallelism	Designed for very high parallelism
Performance Consistency	Consistent performance with low latency	Scales with load; latency can increase as load increases
Cost	Generally lower cost	May incur higher cost due to increased resource usage
Best For	File-based workloads that require consistent low latency	Applications that require high aggregate throughput and can tolerate higher latency

Throughput Modes:

- **Bursting Throughput (**Automatically scales the throughput based on file system size)
- **Provisioned Throughput (**Allows us to allocate throughput without being dependent on the file system size)

Differences between the two throughput modes are as follows:

Feature/Characteristic	**Bursting Throughput Mode**	**Provisioned Throughput Mode**
Use Case	Suitable for most workloads with variable and unpredictable throughput needs	Ideal for applications requiring consistent, high levels of throughput, regardless of file system size
Throughput Allocation	Throughput scales with file system size, allowing bursts based on the amount of stored data	Throughput is provisioned independently of the file system size, ensuring a fixed throughput level
Throughput Range	Scales automatically with storage, up to 100 MB/s per TB (can burst up to 100 MB/s for file systems under 1 TB)	Customizable from 1 MB/s to 1024 MB/s, regardless of file system size
Bursting Capability	Supports bursting when workload needs exceed baseline throughput (depending on size and burst credits)	No bursting; throughput remains constant as provisioned
Performance Consistency	Variable throughput, can burst when needed but may be lower during sustained periods of heavy use	Consistent throughput, ideal for sustained high-performance needs
Cost	Cost-effective for variable workloads, as throughput is tied to file system size	May incur higher costs due to the fixed provisioned throughput, independent of storage usage
Best For	General-purpose workloads, including web serving, backups, and dev/test environments	High-performance workloads, like media processing, big data analytics, and machine learning