AWS EFS

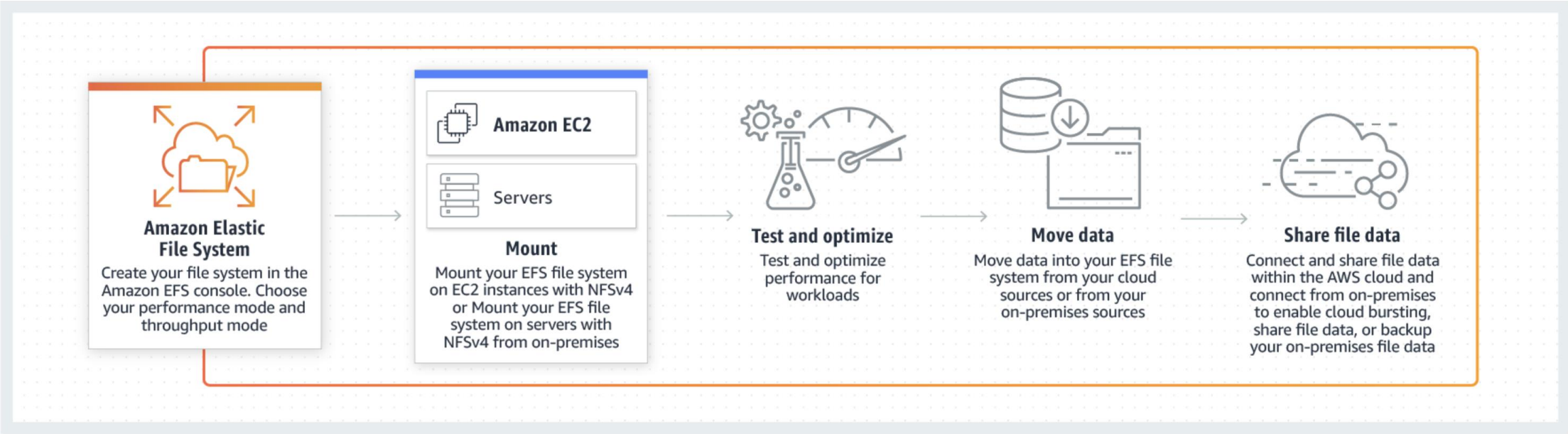
What is AWS EFS

- Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources.
- It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth.

AWS EFS

- Amazon EFS offers two storage classes: the Standard storage class, and the Infrequent Access storage class (EFS IA). EFS IA provides price/performance that's cost-optimized for files not accessed every day. By simply enabling EFS Lifecycle Management on your file system, files not accessed according to the lifecycle policy you choose will be automatically and transparently moved into EFS IA.
- Amazon EFS is a regional service storing data within and across multiple Availability Zones (AZs) for high availability and durability. Amazon EC2 instances can access your file system across AZs, regions, and VPCs, while on-premises servers can access using AWS Direct Connect or AWS VPN.

AWS EFS – How it works?



AWS EFS – Use Cases

- Data scientists can use EFS to create personalized environments, with home directories storing notebook files, training data, and model artefacts. Amazon SageMaker integrates with EFS for training jobs, allowing data scientists to iterate quickly.
- Amazon EFS provides a durable, high throughput file system for content management systems and web serving applications that store and serve information for a range of applications like websites, online publications, and archives.
- Media workflows like video editing, studio production, broadcast processing, sound design, and rendering often depend on shared storage to manipulate large files.

AWS EFS – Use Cases

- Amazon EFS is ideal for container storage providing persistent shared access to a common file repository. Containers are an integral part of building microservices because they're quick to provision, easily portable, and provide process isolation. Containers that need access to the original data each time they start, can take advantage of a shared file system that they can connect to regardless of which instance they're running on.