What is Amazon Elastic File System?

Amazon Elastic File System (Amazon EFS) provides serverless, fully elastic file storage so that you can share file data without provisioning or managing storage capacity and performance. Amazon EFS is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files. Because Amazon EFS has a simple web services interface, you can create and configure file systems quickly and easily. The service manages all the file storage infrastructure for you, meaning that you can avoid the complexity of deploying, patching, and maintaining complex file system configurations.

Amazon EFS supports the Network File System version 4 (NFSv4.1 and NFSv4.0) protocol, so the applications and tools that you use today work seamlessly with AmazonEFS. Amazon EFS is accessible across most types of Amazon Web Services compute instances, including Amazon EC2, Amazon ECS, Amazon EKS, AWS Lambda, and AWS Fargate.

The service is designed to be highly scalable, highly available, and highly durable. Amazon EFS offers the following file system types to meet your availability and durability needs:

- *Regional* (Recommended) – Regional file systems (recommended) store data redundantly across multiple geographically separated Availability Zones within the same AWS Region. Storing data across multiple Availability Zones provides continuous availability to the data, even when one or more Availability Zones in an AWS Region are unavailable.
- *One Zone* – One Zone file systems store data within a single Availability Zone. Storing data in a single Availability Zone provides continuous availability to the data.In the unlikely case of the loss or damage to all or part of the Availability Zone, however, data that is stored in these types of file systems might be lost.

For more information about file system types, see EFS file system types.

Amazon EFS provides the throughput, IOPS, and low latency needed for a broad range of workloads. EFS file systems can grow to petabyte scale, drive high levels of throughput, and allow massively parallel access from compute instances to your data. For most workloads, we recommend using the default modes, which are the General Purpose performance mode and the Elastic throughput modes.

- *General Purpose* – The General Purpose performance mode is ideal for latency-sensitive applications, like web-serving environments, content-management systems, home directories, and general file serving.
- *Elastic* – The Elastic throughput mode is designed to automatically scale throughput performance up or down to meet the needs of your workload activity.

For more information about EFS performance and throughput modes, see Amazon EFS performance.

Amazon EFS provides file-system-access semantics, such as strong data consistency and file locking. For more information, see Data consistency in Amazon EFS. Amazon EFS also supports controlling access to your file systems through Portable Operating System Interface (POSIX) permissions. For more information, see Securing your data in Amazon EFS.

Amazon EFS supports authentication, authorization, and encryption capabilities to help you meet your security and compliance requirements. Amazon EFS supports two forms of encryption for file systems: encryption in transit and encryption at rest. You can enable encryption at rest when creating an Amazon EFS file system. If you do, all of your data and metadata is encrypted. You can enable encryption in transit when you mount the file system. NFS client access to EFS is controlled by both AWS Identity and Access Management (IAM) policies and network security policies, such as security groups. For more information, see Encrypting data in Amazon EFS, Identity and access management for Amazon EFS, and Controlling network access to Amazon EFS file systems for NFS clients.

> **Note**
>
> Using Amazon EFS with Microsoft Windows–based Amazon EC2 instances is not supported.

How Amazon EFS works

Following, you can find a description about how Amazon EFS works, its implementation details, and security considerations.

Topics

- How Amazon EFS works with AWS Direct Connect and AWS Managed VPN
- How Amazon EFS works with AWS Backup
- Implementation summary
- Authentication and access control
- Data consistency in Amazon EFS
- EFS storage classes
- Replication

Overview

Amazon Elastic File System (EFS) provides a simple, serverless, set-and-forget elastic file system. With Amazon EFS, you can create a file system, mount the file system on an Amazon EC2 instance, and then read and write data to and from your file system. You can mount an Amazon EFS file system in your virtual private cloud (VPC), through the Network File System versions 4.0 and 4.1 (NFSv4) protocol. We recommend using a current generation Linux NFSv4.1 client, such as those found in the latest Amazon Linux, Amazon Linux 2, Red Hat, Ubuntu, and macOS Big Sur AMIs, in conjunction with the EFS mount helper. For instructions, see Installing Amazon EFS tools.

For a list of Amazon EC2 Linux and macOS Amazon Machine Images (AMIs) that support this protocol, see NFS support. For some AMIs, you must install an NFS client to mount your file system on your Amazon EC2 instance. For instructions, see Installing the NFS client.

You can access your Amazon EFS file system concurrently from multiple NFS clients, so applications that scale beyond a single connection can access a file system. Amazon EC2 and other AWS compute instances running in multiple Availability Zones within the same AWS Region can access the file system, so that many users can access and share a common data source.

For a list of AWS Regions where you can create an AmazonEFS file system, see the Amazon Web Services General Reference.

To access your Amazon EFS file system in a VPC, you create one or more mount targets in the VPC.

- For Regional file systems, you can create a mount target in each Availability Zone in the AWS Region.
- For One Zone file systems, you create only a single mount target that is in the same Availability Zone as the file system.

For more information, see EFS storage classes.

A *mount target* provides an IP address for an NFSv4 endpoint at which you can mount an Amazon EFS file system. You mount your file system using its Domain Name Service (DNS) name, which resolves to the IP address of the EFS mount target in the same Availability Zone as your EC2 instance. You can create one mount target in each Availability Zone in an AWS Region. If there are multiple subnets in an Availability Zone in your VPC, you create a mount target in one of the subnets. Then all EC2 instances in that Availability Zone share that mount target.

> **Note**
>
> An Amazon EFS file system can have mount targets in only one VPC at a time.

Mount targets themselves are designed to be highly available. As you design for high availability and failover to other Availability Zones, keep in mind that while the IP addresses and DNS for your mount targets in each Availability Zone are static, they are redundant components backed by multiple resources.

After mounting the file system by using its DNS name, you use it like any other POSIX-compliant file system. For information about NFS-level permissions and related considerations, see Working with users, groups, and permissions at the Network File System (NFS) level.

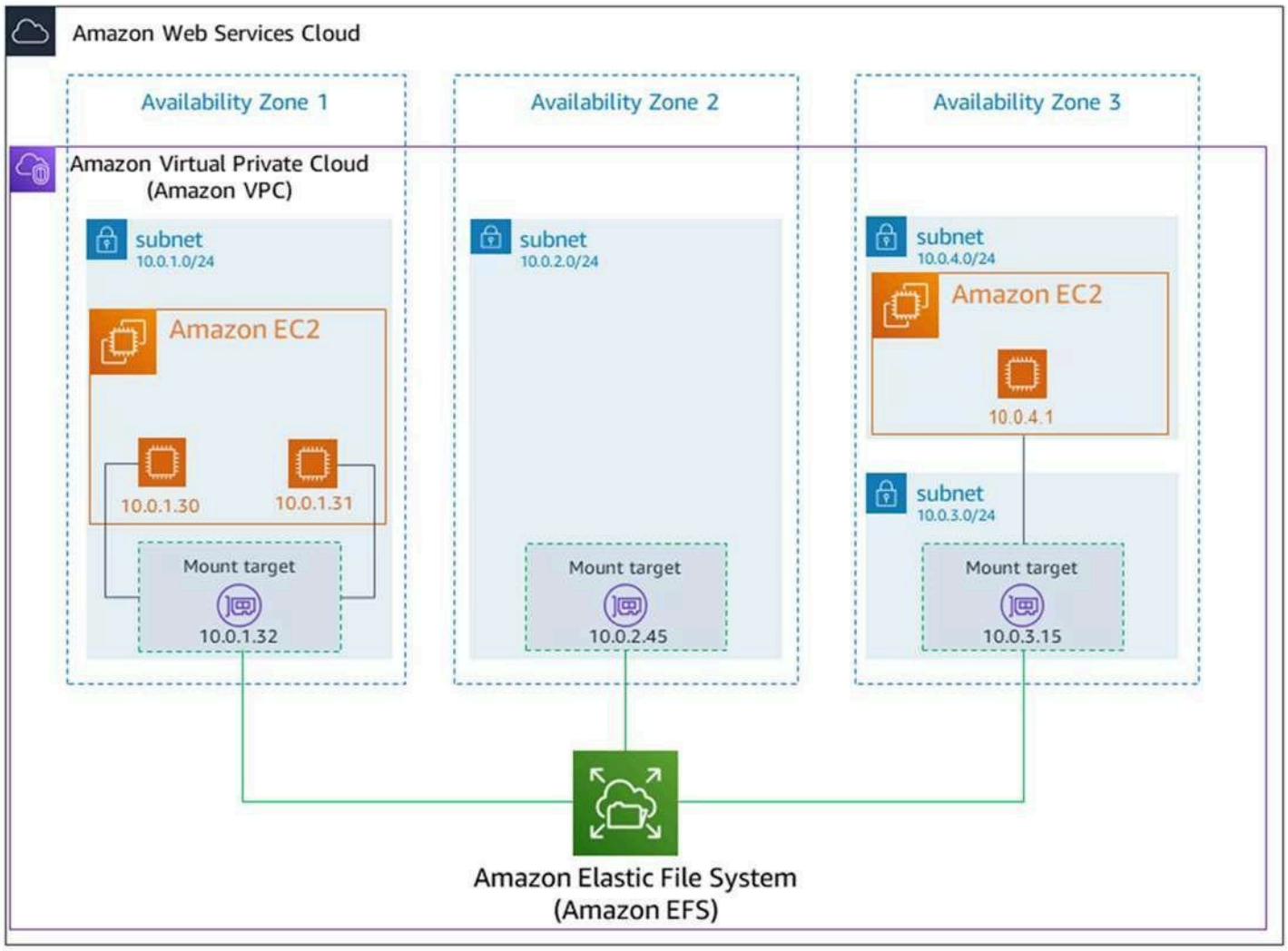
You can mount your Amazon EFS file systems on your on-premises data center servers when connected to your Amazon VPC with AWS Direct Connect or AWS VPN. You can mount your EFS file systems on on-premises servers to migrate datasets to EFS, enable cloud bursting scenarios, or back up your on-premises data to Amazon EFS.

How Amazon EFS works with Amazon EC2

This section explains how Amazon EFS Regional and One Zone file systems are mounted to EC2 instances in an Amazon VPC.

Regional EFS file systems

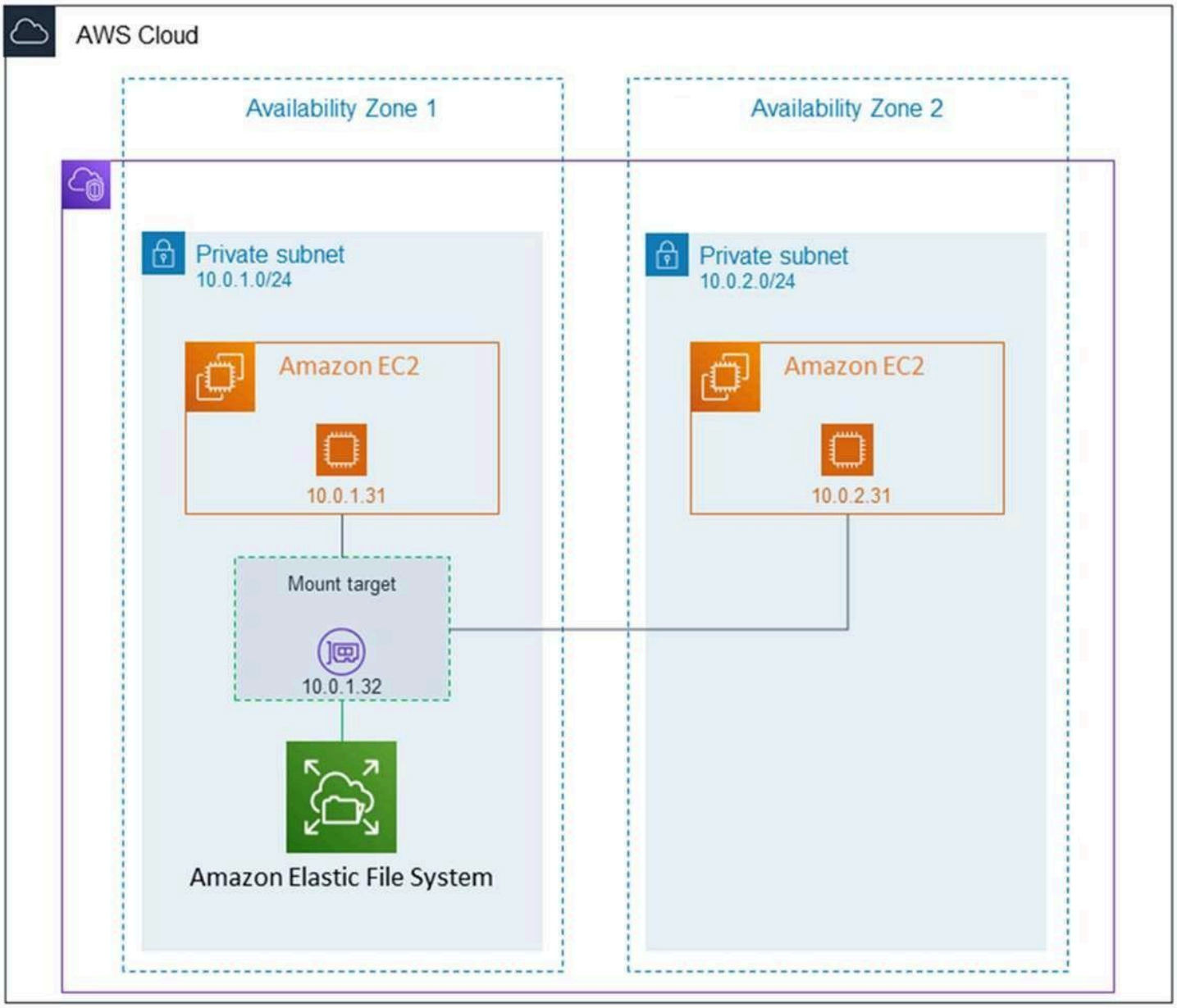
The following illustration shows multiple EC2 instances accessing an Amazon EFS file system that is configured for multiple Availability Zones in an AWS Region.



In this illustration, the virtual private cloud (VPC) has three Availability Zones. Because the file system is Regional, a mount target was created in each Availability Zone. We recommend that you access the file system from a mount target within the same Availability Zone for performance and cost reasons. One of the Availability Zones has two subnets. However, a mount target is created in only one of the subnets. For more information, see Mounting EFS file systems using the EFS mount helper.

One Zone EFS file systems

The following illustration shows multiple EC2 instances accessing a One Zone file system from different Availability Zones in a single AWS Region.



In this illustration, the VPC has two Availability Zones, each with one subnet. Because the file system type is One Zone, it can only have a single mount target. For better performance and cost, we recommend that you access the file system from a mount target in the same Availability Zone as the EC2 instance that you're mounting it on.

In this example, the EC2 instance in the us-west-2c Availability Zone will pay EC2 data access charges for accessing a mount target in a different Availability Zone. For more information, see Mounting One Zone file systems.

How Amazon EFS works with AWS Direct Connect and AWS Managed VPN

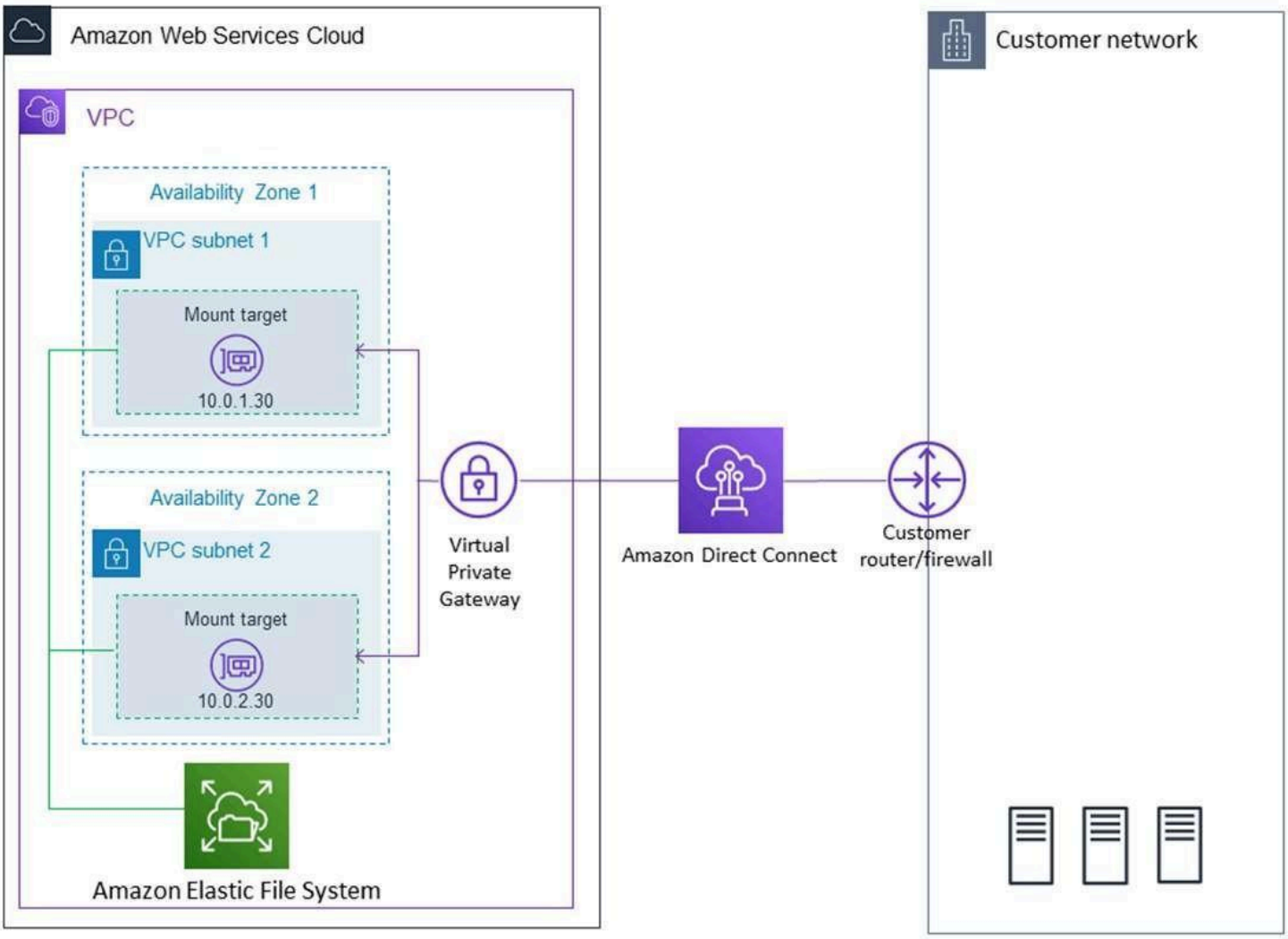
By using an Amazon EFS file system mounted on an on-premises server, you can migrate on-premises data into the AWS Cloud hosted in an Amazon EFS file system. You can also take advantage of bursting. In other words, you can move data from your on-premises servers into Amazon EFS and analyze it on a fleet of Amazon EC2 instances in your Amazon VPC. You can then store the results permanently in your file system or move the results back to your on-premises server.

Keep the following considerations in mind when using Amazon EFS with an on-premises server:

- Your on-premises server must have a Linux-based operating system. We recommend Linux kernel version 4.0 or later.
- For the sake of simplicity, we recommend mounting an Amazon EFS file system on an on-premises server using a mount target IP address instead of a DNS name.

There is no additional cost for on-premises access to your Amazon EFS file systems. You are charged for the AWS Direct Connect connection to your Amazon VPC. For more information, see AWS Direct Connect pricing.

The following illustration shows an example of how to access an Amazon EFS file system from on-premises (the on-premises servers have the file systems mounted).



You can use any mount target in your VPC if you can reach that mount target's subnet by using an AWS Direct Connect connection between your on-premises server and VPC. To access Amazon EFS from an on-premises server, add a rule to your mount target security group to allow inbound traffic to the NFS port (2049) from your on-premises server. For more information, including detailed procedures, see Prerequisites.

How Amazon EFS works with AWS Backup

For a comprehensive backup implementation for your file systems, you can use Amazon EFS with AWS Backup. AWS Backup is a fully managed backup service that makes it easy to centralize and automate data backup across AWS services in the cloud and on-premises. Using AWS Backup, you can centrally configure backup policies and monitor backup activity for your AWS resources. Amazon EFS always prioritizes file system operations over backup operations. To learn more about backing up EFS file systems using AWS Backup, see Backing up EFS file systems.

Implementation summary

In Amazon EFS, a file system is the primary resource. Each file system has properties such as ID, creation token, creation time, file system size in bytes, number of mount targets created for the file system, and the file system lifecycle state. For more information, see CreateFileSystem.

Amazon EFS also supports other resources to configure the primary resource. These include mount targets and access points:

- **Mount target** – To access your file system, you must create mount targets in your VPC. Each mount target has the following properties: the mount target ID, the subnet ID in which it is created, the file system ID for which it is created, an IP address at which the file system may be mounted, VPC security groups, and the mount target state. You can use the IP address or the DNS name in your `mount` command.

  Each file system has a DNS name of the following form.

  ```
  file-system-id.efs.aws-region.amazonaws.com
  ```

  You can specify this DNS name in your `mount` command to mount the Amazon EFS file system. Suppose you create an `efs-mount-point` subdirectory off of your home directory on your EC2 instance or on-premises server. Then, you can use the mount command to mount the file system. For example, on an Amazon Linux AMI, you can use the following `mount` command.

  ```
  $ sudo mount -t nfs -o
   nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport file-
  system-DNS-name:/ ~/efs-mount-point
  ```

  For more information, see Managing mount targets.

- **Access Points** – An access point applies an operating system user, group, and file system path to any file system request made using the access point. The access point's operating system user and group override any identity information provided by the NFS client.The file system path is exposed to the client as the access point's root directory.This ensures that each application always uses the correct operating system identity and the correct directory when accessing shared file-based datasets. Applications using the access point can only access data in its own directory and below. For more information, see Working with Amazon EFS access points.

Mount targets and tags are *subresources* that are associated with a file system. You can only create them within the context of an existing file system.

Amazon EFS provides API operations for you to create and manage these resources. In addition to the create and delete operations for each resource, Amazon EFS supports a describe operation that enables you to retrieve resource information. You have the following options for creating and managing these resources:

- Use the Amazon EFS console– For an example, see Getting started.
- Use the Amazon EFS command line interface (CLI) – For an example, see Tutorial: Create an EFS file system and mount it on an EC2 instance using the AWS CLI.
- You can also manage these resources programmatically as follows:
  - Use the AWS SDKs – The AWS SDKs simplify your programming tasks by wrapping the underlying Amazon EFS API. The SDK clients also authenticate your requests by using access keys that you provide. For more information, see Sample Code and Libraries.
  - Call the Amazon EFS API directly from your application – If you cannot use the SDKs for some reason, you can make the Amazon EFS API calls directly from your application. However, you need to write the necessary code to authenticate your requests if you use this option. For more information about the Amazon EFS API, see Amazon EFS API.

Authentication and access control

You must have valid credentials to make Amazon EFS API requests, such as create a file system. In addition, you must also have permissions to create or access resources.

Users and roles that you create in AWS Identity and Access Management (IAM) must be granted permissions to create or access resources. For more information about permissions, see Identity and access management for Amazon EFS.

IAM authorization for NFS clients is an additional security option for Amazon EFS that uses IAM to simplify access management for Network File System (NFS) clients at scale. With IAM authorization for NFS clients, you can use IAM to manage access to an EFS file system in an inherently scalable way. IAM authorization for NFS clients is also optimized for cloud environments. For more information on using IAM authorization for NFS clients, see Using IAM to control file system data access.

Data consistency in Amazon EFS

Amazon EFS provides the close-to-open consistency semantics that applications expect from NFS.

In Amazon EFS, write operations for Regional file systems are durably stored across Availability Zones in these situations:

- An application performs a synchronous write operation (for example, using the open Linux command with the O_DIRECT flag, or the fsync Linux command).
- An application closes a file.

Depending on the access pattern, Amazon EFS can provide stronger consistency guarantees than close-to-open semantics. Applications that perform synchronous data access and perform non-appending writes have read-after-write consistency for data access.

File locking

NFS client applications can use NFS version 4 file locking (including byte-range locking) for read and write operations on Amazon EFS files.

Remember the following about how Amazon EFS locks files:

- Amazon EFS only supports advisory locking and read/write operations don't check for conflicting locks before executing. For example, to avoid file synchronization issues with atomic operations, your application must be aware of NFS semantics (such as close-to-open consistency).
- Any one particular file can have up to 512 locks across all instances connected and users accessing the file.

EFS storage classes

Amazon EFS provides different storage classes for different data storage needs. Standard is the first storage class to which data is written and is the storage class for data that is accessed frequently. For less frequently accessed files, Amazon EFS offers the EFS Infrequent Access (IA) and EFS Archive storage classes. The IA storage class is cost-optimized for data that is accessed a few times each quarter and the Archive storage class is cost-optimized for data that is accessed only a few times each year or less. For more information about Amazon EFS storage classes, see EFS storage classes.

Lifecycle management

To manage your file systems so that they are stored cost effectively throughout their lifecycle, use lifecycle management. Lifecycle management automatically transitions data between storage classes according to the lifecycle configuration defined for the file system. The lifecycle configuration is a set of *lifecycle policies* that define when to transition the file system data to another storage class. For more information, see Managing file system storage.

Replication

You can create a replica of your Amazon EFS file system in the AWS Region of your preference using replication. Replication automatically and transparently replicates the data and metadata on your EFS file system to a new destination EFS file system that is created in an AWS Region that you choose. EFS automatically keeps the source and destination file systems synchronized. Replication is continual and designed to provide a recovery point objective (RPO) and a recovery time objective (RTO) of minutes. These features assist you in meeting your compliance and business continuity goals. For more information, see Replicating EFS file systems.

Are you a first-time user of Amazon EFS?

If you are a first-time user of Amazon EFS, we recommend that you read the following sections in order:

1. For an Amazon EFS product and pricing overview, see Amazon EFS.
2. For an Amazon EFS technical overview, see How Amazon EFS works.
3. Try the Getting started exercise.

If you want to learn more about Amazon EFS, the following topics discuss the service in greater detail:

- Working with Amazon EFS resources
- Managing EFS file systems
- Amazon EFS API