

# Amazon EFS

- Presented by

Piyusha Supe (23CO315), Iffa Shaikh (23CO314), Shruti Patil (22CO088), Dolly Singh (22CO116)

Under the guidance of  
Prof. Vrushali Kanavde

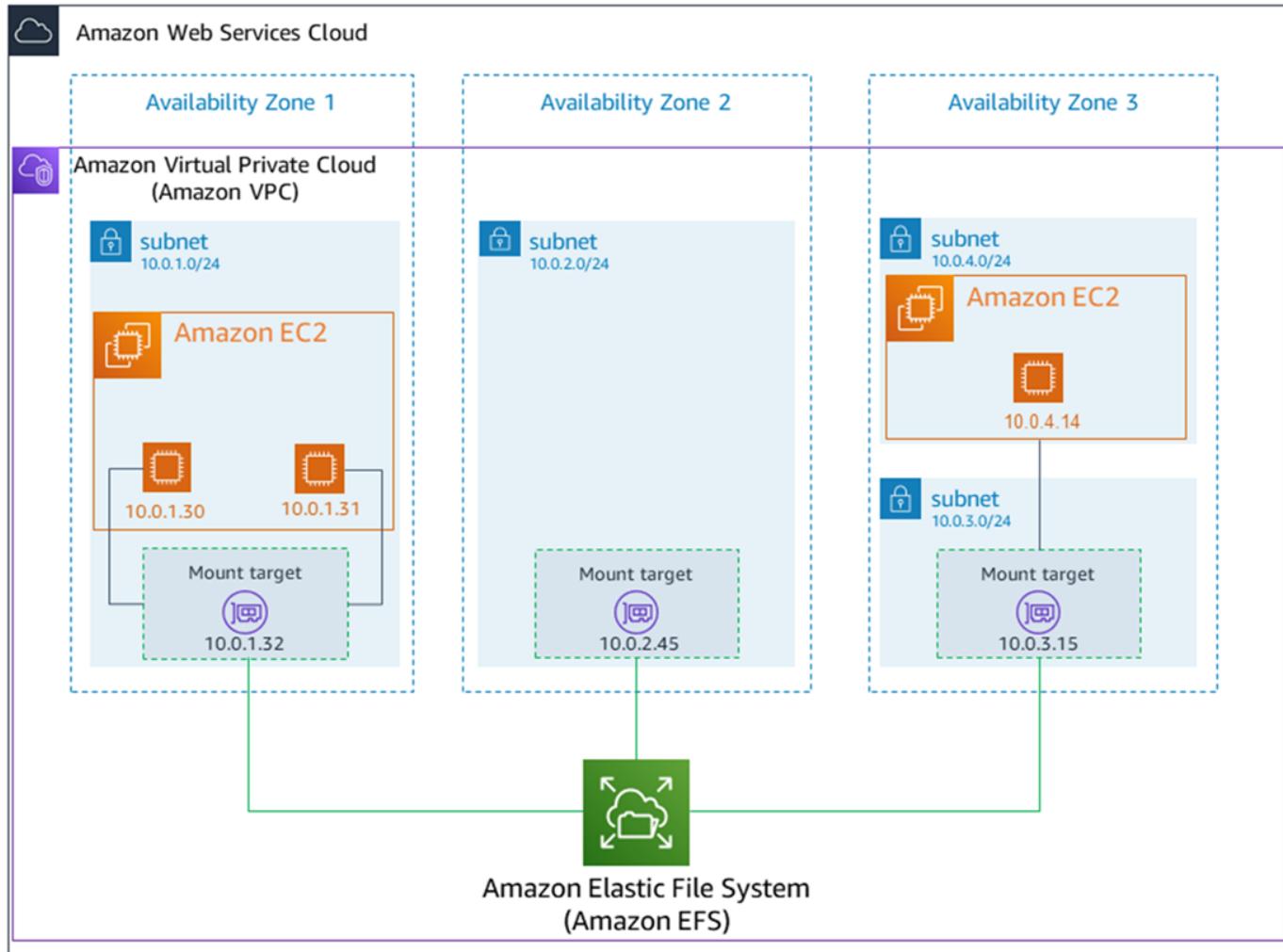
# What is Amazon EFS?

- ▶ **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 Amazon EFS. 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

# 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 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 Amazon EFS is controlled by both AWS Identity and Access Management (IAM) policies and network security policies, such as security groups

# Block diagram



# Features of AWS EFS

- ▶ **Fully managed** – No servers to manage or storage to provision.
- ▶ **Elastic** – Automatically scales up or down depending on how much data you store.
- ▶ **Shared access** – Multiple EC2 instances can read/write to the same files at the same time.
- ▶ **Accessible via NFS** – Uses the standard Network File System (NFS v4/v4.1) protocol.
- ▶ **High availability & durability** – Data is stored across multiple Availability Zones (AZs).

# Mounting the File Systems

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

# Using mount target

- ▶ 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`

# Using access points

- ▶ **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
- ▶ 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.

# States of the file system

File system state	Description
AVAILABLE	The file system is in a healthy state, and is reachable and available for use.
CREATING	Amazon EFS is in the process of creating the new file system.
DELETING	Amazon EFS is deleting the file system in response to a user-initiated delete request. For more information, see <a href="#">Deleting EFS file systems</a> .
DELETED	Amazon EFS has deleted the file system in response to a user-initiated delete request. For more information, see <a href="#">Deleting EFS file systems</a> .
UPDATING	The file system is undergoing an update in response to a user-initiated update request.
ERROR	Applicable for One Zone file systems, including file systems in a replication configuration. The file system is in a failed state and is unrecoverable. To access the file system data, restore a backup of this file system to a new file system. For more information, see <a href="#">Protecting your data in Amazon EFS</a>

# Managing the storage life cycle

- ▶ When you create an EFS file system that has the recommended settings using the AWS Management Console, the file system is automatically configured with the following default lifecycle configuration:
- ▶ **Transition into IA(Infrequent access)** is set to **30 days since last access**.
- ▶ **Transition into Archive** is set to **90 days since last access**.
- ▶ **Transition into Standard** is set to **None**.
- ▶ You can change the default lifecycle policies when creating a file system with customized settings using the AWS Management Console or when creating a file system using the AWS CLI. Alternately, you can change the policies after the file system is created, as described in the following procedures

# Storage Classes Breakdown:

## Standard

- ▶ Default, high-performance, and highest cost.
- ▶ Files you access frequently.

## Infrequent Access (IA)

- ▶ Cheaper than Standard.
- ▶ For files not accessed often (e.g., logs, backups).
- ▶ Slightly higher latency and retrieval cost.

## Archive

- ▶ Lowest-cost storage.
- ▶ For files rarely accessed.
- ▶ Takes longer to retrieve data (hours potentially).

# Automated Monitoring tools

- ▶ You can use the following automated monitoring tools to watch Amazon EFS and report when something is wrong:
- ▶ **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Amazon EC2 Auto Scaling policy
- ▶ **Amazon CloudWatch Logs** – Monitor, store, and access your log files from AWS CloudTrail or other sources.
- ▶ **Amazon CloudWatch Events** – Match events and route them to one or more target functions or streams to make changes, capture state information, and take corrective action..
- ▶ **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail

# Manual Monitoring

- ▶ From the Amazon EFS console, you can find the following items for your file systems:
  - ▶ The current metered size
  - ▶ The number of mount targets
  - ▶ The lifecycle state
- ▶ CloudWatch home page shows:
  - ▶ Current alarms and status
  - ▶ Graphs of alarms and resources
  - ▶ Service health status
- ▶ In addition, you can use CloudWatch to do the following:
  - ▶ Create [customized dashboards](#) to monitor the services that you use.
  - ▶ Graph metric data to troubleshoot issues and discover trends.
  - ▶ Search and browse all your AWS resource metrics.
  - ▶ Create and edit alarms to be notified of problems.

# Thank you!

References:

<https://docs.aws.amazon.com/efs/>