## Amazon Elastic File System (EFS):

**Amazon Elastic File System (EFS)** is a fully managed, scalable, and elastic file storage service provided by AWS. It is designed to be used with AWS cloud services and on-premises resources, offering a simple and cost-effective solution for file storage that can be easily accessed and shared across multiple Amazon EC2 instances.

## Key Features of Amazon EFS:

1. **Scalability**:
    - EFS automatically scales your file system as you add or remove files, accommodating petabytes of data without requiring management of storage capacity.
    - You pay only for the storage you use, and there are no minimum fees or setup costs.
2. **Elasticity**:
    - EFS dynamically grows and shrinks as files are added and deleted, without needing to provision storage in advance.
3. **High Availability and Durability**:
    - EFS is designed for 99.999999999% (11 9's) durability by storing data redundantly across multiple Availability Zones (AZs) within an AWS region.
    - It also provides a high level of availability, ensuring that your data is accessible even if an entire AZ goes down.
4. **Performance Modes**:
    - **General Purpose Mode**: Suitable for latency-sensitive use cases, like web serving and content management systems.
    - **Max I/O Mode**: Designed for applications that require higher levels of aggregate throughput and can tolerate slightly higher latencies, such as big data processing.
5. **Throughput Modes**:
    - **Bursting Throughput**: Suitable for most applications, allowing throughput to scale with the size of the file system.
    - **Provisioned Throughput**: Allows you to specify a fixed throughput level independent of the file system's size, useful for applications with high and consistent throughput requirements.
6. **Security**:
    - **Encryption at Rest**: EFS data can be encrypted at rest using AWS Key Management Service (KMS).

    - **Encryption in Transit**: Data can also be encrypted in transit using industry-standard Transport Layer Security (TLS).
    - **Access Control**: You can control access to your EFS file systems using AWS Identity and Access Management (IAM) policies, security groups, and Network File System (NFS) permissions.
7. **Cost-Effectiveness**:
    - EFS offers two storage classes: **Standard** (for frequently accessed data) and **Infrequent Access (IA)** (for data that is not accessed often), which helps reduce storage costs.
8. **Integration with AWS Services**:
    - EFS integrates seamlessly with other AWS services, such as EC2, ECS, EKS, AWS Lambda, and AWS Backup.
    - It also supports on-premises access through AWS Direct Connect or AWS VPN.
9. **Use Cases**:
    - **Web Serving and Content Management**: EFS can store and serve static and dynamic content for websites and web applications.
    - **Big Data and Analytics**: It provides scalable storage for big data applications that require high throughput.
    - **Backup and Restore**: EFS can be used as a target for backup and restore operations, supporting data resilience strategies.
    - **Development and Test Environments**: EFS allows multiple EC2 instances to share a common file system, making it useful for development, testing, and staging environments.

## How to Use Amazon EFS:

1. **Create an EFS File System**:
    - In the AWS Management Console, navigate to **Amazon EFS** and create a new file system.
    - Choose the appropriate VPC and configure the network settings.
    - Select the desired performance and throughput modes.
2. **Mount EFS to EC2 Instances**:
    - After creating the file system, you can mount it to your EC2 instances using the EFS mount helper or standard NFS client.
    - EFS can be mounted to multiple EC2 instances simultaneously, allowing data to be shared across instances.
3. **Manage and Monitor**:
    - Use the EFS console, CloudWatch, and AWS CloudTrail to monitor performance, access logs, and manage your file system.

Amazon EFS is a versatile solution for applications that require shared file storage in the cloud, offering the flexibility and scalability needed for a wide range of workloads.

## Setups of AWS EFS

Setting up Amazon Elastic File System (EFS) involves creating the EFS file system, configuring network settings, and mounting the file system to your EC2 instances. Here's a step-by-step guide:

### Step 1: Create an Amazon EFS File System

1. **Sign in to AWS Management Console**:
    - Go to the AWS Management Console and sign in.
2. **Navigate to the Amazon EFS Service**:
    - In the AWS Management Console, search for "EFS" or go to **Services** > **Storage** > **Elastic File System**.
3. **Create a File System**:
    - Click **Create file system**.
    - Choose the VPC where your EC2 instances are running.
    - Select the subnets and security groups for network access. Ensure the security groups allow NFS traffic (port 2049).
4. **Configure File System Settings**:
    - Choose **Availability and Durability** options:
        - **Regional**: Default option for high availability across multiple AZs.
        - **One Zone**: Lower cost, but limited to a single AZ.
    - Choose **Performance Mode**:
        - **General Purpose**: Suitable for most workloads.
        - **Max I/O**: For high throughput, latency-tolerant applications.
    - Choose **Throughput Mode**:
        - **Bursting**: Scales with file system size.
        - **Provisioned**: Specify a fixed throughput level.
    - **Lifecycle Management**: Optionally enable this to move infrequently accessed files to the Infrequent Access (IA) storage class.
    - **Encryption**: Enable encryption at rest if needed.
5. **Create and Review**:
    - Review the settings and click **Create**. The file system will be created and available in a few minutes.

### Step 2: Mount EFS to EC2 Instances

1. **Prepare EC2 Instances**:

    - Ensure that your EC2 instances are in the same VPC and security group allows NFS traffic (port 2049).
    - SSH into your EC2 instance.
2. **Install NFS Client**:
    - Install the NFS client on your EC2 instance if it's not already installed:
        - For Amazon Linux 2 or CentOS:
          ```
          sudo yum install -y nfs-utils
          ```
        - For Ubuntu:
          ```
          sudo apt-get install -y nfs-common
          ```
3. **Get the EFS Mount Command**:
    - Go back to the AWS Management Console > EFS.
    - Select your file system and click **Attach** to get the mount command.
4. **Mount the File System**:
    - Create a directory on your EC2 instance to mount the EFS file system:
      ```
      sudo mkdir /mnt/efs
      ```
    - Use the mount command provided by the AWS console (replace the `fs-xxxxxx` with your file system ID):
      ```
      sudo mount -t nfs4 -o nfsvers=4.1
      fs-xxxxxx.efs.<region>.amazonaws.com:/ /mnt/efs
      ```
5. **Verify the Mount**:
    - Check that the file system is mounted correctly by listing the contents of the mount directory:
      ```
      ls /mnt/efs
      ```
    - If successful, the mount directory should be accessible, and you can start using the EFS.

### Step 3: Automate Mounting on Reboot

To ensure that the EFS file system is automatically mounted after a reboot:

1. **Edit the /etc/fstab file**:
    - Open the `/etc/fstab` file in an editor:
      ```
      sudo nano /etc/fstab
      ```
    - Add the following line to the file (replace `fs-xxxxxx` with your EFS file system ID and adjust the region):
      ```
      fs-xxxxxx.efs.<region>.amazonaws.com:/ /mnt/efs nfs4
      defaults,_netdev 0 0
      ```
    - Save and close the file.
2. **Test the Configuration**:

    - Unmount the file system:
      `sudo umount /mnt/efs`
    - Remount all file systems listed in `/etc/fstab`:
      `sudo mount -a`
    - Verify that the EFS file system is mounted again by checking `/mnt/efs`.

### Step 4: Manage and Monitor EFS

- Use the **Amazon EFS Console** or **AWS CLI** to monitor usage, manage file systems, and configure access permissions.
- Integrate with **AWS CloudWatch** for monitoring and setting up alerts based on your EFS usage.

That's it! Your EFS file system should now be set up and accessible from your EC2 instances.