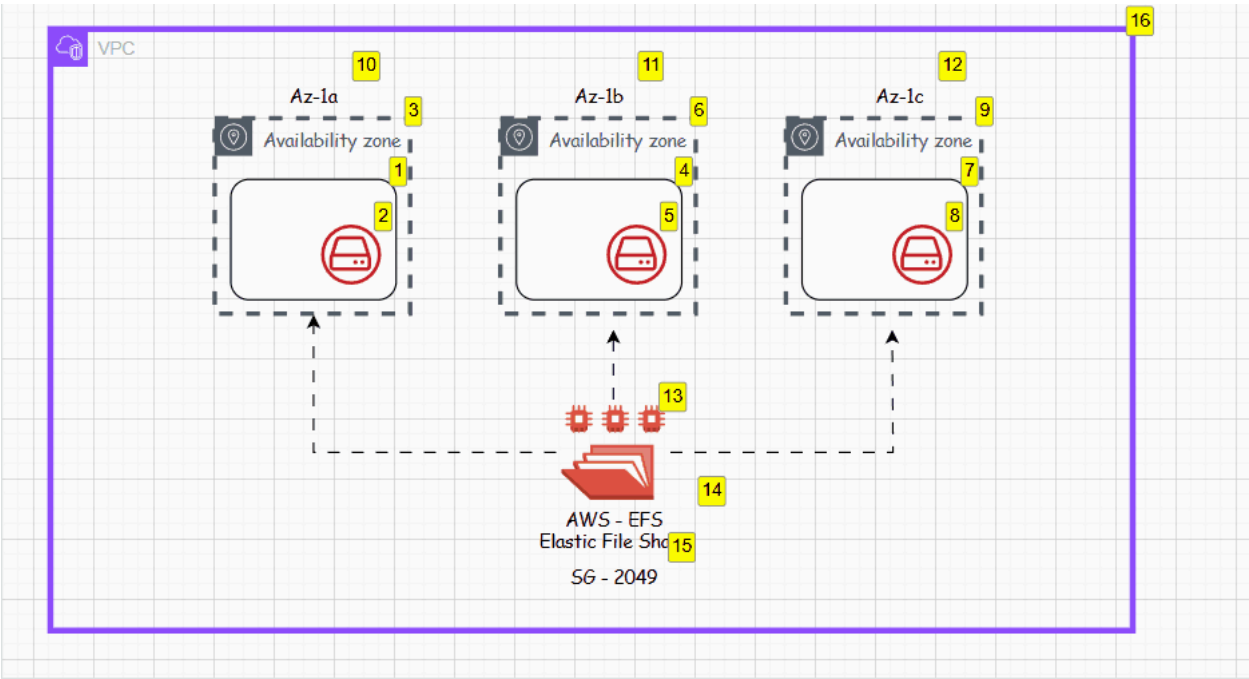

📦 What is AWS EFS (Elastic File System)?

Amazon EFS is a **fully managed, scalable NFS file system** that can be **mounted concurrently by multiple EC2 instances** (and other services) across multiple **Availability Zones (AZs)**. It's ideal for use cases requiring **shared, concurrent access** to data with **POSIX-compliant file systems**.



🔧 Key Features

Feature	Description
Fully managed	No server provisioning or management
Scalable	Grows/shrinks automatically up to petabytes
Multi-AZ	Supports high availability by spanning multiple AZs

POSIX compliant	Standard NFSv4 protocol support
Pay-as-you-go	Pay for the storage used (GB/month)
Integrated with IAM	Supports fine-grained access control via IAM

🔩 EFS Components and Architecture

1. File System

- The top-level resource in EFS.
- Automatically spans multiple AZs in a region for HA.
- Can be mounted from multiple instances concurrently.

---

2. Mount Targets

- Network endpoints in **each AZ** that let you mount the file system via NFS.
- You must create a mount target **per subnet/AZ** where you want to access EFS.
- Associated with a **VPC and security group**.

---

3. Performance Modes

Mode	Description
General Purpose (default)	Good for latency-sensitive workloads (e.g., CMS, home dirs)
Max I/O	For massively parallel workloads (e.g., big data, genomics). Higher latency, but better throughput and scaling

---

4. Throughput Modes

Mode	Description
Bursting (default)	Throughput scales with size of the file system
Provisioned	Set fixed throughput (good for predictable workloads)

---

5. Storage Classes

Class	Description
EFS Standard	For frequently accessed files
EFS Infrequent Access (IA)	92% cheaper for cold data, automatically moved via lifecycle policy

---

6. Access Points

- Acts like **soft mount points** with:
    - Different user/permission contexts
    - Root directories
- Use case: containerized applications needing isolated paths in the same EFS

---

7. Lifecycle Management

- Automatically move files to **IA (Infrequent Access)** based on age (7, 14, 30, 60, or 90 days).

- Saves costs by tiering unused files to cheaper storage.

---

8. Security Features

- **IAM**: Control access at API level
- **Security groups**: Control NFS network access
- **Encryption**:
    - **At rest**: Managed with KMS (enabled by default)
    - **In transit**: Via TLS (when mounted using amazon-efs-utils)

🧰 Integration with AWS Services

Service	Integration
EC2	Mount EFS via NFS
ECS (Fargate or EC2)	Use EFS as persistent storage for containers
Lambda	Supports EFS mount for large libraries or stateful functions
EKS	CSI driver allows mounting EFS as PersistentVolumes
CloudWatch	Monitor EFS metrics (throughput, IOPS, latency)

🚀 Use Cases

Use Case	Description

Shared file storage	Web server clusters (e.g., WordPress) needing shared content
Machine learning	Multiple training nodes sharing input/output datasets
CI/CD	Shared build cache, codebases
Containers	Shared volumes between pods or tasks
Home directories	User-specific directories in managed Linux environments

---

⚠️ Considerations

- **EFS is region-specific**, not cross-region.
- **Latency** is higher than EBS – not ideal for low-latency DB workloads.
- Best for **shared reads/writes** or workloads that need concurrent access.

🛠️ Example: Mounting EFS on EC2

1. Install EFS mount helper:

```
sudo yum install -y amazon-efs-utils
```

2. Mount EFS:

```
sudo mount -t efs -o tls fs-xxxxxxxx:/ /mnt/efs
```

3. Make it persistent (add to /etc/fstab):

fs-xxxxxxxx:/ /mnt/efs efs _netdev,tls 0 0

📊 Monitoring

Use **CloudWatch** to track:

- BurstCreditBalance
- ClientConnections
- PercentIOLimit
- TotalIOBytes

Would you like:

- An **EFS Terraform/CloudFormation module** example?
- Or a **cost optimization cheat sheet** comparing EBS, EFS, and S3?

Let me know!