

Here is a carefully curated list of **production-level AWS EFS (Elastic File System) interview questions and answers**, suitable for roles like **DevOps Engineer**, **Cloud Engineer**, and **Site Reliability Engineer**. These questions reflect real-world, high-availability, performance, and cost concerns seen in production environments.

---

## AWS EFS – Production Interview Questions & Answers

---

### ♦ Q1: What is AWS EFS and when would you use it in production?

**Answer:**

AWS EFS is a **scalable, elastic, shared NFS-based file system** for Linux workloads. In production, it's used when:

- Multiple EC2 instances need **shared access** to the same data
  - Applications require a **POSIX-compliant** file system
  - High **availability** and **durability** across multiple AZs are required
  - Serverless applications like AWS Lambda need persistent storage
- 

### ♦ Q2: What performance modes does EFS support? How do you choose?

**Answer:**

- **General Purpose (default)**: Best for latency-sensitive applications like CMS, developer tools, etc.
- **Max I/O**: Ideal for massively parallel workloads (e.g., big data, genomics), allows more throughput with slightly higher latency.

♦ **Use General Purpose** unless your workload has thousands of concurrent connections and can tolerate extra latency.

---

♦ **Q3: How does AWS EFS achieve high availability and durability?**

**Answer:**

- EFS is **region-wide** and **spans multiple Availability Zones (AZs)**
  - Data is automatically **replicated across multiple AZs**
  - No single point of failure: if one AZ fails, others still provide access
- 

♦ **Q4: What is EFS Lifecycle Management?**

**Answer:**

EFS lifecycle management **automatically moves files to the Infrequent Access (IA) storage class** after a defined period of inactivity (7, 14, 30, 60, 90 days). It helps reduce cost by storing cold data more efficiently.

---

♦ **Q5: What are EFS Access Points, and when should you use them?**

**Answer:**

Access Points provide **application-specific access paths** into an EFS file system with:

- Specific user IDs and root directories
- Isolation per app/service
- Simplified multi-tenant access

Useful in **ECS, EKS, and Lambda** environments where multiple services need controlled access.

---

♦ Q6: How do you mount an EFS file system securely?

Answer:

- Use the `amazon-efs-utils` package
- Mount using **TLS**:

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

This ensures **data in transit** is encrypted. Encryption at rest is also available via KMS.

---

♦ Q7: Can you use EFS across VPCs or regions?

Answer:

- **Cross-region:** ❌ Not natively supported
  - **Cross-VPC (same region):** ✅ Possible using **VPC Peering** or **Transit Gateway**, but must manage DNS resolution and mount target access carefully.
- 

♦ Q8: How does throughput work in EFS?

Answer:

- **Bursting Mode (default):** Throughput scales with the size of the file system

- **Provisioned Mode:** You can explicitly provision throughput (good for predictable workloads)

📌 **BurstCreditBalance** is important — if it depletes, performance drops.

---

◆ **Q9: What are common EFS performance bottlenecks and how do you troubleshoot?**

**Answer:**

- **Burst credit exhaustion:** Check **BurstCreditBalance** in CloudWatch
  - **Throughput limits:** Use Provisioned mode if needed
  - **Concurrent connections:** Use Max I/O mode if thousands of clients
  - **Mount point delays:** Ensure proper SGs and NFS port (2049) are open
- 

◆ **Q10: What are security best practices for EFS in production?**

**Answer:**

- Restrict NFS access (port 2049) via **tight Security Group rules**
  - Use **IAM policies** with Access Points for least privilege
  - Encrypt at rest with **KMS**
  - Encrypt in transit with **TLS**
  - Enable logging and use **CloudTrail** for monitoring access
-

♦ Q11: What's the difference between EFS, EBS, and S3?

Feature	EFS	EBS	S3
Type	Shared block storage (NFS)	Block storage	Object storage
Use Case	Shared access, CMS, Lambda	Databases, OS disks	Static assets, backups
Multi-AZ	Yes	No (1 AZ)	Yes
Concurrent EC2	Yes	No	N/A
Mountable	Yes	Yes (1 EC2)	No (via SDK/API)

---

♦ Q12: How would you use EFS with containers (ECS or EKS)?

Answer:

- With ECS: Use **EFS volume integration**, defined in task definition
  - With EKS: Use the **EFS CSI driver** to mount EFS volumes as persistent volumes
  - With Lambda: Mount EFS using **Access Points** and define mount path in function settings
- 

♦ Q13: How do you automate EFS setup in production?

Answer:

Use **Infrastructure as Code** tools:

- **Terraform** or **CloudFormation** to:
  - Create EFS file system
  - Create mount targets in all subnets

- Set lifecycle policies
  - Configure access points
  - Automate mounting in EC2 **User Data scripts**
- 

♦ Q14: What metrics do you monitor for EFS in production?

Answer:

- **BurstCreditBalance**: Prevent throttling
- **ClientConnections**: See number of connected clients
- **TotalIOBytes**: Understand traffic volume
- **PercentIOLimit**: To check nearing throughput limits

Use **CloudWatch Alarms** for proactive alerting.

---

♦ Q15: What's a real-world use case where you chose EFS over EBS?

Answer:

Use case: A **shared file store** for a WordPress cluster behind ALB.

Why EFS:

- Needed all EC2 instances to **access the same wp-content folder**
- Required **multi-AZ failover**
- CMS required **POSIX compliance**

- Chose **General Purpose + Lifecycle Policy** for cost optimization
- 

## Summary

EFS is great for **shared, scalable, highly available file storage** where **multi-instance access, POSIX compliance**, and **simplicity** are important. But you must handle **performance and cost** carefully in production.

---