PROJECT TITLE

SHARED FILE SYSTEM USING AWS

NAME: SANDHYA

CERTIFICATION: AWS SOLUTIONS ARCHITECT: SAA C03

VALIDATION NUMBER:BB8SQRWBQJR11193

INTRODUCTION

Hello, I'm Sandhya, a certified Solutions Architect with hands-on experience in AWS foundational services and Infrastructure Management. I've successfully completed certifications in AWS (SAA-C03), Azure (AZ-104), and DevOps (AZ-400), which have equipped me with a robust understanding of cloud technologies and best practices.

In this project, we embarked on a journey to design and implement a highly available, secure, and scalable cloud architecture. Our objectives were clear: to ensure data integrity, facilitate seamless collaboration, and maintain stringent security standards.

As a Solutions Architect, I'm passionate about leveraging cloud technologies to solve complex business challenges and drive innovation. I'm looking forward to discussing how my skills and experiences align with the needs of your organization.

PROBLEM STATEMENT

ABC, an India-based entertainment production company focusing on Northeast and East Indian cinema, requires a highly available and reliable storage solution for their on-premises data.

Challenges with Existing Infrastructure:

- *Limited scalability of NAS storage.*
- *Desire for centralized storage with AWS cloud application integration.*
- *Security vulnerabilities with lack of encryption*
- *Manual intervention was needed to change the storage type and transfer files to infrequent storage.*
- *The client is unable to scale up the infrastructure due to high capital costs for new*
- *hardware.*
- *Need for low-cost storage options for both frequent and infrequent data.*
- *Highly available, secure, and persistent shared File system in AWS cloud with EFS*

CHALLENGES & SOLUTION

Challenge Overview: ABC found itself constrained by a legacy NAS storage system, which lacked the necessary scalability and posed significant security vulnerabilities. The company's vision to migrate to a cloud-based application with centralized data synchronization was hindered by the existing infrastructure's limitations.

Strategic Solution:

We created a highly available, reliable, and secure storage distributed file system using **Amazon Elastic File System (EFS)**, which allows multiple EC2 instances to share data efficiently and securely.

Security and Reliability:

We secured data in Amazon EFS with AWS KMS encryption, addressing encryption concerns at rest and in transit

Infrastructure Setup:

Configured EC2 instances with custom security groups in a VPC for EFS, ensuring secure data transfer.

Cost-Effective Scaling: Leveraged AWS cloud for cost-effective storage scaling without hardware investments.

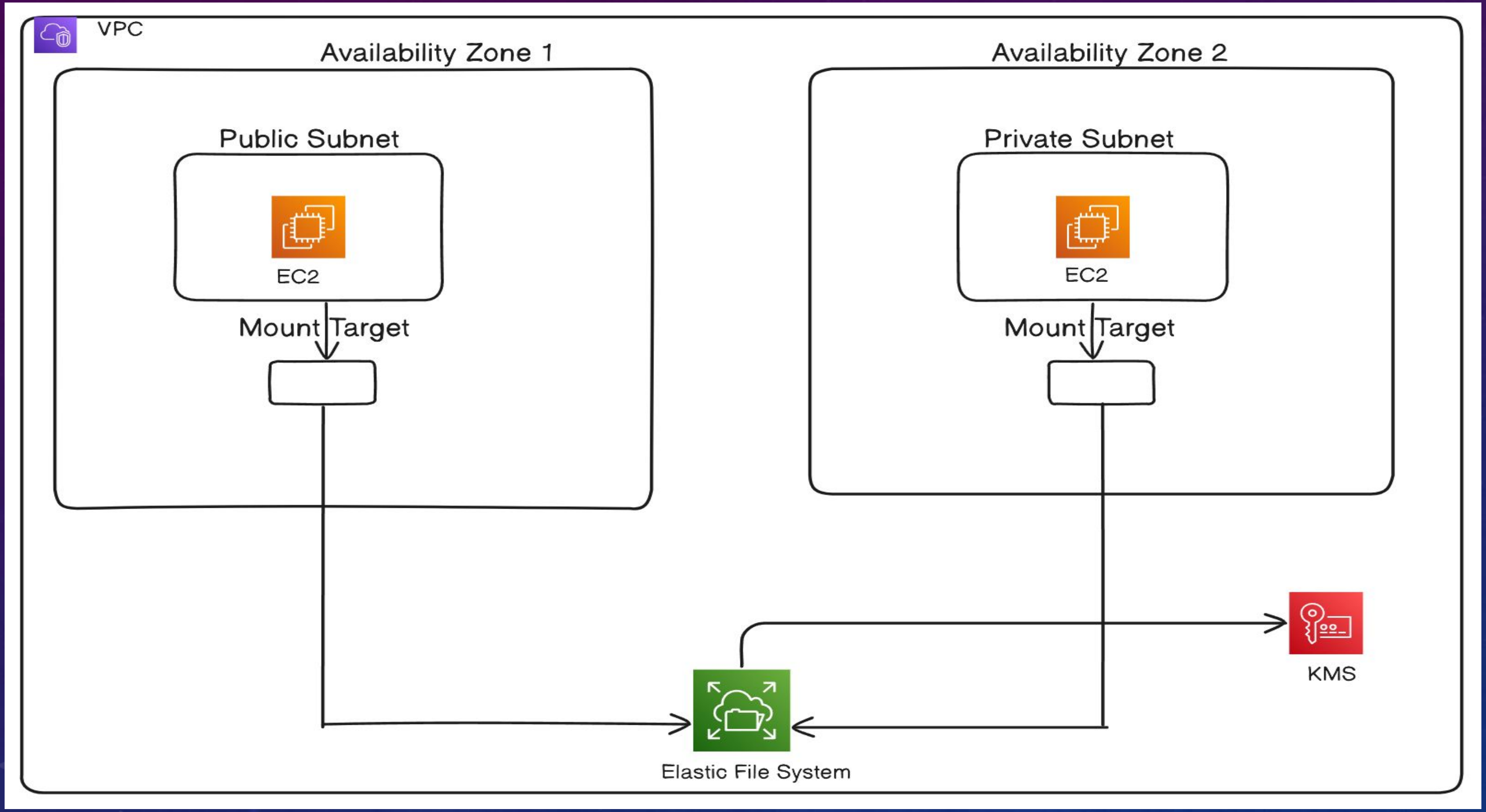
Resiliency:

Deployed EC2 instances across zones for system robustness.

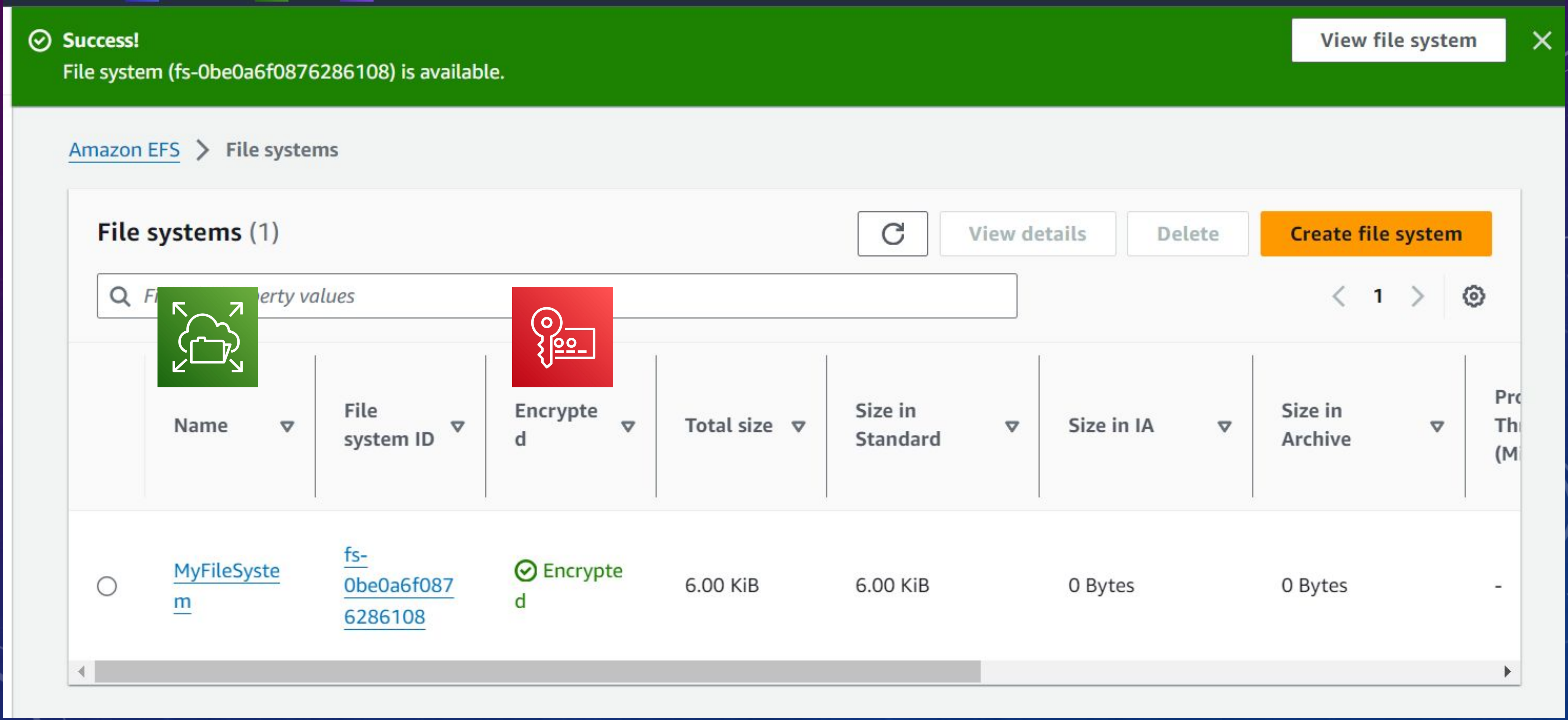
Hands-On Execution:

Connected to instances, installed utilities, and mounted EFS for real-time data sync.

ARCHITECTURAL DIAGRAM



CREATING ELASTIC FILE SYSTEM WITH KMS



Success!
File system (fs-0be0a6f0876286108) is available.

View file system

Amazon EFS > File systems

File systems (1)

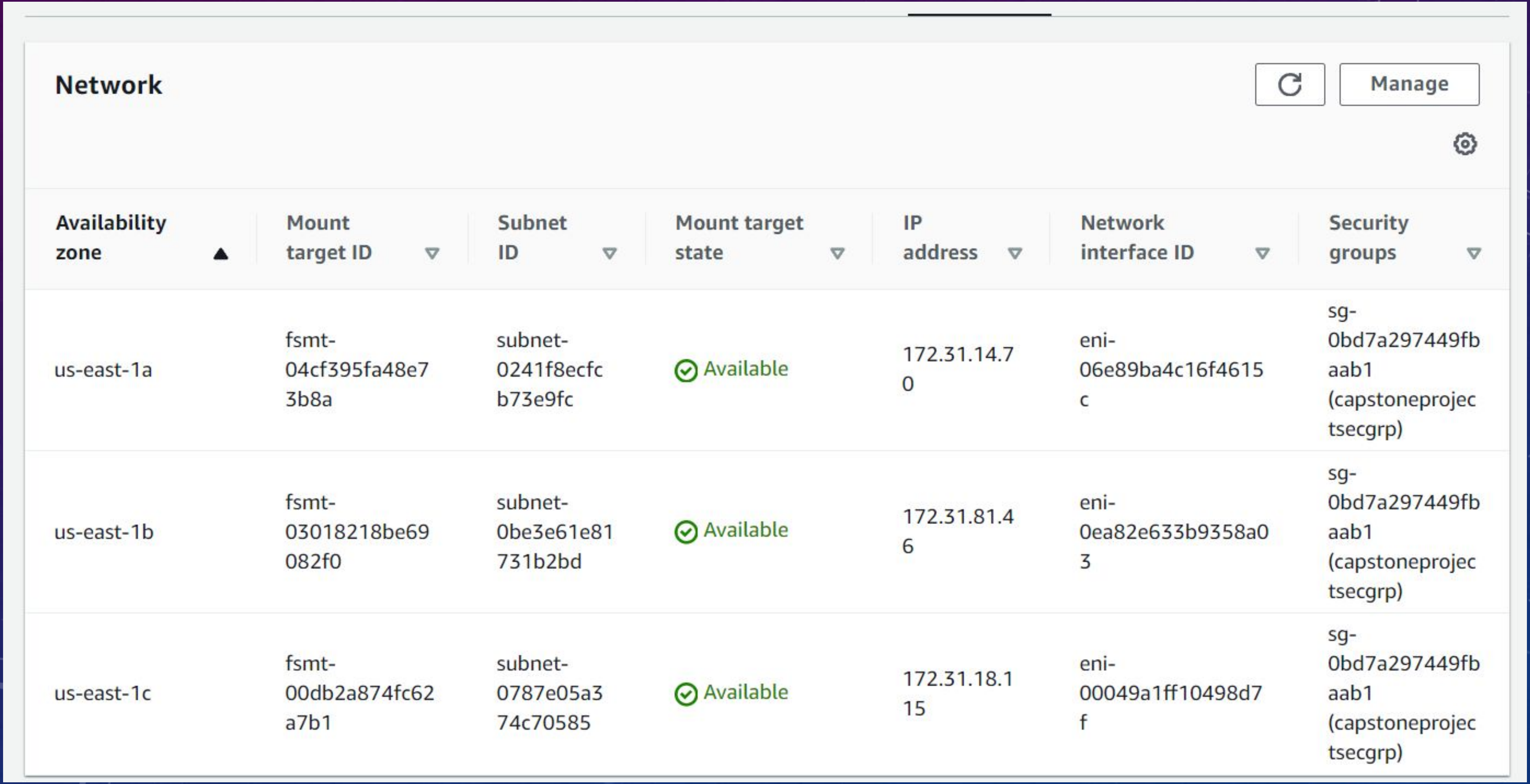
View details | Delete | Create file system

Find ...erty values

1

	Name	File system ID	Encrypted	Total size	Size in Standard	Size in IA	Size in Archive	Pro Th (M
○	MyFileSystem	fs-0be0a6f0876286108	✓ Encrypted	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes	-

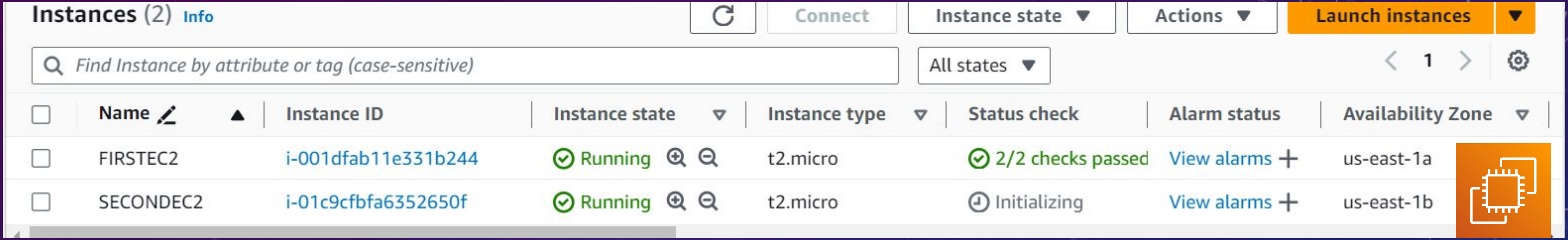
CONFIGURED AN INBOUND RULE FOR NFS (NETWORK FILE SYSTEM) THAT ALLOWS ACCESS FROM ANY IP ADDRESS (REPLACED A NEW SG)



Network

Availability zone	Mount target ID	Subnet ID	Mount target state	IP address	Network interface ID	Security groups
us-east-1a	fsmt-04cf395fa48e73b8a	subnet-0241f8ecfcb73e9fc	✓ Available	172.31.14.70	eni-06e89ba4c16f4615c	sg-0bd7a297449fbaab1 (capstoneprojectsecgrp)
us-east-1b	fsmt-03018218be69082f0	subnet-0be3e61e81731b2bd	✓ Available	172.31.81.46	eni-0ea82e633b9358a03	sg-0bd7a297449fbaab1 (capstoneprojectsecgrp)
us-east-1c	fsmt-00db2a874fc62a7b1	subnet-0787e05a374c70585	✓ Available	172.31.18.115	eni-00049a1ff10498d7f	sg-0bd7a297449fbaab1 (capstoneprojectsecgrp)

CREATED 2 INSTANCES IN 2 DIFF AZ

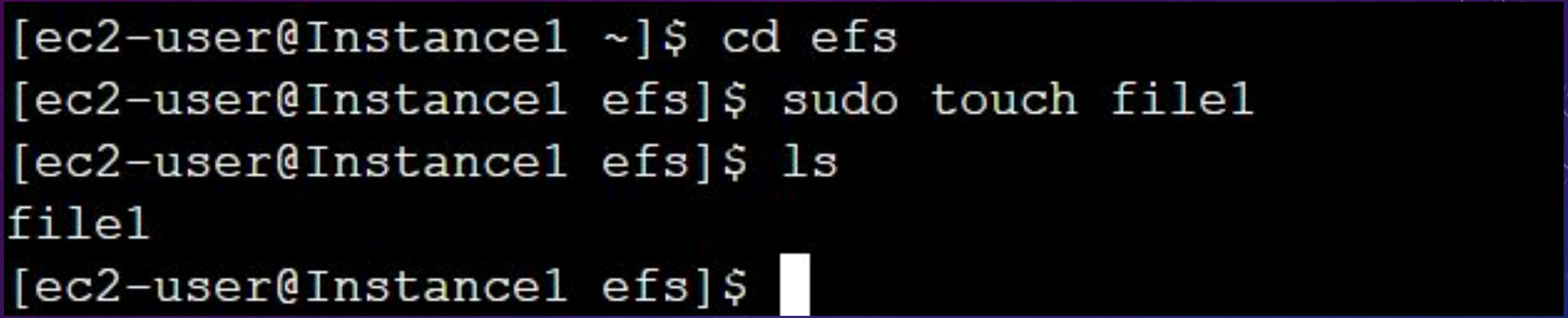


	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	FIRSTEC2	i-001dfab11e331b244	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1a
☐	SECONDEC2	i-01c9cfbfa6352650f	✓ Running	t2.micro	Initializing	View alarms +	us-east-1b

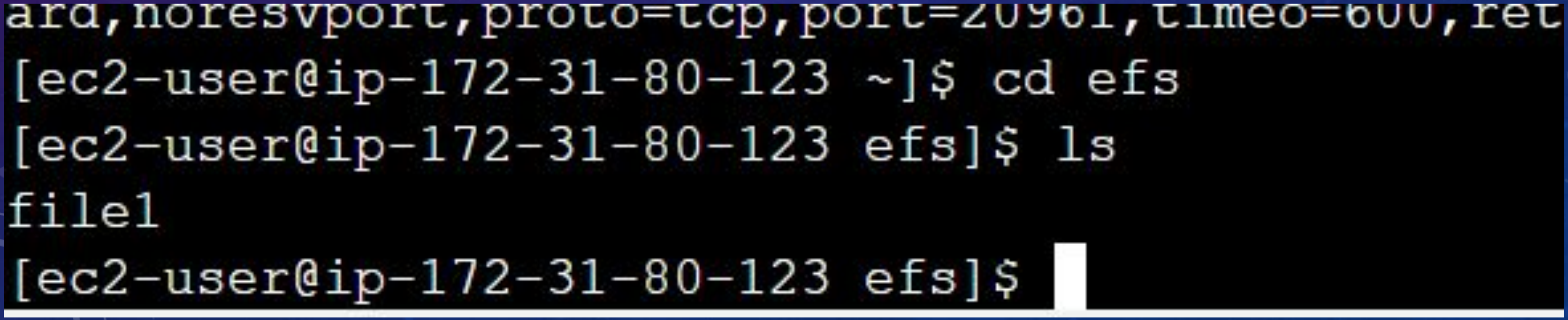
- ***<u>Detailing the steps to set up and verify a shared file system using Amazon EFS across two EC2 instances.</u>***
- Install EFS utilities: sudo yum install amazon-efs-utils
- Mount the EFS File System: sudo mount -t efs -o tls fs-0be0a6f0876286108:/ efs
- Verify the Mount: df –k
- Check mount details: mount | column -t

TESTING EFS ACROSS INSTANCES

You can access file 1 created by FIRSTEC2 through SECONDEC2. Hence, shared file the system is working with both EC2.



```
[ec2-user@Instance1 ~]$ cd efs
[ec2-user@Instance1 efs]$ sudo touch file1
[ec2-user@Instance1 efs]$ ls
file1
[ec2-user@Instance1 efs]$
```



```
ard,noresvport,proto=tcp,port=20961,timeo=600,ret
[ec2-user@ip-172-31-80-123 ~]$ cd efs
[ec2-user@ip-172-31-80-123 efs]$ ls
file1
[ec2-user@ip-172-31-80-123 efs]$
```

BEST PRACTICES

- **Amazon EFS Best Practices:**
- Performance Mode: Choose the right performance mode (General Purpose or Max I/O) based on your workload requirements.
- Lifecycle Management: Implement EFS Lifecycle Management to automatically move infrequently accessed files to a cost-effective storage class.
- Monitoring: Use Amazon CloudWatch to monitor file system performance and set up alarms for unusual activity.

- **AWS KMS Best Practices:**
- Key Rotation: Enable automatic key rotation for your customer master keys (CMKs) to enhance security.
- Least Privilege: Apply the principle of least privilege by granting only necessary permissions to IAM roles and users for KMS operations.
- Audit and Compliance: Regularly audit your KMS usage with AWS CloudTrail to ensure compliance with security policies.

BEST PRACTICES

- **Amazon EC2 Best Practices:**
- Security Groups: Restrict security group rules to allow only necessary traffic. Avoid using 0.0.0.0/0 (open to the world) unless absolutely necessary.
- Instance Types: Choose the right EC2 instance type that matches your performance and cost requirements.
- Elastic IP (EIP): Use EIPs judiciously and release them when not in use to avoid unnecessary charges.
- Backup and Recovery: Regularly create snapshots of your EC2 instances to facilitate quick recovery in case of failure.

- **General AWS Best Practices:**
- Multi-AZ Deployment: Deploy resources across multiple Availability Zones for high availability.
- IAM Roles: Use IAM roles for EC2 instances to securely access other AWS services without the need to manage credentials.
- Encryption: Encrypt data at rest and in transit to ensure data security and privacy.
- Cost Management: Utilize AWS Budgets and Cost Explorer to monitor and manage your AWS spending.

Challenges:

Encountered a mounting issue with EFS due to an incorrect directory path. Initially attempted to set the mount point within a newly created directory, which was not effective. After troubleshooting, I realized the command needed to be executed from the parent directory. This adjustment resolved the mounting problem, allowing for successful attachment of the mount point to EFS

Thank you!