***Case Study-4:***

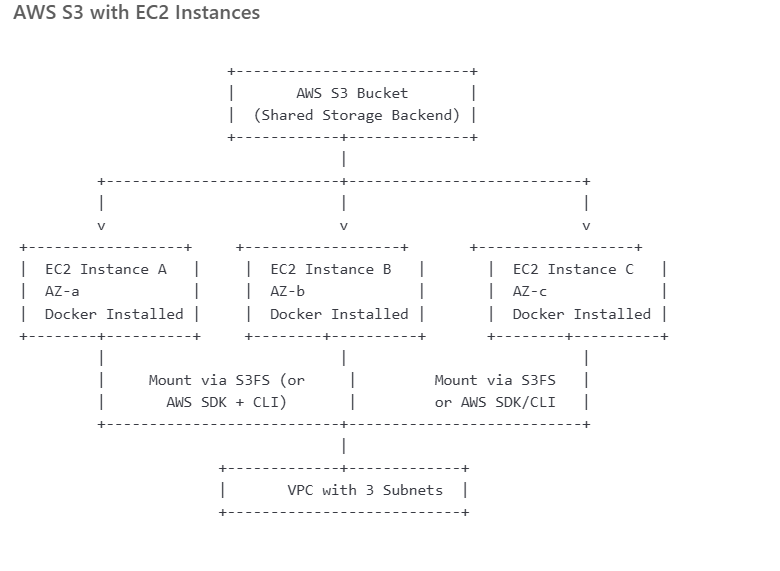
***Problem Statement:*** *Client wants to provision an AWS S3 to provide* ***shared storage*** *across 3 AWS EC2 instances. The EC2 instances must exist in the same VPC with each instance in separate availability zone. Implement a solution to ensure that all the 3 EC2 instances can* ***mount the S3 bucket****. All these EC2 instances have* ***docker installed*** *on it. Use appropriate IaC (Infrastructure as code) tool to develop AWS infrastructure.*

***Duration:*** *2 days*

***Tech Stack:***

*AWS CloudFormation/Terraform:*

**Implementation diagram:**

**

1. **Create S3 Bucket using Terraform**
2. **Create IAM Role for EC2 to Access S3**
3. **Set Up Networking - VPC, Subnets, IGW, Route Table**
4. **Create Security Group**
5. **User Data Script (for Docker + S3FS)**
6. **Launch 3 EC2 Instances (1 per AZ)**

**Key Components:**

**✅ VPC with 3 subnets in separate AZs**

**✅ Internet Gateway and public route table**

**✅ 3 EC2 instances in different AZs**

**✅ IAM role and instance profile with AmazonS3FullAccess**

**✅ Security group allowing SSH (port 22)**

**✅ S3 bucket for shared storage**

**✅ Docker and S3FS setup assumed in user\_data.sh**

| **Subnet Name** | **CIDR Block** | **AZ** |
| --- | --- | --- |
|  |  |  |

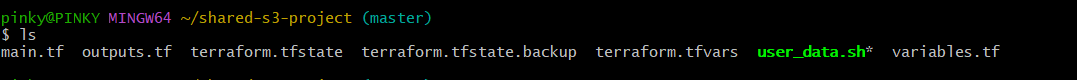
|  |  |  |
| --- | --- | --- |
| Subnet 1 (AZ1) | 10.0.1.0/24 | us-east-1a |

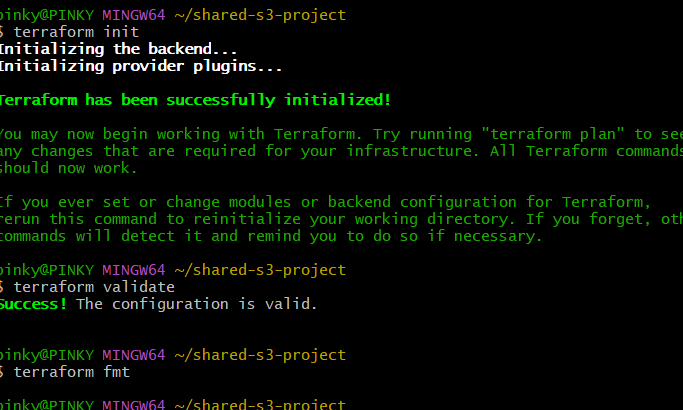
|  |  |  |
| --- | --- | --- |
| Subnet 2 (AZ2) | 10.0.2.0/24 | us-east-1b |

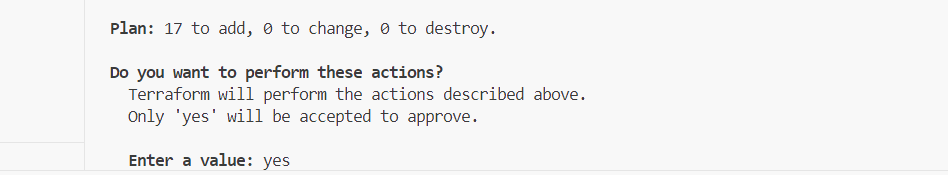
|  |  |  |
| --- | --- | --- |
| Subnet 3 (AZ3) | 10.0.3.0/24 | us-east-1c |

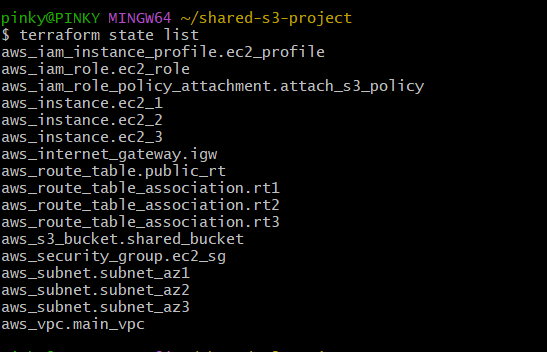
vpc\_cidr = "10.0.0.0/16"

**I have created the resources using Terraform.**

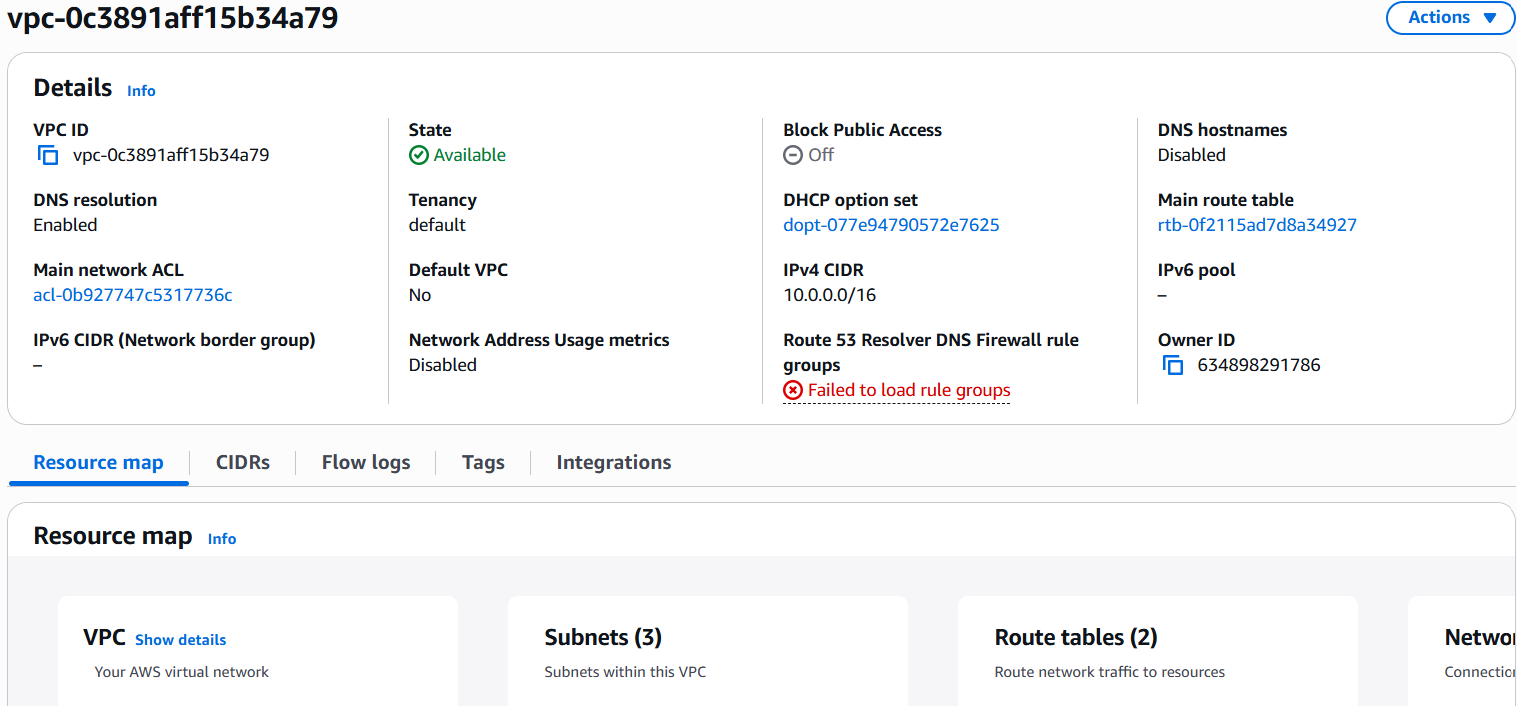
****

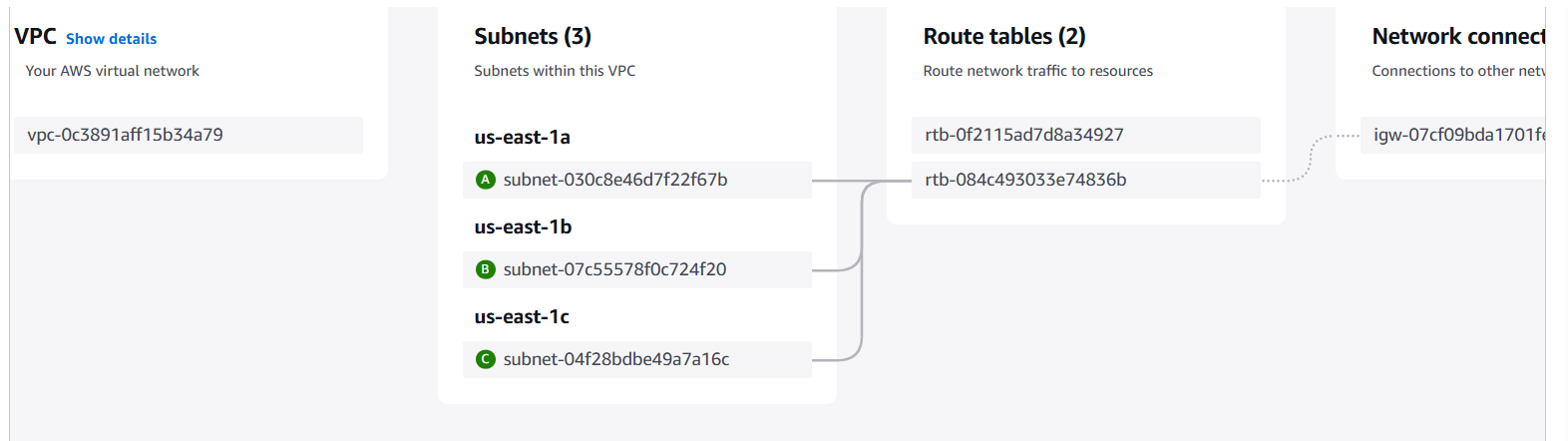
****

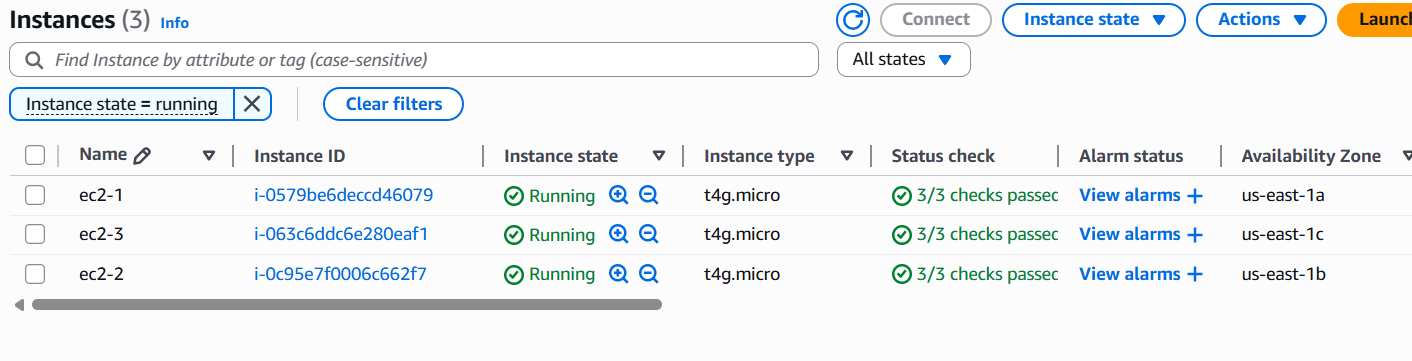
****

****

**3 subnets and 3 ec2 instances and are created in the vpc as seen above.**

****

****

****

**NOTE:**

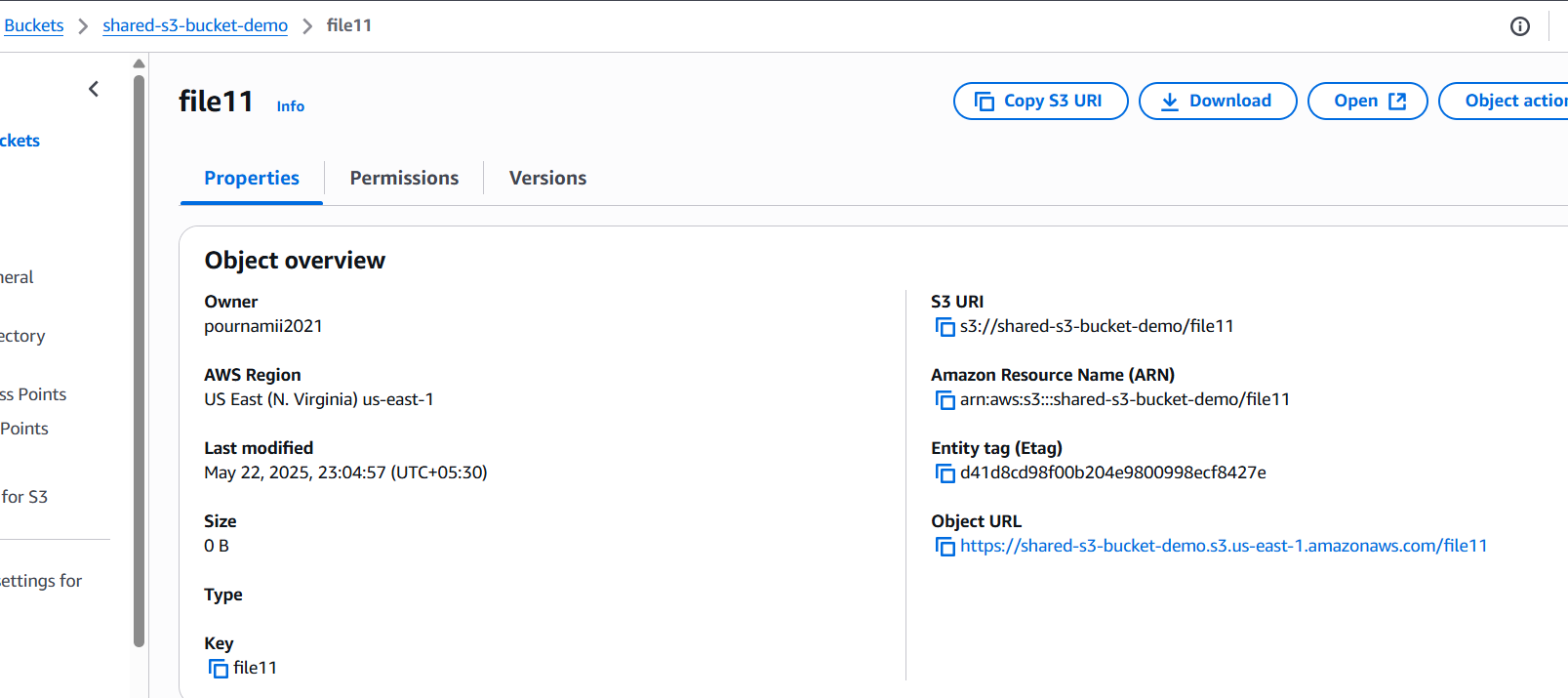
**-------------------------------------------------------------------------------------------------------------------------------------**

* **Each EC2 instance mounts the same S3 bucket using s3fs**
* **The S3 bucket becomes a shared storage across the 3 EC2s**
* **You can use the bucket to store or read files from containers or host OS**

**s3fs is a FUSE-based (Filesystem in Userspace) tool that allows you to mount an Amazon S3 bucket as a local filesystem on a Linux or macOS system. This means you can access your S3 bucket using regular file system commands (ls, cp, mv, etc.) as if it were a local disk.**

**=========================================================================**

S3 bucket is created as seen below and it has file11 also.

****

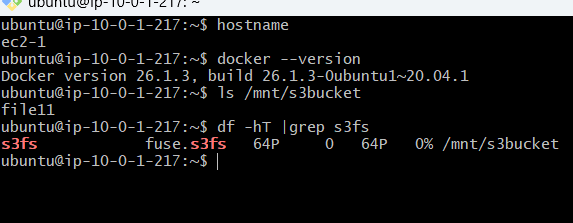
**SSH into Each EC2 instance and check:**

Installed it using the script

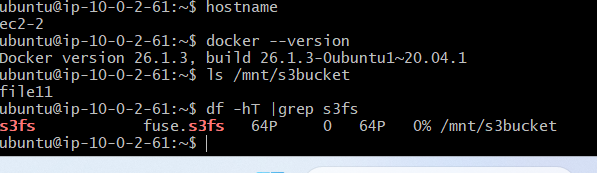
* **Hostname:**
* **Docker version**
* **S3fs installation**
* **Able to mount the s3 and list the mounted s3 bucket**
* **Please see the below screenshots:**

*using s3fs I am able to access the same s3 bucket in each ec2 instances which are in different AZ and one VPC.*

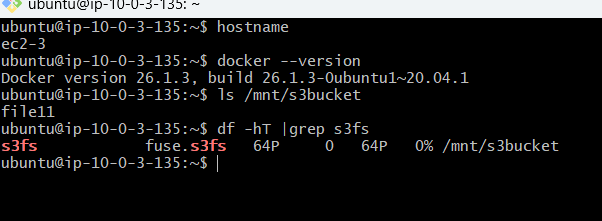
**EC2-1 (First Instance)**

****

**EC2-2 (second Instance)**

****

**EC2-3 (third Instance)**

****

* **File 11 is able to view from each ec2 instances**
* **Thus I am able to mount the same s3 bucket as shared storage in each ec2 instances in different AZs of one vpc**