

# **Centralized Cloud IT Infrastructure Setup for a Growing SME**

**Name: Harini.P**

**Register Number: 727823TUCY014**

## **1. Problem Statement**

A small enterprise with 25 employees is facing frequent system downtime, unstructured data storage, and lack of remote access. All applications and files currently run on individual employee systems with no centralized control. This leads to poor manageability, higher risk of data loss, and difficulty in scaling IT operations.

The IT head wants to move to a cloud-based infrastructure that is stable, manageable, and scalable without increasing operational complexity. However, the organization has no prior cloud experience, and a junior intern is assigned to design and implement a basic but production-ready cloud infrastructure.

## **2. Project Objectives**

- To design and deploy a centralized cloud-based IT infrastructure
- To replace decentralized employee systems with a central server
- To provide secure remote administrative access
- To allocate CPU, memory, and storage appropriately
- To implement backup and recovery using snapshots
- To simulate real-world IT administration tasks
- To document infrastructure decisions clearly

## **3. Scope of the Project**

- Cloud platform: AWS Free Tier
- Single centralized virtual machine
- Manual configuration only (no automation tools)
- Moderate workload (10–15 concurrent users)
- Focus on stability and reliability

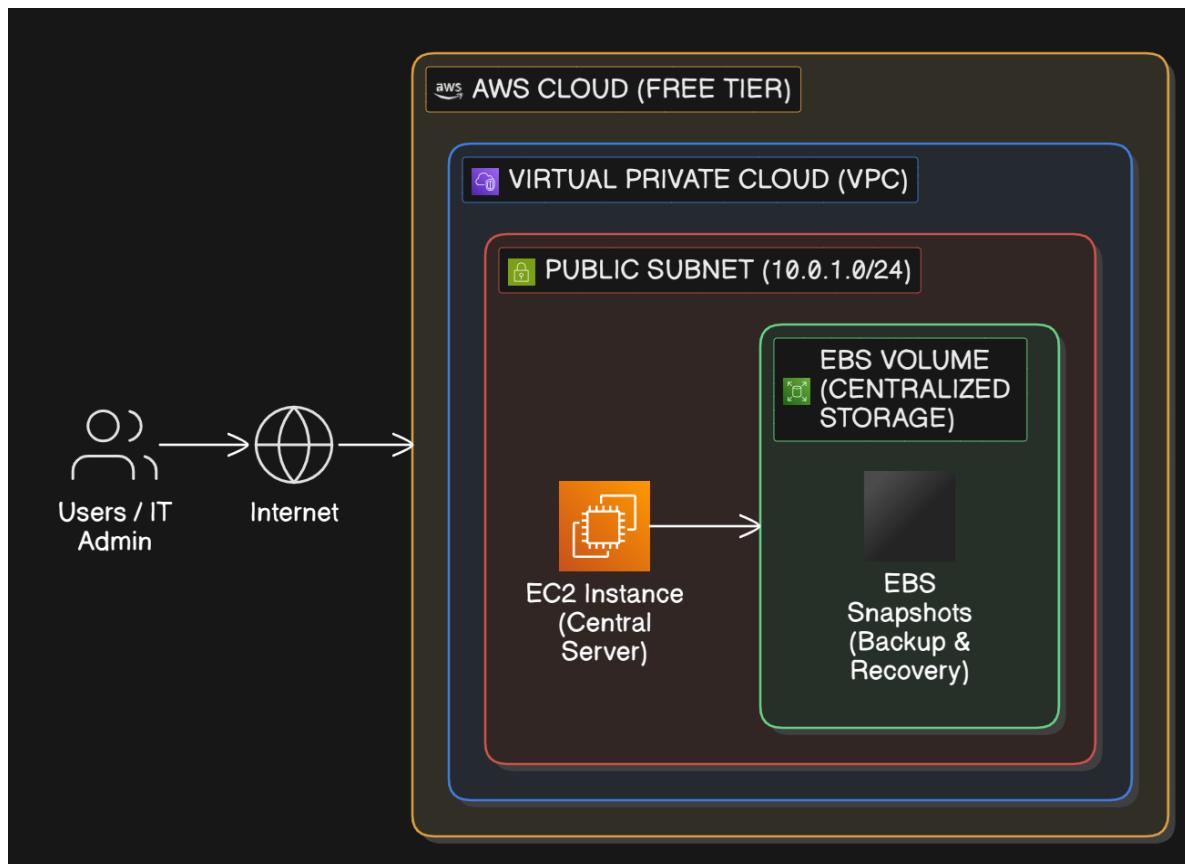
## 4. Architecture Overview

The proposed architecture provides a simple, secure, and cost-effective centralized IT infrastructure using AWS Free Tier services. A dedicated VPC ensures network isolation, while a public subnet and internet gateway allow secure remote access to the EC2 instance.

The EC2 instance acts as a centralized server for all users, and EBS volumes provide persistent storage. EBS snapshots are used for backup and recovery, ensuring data protection and business continuity. Security Groups and IAM users improve access control and overall security.

This architecture is well-suited for a small enterprise with moderate workload and limited cloud experience. While it lacks advanced features like auto scaling and monitoring, it meets current requirements and provides a strong foundation for future enhancements.

## 5. Block Diagram of the System



The above block diagram represents how end users and administrators securely access a centralized cloud server hosted on AWS. The EC2 instance acts as the

main server, while EBS provides persistent storage and snapshots ensure backup and recovery.

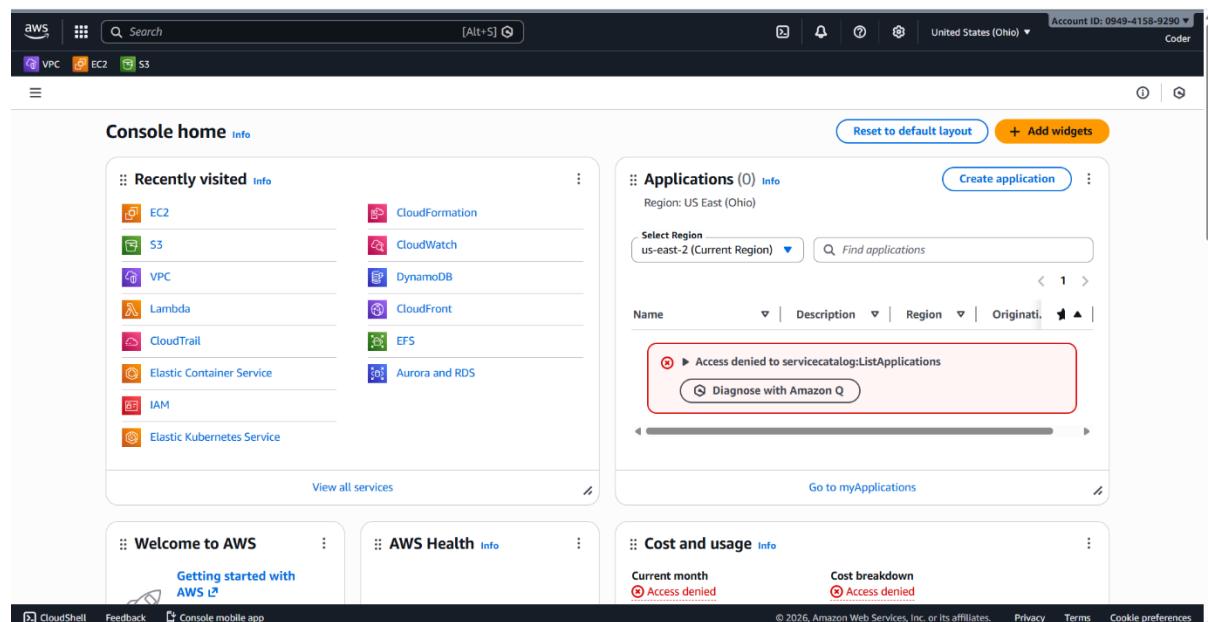
## 6. Step-by-Step Implementation Guide

### Step 1: AWS Account Setup

Creating an AWS account is the foundation of this project. The Free Tier allows limited but sufficient resources to complete this setup without cost.

- Sign up for an AWS account using an email ID
- Activate Free Tier services
- Use the root account only for billing and initial setup
- Enable basic security settings

This step ensures a secure and cost-controlled cloud environment.



### Step 2: IAM User Creation

IAM (Identity and Access Management) is used to manage users and permissions securely.

- Navigate to IAM → Users → Create user
- Enter username: Coder
- Enable AWS Management Console access
- Assign a custom password
- Attach the policy AdministratorAccess

This avoids using the root account for daily operations and follows AWS security best practices.

The screenshot shows the AWS Identity and Access Management (IAM) service interface. In the left sidebar, under 'Access Management', the 'Users' option is selected. The main pane displays a table titled 'Users (1)'. The table has one row for a user named 'Coder'. The columns include 'User name' (Coder), 'Path' (/), 'Group' (0), 'Last activity' (41 minutes ago), 'MFA' (-), 'Password age' (24 hours), and 'Console last sign-in' (41 minutes ago). At the top right of the table, there are 'Delete' and 'Create user' buttons. The top navigation bar shows 'Account ID: 0949-4158-9290' and the current user 'Coder'.

### Step 3: Create Virtual Private Cloud (VPC)

A VPC is an isolated virtual network inside AWS that provides control over IP addressing and networking.

- Go to VPC → Create VPC
- Name: SME-VPC
- IPv4 CIDR block: 10.0.0.0/16
- Enable DNS resolution and DNS hostnames

This VPC acts as the private network for hosting the central server securely.

The screenshot shows the AWS VPC dashboard. In the left sidebar, under 'Virtual private cloud', the 'Your VPCs' option is selected. The main pane displays a table titled 'Your VPCs (2)'. The table has two rows: one for a default VPC represented by a dash symbol and another for a VPC named 'sme-vpc'. The columns include 'Name', 'VPC ID', 'State', 'Encryption controls', 'Encryption control ...', 'Block Public...', and 'IPv4'. Both VPCs are listed as 'Available'. At the top right of the table, there are 'Actions' and 'Create VPC' buttons. The top navigation bar shows 'Account ID: 0949-4158-9290' and the current user 'Coder'.

### Step 4: Create Public Subnet

Subnets divide the VPC into smaller network segments.

- Create a subnet named Public-Subnet
- CIDR block: 10.0.1.0/24
- Select any availability zone
- Enable auto-assign public IPv4 address

The public subnet allows the EC2 instance to be accessed remotely via the internet.

The screenshot shows the AWS VPC dashboard with the Subnets section selected. A success message at the top indicates "You have successfully created 1 subnet: subnet-05046607207be783b". The main table lists one subnet:

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
Public Subnet	subnet-05046607207be783b	Available	vpc-0f8e9c10103f35156   sme...	Off	10.0.1.0/24

Below the table, the details for the subnet are shown:

**subnet-05046607207be783b / Public Subnet**

**Details** | Flow logs | Route table | Network ACL | CIDR reservations | Sharing | Tags

<b>Details</b>	<b>Subnet ID</b> subnet-05046607207be783b	<b>Subnet ARN</b> arn:aws:ec2:us-east-2:09494158929 0:subnet/subnet-05046607207be783b	<b>State</b> Available	<b>Block Public Access</b> Off
	<b>IPv4 CIDR</b> 10.0.1.0/24	<b>Available IPv4 addresses</b> 251	<b>IPv6 CIDR</b> -	<b>IPv6 CIDR association ID</b> -

## Step 5: Internet Gateway and Routing

An Internet Gateway enables communication between resources in the VPC and the internet.

- Create an Internet Gateway named SME-IGW
- Attach it to SME-VPC
- Create a Route Table named Public-RT
- Add route: 0.0.0.0/0 → SME-IGW
- Associate the route table with Public-Subnet

This configuration ensures outbound and inbound internet connectivity.

The screenshot shows the AWS VPC Internet gateways console. On the left, there's a navigation sidebar with options like VPC dashboard, AWS Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), and PrivateLink and Lattice. The main area displays a table titled "Internet gateways (1/1) Info". The table has columns for Name (SME-IGW), Internet gateway ID (igw-08d53efa91671119e), State (Attached), VPC ID (vpc-0f8e9c10103f35156 | sme-vpc), and Owner (094941589290). Below the table, a specific gateway entry is expanded, showing its details: Internet gateway ID (igw-08d53efa91671119e), State (Attached), VPC ID (vpc-0f8e9c10103f35156 | sme-vpc), and Owner (094941589290).

The screenshot shows the AWS VPC Route tables console. The left sidebar includes options for VPC dashboard, AWS Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), and PrivateLink and Lattice. A green notification bar at the top says "Updated routes for rtb-0f2e4b10c00bc90b9 successfully" with a link to "Details". The main area shows a table titled "Route tables (1/2) Info". The table has columns for Name (rtb-0e96a609e289b0e5f, rtb-0f2e4b10c00bc90b9), Route table ID (rtb-0e96a609e289b0e5f, rtb-0f2e4b10c00bc90b9), Explicit subnet associations (None, subnet-05046607207be783b / Public Subnet), Edge associations (None, -), Main (Yes, Yes), and VPC (vpc-0ab61ca2885228690, vpc-0f8e9c10103f35156 | sme-vpc). Below the table, a specific route table entry is expanded, showing its details: Route table ID (rtb-0f2e4b10c00bc90b9), Main (Yes), Owner ID (094941589290), Explicit subnet associations (subnet-05046607207be783b / Public Subnet), and Edge associations (-).

## Step 6: Launch EC2 Instance (Central Server)

The EC2 instance acts as the centralized server replacing on-premise systems.

- Navigate to EC2 → Launch Instance
- Instance name: SME-Central-Server
- Choose AMI: Amazon Linux 2023
- Instance type: t2.micro or t3.micro (Free Tier)
- Create a new key pair named sme-admin-key.pem

This virtual machine provides compute resources for hosting enterprise services.

The screenshot shows the AWS EC2 'Launch an instance' wizard. The top navigation bar includes the AWS logo, search bar, and tabs for VPC, EC2, and S3. The main navigation path is EC2 > Instances > Launch an instance.

**Name and tags** Info

Name: SME-Central-Server [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

**Recents** **Quick Start**

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian.

**Browse more AMIs** Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Amazon Linux 2023 kernel-6.1 AMI Free tier eligible

ami-03ea746da1a2e36e7 (64-bit (x86), uefi-preferred) / ami-0be5a830e851483f9 (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Instance type** Info | Get advice

**Instance type**: t3.micro Free tier eligible

Family: t3 2 vCPU 1 GiB Memory Current generation: true  
On-Demand RHEL base pricing: 0.0392 USD per Hour  
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour  
On-Demand Windows base pricing: 0.0196 USD per Hour  
On-Demand SUSE base pricing: 0.0104 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour

All generations [Compare instance types](#)

**Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**: sme-admin-key.pem [Create new key pair](#)

## Step 7: Configure Storage (EBS)

Elastic Block Store (EBS) provides persistent storage for the EC2 instance.

- Root volume type: gp3
- Storage size: 20 GB
- Used for application data and centralized files

EBS ensures data is retained even if the EC2 instance is stopped.

▼ Configure storage [Info](#) [Advanced](#)

1x  GiB  Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

Click refresh to view backup information  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

## Step 8: Security Group Configuration

Security Groups act as virtual firewalls controlling inbound and outbound traffic.

- Create a Security Group named SME-SG
- Allow SSH (port 22) from My IP
- Optionally allow HTTP (80) and HTTPS (443)

Restricting SSH access improves server security.

▼ Network settings [Info](#)

VPC - required | [Info](#)

vpc-0f8e9c10103f35156 (sme-vpc) [Edit](#)

Subnet | [Info](#)

subnet-05046607207be783b Public Subnet [Edit](#) [Create new subnet](#)

Auto-assign public IP | [Info](#)

Enable [Edit](#)

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Create security group](#)  [Select existing security group](#)

Security group name - required

SME-SG [Edit](#)

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-:/()#@+=;&;!\$\*

Description - required | [Info](#)

launch-wizard-1 created 2026-02-07T05:12:41.911Z [Edit](#)

**Inbound Security Group Rules**

▼ Security group rule 1 (TCP, 22, 115.247.219.102/32)

Type   Info	Protocol   Info	Port range   Info
ssh	TCP	22

Source type | Info

Name | Info

Description - optional | Info

My IP

115.247.219.102/32 X

e.g. SSH for admin desktop

Remove

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type   Info	Protocol   Info	Port range   Info
HTTP	TCP	80

Source type | Info

Source | Info

Description - optional | Info

Anywhere

0.0.0.0/0 X

e.g. SSH for admin desktop

Remove

▼ Security group rule 3 (TCP, 443, 0.0.0.0/0)

Type   Info	Protocol   Info	Port range   Info
HTTPS	TCP	443

Remove

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state: All states

Instance ID: i-032d148ad472b50ba

Name: SME-Central-S...

Status: Running

Instance type: t3.micro

Alarm status: Initializing

Availability Zone: us-east-2a

Public IP: 3.140.195.157

**i-032d148ad472b50ba (SME-Central-Server)**

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID: i-032d148ad472b50ba

Public IPv4 address: 3.140.195.157

Private IPv4 addresses: 10.0.1.119

IPv6 address: -

Instance state: Running

Public DNS: -

Hostname type: ip-name

Private IP DNS name (IPv4 only): 3.140.195.157

## Step 9: Connect to EC2 Instance

Secure Shell (SSH) is used for remote access to the EC2 instance. In this project, the EC2 instance was accessed using **AWS EC2 Instance Connect**, which provides a secure, browser-based SSH connection without requiring local key pair configuration.

## Steps followed:

- Navigate to **EC2 → Instances**
- Select the instance **SME-Central-Server**
- Click **Connect**
- Choose **EC2 Instance Connect**
- Click **Connect**

Upon successful login, the Amazon Linux 2023 terminal is displayed, confirming that the EC2 instance is running correctly and is securely accessible for administrative tasks.



The screenshot shows a terminal window with a dark background. At the top, there's a header bar with the AWS logo, a search bar containing 'Search', and some icons. The main terminal area displays a login banner for 'Amazon Linux 2023'. The banner features a stylized tree or root icon on the left and the text 'Amazon Linux 2023' and a URL 'https://aws.amazon.com/linux/amazon-linux-2023' on the right. Below the banner, the terminal prompt '[ec2-user@ip-10-0-1-48 ~]:' is visible.

## Step 10: User and Admin Simulation

Creating users simulates multiple administrators managing the server.

```
sudo adduser admin1
```

```
sudo passwd admin1
```

This reflects real-world IT operations in an enterprise environment.

```
aws | Search [Alt+S] 🔍

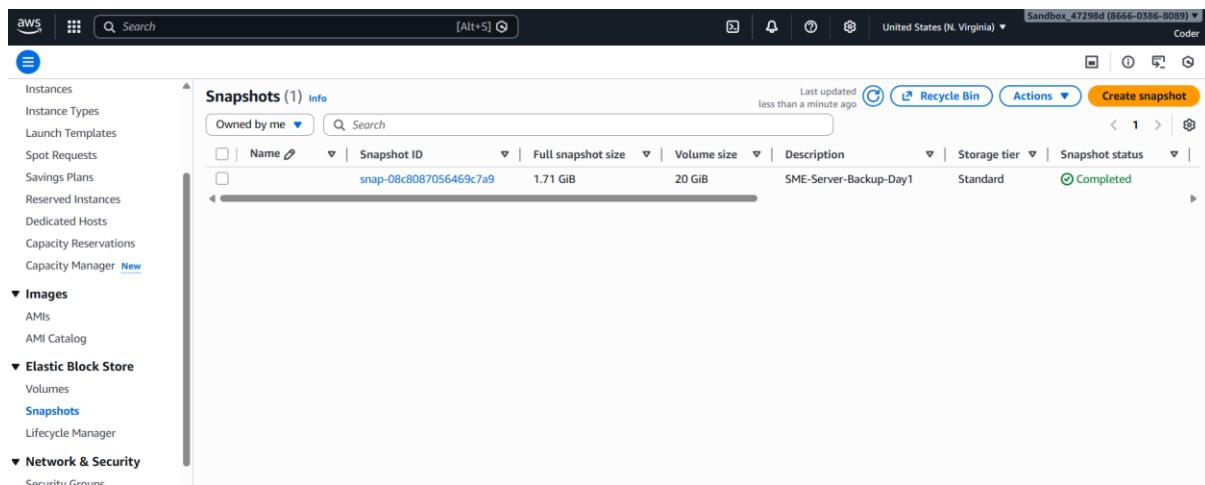
          #
         #_#
        #####_ Amazon Linux 2023
       \###|
      \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
     V~' .__>
      /
     /_/
    /m/ \
Last login: Sat Feb  7 08:55:43 2026 from 18.206.107.29
[ec2-user@ip-10-0-1-48 ~]$ sudo adduser admin
[ec2-user@ip-10-0-1-48 ~]$ sudo passwd admin
Changing password for user admin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-1-48 ~]$ sudo adduser admin5
[ec2-user@ip-10-0-1-48 ~]$ sudo passwd admin5
Changing password for user admin5.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-1-48 ~]$ 
```

## **Step 11: Backup Using Snapshots**

Snapshots are point-in-time backups of EBS volumes.

- Navigate to EC2 → Volumes
  - Select the root EBS volume
  - Create a snapshot named SME-Server-Backup-Day1

Snapshots provide data protection and disaster recovery.



## Step 12: Recovery Simulation

This step demonstrates system recovery in case of failure.

- Create a new EBS volume from the snapshot

- Attach it to the EC2 instance
- Mount the volume to access recovered data

lsblk

sudo mount /dev/xvdf /mnt/recovery

This proves the effectiveness of the backup strategy.

**Volumes (2) Info**

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Source volume ID	Created
vol-079fe6744a9b028cd	gp3	20 GiB	3000	125	snapshot-08c6087...	-	2026/02/07	
vol-060395ab916b27d84	gp3	20 GiB	3000	125	snapshot-09c6080...	-	2026/02/07	

**Fault tolerance for all volumes in this Region**

**Snapshot summary**

Recently backed up volumes / Total # volumes: 0 / 1

Last updated on Sat, Feb 07, 2026, 02:30:23 PM (GMT+05:30)

Data Lifecycle Manager default policy for EBS Snapshots status: No default policy set up | Create policy

**Successfully attached volume vol-079fe6744a9b028cd to instance i-081fd6e7bf5e989ba.**

**Volumes (1/2) Info**

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Source volume ID	Created
<input checked="" type="checkbox"/> vol-079fe6744a9b028cd	gp3	20 GiB	3000	125	snapshot-08c6087...	-	2026/02/07	
<input type="checkbox"/> vol-060395ab916b27d84	gp3	20 GiB	3000	125	snapshot-09c6080...	-	2026/02/07	

**Volume ID: vol-079fe6744a9b028cd**

**Details**

Volume ID vol-079fe6744a9b028cd	Size 20 GiB	Type gp3	Status check Okay
AWS Compute Optimizer finding <small>Opt-in to AWS Compute Optimizer for recommendations.   Learn more</small>	Volume state In-use	IOPS 3000	Throughput 125
Fast snapshot restored No	Availability Zone use1-az4 (us-east-1a)	Created Sat Feb 07 2026 14:40:01 GMT+0530 (India Standard Time)	Multi-Attach enabled No

```
[ec2-user@ip-10-0-1-48 ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1    259:0   0  20G  0 disk
└─nvme0n1p1 259:1   0  20G  0 part /
└─nvme0n1p27 259:2   0   1M  0 part
└─nvme0n1p128 259:3   0 10M  0 part /boot/efi
nvme1n1    259:4   0  20G  0 disk
└─nvme1n1p1 259:5   0  20G  0 part
└─nvme1n1p27 259:6   0   1M  0 part
└─nvme1n1p128 259:7   0 10M  0 part
[ec2-user@ip-10-0-1-48 ~]$ sudo mkdir /mnt/recovery
```

```
[ec2-user@ip-10-0-1-48 ~]$ sudo mount -t xfs -o ro,nouuid /dev/nvme1n1p1 /mnt/recovery
[ec2-user@ip-10-0-1-48 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/tmpfs       4.0M   0  4.0M  0% /dev
tmpfs           459M   0  459M  0% /dev/shm
tmpfs           184M  448K 183M  1% /run
/dev/nvme0n1p1   20G  1.6G  19G  8% /
tmpfs           459M   0  459M  0% /tmp
/dev/nvme0n1p128 10M  1.3M  8.7M 13% /boot/efi
tmpfs            92M   0   92M  0% /run/user/1000
/dev/nvme1n1p1   20G  1.5G  19G  8% /mnt/recovery
[ec2-user@ip-10-0-1-48 ~]$ █
```

## 7. Backup and Recovery Strategy

- Regular EBS snapshots
- Manual restoration in case of failure
- Ensures business continuity

## 8. Challenges Faced

- Learning AWS networking concepts
- Understanding security groups and access control
- Manual configuration without automation tools

## 9. Conclusion

This project demonstrates how a small enterprise can migrate from decentralized systems to a centralized, cloud-based infrastructure using AWS. The setup improves reliability, security, and manageability while staying within free-tier constraints.

## Future Enhancements

- Auto Scaling and Load Balancer
- S3-based backups
- CloudWatch monitoring
- IAM role-based access control