



PROJECT2:

AWS Routing & Switching using CentOS Server Administration

15.10.2024

Prepared By:

K. Dushyant Reddy

Gaurav Gupta

Mainak Roy

Mansi Singh

Anmol Baghel

Anuja Dixit

Guided By:

Zakir Hussain



INDEX

S.NO.	TABLE OF CONTENTS	PAGES
1.	Objective	03
2.	Introduction	04
3.	Services Used	05-07
4.	Architecture Design	08 - 09
5.	Implementation Steps	10-178
6.	Real-Time Scenario	189-1920
7.	Future Scope	201
8.	Conclusion	212

OBJECTIVE

The primary objective of implementing and maintaining a switch and routing network in AWS using CentOS server administration for a centralized large network is to establish a robust, secure, and highly available network infrastructure that can efficiently manage and monitor all traffic flowing through the environment. This centralized network architecture aims to provide a single pane of glass for administrators to control and inspect all ingress and egress traffic, thereby enhancing network security and reducing the risk of unauthorized access.

Additionally, this architecture enables better monitoring and troubleshooting capabilities, allowing administrators to quickly identify and resolve issues, thereby minimizing downtime and ensuring high network availability. Overall, the objective is to create a scalable, flexible, and secure network infrastructure that can support the growing demands of a large network, while also providing a solid foundation for future network expansion and development.

INTRODUCTION

Implementing and maintaining a switch and routing network in AWS using CentOS server administration is crucial for a large-scale network. Amazon Web Services (AWS) provides a robust platform for building and managing a switch and routing network, and CentOS server administration can be used to centrally manage and monitor the network infrastructure. The benefits of implementing a switch and routing network in AWS include scalability, security, high availability, and centralized management. A virtual private cloud (VPC) is a virtual network dedicated to the AWS account, providing a secure and isolated environment for the network infrastructure. Subnets, route tables, switches, routers, and EC2 instances running CentOS are key components of a switch and routing network in AWS.

CentOS server administration plays a vital role in centralized network management. It can be used to configure and manage network interfaces, routing tables, and firewall rules. Additionally, CentOS can be used to monitor and log network activity, providing real-time insights into network performance and security. Automation and scripting can also be used to automate network tasks and scripts, making it easier to manage and maintain the network infrastructure. By implementing a switch and routing network in AWS using CentOS server administration, organizations can build a scalable, secure, and highly available network infrastructure that can be centrally managed and monitored. This approach enables organizations to efficiently manage their network resources, reduce costs, and improve overall network performance.

In a large-scale network, a switch and routing network is essential for efficient communication, scalability, and security. AWS provides a highly scalable infrastructure that can easily adapt to growing network demands. The use of CentOS server administration provides a centralized management system, making it easier to troubleshoot and resolve issues. The combination of AWS and CentOS server administration provides a robust and secure network infrastructure that can meet the demands of a large-scale network. By leveraging the benefits of AWS and CentOS server administration, organizations can build a network infrastructure that is highly available, scalable, and secure.

SERVICES USED

Compute Services

EC2 (Elastic Compute Cloud): A virtual machine service that allows you to run your own applications and operating systems on Amazon's cloud infrastructure. You can choose from a variety of instance types, configure security and networking, and scale your instances as needed.

AMI (Amazon Machine Image): A pre-configured template that contains the operating system, applications, and configuration settings required to launch an EC2 instance. You can create your own custom AMIs or use public AMIs provided by AWS.

Scalability and Load Balancing

Autoscaling: A service that automatically adds or removes EC2 instances based on demand, ensuring that your application has the necessary resources to handle changes in traffic. You can configure scaling policies based on metrics such as CPU utilization, request latency, or queue length.

Application Load Balancer: A service that distributes incoming traffic across multiple EC2 instances, ensuring that no single instance is overwhelmed and becomes a bottleneck. You can configure load balancers to route traffic based on application-specific logic, such as URL paths or query parameters.


Networking

VPC (Virtual Private Cloud): A virtual network that allows you to launch AWS resources, such as EC2 instances, into a virtual network that you define. You can configure subnets, route tables, and security groups to control traffic flow and security.

Public Subnet: A subnet that has a route to the Internet Gateway, allowing instances launched in that subnet to access the Internet.

Private Subnet: A subnet that does not have a route to the Internet Gateway, and is therefore isolated from the Internet.

Database Services



RDS (Relational Database Service): A managed relational database service that supports popular database engines such as MySQL, PostgreSQL, and Oracle. You can create a primary database instance and one or more read replicas to improve performance and availability.

RDS Read Replica: A read-only copy of your primary database instance, which can be used to offload read traffic and improve performance.

Networking and Security

Transit Gateway: A service that enables you to connect multiple VPCs and on-premises networks to a single gateway, simplifying network architecture and improving security.

Transit Attachment: A connection between a VPC or on-premises network and a Transit Gateway.

Monitoring and Logging

CloudWatch: A monitoring and logging service that provides metrics, logs, and alarms for your AWS resources. You can use CloudWatch to monitor performance, detect anomalies, and troubleshoot issues.

VPC Flow Log: A feature that captures information about the IP traffic going to and from network interfaces in your VPC. You can use VPC Flow Logs to monitor traffic patterns, detect security threats, and troubleshoot connectivity issues.

Security and Identity

IAM Role: A role that defines a set of permissions that can be assumed by an AWS service or user. You can use IAM roles to grant access to AWS resources and services.

CloudAlarm: A service that provides real-time monitoring and alerting for your AWS resources. You can use CloudAlarm to detect security threats, performance issues, and other anomalies.

Storage

S3 Bucket: A storage container that can hold objects such as files, images, and videos. You can use S3 buckets to store and serve static content, backup data, and more.



Cost Management

AWS Budget: A service that allows you to set budget thresholds and receive alerts when your costs exceed those thresholds. You can use AWS Budget to monitor and control your AWS costs.

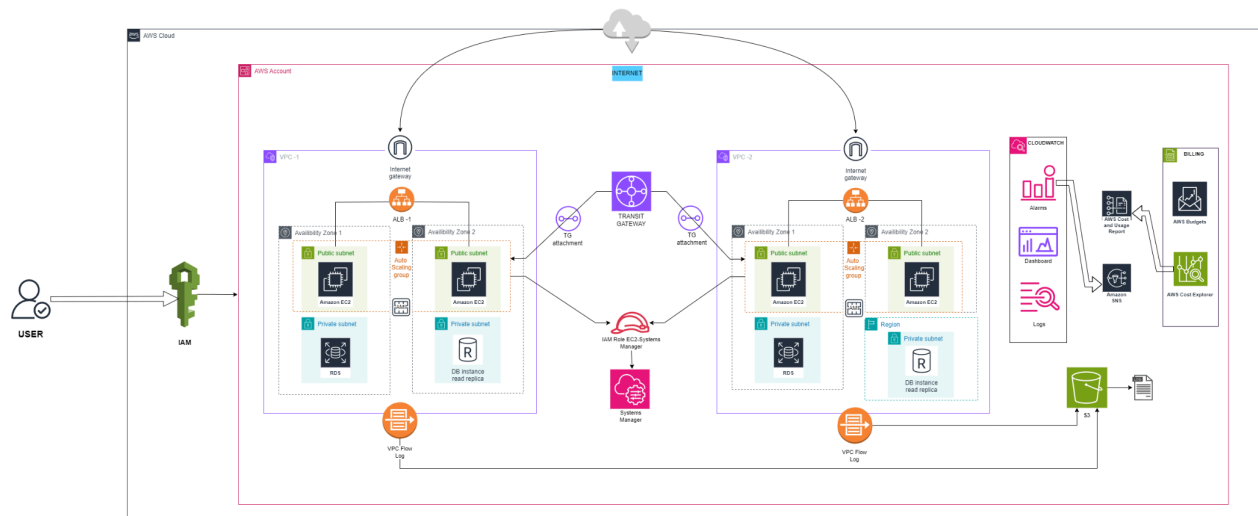
AWS Cost Explorer: A service that provides detailed cost and usage reports for your AWS resources. You can use AWS Cost Explorer to analyze your costs, identify trends, and optimize your resource utilization.

Other Services

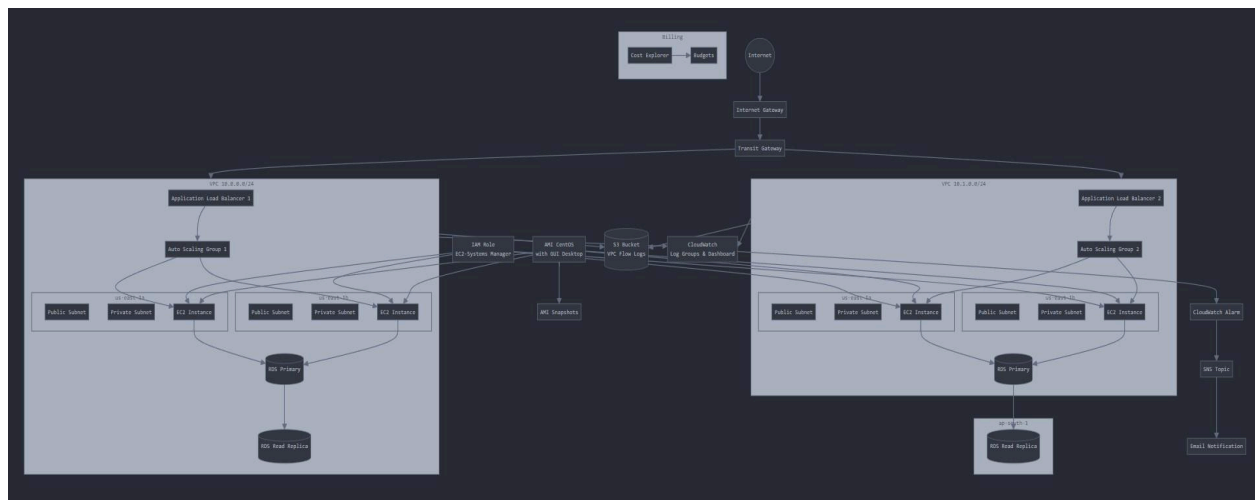
Internet Gateway: A service that connects your VPC to the Internet, allowing instances in your VPC to access the Internet.

SNS (Simple Notification Service): A service that provides a scalable and flexible way to send notifications to users and applications. You can use SNS to send alerts, notifications, and other messages.

ARCHITECTURE DESIGN



AWS Routing & Switching Server Configuration Architecture Diagram



AWS Routing & Switching Server Configuration Flow Diagram

This diagram illustrates a basic AWS architecture with an Application Load Balancer serving as the entry point to the system. Behind the Load Balancer, an Auto Scaling Group provides high availability by scaling the number of EC2 instances up or down based on demand.

The EC2 instances, running the AMI CentOS with GUI Desktop, are launched in a private subnet for security purposes and communicate with a database hosted by an RDS instance. The RDS instance is configured as a primary database with a read replica for redundancy and high availability.



Here's a breakdown of the architecture's key components and their roles:

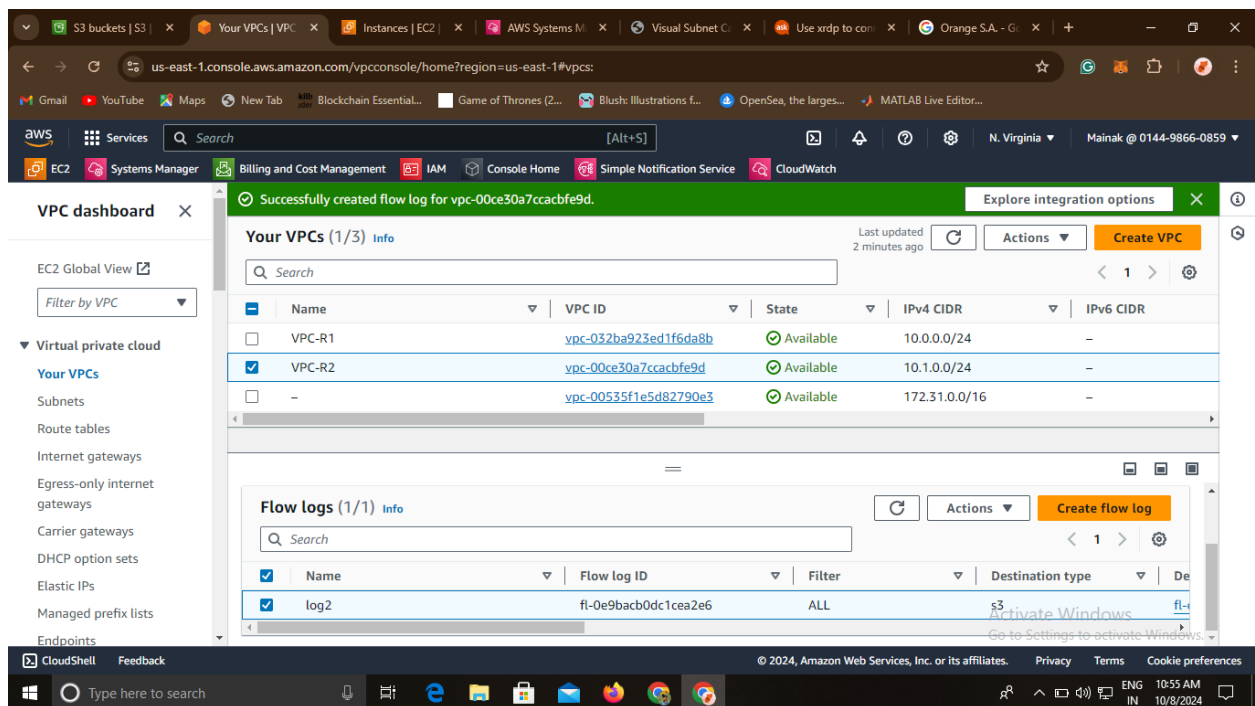
- Internet Gateway: Serves as the entry point for traffic from the internet.
- Transit Gateway: This is a highly scalable and efficient way to connect multiple VPCs. It's used to connect two VPCs in this diagram.
- Application Load Balancer (ALB): The main entry point for incoming traffic. It distributes incoming traffic to healthy EC2 instances behind it.
- Auto Scaling Group: Dynamically manages the number of EC2 instances running to ensure performance and availability.
- EC2 Instances: The application servers that handle requests from the ALB. They are launched in a private subnet for enhanced security.
- Private Subnets: Subnet designed to secure internal applications and data. EC2 instances are launched in this subnet.
- Public Subnet: Subnet designed for internet-facing services. In this example, it's empty.
- RDS Primary: The primary database, responsible for writing and receiving updates.
- RDS Read Replica: A read-only replica of the primary database, offering improved read performance and availability.
- CloudWatch: AWS's monitoring service that tracks various metrics. It's used to trigger alarms and send notifications.
- CloudWatch Alarm: Configured to monitor key metrics in the system and trigger actions, like sending notifications to an SNS topic, when thresholds are breached.
- SNS Topic: A communication channel for sending notifications.
- Email Notification: Receives alerts from the SNS topic.

IMPLEMENTATION STEPS

, Here are the implementation steps for setting up the network in AWS:

1. VPC and Subnet Configuration

- Create two VPCs (e.g., VPC 10.0.0.0/24 and VPC 10.1.0.0/24) to separate the environments. This will allow for better isolation and control over network resources.



- Configure subnets within each VPC:

- Create public and private subnets in multiple Availability Zones (AZs) within each VPC to distribute resources across different zones for high availability.

- For each VPC, include at least one public subnet for internet-facing components and one private subnet for internal services.

Your VPCs (1/2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main
VPC-R1	vpc-08faf3b03b4376287	Available	10.0.0.0/24	-	dopt-Od128a97dc62e7...	rtb-0f4b6f26d029c7f3
-	vpc-00535f1e5d82790e3	Available	172.31.0.0/16	-	dopt-Od128a97dc62e7...	rtb-0f4b6f26d029c7f3

Resource map Info

VPC-R1

Subnets (2)
Subnets within this VPC

- us-east-1a
 - PUB-AZ-1
 - PRI-AZ-1

Route tables (2)
Route network traffic to resources

- rtb-0f4b6f26d029c7f3
- RT-1

Network connections (1)
Connections to other networks

- IGW-1

Your VPCs (1/3) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main
VPC-R1	vpc-08faf3b03b4376287	Available	10.0.0.0/24	-	dopt-Od128a97dc62e7...	rtb-0f4b6f26d029c7f3
-	vpc-00535f1e5d82790e3	Available	172.31.0.0/16	-	dopt-Od128a97dc62e7...	rtb-0f4b6f26d029c7f3

VPC-R2

Subnets (2)
Subnets within this VPC

- us-east-1c
 - PUB-AZ-2
- us-east-1d
 - PRI-AZ-2

Route tables (2)
Route network traffic to resources

- RT-2
- rtb-00020f89a7f731ea3

Network connections (1)
Connections to other networks

- IGW-2

2. Networking Gateways Setup

- Internet Gateway: Attach an internet gateway to both VPCs to enable public access for resources in the public subnets.

VPC dashboard

Internet gateway igw-0e0a6056a9f02551f successfully attached to vpc-032ba923ed1f6da8b

VPC > Internet gateways > igw-0e0a6056a9f02551f

igw-0e0a6056a9f02551f / IGW-1

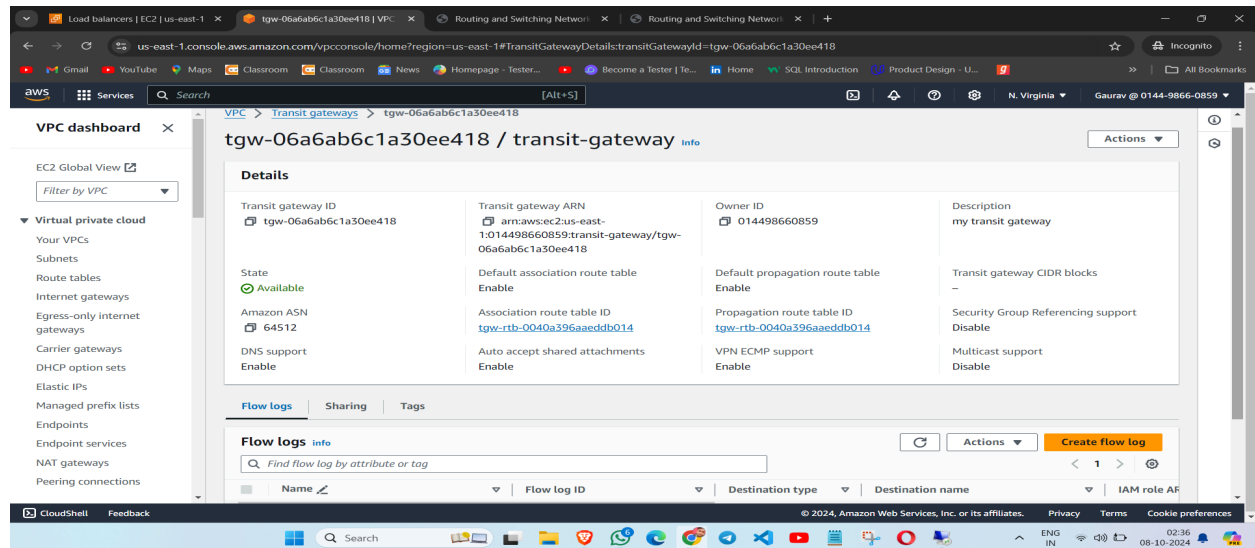
Details Info

Internet gateway ID	igw-0e0a6056a9f02551f	State	Attached	VPC ID	vpc-032ba923ed1f6da8b VPC-R1	Owner	014498660859
---------------------	-----------------------	-------	----------	--------	--------------------------------	-------	--------------

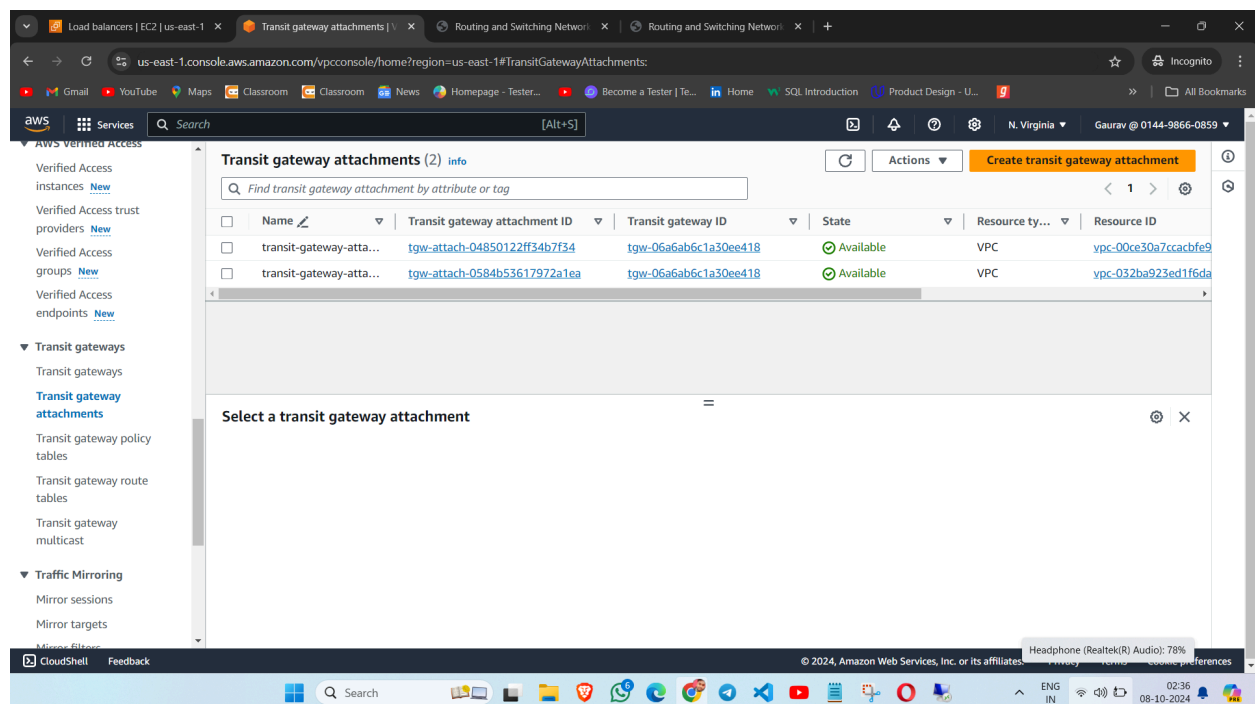
Tags

Key	Value
Name	IGW-1

- Transit Gateway: Set up a transit gateway to connect the two VPCs, allowing communication between them. Configure route tables to route traffic through the transit gateway.



Transit Attachment: A connection between a VPC or on-premises network and a Transit Gateway.



- Route Tables: Update the route tables to enable traffic routing between subnets, to/from the internet, and across VPCs via the transit gateway.

The screenshot shows the AWS VPC console interface. At the top, a green notification bar states: "You have successfully updated subnet associations for rtb-0ab273658cc758dcd / RT-1." The breadcrumb navigation is "VPC > Route tables > rtb-0ab273658cc758dcd". The main heading is "rtb-0ab273658cc758dcd / RT-1".

Details Info

Route table ID rtb-0ab273658cc758dcd	Main No	Explicit subnet associations 2 subnets	Edge associations -
VPC vpc-032ba923ed1f6da8b VPC-R1	Owner ID 014498660859		

Subnet associations

Explicit subnet associations (2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Pub-Sub-AZ2	subnet-0b5a3a51caae00e84	10.0.0.128/26	-
Pub-Sub-AZ1	subnet-01f963de3f417511d	10.0.0.0/26	-

The screenshot shows the AWS VPC console interface. At the top, a green notification bar states: "You have successfully updated subnet associations for rtb-0b1223f54deca2851 / RT-2." The breadcrumb navigation is "VPC > Route tables > rtb-0b1223f54deca2851". The main heading is "rtb-0b1223f54deca2851 / RT-2".

Details Info

Route table ID rtb-0b1223f54deca2851	Main No	Explicit subnet associations 2 subnets	Edge associations -
VPC vpc-00ce30a7ccacbf9d VPC-R2	Owner ID 014498660859		

Subnet associations

Explicit subnet associations (2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Pub-Sub-AZ1	subnet-0978f33978a873ada	10.1.0.0/26	-
Pub-Sub-AZ2	subnet-0d0c09f452011cb84	10.1.0.128/26	-

3. Security Configurations

- Configure Security Groups and Network ACLs: Set up security groups to allow or deny traffic based on the application requirements. Use Network ACLs for an additional layer of network security.

- IAM Roles and Policies: Create IAM roles and policies to control access to EC2 instances, RDS, and other AWS services. For example, an IAM role with permissions for EC2 Systems Manager should be attached to instances for remote management.

The screenshot shows the AWS IAM console 'Role details' page. The role name is 'EC2-SSM-RoId'. The description is 'Allows EC2 instances to call AWS services on your behalf.' The page includes a sidebar with navigation links for 'Step 2: Add permissions', 'Step 3: Name, review, and create', and a search bar at the top.

4. EC2 Instances Setup

- Launch EC2 instances in the private subnets for internal services and public subnets for instances that need to be accessible externally.

The screenshot shows the AWS Management Console 'Instances' page. It displays a table of EC2 instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. One instance, 'web-server-R1', is shown in the 'Running' state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
web-server-R1	i-0a66969f0aa161285	Running	t3.medium	Initializing	View alarms +	us-east-1a	-	34.235....

- Configure Auto Scaling Groups (ASG): Set up Auto Scaling Groups (ASG 1 and ASG 2) to handle traffic spikes by automatically scaling the number of EC2 instances based on demand.

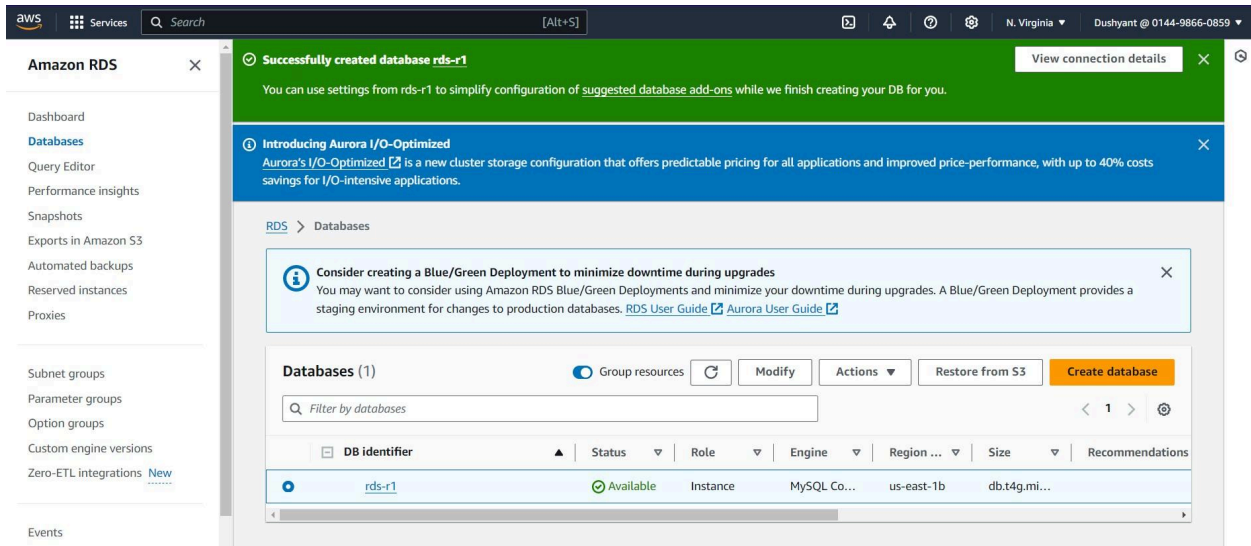
- Application Load Balancers (ALB): Deploy ALBs in front of the Auto Scaling Groups to distribute incoming requests evenly across the instances. ALB 1 and ALB 2 can be configured to handle traffic for each VPC.

The screenshot shows the AWS Management Console 'Load balancers' page. It displays a table of Application Load Balancers. The table has columns for Name, DNS name, State, VPC ID, Availability Zones, and Type. Two load balancers, 'elb-r2' and 'elb-r1', are shown in the 'Active' state.

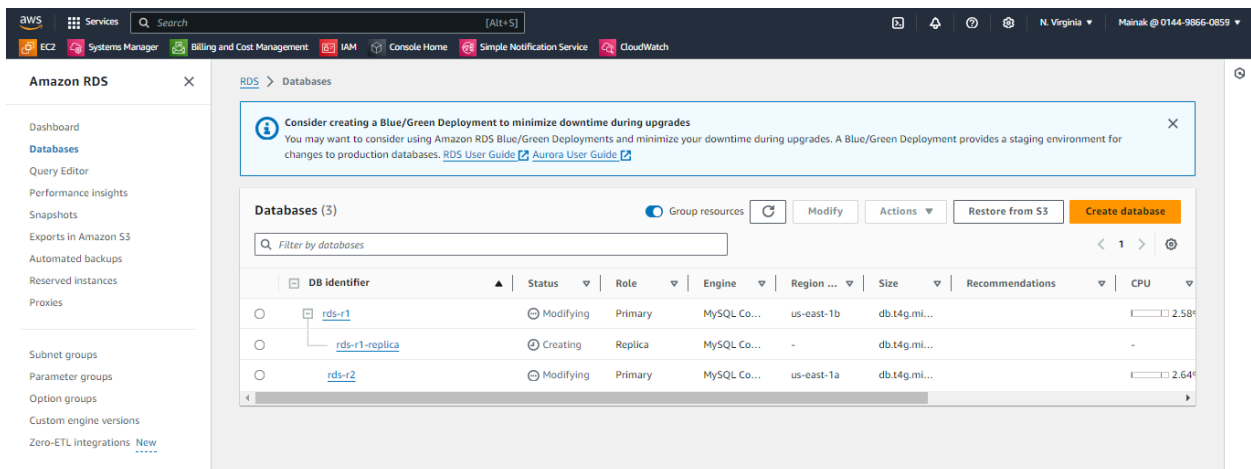
Name	DNS name	State	VPC ID	Availability Zones	Type
elb-r2	elb-r2-882009896.us-east-1.elb.amazonaws.com	Active	vpc-00ce30a7ccacbf9d	2 Availability Zones	application
elb-r1	elb-r1-1220113217.us-east-1.elb.amazonaws.com	Active	vpc-032ba923ed1f6da...	2 Availability Zones	application

5. Database Setup

- *Amazon RDS Deployment: Set up Amazon RDS instances for the primary database in each VPC, ensuring that they are in the private subnets for security reasons.

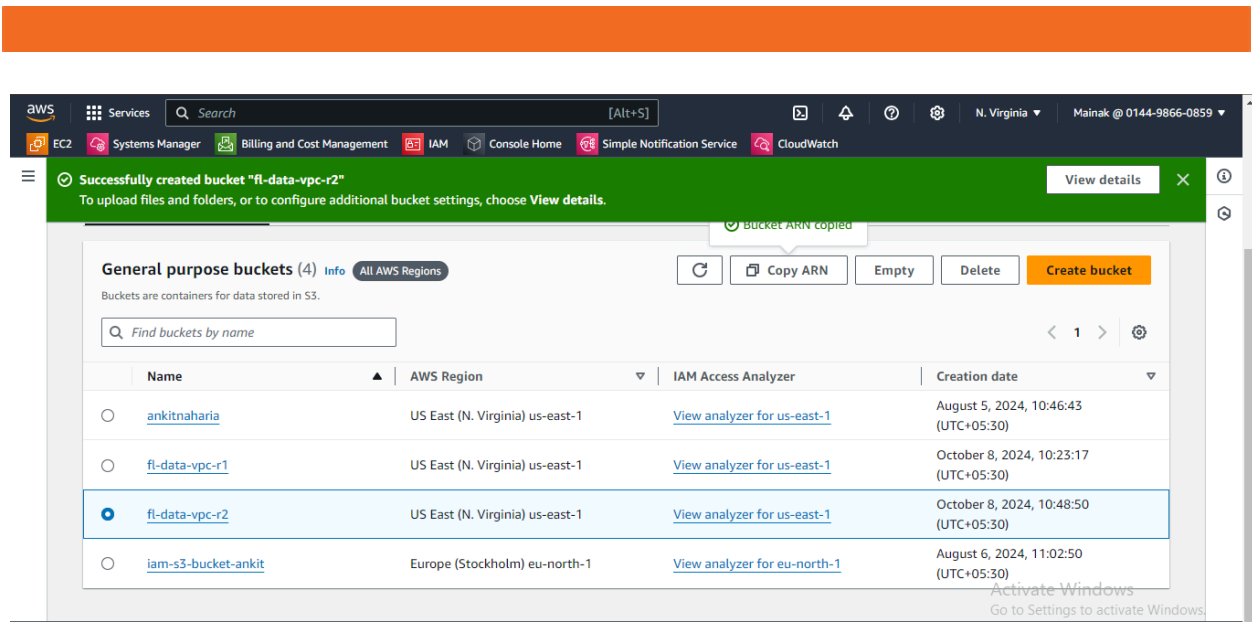


- *Read Replicas: Configure RDS read replicas in each VPC to offload read-heavy traffic from the primary database. This helps with scaling and ensures data availability.

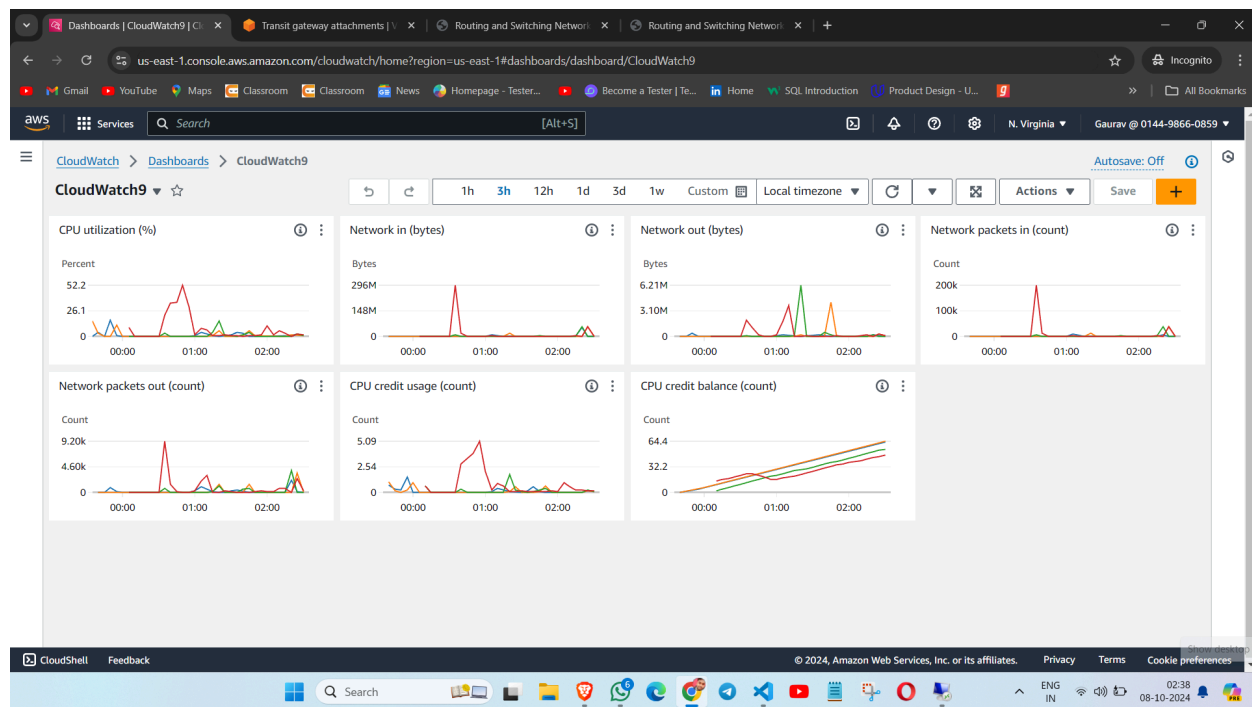


6. Logging and Monitoring

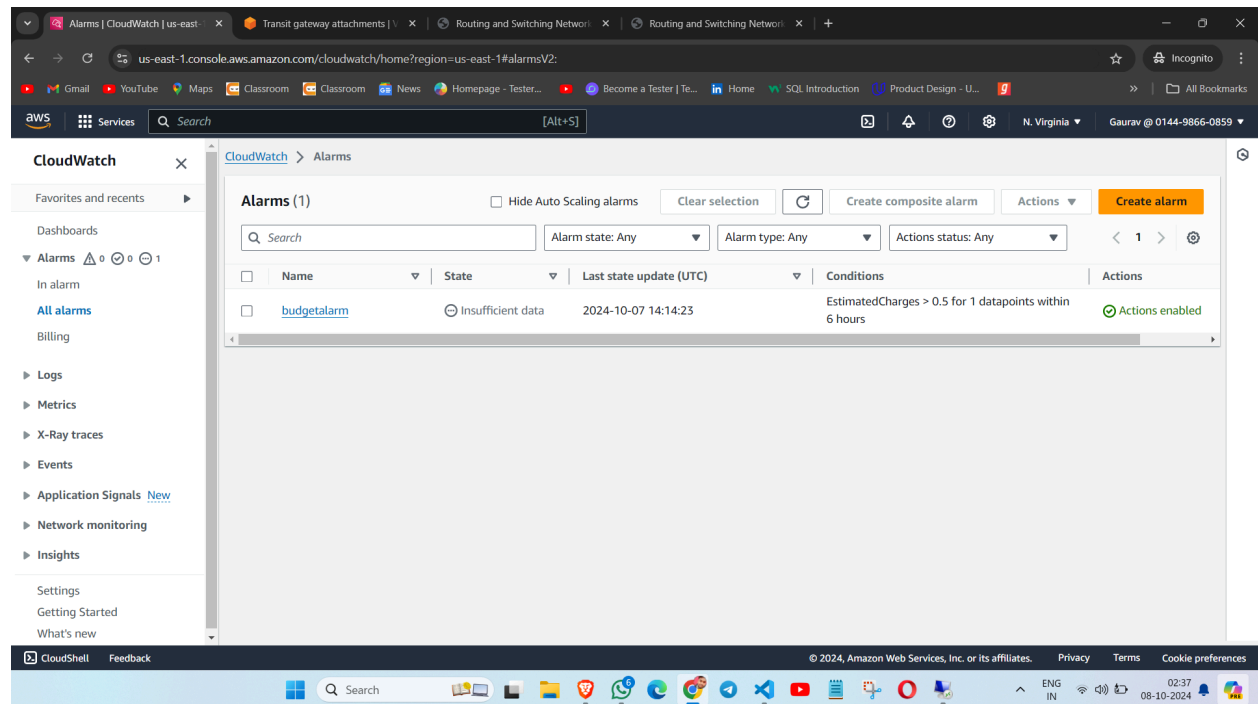
- VPC Flow Logs: Enable VPC flow logs to capture information about the traffic going in and out of the network interfaces in each VPC. Store the logs in an S3 bucket for analysis.



- CloudWatch Monitoring: Set up CloudWatch logs and dashboards to monitor the health of instances, load balancers, and databases. Configure alarms for resource utilization, downtime, or other anomalies.

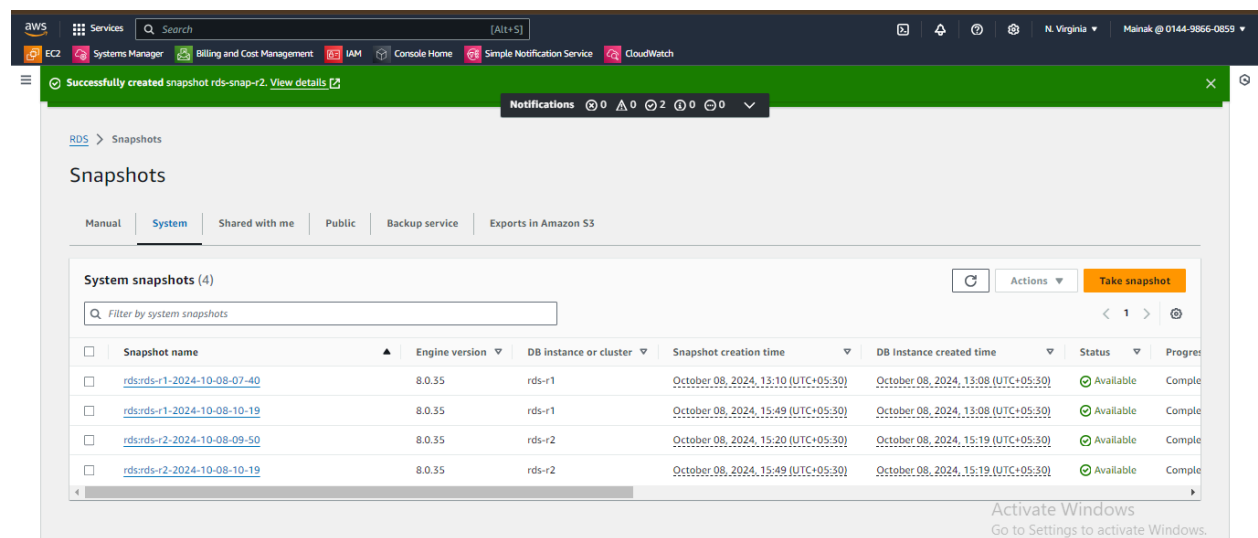


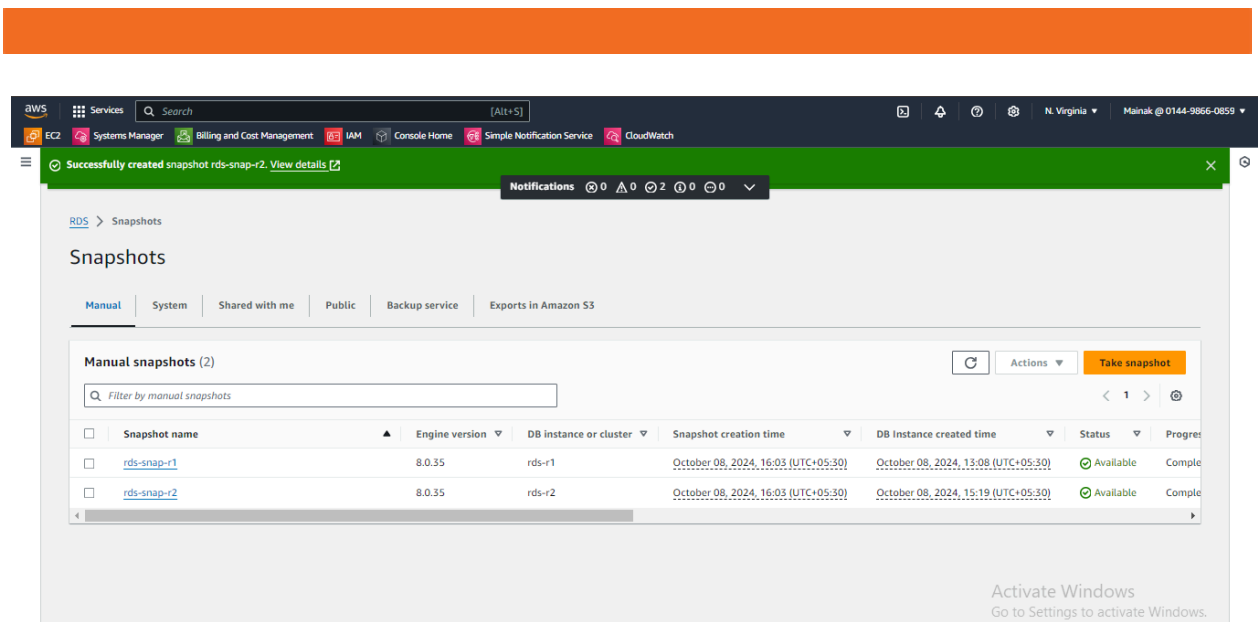
- CloudWatch Alarms and Notifications: Create CloudWatch alarms to monitor critical metrics (e.g., high CPU usage) and trigger notifications via SNS (Simple Notification Service) to send email alerts for any issues.



7. Backup and Recovery

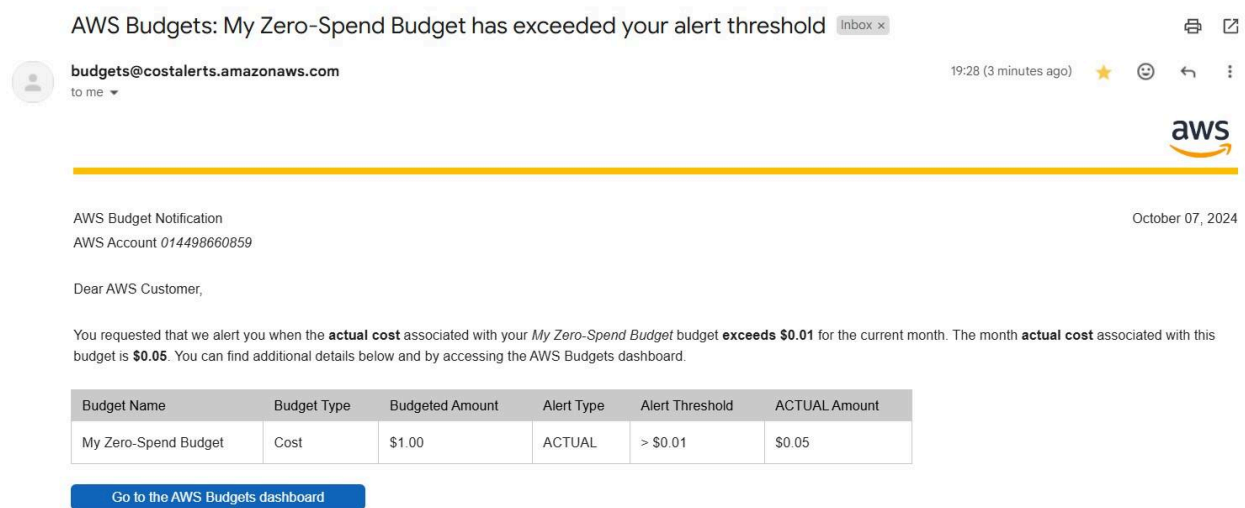
- EC2 Snapshots: Regularly create snapshots of EC2 instances for backup purposes. Store these in S3 or use them to create AMIs (Amazon Machine Images) for disaster recovery.
- Automate Backups for RD*: Enable automated backups and manual snapshots for the RDS databases to ensure data recovery in case of failure.





8. Cost Management

- Billing and Cost Management: Use AWS Cost Explorer and Budgets to monitor usage and control costs. Configure budgets and set alerts for when usage exceeds certain thresholds.



These steps will help establish a robust and secure network infrastructure as depicted in the diagram, ensuring scalability, high availability, and proper monitoring.

REAL-TIME SCENARIO

Scenario: Centralized Network for a Large E-Commerce Company

Company Background:


Founded in 2010, ShopAll is a rapidly growing e-commerce company that started as a small online retail store specializing in electronics. Over the years, it expanded its offerings to include clothing, home appliances, and beauty products. As the customer base grew globally, ShopAll invested heavily in its digital transformation to accommodate millions of daily users and support its expanding catalog. The company's infrastructure now spans multiple departments, including marketing, sales, customer support, development, and logistics. Each department relies on various applications and services to manage day-to-day operations, customer interactions, and business growth.

Objective:

The objective of the project is to establish a centralized network infrastructure in AWS to unify the company's fragmented systems and services into a cohesive and manageable network. The infrastructure will efficiently manage traffic across departments, ensure secure communication between services, and provide VPN access for remote employees, enabling seamless collaboration. This centralized approach aims to enhance network security, improve monitoring capabilities, and support future growth.

Challenges Faced by ShopAll Before Centralization:

Siloed Infrastructure: Each department had its own set of services hosted on different cloud providers and on-premises servers, leading to inefficiencies, integration issues, and high maintenance costs.



Security Concerns: With no unified security framework, the company faced risks of data breaches, as different teams managed separate security protocols and policies.

High Operational Costs: Running multiple fragmented systems resulted in excessive costs for hardware, software licenses, and maintenance.

Scalability Issues: Handling increased traffic during sales events like "Black Friday" was challenging due to the limited scalability of the existing on-premises servers.

Complex Troubleshooting: With systems scattered across different environments, identifying and resolving network issues often took longer, causing potential revenue loss due to downtime.

Proposed Solution:

The solution involves implementing a centralized network using AWS services to consolidate all departmental infrastructures, allowing ShopAll to efficiently manage traffic, enhance security, and support scalability.

Benefits of the Centralized Network for ShopAll:

Increased Security: By unifying security policies and consolidating resources within AWS, the risk of breaches is significantly reduced.

Operational Cost Savings: With a pay-as-you-go model, ShopAll can eliminate the expenses associated with maintaining separate infrastructures.

Scalability for Growth: The new architecture supports seamless scaling during peak times, ensuring that users experience consistent performance.

Improved Troubleshooting: Centralized logging and monitoring allow for quicker identification and resolution of issues.

Efficient Collaboration: A unified infrastructure ensures that all departments can easily share data

FUTURE SCOPE

The future goals of this project can evolve to address changing business requirements, adopt new technologies, and maintain a competitive edge while ensuring a stable and scalable network infrastructure.

Future Goals for the Centralized Network Project

Global Expansion: Extend the network to new regions for better performance and international reach.

Enhanced Security: Integrate advanced security tools and automate compliance checks.

Cost Optimization: Utilize cost-monitoring tools and automate resource scheduling.

AI Integration: Leverage machine learning for traffic analysis, anomaly detection, and predictive scaling.

Automated Response: Implement self-healing infrastructure and automated incident handling.

Hybrid and Multi-Cloud Support: Expand to support hybrid environments with seamless cloud integrations.

Monitoring Improvements: Enhance monitoring and logging for better issue detection.

DevOps Support: Automate deployments and updates through CI/CD and infrastructure-as-code.

These goals aim to enhance scalability, security, and cost-efficiency while supporting future growth.

CONCLUSION

In conclusion, the project's implementation of a secure, scalable, and highly available network infrastructure in AWS using CentOS provides a robust foundation for supporting a centralized large network. The architecture not only ensures optimal performance, security, and cost-efficiency but also facilitates rapid fault detection and resolution to minimize downtime. With strategies for future growth, including scaling, enhanced segmentation, integration of new services, and global expansion, the network is well-equipped to adapt to evolving business needs and traffic demands. This approach ensures long-term sustainability, enabling the organization to efficiently manage current operations while seamlessly accommodating future growth. This project helps mitigate real-time issues by providing rapid fault detection, minimizing downtime, and enabling quick recovery from failures, thus maintaining network performance and security.