



PROJECT2:

AWS Routing & Switching using CentOS Server Administration

16.10.2024

—

Prepared By:

K. Dushyant Reddy



Guided By:

Zakir Hussain



INDEX

S.NO.	TABLE OF CONTENTS	PAGES
1.	Objective	04
2.	Introduction	05 - 06
3.	Prerequisites	07 - 08
4.	Real-Time Scenario	09 - 10
5.	Architecture Design	11 - 12
6.	Implementation Steps	13 - 73
7.	Project Output	74 - 77
7.	Future Scope	78
8.	Conclusion	79

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.

PREREQUISITES

Basic knowledge on -

1. **Amazon VPC (Virtual Private Cloud)** - Isolated virtual network environment within AWS, allowing customization of network settings and security configurations.

Subnet - Segments of a VPC, dividing the network into smaller, manageable parts. Subnets can be public (accessible from the internet) or private (isolated from the internet).
2. **IAM(Identity and Access Management)** - solution that enables to manage users' access to AWS resources.
3. **Amazon EC2 (Elastic Compute Cloud)** - Web service which provides resizable compute capacity in the cloud.

Instance - Virtual computing environments
4. **AWS RDS (Relational Database Service)** - Managed relational database service, providing easy setup, scaling, and maintenance of SQL databases like MySQL, PostgreSQL, etc.
5. **Elastic Load Balancing** - manages the workload on the instances and distributes them to other instances in case of an instance failure.

Application Load Balancers - Ideal for routing HTTP/HTTPS traffic and performing advanced traffic routing and content-based routing.
6. **Auto Scaling** - helps you maintain application availability and allows you to automatically add or remove Amazon EC2 instances according to conditions you define.

7. **AMI (Amazon Machine Image)** - an image that provides the software that is required to set up and boot an Amazon EC2 instance.
8. **AWS Transit Gateway** - a network transit hub that you can use to interconnect your virtual private clouds (VPCs) and on-premises networks
9. **AWS Systems Manager** - helps to manage applications and infrastructure running in the AWS. It simplifies application and resource management, shortens the time to detect, resolve operational problems, and helps to manage AWS resources securely at scale
10. **Amazon S3 (Simple Storage Service)** - a highly scalable object storage service used for a wide range of storage solutions, including websites, mobile applications, backups, and data lakes.
11. **Amazon SNS (Simple Notification Service)** - a managed service that provides message delivery from publishers to subscribers
12. **Amazon CloudWatch** - used to collect and track metrics, which are variables that can be measured for the resources and application.
13. **AWS Cost Explorer** - a tool that enables you to view and analyze your costs and usage
14. **Budgets** - lets you set custom cost and usage budgets that alert you when your budget thresholds are exceeded (or forecasted to exceed)
15. **AWS Cost and Usage Report** - tracks your AWS usage and provides estimated charges associated with your account.

REAL-TIME SCENARIO

Scenario: Centralized Network for a Large E-Commerce Company

Company Background:

Founded in 2010, TechRev 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, TechRev 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.

Current Scenario:

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. Problems faced:

- 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.

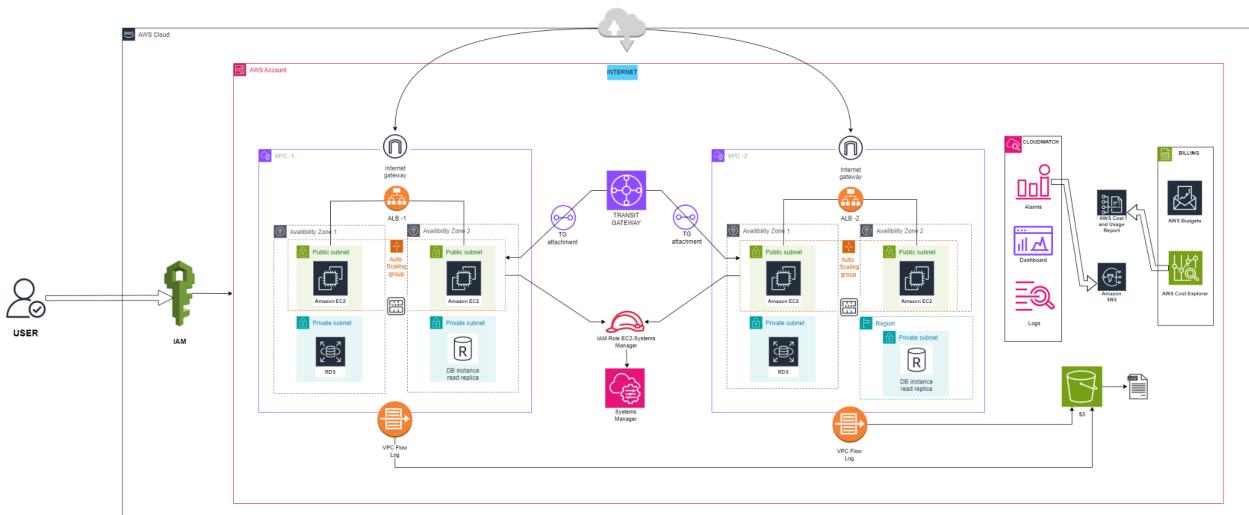
Solution:

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

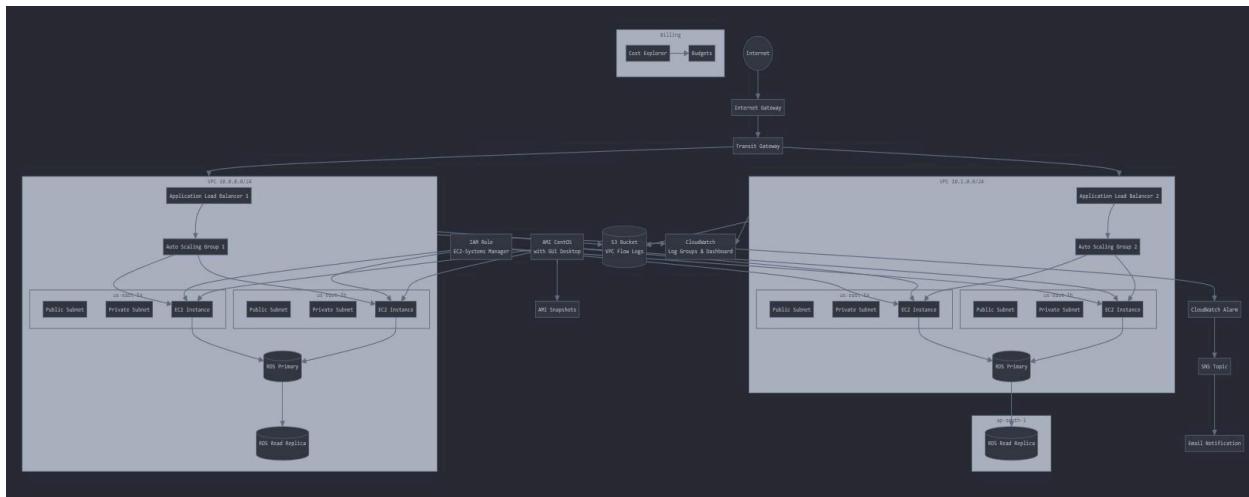
Benefits:

- 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, TechRev 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.

ARCHITECTURE DIAGRAM



AWS Routing & Switching Server Configuration Arch Diagram



AWS Routing & Switching Server Configuration Flow Diagram

Data Flow:

This architecture represents an 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: Subnets 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

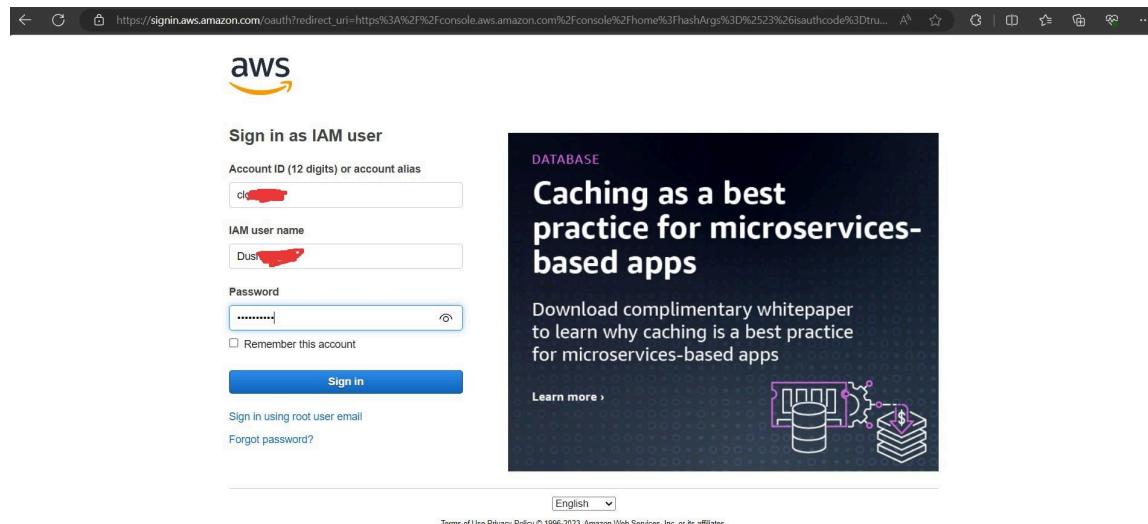
1. Console Login Configuration

Console Login Configuration:

- Go to <https://aws.amazon.com/console/> and click on "Sign in to the Console".
- If you're using an AWS root account, enter your email address associated with the account. If you're using an IAM user account, select "IAM user" and enter your account ID or account alias.
- Once logged in, you'll be directed to the AWS Management Console, where you can manage your AWS services.

Step 1: Login into AWS Console

1.1 Create an account and login into it by providing the credentials.

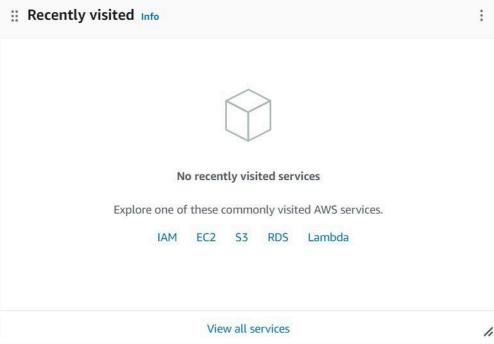


1.2 Give access to AWS Dashboard

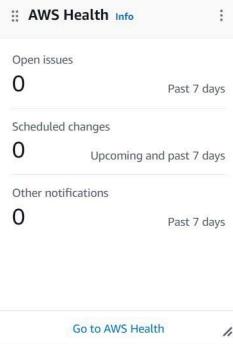
AWS Services Search [Alt+S] Mumbai Dash [REDACTED]

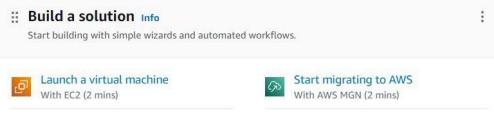
Console Home [Info](#)

[Reset to default layout](#) [+ Add widgets](#)


Recently visited [Info](#)
No recently visited services
Explore one of these commonly visited AWS services.
[IAM](#) [EC2](#) [S3](#) [RDS](#) [Lambda](#)
[View all services](#)


Welcome to AWS [Info](#)
Getting started with AWS [Info](#)
Learn the fundamentals and find valuable information to get the most out of AWS.
 Training and certification [Info](#)
Learn from AWS experts and advance your skills and knowledge.
 What's new with AWS? [Info](#)
Discover new AWS services, features, and Regions.
[Go to AWS Health](#)


AWS Health [Info](#)
Open issues 0 Past 7 days
Scheduled changes 0 Upcoming and past 7 days
Other notifications 0 Past 7 days


Cost and usage [Info](#)

Build a solution [Info](#)
Start building with simple wizards and automated workflows.


Launch a virtual machine With EC2 (2 mins)


Start migrating to AWS With AWS MGN (2 mins)

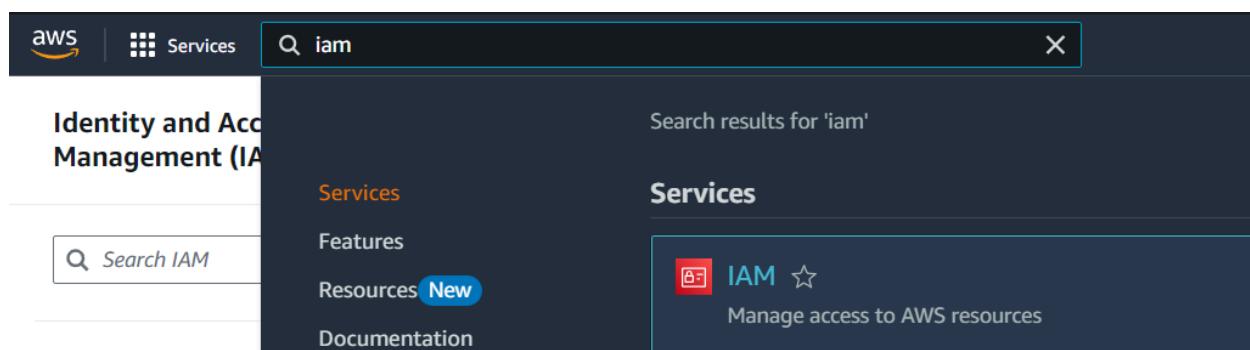
2. IAM Configuration

IAM Groups and Users Configuration:

- In the IAM dashboard, select **User groups** from the left sidebar. Enter a name for the group. Attach policies to the group to define permissions
- Repeat the steps above to create 2 more groups.
- **Groups:** P2-Networking, P2-Infrastructure and P2-Monitoring
- In the IAM dashboard, select **Users** from the left sidebar. Enter a username and choose the access type AWS Console access.
- Add the user to one of the groups you created earlier.
- Repeat the steps above to create 6 more users.
- **Users:** Anuja, Anmol, Mansi, Gaurav, Mainak and Dushyant
- Then go to the Role section and create a role for EC2 and Systems Manager and give permission for AmazonSSMManagedInstanceCore and click Create Role.

Step 2: Launch IAM in Console

2.1 In Dashboard, search for IAM service, click **IAM** to open it.



2.2 Under IAM resources, user groups and users are being assigned.



Screenshot of the AWS IAM Dashboard. The sidebar shows navigation options like Dashboard, Access management, and Access reports. The main area displays security recommendations (Add MFA for root user, Add MFA for yourself, Your user, Dushyant, does not have any active access keys that have been unused for more than a year) and IAM resources (User groups: 4, Users: 8, Roles: 11, Policies: 1, Identity providers: 0).

2.3 Created 6 users - Dushyant, Mainak, Mansi, Gaurav, Anuja and Anmol and added the permissions policies individually.

Screenshot of the AWS IAM Users page. It shows a list of 8 users: ankit, ankit_s3, Anmol, Anuja, Dushyant, Gaurav, Mainak, and Mansi. Each user has their last activity, MFA status, password age, and console sign-in information.

Screenshot of the AWS IAM Permissions policies page. It lists 9 policies: IAMFullAccess, AmazonSNSFullAccess, AmazonVPCFullAccess, CloudWatchActionsEC2Access, CloudWatchEventsFullAccess, AWSCostAndUsageReportAutomation, AWSBudgetsActions_RolePolicyForRe..., and AWSSystemsManagerEnableConfigRe... All policies are AWS managed and attached via Group P2-Networking or P2-Monitoring.

2.4 Created 3 user groups - P2-Infrastructure, P2-Monitoring and P2-Networking and attached the users with the policies

The screenshot shows two screenshots of the AWS IAM console.

IAM > User groups:

- User groups (4) Info:** A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.
- Table:**| Group name | Users | Permissions | Creation time |
| --- | --- | --- | --- |
| ankit_s3_ec2 | 2 | Defined | 2 months ago |
| P2-Infrastructure | 3 | Defined | 7 days ago |
| P2-Monitoring | 2 | Defined | 7 days ago |
| P2-Networking | 4 | Defined | 7 days ago |

2.6 Create a Role for connection of EC2 to Systems Manager.

The screenshot shows the AWS IAM Roles creation process at Step 1: Select trusted entity.

Step 1 Select trusted entity

Select trusted entity type:

- AWS service** (selected): Allows AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**: Allows entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation**: Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**: Create a custom trust policy to enable others to perform actions in this account.

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Step 1: Select trusted entity'. The 'Trusted entity type' section is displayed, with 'AWS service' selected. Other options like 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy' are also shown. The sidebar on the left lists 'Step 2: Add permissions' and 'Step 3: Name, review, and create'. The bottom of the screen shows the Windows taskbar with various icons and system status.

2.7 Assign the permissions policies

The screenshot shows the 'Create role' wizard in the AWS IAM console, specifically the 'Add permissions' step. The 'Permissions policies' section is displayed, showing a search bar with 'ssm' entered and a list of AWS managed policies. The 'AmazonSSMManagedInstanceCore' policy is selected, indicated by a checked checkbox. The sidebar on the left shows 'Step 1: Select trusted entity' and 'Step 3: Name, review, and create'. The bottom of the screen shows the Windows taskbar.

2.8 Enter the Role name and add the trust policy for so.

The screenshot shows a browser window for the AWS IAM console at us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?trustedEntityType=AWS_SERVICE&selectedService=EC2&selectedRoleType=ServiceRole. The page is titled "Role details".

Step 2: Add permissions

Step 3: Name, review, and create

Role name: EC2-SSM-Rol[
Maximum 64 characters. Use alphanumeric and "+-,.,@-_ " characters.

Description: Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: "+-,.,@-_ ".

Step 1: Select trusted entities

Trust policy:

```
1+ [ {  
2+   "Version": "2012-10-17",  
3+   "Statement": [  
4+     {  
5+       "Effect": "Allow",  
6+       "Action": [  
7+         "sts:AssumeRole"  
8+       ],  
9+       "Principal": {  
10+         "Service": [  
11+           "ec2.amazonaws.com"  
12+         ]  
13+       }  
14+     }  
15+   ]  
16+ }]
```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Type here to search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 8:12 PM IN 10/6/2024

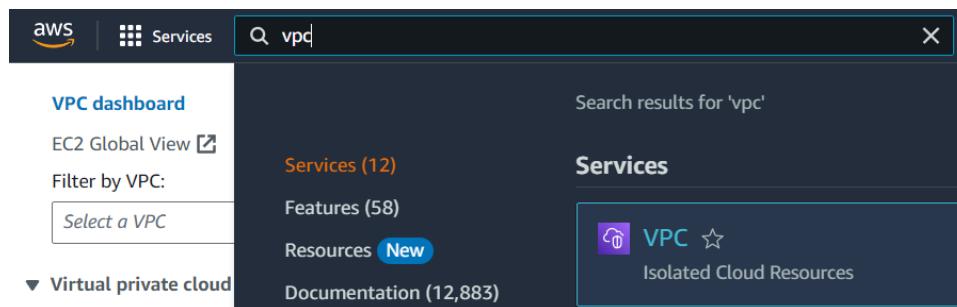
3. VPC Configuration

VPCs Configuration:

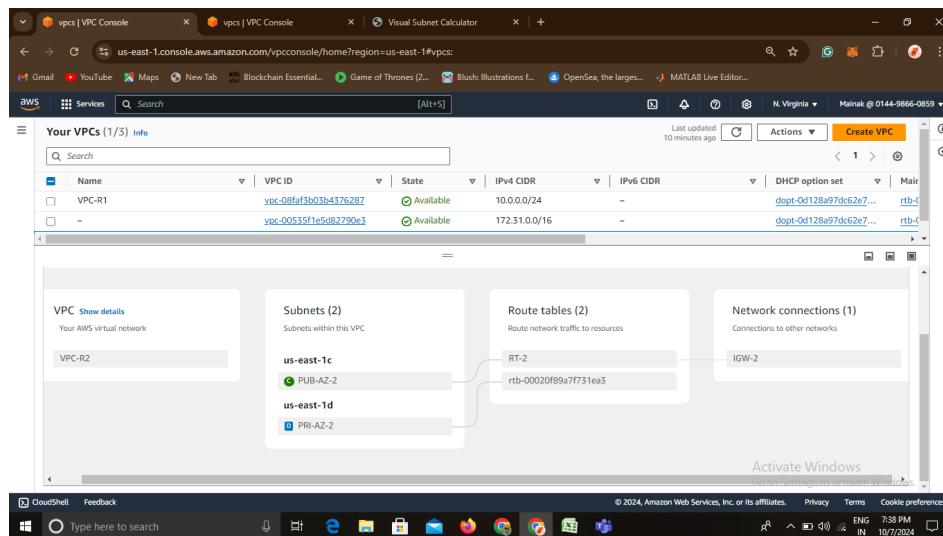
- In the VPC dashboard, click on Create VPC. Enter a name and add a IPv4 CIDR block of **10.0.0.0/24**. Then create 2 public and 2 private subnets in diff AZs.
- Create Internet Gateway and NAT Gateway and attach to VPC and associate the IG and NG in the Route Tables resp.
- Similarly create one more VPC of IPv4 CIDR block of **10.1.0.0/24** and configure it.

Step 3: Launch VPC in Console

3.1 In Dashboard, search for VPC service, click **VPC** to open it.



3.2 Click **Create VPC** to configure.



3.3 Type the **Name** which you prefer for naming the VPC and allocate the IPv4 CIDR block as **10.0.0.0/24**.

The screenshot shows the AWS VPC Console interface. At the top, there's a search bar and a 'Create VPC' button. Below it, a table lists two VPCs: 'VPC-R1' and another one with a partially visible ID. The 'Resource map' section shows a network diagram with nodes: 'VPC Show details', 'Subnets (2)', 'Route tables (2)', and 'Network connections (1)'. Arrows indicate connections between these components. The 'Subnets' node shows two subnets: 'PUB-AZ-1' and 'PRI-AZ-1' under the 'us-east-1a' availability zone.

3.4 Now, select the subnets in the left pane and create **4 Subnets**, two Public for EC2 and other two Private for RDS with CIDR as **10.0.0.0/26**, **10.0.0.128/26** and **10.0.0.64/26**, **10.0.0.92/26** respectively.

The screenshot shows the 'Create subnet' wizard. It starts with a 'VPC' selection step where 'VPC-R1' is chosen from a dropdown. The next step, 'Associated VPC CIDRs', shows an IPv4 CIDR of '10.0.0.0/24' selected. The wizard has three more steps: 'Configure route tables', 'Configure security groups', and 'Review and launch'.

Subnet 1 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 64 IPs
 < > ^ v

Tags - optional

Key	Value - optional
<input type="text" value="Name"/> X	<input type="text" value="Pub-Sub-AZ1"/> X

[Add new tag](#)
You can add 49 more tags.

[Remove](#)

Subnet 2 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 64 IPs
 < > ^ v

Tags - optional

Key	Value - optional
<input type="text" value="Name"/> X	<input type="text" value="Prv-Sub-AZ1"/> X

[Add new tag](#)
You can add 49 more tags.

[Remove](#)

Subnet 3 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 64 IPs
 < > ^ v

Tags - optional

Key	Value - optional
<input type="text" value="Name"/> X	<input type="text" value="Pub-Sub-AZ2"/> X

[Add new tag](#)
You can add 49 more tags.

[Remove](#)

Subnet 4 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 64 IPs
 < > ^ v

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Prv-Sub-AZ2"/>
Add new tag	
You can add 49 more tags.	
Remove	

[Add new subnet](#)

3.5 After creating the subnets now create an internet gateway, NAT gateway and Route table.

VPC dashboard

You have successfully updated subnet associations for rtb-0ab273658cc758dc / RT-1.

[VPC](#) > [Route tables](#) > [rtb-0ab273658cc758dc / RT-1](#)

rtb-0ab273658cc758dc / RT-1

[Actions](#)

Details [Info](#)

Route table ID	Main	Explicit subnet associations	Edge associations
<input type="text" value="rtb-0ab273658cc758dc"/>	<input type="checkbox"/> No	2 subnets	-
VPC	Owner ID		
vpc-032ba923ed1f6da8b VPC-R1	<input type="text" value="014498660859"/>		

[Routes](#) [Subnet associations](#) [Edge associations](#) [Route propagation](#) [Tags](#)

Explicit subnet associations (2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Pub-Sub-AZ2	subnet-0b5a3a31caaee0e84	10.0.0.128/26	-
Pub-Sub-AZ1	subnet-01f963de3f417511d	10.0.0.0/26	-

[Edit subnet associations](#)

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="IGW-1"/>
Add new tag	
You can add 49 more tags.	

[Cancel](#) [Create internet gateway](#)

The screenshot shows the AWS VPC dashboard. A green banner at the top indicates that the Internet gateway `igw-0e0a6056a9f02551f` has been successfully attached to the VPC `vpc-032ba923ed1f6da8b`. The main pane displays the details of the Internet gateway, including its ID, state (Attached), VPC ID, and owner. The 'Tags' section shows a single tag named 'IGW-1'. The left sidebar lists various VPC-related options like 'Your VPCs', 'Subnets', 'Route tables', and 'Internet gateways'.

The screenshot shows the AWS VPC dashboard. A green banner at the top indicates that a new Internet gateway `igw-00101c20c3ee82dc4` has been created and is now detached from any VPC. The main pane displays the details of this new Internet gateway, including its ID, state (Detached), and owner. The 'Tags' section shows a single tag named 'IGW-2'. The left sidebar lists various VPC-related options like 'Your VPCs', 'Subnets', 'Route tables', and 'Internet gateways'.

3.6 Under Route Tables created associate the subnets based upon the resource enlisted such as public or private subnets.

The screenshot shows the AWS Route Tables interface. It displays a table of routes for a specific route table. One route is listed with the destination `10.0.0.0/24`, target `local`, status `Active`, and propagated status `No`. Another route is listed with the destination `0.0.0.0/0`, target `Internet Gateway`, status `-`, and propagated status `No`. The target dropdown for this route is set to `igw-0e0a6056a9f02551f`. At the bottom, there are buttons for `Add route`, `Cancel`, `Preview`, and `Save changes`.

The screenshot shows the AWS VPC dashboard. A green success message at the top states: "You have successfully updated subnet associations for rtb-0b1223f54deca2851 / RT-2." The main pane displays route table details for "rtb-0b1223f54deca2851 / RT-2". Under the "Subnet associations" tab, there are two entries: "Pub-Sub-AZ1" and "Pub-Sub-AZ2", each associated with a specific subnet ID and IPv4 CIDR range.

3.7 Attaching internet gateway for Public route table.

The screenshot shows the "Edit routes" section for a route table. A new route is being added for destination "0.0.0.0/0" with target "Internet Gateway" and ID "igw-0e0a6056a9f02551f". The status is set to "Active". The "Save changes" button is highlighted.

The screenshot shows the "Edit routes" section for a route table. A new route is being added for destination "0.0.0.0/0" with target "Internet Gateway" and ID "igw-00101c20c3ee82dc4". The status is set to "Active". The "Save changes" button is highlighted.

3.8 Similarly, create a route table for Private and allocate the nat gateway and association of it.



VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

Prv-RT

VPC
The VPC to use for this route table.

vpc-0cfb29b22f705341b (food-vpc)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Q Name	Q Prv-RT

Add new tag

You can add 49 more tags.

Cancel **Create route table**

services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

food-ng

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

subnet-0f1776cb1f3b5cd1e (Pub-sub)

Connectivity type
Select a connectivity type for the NAT gateway.

Public

Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.

eipalloc-0133d6ce8d67ffdb6

Allocate Elastic IP

► Additional settings Info

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Q Name	Q food-ng

Add new tag

You can add 49 more tags.

Cancel **Create NAT gateway**

<input checked="" type="checkbox"/> Prv-RT	rtb-007c1a9549b2519e6	-	-	No	vpc-0cfb29b22f705341b
=					
rtb-007c1a9549b2519e6 / Prv-RT					
Details	Routes	Subnet associations	Edge associations	Route propagation	Tags
Routes (1)					
<input type="button" value="Both"/> <input type="button" value="Edit routes"/> <input type="button" value="Filter routes"/> < 1 > <input type="button" value=""/>					
Destination	▼	Target	▼	Status	▼ Propagated
10.0.0.0/16		local		<input checked="" type="checkbox"/> Active	No

3.9 After creating the route table, add a route to allow Internet access.

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
<input type="text" value="Q_ 0.0.0.0"/>	<input type="text" value="Q_ local"/>		
<input type="text" value="Q_ 0.0.0.0/0"/>	<input type="text" value="Q_ NAT Gateway"/>		
	<input type="text" value="Q_ nat-0bd6bdb687eee301f"/>		

Add route Cancel Preview Save changes

3.10 Now we will associate subnets with Route Tables.

You have successfully updated subnet associations for rtb-Oab273658cc758dc / RT-1.

Details																							
Route table ID rtb-Oab273658cc758dc	Main No	Explicit subnet associations 2 subnets	Edge associations -																				
VPC vpc-032ba923ed1f6da8b VPC-R1	Owner ID 014498660859																						
Routes Subnet associations Edge associations Route propagation Tags																							
Explicit subnet associations (2) <table border="1"> <thead> <tr> <th colspan="4">Edit subnet associations</th> </tr> <tr> <td colspan="4"> <input type="text" value="Q_ Find subnet association"/> < 1 > ⌂ </td> </tr> <tr> <th>Name</th> <th>Subnet ID</th> <th>IPv4 CIDR</th> <th>IPv6 CIDR</th> </tr> </thead> <tbody> <tr> <td>Pub-Sub-AZ2</td> <td>subnet-0b5a3a31caae00e84</td> <td>10.0.0.128/26</td> <td>-</td> </tr> <tr> <td>Pub-Sub-AZ1</td> <td>subnet-01f963de3f417511d</td> <td>10.0.0.0/26</td> <td>-</td> </tr> </tbody> </table>				Edit subnet associations				<input type="text" value="Q_ Find subnet association"/> < 1 > ⌂				Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Pub-Sub-AZ2	subnet-0b5a3a31caae00e84	10.0.0.128/26	-	Pub-Sub-AZ1	subnet-01f963de3f417511d	10.0.0.0/26	-
Edit subnet associations																							
<input type="text" value="Q_ Find subnet association"/> < 1 > ⌂																							
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR																				
Pub-Sub-AZ2	subnet-0b5a3a31caae00e84	10.0.0.128/26	-																				
Pub-Sub-AZ1	subnet-01f963de3f417511d	10.0.0.0/26	-																				

You have successfully updated subnet associations for rtb-0b1223f54deca2851 / RT-2.

Details																							
Route table ID rtb-0b1223f54deca2851	Main No	Explicit subnet associations 2 subnets	Edge associations -																				
VPC vpc-00ce30a7ccacbfef9d VPC-R2	Owner ID 014498660859																						
Routes Subnet associations Edge associations Route propagation Tags																							
Explicit subnet associations (2) <table border="1"> <thead> <tr> <th colspan="4">Edit subnet associations</th> </tr> <tr> <td colspan="4"> <input type="text" value="Q_ Find subnet association"/> < 1 > ⌂ </td> </tr> <tr> <th>Name</th> <th>Subnet ID</th> <th>IPv4 CIDR</th> <th>IPv6 CIDR</th> </tr> </thead> <tbody> <tr> <td>Pub-Sub-AZ1</td> <td>subnet-0978f33978a873ada</td> <td>10.1.0.0/26</td> <td>-</td> </tr> <tr> <td>Pub-Sub-AZ2</td> <td>subnet-0d0c09f452011cb84</td> <td>10.1.0.128/26</td> <td>-</td> </tr> </tbody> </table>				Edit subnet associations				<input type="text" value="Q_ Find subnet association"/> < 1 > ⌂				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	-
Edit subnet associations																							
<input type="text" value="Q_ Find subnet association"/> < 1 > ⌂																							
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	-																				

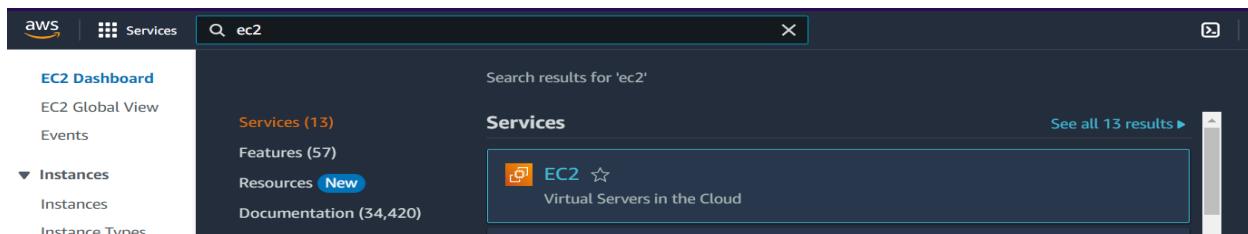
4. EC2 Configuration

EC2s Configuration:

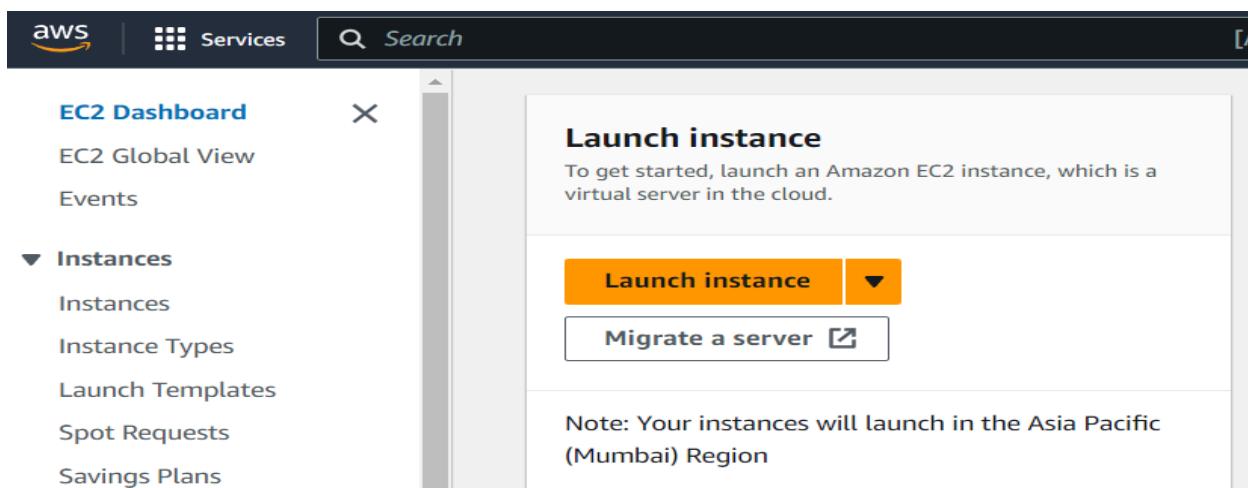
- In the EC2 dashboard, click on **Launch Instance**.
- On the **Choose an Amazon Machine Image (AMI)** page, click on the **AWS Marketplace** tab.
- Browse through the available AMIs and search for CentOS 9 Desktop - GUI Gnome with RDP by Arara Solutions.
- Configure the Instance details, network, storage and security group and attach the EC2-Systems Manager Role and launch the instance.

Step 4: Launch EC2 in Console

4.1 In Dashboard, search for EC2 service, click **EC2** to open it.



4.2 Click **Launch instance** to configure.



4.3 Type the **Name** which you prefer for naming the instance and Choose the **AMI**, so here we are using **CentOS**.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The configuration steps are as follows:

- Name and tags**: The instance is named "web-server-R1".
- Application and OS Images (Amazon Machine Image)**: The AMI selected is "Arara Solutions - CentOS Stream 9 - Desktop RDP", with the ID "-v0323-1bf72720-8a45-4d52-a8a8-".
- Summary**: Shows 1 instance, Software Image (AMI) as CentOS 9 Desktop - GUI Gnome, Virtual server type as t3.medium, Firewall as Sg-web-v1, and Storage (volumes) as 1 volume(s) - 20 GiB.
- Free tier** callout: In the first year, it includes 750 hours of t2.micro (or t3.micro in regions where t2.micro is unavailable), instance usage on free-tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth.

4.4 Select the instance type **t2.micro** and then create the pair to do that click on **Create new pair key** for secure connection.

The screenshot shows the 'Create key pair' dialog. The key pair name is set to "OCT_2024_NV_V1". The key pair type is selected as RSA. The private key file format is chosen as ".pem". A note at the bottom states: "When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance." A "Create key pair" button is visible at the bottom right.

The screenshot shows the final step of the 'Create key pair' wizard. The key pair name is "OCT_2024_NV_V1". The key pair type is RSA. The private key file format is ".pem". A note at the bottom states: "When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance." A "Create key pair" button is visible at the bottom right.

4.5 Under Network Settings, click **Edit** and under Firewall (security groups) choose **Create security group** option and then check in inbound security group rules that Type field is **HTTP** and source type field is **Anywhere**. If the above fields are not present then click **Add security group rule** and click those options.

4.6 Configure Security Group.

Name	Security group ID	Security group name	VPC ID	Description
-	sg-019fe02489ad63e41	default	vpc-00535f1e5d82790e3	default VP
-	sg-0fd363896a36c5a53	SG-WEB-V1	vpc-032ba923ed1f6da8b	CentOS 9 I
-	sg-00b04d02335e9a3c6	default	vpc-00ce30a7ccacbfef9d	default VP
-	sg-0d42c44076bfb05bb	SG-WEB-V2	vpc-00ce30a7ccacbfef9d	sg web for
-	sg-047e6aa4f2f1c8102	default	vpc-032ba923ed1f6da8b	default VP

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0668f72f722f23df8	IPv4	HTTP	TCP	80
-	sgr-0211f8f7939b165a3	IPv4	RDP	TCP	3389
-	sgr-02d777e43ba27ff66	IPv4	SSH	TCP	22
-	sgr-078855439b9dfff16	IPv4	HTTPS	TCP	443
-	sgr-0dfc5c0446d73d5ab	IPv4	DNS (TCP)	TCP	53

4.7 Go to the Advanced Details and under IAM instance profile allocate the role of EC2-SSM Role.

The screenshot shows the AWS CloudFormation console with the following details:

- Configure storage:** 1x 20 GiB gp2 Root volume (Not encrypted)
- Advanced details:** IAM instance profile: EC2-SSM-Role
- Summary:** Number of instances: 1
- Software Image (AMI):** CentOS 9 Desktop - GUI Gnome w...read more ami-07c7123aa10b5d95a
- Virtual server type (instance type):** t3.medium
- Firewall (security group):** Sg-web-v1
- Storage (volumes):** 1 volume(s) - 20 GiB

A tooltip for the free tier is displayed, stating: "Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage".

4.8 Leave rest configuration as default will come back later. Click on **Launch Instance**.

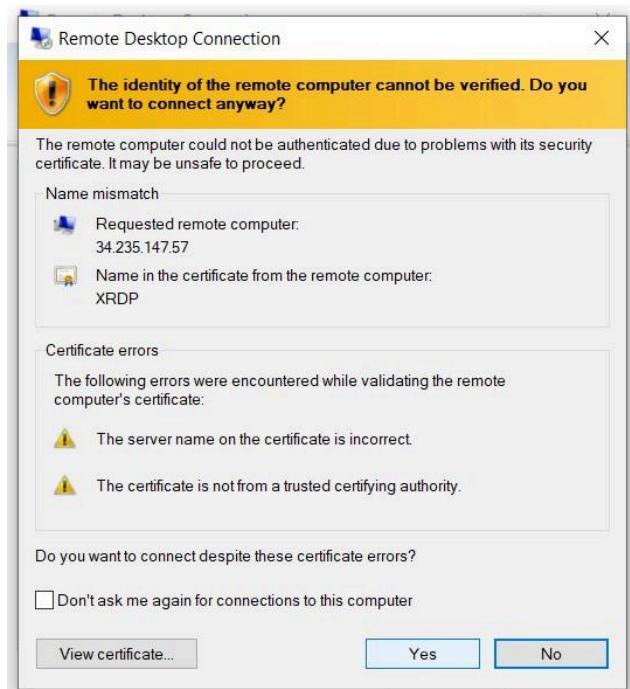
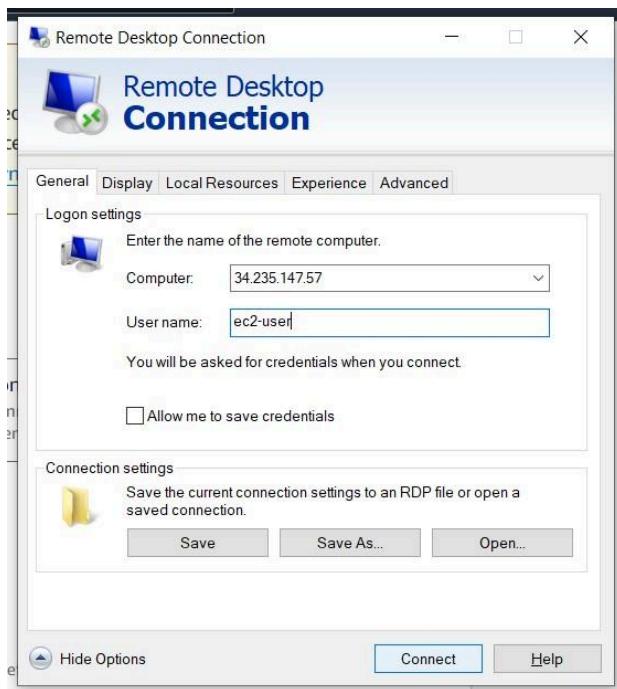
The screenshot shows the AWS CloudFormation console in the 'Summary' step of a new stack creation, with the following details:

- Number of instances:** 1
- Software Image (AMI):** CentOS 9 Desktop - GUI Gnome w...read more ami-07c7123aa10b5d95a
- Virtual server type (instance type):** t3.medium
- Firewall (security group):** Sg-web-v1
- Storage (volumes):** 1 volume(s) - 20 GiB

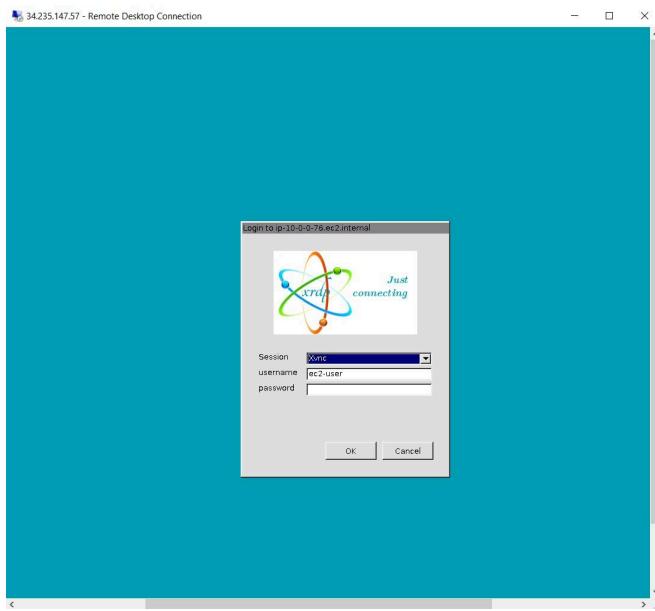
A tooltip for the free tier is displayed, stating: "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet."

At the bottom, there are three buttons: "Cancel", "Launch instance" (highlighted in orange), and "Review commands".

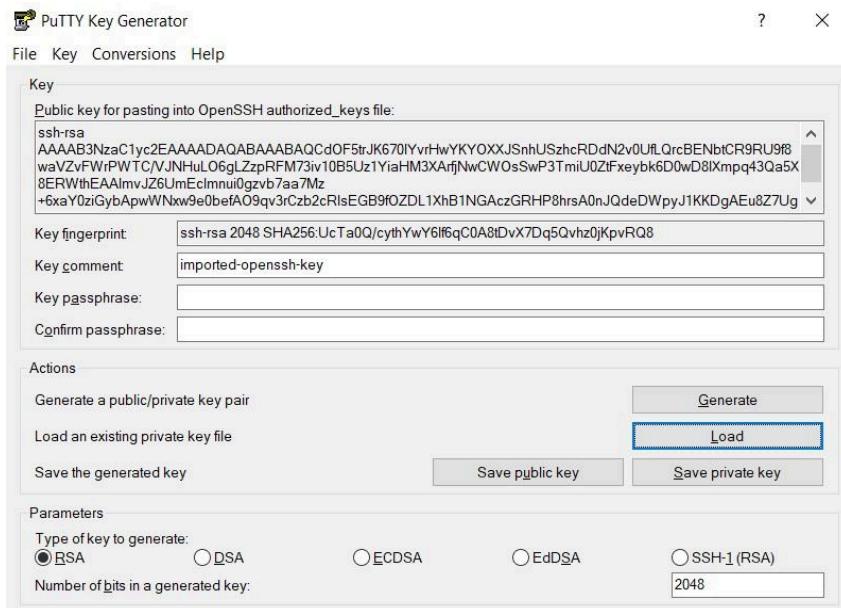
4.9 Using RDP to access an EC2 instance remotely.



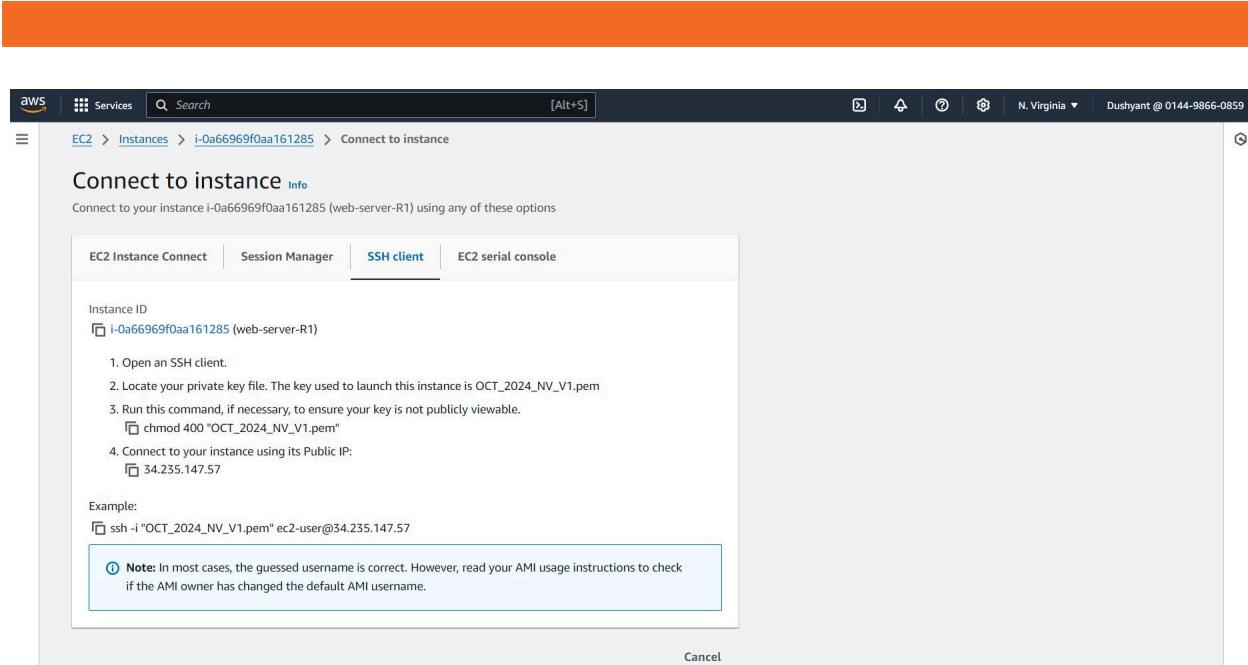
4.10 To login the XRD CentOS GUI first have to setup the password.



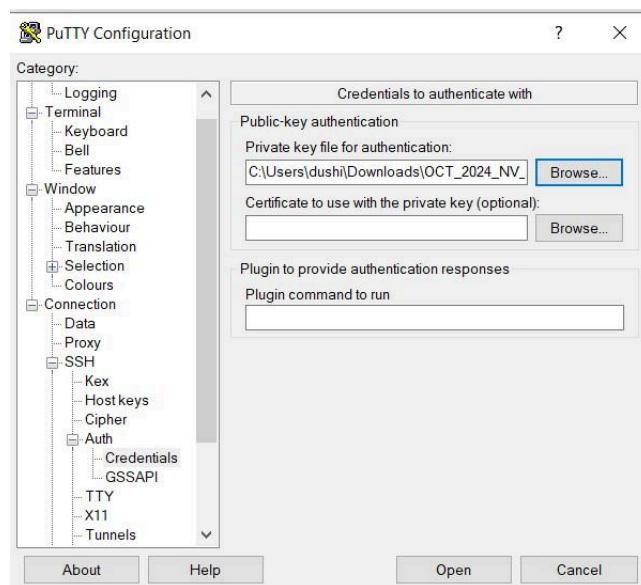
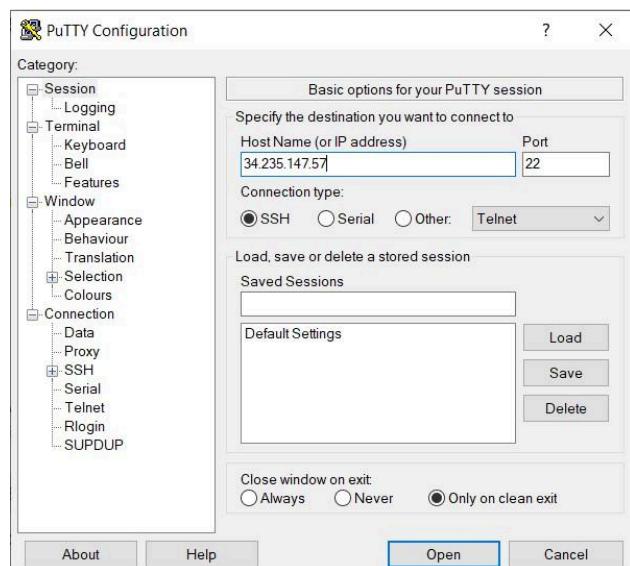
4.11 Convert the .pem file to .ppk file in puttygen



4.12 Using Putty to access an EC2 instance remotely.

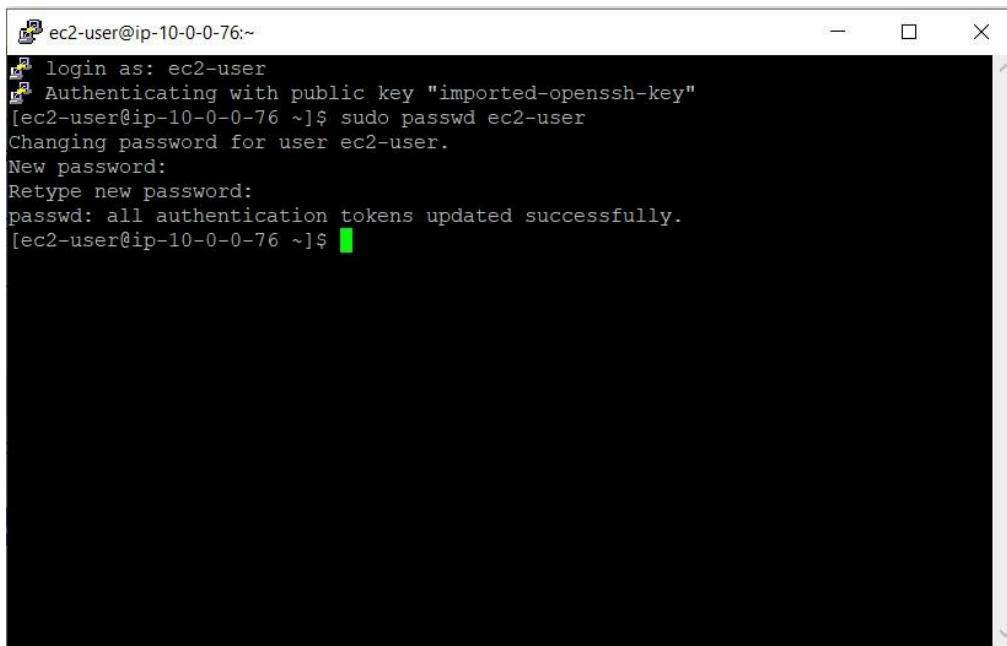


The screenshot shows the AWS EC2 'Connect to instance' dialog. At the top, it displays the navigation path: AWS Services > Instances > i-0a66969f0aa161285 > Connect to instance. The title is 'Connect to instance' with an 'Info' link. Below the title, a message says 'Connect to your instance i-0a66969f0aa161285 (web-server-R1) using any of these options'. There are three tabs: 'EC2 Instance Connect', 'Session Manager', and 'SSH client' (which is selected). Under 'SSH client', the 'Instance ID' is listed as 'i-0a66969f0aa161285 (web-server-R1)'. Below this are four numbered steps: 1. Open an SSH client, 2. Locate your private key file. The key used to launch this instance is OCT_2024_NV_V1.pem, 3. Run this command, if necessary, to ensure your key is not publicly viewable. (checkbox: chmod 400 "OCT_2024_NV_V1.pem"), 4. Connect to your instance using its Public IP: (checkbox: 34.235.147.57). An example command is provided: ssh -i "OCT_2024_NV_V1.pem" ec2-user@34.235.147.57. A note at the bottom states: 'Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' A 'Cancel' button is at the bottom right.



4.13 Assign the password by typing the command

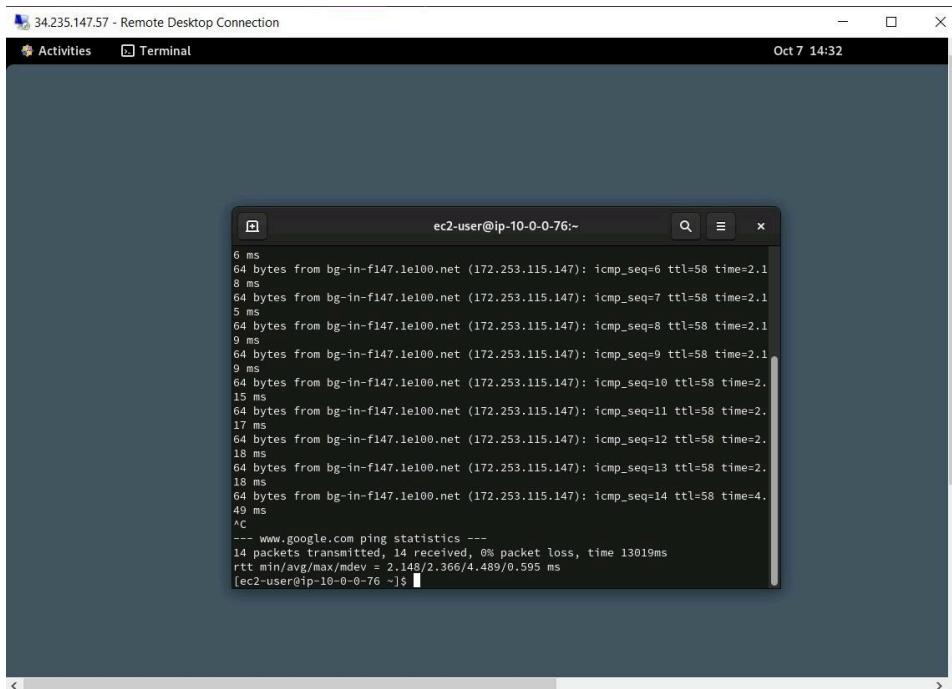
```
sudo passwd ec2-user
```



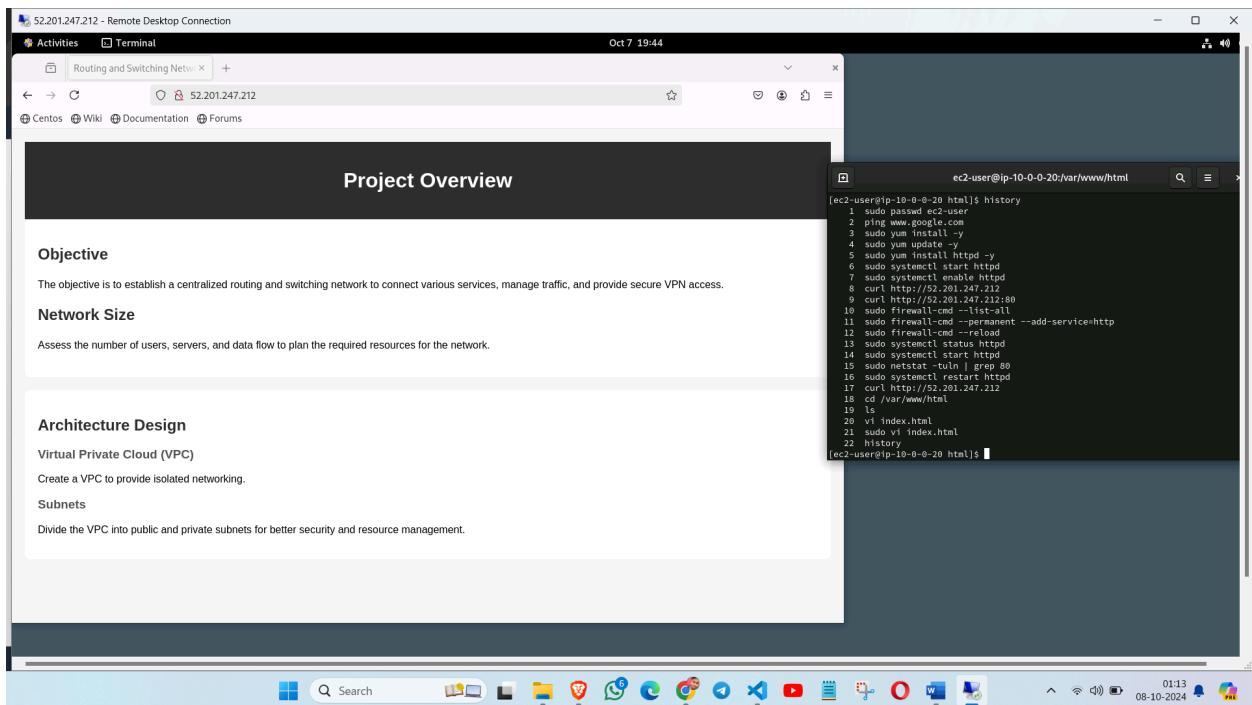
The screenshot shows a terminal window titled "ec2-user@ip-10-0-0-76:~". It displays the following command and its execution:

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-10-0-0-76 ~]$ sudo passwd ec2-user
Changing password for user ec2-user.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-0-76 ~]$
```

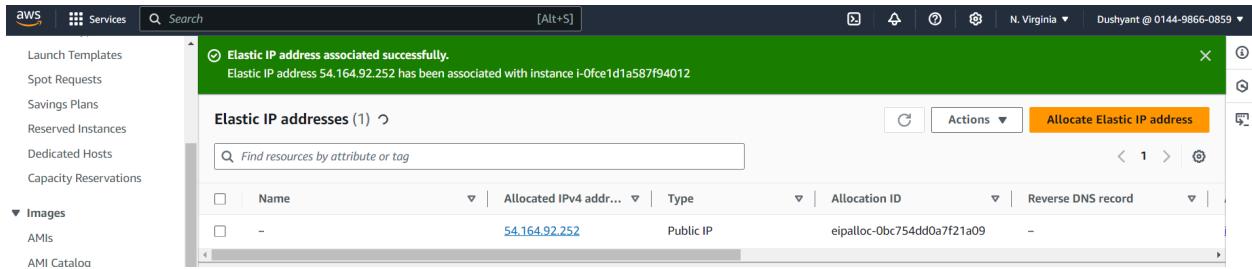
4.14 Finally, the CentOS GUI is launched and running.



4.15 Now launch the instances and configure the web server for adding the HTML files.



4.16 Create an Elastic IP and associate the EC2 instance.



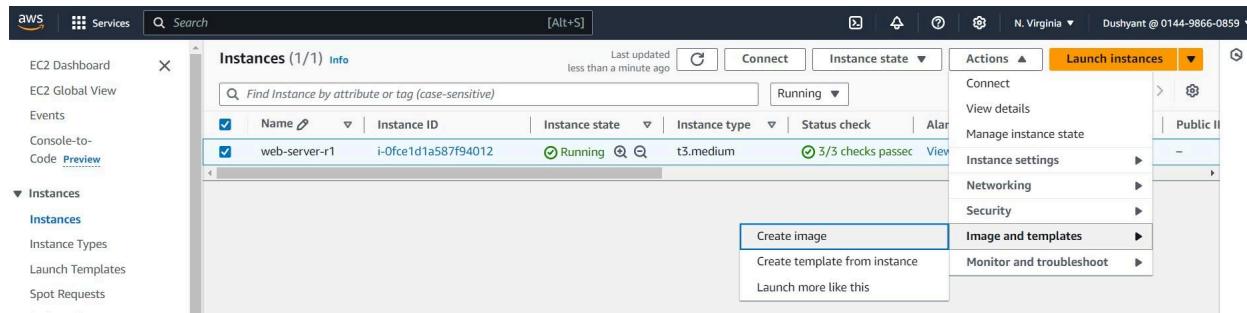
5. AMI Configuration

AMI Configuration:

- Go to build Instance, click on **Actions** navigate to Image and Templates.
- Click Create Image and assign a name, check the description and disable the reboot checkbox and click Create Image
- Now, with the help of AMI, launch the other 3 EC2 instances and configure it.

Step 5: Build AMI from configured EC2 instance

5.1 In Dashboard, go to the Instances click on Actions and select Image and templates and Create Image.



5.2 Configure the description and disable the reboot checkbox.

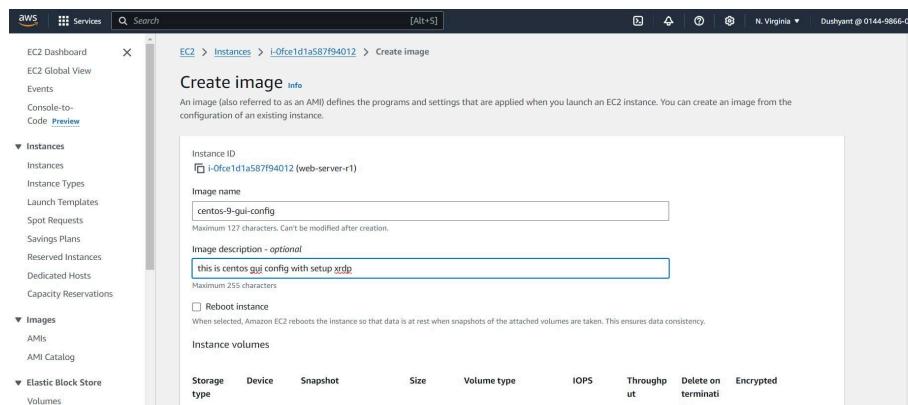


Image summary for ami-0fc95d4393026a23			
		Actions	
AMI ID	Image type	Platform details	Root device type
<input type="checkbox"/> ami-0fc95d4393026a23	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
<input type="checkbox"/> centos-9-gui-config	014498660859	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
<input type="checkbox"/> /dev/sda1	Available	<input type="checkbox"/> 014498660859/centos-9-gui-config	hvm
Boot mode	State reason	Creation date	Kernel ID
-	-	<input type="checkbox"/> Mon Oct 07 2024 21:12:37 GMT+0530 (India Standard Time)	-
Description	Product codes	RAM disk ID	Deprecation time
<input type="checkbox"/> this is centos gui config with setup xrdp	<input type="checkbox"/> marketplace:1nloot6yyt4cdqma9d8ldy5e	-	-
Last launched time	Block devices	Deregistration protection	
Tue Oct 08 2024 00:07:25 GMT+0530 (India Standard Time)	<input type="checkbox"/> /dev/sda1=snap- 00966d3aa24356213:20:true:gp2	<input checked="" type="checkbox"/> Disabled	

Amazon Machine Images (AMIs) (1) Info						
Owned by me		<input type="text"/> Find AMI by attribute or tag			Actions	
<input type="checkbox"/>	Name	AMI name	AMI ID	Source	Owner	Visibility
<input type="checkbox"/>	centos-9-gui-config	ami-0fc95d4393026a23	014498660859/centos-9-gui-config	014498660859	014498660859	Private

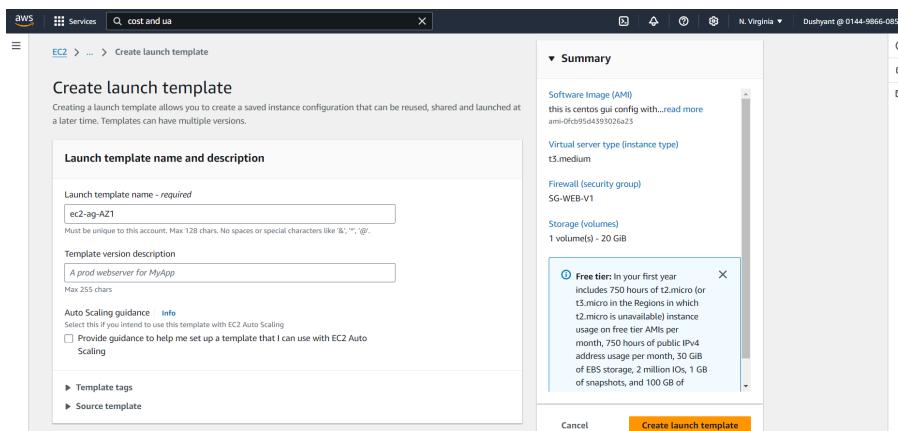
6. Auto Scaling Configuration

Auto Scaling Configuration:

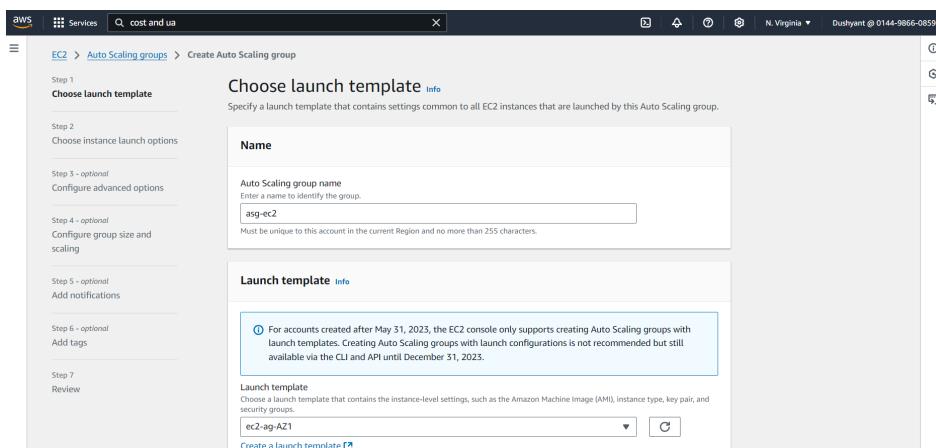
- In the EC2 dashboard, click on **Launch Templates**, assign the name and instance type and click **Create launch template**.
- Now, click **Create Auto Scaling group** and select the VPC and subnets.
- Configure the group size and scaling policies and click Create Auto Scaling Group.

Step 6: Build Launch Template to configure Auto Scaling instances

6.1 Build a launch template from the AMI we have created.



6.2 Now, create the auto scaling group and attach the launch template.



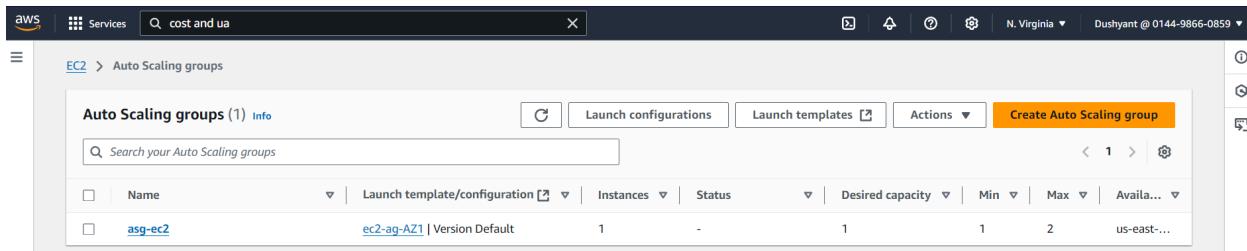
6.3 Configure it and review the details

The screenshot shows the 'Review' step of creating an Auto Scaling group. On the left, a sidebar lists steps from 1 to 5, with 'Step 1 Choose launch template' selected. The main area displays 'Step 1: Choose launch template' with a table for 'Group details'. It shows the 'Auto Scaling group name' as 'asg-ec2' and the 'Launch template' as 'ec2-ag-AZ1'. An 'Edit' button is at the top right of the table.

The screenshot shows 'Step 2: Choose instance launch options'. It includes sections for 'Network' (VPC: vpc-032ba923ed1f6da8b), 'Availability Zone' (us-east-1a and us-east-1b), and 'Instance type requirements'. Below this is 'Step 3: Configure advanced options' with a 'Load balancing' section for 'Load balancer 1' (Name: elb-r1, Type: Application/HTTP, Target group: tg-r1).

The screenshot shows 'Step 4: Configure group size and scaling policies'. It includes sections for 'Group size' (Desired capacity: 1), 'Scaling' (Minimum desired capacity: 1, Maximum desired capacity: 2), 'Instance maintenance policy' (Replacement behavior: No policy, Min healthy percentage: -, Max healthy percentage: -), and 'Instance scale-in protection' (Enable instance protection from scale in: checked).

6.4 Successfully configures the auto scaling group.



The screenshot shows the AWS EC2 Auto Scaling groups page. At the top, there are navigation links for Services, a search bar containing "cost and ua", and account information for "N. Virginia" and "Dushyant @ 0144-9866-0859". Below the header, the breadcrumb trail shows "EC2 > Auto Scaling groups". The main content area is titled "Auto Scaling groups (1) Info". It includes a search bar and a table with one row. The table columns are: Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Available. The single row shows "asg-ec2" with "ec2-ag-AZ1 | Version Default" under Launch template/configuration, "1" under Instances, and "1" under Desired capacity. The "Max" column shows "1" and "2", and the "Available" column shows "us-east-...".

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Available
asg-ec2	ec2-ag-AZ1 Version Default	1	-	1	1	2	us-east-...

7. Load Balancer Configuration

Load Balancer Configuration:

- Click the **Create Load Balancer** button and Select **Application Load Balancer** and assign the name, configure HTTP and HTTPS in security group under inbound rules
- Create a Target group to register the instances and provision the health checks.
- Go back to the Load balancer page and add the listeners and review the configuration and click **Create Load Balancer**.

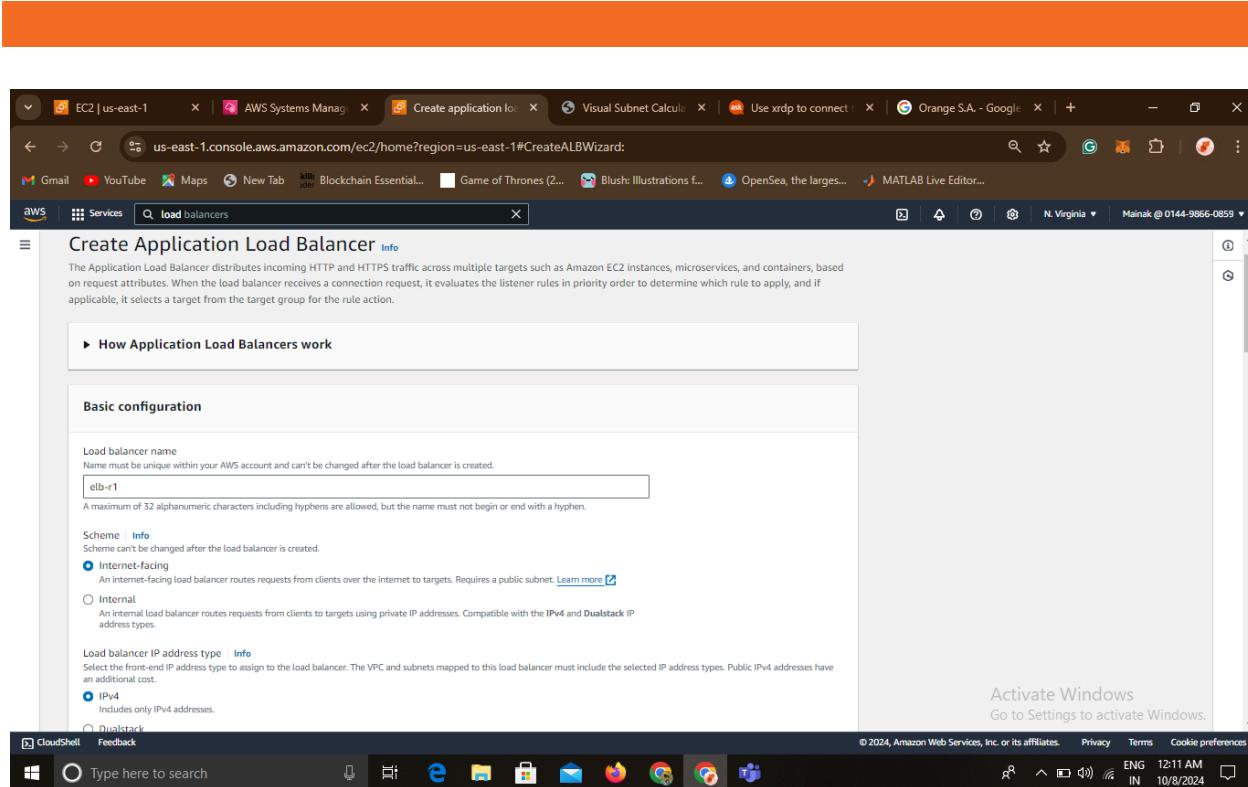
Step 7: Launch ELB to distribute Load

7.1 In the EC2 left pane scroll down to find **Load Balancer** click on it and create load balancer.

The screenshot shows the AWS EC2 Load Balancers page. At the top, there is a navigation bar with 'EC2 > Load balancers'. Below the navigation bar, there is a search bar labeled 'Filter load balancers' and a 'Create load balancer' button. A table header row includes columns for Name, DNS name, State, VPC ID, Availability Zones, and Type. The main content area displays a message: 'No load balancers' and 'You don't have any load balancers in ap-south-1'. At the bottom of the page, there is another 'Create load balancer' button.

7.2 So here, we will choose **Application Load Balancer** to request HTTP traffic.

The screenshot shows the 'Load balancer types' selection page. It features three cards: 'Application Load Balancer' (selected), 'Network Load Balancer', and 'Gateway Load Balancer'.
The 'Application Load Balancer' card includes a diagram showing traffic from a client through an ALB to multiple targets (Lambda, API Gateway, Container). It describes the feature set for applications with HTTP and HTTPS traffic.
The 'Network Load Balancer' card includes a diagram showing traffic from a client through an NLB to multiple targets (VPC, VPCe, Lambda, API Gateway, Container). It describes the feature set for ultra-high performance, TLS offloading, and centralized certificate deployment.
The 'Gateway Load Balancer' card includes a diagram showing traffic from a client through a GWLB to multiple targets (AWS services like VPC, Lambda, API Gateway, Container). It describes the feature set for deploying and managing third-party virtual appliances.
Each card has a 'Create' button at the bottom.



7.3 Add the VPC and the mappings for Availability zones with the existing security group

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

tiervp	Edit
	Delete
	Copy
	Share

Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

ap-south-1a (aps1-az1)

Subnet: subnet-0761341a7fee23ec7 (Pubsub1)

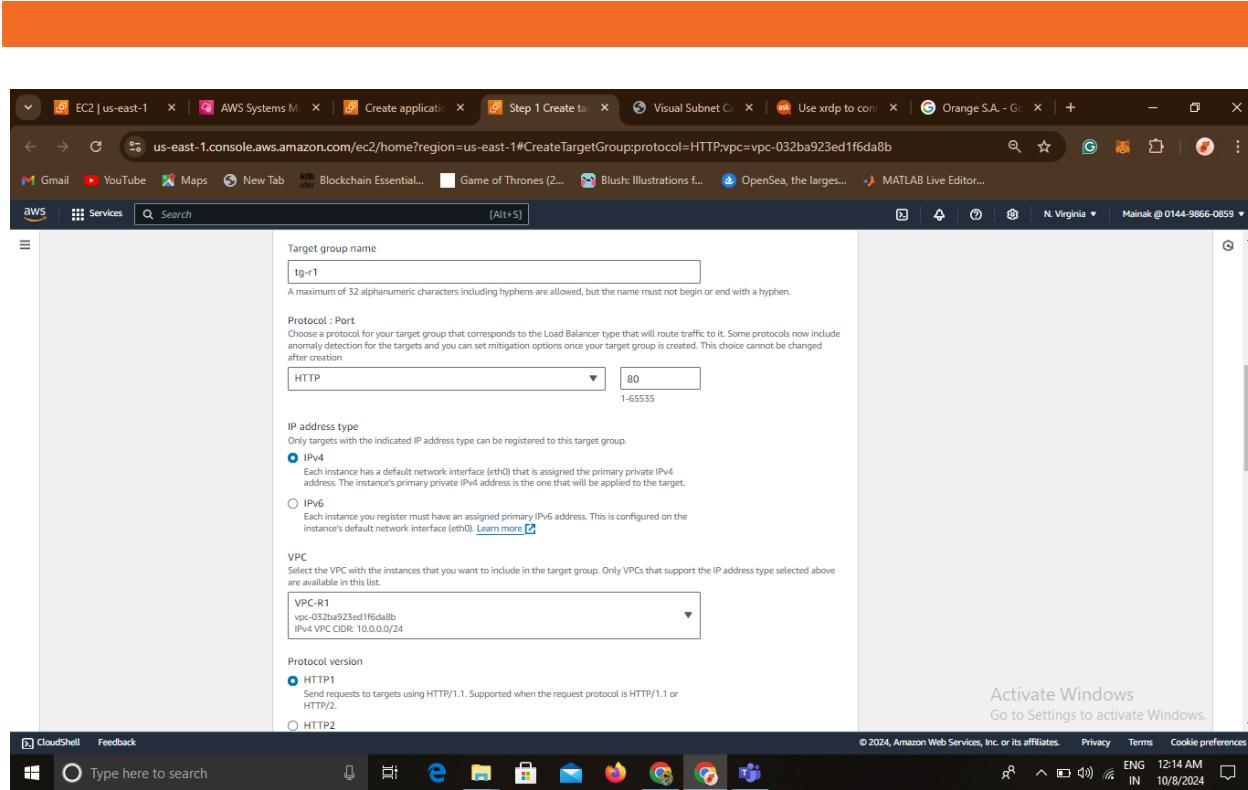
IPv4 address: Assigned by AWS

ap-south-1b (aps1-az3)

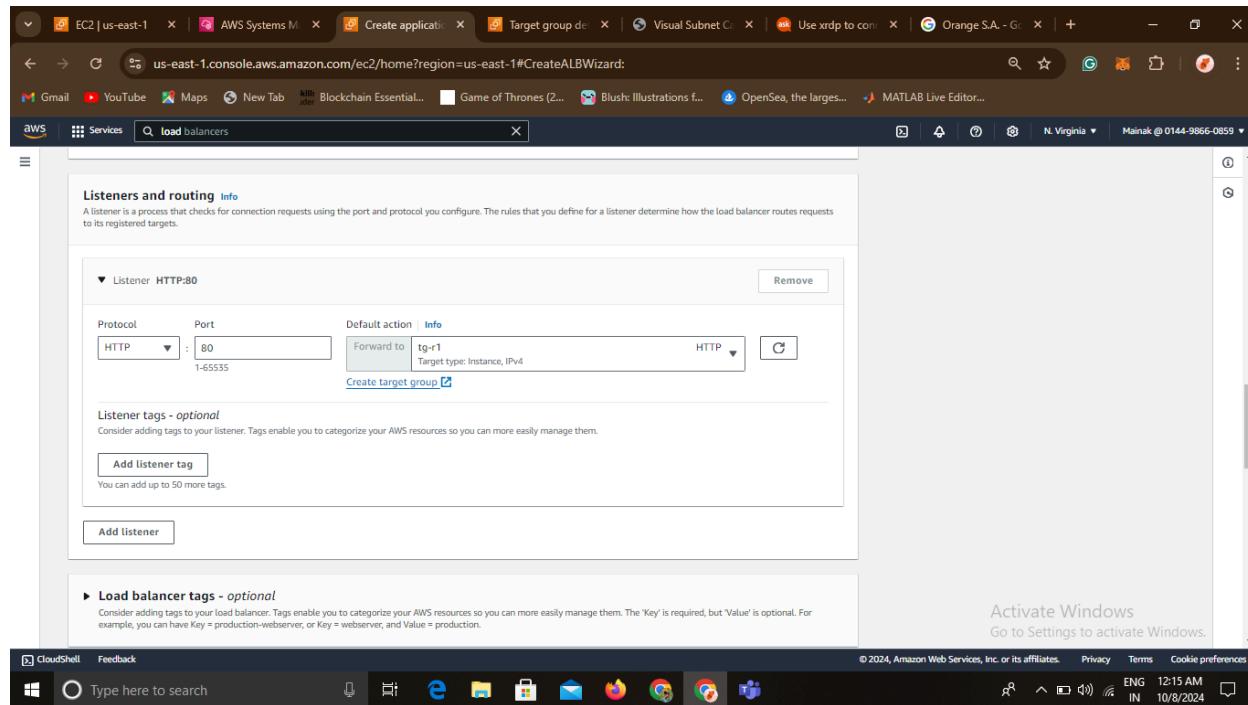
Subnet: subnet-082b12e931b78231a (Pubsub2)

IPv4 address: Assigned by AWS

7.4 Create a **Target group** with register targets, Click **Next** and then register the targets by **Include as pending below**.



7.5 Now associate the build **Target group** in Application load balancer and click **Create Load balancer**.



Successfully created load balancer: elb-r1

It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

EC2 > Load balancers > elb-r1

elb-r1

Details

Load balancer type	Status	VPC	Load balancer IP address type
Application	Provisioning	vpc-032ba923ed1f6da8b	IPv4
Scheme	Hosted zone	Availability Zones	Date created
Internet-facing	Z35SXDOTRQ7X7K	subnet-05sa3a31caa0e84 (us-east-1a) subnet-01963de3f417511d (us-east-1a)	October 8, 2024, 00:15 (UTC+05:30)
Load balancer ARN		DNS name info	
arn:aws:elasticloadbalancing:us-east-1:014498660859:loadbalancer/app/elb-r1/90c1048ba02afc47		elb-r1-1220113217.us-east-1.elb.amazonaws.com (A Record)	

Listeners and rules Network mapping Resource map - new Security Monitoring Integrations Attributes Tags

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Type here to search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 12:15 AM 10/8/2024

7.6 Similarly, configure one more load balancer for other 2 EC2 instances.

Load balancers (2)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type
elb-r2	elb-r2-882009896.us-east...	Active	vpc-00ce30a7ccacbfef9d	2 Availability Zones	application
elb-r1	elb-r1-1220113217.us-eas...	Active	vpc-032ba923ed1f6da...	2 Availability Zones	application

0 load balancers selected

Select a load balancer above.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 02:36 08-10-2024

7.7 Now, copy the DNS name and paste it in a new tab to check the distribution of load.

ELB-1

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Project Overview". The content area contains a section titled "Objective" with a brief description: "The objective is to establish a centralized routing and switching network to connect various services, manage traffic, and provide secure VPN access." Below this is a section titled "Network Size" with a note: "Assess the number of users, servers, and data flow to plan the required resources for the network." At the bottom of the content area is a standard Windows taskbar.

Project Overview

Objective

The objective is to establish a centralized routing and switching network to connect various services, manage traffic, and provide secure VPN access.

Network Size

Assess the number of users, servers, and data flow to plan the required resources for the network.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "CentOS Server Configuration". The content area contains a section titled "EC2 Instances" with a note: "Deploy CentOS instances in private subnets for network administration tasks." Below this is a section titled "Network Configuration" with two bullet points: "Use ifconfig or ip to configure network interfaces." and "Edit /etc/sysconfig/network-scripts/ for persistent configurations." Further down is a section titled "Firewall Configuration" with a note: "Configure firewalld or iptables to manage inbound and outbound traffic." At the bottom of the content area is a standard Windows taskbar.

CentOS Server Configuration

EC2 Instances

Deploy CentOS instances in private subnets for network administration tasks.

Network Configuration

- Use `ifconfig` or `ip` to configure network interfaces.
- Edit `/etc/sysconfig/network-scripts/` for persistent configurations.

Firewall Configuration

Configure firewalld or iptables to manage inbound and outbound traffic.

ELB-2

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Not secure elb-r2-882009896.us-east-1.elb.amazonaws.com". The page content is as follows:

Security, Backup, and Conclusion

Security Best Practices

- Implement least privilege access using IAM roles and policies.
- Use Security Groups and Network ACLs to manage traffic.
- Ensure encryption for data in transit and at rest.

Backup and Recovery

- Set up automated snapshots of EC2 instances and RDS databases.
- Replicate critical data across regions for disaster recovery.

Cost Management

Use AWS Budgets and Cost Explorer to track and manage spending effectively.

At the bottom of the browser window, the taskbar shows various pinned icons and system status indicators.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Not secure elb-r2-882009896.us-east-1.elb.amazonaws.com". The page content is as follows:

Routing and Switching Configuration

Core AWS Services

Amazon VPC

- Public Subnets: Load balancers, NAT gateways, and services that need internet access.
- Private Subnets: Application servers and databases that shouldn't be directly accessible from the internet.

Elastic IPs

Use Elastic IPs for static public IP addresses.

Route Tables

Configure route tables to manage traffic between subnets and the internet.

Routing and Switching Setup

AWS Transit Gateway

Use AWS Transit Gateway for centralized routing between multiple VPCs and on-premises networks.

At the bottom of the browser window, the taskbar shows various pinned icons and system status indicators.

8. Transit Gateway Configuration

Transit Gateway Configuration:

- In the VPC page, click on Transit Gateway to assign a name.
- Enter the ASN for Transit Gateway, then create transit gateway attachments, attach the VPCs that we created.
- Create the attachment and configure the security group for it.

Step 8: Launch Transit Gateway in Console

8.1 In Dashboard, search for Transit Gateway service, click **VPC** to open it.

The screenshot shows the AWS Services search interface. The search bar at the top contains the query "transit". Below the search bar, there are two main sections: "Identity and Access Management (IAM)" on the left and "Search results for 'transit'" on the right. The "Search results" section includes a sidebar with "Features" (Services, Resources New, Documentation) and a main content area titled "Transit gateways" which is described as a "VPC feature".

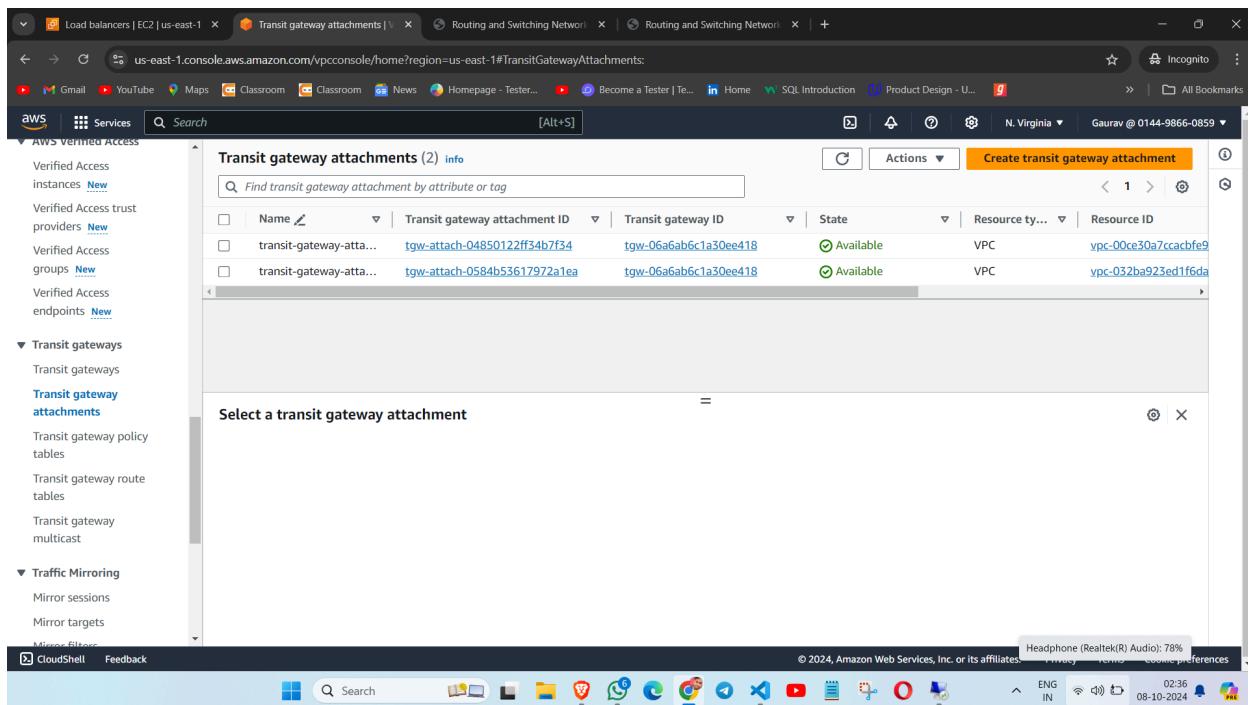
8.2 Create a transit gateway with default ASN 64512 and attach the VPC

The screenshot shows the AWS VPC dashboard. On the left, there is a sidebar with various VPC-related options like EC2 Global View, Virtual private cloud, Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The main content area displays the details of a transit gateway named "tgw-06a6ab6c1a30ee418". The "Details" section shows the following configuration:

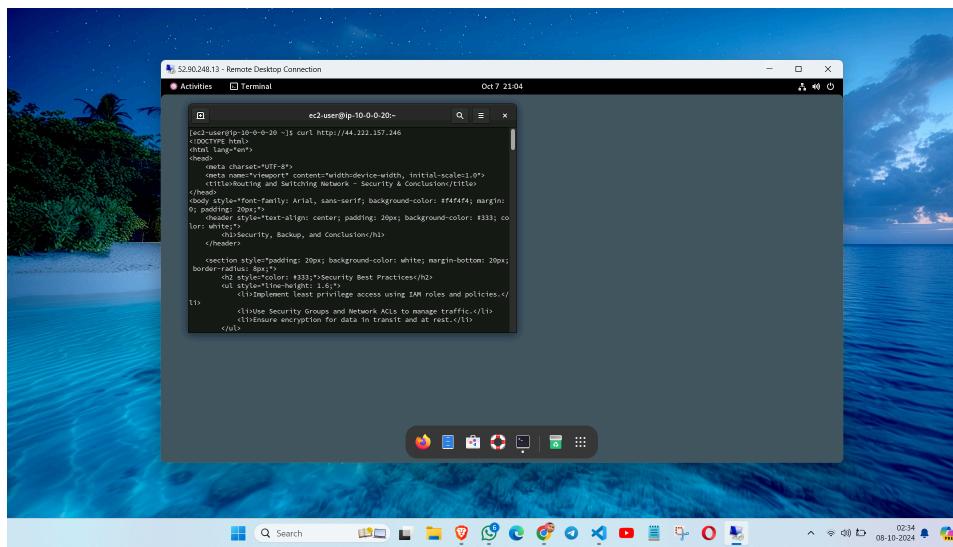
Transit gateway ID	tgw-06a6ab6c1a30ee418	Transit gateway ARN	arn:aws:ec2:us-east-1:014498660859:transit-gateway/tgw-06a6ab6c1a30ee418	Owner ID	014498660859	Description	my transit gateway
State	Available	Default association route table	Enable	Default propagation route table	Enable	Transit gateway CIDR blocks	-
Amazon ASN	64512	Association route table ID	tgw-rtb-0040a396aaeddb014	Propagation route table ID	tgw-rtb-0040a396aaeddb014	Security Group Referencing support	Disable
DNS support	Enable	Auto accept shared attachments	Enable	VPN ECMP support	Enable	Multicast support	Disable

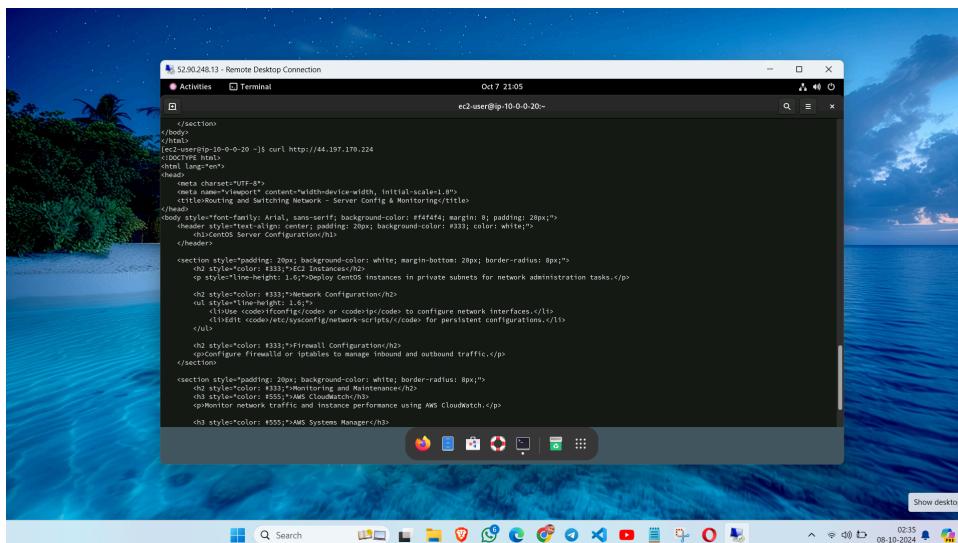
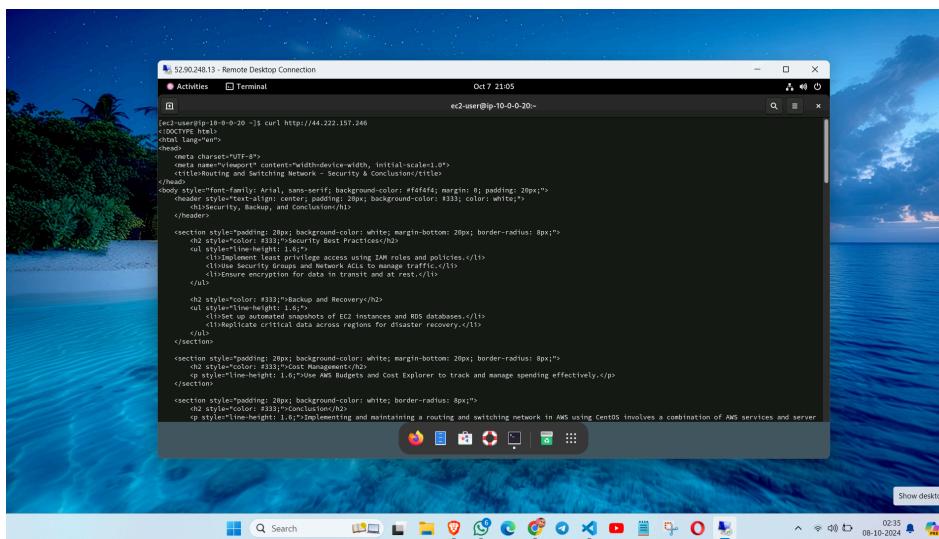
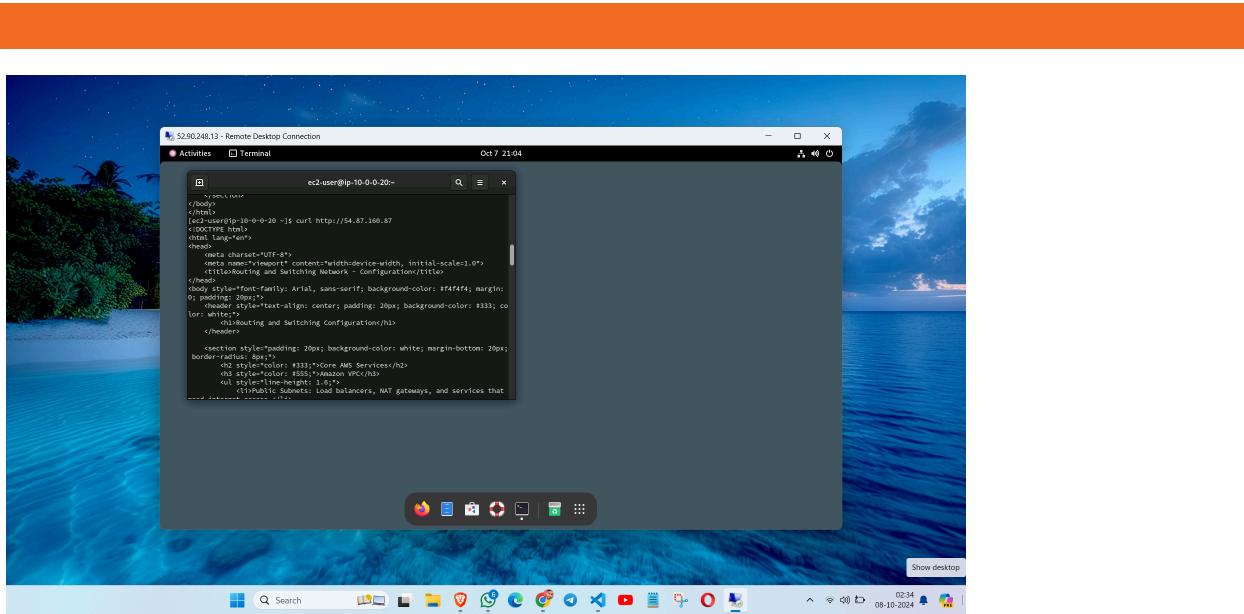
Below the details, there are tabs for "Flow logs", "Sharing", and "Tags". The "Flow logs" tab is active, showing a search bar and a table with columns: Name, Flow log ID, Destination type, Destination name, and IAM role AF. The table currently has one row with a "Name" column value of "tgw-06a6ab6c1a30ee418".

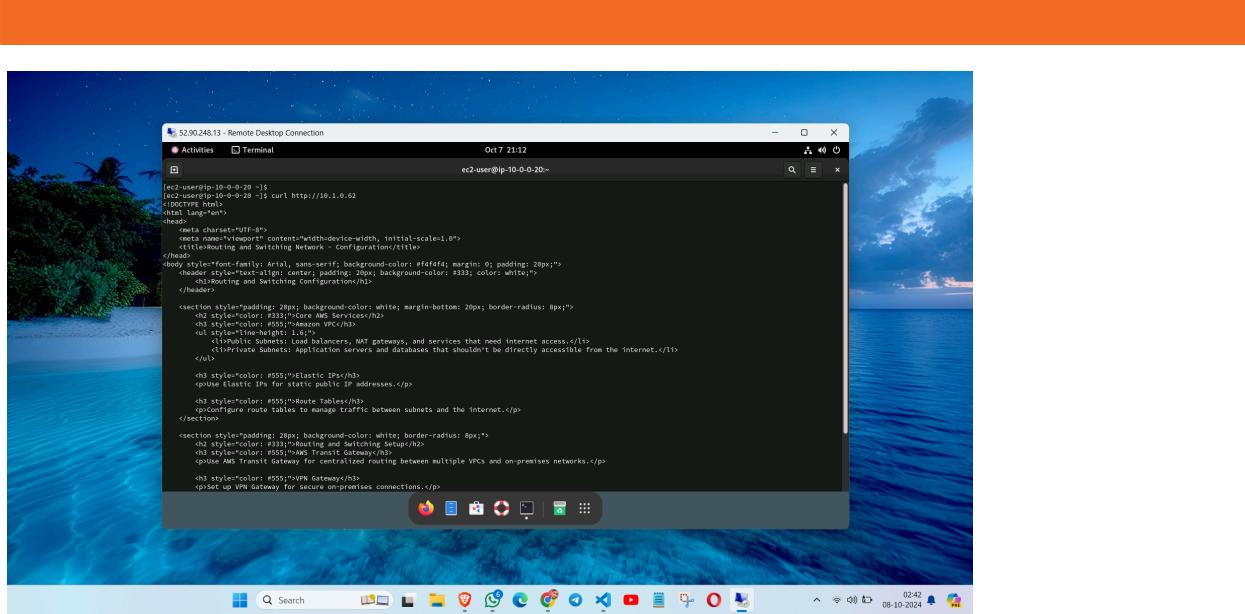
8.3 Configure the attachments for the 2 VPCs



8.4 To test the connectivity open the EC2 instance ping the ip addresses of different VPC.







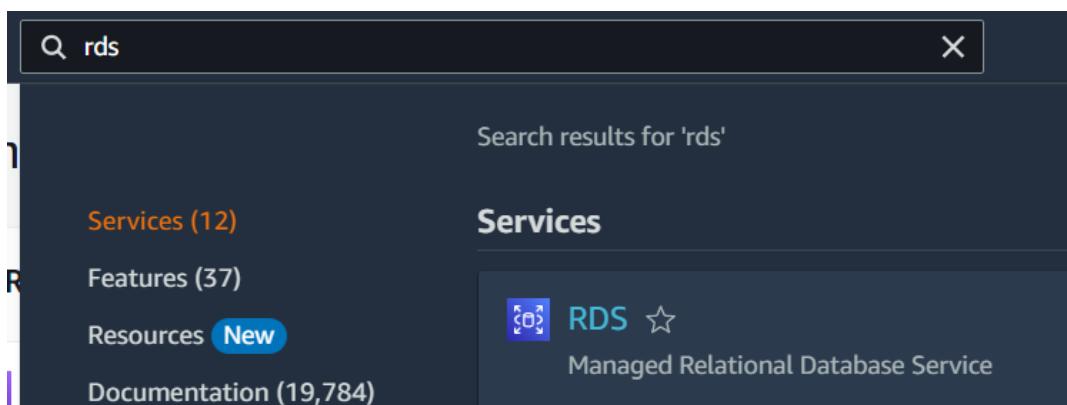
9. RDS Configuration

RDS Configuration:

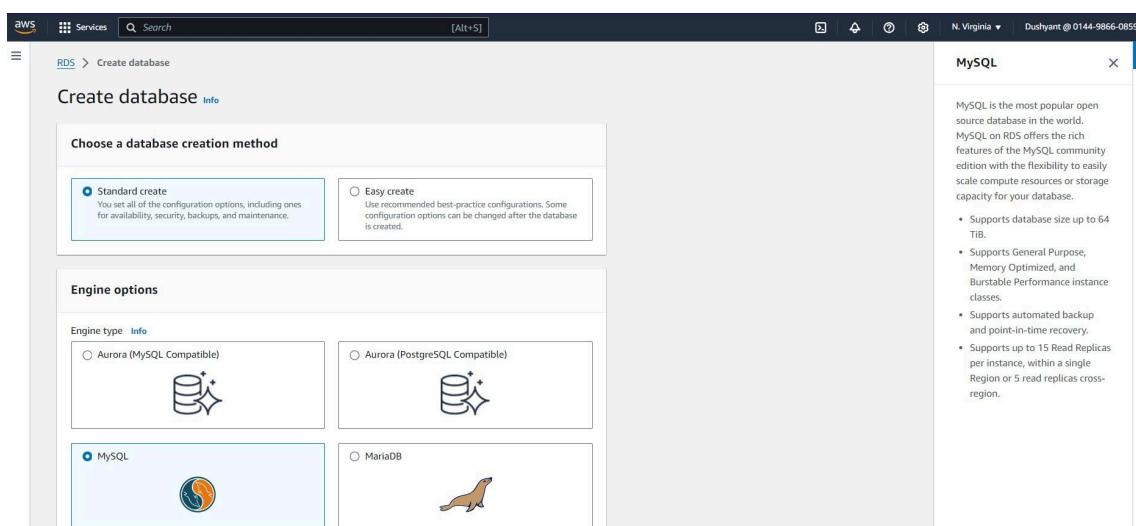
- Click on Databases and choose the Standard creation method.
- Select the MySQL database engine and specify the DB details.
- Add the VPC and custom as well as default security groups and configure the backup snapshots.
- Also configure the Read replica of the RDS in different region

Step 9: Launch RDS to build the database

9.1 In Dashboard, search for RDS service, click **RDS** to open it.



9.2 Keep the database creation **Standard create** and select engine as **MySQL**.



9.3 Now, under the templates select **Free tier**, with the db instance as **rds-r1**.

The screenshot shows the AWS RDS MySQL template creation interface. In the 'Templates' section, the 'Free tier' option is selected. The 'Availability and durability' section lists deployment options: Multi-AZ DB Cluster, Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot), and Single DB instance (not supported for Multi-AZ DB cluster snapshot). The 'Settings' section shows the DB instance identifier set to 'rds-r1'. A right-hand sidebar provides details about MySQL on RDS.

9.4 Add the credentials for database creation with password

The screenshot shows the AWS RDS MySQL credential settings interface. It displays the master username 'admin' and the password strength as 'Weak'. The 'Self managed' password option is selected. A right-hand sidebar provides details about MySQL on RDS.

9.5 Make the instance configuration to default **Bustable classes**.

Storage

Storage type General Purpose SSD (gp3)

Allocated storage 20 GiB

After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. [Learn more](#)

Advanced settings Baseline IOPS of 3,000 IOPS and storage throughput of 125 MiBps are included for allocated storage less than 400 GiB.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

9.6 Similarly, attach the VPC made before here as well and rest leave it default and click **create a database**.

Connectivity

Compute resource Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) VPC-R1 (vpc-032ba923ed1f6da8b)
4 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB subnet group Create new DB Subnet Group

Public access Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

VPC security group (firewall)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Existing VPC security groups Choose one or more options
default X SG-DB-V1 X

Availability Zone Info
No preference

RDS Proxy RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

Certificate authority - optional Info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expiry: May 26, 2061

If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

The screenshot shows the 'Create database' wizard in the AWS RDS console. It includes sections for 'Database authentication options', 'Monitoring' (with an unchecked checkbox for 'Enable Enhanced Monitoring'), 'Additional configuration' (with a note about database options), and 'Estimated monthly costs' (mentioning the Amazon RDS Free Tier). At the bottom are 'Cancel' and 'Create database' buttons.

9.7 Successfully created the database.

The screenshot shows the 'Databases' page in the AWS RDS console. It displays a table with one row for 'rds-r1', which is listed as 'Available' and running on 'db.t4g.micro'. There are tabs for 'Group resources', 'Actions', 'Restore from S3', and 'Create database'. A success message at the top says 'Successfully created database rds-r1'.

9.8 Wait for some time then the endpoint will also get updated.

The screenshot shows the detailed view for the 'rds-r1' database in the AWS RDS console. Under the 'Summary' tab, it shows the DB identifier as 'rds-r1', status as 'Available', role as 'Instance', engine as 'MySQL Community', and region as 'us-east-1b'. Under the 'Connectivity & security' tab, it shows the endpoint as 'rds-r1.ct4emo8ei3qa.us-east-1.rds.amazonaws.com' and the port as '3306'. It also lists VPC security groups: 'default (sg-047ea4af2f1c8102)' and 'SG-DB-V1 (sg-0325d4568f3a38195)', both marked as 'Active'.

9.9 Now, open EC2 instance and connect it and type the commands to connect “**mysql -h (endpoint of database) -P 3306 -u admin -p**”

```
ec2-user@ip-10-0-0-20:~$ Enter password: Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 30 Server version: 8.0.35 Source distribution Copyright (c) 2000, 2024, Oracle and/or its affiliates. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement. mysql> show databases; +-----+ | Database | +-----+ | information_schema | | mysql | | performance_schema | | sys | +-----+ 4 rows in set (0.00 sec) mysql> CREATE DATABASE Attack_Detection; Query OK, 1 row affected (0.01 sec) mysql> CREATE DATABASE Network_Traffic; Query OK, 1 row affected (0.01 sec) mysql> CREATE DATABASE System_Resources; Query OK, 1 row affected (0.00 sec) mysql> CREATE DATABASE Incident_Response; Query OK, 1 row affected (0.01 sec)
```

9.10 Now, create a **Replica DB instance** for processing the primary DB data into it.

The screenshot shows the AWS RDS 'Create read replica' configuration page. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Search' bar, and user info 'Dushyant @ 0144-9866-0859'. Below the navigation is a breadcrumb trail: 'RDS > Databases > Create read replica'. The main section is titled 'Create read replica' with a sub-instruction: 'You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB security groups and DB parameter groups.' There are two tabs: 'Settings' (selected) and 'Instance configuration'. Under 'Settings', the 'Replica source' dropdown is set to 'rds-r1' (Role: Instance). The 'DB instance identifier' field contains 'rds-r1-replica'. Under 'Instance configuration', the 'DB instance class' dropdown is set to 'Info', with 'Hide filters' and 'Include previous generation classes' options below it. The entire interface has a light gray background with white and blue text.

AWS Region

Storage

Storage type [Info](#)
Provisioned IOPS SSD (io2) storage volumes are now available.
General Purpose SSD (gp3)
Performance scales independently from storage

Allocated storage [Info](#)
20 GiB
Minimum: 20 GiB. Maximum: 6,144 GiB

Storage configuration upgrade [Info](#)
 Storage file system configuration upgrade
RDS recommends a storage file system configuration upgrade for your selected database instance.

You are on the latest storage configuration.

Availability

Deployment options [Info](#)
The following deployment options are limited to those supported by the engine.

Multi-AZ DB Cluster
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

Multi-AZ DB instance
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

Single DB instance
Creates a writer DB instance with no reader DB instances.

Connectivity

Network type [Info](#)
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

DB subnet group [Info](#)
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

9.11 Successfully created the replica database '**rds-r1-replica**'. Similarly create one more replica.

Amazon RDS

Databases

Databases (3)

DB identifier	Status	Role	Engine	Region &...	Size	Recommendations
rds-r1	Available	Primary	MySQL Com...	us-east-1b	db.t4g.micro	2 Infor
rds-r1-replica	Available	Replica	MySQL Com...	us-east-1b	db.t4g.micro	1 Infor
rds-r2	Available	Primary	MySQL Com...	us-east-1a	db.t4g.micro	2 Infor

Introducing Aurora I/O-Optimized
Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% cost savings for I/O-intensive applications.

Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

Databases (1)

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations
rds-r2-replica	Available	Replica	MySQL Co...	ap-south-1b	db.t4g.mi...	1 Informational

9.12 To keep the backup of the database also created snapshots of the RDS.

Snapshots

Snapshot name	Engine version	DB instance or cluster	Snapshot creation time	DB instance created time	Status	Progress
rds:rds-r1-2024-10-08-07-40	8.0.35	rds-r1	October 08, 2024, 13:10 (UTC+05:30)	October 08, 2024, 13:08 (UTC+05:30)	Available	Complete
rds:rds-r1-2024-10-08-10-19	8.0.35	rds-r1	October 08, 2024, 15:49 (UTC+05:30)	October 08, 2024, 13:08 (UTC+05:30)	Available	Complete
rds:rds-r2-2024-10-08-09-50	8.0.35	rds-r2	October 08, 2024, 15:20 (UTC+05:30)	October 08, 2024, 15:19 (UTC+05:30)	Available	Complete
rds:rds-r2-2024-10-08-10-19	8.0.35	rds-r2	October 08, 2024, 15:49 (UTC+05:30)	October 08, 2024, 15:19 (UTC+05:30)	Available	Complete

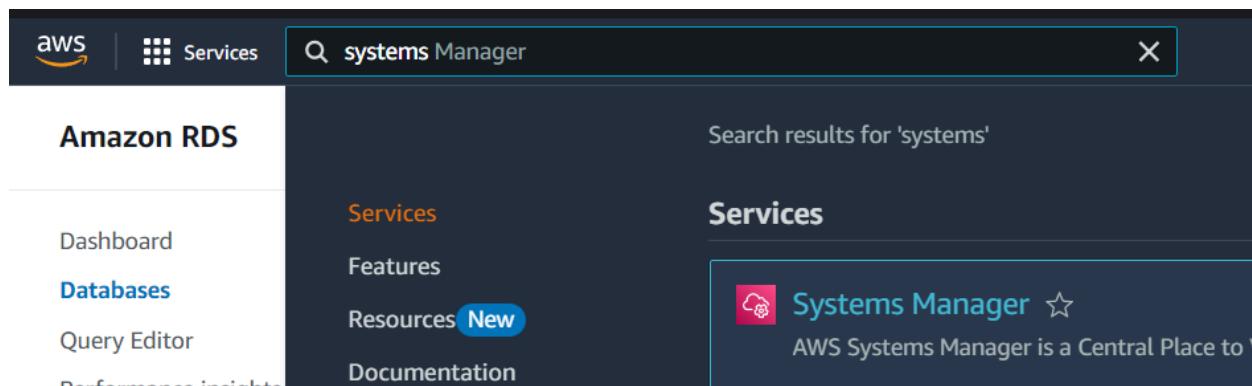
10. Systems Manager Configuration

Systems Manager Configuration:

- Go to the EC2 and connect via SSM Agent and check the status of the instance.
- Then go to Systems Manager and streamline the fleet manager in Node management to understand the file system hierarchy used in CentOS configuration

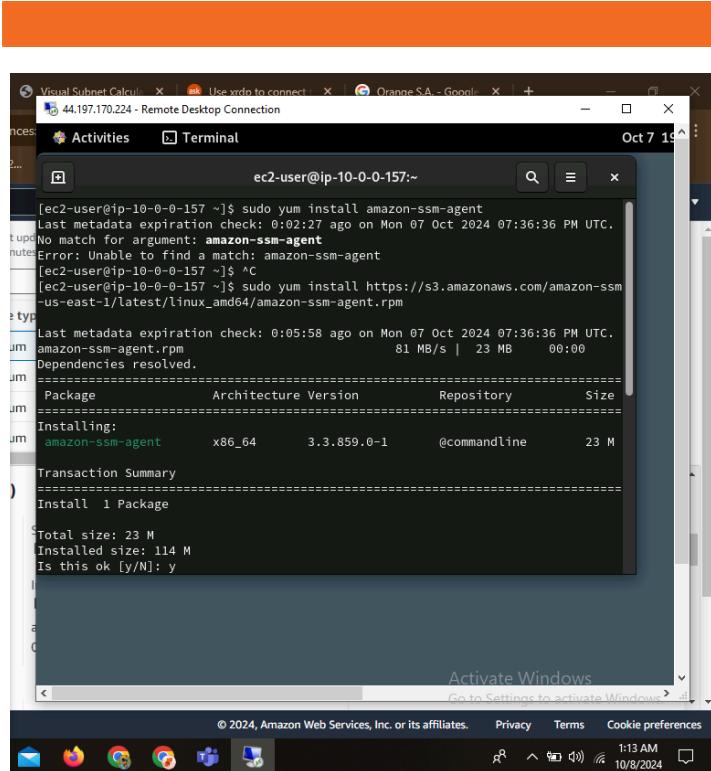
Step 10: Launch Fleet Manager under Node Management in Console

10.1 In Dashboard, search for Systems Manager service, click **Systems Manager** to open it.



10.2 Access the file structure hierarchy to do that SSM Agent has to be configured in EC2.
Type in the command:

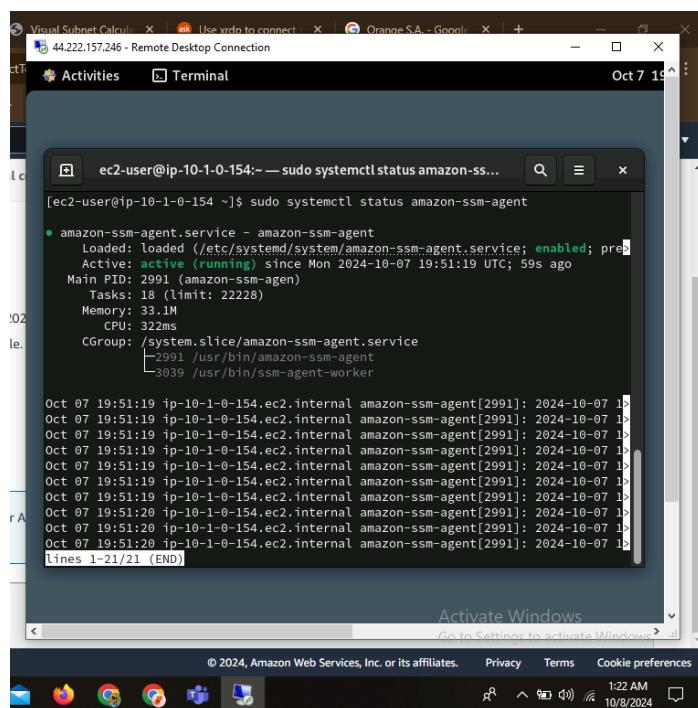
```
sudo yum install amazon-ssm-agent
```



```
[ec2-user@ip-10-0-0-157 ~]$ sudo yum install amazon-ssm-agent
Last metadata expiration check: 0:02:27 ago on Mon 07 Oct 2024 07:36:36 PM UTC.
Error: No match for argument: amazon-ssm-agent
Error: Unable to find a match: amazon-ssm-agent
[ec2-user@ip-10-0-0-157 ~]$ sudo yum install https://s3.amazonaws.com/amazon-ssm-us-east-1/latest/linux_amd64/amazon-ssm-agent.rpm
Last metadata expiration check: 0:05:58 ago on Mon 07 Oct 2024 07:36:36 PM UTC.
Dependencies resolved.
Package           Architecture Version      Repository      Size
amazon-ssm-agent.x86_64   3.3.859.0-1    @commandline   23 M
Transaction Summary
Install  1 Package
Total size: 23 M
Installed size: 114 M
Is this ok [y/N]: y
```

10.3 To check the status of the SSM Agent service type in the command:

```
sudo systemctl status amazon-ssm-agent
```

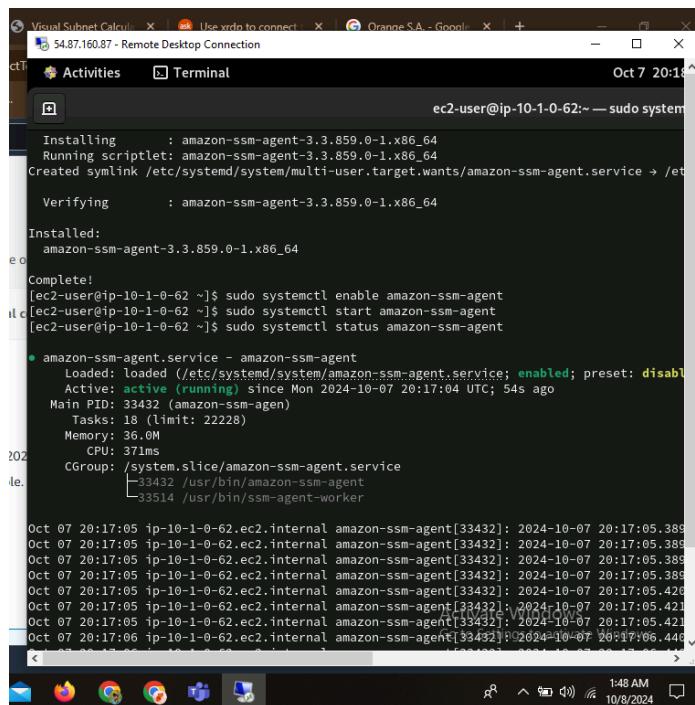


```
[ec2-user@ip-10-1-0-154 ~]$ sudo systemctl status amazon-ssm-agent
● amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; pre
   Active: active (running) since Mon 2024-10-07 19:51:19 UTC; 59s ago
     Main PID: 2991 (amazon-ssm-agent)
        Tasks: 18 (limit: 22228)
       Memory: 33.1M
          CPU: 322ms
         CGroup: /system.slice/amazon-ssm-agent.service
                   ├─2991 /usr/bin/amazon-ssm-agent
                   ├─3039 /usr/bin/ssm-agent-worker
                   └─3039 /usr/bin/ssm-agent-worker

Oct 07 19:51:19 ip-10-1-0-154.ec2.internal amazon-ssm-agent[2991]: 2024-10-07 19:51:19
Oct 07 19:51:20 ip-10-1-0-154.ec2.internal amazon-ssm-agent[2991]: 2024-10-07 19:51:20
Oct 07 19:51:20 ip-10-1-0-154.ec2.internal amazon-ssm-agent[2991]: 2024-10-07 19:51:20
Oct 07 19:51:20 ip-10-1-0-154.ec2.internal amazon-ssm-agent[2991]: 2024-10-07 19:51:20
[lines 1-21/21 (END)]
```

10.4 Now, enable and start the service for that type in the command:

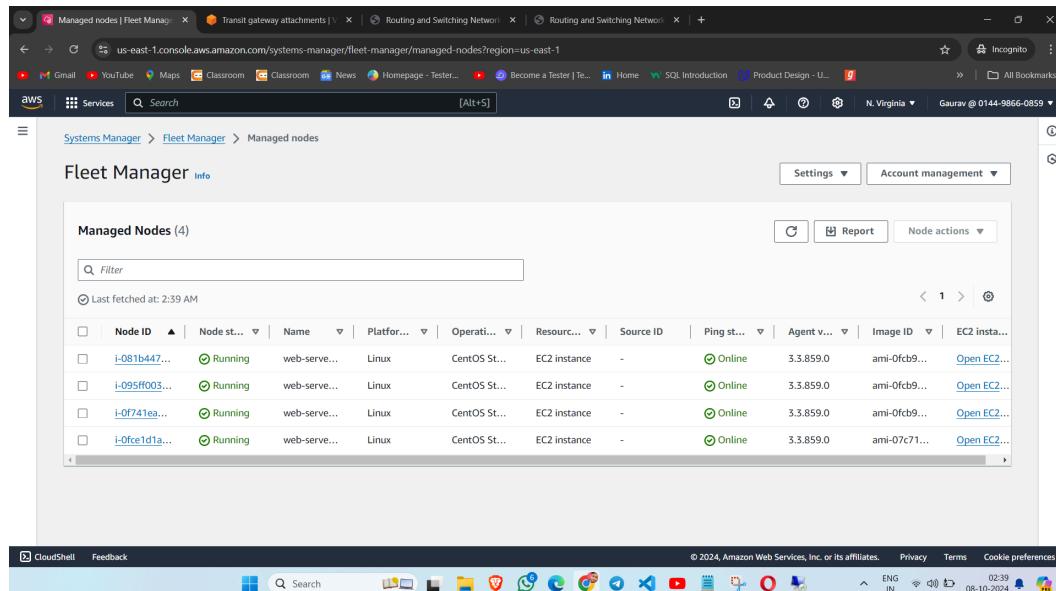
```
sudo systemctl enable amazon-ssm-agent  
sudo systemctl start amazon-ssm-agent
```



A screenshot of a Lambda function execution interface. The terminal window shows the command output for enabling and starting the Amazon SSM Agent service. The output includes the installation of the agent package, creation of a symlink, verification of the package, and the status of the service after enabling and starting it.

```
Installing      : amazon-ssm-agent-3.3.859.0-1.x86_64  
Running scriptlet: amazon-ssm-agent-3.3.859.0-1.x86_64  
Created symlink /etc/systemd/system/multi-user.target.wants/amazon-ssm-agent.service → /etc/  
  
Verifying      : amazon-ssm-agent-3.3.859.0-1.x86_64  
  
Installed:  
amazon-ssm-agent-3.3.859.0-1.x86_64  
e o  
Complete!  
[ec2-user@ip-10-1-0-62 ~]$ sudo systemctl enable amazon-ssm-agent  
[ec2-user@ip-10-1-0-62 ~]$ sudo systemctl start amazon-ssm-agent  
[ec2-user@ip-10-1-0-62 ~]$ sudo systemctl status amazon-ssm-agent  
  
● amazon-ssm-agent.service - amazon-ssm-agent  
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; preset: disabled)  
   Active: active (running) since Mon 2024-10-07 20:17:04 UTC; 54s ago  
     Main PID: 33432 (amazon-ssm-agent)  
       Tasks: 18 (limit: 22228)  
         Memory: 36.0M  
        CPU: 371ms  
       CGroup: /system.slice/amazon-ssm-agent.service  
              └─33432 /usr/bin/amazon-ssm-agent  
                  ├─33514 /usr/bin/ssm-agent-worker  
  
Oct 07 20:17:05 ip-10-1-0-62.ec2.internal amazon-ssm-agent[33432]: 2024-10-07 20:17:05.389  
Oct 07 20:17:05 ip-10-1-0-62.ec2.internal amazon-ssm-agent[33432]: 2024-10-07 20:17:05.420  
Oct 07 20:17:05 ip-10-1-0-62.ec2.internal amazon-ssm-agent[33432]: 2024-10-07 20:17:05.421  
Oct 07 20:17:05 ip-10-1-0-62.ec2.internal amazon-ssm-agent[33432]: 2024-10-07 20:17:05.421  
Oct 07 20:17:06 ip-10-1-0-62.ec2.internal amazon-ssm-agent[33432]: 2024-10-07 20:17:06.446
```

10.5 Now, go to the Fleet Manager under Node Management to check the files.



A screenshot of the AWS Fleet Manager interface. The page displays a list of managed nodes, showing 4 entries. Each node entry includes details such as Node ID, Name, Platform, Operation, Resource ID, Ping status, Agent version, Image ID, and EC2 instance ID. Buttons for 'Open EC2...' and 'Report' are also present.

Node ID	Name	Platform	Operation	Resource ID	Ping status	Agent version	Image ID	EC2 instance ID
i-081b447...	Running	web-server...	Linux	CentOS St...	EC2 instance	-	3.3.859.0	ami-0fcfb9...
i-095ff003...	Running	web-server...	Linux	CentOS St...	EC2 instance	-	3.3.859.0	ami-0fcfb9...
i-0f741ea...	Running	web-server...	Linux	CentOS St...	EC2 instance	-	3.3.859.0	ami-0fcfb9...
i-0fce1d1...	Running	web-server...	Linux	CentOS St...	EC2 instance	-	3.3.859.0	ami-07c71...

The screenshot shows the AWS Systems Manager Fleet Manager interface for managing a fleet of nodes. The current view is for a specific node named "web-server-rb1".

The left sidebar displays the navigation path: Systems Manager > Fleet Manager > Managed nodes > i-081b4474868308ee9 > File system.

The main content area is titled "File system" and lists the following file systems:

File name	Date modified	Owner	Group	Mode	Size
afs	Mon, 09 Aug 2021 20:40:26 GMT	root	root	dr-xr-xr-x	-
bin	Mon, 07 Oct 2024 20:54:43 GMT	root	root	dr-xr-xr-x	-
boot	Mon, 07 Oct 2024 15:30:11 GMT	root	root	dr-xr-xr-x	-
data	Mon, 06 Jun 2022 14:00:56 GMT	root	root	drwxr-xr-x	-
dev	Mon, 07 Oct 2024 18:10:12 GMT	root	root	drwxr-xr-x	-
etc	Mon, 07 Oct 2024 20:54:31 GMT	root	root	drwxr-xr-x	-
home	Tue, 21 Feb 2023 09:48:04 GMT	root	root	drwxr-xr-x	-
lib	Tue, 21 Feb 2023 09:58:03 GMT	root	root	dr-xr-xr-x	-
lib64	Mon, 07 Oct 2024 20:09:16 GMT	root	root	dr-xr-xr-x	-

On the right side of the interface, there is a "Node actions" dropdown menu. The bottom of the screen shows the Windows taskbar with various pinned icons and the system clock indicating it's 02:39 on 08-10-2024.

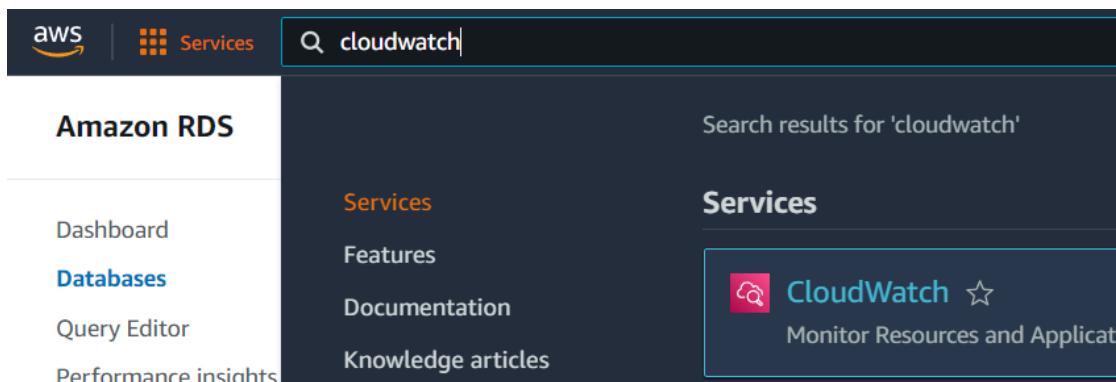
11. CloudWatch Configuration

CloudWatch Configuration:

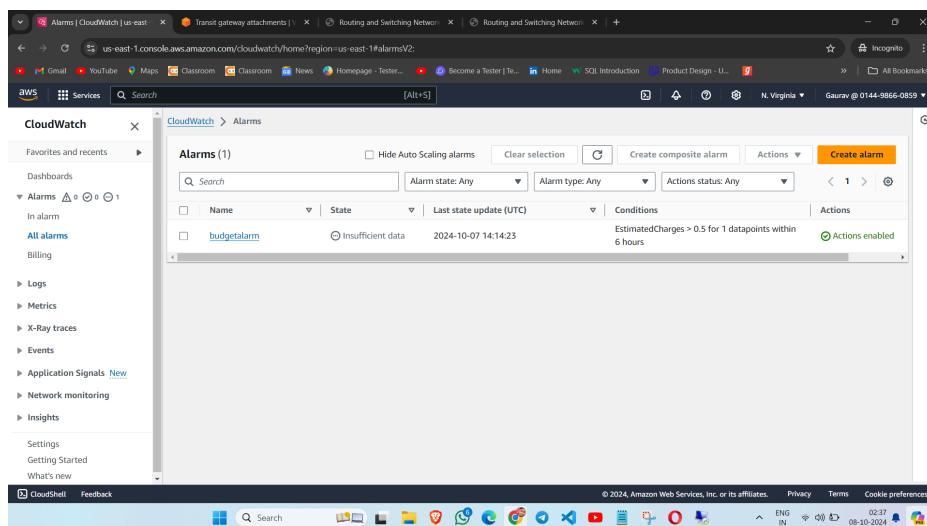
- Create an **Alarm** and choose the metric as Billing and EstimatedCharges and threshold value.
- Create a **Dashboard** and add the widgets to visualize the metrics.
- Create the **Logs** and stream for EC2 instances in the log groups

Step 11: Launch CloudWatch to monitor the Data

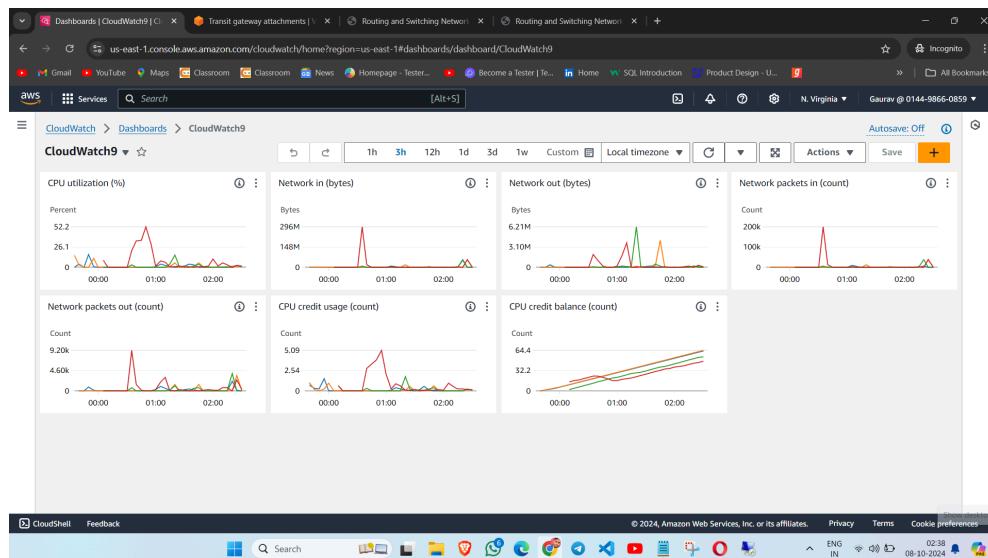
11.1 In Dashboard, search for CloudWatch service, click **CloudWatch** to open it.



11.2 Create an Alarm and choose the metric as Billing and EstimatedCharges and threshold value.



11.3 Create the Dashboard and add the widgets to visualize the metrics.



11.4 Create the Logs and stream for EC2 instances in the log groups.

The screenshot shows the "Create log group" wizard in the AWS CloudWatch service. The left sidebar shows the navigation path: CloudWatch > Log groups > Create log group. The main form has the following fields:

- Log group details**:
 - Log group name: data-logs
 - Retention setting: Never expire
 - Log class: info
 - KMS key ARN - optional: (empty)
- Tags**:
 - Info: You can assign tags to an Amazon Web Services resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your Amazon Web Services costs.
 - No tags are associated with this log group.
 - Add new tag: (button)
 - You can add up to 50 more tags.

At the bottom are "Cancel" and "Create" buttons.

11.5 Assign the name and associate the metrics, subscription filters.

The screenshot shows the AWS CloudWatch Log groups page. The left sidebar shows the navigation path: CloudWatch > Log groups. The main table lists the following log groups:

Log group	Log class	Anomaly d...	Data pr...	Sensiti...	Retent...	Me...
/aws/lambda/vid-aud-function	Standard	Configure	-	-	Never expire	-
data-logs	Standard	Configure	-	-	Never expire	-

At the top right of the table area is a "Create log group" button.

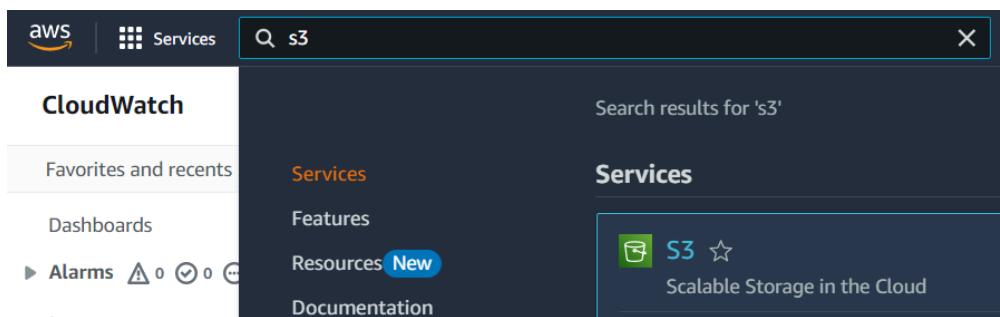
12. S3 Configuration

S3 Configuration:

- Create S3 buckets for capturing the flow logs of 2 VPCs.
- Go to the VPC and configure the flow filtering the traffic to ALL.
- Assigning the destination as S3 bucket which we have created.

Step 12: Build VPC Flow logs to S3 Destination

12.1 In Dashboard, search for S3 service, click **S3** to open it.



12.2 Create the buckets to feed the data.

<input type="radio"/> fl-data-vpc-r1	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 8, 2024, 10:23:17 (UTC+05:30)
<input type="radio"/> fl-data-vpc-r2	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 8, 2024, 10:48:50 (UTC+05:30)

12.3 Go to the VPC and configure the flow filtering the traffic to ALL.

A screenshot of the AWS VPC dashboard. The left sidebar shows 'VPC dashboard', 'EC2 Global View', and 'Virtual private cloud' sections. The main area displays a table titled 'Your VPCs (1/3) Info' with three rows: 'VPC-R1', 'VPC-R2', and a third row starting with '...'. The 'Actions' column for VPC-R1 has a context menu open, with 'Create flow log' highlighted. Other options in the menu include 'Create default VPC', 'Edit VPC settings', 'Edit CIDRs', 'Manage middlebox routes', 'Manage tags', and 'Delete VPC'. The top right corner shows the user 'Dushyant @ 0144-9866-0859'.

12.4 Add the destination to S3 bucket

The screenshot shows the 'Create flow log' configuration page for a VPC. The 'Selected resources' section lists 'VPC-R1' with its ID and state. In the 'Flow log settings' section, the name is set to 'fl-flow-log-01'. The 'Filter' dropdown is set to 'All'. The 'Maximum aggregation interval' is set to '1 minute'. The 'Destination' section shows 'Send to an Amazon S3 bucket' is selected. The 'S3 bucket ARN' field contains 'arnaws:s3::fl-data-vpc-r1'. The 'Log record format' section shows 'AWS default format' is selected. The 'Format preview' shows the log format: '\$(version) \$(account-id) \$(interface-id) \$(srcaddr) \$(dstaddr) \$(sport) \$(dport) \$(protocol) \$(packets) \$(bytes) \$(start) \$(end) \$(action) \$(log-status)'. The 'Log file format' section shows 'Text (default)' is selected. The 'Hive-compat-S3 prefix' section shows 'Enable' is selected. The 'Partition log by time' section shows 'Every 24 hours (default)' is selected. The 'Tags' section shows no tags are associated with the resource.

The screenshot shows the confirmation page for creating the flow log. It displays the summary of the configuration: VPC-R1, fl-flow-log-01, AWS default format, and arnaws:s3::fl-data-vpc-r1. The 'Create flow log' button is visible at the bottom.

12.5 Successfully configured the VPC flow logs with S3 destination.

The screenshot shows the AWS VPC dashboard. The 'Your VPCs' table lists three VPCs: VPC-R1, VPC-R2, and VPC-3. The 'Flow logs' tab for VPC-R1 is selected, showing one flow log entry: 'fl-0d6d46a81d43f3b47' with 's3' as the destination type and 'fl-data-vpc-r1' as the destination name. The 'Details' tab for VPC-R1 shows its configuration: IPv4 CIDR 10.0.0.0/24, IPv6 CIDR -, and DHCP options dopt-0c.

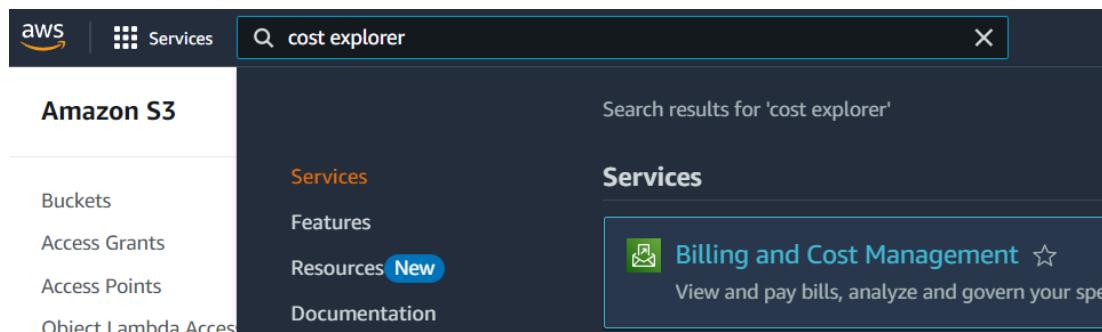
13. Cost Explorer Configuration

Cost Explorer Configuration:

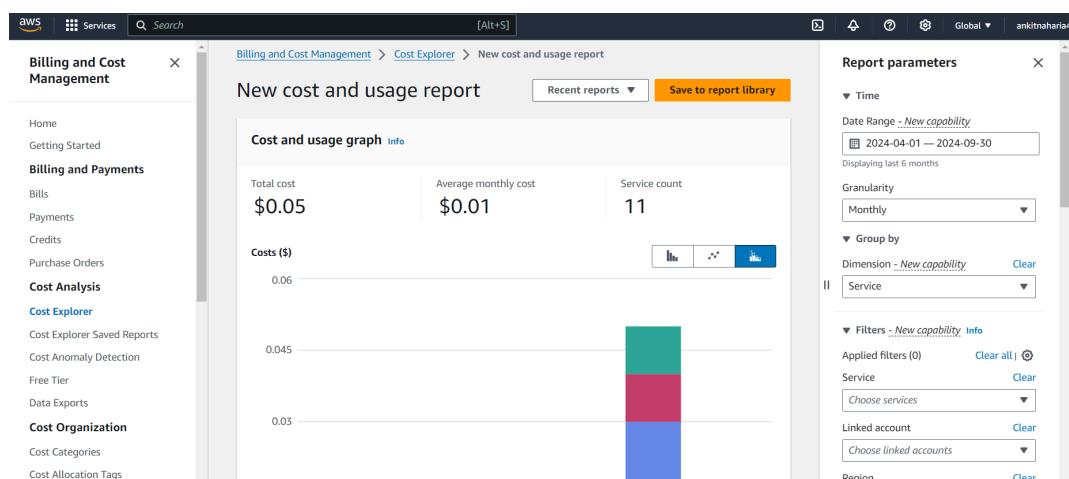
- Go to Cost Explorer and just enable it.
- Once enabled, visualize the cost of daily, cost allocation, service breakdowns.
- Option to filter the time period and services to analyze.

Step 13: Enable the Cost Explorer in Console

13.1 In Dashboard, search for Cost Explorer service, click **Billing and Cost management** to open it.



13.2 Make a report of it by clicking Save to report library.



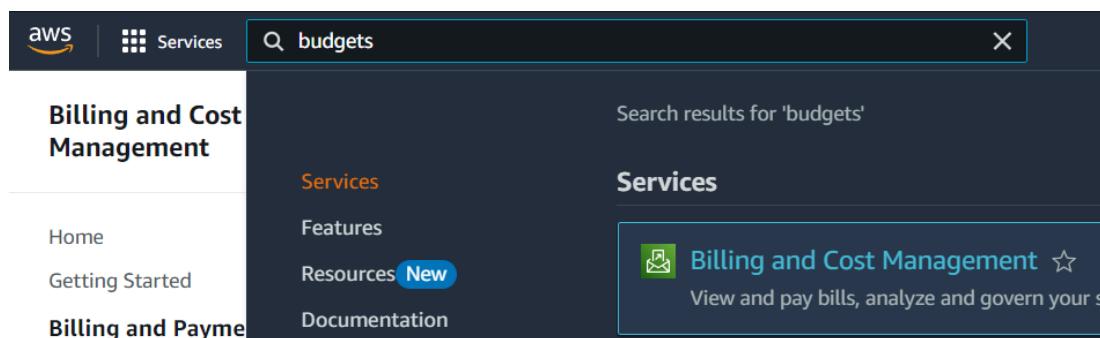
14. Budget Configuration

Budgets Configuration:

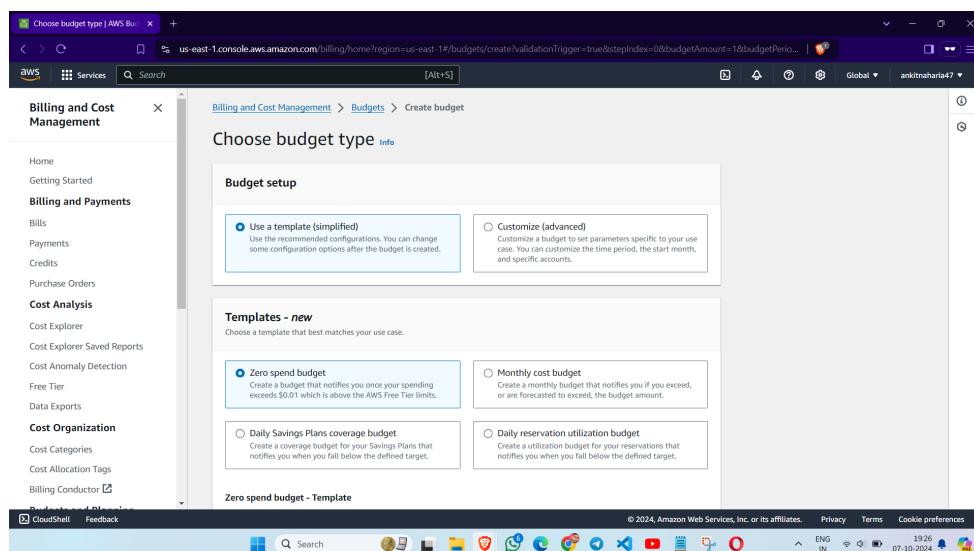
- Click on **Budgets** under the **Billing** section, select the Cost budget type and budget period monthly.
- Similarly create one more Budget for Savings plan budget type and budget period daily.

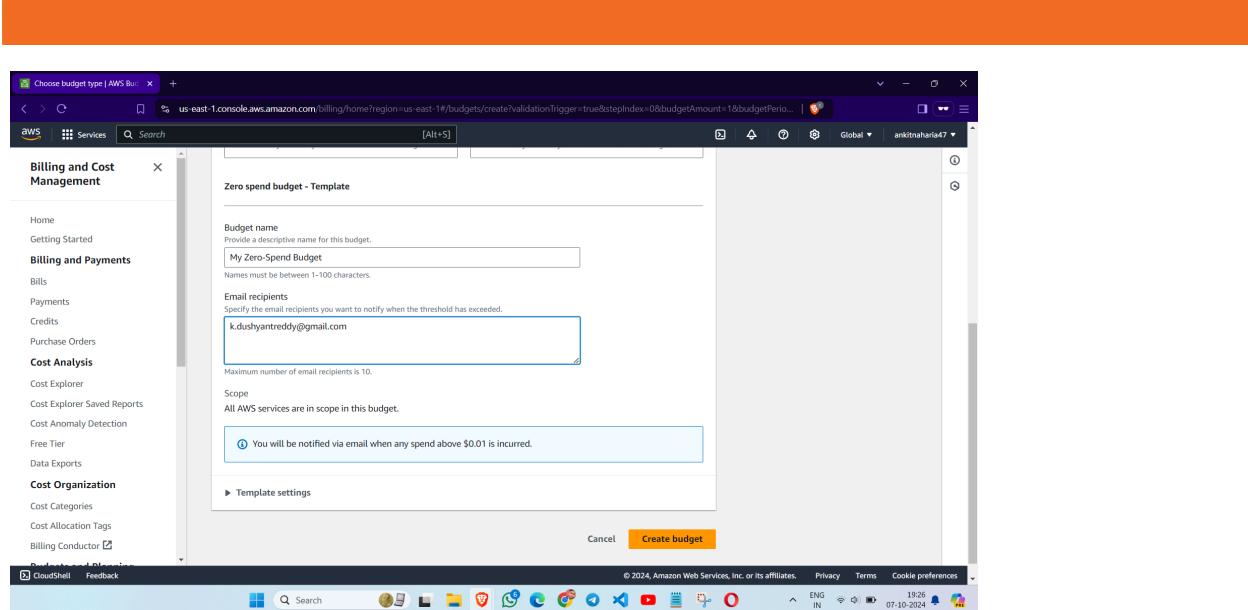
Step 14: Allotting the Budget in Console

14.1 In Dashboard, search for Budgets service, click **Billing and Cost management** to open it.



14.2 Choose the Budget type and add the email ID to confirm.





14.3 Create one more Budget of type Daily budget.

Name	Thresholds	Budget	Amount used	Forecasted ...	Current vs. budgeted
Daily_budget	Exceeded (1)	100.00%	0.00%	-	0.00%
My Zero-Spend Budget	Exceeded (1)	\$1.00	\$53.89	\$99.76	5388.90%

14.4 Received the mail for exceeding the threshold

AWS Budgets Notification
AWS Account 014498660859
October 07, 2024

Budget Name	Budget Type	Budgeted Amount	Alert Type	Alert Threshold	ACTUAL Amount
My Zero-Spend Budget	Cost	\$1.00	ACTUAL	> \$0.01	\$0.05

Go to the AWS Budgets dashboard

15. SNS Configuration

SNS Configuration:

- Navigate to **SNS**, create a Topic of standard type and assign a name for it.
- Go to CloudWatch alarms under Notification choose the SNS topic which you have created and configure via email.

Step 15: Build CloudWatch Alarm subscribed to SNS Email

15.1 Create a Topic of standard type named '**Default_CloudWatch_Alarm_Topic**'.

The screenshot shows the AWS SNS Topics page. On the left, there's a sidebar with links for Dashboard, Topics (which is selected), Subscriptions, Mobile (Push notifications, Text messaging (SMS)), and AWS services (FC2, CloudWatch, Systems Manager, Billing and Cost Management). The main content area has a header 'Topics (3)' with buttons for Edit, Delete, Publish message, and Create topic. Below is a table with columns Name, Type, and ARN. The topics listed are:

Name	Type	ARN
complete	Standard	arn:aws:sns:us-east-1:014498660859:complete
Default_CloudWatch_Alarms_Topic	Standard	arn:aws:sns:us-east-1:014498660859:Default_CloudWatch_Alarms_Topic
error	Standard	arn:aws:sns:us-east-1:014498660859:error

15.2 Create the Subscription of the assigning the Email to get the notification.

The screenshot shows the AWS SNS Subscription details page for the topic 'Default_CloudWatch_Alarms_Topic'. The URL in the browser is 'Subscription: e25ef94d-ad8e-4425-a050-1ef7acae644f'. The left sidebar is identical to the one in the previous screenshot. The main content area shows a table with columns ARN, Status, Protocol, and Endpoint. The subscription details are:

ARN	Status	Protocol	Endpoint
arn:aws:sns:us-east-1:014498660859:Default_CloudWatch_Alarms_Topic:e25ef94d-ad8e-4425-a050-1ef7acae644f	Confirmed	EMAIL	k.dushyantreddy@gmail.com

Below the table, there are buttons for Edit and Delete, and tabs for 'Subscription filter policy' and 'Redrive policy (dead-letter queue)'. At the bottom, there's a footer with links for cloudShell, Feedback, and various AWS services, along with system status indicators like ENG IN and 03:44 08-10-2024.

The screenshot shows the AWS SNS Subscriptions page. On the left, there's a sidebar with 'Amazon SNS' at the top, followed by 'Dashboard', 'Topics', and 'Subscriptions'. Under 'Mobile', it lists 'Push notifications' and 'Text messaging (SMS)'. The main area is titled 'Subscriptions (3)' and contains a table with columns: ID, Endpoint, Status, Protocol, and Topic. The table shows three rows:

ID	Endpoint	Status	Protocol	Topic
ccfaf151-0a54-4e5c-955...	ankitnaharia47@gmail.com	Confirmed	EMAIL	complete
e25ef94d-ad8e-4425-a0...	k.dushyanreddy@gmail...	Confirmed	EMAIL	Default_CloudWatch_Ala...
1874a57f-b9a3-4463-a6...	ankitnaharia.gaming@g...	Confirmed	EMAIL	error

15.3 Received the confirmation mail to subscribe.

The screenshot shows an email inbox with the subject 'AWS Notification - Subscription Confirmation' from 'AWS Notifications <no-reply@sns.amazonaws.com>' sent 'to me' on 'Mon, 7 Oct, 19:43 (7 days ago)'. The email content is as follows:

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:014498660859:Default_CloudWatch_Alarms_Topic

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#).

At the bottom, there are buttons for 'Reply', 'Forward', and 'AI Reply'.

16. Cost and Usage Report Configuration

Cost and Usage Report Configuration:

- Click on **Cost Management** and create a new report and assign a name.
- Configure the resource IDs and add the S3 destination and click Create Report.

Step 16: Build Report to S3 Destination

16.1 Create Cost and Usage Report with report name and adding the Resource IDs.

The screenshot shows the 'Specify report details' step of the 'Create Cost & Usage Report' wizard. The 'Report name' field contains 'Cost-Analysis'. The 'Report content' section lists several options, most of which are checked. The 'Include resource IDs' checkbox is checked. The 'Additional content' section has one option, 'Split cost allocation data', which is unchecked.

16.2 Configure S3 bucket by choosing the selector of the existing bucket and click Save.

The screenshot shows the 'S3 Bucket Selector' dialog box. It displays a list of existing S3 buckets: 'ankitnaharia', 'cost-usage-data-rpt' (which is selected and highlighted with a blue border), 'fl-data-vpc-r1', 'fl-data-vpc-r2', and 'iam-s3-bucket-ankit'. At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

16.3 Click Next and review the details and click Create report.

Configure S3 Bucket

Report delivery options

- S3 path prefix - required: usage-data
- Report data time granularity: Daily
- Report versioning: Create new report version
- Report data integration: Amazon Athena, Amazon Redshift, Amazon QuickSight
- Compression type: ZIP

Cancel Previous Next

Review and create

Default content

- Account identifiers
- Invoice and Bill Information
- Usage amount and unit
- Rates and cost
- Product attributes (instance type, operating system, and region)
- Pricing attributes (offer types and lease lengths)
- Reservation identifiers and related details (for Reserved Instances only)

Data refresh settings

Opted in

Report delivery options

- S3 bucket: cost-usage-data-rpt
- S3 path prefix: usage-data/Cost-Analysis/date-range/
- Time granularity: Daily
- Report versioning: Create new report version
- Compression type: ZIP
- File format: text/csv

Cancel Previous Create report

Report created successfully

In the next 24 hours, your first report will be delivered to the Amazon S3 bucket you configured during report creation.

Billing and Cost Management > Cost and Usage Reports

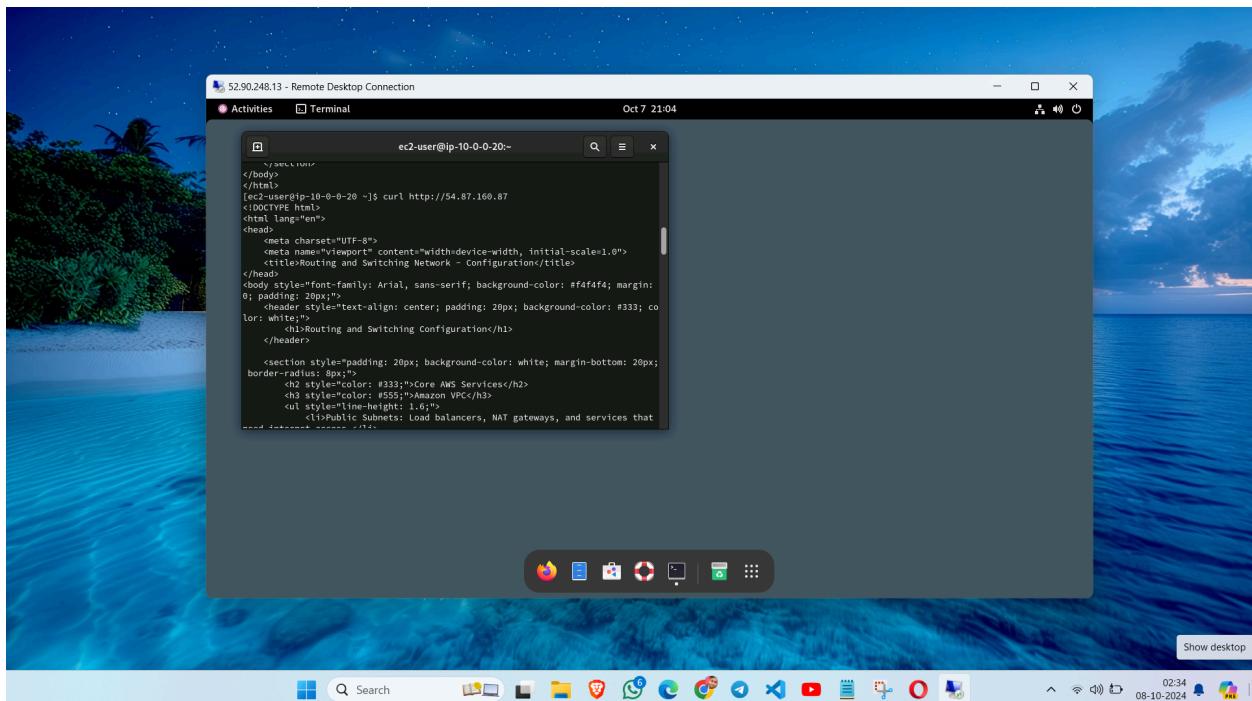
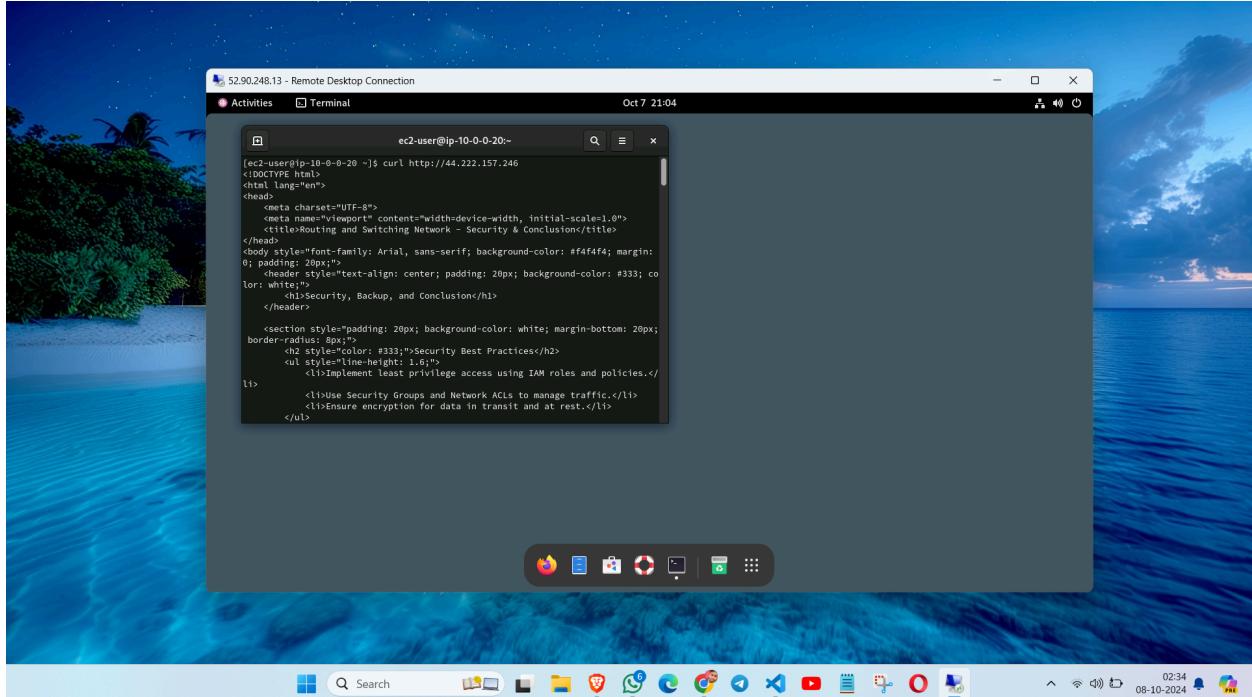
Cost and Usage Reports info

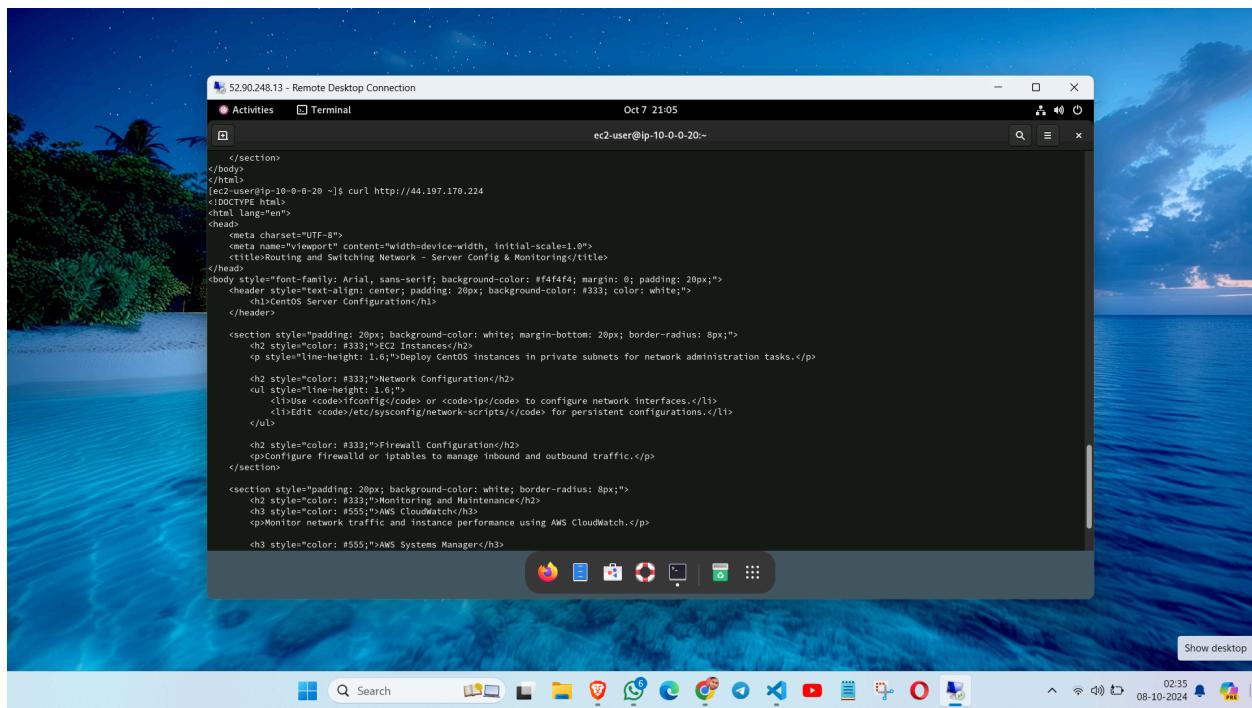
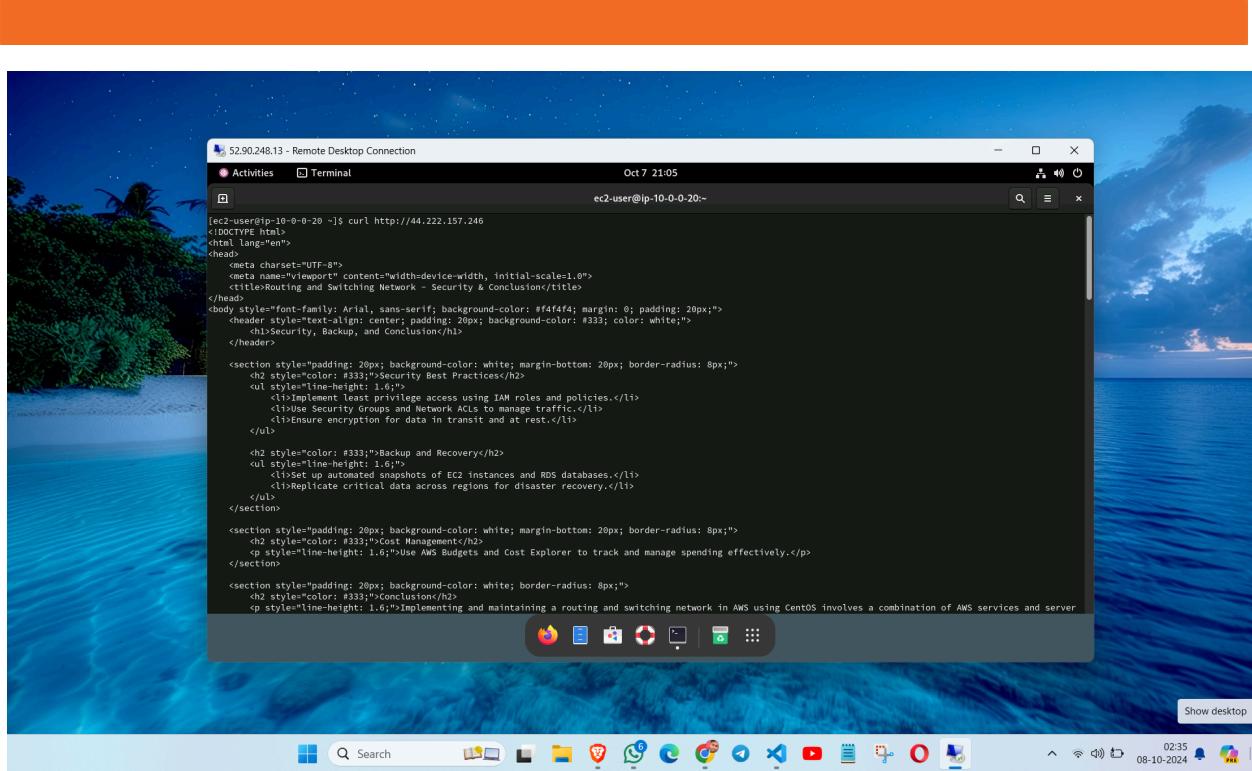
Cost and Usage Reports (1)				
Report name	S3 bucket	Time granularity	Data last refreshed	Actions
Cost-Analysis	cost-usage-data-rpt	Daily	N/A	Create report

PROJECT OUTPUT

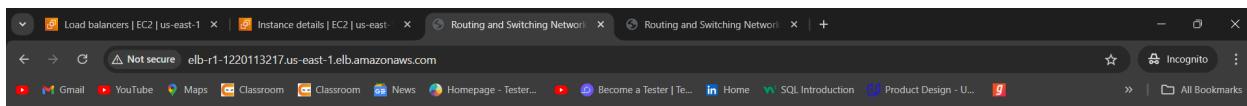
In this project, we have configured the switching and routing through:

Ping VPCs





Web Pages



Objective

The objective is to establish a centralized routing and switching network to connect various services, manage traffic, and provide secure VPN access.

Network Size

Assess the number of users, servers, and data flow to plan the required resources for the network.

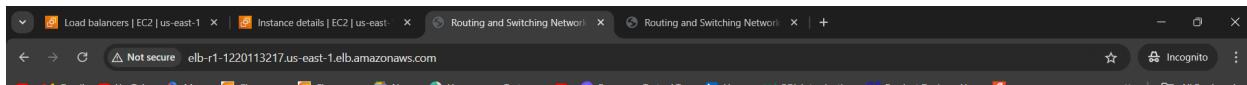
Architecture Design

Virtual Private Cloud (VPC)

Create a VPC to provide isolated networking.

Subnets

Divide the VPC into public and private subnets for better security and resource management.



EC2 Instances

Deploy CentOS instances in private subnets for network administration tasks.

Network Configuration

- Use `ifconfig` or `ip` to configure network interfaces.
- Edit `/etc/sysconfig/network-scripts/` for persistent configurations.

Firewall Configuration

Configure firewalld or iptables to manage inbound and outbound traffic.

Monitoring and Maintenance

AWS CloudWatch

Monitor network traffic and instance performance using AWS CloudWatch.

AWS Systems Manager





Load balancers | EC2 | us-east-1 | Instance details | EC2 | us-east-1 | Routing and Switching Network | Routing and Switching Network | +

Not secure elb-r2-882009896.us-east-1.elb.amazonaws.com

Gmail YouTube Maps Classroom Classroom News Homepage - Tester... Become a Tester | Te... Home SQL Introduction Product Design - U... All Bookmarks

Security, Backup, and Conclusion

Security Best Practices

- Implement least privilege access using IAM roles and policies.
- Use Security Groups and Network ACLs to manage traffic.
- Ensure encryption for data in transit and at rest.

Backup and Recovery

- Set up automated snapshots of EC2 instances and RDS databases.
- Replicate critical data across regions for disaster recovery.

Cost Management

Use AWS Budgets and Cost Explorer to track and manage spending effectively.

Show desktop

Search

ENG IN 02:36 08-10-2024

Load balancers | EC2 | us-east-1 | Instance details | EC2 | us-east-1 | Routing and Switching Network | Routing and Switching Network | +

Not secure elb-r2-882009896.us-east-1.elb.amazonaws.com

Gmail YouTube Maps Classroom Classroom News Homepage - Tester... Become a Tester | Te... Home SQL Introduction Product Design - U... All Bookmarks

Routing and Switching Configuration

Core AWS Services

Amazon VPC

- Public Subnets: Load balancers, NAT gateways, and services that need internet access.
- Private Subnets: Application servers and databases that shouldn't be directly accessible from the internet.

Elastic IPs

Use Elastic IPs for static public IP addresses.

Route Tables

Configure route tables to manage traffic between subnets and the internet.

Routing and Switching Setup

AWS Transit Gateway

Use AWS Transit Gateway for centralized routing between multiple VPCs and on-premises networks.

Search

ENG IN 02:36 08-10-2024

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:

- **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.