DevOps Shack

HIGHLY AVAILABLE WEB APPLICATION

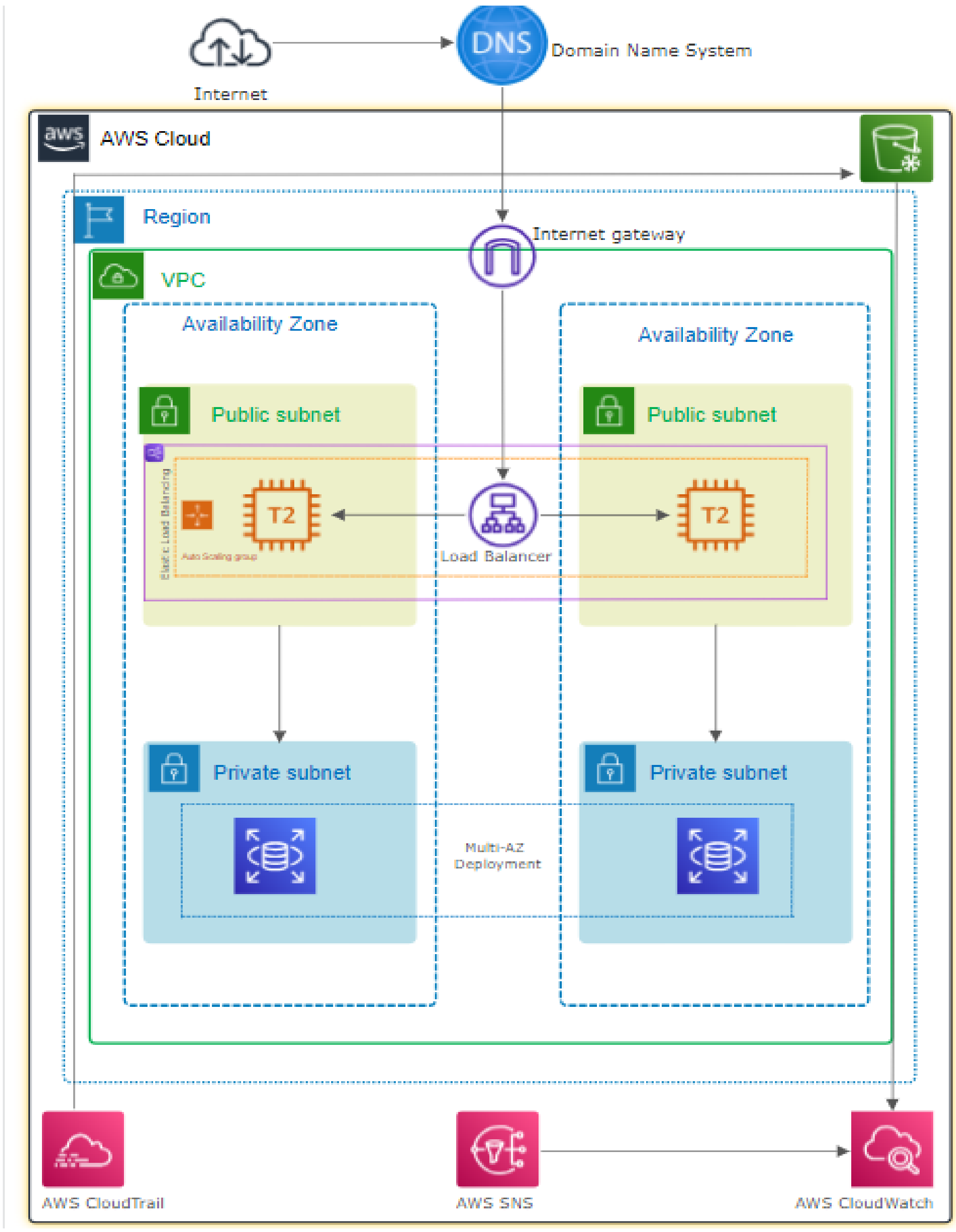
INTRODUCTION

In the dynamic environment of today's web applications, finding uninterrupted service and good performance becomes important. The need for powerful and capable devices has never been more important as businesses strive to meet the ever-increasing demands of their users. This project provides an in-depth look at creating and deploying web applications that can run on Amazon Web Services (AWS).

The importance of versatility cannot be underestimated, especially in the context of web application services, which are the lifeblood of countless businesses. Users expect uninterrupted access to services at all times, so interruptions and interruptions are unacceptable. High availability not only provides a consistent user experience, but also reduces the impact of product failure, network issues, or unexpected disasters.

Using the power of AWS, our goal is to create a design that not only meets the immediate needs of the web, but also adapts and adapts to future initial needs.The main goal of this project is to demonstrate the effectiveness of the web hosting model hosted on AWS. By applying a range of methods that include multi-region availability, load balancing, auto-scaling, and the use of other AWS services, we work to create systems that can be relied upon across projects to differentiate while ensuring service continuity.

DATA FLOW DIAGRAMS (DFD)



BACKGROUND

In the contemporary digital landscape, the relentless availability of web applications is not merely a convenience but a fundamental expectation. Users demand seamless access to services 24/7, placing an emphasis on high availability to ensure a consistently positive user experience. Simultaneously, businesses recognize that downtime can translate into significant financial losses, disrupt operations, and tarnish their reputation.

Amazon Web Services (AWS) emerges as a pivotal solution in meeting the demands of high availability for web applications. With its architecture built on multiple geographically distributed data centers called Availability Zones, AWS provides a foundation for redundancy and fault tolerance. Elastic Load Balancing optimizes the distribution of incoming traffic across instances, preventing the risk of a single point of failure. Auto Scaling, another key feature, enables dynamic adjustments of computing resources based on demand, ensuring optimal performance during peak periods and cost efficiency during troughs.

Managed database services, such as Amazon RDS, contribute to high availability by offering features like automated backups, failover support, and seamless scalability. AWS CloudFront, a content delivery network, enhances user experience by caching and delivering content from edge locations globally, reducing latency and improving responsiveness.

Crucially, AWS's management of a wide array of services, from infrastructure to security, allows businesses to focus on application development rather than the complexities of infrastructure management. The robust security features, coupled with compliance certifications, further reinforce the reliability and resilience of applications hosted on AWS.

REQUIREMENTS

System Architecture

1. **Amazon EC2 Instances:**

- Deploy two EC2 instances in a public subnet inside a VPC
- .
- Set up Auto Scaling Groups to dynamically scale instances based on demand.
- Ensure instances are distributed across different Availability Zones for enhanced fault tolerance.

2. **Load Balancer:**

- Implement a load balancer to distribute user traffic evenly across EC2 instances.
- Enhance system fault tolerance and availability through effective load balancing.

3. **Amazon RDS:**

- Establish two RDS instances in a private subnet.
- Choose different Availability Zones for each RDS instance to ensure redundancy.
- Set up database replication for real-time data synchronization.
- Enable multi-AZ deployment for improved fault tolerance and reliability.

Security Considerations

1. **Networking:**

- Utilize private and public subnets to segregate resources for improved security.

- Implement security groups and Network ACLs to control inbound and outbound traffic effectively.

2. **Encryption:**

- Enable encryption for data in transit and at rest within RDS instances.

Monitoring and Logging

1. **CloudWatch:**

- Integrate CloudWatch to monitor system metrics, logs, and performance.
- Set up alarms to proactively respond to issues and maintain optimal performance.

2. **CloudTrail:**

- Implement AWS CloudTrail to capture API calls for auditing and compliance.
- Store logs securely in an S3 bucket for centralized access.

Notification Services

1. **Amazon SNS:**

- Integrate Amazon SNS for sending notifications.
- Configure SNS to send timely alerts based on CloudWatch alarms, ensuring prompt responses to system issues.

Scalability and Redundancy

1. **Auto Scaling:**

- Implement Auto Scaling Groups to allow automatic scaling of EC2 instances based on varying demand.

2. Redundancy:

- Ensure multi-AZ deployment for both EC2 instances and RDS, contributing to system redundancy and high availability.

OVERVIEW

In this project, the goal is to construct a robust and highly available web application leveraging Amazon Web Services (AWS). The architecture is meticulously designed to ensure uninterrupted service, scalability, and reliability, with a focus on the following key components:

1. **Amazon EC2 Instances:**

Two EC2 instances are deployed across different Availability Zones in a public subnet.

Auto Scaling Groups are implemented to dynamically scale resources based on demand, ensuring optimal performance under varying workloads.

2. **Load Balancer:**

A load balancer is employed to evenly distribute incoming traffic across the EC2 instances, enhancing fault tolerance and availability.

3. **Amazon RDS:**

Two RDS instances reside in a private subnet, each in a different Availability Zone.

Database replication is implemented for real-time synchronization, ensuring data availability and reliability.

Multi-AZ deployment enhances fault tolerance, providing continuous service even in the face of potential disruptions.

4. **Security Considerations:**

Networking is optimized through private and public subnets, segregating resources for improved security.

Encryption mechanisms are implemented to secure data both in transit and at rest within RDS instances.

5. Monitoring and Logging:

CloudWatch is integrated for real-time monitoring of system metrics, logs, and overall performance.

AWS CloudTrail captures API calls, storing logs securely in an S3 bucket for auditing and compliance.

6. Notification Services:

Amazon Simple Notification Service (SNS) is utilized to send notifications, configured to alert based on CloudWatch alarms, ensuring swift responses to system issues.

Scalability and Redundancy:

Auto Scaling Groups and multi-AZ deployment contribute to system scalability and redundancy, accommodating varying workloads and enhancing fault tolerance

DESCRIPTION

1.Amazon EC2 Instances:

Role: Serve as the compute infrastructure hosting the web application.

Functionality:

Two EC2 instances are deployed, each in a different Availability Zone for fault tolerance.

Auto Scaling Groups are employed for dynamic scaling based on varying workloads.

Instances run the Ubuntu operating system and host the web application.

2. Load Balancer:

Role: Distribute incoming traffic across multiple EC2 instances.

Functionality:

Ensures even distribution of traffic, preventing overload on individual instances.

Enhances fault tolerance by redirecting traffic away from unhealthy instances.

Facilitates seamless scaling by adjusting to changes in the number of instances.

3. Amazon RDS:

Role: Host and manage the relational database for the web application.

Functionality:

Two RDS instances are deployed in different Availability Zones to ensure redundancy.

Database replication is implemented for real-time synchronization between primary and standby instances.

Multi-AZ deployment enhances fault tolerance by automatically failing over to the standby instance in case of a primary instance failure.

Supports MySQL or another relational database engine.

4. Security Considerations:

Role: Ensure the security and integrity of the entire system.

Functionality:

Utilizes private and public subnets for resource segregation.

Implements security groups to control inbound and outbound traffic to EC2 instances.

Network ACLs are configured to add an additional layer of security to the subnet.

Encryption is applied for data in transit and at rest within RDS instances, ensuring data security.

5. Monitoring and Logging:

Role: Provide real-time insights into system performance and activity.

Functionality:

CloudWatch is integrated for monitoring EC2 instances, load balancer, and RDS metrics.

Alarms are configured to notify administrators of predefined threshold breaches.

AWS CloudTrail captures API calls, providing a detailed history of changes and activities within the AWS environment.

Logs are stored securely in an S3 bucket for auditing and compliance.

6. Notification Services (Amazon SNS):

Role: Notify administrators of critical events or issues.

Functionality:

Amazon SNS is configured to send notifications based on CloudWatch alarms.

Alerts are sent to predefined recipients, ensuring timely responses to system issues.

7. Scalability and Redundancy:

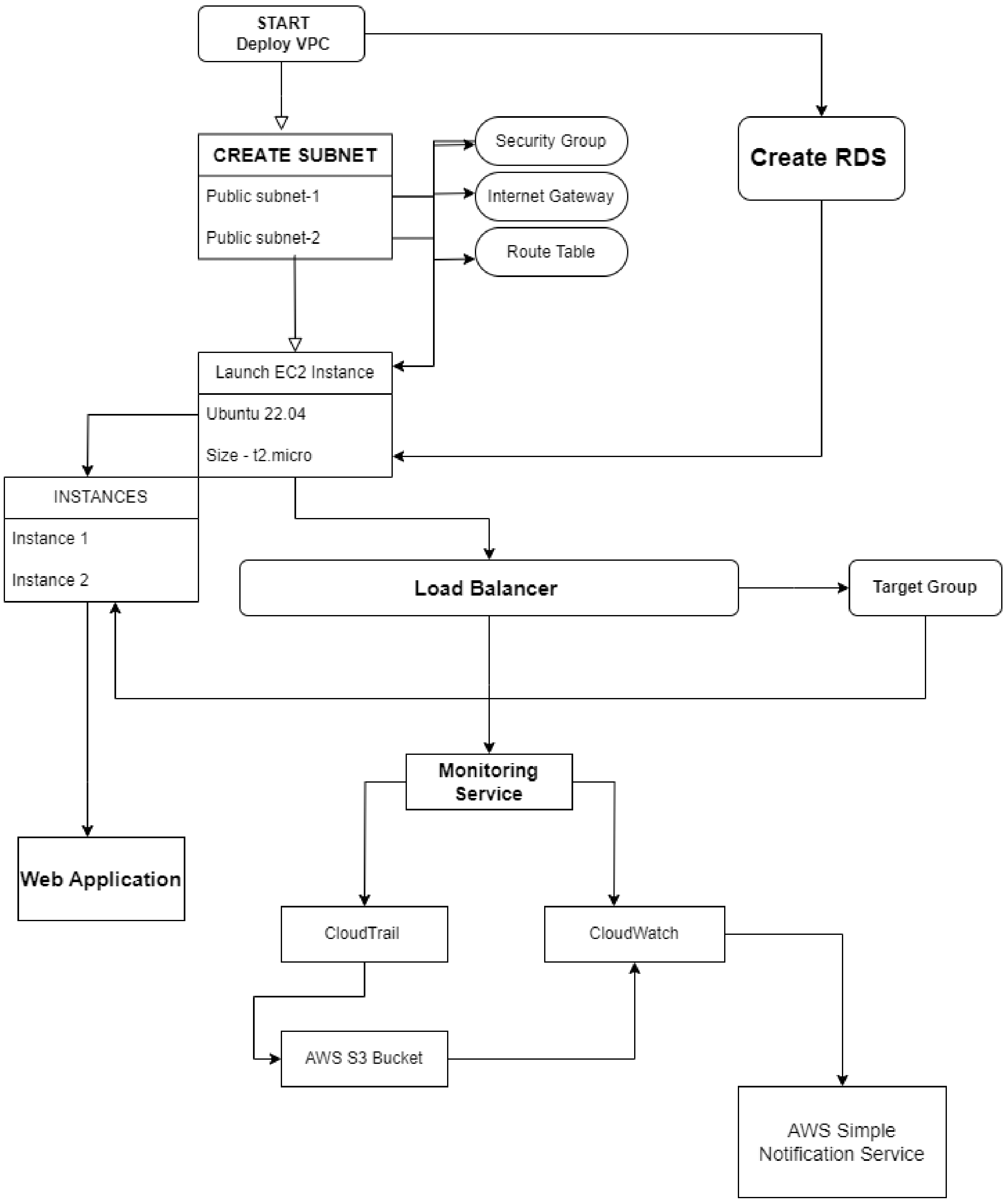
Role: Ensure the system can handle varying workloads and maintain continuous service availability.

Functionality:

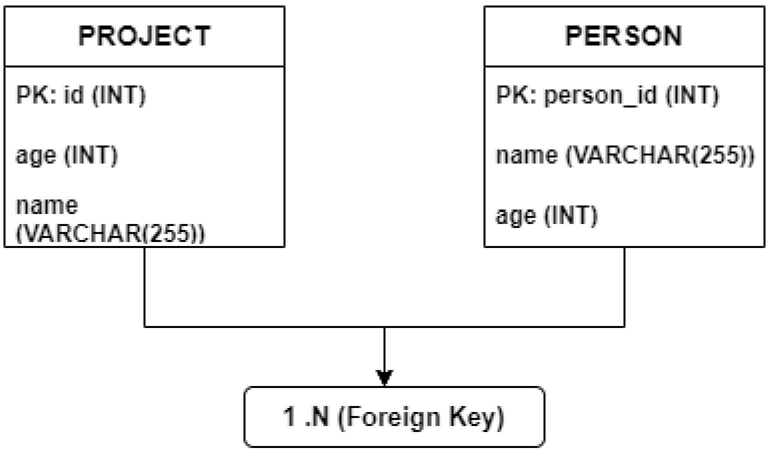
Auto Scaling Groups dynamically adjust the number of EC2 instances based on demand.

Multi-AZ deployment of EC2 instances and RDS provides redundancy and fault tolerance, minimizing the impact of potential failures.

FLOWCHART (MODULES & SUB-MODULES)



DATA DESCRIPTION (E-R DIAGRAMS)



PROJECT Table Entries		
ID	age	Name
1	25	Shubham
2	30	Raj
3	22	Abhishek
4	28	Riya
5	35	Helen
6	35	Rayan
7	40	Sayan
8	32	Aman
9	24	Reha
10	32	Mehta

SAMPLE CODE

Script 1: RDS connection from ec2

```
sudo su
sudo apt update
sudo apt install mysql-server
sudo systemctl start mysql
sudo systemctl enable mysql
mysql -h mysql.crlw2n0fdtpd.us-east-1.rds.amazonaws.com -u admin -p
Enter password - admin123
```

Script 2: RDS dummy data insertion

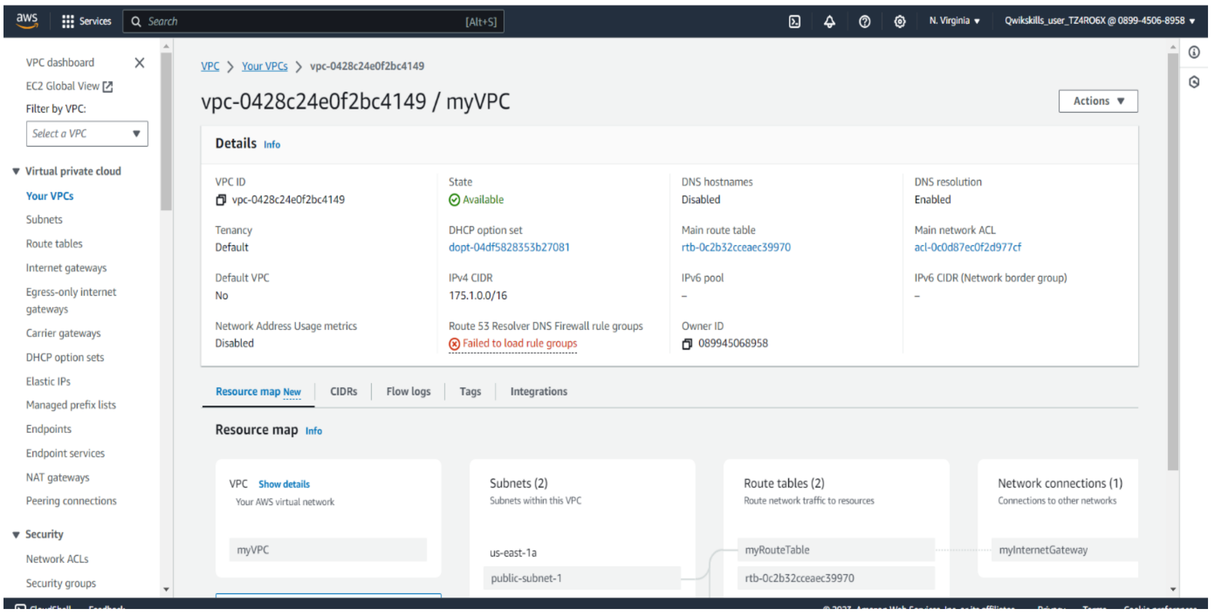
```
CREATE TABLE PROJECT (
   id INT PRIMARY KEY,
   age INT,
   name VARCHAR(255)
);

INSERT INTO PROJECT (id, age, name) VALUES
(1, 25, 'Shubham'),
(2, 30, 'Raj'),
(3, 22, 'Abhishek'),
(4, 28, 'Riya'),
(5, 35, 'Helen');

SELECT * FROM PROJECT;

INSERT INTO PROJECT (id, age, name) VALUES
(6, 35, 'Rayan'),
(7, 40, 'Sayan'),
(8, 32, 'Aman'),
(9, 24, 'Reha'),
(10, 32, 'Mehta');
```

Script 3: Apache server fetches data from RDS

```
#!/bin/bash

# Database configuration
host="mysql.crlw2n0fdtpd.us-east-1.rds.amazonaws.com"
username="admin"
password="admin123"
database="mydatabase"
table="PROJECT"

# Install required packages
sudo apt update
sudo apt install -y apache2 php libapache2-mod-php php-mysql

# Create a PHP script to fetch data from the database
sudo tee /var/www/html/index.php >/dev/null <<EOF
<?php
\$conn = new mysqli("$host", "$username", "$password", "$database");
if (\$conn->connect_error) {
    die("Connection failed: " . \$conn->connect_error);
}
\$sql = "SELECT * FROM $table";
\$result = \$conn->query(\$sql);
if (\$result->num_rows > 0) {
    echo "<table><tr><th>ID</th><th>Name</th></tr>";
    while(\$row = \$result->fetch_assoc()) {
        echo "<tr><td>" . \$row["id"] . "</td><td>" . \$row["name"] . "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}
\$conn->close();
?>
EOF

# Configure Apache to serve PHP files
sudo sed -i "s/index.html/index.php/g" /etc/apache2/mods-enabled/dir.conf

# Restart Apache
sudo systemctl restart apache2

# Display public IP
echo "Web application is now accessible at: http://$(curl -s
http://checkip.amazonaws.com)/"
```

SCREENSHOTS

Vpc



VPC > Your VPCs > vpc-0428c24e0f2bc4149

vpc-0428c24e0f2bc4149 / myVPC

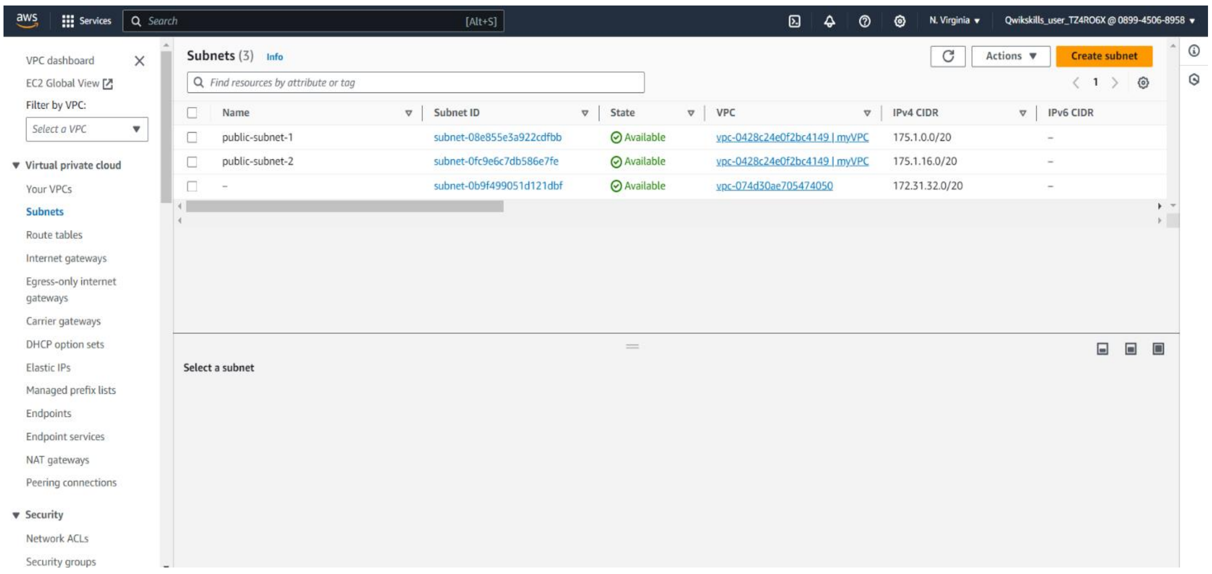
Details

VPC ID: vpc-0428c24e0f2bc4149	State: Available	DNS hostnames: Disabled	DNS resolution: Enabled
Tenancy: Default	DHCP option set: dopt-04df5828353b27081	Main route table: rtb-0c2b32cceaec39970	Main network ACL: acl-0c0d87ec0f2d977cf
Default VPC: No	IPv4 CIDR: 175.1.0.0/16	IPv6 pool: –	IPv6 CIDR (Network border group): –
Network Address Usage metrics: Disabled	Route 53 Resolver DNS Firewall rule groups: Failed to load rule groups	Owner ID: 089945068958	

Resource map

VPC: myVPC (Your AWS virtual network)
Subnets (2): us-east-1a, public-subnet-1
Route tables (2): myRouteTable, rtb-0c2b32cceaec39970
Network connections (1): myInternetGateway

Subnets

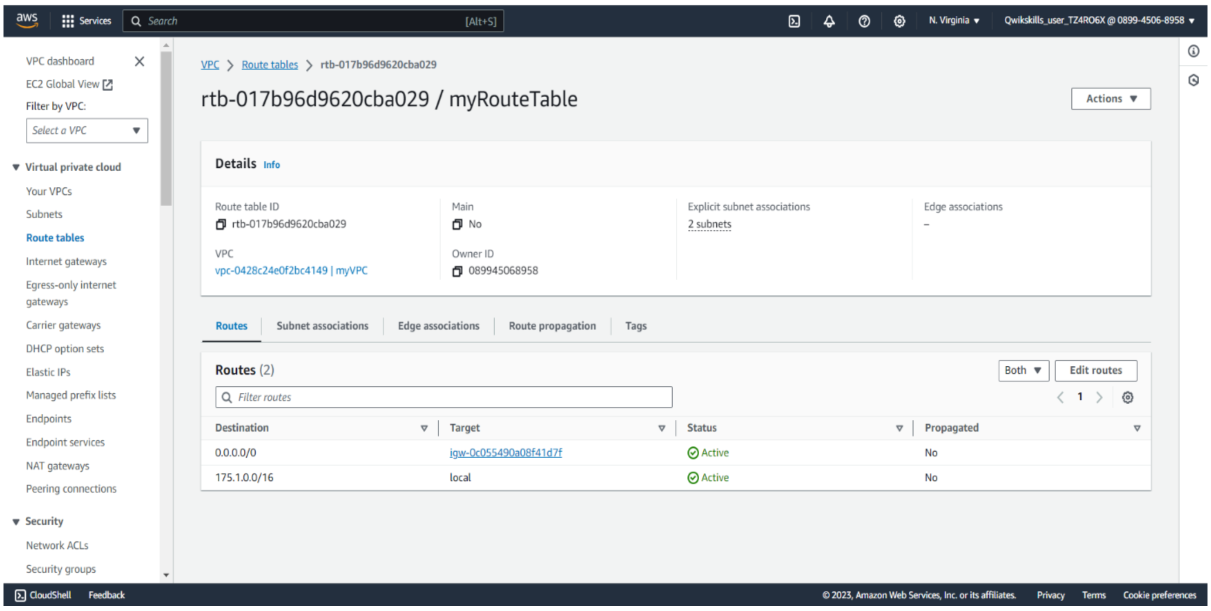


Subnets (3)

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
public-subnet-1	subnet-08e855e3a922cdfbb	Available	vpc-0428c24e0f2bc4149 | myVPC	175.1.0.0/20	–
public-subnet-2	subnet-0fc9e6c7db586e7fe	Available	vpc-0428c24e0f2bc4149 | myVPC	175.1.16.0/20	–
–	subnet-0b9f499051d121dbf	Available	vpc-074d30ae705474050	172.31.32.0/20	–

Select a subnet

Route tables



VPC > Route tables > rtb-017b96d9620cba029

rtb-017b96d9620cba029 / myRouteTable

Actions

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-017b96d9620cba029	No	2 subnets	–
VPC	Owner ID		
vpc-0428c24e0f2bc4149 | myVPC	089945068958		

Routes | Subnet associations | Edge associations | Route propagation | Tags

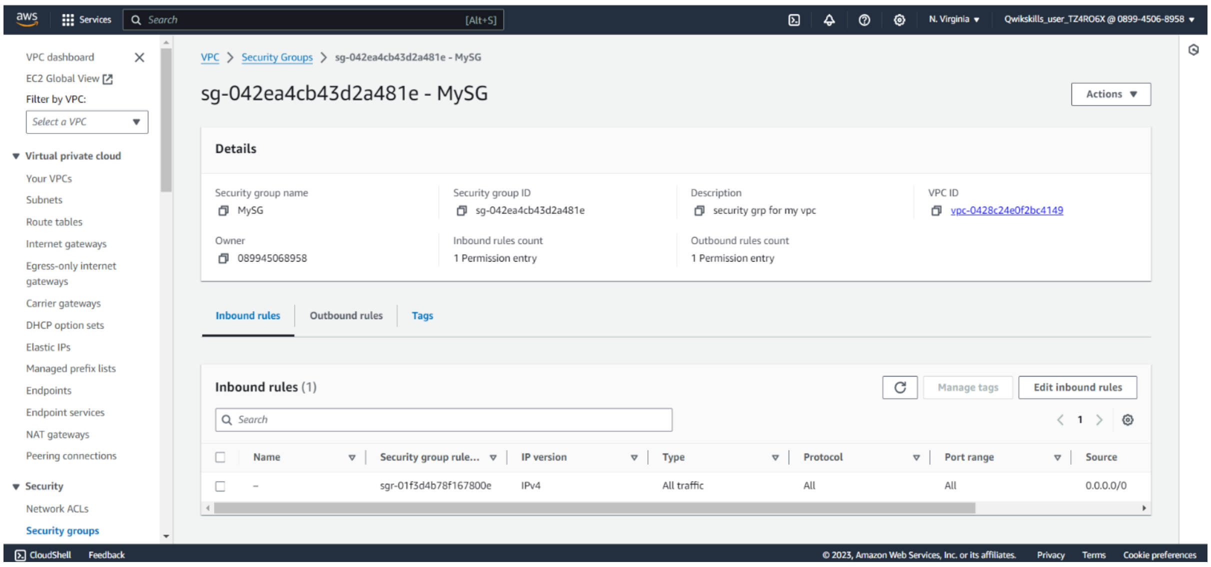
Routes (2)

Both | Edit routes

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0c055490a08f41d7f	Active	No
175.1.0.0/16	local	Active	No

Security group



VPC > Security Groups > sg-042ea4cb43d2a481e - MySG

sg-042ea4cb43d2a481e - MySG

Actions

Details

Security group name	Security group ID	Description	VPC ID
MySG	sg-042ea4cb43d2a481e	security grp for my vpc	vpc-0428c24e0f2bc4149
Owner	Inbound rules count	Outbound rules count	
089945068958	1 Permission entry	1 Permission entry	

Inbound rules | Outbound rules | Tags

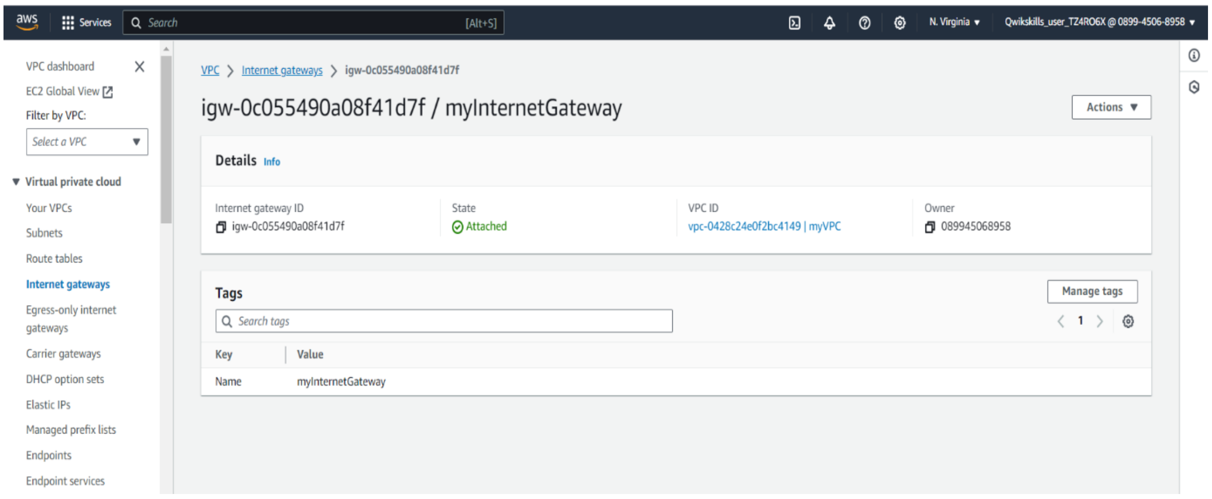
Inbound rules (1)

Manage tags | Edit inbound rules

Search

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-01f3d4b78f167800e	IPv4	All traffic	All	All	0.0.0.0/0

Internet gateway



VPC > Internet gateways > igw-0c055490a08f41d7f

igw-0c055490a08f41d7f / myInternetGateway

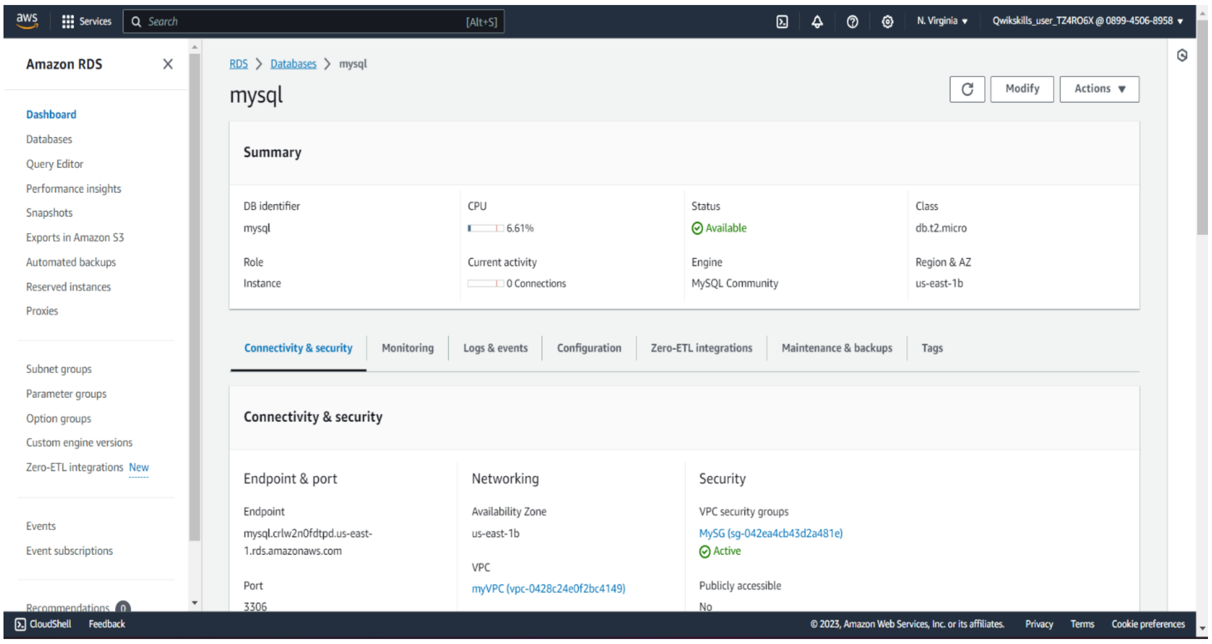
Details

Internet gateway ID	State	VPC ID	Owner
igw-0c055490a08f41d7f	Attached	vpc-0428c24e0f2bc4149 | myVPC	089945068958

Tags

Key	Value
Name	myInternetGateway

RDS instances



RDS > Databases > mysql

mysql

Summary

DB identifier	CPU	Status	Class
mysql	6.61%	Available	db.t2.micro
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0 Connections	MySQL Community	us-east-1b

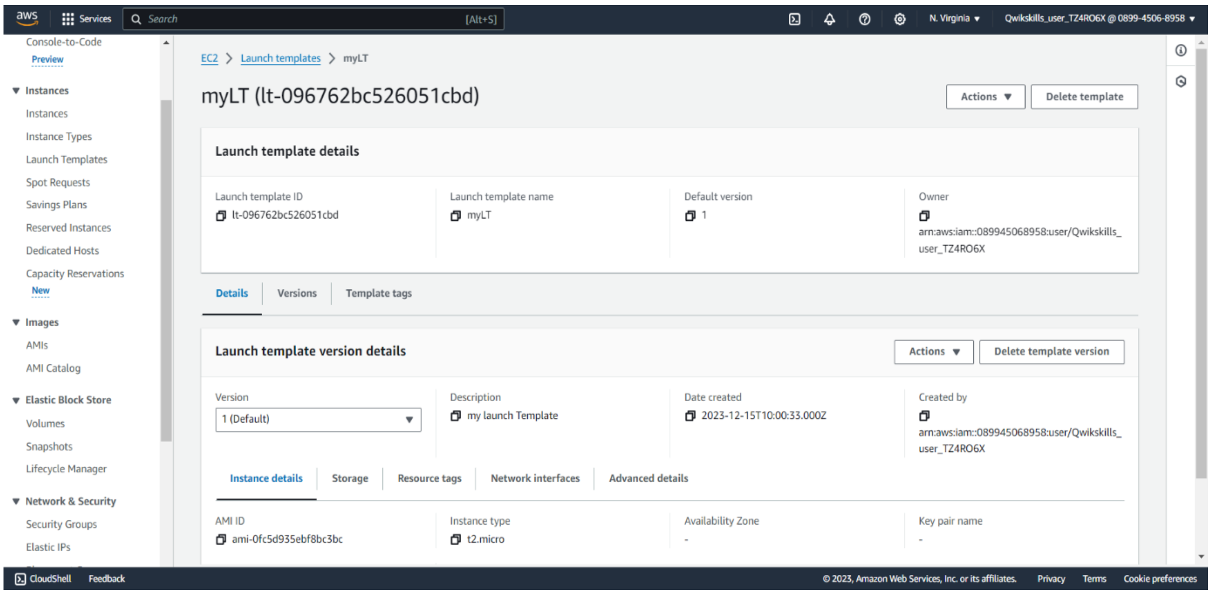
Connectivity & security

Endpoint & port
Endpoint
mysql.crlw2n0fdtpd.us-east-1.rds.amazonaws.com
Port
3306

Networking
Availability Zone
us-east-1b
VPC
myVPC (vpc-0428c24e0f2bc4149)

Security
VPC security groups
MySG (sg-042ea4cb43d2a481e)
Active
Publicly accessible
No

Launch Template



myLT (lt-096762bc526051cbd)

Launch template details

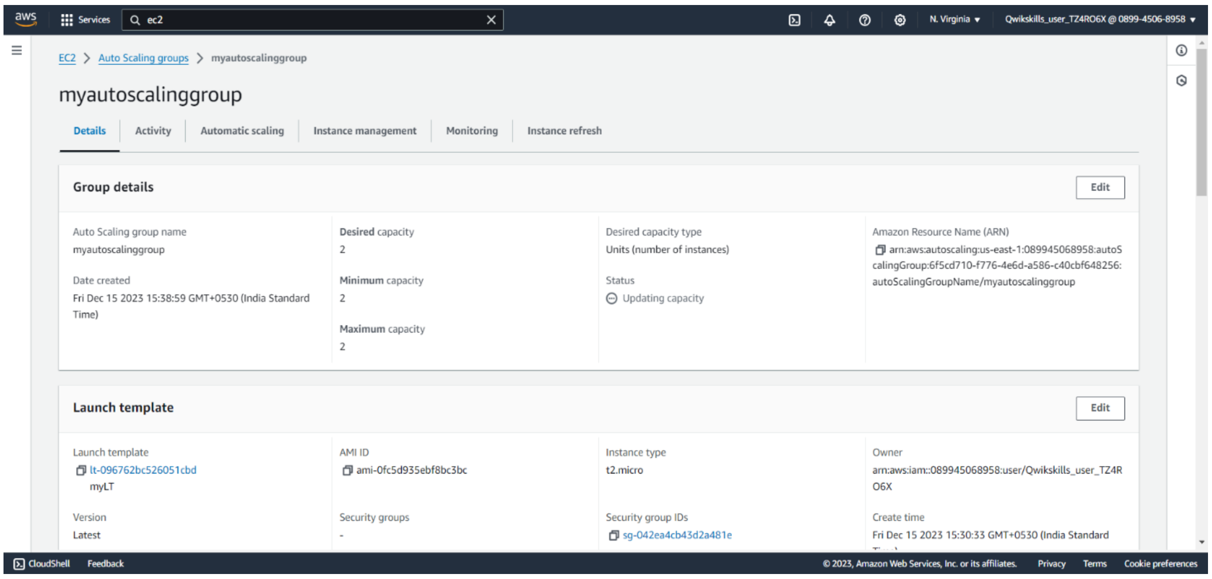
Launch template ID	Launch template name	Default version	Owner
lt-096762bc526051cbd	myLT	1	arn:aws:iam::089945068958:user/Qwikskills_user_TZ4RO6X

Launch template version details

Version	Description	Date created	Created by
1 (Default)	my launch Template	2023-12-15T10:00:33.000Z	arn:aws:iam::089945068958:user/Qwikskills_user_TZ4RO6X

AMI ID	Instance type	Availability Zone	Key pair name
ami-0fc5d935ebf8bc3bc	t2.micro	-	-

Auto Scaling Group



myautoscalinggroup

Group details

Auto Scaling group name
myautoscalinggroup

Date created
Fri Dec 15 2023 15:38:59 GMT+0530 (India Standard Time)

Desired capacity
2

Minimum capacity
2

Maximum capacity
2

Desired capacity type
Units (number of instances)

Status
Updating capacity

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:089945068958:autoScalingGroup:6f5cd710-f776-4e6d-a586-c40cbf648256:autoScalingGroupName/myautoscalinggroup

Launch template

Launch template
lt-096762bc526051cbd
myLT

Version
Latest

AMI ID
ami-0fc5d935ebf8bc3bc

Security groups
-

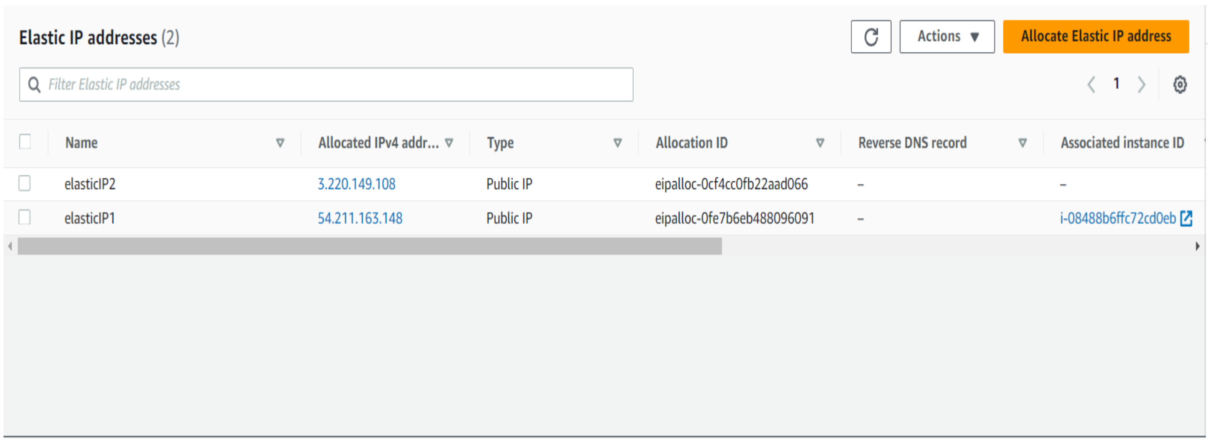
Instance type
t2.micro

Security group IDs
sg-042ea4cb43d2a481e

Owner
arn:aws:iam::089945068958:user/Qwikskills_user_TZ4RO6X

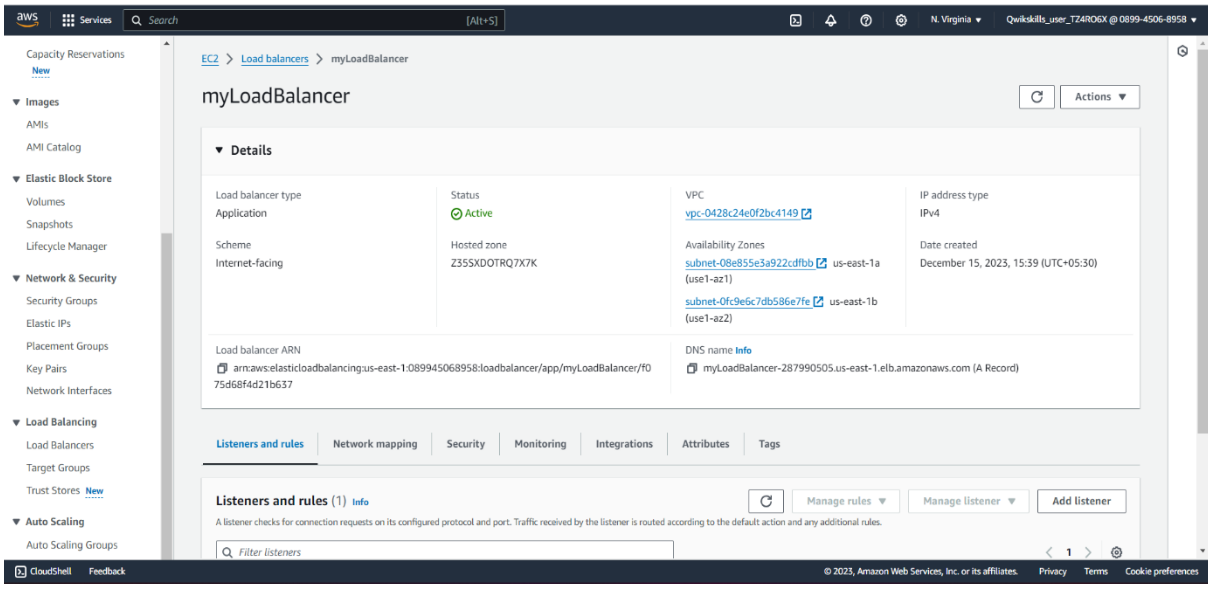
Create time
Fri Dec 15 2023 15:30:33 GMT+0530 (India Standard

Elastic Ips


Elastic IP addresses (2)

Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record	Associated instance ID
elasticIP2	3.220.149.108	Public IP	eipalloc-0cf4cc0fb22aad066	–	–
elasticIP1	54.211.163.148	Public IP	eipalloc-0fe7b6eb488096091	–	i-08488b6ffc72cd0eb

Load Balancer


EC2 > Load balancers > myLoadBalancer

myLoadBalancer

Details

Load balancer type: Application	Status: Active	VPC: vpc-0428c24e0f2bc4149	IP address type: IPv4
Scheme: Internet-facing	Hosted zone: Z35SXDOTRQ7X7K	Availability Zones: subnet-08e855e3a922cdfbb us-east-1a (use1-az1); subnet-0fc9e6c7db586e7fe us-east-1b (use1-az2)	Date created: December 15, 2023, 15:39 (UTC+05:30)

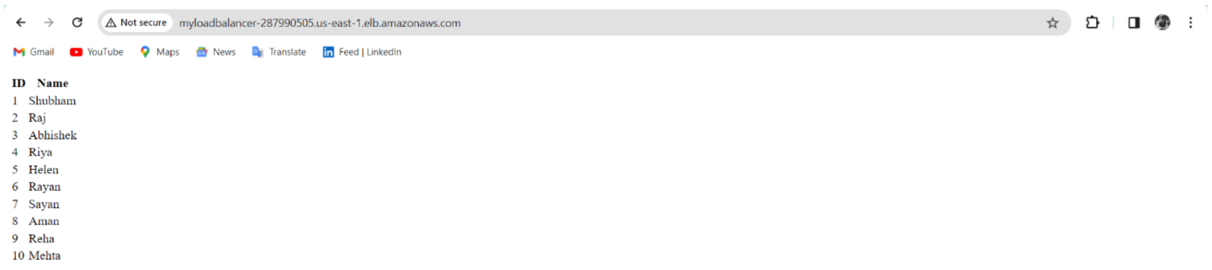
Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:089945068958:loadbalancer/app/myLoadBalancer/f075d68f4d21b637

DNS name: myLoadBalancer-287990505.us-east-1.elb.amazonaws.com (A Record)

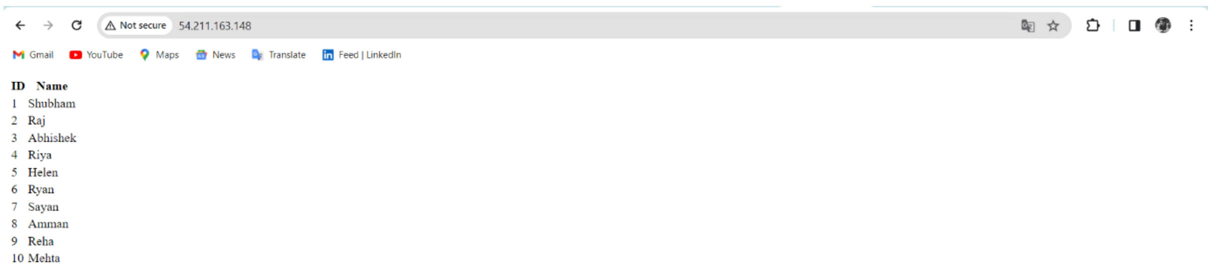
Listeners and rules (1)

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

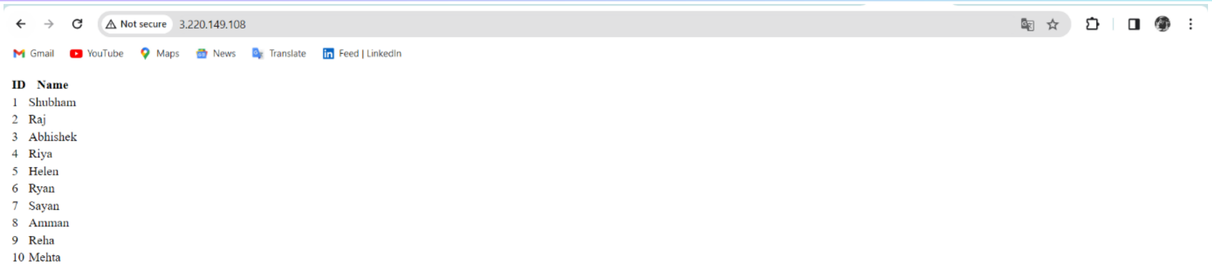
DNS result, and result from both EC2 Instances with their public IP



```
ID  Name
1   Shubham
2   Raj
3   Abhishek
4   Riya
5   Helen
6   Rayan
7   Sayan
8   Aman
9   Reha
10  Mehta
```

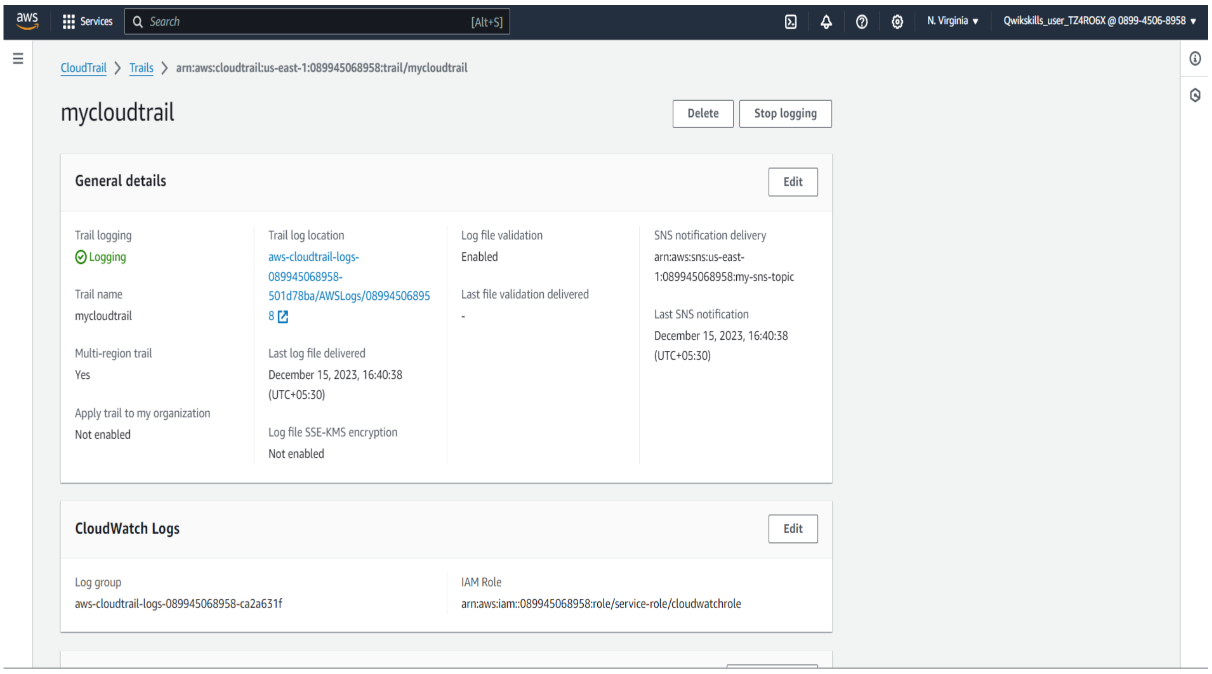


```
ID  Name
1   Shubham
2   Raj
3   Abhishek
4   Riya
5   Helen
6   Ryan
7   Sayan
8   Amman
9   Reha
10  Mehta
```



ID	Name
1	Shubham
2	Raj
3	Abhishek
4	Riya
5	Helen
6	Ryan
7	Sayan
8	Amman
9	Reha
10	Mehta

cloudTrail



arn:aws:cloudtrail:us-east-1:089945068958:trail/mycloudtrail

mycloudtrail

General details

Trail logging: Logging

Trail name: mycloudtrail

Multi-region trail: Yes

Apply trail to my organization: Not enabled

Trail log location: aws-cloudtrail-logs-089945068958-501d78ba/AWSLogs/089945068958

Last log file delivered: December 15, 2023, 16:40:38 (UTC+05:30)

Log file SSE-KMS encryption: Not enabled

Log file validation: Enabled

Last file validation delivered: -

SNS notification delivery: arn:aws:sns:us-east-1:089945068958:my-sns-topic

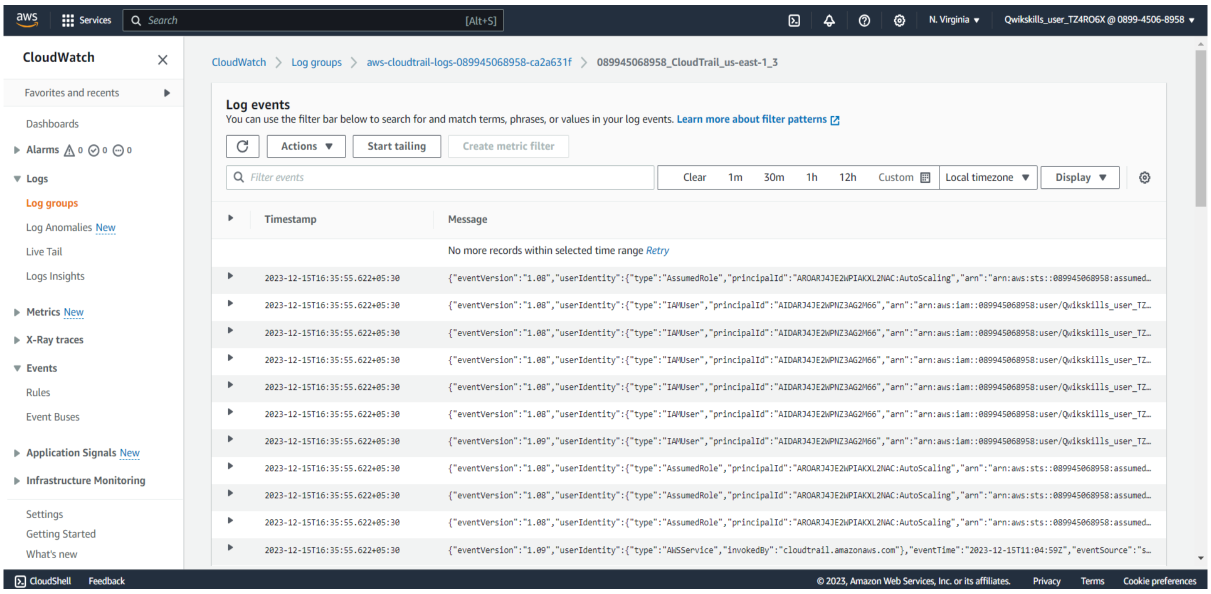
Last SNS notification: December 15, 2023, 16:40:38 (UTC+05:30)

CloudWatch Logs

Log group: aws-cloudtrail-logs-089945068958-ca2a631f

IAM Role: arn:aws:iam::089945068958:role/service-role/cloudwatchrole

CloudWatch



CloudWatch > Log groups > aws-cloudtrail-logs-089945068958-ca2a631f > 089945068958_CloudTrail_us-east-1_3

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

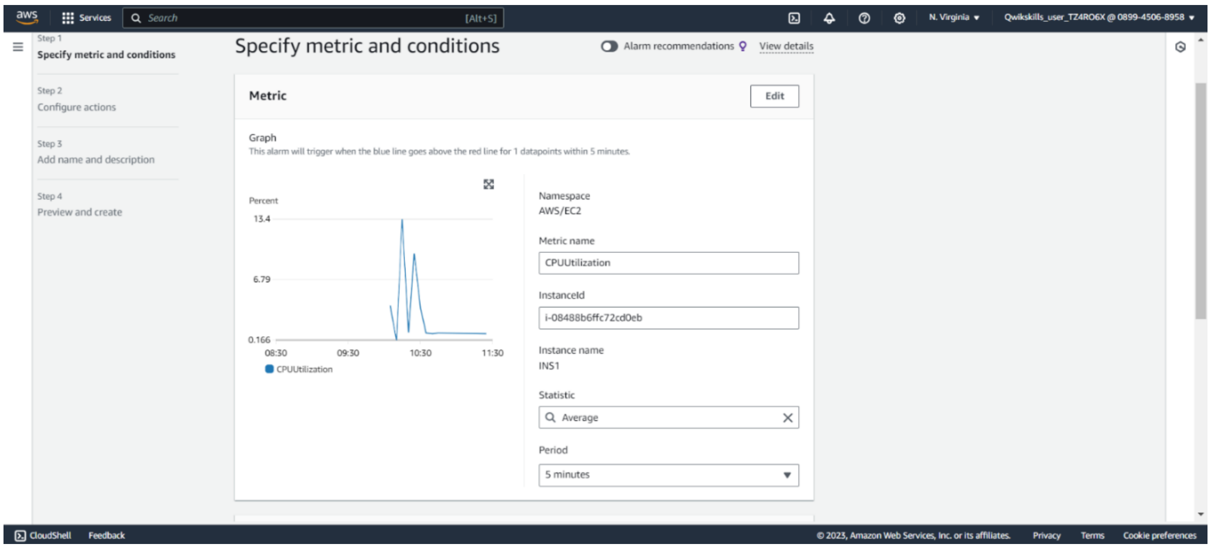
Actions | Start tailing | Create metric filter

Filter events | Clear | 1m | 30m | 1h | 12h | Custom | Local timezone | Display

```
Timestamp                      Message
                               No more records within selected time range Retry
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"AssumedRole","principalId":"AROARJ4JE2WPIAKXL2NAC:AutoScaling","arn":"arn:aws:sts::089945068958:assumed…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"IAMUser","principalId":"AIDARJ4JE2WPNZ3AG2M66","arn":"arn:aws:iam::089945068958:user/Qwikskills_user_TZ…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"IAMUser","principalId":"AIDARJ4JE2WPNZ3AG2M66","arn":"arn:aws:iam::089945068958:user/Qwikskills_user_TZ…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"IAMUser","principalId":"AIDARJ4JE2WPNZ3AG2M66","arn":"arn:aws:iam::089945068958:user/Qwikskills_user_TZ…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"IAMUser","principalId":"AIDARJ4JE2WPNZ3AG2M66","arn":"arn:aws:iam::089945068958:user/Qwikskills_user_TZ…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"IAMUser","principalId":"AIDARJ4JE2WPNZ3AG2M66","arn":"arn:aws:iam::089945068958:user/Qwikskills_user_TZ…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.09","userIdentity":{"type":"IAMUser","principalId":"AIDARJ4JE2WPNZ3AG2M66","arn":"arn:aws:iam::089945068958:user/Qwikskills_user_TZ…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"AssumedRole","principalId":"AROARJ4JE2WPIAKXL2NAC:AutoScaling","arn":"arn:aws:sts::089945068958:assumed…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"AssumedRole","principalId":"AROARJ4JE2WPIAKXL2NAC:AutoScaling","arn":"arn:aws:sts::089945068958:assumed…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.08","userIdentity":{"type":"AssumedRole","principalId":"AROARJ4JE2WPIAKXL2NAC:AutoScaling","arn":"arn:aws:sts::089945068958:assumed…
2023-12-15T16:35:55.622+05:30  {"eventVersion":"1.09","userIdentity":{"type":"AWSService","invokedBy":"cloudtrail.amazonaws.com"},"eventTime":"2023-12-15T11:04:59Z","eventSource":"s…
```

© 2023, Amazon Web Services, Inc. or its affiliates.

CloudWatch Alert



Step 1
Specify metric and conditions

Step 2
Configure actions

Step 3
Add name and description

Step 4
Preview and create

Specify metric and conditions

Metric

Graph

This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

Namespace
AWS/EC2

Metric name
CPUUtilization

InstanceId
i-08488b6ffc72cd0eb

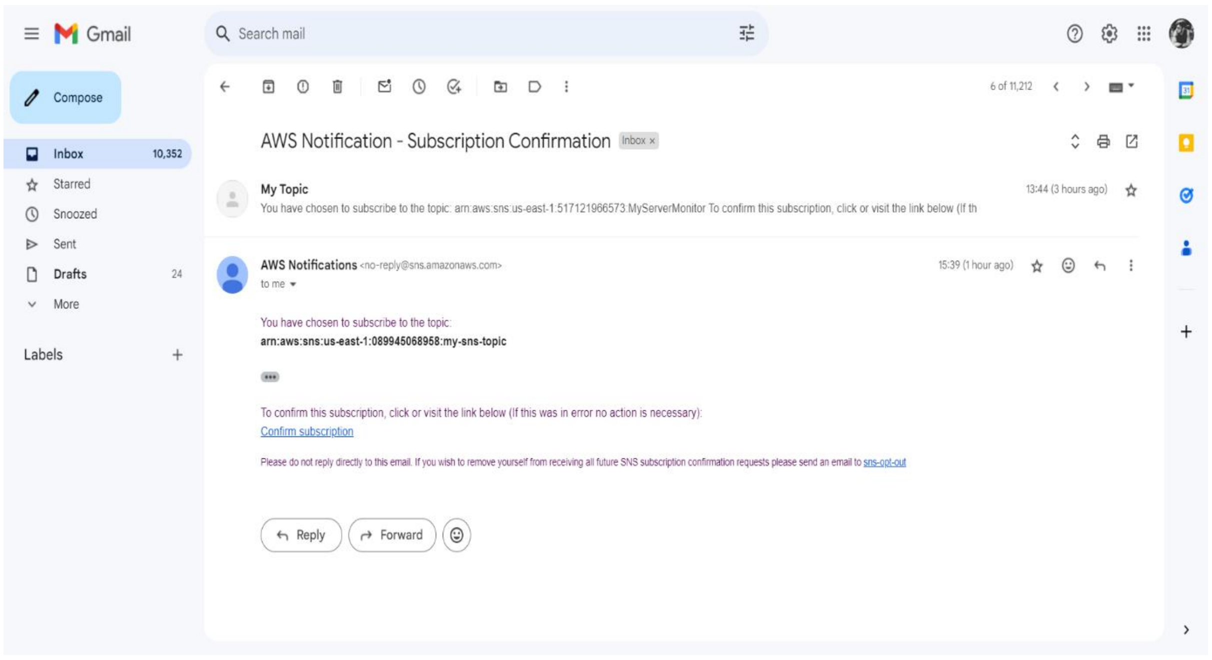
Instance name
INS1

Statistic
Average

Period
5 minutes

© 2023, Amazon Web Services, Inc. or its affiliates.

Email for SNS subscription



AWS Notification - Subscription Confirmation Inbox ×

My Topic 13:44 (3 hours ago)
You have chosen to subscribe to the topic: arn:aws:sns:us-east-1:517121966573:MyServerMonitor To confirm this subscription, click or visit the link below (If th

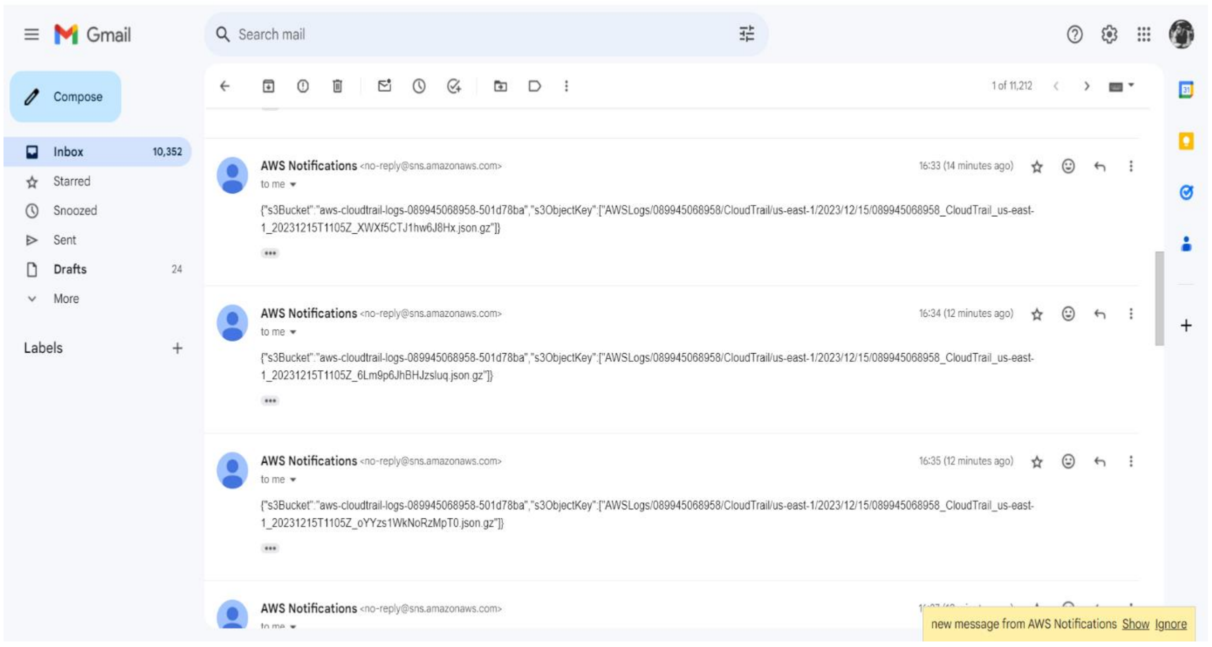
AWS Notifications <no-reply@sns.amazonaws.com> 15:39 (1 hour ago)
to me

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:089945068958:my-sns-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

SNS Notification on Email



AWS Notifications <no-reply@sns.amazonaws.com> 16:33 (14 minutes ago)
to me
```
{"s3Bucket":"aws-cloudtrail-logs-089945068958-501d78ba","s3ObjectKey":["AWSLogs/089945068958/CloudTrail/us-east-1/2023/12/15/089945068958_CloudTrail_us-east-1_20231215T1105Z_XWXf5CTJ1hw6J8Hx.json.gz"]}
```

AWS Notifications <no-reply@sns.amazonaws.com> 16:34 (12 minutes ago)
to me
```
{"s3Bucket":"aws-cloudtrail-logs-089945068958-501d78ba","s3ObjectKey":["AWSLogs/089945068958/CloudTrail/us-east-1/2023/12/15/089945068958_CloudTrail_us-east-1_20231215T1105Z_6Lm9p6JhBHJzsluq.json.gz"]}
```

AWS Notifications <no-reply@sns.amazonaws.com> 16:35 (12 minutes ago)
to me
```
{"s3Bucket":"aws-cloudtrail-logs-089945068958-501d78ba","s3ObjectKey":["AWSLogs/089945068958/CloudTrail/us-east-1/2023/12/15/089945068958_CloudTrail_us-east-1_20231215T1105Z_oYYzs1WkNoRzMpT0.json.gz"]}
```

AWS Notifications <no-reply@sns.amazonaws.com>
to me

new message from AWS Notifications Show Ignore

CONCLUSION

In culmination, the design and implementation of the highly available web application on AWS represent a significant achievement in building a robust and scalable infrastructure. The project's success is underscored by the seamless accessibility achieved through both the load balancer DNS name and public IPs, showcasing the efficacy of load balancing and redundancy mechanisms.

Key components of the project's architecture, including the creation of a VPC (175.1.0.0/16) with two public subnets (175.1.0.0/20 and 175.1.16.0/20) strategically placed in two availability zones (us-east1a and us-east1b), Amazon RDS for efficient database management, and strategically deployed EC2 instances across different availability zones, collectively contribute to a resilient framework capable of handling diverse workloads while ensuring uninterrupted accessibility.

The incorporation of an auto-scaling group and load balancer further enhances the project's scalability and fault tolerance, ensuring a seamless user experience even under varying traffic conditions.

The integration of monitoring services, including CloudWatch, CloudTrail, and SNS, enhances the project's overall resilience. Real-time visibility into metrics and logs, coupled with proactive notification mechanisms, adds a layer of observability critical for maintaining system health and addressing potential issues promptly.

As the project journey concludes, it is valuable to reflect on the lessons learned, challenges overcome, and areas poised for future enhancement

In summary, the project not only fulfills the outlined requirements for a highly available web application on AWS but also exemplifies a commitment to best practices in cloud architecture. Congratulations on reaching this milestone, and may the lessons learned pave the way for continued success in future endeavors.

SALIENT FEATURES

1. Highly Available Infrastructure:

The project establishes a VPC (175.1.0.0/16) with two public subnets (175.1.0.0/20 and 175.1.16.0/20) distributed across distinct availability zones (us-east1a and us-east1b). This design ensures redundancy and fault tolerance, contributing to the high availability of the web application.

2. Elastic Compute Capacity:

Leveraging Amazon EC2 instances and an auto-scaling group, the project dynamically adjusts compute capacity based on demand. This elasticity ensures optimal performance even during peak traffic, enhancing the scalability of the application.

3. Managed Database Service:

Amazon RDS is employed for efficient database management. This managed service automates routine administrative tasks, such as backups and software patching, allowing focus on application development rather than database maintenance.

4. Load Balancing for Even Workload Distribution:

The implementation of a load balancer intelligently distributes incoming traffic across multiple EC2 instances, preventing overloading of individual instances. This load balancing mechanism enhances the overall performance and reliability of the web application.

5. Automated Script for Quick Deployment:

The project includes a Bash script that automates the deployment of the web application, streamlining the setup process for reproducibility and ease of deployment in various environments.

6. Monitoring and Alerting System:

Integration with AWS monitoring services, including CloudWatch, CloudTrail, and SNS, provides real-time insights into system health, performance metrics, and logs. Proactive alerting ensures timely responses to potential issues, contributing to the robustness of the application.

7. Public and Load Balancer Accessibility:

The web application is made accessible through both public IP addresses and a load balancer DNS name, showcasing effective load balancing and redundancy measures. This dual accessibility enhances the availability and reliability of the application.

8. Security Best Practices:

Adherence to AWS security best practices, including the use of security groups and the implementation of secure communication protocols, ensures the integrity and confidentiality of data transmitted within the application.

9. Future-Ready Architecture:

The project's architecture is designed with scalability and adaptability in mind, providing a foundation for future enhancements. The inclusion of a reflective documentation process enables continuous improvement based on lessons learned.

In summary, these salient features collectively define a highly available, scalable, and secure web application on AWS, showcasing a commitment to best practices in cloud architecture

FUTURE SCOPE

As the project progresses, key future development areas emerge:

- Feature Expansion:

  - Prioritizing the integration of advanced features like user authentication, real-time collaboration, and sophisticated data analytics to enhance the web application's utility and responsiveness.

- Global Deployment:

  - Strategically extending the project's reach by deploying instances in additional AWS regions to reduce latency, fortify redundancy, and ensure continuous service availability globally.

- CI/CD Implementation:

  - Implementing Continuous Integration and Deployment (CI/CD) pipelines to automate testing, building, and deployment processes. This accelerates release cycles, improves code quality, and seamlessly integrates updates.

- Microservices Exploration:

  - Exploring the adoption of a microservices architecture for scalable and maintainable development. Breaking down the monolithic structure into independent services allows for flexible deployment and agile development practices.

- Advanced Security Measures:

  - Advancing the project's security posture through the integration of AWS WAF, robust encryption practices, and regular security audits. These measures ensure ongoing integrity and confidentiality of user data.

In summary, the future project scope revolves around enriching features, extending global accessibility, optimizing development processes, exploring architectural improvements, and fortifying security measures. This collective effort aims to establish a resilient, feature-rich, and secure web application.