

A CAPSTONE PROJECT ON  
  
HIGHLY AVAILABLE WEB APPLICATION  
  
UNDERTAKEN AT  
QWIKSKILLS  
By  
Ms. Chandana Yandapalli

SUBMITTED TO



**QWIKSKILLS**

As Partial Fulfilment of the Requirements  
for the Program  
**AWS ESSENTIALS**  
**JUNE - 2023**

GUIDED By  
**Mr. SURAJ VERMA**

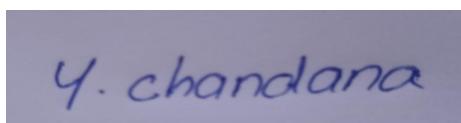
Reference No.

Date:02/12/2023

## DECLARATION

I Chandana Yandapalli, a student of the AWS Essentials Program, declared that the project report submitted by me is a bonafide work for the partial fulfilment of the requirement of the AWS Essentials project work. I have incorporated all the suggestions provided by my guide from time to time.

I further declare that to the best of my knowledge this dissertation contains my original work and does not contain any part of any work which has been submitted for any work entitlement either in this organisation or in any other university/Deemed university/institute organisation etc. without proper citation and I shall be fully responsible for any plagiarism found at any stage.



A handwritten signature in blue ink, appearing to read "Y. chandana".

Your Signature

Name: Chandana Yandapalli

Reference No.

Date:02/12/2023

## Capstone Project Approval Certificate

This is to certify that MS. Chandana Yandapalli, a student of “AWS Essentials” at QwikSkills, has successfully completed the project work entitled “ Highly Available Web Application ” as per guidance. I have thoroughly assessed the progress of the work and reviewed the end result. The student has incorporated all the suggestions provided by instructors in this dissertation. This dissertation is a bona fide piece of work of the standard of capstone project work carried out by the student under instructor’s supervision. Internal examination has been successfully completed.

*Inventor*

Name: Mr. Suraj Verma  
(AWS Cloud Engineer)

S.NO.	INDEX	PAGE NO.
1	<b>Introduction</b>	5-6
2	<b>Background</b>	7-8
3	<b>Requirements</b>	9-10
4	<b>Overview</b>	11
5	<b>Descriptions</b>	12-13
6	<b>Data Flow Diagrams (DFD)</b>	14
7	<b>Flowchart (Modules &amp; Sub-Modules)</b>	15
8	<b>Data Description (E-R diagrams)</b>	16
9	<b>Sample Code</b>	17-19
10	<b>Screenshots</b>	20-64
11	<b>Conclusion</b>	65-66
12	<b>Salient Features</b>	67-68
13	<b>Future Scope</b>	69
14	<b>Bibliography/ References</b>	70

# INTRODUCTION

To build and manage a highly available web application in AWS, it's important to balance cost-effectiveness with peak performance, use modern cloud architecture best practices, and take advantage of AWS services to simplify management tasks. By deploying across multiple availability zones, using auto-scaling groups, and using tools such as AWS Cloud Formation, Elastic Beanstalk, and Lambda, you can streamline deployment, monitoring, and troubleshooting to maintain a healthy application.

When it comes to running a successful business, achieving high availability, minimising downtime, scalability, fault tolerance, automated management, security and compliance, monitoring, and analytics are all crucial objectives to meet.

High availability means that your system is up and running at all times, which is essential for your business to operate smoothly. Minimising downtime is also important as it ensures that any issues that arise are resolved quickly, and your system is back up and running as soon as possible.

Scalability is important as your business grows, and your system needs to be able to handle increased traffic and usage. Fault tolerance is crucial as it ensures that your system is resilient to any faults or errors that may occur.

Automated management is essential as it allows your system to run efficiently and effectively without manual intervention. Security and compliance are also critical objectives to meet as they ensure that your system is secure and meets regulatory requirements.

Finally, monitoring and analytics are essential as they allow you to monitor your system's performance, identify any issues that may arise and make data-driven decisions to improve your system's performance and efficiency. By meeting all these objectives, you can ensure that your business runs smoothly, efficiently, and effectively.

## **Objectives:-**

Achieve Highly Availability, Minimise downtime, Scalability

Fault Tolerance, Automated management, Security and

Compliance, Monitoring and analytics.

# BACKGROUND

Highly Available Web Application in Web Services using AWS (Amazon Web Services) involves the strategic utilisation of various services to ensure reliability, scalability, and fault tolerance. In this context, let's explore how components like Amazon VPC (Virtual Private Cloud), EC2 (Elastic Compute Cloud), RDS (Relational Database Service), CloudWatch, CloudTrail, SNS (Simple Notification Service), and Load Balancer work together to create a robust and highly available architecture.

Amazon VPC (Virtual Private Cloud):

- VPC allows you to create a logically isolated section of the AWS Cloud where you can launch AWS resources.
- It provides control over the virtual networking environment, including IP address ranges, subnets, and route tables.

EC2 (Elastic Compute Cloud):

- EC2 instances are virtual servers in the cloud that can be configured to run applications.
- For high availability, multiple EC2 instances are distributed across different availability zones (AZs) within a region.

RDS (Relational Database Service):

- RDS simplifies database management tasks and enhances the availability and scalability of the application's database.
- Multi-AZ deployments are used for high availability, where a standby instance is automatically created in a different AZ.

CloudWatch:

- CloudWatch is used for monitoring and collecting logs, metrics, and events.
- It helps in setting up alarms and responding to changes in the environment to maintain application availability.

CloudTrail:

- CloudTrail records API calls and creates log files for auditing purposes.
- It helps in tracking changes, understanding user activity, and responding to security incidents.

SNS (Simple Notification Service):

- SNS enables the sending of notifications or alerts to a set of recipients.
- It can be used to notify administrators or operations teams about events that may impact application availability.

Load Balancer:

- A Load Balancer distributes incoming application traffic across multiple EC2 instances to ensure no single instance is overwhelmed.
- It enhances fault tolerance by automatically rerouting traffic in case of instance failures.

Key Design Considerations for High Availability:

- Multi-AZ Deployment: Distributing resources across multiple availability zones to ensure redundancy and fault tolerance.
- Auto Scaling: Automatically adjusting the number of EC2 instances based on traffic and resource utilisation.
- Database Replication: Setting up read replicas and multi-AZ configurations for RDS to enhance database availability.
- Continuous Monitoring: Leveraging CloudWatch for real-time monitoring and setting up alarms for proactive issue resolution.
- Logging and Auditing: Using CloudTrail for auditing API calls and ensuring security and compliance.
- Event Notification: Utilising SNS to send alerts for important events, such as system failures or performance issues.

# REQUIREMENTS

To launch a highly available web application in Amazon Web Services (AWS), you need to design your infrastructure with fault tolerance and redundancy in mind. AWS offers a variety of services and best practices to achieve high availability.

Here's a general guideline for the key components and steps:

- An authorised AWS account with adequate permissions to configure and provide a robust Highly Available Web Application.
- Multiple Availability Zones - Deploy your application across multiple AWS Availability Zones within a region. This ensures that if one AZ experiences issues, your application can continue running in another AZ.
- AWS VPC (Virtual Private Cloud) to configure a new VPC, public subnets, an Internet Gateway and Route Table.
- AWS EC2 (Elastic Cloud Compute) to create a Launch Template, an Auto Scaling Group and the Elastic Load Balancer.
- Elastic Load Balancer (ELB): Use an Elastic Load Balancer to distribute traffic across multiple instances. This can be the Classic Load Balancer, Application Load Balancer, or Network Load Balancer, depending on your needs.
- Auto Scaling - Set up Auto Scaling groups to automatically adjust the number of instances in response to traffic demands. This helps ensure that your application can handle increased loads without manual intervention.
- Amazon RDS Multi-AZ - If you use a relational database, use Amazon RDS with Multi-AZ deployment for database high availability. This replicates your database to a standby instance in another AZ for failover.
- Security - Implement security best practices, such as using AWS Identity and Access Management (IAM), Network Security Groups (NSGs), and encryption for data at rest and in transit.

- Content Delivery - Use Amazon CloudFront for content delivery. CloudFront is a content delivery network (CDN) that caches your content at edge locations, reducing latency and increasing availability.
- Amazon S3 for Static Assets - Store static assets (e.g., images, CSS, JavaScript) in Amazon S3 for durability and scalability. You can also use S3 to host static websites.
- High Availability Databases - consider services like Amazon RDB, which provides high availability and scalability out of the box.
- Monitoring and Alerts - Set up AWS Cloud Watch for monitoring your resources and creating alarms for automated responses. Cloud Watch can also be integrated with AWS Lambda for auto scaling or other automated actions.

# OVERVIEW

This project outlines the use of various AWS services to construct a robust, scalable, and highly available web application infrastructure. Beginning with the creation of an Amazon VPC, public subnets are established across different availability zones for redundancy. Internet traffic is facilitated through an Internet Gateway, and an Application Load Balancer is set up to distribute incoming traffic among multiple EC2 instances. An Auto Scaling Group dynamically adjusts the number of EC2 instances based on traffic demand.

To ensure high availability and scalability of the database, Amazon RDS is configured in a multi-AZ setup, automatically replicating data to a standby instance. Static content, stored in Amazon S3, is served globally through Amazon CloudFront, optimising availability and reducing latency for users worldwide. AWS CloudWatch is employed for real-time performance monitoring and cost analysis, while AWS CloudTrail maintains logs of every event and API call.

# DESCRIPTION

## A description of each component follows:

**EC2 instances:** The first module comprises launching and configuring EC2 Instances to host the web application. EC2 instances are cloud-based virtual Servers that supply computing resources. We may distribute the workload and assure availability even if a single instance fails by launching numerous instances. Selecting the right Amazon Machine Image (AMI), instance type, storage options, security settings, networking options such as Virtual Private Cloud (VPC) and subnets, and additional storage volumes if needed are all part of the configuration.

**RDS setup:** The second segment concentrates on configuring Amazon RDS to offer the web application with a managed database service. Amazon RDS is a dependable and scalable database service. In this phase, we select the database engine (MySQL), as well as the instance parameters such as instance type, storage, and connectivity options. We additionally configured the database's login, password, and authentication. More importantly, backup and maintenance options are configured to assure data durability and availability.

**Auto Scaling Group configuration:** This module entails setting up an Auto Scaling group to automatically scale EC2 instances based on demand. Auto Scaling allows us to automatically alter the number of instances to handle different traffic loads. We can create scaling strategies based on measurements like CPU usage or network traffic. This assures that the application will be able to handle increasing needs without the need to obtain manual intervention. We additionally define the instance's minimum, maximum, and desired capacity, along with the ability to scale cool down periods and notifications.

**Load Balancer setup:** The fourth component deals with configuring an Elastic Load Balancer to allocate incoming traffic across the EC2 instances. Load balancing improves the web application's availability and fault tolerance. The load balancer ensures that no single instance becomes overloaded by distributing traffic equally. We select the appropriate load balancer type, such as Application Load Balancer or Network Load Balancer, then configure the load balancer parameters, including listener ports, protocols, and, if necessary, SSL certificates. We additionally set up the availability zones and subnets in which the load balancer will allocate traffic, as well as the health checks that will be used to monitor the EC2 instances' health.

**S3 Usage:** In our project, it is used for preserving the CloudTrail logs produced by the web application. We use S3 to verify that our log data is consistently saved and readily accessible by additional Amazon Web Services, especially CloudWatch.

**Cloud Trail creation:** It lets us track and log AWS API calls performed within our AWS account. It offers us comprehensive records of actions carried out by users, services, or resources, allowing you to keep track of changes committed to the infrastructure. Cloud Trail has been set up in your application's settings to store logs on Amazon S3, giving us a long-term and scalable storage solution for our log data.

**CloudWatch utilisation:** Metrics, logs, and events from numerous AWS Resources will be collected and tracked. CloudWatch is set up in our project to get logs from the S3 bucket where our CloudTrail logs are maintained. CloudWatch can identify if any EC2 instances in our application are terminated by watching these logs. It lets us to take preemptive actions and assures your web application's high availability

**SNS alerts:** Send alerts to endpoints or clients who have subscribed. When CloudWatch detects a stopped EC2 instance, SNS is utilised in your project to deliver email alerts to a destination. We can rapidly warn the proper parties and take suitable measures to solve any possible concerns by integrating SNS with CloudWatch, hence ensuring the availability and dependability of your online service

# DATA FLOW DIAGRAMS (DFD)

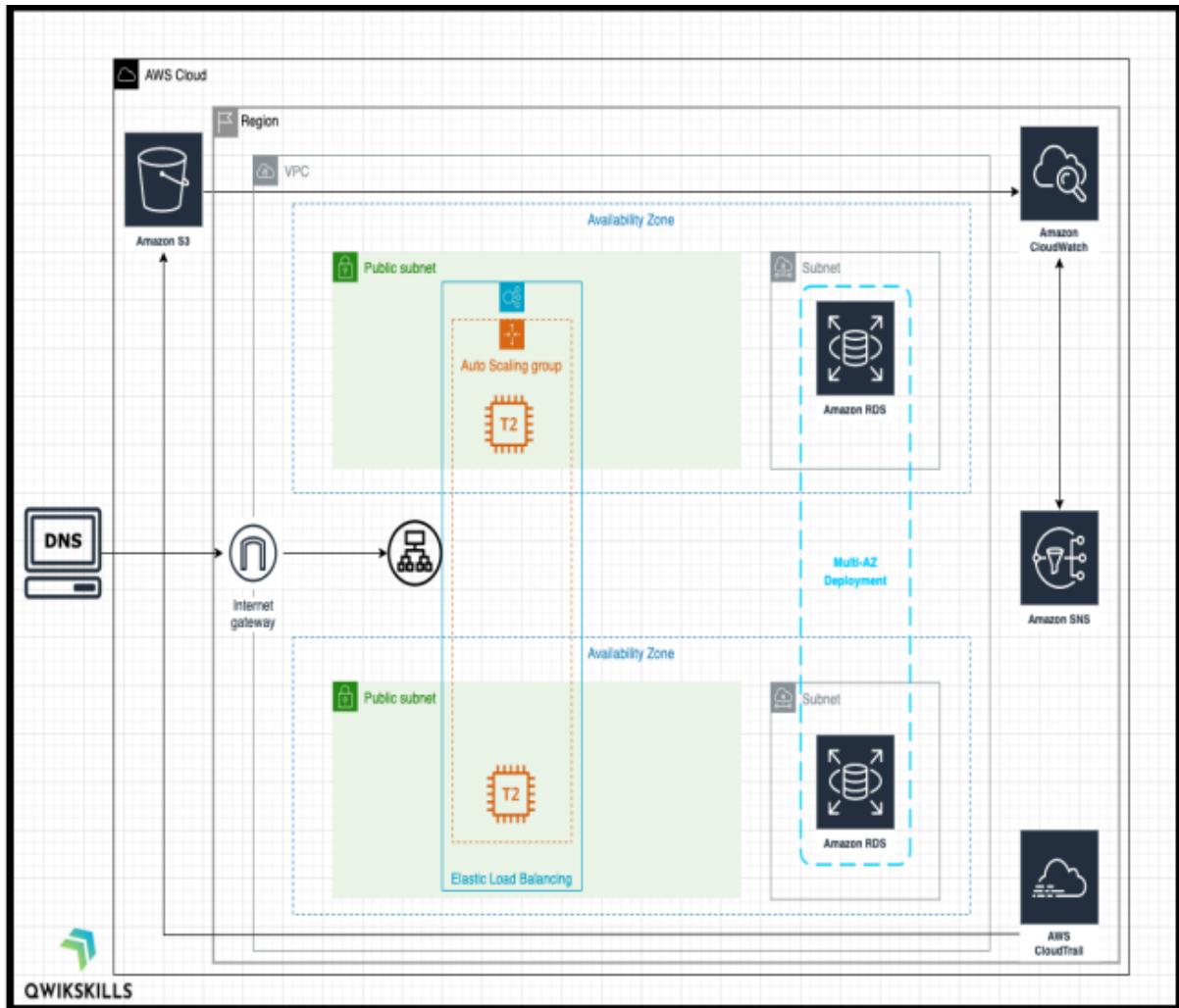


Fig 1 : Following diagram mentioned above represents the Data Flow Diagram.

# FLOWCHART (MODULES&SUB-MODULES)



# DATA DESCRIPTION (E-R DIAGRAMS)

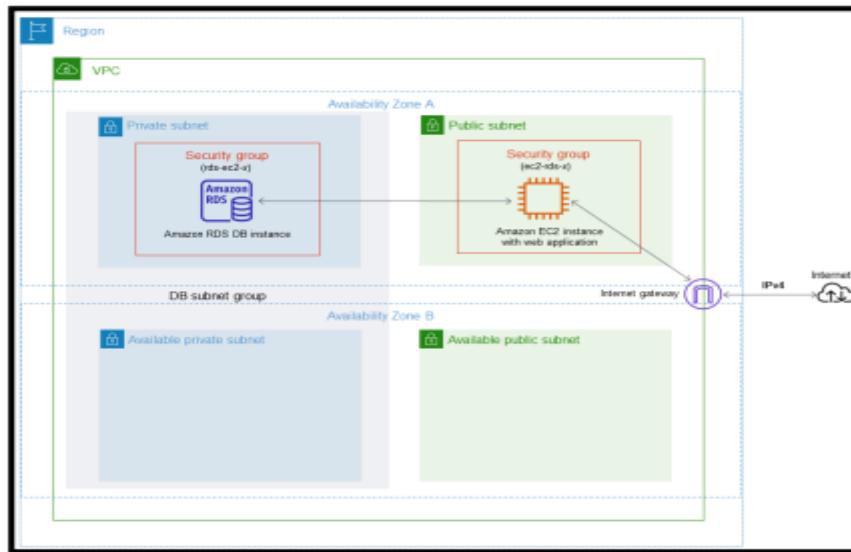


Fig 2 :Following diagram mentioned above show the Data Flow

ID	Name	AGE
001	Rahul	22
002	Budi	21
003	Chandu	20



1	RDS	Database
2	Cloud Trail	Logs
3	EC2	instances
4	VPC	Networking

# SAMPLE CODE

## Script 1: RDS connection from ec2

```
★ sudo su
★ sudo apt update
★ sudo apt-get install mysql-server -y
★ mysql --version
★ mysqlmydatabase.cedvynohrzwi.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
★ Enter password: admin123
```

## Script 2: RDS dummy data insertion

Upon using the commands given in the script 1 and entering the “mysql” database through the RDS endpoint on an EC2 instance.

Use the following.

### SQL commands to insert dummy data:

```
#!/bin/bash
```

```
# MySQL connection details host
my-database.c7k13rii9ip.us-east-1.rds.amazonaws.com
username=admin
password=admin123
database=mydatabase
```

### # SQL query to create table

```
create_table_query="CREATE TABLE details (id INT
PRIMARY KEY, name VARCHAR (50);"
```

```
# SQL query to insert dummy data
insert_data_query="INSERT INTO details (id, name)
VALUES (1, Rahul), (2, 'Budi'), (3, 'Chandana');"

# Connect to MySQL and execute queries
mysql -h $host -u $username -p$password -D $database
-e "$create_table_query" &&

mysql -h $host -u $username -p$password -D $database
-e "$insert_data_query" echo "Table created with dummy
data."
```

### Script 3: Apache server fetches data from RDS

```
#!/bin/bash

# Database configuration
host=<RDS_endpoint>
username="admin"
password="admin123"
database="mydatabase"
table="your_table_name"

# 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
```

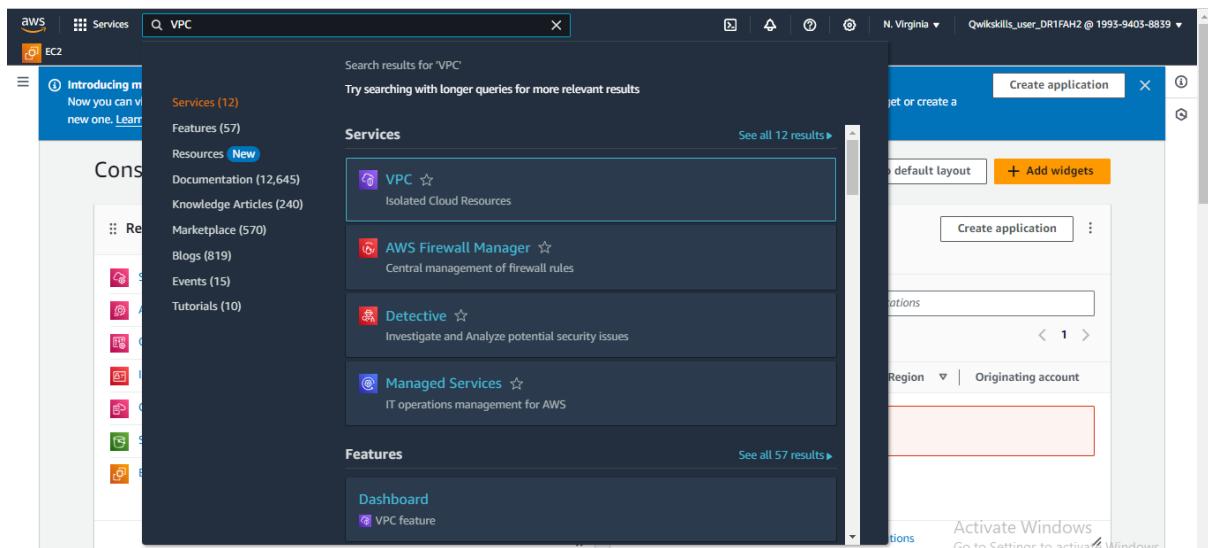
```
# Restart Apache
sudo systemctl restart apache2
```

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

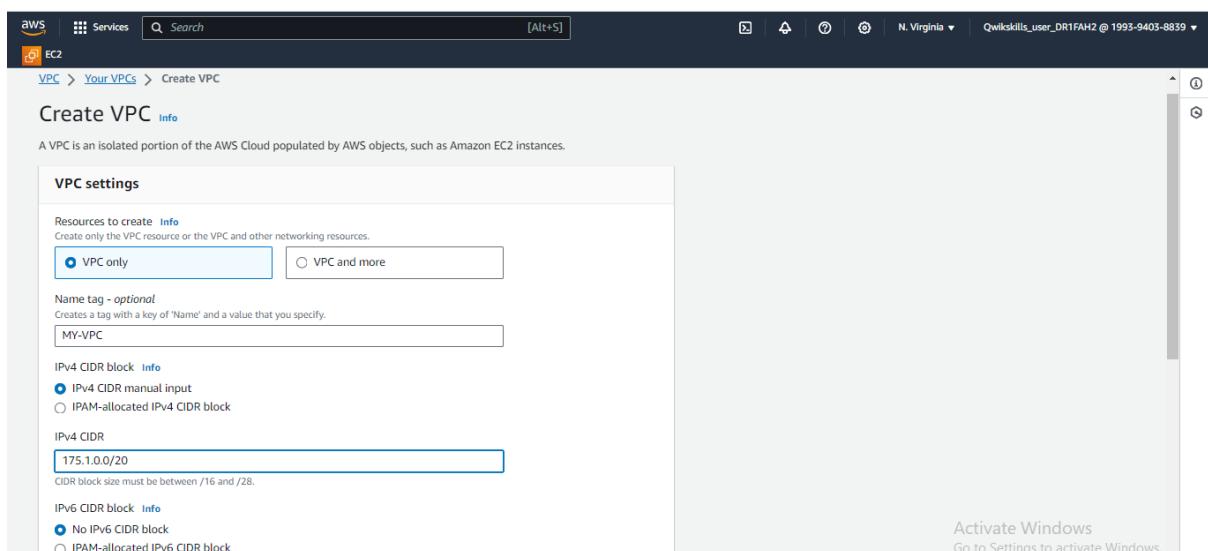
# SCREENSHOTS

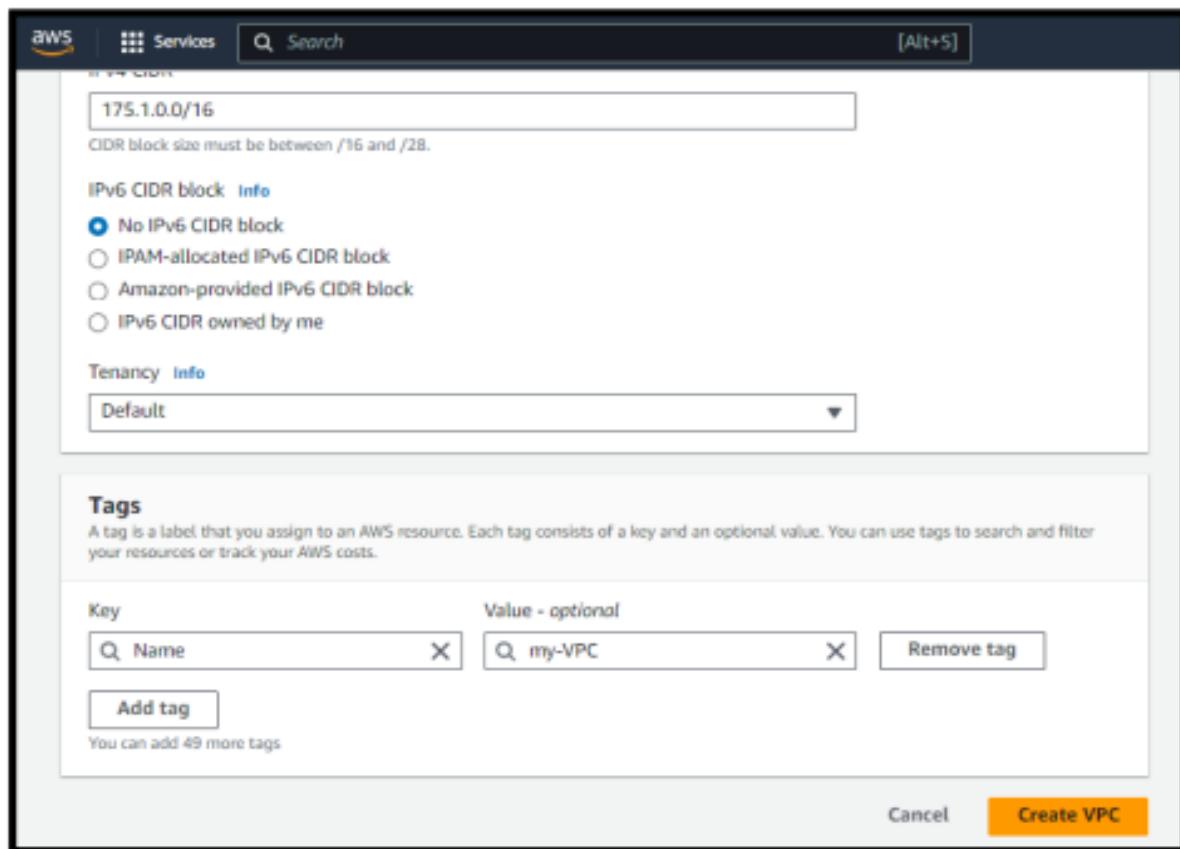
## Vpc :

★ In search bar type “VPC” and hit enter



## ★ Creating VPC





## In VPC settings

- ❖ Select “VPC only”
- ❖ Name tag= “my-VPC”
- ❖ IPv4 CIDR= “175.1.0.0/16” – [Manual input]
- ❖ Default=No ipv6 CIDR block.
- ❖ Click the “Create VPC” button.
- ❖ Review “my-VPC”.

Check that the VPC has been successfully created.

## Subnets :

Go to Subnet in VPC service, \

- ❖ click “Create Subnet” VPC ID= “my-VPC”
- ❖ Subnet name=“public-subnet-1”
- ❖ Availability Zone=“us-east-1a”
- ❖ IPv4 CIDR= “175.1.0.0/20” – [Manual input]
- ❖ Click “Create subnet”

Go to Subnet,

- ❖ click “Create Subnet” VPC ID= “my-VPC”
- ❖ Subnet name=“ public-subnet-2”
- ❖ Availability Zone=“us-east-1b”
- ❖ IPv4 CIDR= “175.1.16.0/20” – [Manual input]
- ❖ Click “Create subnet”

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with options like VPC Dashboard, EC2 Global View, and a dropdown for 'Select a VPC'. The main area has a table titled 'Subnets (6) Info' with columns: Name, Subnet ID, State, VPC, and IPv4 CIDR. The table lists six subnets, all of which are 'Available'. The last subnet listed is 'subnet-0f3f9fdb599c3233' with an IPv4 CIDR of '172.31.48.0/20'. Below the table, there's a section labeled 'Select a subnet'.

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-0bd248013c8d8504d	Available	my-VPC	172.31.64.0/20
-	subnet-0613ffa157cb430cf	Available	my-VPC	172.31.80.0/20
-	subnet-050bddf447b49c5db	Available	my-VPC	172.31.0.0/20
-	subnet-05620a4109f6b42b0	Available	my-VPC	172.31.32.0/20
-	subnet-09c6549fd3183888e	Available	my-VPC	172.31.16.0/20
-	subnet-0f3f9fdb599c3233	Available	my-VPC	172.31.48.0/20

The screenshot shows the AWS EC2 VPC configuration page. At the top, it says "VPC ID" and "Create subnets in this VPC." Below that, the VPC ID is listed as "vpc-0ce805cee40203ff4 (MY-VPC)". Under "Associated VPC CIDRs", it shows "IPv4 CIDRs" with "175.1.0.0/20". In the "Subnet settings" section, "Subnet 1 of 2" is being configured. The "Subnet name" field contains "public-subnet-1". The "Availability Zone" dropdown shows "US East (N. Virginia) / us-east-1a". A watermark for "Activate Windows" is visible on the right.

The screenshot shows the continuation of the VPC configuration for "Subnet 2 of 2". The "Subnet name" is set to "public-subnet-2". The "Availability Zone" is set to "US East (N. Virginia) / us-east-1b". Under "IPv4 VPC CIDR block", the value "175.1.0.0/20" is selected. The "IPv4 subnet CIDR block" is set to "175.1.16.0/20", which covers "4,096 IPs". In the "Tags - optional" section, there is one tag: "Name" with value "public-subnet-2". A watermark for "Activate Windows" is visible on the right.

- ❖ Review both the subnets created as per the details mentioned.
- ✓ public-subnet-1
- ✓ public-subnet-2

# INTERNET GATEWAY: -

- ❖ Go to the Internet Gateway section.
- ❖ Click “Create internet gateway.”
- ❖ Name tag=“MY-INTERNET-GATEWAY”
- ❖ Click “Create internet gateway.”

The screenshot shows the AWS VPC console with the 'Internet gateways' section selected. A single internet gateway is listed:

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-0c5d59dc7cb3ab80a	Attached	vpc-039bb898fa363fc63	199394038839

The sidebar on the left shows other VPC-related options like Subnets, Route tables, and Internet gateways. The bottom status bar indicates it's 22°C Rain at 18:37 on 04-12-2023.

The screenshot shows the 'Create internet gateway' wizard step. It has two main sections:

- Internet gateway settings:** A 'Name tag' field contains 'my-igw'.
- Tags - optional:** A table shows one tag: 'Name' with value 'my-igw'. There is a note that 49 more tags can be added.

At the bottom are 'Cancel' and 'Create internet gateway' buttons. The status bar at the bottom indicates it's 22°C Rain at 18:37 on 04-12-2023.

# Route Tables :

★ Go to Route tables, select the Route Table of my-VPC

The screenshot shows the AWS VPC Route Tables page. On the left, there's a sidebar with options like 'Virtual private cloud', 'Route tables', and 'Internet gateways'. The main area displays a table titled 'Route tables (3)'. The table has columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. The rows show three route tables associated with different VPCs. A modal window titled 'Select a route table' is open at the bottom.

- ★ Click “Route table associated with VPC “my-VPC”
- ★ Click “Edit routes and under “routes” create destination 0.0.0.0/0 and select the internet gateway created as shown in below
  - Click the “Add route” button.
  - Destination=”0.0.0.0/0”
  - Target= Internet gateway “MY-INTERNET-GATEWAY”
  - Click “Save changes” button.

The screenshot shows the 'Edit routes' page for a specific route table. The URL in the address bar is [VPC > Route tables > rtb-0c1c5dcf8dd97228 > Edit routes](#). The page title is 'Edit routes'. It contains a table with columns for Destination, Target, Status, and Propagated. There are two entries: one for '175.1.0.0/16' with 'local' as the target, and another for '0.0.0.0/0' with 'Internet Gateway' as the target. The '0.0.0.0/0' entry has a dropdown menu showing 'igw-0dc8f9c580df57f4f'. At the bottom, there are buttons for 'Add route', 'Cancel', 'Preview', and 'Save changes'.

”

Route table ID: rtb-0c12c5dcf8dd97228

Main

Explicit subnet associations: 2 subnets

VPC: vpc-0c380865996110df1 | MY-VPC1

Owner ID: 199394038839

Subnet associations tab selected.

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
public-subnet-1	subnet-0dad53cfcd61299b9	175.1.0.0/20	-
public-subnet-2	subnet-0e02fcab9f31f5098	175.1.16.0/20	-

- ❖ Subnets are associated with MY-ROUTE-TABLE successfully.

## SECURITY GROUP: -

- ❖ Go to the security group.
- ❖ Click “Create security group.”
- ❖ Security group name=“MY-SG”
- ❖ Description=“my security group”
- ❖ VPC=“my-VPC” (scroll and select )

Basic details

Security group name: MY-SG

Description: Allows SSH access to developers

VPC: my-VPC1

# Configure the Inbound and outbound rules.

- ❖ Inbound rules: (Add rule)

Traffic=" All Traffic"

Source="0.0.0.0/0"

- ❖ Outbound rules: (Add rule)

Traffic=" All Traffic"

Destination="0.0.0.0/0"

The screenshot shows two panels of the AWS VPC Console. The top panel is titled "Inbound rules" and the bottom panel is titled "Outbound rules". Both panels have tabs for Type, Protocol, Port range, Source, and Description. In the Inbound rules panel, the source is set to "0.0.0.0/0" and a warning message is displayed: "⚠ Rules with source of 0.0.0.0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." In the Outbound rules panel, the destination is set to "0.0.0.0/0". A watermark for "Activate Windows" is visible in the bottom right corner of the Outbound rules panel.

Click “Create security group.”

The screenshot shows the AWS VPC Console with a success message: "Security group (sg-04593fa9768ccc5c6 | MY-SG) was created successfully". The main panel displays the details of the new security group "sg-04593fa9768ccc5c6 - MY-SG". It shows the security group name is "MY-SG", the ID is "sg-04593fa9768ccc5c6", the owner is "199394038839", and it has 1 permission entry. Below this, there are tabs for Inbound rules (1), Outbound rules, and Tags. At the bottom, there are buttons for Manage tags, Edit inbound rules, and Activate Windows.

# RDS (Relational Databases) instances: -

- ❖ In search bar type “RDS” and hit enter
- ❖ Go to “Databases” and click the “Create database” button.

The screenshot shows the AWS VPC Console interface. In the top navigation bar, there are tabs for 'AWS Services' and 'VPC'. A search bar contains the query 'RDS'. Below the search bar, the results are displayed under 'Services' and 'Features' categories. Under 'Services', the 'RDS' service is listed as 'Managed Relational Database Service'. Under 'Features', 'Reserved instances' and 'RDS feature' are listed. On the right side of the screen, there is a detailed view of a VPC entry, showing the 'VPC ID' as 'vpc-0c380865996110df1'. The status bar at the bottom indicates 'Activate Windows'.

The screenshot shows the AWS RDS Management Dashboard. The left sidebar has a 'Amazon RDS' section with links like 'Dashboard', 'Databases', 'Query Editor', etc. A modal window is open, prompting the user to try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL. It includes a 'Create database' button and a link to 'Restore Multi-AZ DB Cluster from Snapshot'. The main panel shows 'Resources' and lists various RDS resources: DB Instances (0/40), DB Clusters (0/40), Reserved instances (0/40), Snapshots (0), and DB Clusters (0). On the right, there are sections for 'Parameter groups (2)', 'Option groups (2)', and 'Subnet groups (0/50)'. The status bar at the bottom indicates 'Activate Windows'.

## Select “Standard create” and select Engine type=” MySQL

The screenshot shows the AWS RDS console with the URL [us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance;isHermesCreate=true](https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance;isHermesCreate=true). The page displays various database engine options:

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL** (selected)
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server
- IBM Db2

The MySQL section is expanded, showing its details:

**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.

Activate Windows  
Go to Settings to activate Windows.

CloudShell Feedback Type here to search 22°C Rain 19:21 04-12-2023

### ❖ Templates=“ Dev/Test”

### ❖ Availability and durability=“ Multi-AZ DB instance”

The screenshot shows the AWS RDS console with the URL [us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance;isHermesCreate=true](https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance;isHermesCreate=true). The page displays the following sections:

#### Templates

Choose a sample template to meet your use case.

- Production
- Dev/Test** (selected)
- Free tier

#### Availability and durability

Deployment options [Info](#)

The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB Cluster
- Multi-AZ DB Instance** (selected)
- Single DB Instance

Activate Windows  
Go to Settings to activate Windows.

CloudShell Feedback Type here to search 22°C Rain 19:21 04-12-2023

## In Settings,

- ❖ DB instance identifier=" mydatabase"
- ❖ Master username=" admin"
- ❖ Master password=" admin123"

The screenshot shows the AWS RDS Settings page for creating a new database instance. The 'DB instance identifier' is set to 'mydatabase'. Under 'Credentials Settings', the 'Master username' is 'admin'. A note indicates that managing master user credentials in Secrets Manager is not supported. Other options like 'Auto generate a password' are also shown.

## In Instance configuration,

- ❖ DB instance class=" Burstable classes (includes t classes)"
- ❖ Click "Include previous generation classes."
- ❖ Select "db.t2. Micro"

The screenshot shows the AWS RDS Instance configuration page. The 'DB instance class' is set to 'db.t2.micro', which is described as having 1 vCPU, 1 GiB RAM, and being Not EBS Optimized. The 'Storage type' is set to 'General Purpose SSD (gp3)'. A note states that performance scales independently from storage.

## In Connectivity,

- ❖ VPC="my-VPC"
- ❖ Security Group=MY-SG

The screenshot shows the AWS RDS configuration page for a database instance. Under the 'Compute resource' section, the 'Don't connect to an EC2 compute resource' option is selected. Under 'Virtual private cloud (VPC)', the VPC 'MY-VPC1' is chosen. A note states: 'After a database is created, you can't change its VPC.' Under 'DB subnet group', a new DB subnet group is being created. Under 'Public access', the 'Yes' option is selected. The bottom of the screen shows the AWS navigation bar and a taskbar.

The screenshot shows the AWS RDS configuration page. Under 'Public access', the 'Yes' option is selected, allowing public IP access. Under 'VPC security group (firewall)', the 'Choose existing' option is selected, with 'MY-SG' chosen from the dropdown. Under 'RDS Proxy', the 'Create an RDS Proxy' checkbox is unchecked. Under 'Certificate authority - optional', the 'rds-ca-2019 (default)' option is selected. The bottom of the screen shows the AWS navigation bar and a taskbar.

- ❖ Review Enable deletion protection turned off.
- ❖ Click the “Create database” button.

The screenshot shows the AWS RDS console with a success message: "Successfully created database mydatabase". The message indicates that settings from "mydatabase" can be used to simplify configuration of suggested database add-ons. The main interface displays a list of databases, including "mydatabase", which is currently being configured with enhanced monitoring. The database details show it's an MySQL Community instance in us-east-1b, db.t2.micro.

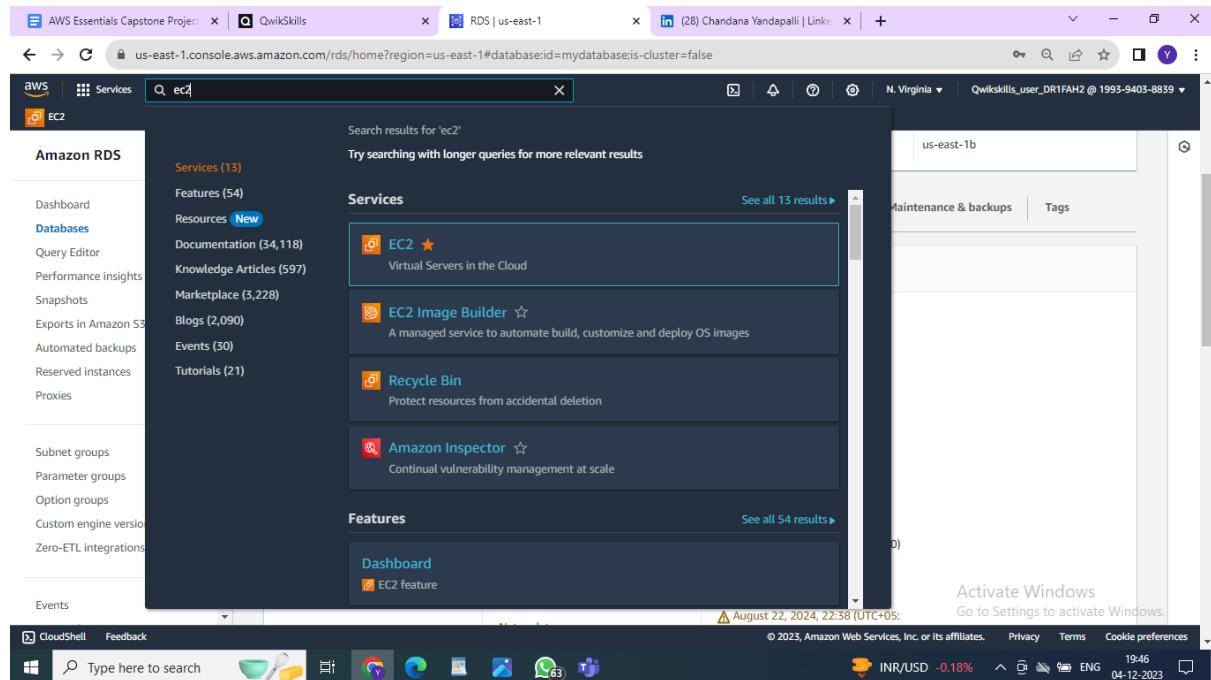
- ❖ Review “mydatabase” and copy its Endpoint.
- ❖ **Endpoint:** mydatabase.cedvynohrzwi.us-east-1.rds.amazonaws.com

The screenshot shows the "Connectivity & security" tab for the "mydatabase" endpoint. Key details include:

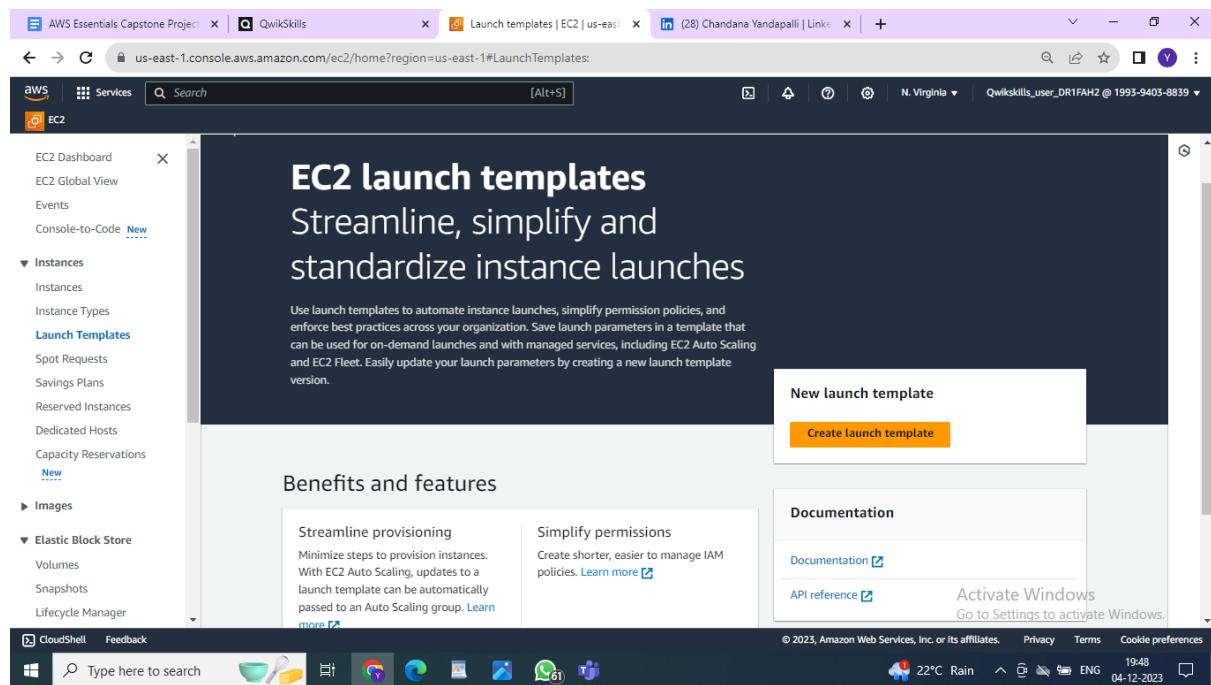
- Endpoint:** mydatabase.cedvynohrzwi.us-east-1.rds.amazonaws.com
- Port:** 3306
- Networking:** Availability Zone: us-east-1b; VPC: MY-VPC1 (vpc-0c380865996110df1); Subnet group: default-vpc-0c380865996110df1; Subnets: subnet-0dad53cfcc61299b9, subnet-0e02fcab9f31f5098
- Security:** VPC security groups: MY-SG (sg-04593fa9768cc5c6); Active status: Active
- Publicly accessible:** No
- Certificate authority:** rds-ca-2019
- DB instance certificate expiration date:** August 22, 2024, 22:38 (UTC+05:30)

# Launch Template: -

❖ In search bar type “EC2” and hit enter



❖ Under EC2 service click Launch templates.



- ❖ Click “Launch Template”
- ❖ Click “Create launch template.”
- ❖ Launch template name=“MY-LT”
- ❖ Template version description=“my Launch Template”
- ❖ AMI=“ Ubuntu Server 22.04 LTS (HVM), SSD Volume Type.
- ❖ Instance type =“t2. micro”
- ❖ Key pair = Don’t include in launch template
- ❖ Security groups=“MY-SG”
- ❖ Click “Create launch template” button

**Create launch template**

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

**Launch template name and description**

Launch template name - required  
MY-LT

Template version description  
A prod webserver for MyApp

Auto Scaling guidance [Info](#)  
Select this if you intend to use this template with EC2 Auto Scaling  
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Template tags  
▶ Source template

**Summary**

Software Image (AMI)  
-

Virtual server type (instance type)  
-

Firewall (security group)  
-

Storage (volumes)  
-

**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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel **Create launch template** Activate Windows Settings to activate Windows.

**Create launch template**

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

**Launch template name and description**

Launch template name - required  
MY-LT

Template version description  
A prod webserver for MyApp

Auto Scaling guidance [Info](#)  
Select this if you intend to use this template with EC2 Auto Scaling  
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Template tags  
▶ Source template

**Summary**

Software Image (AMI)  
-

Virtual server type (instance type)  
-

Firewall (security group)  
-

Storage (volumes)  
-

**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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel **Create launch template** Activate Windows Settings to activate Windows.

AWS Essentials Capstone Project | QwikSkills | Create launch template | EC2 | (28) Chandana Yandapalli | Link | +

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

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

Quick Start

Don't include in launch template | Amazon Linux | macOS | Ubuntu | Windows | Red Hat | Browse more AMIs

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type  
ami-0fc5d935ebf8bc3bc (64-bit (x86)) / ami-016405166ec7fa705 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: obs

Free tier eligible

Description: Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-09-19

Architecture: 64-bit (x86) | AMI ID: ami-0fc5d935ebf8bc3bc | Verified provider

CloudShell Feedback

Type here to search

Activate Windows

Create launch template

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

22°C Rain 19:49 04-12-2023

AWS Essentials Capstone Project | QwikSkills | Create launch template | EC2 | (28) Chandana Yandapalli | Link | +

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:

Description: Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-09-19

Architecture: 64-bit (x86) | AMI ID: ami-0fc5d935ebf8bc3bc | Verified provider

▼ Instance type [Info](#) | [Get advice](#)

Advanced

Instance type: t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.0716 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

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

▼ Summary

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ... [read more](#)  
ami-0fc5d935ebf8bc3bc

Virtual server type (instance type): t2.micro

Firewall (security group):

Storage (volumes): 1 volume(s) - 8 GiB

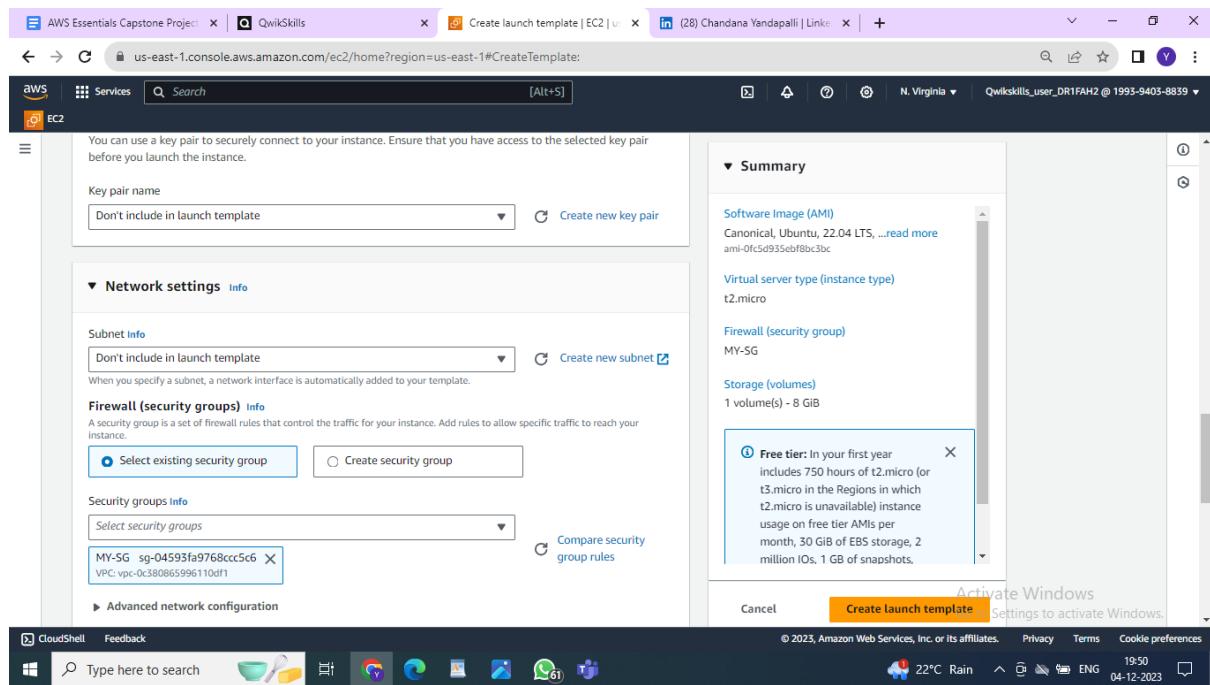
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots.

Cancel Create launch template

Activate Windows

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

22°C Rain 19:50 04-12-2023



## ❖ Review “MY-LT”

## Auto Scaling Group : -

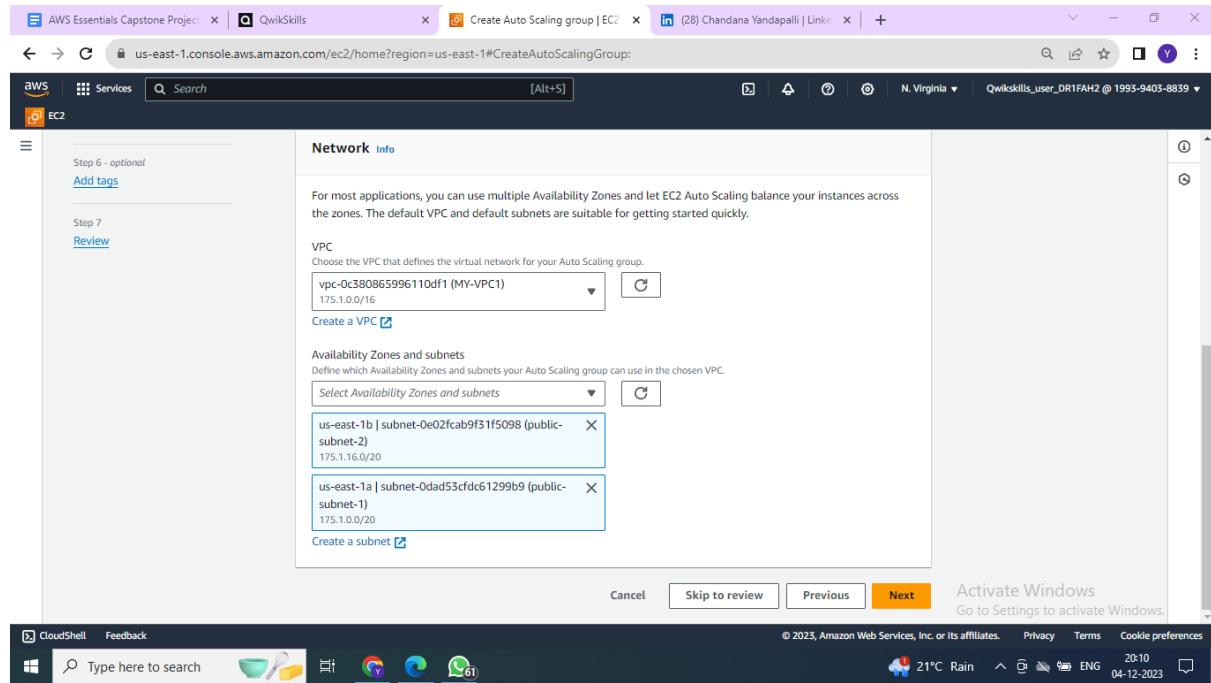
- ❖ Go to the Auto Scaling section in EC2 service.
- ❖ Auto Scaling group name="MY-ASG"
- ❖ Launch template="MY-LT"
- ❖ Click "Next"

The screenshot shows the AWS EC2 Auto Scaling landing page. On the left, there is a navigation sidebar with links for Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling (with Auto Scaling Groups selected). The main content area features a large heading "Amazon EC2 Auto Scaling helps maintain the availability of your applications". Below it, a sub-section titled "How it works" contains a diagram showing a central "Auto Scaling group" box connected to four smaller boxes representing EC2 instances. To the right, there is a "Create Auto Scaling group" button and a "Pricing" section. At the bottom, there is a search bar and a footer with standard AWS links and weather information.

The screenshot shows the "Create Auto Scaling group" configuration step. On the left, a sidebar lists steps from Step 2 to Step 7. Step 2 is "Choose instance launch options", Step 3 is "Configure advanced options", Step 4 is "Configure group size and scaling", Step 5 is "Add notifications", Step 6 is "Add tags", and Step 7 is "Review". The main form is for Step 2, showing fields for "Name" (set to "MY-ASG") and "Launch template". The "Launch template" dropdown is set to "MY-LT". Other fields include "Description" (empty), "Launch template" (set to "MY-LT"), and "Instance type" (set to "t2.micro"). A "Switch to launch configuration" link is also present. At the bottom, there is a "Create" button and a "Cancel" button. The footer includes standard AWS links and weather information.

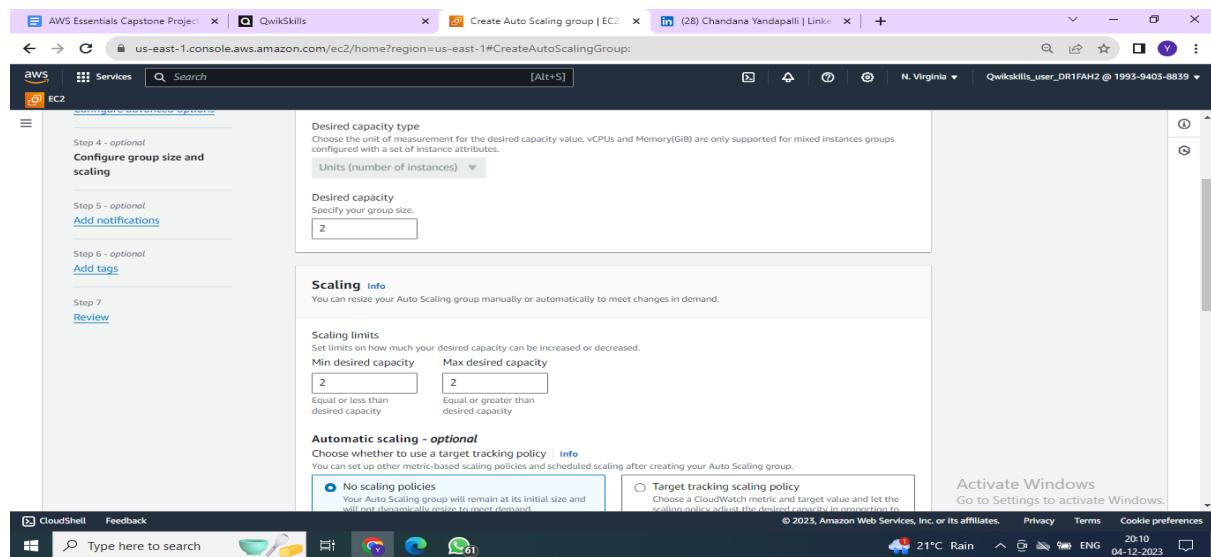
- ❖ VPC=Select “MY-VPC”
- ❖ Availability zone and subnets = Select both the subnets (us-east-1a, us east-1b).

Click “Next” and Select “No load balancer” and click the “Next” button.



### Group size

- ❖ Desired capacity=”2”
- ❖ Minimum capacity=”2”
- ❖ Maximum capacity=”2”
- ❖ Scaling Policies=None
- ❖ Click “Next” button



The screenshot shows the AWS CloudShell interface. At the top, there are several tabs: "AWS Essentials Capstone Project", "QwikSkills", "Auto Scaling groups | EC2 | us-east-1", and "(28) Chandana Yandapalli | Links". The main content area is titled "Auto Scaling groups (1) Info". It displays a table with one row for "MY-ASG". The columns include Name (MY-ASG), Launch template/configuration (MY-LT | Version Default), Instances (0), Status (Updating capacity...), Desired capacity (2), Min (2), Max (2), and Avail. (us-east-1). Below the table, a message says "0 Auto Scaling groups selected" and "Select an Auto Scaling group". At the bottom of the screen, there is a Windows taskbar with icons for CloudShell, Feedback, Start button, search bar, Task View, Edge browser, File Explorer, and a green icon with the number 60. The status bar at the bottom right shows "© 2025, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences", along with weather information (22°C Rain), system status (20:21), and the date (04-12-2023).

❖ Auto scaling group has been created successfully

## Elastic IPs :-

- ❖ Under EC2 service go to Elastic Ips to associate one Elastic IP address with each EC2 Instance
- ❖ Go to running instances,
- ❖ Select instance, click Edit Name and Type “instance01 and instance02 simultaneously.”

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups>. The browser tab bar includes "AWS Essentials Capstone Project", "QwikSkills", "Auto Scaling groups | EC2 | us-east-1", and "(28) Chandana Yandapalli | Links". The main content area displays the "Auto Scaling groups" section with a table. One row is selected for "MY-ASG" with a Launch template/configuration of "MY-LT" and a status of "Updating capacity...". A modal dialog at the bottom left says "0 Auto Scaling groups selected" and "Select an Auto Scaling group". The top navigation bar has tabs for "Services" (selected), "Search", and "Actions". The bottom navigation bar includes "CloudShell", "Feedback", and links for "Privacy", "Terms", and "Cookie preferences". A system tray at the bottom right shows "22°C Rain", "ENG", and the date "04-12-2023".

- ❖ Click on Elastic Ips and “allocate Elastic IP address.”

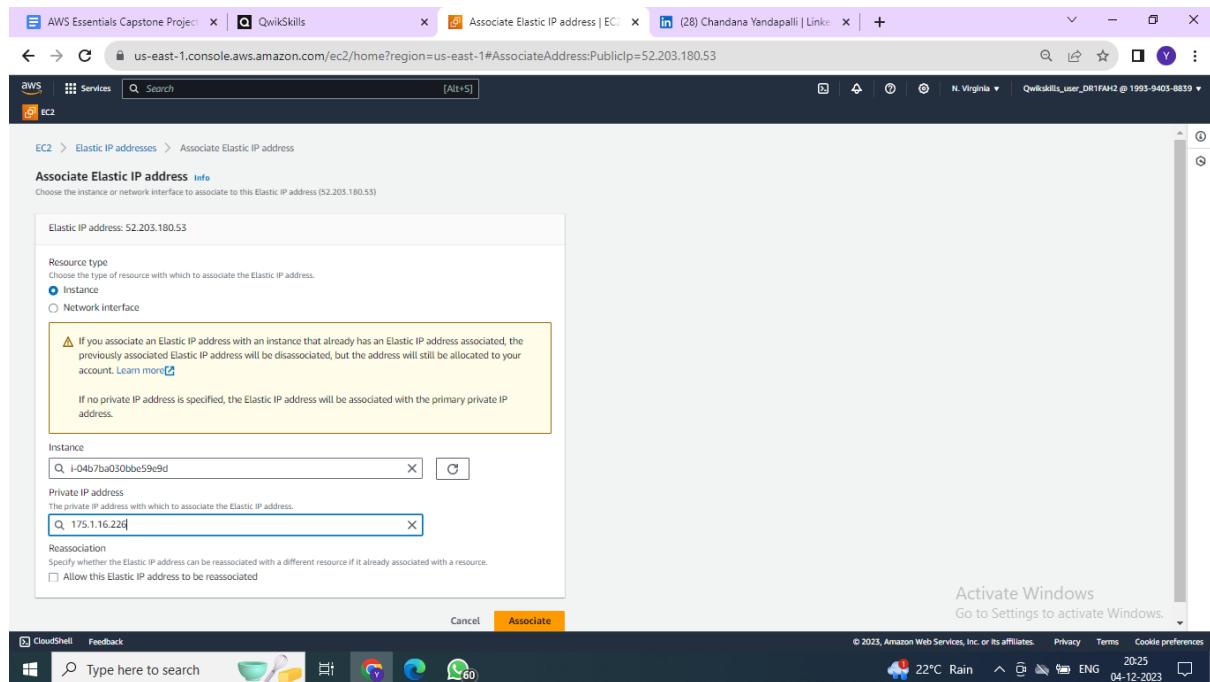
The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Addresses>. The browser tab bar includes "AWS Essentials Capstone Project", "QwikSkills", "Elastic IP addresses | EC2 | us-east-1", and "(28) Chandana Yandapalli | Links". The main content area displays the "Elastic IP addresses" section with a table. A note at the bottom says "View IP address usage and recommendations to release unused IPs with Public IP insights." The left sidebar shows navigation categories like "Reserved Instances", "Dedicated Hosts", "Capacity Reservations", "Images", "Elastic Block Store", "Network & Security" (selected), "Load Balancing", and "Trust Stores". The top navigation bar has tabs for "Services" (selected), "Search", and "Actions". The bottom navigation bar includes "CloudShell", "Feedback", and links for "Privacy", "Terms", and "Cookie preferences". A system tray at the bottom right shows "22°C Rain", "ENG", and the date "04-12-2023".

## ❖ Click on Associate.

The screenshot shows the 'Elastic IP address settings' page in the AWS Management Console. The 'Associate' button is highlighted in orange at the bottom right of the main content area. The page includes sections for 'Network Border Group', 'Public IPv4 address pool' (Amazon's pool), and 'Tags - optional'. A status message at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

## ❖ Associate one Elastic IP address with EC2 Instance01.

The screenshot shows the 'Associate Elastic IP address' page for EC2 Instance01. It displays the instance ID (i-04b7ba030bbe59e9d) and private IP address (175.116.224). A warning message states: 'If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account.' Below this, a note says: 'If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.' The 'Associate' button is visible at the bottom right. A status message at the bottom right says 'Activate Windows Go to Settings to activate Windows.'



- ❖ Review whether it's assigned to instance01 and do same to instance02 as mentioned above
- ❖ Elastic Ips assigned for both the Instances.
  - ✓ 52.203.180.53 (instance01)
  - ✓ 34.226.181.247(instance02)

# Load Balancer:-

- ❖ Go to the Load balancer section.
- ❖ Click “Create load balancer.”

The screenshot shows the AWS EC2 Load Balancers page. On the left, there is a navigation sidebar with various services like Savings Plans, Reserved Instances, and Load Balancing. Under Load Balancing, 'Load Balancers' is selected. The main content area is titled 'Load balancers' and contains a table with one row: 'No load balancers'. Below the table is a button labeled 'Create load balancer'. The top of the page has tabs for 'AWS Services' and 'QwikSkills', and the URL is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers:'.

The screenshot shows the 'Compare and select load balancer' wizard. It displays three options: 'Application Load Balancer', 'Network Load Balancer', and 'Gateway Load Balancer'. Each option has a diagram and a brief description. The 'Application Load Balancer' section says: 'Choose an Application Load Balancer when you need a flexible feature set for your applications with integrated security. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.' The 'Network Load Balancer' section says: 'Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, and certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.' The 'Gateway Load Balancer' section says: 'Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GEN2 features. These enable you to improve security, compliance, and policy controls.' At the bottom of the wizard, there is a link 'Classic Load Balancer - previous generation'.

- ❖ Select “Application Load Balancer”
- ❖ Load balancer name=“ Application-Load-Balancer”
- ❖ Scheme=“ Internet-Facing”
- ❖ VPC=“my-VPC”

### Mappings:

- ❖ Us-east-1a → public-subnet-1
- ❖ Us-east-1b → public-subnet-2
- ❖ Security groups=“MY-SG”

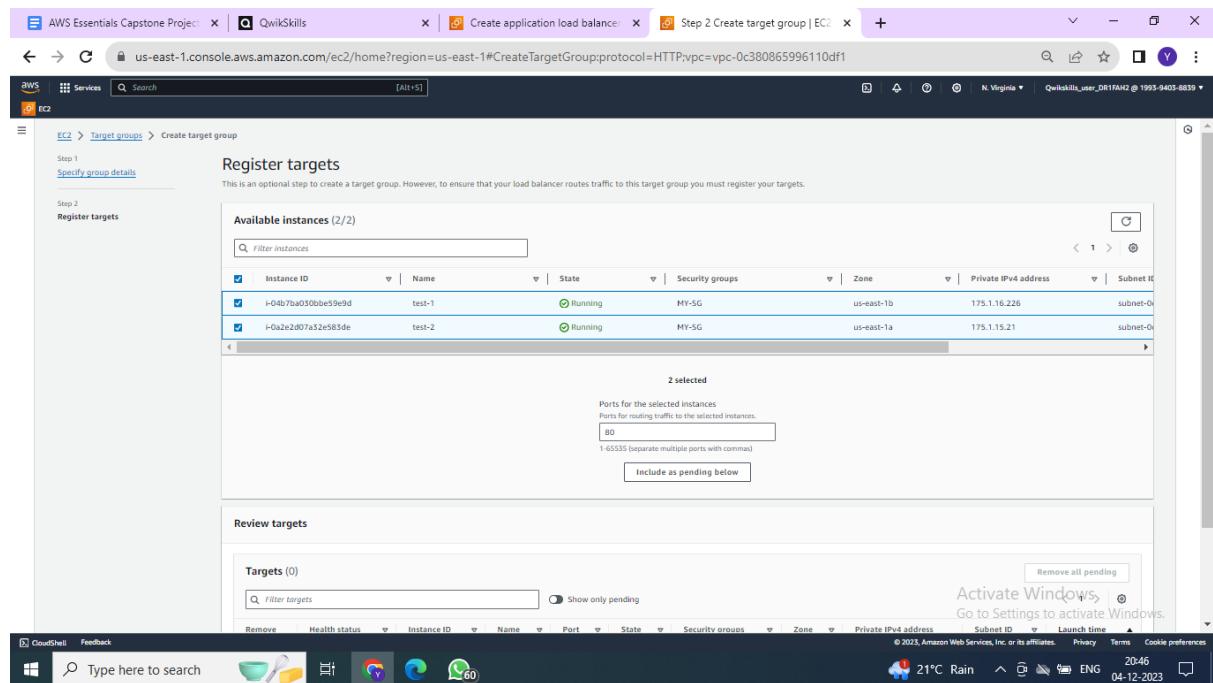
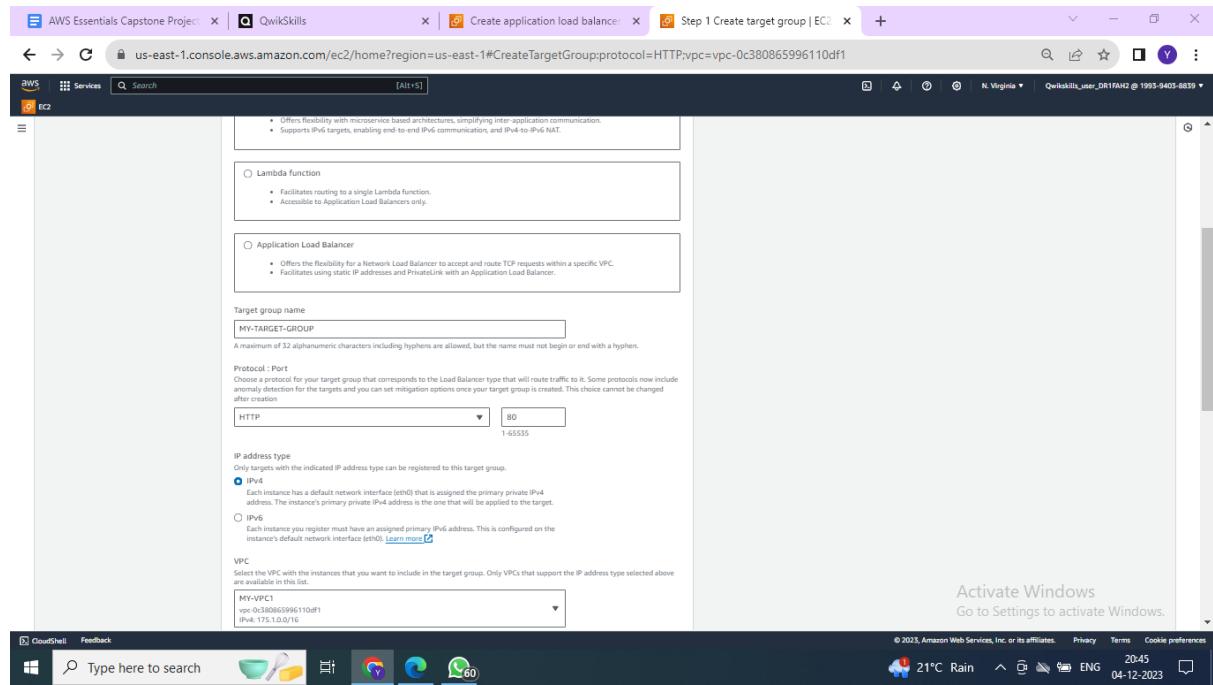
The screenshot shows the AWS CloudFormation Create application load balancer wizard. In the first step, 'Basic configuration', the 'Load balancer name' is set to 'MY-LB'. In the second step, 'Network mapping', the 'VPC' dropdown is set to 'my-VPC' (id: vpc-0360c996110ef1). Under 'Mappings', two subnets are selected: 'public-subnet-1' for 'us-east-1a (use1-aaz)' and 'public-subnet-2' for 'us-east-1b (use1-aaz)'. Both subnets are assigned by AWS.

The screenshot shows the AWS CloudFormation Create application load balancer wizard. In the final step, 'Security groups', the 'Security groups' dropdown is set to 'Select up to 5 security groups'. The 'Default' option is selected. The status bar at the bottom indicates the creation is in progress: 'Creating resources... 20:43 04-12-2023'.

- ❖ Create Target Group
- ❖ Target type= " Instances"
- ❖ Target group name="MY-TargetGroup"
- ❖ VPC="my-VPC" and Click "Next"

The screenshot shows the AWS CloudShell interface with multiple tabs open. The active tab is titled 'Create application load balance'. In the 'Listeners and routing' section, there is a configuration for a listener on port 80. The 'Default action' dropdown is set to 'Select a target group', which has 'MY-SG' selected. Other tabs visible include 'AWS Essentials Capstone Project' and 'QwikSkills'.

The screenshot shows the AWS CloudShell interface with multiple tabs open. The active tab is titled 'Step 1 Create target group | EC2'. In the 'Specify group details' section, 'Instances' is selected as the target type. Other tabs visible include 'AWS Essentials Capstone Project' and 'QwikSkills'.



- ❖ Register “Test-1” and “Test-2” as Targets
- ❖ Click “Create target group.”

Screenshot of the AWS Management Console showing the creation of a target group for an application load balancer.

The browser tabs are:

- AWS Essentials Capstone Project
- QwikSkills
- Create application load balancer
- Step 2 Create target group | EC2

The EC2 service page shows two instances selected for the target group:

Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID
i-04b7ba030bbe59e9d	test-1	Running	MY-SG	us-east-1b	175.1.16.226	subnet-0e2fcab9f31f5098
i-0a2e2d07a32e585de	test-2	Running	MY-SG	us-east-1a	175.1.15.21	subnet-0dad53cfcd61299b9

Ports for the selected instances are set to 80 and 1-65535.

**Review targets**

**Targets (2)**

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
X	Pending	i-04b7ba030bbe59e9d	test-1	80	Running	MY-SG	us-east-1b	175.1.16.226	subnet-0e2fcab9f31f5098	December 4, 2023, 20:1
X	Pending	i-0a2e2d07a32e585de	test-2	80	Running	MY-SG	us-east-1a	175.1.15.21	subnet-0dad53cfcd61299b9	December 4, 2023, 20:1

2 pending

Buttons: Cancel, Go to previous, Create target group, Next Step

Screenshot of the AWS Management Console showing the creation of a target group for an application load balancer.

The browser tabs are:

- AWS Essentials Capstone Project
- QwikSkills
- Create application load balancer
- Target group details | EC2 | us-east-1

The EC2 service page shows the target group "MY-TARGET-GROUP" successfully created:

Successfully created the target group: MY-TARGET-GROUP. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.

**MY-TARGET-GROUP**

**Details**

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-0c380865996110df1
IP address type	Load balancer IPv4 None associated		

Total targets: 2 (0 Healthy, 0 Unhealthy, 2 Unused, 0 Initial, 0 Draining)

Distribution of targets by Availability Zone (AZ): 0 Anomalous

**Registered targets (2)**

Anomaly mitigation: Not applicable

Buttons: Go to Settings to activate Windows, Deregister, Register targets

❖ Configuring a new Target Group that is to be attached with the Load Balancer

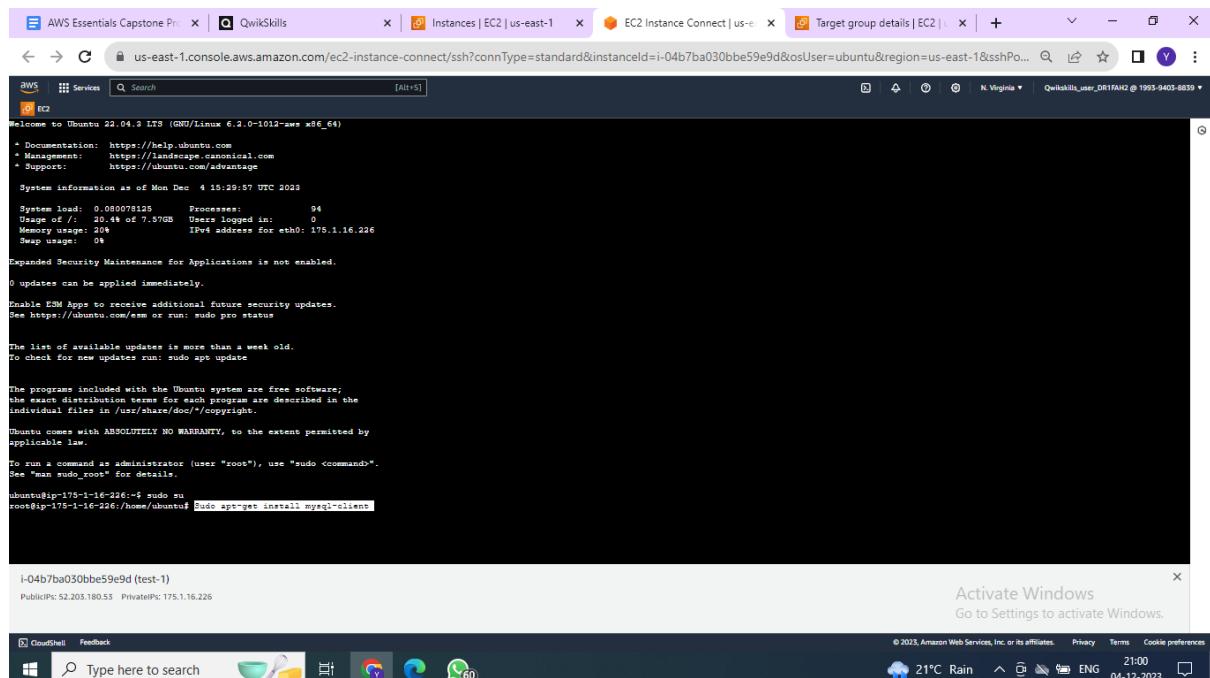
The screenshot shows the AWS CloudShell interface with multiple tabs open. The main tab displays the configuration of an Application Load Balancer (ALB) in the 'Create application load balancer' wizard. The 'Listeners and routing' section shows a listener for port 80 (HTTP) forwarding to a target group named 'MY-TARGET-GROUP'. The target group is associated with an EC2 instance. Other tabs visible include 'AWS Essentials Capstone Project', 'QwikSkills', 'Target group details | EC2 | us-east-1', and '+'. The bottom of the screen shows the Windows taskbar with various pinned icons.

❖ Successfully configured and created the Application Load Balancer.

❖ Go to running instances

✓ Connect to “Instance01” EC2 instance. → 52.203.180.53(instance01)

The screenshot shows the AWS CloudShell interface with multiple tabs open. The main tab displays the 'Connect to instance' configuration for an EC2 instance with ID i-04b7ba030bbe59e9d. The 'EC2 Instance Connect' option is selected, and the connection is set to use a browser-based client. The public IP address listed is 52.203.180.53. Other tabs visible include 'AWS Essentials Capstone Project', 'QwikSkills', 'Connect to instance | EC2 | us-east-1', and 'Target group details | EC2 | us-east-1'. The bottom of the screen shows the Windows taskbar with various pinned icons.



In Instance01" EC2 instance. → 52.203.180.53 (instance01)

- ❖ Type “sudo su” enter.
  - ❖ “Sudo apt-get install mysql-client” (Enter)
  - ❖ mysql -h my-database.c7k13rii9ip.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
  - ❖ Enter password: admin123 ❖ CREATE DATABASE mydatabase;
  - ❖ USE mydatabase;
  - ❖ Exit;

## **Script 1: Connect to RDS and create a new table with dummy**

```
#!/bin/bash
```

## # MySQL connection details

host= my-database.c7k13rii9ip.us-east-1.rds.amazonaws.com

**username=admin**

password=admin123

database=mydatabase

## # SQL query to create table

```
create_table_query="CREATE TABLE details (id INT PRIMARY KEY,  
name VARCHAR (50)); "
```

```
# SQL query to insert dummy data
INSERT INTO chandana(ID, NAME, AGE) VALUES (001, 'Rahul', 22);
INSERT INTO chandana(ID, NAME, AGE) VALUES (002, 'Budi', 21);
INSERT INTO chandana(ID, NAME, AGE) VALUES (003, 'Chandu', 20);
```

```
# Connect to MySQL and execute queries
mysql -h $host -u $username -p$password -D $database -e
"$create_table_query"
&& \
mysql -h $host -u $username -p$password -D $database -e
"$insert_data_query" echo "Table created with dummy data."
```

**Note:** - Connect to both EC2 instances in two separate browser tabs and run the below script.

```
#!/bin/bash
```

### **# Database configuration**

```
host= my-database.c7k13rii9ip.us-east-1.rds.amazonaws.com
username=admin
password=admin123
database=mydatabase
table=details
```

### **# 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)/"
```

```

AWS Essentials Capstone Proj | QwikSkills | RDS | us-east-1 | Load balancers | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | + | - | X
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0f840592b9c645146&osUser=ubuntu&s... | Search | Download | Print | Copy | Share | N. Virginia | QwikSkills_user_PBBCKQ6 @ 395B-75B3-0B55 |
AWS Services Search [Alt+S] | EC2
Enabling module mime.
Enabling module negotiation.
Enabling module rewrite.
Enabling module filter.
Enabling module deflate.
Enabling module headers.
Enabling module expires.
Enabling module requireoud.
Enabling conf charset.
Enabling conf localized-exerror-pages.
Enabling conf other-hosts-access-log.
Enabling conf log-symlink.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Module mod_event disabled.
Enabling module mpm_prefork.
Info: Executing deferred 'wtmpned php5.1' for package libapache2-mod-php5.1
Enabling module php5.
Create symlink /etc/systemd/system/multi-user.target.wants/apache2.service -> /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service -> /lib/systemd/system/apache-htcacheclean.service.
Setting up php5.1 (8.1.2-1ubuntu2.14) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libphp5-bin (2.25-0ubuntu9.3) ...
Processing triggers for libphp5.1-cli (8.1.2-1ubuntu2.14) ...
Processing triggers for libapache2-mod-php5.2 (8.1.2-1ubuntu2.14) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
Web application is now accessible at: http://44.221.177.91/
root@ip-175-1-9-258:/home/ubuntu#

```

i-0f840592b9c645146 (Test-1)  
PublicIP: 44.221.177.91 PrivateIP: 175.1.9.258

Activate Windows  
Go to Settings to activate Windows.

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 21°C Rain 12:36 ENG 05-12-2023

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

- ❖ I-0f840592b9c645146 (**instance-id**)
- ❖ **Public ip : 52.203.180.53**
- ❖ I-0ec303bf983174665 (**instance-id**)
- ❖ **Public ip : 34.226.181.247**

DNS name info : MY-LB-545557948.us-east-1.elb.amazonaws.com

## cloudTrail :-

- ❖ In search bar type “CloudTrail” then hit enter

The screenshot shows the AWS CloudTrail Dashboard. On the left, there's a sidebar with options like Dashboard, Insights, Lake, Pricing, Documentation, Forums, and FAQs. The main area has sections for "Introducing CloudTrail Lake" (with a note about querying multiple event fields), "CloudTrail Insights" (disabled), and "Event history". The "Event history" section lists several events from December 5, 2023, such as "AutherizeSecurityGroup", "SendSSHPublicKey", and "CreateLogStream". At the bottom right, there's a Windows taskbar with icons for CloudShell, Feedback, Start, Task View, File Explorer, Edge, and File Explorer. The status bar shows "Activate Windows Go to Settings to activate Windows.", "© 2023, Amazon Web Services, Inc. or its affiliates.", "Privacy Terms Cookie preferences", "21°C Rain 12:51 05-12-2023", and "CloudTrail Insights is not enabled".

- ❖ Trail name=“MY-cloud trial”
- ❖ Storage location=“ Create new S3 bucket”
- ❖ And enter details as shown below.

This screenshot shows the "Choose trail attributes" step of the "Create trail" wizard. It's divided into three tabs: Step 1 (Choose trail attributes), Step 2 (Choose log events), and Step 3 (Review and create). The Step 1 tab is active. It contains a "General details" section with a note that a multi-region trail is created in the console. The "Trail name" field is set to "My\_cloudtrail". Below it is a note about character restrictions: "3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed." There are two options for "Storage location": "Create new S3 bucket" (selected) and "Use existing S3 bucket". The "Create new S3 bucket" option includes a note: "Create a bucket to store logs for the trail." The "Use existing S3 bucket" option includes a note: "Choose an existing bucket to store logs for this trail." At the bottom, there's a "Trail log bucket and folder" section with a note: "Enter a new S3 bucket name and folder (prefix) to store your logs. Bucket names must be globally unique." A pre-filled value "aws-cloudtrail-logs-415356710414-20128d5e" is shown in the input field. A small note at the bottom right says: "I now will be stored in [aws-cloudtrail-logs-415356710414-20128d5e.s3.amazonaws.com](#)".

**CloudWatch Logs - optional**

Configure CloudWatch Logs to monitor your trail logs and notify you when specific activity occurs. Standard CloudWatch and CloudWatch Logs charges apply. Learn more [\[?\]](#)

CloudWatch Logs | [Info](#)

Enabled

New  
 Existing

Log group name

aws-cloudtrail-logs-415356710414-125b9c18

1-512 characters. Only letters, numbers, dashes, underscores, forward slashes, and periods are allowed.

AWS CloudTrail assumes this role to send CloudTrail events to your CloudWatch Logs log group.

New  
 Existing

Role name

qwik\_role

► Policy document

**Tags - optional** [Info](#)

You can add one or more tags to help you manage and organize your resources, including trails.

Key	Value - optional	Remove
Enter key	Enter value	<a href="#">Remove</a>

[Add tag](#)

You can add 49 more tags

[Cancel](#) [Next](#)

- ❖ Role name:- `qwik_role`
- ❖ Click “Next”

CloudTrail > Dashboard > Create trail

Step 1 [Choose trail attributes](#)

Step 2 [Choose log events](#)

Step 3 [Review and create](#)

### Choose log events

**Events** [Info](#)  
 Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply \[?\]](#)

Event type  
 Choose the type of events that you want to log.

<input checked="" type="checkbox"/> Management events Capture management operations performed on your AWS resources.	<input checked="" type="checkbox"/> Data events Log the resource operations performed on or within a resource.	<input checked="" type="checkbox"/> Insights events Identify unusual activity, errors, or user behavior in your account.
---	---	---

- ❖ Choose log events (Management events, Data event, Insight events)
- ❖ Management events (Read, Write)
- ❖ Click on switch to advance event selector.
- ❖ Data event source=S3
- ❖ Click on Next
- ❖ Review and then click on Create Trail

- ❖ Enter details as shown below and then click “Next”

**Insights events** Info

Identify unusual activity, errors, or user behavior in your account. Additional charges apply.

Choose Insights types

Insights measure unusual activity against a seven-day baseline.

**API call rate**  
A measurement of write-only management API calls that occur per minute against a baseline API call volume.

**API error rate**  
A measurement of management API calls that result in error codes. The error is shown if the API call is unsuccessful.

Cancel Previous Next

- ❖ Review and then click “Create trail” button

	Name	Home region	Multi-region trail	Insights	Organization trail	S3 bucket	Log file prefix	CloudWatch Log Log group	Status
<input type="radio"/>	transformer-logs-events	US East (N. Virginia)	Yes	Disabled	Yes	logs-136169858-184-10fa4e4f	-	amazonawslog-size-east-1:415356710414-log-groups-cloudtrail-logs-415356710414-125fb9c18*	<span>Logging</span>
<input type="radio"/>	My_cldtrail	US East (N. Virginia)	Yes	Enabled	No	aws-cloudtrail-logs-d15356710414-20128d5e	-	amazonawslog-size-east-1:415356710414-log-groups-cloudtrail-logs-415356710414-125fb9c18*	<span>Logging</span>

- ❖ Cloud trial created successfully.

## S3: - (Simple Storage Service).

- ❖ Bucket created by clouptrail to store the logs generated

**Amazon S3**

**Amazon S3**

**Account snapshot**

Total storage: Pending Object count: Pending Average object size: Pending

You can enable advanced metrics in the "default-account-dashboard" configuration.

**General purpose buckets** (1) Info

Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
aws-cloudtrail-logs-395875830855-0216d611	US East (N. Virginia) us-east-1	Bucket and objects not public	December 5, 2023, 12:56:14 (UTC+05:30)

Activate Windows  
Go to Settings to activate Windows.

CloudShell Feedback Type here to search 21°C Rain 13:16 ENG 05-12-2023

## CLOUDWATCH: -

- ❖ Go to CloudWatch → Log groups.
- ❖ Select Log group of “my-trail”

The screenshot shows the AWS CloudWatch Log Groups interface. The left sidebar navigation includes: Dashboards, Alarms, Logs (selected), Metrics, X-Ray traces, Events, Application Signals, and CloudShell. The main content area displays 'Log groups (2)'. A table lists the log groups: 'RDSMetrics' (Standard, Configure, 1 month retention) and 'aws-cloudtrail-logs-395875830855-32625d2d' (Standard, Configure, Never expire). Action buttons include 'Actions' (Delete, Edit, Create metric filter, Create contributor insights rules, Create data protection policy, Anomaly detection - new, Subscription filters, Export data to Amazon S3, View all exports to Amazon S3), 'View in Logs Insights', 'Start tailing', and 'Create log group'.

This screenshot is identical to the one above, but a context menu is open over the 'aws-cloudtrail-logs...' log group. The visible options in the 'Actions' menu are: Delete (log groups), Edit (retention setting(s)), Create metric filter (highlighted in blue), Create contributor insights rules, Create data protection policy, Anomaly detection - new, Subscription filters, Export data to Amazon S3, and View all exports to Amazon S3.

- ❖ Click Actions → Create metric filter.
- ❖ Enter `$.eventName="StopInstances"` in Filter pattern
- ❖ Select Log data which is in the format as shown below!
- ❖ Click “Next”

The screenshot shows the AWS CloudWatch Metrics Filter Pattern creation interface. It's a three-step wizard:

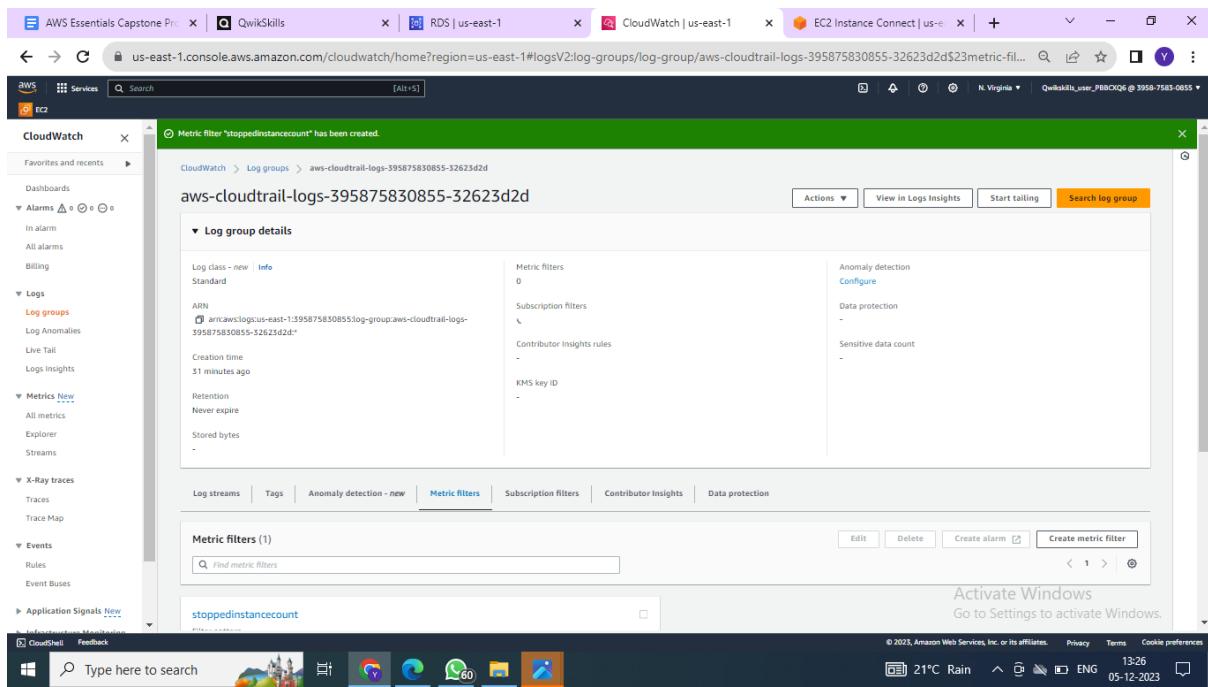
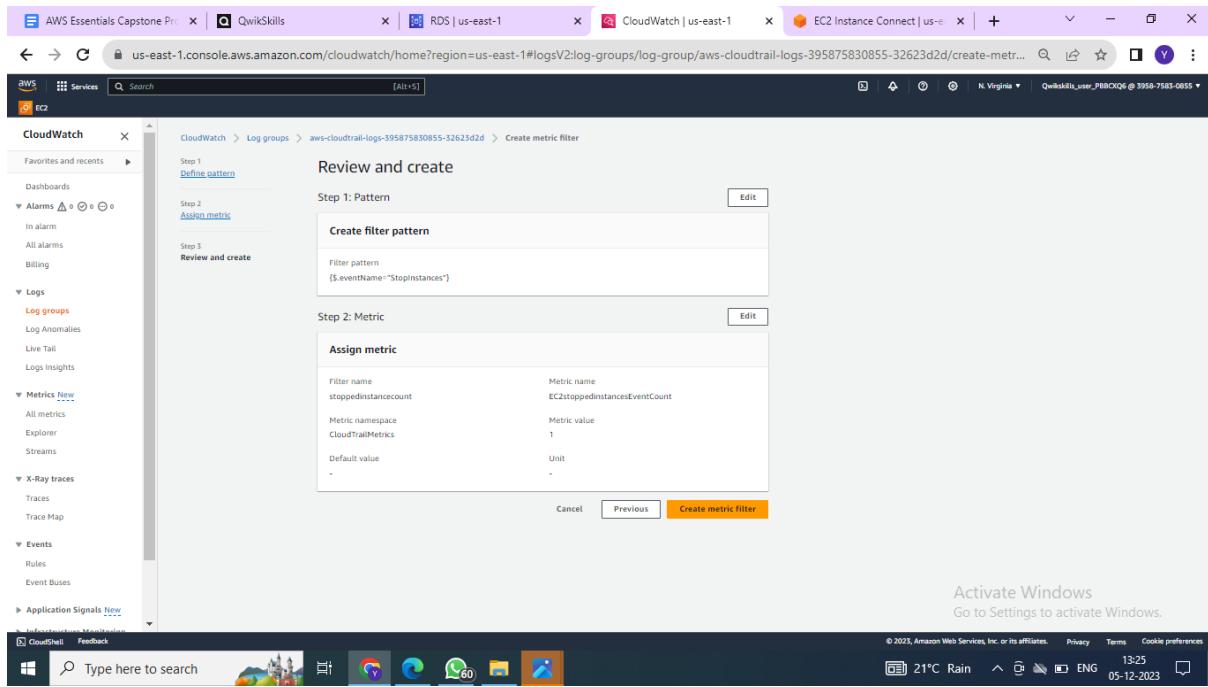
- Step 1: Define pattern**: Shows the filter pattern `[$eventName:="StopInstances"]`.
- Step 2: Assign metric**: Shows the filter name `stoppedinstancecount`, the metric namespace `CloudTrailMetrics`, and the metric name `EC2stoppedInstanceEventCount`.
- Step 3: Review and create**: Shows the metric details: Metric namespace `CloudTrailMetrics`, Metric name `EC2stoppedInstancesEventCount`, Metric value `1`, and Default value `Enter default value`.

- ❖ Filter Name=stopped instance count
- ❖ Metric namespace= CloudTrailMetrics
- ❖ Metric name= EC2stoppedInstanceEventCount
- ❖ Metric value=1
- ❖ Click on Next

The screenshot shows the AWS CloudWatch Metrics Filter Pattern creation interface. It's a three-step wizard:

- Step 1: Define pattern**: Shows the filter pattern `[$eventName:="StopInstances"]`.
- Step 2: Assign metric**: Shows the filter name `stoppedinstancecount`, the metric namespace `CloudTrailMetrics`, and the metric name `EC2stoppedInstanceEventCount`.
- Step 3: Review and create**: Shows the metric details: Metric namespace `CloudTrailMetrics`, Metric name `EC2stoppedInstancesEventCount`, Metric value `1`, and Default value `Enter default value`.

## ❖ Review and click “Create metric filter.”



## CloudWatch Alert :

- ❖ Go to “Metric filters”
- ❖ Select “stoppedInstancecount” and then click “Create alarm.”

Specify the metric conditions as below:

- ❖ Namespace=CloudTrailMetrics(default)
- ❖ Metric name=EC2stoppedInstancesEventCount(default)
- ❖ Statistic=Sum(default)
- ❖ Period=5 minutes
- ❖ Threshold type=Static
- ❖ Whenever EC2stoppedInstancesEvent count is=Greater /Equal than 1

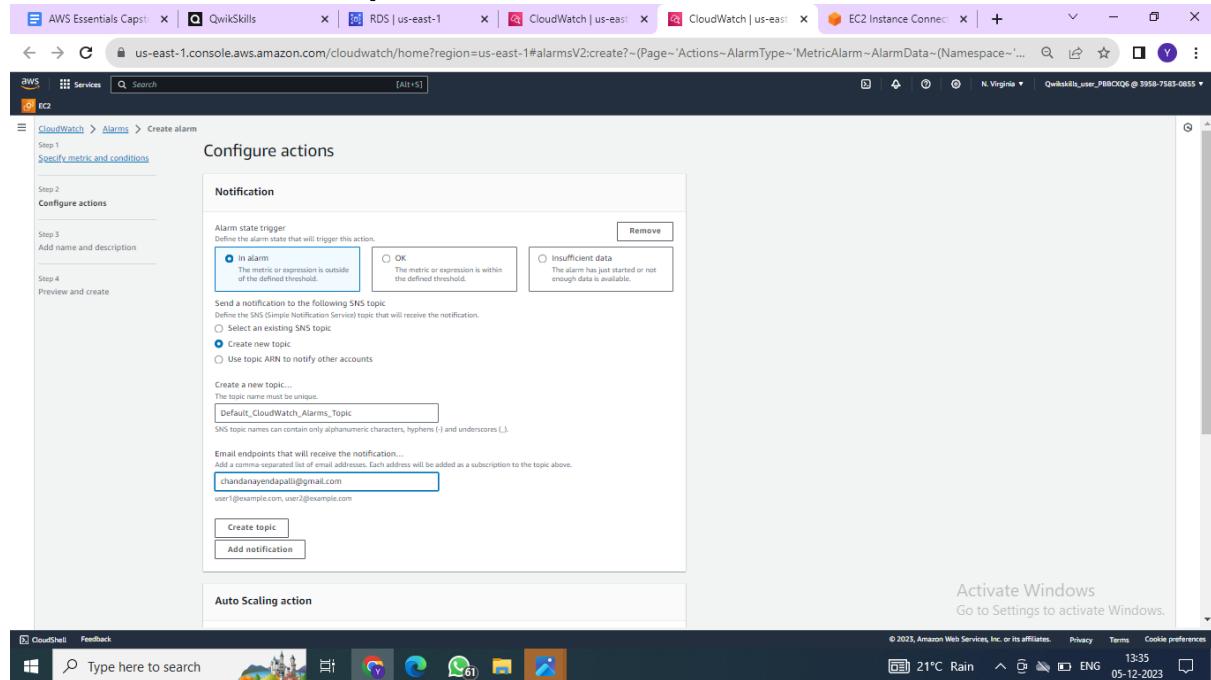
The screenshot shows the 'Create alarm' wizard on the 'Specify metric and conditions' step. The 'Metric' section displays a graph of EC2stoppedInstancesEventCount over a one-minute period, with the threshold set at 1. The 'Conditions' section shows the threshold type selected as 'Static' (Greater/Equal). Below the graph, it says 'Whenever EC2stoppedInstancesEventCount is...'. The 'Greater/Equal' option is selected. The rest of the wizard steps are visible on the left.

❖ Preview and click on create alarm

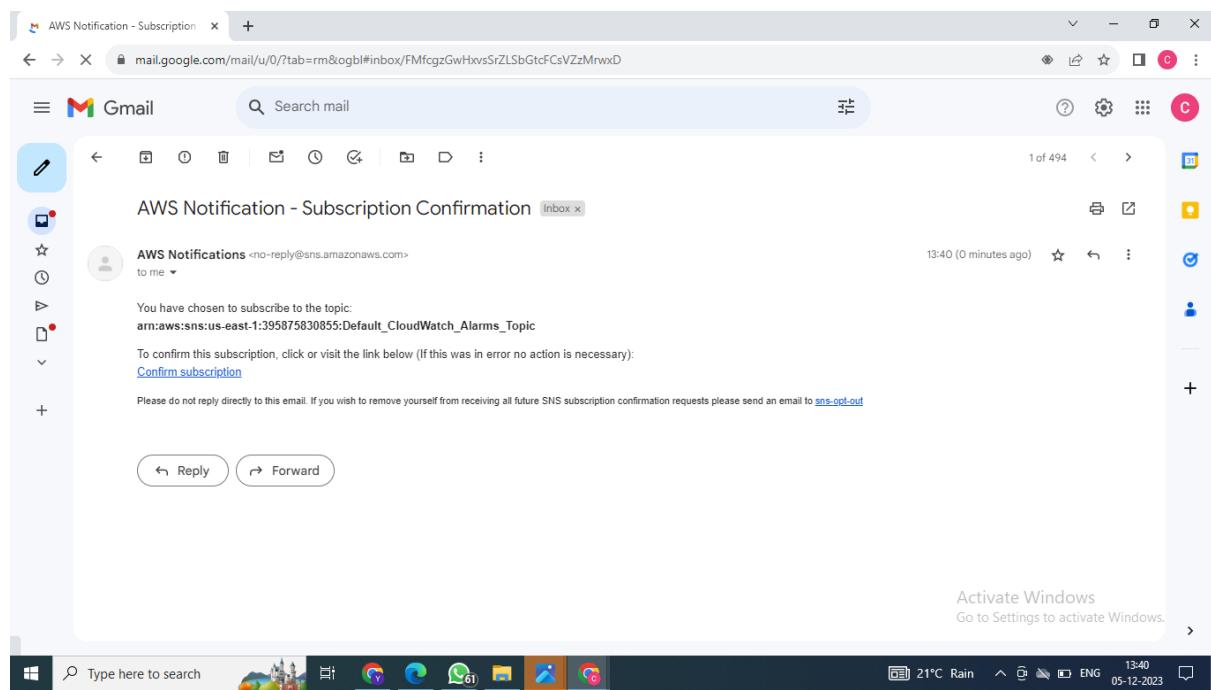
- ❖ Configure actions=In alarm
- ❖ Select an existing SNS topic.
- ❖ Send a notification to mail id:- [chandanayendapalli@gmail.com](mailto:chandanayendapalli@gmail.com)

The screenshot shows the 'Configure actions' step of the wizard. Under the 'Notification' section, the 'In alarm' option is selected. Below it, an SNS topic named 'Default\_CloudWatch\_Alarms\_Topic' is listed under 'Send a notification to the following SNS topic'. The 'Email endpoints that will receive the notification...' field contains the email address 'chandanayendapalli@gmail.com'. The 'Create topic' and 'Add notification' buttons are visible at the bottom.

## Email for SNS subscription :



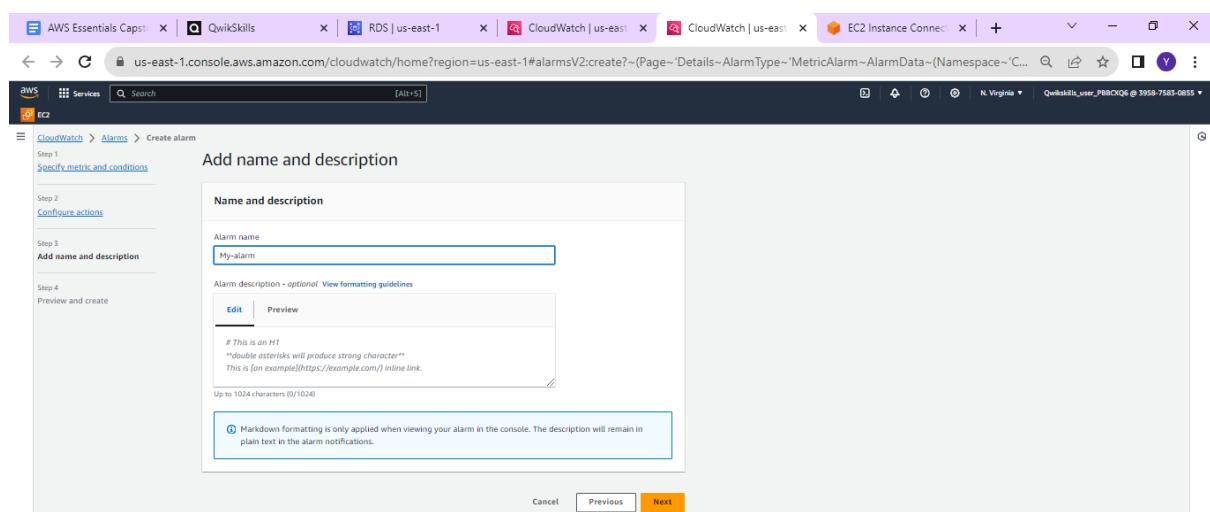
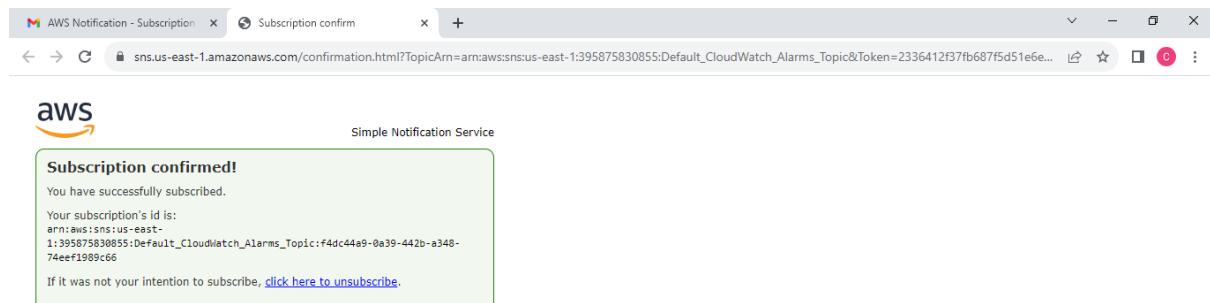
- ❖ Go to the email you entered earlier, check for any email from AWS Subscribe to SNS.

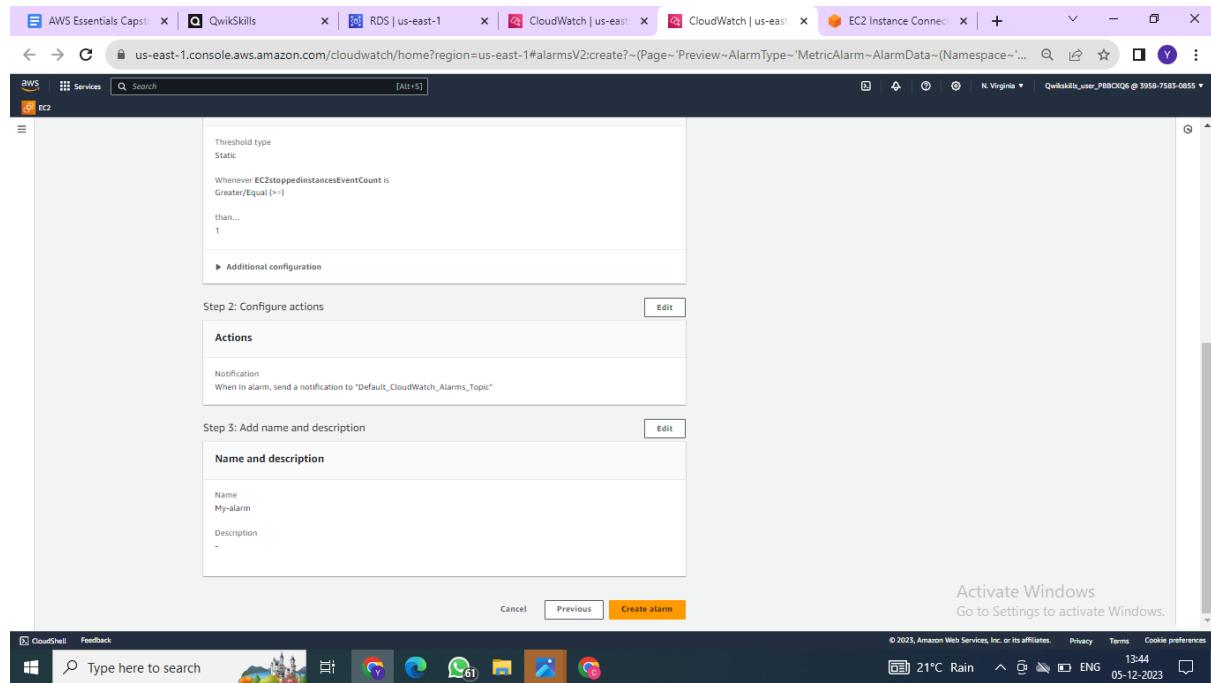


## SNS (Simple Notification Service):-

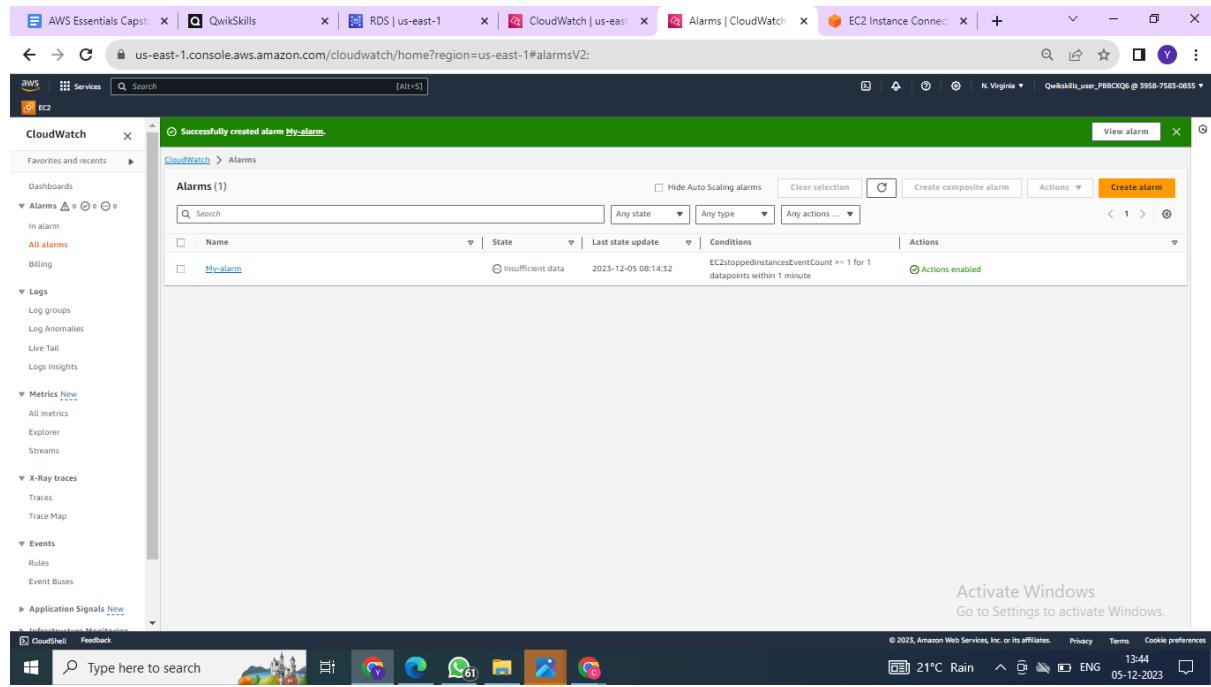
Notification on Email

- ❖ Go to the email you entered earlier, check for any email from AWS Subscribe to SNS





## ❖ Alarm created successfully.



## ❖ Go to EC2 Console and Launch an Instance with the following configuration

**Name and tags**

Name: MyEC2Server

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

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

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

Amazon Machine Image (AMI): Amazon Linux 2023 AMI (ami-0230bd0aa40260c6 [64-bit (x86)] / ami-04-97462b19d5f3f1 [64-bit (Arm)])

**Summary**

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.2.2... (read more)

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) > 8 GiB

**Launch Instance**

Free tier: In your first year includes: 750 hours of t2.micro (or t3.micro) in the Regions in which t2.micro is unavailable. Includes up to 100 hours per Month, 20 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

Activate Windows: Go to Settings to activate Windows.

- ❖ Wait till the “MyEC2Server” instance get into Running state.
- ❖ Stop the “MyEC2Server” instance.

**Instances (3) info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPV6
Test-1	i-0f840592b0c645146	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	44.221.177.91	44.221.177.91	-	-
MyEC2Server	i-0a5a752437d6b1b05	Running	t2.micro	Initializing	No alarms	us-east-1a	-	-	-	-
Test-2	i-0ec303bf083174665	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	54.253.243.179	54.253.243.179	-	-

Select an instance:

Activate Windows: Go to Settings to activate Windows.

**Instances (1/3) info**

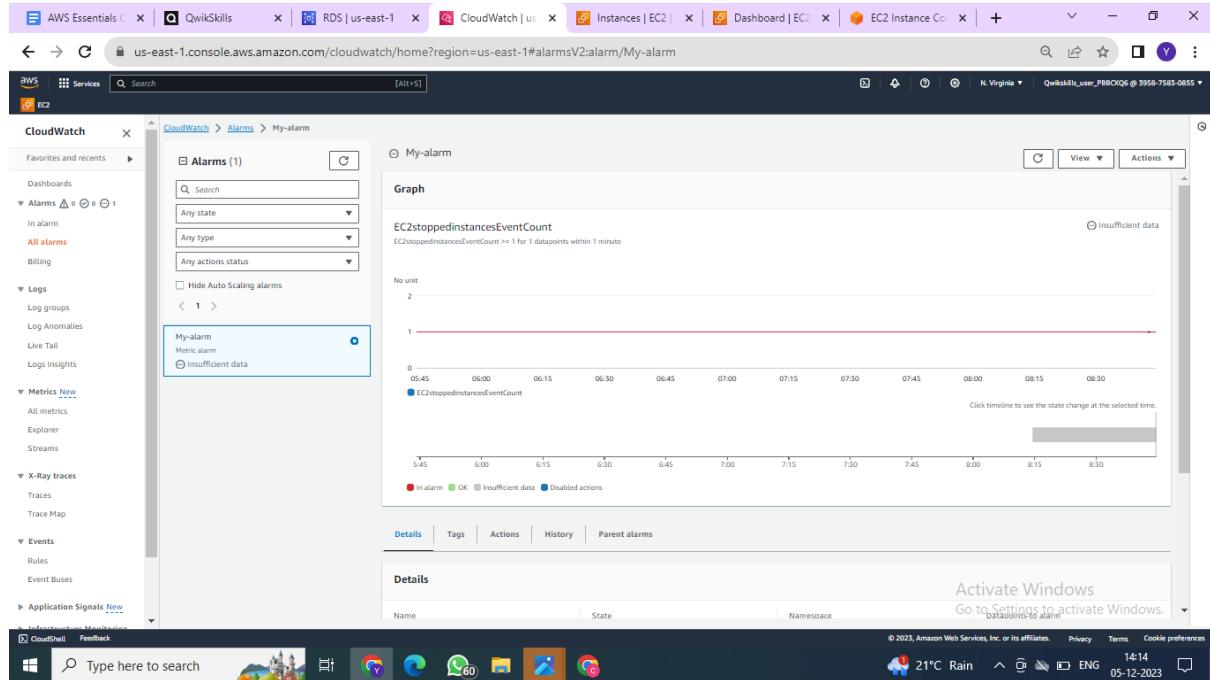
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPV6
Test-1	i-0f840592b0c645146	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	44.221.177.91	44.221.177.91	-	-
MyEC2Server	i-0a5a752437d6b1b05	Stopped	t2.micro	Initializing	No alarms	us-east-1a	-	-	-	-
Test-2	i-0ec303bf083174665	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	54.253.243.179	54.253.243.179	-	-

Instance: i-0a5a752437d6b1b05 (MyEC2Server)

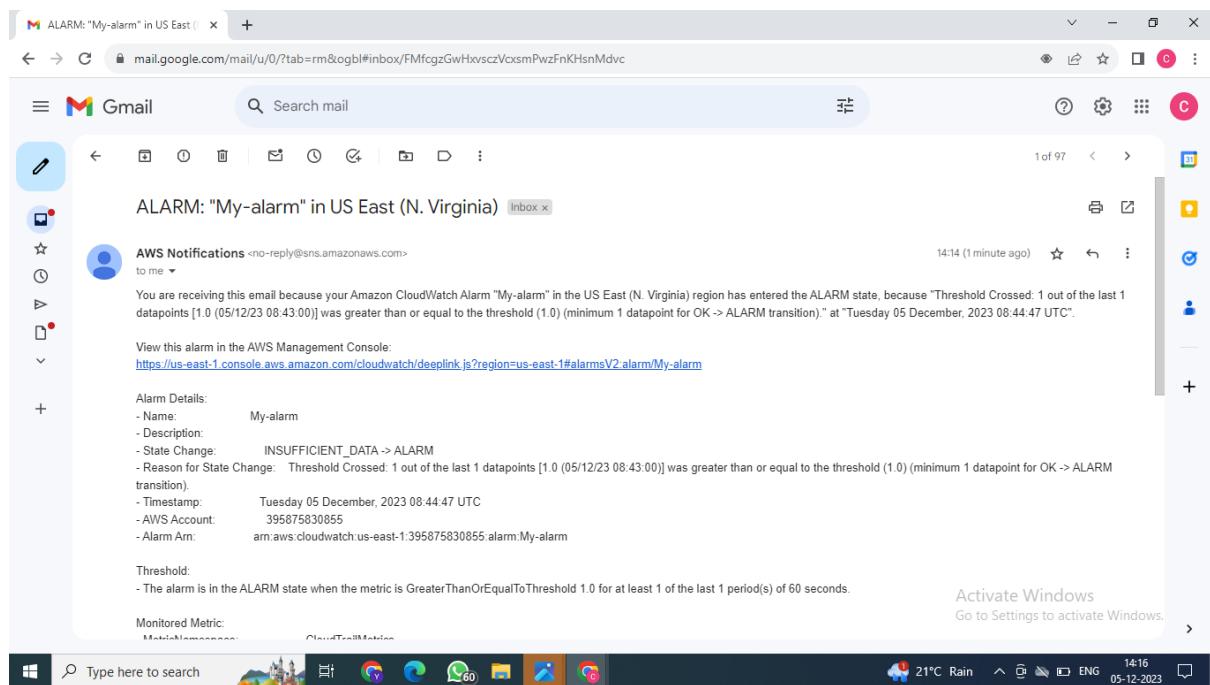
Activate Windows: Go to Settings to activate Windows.

## CLOUDWATCH ALERT: -

- ❖ You can check the alert found in the alarm "My\_stopped\_ec2\_alarm" as mentioned below.



- ❖ You will get an SNS email at your email address



# CONCLUSION

The successful launch of a high-availability web application is a testament to careful planning, meticulous execution, and the dedication of the entire team. Achieving high availability in a web application is a complex task, but when done right, it can result in a seamless and reliable user experience.

The project carried out has underscored the fact that AWS is an invaluable resource for devising scalable, secure, and productive IT structures. By deploying AWS services such as **EC2 instances, RDS, S3 Security Groups, VPC, Cloud Trail, and Cloud Watch**, we have illuminated AWS's broad capabilities for creating an extensively and Highly Available Web Application.

The **RDS** service automated laborious tasks such as hardware provisioning, database setup, patching, and backups, thereby allowing us to focus on enhancing the application to provide swift, reliable services to end users.

With its ability to scale, provide security, and boost performance, Amazon S3 has been a linchpin of our project. Its high durability, constant availability, and extraordinary scalability have been invaluable in handling, analysing, and managing vast and diverse datasets effortlessly. We put the logs stored in the S3 to use by implementing the **Cloud Trail**.

We utilised **Security Groups** to act as a virtual firewall for our EC2 instances, regulating inbound and outbound traffic and ensuring protection from unauthorised access, which boosted our system's overall security by permitting the All Web Traffic.

Lastly, **Cloud Watch** played a significant role by offering a platform to monitor applications, gather and track metrics, and react to system-wide performance changes in each user account.

In wrapping up, it's clear that AWS services play a crucial role in creating resilient, expandable, and efficient Highly Available Web Applications... The successful realisation of our project serves as a solid proof of AWS's proficiency in optimising the cloud computing experience.

The successful launch of a high-availability web application not only provides an excellent user experience but also instil confidence in both users and stakeholders. However, it's important to remember that high availability is an ongoing commitment, and vigilance is required to maintain it. As the application evolves and grows, the team should remain proactive in addressing challenges and staying ahead of potential issues to ensure its continued success.

# SALIENT FEATURES

By implementing these AWS features and best practices, you can create a highly available web application that can withstand various failures and provide a reliable experience for your users.

Building a highly available web application on Amazon Web Services (AWS) requires careful planning and implementation of various services and features. Here are some of the salient features and best practices to achieve high availability on AWS:

- 1. Multi-Availability Zone Deployment:** Distribute your application across multiple AWS Availability Zones (data centres) within a region. This ensures redundancy and fault tolerance. AWS offers automatic failover between Availability Zones.
- 2. Elastic Load Balancing:** Use Elastic Load Balancers (ELB) to distribute incoming traffic across multiple instances in different Availability Zones. ELBs also provide automatic scaling and health checks.
- 3. Auto Scaling:** Implement Auto Scaling groups to automatically adjust the number of instances in your application based on traffic or resource utilisation. This helps maintain performance and availability during traffic spikes.
- 4. Database Replication:** Use AWS RDS (Relational Database Service) with Multi-AZ deployments for database high availability. This replicates your database across different Availability Zones with automatic failover.
- 5. Content Delivery:** Utilise Amazon CloudFront for content delivery to cache and distribute content from edge locations, reducing latency and enhancing availability.
- 6. Global Content Distribution:** For global availability, leverage AWS services like Amazon Route 53 for DNS-based routing to direct users to the nearest AWS region or edge location.
- 7. Redundant Data Storage:** Use Amazon S3 for object storage and maintain data redundancy across multiple Availability Zones to ensure

data availability.

**8. Caching:** Implement in-memory caching using Amazon Elasticache (Redis or Memcached) to reduce database load and improve application performance.

**9. Application Load Balancers:** Use Application Load Balancers (ALB) to route HTTP/HTTPS traffic to different instances, enabling fine-grained control over traffic routing and fault tolerance.

**10. Monitoring and Alerts:** Set up AWS Cloud Watch for real-time monitoring of application and infrastructure health. Configure alarms to notify you when there are issues.

**11. Disaster Recovery:** Create backups of your data and systems using AWS services like Amazon Backup or snapshots. Implement a disaster recovery plan to recover from catastrophic failures.

**12. Security:** Implement security best practices, such as network security groups, AWS Identity and Access Management (IAM), and VPC (Virtual Private Cloud) for isolation and access control.

**13. Documentation and Testing:** Document your architecture and disaster recovery procedures thoroughly, and regularly test your failover and recovery processes to ensure they work as expected.

**14. Session Management:** Utilise stateless application design or external session management to ensure that user sessions are not tied to a specific instance.

**15. Immutable Infrastructure:** Use infrastructure as code (IAC) and tools like AWS CloudFormation to create and manage infrastructure in a repeatable and predictable manner.

**16. Blue-Green Deployments:** Implement blue-green deployment strategies to minimise downtime during updates and ensure seamless application availability.

**17. Distributed Databases:** For large-scale applications, consider MySQL databases like Amazon RDS for high availability and scalability.

## FUTURE SCOPE

The successful establishment of a highly available web application on Amazon Web Services (AWS) not only signifies a major accomplishment but also opens up numerous opportunities for future growth.

This project's achievements serve as a solid foundation for potential advancements, leveraging the robust AWS infrastructure and established principles.

Key areas for future development include implementing advanced features, expanding globally, adopting microservices architecture, embracing continuous improvement, exploring multi-cloud and hybrid solutions, integrating edge computing, tailoring industry-specific solutions, enhancing data analytics, adopting serverless architecture, exploring IoT integration, and incorporating green initiatives.

Overall, the project's success positions it as a dynamic solution, ready to adapt to emerging trends and user needs to remain a leader in the digital landscape.

# BIBLIOGRAPHY/REFERENCES

- QwikSkills AWS access and other references taken from  
<https://dashboard.qwikskills.com/dashboard>
- AWS VPC documentation:  
<https://docs.aws.amazon.com/vpc/index.html>
- Amazon free tier account: <https://aws.amazon.com/console/>
- EC2 Documentation for all the EC2 instances Configuration.  
<https://docs.aws.amazon.com/ec2/index.html>
- Amazon Simple Storage Service (S3) Documentation:  
<https://docs.aws.amazon.com/s3/index.html>
- Amazon RDS (Relational Database Service) Documentation  
<https://docs.aws.amazon.com/rds/index.html>
- CloudWatch documentation  
<http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/UsingSetupSNS.html>
- Component based documentation for multiple References of AWS.  
<https://docs.aws.amazon.com/imagebuilder/latest/userguide/toe-use-documents.htm>