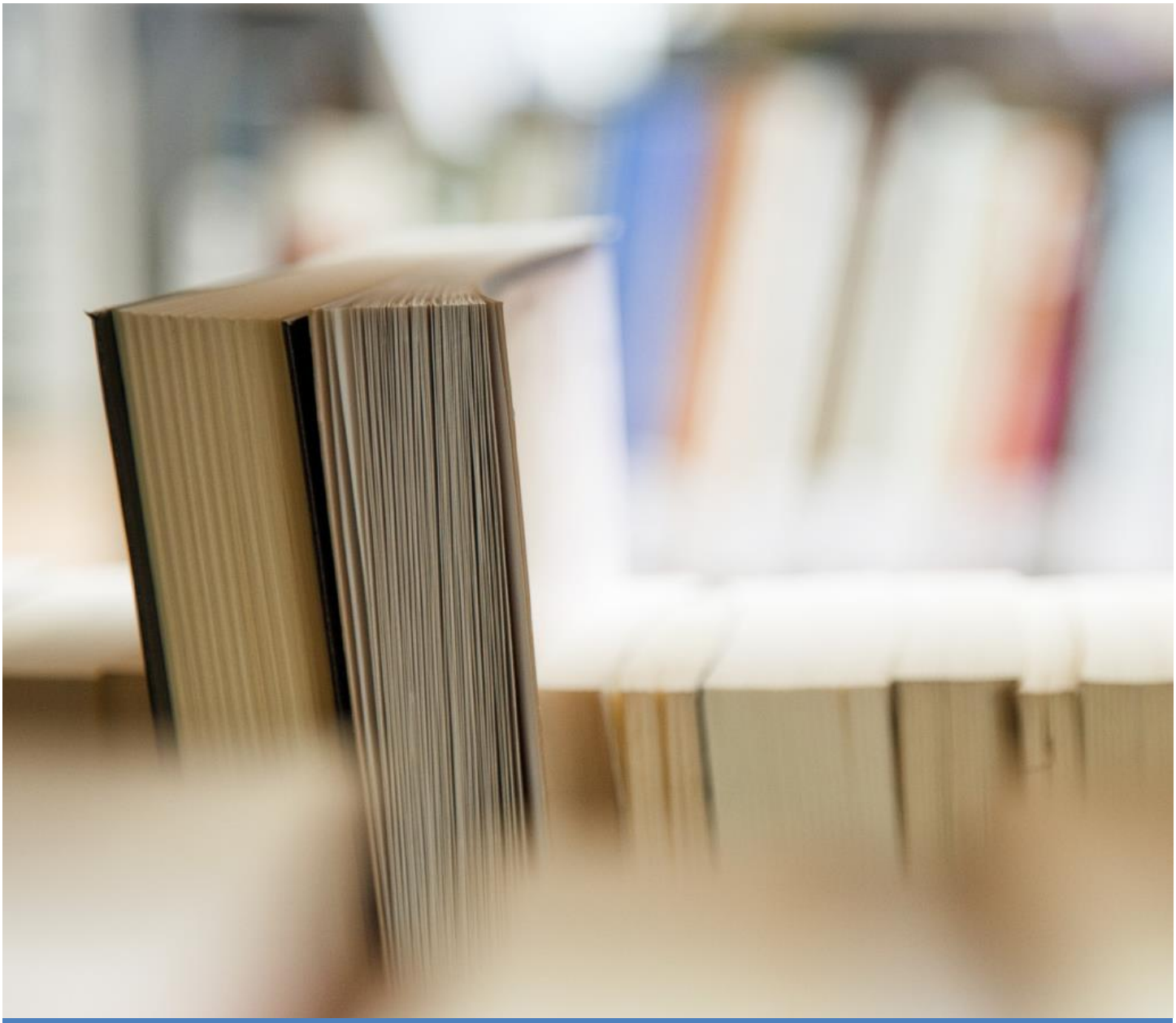


Virtualization and Cloud Computing

Assignment Submitted by: Purushothaman S

Roll number: G24AI1042

Submitted to: Dr. Sumit Kalra **3/22/25** **Assignment No. 3**

Virtual Machine Creation with Resource Monitoring along with Auto Scaling

Introduction

Deploying scalable and resilient infrastructure is essential in today's cloud-driven environment. This document provides a comprehensive guide to setting up a Windows-based Virtual Machine (VM) in Amazon Web Services (AWS) with integrated resource monitoring, alert systems, auto scaling, and load balancing using VPC networking components. The configuration ensures performance, availability, and cost-efficiency for dynamic workloads.

VPC- Virtual Private Cloud

A **VPC (Virtual Private Cloud)** in AWS is a **virtual network** that you create in the AWS cloud. It is like having your own private data center in the cloud where you can launch and manage AWS resources (like EC2 instances, databases, etc.) securely.

A **Virtual Private Cloud (VPC)** in AWS is a logically isolated network that allows you to securely launch and manage cloud resources. It offers customizable IP address ranges using CIDR, and lets you divide the network into public subnets (with internet access via an Internet Gateway) and private subnets (for internal communication). VPCs use route tables, security groups, and network ACLs to control traffic flow and enhance security. With NAT Gateways, private subnets can securely access the internet without being exposed. VPCs also support connections to other networks through peering or VPN. Overall, VPCs provide strong security, flexible network configurations, and scalability for cloud infrastructure

Step-by-Step Summary

1. **VPC and Networking Setup:** Start by creating a Virtual Private Cloud (VPC) with a public subnet, internet gateway, and route table to ensure your Windows VM is internet-accessible.
2. **Launch Windows EC2 Instance:** Use a Windows AMI to create an EC2 instance in the configured subnet. Assign a public IP and allow RDP access for remote desktop connectivity.

3. **CloudWatch Monitoring and Alarms:** Enable monitoring of CPU utilization using Amazon CloudWatch. Set up an alarm to notify via email if CPU usage exceeds 75% using SNS.
4. **Launch Template and Auto Scaling Group:** Create a launch template for your instance configuration, then define an Auto Scaling Group (ASG) that manages the number of instances based on CPU usage.
5. **Target Group and Load Balancer:** Create a target group and configure an Application Load Balancer to distribute traffic among instances. Associate the ALB with your ASG for dynamic instance management.
6. **Load Balancer Template:** Use the provided AWS CloudFormation template to automate load balancer creation and configuration.

By following these summarized steps, you build a resilient infrastructure capable of scaling with demand, complete with automated monitoring and notification.

1. VPC and Networking Setup

Step 1: Create a VPC

- Go to the VPC dashboard.
- Click "Create VPC".
- Name: "**test-vpc".**
- IPv4 CIDR: 12.0.0.0/16.

Step 2: Create Subnet

- VPC: "**test-vpc**'' // select the created VPC by using the VPC name.
- Name: ''**Test-Public-Subnet-1a''** // Name the subnet.
- Availability Zone: Choose one (e.g., ap-south-1a) // Select the Zone Asia Pacific-Mumbai.
- CIDR: **"2.0.1.0/16"** // Ensure the Higher number of availability such as 256.

Step 3: Create Internet Gateway

- Create a new IGW named **"Igw-test"** // Click the Create Internet Gate way Name it **.**
- Attach it to "**test-vpc**" // Attach the created VPC by clicking Action.

Step 4: Create Route Table

- Create route table named "**rt-test-public**''. // Click Create Route Table and Name it.

- Select the VPC // Select the test-vpc which has been created.
- Click Subnet Associations and select the created Subnets.
- Click the Route / Edit Route /Add Route 0.0.0.0/0.
- Click the Dropdown and Select the Created the Internet gateway.
- Click SAVE changes.

Step 4: Target Group Creation

- Click "**AWS**" Home Button.
- Click "**EC2**".
- Click "**Load Balancing**".
- Click "**Target Group Click".**
- Click "**Create Target Group**".
- Select "**Instance**".
- Give the Target Group Name "**tr-ec2-apache2**".
- Select Protocol: Port.
- HTTP / 80.
- IP Address Type: IPv4.
- Select The VPC – Which we created "**test-vpc**".
- Protocol Version / HTTP 1: Click "**Next**" Scroll Down to "**Create Target Group**".

Step 5: Load Balancer Creation

- Click "**AWS**" Home Button.
- Click "**EC2**".
- Click "**Load Balancing**".
- Click "**Application Load Balancer".**
- Click "**Create**".
- Give the Name "**alb-ec2-instances-with-asg**".
- Select Internet Facing.
- Select VPC // "**test-vpc**" that we have created
- Select Both Subnet

- Select Security Group. Our Application Load Balancer Need to access the Internet so we need to create a new securing rule we need to enable port 80 and give name "**alb-sg-for-http-req".**
- Type – **Allow http request.**
- VPC – Select "**test-vpc**".
- Click **Create Security Group.**
- Copy the Name of Security Group and go to the previous page and Select the Created Security Group.
- Listener and Routing.
- Select the **Target Group**
- Select the **Created Target Group**

Step 6: Auto Scaling

- Click "**AWS**" Home Button.
- Click "**EC2**".
- Click "**Auto Scaling**".
- Click "**Auto Scaling Groups".**
- Click "**Create Auto Scaling Groups**".
- Give the Name "**Asg-ec2-instances-test-demo**".
- Click "**Create a Launch Template**".
- Give the template Name "**lt-ec2-instances-apache2**"
- Click **AMI** => Browse / Select **Windows Image**
- Instance Type – **t2Micro** or **t3Micro**
- Select Security Group now we must create a Security Group
  Name it "**lt-sg-ec2-instances-apache2**" here allow SSH and HTTP request
- VPC "**test-vpc**" select the created vpc
- Add **HTTP /80/ 0.0.0.0/0**
- Add **SSH /22/ 0.0.0.0/0**
- Click "**EC2**".
  Then go back to previous page and select the created Security Group

- Goto Network Interface
- Auto **Enable-Public ID / Enable** it
- Scaling Policy Type: Target Tracking
- Metric Type: Average CPU Utilization
- Target value: 50%
- Desire Instance: 1
- Min Instances: 1
- Max Instances: 3

Monitoring The Resource

Set up CloudWatch Monitoring and Alarm

Step 1: Create Alarm for CPU Utilization

- Go to CloudWatch > Alarms > Create Alarm
- Select metric: EC2 > Per-Instance Metrics > CPU Utilization
- Conditions: Threshold type: Static, CPU Utilization > 75%
- Period: 5 minutes

Step 2: Notification

- Create new SNS topic (e.g., HighCPUAlert)
- Add email subscription to SNS topic
- Confirm subscription via email

Step 3: Attach Alarm to Instance

- Actions: Send notification to HighCPUAlert

Step 4: Add Auto Scale

- Select the Auto Scaling Policies and enable it

Instance Creation and Remote Desktop Connection

Launch a VM (EC2) Instance

1. Log in to the AWS Management Console.
2. Go to EC2 under "Compute" services.
3. Click “Launch Instance.”
4. Configure the following:
    - Name: Give your instance a name.
    - AMI: Choose a Windows AMI (e.g., Windows Server).

- Instance Type: Choose a type (e.g., t2. micro for free tier).
- Key Pair: Create or select an existing key pair (you will need this to decrypt your password).
- Network Settings:
  - Allow RDP (port 3389) in the security group.
  - Choose a public subnet and auto-assign public IP (for external access).
- Click Launch Instance.

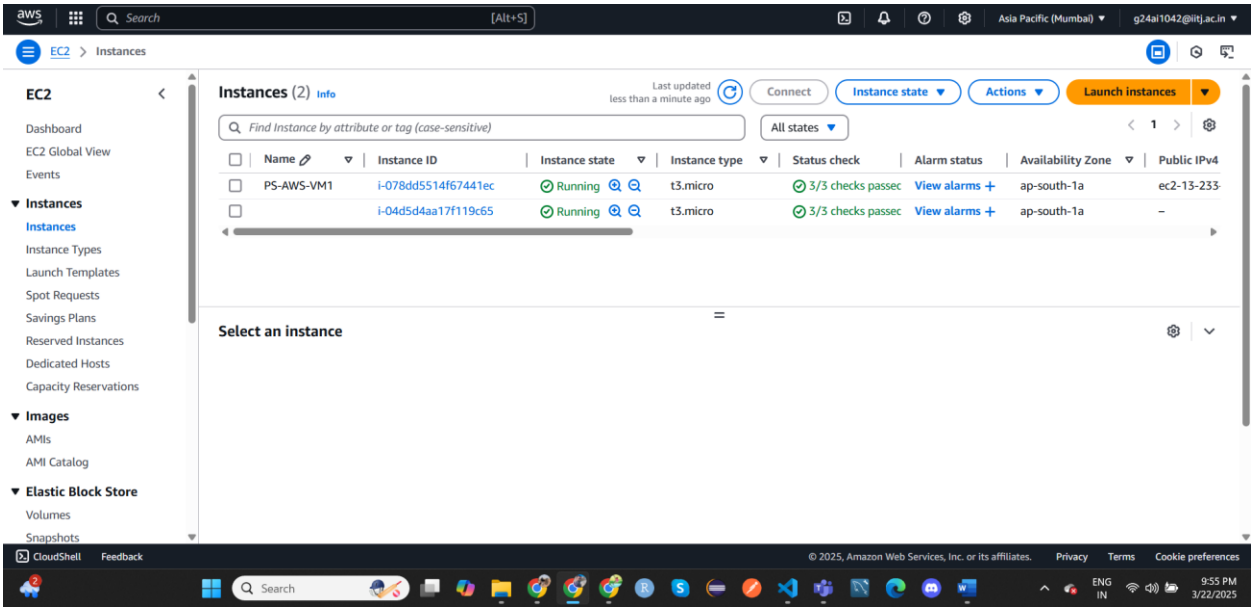
Step 2: Get Windows Password

1. Once the instance is running, go to the Instances page.
2. Select your instance → Click “Connect” → Go to RDP Client tab.
3. Click “Get Password”.
   - Upload the .pem file (your key pair).
   - AWS will decrypt and show the Windows administrator password.

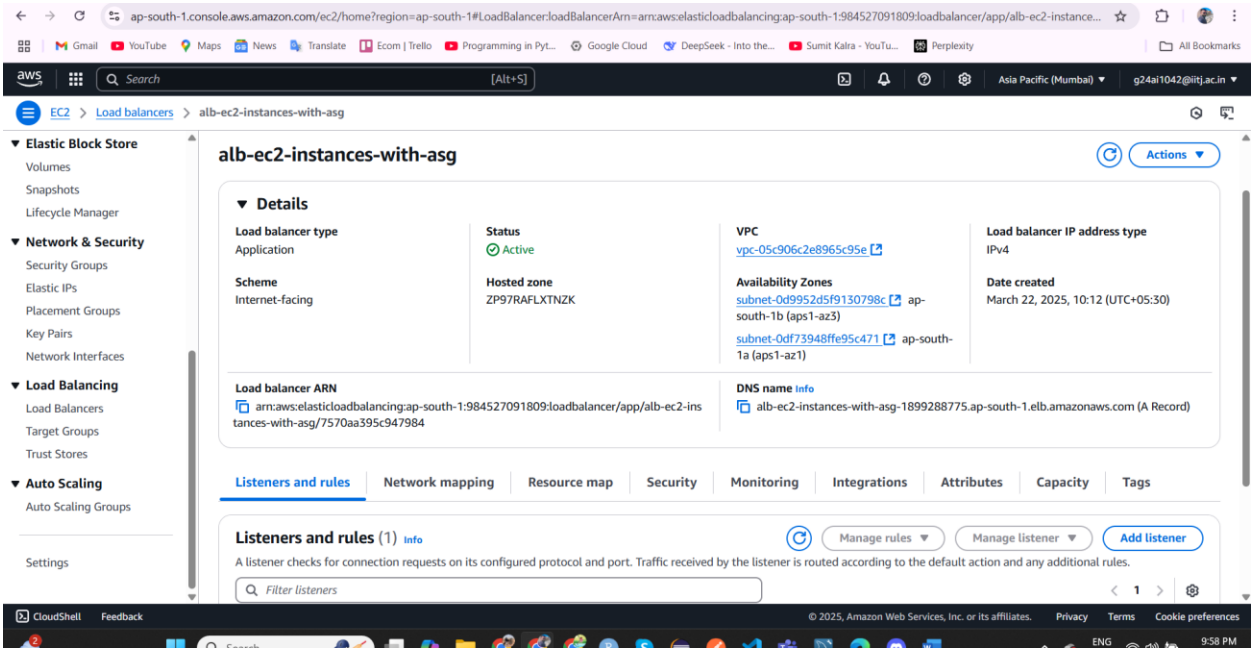
Step 3: Connect Using Remote Desktop

1. Open Remote Desktop Connection (on Windows: search for "mstsc").
2. In the Computer field, enter your instance’s public IP address.
3. Use the username Administrator and the decrypted password.
4. Click Connect – accept the certificate warning if prompted.

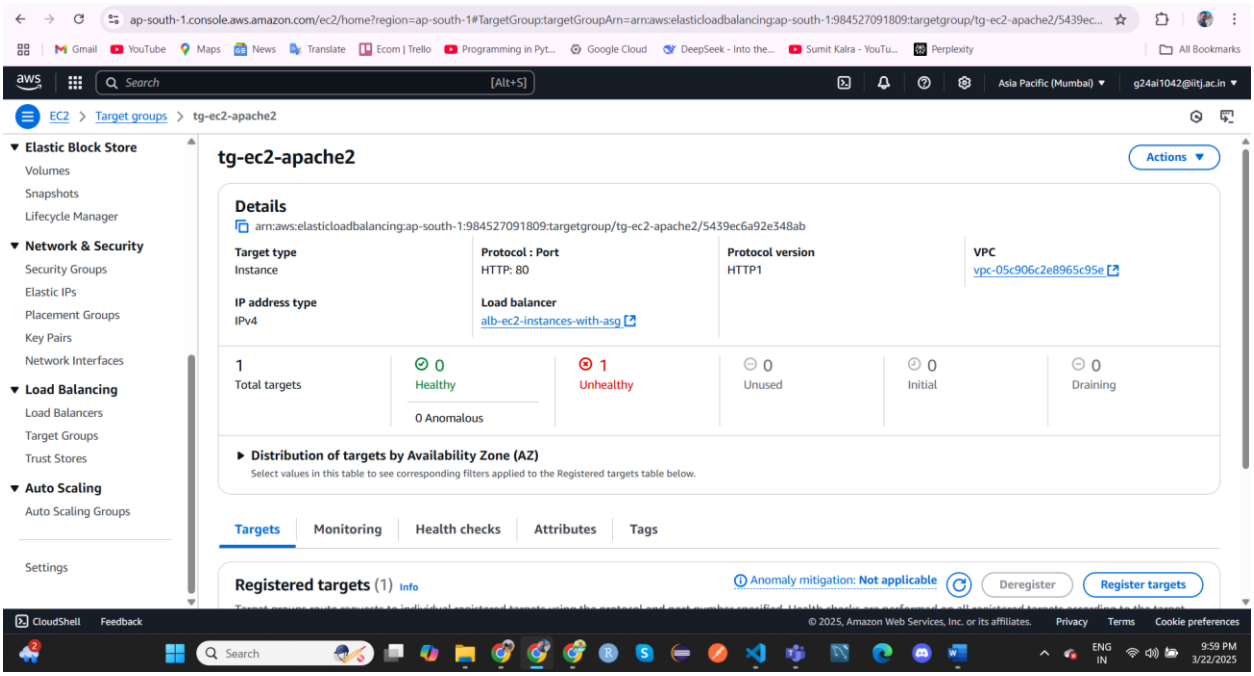
Creates Instance and Auto scaled Instance



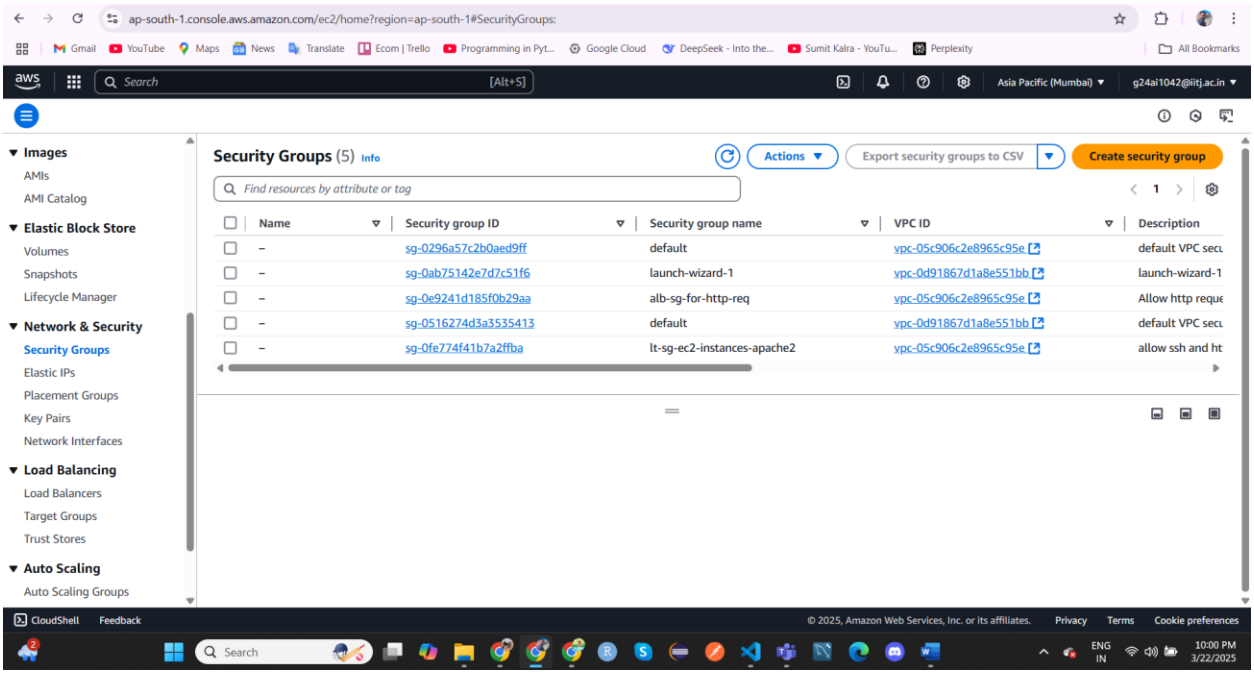
Load Balancer



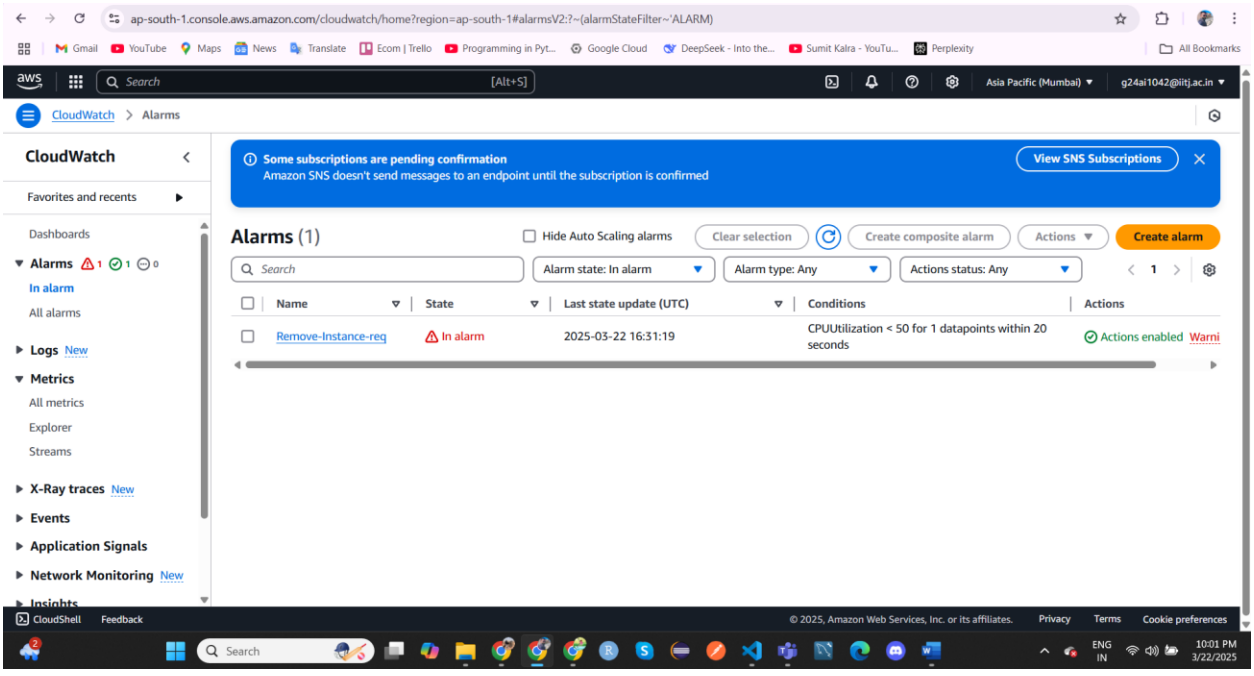
Targeted Group



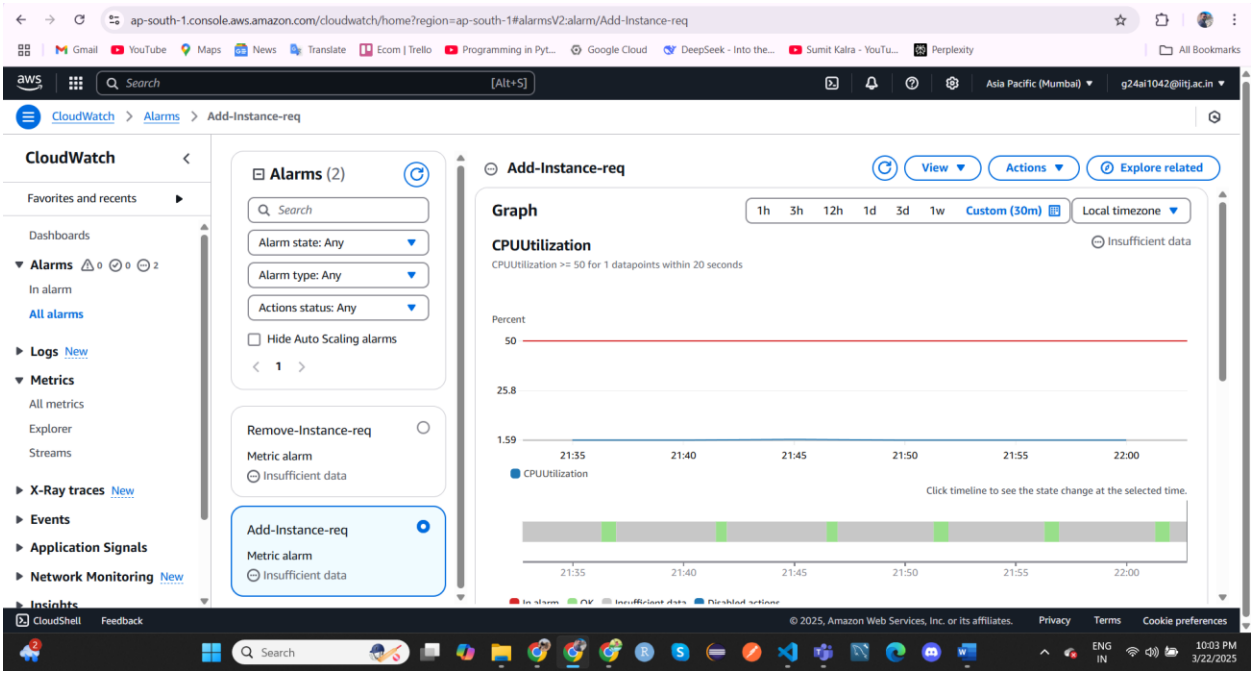
Security Group



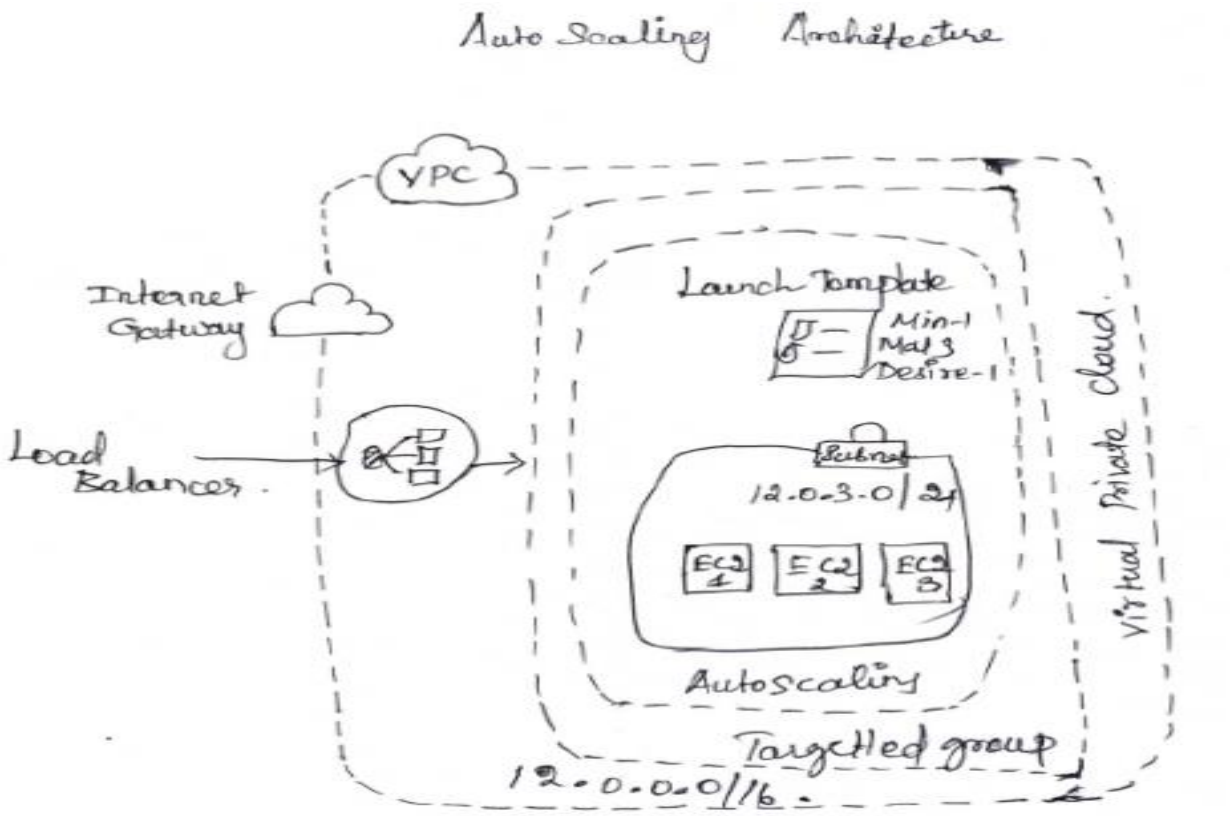
Cloud Watch



CPU Utilization Monitoring



Architecture Diagram



Conclusion

Through this guide, you have created a complete and scalable Windows server environment in AWS. You have set up foundational networking components, launched a Windows VM, and configured remote access. Furthermore, you've enabled CloudWatch monitoring with alerting, established an Application Load Balancer, and configured an Auto Scaling Group to maintain performance under varying loads. This architecture enhances uptime, optimizes performance, and automates response to high CPU utilization.

Web References

- Amazon EC2 Documentation
- Amazon VPC Documentation
- Amazon CloudWatch Documentation
- Auto Scaling Documentation
- Load Balancer Documentation
- CloudFormation Template Reference