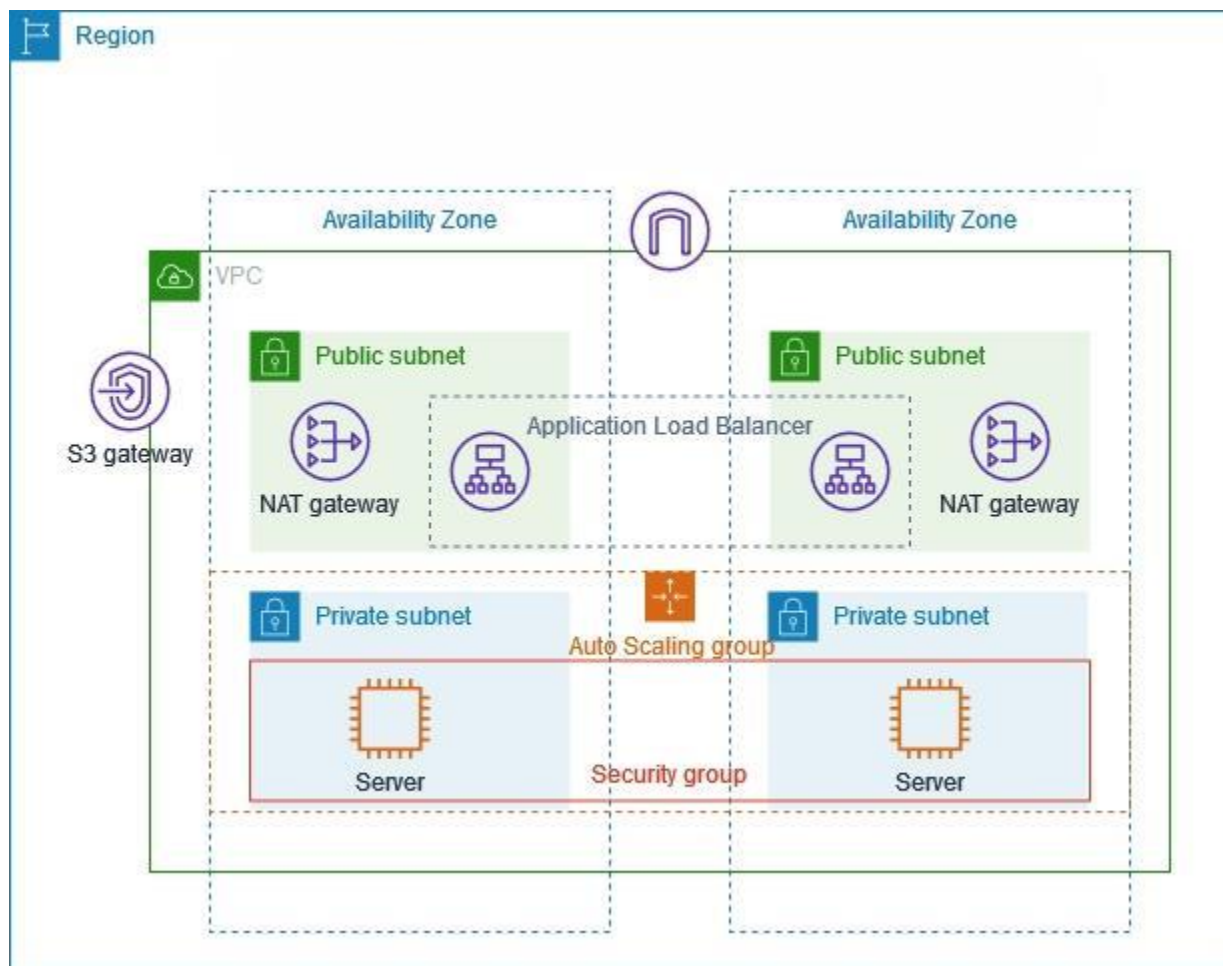


AWS Multi-AZ Private EC2 Application Deployment with Load Balancer and Auto Scaling

Overview :-

This project demonstrates a secure, scalable AWS architecture implemented as an IAM user, with key components like VPC segmentation, load balancing, auto-scaling, and NAT gateway configuration. The architecture is designed to ensure scalability, security, and high availability.



Architecture Highlights :-

- VPC: Isolated public and private subnets to enhance network segmentation and security.
- Application Load Balancer: Distributes traffic across instances in multiple Availability Zones (AZs) to ensure high availability.
- Auto Scaling: Auto Scaling Group adjusts the number of instances dynamically based on CPU utilization.
- NAT Gateway: Provides secure internet access for instances in private subnets.
- Security Groups: Granular control over instance access, enforcing security best practices.

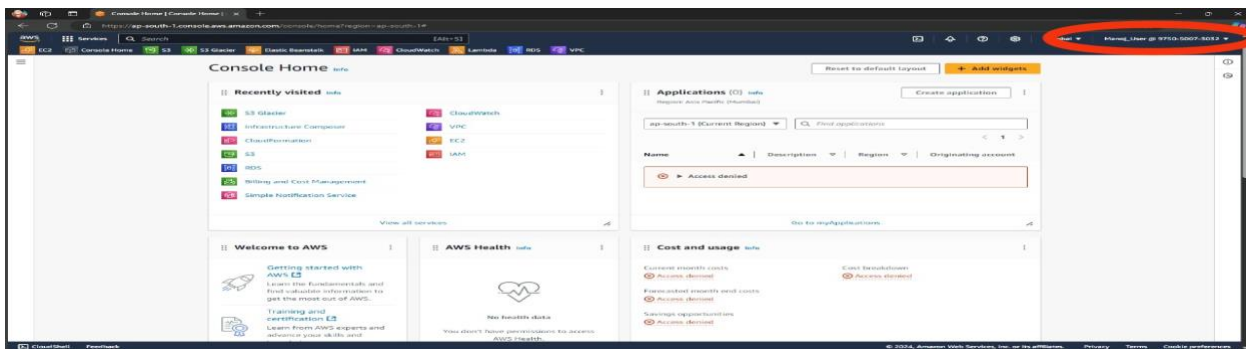
Benefits :-

- Enhanced Security: Hosting instances in private subnets protects them from direct internet exposure, reducing potential attack surfaces.
- Cost Efficiency: By using the NAT Gateway for outbound access, the need for public IPs on instances is eliminated, helping reduce costs.

Setup Steps :-

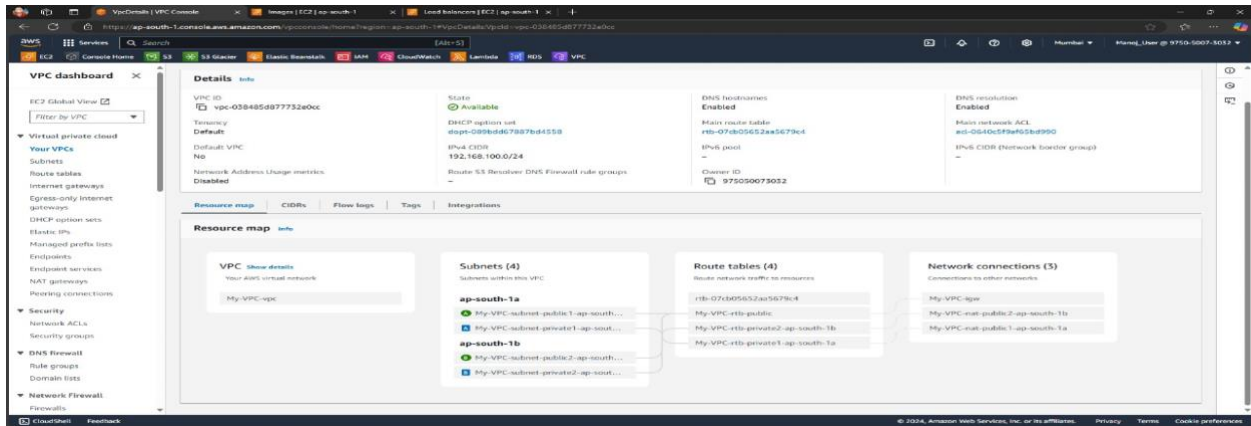
1. IAM User Permissions

- set up an IAM user with the necessary permissions to manage resources such as VPC, EC2, Auto Scaling, and Load Balancer.
- Apply the principle of least privilege, granting only the permissions required to ensure secure access management.



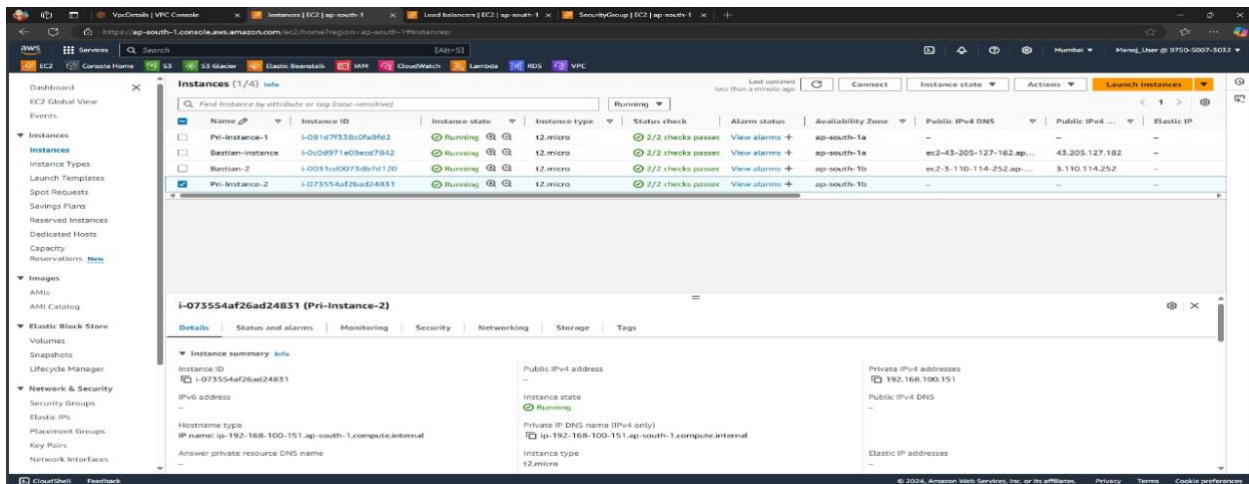
2. VPC & Subnets

- Create a new VPC to isolate resources and organize your network.
- Create public and private subnets in multiple Availability Zones for improved availability and security.



3. EC2 Instances in Private Subnets

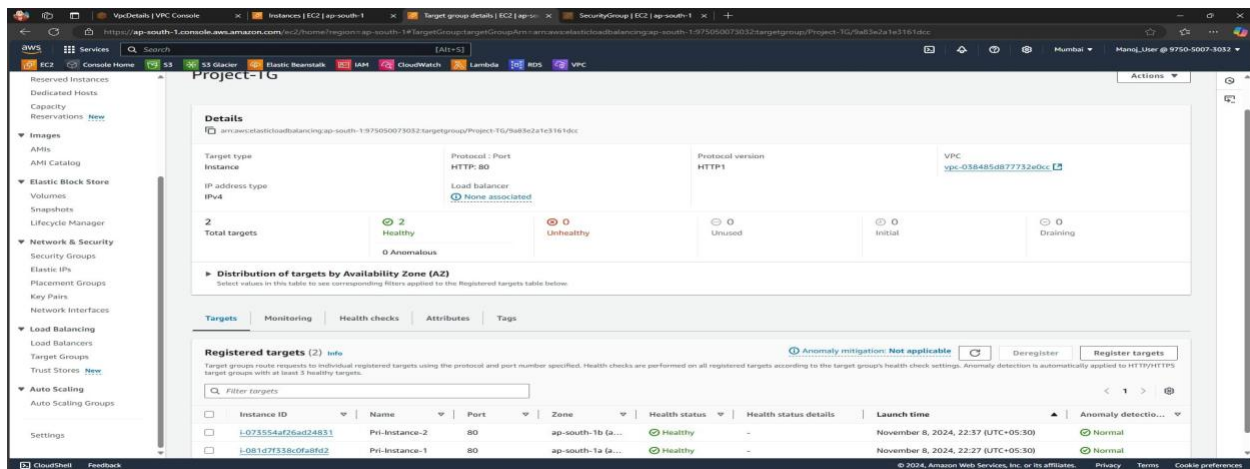
- Launch EC2 instances in private subnets across different AZs to enhance security and availability.
- Configure Security Groups to control access to these instances as per the application requirements.



4. Application Load Balancer (ALB)

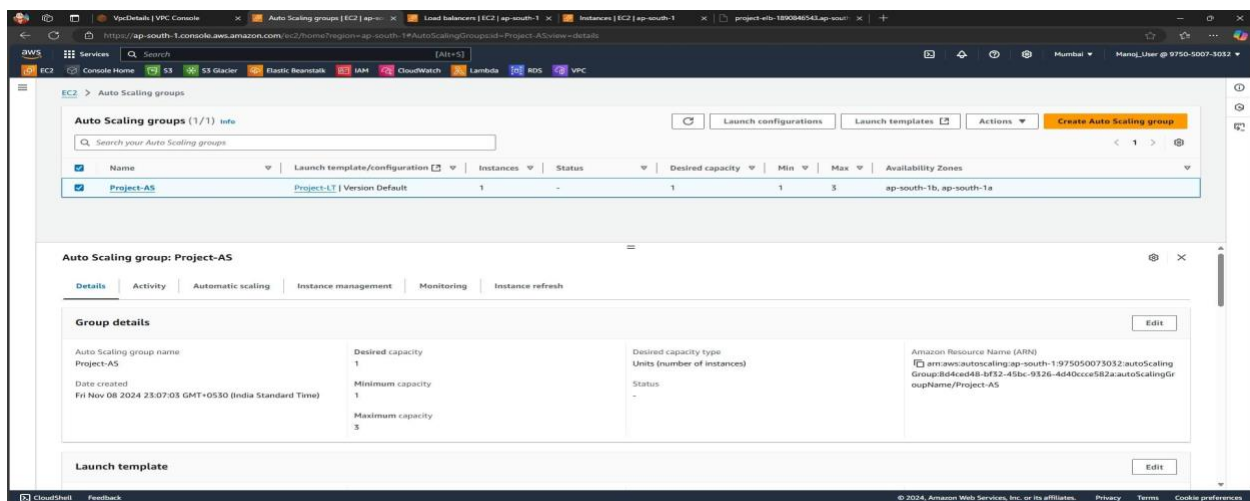
- Set up an Application Load Balancer in the public subnets to route traffic to EC2 instances in private subnets.

- Attach EC2 instances to the load balancer's target group to distribute traffic efficiently.



5. Auto Scaling

- Configure an Auto Scaling Group linked to the load balancer to handle traffic spikes. Set CPU utilization to 80% as the threshold for scaling.



Project result and outcomes:- [Click here to open the link :-](#)

1.NAT Gateway Efficiency: Using a NAT Gateway for outbound internet in private subnets removes the need for public IPs, reducing costs for public-facing instances.

2. Auto Scaling: Automatically adjusts instances based on demand, minimizing costs by scaling down during low traffic.

3. Load Balancer Savings: The Application Load Balancer distributes traffic efficiently, charging only for actual usage.

4. Reserved Instances/Savings Plans: For consistent workloads, these options lock in discounts, further reducing EC2 and NAT Gateway expenses.