**Traffic Distribution Using Load Balancer and autoscaling group**

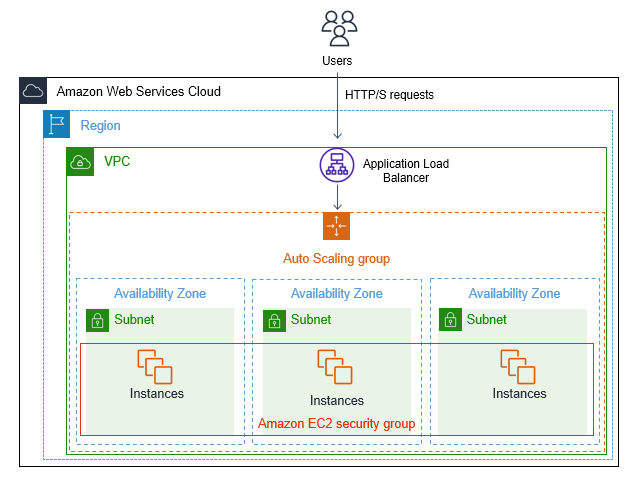
**Summary :- In our project, we're incorporating both a load balancer and an autoscaling group.**

**The load balancer evenly distributes incoming web traffic across our servers, preventing any single server from becoming overwhelmed.**

**Meanwhile, the autoscaling group automatically adjusts the number of servers based on demand, scaling up during peak periods and scaling down during quieter times.**

**This setup ensures that our system can handle fluctuating traffic loads efficiently, providing a reliable and responsive experience for our users while minimizing operational management.**

Diagram:-Top of Form



Execution Steps:-

1. \*Create Launch Configuration\*: Navigate to the Amazon EC2 service in the AWS Management Console. Create a launch configuration specifying your AMI, instance type, key pair, security groups, etc.

2. \*Create Auto Scaling Group\*: In the EC2 service, create an Auto Scaling group using the launch configuration you created. Specify the desired capacity, minimum and maximum capacity, subnets, and any scaling policies.

3. \*Set up Health Checks\*: Configure health checks for your Auto Scaling group. Define the protocol, port, and the endpoint for health checks. Auto Scaling will use these checks to determine the health of instances.

4. \*Create Load Balancer\*: Navigate to the Elastic Load Balancing service in the AWS Management Console. Create a load balancer, specifying the load balancer type (Application Load Balancer or Classic Load Balancer), listener configuration, availability zones, security settings, etc.

5. \*Configure Target Group\*: If using an Application Load Balancer, create a target group for your Auto Scaling instances. Define the protocol, port, and health check settings for the target group.

6. \*Associate Target Group with Load Balancer\*: Associate the target group you created with the listener on your load balancer.

7. \*Set up Auto Scaling Policies\*: Define scaling policies for your Auto Scaling group based on metrics like CPU utilization, network traffic, etc. Configure scaling actions for scale-out and scale-in events.

8. \*Test the Configuration\*: Deploy your application on the instances launched by the Auto Scaling group. Ensure that the load balancer distributes traffic evenly across instances and that scaling policies trigger appropriately based on workload.

9. \*Monitor and Fine-Tune\*: Regularly monitor the performance of your Auto Scaling group and load balancer. Adjust scaling policies and thresholds as needed to optimize resource utilization and ensure reliability.