AWS Lab 33

Elastic Load Balancer - Application Load Balancer

Overview of the lab

In this lab you will learn how to create a highly available web server running in two availability zones (public subnets) and traffic load balanced with application load balancer.

High Availability

Application access can withstand instance/OS/application failure, because multiple instances are configured to run the same application

LoadBalancer

It is a physical or virtual hardware for load balancing

LoadBalancing

For high availability multiple instances can be configured with same application, so incoming application traffic can be load balanced to different instances

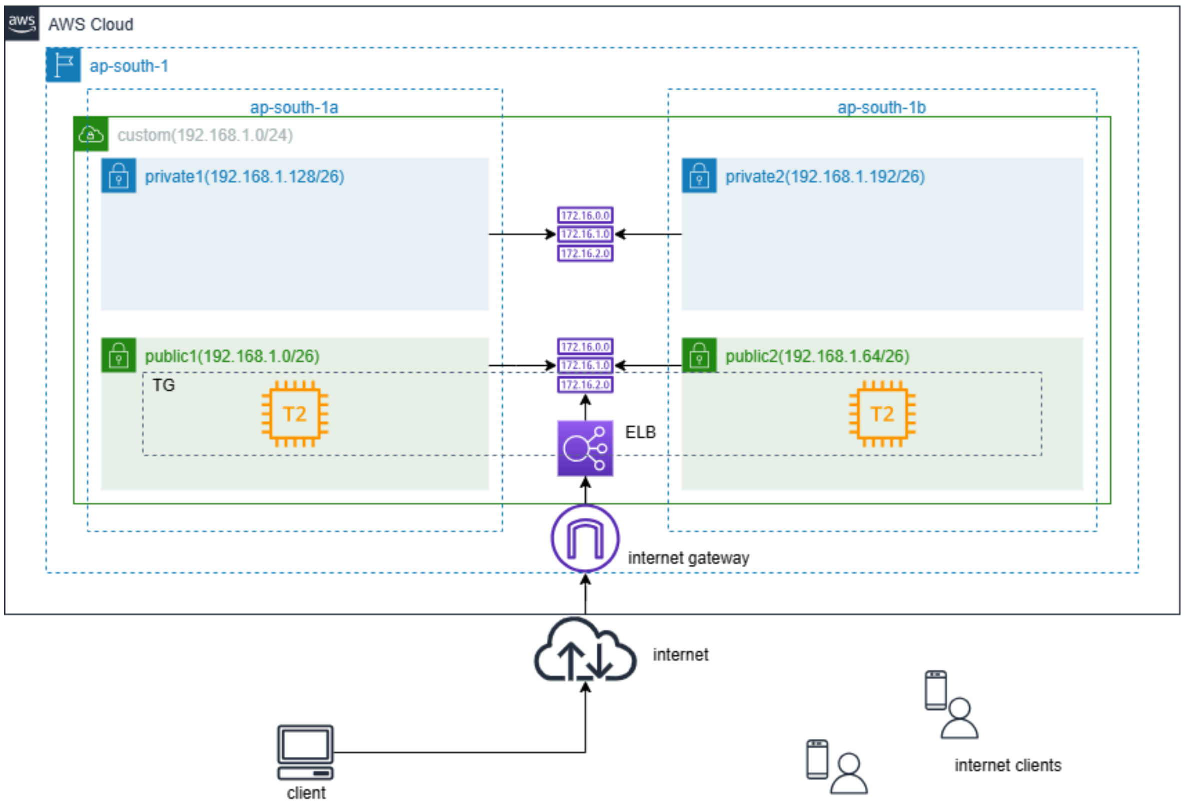
LoadBalancer in AWS

It is a virtual device in aws cloud managed by aws

Target Group

Grouping multiple instances

Architecture



Step by Step Lab

Launch instance(s)

1. In EC2 management console, launch instance
   1.1. Name and tag – linux-webserver1
   1.2. Application and OS Images – RedHat

1.3. Instance type - t2.micro
1.4. Key pair – select the existing keypair
1.5. Edit Network settings
1.5.1. VPC - custom-vpc
1.5.2. Subnet – custom-vpc-public1(ap-south-1a)
1.5.3. Auto-assign public IP - Enable
1.5.4. Firewall - Select existing security group
1.6. In Advanced Details(scroll down to bottom), copy the below bash script in userdata section

```
#!/bin/bash
dnf install httpd -y
systemctl start httpd
systemctl enable httpd
echo $HOSTNAME is running in ap-south-1a > /var/www/html/index.html
```

1.7. Number of instances - 1
1.8. Click on Launch instance

2. Launch instance
2.1. Name and tag – linux-webserver2
2.2. Application and OS Images – RedHat
2.3. Instance type - t2.micro
2.4. Key pair – select the existing keypair
2.5. Edit Network settings
2.5.1. VPC - custom-vpc
2.5.2. Subnet – custom-vpc-public2(ap-south-1b)
2.5.3. Auto-assign public IP - Enable

2.5.4. Firewall - Select existing security group

2.6. In Advanced Details(scroll down to bottom), copy the below bash script in userdata section

```
#!/bin/bash
dnf install httpd -y
systemctl start httpd
systemctl enable httpd
echo $HOSTNAME is running in ap-south-1b > /var/www/html/index.html
```

2.7. Number of instances - 1

2.8. Click on Launch instance

Create Target Group

3. Click on Target Groups and Click on Create target group
4. Basic Configuration
   - 4.1. Choose a target type - Instances
   - 4.2. Target group name - demo-tg1
   - 4.3. Vpc - custom-vpc
5. Health checks
   - 5.1. Click on Advanced health check settings
   - 5.2. Healthy threshold - 2
   - 5.3. Click on Next
6. Register targets
   - 6.1. Select the instances and Click on Include as pending below
7. Click on Create target group

Create ELB - Application Load Balancer

8. Click on Load Balancers and Click on Create load balancer
9. Application Load Balancer - Create
10. Basic configuration
    10.1. Load balancer name - demo-alb
    10.2. Scheme - Internet-facing
11. Network mapping
    11.1. VPC - custom-vpc
    11.2. Mappings
        11.2.1. ap-south-1a - custom-vpc-public1
        11.2.2. ap-south-1b - custom-vpc-public2
12. Security groups - select the existing
13. Listeners and routing
    13.1. Protocol - HTTP
    13.2. Port - 80
    13.3. Default action - select demo-tg1
14. Click on Create load balancer

(Once the load balancer is created accessing the web page using DNS name of the load balancer)

Clean Up Step

1. Select the instances and terminate it
2. Click on Load balancers - Select the demo-alb (load balancer) and in Actions - Click on Delete load balancer and confirm
3. Click on Target groups - select the demo-tg1 (target group) and in Actions - Click on Delete