

Task-7:

Implement load balancing and high availability to opensource application.

This task is in two parts:

1. Where you need to explore the methodologies and possibilities that can be implement in load balancing and creating an application with high availability and document them.
2. The task is to implement one of the explored methodologies in real time and document it.

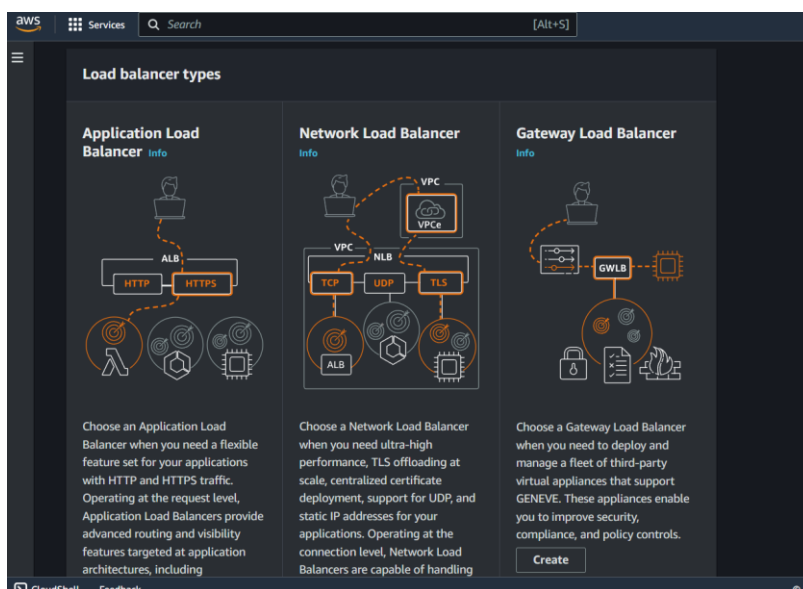
Start:

Part-1

Load balancing methodologies and possibilities

I am using AWS cloud.

AWS provides these types of load balancers.



These are elastic load balancer types, pretty explain these types of load balancers.

- Application load balancer (ALB): operates at the application layer (HTTP/HTTPS) and allows advanced routing based on content and

visibility features targeted at application architectures, microservices and containers.

- Network load balancer (NLB): operates at the transport layers (TCP/UDP) and is optimized for extreme performance. Operating at the connection level, network load balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.
- Gateway load balancer (GWLb): exchange traffic across VPC boundaries securely. GWLB ensure high availability and realibility by routing traffic flows through healthy virtual appliance and rerouting flows when a virtual appliance becomes unhealthy.

High availability zones

- Auto scaling groups: automatically adjust the number of instances in response to changing demand, ensuring the availability of the application. ASG let you launch a fleet composed of on-demand ec2 instance and spot instances.

Load balancing:

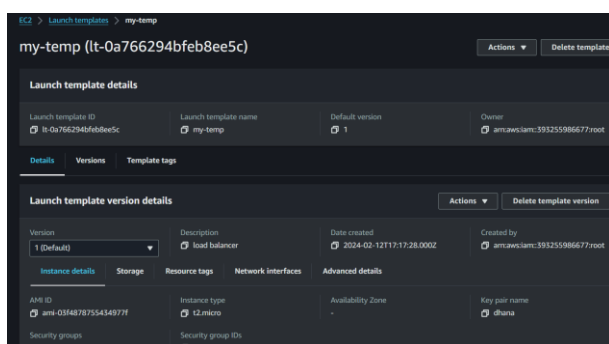
Load balancing is the process of distributing network traffic across multiple servers.

Part-2:

Step-1:

Create launch configuration

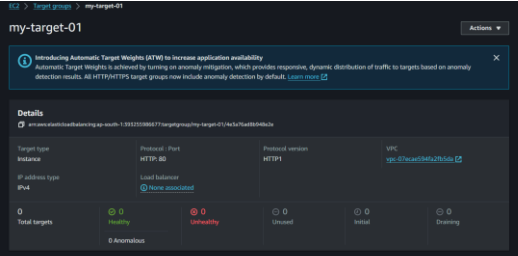
Create instance template



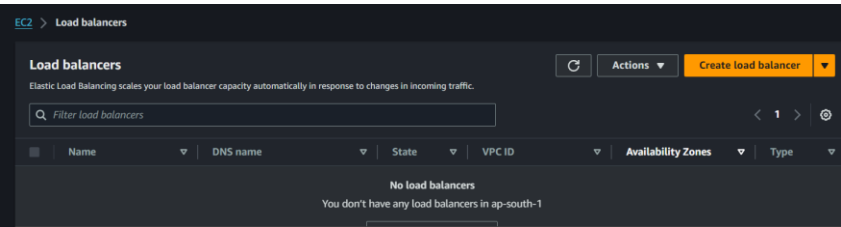
Create load balancer

Step-1:

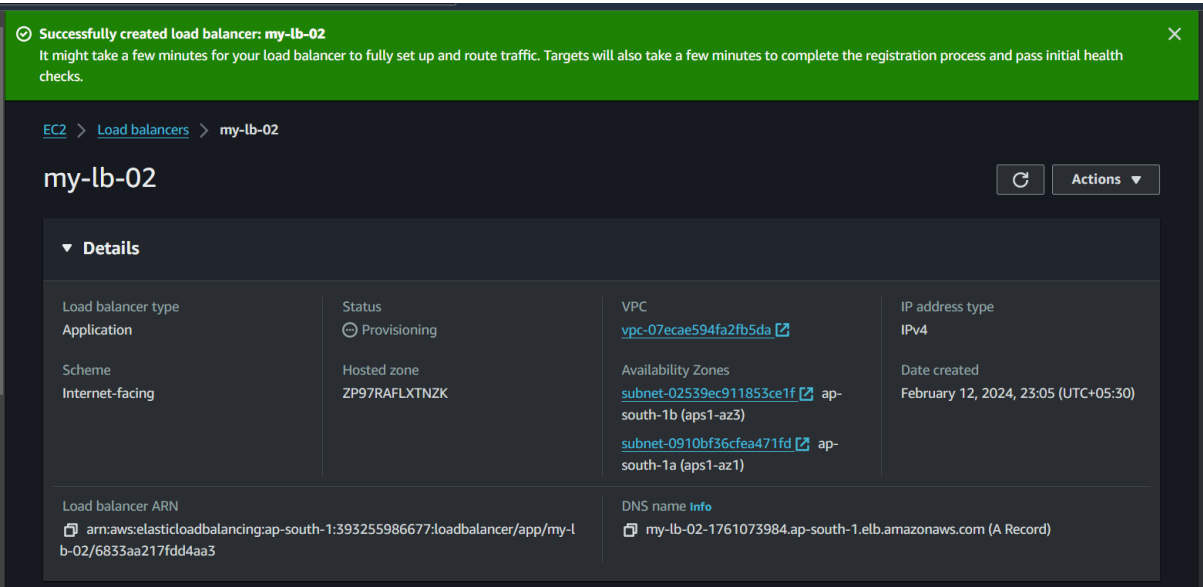
Create target group



Open load balancer, then create load balancer.



I am choosing application load balancer and create load balancer including VPC, security groups, target group details.



Create ASG

Step-1: > choose launch template

The screenshot shows the 'Choose launch template' step of the 'Create Auto Scaling group' wizard. On the left, a sidebar lists steps: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 (optional, Configure advanced options), Step 4 (optional, Configure group size and scaling), Step 5 (optional, Add notifications), Step 6 (optional, Add tags), and Step 7 (Review). The main content area is titled 'Choose launch template' with a sub-header 'Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.' It includes a 'Name' field with the value 'my-asg-lb' and a 'Launch template' section with a blue information box stating: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.'

Step-2: > choose instance launch options

The screenshot shows the 'Instance type requirements' step. It includes an 'Override launch template' button. A table lists the launch template 'my-temp' with version 'Default' and description 'load balancer'. The instance type is 't2.micro'. The 'Network' section shows the VPC 'vpc-07ec2e594fa2fb5da (my-vpc-lb-1)' and a dropdown for 'Availability Zones and subnets'.

Step-3: > configure advanced options

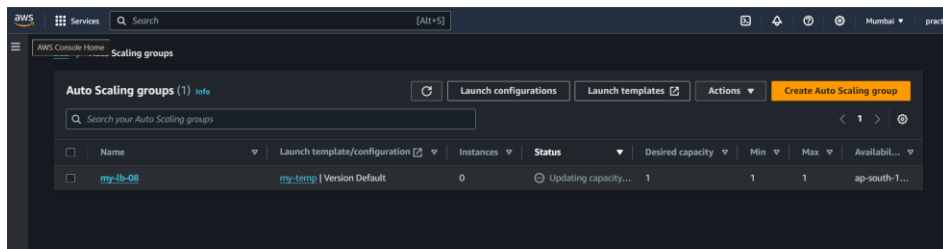
The screenshot shows the 'Configure advanced options' step. It includes a 'Load balancing' section with three radio buttons: 'No load balancer', 'Attach to an existing load balancer' (selected), and 'Attach to a new load balancer'. Below this, the 'Attach to an existing load balancer' section shows a dropdown for 'Existing load balancer target groups'.

Step-4 > configure group size and scaling

The screenshot shows the 'Configure group size and scaling' step. It includes a 'Group size' section with a 'Desired capacity type' dropdown set to 'Units (number of instances)' and a 'Desired capacity' input field set to '1'. The 'Scaling' section has a 'Scaling limits' subsection.

Next steps leaving in the default settings

Now created load balancer



Now automatically launched two ec2 instances.