**AUTO SCALING GROUPS**

Amazon Auto Scaling helps you ensure that you have correct number of Amazon EC2 instances available to handle the load for your application

Whenever there is a dement on traffic Autoscaling will be **SCALE OUT** and **SCALE IN**

**SCALE OUT** : Adding instance based on the traffic

**SCALE IN** : Removing instance based on the traffic

Ensure that we have a minimum and maximum number of machines running Automatically register new instance to the load balancer

**Min** : The minimum number of EC2 instances that the ASG should have

**Max** : The maximum number of EC2 instances that the ASG should have

**Desired** : The number of EC2 instances that you wish/desired to launch initially

(It should be between min and max number)

**Benefits:**

* Better fault tolerance, Amazon EC2 Auto scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone become un-available, Amazon EC2 Auto Scaling can launch instances in another one
* Better Availability, Amazon EC2 Auto scaling can help you ensure that your application always has the right amount of capacity to handle the current traffic demand.
* Better cost management, Amazon EC2 Auto scaling can dynamically increase and decrease capacity as needed. Because you pay for EC2 instances you use, you save money by launching instances when they are actually needed and terminating them, they aren’t needed
* Minimum and maximum of instances running
* Launch or terminate instances to meet desired capacity
* Never start more than maximum number of instances
* Keeps capacity balanced across AZs
* EC2 instances are managed by Auto Scaling Group

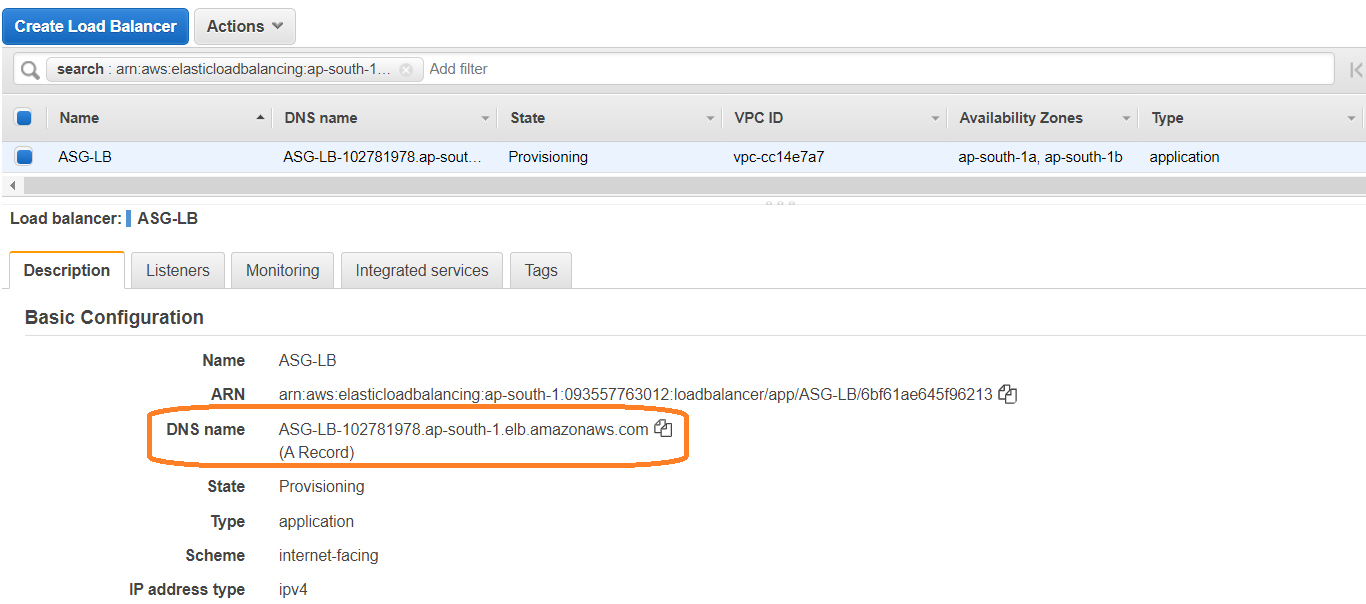
**We have 3 types of scaling option**

* **Manual Scaling** : if you are manually modifying Min / Max / Desired capacity
* **Scheduled Scaling**: Based on the time period of day
* **Dynamic Scaling** : Based on the load (CPU, no of requests , network)

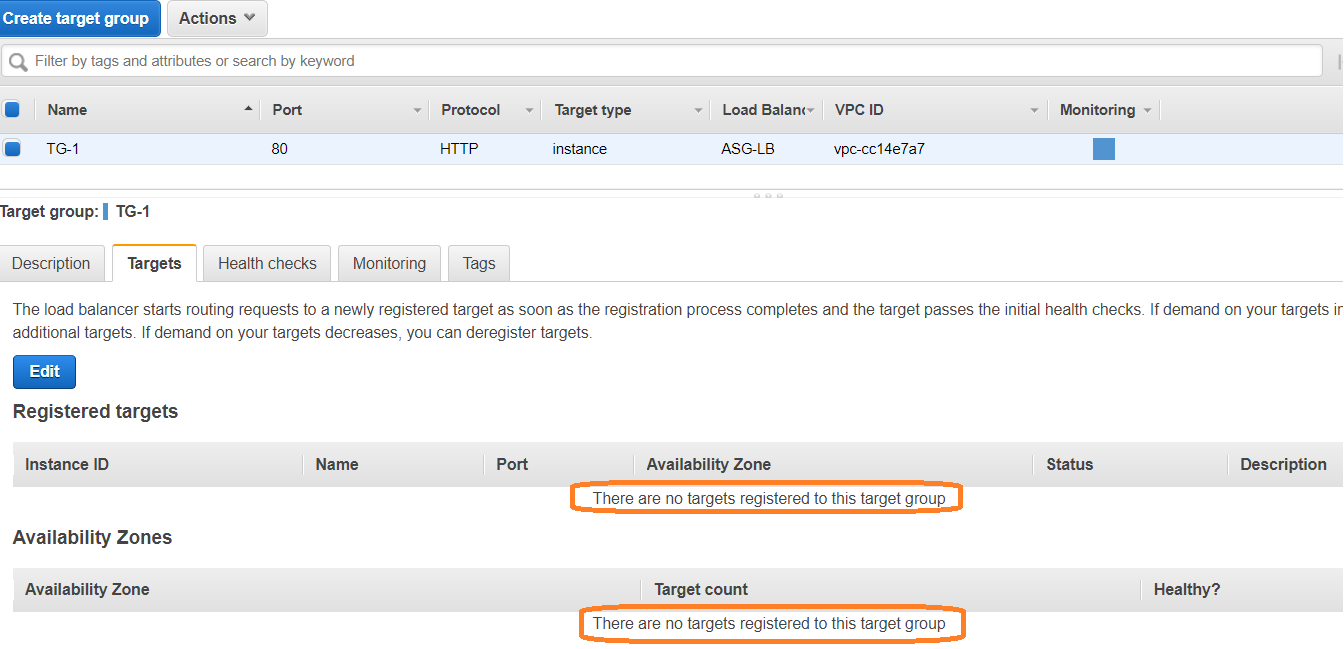
1. **Create ELB with Empty Target Group**

**Keep HTTP in your SG, and make sure you have default rule ALL TRAFFIC with self**

**Load Balancer**

****

**Empty Target Group**

****

**2.** **Create Launch Configuration**

**EC2 🡪 Launch Configurations 🡪 Create Launch Configuration**

* Launch Configuration name : Give any name (ASG-LC)
* Amazon machine image (AMI) : select AMI (ami-06a0b4e3b7eb7a300)

(We can use our own AMI)

* Instance Type : t2. micro (1 vCPUs,1 GiB, EBS only)
* Storage (Volumes) : select default

(If you want, we can add additional volumes)

* Additional Configuration -optional

Purchasing option

Request spot instances

IAM role : We can attach some IAM role

Monitoring

Enable EC2 instance detailed monitoring within CloudWatch

* Advanced details : update user data (bootstrap script)
* Security groups : select default
* Key pair(login) : select key pair

**Bootstrapping**

Installation & setup needs to be fully automated

- use Amazon Machine Image (AMI) with all required configurations & software

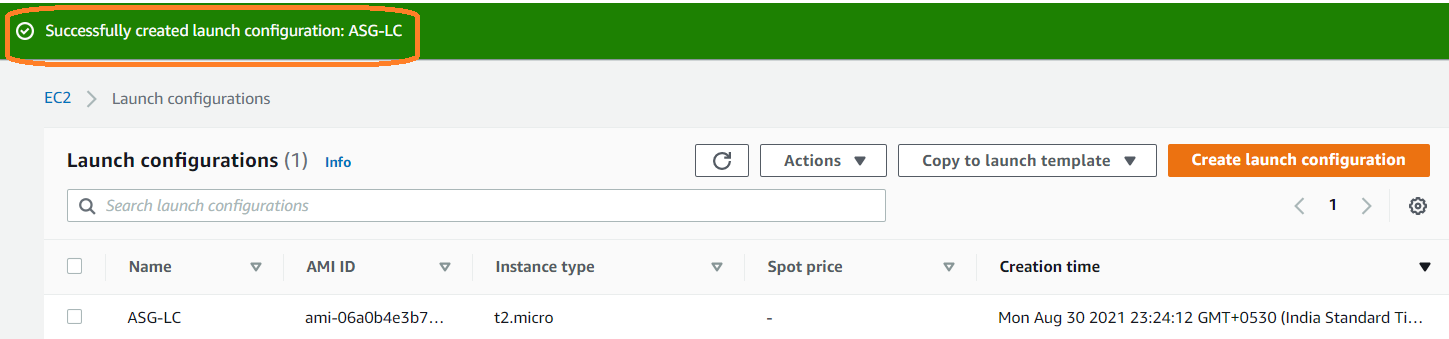
(“Golden Image”)

- Base AMI + install code & configuration as needed

- Via User data

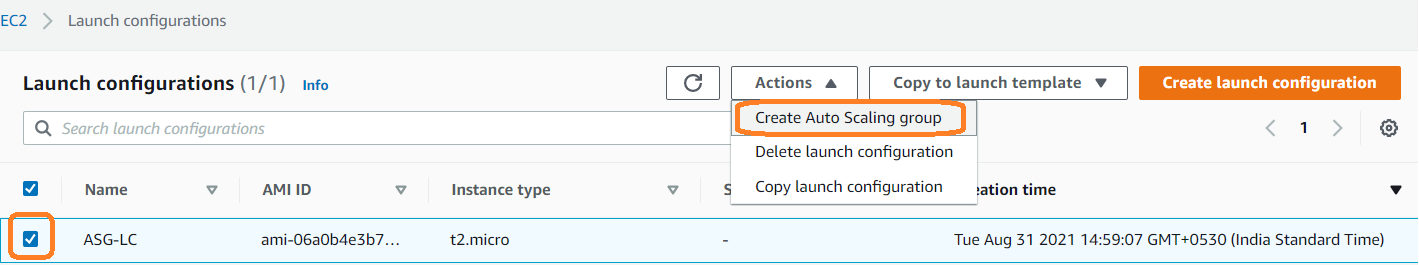
- Via Chef/puppet/Ansible

- Using AWS code deploy



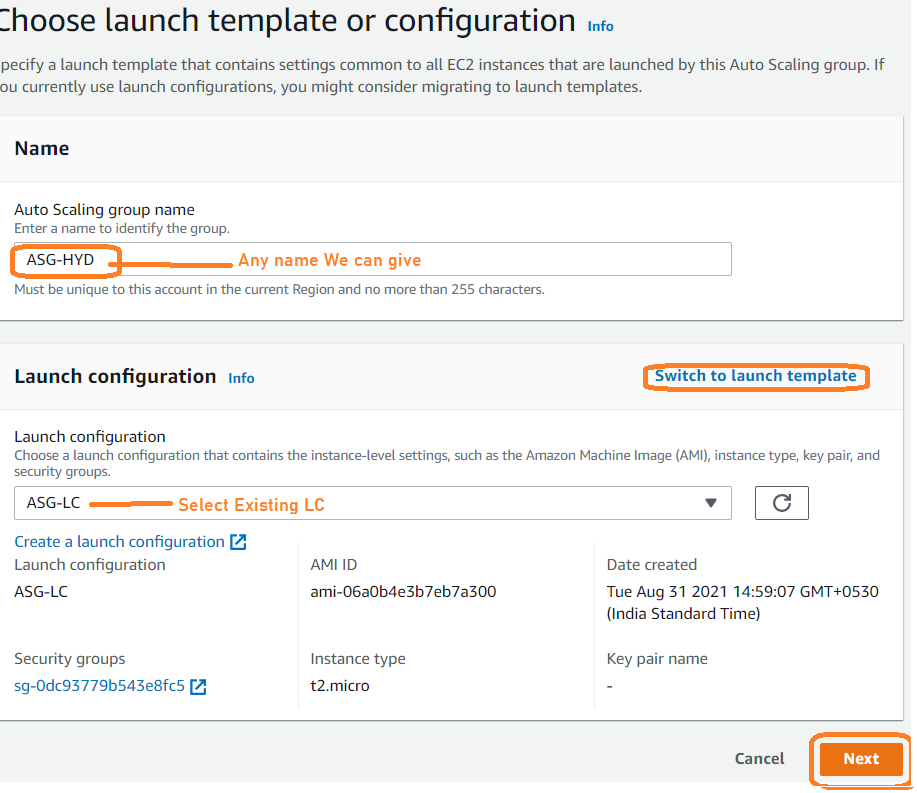
**3.Create ASG from the created Launch configuration**

**EC2 🡪 Launch configuration OR left panel we have option to open Auto Scaling Groups under AUTO SCALING**

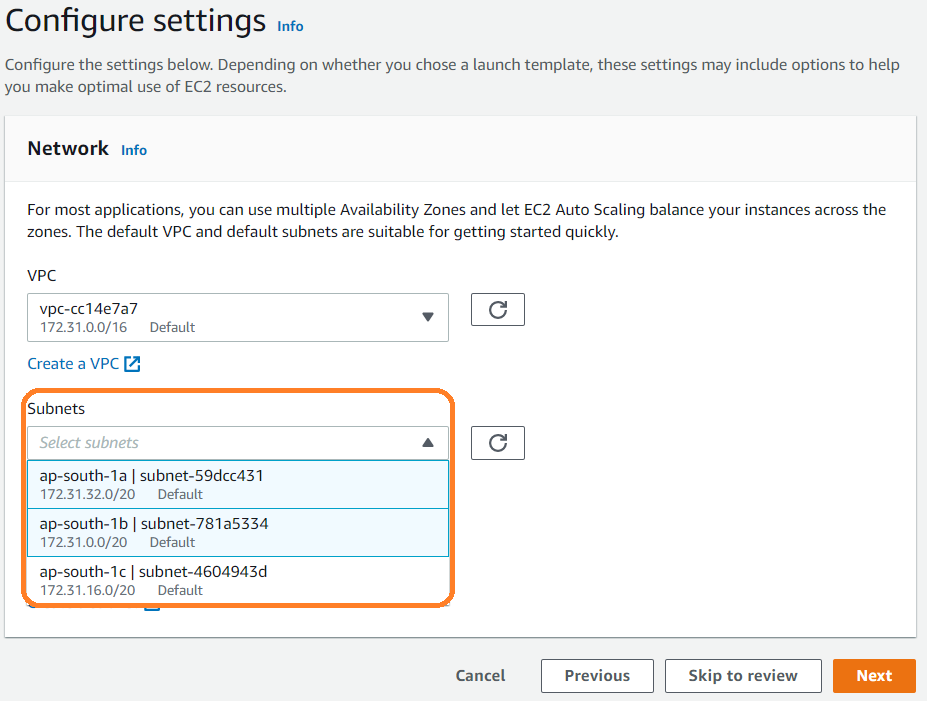
****

We have 7 steps for creating AutoscalingGroup

1.Choose launch template or configuration

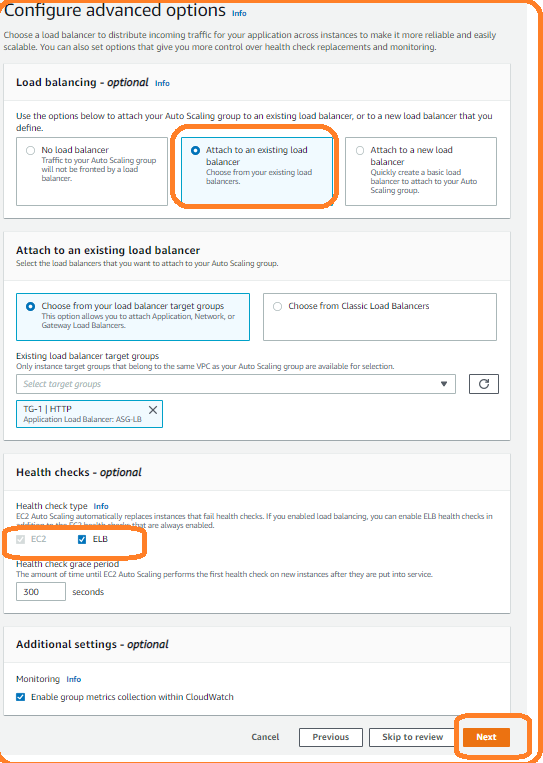


2.Configure settings



We can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones.

3.Configure advanced options



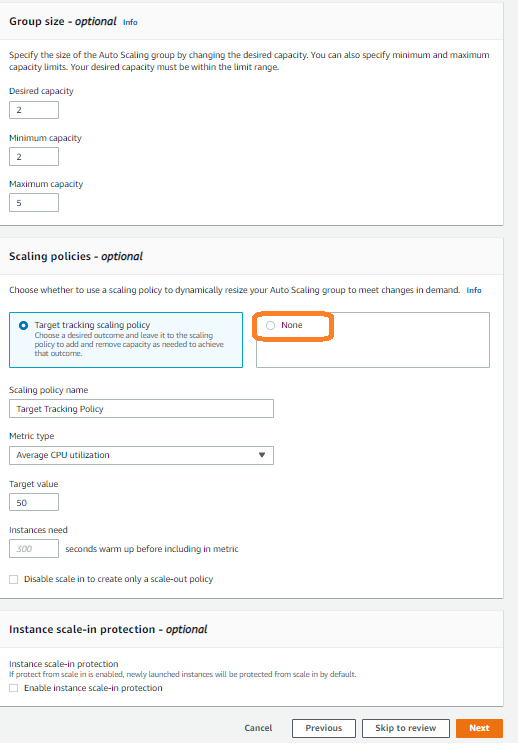
ASG also perform the health checks

by default, it will perform on EC2 instance

**ELB** 🡪 It will perform health check if application is up and running and Tomcat is down it will consider as server down and it will create new instance

Enable group metrics collection within CloudWatch

4.Configure group size and scaling policies (choose **Target Tracking Scaling policy**)



Select min/max/desired capacity

Desired capacity should be min and max value

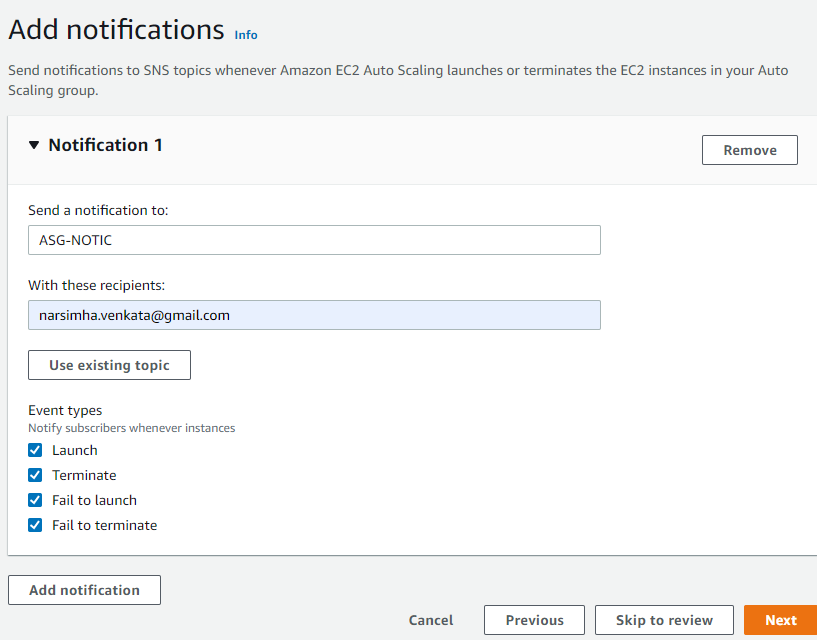
Under scaling polices – optional

If we select **None** 🡪 never scaling

A scaling plan tells Auto Scaling when and how to scale.

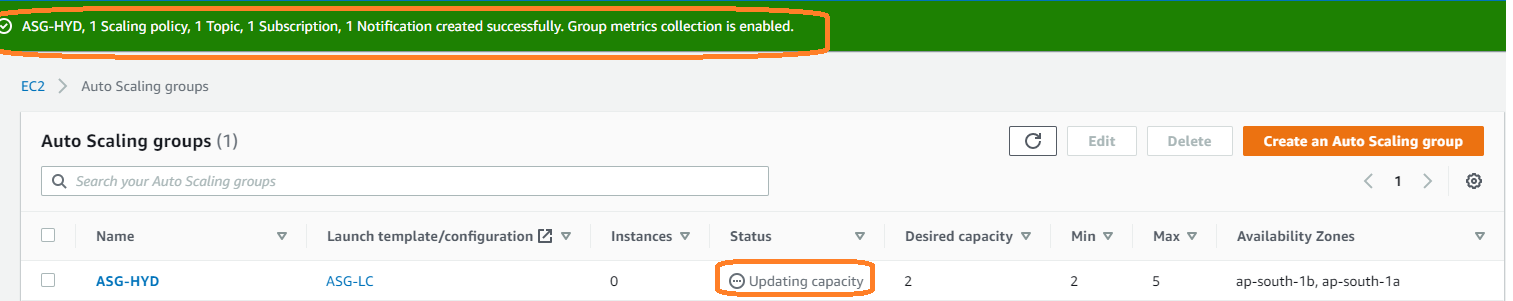
Create a scaling plan based on the occurrence of specified conditions (dynamic scaling) OR create a plan based on a specific schedule.

5.Add notifications



6.Add tags

7.Review



Based on the configuration servers will be updated under instances section and target group section

**Target Group instance status**

1.initial

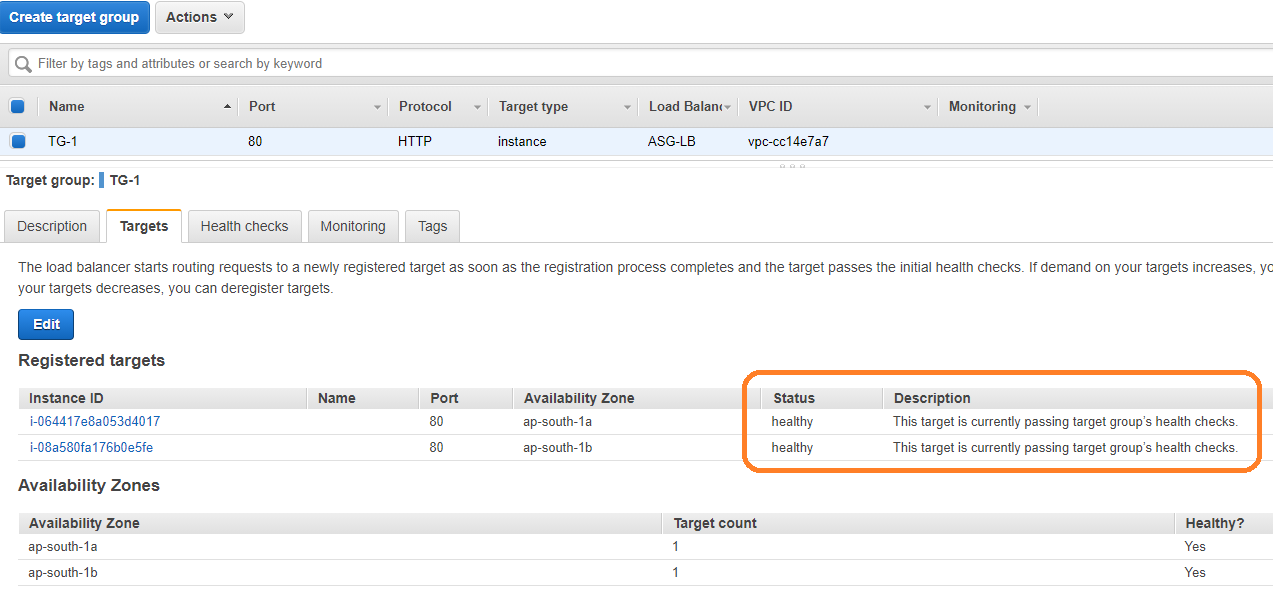
2.healthy

3.unhealthy

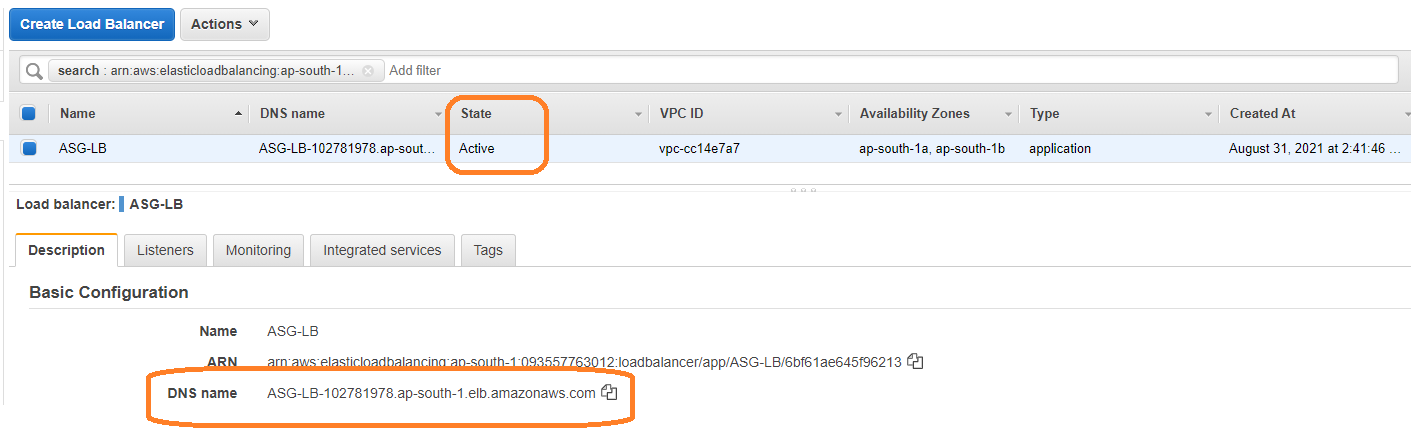
If any instance deregistering under Target Group status will be **draining**

**Target Group**

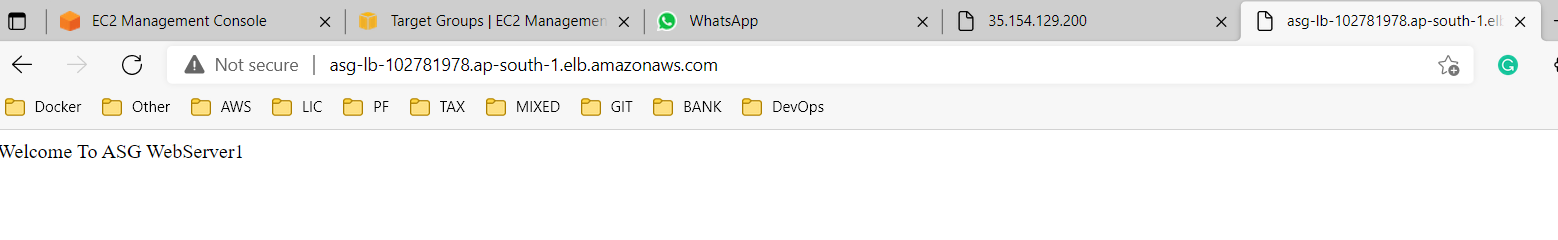
Check your targets in target groups, it should be healthy

****

**Load Balancer**



Access our application using DNS name



**Deleting process**

1) Delete Autoscaling

2) Delete launch configuration (Instances will be terminated automatically)

3) Delete Load balancer

4) Delete Topic

5) Delete Alarm