ASG & ELB

serverless compute service

Auto Scaling Group

Elastic Load Balancer

Auto Scaling Groups

>> Auto Scaling Groups (ASG) automates the process of adding/removing EC2 instances based on demand.

>> It helps ensuring that you have the correct number of EC2 instances available to handle the load of the application

eg: Amazon Great Indian Festival Sale

1. Set up some values:

   - Minimum EC2 instances -> 1
   - Desired EC2 instances -> 3
   - Maximum EC2 instances -> 10

2. Attach CloudWatch Alarm:

   - Avg. Cpu utilization > 70% -> ASG will add 2 instances
   - Avg. Cpu utilization < 30% -> ASG will remove 1 instances

3. Attach CloudWatch Alarm:

   - If any instances crashes, ASG will automatically replace it.

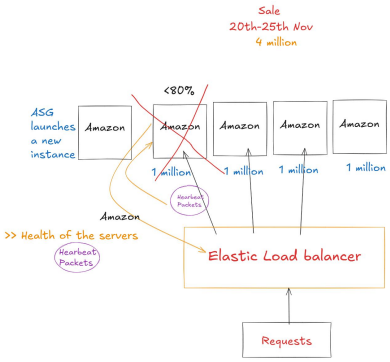
Main goals of ASG:

1. Scalability: Automatically scales up by adding more instances during high traffic and scales down by removing instances when demand decreases.

2. High Availability: It maintains the minimum amount of instances & even if any instance fails, it replaces them with a new one.

3. Cost efficiency: By automatically scaling down during the periods of low usage, an ASG helps us avoid paying extra for the resources.

Elastic Load balancer



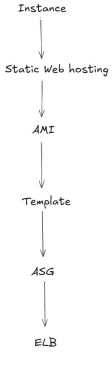
HOW ELB & ASG WORK TOGETHER

>> The true power lies in the integration of ASG & ELB.

>> An ELB distributes traffic to the instances which are managed by ASG.

>> ASG will use metrics from the ELB (like requests per second), to decide when to scale.

>> If the ELB detects that no. of requests is increasing & also CPU utilization of the instances are rising, so ASG will automatically launch the new instances. -> Once new instances are available & ready, they automatically get registered with the ELB, which starts distributing the traffic to them as well.



1) Create an instance while creating the instance, add security groups.
2) Under Inbound Security Group Rules click on add security group rule.
3) Under type select HTTP, SSH and all traffic and under source type Anywhere for all three types.
4) Create an Image of the instance.
5) After creating the Image, in the sidebar under instances select launch templetes and click on create launch templete.
6) Give templete name and templete version and tick on "Auto Scaling guidance" Provide guidance to help me set up a template that I can use with EC2 Auto Scaling.
7) Under "Application and OS Images (Amazon Machine Image)" select my AMIs and select instance type and key pair and after that click on create launch templete.
8) After that go to the sidebar, scroll down to the end and under "Auto scaling" click on auto scaling groups and click on create auto scaling group.
9) Give "Auto Scaling group name" and choose the template under launch template and click on next.
10) Choose atleast two AZ and choose balanced best effort option and click on next.
11) Under "load balancing" option select attach to a new load balancer option.
12) Under attach to a new load balancer option select application load balancer and give a name to the load balancer and select internet facing option.
13) Under default routing select the search bar and click on create a target group.
14) Under health checks tick on "Turn on elastic load balancing health checks" then click on next.
15) Select desired capacity and also select minimum desired capacity and maximum desired capacity.
16) Click on target tracking scaling policy and under that select the metric type and click on next.
17) Ignore the 5th step and click on next.
18) For the 6th step, click on add tag and under key write "Name" and under value write the name of the instance which you want to give for the asg instance which you are creating. Then click on next.
19) Then comes the summary part. Click on "Create auto scaling groups".
20) If you go to the instance and refresh you will get asg instance runnning. Copy the IPv4 address and paste it in the search bar. It should load the static web page. If it is not loading check the security group and create SSH, HTTP and all traffic. If the given limit is reached, it should create a new instance.