AUTO SCALING

* ASG automatically scales up and down the instances based on load of your application.
* Based on your configuration, when load went up to specific percentage, it will automatically scales up the specific no of instances as we mentioned and if load went down to specific percentage, it will scales down the instances as per your requirement.
* You can add multiple az for scalability. If anyone AZ is unavailable, auto scaling will launch instances in another unaffected AZ.

**Launch configuration per region = 100. ASG per region = 20.**

**Scaling policies per ASG = 50.**

CREATE A LAUNCH CONFIGURATION

* In launch config, you can create instances to add to your auto scaling groups.
* Go to ec2.

**Click launch configuration.**

**Click create.**

**Select AMI.**

**Type a name to your launch configuration.**

**key-pairs, security groups, assign public ip.**

**Click Create launch configuration.**

* If you already have running instances that you want to add to ASG. No need to create a launch configuration.

**Select your instance Click Actions. Instance settings.**

**select Attach to auto scaling group.**

**Choose ASG** (or) **create a new ASG option**

**If you select new ASG, type a name and attach.**

**Launch configuration also create with same name as ASG.**

**It will create ASG with default settings.**

**To change, go to ASG, click edit**.

* You can't edit the launch configuration once you saved it.
* To edit launch configuration, you have to copy the current launch configuration and edit the copied one.
* After editing the launch configuration, add this updated launch config to ASG by clicking edit and select new launch config and save it.
* When an new instance is launched within auto scaling, it will launch based on the ami we selected in launch configuration.
* If your want your newly launched instances to be updated with your latest application revision, you have to edit the launch configuration, add your latest AMI and save it.

CREATE AN AUTO SCALING GROUP

* An auto scaling group contains collection of ec2 instances treated as logical group for automated scaling.
* It will scale up and down the instances based on load of your application. You can create scaling policies to scale number of instances in ASG.
* To create an ASG,

**Go to ec2.**

**Select auto scaling on left pane.**

**Click Create an auto scaling group.**

**If you don't have launch config, first it will create a launch config and it creates ASG. If you have launch config, it will directly go to ASG creation page.**

**Type group name.**

**Select how many instances to start after the creation of ASG. Select VPC.**

**Select Subnets(az).**

**If you select two subnets, if one subnet went down, asg will launch instances in another subnet. it is useful for scalability.**

* Click, Advanced Details,

**Select Receive traffic from load balancers. Select your load balancer.**

**Health check type = elb, ec2(your choice).**

* In Configuring scaling policies page,

**Select keep this as initial size (default policies) (or) Select use scaling policies (Manually).**

**Select min and max instances for the group size.**

* In Increased Group size,

**Type a name.**

**Click create a alarm.**

**In alarm page, Choose create topic.**

**Type a name for sns topic and emails to send notifications when the scaling happens.**

**Give the metrics, when the CPU % is greater than 80 for at least 1 for 5 minutes.**

**In take action, type your desired number to add instances when the CPU is reached you desired capacity.**

* In Decreased Group size,

**Type a name.**

**Click Create alarm.**

**In alarm page, Choose Create topic.**

**Select your desired load, when to reduce instances from group. And how many instances to reduce.**

**In take action, type a number to reduce when the load is decreased.**

* On review page, check all your settings and Click Create.
* yes > /dev/null & to increase cpu load

INSTANCE PROTECTION

* By default aws terminates ec2 instances while the load is reduced. But we are not sure which instance it will terminate. For this, we use instance protection.
* After we set protection for an instance, whenever the load is reduced, it won't terminate that protected instance, it will choose other instances to terminate.
* If you set instance protection for all instance, asg won't terminate any instances and scaling policies doesn't work.
* There are certain termination policies for the ec2 instances in ASG like

**Instance which is near to next billing hour.**

**Instance which is old.**

**Instance which is new.**

* To set protection, Go to EC2,

**Select instance (to protect from termination).**

**Click Actions.**

**Choose instance settings.**

**Click Change termination protection.**

**Choose enable.**

**LIFE CYCLE HOOKS:**

* Lifecycle hooks enables you to perform custom actions by pausing instances as an asg launch and terminates them. The instances will be in that desired state for a time period where you can perform actions like for ex: when an newly launched instance is paused, you can install and configure software in it.
* There is a limit for lifecycle hooks per asg max **50**.
* Lifecycle hook puts instances in waitstate (pending:watr or terminating:wait). The instances will be in that state until you continue (or) time period ends.
* By default, the instance will be in wait state for 1 hour and max is 2 hrs. if you need more time, you can reset the time period by recording a heartbeat.
* **Cooldownperiod =** the number of seconds to wait after an action completes, before starting a new action.
* If you add a health check period, it won’t start until the lifecycle action is completed.
* The result of a lifecycle hook will be **ABONDON** (or) **CONTINUE**.
* If you **launching** an instance, **continue** indicates that your actions were successful, put the instance in service state.
* If it indicates **abondon**, you actions were unsuccessful and the instance will be terminated.
* If you are **terminating** an instance, both **abondon** and **continue** will allow instance to terminate.
* You can find aws lifecycle hooks in aws console, create a hook and enter metadata in notification metadata secton.