AWS Auto Scaling

Overview:

- Auto Scaling provides the ability to ensure a correct number of EC2 instances are always running to handle the load of the application
- Auto Scaling helps
  - to achieve better fault tolerance, better availability and cost management.
  - helps specify scaling policies that can be used to launch and terminate EC2 instances to handle any increase or decrease in demand.
- Auto Scaling attempts to distribute instances evenly between the AZs that are enabled for the Auto Scaling group.
- Auto Scaling does this by attempting to launch new instances in the AZ with the fewest instances. If the attempt fails, it attempts to launch the instances in another AZ until it succeeds
- Auto Scaling ensures a steady minimum (or desired if specified) count of Instances will always be running.
- If an instance is found unhealthy, Auto Scaling will terminate the Instance and launch a new one.
- ASG determines the health state of each instance by periodically checking the results of EC2 instance status checks.
- ASG can be associated with an Elastic load balancer enabled to use the Elastic Load Balancing health check, Auto Scaling determines the health status of the instances by checking the results of both EC2 instance status and Elastic Load Balancing instance health.
- Auto Scaling marks an instance unhealthy and launches a replacement if
  - the instance is in a state other than *running*,
  - the system status is *impaired*, or
  - Elastic Load Balancing reports the instance state as *OutOfService*.
  -

Auto Scaling Components:

Auto Scaling Groups – ASG

- ASG requires
  - **Launch configuration** OR **Launch Template**
    - determine the EC2 template to use for launching the instance
  - **Minimum & Maximum capacity**

    - determine the number of instances when an autoscaling policy is applied.
    - Number of instances cannot grow beyond these boundaries
  - **Desired capacity**
    - to determine the number of instances the ASG must maintain at all times. If missing, it equals the minimum size.
    - Desired capacity is different from minimum capacity.
    - An Auto Scaling group's desired capacity is the default number of instances that should be running. A group's minimum capacity is the fewest number of instances the group can have running
  - **Availability Zones or Subnets** in which the instances will be launched.
  - **Metrics & Health Checks**
    - metrics to determine when it should launch or terminate instances and health checks to determine if the instance is healthy or not
- ASG starts by launching a desired capacity of instances and maintains this number by performing periodic health checks.
- If an instance becomes unhealthy, the ASG terminates and launches a new instance.
- ASG can also use scaling policies to increase or decrease the number of instances automatically to meet changing demands
- An ASG can contain EC2 instances in one or more AZs within the same region.
- ASGs cannot span multiple regions.
- ASG can launch **On-Demand Instances, Spot Instances, or both** when configured to use a launch template.
- ASG can be associated with a single launch configuration or template
- When the launch configuration for the ASG is changed, any new instances launched, use the new configuration parameters, but the existing instances are not affected.

Launch Template

- Launch Template allows **multiple versions of a template** to be defined.
- With versioning, a subset of the full set of parameters can be created and then reused to create other templates or template versions *for e.g, a default template that defines common configuration parameters can be created and allow the other parameters to be specified as part of another version of the same template*.

AWS Auto Scaling Policies

- EC2 Auto Scaling Policies provide several ways for scaling the Auto Scaling group.
  - Manual Scaling

- Scheduled Scaling
- Dynamic Scaling
- Predictive Scaling

Scheduled Scaling

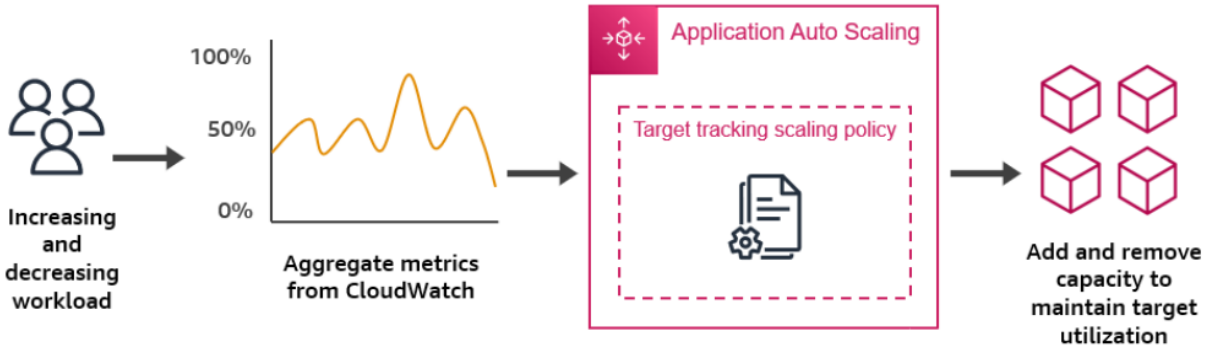
- Scaling based on a schedule allows you to scale the application in response to **predictable load changes** *for e.g. last day of the month, the last day of a financial year.*
- Scheduled scaling requires the configuration of Scheduled actions, which tells Auto Scaling to perform a scaling action at a certain time in the future, with the start time at which the scaling action should take effect, and the new minimum, maximum, and desired size of group should have.
- Auto Scaling guarantees the order of execution for scheduled actions within the same group, but not for scheduled actions across groups.
- Multiple Scheduled Actions can be specified but should have **unique** time values and they **cannot** have overlapping times scheduled which will lead to their rejection.

Dynamic Scaling

- Allows automatic scaling in response to the changing demand *for e.g. scale-out in case CPU utilization of the instance goes above 70% and scale in when the CPU utilization goes below 30%*
- ASG uses a combination of **alarms & policies** to determine when the conditions for scaling are met.

Target tracking scaling

- Increase or decrease the current capacity of the group based on a target value for a specific metric.



Step scaling

- Increase or decrease the current capacity of the group based on a set of scaling adjustments, known as *step adjustments,* that vary based on the size of the alarm breach.

Simple Scaling

Multiple Policies

- ASG can have more than one scaling policy attached at any given time.
- Each ASG would have at least two policies: one to scale the architecture out and another to scale the architecture in.
- If an ASG has multiple policies, there is always a chance that both policies can instruct the Auto Scaling to Scale Out or Scale In at the same time.
- When these situations occur, Auto Scaling chooses the policy that has the **greatest impact** i.e. **provides the largest capacity for both scale out and scale in** on the ASG *for e.g. if two policies are triggered at the same time and Policy 1 instructs to scale out the instance by 1 while Policy 2 instructs to scale out the instances by 2, Auto Scaling will use the Policy 2 and scale out the instances by 2 as it has a greater impact*.

Predictive Scaling

- Predictive scaling can be used to increase the number of EC2 instances in the ASG in advance of daily and weekly patterns in traffic flows.
- Predictive scaling is well suited for situations where you have:
  - Cyclical traffic, such as high use of resources during regular business hours and low use of resources during evenings and weekends
  - Recurring on-and-off workload patterns, such as batch processing, testing, or periodic data analysis
  - Applications that take a long time to initialize, causing a noticeable latency impact on application performance during scale-out events
- Predictive scaling provides proactive scaling that can help scale faster by launching capacity in advance of forecasted load, compared to using only dynamic scaling, which is reactive in nature.
- Predictive scaling uses machine learning to predict capacity requirements based on historical data from CloudWatch. The machine learning algorithm consumes the available historical data and calculates the capacity that best fits the historical load pattern, and then continuously learns based on new data to make future forecasts more accurate.

- Predictive scaling supports **forecast only** mode so that you can evaluate the forecast before you allow predictive scaling to actively scale capacity
- When you are ready to start scaling with predictive scaling, switch the policy from **forecast only** mode to **forecast and scale** mode.

Auto Scaling Cooldown Period

- Auto Scaling Cooldown period is a configurable setting for the ASG that helps to ensure that Auto Scaling doesn't launch or terminate additional instances before the previous scaling activity takes effect and allows the newly launched instances to start handling traffic and reduce load
- When ASG dynamically scales using a simple scaling policy and launches an instance, Auto Scaling suspends the scaling activities for the cooldown period (default 300 seconds) to complete before resuming scaling activities
- Note that if an instance becomes unhealthy, Auto Scaling does not wait for the cooldown period to complete before replacing the unhealthy instance.
- Cooldown periods are automatically applied to dynamic scaling activities for simple scaling policies and are not supported for step scaling policies.

Auto Scaling Termination Policy

- Termination policy helps Auto Scaling decide which instances it should terminate first when Auto Scaling automatically scales in.

Instance Refresh

- Instance refresh can be used to update the instances in the ASG instead of manually replacing instances a few at a time.
- An instance refresh can be helpful when you have a new AMI or a new user data script.
- Instance refresh also helps configure the minimum healthy percentage, instance warmup, and checkpoints.
- To use an instance refresh
  - Create a new launch template that specifies the new AMI or user data script.
  - Start an instance refresh to begin updating the instances in the group immediately.
  - EC2 Auto Scaling starts performing a rolling replacement of the instances.

Instance Protection

- Instance protection controls whether Auto Scaling can terminate a particular instance or not.
- Instance protection can be enabled on an ASG or an individual instance as well, at any time

Standby State

Auto Scaling allows putting the InService instances in the Standby state during which the instance is still a part of the ASG but does not serve any requests. This can be used to either troubleshoot an instance or update an instance and return the instance back to service.

- An instance can be put into Standby state and it will continue to remain in the Standby state unless exited.
- Auto Scaling, by default, decrements the desired capacity for the group and prevents it from launching a new instance. If no decrement is selected, it would launch a new instance

Auto Scaling Lifecycle Transition