

Auto Scaling group

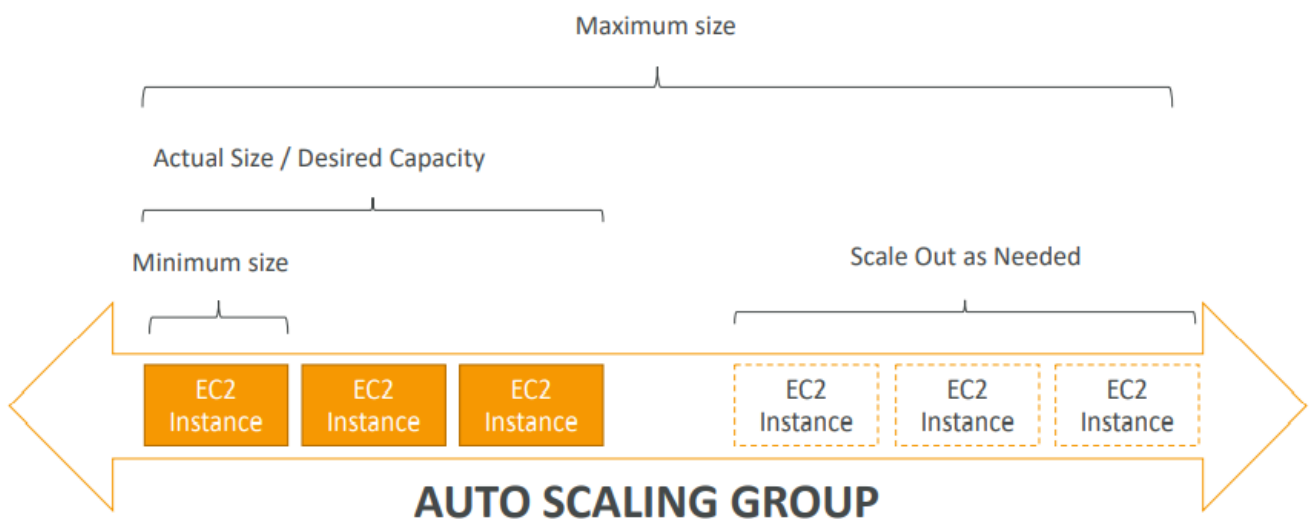
09:37 PM



What's an Auto Scaling Group?

- In real-life, the load on your websites and application can change
- In the cloud, you can create and get rid of servers very quickly
- The goal of an Auto Scaling Group (ASG) is to:
 - Scale out (add EC2 instances) to match an increased load
 - Scale in (remove EC2 instances) to match a decreased load
 - Ensure we have a minimum and a maximum number of machines running
 - Automatically Register new instances to a load balancer

Auto Scaling Group in AWS



ASGs have the following attributes

- A launch configuration
 - AMI + Instance Type
 - EC2 User Data
 - EBS Volumes
 - Security Groups
 - SSH Key Pair
- Min Size / Max Size / Initial Capacity
- Network + Subnets Information

- Load Balancer Information
 - Scaling Policies
-

Auto Scaling New Rules

- It is now possible to define "better" **auto** scaling rules that are directly managed by EC2
 - Target Average CPU Usage
 - Number of requests on the ELB per instance
 - Average Network In
 - Average Network Out
- These rules are easier to set up and can make more sense

Auto Scaling Custom Metric

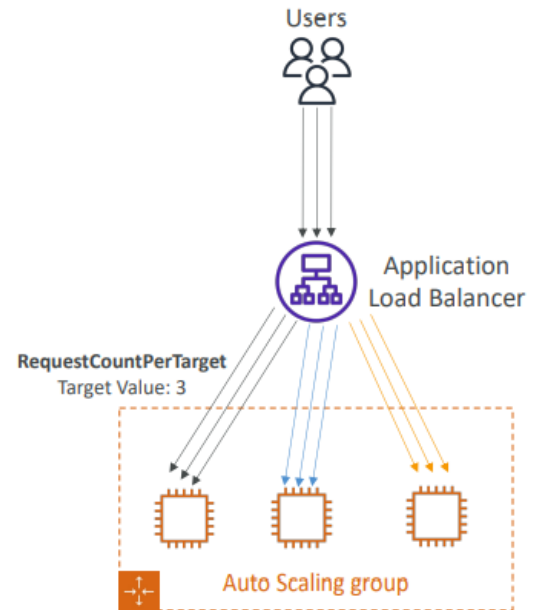
- We can auto scale based on a custom metric (ex: number of connected users)
- 1. Send custom metric from application on EC2 to CloudWatch (PutMetric API)
- 2. Create CloudWatch alarm to react to low / high values
- 3. Use the CloudWatch alarm as the scaling policy for ASG

Auto Scaling Groups – Dynamic Scaling Policies

- **Target Tracking Scaling**
 - Most simple and easy to set-up
 - Example: I want the average ASG CPU to stay at around 40%
- **Simple / Step Scaling**
 - When a CloudWatch alarm is triggered (example CPU > 70%), then add 2 units
 - When a CloudWatch alarm is triggered (example CPU < 30%), then remove 1
- **Scheduled Actions**
 - Anticipate a scaling based on known usage patterns
 - Example: increase the min capacity to 10 at 5 pm on Fridays

Good metrics to scale on

- **CPUUtilization:** Average CPU utilization across your instances
- **RequestCountPerTarget:** to make sure the number of requests per EC2 instances is stable
- **Average Network In / Out** (if you're application is network bound)
- **Any custom metric** (that you push using CloudWatch)



Auto Scaling Groups - Scaling Cooldowns

- After a scaling activity happens, you are in the **cooldown period (default 300 seconds)**
- During the cooldown period, the ASG will not launch or terminate additional instances (to allow for metrics to stabilize)
- Advice: Use a ready-to-use AMI to reduce configuration time in order to be serving request faster and reduce the cooldown period

