AWS ELB and AutoScaling

CS516 – Cloud Computing

Computer Science Department

Maharishi International University

Maharishi International University - Fairfield, Iowa

Content

- Elastic Load Balancer (ELB)
  - Listener
  - Listener rule
  - Target groups
- Types of ELB (OSI Model)
  - Application Load Balancer (ALB)
  - Network Load Balancer (NLB)
  - Classic Load Balancer
  - Gateway Load Balancer
- Auto Scaling (with CloudWatch)
  - Target Tracking Scaling policy

Elastic Load Balancer (ELB)

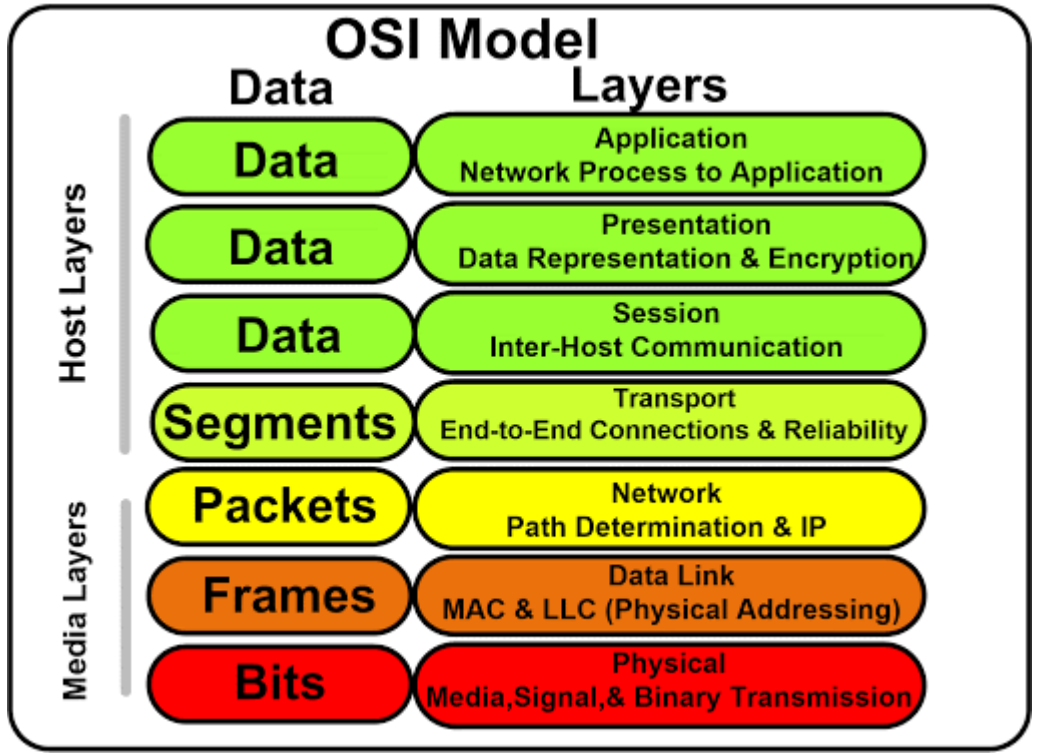
ELB automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables **fault-tolerance** in applications, seamlessly providing the required amount of load-balancing capacity needed to route application traffic.

A tool that distributes incoming web traffic (visitors to a web site) and **equally across multiple EC2** instances that are running a web site.

Helps prevent one server from being overloaded while another server can handle more visitors.

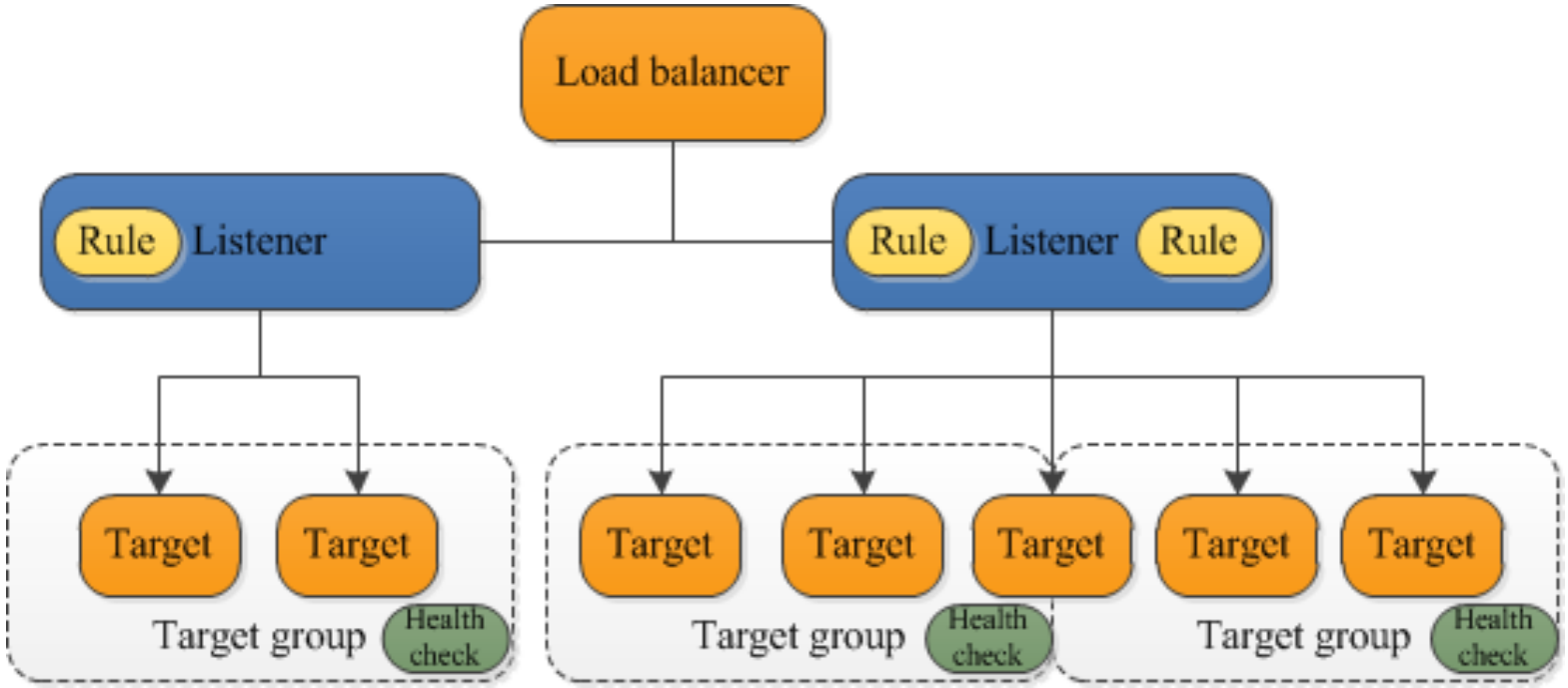
Types of ELB

1. **Classic Load Balancer** (CLB) – Old generation, not recommended for new apps. Performs routing at Layer 4 and Layer 7.
2. **Network Load Balancer** (NLB) – Routes connections based on IP protocol data (layer 4). Ultra high performance and low latency. Supports UDP and static IP addresses as targets.
3. **Application Load Balancer** (ALB) – Routes based on the content of the request (layer 7). Supports path-based, host-based, query string, parameter-based, and source IP based routing. Supports IP addresses, Lambda, containers as targets.
4. **Gateway Load Balancer** – Operates at Layer 3 (Network). Gateway Load Balancers enable you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems.



Application Load Balancer (ALB)

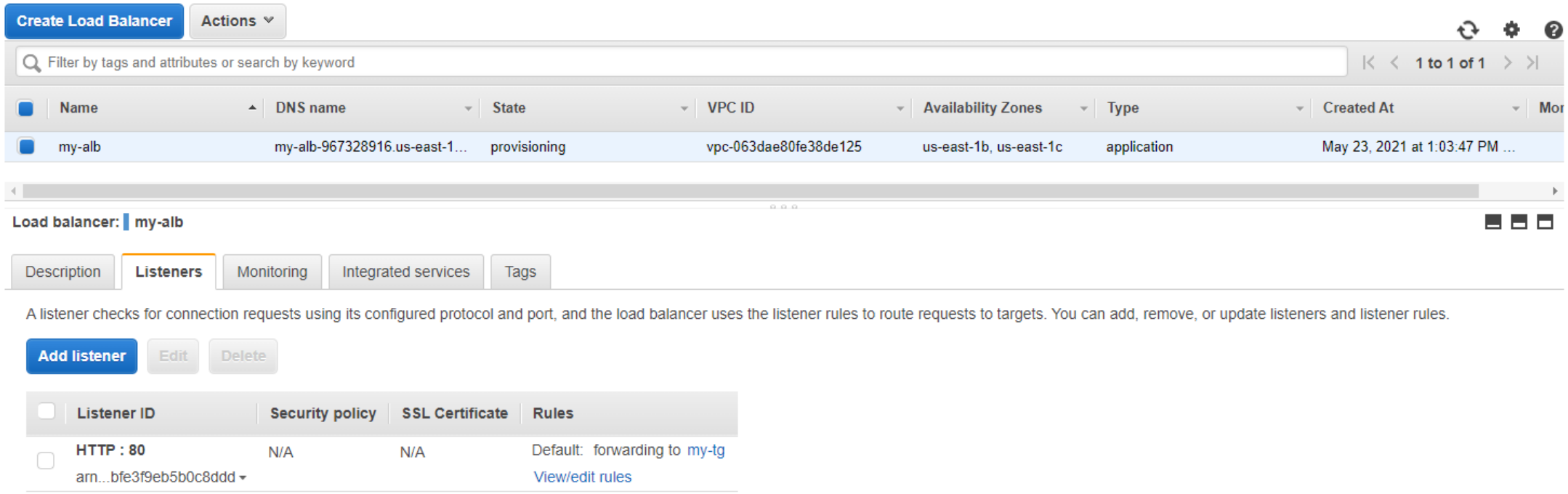
A load balancer serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances in multiple Availability Zones. This increases the availability of your application. You add one or more listeners to your load balancer.



ALB Listener

A listener checks for connection requests from clients, using the **protocol** and **port** that you configure.

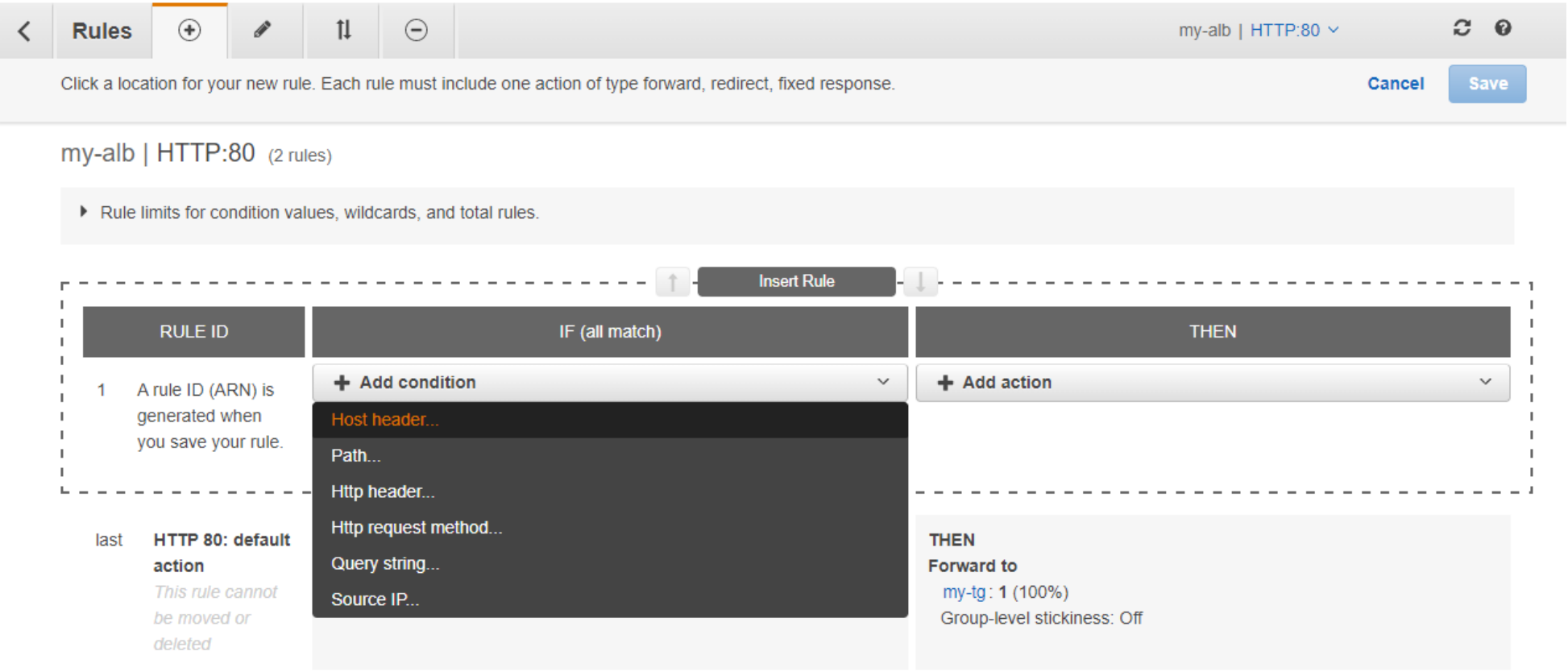
A **certificate** is attached to the listener. You must define a default certificate if using https. AWS and custom certificates are stored on Amazon Certificate Manager (ACM). AWS certificates are free!



ALB Listener Rules

The **rules** that you define for a listener determine how the load balancer routes requests to its **registered targets**.

Each rule consists of a **priority, actions, conditions**. When the conditions for a rule are met, then its actions are performed. You must define a default rule for each listener, and you can optionally define additional rules.



ALB Target Groups

Each target group routes requests to one or more registered targets, such as EC2 instances, using the protocol and port number that you specify.

You can configure health checks on a per target group basis. Health checks are performed on all targets registered to a target group.

Note: When you associate a resource with an ALB, you associate it with its target group. When creating an ALB, target group can have no resources. But the important thing is you must specify the right target group type (ip, instance, lambda).

ALB Target Groups – Why is an instance unhealthy?

EC2 > Target groups > my-tg

my-tg

Delete

arn:aws:elasticloadbalancing:us-east-1:637278349805:targetgroup/my-tg/1a341e1a092d8a30

Details

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-063dae80fe38de125
Load balancer			
my-alb			

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
1	0	1	0	0	0

Targets | Monitoring | Health checks | Attributes | Tags

Health check settings

Edit

Protocol	Path	Port	Healthy threshold
HTTP	/	Traffic port	2 consecutive health check successes
Unhealthy threshold	**Timeout**	**Interval**	**Success codes**
2 consecutive health check failures	120 seconds	300 seconds	200

Instance’s SG only allowed the developer’s IP address

EC2 > Security Groups > sg-0bde078b4803140da - allow-my-http-ssh

sg-0bde078b4803140da - allow-my-http-ssh

Actions

Details

Security group name	Security group ID	Description	VPC ID
allow-my-http-ssh	sg-0bde078b4803140da	launch-wizard-1 created 2021-05-22T12:26:31.891-05:00	vpc-063dae80fe38de125

Owner	Inbound rules count	Outbound rules count
637278349805	2 Permission entries	1 Permission entry

Inbound rules | Outbound rules | Tags

Inbound rules (2)

Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	69.18.21.229/32	–
SSH	TCP	22	69.18.21.229/32	–

Whitelisting the ALB SG in Instance's SG

EC2 > Security Groups > sg-0bde078b4803140da - allow-my-http-ssh

sg-0bde078b4803140da - allow-my-http-ssh

Actions

Details

Security group name	Security group ID	Description	VPC ID
allow-my-http-ssh	sg-0bde078b4803140da	launch-wizard-1 created 2021-05-22T12:26:31.891-05:00	vpc-063dae80fe38de125

Owner	Inbound rules count	Outbound rules count
637278349805	3 Permission entries	1 Permission entry

Inbound rules | Outbound rules | Tags

Inbound rules (3)

Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	69.18.21.229/32	–
HTTP	TCP	80	sg-0ebb0b5c99091815d / my-alb-sg	–
SSH	TCP	22	69.18.21.229/32	–

Health check is now green

EC2 > Target groups > my-tg

my-tg

Delete

arn:aws:elasticloadbalancing:us-east-1:637278349805:targetgroup/my-tg/1a341e1a092d8a30

Details

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-063dae80fe38de125

Load balancer
my-alb

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
1	1	0	0	0	0

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (1/1)

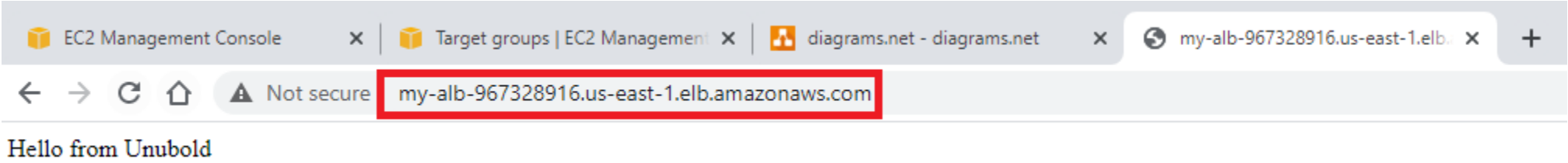
Deregister | Register targets

Filter resources by property or value

1

Instance ID	Name	Port	Zone	Status	Status details
i-0c334940879eef22e		80	us-east-1b	healthy	

ALB to EC2 result



Hello from Unubold

Routing algorithms

The routing algorithms choose the target in the target group.

Round-robin (default) load balancing is one of the simplest methods for distributing client requests across a group of servers. Going down the list of servers in the group, the round-robin load balancer forwards a client request to each server in turn. When it reaches the end of the list, the load balancer loops back and goes down the list again.

Least outstanding requests is an algorithm that choses which instance receives the next request by selecting the instance that, at that moment, has the lowest number of outstanding (pending, unfinished) requests.

Auto Scaling benefits

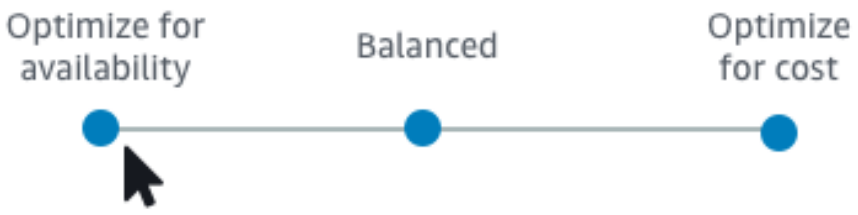
- **Better fault tolerance** - Auto Scaling can detect when a resource is unhealthy, terminate it, and launch a resource to replace it. You can also configure resources to use multiple AZs. If one AZ becomes unavailable, Auto Scaling can launch resources in another one to compensate.
- **Better availability** - Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.
- **Better cost management** - Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the resources you use.

AWS Auto Scaling

Auto Scaling is enabled by **CloudWatch**

Scaling Strategy - Your own custom strategy. You can optimize

- for availability
- for cost
- a balance of both.

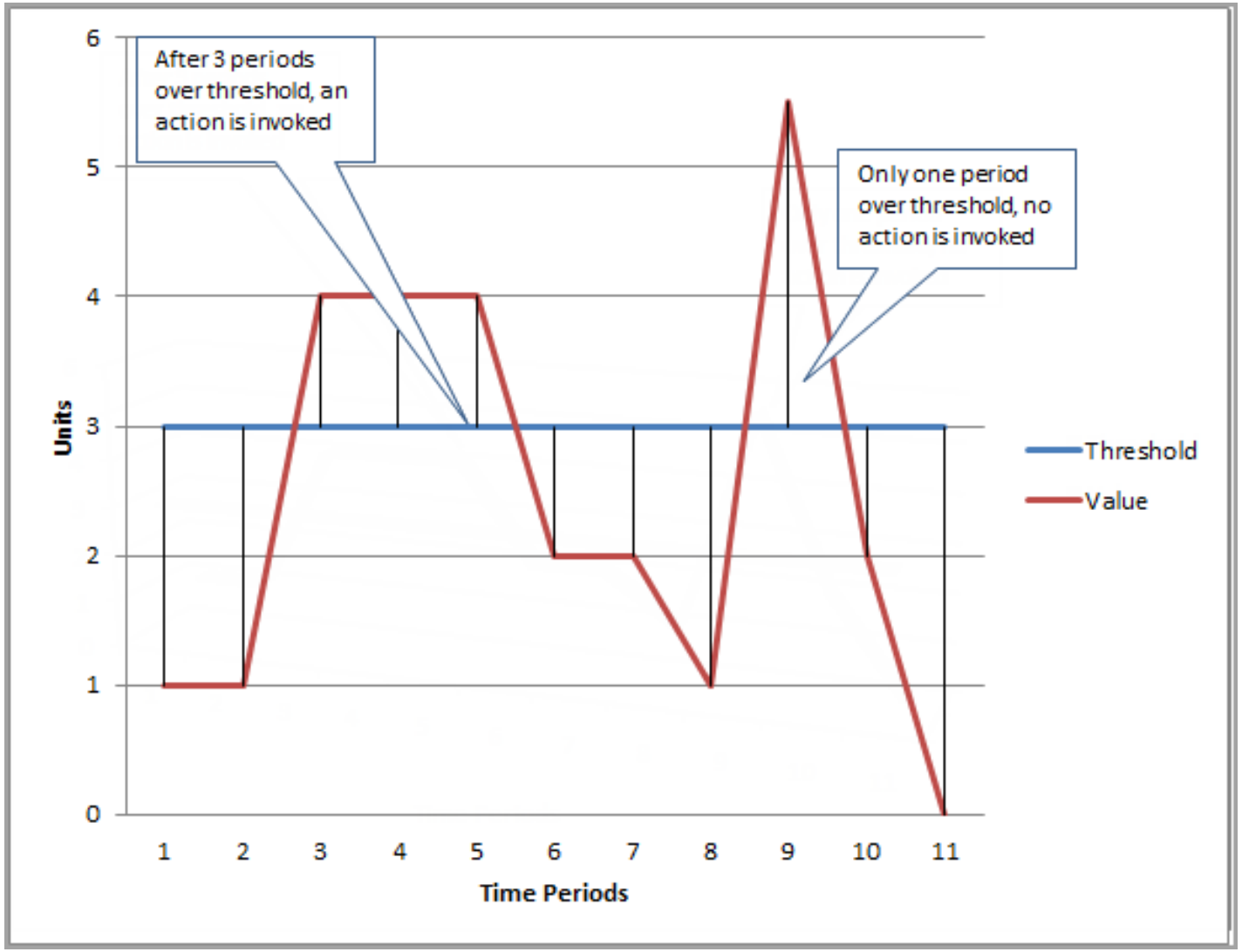


What is Auto Scaling?

Auto Scaling is nothing but it changes the desired capacity of the services such as EC2 based on the CloudWatch Alarms.

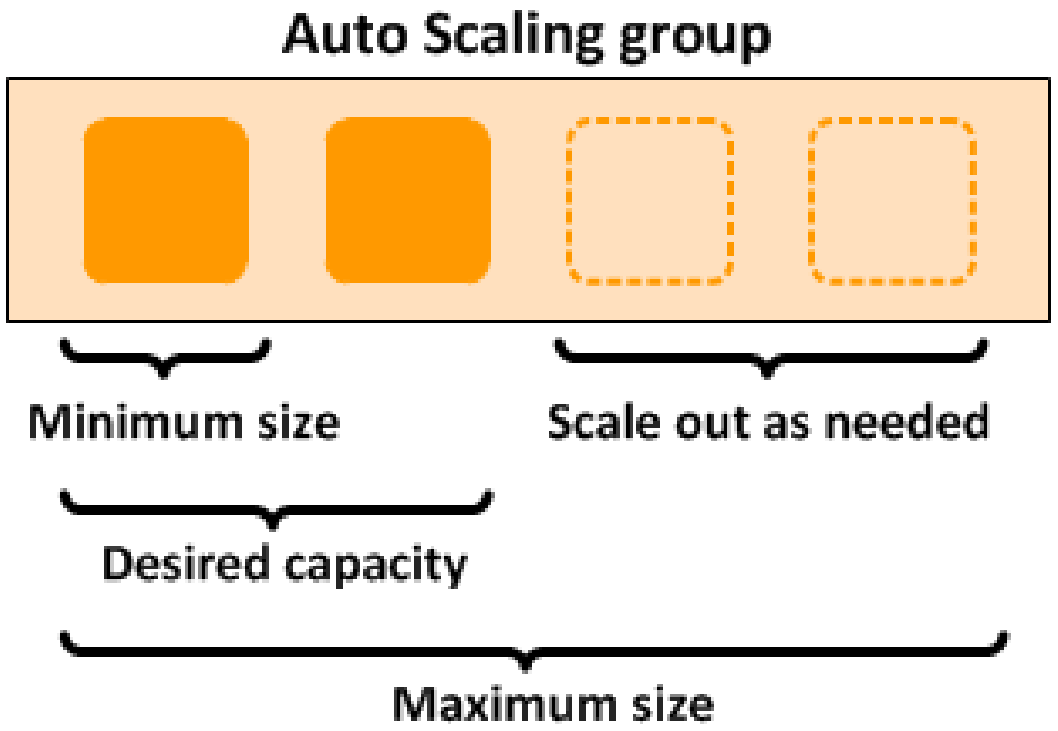
AWS Auto Scaling notifies the service that it is time to scale out or scale in. Then scaling happens on the service.

CloudWatch Alarm for AutoScaling



Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called **Auto Scaling groups**.



EC2 Auto Scaling components

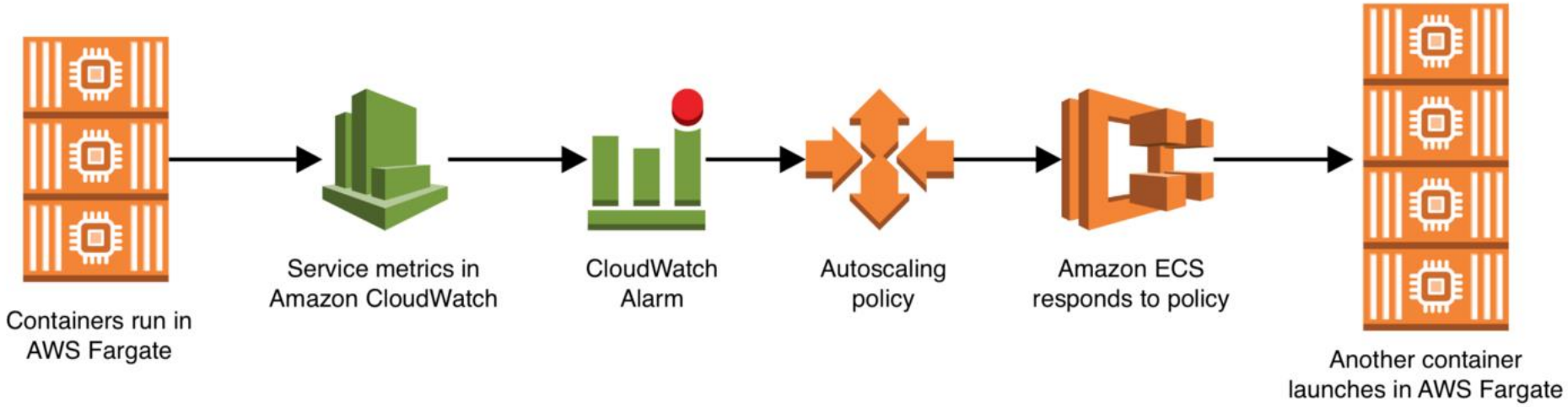
Groups - Your EC2 instances are organized into groups so that they can be treated as a logical unit for the purposes of scaling and management.

Configuration templates - Your group uses a launch (configuration) template where you can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances.

Scaling options - Amazon EC2 Auto Scaling provides several ways for you to scale your Auto Scaling groups. For example, you can configure a group to scale based on the occurrence of specified conditions (dynamic scaling) or on a schedule.

Application Auto Scaling

Application Auto Scaling is a web service for developers and system administrators who need a solution for automatically scaling their scalable resources for individual AWS services beyond Amazon EC2. You can scale Elastic Container Service (ECS), Lambda, Aurora replicas (RDS), DynamoDB, and more.



Features of Application Auto Scaling

Application Auto Scaling allows you to automatically scale your scalable resources according to conditions that you define:

1. **Target tracking scaling** - Scale a resource based on a target value for a specific CloudWatch metric. Like thermostat at your home.
2. **Step scaling** - Scale a resource based on a set of scaling adjustments that vary based on the size of the alarm breach.
3. **Scheduled scaling** - Scale a resource based on the date and time.
4. **Predictive scaling** - Uses machine learning to analyze each resource's historical workload and regularly forecasts the future load for the next two days.

Cooldown Period

The amount of time to wait for a previous scaling activity to take effect is called the cooldown period.

There are two types of cooldown periods:

1. **Scale-out** cooldown period, the intention is to continuously (but not excessively) scale out. After successfully scales out, it starts to calculate the cooldown time. The scaling policy won't increase the **desired capacity** again unless either a larger scale out is triggered or the cooldown period ends.
2. **Scale-in** cooldown period, the intention is to scale in conservatively to protect your application's availability, so scale-in activities are blocked until the cooldown period has expired. However, if another alarm triggers a scale-out activity during the scale-in cooldown period, Application Auto Scaling scales out the target immediately.