**Auto Scaling**

An Auto Scaling group is a set of Amazon EC2 instances treated as a unit. You configure group settings, instance details, and set minimum, maximum, and desired capacity. Different min and max capacity values define the scaling bounds, allowing the group to adjust to application demand spikes. Scaling is achieved by manual adjustments or automatic addition/removal of capacity by Amazon EC2 Auto Scaling.

When launching instances, specify the percentage fulfilled by On-Demand or Spot Instances to save costs. EC2 Auto Scaling optimizes availability by balancing capacity across Availability Zones. It offers lifecycle hooks, health checks, and scheduled scaling for automated capacity management.

**Benefits and features:**

- Fault tolerance: Detect and replace unhealthy instances automatically for improved system reliability.

- Cost Management: Save costs by dynamically launching instances based on demand and terminating them when not needed.

- Availability: Ensure optimal capacity to handle current traffic demand, enhancing application availability.

**Related services:**

- Amazon EC2: Create and run virtual machines in the cloud with varying CPU, memory, storage, and networking capacities.

- Elastic Load Balancing: Automatically distribute incoming traffic across Auto Scaling group instances.

- Amazon CloudWatch: Enable scaling policies and monitor metrics for Auto Scaling groups and EC2 instances.

**Notes:** Amazon EC2 Auto Scaling ensures the right number of instances for your application load. Create Auto Scaling groups with specified min, max, and desired capacities. It maintains instances above the minimum and below the maximum. Define scaling policies for automatic instance adjustments based on application demand. For instance, an Auto Scaling group with a minimum of one, desired of two, and a maximum of four instances can dynamically scale within these bounds as defined by your scaling policies.


   An illustration of a basic Auto Scaling group.
  

**Amazon EC2 Auto Scaling offers several features to enhance scalability, resilience, and management:**

1. Auto Scaling Groups: Organizes EC2 instances into logical units for scaling and management.

2. Launch Templates: Configures EC2 instances within Auto Scaling groups.

3. Health Monitoring:

- Utilizes EC2 health checks to monitor and maintain instance health.

- Allows custom health checks specific to your application.

4. Availability Zone Balancing:

- Supports multiple Availability Zones for even instance distribution.

- Enhances high availability by preventing failures in a single location.

5. Instance Type and Purchase Options:

- Launches multiple instance types within a single Auto Scaling group.

- Supports Spot and On-Demand Instances for cost optimization.

- Utilizes Reserved Instance and Savings Plan discounts.

6. Automated Spot Instance Replacement:

- Automatically requests replacement Spot capacity if interrupted.

- Implements Capacity Rebalancing to proactively replace Spot Instances at risk.

7. Load Balancing:

- Integrates with Elastic Load Balancing for even traffic distribution.

- Registers and deregisters instances automatically with load balancer changes.

8. Scalability:

- Allows auto scaling for maintaining availability and cost efficiency.

- Enables manual adjustment of Auto Scaling group size.

9. Instance Refresh:

- Facilitates rolling updates for instances when updating AMI or launch template.

- Supports phased canary deployment for testing on a subset of instances.

10. Lifecycle Hooks:

- Defines custom actions during instance launch or termination.

- Useful for event-driven architectures and instance lifecycle management.

11. Support for Stateful Workloads:

- Offers lifecycle hooks for persisting state on shut down.

- Provides scale-in protection and custom termination policies for stateful applications.

12. Instance State Continuity:

- Ensures continuity for stateful applications during scale-in processes.

Pricing for Amazon EC2 Auto Scaling is transparent, with no additional fees. Users only pay for the AWS resources (e.g., EC2 instances, EBS volumes, and CloudWatch alarms) they consume.

**launch template**

A launch template in Amazon EC2 specifies instance configuration, similar to a launch configuration. It includes AMI ID, instance type, key pair, security groups, and other launch parameters. The key difference is the ability to have multiple versions of a launch template, providing flexibility.

1. Versioning:

- Enables multiple versions of a launch template.

- Allows creating subsets of parameters for different configurations.

2. Example Usage:

- Create a base launch template without AMI or user data.

- Generate new versions to add the latest AMI and user data for testing.

- Maintain a base configuration and create versions as needed.

3. Flexibility:

- Easily modify and update specific parameters in different versions.

- Useful for managing configurations across application development stages.

4. Testing and Clean-up:

- Create versions for testing with the latest application configurations.

- Delete versions when no longer needed, ensuring a clean configuration history.

By utilizing launch templates with versioning, users can efficiently manage and update EC2 instance configurations with flexibility and control over different application versions and testing scenarios.

## Notes:

## What Is Horizontal Scaling?

Horizontal scaling (aka scaling out) refers to adding additional [nodes](https://medium.com/coinmonks/what-are-nodes-nodes-in-a-nutshell-f5d567bc9662) or machines to your [infrastructure](https://www.cloudzero.com/blog/cloud-infrastructure) to cope with new demands. If you are hosting an application on a server and find that it no longer has the capacity or capabilities to handle traffic, adding a server may be your solution.

## What Is Vertical Scaling?

Vertical scaling (aka scaling up) describes adding additional resources to a system so that it meets demand. How is this different from horizontal scaling?

While horizontal scaling refers to adding additional nodes, vertical scaling describes adding more power to your current machines. For instance, if your server requires more processing power, vertical scaling would mean upgrading the CPUs. You can also vertically scale the memory, storage, or network speed.