



AWS AutoScaling

Khalid Bin Sattar



Agenda

- Overview of AWS Auto Scaling.
- Introduction to AWS Auto Scaling .
- What Is Auto Scaling?
- Concept of Auto Scaling.
- Benefits of Auto Scaling.
- Auto Scaling Lifecycle.
- Launch Configurations.
- Auto Scaling Groups.
- Scaling Your Group.
- Auto Scaling Limits.
- Demo of Auto Scaling.



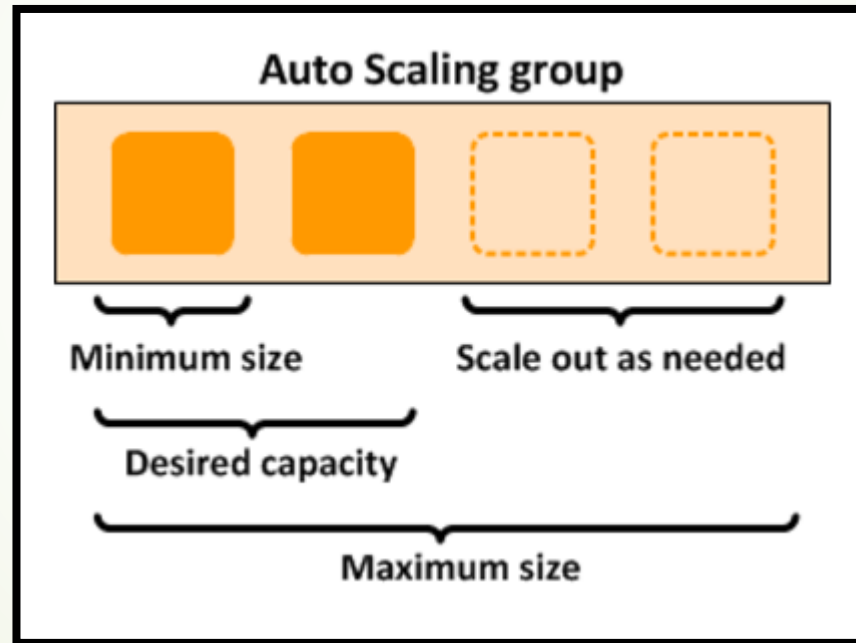
Overview of AWS AutoScaling

What Is Auto Scaling?

- 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.
- You can specify the minimum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes above this size.
- If you specify the desired capacity, either when you create the group or at any time thereafter, Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

Overview of AWS AutoScaling

Architecture of Auto Scaling





Benefits of Auto Scaling



Benefits of Auto Scaling

Adding Auto Scaling to your application architecture is one way to maximize the benefits of the AWS cloud. When you use Auto Scaling, your applications gain the following benefits:

- Better fault tolerance. Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Auto Scaling can launch instances in another one to compensate.
- Better availability. Auto Scaling can help you ensure that your application always has the right amount of capacity to handle the current traffic demands.
- Better cost management. Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are actually needed and terminating them when they aren't needed.

Benefits of Auto Scaling

Example: Covering Variable Demand.

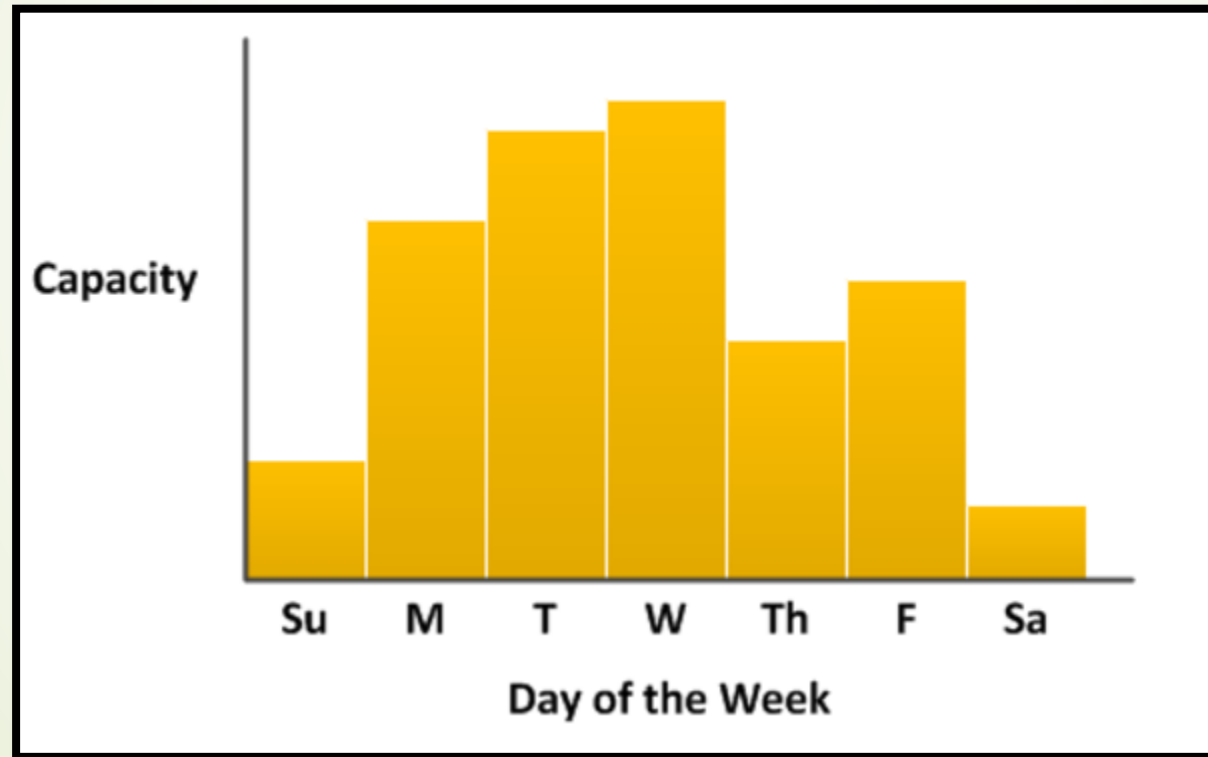


Fig. 1

Benefits of Auto Scaling

Example: Covering Variable Demand.

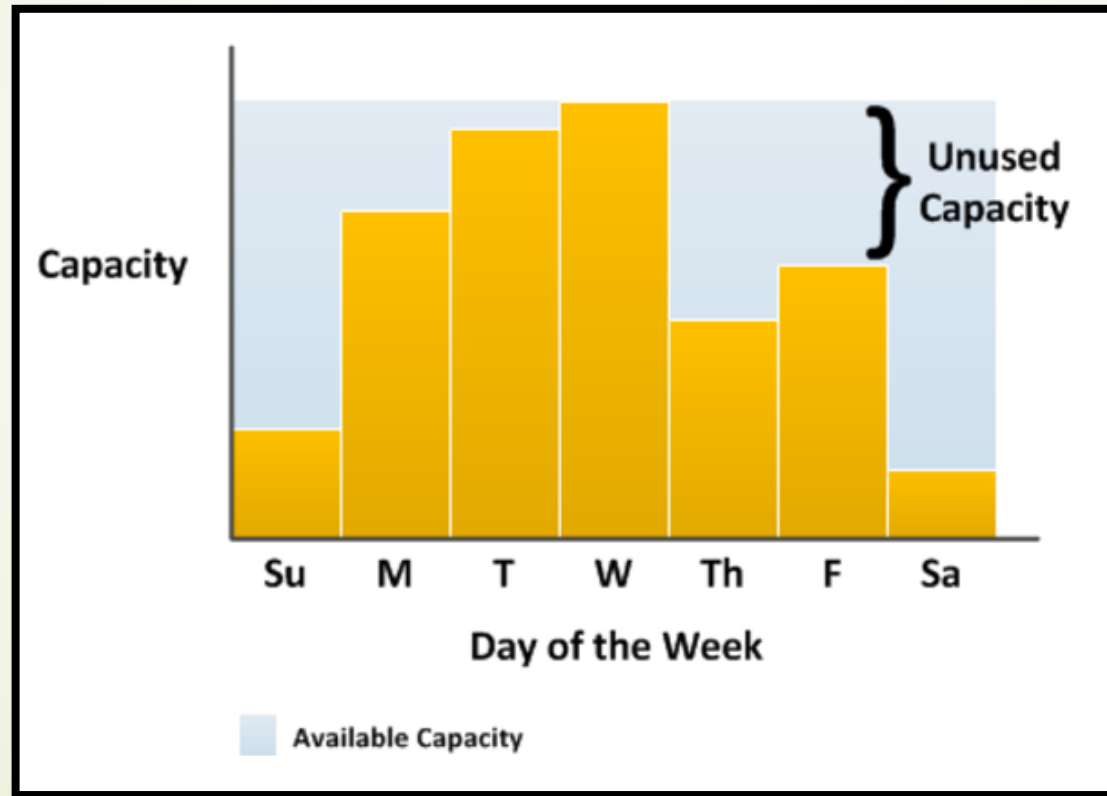


Fig. 2

Benefits of Auto Scaling

Example: Covering Variable Demand.

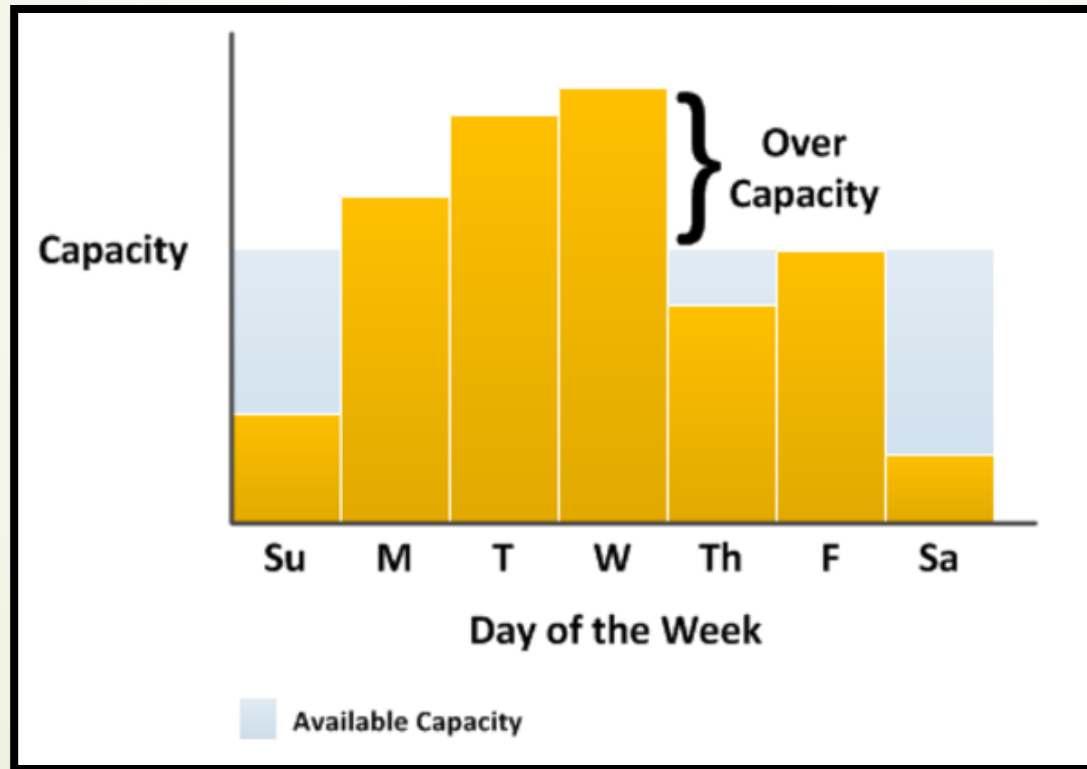


Fig. 3

Benefits of Auto Scaling

Example: Covering Variable Demand.

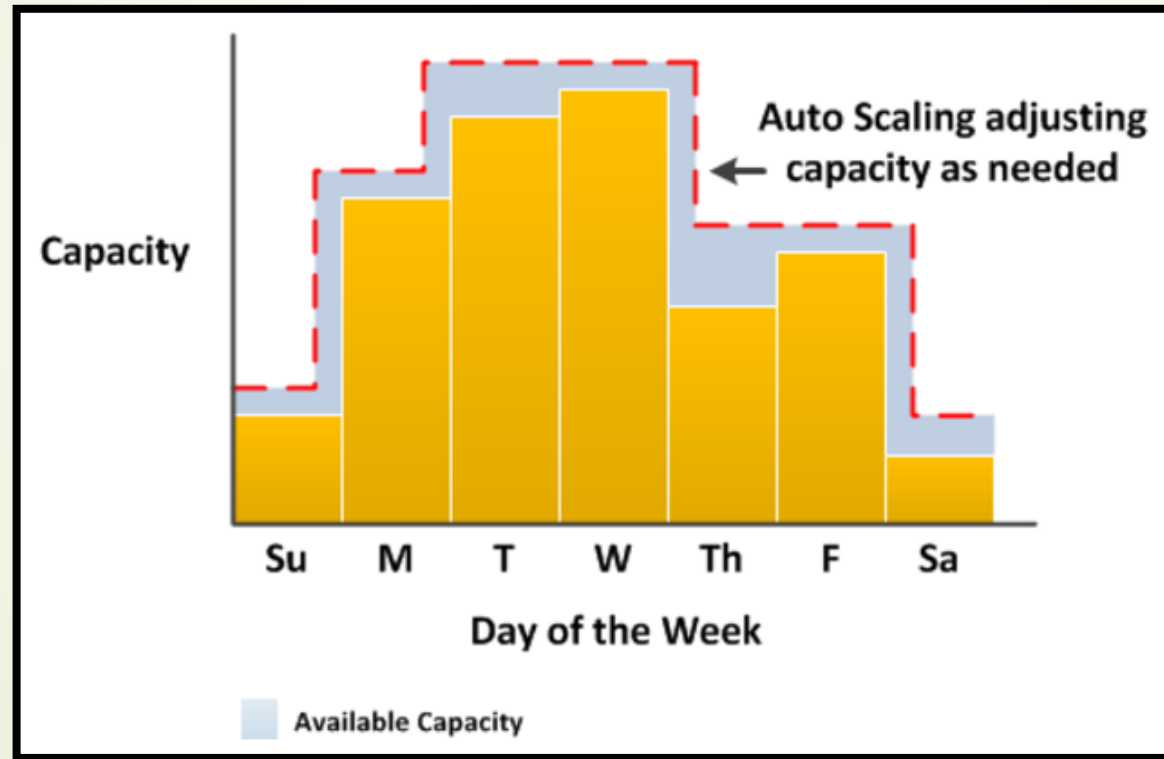


Fig. 4

Benefits of Auto Scaling

Example: Web App Architecture.

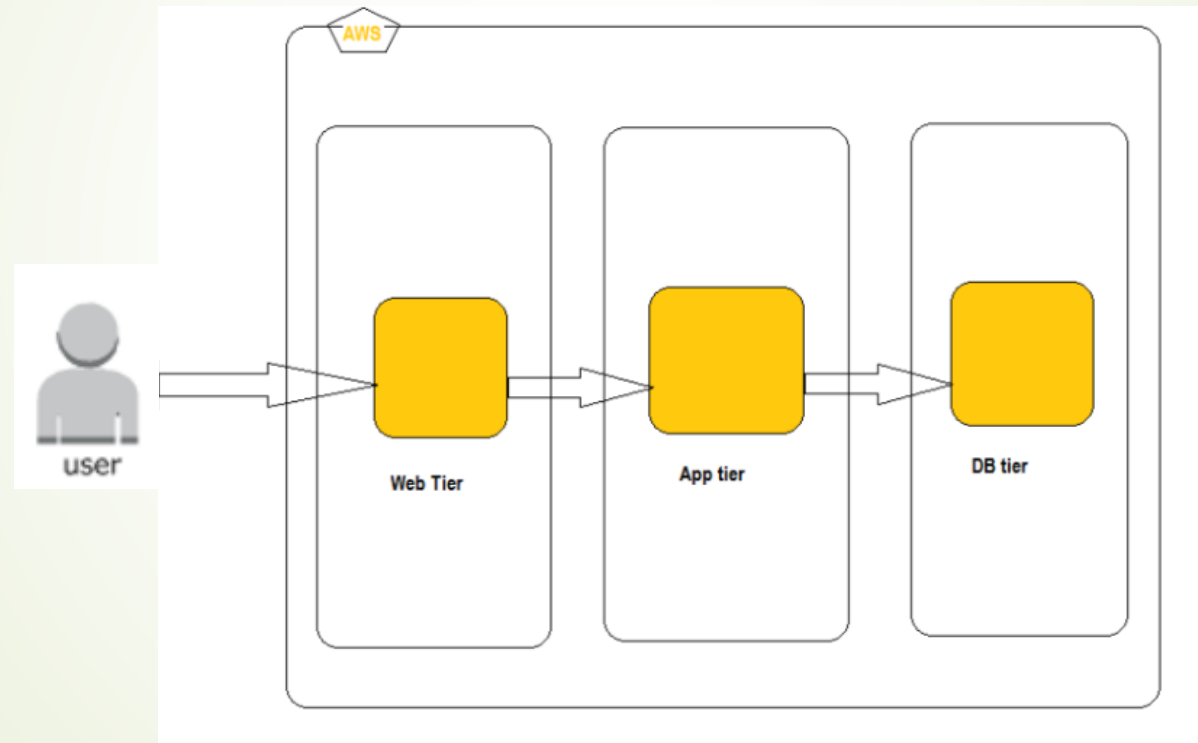


Fig. 1

Benefits of Auto Scaling

Example: Web App Architecture.

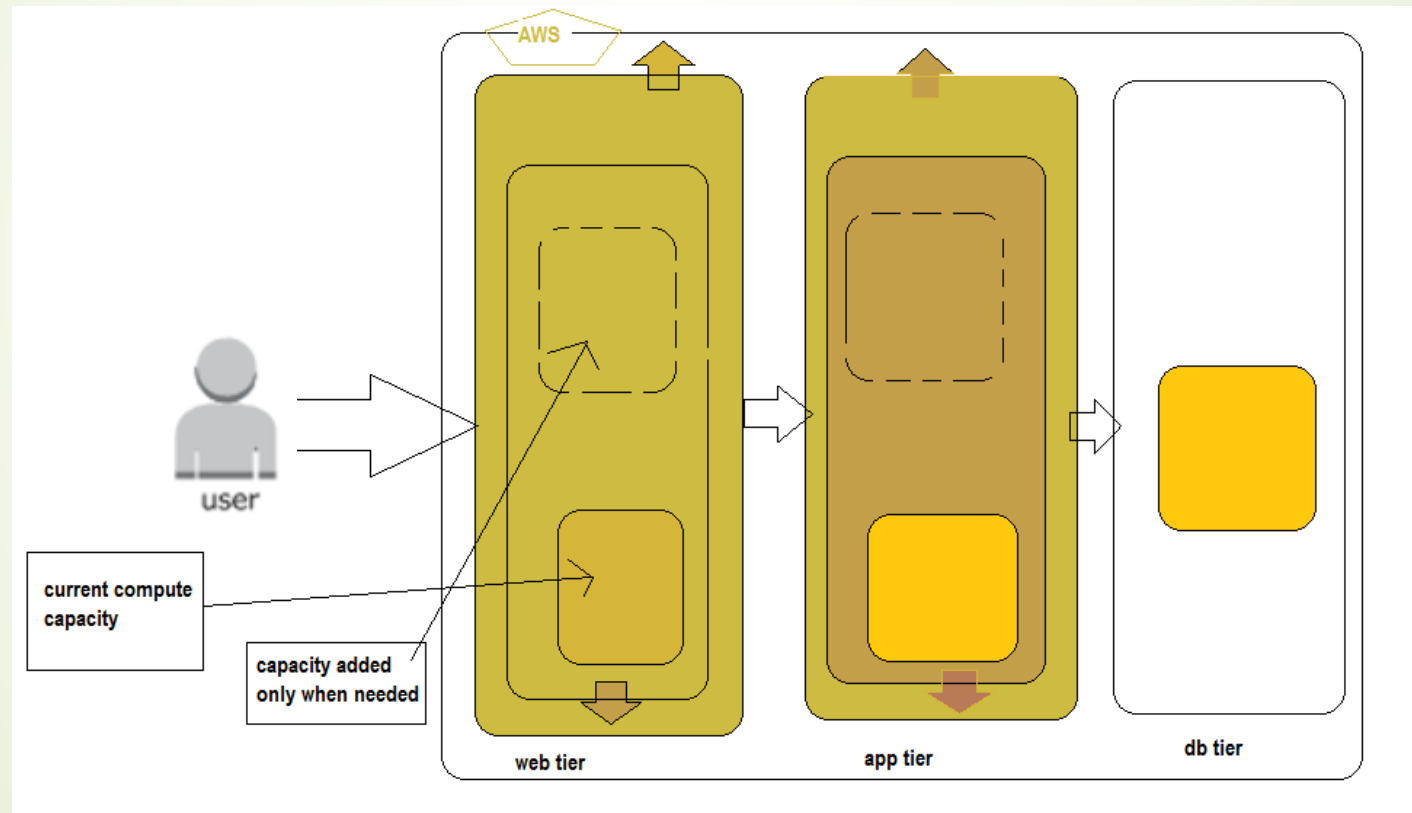


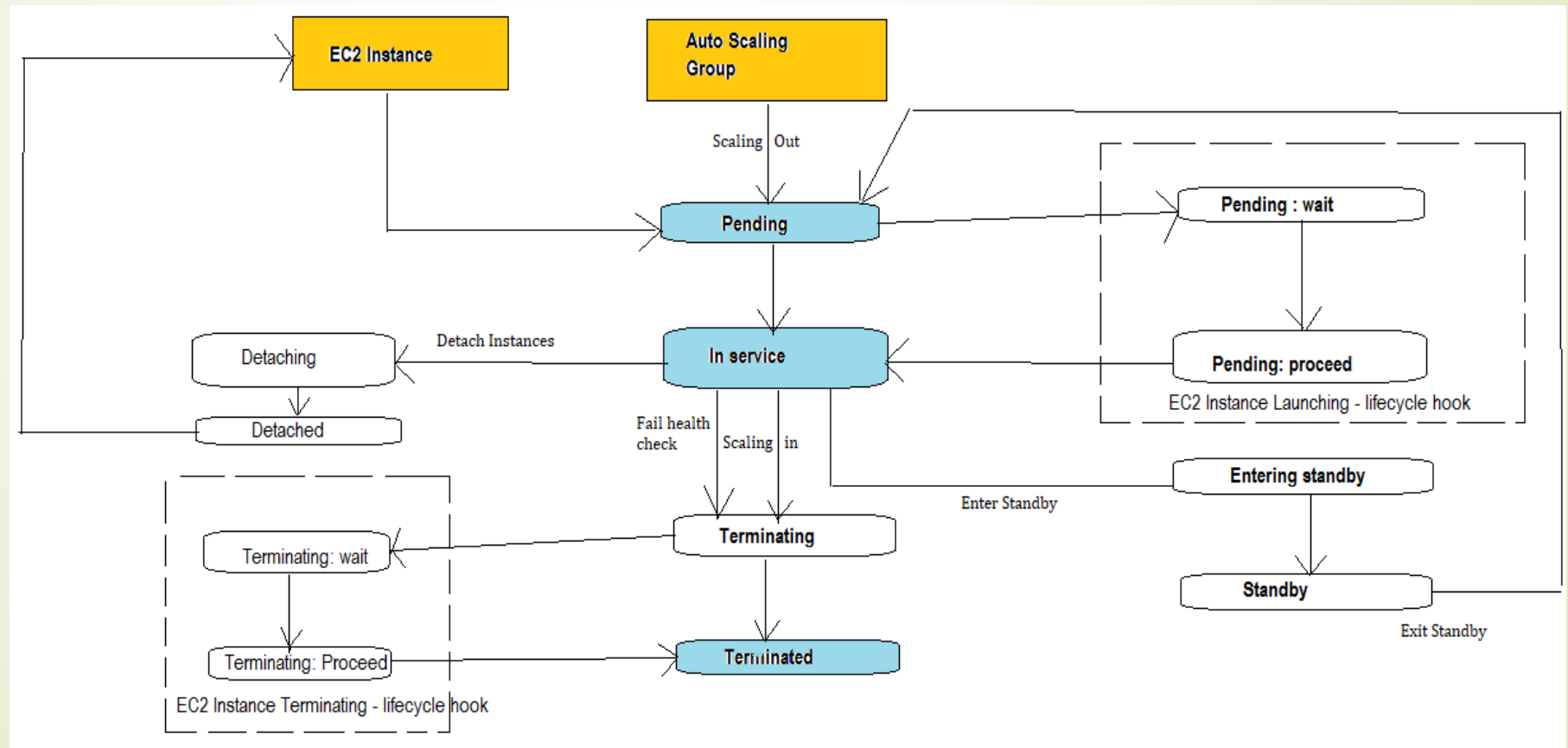
Fig. 2



Auto Scaling Lifecycle

Auto Scaling Lifecycle

The EC2 instances in an Auto Scaling group have a path, or lifecycle, that differs from that of other EC2 instances. The lifecycle starts when the Auto Scaling group launches an instance and puts it into service. The lifecycle ends when you terminate the instance, or the Auto Scaling group takes the instance out of service and terminates it.



Auto Scaling Lifecycle

Scale Out :- The following scale out events direct the Auto Scaling group to launch EC2 instances and attach them to the group.

1. You manually increase the size of the group.
 2. You create a scaling policy to automatically increase the size of the group based on a specified increase in demand.
 3. You set up scaling by schedule to increase the size of the group at a specific time.
- When a scale out event occurs, the Auto Scaling group launches the required number of EC2 instances, using its assigned launch configuration. These instances start in the Pending state. If you add a lifecycle hook to your Auto Scaling group, you can perform a custom action here.
 - When each instance is fully configured and passes the Amazon EC2 health checks, it is attached to the Auto Scaling group and it enters the InService state. The instance is counted against the desired capacity of the Auto Scaling group.

Auto Scaling Lifecycle

Instances In Service:-

A scale in event occurs, and Auto Scaling chooses to terminate this instance in order to reduce the size of the Auto Scaling group. You put the instance into a Standby state.

- You detach the instance from the Auto Scaling group.
- The instance fails a required number of health checks, so it is removed from the Auto Scaling group, terminated, and replaced.

Scale In:-

It is important that you create a scale in event for each scale out event that you create. This helps ensure that the resources assigned to your application match the demand for those resources as closely as possible.

1. You manually decrease the size of the group.
2. You create a scaling policy to automatically decrease the size of the group based on a specified decrease in demand.
3. You set up scaling by schedule to decrease the size of the group at a specific time.

When a scale in event occurs, the Auto Scaling group detaches one or more instances. The Auto Scaling group uses its termination policy to determine which instances to terminate. Instances that are in the process of detaching from the Auto Scaling group and shutting down enter the Terminating state, and can't be put back into service.



Auto Scaling Lifecycle

Attach an Instance:-

You can attach a running EC2 instance that meets certain criteria to your Auto Scaling group. After the instance is attached, it is managed as part of the Auto Scaling group.

Detach an Instance:-

You can detach an instance from your Auto Scaling group. After the instance is detached, you can manage it separately from the Auto Scaling group or attach it to a different Auto Scaling group.

Lifecycle Hooks:-

- You can add a lifecycle hook to your Auto Scaling group so that you can perform custom actions when instances launch or terminate.
- When Auto Scaling responds to a scale out event, it launches one or more instances. These instances start in the Pending state. If you added an autoscaling:EC2_INSTANCE_LAUNCHING lifecycle hook to your Auto Scaling group, the instances move from the Pending state to the Pending:Wait state. After you complete the lifecycle action, the instances enter the Pending:Proceed state. When the instances are fully configured, they are attached to the Auto Scaling group and they enter the InService state.



Auto Scaling Lifecycle

Lifecycle Hooks:-

When Auto Scaling responds to a scale in event, it terminates one or more instances. These instances are detached from the Auto Scaling group and enter the Terminating state. If you added an `autoscaling:EC2_INSTANCE_TERMINATING` lifecycle hook to your Auto Scaling group, the instances move from the Terminating state to the Terminating:Wait state. After you complete the lifecycle action, the instances enter the Terminating:Proceed state. When the instances are fully terminated, they enter the Terminated state.

Enter and Exit Standby:-

- You can put any instance that is in an InService state into a Standby state. This enables you to remove the instance from service, troubleshoot or make changes to it, and then put it back into service.
- Instances in a Standby state continue to be managed by the Auto Scaling group. However, they are not an active part of your application until you put them back into service.



Getting Started with Auto Scaling



Getting Started with Auto Scaling

Whenever you plan to use Auto Scaling, you must use certain building blocks to get started. This tutorial walks you through the process for setting up the basic infrastructure for Auto Scaling.

The following step-by-step instructions help you create a template that defines your EC2 instances, create an Auto Scaling group to maintain the healthy number of instances at all times, and optionally delete this basic Auto Scaling infrastructure. This tutorial assumes that you are familiar with launching EC2 instances and have already created a key pair and a security group.

Tasks:-

- Step 1: Create a Launch Configuration.
- Step 2: Create an Auto Scaling Group.
- Step 3: Verify Your Auto Scaling Group.
- Step 4: (Optional) Delete Your Auto Scaling Infrastructure.



Getting Started with Auto Scaling



➤ Step 1: Create a Launch Configuration.

A launch configuration specifies the type of EC2 instance that Auto Scaling creates for you. You create the launch configuration by including information such as the Amazon Machine Image (AMI) ID to use for launching the EC2 instance, the instance type, key pairs, security groups, and block device mappings, among other configuration settings.

➤ Step 2: Create an Auto Scaling Group

An Auto Scaling group is a collection of EC2 instances, and the core of the Auto Scaling service. You create an Auto Scaling group by specifying the launch configuration you want to use for launching the instances and the number of instances your group must maintain at all times. You also specify the Availability Zone in which you want the instances to be launched.



Getting Started with Auto Scaling

➤ Step 3: Verify Your Auto Scaling Group.

Now that you have created your Auto Scaling group, you are ready to verify that the group has launched an EC2 instance.

➤ Step 4: (Optional) Delete Your Auto Scaling Infrastructure.

You can either delete your Auto Scaling set up or delete just your Auto Scaling group and keep your launch configuration to use at a later time.

Tutorial: Set Up a Scaled and Load-Balanced Application



Set Up a Scaled and Load-Balanced Application

You can attach a load balancer to your Auto Scaling group. The load balancer automatically distributes incoming traffic across the instances in the group.

Configure Scaling and Load Balancing Using the AWS Management Console:-

Complete the following tasks to set up a scaled and load-balanced application when you create your Auto Scaling group.

Task:-

- Create or Select a Launch Configuration.
- Create an Auto Scaling Group.
- Create an Load Balancer and Target Group.
- (Optional) Verify that Your Load Balancer is Attached to Your Auto Scaling Group.
- Delete the Instance which is attached to Auto Scaling and Test Auto Scaling Feature.



Tutorial: Launch Configurations



Launch Configurations

- ❖ A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. If you've launched an EC2 instance before, you specified the same information in order to launch the instance.
- ❖ When you create an Auto Scaling group, you must specify a launch configuration. You can specify your launch configuration with multiple Auto Scaling groups. However, you can only specify one launch configuration for an Auto Scaling group at a time, and you can't modify a launch configuration after you've created it. Therefore, if you want to change the launch configuration for your Auto Scaling group, you must create a launch configuration and then update your Auto Scaling group with the new launch configuration.

Contents :-

- Creating a Launch Configuration.
- Creating a Launch Configuration Using an EC2 Instance.
- Changing the Launch Configuration for an Auto Scaling Group.
- Launching Auto Scaling Instances in a VPC.



Tutorial: Auto Scaling Groups



Auto Scaling Groups

- ❖ An Auto Scaling group contains a collection of EC2 instances that share similar characteristics and are treated as a logical grouping for the purposes of instance scaling and management. For example, if a single application operates across multiple instances, you might want to increase the number of instances in that group to improve the performance of the application, or decrease the number of instances to reduce costs when demand is low. You can use the Auto Scaling group to scale the number of instances automatically based on criteria that you specify, or maintain a fixed number of instances even if an instance becomes unhealthy. This automatic scaling and maintaining the number of instances in an Auto Scaling group is the core functionality of the Auto Scaling service.
- ❖ An Auto Scaling group starts by launching enough EC2 instances to meet its desired capacity. The Auto Scaling group maintains this number of instances by performing periodic health checks on the instances in the group. If an instance becomes unhealthy, the group terminates the unhealthy instance and launches another instance to replace it.
- ❖ You can use scaling policies to increase or decrease the number of running EC2 instances in your group automatically to meet changing conditions. When the scaling policy is in effect, the Auto Scaling group adjusts the desired capacity of the group and launches or terminates the instances as needed. If you manually scale or scale on a schedule, you must adjust the desired capacity of the group in order for the changes to take effect.



Auto Scaling Groups



❑ Contents:-

- ❖ Creating an Launch Configuration using CLI.
- ❖ Creating an Auto Scaling Group using CLI.
- ❖ Creating an Auto Scaling Group Using an EC2 Instance.
- ❖ Tagging Auto Scaling Groups.
- ❖ Using a Load Balancer With an Auto Scaling Group.
- ❖ Deleting Your Auto Scaling Infrastructure.

Scaling the Size of Your Auto Scaling Group





Scaling the Size of Your Auto Scaling Group

- ❖ Scaling is the ability to increase or decrease the compute capacity of your application. Scaling starts with an event, or scaling action, which instructs Auto Scaling to either launch or terminate EC2 instances.
- ❖ Auto Scaling provides a number of ways to adjust scaling to meet the needs of your applications. As a result, it's important that you have a good understanding of your application. You have to Keep 3 points in mind:
 1. What role do you want Auto Scaling to play in your application's architecture? It's common to think about Auto Scaling as a way to increase and decrease capacity, but it's also useful for maintaining a steady number of servers.
 2. What cost constraints are important to you? Because Auto Scaling uses EC2 instances, you only pay for the resources that you use. Knowing your cost constraints helps you decide when to scale your applications, and by how much.
 3. What metrics are important to your application? CloudWatch supports a number of different metrics that you can use with your Auto Scaling group. AWS recommend reviewing them to see which of these metrics are the most relevant to your application.

Scaling Plans





Scaling Plans

Auto Scaling provides several ways for you to scale your Auto Scaling group.

- ❖ Maintain current instance levels at all times:-You can configure your Auto Scaling group to maintain a minimum or specified number of running instances at all times. To maintain the current instance levels, Auto Scaling performs a periodic health check on running instances within an Auto Scaling group. When Auto Scaling finds an unhealthy instance, it terminates that instance and launches a new one.
- ❖ Manual scaling:-Manual scaling is the most basic way to scale your resources. Specify only the change in the maximum, minimum, or desired capacity of your Auto Scaling group. Auto Scaling manages the process of creating or terminating instances to maintain the updated capacity.
- ❖ Scale based on a schedule:-Sometimes you know exactly when you will need to increase or decrease the number of instances in your group, simply because that need arises on a predictable schedule. Scaling by schedule means that scaling actions are performed automatically as a function of time and date.



Scaling Plans

Scale based on demand:-


- ❖ A more advanced way to scale your resources, scaling by policy, lets you define parameters that control the Auto Scaling process. For example, you can create a policy that calls for enlarging your fleet of EC2 instances whenever the average CPU utilization rate stays above ninety percent for fifteen minutes. This is useful when you can define how you want to scale in response to changing conditions, but you don't know when those conditions will change. You can set up Auto Scaling to respond for you.
- ❖ You should have two policies, one for scaling in (terminating instances) and one for scaling out (launching instances), for each event to monitor. For example, if you want to scale out when the network bandwidth reaches a certain level, create a policy specifying that Auto Scaling should start a certain number of instances to help with your traffic. But you may also want an accompanying policy to scale in by a certain number when the network bandwidth level goes back down.



Scaling Plans

Multiple Scaling Policies:-

- ❖ An Auto Scaling group can have more than one scaling policy attached to it any given time. In fact, AWS recommend that each Auto Scaling group has at least two policies: one to scale your architecture out and another to scale your architecture in. You can also combine scaling policies to maximize the performance of an Auto Scaling group.
- ❖ When you have more than one policy attached to an Auto Scaling group, there's a chance that both policies could instruct Auto Scaling to scale out (or in) at the same time.



Maintaining the Number of Instances in Your Auto Scaling Group

Maintaining the No. of Instances in Auto Scaling Group

- ❖ After you have created your launch configuration and Auto Scaling group, the Auto Scaling group starts by launching the minimum number of EC2 instances (or the desired capacity, if specified). If there are no other scaling conditions attached to the Auto Scaling group, the Auto Scaling group maintains this number of running instances at all times.
- ❖ To maintain the same number of instances, Auto Scaling performs a periodic health check on running instances within an Auto Scaling group. When it finds that an instance is unhealthy, it terminates that instance and launches a new one.
- ❖ All instances in your Auto Scaling group start in the healthy state. Instances are assumed to be healthy unless Auto Scaling receives notification that they are unhealthy. This notification can come from one or more of the following sources: Amazon EC2, Elastic Load Balancing, or your customized health check.

Maintaining the No. of Instances in Auto Scaling Group

Determining Instance Health

- ❖ By default, the Auto Scaling group determines the health state of each instance by periodically checking the results of EC2 instance status checks. If the instance status is any state other than running or if the system status is impaired, Auto Scaling considers the instance to be unhealthy and launches a replacement.
- ❖ If you have associated your Auto Scaling group with a load balancer or a target group and have chosen to use the ELB health checks, Auto Scaling determines the health status of the instances by checking both the instance status checks and the ELB health checks. Auto Scaling marks an instance as unhealthy if the instance is in a state other than running, the system status is impaired, or Elastic Load Balancing reports that the instance failed the health checks.
- ❖ You can customize the health check conducted by your Auto Scaling group by specifying additional checks. Or, if you have your own health check system, you can send the instance's health information directly from your system to Auto Scaling.

Maintaining the No. of Instances in Auto Scaling Group

Replacing Unhealthy Instances

- ❖ After an instance has been marked unhealthy because of an Amazon EC2 or Elastic Load Balancing health check, it is almost immediately scheduled for replacement.
- ❖ Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance.
- ❖ When your instance is terminated, any associated Elastic IP addresses are disassociated and are not automatically associated with the new instance. You must associate these Elastic IP addresses with the new instance manually. Similarly, when your instance is terminated, its attached EBS volumes are detached. You must attach these EBS volumes to the new instance manually.

Manual Scaling





Manual Scaling

At any time, you can change the size of an existing Auto Scaling group. Update the desired capacity of the Auto Scaling group, or update the instances that are attached to the Auto Scaling group.

Contents:-

- ❖ Change the Size of Your Auto Scaling Group Using the Console.
- ❖ Change the Size of Your Auto Scaling Group Using the AWS CLI.
- ❖ Attach EC2 Instances to Your Auto Scaling Group.
- ❖ Detach EC2 Instances from Your Auto Scaling Group.

Scheduled Scaling





Scheduled Scaling

- ❖ Scaling based on a schedule allows you to scale your application in response to predictable load changes. For example, every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can plan your scaling activities based on the predictable traffic patterns of your web application.
- ❖ To configure your Auto Scaling group to scale based on a schedule, you create a scheduled action, which tells Auto Scaling to perform a scaling action at specified times. To create a scheduled scaling action, you specify the start time when you want the scaling action to take effect, and the new minimum, maximum, and desired sizes for the scaling action. At the specified time, Auto Scaling updates the group with the values for minimum, maximum, and desired size specified by the scaling action.
- ❖ You can create scheduled actions for scaling one time only or for scaling on a recurring schedule.



Scheduled Scaling

Considerations for Scheduled Actions:-

When you create a scheduled action, keep the below points in mind.

- ❖ Auto Scaling guarantees the order of execution for scheduled actions within the same group, but not for scheduled actions across groups.
- ❖ A scheduled action generally executes within seconds. However, the action may be delayed for up to two minutes from the scheduled start time. Because Auto Scaling executes actions within an Auto Scaling group in the order they are specified, scheduled actions with scheduled start times close to each other can take longer to execute.
- ❖ You can create a maximum of 125 scheduled actions per Auto Scaling group.
- ❖ A scheduled action must have a unique time value. If you attempt to schedule an activity at a time when another scaling activity is already scheduled, the call is rejected with an error message noting the conflict.
- ❖ Cooldown periods are not supported.



Scheduled Scaling

Contents:-

- ❖ Create a Scheduled Action Using the Console.
- ❖ Update a Scheduled Action.
- ❖ Create or Update a Scheduled Action Using the AWS CLI.
- ❖ Delete a Scheduled Action.

Dynamic Scaling





Dynamic Scaling

When you use Auto Scaling to scale dynamically, you must define how you want to scale in response to changing demand. For example, say you have a web application that currently runs on two instances and you do not want the CPU utilization of the Auto Scaling group to exceed 70 percent. You can configure your Auto Scaling group to scale automatically to meet this need. The policy type determines how the scaling action is performed.

Scaling Policy Types:-

- ✓ Simple scaling—Increase or decrease the current capacity of the group based on a single scaling adjustment.
- ✓ 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.
- ✓ Target tracking scaling—Increase or decreases the current capacity of the group based on a target value for a specific metric. This is similar to the way that your thermostat maintains the temperature of your home – you select a temperature and the thermostat does the rest.



Dynamic Scaling

Simple and Step Scaling Policies:-

- ✓ Auto Scaling originally supported only simple scaling policies. If you created your scaling policy before target tracking and step policies were introduced, your policy is treated as a simple scaling policy.
- ✓ AWS recommend that you use step scaling policies instead of simple scaling policies even if you have a single step adjustment, because AWS continuously evaluate alarms and do not lock the group during scaling activities or health check replacements. If you are scaling based on a metric that is a utilization metric that increases or decreases proportionally to the number of instances in the Auto Scaling group, AWS recommend that you use a target tracking scaling policy instead.



Dynamic Scaling

Simple Scaling Policies:-

- ✓ After a scaling activity is started, the policy must wait for the scaling activity or health check replacement to complete and the cooldown period to expire before it can respond to additional alarms. Cooldown periods help to prevent Auto Scaling from initiating additional scaling activities before the effects of previous activities are visible. You can use the default cooldown period associated with your Auto Scaling group, or you can override the default by specifying a cooldown period for your policy.

Step Scaling Policies:-

- ✓ After a scaling activity is started, the policy continues to respond to additional alarms, even while a scaling activity or health check replacement is in progress. Therefore, all alarms that are breached are evaluated by Auto Scaling as it receives the alarm messages. If you are creating a policy to scale out, you can specify the estimated warm-up time that it takes for a newly launched instance to be ready to contribute to the aggregated metrics.



Create an Auto Scaling Group with Step Scaling Policies



Target Tracking Scaling Policies



Target Tracking Scaling Policies

- ✓ Target tracking scaling policies simplify how you configure dynamic scaling. You select a predefined metric or configure a customized metric, and set a target value. Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to the fluctuations in the metric due to a fluctuating load pattern and minimizes rapid fluctuations in the capacity of the Auto Scaling group.

For example, you could use target tracking scaling to:

- ✓ Configure a target tracking scaling policy to keep the average aggregate CPU utilization of your Auto Scaling group at 50 percent.
- ✓ Configure a target tracking scaling policy to keep the request count per target of your Elastic Load Balancing target group at 1000 for your Auto Scaling group.



Create an Auto Scaling Group with Target Tracking Scaling Policies



Auto Scaling Lifecycle Hooks



Auto Scaling Lifecycle Hooks

- ✓ Auto Scaling lifecycle hooks enable you to perform custom actions by pausing instances as Auto Scaling launches or terminates them. For example, while your newly launched instance is paused, you could install or configure software on it.
- ✓ Each Auto Scaling group can have multiple lifecycle hooks.

How Lifecycle Hooks Work:-

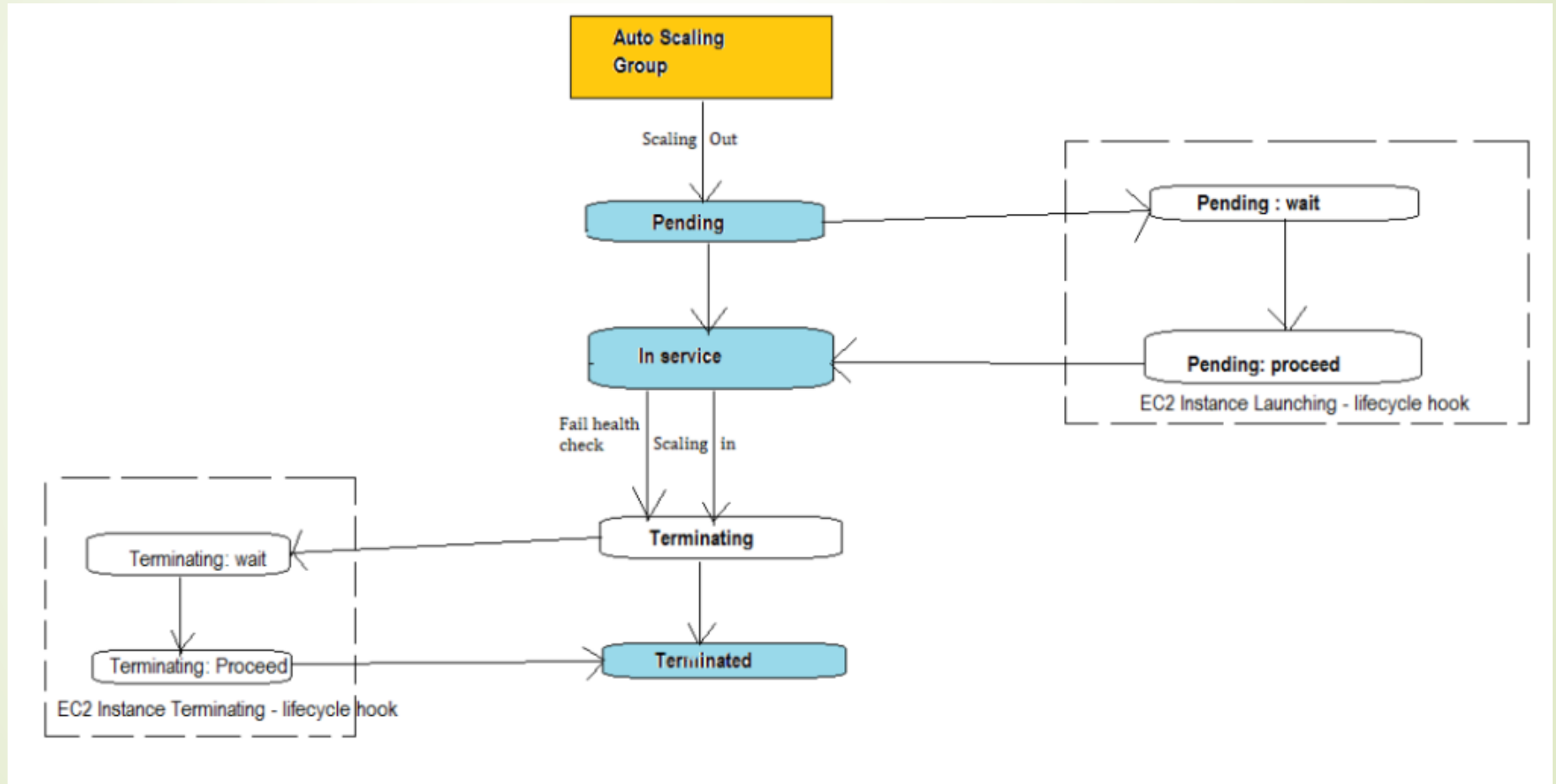
- ✓ Auto Scaling responds to scale out events by launching instances and scale in events by terminating instances.
- ✓ Auto Scaling puts the instance into a wait state (*Pending:Wait* or *Terminating:Wait*). The instance is paused until either you tell Auto Scaling to continue or the timeout period ends.



Auto Scaling Lifecycle Hooks

- ✓ You can perform a custom action like Create a script that runs on the instance as the instance starts. The script can control the lifecycle action using the ID of the instance on which it runs.
- ✓ By default, the instance remains in a wait state for one hour, and then Auto Scaling continues the launch or terminate process (Pending:Proceed or Terminating:Proceed). If you need more time, you can restart the timeout period by recording a heartbeat. If you finish before the timeout period ends, you can complete the lifecycle action, which continues the launch or termination process.

Auto Scaling Lifecycle Hooks Flow Diagram





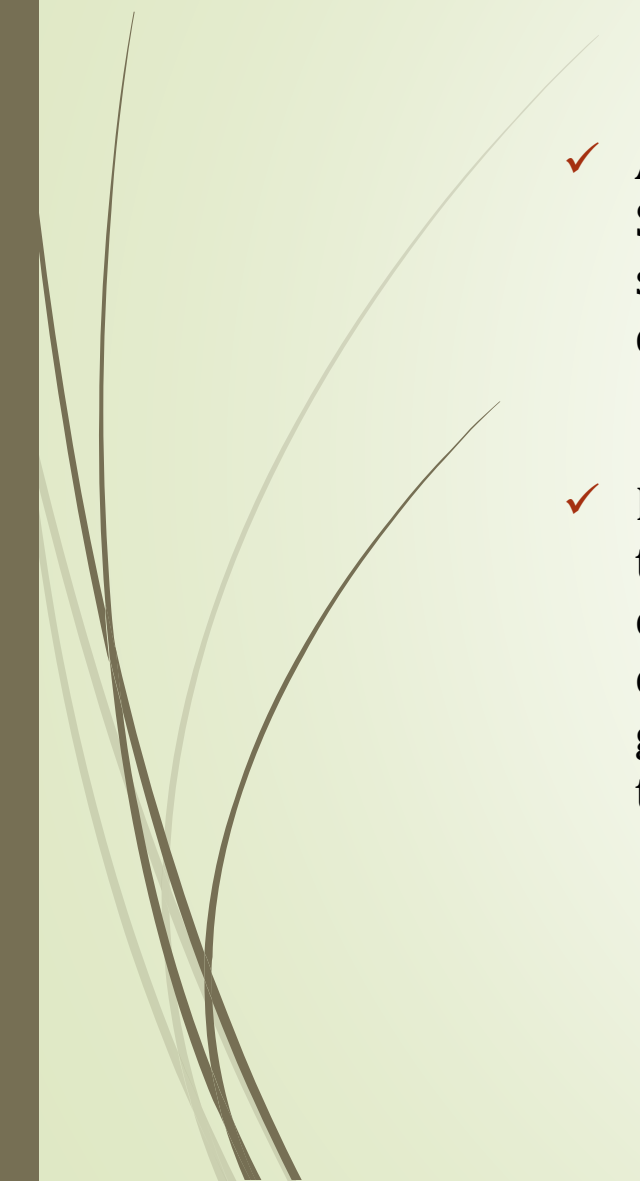
Create an Auto Scaling Group and Create Lifecycle Hooks



Temporarily Removing Instances from Your Auto Scaling Group



Removing Instances from Your Auto Scaling Group

- 
- ✓ Auto Scaling enables you to put an instance that is in the InService state into the Standby state, update or troubleshoot the instance, and then return the instance to service. Instances that are on standby are still part of the Auto Scaling group, but they do not actively handle application traffic.
 - ✓ For example, you can change the launch configuration for an Auto Scaling group at any time, and any subsequent instances that the Auto Scaling group launches use this configuration. However, the Auto Scaling group does not update the instances that are currently in service. You can either terminate these instances and let the Auto Scaling group replace them, or you can put the instances on standby, update the software, and then put the instances back in service.



Removing Instances from Your Auto Scaling Group

How the Standby State Works:-

- ✓ You put the instance into the standby state. The instance remains in this state until you exit the standby state.
- ✓ If there is a load balancer or target group attached to your Auto Scaling group, the instance is deregistered from the load balancer or target group.
- ✓ By default, Auto Scaling decrements the desired capacity of your Auto Scaling group when you put an instance on standby. This prevents Auto Scaling from launching an additional instance while you have this instance on standby. Alternatively, you can specify that Auto Scaling does not decrement the capacity. This causes Auto Scaling to launch an additional instance to replace the one on standby.
- ✓ You can update or troubleshoot the instance.



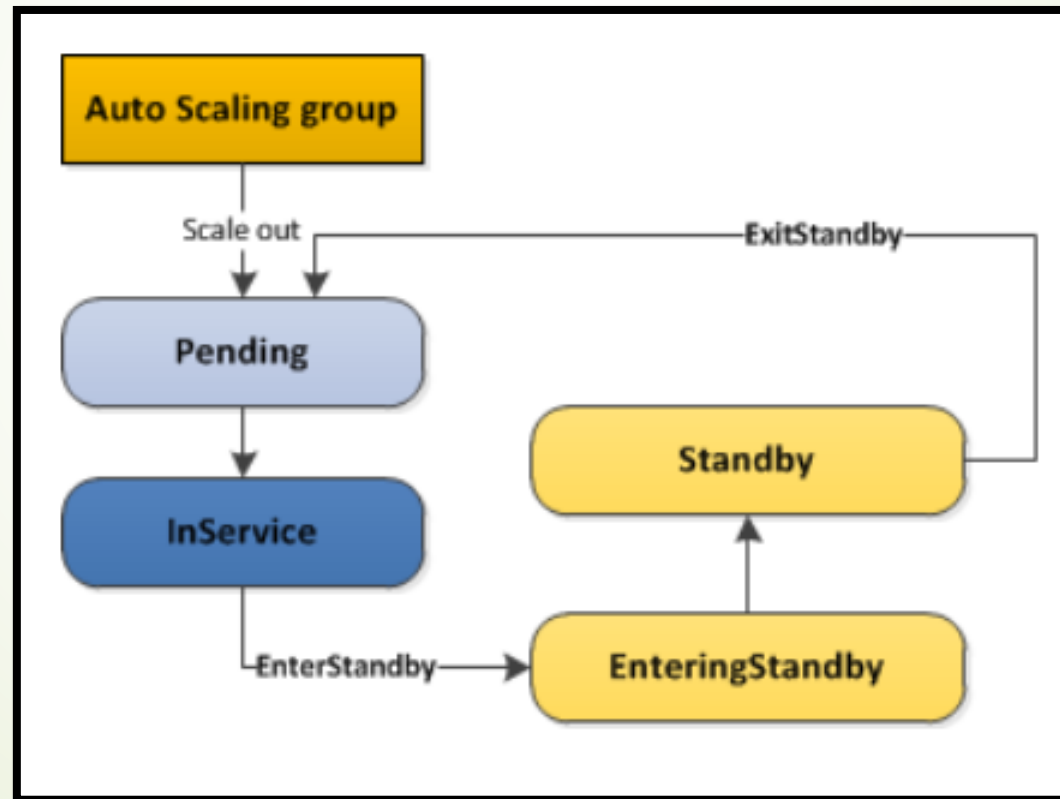
Removing Instances from Your Auto Scaling Group

How the Standby State Works:-

- ✓ You return the instance to service by exiting the standby state.
- ✓ Auto Scaling increments the desired capacity when you put an instance that was on standby back in service. If you did not decrement the capacity when you put the instance on standby, Auto Scaling detects that you have more instances than you need, and applies the termination policy in effect to reduce the size of your Auto Scaling group.
- ✓ If there is a load balancer or target group attached to your Auto Scaling group, the instance is registered with the load balancer or target group.

Removing Instances from Your Auto Scaling Group

Transitions between instance states in this process:-





Removing Instances from Your Auto Scaling Group

Health Status of an Instance in a Standby State:-

- ✓ Auto Scaling does not perform health checks on instances that are in a standby state. While the instance is in a standby state, its health status reflects the status that it had before you put it on standby. Auto Scaling does not perform a health check on the instance until you put it back in service.
- ✓ For example, if you put a healthy instance on standby and then terminate it, Auto Scaling continues to report the instance as healthy. If you return the terminated instance to service, Auto Scaling performs a health check on the instance, determines that it is unhealthy, and launches a replacement instance.



Temporarily Remove an Instance Using the AWS Management Console

Suspending and Resuming Auto Scaling Processes





Suspending and Resuming Auto Scaling Processes

- ✓ Auto Scaling enables you to suspend and then resume one or more of the Auto Scaling processes in your Auto Scaling group. This can be useful when you want to investigate a configuration problem or other issue with your web application and then make changes to your application, without triggering the Auto Scaling process.
- ✓ Auto Scaling might suspend processes for Auto Scaling groups that repeatedly fail to launch instances. This is known as an administrative suspension, and most commonly applies to Auto Scaling groups that have been trying to launch instances for over 24 hours but have not succeeded in launching any instances. You can resume processes suspended for administrative reasons.

Auto Scaling Limits





Auto Scaling Limits

Resource	Default Limit
Launch configurations per region	100
Auto Scaling groups per region	20
Scaling policies per Auto Scaling group	50
Scheduled actions per Auto Scaling group	125
Lifecycle hooks per Auto Scaling group	50
SNS topics per Auto Scaling group	10
Load balancers per Auto Scaling group	50
Target groups per Auto Scaling group	50
Step adjustments per scaling policy	20