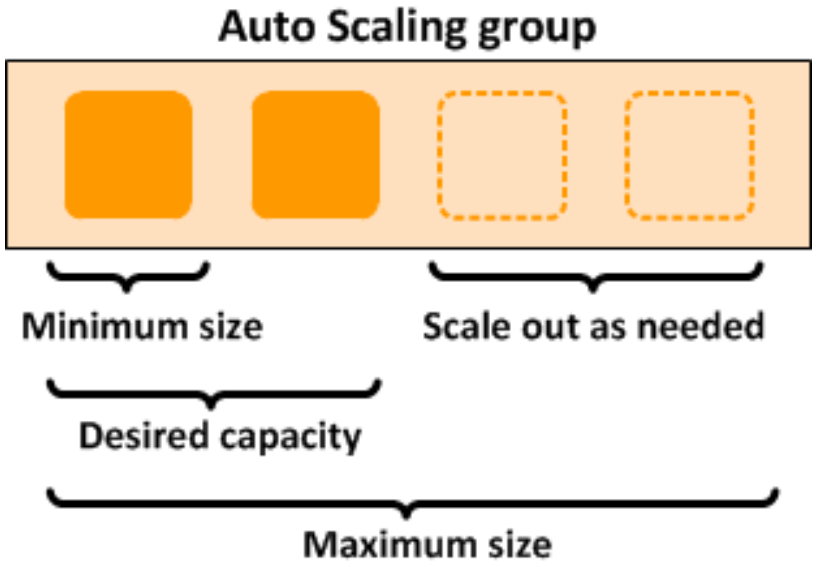
Prior to Cloud computing, there were limited number of servers to handle the application load. When the number of requests increases, the load on the servers also increases which results in poor performance and failures.

AWS provides Amazon EC2 Auto Scaling service to overcome this kind of failure scenario. Auto Scaling ensures that Amazon EC2 instances are sufficient to run our application. We can create an auto-scaling group which contains a collection of EC2 instances. We can specify a minimum number of EC2 instance in that group and auto-scaling service will maintain and ensure the minimum number of EC2 instances. We can also specify a maximum number of EC2 instances in each auto scaling group so that auto-scaling will ensure instances never go beyond that maximum limit.



We can also specify desired capacity and auto-scaling policies for the Amazon EC2 auto-scaling. By using the scaling policy, auto-scaling can launch or terminate the EC2 instances depending on the demand.

Auto Scaling Components

1. Groups

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables us to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service.

The size of an Auto Scaling group depends on the number of instances that we set as the desired capacity. We can adjust its size to meet demand, either manually or by using automatic scaling.

An Auto Scaling group starts by launching enough instances to meet its desired capacity. It maintains this number of instances by performing periodic health checks on the instances in the group. The Auto Scaling group continues to maintain a fixed number of instances even if an instance becomes unhealthy. If an instance becomes unhealthy, the group terminates the unhealthy instance and launches another instance to replace it.

We can use scaling policies to increase or decrease the number of instances in our group dynamically to meet changing conditions. When the scaling policy is in effect, the Auto Scaling group adjusts the desired capacity of the group, between the minimum and maximum capacity values that we specify, and launches or terminates the instances as needed. We can also scale on a schedule.

An Auto Scaling group can launch On-Demand Instances, Spot Instances, or both. We can specify multiple purchase options for our Auto Scaling group only when we configure the group to use a launch template. (AWS recommends that we use launch templates instead of launch configurations to make sure that we can use the latest features of Amazon EC2.)

Spot Instances provides us with access to unused Amazon EC2 capacity at steep discounts relative to On-Demand prices. There are key differences between Spot Instances and On-Demand Instances:

- The price for Spot Instances varies based on demand
- Amazon EC2 can terminate an individual Spot Instance as the availability of, or price for, Spot Instances changes.

When a Spot Instance is terminated, the Auto Scaling group attempts to launch a replacement instance to maintain the desired capacity for the group.

When instances are launched, if we specified multiple Availability Zones, the desired capacity is distributed across these Availability Zones. If a scaling action occurs, Amazon EC2 Auto Scaling automatically maintains balance across all of the Availability Zones that we specify.

2. Launch Configuration

A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When we create a launch configuration, we specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups. If we have launched an EC2 instance before, we have specified the same information in order to launch the instance.

We can specify our launch configuration with multiple Auto Scaling groups. However, we can only specify one launch configuration for an Auto Scaling group at a time, and we cannot modify a launch configuration after we have created it. To change the launch configuration for an Auto Scaling group, we must create a launch configuration and then update our Auto Scaling group with it.

Keep in mind that whenever we create an Auto Scaling group, we must specify a launch configuration, a launch template, or an EC2 instance. When we create an Auto Scaling group using an EC2 instance, Amazon EC2 Auto Scaling automatically creates a launch configuration for us and associates it with the Auto Scaling group.

https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-asg.html

3. Scaling

Scaling is the ability to increase or decrease the compute capacity of your application. Scaling starts with an event or scaling action, which instructs an Auto Scaling group to either launch or terminate Amazon EC2 instances.

Amazon EC2 Auto Scaling provides a number of ways to adjust scaling to best meet the needs of our applications. It is important that we have a good understanding of our application. Keep the following considerations in mind:

- What role should Amazon EC2 Auto Scaling play in our application's architecture? It is common to think about automatic scaling primarily as a way to increase and decrease capacity, but it's also useful for maintaining a steady number of servers.

- What cost constraints are important to us? Because Amazon EC2 Auto Scaling uses EC2 instances, we only pay for the resources that we use. Knowing our cost constraints helps us decide when to scale our applications and by how much.

- What metrics are important to our application? Amazon CloudWatch supports a number of different metrics that we can use with our Auto Scaling group.

Scaling options

Amazon EC2 Auto Scaling provides several ways for us to scale the Auto Scaling group.

Maintain current instance levels at all times

We can configure our Auto Scaling group to maintain a specified number of running instances at all times. To maintain the current instance levels, Amazon EC2 Auto Scaling performs a periodic health check on running instances within an Auto Scaling group. When Amazon EC2 Auto Scaling finds an unhealthy instance, it terminates that instance and launches a new one. For more information, see Maintaining a Fixed Number of Instances in Your Auto Scaling Group.

Scale manually

Manual scaling is the most basic way to scale your resources, where you specify only the change in the maximum, minimum, or desired capacity of your Auto Scaling group. Amazon EC2 Auto Scaling manages the process of creating or terminating instances to maintain the updated capacity. For more information, see Manual Scaling for Amazon EC2 Auto Scaling.

Scale based on a schedule

Scaling by schedule means that scaling actions are performed automatically as a function of time and date. This is useful when you know exactly when to increase or decrease the number of instances in your group, simply because the need arises on a predictable schedule. For more information, see Scheduled Scaling for Amazon EC2 Auto Scaling.

Scale based on demand

A more advanced way to scale your resources, using scaling policies, lets you define parameters that control the scaling process. For example, let's say that you have a web application that currently runs on two instances and you want the CPU utilization of the Auto Scaling group to stay at around 50 percent when the load on the application changes. This method is useful for scaling in response to changing conditions, when you don't know when those conditions will change. You can set up Amazon EC2 Auto Scaling to respond for you. For more information, see Dynamic Scaling for Amazon EC2 Auto Scaling.

Scaling Policy Types

Amazon EC2 Auto Scaling supports the following types of scaling policies:

Target tracking scaling: Increase or decrease the current capacity of the group based on a target value for a specific metric. This is similar to the way that our thermostat maintains the temperature of our home - we select a temperature and the thermostat does the rest.

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: Increase or decrease the current capacity of the group based on a single scaling adjustment.

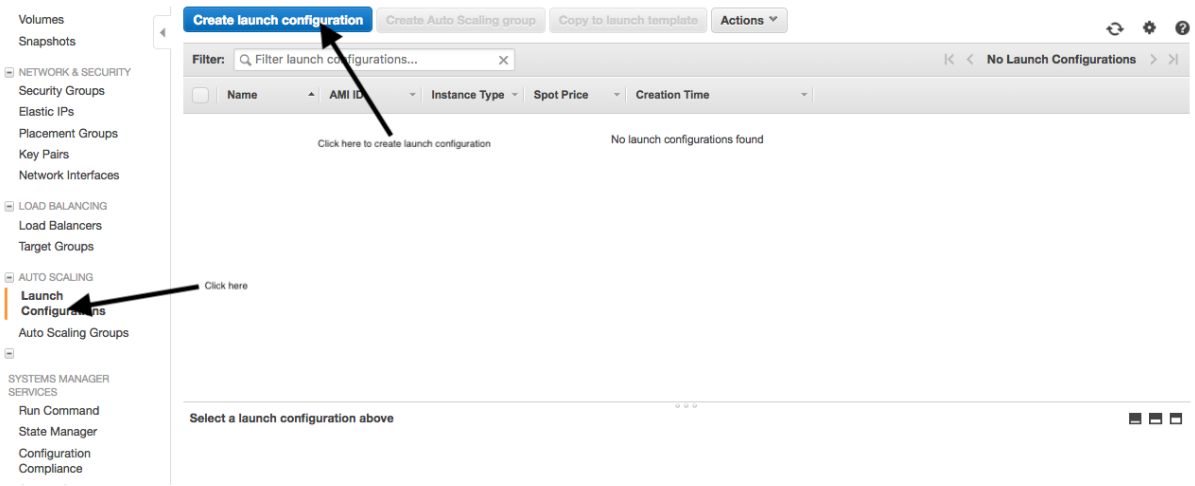
If we are scaling based on a utilization metric that increases or decreases proportionally to the number of instances in an Auto Scaling group, we recommend that we use target tracking scaling policies. Otherwise it is recommended that we use step scaling policies.

Setup

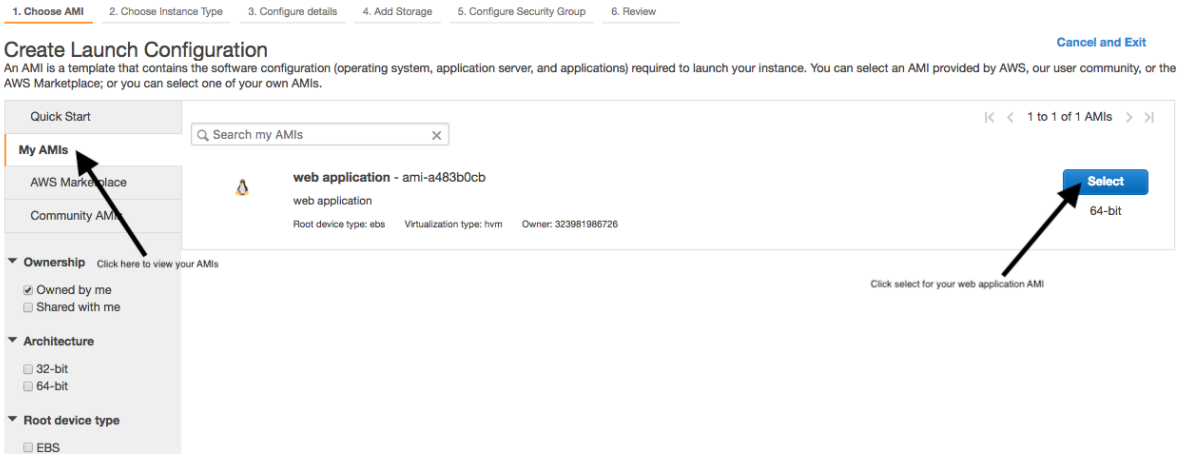
As a pre-requisite, we need to create an AMI of our application which is running on our EC2 instance.

- **Setup: Launch Configuration:**

1. Go to EC2 console and click on Launch Configuration from Auto Scaling

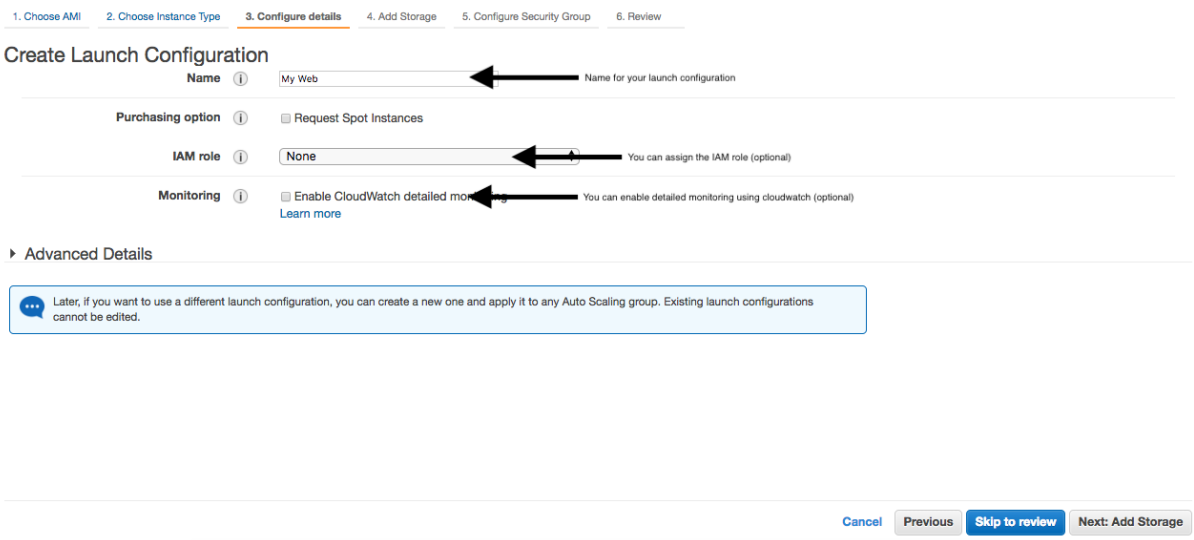


2. From Choose AMI, select the Amazon Machine Image from My AMIs tab, which was used to create the image for our web application.



3. Then, select the instances type which is suitable for our web application and click Next: Configure details.

4. On Configure details, name the launch configuration, we can assign if any specific IAM role is assigned for our web application and we can enable the detailed monitoring.



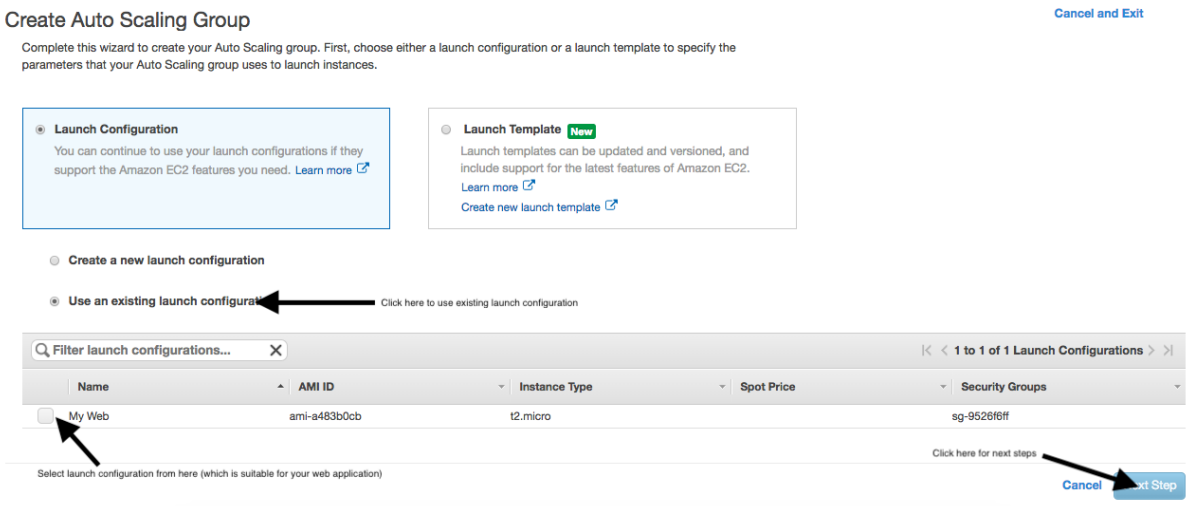
5. After that, Add the storage and Security Groups, then go for review. Note: Open the required ports for our application to run.

6. Click on Create launch configuration and choose the existing key pair or create new key pair

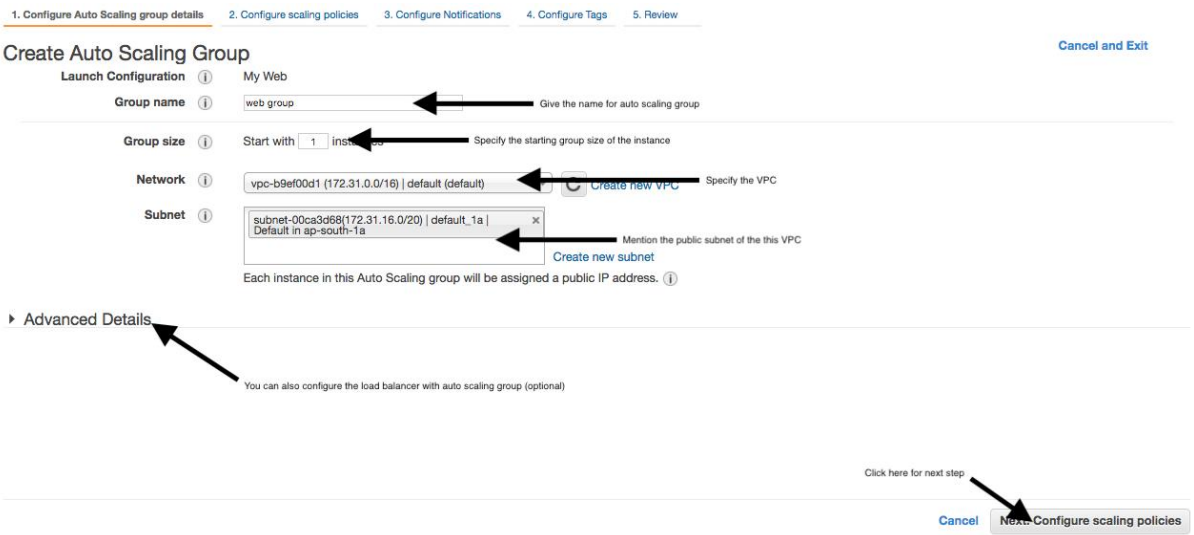
Setup: Auto Scaling Group:

1. From EC2 console, click on Auto Scaling Group which is below the launch configuration. Then click on create auto scaling group.

2. From Auto scaling Group page, we can create either using launch configuration or Launch Template. Here we have created using Launch Configuration. We can create new Launch Configuration from this page also. Since we had already created the launch configuration, we can go for creating auto scaling group by using “Use a existing launch configuration”.

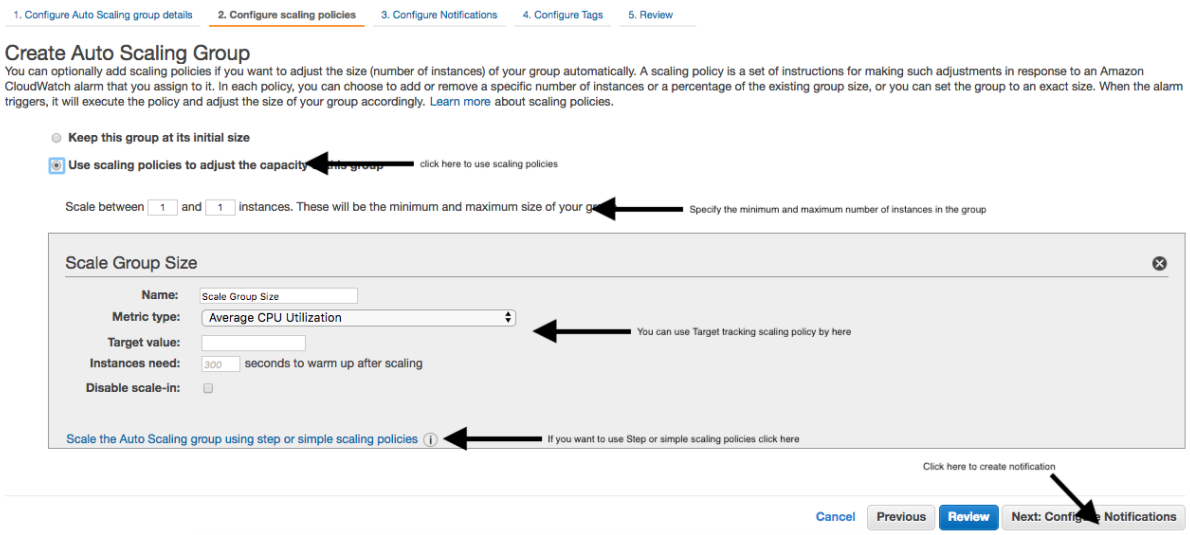


3. After clicking on next step, we can configure group name, group initial size, and VPC and subnets. We can also configure load balance with auto scaling group by clicking Advanced Details.

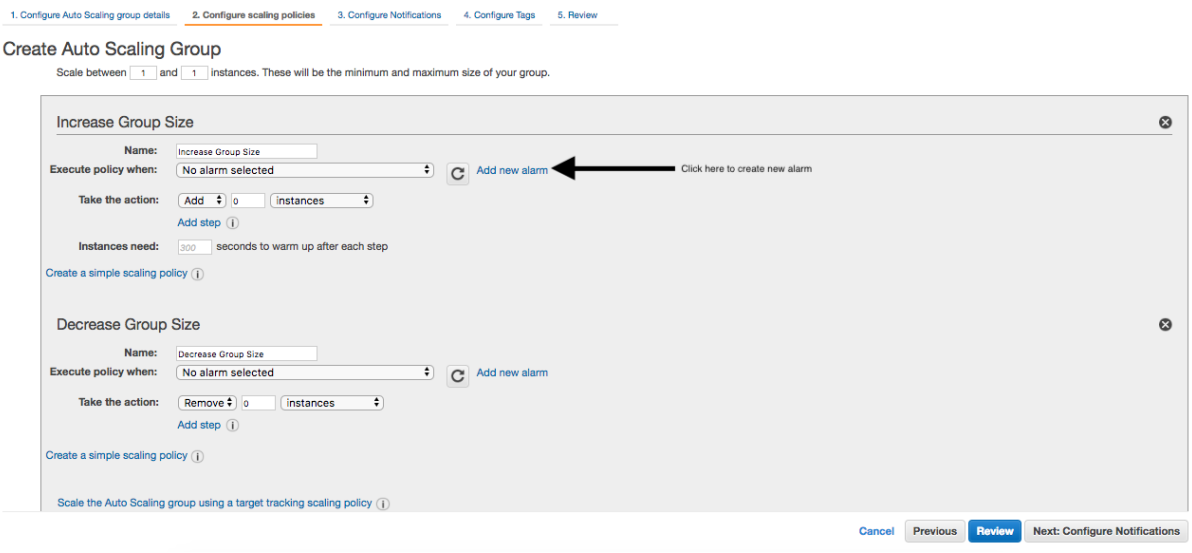


After that click on next to configure scaling policies

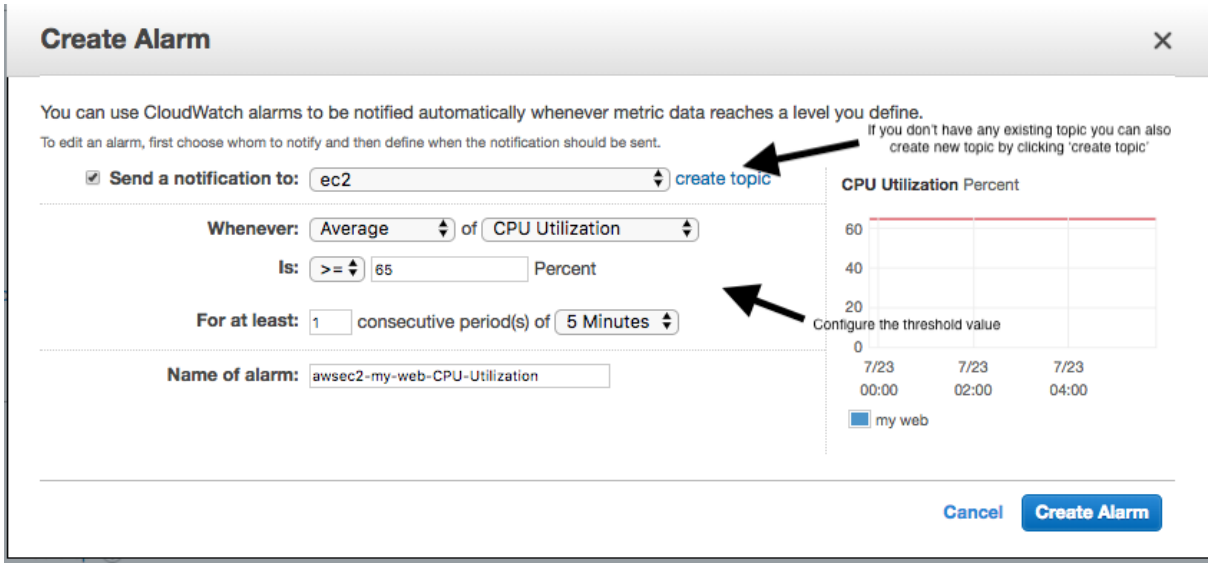
4.On scaling policy page, we can specify the minimum and maximum number of instances in this group. Here we can use target tracking policy to configure the scaling policies. In metric type we can specify such as CPU utilization and Network In or Out and we can give the target value as well. Depending on the target value the scaling policy will work. We can also disable scale-in from here.



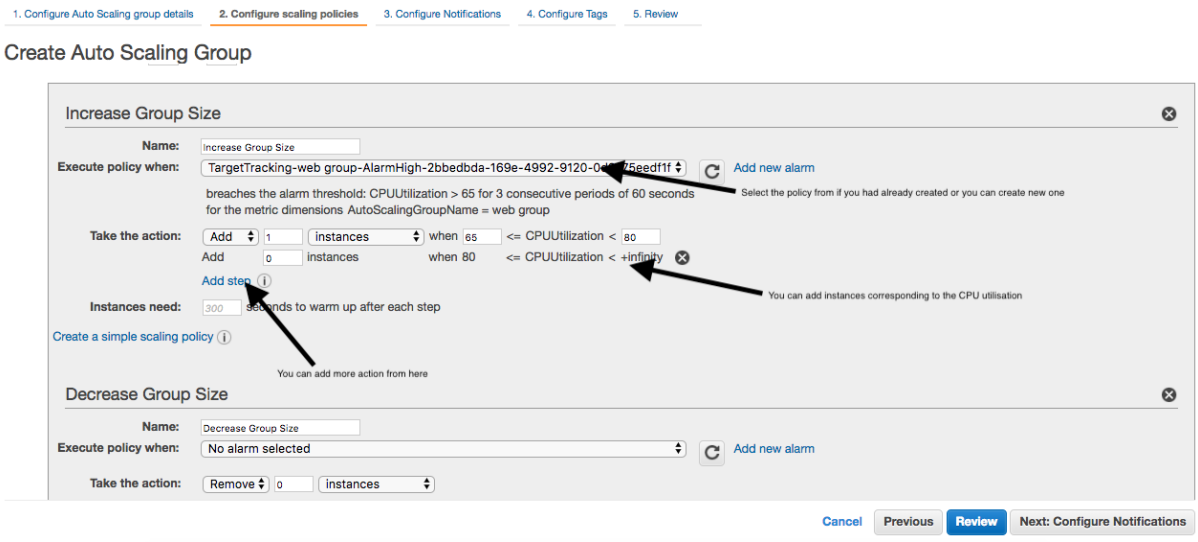
We can use Step and Simple scaling policies.



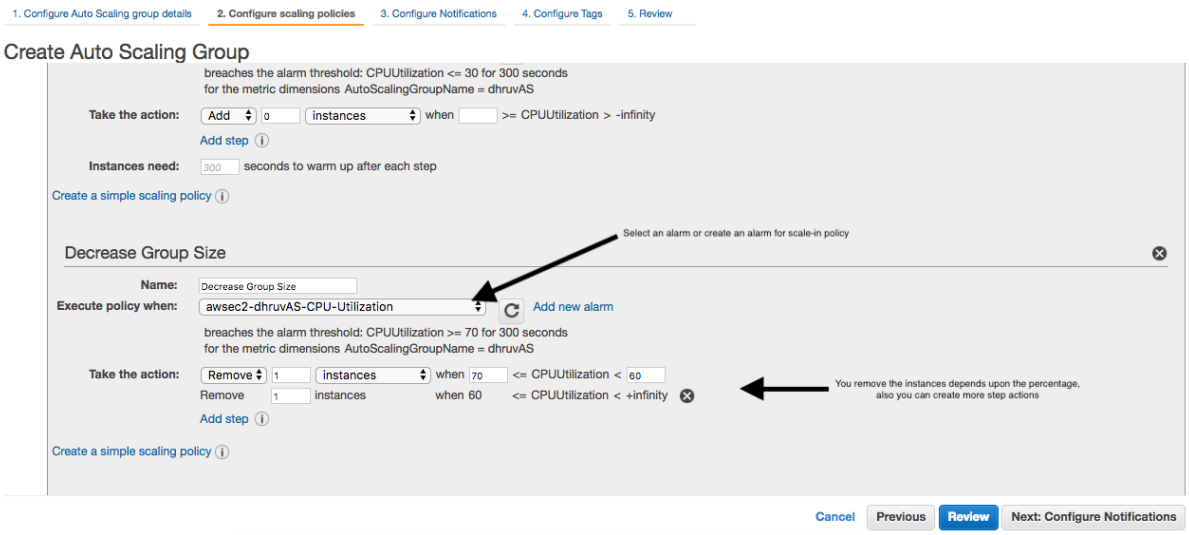
It works based on alarm, so first create the alarm by clicking on 'add new alarm'.



Here the alarm created is based on CPU utilization above 65%. If CPU utilization crosses 65% the auto scaling launches new instances based on the step action.



We can specify more step actions based on our load, but in simple policy you can't categorize depending on the percentage of CPU utilization. Also, we need to configure scale-in policies once the traffic become low, as it reduces the billing.



5. Next click on ‘Next: Configure Notification’ to get the notification based on launch, terminate, and fail etc. to your mail ID, and enter the tag and click on ‘Create auto scaling group’.

Note: We need to create an Elastic Load balancer on top of an auto scaling group. Please refer to the Elastic Load Balancing topic for more information.