**EXPERIMENT-3**

**Show how Autoscaling can be performed on EC2 instances( Scale manually & Scale on demand)**

**QUESTIONS**

**Ques-1:** List the key components of Autoscaling.

**Ans:** The key components of autoscaling in Amazon Web Services (AWS) are:

**1)Auto Scaling Group:** An Auto Scaling group is a collection of Amazon Elastic Compute Cloud (EC2) instances that are created and managed together. The group can automatically scale the number of instances based on demand.

**2)Launch Configuration:** A launch configuration is a blueprint that defines the EC2 instances to be launched, including the Amazon Machine Image (AMI), instance type, and storage volumes. The launch configuration acts as a template for creating new EC2 instances in the Auto Scaling group.

**3)Scaling Options:** EC2 Auto Scaling provides several options for automatically scaling the number of EC2 instances based on demand. These options include target tracking scaling, step scaling, scheduled scaling, dynamic scaling, and manual scaling. Each option has its own set of rules and parameters for scaling the number of instances in response to changes in demand.

In summary, the Auto Scaling group, launch configuration, and scaling options work together to automatically scale the number of EC2 instances in response to changes in demand, ensuring that your applications have the resources they need to operate effectively and efficiently.

**Ques-2:** Differentiate between Launch Template and Launch Configuration. Which is recommended and Why?

**Ans:** Launch Configuration and Launch Template are both used to create Amazon EC2 instances in AWS Auto Scaling groups, but there are some key differences between the two.

A Launch Configuration is a blueprint that describes all the settings for an Amazon EC2 instance. It includes information such as the Amazon Machine Image (AMI), instance type, key pair, and security group. A Launch Configuration is used to create a new Amazon EC2 instance, and it can only be used to launch a single instance at a time.

A Launch Template is a more flexible and powerful alternative to a Launch Configuration. A Launch Template includes all the same information as a Launch Configuration, but it can also include additional parameters such as block device mappings, network interfaces, and tags. Additionally, a Launch Template can be used to launch multiple instances at once.

AWS recommends using Launch Templates over Launch Configuration for several reasons:

**1)**Launch Templates provide more flexibility and control over the instances that are launched.

**2)**Launch Templates can be versioned, so you can easily roll back to a previous version if needed.

**3)**Launch Templates can be shared across multiple Auto Scaling groups and accounts, making it easier to manage and maintain.

**4)**Launch Templates provide more detailed control over the instances, such as Network interfaces and block device mappings.

In summary, Launch Templates provide more advanced features and flexibility than Launch Configuration, and it is recommended to use them for more complex use cases, especially when you need versioning and sharing capabilities.

**Ques-3:** Explain the different AutoScaling options available in EC2 with the help of suitable options.

**Ans:** Amazon EC2 Auto Scaling provides several options to automatically scale the number of EC2 instances based on demand. These options include:

**1)Target Tracking Scaling:** This option adjusts the number of instances based on a specified metric, such as CPU utilization or network traffic, relative to a target value. The Auto Scaling group automatically adjusts the number of instances to maintain the target value.

**2)Step Scaling**: This option adjusts the number of instances based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the increase or decrease in demand. This option allows you to specify different scaling adjustments for different ranges of the metric.

**3)Scheduled Scaling**: This option adjusts the number of instances based on a schedule that you define. This is useful for anticipating predictable changes in demand, such as during business hours or on weekends.

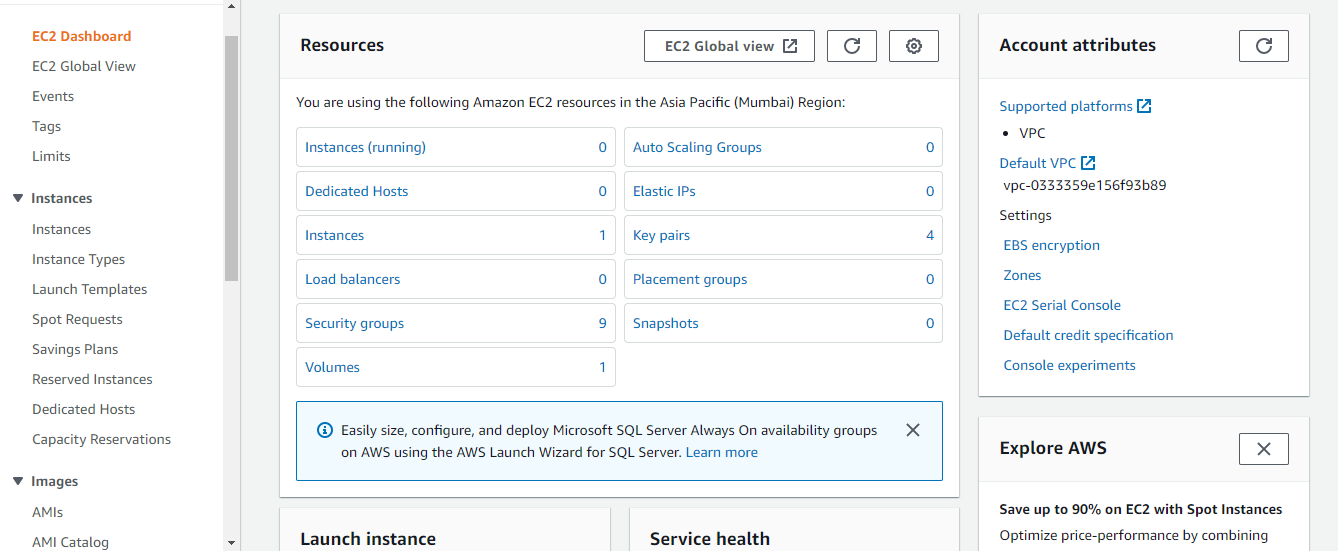
**4)Dynamic Scaling**: This option adjusts the number of instances based on a custom metric that you provide. This is useful for scaling based on custom application-specific metrics, such as the number of active users or the size of a database.

**5)Manual Scaling**: This option allows you to manually adjust the number of instances in your Auto Scaling group. This is useful for making one-time changes to the number of instances, or for testing purposes.

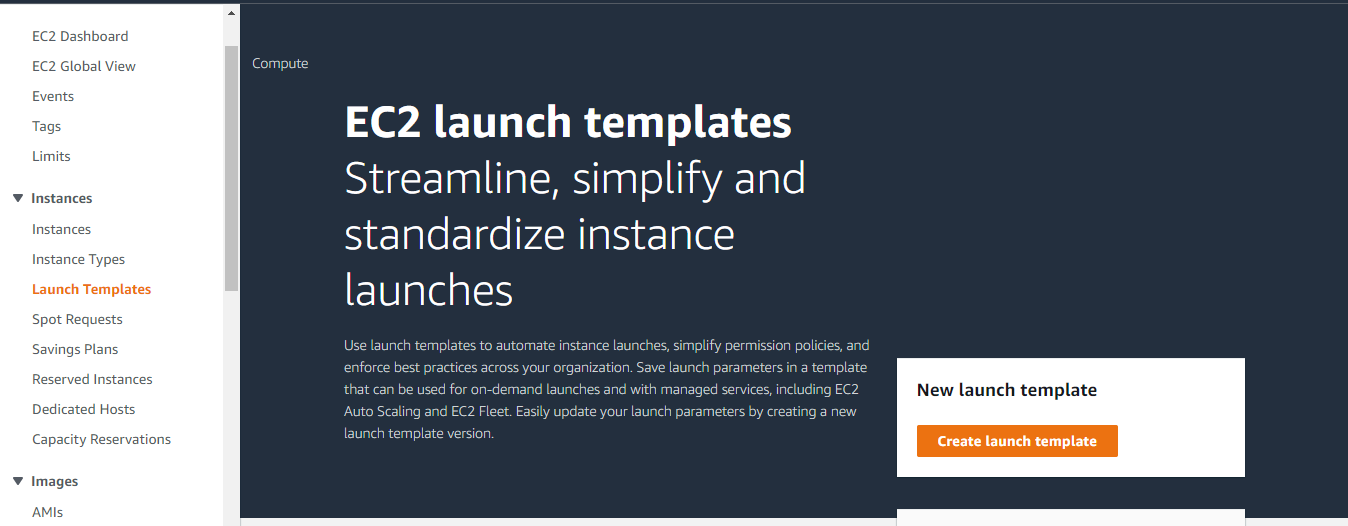
In addition to these scaling options, EC2 Auto Scaling also provides features such as health check replacement, which automatically replaces any instances that become unhealthy, and scale-out protection, which helps prevent over-scaling by adding a limit on the maximum number of instances that can be added during a scaling event.

**STEPS FOR AUTOSCALING USING LINUX AMI MANUALLY AND ON DEMAND**

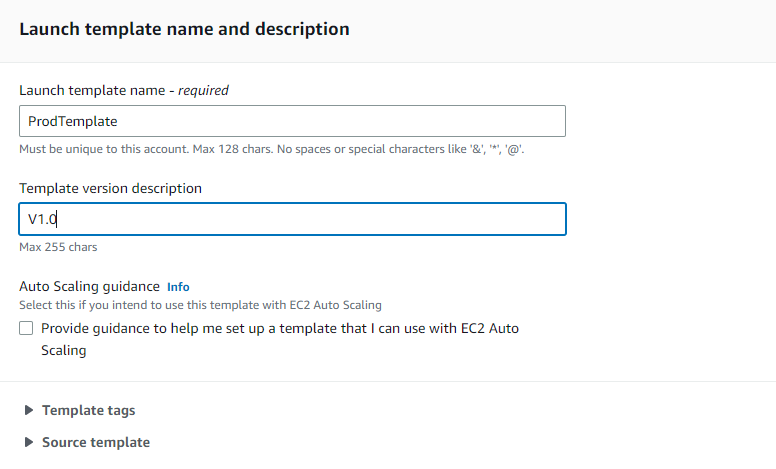
**Step-1:** Login to the AWS console andgo to EC2 dashboard.



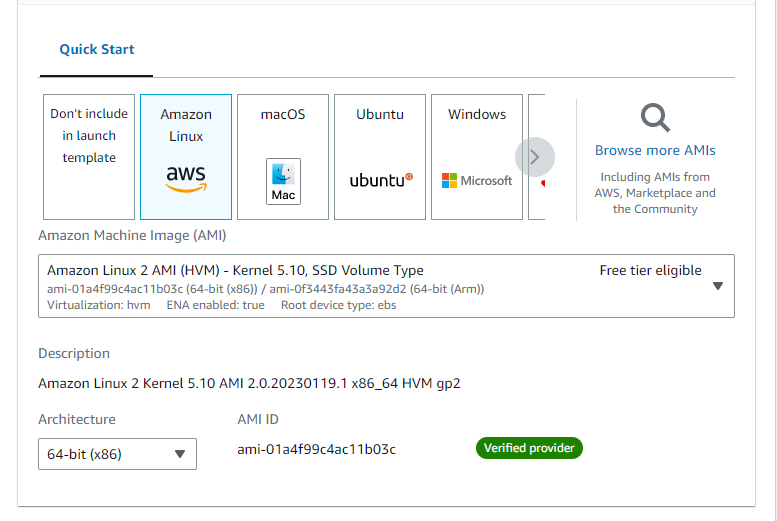
**Step-2:** Now go to launch templates and create new launch template.



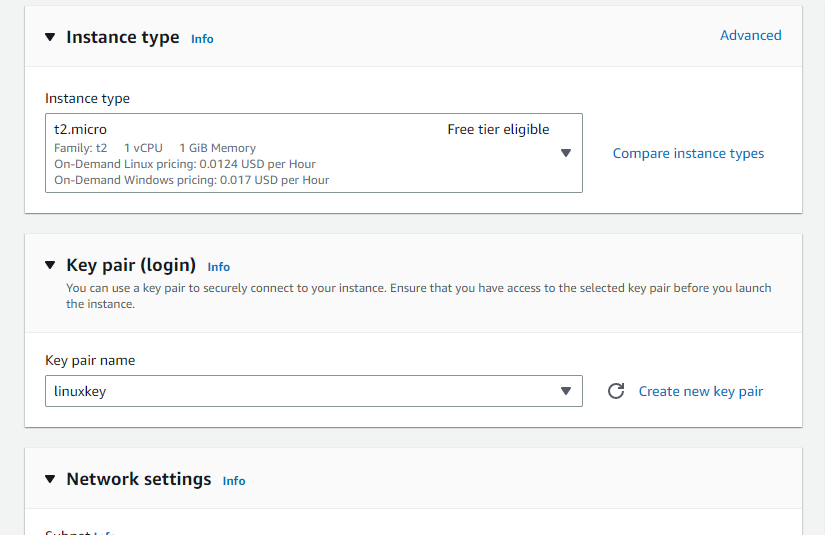
**Step-3:** Now give the launch template name and version description.



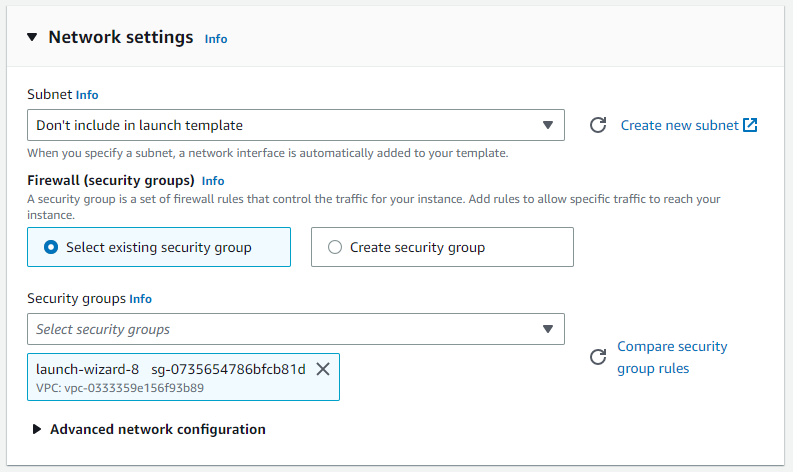
**Step-4:** Now choose the AMI you want to choose for launch template.



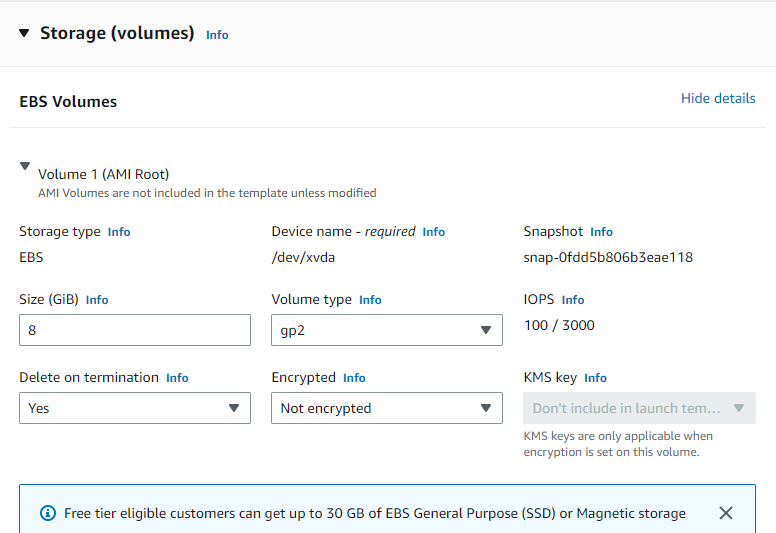
**Step-5:** In this step, choose the instance type and the key pair(existing or make a new pair).



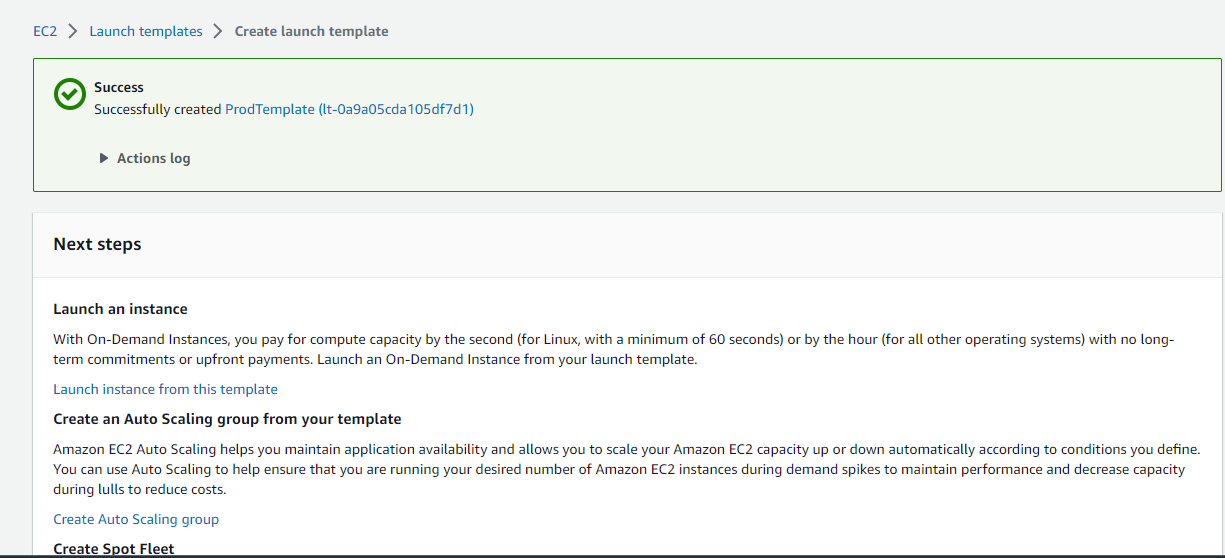
**Step-6:**Choose the network settings and security groups.



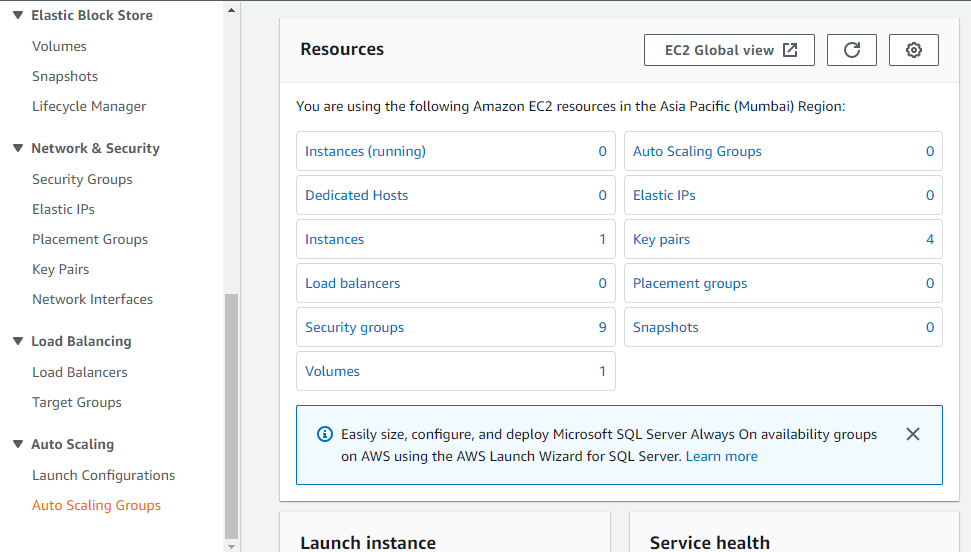
**Step-7:** Choose the storage size and type and click on launch template.



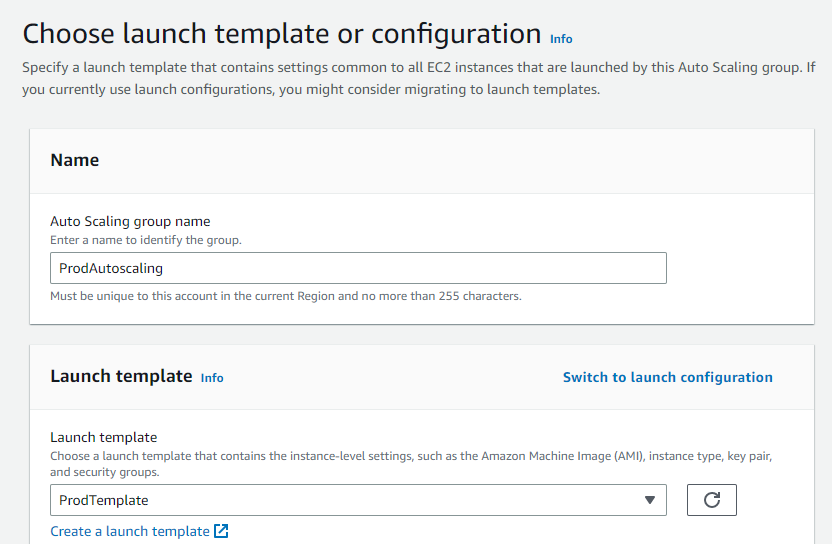
**Step-8: ‘**Prodtemplate’ has been successfully created.



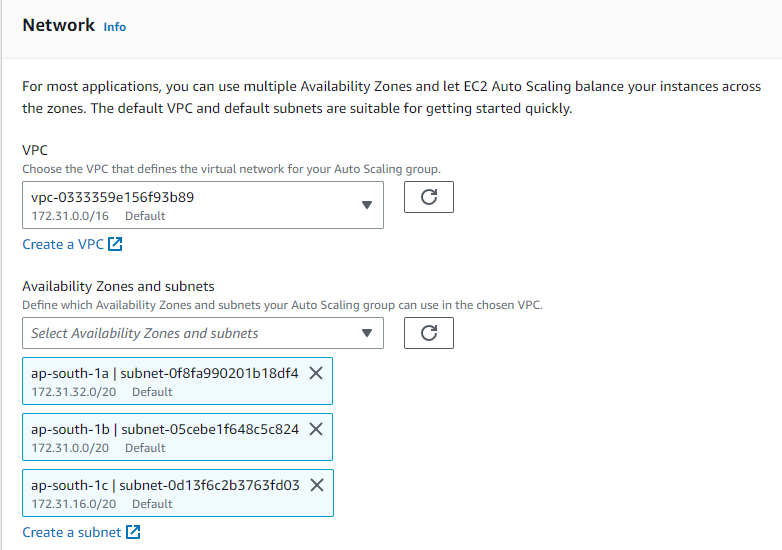
**Step-9:** Now go to EC2 dashboard again and click on auto scaling group.



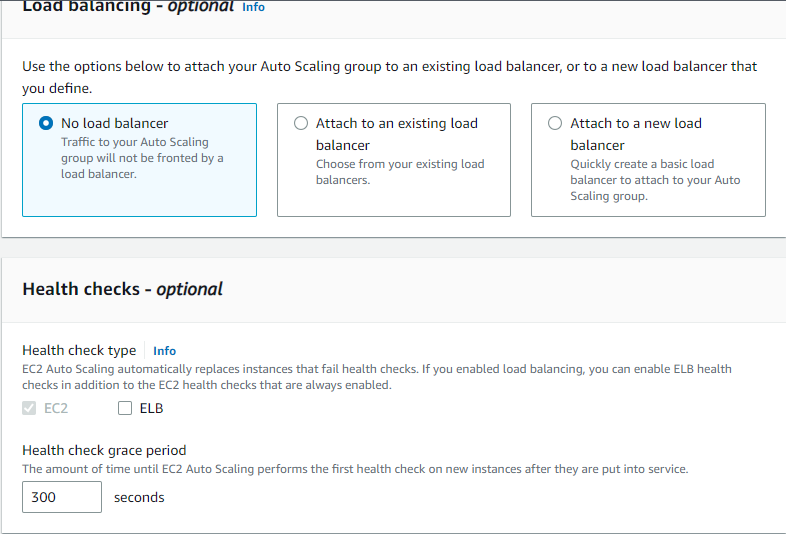
**Step-10:** Name the autoscaling group and select the launch template that we have created.



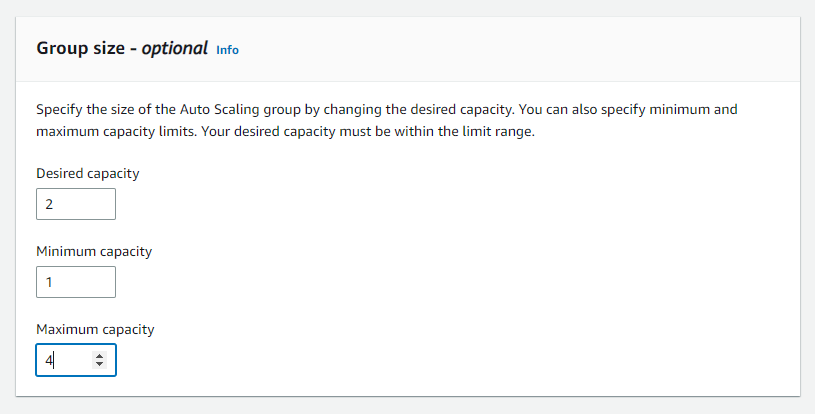
**Step-11:** Select the availabilities zones and proceed.



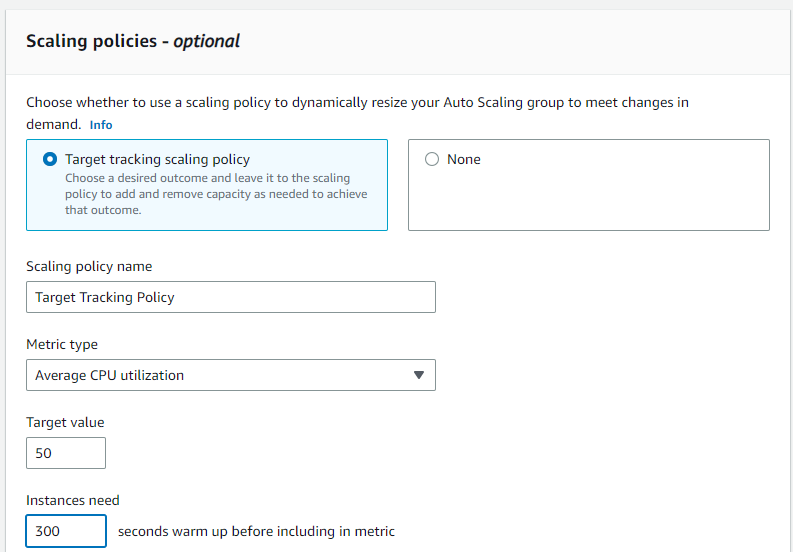
**Step-12:** Select for no load balancer and proceed.



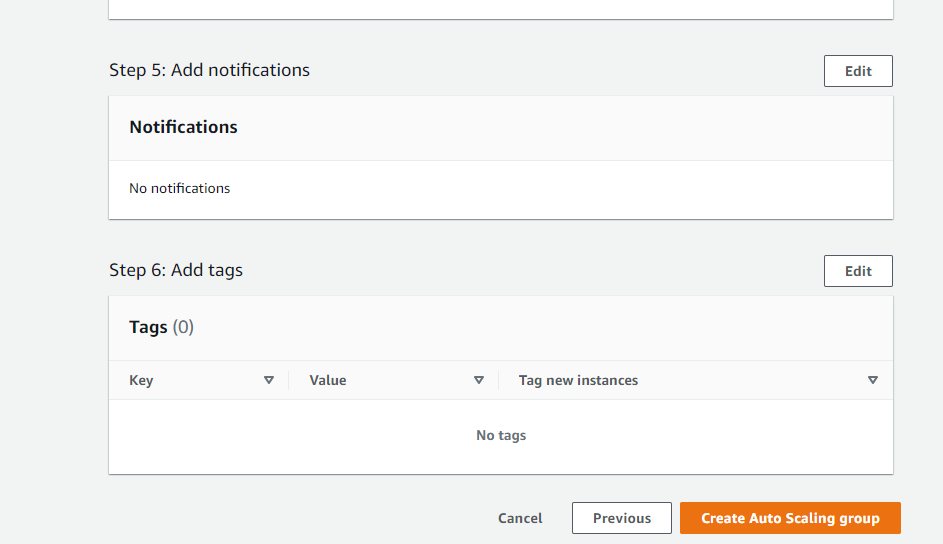
**Step-13:** Now select the desired capacity as 2, minimum capacity as 1 and maximum capacity as either 3 or 4.



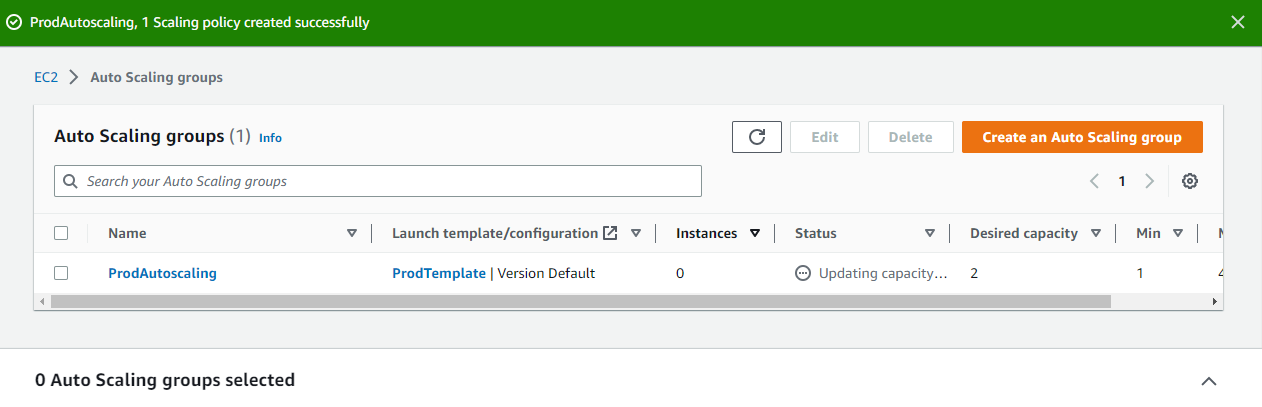
**Step-12:** Select for the target tracking scaling policy and fill the instances need and target value.



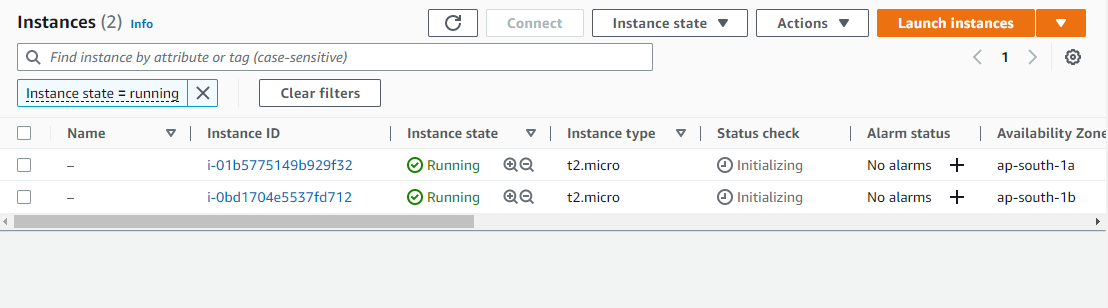
**Step-13:** No notifications and tags as of now and click on create auto scaling group.



**Step-14:**Autoscaling group, “ProdAutoscaling" has been successfully created.



**Step-15:**Now go to the instances you can see that the desired number of instances i.e. 2 have been created and are running.



**Step-16:** Now we will do stress testing. Install stress on the linux ami 2 using the following commands.

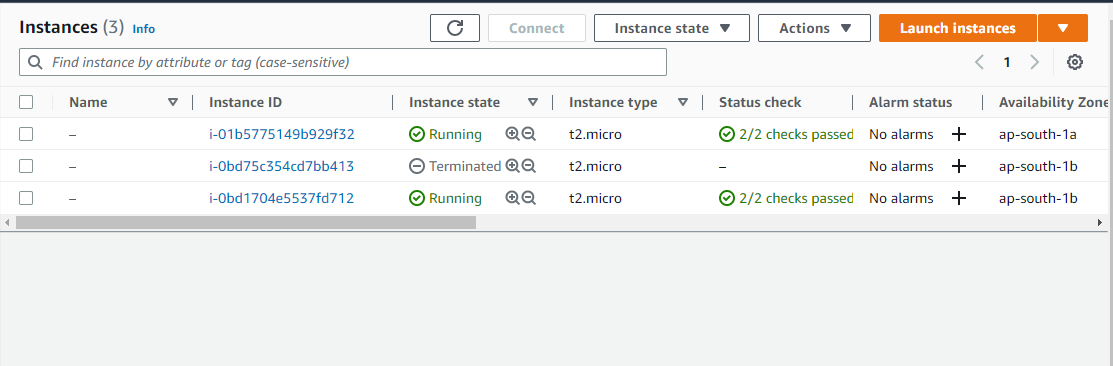
sudo amazon-linux-extras install epel -y

sudo yum install stress -y

**Step-17:** Now increase the load on your cpu using the following command

‘sudo stress --cpu 8 --vm-bytes $(awk '/MemAvailable/{printf "%d\n", $2 \* 0.9;}' < /proc/meminfo)k --vm-keep -m 1’

**Step-18:** Now we can see that an additional instance has been created when the load on the cpu is increased and the among the two desired instances, one is terminated.



**STEPS FOR AUTOSCALING USING WINDOWS AMI MANUALLY AND ON DEMAND**

