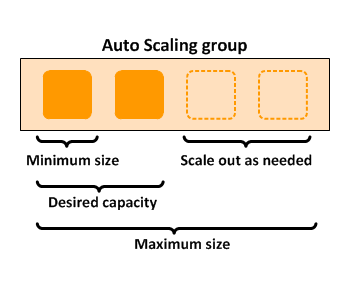
**Domain: Deployment and Provisioning**

**Topic:** EC2 Autoscaling groups

**Service description:**

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. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 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 Amazon EC2 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, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, the following Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.



**Use cases:** You should use EC2 Auto Scaling if you only need to scale Amazon EC2 Auto Scaling groups, or if you are only interested in maintaining the health of your EC2 fleet. You should also use EC2 Auto Scaling if you need to create or configure Amazon EC2 Auto Scaling groups, or if you need to set up scheduled or step scaling policies (as AWS Auto Scaling supports only target tracking scaling policies).

Adding Amazon EC2 Auto Scaling to your application architecture is one way to maximize the benefits of the AWS Cloud. When you use Amazon EC2 Auto Scaling, your applications gain the following benefits:

* Better fault tolerance. Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Amazon EC2 Auto Scaling can launch instances in another one to compensate.
* Better availability. Amazon EC2 Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.
* Better cost management. Amazon EC2 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 needed and terminating them when they aren't.

**Limits:** Your AWS account has the default quotas , for Amazon EC2 Auto Scaling Groups, Scaling policy quotas and API-specific limits

To view the current quotas for your account, open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/> and navigate to the **Limits** page. You can also use the [describe-account-limits](https://docs.aws.amazon.com/cli/latest/reference/autoscaling/describe-account-limits.html) command. To request an increase, use the [Auto Scaling Limits form](https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-auto-scaling).

**Task: Creating** EC2 Autoscaling groups from Launch template

**Problem to Be Solved**

You are a System Engineer witch has to administrate online supermarket. Next month you are expected big sales in your company and this will generate huge traffic on the web site and overload it. So in this task you will learn how create ES2 Autoscaling group from Launch template witch will scale up and down instances (web servers) group depending on the instance CPU load.

**NOTICE**: in the future task you will configure Application Load Balancer to load balance between the instance inside the Autoscaling group.

**Explanation of the Solution**

[*https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html*](https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html)

[*https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html*](https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html)

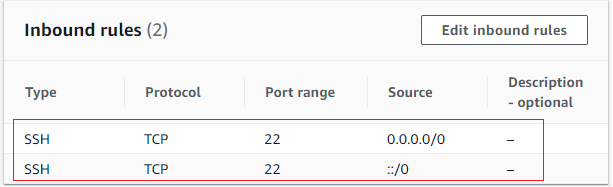
[*https://docs.aws.amazon.com/autoscaling/ec2/userguide/GettingStartedTutorial.html*](https://docs.aws.amazon.com/autoscaling/ec2/userguide/GettingStartedTutorial.html)

**Implementation Details**

**Required**: Key Pairs, default VPC and default Subnets (check if they exist).

Preset new Security group: Navigate to **EC2 > Instances** > **Security Groups,** clck “**Create Security group**”. Set Security group name “**SSHSecurityGroup**”, Descripion “allows ssh” and in VPC choose “default”.

In “Inbound rules” section click “**Add rule**” and enter the next settings – Type: **SSH**; Protocol: **TCP** (default); Port range: **22** (default); Source type: **Anywhere**; Source: **0.0.0.0/0** and **::/0** (default). Press “**Create Security group**”. After this you will see in inbound rules records like this:



These rules will allow you SSH connect into EC2 instances.

1. Create a Launch Template
2. Navigate to **EC2 > Instances** > **Launch Templates**.
3. Click **Create launch template**, call it “**MyLaunchTemplate**”, and mark **checkbox Auto Scaling guidance** .
4. Select for “AMI” **Amazon Linux 2 AMI (HVM) (64-bit x86) image.**
5. Set the instance type as **t2.micro**.
6. Select the key pair you created earlier or create new.
7. **Network settings set VPC and “SSHSecurityGroup” security group.**
8. Storage will be automatically set, don’t change it.
9. Expand **Advanced Details**, and paste in the **User data** provision script to install app. For example: “sudo yum –y install httpd”
10. Click **Create Launch Template**.
11. Create a Autoscaling group
12. Navigate to **EC2**  > **Auto Scaling** > **Auto Scaling Groups.**
13. Click **Create an Auto Scaling group**.
14. Call the group **MyASG**.
15. Select **Launch Template**, and choose the template you just created “**MyLaunchTemplate**”.
16. Select **Adhere to Launch Template**.
17. Version will set by “**Default (1)**”, don’t change it.
18. Click “**Next**”.
19. Set Network VPC “default”, and one default Subnets.
20. Click “**Next**”.
21. Load balance optional set to “**No load balancer**”.
22. Health check don’t change.
23. Additional setting don’t change.
24. Click “**Next**”.
25. In Group Size enter next

*Desired Capacity*: **1**

*Minimum Capacity*: **1**

*Maximum Capacity*: **2**

1. Scaling policies set to “**Target tracking scaling policy**”, don’t change “**Scaling policy name”**, “**Metric type”** and **“Target value”**.
2. Click “**Next**”.
3. Don’t change Notifications
4. Click “**Next**”.
5. Click “**Next**”.
6. Click “**Create Auto Scaling group**”.
7. Test Horizontal Scaling
8. Connect to one of the EC2 instances via SSH.
9. Install stress test utility, run

sudo amazon-linux-extras install epel -y

sudo yum install -y stress

1. Run stress test

stress --cpu 2 --timeout 300

1. Wait 5-10 minutes, go to **EC2 > Auto Scaling groups > MyASG** click “**Monitoring**” then “**EC2**” and see “**CPU Utilization**” graph. Then check on the page **EC2**  > **Instance and check the** number of Runnig Instance.

**Benefits / Outcomes / Pros and Cons / Summary**

**(say a few words about how ASGs connected with HA)**

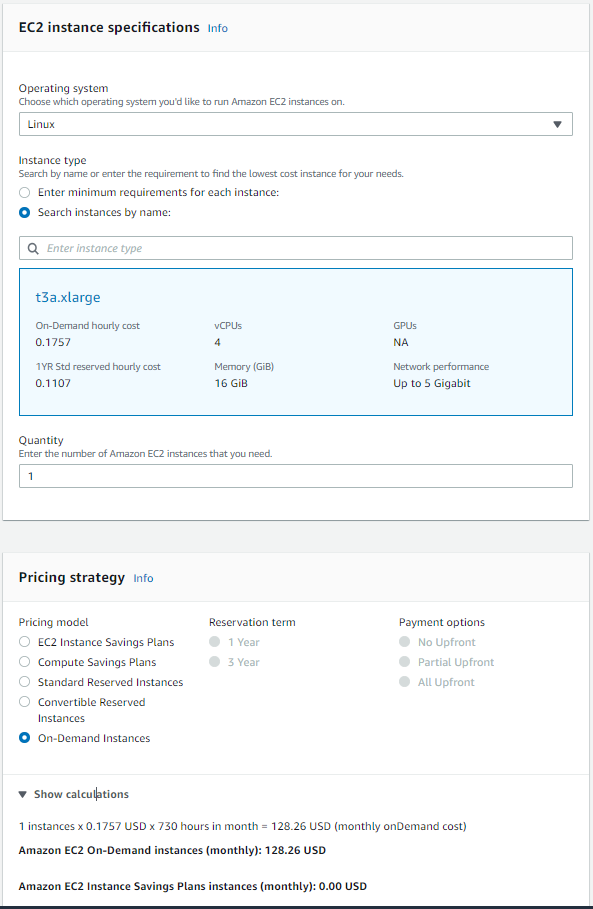
*In this task you known how to create Auto scaling group and how to manage the size of the Auto Scaling group by changing tracking scaling policy (CPU Utilisation).*

**Pricing**

*There are no additional fees with Amazon EC2 Auto Scaling, so it's easy to try it out and see how it can benefit your AWS architecture. You only pay for the AWS resources (for example, EC2 instances and CloudWatch alarms) that you use.*

*Link to EC2 cost calculator*: <https://calculator.aws/#/createCalculator/EC2>

 For example pricing of EC2 t3a.xlarge:



**Tearing down**

1. Delete Auto Scaling Ggroups
2. Navigate to **EC2**  > **Auto Scaling** > **Auto Scaling Groups.**
3. Mark “**MyASG**” group and click “**Delete**”. On the next page type in the field “**delete**” and click “**Delete**”.
4. Delete **Launch Templates**
5. Navigate to **EC2**  > **EC2 > Instances** > **Launch Templates.**
6. Mark “**MyLaunchTemplate**” click on “Action” and set “**Delete template**”. On the next page type in the field “**delete**” and click “**Delete**”.