**Menu**

# 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](http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/GettingStartedTutorial.html#gs-create-lc)
* [Step 2: Create an Auto Scaling Group](http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/GettingStartedTutorial.html#gs-create-asg)
* [Step 3: Verify Your Auto Scaling Group](http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/GettingStartedTutorial.html#gs-verify-asg)
* [Step 4: (Optional) Delete Your Auto Scaling Infrastructure](http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/GettingStartedTutorial.html#gs-delete-asg)

## 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.

**To create a launch configuration**

1. Open the Amazon EC2 console.
2. In the navigation pane, under Auto Scaling, click Launch Configurations.
3. Select a region. The Auto Scaling resources that you create are tied to the region you specify and are not replicated across regions.
4. On the Welcome to Auto Scaling page, click Create Auto Scaling group.
5. On the Create Auto Scaling Group page, click Create launch configuration.
6. On the Choose AMI page displays a list of basic configurations, called Amazon Machine Images (AMIs), that serve as templates for your instance. Select the 64-bit Amazon Linux AMI.
7. On the Choose Instance Type page, select a hardware configuration for your instance. We recommend that you use the t2.micro instance that is selected by default. Click Next: Configure details.
8. On the Configure Details page, do the following:
   1. In the Name field, enter a name of your launch configuration (for example, my-first-lc).
   2. Under Advanced Details, select an IP address type. If you want to connect to an instance in a VPC, you must select an option that assigns a public IP address. If you want to connect to your instance but aren't sure whether you have a default VPC, select Assign a public IP address to every instance.
   3. Click Skip to review.
9. On the Review page, click Edit security groups, follow the instructions to choose an existing security group, and then click Review.
10. On the Review page, click Create launch configuration.
11. In the Select an existing key pair or create a new key pair dialog box, select one of the listed options. Note that you won't connect to your instance as part of this tutorial. Therefore, you can select Proceed without a key pair unless you intend to connect to your instance.
12. Click Create launch configuration to create your launch configuration.

## Step 2: Create an Auto Scaling Group

Auto Scaling groups are the core of the Auto Scaling service. An Auto Scaling group is a collection of EC2 instances. 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.

**To create an Auto Scaling group**

1. On the Configure Auto Scaling group details page, do the following:
   1. In Group name, enter a name for your Auto Scaling group (for example, my-first-asg.
   2. Leave Group size set to the default value of 1 instance for this tutorial.
   3. If you are launching a t2.micro instance, you must select a VPC in Network.
   4. If you selected a VPC in the previous step, select a subnet from Subnet. If you selected EC2-Classic in the previous step, select an Availability Zone from Availability Zone(s).
   5. Click Next: Configure scaling policies.
2. In the Configure scaling policies page, select Keep this group at its initial size for this tutorial and click Review.
3. On the Review page, click Create Auto Scaling group.
4. On the Auto Scaling group creation status page, click Close.

## 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 your EC2 instance.

**To verify that your Auto Scaling group has launched your EC2 instance**

1. On the Auto Scaling Groups page, select the Auto Scaling group that you just created.
2. The Details tab provides information about the Auto Scaling group.
3. Select the Scaling History tab. The Status column contains the current status of your instance. When your instance is launching, the status column shows In progress. The status changes to Successful after the instance is launched. You can also click the refresh button to see the current status of your instance.
4. Select the Instances tab. The Lifecycle column contains the state of your newly launched instance. You can see that your Auto Scaling group has launched your EC2 instance, and it is in the InService lifecycle state. The Health Status column shows the result of the EC2 instance health check on your instance.

f

1. (Optional) If you want, you can try the following experiment to learn more about Auto Scaling. The minimum size for your Auto Scaling group is 1 instance. Therefore, if you terminate the running instance, Auto Scaling must launch a new instance to replace it.
   1. On the Instances tab, click the ID of the instance. This takes you to the Instances page and selects the instance.
   2. Click Actions, select Instance State, and then click Terminate. When prompted for confirmation, click Yes, Terminate.
   3. In the navigation pane, select Auto Scaling Groups and then select the Scaling History tab. The default cooldown for the Auto Scaling group is 300 seconds (5 minutes), so it takes about 5 minutes until you see the scaling activity. When the scaling activity starts, you'll see an entry for the termination of the first instance and an entry for the launch of a new instance. The Instances tab shows the new instance only.
   4. In the navigation pane, select Instances. This page shows both the terminated instance and the running instance.

Go to the next step if you would like to delete your Auto Scaling set up. Otherwise, you can use this Auto Scaling infrastructure as your base and try one or more of the following:

## 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.

**To delete your Auto Scaling group**

1. Open the Amazon EC2 console.
2. In the navigation pane, under Auto Scaling, click Auto Scaling Groups.
3. On the Auto Scaling groups page, select your Auto Scaling group (for example, my-first-asg).
4. Click Actions and then click Delete. When prompted for confirmation, click Yes, Delete.

The Name column indicates that the Auto Scaling group is being deleted. The Desired, Min, and Max columns shows 0 instances for the Auto Scaling group.

Skip this procedure if you would like keep your launch configuration.

**To delete your launch configuration**

1. In the navigation pane, under Auto Scaling, click Launch Configurations.
2. On the Launch Configurations page, select your launch configuration (for example, my-first-lc).
3. Click Actions and select Delete launch configuration. When prompted for confirmation, click Yes, Delete.

Refer <http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/how-as-works.html#arch-AutoScalingMultiAZ>

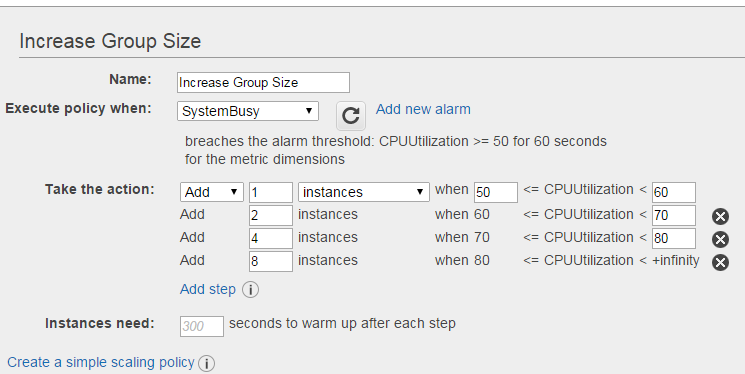
for more information

New Updates to AUTOSCALINEG:[ Jul 7, 2015] Extract from <https://aws.amazon.com/blogs/aws/auto-scaling-update-new-scaling-policies-for-more-responsive-scaling/>

New Scaling Policies With Steps  
Today we are making Auto Scaling even more flexible with the addition of new scaling policies with steps.

Our goal is to allow you to create systems that can do an even better job of responding to rapid and dramatic changes in load. You can now define a scaling policy that will respond to the magnitude of the alarm breach in a proportionate and appropriate way. For example, if you try to keep your average CPU utilization below 50% you can have a standard response for a modest breach (50% to 60%), two more for somewhat bigger breaches (60% to 70% and 70% to 80%), and a super-aggressive one for utilization that exceeds 80%.

Here’s how I set this up for my Auto Scaling group:



In this example I added a fixed number (1, 2, 4, or 8) of instances to the group. I could have chosen to define the policies on a percentage basis, increasing the instance count by (say) 50%, 100%, 150%, and 200% at the respective steps. The empty upper bound in the final step is effectively positive infinity. You can also define a similar set of increasingly aggressive policies for scaling down.

As you can see from the example above, you can also tell Auto Scaling how long it should take for an instance to warm up and be ready to start sharing the load. While this waiting period is in effect, Auto Scaling will include the newly launched instances when it computes the current size of the group. However, during this scaling time, the instances are not factored in to the CloudWatch metrics for the group. This avoids unnecessary scaling while the new instances prepare themselves to take on their share of the load.

Step policies continuously evaluate the alarms during a scaling activity and while unhealthy instances are being replaced with new ones. This allows for faster response to changes in demand. Let’s say the CPU load increases and the first step in the policy is activated. During the specified warm up period (300 seconds in this example), the load might continue to increase and a more aggressive response might be appropriate. Fortunately, Auto Scaling is in violent agreement with this sentiment and will switch in to high gear (and use one of the higher steps) automatically. If you create multiple step scaling policies for the same resource (perhaps based on CPU utilization and inbound network traffic) and both of them fire at approximately the same time, Auto Scaling will look at both policies and choose the one that results in the change of the highest magnitude.

f