



Immersion Day

*Using EC2 Auto Scaling to Build Highly Available
Environments on AWS*

JUNE 2019

Table of Contents

Overview.....	3
Task 1: Creating a Launch Template.....	5
Task 2: Creating an Auto Scaling Group	13
Task 3: Creating Scaling Policies	17
Task 4: Testing the Auto Scaling Group	19
Task 5: Clean Up.....	24
Appendix – Additional Reading.....	25

Overview

Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling helps you to maintain application availability by scaling your infrastructure as the need or demand arises. It helps support your application's workload by making sure you have the right number of EC2 instances available. You can create **Auto Scaling Groups**, which are collections of EC2 instances that have the same characteristics and applications.

The number of EC2 instances can be scaled in or out as Auto Scaling responds to the metrics you define when creating these groups.

- You can specify the **minimum number** of instances in each Auto Scaling Group, so that your group never goes *below* this size.
- You can specify the **maximum number** of instances in each Auto Scaling Group, so that your group never goes *above* this size.
- You can specify a **desired capacity** so that Auto Scaling ensures your group always has a certain number of instances.
- You can specify **scaling policies** so that Auto Scaling will modify the desired target capacity mentioned in the previous point. It will launch or terminate instances as demand on your application increases or decreases.

There are multiple components of Auto Scaling on AWS. They are:

1. Groups:

Your EC2 instances are organized into *groups* so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and desired number of EC2 instances.

2. Launch Template:

A Launch Template is a capability of EC2 Auto Scaling that allows a way to templatize your launch requests. It enables you to store launch parameters so that you do not have to specify them every time you launch an instance. For example, a launch template can contain an Amazon Machine Image, instance type, storage, and networking settings that you typically use to launch instances. You can also specify advanced configurations like user data, as well

as the ability to choose T2/T3 Unlimited EC2 instances, where you can decide whether to enable applications to burst beyond the baseline CPU performance for as long as needed. For each Launch Template, you can create one or more numbered Launch Template Versions. Each version can have different launch parameters. When you create an Auto Scaling Group that is backed by a Launch Template, you also have the option of launching one type of instance, or a combination of instance types and purchase options.

3. Scaling Policies:

A Scaling Policy tells Auto Scaling when and how to scale. Scaling can occur manually, on a schedule, on demand or you can use Auto Scaling to maintain a specific number of instances.

Auto Scaling is well suited for applications that have unpredictable demand patterns that can experience hourly, daily, or weekly variability in usage. This helps you to manage your cost and eliminate over-provisioning of capacity during times when it is not needed. Auto Scaling can also find an unhealthy instance, terminate that instance, and launch a new one based on the scaling plan

Getting Started

This lab will walk you through the process of building out these components. For the purposes of this lab, you will be using the AWS Management Console. Auto Scaling can also be configured from the AWS CLI or Windows PowerShell if you prefer a command line interface. You can also do this via an available SDK.

This lab assumes that you have an AWS account and the appropriate permissions to use the services involved.

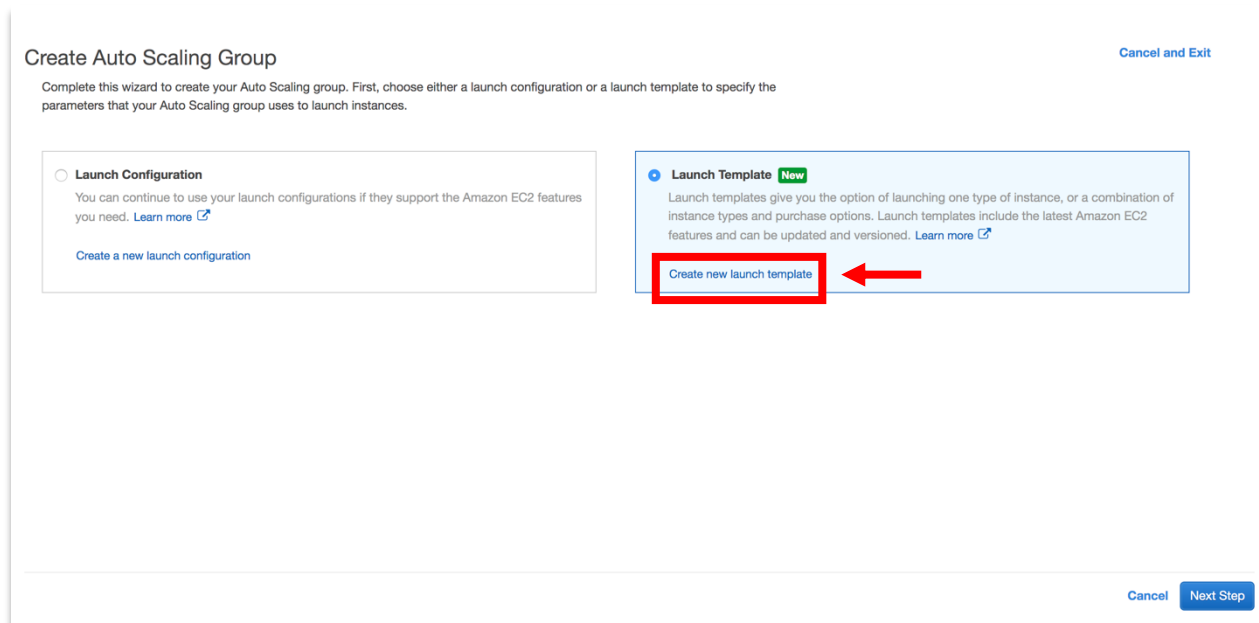
Task 1: Creating a Launch Template

When you create an Auto Scaling Group, you must specify a Launch Template. The first step in this lab is to create the Launch Template for an EC2 Auto Scaling Group.

1. Sign into the AWS Management Console and on the Services menu, click **EC2**.
2. In the left navigation pane, find AUTO SCALING and click **Auto Scaling Groups**.
3. Click **Create Auto Scaling group**.

The console explains that the first step is to define a Launch Template and the second step is to create the Auto Scaling Group.

4. Click **Get Started** if prompted.
5. Select **Launch Template**, and then click **Create a new launch template**.
 - a. If you do not see the option to create a Launch Template at this step, look for INSTANCES in the left navigation pane and select **Launch Templates** to get started instead



The screenshot shows the 'Create Auto Scaling Group' wizard in the AWS Management Console. The title is 'Create Auto Scaling Group' with a 'Cancel and Exit' link in the top right. Below the title is a descriptive paragraph: 'Complete this wizard to create your Auto Scaling group. First, choose either a launch configuration or a launch template to specify the parameters that your Auto Scaling group uses to launch instances.' There are two main options: 'Launch Configuration' (unselected) and 'Launch Template' (selected, marked with a blue dot and a 'New' badge). The 'Launch Template' section contains a description: 'Launch templates give you the option of launching one type of instance, or a combination of instance types and purchase options. Launch templates include the latest Amazon EC2 features and can be updated and versioned.' followed by a 'Learn more' link. At the bottom of the 'Launch Template' section, the button 'Create new launch template' is highlighted with a red rectangle, and a red arrow points to it from the right. At the bottom right of the wizard, there are 'Cancel' and 'Next Step' buttons.

6. This takes you to the *Create launch template* page where you can define the following configurations for your launch template:

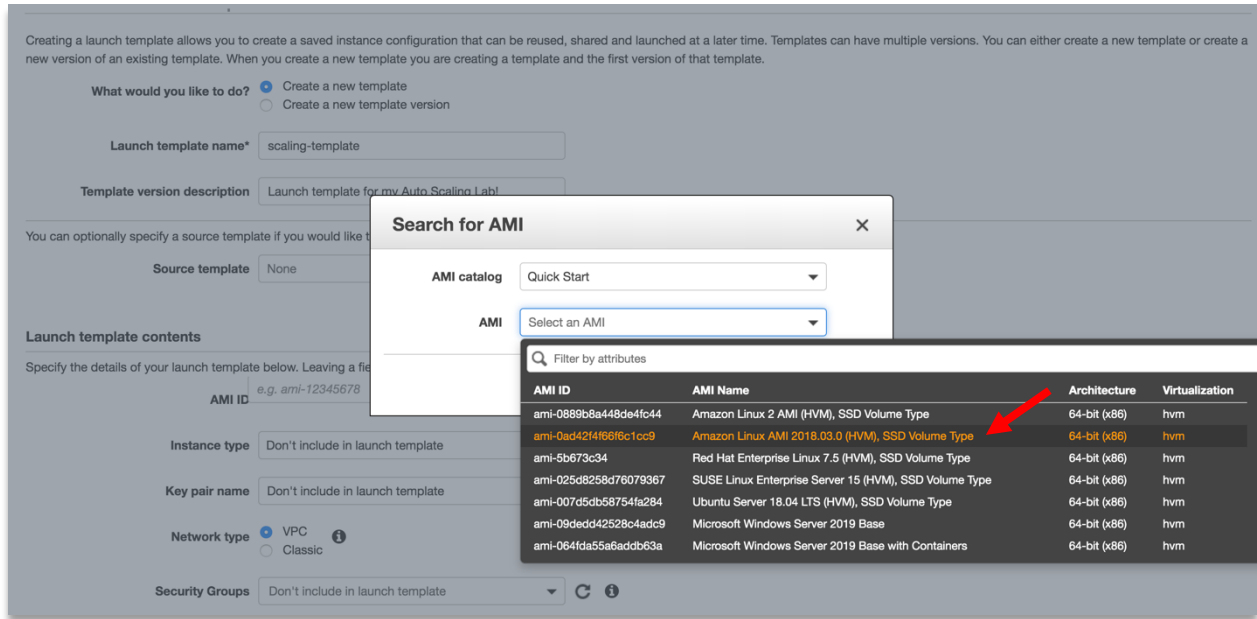
- a. **What would you like to do?** Create a new template
- b. **Launch template name:** scaling-template
- c. **Template version description:** Launch template for my Auto Scaling lab!

7. Next, select an AMI ID by clicking the **Search for AMI** link to the right of this configuration field.

- a. For the AMI Catalog, select Quick Start
- b. For the AMI, select the **Amazon Linux AMI**.

You are selecting an **Amazon Machine Image (AMI)**, which provides the information required to launch an instance. It is a template for the root volume for the instance that can contain an operating system, application server, and applications. You use the AMI to launch an EC2 instance, which is a copy of the AMI running as a virtual server in the cloud.

AMIs are available for various versions of Windows and Linux. In this lab, you are going to launch an instance running *Amazon Linux*.



8. Next, select the instance type to be **t2.micro**.

EC2 offers a wide selection of instance types that are optimized to fit different use cases. For example, there are memory optimized, compute optimized, and storage optimized instances, among others. For this lab, you are going to use a t2.micro instance, which is a general purpose instance. This instance type is also covered under EC2 Free Tier pricing, which includes 750 hours of Linux and Windows t2.micro instances each month for one year.

9. Your configurations up to this point should look like this:

Launch Templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions. You can either create a new template or create a new version of an existing template. When you create a new template you are creating a template and the first version of that template.

What would you like to do? ☒ Create a new template
☐ Create a new template version

Launch template name*

Template version description

You can optionally specify a source template if you would like to create a template from another existing template.

Source template

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

AMI ID [Search for AMI](#) ⓘ

Instance type ⓘ

Key pair name ⓘ

Network type ☒ VPC ⓘ
☐ Classic

10. Now, you are going to create a security group. Open up the AWS Management Console in a new tab and on the Services menu, click **EC2**.

11. On the left navigation pane, find NETWORK & SECURITY and select **Security Groups**.

12. Click **Create Security Group** and enter the following configurations:

- a. **Security group name:** scaling-lab-sg
- b. **Description:** Security group for my Auto Scaling lab!
- c. Add an **Inbound rule** with Type HTTP

Your final configurations should look like this:

Create Security Group

Security group name ⓘ scaling-lab-sg

Description ⓘ Security group for my Auto Scaling lab!

VPC ⓘ vpc-7da7ff15 (default)

Security group rules:

Inbound Outbound

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin I

Add Rule

Cancel Create

By creating this security group and adding it to your Launch Template, you are allowing inbound HTTP traffic on port 80 to the EC2 instances in your Auto Scaling Group so that you can access them through your web browser.

13. After this, go back to the first tab where you are configuring your Launch Template.
14. Hit the refresh button next to Security Groups. Find and select the Security Group that you just created.
15. Scroll down to Storage Volumes and select **Add new volume** and configure:
 - a. **Device Name:** /dev/xvda
 - b. **Size:** 8
 - c. **Volume type:** General Purpose
 - d. **Delete on termination:** Yes

In a production environment, you might want to add persistent block storage based on your requirements. You can specify an EBS volume

type, size, and termination requirement. You can also enable encryption on the EBS volume.

16. Under Instance Tags, select **Add Tag** and configure:

- a. **Key:** Name
- b. **Value:** Scaling Lab Instances

You can add tags for better organization and management of your AWS resources.

17. Your configurations up to this point should look like this:

18. Expand **Advanced details**

- a. **Monitoring:** Enabled

Here, you are enabling CloudWatch Detailed Monitoring. By default, your instance is enabled for **basic monitoring** with a 5-minute period for the instances. You can optionally enable **detailed monitoring** for an added cost, however 10 Detailed Monitoring Metrics at a 1-minute frequency are covered under the Free Tier. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance.

b. Enter the following **user data** as text:

```
#!/bin/sh
yum -y install httpd php mysql php-mysql
chkconfig httpd on
/etc/init.d/httpd start
cd /tmp
wget http://us-east-1-aws-
training.s3.amazonaws.com/self-paced-lab-
4/examplefiles-as.zip
unzip examplefiles-as.zip
mv examplefiles-as/* /var/www/html
```

When you launch EC2 instances, you have the option to pass **user data** to the instance which can be used to perform common automated configuration tasks and even run scripts after the instance starts. Here, you are installing scripts needed for this lab when the instance is created and launched.

19. Your final configurations for your Launch Template should be similar to this:

▼ Advanced details

Purchasing option	<input type="checkbox"/> Request Spot instances	
Request Spot instances at the Spot price, capped at the On-Demand price		

IAM instance profile	<input type="text" value="e.g. arn:aws:iam::123456789012:instance-profile/MyProfile"/>	
----------------------	--	--

Shutdown behavior	<input type="text" value="Don't include in launch template"/>	
-------------------	---	--

Stop - Hibernate behavior	<input type="text" value="Don't include in launch template"/>	
---------------------------	---	--

Termination protection	<input type="text" value="Don't include in launch template"/>	
------------------------	---	--

Monitoring	<input type="text" value="Enable"/>	
------------	-------------------------------------	--

Elastic Graphics	<input type="text" value="Don't include in launch template"/>	
------------------	---	--

T2/T3 Unlimited	<input type="text" value="Don't include in launch template"/>	
-----------------	---	--

Placement group name	<input type="text" value="Don't include in launch template"/>	
----------------------	---	--

EBS-optimized instance	<input type="text" value="Don't include in launch template"/>	
------------------------	---	--

Capacity Reservations	<input type="text" value="Don't include in launch template"/>	
-----------------------	---	--

Tenancy ⓘ

RAM disk ID ⓘ

Kernel ID ⓘ

License Configurations ⓘ

User data ⓘ

```
#!/bin/sh
yum -y install httpd php mysql php-mysql
chkconfig httpd on
/etc/init.d/httpd start
cd /tmp
wget http://us-east-1-aws-training.s3.amazonaws.com/self-paced-lab-4/examplefiles-as.zip
unzip examplefiles-as.zip
mv examplefiles-as/* /var/www/html
```

* Required

[Cancel](#) [Create launch template](#)

20. When you are sure the configurations are correct, click **Create launch template** and now you are finished creating your Launch Template!

Task 2: Creating an Auto Scaling Group

You have officially created a Launch Template, which defines *what* should be launched. Now, it is time to create an Auto Scaling Group so that you can define *how many* EC2 instances should be launched and *where* to launch them.

After you finish creating the Launch Template, follow these steps:

21. Go back to the Services menu and click **EC2**.
22. In the left navigation pane, find AUTO SCALING and click **Auto Scaling Groups**
23. Click **Create Auto Scaling Group**.
24. Select **Launch Template** and choose the one you have just created.

Create Auto Scaling Group Cancel and Exit

Complete this wizard to create your Auto Scaling group. First, choose either a launch configuration or a launch template to specify the parameters that your Auto Scaling group uses to launch instances.

☐ Launch Configuration

You can continue to use your launch configurations if they support the Amazon EC2 features you need. [Learn more](#)

[Create a new launch configuration](#)

☒ Launch Template New

Launch templates give you the option of launching one type of instance, or a combination of instance types and purchase options. Launch templates include the latest Amazon EC2 features and can be updated and versioned. [Learn more](#)

[Create new launch template](#)

Filter launch templates...

< 1 to 1 of 1 Launch Templates >

Name	Launch Template Id	Default Version	Latest Version	Create Time	Created by
scaling-template	lt-0c6aca0913cdc872d	1	1	Tue May 07 19:52:46 GMT-700 2019	arn:aws:iam::348179322981:user/glizzig

Cancel Next Step

25. Click **Next Step**.
26. On the **Configure Auto Scaling group details** page, configure the following:
 - a. **Group name:** auto-scaling-lab
 - b. **Fleet composition:** combine purchase options and instances

This is a feature that lets you create a mixed Auto Scaling Group with different instance types as well as different purchasing options. This way, you can harness the power of multiple EC2 instance types to

support your workloads. You can also cost optimize by having the option to use On-Demand, Reserved, and Spot instances.

- c. Try adding a new instance type by selecting a different type, like
t3.micro
- d. **Group size:** start with 1 instance
- e. **Network:** keep this the default VPC
- f. **Subnet:** select the first subnet that appears

27. Expand **Advanced Details**.

Here you can see further configurations that you can make with your Auto Scaling Group, like the ability to associate your group with a load balancer. You can use an Elastic Load Balancer with your Auto Scaling Group to help automatically distribute incoming application traffic across multiple targets, like your EC2 instances. It can handle varying load of your application traffic across multiple Availability Zones.

You can use an Elastic Load Balancer with your Auto Scaling Group to create a fault tolerant and highly available environment. If you associate an Elastic Load Balancer to the Auto Scaling Group, then all newly instantiated instances will register themselves with that load balancer and then it will distribute incoming traffic across those instances. For now, you will leave this unselected and move on.

- a. Select **Enable CloudWatch detailed monitoring** under Advanced Details.

28. Your configurations should be similar to:

For now, you are going to skip this step.

31. Click **Next: Configure Tags**.

32. Add a tag and configure the following:

a. **Key:** Name

b. **Value:** Auto Scaling Group

The screenshot shows the 'Create Auto Scaling Group' wizard in the AWS Management Console, specifically the 'Configure Tags' step. The wizard has five steps: 1. Configure Auto Scaling group details, 2. Configure scaling policies, 3. Configure Notifications, 4. Configure Tags (current step), and 5. Review. The title is 'Create Auto Scaling Group'. Below the title, a note states: 'A tag consists of a case sensitive key-value pair that you can use to identify your group. For example, you could define a tag with Key = Environment and Value = Production. You can optionally choose to apply these tags to instances in the group when they launch. [Learn more](#).' The main area contains a table with two columns: 'Key' and 'Value'. The first row has 'Name' in the 'Key' column and 'Auto Scaling Group 1' in the 'Value' column. To the right of the 'Value' column is a checkbox labeled 'Tag New Instances' with an information icon. Below the table, there is an 'Add tag' button and a counter '49 remaining'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Review'.

Key	Value	Tag New Instances
Name	Auto Scaling Group 1	<input checked="" type="checkbox"/>

[Add tag](#) 49 remaining

[Cancel](#) [Previous](#) [Review](#)

33. Select **Review** and then **Create Auto Scaling group** and now you have successfully created your Auto Scaling Group!

Task 3: Creating Scaling Policies

Now that you have created an Auto Scaling Group, you are going to create a scaling policy so that your Auto Scaling Group knows when to add capacity to your applications when they need it, and remove capacity when they don't.

You can use AWS Auto Scaling to do this. This is a service that monitors your applications and automatically adjusts capacity to maintain steady, predictable performance. It offers different built-in scaling strategies that you can choose from, as well as predictive and dynamic scaling. Predictive scaling uses machine learning to predict future traffic, including regularly occurring spikes, and provisions the right number of EC2 instances in advance of these predicted changes. Dynamic scaling lets you define how to scale in response to changing demand by using target tracking, which is a feature that lets you set a target value and specify a scaling metric.

34. To start, go to the Services menu and select **AWS Auto Scaling**

35. Click **Get Started** if prompted.

36. Under *Find Scalable Resources*, select **Choose EC2 Auto Scaling groups** and select the one you have just created.

The screenshot shows the AWS Auto Scaling console interface for creating a scaling plan. The breadcrumb trail at the top reads 'AWS Auto Scaling > Scaling plans > Create scaling plan'. On the left, a sidebar lists four steps: 'Step 1 Find scalable resources' (active), 'Step 2 Specify scaling strategy', 'Step 3 Configure advanced settings (optional)', and 'Step 4 Review and create'. The main content area is titled 'Find scalable resources' with a subtitle 'Automatically discover or manually choose resources to add to your scaling plan. Info'. Under 'Choose a method', there are three radio button options: 'Search by CloudFormation stack' (disabled), 'Search by tag' (disabled), and 'Choose EC2 Auto Scaling groups' (selected). Below these, the 'Choose Auto Scaling groups' section features a dropdown menu labeled 'Auto Scaling groups' with the placeholder text 'Choose Auto Scaling groups'. A tag 'auto-scaling-lab' is shown with a close button. At the bottom right, there are 'Cancel' and 'Next' buttons.

37. Click **Next** so that you can specify a scaling strategy and define:

a. **Name:** scaling-plans

Here you can choose a built-in scaling strategy. There are ones optimized for availability, cost, or both. You can also define your own custom metrics for optimization. You can also enable predictive and dynamic scaling here.

38. Select the **Optimized for availability** strategy

39. Disable predictive scaling for now and leave the rest of the configuration

details the same

The default settings are configured to maintain an average CPU utilization of 40%. Your resources will scale in and out as needed to achieve this.

The screenshot shows the 'Specify scaling strategy' step in the AWS IAM console. On the left, a sidebar lists four steps: Step 2 (Specify scaling strategy), Step 3 (Configure advanced settings (optional)), Step 4 (Review and create), and Step 5 (Create). The main content area is titled 'Scaling plan details' and includes a 'Name' field with the value 'scaling-plans'. Below this, it states 'Resources: 1 Auto Scaling group was selected.' The 'Auto Scaling groups (1)' section shows a table with one group, 'scaling-plans', and a checkbox 'Include in scaling plan' which is checked. The 'Scaling strategy' section has four radio button options: 'Optimize for availability' (selected), 'Balance availability and cost', 'Optimize for cost', and 'Custom'. Below these are two checkboxes: 'Enable predictive scaling' (unchecked) and 'Enable dynamic scaling' (checked). A 'Configuration details' link is at the bottom.

Step 2
Specify scaling strategy

Step 3
Configure advanced settings (optional)

Step 4
Review and create

Step 5
Create

Scaling plan details

Name
scaling-plans
Must be 1-128 characters long and should not contain the pipe "|", colon ":", and forward slash "/" characters.

Resources
1 Auto Scaling group was selected.

Auto Scaling groups (1)
Specify a scaling strategy for 1 Auto Scaling group.

☒ Include in scaling plan

Scaling strategy
The strategy defines the scaling metric and target value used to scale your resources.

☒ **Optimize for availability**
Keep the average CPU utilization of your Auto Scaling groups at 40% to provide high availability and ensure capacity to absorb spikes in demand.

☐ **Balance availability and cost**
Keep the average CPU utilization of your Auto Scaling groups at 50% to provide optimal availability and reduce costs.

☐ **Optimize for cost**
Keep the average CPU utilization of your Auto Scaling groups at 70% to ensure lower costs.

☐ **Custom**
Choose your own scaling metric, target value, and other settings.

☐ **Enable predictive scaling**
Support your scaling strategy by continually forecasting load and proactively scheduling capacity ahead of when you need it. [Info](#)

☒ **Enable dynamic scaling**
Support your scaling strategy by creating target tracking scaling policies to monitor your scaling metric and increase or decrease capacity as you need it. [Info](#)

[► Configuration details](#)

40. Hit next twice and create the scaling plan. You have successfully created your scaling strategy!

Task 4: Testing the Auto Scaling Group

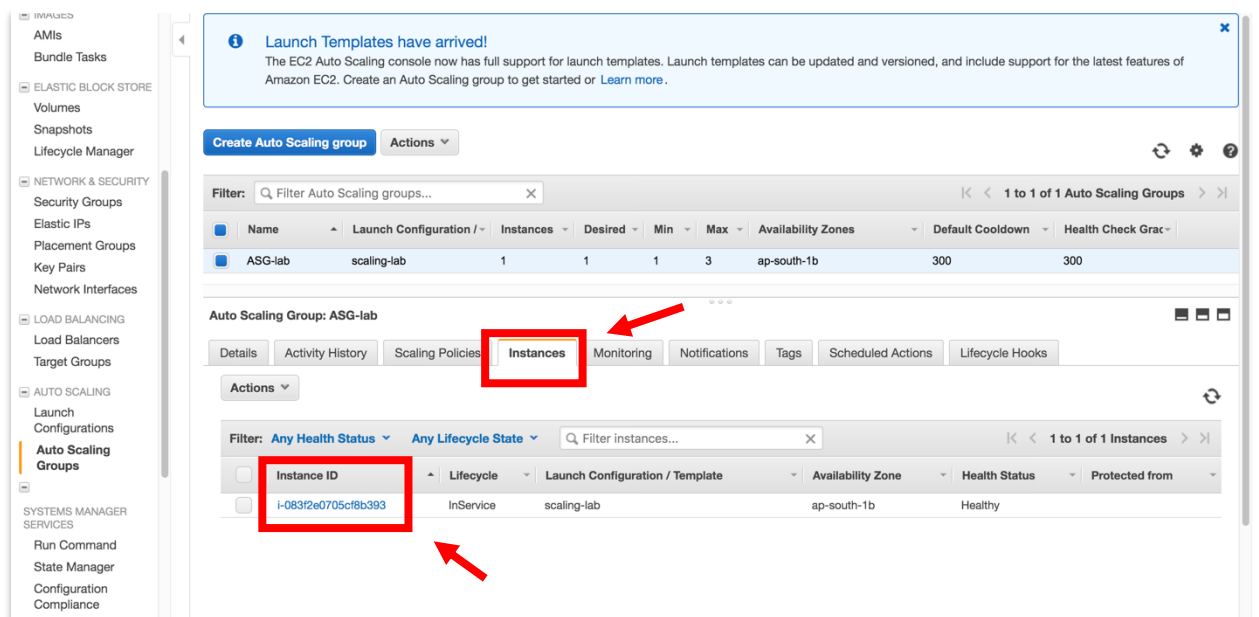
Now that you have created your Auto Scaling Group and scaling strategy, you can test it to ensure that it works correctly.

41. Hit **Close** to return back to the Auto Scaling Groups page under Auto Scaling listed in the navigation pane on the left.

42. Select the Auto Scaling Group you have just created.

This brings up details about your Auto Scaling Group, like the active history, scaling policies, instances, and allows you to monitor different metrics.

43. Select the **Instances** tab to see that your Auto Scaling Group is in the process of spinning up an EC2 instance, as shown below.



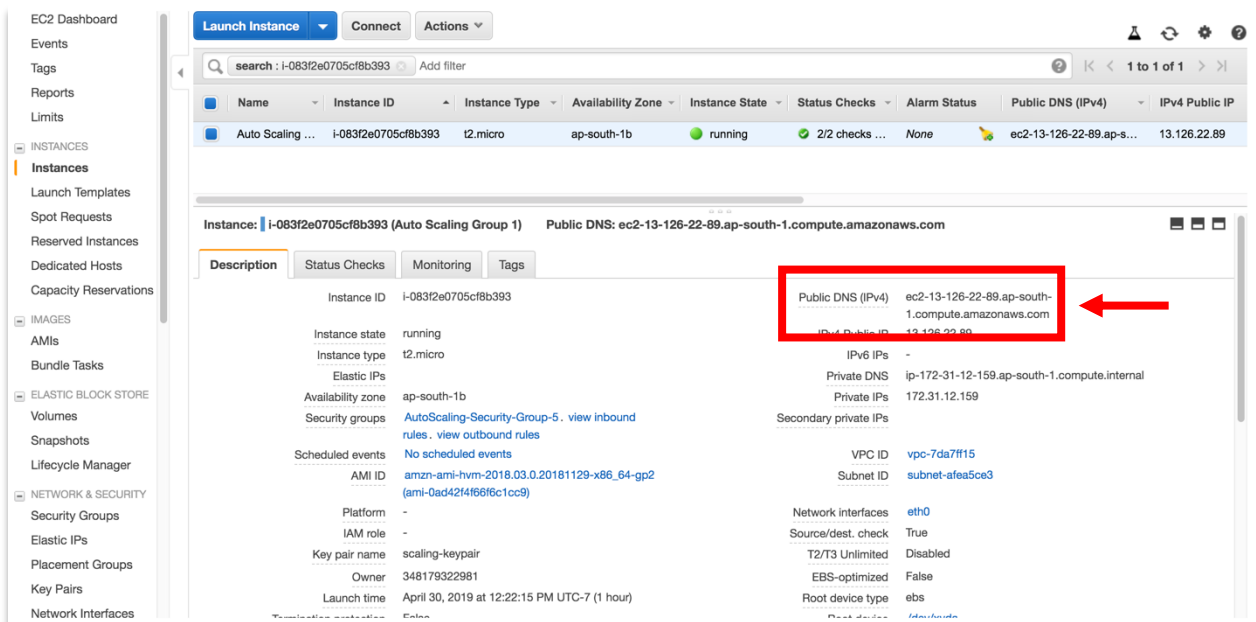
44. Select the **Instance ID**, which will take you to the EC2 Management Console.

On this page, you can see details about specific EC2 instances, like the public DNS name, IPv4 address, and status checks. Since you have enabled detailed monitoring, you can also monitor metrics at a 1-minute

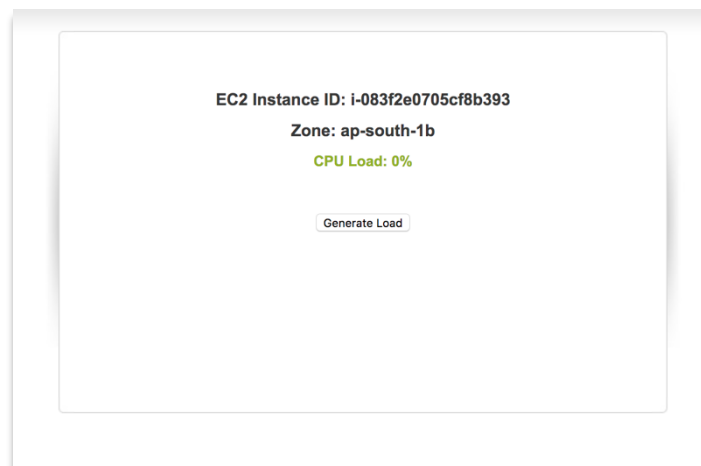
interval under the **Monitoring** tab. These metrics include CPU utilization, disk reads, disk writes, and much more.

45. Wait for the instance state to say running and status checks to say 2/2 before moving forward.

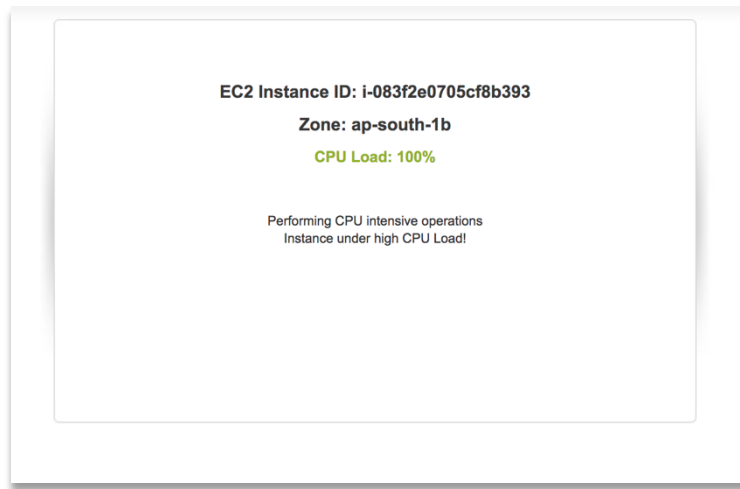
46. Copy the **Public DNS** name.



47. Open a new tab in your web browser and go to that public DNS name. You should see this:



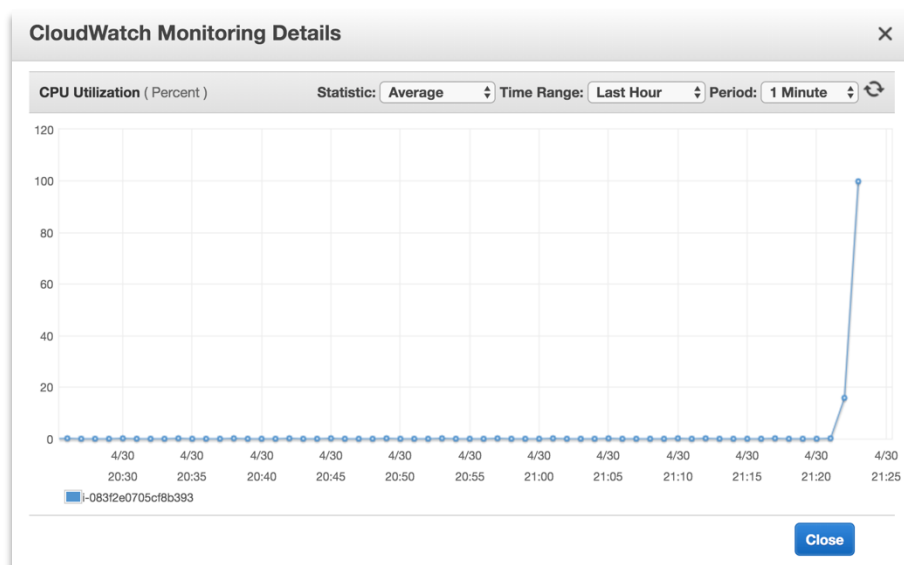
48. Click **Generate Load**. You might have to do this a couple times to ensure you are keeping a constant load on your instance.



Now that you are maxing out the load on that EC2 instance to 100%, your Auto Scaling Group should respond by spinning up more instances to help support the increase in load.

49. Go back to the EC2 console, make sure your instance is selected, and click the **Monitoring** tab so you can see CloudWatch detailed monitoring metrics.

It will take a minute for data to start populating in CloudWatch. Since you are maxing out the load on your EC2 instance, your CPU utilization should look similar to this:

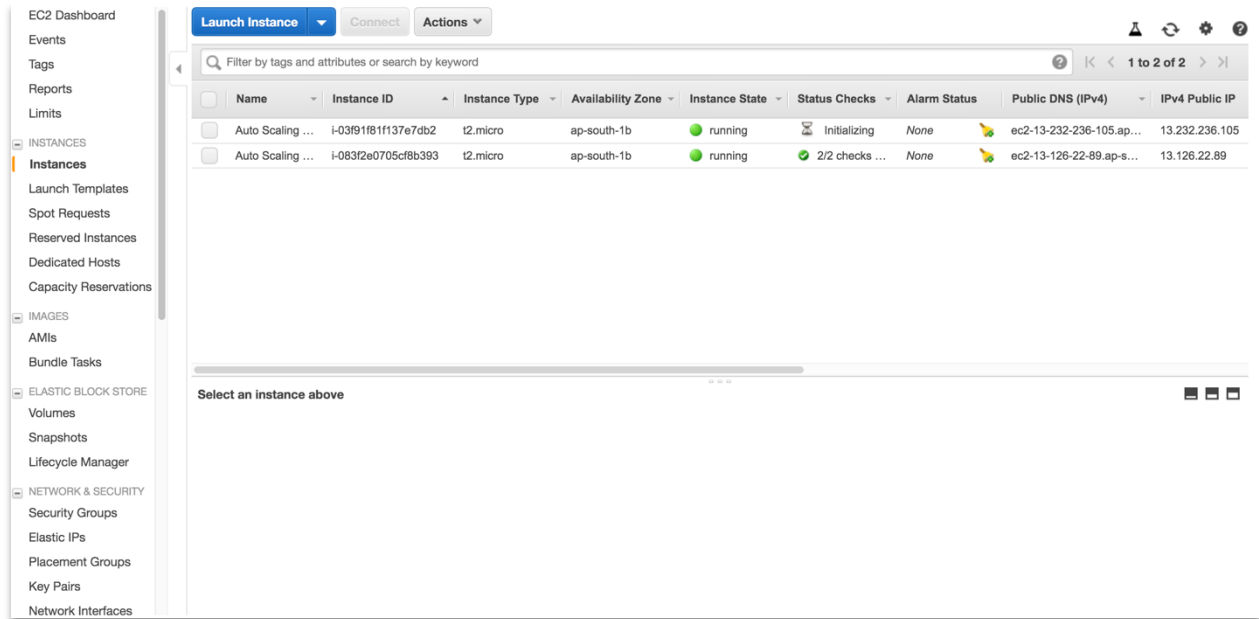


You can also view metrics in the AWS Auto Scaling page if you select your scaling plan:



Based on the scaling policy you have set in your Auto Scaling Group, your group should spin up a new instance to help support this increase in demand.

This might take a couple of minutes. Refresh the EC2 instances page and you should soon see a new instance spinning up automatically.



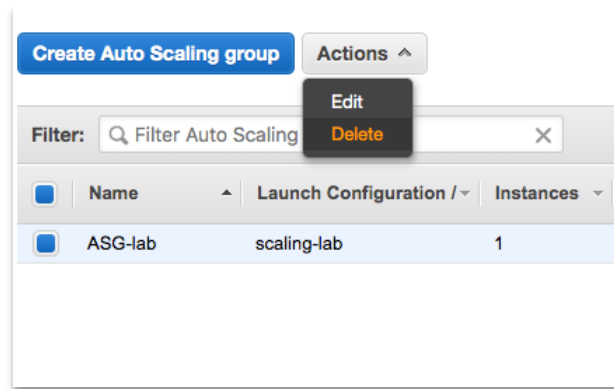
You can also see this in the Auto Scaling Group page. If you look at the details under the Active History tab, you can see that the new instance is warming up. You can look at the Instances tab to see how many instances there are in your group currently. The monitoring tab shows you different metrics like group size, pending instances, total instances, and much more.

Congratulations! You have successfully created an EC2 Auto Scaling Group!

Task 5: Clean Up

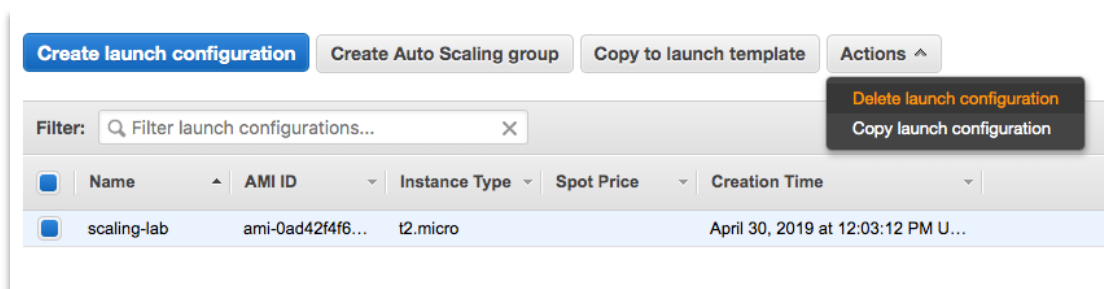
Now that you have successfully created your Auto Scaling Group and tested it to ensure that it works, you can clean up your environment by spinning down these resources.

50. First, delete your Auto Scaling Group by selecting your group, hitting **Actions**, and then **Delete**.



Deleting your Auto Scaling Group also deletes all the EC2 instances associated with it.

51. Finally, delete your Launch Template by going to the Launch Templates in the left navigation pane, selecting the template, hitting **Actions**, and then **Delete**.



Appendix – Additional Reading

What is Amazon EC2 Auto Scaling?

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>

What is Elastic Load Balancing?

<https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html>

What is Amazon VPC?

<https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>

Whitepaper: Building Fault-Tolerant Applications on AWS

https://media.amazonwebservices.com/AWS_Building_Fault_Tolerant_Applications.pdf