Maintaining High Availability with Auto Scaling

**SPL-04 - Version 4.3.23**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

Auto Scaling allows you to scale your Amazon Elastic Compute Cloud (Amazon EC2) capacity up or down automatically according to conditions you define. With Auto Scaling, you can ensure that the number of Amazon EC2 instances you’re using increases seamlessly during demand spikes to maintain performance and decreases automatically during demand lulls to minimize costs. Auto Scaling is particularly well suited for applications that experience hourly, daily, or weekly variability in usage.

But Auto Scaling represents more than a way to add and subtract servers. It is also a mechanism to handle failures similar to the way load balancing handles unresponsive servers. This lab demonstrates configuring Auto Scaling to automatically launch, monitor, and update the load balancer associated with your Amazon EC2 instances.

There are two important things to know about Auto Scaling. First, Auto Scaling is a way to set the “cloud temperature.” You use policies to “set the thermostat,” and under the hood, Auto Scaling controls the heat by adding and subtracting Amazon EC2 resources on an as-needed basis in order to maintain the “temperature” (capacity).

An Auto Scaling policy consists of:

* A **launch configuration** that defines the servers that are created in response to increased demand.
* An **Auto Scaling group** that defines when to use a launch configuration to create new server instances and in which Availability Zone (AZ) and load balancer context they should be created.

Second, Auto Scaling assumes a set of homogeneous servers. That is, Auto Scaling does not know that Server A is a 64-bit extra-large instance and more capable than a 32-bit small instance. In fact, this is a core tenet of cloud computing: scale horizontally using a fleet of fungible resources; individual resources are secondary to the fleet itself.

OBJECTIVES

By the end of this lab, you should be able to do the following:

* Create a new launch configuration using command-line tools.
* Create a new Auto Scaling group using command line tools.
* Configure Auto Scaling notifications that are triggered when instance resources become too high or too low.
* Create policies to scale up or scale down the number of currently running instances in response to changes in resource utilization.

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab, you should be familiar with basic Linux server administration and comfortable using the Linux command-line tools. You should also be proficient at this point with the basics of creating new Amazon EC2 server instances and configuring Elastic Load Balancing.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Caution:** Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).
* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Learn more:** Where to find more information.
* **Note:** A hint, tip, or important guidance.
* **Refresh:** A time when you might need to refresh a web browser page or list to show new information.
* **Task complete:** A conclusion or summary point in the lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Warning:** Do not change the **Region** unless instructed.

COMMON SIGN-IN ERRORS

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

SERVICES USED IN THIS LAB

**Key components of Auto Scaling**

When you launch a server manually, you provide parameters such as which Amazon Machine Image (AMI), which instance type, and which security group to launch in. Auto scaling calls this a *launch configuration*. It is simply a set of parameters describing what kind of instances to launch.

*Auto Scaling groups* tell the system what to do with an instance after it is launched. This is where you specify which Availability Zones your instances should be launched in, which load balancers they will receive traffic from, and the minimum and maximum number of instances to run at any given time.

You need rules that tell the system when to add or subtract instances. These are known as *scaling policies*, and have rules such as “scale the fleet out by 10%” and “scale in by 1 instance.”

**Timing matters**

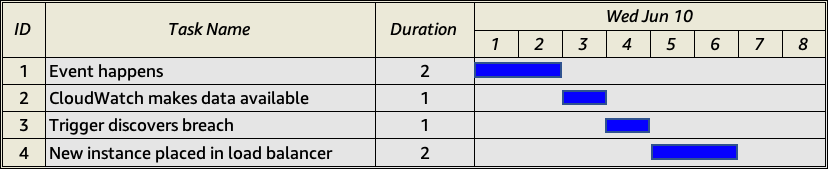
There are costs related to using Auto Scaling. There are two important factors that directly affect the cost of AWS and also the manner in which your application scales: cost and time.

**Amazon EC2 Linux instances are charged per-second**

This means that you can scale-out the servers when there is a lot of activity, then scale-in to reduce costs when less capacity is required.

**Scaling takes time**

Consider the following table. In most situations, a considerable amount of time passes between when the *need* for a scaling event occurs and *when* the scaling event happens.



*Image description: The preceding diagram depicts a tabular representation of different tasks associated with a scaling event for a generic application architecture.*

* In this example, the rule says that you must be in a particular condition for at least two minutes.
* CloudWatch is the underlying data collection system that monitors statistics such as CPU utilization. It is a polling protocol, and in general takes 60 seconds to aggregate new data.
* Auto Scaling is also a polling system, and it takes another 60 seconds.
* Then there is boot time for your server. A large, complex server may take many minutes to launch.
* Finally, the load balancer needs to poll the server for a few cycles before it is comfortable that the server is healthy and accepting requests.

AWS SERVICES NOT USED IN THIS LAB

AWS service capabilities used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

**Task 1: Connect to your EC2 Instance**

In this task, you connect to your EC2 instance using Session Manager.

1. Copy the **CommandHostSessionUrl** value that is listed to the left of these instructions. Paste the URL into a new web browser tab and press **Enter**.

This opens up a Session Manager session to your EC2 instance.

**Task complete:** You have successfully connected to your EC2 instance.

**Task 2: Configure AWS CLI**

In this task, you configure the AWS Command Line Interface (CLI) tool to execute the commands.

1. **Command:** To configure AWS CLI, run the following command in your Session Manager session:

aws configure

* For **AWS Access Key ID [None]:**, press **Enter**.
* For **AWS Secret Access Key [None]:**, press **Enter**.
* For **Default region name [None]:**, enter the value of **REGION** located to the left of these instructions
* For **Default output format [None]:**, press **Enter**.

**Note:** You omitted the Access Key ID and Secret Access Key values as the credentials are being passed to the instance using an Identity and Access Management (IAM) role created during setup.

**Learn more:** Refer to *IAM roles* in the **Additional resources** section for more information.

The default output format if not mentioned is **json**.

**Task complete:** You have successfully configured the AWS CLI.

**Task 3: Create a Launch Configuration**

In this task, you create an autoscaling launch configuration. This launch configuration specifies a machine image (Amazon Machine Image or *AMI*) that launches when Auto Scaling adds new servers. You use the same AMI that was used to create your current running instance.

1. **Command:** To create an autoscaling launch configuration, run the following command:

* Replace the **AMIID** placeholder value with the **AMIID** value that is listed to the left of these instructions.
* Replace the **EC2SECURITYGROUPID** placeholder value with the **EC2SECURITYGROUPID** value that is listed to the left of these instructions.

aws autoscaling create-launch-configuration --image-id AMIID --instance-type t3.micro --security-groups EC2SECURITYGROUPID --user-data file:///home/ec2-user/as-bootstrap.sh --launch-configuration-name lab-lc

**Task complete:** You have successfully created an autoscaling launch configuration.

**Task 4: Create an Auto Scaling Group**

In this task, you create a new Auto Scaling group in your current region and Availability Zone. The group will ensure that there is always one server running by establishing a minimum Auto Scaling group size of one. You also specify that the maximum number of servers in this group must not exceed four.

1. **Command:** To create an autoscaling group, run the following command:

* Replace the **LOADBALANCER** placeholder value with the **LOADBALANCER** value that is listed to the left of these instructions.
* Replace the **SUBNET1** placeholder value with the **SUBNET1** value that is listed to the left of these instructions.
* Replace the **SUBNET2** placeholder value with the **SUBNET2** value that is listed to the left of these instructions.

aws autoscaling create-auto-scaling-group --auto-scaling-group-name lab-as-group --launch-configuration-name lab-lc --load-balancer-names LOADBALANCER --max-size 4 --min-size 1 --vpc-zone-identifier SUBNET1,SUBNET2

**Task complete:** You have successfully created an autoscaling group in your current region and Availability Zone.

**Task 5: Verifying Auto Scaling**

In this task, you verify your lab environment and test your Auto Scaling group.

TASK 5.1: VERIFY THAT THE AUTO SCALING SERVERS WERE LAUNCHED

Use the AWS Management Console to inspect your instance count. You should see two instances in your fleet because you set the minimum size to one (you may need wait a few minutes before it appears), and you had one instance already running.

1. Switch back to the **AWS Management Console**.
2. At the top of the AWS Management Console, in the search bar, search for and choose

EC2

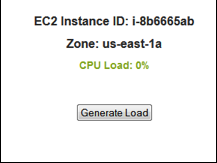
.

1. In the left navigation pane, choose **Instances**.
2. Select the checkbox next to the instance without a name (*not* the instance named “Command-Line Tools”).
3. Verify that the **Status check** column has changed from *Initializing* to *2/2 checks passed*. You can use the refresh icon to refresh the status.
4. Choose the **Details** tab in the lower pane.
5. Copy the value of **Public IPv4 DNS** for your new instance and save it in your notepad.

It should be similar to *ec2-3-233-214-200.compute-1.amazonaws.com*.

1. Paste the **Public IPv4 DNS** value into the address bar of a new browser window to verify that the instance is running.

You should see a page that looks like this:



*Image description: The preceding diagram depicts a sample web page confirming that the instance is running.*

TASK 5.2: VERIFY THAT AUTO SCALING WORKS

1. Switch back to the **Instances** page of the **AWS Management Console**.
2. Select the checkbox next to the instance without a name (*not* the instance named “Command-Line Tools”).
3. Choose the **Instance state** dropdown menu, and then choose **Terminate instance**.
4. On the **Terminate instance?** pop-up window, choose **Terminate**.

In a few minutes a new **t3.micro** instance should appear because Auto Scaling detects that the fleet size is below the minimum size.

1. **Refresh:** Use the refresh icon to refresh the status.

**Note:** It may take a few minutes for the new instance to appear.

1. Wait until the **Instance state** of the new instance moves to *Running*.
2. Select the checkbox next to the new instance that was launched in the previous step (*not* the instance named “Command-Line Tools”).
3. Choose the **Instance state** dropdown menu, and then choose **Stop instance**.
4. On the **Stop instance?** pop-up window, choose **Stop**.

Auto Scaling should detect that the instance is non-responsive after a few minutes and should automatically terminate it and launch a replacement instance for you.

TASK 5.3: TAG AUTO SCALING RESOURCES

Notice that the Auto Scaling instances are launched without names. There are two ways to help you better identify these instances. The first is by adding a new column to the Management Console.

1. On the **Instances** page, choose the **Gear** icon from the top right corner to display the **Preferences** dialog box.
2. On the **Preferences** dialog box, navigate to the **Tag columns** selection and use the dropdown menu to select the **aws:autoscaling:groupName** option.
3. Choose **Confirm**.
4. **Refresh:** Use the refresh icon to refresh the status.

Auto Scaling automatically creates and populates a tag called **aws:autoscaling:groupName** for your Auto Scaling instances.

The second way you can better identify your Auto Scaling instances is to modify your Auto Scaling group to populate the Name tag for you. You could have created a Name tag for the Auto Scaling group when you created it by using the --tags “Key=Name, Value=AS-Web-Server” option. Instead, because the Auto Scaling group already exists, modify the existing tags.

1. Switch back to the **Session Manager** session for your command host instance.
2. If your session has been terminated, copy the **CommandHostSessionUrl** value that is listed to the left of these instructions. Paste the URL into a new web browser tab and press Enter.
3. **Command:** To update the tags for your autoscaling group, run the following command into your Session Manager session:

aws autoscaling create-or-update-tags --tags "ResourceId=lab-as-group, ResourceType=auto-scaling-group, Key=Name, Value=AS-Web-Server, PropagateAtLaunch=true"

If the command ran successfully, you should be returned to the shell prompt.

Now you verify that the configuration was updated.

1. Switch back to the **Instances** page of the **AWS Management Console**.
2. Select the checkbox next to the instance without a name (*not* the instance named “Command-Line Tools”).
3. Choose the **Instance state** dropdown menu, and then choose **Stop instance**.
4. On the **Stop instance?** pop-up window, choose **Stop**.

In a few minutes, a new instance should be created.

1. Verify that the new instance is named **AS-Web-Server**.

**Refresh:** You may need to refresh the browser to see the updates.

TASK 5.4: AUTO SCALING INTEGRATION WITH YOUR LOAD BALANCER

Auto Scaling instances are being added to your load balancer. This was configured with the *–load-balancer-names NameOfYourELB* option when the Auto Scaling group was created. Now confirm that the instances have been added.

1. In the left navigation pane, choose **Load Balancers**.
2. Choose the load balancer whose name starts with **LabStack-**.
3. Choose the **Target instances** tab in the lower pane.

Your instance should be listed by name.

**Task complete:** You have successfully verified your lab environment and tested your Auto Scaling group.

**Task 6: Create Auto Scaling Notifications**

All of these Auto Scaling activities are occurring transparently. You can configure Auto Scaling to notify you when it automatically creates or terminates instances. Auto Scaling has been integrated with Amazon Simple Notification Service (SNS) for precisely this purpose. In this task, you configure notifications for your auto scaling activities using SNS.

SNS is a web service that makes it easy to set up, operate, and send notifications from the cloud. It provides developers with a highly scalable, flexible, and cost-effective ability to publish messages from an application and immediately deliver them to subscribers or other applications.

TASK 6.1: CREATE AN AMAZON SNS TOPIC

First, create an Amazon SNS topic used to send notifications.

1. At the top of the AWS Management Console, in the search bar, search for and choose

Simple Notification Service

.

1. In the left navigation pane, choose **Topics**.

**Caution:** You may need to choose the  icon first, and then choose **Topics**.

1. Choose **Create topic**.
2. On the **Create topic** page, in the **Details** section, configure the following:

* For **Type**, choose **Standard**.
* For **Name**, enter

lab-as-topic

.

1. Choose **Create topic**.
2. On your **lab-as-topic** topic page, choose **Create subscription**.
3. On the **Create subscription** page, in the **Details** section, configure the following:

* For **Protocol**, use the dropdown menu and choose **Email**.
* For **Endpoint**, enter the email address used to receive email notifications.

1. Choose **Create subscription**.

**Caution:** You need to validate your SNS subscription request, so the email address must be a valid one that you can access.

1. Check your email and choose the **Confirm subscription** link to confirm your subscription to the topic.

Now that SNS is set up, you need the Amazon Resource Number (ARN) for the SNS topic to use with Auto Scaling.

1. In the AWS Management Console, return to the SNS configuration page.
2. In the left navigation pane, choose **Topics**.
3. Copy the ARN for your topic **lab-as-topic** and save it in your notepad.

Your ARN should look similar to: *arn:aws:sns:us-east-1:012345678910:lab-as-topic*

TASK 6.2: CREATE AUTO SCALING NOTIFICATIONS

You can use the

aws autoscaling describe-auto-scaling-notification-types

 command on your server’s Linux command line to determine the types of Auto Scaling notifications that are supported. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE COMMAND \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws autoscaling describe-auto-scaling-notification-types

"AutoScalingNotificationTypes": [

"autoscaling:EC2\_INSTANCE\_LAUNCH",

"autoscaling:EC2\_INSTANCE\_LAUNCH\_ERROR",

"autoscaling:EC2\_INSTANCE\_TERMINATE",

"autoscaling:EC2\_INSTANCE\_TERMINATE\_ERROR",

"autoscaling:TEST\_NOTIFICATION"

]

1. Switch back to the **Session Manager** session for your command host instance.
2. If your session has been terminated, copy the **CommandHostSessionUrl** value that is listed to the left of these instructions. Paste the URL into a new web browser tab and press Enter.
3. **Command:** To update the notification configuration for your auto scaling group, run the following command:

* Replace the **SNSARN** placeholder value with the value of your **lab-as-topic** SNS topic from your notepad.

aws autoscaling put-notification-configuration --auto-scaling-group-name lab-as-group --topic-arn SNSARN --notification-types autoscaling:EC2\_INSTANCE\_LAUNCH autoscaling:EC2\_INSTANCE\_TERMINATE

If the command ran successfully, you should be returned to the shell prompt.

1. Check your email to verify that you received a test notification email confirming the configuration.

**Task complete:** You have successfully configured notifications for your auto scaling activities using SNS.

**Task 7: Create Auto Scaling Policies**

Currently you have an Auto Scaling group verifying that you have at least one running server. In this task, you configure the Auto Scaling group to automatically scale up whenever the average CPU of the web server fleet is ≥ 40%.

You use the *aws autoscaling put-scaling-policy* command to create two scaling policies that increase the number of servers by one for scale-up events and decrease the number of servers by one for scale-down events. You also specify a “cooldown” period of 300 seconds to let the environment settle before initiating additional scale-up/down events.

TASK 7.1: CREATE A SCALE-UP POLICY

1. **Command:** To create a scale-up policy, run the following command:

aws autoscaling put-scaling-policy --policy-name lab-scale-up-policy --auto-scaling-group-name lab-as-group --scaling-adjustment 1 --adjustment-type ChangeInCapacity --cooldown 300 --query 'PolicyARN' --output text

**Expected output:** An ARN should be returned similar to the following example output:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

arn:aws:autoscaling:us-east-1:012345678910:scalingPolicy:1dddfee0-3fa6-4145-a59e-c2c37e6203b8:autoScalingGroupName/lab-as-group:policyName/lab-scale-up-policy

TASK 7.2: CREATE A SCALE-DOWN POLICY

1. **Command:** To create a scale-down policy, run the following command:

aws autoscaling put-scaling-policy --policy-name lab-scale-down-policy --auto-scaling-group-name lab-as-group --scaling-adjustment -1 --adjustment-type ChangeInCapacity --cooldown 300 --query 'PolicyARN' --output text

Notice the policy name change as well as the slight change enclosing the *–scaling-adjustment -1*, which specifies a value of -1 in order to decrease the number of running instances by one each time this policy is executed.

**Expected output:** An ARN should be returned similar to the following example output:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

arn:aws:autoscaling:us-east-1:012345678910:scalingPolicy:66a6a1f4-a9e9-4a70-a935-319b3b082d17:autoScalingGroupName/lab-as-group:policyName/lab-scale-down-policy

TASK 7.3: CREATE A CLOUDWATCH HIGH CPU ALERT

In this task, you create a CloudWatch alarm to monitor the aggregate average CPU of the Auto Scaling fleet and to trigger the lab-scale-up-policy.

1. Switch back to the AWS Management Console.
2. At the top of the AWS Management Console, in the search bar, search for and choose

CloudWatch

.

1. In the navigation pane at the left of the page, expand the **Alarms** section and choose **In alarm**.
2. Choose **Create alarm**.
3. Choose **Select metric**.
4. Under **Metrics**, choose **EC2**.
5. Choose **By Auto Scaling Group**.
6. In the search  box, enter

CPUUtilization

.

1. Select the checkbox next to **lab-as-group**.
2. Choose the **Graphed metrics** tab, and configure the following:

* For **Statistic**, choose **Average**.
* For **Period** choose **1 minute**.

1. Choose **Select metric**.
2. Scroll down to the **Conditions** section, and configure the following:

* For **Whenever CPU Utilization is…** choose **Greater/Equal**.
* For **than…** enter

40

.

**Note:** This is referring to the instance using 40% CPU utilization or more for 1 datapoint in 1 time interval (1 minute).

1. Choose **Next**.
2. For **Notification**, configure the following:

* For **Alarm state trigger**, choose **In alarm**.
* For **Send a notification to the following SNS topic**, choose **Select an existing SNS topic**.
* For **Send a notification to…**, choose **lab-as-topic**.

**Note:** For **Email (endpoints)**, the email address you added to the SNS subscription will be displayed.

1. In the **Auto Scaling action** section, choose **Add Auto Scaling action**.

* For **Alarm state trigger**, choose **In alarm**.
* For **Resource type** choose **EC2 Auto Scaling group**.
* For **Select a group**, choose **lab-as-group**.
* For **Take the following action…**, choose **lab-scale-up-policy (Add 1 instance)**.

1. Choose **Next**.
2. For **Alarm name**, enter

High CPU Alarm

.

1. Choose **Next**.
2. Review your settings, and choose **Create alarm**.

TASK 7.4: CREATE A CLOUDWATCH LOW CPU ALERT

In this task, (using previous steps as a guide) configure a CPU utilization alert when usage is <=20% to trigger the lab-scale-down-policy.

1. Choose **Create alarm**.
2. Choose **Select metric**.
3. Under **Metrics**, select **EC2**.
4. Choose **By Auto Scaling Group**.
5. In the search  box, enter

CPUUtilization

.

1. Select the checkbox next to **lab-as-group**.
2. Choose the **Graphed metrics** tab, and configure the following:

* For **Statistic**, choose **Average**.
* For **Period** choose **1 minute**.

1. Choose **Select metric**.
2. Scroll down to the **Conditions** section, and configure the following:

* For **Whenever CPU Utilization is…** choose **Lower/Equal**.
* For **than…** enter

20

.

**Note:** This is referring to the instance using 20% CPU utilization or less for 1 datapoint in 1 time interval (1 minute).

1. Choose **Next**.
2. For **Notification**, configure the following:

* For **Alarm state trigger**, choose **In alarm**.
* For **Send a notification to the following SNS topic**, choose **Select an existing SNS topic**.
* For **Send a notification to…**, choose **lab-as-topic**.

**Note:** For **Email (endpoints)**, the email address you added to the SNS subscription will be displayed.

1. In the **Auto Scaling action** section, choose **Add Auto Scaling action**.

* For **Alarm state trigger**, choose **In alarm**.
* For **Resource type** choose **EC2 Auto Scaling group**.
* For **Select a group**, choose **lab-as-group**.
* For **Take the following action…**, choose **lab-scale-down-policy (Remove 1 instance)**.

1. Choose **Next**.
2. For **Alarm name**, enter

Low CPU Alarm

.

1. Choose **Next**.
2. Review your settings, and choose **Create alarm**.

**Note:** AWS recommends configuring Auto Scaling policies to scale up quickly and scale down slowly. This allows the application to better respond to increased traffic loads after a scale-up event and to make more economical use of the AWS hourly billing cycle. The example here is intentionally simple. From a billing perspective, it costs no more if the instance is scaled down after three minutes than if it is allowed to run for 59 minutes.

**Task complete:** You have successfully configured the Auto Scaling group to automatically scale up and scale down based on the average CPU utilization.

**Task 8: Test Auto Scaling**

All pieces are in place to demonstrate Auto Scaling based on application usage. You have an Auto Scaling group with a minimum of two instances and a maximum of four instances. You also have Auto Scaling policies to increase and decrease the group by one instance, and you created CloudWatch alarms to trigger policies when the aggregate average CPU utilization of the group is ≥ 40% and < 20%. Currently, one instance is running because the minimum size is one and the group is not currently under any load. Also, if you view your list of alarms, the two alarms you created should be in two different states: The High CPU alarm should be in the *OK* state (because your total CPU utilization is very low), and the Low CPU alarm should be in the *ALARM* state (because your total CPU utilization indicates that you have a surplus of processing cycles).

Even though your CPU utilization is below 20%, Auto Scaling is not removing instances because the group size is currently at its minimum (one). Also, remember that the cooldown time for the Auto Scaling polices is five minutes (300 seconds). This is important because it influences how quickly you see the Auto Scaling activities.

In this task, you see what happens with the scaling policies you have configured when you put additional strain on the servers.

1. At the top of the AWS Management Console, in the search bar, search for and choose

EC2

.

1. In the left navigation pane, choose **Load Balancers**.
2. Copy the public **DNS name** value of your load balancer to your notepad.
3. In a new browser window, paste the public DNS name in the address bar.

You should only see a single instance.

1. Choose **Generate Load**.

The CPU Load will jump to 100% (you may have to refresh your browser to see the CPU Load increase).

**Note:** This button triggers a simple background process to copy, zip, and unzip ~1 GB of nothing (/dev/zero) for 10–20 minutes.

1. Switch back to the AWS Management Console.
2. At the top of the AWS Management Console, in the search bar, search for and choose

CloudWatch

.

1. In the left navigation pane, choose **All alarms**.
2. Choose **High CPU Alarm**.

In a few minutes you should see the Low CPU alarm state change to *OK* and the High CPU alarm state change to *In alarm*. Additionally, you should receive an email notification from Auto Scaling informing you that a scale-up action was triggered.

Now return to the EC2 console and examine the instance autoscaling created.

1. At the top of the AWS Management Console, in the search bar, search for and choose

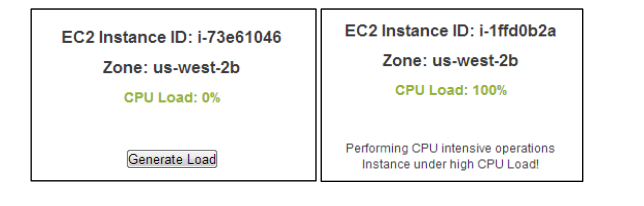
EC2

.

1. In the left navigation pane, choose **Instances**.

A new instance has been added to your group.

1. Switch to the browser tab that points to the A record for your load balancer, and refresh the load balancer several times to see that one server is under heavy load, while the other is not:



*Image description: The preceding diagram depicts a sample web page from the load balancer switching between two EC2 instances showing different CPU loads on those instances.*

1. Return to CloudWatch, to view the CPU utilization of the fleet.

You probably triggered another Auto Scaling scale-up event because your server fleet average CPU equaled 40% (one instance at ~100% and the other at ~0%), and the alarm trigger was set to ≥ 40%.

After 15–20 minutes, your Auto Scaling fleet should have scaled up to two or three instances and then scaled back down to one instance. Also note that the instances were terminated in launch order, meaning that the “oldest” instances were terminated first. This allows you to roll in new changes to your application by updating your launch configuration to a newer AMI and then triggering Auto Scaling events (increase the minimum size).

**Note:** You have probably noticed that this is not an overly realistic test because it essentially simulates a user overloading a single server and does not take advantage of your load balancer. In this case, Auto Scaling helps additional clients but does not load balance this “work” across multiple servers.

**Task complete:** You have successfully verified the autoscaling activities based on the scaling policies when you generated CPU load to strain the servers.

**Task 9: Wrapping Up**

In this task, you review some topics related to auto scaling activities and processes.

VIEWING AUTO SCALING ACTIVITIES

The Auto Scaling API provides a programmatic way to display all of the Auto Scaling activities that have occurred. You can use the

aws autoscaling describe-scaling-activities

 command to demonstrate this capability.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE COMMAND \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws autoscaling describe-scaling-activities --auto-scaling-group-name lab-as-group

SUSPENDING AND RESUMING AUTO SCALING PROCESSES

Auto Scaling also allows you to intentionally tell an Auto Scaling group to suspend or resume Auto Scaling processes using the as-suspend-processes and as-resume processes commands. This can be helpful if you know ahead of time that certain activities (such as maintenance events) will trigger Auto Scaling alerts, but you do not want instances to be automatically added or removed during the course of the event.

HOW LARGE AN AUTO SCALING FARM CAN I HAVE?

By default each account is assigned limits on a *per-region* basis.

**Learn more:** Refer to *Quotas for Auto Scaling resources and groups* in the **Additional resources** section for more information.

All of these defaults can be changed by making a request with AWS Support.

**Learn more:** Refer to *Contact AWS* in the **Additional resources** section for more information on how to make a request with AWS Support.

**Task complete:** You have successfully reviewed some topics related to auto scaling activities and processes.

**Conclusion**

You have successfully done the following:

* Created a new launch configuration using command-line tools.
* Created a new Auto Scaling group using command-line tools.
* Configured Auto Scaling notifications that are triggered when instance resources become too high or too low.
* Created policies to scale up or scale down the number of currently running instances in response to changes in resource utilization.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [IAM roles](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html)
* [Quotas for Auto Scaling resources and groups](https://docs.aws.amazon.com/autoscaling/ec2/userguide/ec2-auto-scaling-quotas.html)
* [Contact AWS](https://aws.amazon.com/contact-us/)
* [Auto Scaling Documentation](https://docs.aws.amazon.com/autoscaling/)
* [Auto Scaling Quick Reference Card](http://awsdocs.s3.amazonaws.com/AutoScaling/latest/as-qrc.pdf)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).