

## Simulating Dynamic Horizontal scaling / LifeCycle-Hooks and

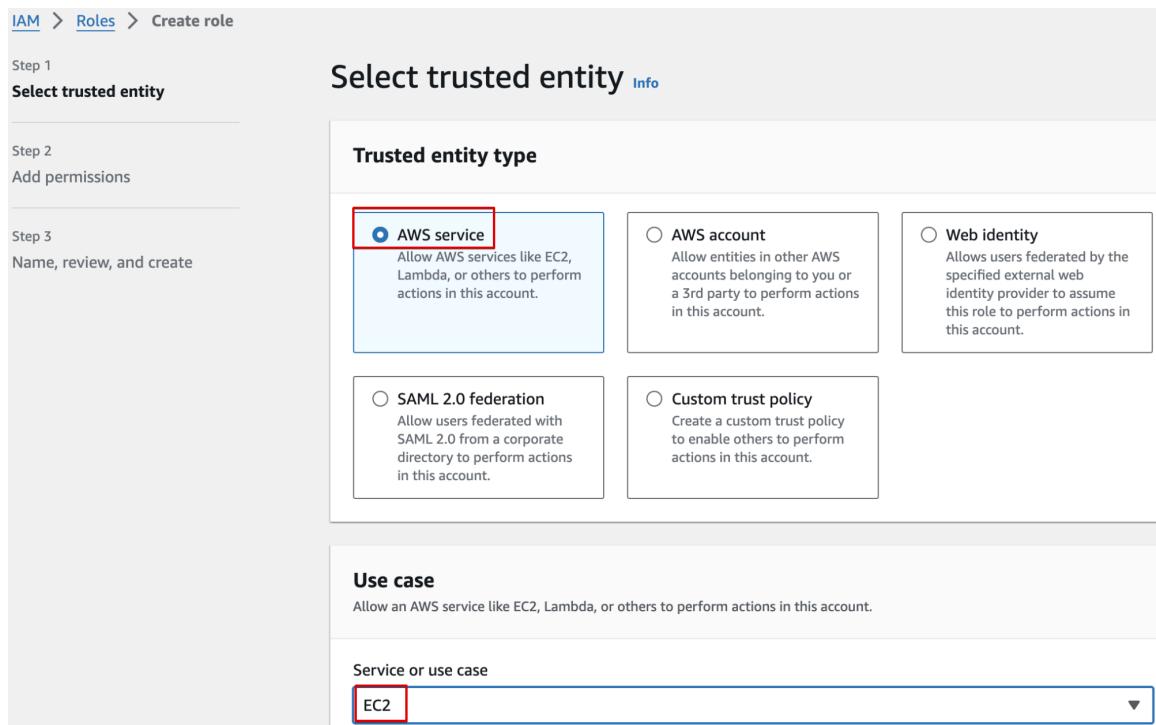
This runbook provides detailed steps to simulate horizontal and vertical scaling in AWS. We will create an EC2 launch template, an Auto Scaling Group (ASG), and configure dynamic scaling using CloudWatch alarms. The process demonstrates how to scale out and scale in based on CPU utilization metrics.

Additionally it will simulate life cycle hooks and termination policies

### 1. Create IAM Role for with AutoScalingFullAccess Policy

This step involves creating an IAM role with the necessary permissions for managing the Auto Scaling Group. We will attach the AutoScalingFullAccess policy to this role.

1. Open the IAM console.
2. In the left navigation pane, choose 'Roles'.
3. Choose '**Create role**'.
4. Select '**AWS service**' and choose '**EC2**'.



5. click on Next and add permissions to the role by attaching the 'AutoScalingFullAccess' policy to the role .

Add permissions Info

Permissions policies (1/947) Info

Choose one or more policies to attach to your new role.

Filter by Type

Policy name	Type
<input type="checkbox"/> <a href="#">AmazonElasticMapReduceforAutoScalingRole</a>	AWS managed
<input type="checkbox"/> <a href="#">ApplicationAutoScalingForAmazonAppStreamAccess</a>	AWS managed
<input type="checkbox"/> <a href="#">AutoScalingConsoleFullAccess</a>	AWS managed
<input type="checkbox"/> <a href="#">AutoScalingConsoleReadOnlyAccess</a>	AWS managed
<input checked="" type="checkbox"/> <a href="#">AutoScalingFullAccess</a>	AWS managed
<input type="checkbox"/> <a href="#">AutoScalingNotificationAccessRole</a>	AWS managed
<input type="checkbox"/> <a href="#">AutoScalingReadOnlyAccess</a>	AWS managed

6. click on **Next** and provide a name for the role (e.g., 'ASGRole') and choose '**Create role**'.

## 2. Create a Launch Template of an Instance

This step involves creating a launch template which defines the configuration details of the EC2 instances.

The template includes details such as the Amazon Machine Image (AMI), instance type, key pair, security groups, and other parameters required to launch an instance.

1. Open the Amazon EC2 console.
2. In the left navigation pane, choose 'Launch Templates'.

The screenshot shows the AWS EC2 Launch Template creation process. Step 1: A modal window titled 'New launch template' with a large orange 'Create launch template' button. Step 2: A main page titled 'Benefits and features' with two sections: 'Streamline provisioning' and 'Simplify permissions'. Step 3: A form titled 'Launch template name and description' where the user has entered 'jjitech-ASG-demo' for the name and 'version1 template for ASG demo' for the description. Other fields include 'Auto Scaling guidance' checked, and a section for 'Template tags' with one tag named 'demo-launch-template'.

3. Choose 'Create launch template'.

4. Provide a name and description and tag(optional) for the launch template.

The screenshot shows the 'Launch template name and description' form. The 'Name' field contains 'jjitech-ASG-demo'. The 'Description' field contains 'version1 template for ASG demo'. The 'Auto Scaling guidance' checkbox is checked. The 'Tags' section shows a single tag named 'demo-launch-template'.

Key	Value
Name	demo-launch-template

5. In the 'Launch template contents' section, specify the AMI, instance type, key pair, security group, and other configurations.

## 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.

### ▼ Application and OS Images (Amazon Machine Image) - required [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

#### Quick Start



Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

#### Amazon Machine Image (AMI)

**Amazon Linux 2023 AMI**

ami-01b799c439fd5516a (64-bit (x86), uefi-preferred) / ami-0e1ef59154d415994 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▾

### ▼ Instance type [Info](#) | [Get advice](#)

#### Instance type

**t2.micro**

Free tier eligible ▾

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.0716 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

[Additional costs apply for AMIs with pre-installed software](#)

### ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to your key pair before you launch the instance.

#### Key pair name

**ASG-launchTemplate**

**▼ Network settings** [Info](#)

Subnet [Info](#)

▼

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to your instance.

Select existing security group  Create security group

Security group name - *required*

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Valid characters: a-z, A-Z, 0-9, spaces, and .\_-:/()#,@[]+=&;{}!\$\*

Description - *required* [Info](#)

VPC [Info](#)

(default) ▼  
 172.31.0.0/16

Inbound Security Group Rules

No security group rules are currently included in this template. Add a new rule to include in the launch template.

6. Leave other settings (e.g storage) as default.

7. Under **Advanced details** sections:

- attach the **ASGrole** created above to the launch Template

▼ Advanced details [Info](#)

**IAM instance profile** [Info](#)

ASGrole  
arn:aws:iam::945685952191:instance-profile/ASGrole

Specify a custom value...

Don't include in launch template

admin  
arn:aws:iam::945685952191:instance-profile/admin

AmazonSSMRoleForInstancesQuickSetup  
arn:aws:iam::945685952191:instance-profile/AmazonSSMRoleForInstancesQuickSetup

**ASGrole**  
arn:aws:iam::945685952191:instance-profile/ASGrole

eks-68c7e6fd-e8d0-18b6-e776-a0de5ce2d7e4  
arn:aws:iam::945685952191:instance-profile/eks-68c7e6fd-e8d0-18b6-e776-a0de5ce2d7e4

OnboardingSpotInstances2021061708434336480000000a

**Stop - Hibernate behavior** [Info](#)

Don't include in launch template

**Termination protection** [Info](#)

- scroll to the bottom and add the provided user-data in the user data section. Either copy and paste or upload from file on local.

**User data - optional** [Info](#)

Upload a file with your user data or enter it in the field.

**Add user-data here**

**Choose file**

```
#!/bin/bash
yum install httpd -y && \
service httpd start && \
INSTANCE_ID=`wget -q -O - http://instance-data/latest/meta-data/instance-id` \
&& \
aws autoscaling complete-lifecycle-action --lifecycle-action-result CONTINUE -- \
instance-id $INSTANCE_ID --lifecycle-hook-name JJtechHook --auto-scaling-group- \
name demo-ASG --region us-east-1 || \
aws autoscaling complete-lifecycle-action --lifecycle-action-result ABANDON -- \
instance-id $INSTANCE_ID --lifecycle-hook-name JJtechHook --auto-scaling-group- \
name demo-ASG --region us-east-1
```

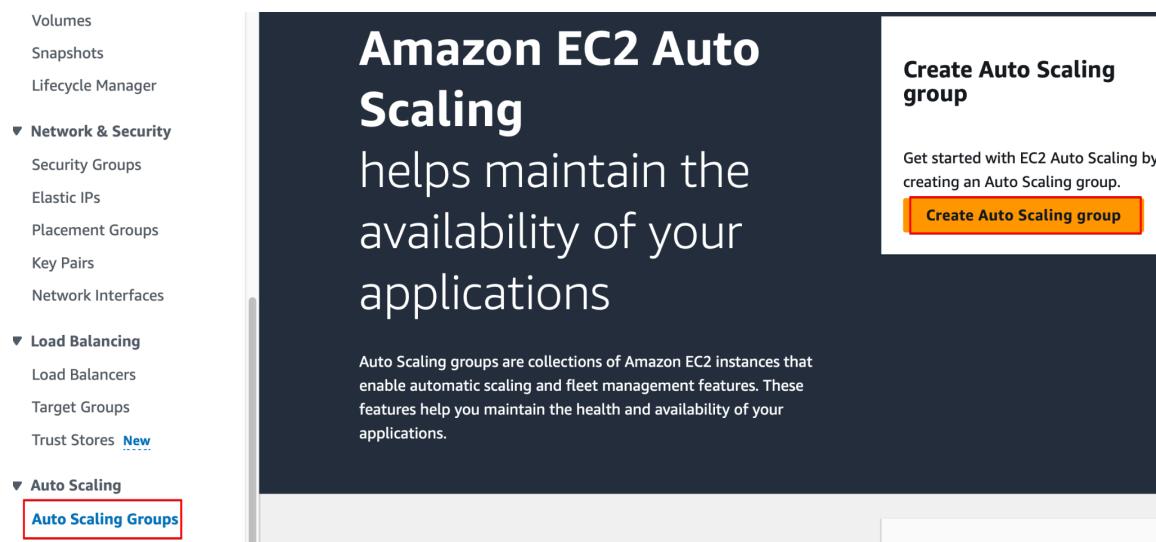
8. Review the settings and choose '**Create launch template**'.

### 3. Create an Auto Scaling Group

This step involves creating an Auto Scaling Group (ASG) using the launch template. The ASG ensures that the number of instances scales automatically based on the defined policies and rules.

1. Open the Amazon EC2 console.

2. In the left navigation pane, choose 'Auto Scaling Groups'.



3. Choose 'Create Auto Scaling group'.

4. Select the launch template created in the previous step and click on **Next**

In the *Choose instance launch options* section,

5. For instance type requirements, leave as default

6. Configure the VPC and subnets for the Auto Scaling group. Select subnets in different AZs in order to enable the ASG to launch instances in that AZ

In the *Configure Advanced Options* section, leave everything as default

In *Configure the group size and scaling* section,

7. Set the minimum capacity to , desired capacity , and maximum capacity .e.g. 2,4,6 respectively.

## Configure group size and scaling - optional Info

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

### Group size Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by automatic scaling.

#### Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▾

#### Desired capacity

Specify your group size.

4 ▾

### Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

#### Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
2	5

Equal or less than desired capacity Equal or greater than desired capacity

8. Leave other settings as default, scroll down and click on **Skip to review**

9. Review the settings and choose '**Create Auto Scaling group**'.

## 4. Configure Dynamic Scaling by Using Metrics

Dynamic scaling automatically adjusts the number of instances in the Auto Scaling group based on specified metrics. there are 3 types of dynamic scaling

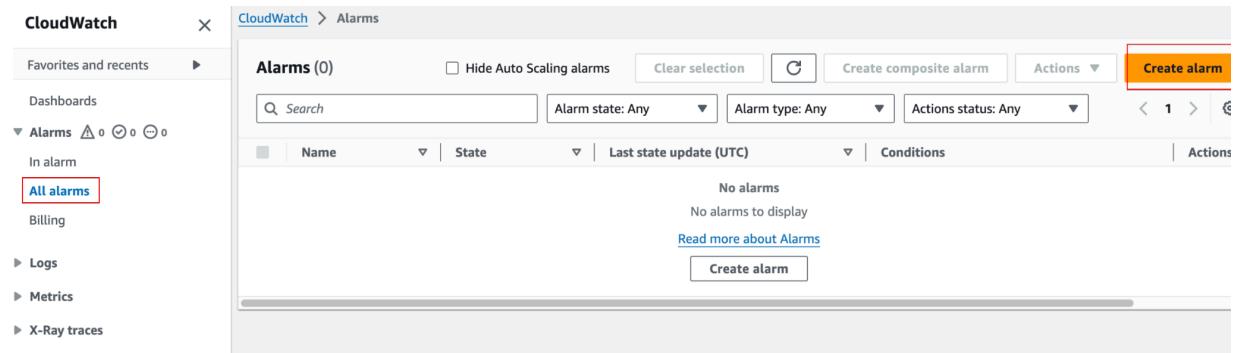
- Target tracking scaling (AWS creates the associated alarm)
- Step scaling, and
- Simple scaling (not recommended)

In this step, we will set up CloudWatch alarms and policies to scale out and scale in based on CPU utilization.

### a. Create CloudWatch Scale Out Alarm (70%)

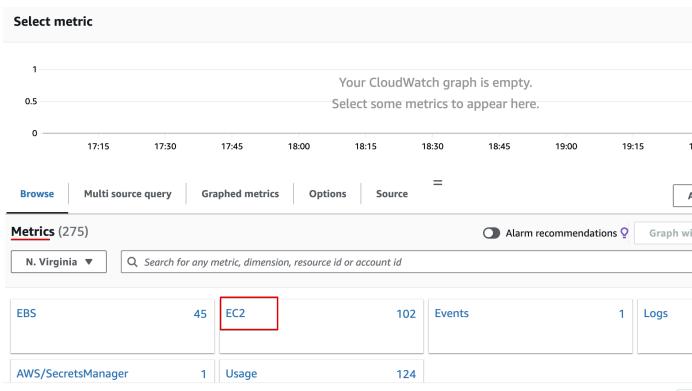
Create a CloudWatch alarm to trigger a scale out action when the CPU utilization exceeds 80%.

1. Open the CloudWatch console.
2. In the left navigation pane, choose 'Alarms', then 'Create Alarm'.

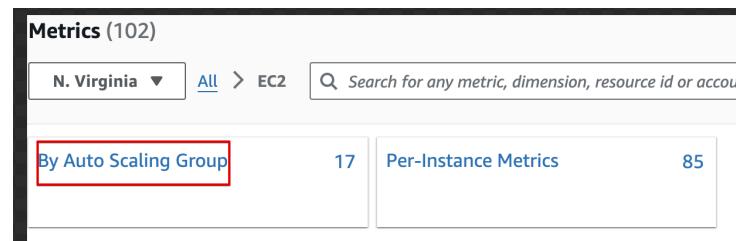


3. Choose Select metric

4. On the **metrics tab** choose EC2,



then Select the **By Auto Scaling Group** option.



If you have multiple ASGs in this region, Enter the ASG name in the search field.

Choose the '**CPUUtilization**' and choose **Select metric**.

The screenshot shows the 'Select metric' dialog box. At the top, there's a graph for 'CPUUtilization' from 17:30 to 20:15. Below the graph is a list of metrics from the 'ASG-demo' namespace. The 'CPUUtilization' metric is selected and highlighted with a red border. At the bottom right, there are 'Cancel' and 'Select metric' buttons, with 'Select metric' being the one highlighted.

5. Under the **Specify metric and Conditions** page, change period 1 minute

The screenshot shows the 'Specify metric and Conditions' page. On the left is a graph for 'CPUUtilization' from 18:00 to 20:00. On the right, there are configuration fields: 'Namespace' set to 'AWS/EC2', 'Metric name' set to 'CPUUtilization', 'AutoScalingGroupName' set to 'ASG-demo', 'Statistic' set to 'Average', and 'Period' set to '1 minute'. The 'Period' field is highlighted with a red border.

Set the threshold type to 'Static' (default) and the condition to '**Greater than 70%**

## Conditions

### Threshold type

Static

Use a value as a threshold

Anomaly detection

Use a band as a threshold

### Whenever CPUUtilization is...

Define the alarm condition.

Greater

> threshold

Greater/Equal

>= threshold

Lower/Equal

<= threshold

Lower

< threshold

than...

Define the threshold value.

70

Must be a number

click on **Next**

6. Notification: In the **Configure actions** page, create an SNS topic to send notifications when the CloudWatch alarm is triggered.

- select **In alarm**, to send notifications only when the alarm is IN ALARM state
- select **Create new topic** to create a new SNS topic
- fill other necessary details and select **create topic**.

## Configure actions

### Notification

#### Alarm state trigger

Define the alarm state that will trigger this action.

**Remove**

In alarm

The metric or expression is outside of the defined threshold.

OK

The metric or expression is within the defined threshold.

Insufficient data

The alarm has just started or not enough data is available.

#### Send a notification to the following SNS topic

Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic

Create new topic

Use topic ARN to notify other accounts

#### Create a new topic...

The topic name must be unique

jjtech-Autoscaling-demo-SNS-topic

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (\_).

#### Email endpoints that will receive the notification...

Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

me@gmail.com

user1@example.com, user2@example.com

**Create topic**

click on **Next**

7. Name the Alarm (e.g dynamicASG-ScaleOut-alarm), click on **Next**

8. Review and create the alarm.

**NB: Open the associated email endpoint and accept the SNS subscription.**

### b. Create CloudWatch Scale In Alarm (40%)

Create a CloudWatch alarm to trigger a scale in action when the CPU utilization falls below 40%. Note that this step creates the alarm, but no scale in action will be executed until the scaling policy is created.

**Repeat steps 1 to 4** above, for creating alarm at 70% utilization

5. Under the **Specify metric and Conditions** page, change period **1 minute** (same as above)

Set the threshold type to 'Static' (default) and the condition to '**lower than 40%**'

**Conditions**

Threshold type

Static  
Use a value as a threshold

Anomaly detection  
Use a band as a threshold

Whenever CPUUtilization is...

Define the alarm condition.

Greater  
> threshold

Greater/Equal  
>= threshold

Lower/Equal  
<= threshold

Lower  
< threshold

than...

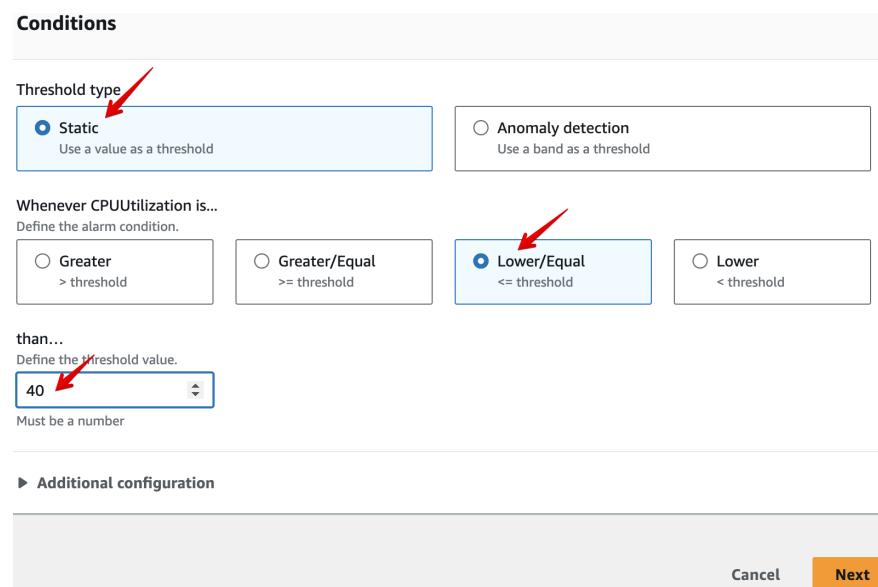
Define the threshold value.

40

Must be a number

► Additional configuration

Cancel      **Next**



6. Configure the actions to send a notification (e.g., email) when the alarm state is triggered.

**Configure actions**

**Notification**

**Alarm state trigger**  
Define the alarm state that will trigger this action.

**In alarm**  
The metric or expression is outside of the defined threshold.

**OK**  
The metric or expression is within the defined threshold.

**Insufficient data**  
The alarm has just started or not enough data is available.

**Send a notification to the following SNS topic**  
Define the SNS (Simple Notification Service) topic that will receive the notification.

**Select an existing SNS topic**

**Create new topic**

**Use topic ARN to notify other accounts**

**Send a notification to...**

jjtech-Autoscaling-demo-SNS-topic X

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Email (endpoints)

 [jjtech7@gmail.com](mailto:jjtech7@gmail.com) - [View in SNS Console](#) 

7. Name (**dynamicASG-ScaleIn-alarm**) the alarm. Review and create the alarm.

### c. Create Auto Scaling Group Scale Out Policy

Create a scaling policy that integrates with the CloudWatch scale out alarm to increase the number of instances when the alarm is triggered.

1. Open the Amazon EC2 console.
2. In the left navigation pane, choose **Auto Scaling Groups**.
3. Select the Auto Scaling group created earlier. This opens a split pane in the bottom.
4. Choose the **Automatic scaling** tab.
5. Choose **Create dynamic scaling policy**

The screenshot shows the AWS EC2 Dashboard with the 'Auto Scaling Groups' section selected. The main pane displays the 'Auto Scaling groups (1/1)' table, which contains one entry: 'ASG-demo' with a status of '0 instances'. Below the table, the 'Auto Scaling group: ASG-demo' details are shown, with the 'Automatic scaling' tab selected. A note about scaling policies is present, and the 'Dynamic scaling policies (0)' section is visible.

6. Set the policy type to **Step scaling** and Configure the policy to add 2 instances when the CPU utilization exceeds **70%**.

NB this policy is triggered when the associated CloudWatch Alarm is triggered.

### Create dynamic scaling policy

**Policy type:** Step scaling

**Scaling policy name:** DynamicASG-ScaleOut

**CloudWatch alarm:** Choose an alarm that can scale capacity whenever: dynamicASG-ScaleOut-alarm

**Take the action:**

- Add step: 2 capacity units when CPUUtilization <= 70

**Instance warmup:** 300 seconds

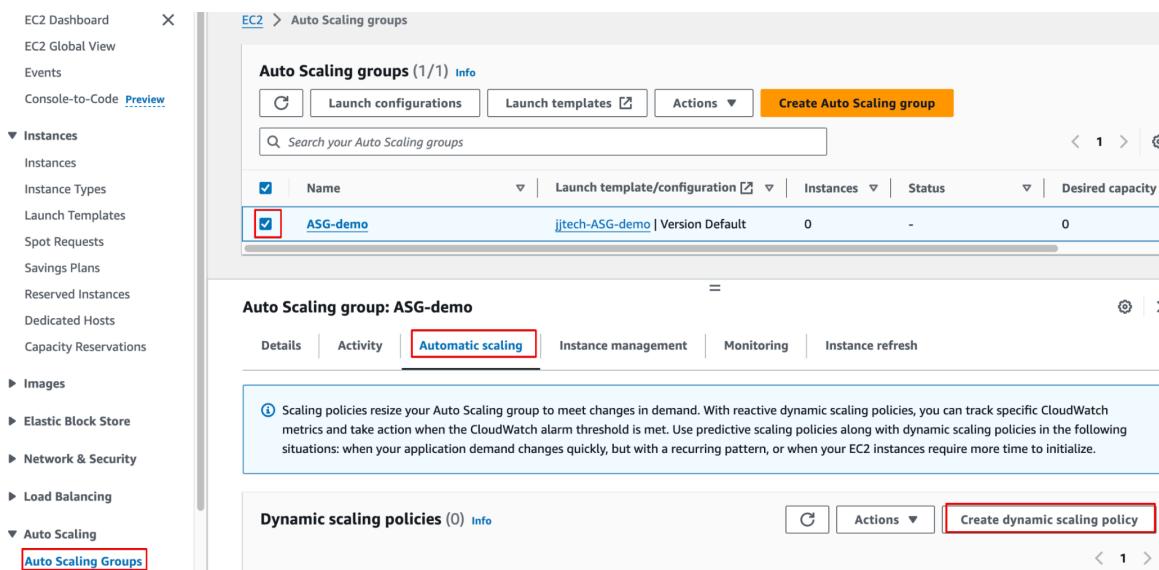
**Create**

8. Save and create the policy.

## d. Create Auto Scaling Group Scale In Policy

Create a scaling policy that integrates with the CloudWatch scale in alarm to decrease the number of instances when the alarm is triggered.

1. Open the Amazon EC2 console.
2. In the left navigation pane, choose **Auto Scaling Groups**.
3. Select the Auto Scaling group created earlier. This opens a split pane in the bottom.
4. Choose the **Automatic scaling** tab.
5. Choose **Create dynamic scaling policy**



6. Set the policy type to **Step scaling** and Configure the policy to remove 1 EC2 instance when the average CPU utilization for the ASG is less than **40%**.

NB this policy is triggered when the associated CloudWatch Alarm is triggered.

## Create dynamic scaling policy

Policy type

Scaling policy name

CloudWatch alarm  
Choose an alarm that can scale capacity whenever:

breaches the alarm threshold: CPUUtilization <= 40 for 1 consecutive periods of 300 seconds for the metric dimensions:

AutoScalingGroupName = ASG-demo

Take the action

capacity units when  >= CPUUtilization > -infinity

7. Save the policy.

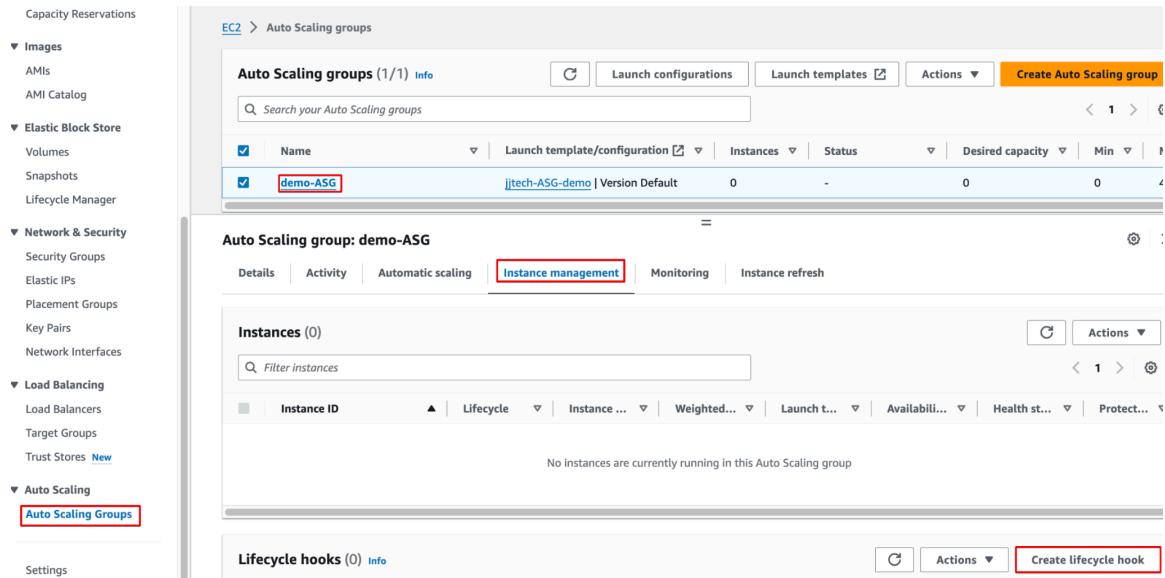
8. Adjust the minimum, desired and maximum capacity for the ASG and see how the Dynamic Scale in policy will keep changing the desired capacity for the ASG based on the rules set in the policy.

The scaleIn policy will adjust desired capacity until when Desired Capacity == Minimum Capacity.

## 5. Create a Lifecycle Hook for Launch Action

This step involves creating a lifecycle hook that will pause the instance in the 'Pending:Wait' state before it is fully launched. The lifecycle hook will be named 'JJtechHook'.

1. Open the Amazon EC2 console.
2. In the left navigation pane, choose **Auto Scaling Groups**.
3. Select the Auto Scaling group **demo-ASG** created earlier.
4. Choose the **Instance management** tab.
5. click on **create lifecycle hook**.



6. Provide the following details:

- Name: *JJtech-ASG-Hook*
- Lifecycle transition: Instance launch
- Default result: ABANDON
- Timeout: 3600 seconds

**Create lifecycle hook**

Lifecycle hook name  
 Must be unique to this group. Max 255 chars. No spaces or special characters except "-", "\_", and "/"

Lifecycle transition  
You can perform custom actions as EC2 Auto Scaling launches or terminates instances.

Heartbeat timeout  
The amount of time, in seconds, for the instances to remain in wait state.  
 seconds  
Minimum: 30, Maximum: 7200

Default result  
The action the Auto Scaling group takes when the lifecycle hook timeout elapses or if an unexpected failure occurs.

Notification metadata *(optional)*  
Additional information to include any time that EC2 Auto Scaling sends a message to the notification target.

[Learn how to receive lifecycle hook notifications](#)

7. Choose **Create**

8. After creating the lifecycle hook, observe it interrupt the EC2 launch cycle by pausing the launch cycle of the EC2 instances with a transition from **Pending** to **Pending:Wait state**, to give time for any custom actions on the instance

- Adjust the ASGs minimum, desired and maximum values so that new instances are launched.
- select the ASG and click on the **Instance management tab**. This opens a split pane displacing all EC2 instances under management by the ASG

**demo-ASG**

Details    Activity    Automatic scaling    **Instance management**    Monitoring    Instance refresh

**Instances (3)**

Instance ID	Lifecycle	Instance ...	Weighted...	Launch t...	Availabil...	Health st...	Protected...
<a href="#">i-03fda2f855fbfadcd</a>	Pending:Wait	t2.micro	-	<a href="#">jjtech-ASG-demo</a>	us-east-1a	<span>Healthy</span>	
<a href="#">i-08b866b7f40e08301</a>	Pending:Wait	t2.micro	-	<a href="#">jjtech-ASG-demo</a>	us-east-1c	<span>Healthy</span>	
<a href="#">i-0b552ef2dce41b23b</a>	Pending	t2.micro	-	<a href="#">jjtech-ASG-demo</a>	us-east-1b	<span>Healthy</span>	

**Lifecycle hooks (1) Info**

Name	Lifecycle transition	Default result	Heartbeat timeout...	Notification target...	Role ARN
JJtech-ASG-Hook	autoscaling:EC2_INST...	ABANDON	3600		

### ***Lifecycle-Hook UserData for the Launch Template.***

```
#!/bin/bash

yum install httpd -y && |

service httpd start && |

INSTANCE_ID="`wget -q -O - http://instance-data/latest/meta-data/instance-id`" && |

aws autoscaling complete-lifecycle-action --lifecycle-action-result CONTINUE --instance-id $INSTANCE_ID
--lifecycle-hook-name JJtech-ASG-Hook --auto-scaling-group-name demo-ASG --region us-east-1 || \

aws autoscaling complete-lifecycle-action --lifecycle-action-result ABANDON --instance-id $INSTANCE_ID
--lifecycle-hook-name JJtech-ASG-Hook --auto-scaling-group-name demo-ASG --region us-east-1
```