

# HW1

## Compute - Amazon EC2

### AWS Compute

Instance	Containers	Serverless	Edge and hybrid	Cost and capacity management
Amazon EC2	Amazon ECS	AWS Lambda	AWS Outposts	AWS Savings Plan
Amazon EC2 Spot	Amazon ECR		AWS Snow Family	AWS Compute Optimizer
Amazon EC2 Auto Scaling	Amazon EKS		AWS Wavelength	AWS Elastic Beanstalk
Amazon Lightsail	AWS Fargate		Vmware Cloud on AWS	EC2 Image Builder
AWS Batch			AWS Local Zones	Elastic Load Balancing

Building and running your organization starts with compute, whether you are building enterprise, cloud-native or mobile apps, or running massive clusters to drive analysis workloads. AWS offers a comprehensive portfolio of compute services allowing you to develop, deploy, run, and scale your applications and workloads in the world's most powerful, secure and innovative compute cloud.

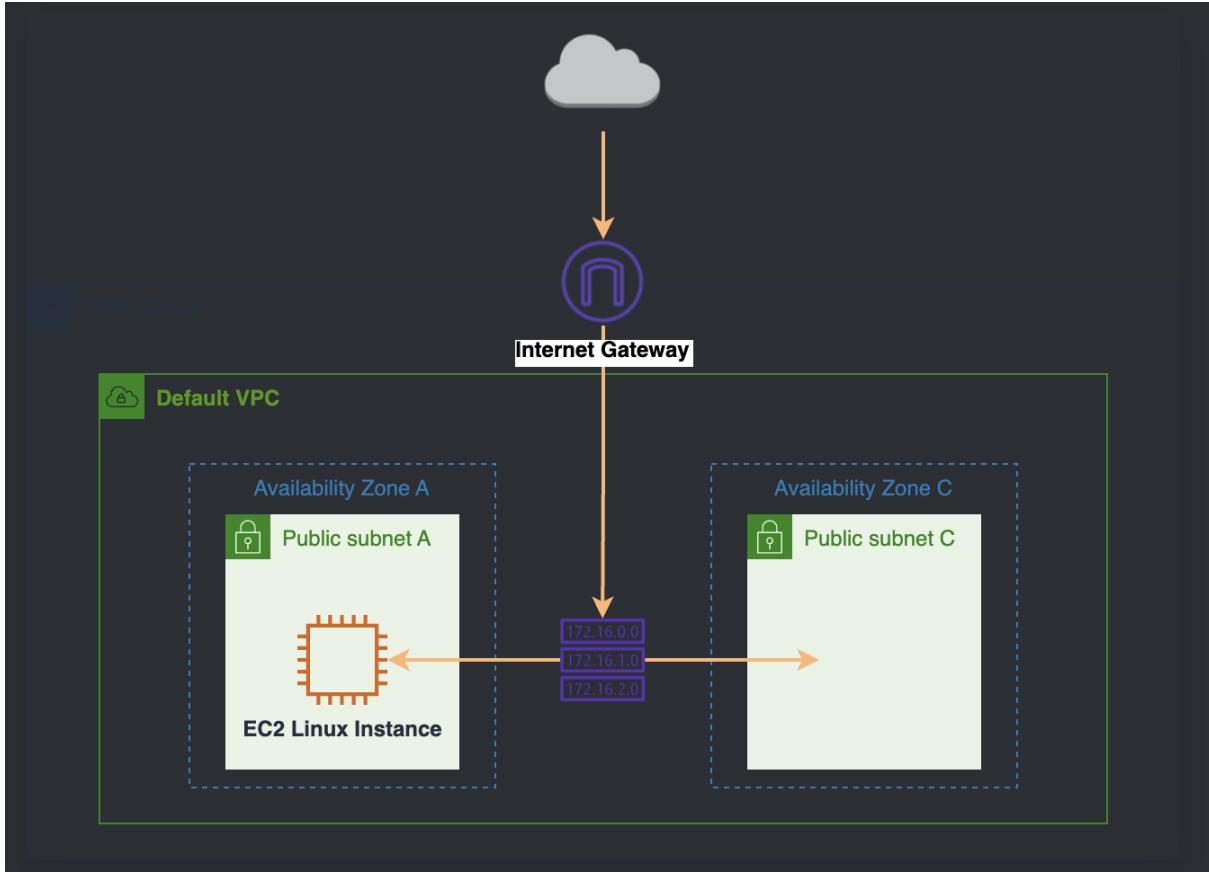
AWS computing services have the following characteristics:

- Right compute for your workloads
- Accelerate from idea to market
- Offer built in security
- Flexibility to optimize costs
- Provide compute resource where you need it

#### ▼ EC2 Linux Hands on Lab

### Amazon EC2 Overview

Amazon EC2 provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.



Create your own web server by going through the labs in the order below:

1. [Create a new key pair](#)

## Create a new Key Pair

### [Create a new Key Pair](#)

In this lab, you will need to create an EC2 instance using an SSH keypair. The following steps outline creating a unique SSH keypair for you to use in this lab.

1. Sign into the AWS Management Console and open the [Amazon EC2 console](#). In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region.
2. Click on **Key Pairs** in the Network & Security section near the bottom of the leftmost menu. This will display a page to manage your SSH key pairs.

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Images' section, there is a 'Key Pairs' link which is highlighted with a red box and a blue arrow pointing to it. The main content area has three sections: 'Launch instance', 'Scheduled events', and 'Migrate a server'. The 'Launch instance' section contains a 'Launch instance' button and a note about launching in the US East (N. Virginia) Region. The 'Scheduled events' section shows 'No scheduled events'. The 'Migrate a server' section provides information about using AWS Application Migration Service.

1. To create a new SSH key pair, click the **Create key pair** button at the top of the browser window.

The screenshot shows the 'Create key pair' wizard. The first step, 'Key pair', is displayed. It includes fields for 'Name' (containing 'AWS-ImmersionDay'), 'Key pair type' (set to RSA), 'Private key file format' (set to '.pem'), and 'Tags - optional'. At the bottom are 'Cancel' and 'Create key pair' buttons.

2. Type **[Your Name]-ImmersionDay** into the Key Pair Name: text box and click **Create key pair** button. For Windows users, please select **ppk** for file format.
3. The page will download the file **[Your Name]-ImmersionDay.pem** to the local drive. Follow the browser instructions to save the file to the default download location. Remember the full path to the key pair file you just downloaded.

## 2. Launch a Web Server Instance

# Launch a Web Server Instance

### Launch a Web Server Instance

We will launch an Amazon Linux 2 instance, bootstrap Apache/PHP, and install a basic web page that will display information about our instance.

1. Click on **EC2 Dashboard** near the top of the leftmost menu. And Click on **Launch instances**.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (with 'Instances New'), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances New', 'Dedicated Hosts', 'Capacity Reservations', and 'Images' (with 'AMIs New' and 'AMI Catalog'). The main area is titled 'Resources' and displays the following resource counts:

Instances (running)	0	Dedicated Hosts	0	Elastic IPs	0
Instances	0	Key pairs	1	Load balancers	0
Placement groups	0	Security groups	1	Snapshots	0
Volumes	0				

A callout box provides information about launching Microsoft SQL Server Always On availability groups using the AWS Launch Wizard for SQL Server.

In the center, there's a 'Launch instance' section with a green checkmark icon and the word 'Click'. Below it are two buttons: 'Launch Instance' and 'Migrate a server'.

On the right, there's a 'Service health' section showing the region as 'Asia Pacific (Seoul)' and a 'Status' link.

2. In **Name**, put the value **Web server for IMD**. And check the default setting for Amazon Machine Image below.

**Name and tags** [Info](#)

Name  
Web server for IMD [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** [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

Recents My AMIs Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat S

Mac ubuntu® Microsoft Red Hat >

Amazon Machine Image (AMI)

**Amazon Linux 2023 AMI** [Free tier eligible](#)

ami-022e1a32d3f742bd8 (64-bit (x86)) / ami-0b54418bdd76353ce (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.0.20230614.0 x86\_64 HVM kernel-6.1

Architecture AMI ID Verified provider

64-bit (x86) ami-022e1a32d3f742bd8

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

3. Select **t2.micro** in Instance Type.

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

Instance type

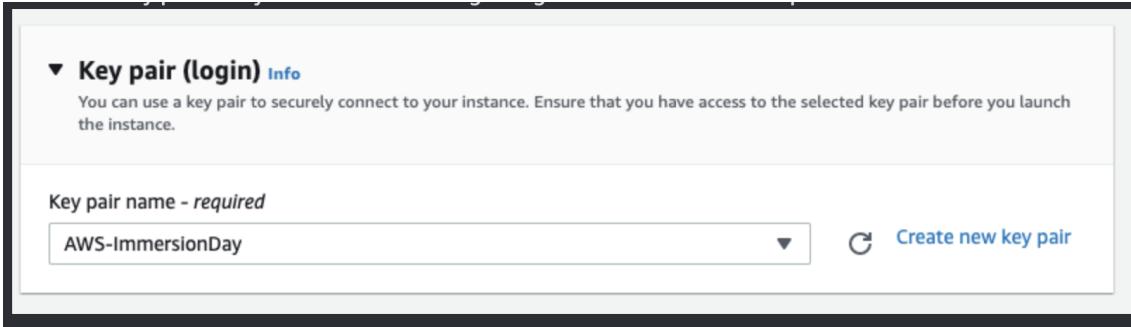
**t2.micro** [Free tier eligible](#)

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

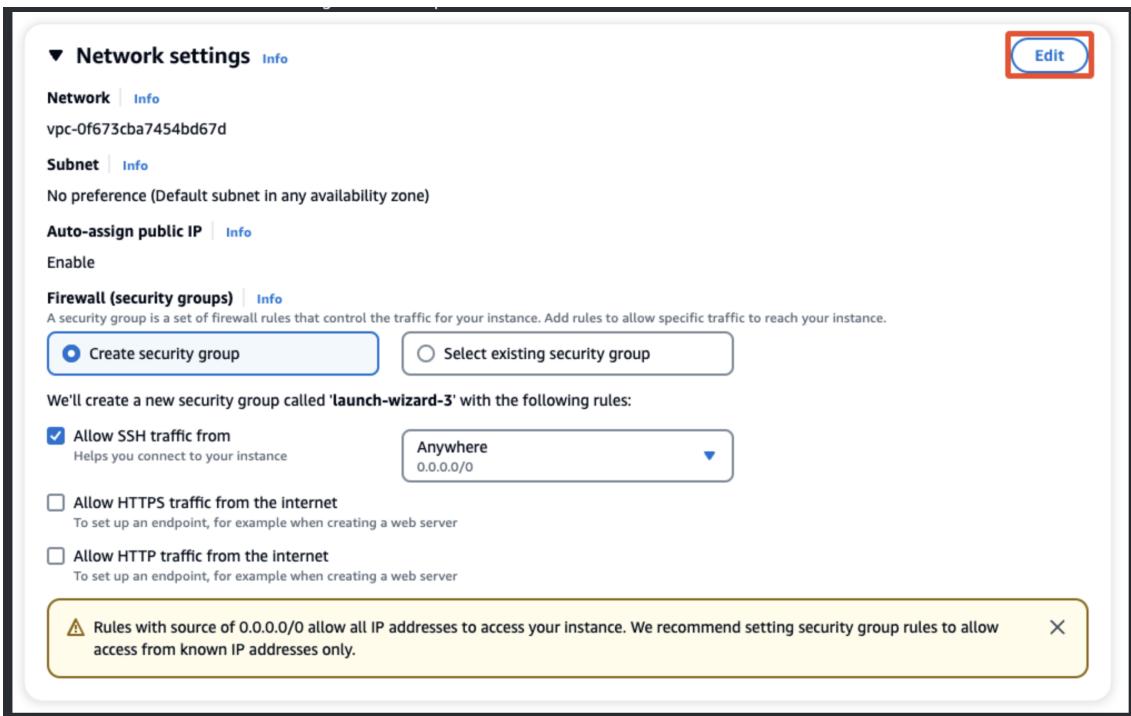
All generations [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

4. Select the key pair that you created in the beginning of this lab from the drop-down.



5. Click the Edit button in Network settings to set the space where EC2 will be located.



Check **default VPC** and **subnet**. **Auto-assign public IP** is set to **Enable**. Right below it, create **Security groups** to act as a network firewall. Security groups will specify the protocols and addresses you want to allow in your firewall policy. For the security group you are currently creating, this is the rule that applies to the EC2 that will be created. After entering *Immersion Day - Web Server* in Security group name and Description, select Add Security group rule and set HTTP to Type.

**▼ Network settings**

**VPC - required Info**

vpc-0587c58f5e4d82b38	(default) ▾	
172.31.0.0/16		

**Subnet Info**

subnet-059b9087c44372eaf	▼	
VPC: vpc-0587c58f5e4d82b38 Owner: 025482651656		
Availability Zone: ap-northeast-2a IP addresses available: 4091		

**Create new subnet**

**Auto-assign public IP Info**

Enable	▼
--------	---

**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 reach your instance.

Create security group     Select existing security group

**Security group name - required**

Immersion Day - Web Server
----------------------------

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-:/()#,@[]+=&;!\$\*

**Description - required Info**

Immersion Day - Web Server
----------------------------

**Inbound security groups rules**

**▼ Security group rule 1 (TCP, 22, 54.239.119.3/32)**

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info	Source Info	Description - optional Info
My IP	<input type="text"/> Add CIDR, prefix list or security group ██████████ X	e.g. SSH for admin desktop

Also allow TCP/80 for Web Service by specifying it. Select **My IP** in the source.

**Inbound security groups rules**

▼ Security group rule 1 (TCP, 22, [REDACTED]) Remove

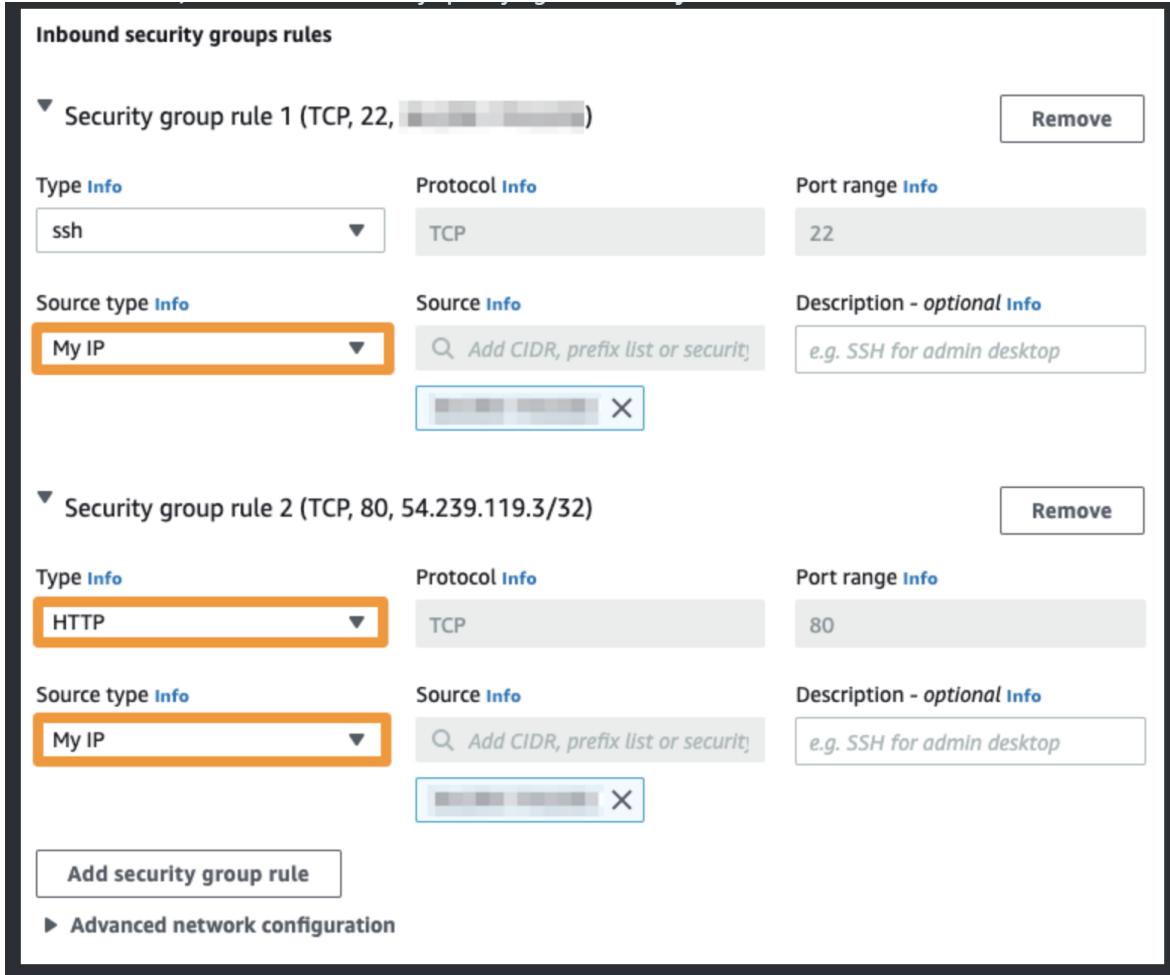
Type <a href="#">Info</a> ssh	Protocol <a href="#">Info</a> TCP	Port range <a href="#">Info</a> 22
Source type <a href="#">Info</a> My IP	Source <a href="#">Info</a> <input type="text"/> Add CIDR, prefix list or security [REDACTED] X	Description - optional <a href="#">Info</a> e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 54.239.119.3/32) Remove

Type <a href="#">Info</a> HTTP	Protocol <a href="#">Info</a> TCP	Port range <a href="#">Info</a> 80
Source type <a href="#">Info</a> My IP	Source <a href="#">Info</a> <input type="text"/> Add CIDR, prefix list or security [REDACTED] X	Description - optional <a href="#">Info</a> e.g. SSH for admin desktop

Add security group rule

► Advanced network configuration



1. All other values accept the default values, expand by clicking on the **Advanced Details** tab at the bottom of the screen.

▼ Configure storage [Info](#) Advanced

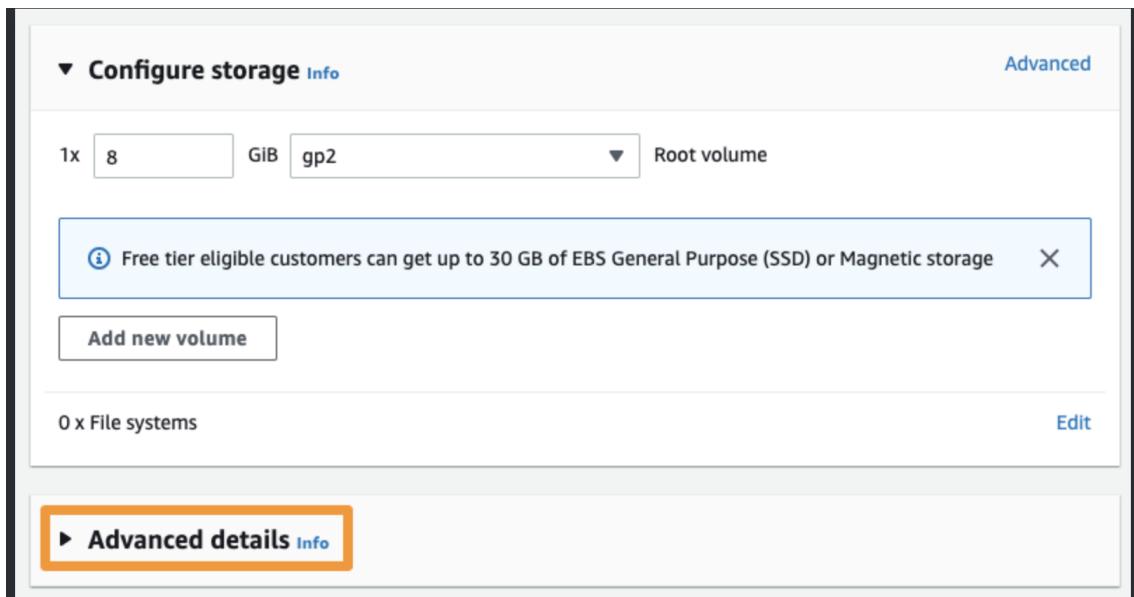
1x <input type="text" value="8"/> GiB	gp2	Root volume
---------------------------------------	-----	-------------

i Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

Add new volume

0 x File systems Edit

► Advanced details [Info](#)



Click the Meta Data version dropdown and select V2 only (token required) Enter the following values in the User data field and select Launch instance.

```
#!/bin/sh

#Install a LAMP stack
dnf install -y httpd wget php-fpm php-mysqli php-json php php-devel
dnf install -y mariadb105-server
dnf install -y httpd php-mbstring

#Start the web server
chkconfig httpd on
systemctl start httpd

#Install the web pages for our lab
if [ ! -f /var/www/html/immersion-day-app-php7.zip ]; then
cd /var/www/html
wget -O 'immersion-day-app-php7.zip' \
https://static.us-east-1.prod.workshops.aws/9c1aea7d-0009-44d3-875b-c0fa6583c6bd/assets/immersion-day-app-php7.zip?Key-Pair-Id=K36Q2WVO3JP7QD&Policy=eyJTdGF0ZW1bnQiOlt7IlJlc291cmNljoiaHR0cHM6Ly9zdGF0aWMudXMtZWFzdCG9csBXufeimzCdYMNEpD1HUsOnwWh2ALitEAcj714EqDw7swM-
ttwIBxv3pVCMq1iZI2BRnZ42LHdE9MGWvuqJ~6HtySrKAIv~ozrnypw4nPXCCAP0RvMbeY7ifCyIPq3Vy4CL7dCUvfFkxz56kNx7HBO8pEbzeUlgSXZIPr6KX65UrrCQHIEpAmKVYfwp3PZqRJIP5eAh2rNqu2NilCw8iibViofIUVZ0Hv6U~pHvypp6Pqrce4OLfUEudy8WMvyRjSjBaagHhjYQ_'
unzip immersion-day-app-php7.zip
fi

#Install the AWS SDK for PHP
if [ ! -f /var/www/html/aws.zip ]; then
cd /var/www/html
mkdir vendor
cd vendor
wget
https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.zip
unzip aws.zip
fi

# Update existing packages
dnf update -y
```

1. Click the **View Instances** button in the lower right hand portion of the screen to view the list of EC2 instances. Once your instance has launched, you will see your Web Server as well as the Availability Zone the instance is in, and the publicly routable **DNS name**. Click the checkbox next to your web server to view details about this EC2 instance.

The screenshot shows the AWS EC2 Instances console. At the top, there's a search bar and a filter for 'Instance state = running'. Below the header is a table with columns: Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. One row is selected, showing 'Web server for IMD' as the name, instance ID 'i-089d22683ed0fc34e', instance state 'Running', instance type 't2.micro', 2/2 checks passed, and no alarms. The 'Status check' column contains a link to 'open address'. In the middle section, under 'Instance summary', there are fields for Instance ID ('i-089d22683ed0fc34e'), Public IPv4 address ('3.35.210.158'), Private IPv4 addresses ('172.31.43.213'), IPv6 address ('-'), Instance state ('Running'), and Public IPv4 DNS ('ec2-3-35-210-158.ap-northeast-2.compute.amazonaws.com'). The 'Public IPv4 DNS' field is highlighted with an orange box.

### Browse the Web Server

Wait for the instance to pass the Status Checks to finish loading. Open a new browser tab and browse the Web Server by entering the EC2 instance's **Public DNS name** into the browser. The EC2 instance's Public DNS name can be found in the console by reviewing the **Public IPv4 DNS** name line highlighted above. You should see a website that looks like the following.

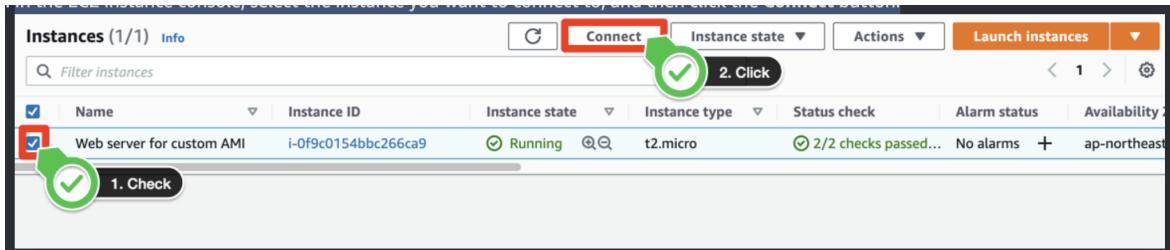
If you are using the Chrome web browser, when you attach the **Public IPv4 DNS** value to the web browser, if it does not run, https may be automatically added in front of the DNS value, so it may not run. Therefore, it is recommended to enter http://.

The screenshot shows a web page with the AWS logo at the top. Below the logo, there are two tabs: 'LOAD TEST' and 'RDS'. Underneath the tabs, there is a table with two rows. The first row has 'Meta-Data' in bold and 'Value' in bold. The second row lists 'Instanceld' with the value 'i-0f9c0154bbc266ca9' and 'Availability Zone' with the value 'ap-northeast-2c'. Below the table, the text 'Current CPU Load: 1%' is displayed.

**Great Job! You have deployed a server and launched a web site in a matter of minutes!**

### 3. Connect to your linux instance

In the EC2 instance console, select the instance you want to connect to, and then click the **Connect** button.



In the **Connect to instance** page, select **SSH client**. Follow the instructions below.

The screenshot shows the 'Connect to instance' page for instance `i-0f9c0154bbc266ca9`. The page has a breadcrumb navigation: EC2 > Instances > `i-0f9c0154bbc266ca9` > Connect to instance. The main title is 'Connect to instance' with an 'Info' link. Below it, a sub-instruction says 'Connect to your instance `i-0f9c0154bbc266ca9` (Web server for custom AMI) using any of these options'. There are three tabs: 'EC2 Instance Connect', 'Session Manager', and 'SSH client' (which is highlighted with a red box). Below the tabs, the 'Instance ID' is listed as `i-0f9c0154bbc266ca9` (Web server for custom AMI). A 'Click' button with a green checkmark icon is positioned next to the 'SSH client' tab. The page then lists steps for connecting via SSH:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is `AWS-ImmersionDay.pem`.
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
`chmod 400 AWS-ImmersionDay.pem`
4. Connect to your instance using its Public DNS:  
`ec2-52-78-23-186.ap-northeast-2.compute.amazonaws.com`

Below these steps is an 'Example:' section with a command:  
`ssh -i "AWS-ImmersionDay.pem" ec2-user@ec2-52-78-23-186.ap-northeast-2.compute.amazonaws.com`

A note in a callout box says: **Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Navigate to where your private key is located and enter the following command. Replace **[Your Name]** with the name you specified prior when you created the key.

EC2 > Instances > i-0f9c0154bbc266ca9 > Connect to instance

### Connect to instance Info

Connect to your instance i-0f9c0154bbc266ca9 (Web server for custom AMI) using any of these options

EC2 Instance Connect | Session Manager | **SSH client**

Instance ID  
 i-0f9c0154bbc266ca9 (Web server for custom AMI)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is AWS-ImmersionDay.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 AWS-ImmersionDay.pem 

4. Connect to your instance using its Public DNS:  
 ec2-52-78-23-186.ap-northeast-2.compute.amazonaws.com

Example:  
 ssh -i "AWS-ImmersionDay.pem" ec2-user@ec2-52-78-23-186.ap-northeast-2.compute.amazonaws.com

**Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

chmod 400 [Your Name]-ImmersionDay.pem

Next, enter the following command in your SSH client to connect to your Linux instance.

EC2 > Instances > i-0f9c0154bbc266ca9 > Connect to instance

### Connect to instance Info

Connect to your instance i-0f9c0154bbc266ca9 (Web server for custom AMI) using any of these options

EC2 Instance Connect Session Manager **SSH client**

Instance ID

**i-0f9c0154bbc266ca9 (Web server for custom AMI)**

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is AWS-ImmersionDay.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 AWS-ImmersionDay.pem
4. Connect to your instance using its Public DNS:  
 ec2-52-78-23-186.ap-northeast-2.compute.amazonaws.com

Example:

ssh -i "AWS-ImmersionDay.pem" ec2-user@ec2-52-78-23-186.ap-northeast-2.compute.amazonaws.com

**Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

```
ssh -i "[Your Name]-ImmersionDay.pem" ec2-user@<Public IPv4 DNS>
```

After connecting continue query to yes, You can check the result as below.

```
joozero ~]~Downloads>
> ssh -i "AWS-ImmersionDay.pem" ec2-user@ec2-54-180-134-238.ap-northeast-2.compute.amazonaws.com
The authenticity of host 'ec2-54-180-134-238.ap-northeast-2.compute.amazonaws.com (54.180.134.238)' can't be established.
ECDSA key fingerprint is SHA256:COXKVOpjh2GLgiUOWXWuZrfFDj7ZUEtIx6T72hfxc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-180-134-238.ap-northeast-2.compute.amazonaws.com,54.180.134.238' (ECDSA) to the list of known hosts.
Last login: Sun Dec 19 23:29:47 2021 from ec2-13-209-1-56.ap-northeast-2.compute.amazonaws.com

  _|_ _|_
  | (   /  Amazon Linux 2 AMI
  _\|_|_|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-0-234 ~]$ ls
[ec2-user@ip-172-31-0-234 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-0-234 ~]$
```

4. Connect to your Linux instance using Session Manager (Optional).

## Connect to your Linux instance using Session Manager (Optional)

Session Manager is a fully managed AWS Systems Manager capability that lets you manage your Amazon EC2 instances through an interactive one-click browser-based shell or through the AWS CLI. You can use Session Manager to start a session with an instance in your account. After the session is started, you can run bash commands as you would through any other connection type.

## Create an IAM instance profile for Systems Manager

1. Sign in to the AWS Management Console and open the [IAM console](#). In the navigation pane, choose Roles, and then choose Create role.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

**Roles**  Click

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCPs)

Introducing the new IAM dashboard experience  
We've redesigned the IAM dashboard experience to make it easier to use. Let us know what you think.

## IAM dashboard

### Security recommendations

**Add MFA for root user**  
Enable multi-factor authentication (MFA) for the root user to improve security for this account.

### IAM resources

User groups	Users	Roles	Policies	Identity providers
0	1	18	3	0

#### What's new?

Update for features in IAM

View all | 

- IAM Access Analyzer helps you generate fine-grained policies that specify the required actions for more than 50 services. 4 months ago
- IAM Access Analyzer helps you generate IAM policies based on access activity found in your organization that is more than 60 days old.
- IAM Access Analyzer adds new policy checks to help validate conditions during IAM policy authoring. 6 months ago
- AWS Amplify announces support for IAM permissions boundaries on Amplify-generated IAM roles. 6 months ago

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

**Roles**  Click

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCPs)

Introducing the new IAM roles experience  
We've redesigned the IAM roles experience to make it easier to use. Let us know what you think.

Home > Roles

### Roles (18)

An IAM role is a temporary identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AMIServerRoleForAmazonDynamoDB	AMIServerRoleForAmazonDynamoDB	AMIServerRoleForAmazonDynamoDB
AMIServerRoleForAmazonCloudWatchLogs	AMIServerRoleForAmazonCloudWatchLogs	AMIServerRoleForAmazonCloudWatchLogs
AMIServerRoleForAmazonCloudWatchMetrics	AMIServerRoleForAmazonCloudWatchMetrics	AMIServerRoleForAmazonCloudWatchMetrics
AMIServerRoleForAmazonCloudWatchLogsAutoscaling	AMIServerRoleForAmazonCloudWatchLogsAutoscaling	AMIServerRoleForAmazonCloudWatchLogsAutoscaling
AMIServerRoleForAmazonCloudWatchMetricsAutoscaling	AMIServerRoleForAmazonCloudWatchMetricsAutoscaling	AMIServerRoleForAmazonCloudWatchMetricsAutoscaling
AMIServerRoleForAmazonCloudWatchLogsScalableTarget	AMIServerRoleForAmazonCloudWatchLogsScalableTarget	AMIServerRoleForAmazonCloudWatchLogsScalableTarget
AMIServerRoleForAmazonCloudWatchMetricsScalableTarget	AMIServerRoleForAmazonCloudWatchMetricsScalableTarget	AMIServerRoleForAmazonCloudWatchMetricsScalableTarget
AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogs	AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogs	AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogs
AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetrics	AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetrics	AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetrics
AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogsCloudWatchLogs	AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogsCloudWatchLogs	AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogsCloudWatchLogs
AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetricsCloudWatchMetrics	AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetricsCloudWatchMetrics	AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetricsCloudWatchMetrics
AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogsCloudWatchLogsCloudWatchLogs	AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogsCloudWatchLogsCloudWatchLogs	AMIServerRoleForAmazonCloudWatchLogsCloudWatchLogsCloudWatchLogsCloudWatchLogs
AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetricsCloudWatchMetricsCloudWatchMetrics	AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetricsCloudWatchMetricsCloudWatchMetrics	AMIServerRoleForAmazonCloudWatchMetricsCloudWatchMetricsCloudWatchMetricsCloudWatchMetrics

 Search

2. Under **Select type of trusted entity**, choose **AWS service**. Immediately under **Choose the service that will use this role**, choose **EC2**, and then choose **Next**.

3. On the **Attach permissions policies** page, do the following: Use the **Search** field to locate the **AmazonSSMManagedInstanceCore**. Select the box next to its name. Choose **Next**.

The screenshot shows the 'Add permissions' step of the AWS IAM Role creation wizard. The title bar says 'Add permissions'. Below it, a section titled 'Permissions policies (Selected 1/730)' with the sub-instruction 'Choose one or more policies to attach to your new role.' A search bar contains 'AmazonSSMManagedInstanceCore' with a magnifying glass icon and a dropdown showing '1 match'. To the right are 'Create Policy' and 'Next Step' buttons. A large red box highlights the search bar and the policy name. Another red box highlights the checkbox next to the policy name 'AmazonSSMManagedInstanceCore'.

4. For Role name, enter a name for your new instance profile, such as **SSMInstanceProfile**. Choose **Create role**. The system returns you to

4. For Role name, enter a name for your new instance profile, such as `SSMInstanceProfile`. Choose **Create role**. The system returns you to Name, review, and create

**Name, review, and create**

**Role details**

**Role name**  
 Role names must be unique to the account.

**Description**  
Add a brief description for this policy.  
 Allow EC2 instances to call AWS services on your behalf

Maximum 1000 characters. Use alphanumeric and '-' characters.

**Step 1: Select trusted entities**

```
1 { "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "sts:AssumeRole", "Principal": "arn:aws:iam::123456789012:root" } ] }
```

**Step 2: Add permissions**

**Permissions policy summary**

Policy name	Type	Attached as
AmazonSSMManagedInstanceCore	AWS managed	Permissions policy

**Tags**

**Add tags (Optional)**  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or control the resource.

No tags associated with this resource

**Add tag**  
You can add up to 500 tags.

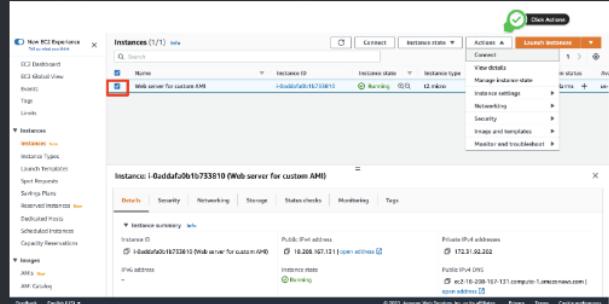
**Create role**

the Roles page.

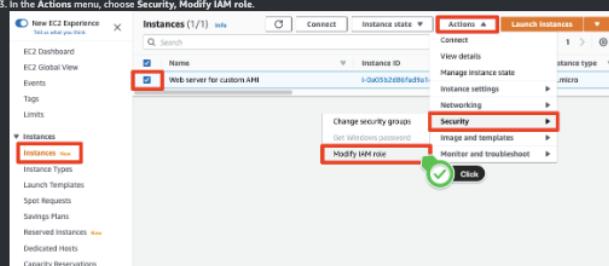
④ Make a note of the role name. You will choose this role when you create new instances that you want to manage by using Systems Manager.

#### Attach the Systems Manager instance profile to an existing instance (console)

1. Sign in to the AWS Management Console and open the Amazon EC2 console at [Amazon EC2 console](#).
2. In the navigation pane, under Instances, choose Instances. Choose your EC2 instance from the list and click Actions.



3. In the Actions menu, choose Security, Modify IAM role.



4. For IAM role, select the instance profile you created `SSMInstanceProfile`.

**EC2 > Instances > i-0a03b2d8fad9a1d6 > Modify IAM role**

**Modify IAM role - info**  
Attach an IAM role to your instance.

**Instance ID**

**IAM role**  
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

**Choose IAM role**  
 Search for this column to detach an IAM role.

**No IAM role**  
Create a new IAM role.

**SSMInstanceProfile** arn:aws:iam::123456789012:instance-profile/SSMInstanceProfile

**TeamRoleInstanceProfile** arn:aws:iam::123456789012:instance-profile/TeamRoleInstanceProfile

**Save**

EC2 > Instances > i-0a03b2d86fad9a1d6 > Modify IAM role

**Modify IAM role** Info

Attach an IAM role to your instance.

Instance ID  
i-0a03b2d86fad9a1d6 (Web server for custom AMI)

IAM role  
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

SSMInstanceProfile

Cancel

5. Choose Save.

**Connect to your Linux instance using Session Manager**

- In the EC2 instance console, select the instance you want to connect to, and then click the **Connect** button.

1. Click the green checkmark icon next to the instance name.

2. Click the 'Connect' button.

- Review the **Session Manager usage** section for advantages of using Session Manager.
- Choose Connect. A new session will be started in a new tab. After the session is started, you can run bash commands as you would through any other connection type.

EC2 > Instances > i-0a03b2d86fad9a1d6 > Connect to instance

**Connect to instance** Info

Connect to your instance i-0a03b2d86fad9a1d6 (Web server for custom AMI) using any of these options

EC2 Instance Connect  **Session Manager**  SSH client  EC2 Serial Console

**Session Manager usage:**

- Connect to your instance without SSH keys or a bastion host.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

Cancel

**⚠️** If you receive an error like shown below, wait for few seconds and refresh your browser. Behind the scenes the EC2 instance is being setup for use with Session Manager

EC2 > Instances > i-034588912ae28c8ae > Connect to instance

**Connect to instance** Info

Connect to your instance i-034588912ae28c8ae using any of these options

EC2 Instance Connect  **Session Manager**  SSM client  EC2 Serial Console

**We weren't able to connect to your instance. Common reasons for this include:**

- SSM Agent isn't installed on the instance. You can install the agent on both [Windows instances](#) and [Linux instances](#).
- The required IAM instance profile isn't attached to the instance. You can attach a profile using [AWS Systems Manager Quick Setup](#).
- Session Manager setup is incomplete. For more information, see [Session Manager Prerequisites](#).

**Session Manager usage:**

- Connect to your instance without SSH keys or a bastion host.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

Cancel

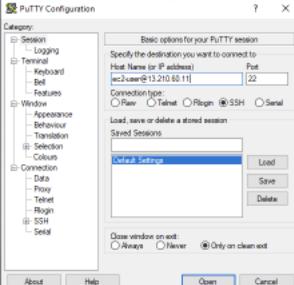
5. Connect to EC2 Instance using PuTTy (Optional)

Event dashboard > Basic Modules > Compute + Amazon EC2 > EC2 Linux Hands on Lab > Connect to EC2 Instance using PuTTy (Optional)

## Connect to EC2 Instance using PuTTy (Optional)

**Connect to EC2 Instance using PuTTy**

1. Start PuTTy if you need to download PuTTY.  
2. In the Category pane, choose Session.  
3. In the Host Name box enter ec2-user@[your public IP of EC2 that you created].  
4. Set the Port value to 22.



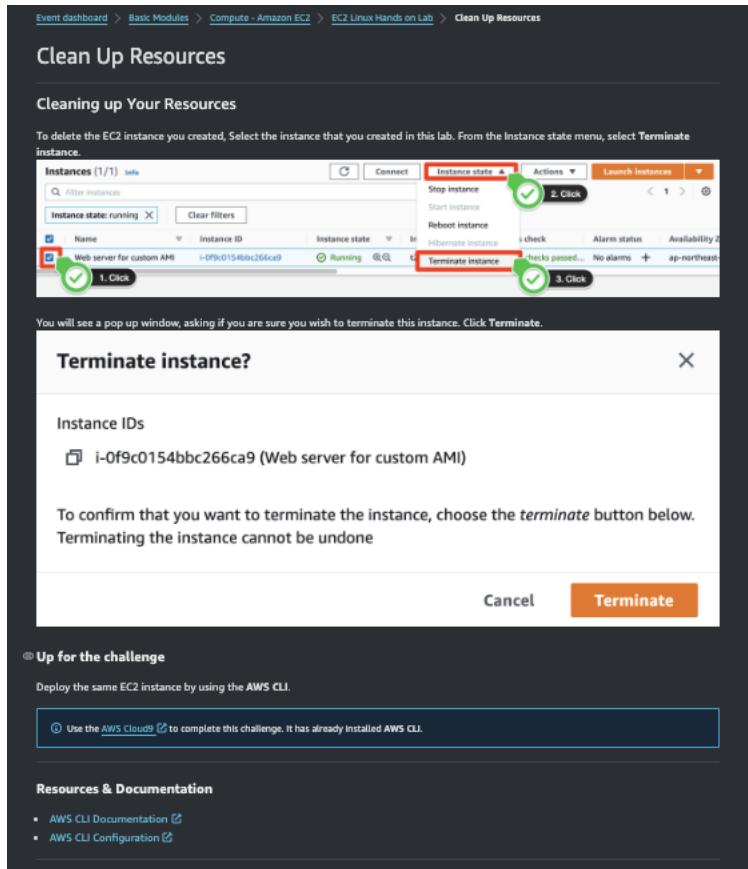
The screenshot shows the PuTTY Configuration window. The 'Session' category is selected in the left sidebar. The 'Host Name (or IP address)' field contains 'ec2-user@13.210.68.11'. The 'Port' field is set to '22'. Under 'Connection type', the radio button for 'SSH' is selected. The 'Saved Sessions' dropdown shows 'Default Settings'. At the bottom, the 'Open' button is highlighted in blue, while 'About', 'Help', and 'Cancel' are greyed out.

5. Under Connection type, select SSH.  
6. In the Category pane, expand Connection, expand SSH, and then choose Auth. Complete the following:

- Choose Browse.
- Select the .ppk file that you generated for your key pair and choose Open.

7. If this is the first time you have connected to this instance, PuTTY displays a security alert dialog box that asks whether you trust the host to which you are connecting. Choose Yes. A window opens and login as ec2-user and you are connected to your instance.

[Previous](#) [Next](#)



## ▼ Auto Scaling on AWS

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes. The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas.

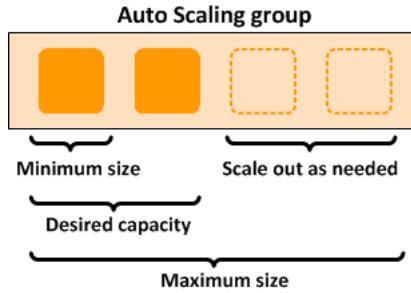
AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you're already using Amazon EC2 Auto Scaling to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

### EC2 Auto Scaling Lab Overview

#### What is Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

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



### Features of Amazon EC2 Auto Scaling

With Amazon EC2 Auto Scaling, your EC2 instances are organized into Auto Scaling groups so that they can be treated as a logical unit for the purposes of scaling and management. Auto Scaling groups use launch templates (or launch configurations) as configuration templates for their EC2 instances.

The following are key features of Amazon EC2 Auto Scaling:

**Monitoring the health of running instances** Amazon EC2 Auto Scaling automatically monitors the health and availability of your instances using EC2 health checks and replaces terminated or impaired instances to maintain your desired capacity.

**Custom health checks** In addition to the built-in health checks, you can define custom health checks that are specific to your application to verify that it's responding as expected. If an instance fails your custom health check, it's automatically replaced to maintain your desired capacity.

**Balancing capacity across Availability Zones** You can specify multiple Availability Zones for your Auto Scaling group, and Amazon EC2 Auto Scaling balances your instances evenly across the Availability Zones as the group scales. This provides high availability and resiliency by protecting your applications from failures in a single location.

**Multiple instance types and purchase options** Within a single Auto Scaling group, you can launch multiple instance types and purchase options (Spot and On-Demand Instances), allowing you to optimize costs through Spot Instance usage. You can also take advantage of Reserved Instance and Savings Plan discounts by using them in conjunction with On-Demand Instances in the group.

**Automated replacement of Spot Instances** If your group includes Spot Instances, Amazon EC2 Auto Scaling can automatically request replacement Spot capacity if your Spot Instances are interrupted. Through Capacity Rebalancing, Amazon EC2 Auto Scaling can also monitor and proactively replace your Spot Instances that are at an elevated risk of interruption.

**Load balancing** You can use Elastic Load Balancing or VPC Lattice load balancing and health checks to ensure an even distribution of application traffic to your healthy instances. Whenever instances are launched or terminated, Amazon EC2 Auto Scaling automatically registers and deregisters the instances from the load balancer. **Scalability** Amazon EC2 Auto Scaling also provides several ways for you to scale your Auto Scaling groups. Using auto scaling allows you to maintain application availability and reduce costs by adding capacity to handle peak loads and removing capacity when demand is lower. You can also manually adjust the size of your Auto Scaling group as needed.

**Instance refresh** The instance refresh feature provides a mechanism to update instances in a rolling fashion when you update your AMI or launch template. You can also use a phased approach, known as a canary deployment, to test a new AMI or launch template on a small set of instances before rolling it out to the whole group.

**Lifecycle hooks** Lifecycle hooks are useful for defining custom actions that are invoked as new instances launch or before instances are terminated. This feature is particularly useful for building event-driven architectures, but it also helps you manage instances through their lifecycle.

**Support for stateful workloads** Lifecycle hooks also offer a mechanism for persisting state on shut down. To ensure continuity for stateful applications, you can also use scale-in protection or custom termination policies to prevent instances with long-running processes from terminating early.

For more information about the benefits of Amazon EC2 Auto Scaling, see [Amazon EC2 Auto Scaling benefits](#).

## **Work with Auto Scaling groups**

You can create, access, and manage your Auto Scaling groups using any of the following interfaces:

- **AWS Management Console** – Provides a web interface that you can use to access your Auto Scaling groups. If you've signed up for an AWS account, you can access your Auto Scaling groups by signing into the AWS Management Console, using the search box on the navigation bar to search for **Auto Scaling groups**, and then choosing **Auto Scaling groups**.
- **AWS Command Line Interface (AWS CLI)** – Provides commands for a broad set of AWS services, and is supported on Windows, macOS, and Linux. To get started, see [Prepare to use the AWS CLI](#). For more information, see [autoscaling](#) in the *AWS CLI Command Reference*.
- **AWS Tools for Windows PowerShell** – Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see the [AWS Tools for Windows PowerShell User Guide](#). For more information, see the [AWS Tools for PowerShell Cmdlet Reference](#).
- **AWS SDKs** – Provides language-specific API operations and takes care of many of the connection details, such as calculating signatures, handling request retries, and handling errors. For more information, see [AWS SDKs](#).
- **Query API** – Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access AWS services. However, it requires your application to handle low-level details such as generating the hash to sign the request, and handling errors. For more information, see the [Amazon EC2 Auto Scaling API Reference](#).
- **AWS CloudFormation** – Supports creating Auto Scaling groups using CloudFormation templates. For more information, see [Create Auto Scaling groups with AWS CloudFormation](#).

To connect programmatically to an AWS service, you use an endpoint. For information about endpoints for calls to Amazon EC2 Auto Scaling, see [Amazon EC2 Auto Scaling endpoints and quotas](#) in the *AWS General Reference*.

An overview of auto scaling on AWS can be found [here](#). This lab will first walk you through the process of creating an Amazon Machine Image (AMI) from a web host created with CloudFormation. Then it will walk you through creating a launch template and setting up the web host within an auto scaling group behind an Application Load Balancer (ALB). The end result will be an auto scaling group behind a load balancer that scales based on CPU utilization of the hosts.

### **Services and concepts covered in this lab:**

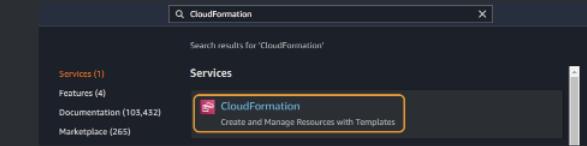
- Primary: Auto Scaling, Launch Templates, Creation and configuration of Security Groups
- Secondary: AMIs, Application Load Balancers

## Lab Prerequisites

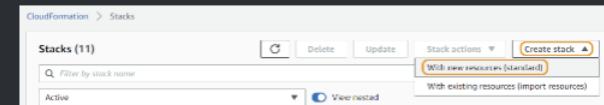
To create an AMI (Amazon Machine Image) for our Auto-Scaling group we will first need to setup a web host. We will generate an AMI from the instance and then auto-scale the instance behind a load balancer. Click on the template link button below to build the web host in EC2 using CloudFormation.

### Download and launch the CloudFormation template

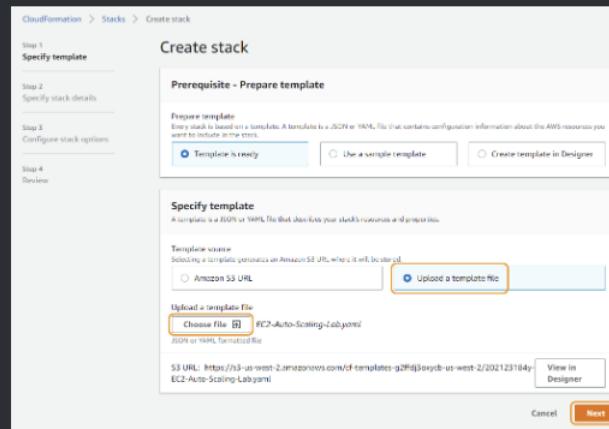
1. Download the "EC2-Auto-Scaling-Lab.yaml" CloudFormation template by right-clicking on this link and save it to your local hard drive.
2. In the AWS Console search for CloudFormation or select the Services menu and click on CloudFormation under "Management & Governance".



3. In the CloudFormation console select the Create stack button and then select With new resources (standard).

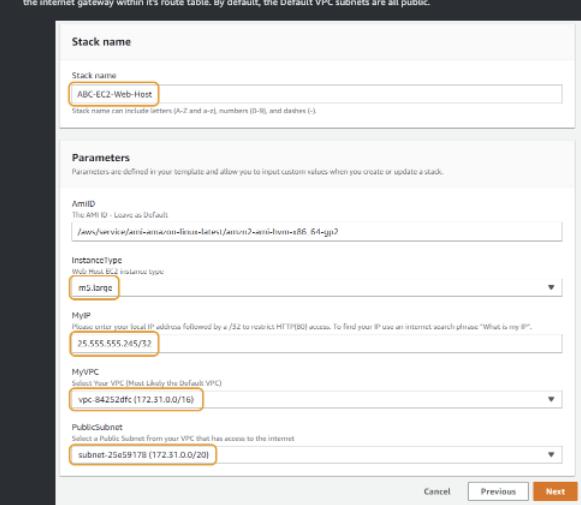


4. Under "Template source" select Upload a template file and then select the Choose file button. Select the "EC2-Auto-Scaling-Lab.yaml" template file you downloaded in the first step. Once you have selected the template file click on the Next button.



5. On the Specify stack details page, fill in the following fields:

- a. Under "Stack name" name yourstack [Your Initials]-EC2-Web-Host.
- b. You can leave "AmiID" as the default, resulting in the use the most recent version of this AMI.
- c. Under "InstanceType" select the m5.large or the t2.micro. It is recommended that you use m5.large size instances to demonstrate real world performance as t2 types instances are not recommended for production workloads. If you have an issue with the m5 instances for any reason, load the CloudFormation template again and select the t2.micro, it is sufficient enough for this lab.
- d. Under "MyIP" input the IP address of your local machine followed by a /32. This will lock down HTTP port 80 to your machine. You can find your local IP by searching What is my IP.
- e. Under "MyVPC" select the VPC you want to use to setup the instance. In most accounts the default VPC will be a good choice and in new AWS accounts it will be the only choice.
- f. Under "PublicSubnet" select a subnet within your VPC that has internet access. A public subnet is defined by a subnet having a route to the internet gateway within its route table. By default, the Default VPC subnets are all public.



6. Once you are done entering the details above, click on **Next**. On the next page, "Configure stack options", you can leave "Tags", "Permissions", and "Advanced options" as default and select **Next**.

7. On the Review [Your Initials]-EC2-Web-Host page, review your settings and click on **Create stack** to start building your web server.

8. Wait till the "Logical ID" "[Your Initials]-EC2-Web-Host" shows a status of "CREATE\_COMPLETE".

The CloudFormation stack creation should be completed in about 3 minutes.

**Confirm the successful setup of your instance:**

1. Navigate to the EC2 service page by selecting **Services** and then **EC2** or search for **EC2** in the search bar.
2. Select **Instances** from the left hand menu. On the "Instances" page, select your instance "[Your Initials]-EC2-Web-Host" and copy the "Public IPv4 DNS" address from your clipboard by clicking on the image of two overlapping squares to the left of the Public IPv4 DNS Address. Paste this address into a new tab on your web browser.

Clicking on the open address link under "Public IPv4 DNS" heading may result in you not being able to see your website. The "open address" link uses <https://> instead of <http://>, which will result with an error because our web host has not been setup with an SSL cert.

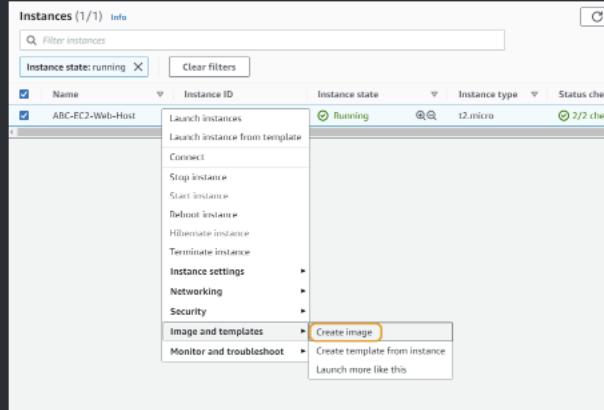
3. You should now see the from page titled "EC2 Instance Metadata".

If the page does not load it is recommended you wait for your Instance "Status check" to show "2/2 checks passed" and then try again. Your metadata will look similar but will not match the image above.

### Generate a custom AMI of the web server created in the EC2 - Linux Lab:

Now that we have our instance setup to host our website, we will generate a custom machine image for our auto scaling group. This will create an image of our web host that will be used by our Auto Scaling group to spin up multiple instances based on server load.

1. In the EC2 Console under Instances, you can create Amazon Machine Images (AMIs) from either running or stopped instances. Return to or open the EC2 console.
2. Right-click your webhost instance named "[Your Initials]-Web Server" and under "Image and templates" choose Create image from the context menu. (This can also be done by selecting the instance and clicking on the Actions menu in the upper right hand corner)



3. On the "Create Image" page, put in the Image name "[Your Initials]\_Auto\_Scaling\_Webhost" and a description. You can leave the Instance volumes as default and then choose Create Image.

**Create Image**

An image (AMI) refers to an AMI definition that defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID: **ami-0c0e0d12b6723f1 (ABC-EC2-Web-Host)**

Image name: **[Your Initials]\_Auto\_Scaling\_Webhost**

Description: **ABC - Auto Scaling Webhost**

Image description: **ABC - Auto Scaling Webhost**

Root volume:  EBS

Instance volumes:

Volume type	Device	Snapshots	Size	Volume type	IOPS	Throughput	Data encryption at termination	Encrypted
EBS	/dev/sda1	Snapshot 1	8	EBS General Purpose SS	100	100	<input checked="" type="checkbox"/> Static	<input type="checkbox"/> Static

**Starting the image creation process. Amazon EC2 creates a snapshot of each of the above volumes.**

Tags (optional):

- Tag image and snapshots together: Tag the image and the snapshots with the same tag.
- Tag image and snapshots separately: Tag the image and the snapshots with different tags.

Tags associated with the resource:

**Add tag**

You can add 50 more tags.

**Create Image**

4. It may take a few minutes for the AMI to be created. In the EC2 console under "Images" in the left hand menu select AMIs. You should see the AMI you just created, it may be in a pending state, but after a few moments it will transition to an available state.

**Amazon Machine Images (AMIs) (1)**

Name	AMI ID	AMI name	Source	Owner	Visibility	Status
-	ami-0c0e0d12b6723f1	[Your Initials]_Auto_Scaling_Webhost	-	-	-	Available

We are done with our new Amazon Machine Image for now, we can now move on to setting up our auto scaling security group.

### Create a new Security Group for our Auto Scaling Group:

Before we get into setting up our Launch Template we will need to setup a special Security Group for our Auto Scaling Group. A security group provides instance (virtual machine) level protection. If you want to know more about security groups, check out this page.

1. Within the console Under "Services" select EC2 or search EC2 in the search bar. On the EC2 page under the "Network & Security" heading in the left-hand menu select Security Groups. You should see other security groups, including the security group for your web server named "[Your Initials]-EC2-Web-Host - Website Security Group". To start the creation of a new security group, click on the Create security group button.

**Security Groups (2)**

Name	Security group ID	Security group name	VPC ID
sg-01c50f12e5844bcb5	ABC - Website Security Group	vpc-046de34d2c5310edf	
sg-07ac96ca5a7d7af1	default	vpc-046de34d2c5310edf	

**Create security group**

2. Name your security group [Your\_Initials] - Auto Scaling SG and you can use the same name for the description as well and make sure you have the correct VPC selected. (Most likely the Default VPC unless you setup a new one for this lab)

3. Under "Inbound rules" there currently are not any rules created, we will leave it empty for now. We will be creating a rule later in this lab but we need our load balancer security group to exist first.

4. "Outbound rules" currently allow all traffic out so there is no need for any additional rules.

Now click on **Create security group**.

**Basic details**

Security group name: ABC - Auto Scaling SG  
Description: ABC - Auto Scaling SG  
VPC: vpc-03a1d083fe223f03 (Default VPC)

**Inbound rules**

This security group has no inbound rules.  
Add rule

**Outbound rules**

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

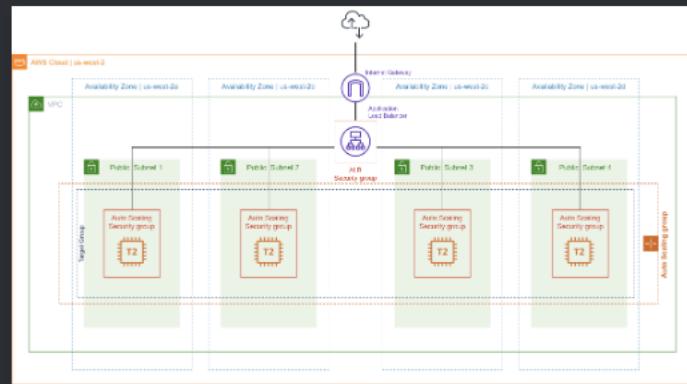
Add rule Cancel Create security group

You are now finished with the prerequisites and ready to move onto the next step

## Creating a Launch Template

### Auto Scaling Lab Architecture:

Below is a diagram of our end state architecture. Let's get building!



### There are three main components to EC2 Auto Scaling on AWS

- Launch Template:** A Launch Template is a feature of EC2 Auto Scaling that allows a way to template 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 a specific Amazon Machine Image, instance type, storage, and networking settings that you typically use to launch instances. For each Launch Template, you can create one or more numbered Launch Template Versions. Each version can have different launch parameters.
- Auto Scaling Groups:** For auto scaling 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.
- 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.

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. (Even if the instances are determined to be unhealthy)
- 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 to specify the number of healthy instances your auto scaling group should have at all times. More information can be found [here](#).
- 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.

### 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.

- Under the "Services" select EC2.
- In the left navigation pane, find "Instances" and select Launch Templates.
- Now select Create launch template.

The screenshot shows the "Launch templates" page in the AWS EC2 console. At the top, there is a search bar and a "Create launch template" button. Below the header, there is a table with columns for "Launch template ID", "Launch template name", and "Default version". A message at the bottom states "You have no launch templates in this region".

4. This takes you to the "Create launch template" page, starting with the "Launch template name and description":

- Launch template name:** [Your Initials]-scaling-template
- Template version description:** This is optional
- Auto scaling guidance:** Check the box to provide guidance
- "Launch Template Contents"** defines the parameters for the instances in the Auto Scaling group:
  - Amazon machine image (AMI):** Select "My AMIs" and "Owned by me". In the drop down search by typing your initials and select the custom AMI you just created [Your Initials]\_Auto\_Scaling\_Webhost. (The new AMI you just created may already be selected)
  - Instance type:** t2.micro
  - Key pair (Login):** Select the Key Pair you created in the first lab, it is most likely call [Your Initials]-KeyPair
  - Networking Settings:**
    - Subnet:** Don't include in launch template
    - Firewall (security groups):** Select "Select existing security group" and then select the security group you created in the first part of this lab named [Your Initials] - Auto Scaling SG
  - Configure storage:** Leave as default
  - Resource tags:** None
  - Advanced Details:** IMPORTANT: Select the arrow to expand "Advanced details" and under "Detailed Cloudwatch monitoring" select Enable. Leave everything else as the default.

Here, you are enabling CloudWatch Detailed monitoring. By default, your instance has basic monitoring in 5-minute intervals for the instances. After you enable detailed monitoring CloudWatch will monitor the instances in your auto scaling group in 1-minute intervals. This will allow the auto scaling group to respond quicker to changes in the group.

**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.

**Launch template name and description**

**Launch template name - required**  
ABC-scaling-template

Must be unique to this account. Max 128 chars. No spaces or special characters like \$, %, &.

**Template version description**  
V1

Max 255 chars.

**Auto Scaling guidance info**  
Select this if you intend to use this template with EC2 Auto Scaling  
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

**Template tags**  
**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.

**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 AMI if you don't see what you are looking for below.

**AMIs**

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

Recent | My AMIs | Quick Start

Owned by me |  Shared with me

Resource types: AMIs  
Networking, security, AWS Lambda, Marketplace and the Community

**Amazon Machine Image (AMI)**

ABC\_Auto\_Scaling\_Webhost  
ami-0322045994957bd  
2023-06-07T19:54:00Z (0) | Virtualization type: HVM | Root device type: ebs

**Description**  
ABC - Auto Scaling Webhost

**Architecture** x86\_64 | **AMI ID** ami-0322045994957bd

**Instance type** Info

**t2.micro** Free tier eligible

Region: US West (Oregon) | 1 vCPU | 1.00 Memory  
On-Demand: 0.016 USD per Hour  
On-Demand: Windows pricing: 0.0162 USD per Hour

**Advanced**

**Key pair (login)** Info  
You can use a key pair to securely connect to your instance. Please that you have access to the selected key pair before you launch the instance.

**Key pair name** ABC-Keypair |

**Network settings** Info

**Subnet info**  
Don't include in launch template |

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 reach your instance.

Select existing security group |  Create security group

**Security group info**  
Select security groups |

ABC - Auto Scaling SG sg-042329f5bc6a20756 X  
VPC ip-042329f5bc6a20756

**Advanced network configuration**  
No network interfaces are currently included in this template. Add a network interface to include it in the launch template.

**Storage (volumes)** Info

**EBS Volumes** Hide details

Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp2))  
AMI volumes are not included in the template unless modified.

Free tier eligible customers can get up to 70 GiB of EBS General Purpose (SSD) or Magnetic storage.

**Resource tags** Info

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

10 remaining (Up to 50 tags maximum)

**Advanced details** Info

Purchasing option Info  
 Request Spot instances  
If Spot is selected you will not be able to create an Auto Scaling group that spans across multiple pricing options and instance types.

IAM Instance profile Info

Hostname type Info

DNS Hostname Info  
 Enable resource-based IPv4 (A record) DNS requests  
 Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery Info

Shutdown behavior Info  
 Not applicable for EC2 Auto Scaling

Stop - Hibernation behavior Info  
 Not applicable for Amazon EC2 Auto Scaling

Termination protection Info

Stop protection Info

Detailed CloudWatch monitoring Info  
 Enable Additional charges apply

**Summary**

Software image (AMI)  
ABC - Auto Scaling Webhost ami-03203f8571fd

Virtual server type (instance type)  
t2.micro

Network (security group)  
ABC - Auto Scaling SG

Storage (volumes)  
1 volume(s) - 8 GiB

6. When you are sure the configurations are correct, click **Create launch template**, then **View launch templates**. You are now finished creating your Launch Template.

You are now ready to move onto the next step

## Create Auto Scaling Group

You have created a Launch Template, which defines the parameters of the instances launched. Now we will create an Auto Scaling Group so that you can define how many EC2 instances should be launched and where to launch them.

1. Make sure you are on the EC2 Service page.
2. In the left navigation pane, find "Auto Scaling" and Select **Auto Scaling Groups**.
3. Click **Create an Auto Scaling group**.
4. Give the Auto Scaling group a name: [Your Initials]-Lab-AutoScaling-Group
5. From the Launch Template drop down choose the launch template named [Your Initials]-scaling-template you created in the previous section and select **Next**.

### Choose launch template or configuration

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

<b>Name</b>	
Auto Scaling group name Enter a name to identify the group. <b>ABC-Lab-AutoScaling-Group</b>	
Must be unique to this account in the current Region and no more than 255 characters.	
<b>Launch template</b> <span style="float: right;">Switch to launch configuration</span>	
Launch template Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups. <b>ABC-scaling-template</b> <span style="float: right;"> </span>	
Create a launch template	
Version <b>Default (1)</b>	
Create a launch template version	
Description <b>Auto Scaling Lab Launch Template</b>	Launch template <b>ABC-scaling-template</b> lt-058f56f2f99a36d2e
AMI ID <b>ami-0ac926fdf993ce578</b>	Security groups -
Key pair name <b>ABC-KeyPair</b>	
Instance type <b>t2.micro</b>	
Additional details	
Storage (volumes) -	Date created Wed Jun 24 2020 12:52:11 GMT-0700 (Pacific Daylight Time)
<b>Cancel</b>	

6. Configure settings page, configure the following and select **Next**:

#### Network:

- VPC: Select your VPC (most likely Default)
- Subnets: Select the subnets where you would like the auto scaling group to use when spinning up the hosts. (If you are using the default VPC, this will most likely be four subnets, as shown below)

A best practice for your Auto Scaling Group would be to select only private subnets. The instances will be sitting behind a load balancer and will not need public IP addresses. For the sake of this lab, they could be either private or public subnets.

### Choose instance launch options

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

<b>Network</b>								
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.								
<b>VPC</b> Choose the VPC that defines the virtual network for your Auto Scaling group.								
<b>vpce-84252dfc</b> <span style="float: right;"> </span>								
Create a VPC								
<b>Availability Zones and subnets</b> Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.								
Select Availability Zones and subnets <span style="float: right;"> </span>								
<table border="1"> <tr> <td>us-west-2a   subnet-079bbedc X</td> </tr> <tr> <td>172.31.32.0/20 Default</td> </tr> <tr> <td>us-west-2b   subnet-28781d50 X</td> </tr> <tr> <td>172.31.16.0/20 Default</td> </tr> <tr> <td>us-west-2c   subnet-25e59178 X</td> </tr> <tr> <td>172.31.0.0/20 Default</td> </tr> <tr> <td>us-west-2d   subnet-516fa76 X</td> </tr> <tr> <td>172.31.48.0/20 Default</td> </tr> </table>	us-west-2a   subnet-079bbedc X	172.31.32.0/20 Default	us-west-2b   subnet-28781d50 X	172.31.16.0/20 Default	us-west-2c   subnet-25e59178 X	172.31.0.0/20 Default	us-west-2d   subnet-516fa76 X	172.31.48.0/20 Default
us-west-2a   subnet-079bbedc X								
172.31.32.0/20 Default								
us-west-2b   subnet-28781d50 X								
172.31.16.0/20 Default								
us-west-2c   subnet-25e59178 X								
172.31.0.0/20 Default								
us-west-2d   subnet-516fa76 X								
172.31.48.0/20 Default								
Create a subnet								
<b>Instance type requirements</b> <span style="float: right;">Override launch template </span>								
You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.								
Launch template <b>ABC-scaling-template</b> <span style="float: right;"> </span>	Version <b>Default</b>							
Description <b>V1</b>								
Instance type <b>t2.micro</b>								
<b>Cancel</b>								

**7. Specify load balancing and health checks:**

- Load balancing: Attach to a new load balancer
- Load balancer type: Application Load Balancer
- Load balancer name: [Your Initials]-Application-Load-Balancer
- Load balancer scheme: Internet-facing
- Networking mapping: You should see all the Availability Zones and subnets you selected in the previous step. (If you had multiple subnets per AZ, this were it would let you choose between them)
- Listeners and routing: Keep the Port as 80 and select Create a target group from the "Default routing (forward to)" dropdown.
- New target group name: [Your Initials]-Target-Group
- The target group is where your load balancer is going to look for instances to distribute traffic. We are setting our auto scaling group to automatically register instances into this group and it will also be associated to our load balancer.
- Health checks & Additional settings: Leave as default and select Next.

### Configure advanced options Info

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

#### Load balancing - optional Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

#### Attach to a new load balancer

Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type  
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the [Load Balancing console](#).

Application Load Balancer  
HTTP, HTTPS

Network Load Balancer  
TCP, UDP, TLS

Load balancer name  
Name cannot be changed after the load balancer is created.  
**ABC-Application-Load-Balances**

Load balancer scheme  
Scheme cannot be changed after the load balancer is created.

Internal

Internet-facing

Network mapping  
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC  
vpc-B4252dfe

Availability zones and subnets  
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

<input checked="" type="checkbox"/> us-west-2c	subnet-25e5917f
<input checked="" type="checkbox"/> us-west-2b	subnet-28781d50
<input checked="" type="checkbox"/> us-west-2a	subnet-8759becd
<input checked="" type="checkbox"/> us-west-2d	subnet-5c16fa76

Listeners and routing  
If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol <b>HTTP</b>	Port <b>80</b>	Default routing (forward to) <b>Create a target group</b>
New target group name An instance target group with default settings will be created. <b>ABC-Target-Group</b>		

Tags - optional  
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add tag  
50 remaining

#### Health checks - optional

Health check type Info  
EC2 Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

EC2  ELB

Health check grace period  
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.  
**300** seconds

#### Additional settings - optional

Monitoring Info  
 Enable group metrics collection within CloudWatch

Default instance warmup Info  
The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.  
 Enable default instance warmup

**Cancel** **Previous** **Skip to review** **Next**

8. Configure the group size and scaling policies below and then select Next.
- Group Size:** The settings below will keep our group size to one EC2 instance unless a scaling policy is triggered.
    - o Desired capacity: 1
    - o Minimum capacity: 1
    - o Maximum Capacity: 5
  - Scaling policies:** Select Target tracking scaling policy
    - o Metric type: Average CPU utilization
    - o Target Value: 25

We are going to set our target CPU utilization low to speed up the lab.

#### 9. Add Notifications:

You can configure your Auto Scaling Group to send notifications to an endpoint that you choose, such as an email address. You can receive notifications whenever a specified event takes place, including the successful launch of an instance, failed instance launch, instance termination, and failed instance termination.

For now, you are going to skip this step, select Next.

#### 10. Add Tags: Add a single tag and then select Next.

- Select the Add tag button and configure the following:
  - o Key: Name
  - o Value: [Your Initials] - Auto Scaling Group

11. Review your settings and then select Create Auto Scaling group. You have now created your Auto Scaling Group, target group and load balancer.

- You will soon see a new instance created by the Auto Scaling group in the EC2 console with the name tag "[Your Initials] - Auto Scaling Group". (You may need to refresh the screen to see the instance)
- If you select Load Balancers under "Load Balancing" in the left hand menu, you will see your load balancer provisioning.

In the next step we will create an additional security group and update the security settings to allow traffic to flow between the ALB and our web hosts.



You are now ready to move onto the next step

## **Configuring Security Groups**

**Creating a Load Balancer Security Group**

When our load balancer was provisioned it was setup with the default security group in our VPC. To allow access to the load balancer via the public DNS, we will need to create and attach a security group to allow inbound traffic on port 80 from the internet.

We will also create an outbound rule that allows outgoing traffic from the load balancer to only be sent to hosts within the Auto Scaling Security Group.

Within the console Under "Services" search and select EC2. On the EC2 page under the "Network & Security" heading in the left-hand menu select **Security Groups**. You should see other security groups, including the security group for your web server named [Your Initials]-EC2-Web-Host - Website Security Group. Click on the **Create security group** button.

- Basic details:**
  - a. **Security group name:** [Your Initials]-SG-Load-Balancer
  - b. **Description:** [Your Initials]-SG-Load-Balancer
  - c. **VPC:** Select your VPC (Most likely the Default VPC)
- Inbound rules:**
  - a. Click on the **Add rule** button
  - b. **Type:** HTTP
  - c. **Source:** Custom: [Input your public IP address followed by a /32] (You can find your local IP by searching [What is my IP](#))
- Outbound rules:**
  - a. Find the "All traffic" rule and click on **Delete** to remove the rule. (All Outbound rules should now be removed)
  - b. Click on the **Add rule** button
  - c. **Type:** HTTP
  - d. Under "Destination" select **Custom** and in the field select your [Your Initials]-Auto Scaling SG as the "Destination". Hint: start by typing sg to get the Security Group list.
  - e. Your security group configuration should look similar to the image below. Select **Create security group** when finished.

**Create security group**

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

**Security group name:** ABC-SG-Load-Balancer  
Name cannot be deleted after creation.

**Description:** ABC-SG-Load-Balancer

**VPC:** vpc-0x4318aef1f949e3 (Default VPC)

**Inbound rules**

Type	Protocol	Port range	Source	Description (optional)
HTTP	TCP	80	Custom	26.155.55.245/32

**Add rule**

**Outbound rules**

Type	Protocol	Port range	Destination	Description (optional)
HTTP	TCP	80	Custom	ip-30aef7f12321ba9

**Add rule**

**Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to identify your resources or track your AWS costs.

No tags associated with the resource.

**Add new tag**

You can add up to 10 more tags.

**Create security group**

**4. Attach your new Load Balancer Security group to your Load Balancer:**

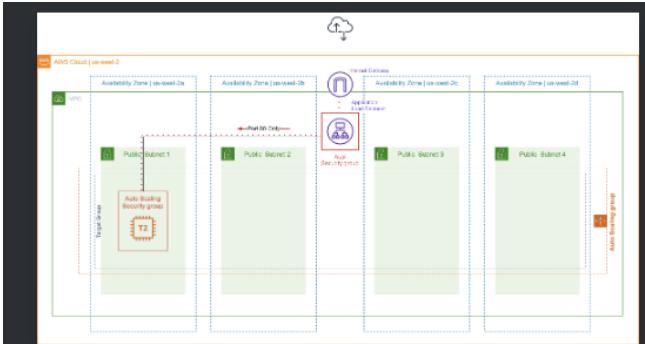
- a. On the EC2 service page left side menu find "Load Balancing" and select **Load Balancers**. Select the load balancer you created. Make sure the State is "Active".
- b. Under the "Description" tab scroll down to the "Security" section and click on **Edit security groups**.
- c. Select the box to the left of your new load balancer sg named [Your Initials]-SG-Load-Balancer.
- d. Make sure you also un-select any other security group and then click on the **Save** button.

**Edit security groups**

Select security groups to associate with your load balancer.

Security group ID	Name	Description
sg-0054479c9d0073	ABC-AutoScalingSG	ABC - Auto Scaling SG
sg-035474167593	ABC-EC2-Web-Host - Website Security Group	All Access to the WebHost on Port 80 & 22
sg-0491fb1f116f	ABC-LoadBalancer	ABC-LoadBalancer
sg-a8969f1	default	default VPC security group

**Cancel** **Save**

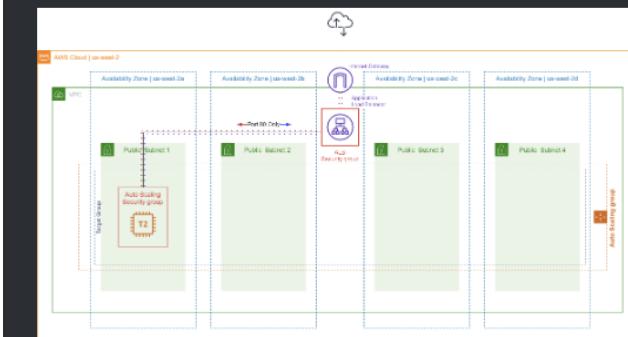


#### Task 4: Add Inbound Rule to the Auto Scaling Security Group

1. We will need to setup a rule to only allow traffic from the new Load Balancer Security Group to the Auto Scaling Security Group. This will be one of the layers of protection that will prevent our webhosts from being directly accessed from the internet.
- a. On the EC2 service page left side menu under "Network & Security" select **Security Groups**.
- b. Select your Auto Scaling Security Group: [Your Initials] - Auto Scaling SG
- c. Select the **Inbound Rules** tab and click on the **Edit inbound rules** button and then the **Add rule** button.
- d. From the "Type" drop down select **HTTP**. Under "Source" select **Custom** and in the field specify your [Your Initials]-SG-Lead-Balancer as the "Source". Hint: start by typing sg to get the Security Group list. Now click on **Save rules**.

Your rule should now look similar to the image below.

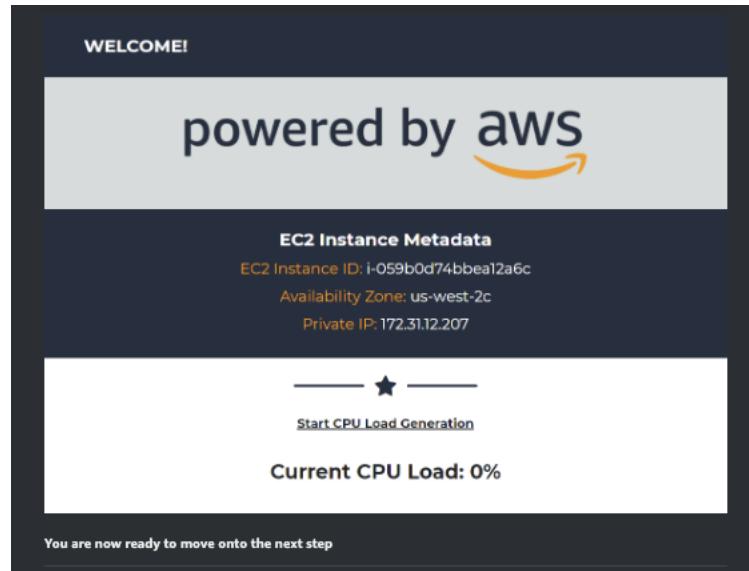
Name	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	sg-0bc0cbcb3204be66 (ABC-SG-Load-Balancer)	



2. We will now test to make sure your load balancer is working. There is currently only one instance (or target) running in the auto scaling group, but you should be able to access the website.

Return to your load balancers page by selecting **Load Balancers** from the left hand menu. Under the "Description" tab copy the DNS name and paste it into a web browser. You should now see the website being loaded from your auto scaling group. Leave this page open, you will need it in the next step.

Name	DNS name	State	VPC ID	Availability Zones	Type
ABC-Application Load Balancer	ABC-Application-Load-Balancer.us-west-2.ellipsis.amazonaws.com	Active	vpc-ace50495	us-west-2a, us-west-2b, us-west-2d, us-west-2e	application

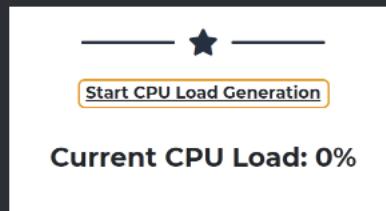


### Testing the Auto Scaling Group

## Testing the Auto Scaling Group

Now that you have created your Auto Scaling Group and load balancer, let's test it to ensure that everything is working correctly.

1. Make sure you are on the website accessed through the Load Balancer DNS address in the previous step.
2. At the bottom of the front page click on the **Start CPU Load Generation** link: Once the CPU load goes above 25% for a sustained period the Auto Scaling policy will begin spinning up the instances specified in the launch template to meet demand. (You may have to do this twice if the first time doesn't generate enough load)



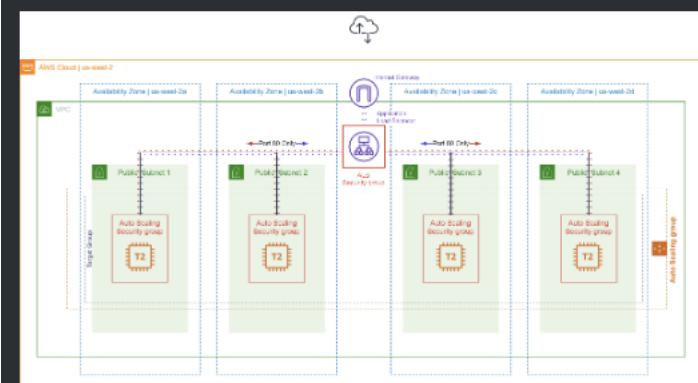
3. In the "Instances" section of the EC2 Console you can watch for the new instances created by Auto Scaling, this might take a couple of minutes. Refresh the EC2 instances page and you should soon see a new instance spinning up automatically. You can select the instance named [Your Initials]-Auto Scaling Group and click on the Monitoring tab below to keep an eye on the "CPU Utilization".

Instances (4) Info							
Filter instances		Instance static pending		Clear filters			
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	ABC - Auto Scaling Group	i-0f75e9edc6077cc0	Pending	t2.micro	-	No alarms +	us-west-2A
<input type="checkbox"/>	ABC - Auto Scaling Group	i-02f8303132d28159	Pending	t2.micro	-	No alarms +	us-west-2B
<input type="checkbox"/>	ABC-EC2 Web Host	i-0860600408293461	Running	m4.large	2/2 checks passed.	No alarms +	us-west-2C
<input type="checkbox"/>	ABC - Auto Scaling Group	i-09a0a39f189722026	Running	t2.micro	2/2 checks passed.	No alarms +	us-west-2D

4. You can also see this by going to the Auto Scaling Groups page. <https://console.aws.amazon.com/ec2autoscaling/> Then select your auto scaling group [Your Initials]-Lab-AutoScaling-Group. If you look at the details under the Instance management tab, you can see if new instances are spinning up. You can look at the instance management 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.

EC2 > Auto Scaling groups > ABC-Lab-AutoScaling-Group																																																							
Details		Activity		Automatic scaling		Instance management																																																	
Instances (4)																																																							
<table border="1"> <thead> <tr> <th colspan="7">Instances (4)</th> </tr> <tr> <th colspan="2">Filter instances</th> <th colspan="2">Actions</th> <th colspan="3"></th> </tr> <tr> <th></th> <th>Instance ID</th> <th>Lifecycle</th> <th>Instance type</th> <th>Launch template/configuration</th> <th>Availability Zone</th> <th>Health status</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>i-0781a01ba47ed709e</td> <td>InService</td> <td>t2.micro</td> <td>ABC-scaling-template   Version 1</td> <td>us-east-1b</td> <td>Healthy</td> </tr> <tr> <td><input type="checkbox"/></td> <td>i-0b1a71f4c3dbe25a</td> <td>InService</td> <td>t2.micro</td> <td>ABC-scaling-template   Version 1</td> <td>us-east-1a</td> <td>Healthy</td> </tr> <tr> <td><input type="checkbox"/></td> <td>i-0f1ef9f65ddfb705</td> <td>InService</td> <td>t2.micro</td> <td>ABC-scaling-template   Version 1</td> <td>us-east-1c</td> <td>Healthy</td> </tr> <tr> <td><input type="checkbox"/></td> <td>i-074cb477f2769ce</td> <td>InService</td> <td>t2.micro</td> <td>ABC-scaling-template   Version 1</td> <td>us-east-1f</td> <td>Healthy</td> </tr> </tbody> </table>							Instances (4)							Filter instances		Actions						Instance ID	Lifecycle	Instance type	Launch template/configuration	Availability Zone	Health status	<input type="checkbox"/>	i-0781a01ba47ed709e	InService	t2.micro	ABC-scaling-template   Version 1	us-east-1b	Healthy	<input type="checkbox"/>	i-0b1a71f4c3dbe25a	InService	t2.micro	ABC-scaling-template   Version 1	us-east-1a	Healthy	<input type="checkbox"/>	i-0f1ef9f65ddfb705	InService	t2.micro	ABC-scaling-template   Version 1	us-east-1c	Healthy	<input type="checkbox"/>	i-074cb477f2769ce	InService	t2.micro	ABC-scaling-template   Version 1	us-east-1f	Healthy
Instances (4)																																																							
Filter instances		Actions																																																					
	Instance ID	Lifecycle	Instance type	Launch template/configuration	Availability Zone	Health status																																																	
<input type="checkbox"/>	i-0781a01ba47ed709e	InService	t2.micro	ABC-scaling-template   Version 1	us-east-1b	Healthy																																																	
<input type="checkbox"/>	i-0b1a71f4c3dbe25a	InService	t2.micro	ABC-scaling-template   Version 1	us-east-1a	Healthy																																																	
<input type="checkbox"/>	i-0f1ef9f65ddfb705	InService	t2.micro	ABC-scaling-template   Version 1	us-east-1c	Healthy																																																	
<input type="checkbox"/>	i-074cb477f2769ce	InService	t2.micro	ABC-scaling-template   Version 1	us-east-1f	Healthy																																																	

Your architecture now looks like this:



5. Once a number of new instances have successfully started (probably 3 or 4), repeatedly refresh your web-browser on you web host. You should now see the Instance ID, Availability Zone and Private IP change as the load balancer distributes the requests across the Auto Scaling group.

EC2 Instance Metadata	EC2 Instance Metadata	EC2 Instance Metadata
EC2 Instance ID: i-0cd1d99666510 Availability Zone: us-west-2a Private IP: 172.31.37.120	EC2 Instance ID: i-0f68074a4e86e4e8 Availability Zone: us-west-2b Private IP: 172.31.21.204	EC2 Instance ID: i-0df1b562f76d43fd Availability Zone: us-west-2c Private IP: 172.31.33.92

Congratulations! You have successfully created an EC2 Auto Scaling Group behind an Application Load Balancer.

Lab Author: Cy Hopkins

If you need to cleanup this lab, move on to the next section

## **Auto Scaling Lab Teardown**

**Deleting your Load Balancer**

- In the console navigate to the "EC2" service. From the left menu under "Load Balancing" select **Load Balancers**.
- Select the load balancer named [Your Initials]-Application-Load-Balancer.
- Click on the **Actions** drop down at the top of the page and select **Delete**. In the following popup window select the **Yes, Delete** button.

4. Your load balancer should be gone immediately if your delete was successful.

**Deleting your Target Group**

- In the console navigate to the "EC2" service. From the left menu under "Load Balancing" select **Target Groups**.
- Select the target group named [Your Initials]-Target-Group
- Click on the **Actions** drop down at the top of the page and select **Delete**. In the following popup window select the **Yes, Delete** button.

4. Your target group should now be deleted.

**Deleting your Auto Scaling Group**

- In the console navigate to the "EC2" service. From the left menu under "Auto Scaling" select **Auto Scaling Groups**.
- Select the auto scaling group named [Your Initials]-Lab-AutoScaling-Group
- Click on the **Delete** button at the top of the page. In the following popup window type **delete** in the text field and select the **Delete** button.

4. All of the instances in your auto scaling group will now be terminated, it may take a few minutes to complete.

**Deleting your Launch Template**

- In the console navigate to the "EC2" service. From the left menu under "Instances" select **Launch Templates**.
- Select the launch template named [Your Initials]-scaling-template.
- Click on the **Actions** drop down at the top of the page and select **Delete template**. In the following popup window type **delete** in the text field and select the **Delete** button.

4. Your launch template should now be deleted.

**Deleting your Security Groups**

- In the console navigate to the "EC2" service. From the left menu under "Network & Security" select Security Groups.
- Select the security group named [Your Initials]-SG-Load-Balancer and click on the Edit inbound rules button.

3. On the "Edit inbound rules" page, click on the Delete button to delete the one rule and then select the Save rules button.

4. Now select the Outbound rules tab and select the Edit outbound rules button.

5. On the "Edit outbound rules" page, click on the Delete button to delete the one rule and then select the Save rules button.

6. Now only select the security group name [Your Initials] - Auto Scaling SG and repeat steps 2 & 3. (No need to remove the outbound rule on this security group)

7. Now select the two security groups named [Your Initials]-SG-Load-Balancer and [Your Initials] - Auto Scaling SG.

8. Click on the Actions drop down at the top of the page and select Delete security groups. (You may have to scroll down in the Actions menu) In the following popup window type delete  in the text field and select the Delete button.

9. Your security groups should now be deleted. If you were not able to remove the security groups you may not have removed all the rules successfully.

**Deleting your CloudFormation Stack**

- In the console open CloudFormation under services or with search.
- Select the stack named [Your Initials]-EC2-Web-Host and then the Delete button.

3. In the popup select Delete stack.

4. The stack will take a few minutes to delete, select the refresh button to see the updated status. The stack will no longer be visible when it is deleted.

**Congratulations! You have completed the Auto Scaling Lab!**