



Academy Cloud Foundations (ACF)  
Lab Guide

Version 1.0.18

100-ACFNDS-10-EN-LG

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved.

This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.

Corrections or feedback on the course, please email us at:

[aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com).

For all other questions, contact us at:

<https://aws.amazon.com/contact-us/aws-training/>.

All trademarks are the property of their owners.

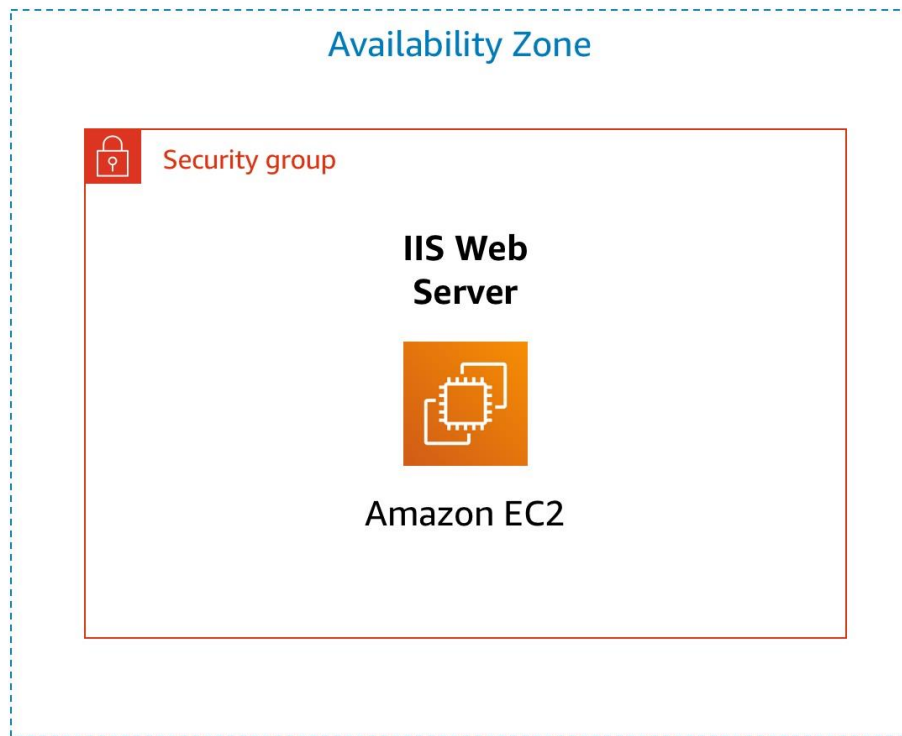
# Contents

ACF Lab 1: Introduction to Amazon EC2	4
ACF Lab 2: Working with EBS	20
ACF Lab 3: Build your VPC and Launch a Web Server	33
ACF Lab 4: Build Your DB Server and Interact With Your DB Using an App	45
ACF Lab 5: Scale and Load Balance Your Architecture	56
ACF Lab 6: Introduction to AWS IAM	70
ACF Lab 7: AWS Playground	81

# ACF Lab 1: Introduction to Amazon EC2

Version 1.1.7 (spl200)

## Overview



This lab provides you with a basic overview of launching, resizing, managing, and monitoring an Amazon EC2 instance.

**Amazon Elastic Compute Cloud (Amazon EC2)** is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

## Topics covered

---

By the end of this lab, you will be able to:

- Launch a web server with termination protection enabled
- Monitor Your EC2 instance
- Modify the security group that your web server is using to allow HTTP access
- Resize your Amazon EC2 instance to scale
- Explore EC2 limits
- Test termination protection
- Terminate your EC2 instance

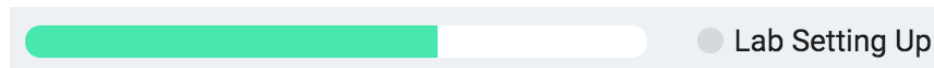
### Duration

This lab takes approximately **45 minutes** to complete.

## Accessing the AWS Management Console

1. At the top of these instructions, click **Start Lab** to launch your lab.
2. When asked *How do you want to start this lab*, click **Launch with credits**.

A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.



3. Click **Open Console**
4. Sign in using the **Username** and **Password** shown to the left of these instructions.

You will be taken to the AWS Management Console.

## Task 1: Launch Your Amazon EC2 Instance

In this task, you will launch an Amazon EC2 instance with *termination protection*. Termination protection prevents you from accidentally terminating an EC2 instance. You will deploy your instance with a User Data script that will allow you to deploy a simple web server.

5. In the **AWS Management Console** on the **Services** menu, click **EC2**.
6. Click **Launch Instance**

### Step 1: Choose an Amazon Machine Image (AMI)

① An **Amazon Machine Image (AMI)** provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes:

- A template for the root volume for the instance (for example, an operating system or an application server with applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it is launched

The **Quick Start** list contains the most commonly-used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.

7. Click **Select** next to **Amazon Linux 2 AMI** (at the top of the list).

## Step 2: Choose an Instance Type

---

① Amazon EC2 provides a wide selection of *instance types* optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more *instance sizes*, allowing you to scale your resources to the requirements of your target workload.

You will use a **t2.micro** instance which should be selected by default. This instance type has 1 virtual CPU and 1 GiB of memory. **NOTE:** You may be restricted from using other instance types in this lab.

8. Click **Next: Configure Instance Details**

## Step 3: Configure Instance Details

---

This page is used to configure the instance to suit your requirements. This includes networking and monitoring settings.

The **Network** indicates which Virtual Private Cloud (VPC) you wish to launch the instance into. You can have multiple networks, such as different ones for development, testing and production.

9. For **Network**, select **Lab VPC**.

The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.

10. For **Enable termination protection**, select ☒ **Protect against accidental termination**.

① When an Amazon EC2 instance is no longer required, it can be *terminated*, which means that the instance is stopped and its resources are released. A terminated instance cannot be started again. If you want to prevent the instance from being accidentally terminated, you can enable *termination protection* for the instance, which prevents it from being terminated.

11. Scroll down, then expand ► **Advanced Details**.

A field for **User data** will appear.

① When you launch an instance, you can pass *user data* to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.

Your instance is running Amazon Linux, so you will provide a *shell script* that will run when the instance starts.

12. Copy the following commands and paste them into the **User data** field:

```
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
```

The script will:

- Install an Apache web server (httpd)
- Configure the web server to automatically start on boot
- Activate the Web server
- Create a simple web page

13. Click **Next: Add Storage**

## Step 4: Add Storage

---

① Amazon EC2 stores data on a network-attached virtual disk called *Elastic Block Store*.

You will launch the Amazon EC2 instance using a default 8 GiB disk volume. This will be your root volume (also known as a 'boot' volume).

14. Click **Next: Add Tags**

## Step 5: Add Tags

---

① Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you have assigned to it. Each tag consists of a Key and a Value, both of which you define.

15. Click **Add Tag** then configure:

- **Key:** Name
- **Value:** Web Server



16. Click **Next: Configure Security Group**

## Step 6: Configure Security Group


---

① A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add *rules* to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.

17. On **Step 6: Configure Security Group**, configure:

- **Security group name:** `Web Server security group`
- **Description:** `Security group for my web server`

In this lab, you will not log into your instance using SSH. Removing SSH access will improve the security of the instance.

18. Delete  the existing SSH rule.

19. Click **Review and Launch**

## Step 7: Review Instance Launch

---

The Review page displays the configuration for the instance you are about to launch.

20. Click **Launch**

A **Select an existing key pair or create a new key pair** window will appear.

① Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

In this lab you will not log into your instance, so you do not require a key pair.

21. Click the **Choose an existing key pair** drop-down and select *Proceed without a key pair*.

22. Select ☒ **I acknowledge that ....**

23. Click **Launch Instances**

Your instance will now be launched.

24. Click **View Instances**

The instance will appear in a *pending* state, which means it is being launched. It will then change to *running*, which indicates that the instance has started booting. There will be a short time before you can access the instance.

The instance receives a *public DNS name* that you can use to contact the instance from the Internet.

Your **Web Server** should be selected. The **Description** tab displays detailed information about your instance.

💬 To view more information in the Description tab, drag the window divider upwards.

Review the information displayed in the **Description** tab. It includes information about the instance type, security settings and network settings.

25. Wait for your instance to display the following:

- **Instance State:** ● running
- **Status Checks:** 2/2 checks passed
- 👏 **Congratulations!** You have successfully launched your first Amazon EC2 instance.

## Task 2: Monitor Your Instance

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions.

26. Click the **Status Checks** tab.

① With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.

Notice that both the **System reachability** and **Instance reachability** checks have passed.

27. Click the **Monitoring** tab.

This tab displays Amazon CloudWatch metrics for your instance. Currently, there are not many metrics to display because the instance was recently launched.

You can click on a graph to see an expanded view.

① Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (five-minute) monitoring is enabled by default. You can enable detailed (one-minute) monitoring.

28. In the **Actions** menu, select **Instance Settings ► Get System Log**.

The System Log displays the console output of the instance, which is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started. If you do not see a system log, wait a few minutes and then try again.

29. Scroll through the output and note that the HTTP package was installed from the **user data** that you added when you created the instance.

```

System Log: i-0bbd359abd033965f (Web Server)

Dependencies Resolved

Package Arch Version Repository Size
Installing: httpd x86_64 2.2.34-1.16.amzn1 amzn-updates 1.2 M
Installing for dependencies:
apr x86_64 1.5.2-5.13.amzn1 amzn-updates 118 k
apr-util x86_64 1.5.4-6.18.amzn1 amzn-updates 99 k
apr-util-ldap x86_64 1.5.4-6.18.amzn1 amzn-updates 19 k
httpd-tools x86_64 2.2.34-1.16.amzn1 amzn-updates 80 k

Transaction Summary
Install 1 Package (+4 Dependent packages)

Total download size: 1.5 M
Installed size: 3.6 M
Downloading packages:
-----
Total 3.8 MB/s | 1.5 MB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : apr-1.5.2-5.13.amzn1.x86_64 1/5
Installing : apr-util-1.5.4-6.18.amzn1.x86_64 2/5
Installing : httpd-tools-2.2.34-1.16.amzn1.x86_64 3/5
Installing : apr-util-ldap-1.5.4-6.18.amzn1.x86_64 4/5
Installing : httpd-2.2.34-1.16.amzn1.x86_64 5/5
Verifying : httpd-tools-2.2.34-1.16.amzn1.x86_64 1/5
  
```

30. Click **Close**

31. In the **Actions** menu, select **Instance Settings ► Get Instance Screenshot**.

This shows you what your Amazon EC2 instance console would look like if a screen were attached to it.

```

Get instance screenshot

Below is a screenshot of i-0bbd359abd033965f (Web Server) at 2018-02-09T14:56:07.784+11:00.

Refresh

Amazon Linux AMI release 2017.09
Kernel 4.9.76-3.78.amzn1.x86_64 on an x86_64

ip-10-0-1-221 login: _
  
```

① If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. This provides visibility as to the status of the instance, and allows for quicker troubleshooting.

32. Click **Close**

👉 **Congratulations!** You have explored several ways to monitor your instance.

## Task 3: Update Your Security Group and Access the Web Server

When you launched the EC2 instance, you provided a script that installed a web server and created a simple web page. In this task, you will access content from the web server.

33. Click the **Description** tab.

34. Copy the **IPv4 Public IP** of your instance to your clipboard.

35. Open a new tab in your web browser, paste the IP address you just copied, then press **Enter**.

**Question:** Are you able to access your web server? Why not?

You are **not** currently able to access your web server because the *security group* is not permitting inbound traffic on port 80, which is used for HTTP web requests. This is a demonstration of using a security group as a firewall to restrict the network traffic that is allowed in and out of an instance.

To correct this, you will now update the security group to permit web traffic on port 80.

36. Keep the browser tab open, but return to the **EC2 Management Console** tab.

37. In the left navigation pane, click **Security Groups**.

38. Select **Web Server security group**.

39. Click the **Inbound** tab.

The security group currently has no rules.

40. Click **Edit** then configure:

- **Type:** *HTTP*
- **Source:** *Anywhere*
- Click **Save**

41. Return to the web server tab that you previously opened and refresh the page.

You should see the message *Hello From Your Web Server!*

👉 **Congratulations!** You have successfully modified your security group to permit HTTP traffic into your Amazon EC2 Instance.

## Task 4: Resize Your Instance: Instance Type and EBS Volume

As your needs change, you might find that your instance is over-utilized (too small) or under-utilized (too large). If so, you can change the *instance type*. For example, if a *t2.micro* instance is too small for its workload, you can change it to an *m5.medium* instance. Similarly, you can change the size of a disk.

### Stop Your Instance

---

Before you can resize an instance, you must *stop* it.

① When you stop an instance, it is shut down. There is no charge for a stopped EC2 instance, but the storage charge for attached Amazon EBS volumes remains.

42. On the **EC2 Management Console**, in the left navigation pane, click **Instances**.

**Web Server** should already be selected.

43. In the **Actions** menu, select **Instance State ► Stop**.

44. Click **Yes, Stop**

Your instance will perform a normal shutdown and then will stop running.

45. Wait for the **Instance State** to display: ● stopped

### Change The Instance Type

---

46. In the **Actions** menu, select **Instance Settings ► Change Instance Type**, then configure:

- **Instance Type:** *t2.small*
- Click **Apply**

When the instance is started again it will be a *t2.small*, which has twice as much memory as a *t2.micro* instance. **NOTE:** You may be restricted from using other instance types in this lab.

### Resize the EBS Volume

---

47. In the left navigation menu, click **Volumes**.

48. In the **Actions** menu, select **Modify Volume**.

The disk volume currently has a size of 8 GiB. You will now increase the size of this disk.

49. Change the size to: 10 **NOTE:** You may be restricted from creating large Amazon EBS volumes in this lab.

50. Click **Modify**

51. Click **Yes** to confirm and increase the size of the volume.

52. Click **Close**

## Start the Resized Instance

---

You will now start the instance again, which will now have more memory and more disk space.

53. In left navigation pane, click **Instances**.

54. In the **Actions** menu, select **Instance State ► Start**.

55. Click **Yes, Start**

👏 **Congratulations!** You have successfully resized your Amazon EC2 Instance. In this task you changed your instance type from *t2.micro* to *t2.small*. You also modified your root disk volume from 8 GiB to 10 GiB.



## Task 5: Explore EC2 Limits

Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create an AWS account, there are default limits on these resources on a per-region basis.

56. In the left navigation pane, click **Limits**.

Note that there is a limit on the number of instances that you can launch in this region. When launching an instance, the request must not cause your usage to exceed the current instance limit in that region.

You can request an increase for many of these limits.

## Task 6: Test Termination Protection

You can delete your instance when you no longer need it. This is referred to as *terminating* your instance. You cannot connect to or restart an instance after it has been terminated.

In this task, you will learn how to use *termination protection*.

57. In left navigation pane, click **Instances**.

58. In the **Actions** menu, select **Instance State ► Terminate**.

Note that there is a message that says: *These instances have Termination Protection and will not be terminated. Use the Change Termination Protection option from the Instances screen Actions menu to allow termination of these instances.*

Also, the **Yes, Terminate** button is dimmed and cannot be clicked.

This is a safeguard to prevent the accidental termination of an instance. If you really want to terminate the instance, you will need to disable the termination protection.

59. Click **Cancel**.

60. In the **Actions** menu, select **Instance Settings ► Change Termination Protection**.

61. Click **Yes, Disable**

You can now terminate the instance.

62. In the **Actions** menu, select **Instance State ► Terminate**.

63. Click **Yes, Terminate**

👏 **Congratulations!** You have successfully tested termination protection and terminated your instance.

## Lab Complete

🏁 Congratulations! You have completed the lab.

Click **End Lab** at the top of this page to clean up your lab environment.

For feedback, suggestions, or corrections, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com)

## Additional Resources

- [Launch Your Instance](#)
- [Amazon EC2 Instance Types](#)
- [Amazon Machine Images \(AMI\)](#)
- [Amazon EC2 - User Data and Shell Scripts](#)
- [Amazon EC2 Root Device Volume](#)
- [Tagging Your Amazon EC2 Resources](#)
- [Security Groups](#)
- [Amazon EC2 Key Pairs](#)
- [Status Checks for Your Instances](#)
- [Getting Console Output and Rebooting Instances](#)
- [Amazon EC2 Metrics and Dimensions](#)
- [Resizing Your Instance](#)
- [Stop and Start Your Instance](#)
- [Amazon EC2 Service Limits](#)
- [Terminate Your Instance](#)
- [Termination Protection for an Instance](#)

# ACF Lab 2: Working with EBS

Version 5.1.3 (spl2)

## Lab Overview



This lab focuses on Amazon Elastic Block Store (Amazon EBS), a key underlying storage mechanism for Amazon EC2 instances. In this lab, you will learn how to create an Amazon EBS volume, attach it to an instance, apply a file system to the volume, and then take a snapshot backup.

## Topics covered

By the end of this lab, you will be able to:

- Create an Amazon EBS volume
- Attach and mount your volume to an EC2 instance
- Create a snapshot of your volume
- Create a new volume from your snapshot
- Attach and mount the new volume to your EC2 instance

## Lab Pre-requisites

To successfully complete this lab, you should be familiar with basic Amazon EC2 usage and with basic Linux server administration. You should feel comfortable using the Linux command-line tools.

## Other AWS Services

---

Other AWS Services than the ones needed for this lab are disabled by IAM policy during your access time in this lab. In addition, the capabilities of the services used in this lab are limited to what's required by the lab and in some cases are even further limited as an intentional aspect of the lab design. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

## What is Amazon Elastic Block Store?

---

**Amazon Elastic Block Store (Amazon EBS)** offers persistent storage for Amazon EC2 instances. Amazon EBS volumes are network-attached and persist independently from the life of an instance. Amazon EBS volumes are highly available, highly reliable volumes that can be leveraged as an Amazon EC2 instances boot partition or attached to a running Amazon EC2 instance as a standard block device.

When used as a boot partition, Amazon EC2 instances can be stopped and subsequently restarted, enabling you to pay only for the storage resources used while maintaining your instance's state. Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores because Amazon EBS volumes are automatically replicated on the backend (in a single Availability Zone).

For those wanting even more durability, Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon Simple Storage Service (Amazon S3) and automatically replicated across multiple Availability Zones. These snapshots can be used as the starting point for new Amazon EBS volumes and can protect your data for long-term durability. You can also easily share these snapshots with co-workers and other AWS developers.

This lab guide explains basic concepts of Amazon EBS in a step-by-step fashion. However, it can only give a brief overview of Amazon EBS concepts. For further information, see the [Amazon EBS documentation](#).

## Amazon EBS Volume Features

---

Amazon EBS volumes deliver the following features:

- **Persistent storage:** Volume lifetime is independent of any particular Amazon EC2 instance.
- **General purpose:** Amazon EBS volumes are raw, unformatted block devices that can be used from any operating system.
- **High performance:** Amazon EBS volumes are equal to or better than local Amazon EC2 drives.
- **High reliability:** Amazon EBS volumes have built-in redundancy within an Availability Zone.
- **Designed for resiliency:** The AFR (Annual Failure Rate) of Amazon EBS is between 0.1% and 1%.
- **Variable size:** Volume sizes range from 1 GB to 16 TB.
- **Easy to use:** Amazon EBS volumes can be easily created, attached, backed up, restored, and deleted.

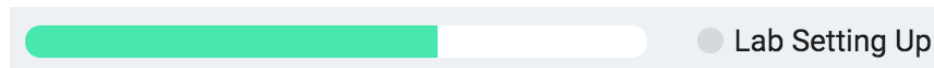
### Duration

This lab takes approximately **45 minutes** to complete.

## Accessing the AWS Management Console

1. At the top of these instructions, click **Start Lab** to launch your lab.
2. When asked *How do you want to start this lab*, click **Launch with credits**.

A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.



3. Click **Open Console**
4. Sign in using the **Username** and **Password** shown to the left of these instructions.

You will be taken to the AWS Management Console.

## Task 1: Create a New EBS Volume

In this task, you will create and attach an Amazon EBS volume to a new Amazon EC2 instance.

5. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
6. In the left navigation pane, click **Instances**.

An Amazon EC2 instance named **Lab** has already been launched for your lab.

7. Note the **Availability Zone** of the instance. It will look similar to *us-west-2a*.
8. In the left navigation pane, click **Volumes**.

You will see an existing volume that is being used by the Amazon EC2 instance. This volume has a size of 8 GiB, which makes it easy to distinguish from the volume you will create next, which will be 1 GiB in size.

9. Click **Create Volume** then configure:

- **Volume Type:** *General Purpose SSD (gp2)*
- **Size (GiB):** 1. **NOTE:** You may be restricted from creating large volumes.
- **Availability Zone:** Select the same availability zone as your EC2 instance. It is also shown to the left of the instructions you are currently reading.
- Click **Add Tag**
- In the Tag Editor, enter:
  - **Key:** Name
  - **Value:** My Volume

10. Click **Create Volume** then click **Close**

Your new volume will appear in the list, and will move from the *creating* state to the *available* state. You may need to click **refresh** to see your new volume.

## Task 2: Attach the Volume to an Instance

You can now attach your new volume to the Amazon EC2 instance.

11. Select **My Volume**.

12. In the **Actions** menu, click **Attach Volume**.

13. Click in the **Instance** field, then select the instance that appears (Lab).

Note that the **Device** field is set to `/dev/sdf`. You will use this device identifier in a later task.

14. Click **Attach**

The volume state is now *in-use*.



## Task 3: Connect to Your Amazon EC2 Instance

### Windows Users: Using SSH to Connect

---

🔊 These instructions are for Windows users only.

If you are using Mac or Linux, skip to the next section.

15. To the left of the instructions you are currently reading, click **Download PPK**.

16. Save the file to the directory of your choice.

You will use PuTTY to SSH to Amazon EC2 instances.

If you do not have PuTTY installed on your computer, [download it here](#).

17. Open PuTTY.exe

18. Configure the PuTTY to not timeout:

- Click **Connection**
- Set **Seconds between keepalives** to 30

This allows you to keep the PuTTY session open for a longer period of time.

19. Configure your PuTTY session:

- Click **Session**
- **Host Name (or IP address):** Copy and paste the **PublicIP** shown to the left of these instructions
- In the **Connection** list, expand **SSH**
- Click **Auth** (don't expand it)
- Click **Browse**
- Browse to and select the PPK file that you downloaded
- Click **Open** to select it
- Click **Open**

20. Click **Yes**, to trust the host and connect to it.

21. When prompted **login as**, enter: `ec2-user`

This will connect to your EC2 instance.

22. Windows Users: Click here to skip ahead to the next task.

## Mac and Linux Users

---

These instructions are for Mac/Linux users only. If you are a Windows user, skip ahead to the next task.

23. To the left of the instructions you are currently reading, click **Download PEM**.

24. Save the file to the directory of your choice.

25. Copy this command to a text editor:

```
chmod 400 KEYPAIR.pem
```

```
ssh -i KEYPAIR.pem ec2-user@PublicIP
```

26. Replace *KEYPAIR.pem* with the path to the PEM file you downloaded.

27. Replace *PublicIP* with the values of PublicIP shown to the left of these instructions.

28. Paste the updated command into the Terminal window and run it.

29. Type *yes* when prompted to allow a first connection to this remote SSH server.

Because you are using a key pair for authentication, you will not be prompted for a password.

## Task 4: Create and Configure Your File System

In this task, you will add the new volume to a Linux instance as an ext3 file system under the `/mnt/data-store` mount point.

① If you are using PuTTY, you can paste text by right-clicking in the PuTTY window.

30. View the storage available on your instance:

```
df -h
```

You should see output similar to:

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	488M	60K	488M	1%	/dev
tmpfs	497M	0	497M	0%	/dev/shm
/dev/xvda1	7.8G	982M	6.7G	13%	/

This is showing the original 8GB disk volume. Your new volume is not yet shown.

31. Create an ext3 file system on the new volume:

```
sudo mkfs -t ext3 /dev/sdf
```

32. Create a directory for mounting the new storage volume:

```
sudo mkdir /mnt/data-store
```

33. Mount the new volume:

```
sudo mount /dev/sdf /mnt/data-store
```

To configure the Linux instance to mount this volume whenever the instance is started, you will need to add a line to `/etc/fstab`.

```
echo "/dev/sdf    /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab
```

34. View the configuration file to see the setting on the last line:

```
cat /etc/fstab
```

35. View the available storage again:

```
df -h
```

The output will now contain an additional line - `/dev/xvdf`:

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	488M	60K	488M	1%	/dev
tmpfs	497M	0	497M	0%	/dev/shm
/dev/xvda1	7.8G	982M	6.7G	13%	/
/dev/xvdf	976M	1.3M	924M	1%	/mnt/data-store

36. On your mounted volume, create a file and add some text to it.

```
sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
```

37. Verify that the text has been written to your volume.

```
cat /mnt/data-store/file.txt
```

## Task 5: Create an Amazon EBS Snapshot

In this task, you will create a snapshot of your EBS volume.

You can create any number of point-in-time, consistent snapshots from Amazon EBS volumes at any time. Amazon EBS snapshots are stored in Amazon S3 with high durability. New Amazon EBS volumes can be created out of snapshots for cloning or restoring backups. Amazon EBS snapshots can also be easily shared among AWS users or copied over AWS regions.

38. In the **AWS Management Console**, select **My Volume**.

39. In the **Actions** menu, click **Create Snapshot**.

40. Click **Add Tag** then configure:

- **Key:** Name
- **Value:** My Snapshot
- Click **Create Snapshot** then click **Close**

Your snapshot will be listed in the **Snapshots** console.

41. In the left navigation pane, click **Snapshots**.

Your snapshot is displayed. The state will start with a state of *pending*, which means that the snapshot is being created. It will then change to a state of *completed*. Only used storage blocks are copied to snapshots, so empty blocks do not take any snapshot storage space.

42. In your remote SSH session, delete the file that created on your volume.

```
sudo rm /mnt/data-store/file.txt
```

43. Verify that the file has been deleted.

```
ls /mnt/data-store/
```

Your file has been deleted.

## Task 6: Restore the Amazon EBS Snapshot

If you ever wish to retrieve data stored in a snapshot, you can **Restore** the snapshot to a new EBS volume.

### Create a Volume Using Your Snapshot

---

44. In the **AWS Management Console**, select **My Snapshot**.
45. In the **Actions** menu, click **Create Volume**.
46. For **Availability Zone** Select the same availability zone that you used earlier.
47. Click **Add Tag** then configure:
  - **Key:** Name
  - **Value:** Restored Volume
  - Click **Create Volume**
  - Click **Close**

When restoring a snapshot to a new volume, you can also modify the configuration, such as changing the volume type, size or Availability Zone.

### Attach the Restored Volume to Your EC2 Instance

---

48. In the left navigation pane, click **Volumes**.
  49. Select **Restored Volume**.
  50. In the **Actions** menu, click **Attach Volume**.
  51. Click in the **Instance** field, then select the instance that appears (Lab).
- Note that the **Device** field is set to `/dev/sdg`. You will use this device identifier in a later task.
52. Click **Attach**  
The volume state is now *in-use*.

### Mount the Restored Volume

---

53. Create a directory for mounting the new storage volume:

```
sudo mkdir /mnt/data-store2
```

54. Mount the new volume:

```
sudo mount /dev/sdg /mnt/data-store2
```

55. Verify that volume you mounted has the file that you created earlier.

```
ls /mnt/data-store2/
```

You should see file.txt.

## Conclusion

👏 Congratulations! You now have successfully:

- Created an Amazon EBS volume
- Attached the volume to an EC2 instance
- Created a file system on the volume
- Added a file to volume
- Created a snapshot of your volume
- Created a new volume from the snapshot
- Attached and mounted the new volume to your EC2 instance
- Verified that the file you created earlier was on the newly created volume

## Lab Complete

🏁 Congratulations! You have completed the lab.

Click **End Lab** at the top of this page to clean up your lab environment.

## Additional Resources

- [Amazon Elastic Block Store features, functions, and pricing](#)
- [AWS Training and Certification](#)

For feedback, suggestions, or corrections, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com)



# ACF Lab 3: Build your VPC and Launch a Web Server

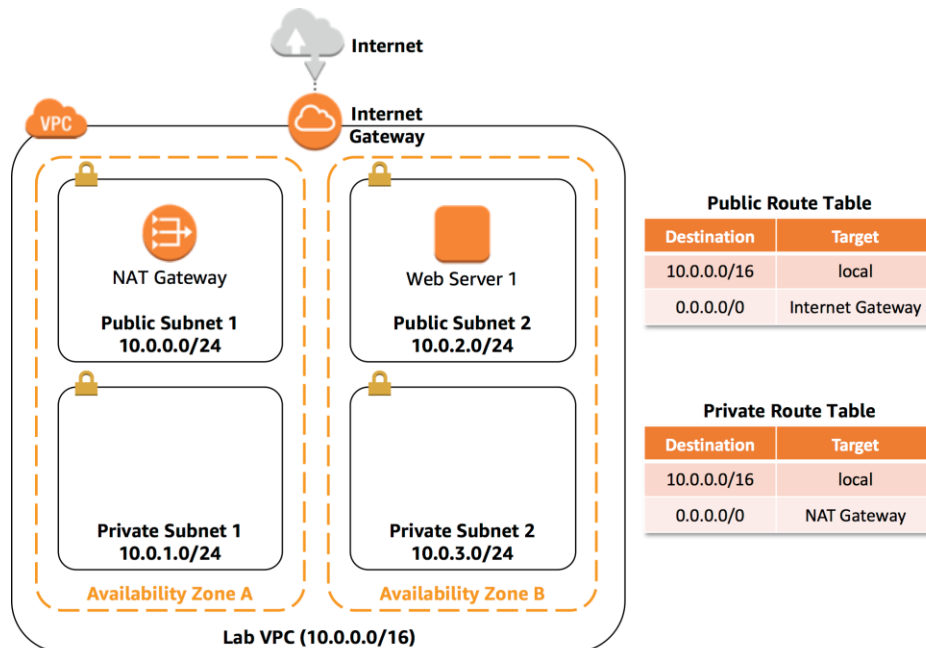
## Version 4.6.6 (TESS1)

In this lab, you will use Amazon Virtual Private Cloud (VPC) to create your own VPC and add additional components to produce a customized network. You will also create security groups for your EC2 instance. You will then configure and customize an EC2 instance to run a web server and launch it into the VPC.

**Amazon Virtual Private Cloud (Amazon VPC)** enables you to launch Amazon Web Services (AWS) resources into a virtual network that you defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS. You can create a VPC that spans multiple Availability Zones.

## Scenario

In this lab you build the following infrastructure:



## Objectives

After completing this lab, you can:

- Create a VPC.
- Create subnets.
- Configure a security group.

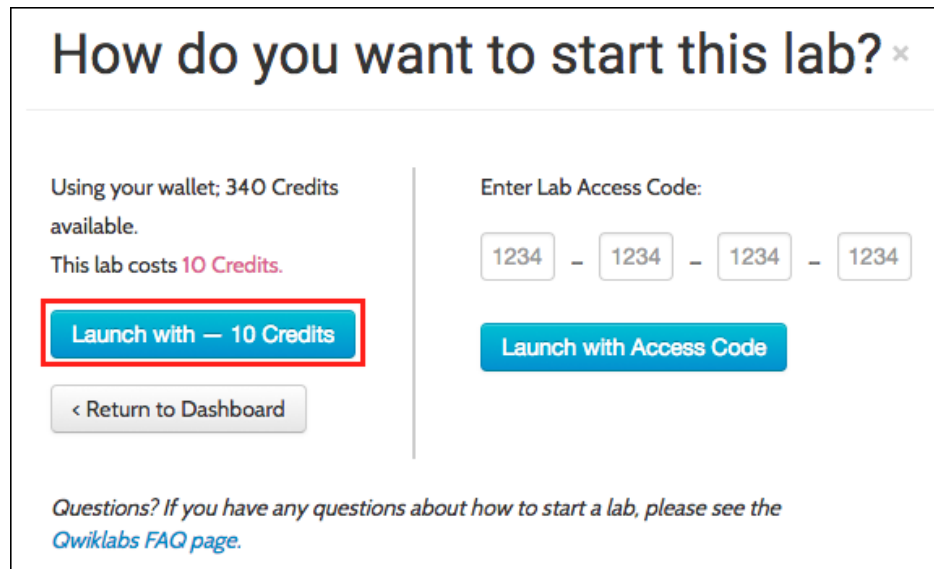
- Launch an EC2 instance into a VPC.

### Duration

This lab takes approximately **45 minutes** to complete.

## Accessing the AWS Management Console

1. At the top of these instructions, click **Start Lab** to launch your lab.
2. When asked *How do you want to start this lab*, click **Launch with credits**.



How do you want to start this lab? ✕

Using your wallet; 340 Credits available.  
This lab costs 10 Credits.

**Launch with — 10 Credits**

< Return to Dashboard

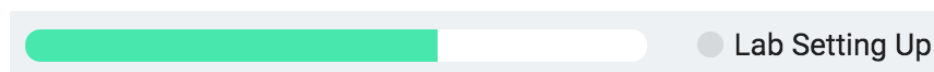
Enter Lab Access Code:

1234 - 1234 - 1234 - 1234

**Launch with Access Code**

Questions? If you have any questions about how to start a lab, please see the [Qwiklabs FAQ page](#).

A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.



Lab Setting Up

3. Click **Open Console**
4. Sign in using the **Username** and **Password** shown to the left of these instructions.

You will be taken to the AWS Management Console.

## Task 1: Create Your VPC

In this task, you will use the VPC Wizard to create a VPC an Internet Gateway and two subnets in a single Availability Zone. An **Internet gateway (IGW)** is a VPC component that allows communication between instances in your VPC and the Internet.

After creating a VPC, you can add **subnets**. Each subnet resides entirely within one Availability Zone and cannot span zones. If a subnet's traffic is routed to an Internet Gateway, the subnet is known as a *public subnet*. If a subnet does not have a route to the Internet gateway, the subnet is known as a *private subnet*.

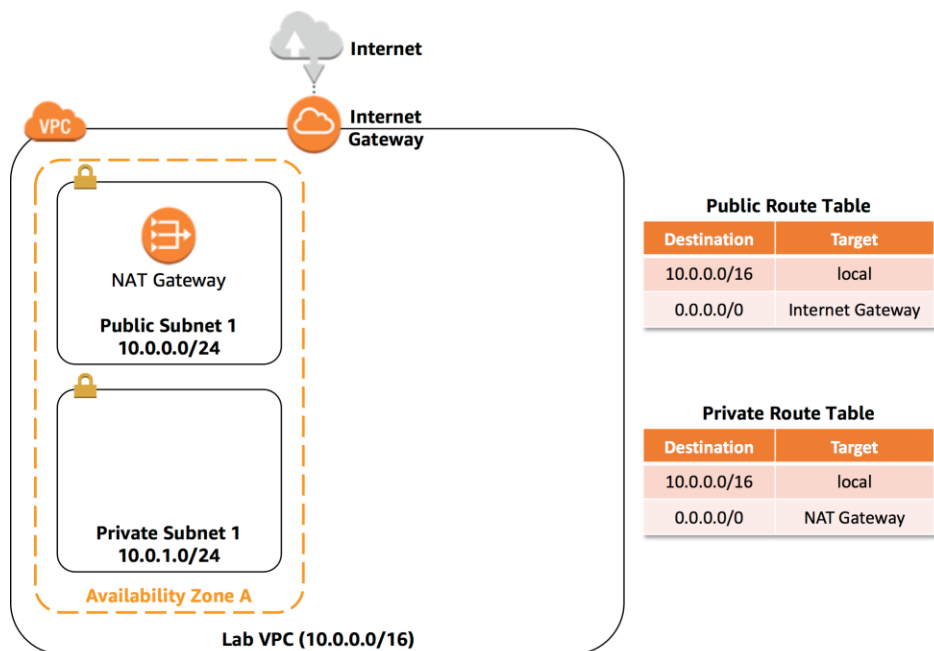
The wizard will also create a *NAT Gateway*, which is used to provide internet connectivity to EC2 instances in the private subnets.

5. In the **AWS Management Console**, on the **Services** menu, click **VPC**.
6. Click **Launch VPC Wizard**
7. In the left navigation pane, click **VPC with Public and Private Subnets** (the second option).
8. Click **Select** then configure:
  - **VPC name:** `Lab VPC`
  - **Availability Zone:** Select the *first* Availability Zone
  - **Public subnet name:** `Public Subnet 1`
  - **Availability Zone:** Select the *first* Availability Zone (the same as used above)
  - **Private subnet name:** `Private Subnet 1`
  - **Elastic IP Allocation ID:** Click in the box and select the displayed IP address
9. Click **Create VPC**

The wizard will create your VPC.

10. Once it is complete, click **OK**

The wizard has provisioned a VPC with a public subnet and a private subnet in the same Availability Zone, together with route tables for each subnet:



The Public Subnet has a CIDR of **10.0.0.0/24**, which means that it contains all IP addresses starting with **10.0.0.x**.

The Private Subnet has a CIDR of **10.0.1.0/24**, which means that it contains all IP addresses starting with **10.0.1.x**.

## Task 2: Create Additional Subnets

In this task, you will create two additional subnets in a second Availability Zone. This is useful for creating resources in multiple Availability Zones to provide *High Availability*.

11. In the left navigation pane, click **Subnets**.

First, you will create a second Public Subnet.

12. Click **Create subnet** then configure:

- **Name tag:** `Public Subnet 2`
- **VPC:** `Lab VPC`
- **Availability Zone:** Select the *second* Availability Zone
- **IPv4 CIDR block:** `10.0.2.0/24`

The subnet will have all IP addresses starting with **10.0.2.x**.

13. Click **Create** then click **Close**

You will now create a second Private Subnet.

14. Click **Create subnet** then configure:

- **Name tag:** `Private Subnet 2`
- **VPC:** `Lab VPC`
- **Availability Zone:** Select the *second* Availability Zone
- **CIDR block:** `10.0.3.0/24`

The subnet will have all IP addresses starting with **10.0.3.x**.

15. Click **Create** then click **Close**

You will now configure the Private Subnets to route internet-bound traffic to the NAT Gateway so that resources in the Private Subnet are able to connect to the Internet, while still keeping the resources private. This is done by configuring a *Route Table*.

A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table; the route table controls routing for the subnet.

16. In the left navigation pane, click **Route Tables**.

17. Select ☒ the route table with **Main = Yes** and **VPC = Lab VPC**. (Expand the *VPC ID* column if necessary to view the VPC name.)

18. In the lower pane, click the **Routes** tab.

Note that **Destination 0.0.0.0/0** is set to **Target nat-xxxxxxx**. This means that traffic destined for the internet (0.0.0.0/0) will be sent to the NAT Gateway. The NAT Gateway will then forward the traffic to the internet.

This route table is therefore being used to route traffic from Private Subnets. You will now add a name to the Route Table to make this easier to recognize in future.

19. In the **Name** column for this route table, click the pencil then type `Private Route Table` and click

20. In the lower pane, click the **Subnet Associations** tab.

You will now associate this route table to the Private Subnets.

21. Click **Edit subnet associations**

22. Select ☒ both **Private Subnet 1** and **Private Subnet 2**.

💡 You can expand the *Subnet ID* column to view the Subnet names.

23. Click **Save**

You will now configure the Route Table that is used by the Public Subnets.

24. Select ☒ the route table with **Main = No** and **VPC = Lab VPC** (and deselect any other subnets).

25. In the **Name** column for this route table, click the pencil then type `Public Route Table`, and click

26. In the lower pane, click the **Routes** tab.

Note that **Destination 0.0.0.0/0** is set to **Target igw-xxxxxxx**, which is the Internet Gateway. This means that internet-bound traffic will be sent straight to the internet via the Internet Gateway.

You will now associate this route table to the Public Subnets.

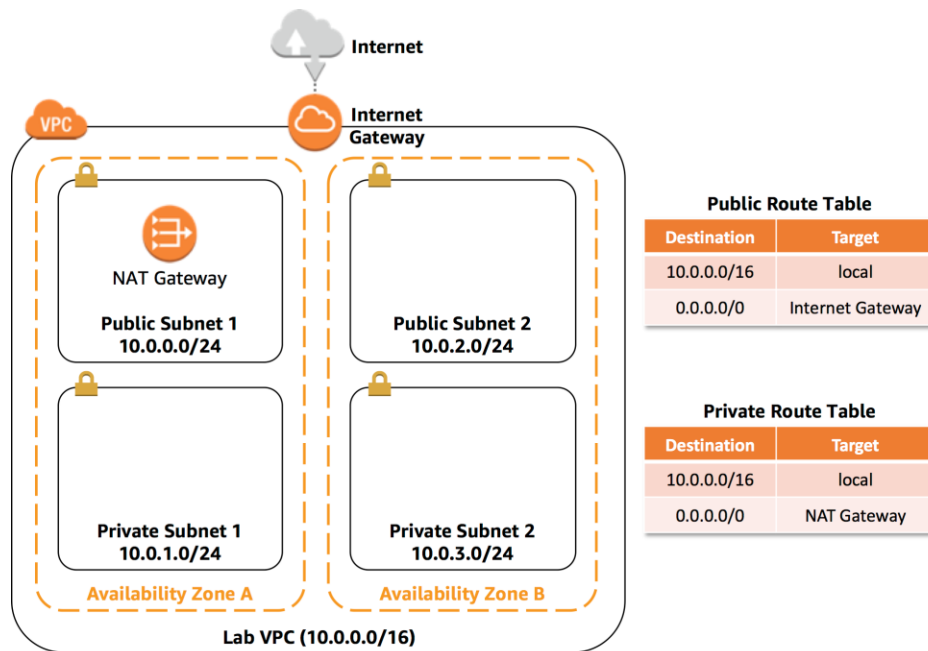
27. Click the **Subnet Associations** tab.

28. Click **Edit subnet associations**

29. Select ☒ both **Public Subnet 1** and **Public Subnet 2**.

30. Click **Save**

Your VPC now has public and private subnets configured in two Availability Zones:



## Task 3: Create a VPC Security Group

In this task, you will create a VPC security group, which acts as a virtual firewall. When you launch an instance, you associate one or more security groups with the instance. You can add rules to each security group that allow traffic to or from its associated instances.

31. In the left navigation pane, click **Security Groups**.

32. Click **Create security group** and then configure:

- **Security group name:** `Web Security Group`
- **Description:** `Enable HTTP access`
- **VPC:** `Lab VPC`

33. Click **Create** then click **Close**

You will now add a rule to the security group to permit inbound web requests.

34. Select ☒ **Web Security Group**.

35. Click the **Inbound Rules** tab.

36. Click **Edit rules**

37. Click **Add Rule** then configure:

- **Type:** `HTTP`
- **Source:** `Anywhere`
- **Description:** `Permit web requests`

38. Click **Save rules** then click **Close**

You will use this security group in the next task when launching an Amazon EC2 instance.



## Task 4: Launch a Web Server Instance

In this task, you will launch an Amazon EC2 instance into the new VPC. You will configure the instance to act as a web server.

39. On the **Services** menu, click **EC2**.

40. Click **Launch Instance**

First, you will select an *Amazon Machine Image (AMI)*, which contains the desired Operating System.

41. In the row for **Amazon Linux 2** (at the top), click **Select**

The *Instance Type* defines the hardware resources assigned to the instance.

42. Select **t2.micro** (shown in the *Type* column).

43. Click **Next: Configure Instance Details**

You will now configure the instance to launch in a Public Subnet of the new VPC.

44. Configure these settings:

- **Network:** *Lab VPC*
- **Subnet:** *Public Subnet 2 (not Private!)*
- **Auto-assign Public IP:** *Enable*

45. Expand the ► **Advanced Details** section (at the bottom of the page).

46. Copy and paste this code into the **User data** box:

```
#!/bin/bash
# Install Apache Web Server and PHP
yum install -y httpd mysql php
# Download Lab files
wget https://us-west-2-tcprod.s3.amazonaws.com/courses/ILT-CUR-100-ACFNDS/v1.0.18/acf-lab3-vpc/scripts/lab-app.zip
unzip lab-app.zip -d /var/www/html/
# Turn on web server
chkconfig httpd on
service httpd start
```

This script will be run automatically when the instance launches for the first time. The script loads and configures a PHP web application.

47. Click **Next: Add Storage**

You will use the default settings for storage.

48. Click **Next: Add Tags**

Tags can be used to identify resources. You will use a tag to assign a Name to the instance.

49. Click **Add Tag** then configure:

- **Key:** Name
- **Value:** Web Server 1

50. Click **Next: Configure Security Group**

You will configure the instance to use the *Web Security Group* that you created earlier.

51. Select **Select an existing security group**

52. Select ☒ **Web Security Group**.

This is the security group you created in the previous task. It will permit HTTP access to the instance.

53. Click **Review and Launch**

54. When prompted with a *warning* that you will not be able to connect to the instance through port 22, click **Continue**

55. Review the instance information and click **Launch**

56. In the **Select an existing keypair** dialog, select ☒ **I acknowledge....**

57. Click **Launch Instances** and then click **View Instances**

58. Wait until **Web Server 1** shows *2/2 checks passed* in the **Status Checks** column.

💡 This may take a few minutes. Click refresh in the top-right every 30 seconds for updates.

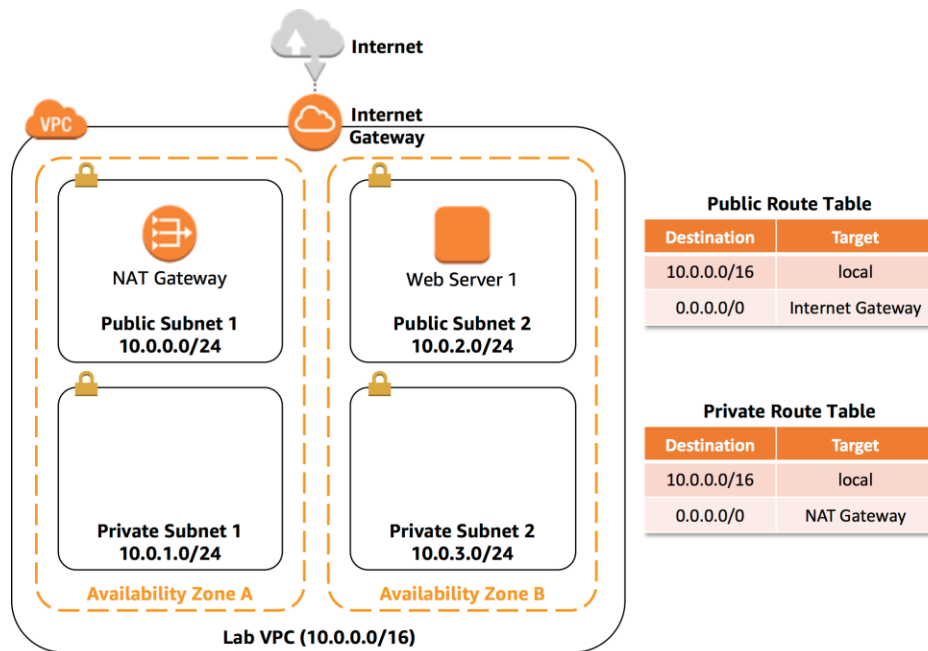
You will now connect to the web server running on the EC2 instance.

59. Copy the **Public DNS (IPv4)** value shown in the **Description** tab at the bottom of the page.

60. Open a new web browser tab, paste the **Public DNS** value and press Enter.

You should see a web page displaying the AWS logo and instance meta-data values. You will use this web application in the next lab.

The complete architecture you deployed is:



## Lab Complete

🏁 Congratulations! You have completed the lab.

Click **End Lab** at the top of this page to clean up your lab environment.

# ACF Lab 4: Build Your DB Server and Interact With Your DB Using an App

## Version 4.6.6 (TESS2)

This lab is designed to reinforce the concept of leveraging an AWS-managed database instance for solving relational database needs.

**Amazon Relational Database Service** (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

## Objectives

After completing this lab, you can:

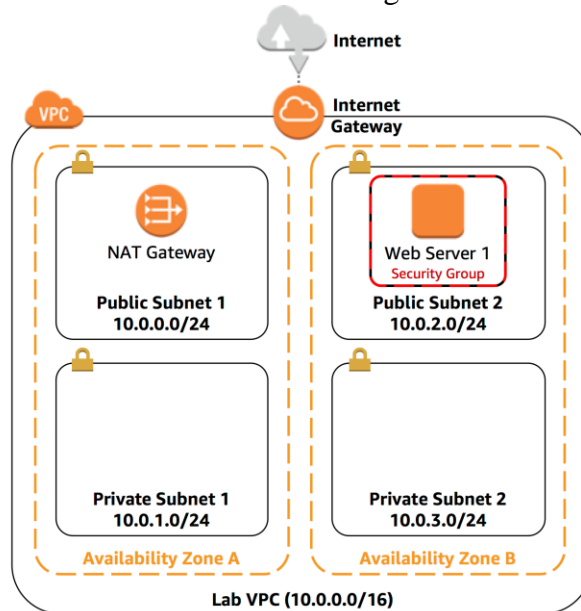
- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

## Duration

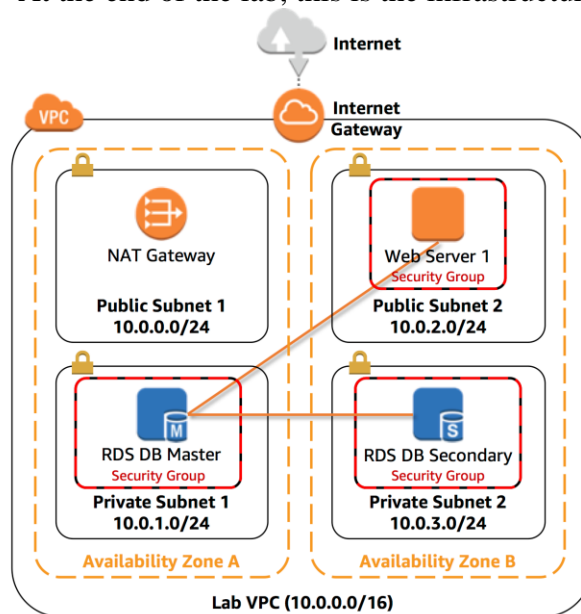
This lab takes approximately **45 minutes**.

## Scenario

You start with the following infrastructure:



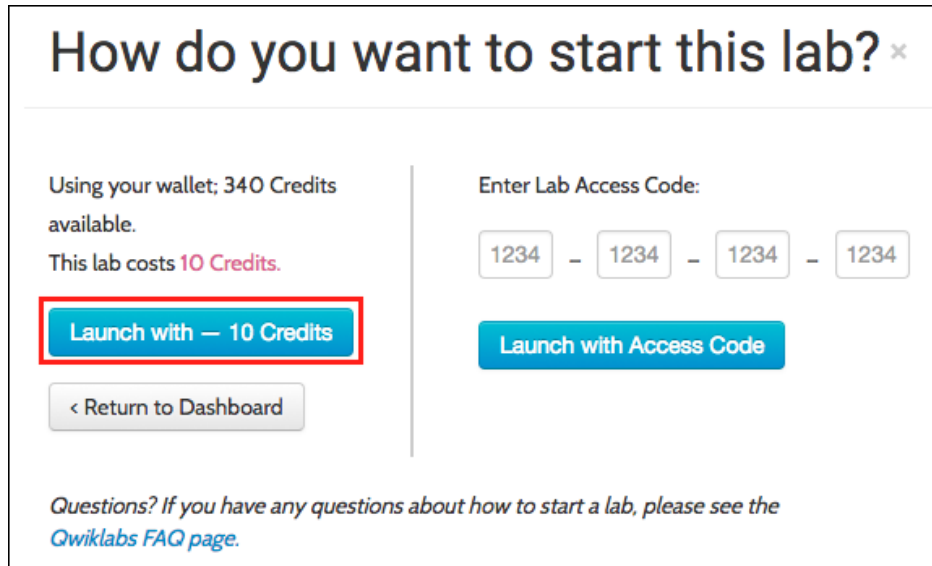
At the end of the lab, this is the infrastructure:



## Accessing the AWS Management Console

⚠ **Windows Users:** Please use Chrome or Firefox as your web browser for this lab. The lab instructions are **not compatible with Internet Explorer** due to a difference in the Amazon RDS console.

1. At the top of these instructions, click **Start Lab** to launch your lab.
2. When asked *How do you want to start this lab*, click **Launch with credits**.



How do you want to start this lab? x

Using your wallet; 340 Credits available.  
This lab costs 10 Credits.

**Launch with — 10 Credits**

< Return to Dashboard

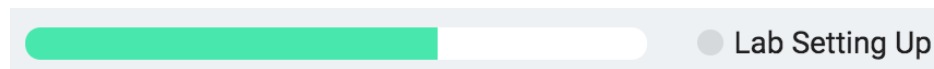
Enter Lab Access Code:

1234 - 1234 - 1234 - 1234

**Launch with Access Code**

Questions? If you have any questions about how to start a lab, please see the [Qwiklabs FAQ page](#).

A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.



3. Click **Open Console**
4. Sign in using the **Username** and **Password** shown to the left of these instructions.

You will be taken to the AWS Management Console.





## Task 1: Create a Security Group for the RDS DB Instance

In this task, you will create a security group to allow your web server to access your RDS DB instance. The security group will be used when you launch the database instance.

5. In the **AWS Management Console**, on the **Services** menu, click **VPC**.

6. In the left navigation pane, click **Security Groups**.

7. Click **Create security group** and then configure:

- **Security group name:** `DB Security Group`
- **Description:** `Permit access from Web Security Group`
- **VPC:** *Lab VPC*

8. Click **Create** then click **Close**

You will now add a rule to the security group to permit inbound database requests.

9. Select ☒ **DB Security Group**.

10. Click the **Inbound Rules** tab.

The security group currently has no rules. You will add a rule to permit access from the *Web Security Group*.

11. Click **Edit rules**

12. Click **Add Rule** then configure:

- **Type:** *MySQL/Aurora (3306)*
- **CIDR, IP, Security Group or Prefix List:** Type `sg` and then select *Web Security Group*.

This configures the Database security group to permit inbound traffic on port 3306 from any EC2 instance that is associated with the *Web Security Group*.

13. Click **Save rules** then click **Close**

You will use this security group when launching the Amazon RDS database.

## Task 2: Create a DB Subnet Group

In this task, you will create a *DB subnet group* that is used to tell RDS which subnets can be used for the database. Each DB subnet group requires subnets in at least two Availability Zones.

14. On the **Services** menu, click **RDS**.

15. In the left navigation pane, click **Subnet groups**.

△ If the navigation pane is not visible, click the ☰ menu icon in the top-left corner.

16. Click **Create DB Subnet Group** then configure:

- **Name:** DB Subnet Group
- **Description:** DB Subnet Group
- **VPC ID:** *Lab VPC*
- **Availability zone:** Select the *first* Availability Zone
- **Subnet:** *10.0.1.0/24*
- Click **Add subnet**

This added Private Subnet 1. You will now add Private Subnet 2.

17. Configure these settings (on the existing screen):

- **Availability zone:** Select the *second* Availability Zone
- **Subnet:** *10.0.3.0/24*
- Click **Add subnet**

These subnets should now be shown in the list: **10.0.1.0/24** and **10.0.3.0/24**

18. Click **Create**

You will use this DB subnet group when creating the database in the next task.

## Task 3: Create an Amazon RDS DB Instance

In this task, you will configure and launch a Multi-AZ Amazon RDS for MySQL database instance.

Amazon RDS **Multi-AZ** deployments provide enhanced availability and durability for Database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ).

19. In the left navigation pane, click **Databases**.

20. Click **Create database**

⚠ If you see **Switch to the new database creation flow** at the top of the screen, please click it.

21. Select **MySQL**.

22. Under **Settings**, configure:

- **DB instance identifier:** `lab-db`
- **Master username:** `master`
- **Master password:** `lab-password`
- **Confirm password:** `lab-password`

23. Under **DB instance size**, configure:

- Select **Burstable classes (includes t classes)**.
- Select `db.t2.micro`

24. Under **Storage**, configure:

- **Storage type:** *General Purpose (SSD)*
- **Allocated storage:** `20`

25. Under **Connectivity**, configure:

- **Virtual Private Cloud (VPC):** *Lab VPC*

26. Expand ► **Additional connectivity configuration**, then configure:

- For **Existing VPC security groups:** click *DB Security Group* to highlight it in blue.

27. Expand ► **Additional configuration**, then configure:

- **Initial database name:** `lab`

- Uncheck **Enable automatic backups**.
- Uncheck **Enable Enhanced monitoring**.

☛ This will turn off backups, which is not normally recommended, but will make the database deploy faster for this lab.

28. Click **Create database**

Your database will now be launched.

29. Click **lab-db** (click the link itself).

You will now need to wait **approximately 4 minutes** for the database to be available. The deployment process is deploying a database in two different Availability zones.

① While you are waiting, you might want to review the [Amazon RDS FAQs](#) or grab a cup of coffee.

30. Wait until **Info** changes to **Modifying** or **Available**.

31. Scroll down to the **Connectivity** section and copy the **Endpoint** field.

It will look similar to: *lab-db.cgqg8lhnxvnx.us-west-2.rds.amazonaws.com*

32. Paste the Endpoint value into a text editor. You will use it later in the lab.

## Task 4: Interact with Your Database

In this task, you will open a web application running on your web server and configure it to use the database.

33. Copy the **WebServer** IP address shown to the left of the instructions you are currently reading.

34. Open a new web browser tab, paste the *WebServer* IP address and press Enter.

The web application will be displayed, showing information about the EC2 instance.

35. Click the **RDS** link at the top of the page.

You will now configure the application to connect to your database.

36. Configure the following settings:

- **Endpoint:** Paste the Endpoint you copied to a text editor earlier
- **Database:** `lab`
- **Username:** `master`
- **Password:** `lab-password`
- Click **Submit**

A message will appear explaining that the application is executing a command to copy information to the database. After a few seconds the application will display an **Address Book**.

The Address Book application is using the RDS database to store information.

37. Test the web application by adding, editing and removing contacts.

The data is being persisted to the database and is automatically replicating to the second Availability Zone.

## Lab Complete

🏁 Congratulations! You have completed the lab.

Click **End Lab** at the top of this page to clean up your lab environment.

## Attributions

---

**Bootstrap v3.3.5** - <http://getbootstrap.com/>

The MIT License (MIT)

Copyright (c) 2011-2016 Twitter, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# ACF Lab 5: Scale and Load Balance Your Architecture

## Version 4.6.6 (TESS3)

This lab walks you through using the Elastic Load Balancing (ELB) and Auto Scaling services to load balance and automatically scale your infrastructure.

**Elastic Load Balancing** automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications by seamlessly providing the required amount of load balancing capacity needed to route application traffic.

**Auto Scaling** helps you maintain application availability and allows you to scale your Amazon EC2 capacity out or in automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances. Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs. Auto Scaling is well suited to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

## Objectives

After completing this lab, you can:

- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

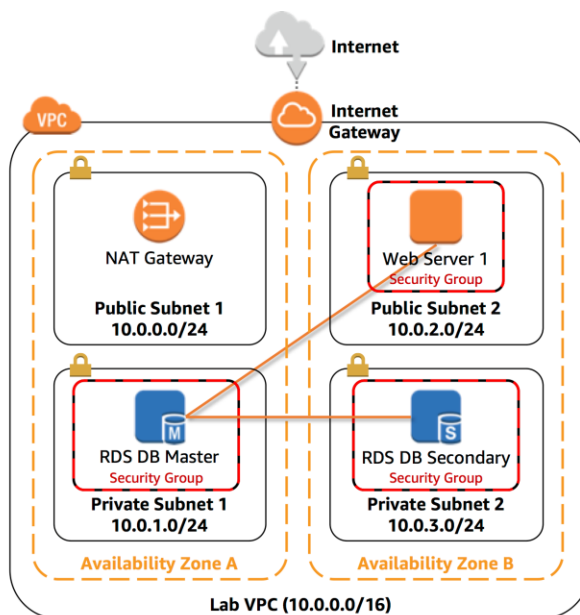
## Duration

This lab takes approximately **45 minutes**.

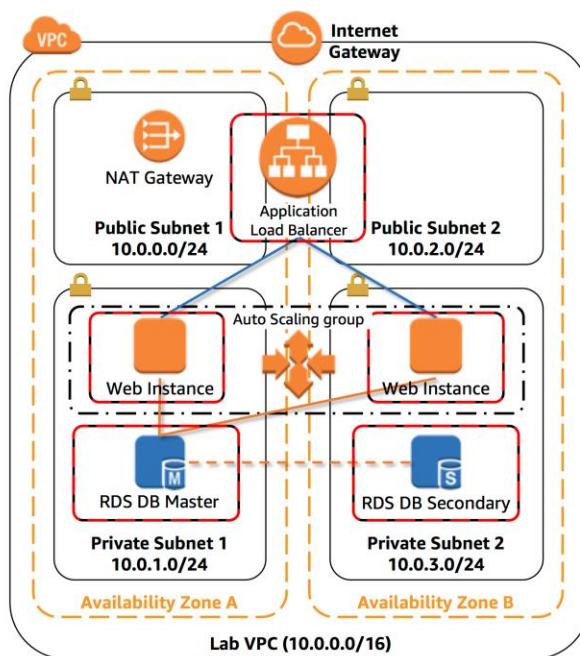
## Scenario

You start with the following infrastructure:



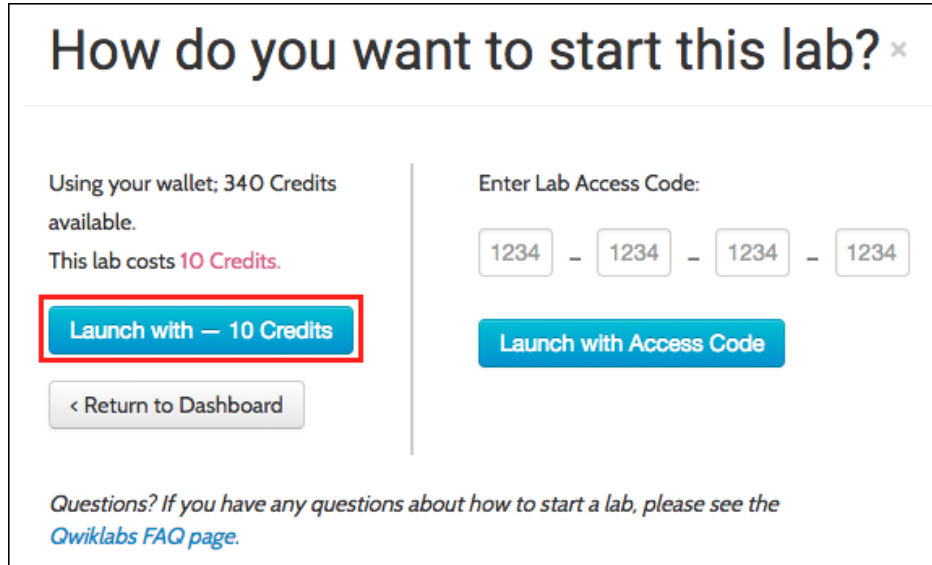


The final state of the infrastructure is:



## Accessing the AWS Management Console

1. At the top of these instructions, click **Start Lab** to launch your lab.
2. When asked *How do you want to start this lab*, click **Launch with credits**.



How do you want to start this lab? ✕

Using your wallet; 340 Credits available.  
This lab costs 10 Credits.

Launch with — 10 Credits

< Return to Dashboard

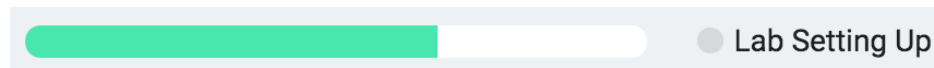
Enter Lab Access Code:

1234 - 1234 - 1234 - 1234

Launch with Access Code

Questions? If you have any questions about how to start a lab, please see the [Qwiklabs FAQ page](#).

A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.



3. Click **Open Console**
4. Sign in using the **Username** and **Password** shown to the left of these instructions.

You will be taken to the AWS Management Console.

## Task 1: Create an AMI for Auto Scaling

In this task, you will create an AMI from the existing *Web Server 1*. This will save the contents of the boot disk so that new instances can be launched with identical content.

5. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
6. In the left navigation pane, click **Instances**.

First, you will confirm that the instance is running.

7. Wait until the **Status Checks** for **Web Server 1** displays *2/2 checks passed*. Click refresh to update.

You will now create an AMI based upon this instance.

8. Select ☒ **Web Server 1**.
9. In the **Actions** menu, click **Image > Create Image**, then configure:
  - **Image name:** `Web Server AMI`
  - **Image description:** `Lab AMI for Web Server`
10. Click **Create Image**

The confirmation screen displays the **AMI ID** for your new AMI.

11. Click **Close**

You will use this AMI when launching the Auto Scaling group later in the lab.

## Task 2: Create a Load Balancer

In this task, you will create a load balancer that can balance traffic across multiple EC2 instances and Availability Zones.

12. In the left navigation pane, click **Load Balancers**.

13. Click **Create Load Balancer**

Several different types of load balancer are displayed. You will be using an *Application Load Balancer* that operates at the request level (layer 7), routing traffic to targets — EC2 instances, containers, IP addresses and Lambda functions — based on the content of the request. For more information, see: [Comparison of Load Balancers](#)

14. Under **Application Load Balancer** click **Create** and configure:

- **Name:** LabELB
- **VPC:** Lab VPC (In the **Availability Zones** section)
- **Availability Zones:** Select ☒ both to see the available subnets.
- Select **Public Subnet 1** and **Public Subnet 2**

This configures the load balancer to operate across multiple Availability Zones.

15. Click **Next: Configure Security Settings**

🗨 You can ignore the "*Improve your load balancer's security.*" warning.

16. Click **Next: Configure Security Groups**

A *Web Security Group* has already been created for you, which permits HTTP access.

17. Select ☒ **Web Security Group** and deselect **default**.

18. Click **Next: Configure Routing**

Routing configures where to send requests that are sent to the load balancer. You will create a *Target Group* that will be used by Auto Scaling.

19. For **Name**, enter: LabGroup

20. Click **Next: Register Targets**

Auto Scaling will automatically register instances as targets later in the lab.

21. Click **Next: Review**

22. Click **Create** then click **Close**

The load balancer will show a state of *provisioning*. There is no need to wait until it is ready. Please continue with the next task.

## Task 3: Create a Launch Configuration and an Auto Scaling Group

In this task, you will create a *launch configuration* for your Auto Scaling group. A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the AMI, the instance type, a key pair, security group and disks.

23. In the left navigation pane, click **Launch Configurations**.

24. Click **Create launch configuration**

25. In the left navigation pane, click **My AMIs**.

You will use the AMI that you created from the existing *Web Server 1*.

26. In the row for **Web Server AMI**, click **Select**

27. Accept the **t2.micro** selection and click **Next: Configure details**

28. Configure these settings:

- **Name:** `LabConfig`
- **Monitoring:** Select ☒ **Enable CloudWatch detailed monitoring**

This allows Auto Scaling to react quickly to changing utilization.

29. Click **Next: Add Storage**

You will use the default storage settings.

30. Click **Next: Configure Security Group**

You will configure the launch configuration to use the *Web Security Group* that has already been created for you.

31. Configure these settings:

- Click **Select an existing security group**
- Select ☒ **Web Security Group**
- Click **Review**

32. Review the details of your launch configuration and click **Create launch configuration**

🗨 You can ignore the warning "Improve security..." warning.

33. On the **Select an existing key pair** dialog:


- Select ☒ **I acknowledge...**

- Click **Create launch configuration**

You will now create an Auto Scaling group that uses this Launch Configuration.

34. Click **Create an Auto Scaling group using this launch configuration**

35. Configure the following settings:

- **Group name:** `Lab Auto Scaling Group`
- **Group size:** Start with: 2 instances
- **Network:** `Lab VPC`
-  You can ignore the message regarding "No public IP address"
- **Subnet:** Select *Private Subnet 1 (10.0.1.0/24)* and *Private Subnet 2 (10.0.3.0/24)*

This will launch EC2 instances in private subnets across both Availability Zones.

36. Expand ► **Advanced Details**, then configure:

- **Load Balancing:** Select ☒ **Receive traffic from one or more load balancers**
- **Target Groups:** `LabGroup`
- **Monitoring:** Select ☒ **Enable CloudWatch detailed monitoring**

This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns.

37. Click **Next: Configure scaling policies**

38. Select **Use scaling policies to adjust the capacity of this group.**

39. Modify the **Scale between** text boxes to scale between **2** and **6** instances.

This will allow Auto Scaling to automatically add/remove instances, always keeping between 2 and 6 instances running.

40. In **Scale Group Size**, configure:

- **Metric type:** *Average CPU Utilization*
- **Target value:** 60

This tells Auto Scaling to maintain an *average* CPU utilization *across all instances* at 60%. Auto Scaling will automatically add or remove capacity as required to keep the metric at, or close to, the specified target value. It adjusts to fluctuations in the metric due to a fluctuating load pattern.

41. Click **Next: Configure Notifications**

Auto Scaling can send a notification when a scaling event takes place. You will use the default settings.

42. Click **Next: Configure Tags**

Tags applied to the Auto Scaling group will be automatically propagated to the instances that are launched.

43. Configure the following:

- **Key:** Name
- **Value:** Lab Instance

44. Click **Review**

45. Review the details of your Auto Scaling group, then click **Create Auto Scaling group**

46. Click **Close** when your Auto Scaling group has been created.

Your Auto Scaling group will initially show an instance count of zero, but new instances will be launched to reach the **Desired** count of 2 instances.



## Task 4: Verify that Load Balancing is Working

In this task, you will verify that Load Balancing is working correctly.

47. In the left navigation pane, click **Instances**.

You should see two new instances named **Lab Instance**. These were launched by Auto Scaling.

● If the instances or names are not displayed, wait 30 seconds and click **refresh** in the top-right.

First, you will confirm that the new instances have passed their Health Check.

48. In the left navigation pane, click **Target Groups** (in the *Load Balancing* section).

49. Click the **Targets** tab.

Two **Lab Instance** targets should be listed for this target group.

50. Wait until the **Status** of both instances transitions to *healthy*. Click **Refresh** in the upper-right to check for updates.

*Healthy* indicates that an instance has passed the Load Balancer's health check. This means that the Load Balancer will send traffic to the instance.

You can now access the Auto Scaling group via the Load Balancer.

51. In the left navigation pane, click **Load Balancers**.

52. In the lower pane, copy the **DNS name** of the load balancer, making sure to omit "(A Record)".

It should look similar to: *LabELB-1998580470.us-west-2.elb.amazonaws.com*

53. Open a new web browser tab, paste the DNS Name you just copied, and press **Enter**.

The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.

## Task 5: Test Auto Scaling

You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

54. Return to the AWS management console, but do not close the application tab — you will return to it soon.

55. On the **Services** menu, click **CloudWatch**.

56. In the left navigation pane, click **Alarms** (*not* **ALARM**).

Two alarms will be displayed. These were created automatically by the Auto Scaling group. They will automatically keep the average CPU load close to 60% while also staying within the limitation of having two to six instances.

57. Click the **OK** alarm, which has *AlarmHigh* in its name.

💡 If no alarm is showing **OK**, wait a minute then click refresh in the top-right until the alarm status changes.

The **OK** indicates that the alarm has *not* been triggered. It is the alarm for **CPU Utilization > 60**, which will add instances when average CPU is high. The chart should show very low levels of CPU at the moment.

You will now tell the application to perform calculations that should raise the CPU level.

58. Return to the browser tab with the web application.

59. Click **Load Test** beside the AWS logo.

This will cause the application to generate high loads. The browser page will automatically refresh so that all instances in the Auto Scaling group will generate load. Do not close this tab.

60. Return to browser tab with the **CloudWatch** console.

In less than 5 minutes, the **AlarmLow** alarm should change to **OK** and the **AlarmHigh** alarm status should change to *ALARM*.

💡 You can click Refresh in the top-right every 60 seconds to update the display.

You should see the **AlarmHigh** chart indicating an increasing CPU percentage. Once it crosses the 60% line for more than 3 minutes, it will trigger Auto Scaling to add additional instances.

61. Wait until the **AlarmHigh** alarm enters the *ALARM* state.

You can now view the additional instance(s) that were launched.

62. On the **Services** menu, click **EC2**.

63. In the left navigation pane, click **Instances**.

More than two instances labeled **Lab Instance** should now be running. The new instance(s) were created by Auto Scaling in response to the Alarm.

## Task 6: Terminate Web Server 1

In this task, you will terminate *Web Server 1*. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.

64. Select ☒ **Web Server 1** (and ensure it is the only instance selected).
65. In the **Actions** menu, click **Instance State > Terminate**.
66. Click **Yes, Terminate**

## Lab Complete

🚩 Congratulations! You have completed the lab.

Click **End Lab** at the top of this page to clean up your lab environment.

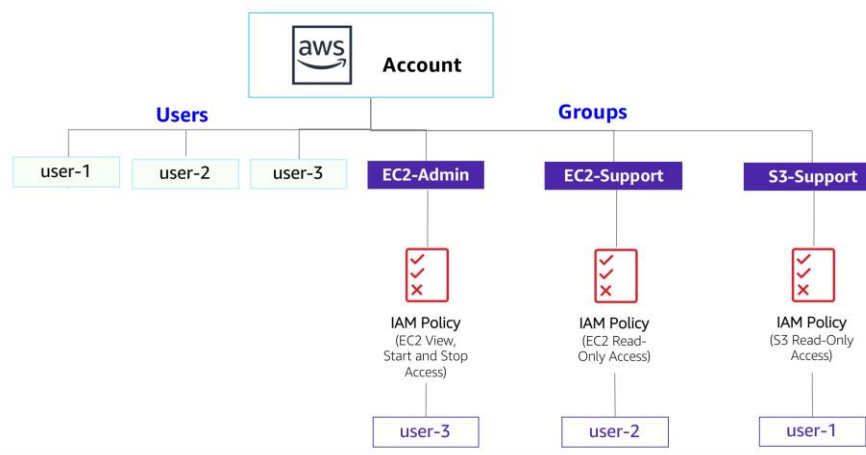
# ACF Lab 6: Introduction to AWS IAM

## Version 3.1.2 (spl66)

**AWS Identity and Access Management (IAM)** is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. With IAM, you can centrally manage **users**, **security credentials** such as access keys, and **permissions** that control which AWS resources users can access.

## Topics covered

This lab will demonstrate:



- Exploring pre-created **IAM Users and Groups**
- Inspecting **IAM policies** as applied to the pre-created groups
- Following a **real-world scenario**, adding users to groups with specific capabilities enabled
- Locating and using the **IAM sign-in URL**
- **Experimenting** with the effects of policies on service access

## Other AWS Services

During this lab, you may receive error messages when performing actions beyond the steps in this lab guide. These messages will not impact your ability to complete the lab.

## AWS Identity and Access Management

AWS Identity and Access Management (IAM) can be used to:

- **Manage IAM Users and their access:** You can create Users and assign them individual security credentials (access keys, passwords, and multi-factor authentication devices). You can manage permissions to control which operations a User can perform.
- **Manage IAM Roles and their permissions:** An IAM Role is similar to a User, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a Role is intended to be *assumable* by anyone who needs it.
- **Manage federated users and their permissions:** You can enable *identity federation* to allow existing users in your enterprise to access the AWS Management Console, to call AWS APIs and to access resources, without the need to create an IAM User for each identity.

### Duration

This lab takes approximately **45 minutes** to complete.

## Accessing the AWS Management Console

1. At the top of these instructions, click **Start Lab** to launch your lab.
2. When asked *How do you want to start this lab*, click **Launch with credits**.

How do you want to start this lab? ✕

Using your wallet; 340 Credits available.  
This lab costs 10 Credits.

**Launch with — 10 Credits**

< Return to Dashboard

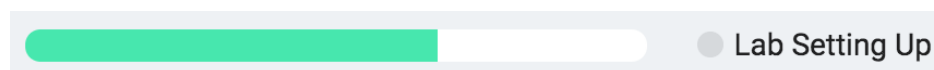
Enter Lab Access Code:

1234 - 1234 - 1234 - 1234

**Launch with Access Code**

Questions? If you have any questions about how to start a lab, please see the [Qwiklabs FAQ page](#).

A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.



3. Click **Open Console**
4. Sign in using the **Username** and **Password** shown to the left of these instructions.

You will be taken to the AWS Management Console.

## Task 1: Explore the Users and Groups

In this task, you will explore the Users and Groups that have already been created for you in IAM.

5. In the **AWS Management Console**, on the **Services** menu, click **IAM**.
6. In the navigation pane on the left, click **Users**.

The following IAM Users have been created for you:

- user-1
- user-2
- user-3

There is also an **awsstudent** user, which you can ignore for this lab.

7. Click **user-1**.

This will bring to a summary page for user-1. The **Permissions** tab will be displayed.

8. Notice that user-1 does not have any permissions.
9. Click the **Groups** tab.

user-1 also is not a member of any groups.

10. Click the **Security credentials** tab.

user-1 is assigned a **Console password**

11. In the navigation pane on the left, click **Groups**.

The following groups have already been created for you:

- EC2-Admin
- EC2-Support
- S3-Support

12. Click the **EC2-Support** group.

This will bring you to the summary page for the **EC2-Support** group.



13. Click the **Permissions** tab.

This group has a Managed Policy associated with it, called **AmazonEC2ReadOnlyAccess**. Managed Policies are pre-built policies (built either by AWS or by your administrators) that can be attached to IAM Users and Groups. When the policy is updated, the changes to the policy are immediately apply against all Users and Groups that are attached to the policy.

14. Under **Actions**, click the **Show Policy** link.

A policy defines what actions are allowed or denied for specific AWS resources. This policy is granting permission to List and Describe information about EC2, Elastic Load Balancing, CloudWatch and Auto Scaling. This ability to view resources, but not modify them, is ideal for assigning to a Support role.

The basic structure of the statements in an IAM Policy is:

- **Effect** says whether to *Allow* or *Deny* the permissions.
- **Action** specifies the API calls that can be made against an AWS Service (eg *cloudwatch:ListMetrics*).
- **Resource** defines the scope of entities covered by the policy rule (eg a specific Amazon S3 bucket or Amazon EC2 instance, or \* which means *any resource*).

15. Close **X** the **Show Policy** window.

16. In the navigation pane on the left, click **Groups**.

17. Click the **S3-Support** group.

The S3-Support group has the **AmazonS3ReadOnlyAccess** policy attached.

18. Below the **Actions** menu, click the **Show Policy** link.

This policy has permissions to Get and List resources in Amazon S3.

19. Close **X** the **Show Policy** window.

20. In the navigation pane on the left, click **Groups**.

21. Click the **EC2-Admin** group.

This Group is slightly different from the other two. Instead of a *Managed Policy*, it has an **Inline Policy**, which is a policy assigned to just one User or Group. Inline Policies are typically used to apply permissions for one-off situations.

22. Under **Actions**, click **Show Policy** to view the policy.

This policy grants permission to view (Describe) information about Amazon EC2 and also the ability to Start and Stop instances.

23. At the bottom of the screen, click **Cancel** to close the policy.

## Business Scenario

For the remainder of this lab, you will work with these Users and Groups to enable permissions supporting the following business scenario:

Your company is growing its use of Amazon Web Services, and is using many Amazon EC2 instances and a great deal of Amazon S3 storage. You wish to give access to new staff depending upon their job function:

User	In Group	Permissions
user-1	S3-Support	Read-Only access to Amazon S3
user-2	EC2-Support	Read-Only access to Amazon EC2
user-3	EC2-Admin	View, Start and Stop Amazon EC2 instances

## Task 2: Add Users to Groups

You have recently hired **user-1** into a role where they will provide support for Amazon S3. You will add them to the **S3-Support** group so that they inherit the necessary permissions via the attached *AmazonS3ReadOnlyAccess* policy.

💡 You can ignore any "not authorized" errors that appear during this task. They are caused by your lab account having limited permissions and will not impact your ability to complete the lab.

### Add user-1 to the S3-Support Group

24. In the left navigation pane, click **Groups**.

25. Click the **S3-Support** group.

26. Click the **Users** tab.

27. In the **Users** tab, click **Add Users to Group**.

28. In the **Add Users to Group** window, configure the following:

- Select ☒ **user-1**.
- At the bottom of the screen, click **Add Users**.

In the **Users** tab you will see that user-1 has been added to the group.

### Add user-2 to the EC2-Support Group

You have hired **user-2** into a role where they will provide support for Amazon EC2.

29. Using similar steps to the ones above, add **user-2** to the **EC2-Support** group.

user-2 should now be part of the **EC2-Support** group.

### Add user-3 to the EC2-Admin Group

You have hired **user-3** as your Amazon EC2 administrator, who manage your EC2 instances.

30. Using similar steps to the ones above, add **user-3** to the **EC2-Admin** group.

user-3 should now be part of the **EC2-Admin** group.

31. In the navigation pane on the left, click **Groups**.

Each Group should have a **1** in the Users column for the number of Users in each Group.

If you do not have a **1** beside each group, revisit the above instructions above to ensure that each user is assigned to a Group, as shown in the table in the Business Scenario section.

## Task 3: Sign-In and Test Users

In this task, you will test the permissions of each IAM User.

32. In the navigation pane on the left, click **Dashboard**.

An **IAM users sign-in link** is displayed. It will look similar to:

*<https://123456789012.signin.aws.amazon.com/console>*

This link can be used to sign-in to the AWS Account you are currently using.

33. Copy the **IAM users sign-in link** to a text editor.

34. Open a private window.

### Mozilla Firefox

- Click the menu bars ☰ at the top-right of the screen
- Select **New Private Window**

### Google Chrome

- Click the ellipsis ⋮ at the top-right of the screen
- Click **New incognito window**

### Microsoft Edge

- Click the ellipsis ⋮ at the top-right of the screen
- Click **New InPrivate window**

### Microsoft Internet Explorer

- Click the **Tools** menu option
- Click **InPrivate Browsing**

35. Paste the **IAM users sign-in link** into your private window and press **Enter**.

You will now sign-in as **user-1**, who has been hired as your Amazon S3 storage support staff.

36. Sign-in with:

- **IAM user name:** `user-1`
- **Password:** `lab-password`

37. In the **Services** menu, click **S3**.

38. Click the name of one of your buckets and browse the contents.

Since your user is part of the **S3-Support** Group in IAM, they have permission to view a list of Amazon S3 buckets and their contents.

Now, test whether they have access to Amazon EC2.

39. In the **Services** menu, click **EC2**.

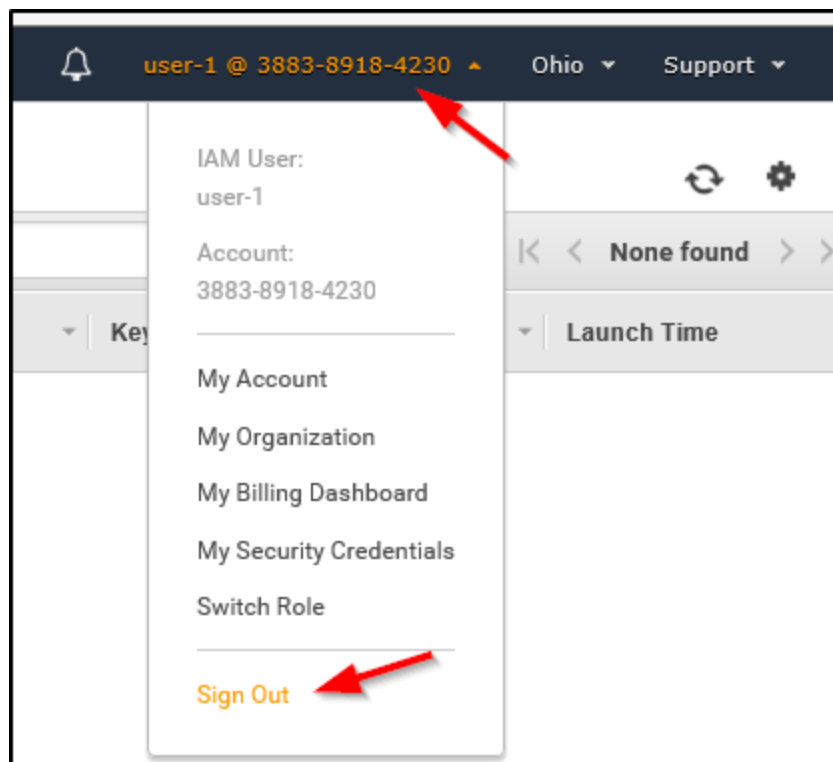
40. In the left navigation pane, click **Instances**.

You cannot see any instances! Instead, it says *An error occurred fetching instance data: You are not authorized to perform this operation..* This is because your user has not been assigned any permissions to use Amazon EC2.

You will now sign-in as **user-2**, who has been hired as your Amazon EC2 support person.

41. Sign user-1 out of the **AWS Management Console** by configuring the following:

- At the top of the screen, click **user-1**
- Click **Sign Out**



42. Paste the **IAM users sign-in** link into your private window and press **Enter**.

This links should be in your text editor.

43. Sign-in with:

- **IAM user name:** `user-2`

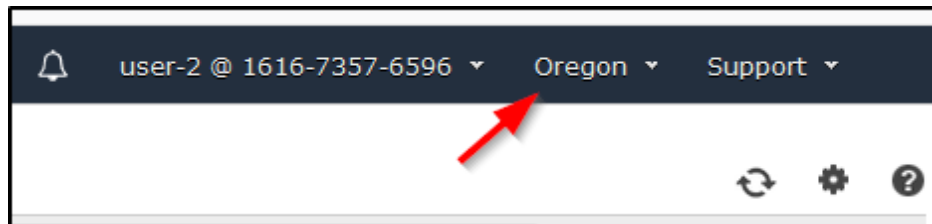
- **Password:** lab-password

44. In the **Services** menu, click **EC2**.

45. In the navigation pane on the left, click **Instances**.

You are now able to see an Amazon EC2 instance because you have Read Only permissions. However, you will not be able to make any changes to Amazon EC2 resources.

⚠ If you cannot see an Amazon EC2 instance, then your Region may be incorrect. In the top-right of the screen, pull-down the Region menu and select the region that you noted at the start of the lab (eg **Oregon**).



Your EC2 instance should be selected ☒. If it is not selected, select ☒ it.

46. In the **Actions** menu, click **Instance State** > **Stop**.

47. In the **Stop Instances** window, click **Yes, Stop**.



You will receive an error stating *You are not authorized to perform this operation*. This demonstrates that the policy only allows you to information, without making changes.

48. At the **Stop Instances** window, click **Cancel**.

Next, check if user-2 can access Amazon S3.

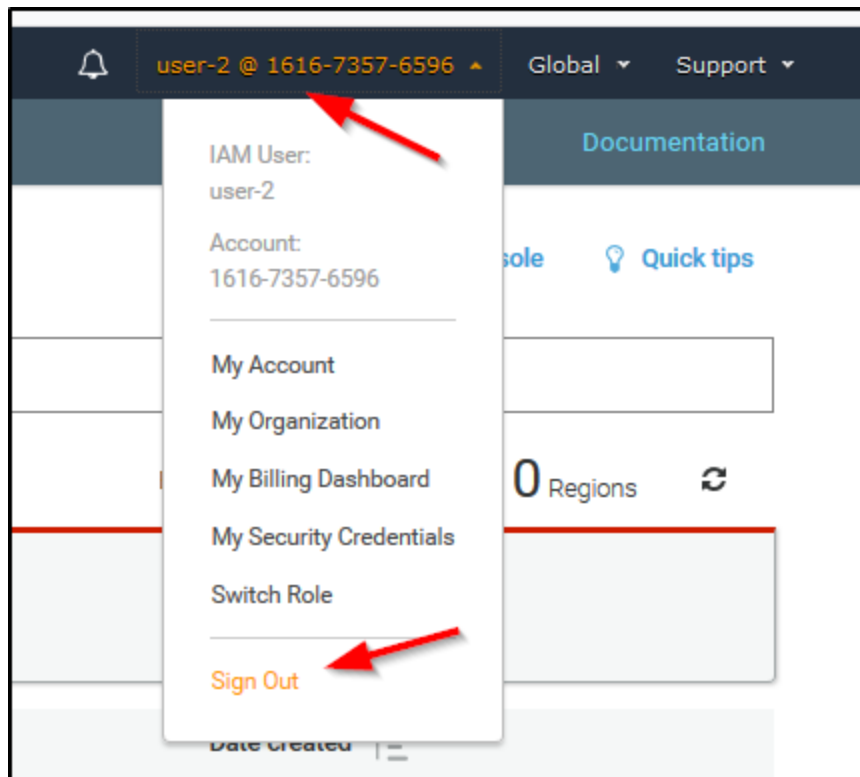
49. In the **Services**, click **S3**.

You will receive an ! **Error Access Denied** because user-2 does not permission to use Amazon S3.

You will now sign-in as **user-3**, who has been hired as your Amazon EC2 administrator.

50. Sign user-2 out of the **AWS Management Console** by configuring the following:

- At the top of the screen, click **user-2**
- Click **Sign Out**



51. Paste the **IAM users sign-in** link into your private window and press **Enter**.
  52. Paste the sign-in link into your web browser address bar again. If it is not in your clipboard, retrieve it from the text editor where you stored it earlier.
  53. Sign-in with:
    - **IAM user name:** user-3
    - **Password:** lab-password
  54. In the **Services** menu, click **EC2**.
  55. In the navigation pane on the left, click **Instances**.
- As an EC2 Administrator, you should now have permissions to Stop the Amazon EC2 instance.
- Your EC2 instance should be selected ☒. If it is not, please select ☒ it.
- ⚠ If you cannot see an Amazon EC2 instance, then your Region may be incorrect. In the top-right of the screen, pull-down the Region menu and select the region that you noted at the start of the lab (eg **Oregon**).
56. In the **Actions** menu, click **Instance State > Stop**.

57. In the **Stop Instances** window, click **Yes, Stop**.

The instance will enter the *stopping* state and will shutdown.

58. Close your private window.

## Lab Complete

🚩 Congratulations! You have completed the lab.

Click **End Lab** at the top of this page to clean up your lab environment.

## Conclusion

👏 Congratulations! You now have successfully:

- Explored pre-created IAM users and groups
- Inspected IAM policies as applied to the pre-created groups
- Followed a real-world scenario, adding users to groups with specific capabilities enabled
- Located and used the IAM sign-in URL
- Experimented with the effects of policies on service access

## Additional Resources

- [AWS Training and Certification](#)
- [For more information about AWS IAM](#)

For feedback, suggestions, or corrections, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com)



# ACF Lab 7: AWS Playground

## Lab Overview

---

This is a sandbox environment for ad-hoc exploration of AWS services.

You will be restricted to the following :

- Amazon EC2
  - Instances - t2 and t3 instances of type t\*.nano to t\*.medium.
  - EBS volumes - size up to 35 GB and type General Purpose SSD (gp2).
  - On-Demand instances.
  - AMIs - Amazon provided Linux and Windows AMIs.
- Amazon RDS
  - DB Instance Class Types - db.t2 and db.t3 instances of type db.t\*.micro to db.t\*.medium.
  - EBS volumes - size up to 100 GB and type General Purpose SSD (gp2).
  - On-Demand DB instance class types.
  - DatabaseEngine - Amazon Aurora, MySQL, PostgreSQL and MariaDB.
- AWS IAM
  - Read access.
- Amazon EC2 Autoscaling
- AWS CloudFormation
- Amazon CloudWatch
- Amazon SNS
- Amazon S3
- Amazon Glacier
- Elastic Load Balancing
- Amazon Route53
- AWS Lambda
- Amazon CloudFront

- Amazon DynamoDB
- AWS CloudTrail
- AWS KMS
  - List access
- Tagging

Please see your tutor for further instructions.