



Academy Cloud Foundations (ACF)
Lab Guide
Version 1.0.5

100-ACFNDS-10-EN-LG

© 2018 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.

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	24
ACF Lab 3: Build your VPC and Launch a Web Server	40
ACF Lab 4: Build your DB Server and Interact with your DB Using an App	52
ACF Lab 5: Scale and Load Balance your Architecture	67
ACF Lab 6: Introduction to AWS IAM	83

ACF Lab 1: Introduction to Amazon EC2

Lab Overview

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

What is Amazon Elastic Compute Cloud (Amazon EC2)?

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.

Amazon EC2 Features

Amazon EC2 provides a number of power features for building scalable, failure resistant, enterprise class applications:

- **Bare Metal Instances:** Provides your applications with direct access to the processor and memory of the underlying server.
- **Optimized Compute Performance and Cost:** Provision compute capacity across EC2 instance type, Availability Zones, and purchase models to optimize scale, performance and cost.
- **GPU Compute Instances:** Massive floating point processing power benefit from next-generation general-purpose GPU compute instances.
- **GPU Graphics:** High graphics quality requirements benefit from GPU graphics instances.
- **High I/O Instances:** Requirements for very high, low latency, random I/O access to data benefit from High I/O instances.
- **Dense Storage Instances:** Very high density per instance and high sequential I/O for data intensive applications benefit from Dense Storage instances.

- **Optimized CPU Configurations:** Greater control of Amazon EC2 instances with the ability to specify a custom number of vCPUs and ability to disable Intel Hyper-Threading Technology.
- **Flexible Storage Options:** Storage to suit workload requirements.
- **Automatic Scaling:** Automatically scale Amazon EC2 capacity up or down according to defined conditions.
- **High Performance Computing Clusters:** Engineered to provide high-performance network capability.
- **Enhanced Networking:** Engineered to provide significantly higher packet per second (PPS) performance, lower network jitter and lower latencies.

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](#).

Technical knowledge prerequisites

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.

Lab Objectives

After completing 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

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.

Duration

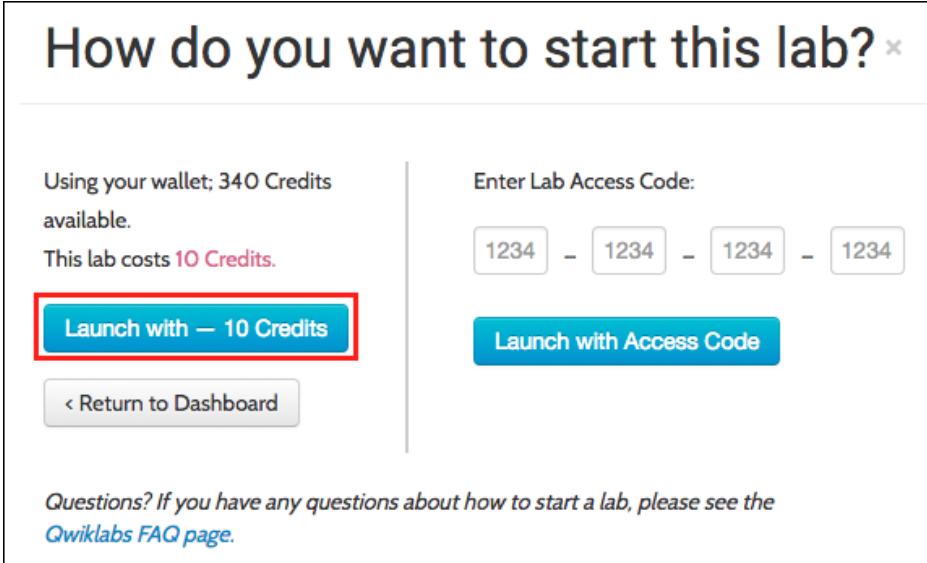
This lab takes approximately **45 minutes** to complete.

Accessing the AWS Management Console

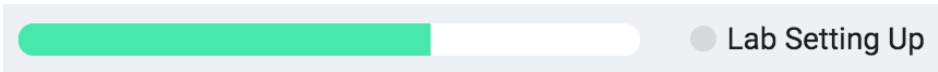
1. At the top of these instructions, click **START LAB** to launch your lab.

A green rectangular button with the text "START LAB" in white, bold, uppercase letters.

2. When asked *How do you want to start this lab*, click **Launch with credits**.

A dialog box titled "How do you want to start this lab?" with a close button (x). It has two main sections. The left section shows "Using your wallet; 340 Credits available." and "This lab costs 10 Credits." Below this is a blue button labeled "Launch with — 10 Credits" which is highlighted with a red rectangle. Below that is a grey button labeled "< Return to Dashboard". The right section is titled "Enter Lab Access Code:" and contains four input boxes, each with "1234", separated by minus signs. Below this is a blue button labeled "Launch with Access Code". At the bottom, there is a link: "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.

A horizontal progress bar with a green segment on the left and a white segment on the right. To the right of the bar is a grey circle followed by the text "Lab Setting Up".

3. Click **Open Console**.

A yellow rectangular button with the text "OPEN CONSOLE" in white, bold, uppercase letters.

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**.

Choose an 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 AMI** (at the top of the list).

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.

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

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 a 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 update
yum -y install httpd
chkconfig httpd on
service httpd start
echo "<html><h1>Hello From Your Web Server!</h1></html>" > /var/www/html/index.html
```

The script will:

- Install system updates.
- 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**.

Add Storage

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

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**.

Add Tags

① A tag is a label that you assign to an AWS resource. 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. The Key is similar to the category for the tag and the Value is just that -- the assigned value for the Key.

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

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

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


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**, add the following text into the "**Security Group Name:**" and "**Description:**" boxes, respectively. This will configure the Security Group.

- **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. Delete the existing SSH rule under the grey banner for listed rules that can be created.

18. Delete  the existing SSH rule under the grey banner for listed rules that can be created.

19. Take a moment to review the details associated with the instance you will launch momentarily. You are able to see the EBS storage, the respective Security Groups, and the Tags you created. Click **Review and Launch**.

Review

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, as indicated by the green text.

27. Click the **Monitoring** tab.

This tab displays CloudWatch metrics for your instance. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications and services that run on AWS, and on-premises servers. Currently, there are not many metrics to display because the instance was recently launched.

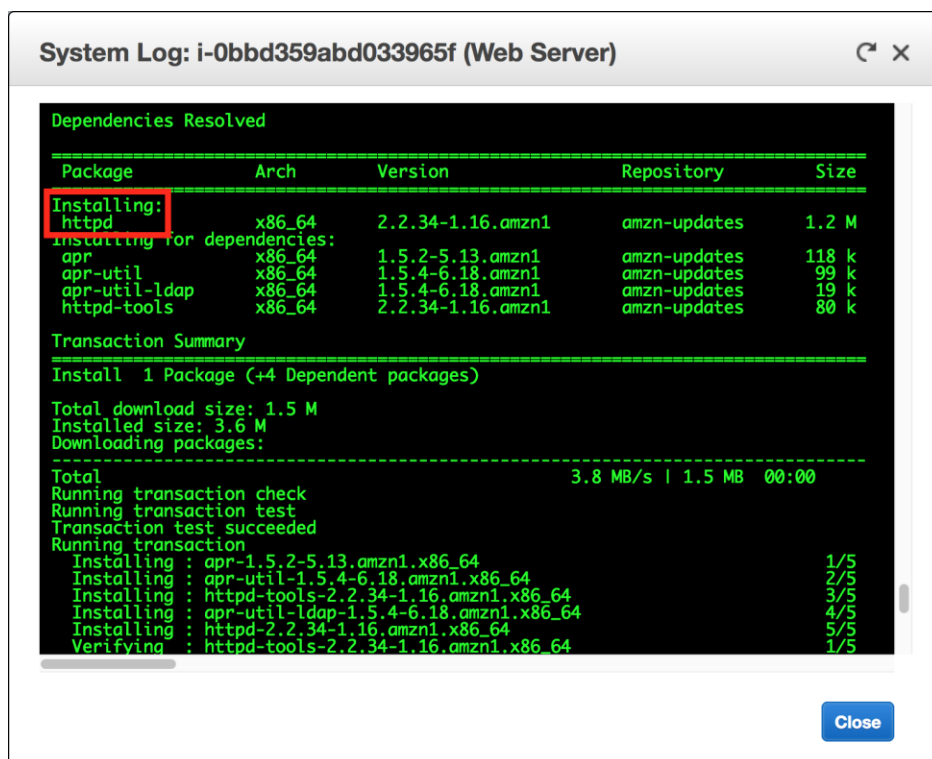
You can click on a graph to see an expanded view.

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 located at the top of the screen, 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.

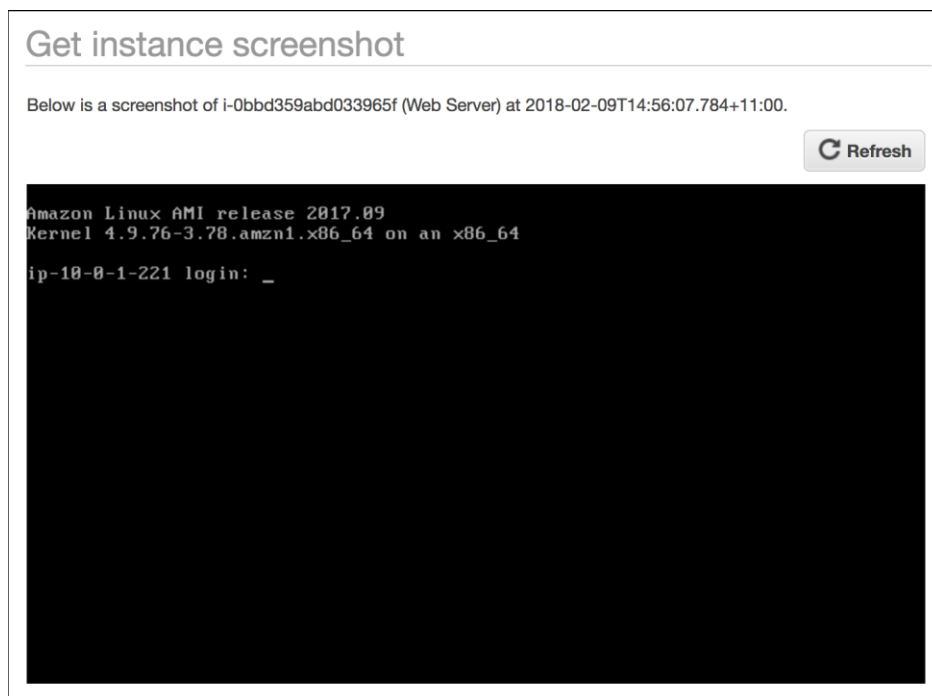
29. Scroll through the output and note that the HTTP package was installed from the **user data** that you added when you created the instance.



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.



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, located at the bottom of the screen at the top of the right hand column.
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 by using the following drop down menus.

- **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 as indicated by the red icon on the screen, which may be yellow as it prepares to stop.

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

Change The Instance Type

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

47. For **Instance Type**, select **t2.small**, using the drop down menu.

48. 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.

Resize the EBS Volume

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

50. 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.

51. Change the size to: 10

52. Click **Modify**.

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

54. Click **Close**.

Start the Resized Instance

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

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

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

57. 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.

58. 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 could request an increase for many of these limits; however, we will not be requesting an increase. This is for informational purposes.

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. Remember, it is important to terminate all instances not in use so no additional charges are incurred for that instance.

In Task 1, we initiated the EC2 instance with termination protection enabled. In this task, you will learn how to use *termination protection*.

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

60. 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 note, that 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.

61. Click **Cancel**.

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

63. Click **Yes, Disable**.

You can now terminate the instance.

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

65. Click **Yes, Terminate**.

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

Conclusion

Congratulations! You now have learned how to:

- Launch your Amazon EC2 instance
- Monitor your instance
- Update your security group and access content from a web server
- Resize your instance type and EBS volume
- Test Termination Protection

Lab Complete

You have successfully completed the lab. To clean up your lab environment, do the following:

66. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
67. Return to the **qwikLABS** page where you launched your lab and click **END LAB**.

Lab Feedback

For feedback, suggestions, or corrections, please email us at *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

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.

What is Amazon Elastic Block Store (Amazon EBS)?

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.

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.

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](#).

The Amazon EBS Volume Lifecycle

You can perform many actions on an EBS volume:

Action	Description
Create	New Amazon EBS volumes are created out of a vast amount of available space. They can have a size of 1 GB to 16 TB.
Attach	An Amazon EBS volume can be attached to an instance. After attachment, it becomes visible to the operating system as a regular block device, just like a hard drive. Each Amazon EBS volume can only be attached to a single instance at a time.
Attached and In Use	The operating system can now format and set up a file system on the Amazon EBS volume and use it as a regular storage device.
Create Snapshot	Snapshots can be created at any time while the volume is in-use.
Detach	When the operating system no longer uses the volume, it can be detached from the instance. Data remains stored on the Amazon EBS volume, and the volume remains available for attachment to any other instance within the same Availability Zone.
Delete	When the volume and its contents are no longer needed, the Amazon EBS volume can be deleted.

Amazon EBS, AMIs, and Termination

You can boot Amazon EC2 instances from Amazon EBS volumes. Such instances are created from Amazon Machine Images (AMIs) that reference Amazon EBS snapshots containing the initial operating system image to boot from. When you create such an instance, a new Amazon EBS volume for the root file system is created from a volume snapshot containing the operating system image.

When an instance is terminated, the default behavior for most instances is to simply delete the Amazon EBS volume because it can be recreated at any time out of the underlying AMI.

If you want to retain the Amazon EBS volume's contents after terminating an instance (for example to keep some customized configuration), you can specify the *DeleteOnTermination = False* flag for the instance. In this case, the Amazon EBS volume will only be detached from the instance when the instance is terminated.

Technical knowledge prerequisites

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

Lab Objectives

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

- Create an Amazon EBS volume
- Attach the volume to an instance
- Configure the instance to use the virtual disk
- Create an Amazon EBS snapshot
- Restore the snapshot

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.

Duration

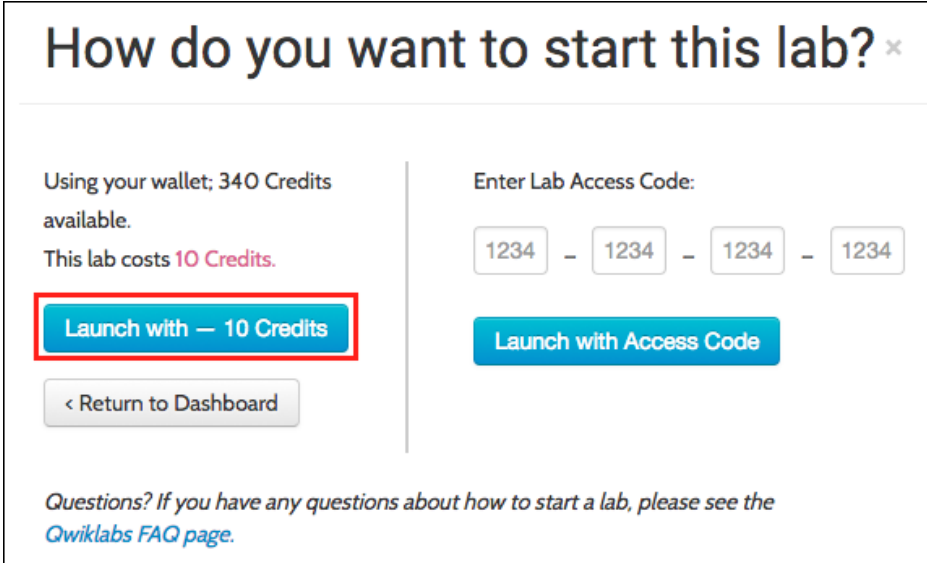
This lab takes approximately **45 minutes** to complete.

Accessing the AWS Management Console

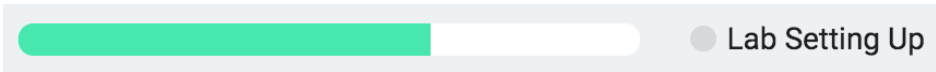
1. At the top of these instructions, click **START LAB** to launch your lab.

A green rectangular button with the text "START LAB" in white, bold, uppercase letters.

2. When asked *How do you want to start this lab*, click **Launch with credits**.

A dialog box titled "How do you want to start this lab?" with a close button (x) in the top right corner. The dialog is split into two columns. The left column contains the text "Using your wallet; 340 Credits available." and "This lab costs 10 Credits." Below this is a blue button labeled "Launch with — 10 Credits" which is highlighted with a red rectangular border. At the bottom of the left column is a grey button labeled "< Return to Dashboard". The right column contains the text "Enter Lab Access Code:" followed by four input boxes, each containing "1234", separated by hyphens. Below this is a blue button labeled "Launch with Access Code". At the bottom of the dialog, there is a line of text: "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.

A horizontal progress bar with a green segment on the left and a white segment on the right. To the right of the bar is a grey circle followed by the text "Lab Setting Up".

3. Click **Open Console**.

A yellow rectangular button with the text "OPEN CONSOLE" in white, bold, uppercase letters.

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* which is located in the blue banner under the Availability Zone header.

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**.

The **Create Volume** page will appear. Point to the information icons ⓘ to obtain information about each field.

10. Enter the following values, leaving other fields at their default values:

- **Volume Type:** General Purpose SSD (GP2)
- **Size (GiB):** 1
- **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.
- **Tags:** Create additional tags. Select Add Tag and enter the following in the respective boxes:
 - **Key:** Name (Case-sensitive, so type it exactly as shown)
 - **Value:** My Volume

11. 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 as indicated by the green and blue icons, respectively under the State column in the banner.

Task 2: Attach the Volume to an Instance

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

12. Select ☒ **My Volume** so that the row is highlighted.

💡 If you cannot see the volume, click the refresh icon.

13. In the **Actions** menu, select **Attach Volume**.

14. 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.

15. Click **Attach**.

The volume state is now *in-use* as indicated by the blue icon turned to green.

Task 3: Login to your Amazon EC2 instance

To perform the next operations on your volume, you will login to the Amazon EC2 instance. Windows users should follow Task 3.1. Mac/Linux users should follow Task 3.2.

Mac/Linux users - [click here for Login instructions](#)

Task 3.1: Windows SSH

This section is for **Windows users only**. If you are running Mac operating system or Linux, please click the "Mac/Linux Users" link above.

In this task, you will download a Keypair and use it to connect to your Amazon EC2 instance using PuTTY.

16. From the qwikLABS page in your browser, in the **Connect** section, click **Download PEM/PPK > Download PPK**. (It is in the section below the Username and Password - scroll down to see it.)

17. Save the file to your **Downloads** folder or any other easy to access location on your local computer. Make note of the location of the saved file for future use.

You will be using PuTTY to connect to the Amazon EC2 instance. If you do not have PuTTY installed on your computer, [download it here](#).

18. Launch **PuTTY** by running the putty.exe file you downloaded. The PuTTY Configuration dialog box that will be used for the next steps will appear.

19. For **Host Name**, enter the public IP address of your EC2 instance, which is shown to the left of the instructions you are currently reading in the Lab's **Connection Details**.

20. In the **Connection** list, expand **SSH**.

21. Click **Auth**.

22. In the **Private key file for authentication** box, browse to the .ppk file that you downloaded earlier, then click **Open**.

23. In the **PuTTY Security Alert** dialog box that opens, click **Yes** to add the key to PuTTY's cache.

24. For **login as:** type `ec2-user` and press **Enter**. You are now logged in to your **Web Server** instance.

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Jun 15 03:49:40 2017 from 205.251.233.178

 _ | _ | _ )
 _ | ( _ | /   Amazon Linux AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/
8 package(s) needed for security, out of 12 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-43-90 ~]$
```

(The actual text you see may differ slightly from the above.)

25. Windows Users: Click [here](#) to skip ahead to the next task.

Task 3.2: Mac/Linux SSH

This section is for **Linux** and **Mac operating system** users only. If you are running **Windows**, click [here](#) to skip ahead to the next task.

In this task, you will download a Keypair and use it to connect to your Amazon EC2 instance.

26. From the qwikLABS page in your browser, in the **Connect** section, click **Download PEM/PPK > Download PEM**. (It is in the section below the Username and Password - scroll down to see it.)

27. Save the file to your computer in a place where you can easily access it.

28. Open the Terminal application on your computer.

29. To connect to your EC2 instance, run the following commands in Terminal (but substituting values as explained below):

```
chmod 400 <path and name of pem file>
```

```
ssh -i <path and name of pem> ec2-user@<Public IP>
```

- For **<path and name of pem file>**, substitute the path/filename to the .pem file you downloaded.
- For **<Public IP>**, enter the public IP address of your EC2 instance, which is shown to the left of the instructions you are currently reading.

Task 4: Create and Configure Your File System

In this task, you will add an 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. Copy and paste this command to view the current 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. Copy and paste this command to 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*.

34. Run this command to add the configuration line:

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

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

```
cat /etc/fstab
```

36. View the available storage again:

```
df -h
```

The output will now contain an additional line, shown here as */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

Compare this to the output generated in step 30.

37. Close your SSH session window.

Task 5: Create an Amazon EBS Snapshot

In this task, you will create a snapshot of your EBS volume. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. Additional information on [snapshots](#) is available on the AWS site.

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. Return to the EC2 Management Console in your web browser.

You should still be on the **Volumes** page. If not, click **Volumes** in the left navigation pane.

39. Select ☒ your volume so that the row is highlighted.

40. In the **Actions** menu, select **Create Snapshot**.

41. In the **Create Snapshot** dialog box, for **Name**, type: `My Snapshot`

42. Click **Create**, then click **Close**.

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

43. 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.

Task 6: Restore the Amazon EBS Snapshot

Creating a snapshot is like backing up your hard drive on Monday morning at 11:00 a.m. Sometimes disaster strikes and on Friday something goes wrong on your laptop and you need to restore your laptop to the state it was on Monday at 11:00 a.m. (when the last snapshot was taken). To retrieve data stored in a snapshot, you can [Restore](#) the snapshot to a new EBS volume.

44. Select ☒ your snapshot so that the row is highlighted.

45. In the **Actions** menu, select **Create Volume**.

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

46. Enter the following values, leaving other fields at their default values:

- **Availability Zone:** Choose a different availability zone this time
- **Tags:** ☒ Create additional tags
- In the Tag Editor, enter:
 - **Key:** Name (Case-sensitive, so type it exactly as shown)
 - **Value:** Restored Volume

47. Click **Create Volume**, then click **Close**.

48. In the left navigation pane, click **Volumes**.

You should now see your new *Restored Volume*.

Conclusion

Congratulations! You now have learned how to:

- Create an Amazon EBS volume
- Attach the volume to an instance
- Configure the instance to use the virtual disk
- Create an Amazon EBS snapshot
- Restore the snapshot

Lab Complete

You have successfully completed the lab. To clean up your lab environment, do the following:

49. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
50. Return to the **qwikLABS** page where you launched your lab and click **END LAB**.

Lab Feedback

For feedback, suggestions, or corrections, please email us at *aws-course-feedback@amazon.com*.

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

Lab Overview

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

What is Amazon Virtual Private Cloud (Amazon VPC)?

Amazon VPC enables you to launch Amazon Web Services (AWS) resources into a virtual network that you define. 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. 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.

An **Internet gateway (IGW)** is a VPC component that allows communication between instances in your VPC and the Internet. 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.

After creating a VPC, you can add one or more subnets in each Availability Zone. 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*.

Amazon VPC Features

Amazon VPC provides a number of power features for build:

- **Secure:** Amazon VPC provides advanced security features such as security groups and network access control lists.
- **Simple:** Amazon VPC's can be created quickly and easily.
- **Scalable and reliable:** All the same scalability and reliability benefits as the rest of the AWS platform.

Technical knowledge prerequisites

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

Lab Objectives

After completing this lab, you will be able to:

- Create a VPC.
- Create subnets.
- Configure a security group.
- Launch an EC2 instance into a VPC.

Other AWS Services

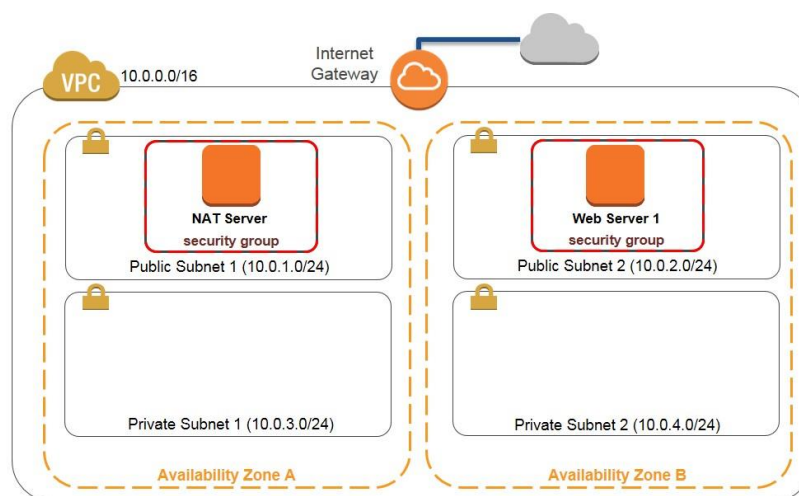
Other AWS Services than the ones needed for this lab are disabled by IAM policy during your access time in this lab. In addition, the capabilities of the services used in this lab are limited to what's required by the lab and in some cases are even further limited as an intentional aspect of the lab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

Duration

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

Scenario

In this lab you build the following infrastructure:

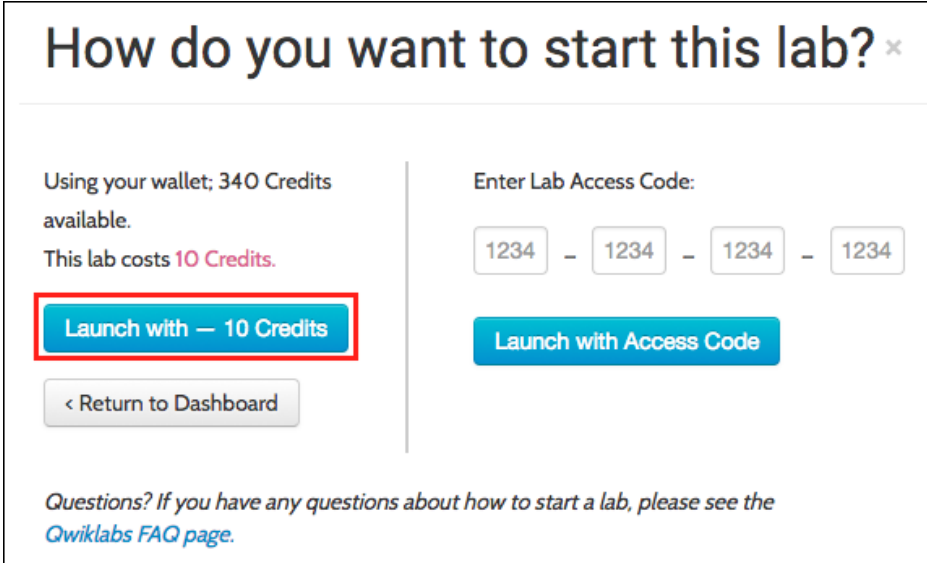


Accessing the AWS Management Console

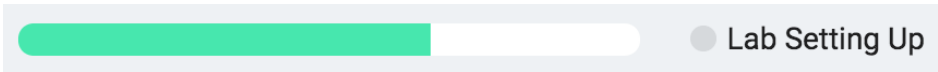
1. At the top of these instructions, click **START LAB** to launch your lab.

A green rectangular button with the text "START LAB" in white capital letters.

2. When asked *How do you want to start this lab*, click **Launch with credits**.

A dialog box titled "How do you want to start this lab?" with a close button (x) in the top right. The dialog is split into two columns. The left column shows "Using your wallet; 340 Credits available." and "This lab costs 10 Credits." Below this is a blue button labeled "Launch with — 10 Credits" which is highlighted with a red rectangle. At the bottom of the left column is a grey button labeled "< Return to Dashboard". The right column is titled "Enter Lab Access Code:" and contains four input boxes, each with "1234", separated by hyphens. Below this is a blue button labeled "Launch with Access Code". At the bottom of the dialog, there is a link: "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.

A horizontal progress bar with a green segment on the left and a white segment on the right. To the right of the bar is a grey circle followed by the text "Lab Setting Up".

3. Click **Open Console**.

A yellow rectangular button with the text "OPEN CONSOLE" in black capital letters.

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

Overview

In this task, you create a VPC with two subnets in one Availability Zone.

5. In the **AWS Management Console**, on the **Services** menu, click **VPC**.
6. Click on **Your VPC**.
7. Click **Create VPC**.
8. In the navigation pane on the left side of the screen, click **VPC with Public and Private Subnets**.
9. Click **Select**.
10. Configure the following settings (and ignore any settings that aren't listed):
 - a. **IPv4 CIDR block**: Type: `10.0.0.0/16`
 - b. **VPC name**: type `My Lab VPC`
 - c. **Public subnet's IPv4 CIDR**: Type `10.0.1.0/24`
You can safely ignore the error:
"Public and private subnet CIDR blocks overlap."
You will fix this when you change the value below.
 - d. **Availability Zone**: Click the *first* Availability Zone.
 - e. **Public subnet name**: type `Public Subnet 1`
 - f. **Private subnet's IPv4 CIDR**: Type `10.0.3.0/24`
 - g. **Availability Zone**: Click the *first* Availability Zone.
The same as used for Public Subnet 1
 - h. **Private subnet name**: type `Private Subnet 1`
 - i. **Specify the details of your NAT gateway**: Click **Use a NAT instance instead**.
On the far right of the screen - you may need to scroll.
 - j. **Key pair name**: Click the **Qwiklabs** key pair.
11. Click **Create VPC**.
12. In the success message, click **OK**.

Task 2: Create Additional Subnets

In this task, you create two additional subnets from the architecture diagram in another Availability Zone and associate the subnets with existing route tables.

13. In the navigation pane, click **Subnets**.
14. Click **Create Subnet**.
15. Now to create the Public subnet, in the **Create Subnet** dialog box, configure the following settings (and ignore any settings that aren't listed):
 - a. **Name tag**: type `Public Subnet 2`
 - b. **VPC**: Click **My Lab VPC**.
 - c. **Availability Zone**: Click the *second* Availability Zone
 - d. **IPv4 CIDR block**: Type `10.0.2.0/24`
16. Click **Yes, Create**.
17. Click **Create Subnet**.
18. Now to create the Private subnet, in the **Create Subnet** dialog box, configure the following settings (and ignore any settings that aren't listed):
 - a. **Name tag**: type `Private Subnet 2`
 - b. **VPC**: Click **My Lab VPC**.
 - c. **Availability Zone**: Select the *second* Availability Zone.
The same as used for Public Subnet 2
 - d. **CIDR block**: Type `10.0.4.0/24`
19. Click **Yes, Create**. You can now see the four subnets you created on the dashboard.
20. In the navigation pane, click **Route Tables**.
21. Select the route table with the VPC **My Lab VPC** and **Yes** under **Main**.
22. Double-click the empty **Name** for this route table, type `Private Route Table`, and click the checkmark to save.
23. In the lower pane, click **Routes** and note that **Destination 0.0.0.0/0** is set to **Target eni-xxxxxxx / i-xxxxxxx**.
This route table is used to route traffic from private subnets to the NAT instance, as identified by an Elastic Network Interface (ENI) and Instance ID.
24. Click **Subnet Associations**, and then click **Edit**.

25. Select **Private Subnet 1** and **Private Subnet 2**.
26. Click **Save**.
27. Select the route table with the VPC **My Lab VPC** and **No** under **Main**.
28. Double-click the empty **Name** for this route table, type `Public Route Table`, and click the checkmark to save.
29. In the lower pane, click **Routes** and note that **Destination 0.0.0.0/0** is set to **Target igw-xxxxxxx**.
This route table is used by public subnets for communication.
30. Click **Subnet Associations**, and then click **Edit**.
31. Select **Public Subnet 1** and **Public Subnet 2**.
32. Click **Save**.

Task 3: Create a VPC Security Group

In this task, you create a VPC security group that permits access for web traffic.

33. In the navigation pane, click **Security Groups**.
34. Click **Create Security Group**.
35. In the **Create Security Group** dialog box, configure the following settings (and ignore any settings that aren't listed):
 - a. **Name tag**: type `WebSecurityGroup`
You can ignore the message:
"A security group description is required."
 - b. **Group name**: Click **WebSecurityGroup**.
This will be entered automatically
 - c. **Description**: type `Enable HTTP access`
 - d. **VPC**: Click **My Lab VPC**.
This is the VPC you created in Task 1
36. Click **Yes, Create**.
37. Select **WebSecurityGroup**.
38. Click the **Inbound Rules** tab.
39. Click **Edit**.
40. For **Type**, click `HTTP (80)`.
41. To allow traffic from any range, click in the **Source** box and type `0.0.0.0/0` (Note: This CIDR notation. To learn more, see the [VPC User's Guide](#)).
42. Click **Save**.

Task 4: Launch Your First Web Server Instance

Overview

In this task, you launch an EC2 instance into the VPC you created and bootstrap the instance to act as a web server.

43. On the **Services** menu, click **EC2**.
44. Click **Launch Instance**.
45. In the row for **Amazon Linux AMI**, click **Select**. If you receive a warning, click **Continue**.
46. On the **Step 2: Choose an Instance Type** page, confirm that **t2.micro** is selected and then click **Next: Configure Instance Details**.
47. On the **Step 3: Configure Instance Details** page, configure the following settings (and ignore any settings that aren't listed):
 - a. **Network**: Click **My Lab VPC**.
This is the VPC you created in Task 1
 - b. **Subnet**: Click the **Public Subnet 2 (10.0.2.0/24)**.
This is the subnet you created in Task 2
 - c. **Auto-assign Public IP**: Click **Enable**.
You can safely ignore the message:
"You do not have permissions to list any IAM roles."
48. Expand the **Advanced Details** section.
49. Click **Copy Code Block** below, and paste it into the **User data** box.

```
#!/bin/bash -ex
yum -y update
yum -y install httpd php mysql php-mysql
chkconfig httpd on
/etc/init.d/httpd start
if [ ! -f /var/www/html/lab2-app.tar.gz ]; then
cd /var/www/html
wget https://us-west-2-aws-training.s3.amazonaws.com/awsu-ilt/AWS-100-ESS/v4.2/lab-2-
configure-website-datastore/scripts/lab2-app.tar.gz
tar xvfz lab2-app.tar.gz
chown apache:root /var/www/html/rds.conf.php
fi
```

The user data transforms the Linux instance into a PHP web application.

50. Click **Next: Add Storage**.
51. Click **Next: Add Tags**.

52. Click **Add Tag**, and configure the following settings (and ignore any settings that aren't listed):
 - a. **Key:** type `Name`
 - b. **Value:** type `Web Server 1`
53. Click **Next: Configure Security Group**.
54. On the **Step 6: Configure Security Group** page, click **Select an existing security group**, and then select the security group you created in Task 3 (**WebSecurityGroup**) by clicking the second radio button.
55. Click **Review and Launch**.
When prompted with a *warning* that you will not be able to connect to the instance through port 22, click **Continue**.
56. Review the instance information and click **Launch**.
Ignore any warning that appears regarding a security group being open to the world. This is expected behavior.
57. Click **Choose an existing key pair**, click the **Qwiklabs** key pair, select the acknowledgment check box, and then click **Launch Instances**.
58. Scroll down and click **View Instances**. You will see two instances – **Web Server 1** and the NAT instance launched by the VPC Wizard. Green indicates everything is good and the yellow indicates a service is in the process of launching.
59. Wait until **Web Server 1** shows *2/2 checks passed* in the **Status Checks** column.
This will take 3 to 5 minutes. Click the refresh icon in the upper right pane to check for updates.
60. Select **Web Server 1** and copy the **Public DNS** value on the **Description** tab.
61. Paste the **Public DNS** value in a new web browser window or tab and press **ENTER**.

You will see a web page displaying the AWS logo and instance meta-data values.

Conclusion

Congratulations! You have learned how to:

- Create a VPC.
- Create subnets and configure a security group.
- Launch an EC2 instance into a VPC.

Lab Complete

You have successfully completed the lab. To clean up your lab environment, do the following:

62. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
63. Return to the **qwikLABS** page where you launched your lab and click **END LAB**.

Lab Feedback

For feedback, suggestions, or corrections, please email us at *aws-course-feedback@amazon.com*.

ACF Lab 4: Build your DB Server and Interact with your DB Using an App

Lab Overview

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

What is Amazon Relational Database Service (Amazon RDS)?

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.

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).

Amazon RDS Features

Some features are specific to different Amazon RDS engines. Some general features:

- **Lower Administrative Burden**
 - **Easy to Use:** Relational database capabilities can be easily accessed via the Amazon RDS Command Line Interface or simple API calls.
 - **Automatic Software Patching:** Database software stays up-to-date with the latest patches.
- **Performance:** Storage that is suitable for a broad range of database workloads.
- **Scalability:** Push button compute scaling.
- **Availability and Durability:** Facilitated by automated backups, database snapshots, multi-AZ deployments, and automatic host replacement.
- **Security:** Facilitated by encryption at rest and in transit, network isolation, and resource-level permissions.

- **Manageability:** Supported by monitoring and metrics provided by CloudWatch, event notifications provided via email, SMS, or Amazon SNS; AWS Config integration to support compliance and enhance security.
- **Cost Effectiveness:** Pay for what you use, save with reserved instances, and easy database stop and start to reduce costs,

Technical knowledge prerequisites

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

Lab Objectives

After completing this lab, you will be able to:

- 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.

Other AWS Services

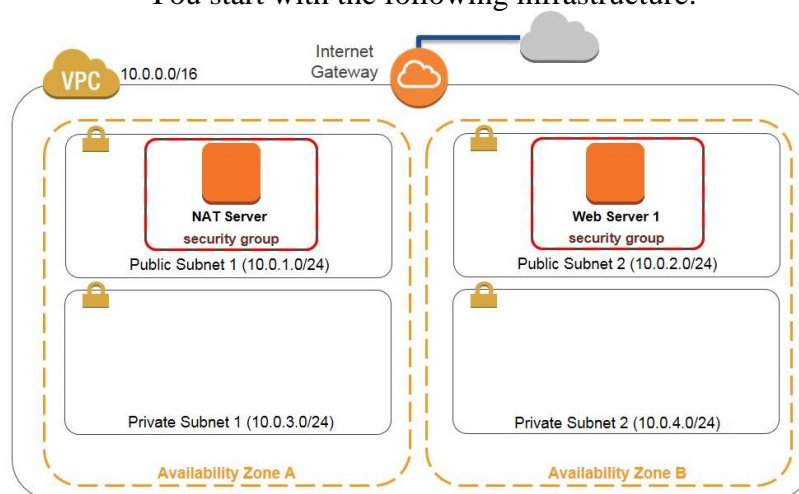
Other AWS Services than the ones needed for this lab are disabled by IAM policy during your access time in this lab. In addition, the capabilities of the services used in this lab are limited to what's required by the lab and in some cases are even further limited as an intentional aspect of the lab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

Duration

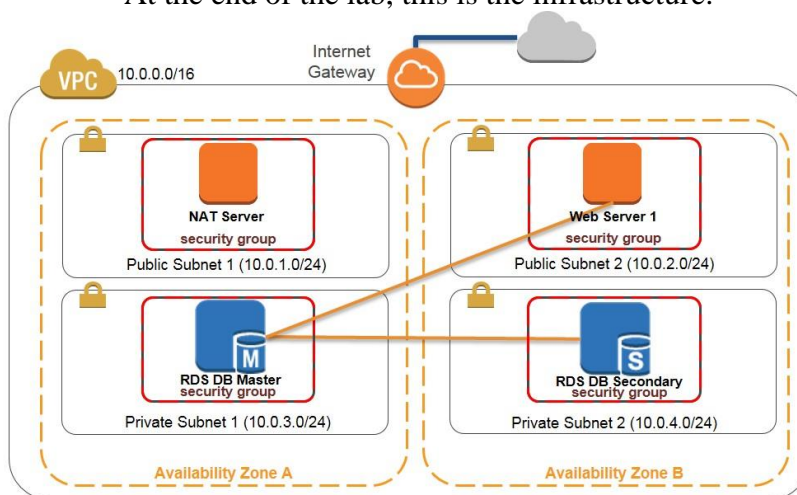
This lab takes approximately **45 minutes** to complete.

Scenario

You start with the following infrastructure:



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

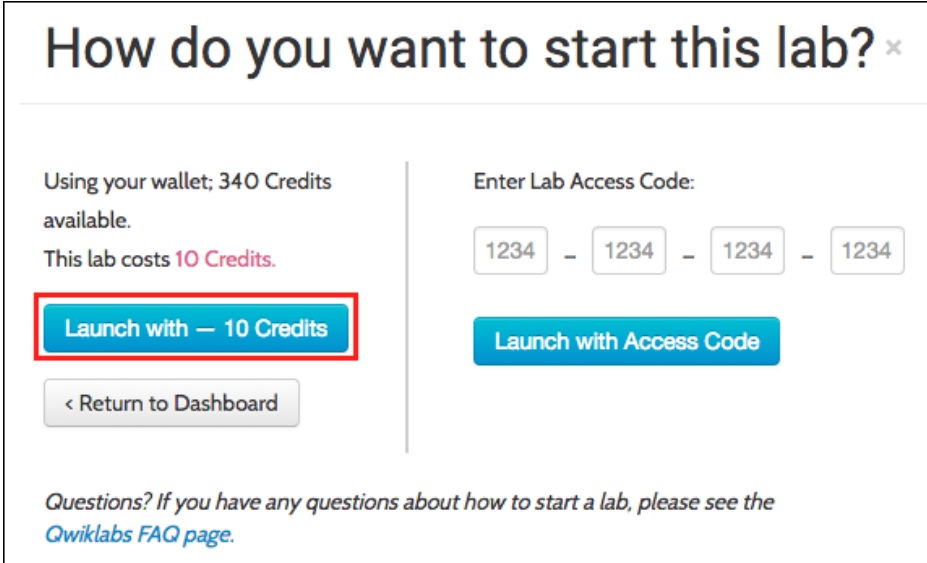


Accessing the AWS Management Console

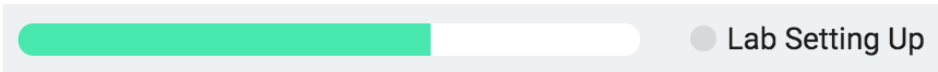
1. At the top of these instructions, click **START LAB** to launch your lab.

A green rectangular button with the text "START LAB" in white capital letters.

2. When asked *How do you want to start this lab*, click **Launch with credits**.

A dialog box titled "How do you want to start this lab?" with a close button (x) in the top right. It has two main sections. The left section shows "Using your wallet; 340 Credits available." and "This lab costs 10 Credits." Below this is a blue button labeled "Launch with — 10 Credits" which is highlighted with a red rectangle. Below that is a grey button labeled "< Return to Dashboard". The right section is titled "Enter Lab Access Code:" and contains four input boxes, each with "1234", separated by minus signs. Below this is a blue button labeled "Launch with Access Code". At the bottom, there is a link: "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.

A horizontal progress bar with a green segment on the left and a white segment on the right. To the right of the bar is a grey circle followed by the text "Lab Setting Up".

3. Click **Open Console**.

A yellow rectangular button with the text "OPEN CONSOLE" in black capital letters.

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 VPC Security Group for the RDS DB Instance

In this task, you create a VPC security group to allow your web server to access your RDS DB instance.

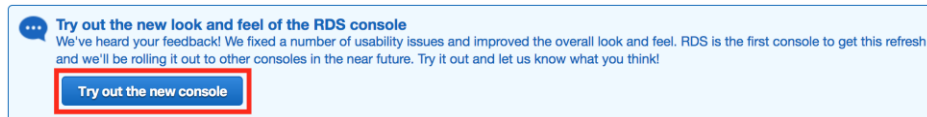
5. In the **AWS Management Console**, on the **Services** menu, click **VPC**.
 6. In the left navigation pane, click **Security Groups**.
 7. Click **Create Security Group**.
 8. In the **Create Security Group** dialog box, configure the following settings (and ignore any settings that aren't listed):
 - **Name tag:** `DB-Security-Group`
 - **Group name:** `DB-Security-Group`
 - **Description:** `DB Instance Security Group`
 - **VPC:** *My Lab VPC*
 9. Click **Yes, Create**.
 10. Select the security group you just created (*DB-Security-Group*) and ensure that all other security groups are not selected.
 11. Click the **Inbound Rules** tab, then click **Edit**.
 12. Configure the following settings (and ignore any settings that aren't listed):
 - **Type:** *MySQL/Aurora (3306)*
 - **Protocol:** *TCP(6)*
 - **Source:** Click in the field and select *Web-Security-Group*
- This is configuring 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**.

Task 2: Create a DB Subnet Group

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

14. On the **Services** menu, click **Relational Database Service**.

⚠ If you see this message in your console, click **Try out the new console** so that these lab instructions match your experience:



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

16. Click **Create DB Subnet Group**.

17. On the **Create DB subnet group** page, configure the following settings (and ignore any settings that aren't listed):

- **Name:** db-subnet-group
- **Description:** DB Instance Subnet Group
- **VPC ID:** *My Lab VPC*

18. For **Availability zone**, select the first Availability Zone.

19. For **Subnet**, select **10.0.3.0/24**.

20. Click **Add subnet**.

You will now add another subnet in a different Availability Zone.

21. For **Availability zone**, select the second Availability Zone.

22. For **Subnet**, click **10.0.4.0/24**.

23. Click **Add subnet**.

24. Click **Create**.

Task 3: Create an RDS DB Instance

In this task, you will configure and launch a MySQL-backed Amazon RDS DB instance.

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

26. Click **Launch DB instance**.

27. Select **MySQL**.

28. Click **Next**.

29. For **Use case**, select **Production - MySQL**.

This will configure a *multi-AZ* database that runs across multiple Availability Zones, making it highly available.

30. Click **Next**.

31. On the **Specify DB details** page, configure the following settings (and ignore any settings that aren't listed):

- **DB instance class:** *db.t2.micro* (The first option in the list)
- **DB instance identifier:** *lab-db*
- **Master username:** *master*
- **Master password:** *lab-password*
- **Confirm password:** *lab-password*

32. Click **Next**.

33. On the **Configure advanced settings** page, configure the following settings (and ignore any settings that aren't listed):

- **VPC:** *My Lab VPC*
- **Subnet group:** *db-subnet-group*
- **VPC security groups:** *Select existing VPC security groups*
- **Select VPC security groups:** *DB-Security-Group (VPC)* and remove *default (VPC)*
- **Database name:** *lab*
- **Backup retention period:** *0 Days*
- **Enhanced monitoring:** *Disable enhanced monitoring*

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

34. Click **Launch DB instance**.

Your database will now be launched.

35. Click **View DB instance details**.

Information about the database will be displayed.

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

① While you are waiting, you might want to review the [Amazon RDS FAQs](#).

36. Wait until the **DB instance Status** is *available* or *modifying*.

💡 You can refresh the web page every couple of minutes to update the status.

37. Scroll down to the **Connect** section and copy the **Endpoint** field.

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

38. 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.

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

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

41. Select ☒ **Web Server 1**.

42. Copy the **IPv4 Public IP** address that appears in the lower pane.

43. Open a new web browser tab, paste the IP address and hit Enter.

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

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

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

45. Configure the following settings:

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

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.

46. 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 successfully configured a relational data store for your website. To clean up your lab environment, do the following:

47. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
48. Return to the **qwikLABS** page where you launched your lab and click **END LAB**.

Conclusion

Congratulations! You have learned how to:

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

Lab Complete

You have successfully completed the lab. To clean up your lab environment, do the following:

49. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
50. Return to the **qwikLABS** page where you launched your lab and click **END LAB**.

Lab Feedback

For feedback, suggestions, or corrections, please email us at *aws-course-feedback@amazon.com*.

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

Lab Overview

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

What is Elastic Load Balancing?

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon Elastic Compute Cloud (Amazon EC2) instances, containers, and IP addresses.

The load balancer serves as a single point of contact for clients, which increases the availability of your application. You can add and remove instance from your load balancer as your needs change. Elastic Load Balancing scales in or out as traffic to your application changes over time. ELB allows you to achieve fault tolerance in your applications by seamlessly providing the required amount of load balancing capacity needed to route application traffic.

Elastic Load Balancing supports three types of load balancers: [Application Load Balancer](#), which routes traffic based on advanced application-level information that includes the content of the request, the [Network Load Balancer](#), which routes traffic to target within Amazon Virtual Private Cloud (VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies, and the [Classic Load Balancer](#), which routes traffic based on either application- or network-level information. The Classic Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instances, and the Application Load Balancer is ideal for applications that need advanced routing capabilities, microservices, and container-based architectures. The Application Load Balancer offers you the ability to route traffic to multiple services or load balance across multiple ports on the same EC2 instance.

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.

Technical knowledge prerequisites

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

Lab Objectives

After completing this lab, you will be able to:

- 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.

Other AWS Services

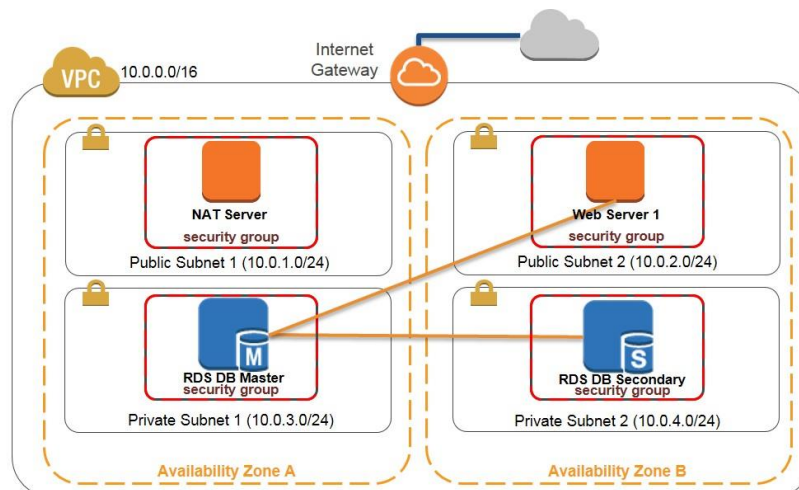
Other AWS Services than the ones needed for this lab are disabled by IAM policy during your access time in this lab. In addition, the capabilities of the services used in this lab are limited to what's required by the lab and in some cases are even further limited as an intentional aspect of the lab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

Duration

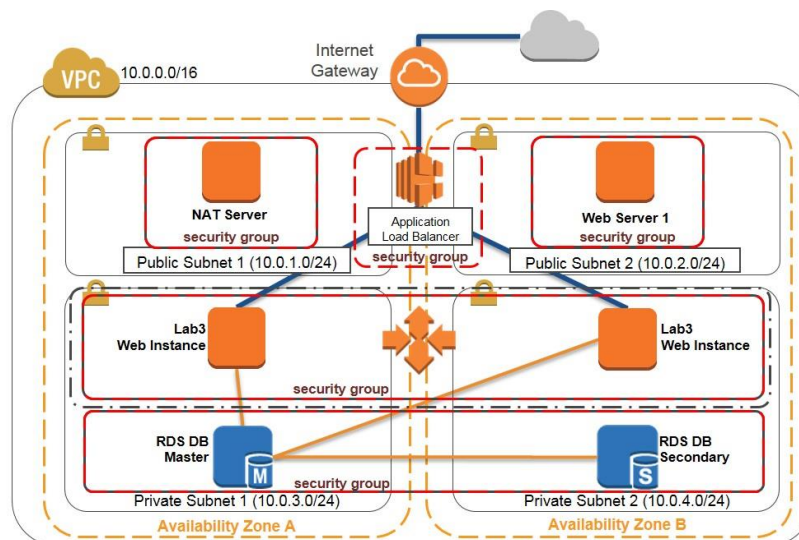
This lab takes approximately **45 minutes** to complete.

Scenario

You start with the following infrastructure:



The final state of the infrastructure is:

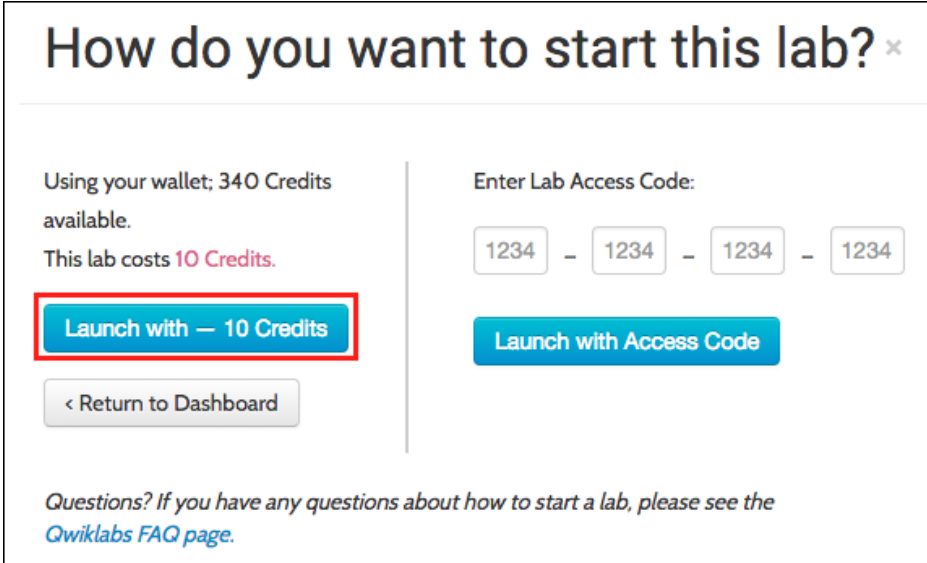


Accessing the AWS Management Console

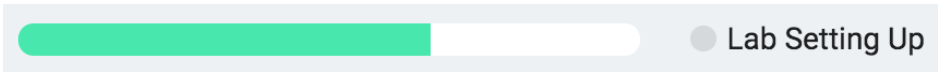
1. At the top of these instructions, click **START LAB** to launch your lab.

A green rectangular button with the text "START LAB" in white, bold, uppercase letters.

2. When asked *How do you want to start this lab*, click **Launch with credits**.

A dialog box titled "How do you want to start this lab?" with a close button (x) in the top right. The dialog is split into two columns. The left column contains the text "Using your wallet; 340 Credits available." and "This lab costs 10 Credits." Below this is a blue button labeled "Launch with — 10 Credits" which is highlighted with a red rectangular border. At the bottom of the left column is a grey button labeled "< Return to Dashboard". The right column contains the text "Enter Lab Access Code:" followed by four input boxes, each containing "1234", separated by hyphens. Below this is a blue button labeled "Launch with Access Code". At the bottom of the dialog, there is a link: "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.

A horizontal progress bar with a green segment on the left and a white segment on the right. To the right of the bar is a grey circle followed by the text "Lab Setting Up".

3. Click **Open Console**.

A yellow rectangular button with the text "OPEN CONSOLE" in white, bold, uppercase letters.

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

Overview

In this task, you create an AMI as the starting point for launching new instances to use with Auto Scaling.

5. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
6. In the navigation pane, click **Instances**.
7. Verify that the **Status Checks** for **Web Server 1** displays *2/2 checks passed*. If it doesn't, wait until it does before proceeding to the next step. Use the refresh icon in the upper right corner to check for updates.
8. Right-click on **Web Server 1**, and then click **Image > Create Image**.
9. Configure the following settings (and ignore any settings that are not listed):
 - a. **Image name:** `Web Server AMI`
 - b. **Image description:** `AMI for Web Server`
10. Click **Create Image**.

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

Task 2: Create a Load Balancer

Overview

In this task, you create a load balancer to balance traffic across several EC2 instances in two Availability Zones.

11. In the navigation pane, click **Load Balancers**.
 12. Click **Create Load Balancer**.
 13. Below **Application Load Balancer**, click **Create**.
 14. Configure the following settings (and ignore any settings that are not listed):
 - a. **Name**: type `Lab-ELB`
 - b. **VPC**: Click **My Lab VPC**.
 - c. **Availability Zones**: Select both to see the available subnets.
Then, select **Public Subnet 1** and **Public Subnet 2**
 15. Click **Next: Configure Security Settings**.
 16. Ignore the following warning: *"Improve your load balancer's security. Your load balancer is not using any secure listener"* and click **Next: Configure Security Groups**.
 17. Select the security group that contains **WebSecurityGroup** in the **Name** and a **Description** of **Enable HTTP access** and clear the **default** check box (indicating the default Security Group).
 18. Click **Next: Configure Routing**.
 19. Under **Target group**, for **Name**, type: `Lab-Group`
 20. Expand **Advanced health check settings**, and configure the following settings (and ignore any settings that aren't listed):
 - a. **Healthy threshold**: type `2`
 - b. **Unhealthy threshold**: type `3`
 - c. **Timeout**: type `10`
 21. Click **Next: Register Targets**.
- Auto Scaling will automatically add instances later. Click **Next: Review**.
22. Review the configuration of your load balancer and click **Create**.
 23. On the "Successfully created load balancer" message, click **Close**.

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

Overview

In this task, you 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, one or more security groups and a block device mapping. An Auto Scaling group contains a collection of EC2 instances that share similar characteristics and are treated as a logical grouping for the purposes of instance scaling and management.

24. In the navigation pane, click **Launch Configurations**.
25. Click **Create Auto Scaling group**.
26. Click **Create launch configuration**.
27. In the navigation pane under Auto Scaling, click **My AMIs**.
28. In the row for **Web Server AMI**, click **Select**.
29. Accept the **t2.micro** selection and click **Next: Configure details**.
30. Configure the following settings (and ignore any settings that aren't listed):
 - a. **Name:** type `Lab-Config`
 - b. **Monitoring:** Click **Enable CloudWatch detailed monitoring**.
31. Click **Next: Add Storage**.
32. Click **Next: Configure Security Group**.
33. Click **Select an existing security group**, and select the security group that contains **WebSecurityGroup** in the **Name** and a **Description** of **Enable HTTP access**.
34. Click **Review**.
35. Review the details of your launch configuration and click **Create launch configuration**. Ignore the "Improve security..." warning; this is expected.
36. Click **Choose an existing key pair**, select the **Qwiklabs** key pair, select the acknowledgement check box, and click **Create launch configuration**.
37. Configure the following settings (and ignore any settings that are not listed):
 - a. **Group name:** type `Lab Scaling Group`
 - b. **Group size Start with:** type `2 (instances)`

- c. **Network:** Click **My Lab VPC**.
Ignore the message regarding "no public IP"; this is expected.
 - d. **Subnet:** Click **Private Subnet 1 (10.0.3.0/24)**, and
Click **Private Subnet 2 (10.0.4.0/24)**.
38. Expand **Advanced Details**, configure the following settings (and ignore any settings that are not listed):
- a. **Load Balancing:** Click **Receive traffic from one or more load balancers**.
 - b. **Target Groups:** Click **Lab-Group**.
 - c. **Health Check Type:** Click **ELB**.
 - d. **Monitoring:** Click **Enable CloudWatch detailed monitoring**.
39. Click **Next: Configure scaling policies**.
40. Select **Use scaling policies to adjust the capacity of this group**.
41. Modify the **Scale between** text boxes to scale between **2** and **6** instances.
42. Click **Scale the Auto Scaling group using step or simple scaling policies**.
43. In **Increase Group Size** section of the dialog box, for **Execute policy when**, click **Add new alarm**.
44. Click to uncheck the checkbox to clear **Send a Notification to..**
45. Configure the following settings (and ignore any settings that aren't listed):
- a. **Whenever:** Click **Average**, and then click **CPU Utilization**.
 - b. **Is:** Click **>=**, and then type **65** (indicating percent).
 - c. **For at least:** type **1**, and then click **1 Minute**.
 - d. **Name of alarm:** Replace exiting entry with: **High CPU Utilization**
46. Click **Create Alarm**.
47. In **Increase Group Size**, configure the following settings (and ignore any settings that aren't listed):
- a. **Take the action:** type **1**, click **instances**, and then type **65**
 - b. **Instances need:** type **60**
(seconds to warm up after each step)
48. In **Decrease Group Size**, for **Execute policy when**, click **Add new alarm**.

49. Click to uncheck the checkbox to clear **Send a notification to**.
50. Configure the following settings (and ignore any settings that aren't listed):
 - a. **Whenever:** Click **Average**, and then click **CPU Utilization**.
 - b. **Is:** Click **<=**, and then type 20
 - c. **For at least:** type 1, and then click **1 Minute**.
 - d. **Name of alarm:** Replace exiting entry with: `Low CPU Utilization`
51. Click **Create Alarm**.
52. In **Decrease Group Size**, for **Take the action:** click **Remove**, type 1, click **instances**, and then type 20
53. Click **Next: Configure Notifications**.
54. Click **Next: Configure Tags**.
55. Configure the following settings (and ignore any settings that aren't listed):
 - a. **Key:** type `Name`
 - b. **Value:** type `Web Instance`
56. Click **Review**.
57. Review the details of your Auto Scaling group, and then click **Create Auto Scaling group**.
58. Click **Close** when your Auto Scaling group has been created.

Task 4: Verify Auto Scaling is Working

Overview

In this task, you verify that Auto Scaling is working correctly.

59. In the navigation pane, click **Instances**.

Four instances are displayed: **Web Server 1**, **NAT Server**, and two new instances labeled as **Web Instance**.

Note: The new instances should appear as running after a few minutes.

60. In the navigation pane, click **Target Groups**.

61. Select **Lab-Group**, and click the **Targets** tab.

Two **Web Instance** instances should be listed for this target group.

62. Wait until the **Status** of both instances transitions to *healthy*. Use the refresh icon in the upper right corner to check for updates.

63. In the navigation pane, click **Load Balancers**.

64. Select **Lab-ELB** and on the **Description** tab in the lower pane, copy the **DNS name** of your load balancer, making sure to omit "(A Record)".

Task 5: Test Auto Scaling

Overview

You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. You created Auto Scaling policies to increase and decrease the group by one instance. You created Amazon CloudWatch alarms to trigger these policies when the aggregate average CPU of the group is $\geq 65\%$ and $\leq 20\%$ respectively. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now monitor this infrastructure using the CloudWatch alarms that you created.

In this task you test the Auto Scaling configuration you implemented.

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

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

The two alarms **High CPU Utilization** and **Low CPU Utilization** are displayed that you created in the last task within this lab. **Low CPU Utilization** has a **State** of *ALARM* and **High CPU Utilization** has a **State** of *OK*. This is because the current group CPU Utilization is $< 20\%$. Auto Scaling is not removing any instances because the group size is currently at its minimum (2).

67. Paste the load balancer's DNS name that you copied earlier into a new browser window or tab and press *ENTER*.

68. Click **Load Test** next to the AWS logo. The application load tests your instances and auto-refreshes in 5 seconds. The Current CPU Load jump to 100%. The **Load Test** link triggers a simple background process. Do not close this tab.

69. Return to the window or tab with the **AWS CloudWatch console**.

In less than 5 minutes, the **Low CPU** alarm status changes to *OK* and the **High CPU** alarm status changes to *ALARM*. Click the refresh icon to see the changes.

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

71. In the navigation pane, click **Instances**.

More than two instances labeled **Web Instance** are now running. They may be in creation, and the tags may not appear immediately. The new instance was created by Auto Scaling based on the CloudWatch Alarm you created in an earlier step.

Task 6 (Optional): Terminate Web Server 1

Overview

In this task, you terminate Web Server 1 in Public Subnet 2. Your Auto Scaling group launched instances into private subnets, and the original publicly accessible web server is no longer needed.

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

73. In the navigation pane, click **Instances**.

74. Right-click **Web Server 1**, and click **Instance State > Terminate**.

75. Click **Yes, Terminate**.

Lab Complete

Congratulations! You have successfully managed your architecture using Auto Scaling and Elastic Load Balancing. To clean up your lab environment, do the following:

76. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
77. Return to the **qwikLABS** page where you launched your lab and click **END LAB**.

Conclusion

Congratulations! You have learned how to:

- 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.

Lab Complete

You have successfully completed the lab. To clean up your lab environment, do the following:

78. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
79. Return to the **qwikLABS** page where you launched your lab and click **END LAB**.

Lab Feedback

For feedback, suggestions, or corrections, please email us at *aws-course-feedback@amazon.com*.

80. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.

81. Return to the **Qwiklabs** page where you launched your lab from and click **END LAB**.

ACF Lab 6: Introduction to AWS IAM

Lab Overview

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.

What is 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.

AWS IAM Features

AWS IAM assists in creating roles and permissions. It allows you to:

- **Manage IAM users and their access:** Create users and assign them individual security credentials.
- **Manage IAM roles and permissions:** Create IAM roles in IAM and manage permissions.
- **Performance:** Storage that is suitable for a broad range of database workloads.

Technical knowledge prerequisites

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

Lab Objectives

After completing this lab, you will be able to:

- Explore existing (pre-created) **IAM Users and Groups**
- Inspect **IAM policies** as applied to the pre-created Groups
- Follow a **real-world scenario** to add Users to Groups with specific capabilities enabled
- Locate and use the **IAM sign-in URL**
- **Experiment** with the effects of policies on service access

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.

Duration

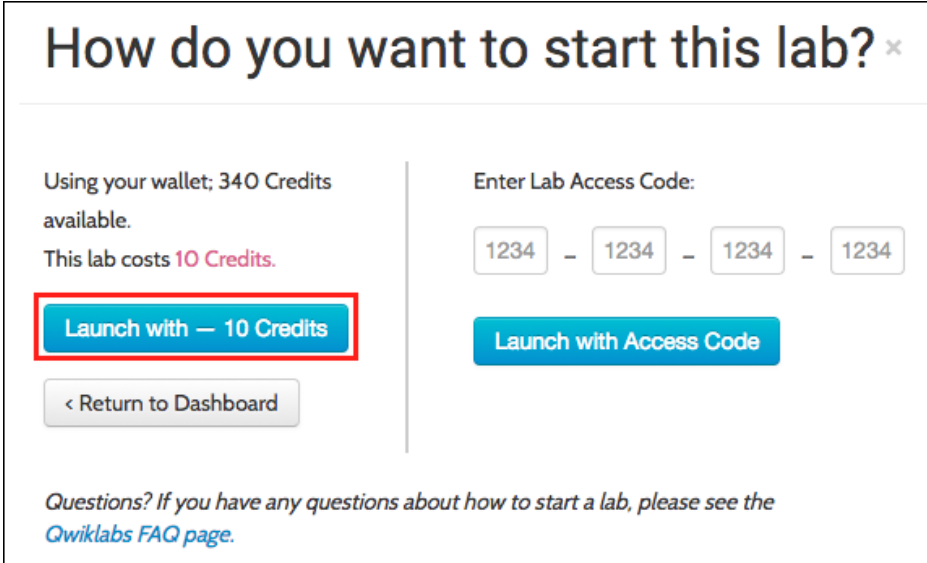
This lab takes approximately **45 minutes** to complete.

Accessing the AWS Management Console

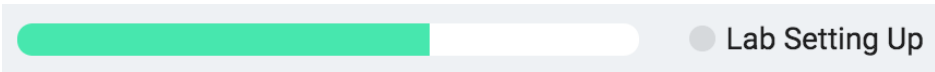
1. At the top of these instructions, click **START LAB** to launch your lab.

A green rectangular button with the text "START LAB" in white, bold, uppercase letters.

2. When asked *How do you want to start this lab*, click **Launch with credits**.

A dialog box titled "How do you want to start this lab?" with a close button (x) in the top right. It is divided into two columns. The left column shows "Using your wallet; 340 Credits available." and "This lab costs 10 Credits." Below this is a blue button labeled "Launch with — 10 Credits" which is highlighted with a red rectangle. At the bottom of the left column is a grey button labeled "< Return to Dashboard". The right column is titled "Enter Lab Access Code:" and contains four input boxes, each with "1234", separated by hyphens. Below this is a blue button labeled "Launch with Access Code". At the bottom of the dialog, there is a link: "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.

A horizontal progress bar with a green segment on the left and a white segment on the right. To the right of the bar is a grey circle followed by the text "Lab Setting Up".

3. Click **Open Console**.

A yellow rectangular button with the text "OPEN CONSOLE" in white, bold, uppercase letters.

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, click **Services** then click **IAM** (Identity & Access Management).
6. In the left navigation pane, click **Users**.

Some IAM Users have already 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 on the name **user-1** to reveal details about the User.
8. In the User details, find the following facts about this User:
 - It currently has no policies attached (**Permissions** tab)
 - It is not a member of any group (**Groups** tab)
 - It has a password assigned already (**Security credentials** tab)
9. In the left navigation pane, click **Groups**.

The following groups have already been created for you:

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

10. Click on the group name for **EC2-Support**.
11. 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 immediately apply against all Users and Groups that are attached to the policy.

12. Under **Actions**, click **Show Policy**.

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.

Note the basic structure of the statements in the IAM Policies:

- **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*).

13. Close the dialog box, click **Groups** again and look at the permissions assigned to the **S3-Support** group.

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

14. Close the dialog box, click **Groups** again and look at the permissions assigned to the **EC2-Admin** group.

This Group is different. Instead of a *Managed Policy*, it has an **Inline Policy**, which is a policy assigned to just one User or Group. Inline Policies are used to override standard permissions for specific situations.

15. Under **Actions**, click **Edit Policy** to view the policy.

The policy is granting permission to view (Describe) information about Amazon EC2 and also the ability to Start and Stop instances.

16. 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.

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

18. Click on the **S3-Support** group name.

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

20. Select **user-1** and click the blue **Add Users** button in the lower-right.

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

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

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

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

23. When you are finished, again click **Groups** in the left navigation pane. 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 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.

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

An **IAM users sign-in link** is displayed. It should 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.

25. Copy the **IAM users sign-in link**.

26. Paste a copy of the sign-in link in a text editor because you will be using it again several times.

27. Paste the link into your web browser address bar and hit Enter. This will take you to the Sign-In screen.

💡 If you receive a message stating *"You must first log out before logging into a different AWS account"*, then click the "To logout, click here" link, then paste the link into your web browser address bar again and hit Enter. This will take you to the Sign-In screen.

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

28. Sign-in with:

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

29. From the main AWS Console, click **Services** and then **S3**.

30. Click the name of a bucket and browse the contents.

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

Now, test whether they have access to Amazon EC2.

31. From the main AWS Console, click **Services** and then **EC2**.

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

You cannot see any instances! Instead, it says *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.

33. 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.

34. Sign-in with:

- **IAM user name:** `user-2`
- **Password:** `lab-password`

35. From the main AWS Console, click **Services** and then **EC2**.

36. In the left navigation pane, 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**).

37. Select the EC2 instance.

38. In the **Actions** menu, under **Instance State**, click **Stop**.

39. Click **Yes, Stop**.

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

Next, check whether they can access Amazon S3.

40. From the main AWS Console, click **Services** and then **S3**.

You receive an *Access Denied* error because this User does not permission to use Amazon S3.

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

41. 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.

42. Sign-in with:

- **IAM user name:** `user-3`
- **Password:** `lab-password`

43. From the main AWS Console, click **Services** and then **EC2**.

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

As an EC2 Administrator, you should now have permissions to Stop the Amazon EC2 instance.

45. Select the EC2 instance.

⚠ 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**).

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

47. Click **Yes, Stop**.

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

Conclusion

Congratulations! You now have:

- 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

Lab Complete

You have successfully completed the lab. To clean up your lab environment, do the following:

48. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
49. Return to the **qwikLABS** page where you launched your lab and click **END LAB**.

Lab Feedback

For feedback, suggestions, or corrections, please email us at *aws-course-feedback@amazon.com*.