AWS Technical Essentials
Lab Guide
Version 4.1
100-ESS-41-EN-LG

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/.

Contents

Lab 0
Signing In to the AWS Management Console

Introduction

In this lab, you will learn how to sign in to the student account created for you as part of this course.

Prerequisites

All labs require the following:

- Access to a computer with Wi-Fi running Microsoft Windows, Mac OS X, or Linux (Ubuntu, SuSE, or Red Hat).
    - The qwikLABS lab environment is not accessible using an iPad or tablet device, but you can use these devices to access the student guide.
- An Internet browser such as Chrome, Firefox, or IE9 or later (previous versions of Internet Explorer are not supported).

Student Errata

Any errata that have been published since the release of this version can be found here:

http://d2lrzjb0vjvpn5.cloudfront.net/AWS-100-ESS/v4.1/errata/static/AWSTechnicalEssentials_4.1_Student_Errata.pdf

Task 1.1: Signing In

These instructions walk you through signing in to the AWS Management Console.

1.1.1 From the **Class Details** page in qwikLABS, find the current lab, and click **Select**.

1.1.2 Click **Start Lab**.

1.1.3 On the lab page, wait until the text *Create in Progress...* disappears from the screen. For some labs, this may happen instantly; for other labs, it may take from five to 10 minutes for your lab to initialize.

Note Make sure to wait until the lab creation process has been completed before you continue to the next step.

1.1.4 Under **AWS Management Console**, you will see the values of **User Name** and **Password**. These are your AWS account credentials. Select and copy the **Password** value.

1.1.5 Click **Open Console**. This will open the AWS Management Console, and pre-populate it with the AWS account ID.

1.1.6 For **User Name**, type **awsstudent**.

1.1.7 For **Password**, paste the password that you copied from Step 1.1.4.

1.1.8 Click **Sign In**.

Lab 1
Build a VPC and Deploy a Web Server

Overview

In this lab session, you will 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 will configure and customize the EC2 instance to run a web server and launch it into the VPC.

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd 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*.

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.

Duration

This lab will take approximately 45 minutes.

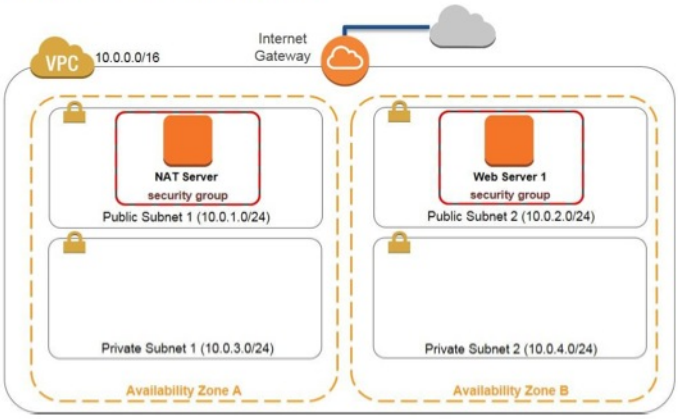
Task 1: Create Your VPC

Overview

In this section you create your VPC.

Scenario

In this lab you build the following infrastructure:



Task 1.1: Create Your VPC

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

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

1.1.2 Click **Start VPC Wizard**.

1.1.3 In the navigation pane, click **VPC with Public and Private Subnets**.

1.1.4 Click **Select**.

1.1.5 Do the following:

a. For **IP CIDR block**, type **10.0.0.0/16**

b. For **VPC name**, type **My Lab VPC**

c. For **Public subnet**, type **10.0.1.0/24**
(You can safely ignore the error "Public and private subnet CIDR blocks overlap.")

d. For **Availability Zone**, click the first Availability Zone. This is referred to as the *first Availability Zone* in further sections.

e. For **Public subnet name**, type **Public Subnet 1**

f. For **Private subnet**, type **10.0.3.0/24**

g. For **Availability Zone**, click the first Availability Zone also used for Public Subnet 1.

h. For **Private subnet name**, type **Private Subnet 1**

1.1.6 In **Specify the details of your NAT gateway**, click **Use a NAT instance instead**.

1.1.7 Select the first instance type listed in **Instance type**.

1.1.8 For **Key pair name**, click the **qwikLABS** key pair from the drop-down list. Leave the remaining options as their default values.

1.1.9 Click **Create VPC**.

1.1.10 In the message that your VPC was successfully created, click **OK**.

Task 1.2: Create Additional Subnets

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

1.2.1 In the navigation pane, click **Subnets**.

1.2.2 Click **Create Subnet**.

1.2.3 In the **Create Subnet** dialog box, do the following:

a. For **Name tag**, type **Public Subnet 2**

b. For **VPC**, click **My Lab VPC**

c. For **Availability Zone**, select the second Availability Zone. This is referred to as the *second Availability Zone* in further sections.

d. For **CIDR block**, type **10.0.2.0/24**

1.2.4 Click **Yes, Create**.

1.2.5 Click **Create Subnet**.

1.2.6 In the **Create Subnet** dialog box, do the following:

a. For **Name tag**, type **Private Subnet 2**

b. For **VPC**, click **My Lab VPC**

c. For **Availability Zone**, click the second Availability Zone also used for Public Subnet 2.

d. For **CIDR block**, type **10.0.4.0/24**

1.2.7 Click **Yes, Create**.

1.2.8 In the navigation pane, click **Route Tables**.

1.2.9 Select the route table with the CIDR range **10.0.0.0/16** and with the option **Yes** under **Main**.

1.2.10 Double-click the empty **Name** for this route table, type **Private Route Table**, and click the checkmark to save.

1.2.11 In the lower pane, click **Routes** and note that **Destination 0.0.0.0/0** is set to **Target eni-xxxxxxxx / i-xxxxxxxx**. 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.

1.2.12 Click **Subnet Associations** and then click **Edit**.

1.2.13 Select **Private Subnet 1** and **Private Subnet 2**.

1.2.14 Click **Save**.

1.2.15 In the upper pane select the route table with the CIDR range **10.0.0.0/16** and with the option **No** under **Main**.

1.2.16 Double-click the empty **Name** for this route table, type **Public Route Table**, and click the checkmark to save.

1.2.17 In the lower pane, click **Routes** and note that **Destination 0.0.0.0/0** is set to **Target igw-xxxxxxxx**. This route table is used by public subnets for communication.

1.2.18 Click **Subnet Associations** and then click **Edit**.

1.2.19 Select **Public Subnet 1** and **Public Subnet 2**.

1.2.20 Click **Save**.

Task 1.3: Create a VPC Security Group

You will create a VPC security group that permits access for web traffic.

1.3.1 In the navigation pane, click **Security Groups**.

1.3.2 Click **Create Security Group**.

1.3.3 In the **Create Security Group** dialog box, do the following:

a. For **Name tag**, type **WebSecurityGroup**

b. For **Group name**, type **WebSecurityGroup**

c. For **Description**, type **Enable HTTP access**

d. For **VPC**, click the VPC you created in Task 1.1 (**My Lab VPC**).

1.3.4 Click **Yes, Create**.

1.3.5 Select **WebSecurityGroup**.

1.3.6 Click the **Inbound Rules** tab.

1.3.7 Click **Edit**.

1.3.8 For **Type**, click **HTTP (80)**.

1.3.9 Click in the **Source** box and type **0.0.0.0/0**

1.3.10 Click **Save**.

Task 2: Launch Your Web Server

Overview

Now that you have launched your VPC, you will launch an EC2 instance into it and bootstrap the instance to act as a web server.

Command Reference File

Use the command reference file when copying text provided in this lab manual. The command reference file is available in the qwikLABS page for the lab.

You should not copy and paste commands directly from this lab manual, because the manual's rich formatting may inject characters that could introduce errors to your lab experience. Download the reference file to your computer instead.

Task 2.1: Launch Your First Web Server Instance

This task walks you through launching an EC2 instance into your VPC. This instance will act as your web server.

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

2.1.2 Click **Launch Instance**.

2.1.3 In the row for **Amazon Linux AMI**, click **Select**.

2.1.4 On the **Step 2: Choose an Instance Type** page, confirm that **t2.micro** is selected and then click **Next: Configure Instance Details**.

2.1.5 On the **Step 3: Configure Instance Details** page, do the following and leave all other values with their default:

a. For **Network**, click the VPC that you created in Task 1.1 (**My Lab VPC**).

b. For **Subnet**, click the **Public Subnet 2 (10.0.2.0/24)** you created in Task 1.2.

c. For **Auto-assign Public IP**, click **Enable**.

2.1.6 Expand the **Advanced Details** section.

2.1.7 Copy the following user data from the command reference file, and then paste it in 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/v4.1/lab-2-configure-website-
datastore/scripts/lab2-app.tar.gz
tar xvfz lab2-app.tar.gz
chown apache:root /var/www/html/lab2-app/rds.conf.php
fi
```

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

2.1.8 Click **Next: Add Storage**.

2.1.9 Click **Next: Tag Instance**.

2.1.10 On the **Step 5: Tag Instance** page, do the following:

a. For **Key**, type **Name**

b. For **Value**, type **Web Server 1**

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

2.1.12 On the **Step 6: Configure Security Group** page, click **Select an existing security group** and then select the security group you created in Task 1.3 (**WebSecurityGroup)**.

2.1.13 Click **Review and Launch**.

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

2.1.15 Click **Choose an existing key pair**, click the **qwikLABS** key pair, select the acknowledgement check box, and then click **Launch Instances**.

2.1.16 Scroll down and click **View Instances**.

You will see two instances – **Web Server 1** and the NAT instance launched by the VPC Wizard.

2.1.17 Wait until **Web Server 1** shows *2/2 checks passed* in the **Status Checks** column. This will take 3–5 minutes. Use the refresh icon  at the upper right to check for updates.

2.1.18 Select Web Server 1 and copy the **Public DNS** value that appears on the **Description** tab.

2.1.19 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.

Lab Complete

Congratulations! You have successfully created a VPC and launched an EC2 instance into it. To clean up your lab environment, do the following:

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

Lab 2
Build Your Database Server and Connect to it

Overview

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

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you up to focus on your applications and business. Amazon RDS provides you 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).

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.

Duration

This lab will take approximately 45 minutes.

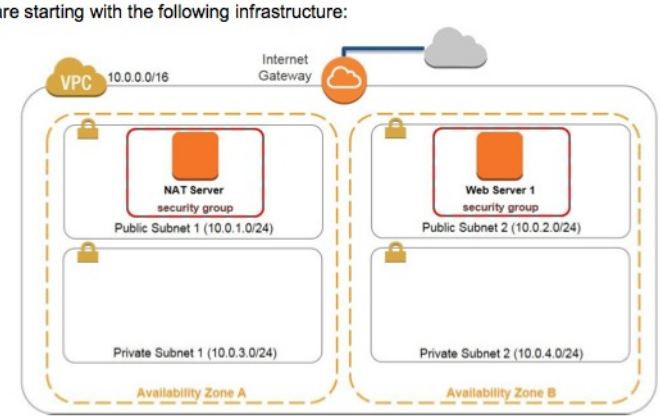
Task 1: Launch an Amazon RDS DB Instance

Overview

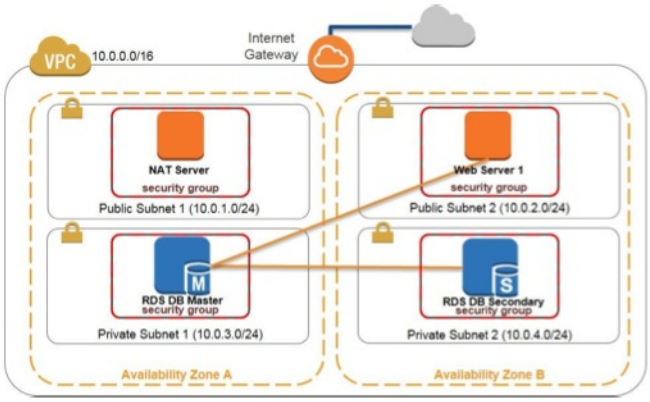
In this task, you launch an Amazon RDS DB instance backed by MySQL.

Scenario

You are starting with the following infrastructure:



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



Task 1.1: 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.

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

1.1.2 In the navigation pane, click **Security Groups**.

1.1.3 Click **Create Security Group**.

1.1.4 In the **Create Security Group** dialog box, do the following:

a. For **Name tag**, type **DBSecurityGroup**

b. For **Group name**, type **DBSecurityGroup**

c. For **Description**, type **DB Instance Security Group**

d. For **VPC**, click **My Lab VPC**.

1.1.5 Click **Yes, Create**.

1.1.6 Select **DBSecurityGroup** you just created and ensure that all other security groups are cleared.

1.1.7 Click the **Inbound Rules** tab, and then click **Edit**.

1.1.8 Do the following:

a. For **Type**, click **MySQL/Aurora (3306)**.

b. For **Protocol**, click **TCP(6)**.

c. For **Source**, click **WebSecurityGroup**.

1.1.9 Click **Save**.

Task 1.2: 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.

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

1.2.2 In the navigation pane, click **Subnet Groups**.

1.2.3 Click **Create DB Subnet Group**.

1.2.4 On the **Create DB Subnet Group** page, do the following:

a. For **Name**, type **dbgroup**

b. For **Description**, type **Lab DB Subnet Group**

c. For **VPC ID**, click **My Lab VPC**.

1.2.5 For **Availability Zone**, click the first Availability Zone.

1.2.6 For **Subnet ID**, click **10.0.3.0/24**.

1.2.7 Click **Add**.

1.2.8 For **Availability Zone**, click the second Availability Zone. This adds another subnet to the DB subnet group.

1.2.9 For **Subnet ID**, click **10.0.4.0/24**

1.2.10 Click **Add**.

1.2.11 Click **Create**.

1.2.12 If you do not see your new subnet group, click the refresh icon in the upper-right corner of the console.

Task 1.3: Create an RDS DB Instance

In this task you configure and launch your MySQL-backed Amazon RDS DB instance.

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

1.3.2 Click **Launch DB Instance**.

1.3.3 Click **MySQL** > **Select**.

1.3.4 Under **Production**, click **MySQL**.

1.3.5 Click **Next Step**.

1.3.6 On the **Specify DB Details** page, do the following:

a. For **DB Instance Class**:, click the first option in the list (for example: *db.t2.micro*).

b. For **Multi-AZ Deployment**:, click **Yes**.

c. For **DB Instance Identifier**, type **DB1**

d. For **Master Username**, type **labuser**

e. For **Master Password**, type **labpassword**

f. For **Confirm Password**, type **labpassword**

1.3.7 Click **Next Step**.

1.3.8 On the **Configure Advanced Settings** page, do the following, leaving all other values and sections with their default:

a. For **VPC**, click **My Lab VPC**.

b. For **Subnet Group**, click **dbgroup**.

c. For **Publicly Accessible**, click **No**.

d. For **VPC Security Group(s)**, click **DBSecurityGroup (VPC)**.

e. For **Database Name**, type **sampleDB**

f. For **Enable Enhanced Monitoring**, click **No**.

1.3.9 Click **Launch DB Instance**.

1.3.10 Click **View Your DB Instances**.

1.3.11 Select **DB1** and wait until the endpoint is *available*, *modifying*, or has transitioned from "Not available yet" to a string ending with "3306" – this may take up to 10 minutes. Click the refresh icon in the upper-right corner to check for updates.

1.3.12 Copy and save the value of the endpoint in a text file, **making sure to omit the :3306**. Your endpoint should look similar to the following example: *db1.cze6p5rivinc.us-west-2.rds.amazon.com*

Task 2: Interact with Your Database

Overview

In this task you interact with your database through a PHP web application that was deployed to the web server in Lab 1.

Task 2.1: Access the Database Web Application

Open a web application running on your web server.

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

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

2.1.3 Select **Web Server 1**, ensure that all other instances are cleared, and view the **Description** tab in the lower pane.

2.1.4 Copy the **Public IP** address of **Web Server 1** that appears in the lower pane.

2.1.5 Paste the IP address in a new browser tab or window. A web application is displayed with the web server's instance metadata.

2.1.6 Click the **RDS** link.

2.1.7 Do the following:

a. For **Endpoint**, paste the endpoint you copied previously, **making sure to omit the :3306**.

b. For **Database**, type **sampleDB**

c. For **Username**, type **labuser**

d. For **Password**, type **labpassword**

2.1.8 Click **Submit**. The connection string is displayed and then the page is redirected. Two new records are added to the address table and displayed.

2.1.9 Test whether the PHP web application can communicate with the RDS DB database. To add another contact, click **Add Contact** and enter a **Name**, **Phone**, and **Email**, and then click **Submit**.

2.1.10 To edit a contact, click **Edit**, modify one of the fields, and then click **Submit**.

2.1.11 To remove a record, click **Remove**. You can now close this browser tab or window.

Lab Complete

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

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

Lab 3
Scale and Load Balance your Architecture

Overview

This lab builds on the previous lab and walks you through using the Elastic Load Balancing (ELB) and Auto Scaling services to load balance and autoscale your infrastructure.

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications, seamlessly providing the required amount of load balancing capacity needed to route application traffic. Elastic Load Balancing offers two types of load balancers that both feature high availability, automatic scaling, and robust security. These are the Classic Load Balancer which routes traffic based on either application- or network-level information, and the Application Load Balancer which routes traffic based on advanced application-level information that includes the content of the request. 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.

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.
- Autoscale new instances within a private subnet.
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

Duration

This lab will take approximately 45 minutes.

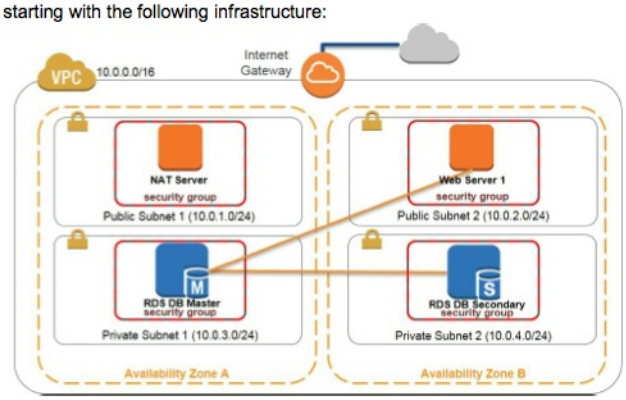
Task 1: Auto Scaling

Overview

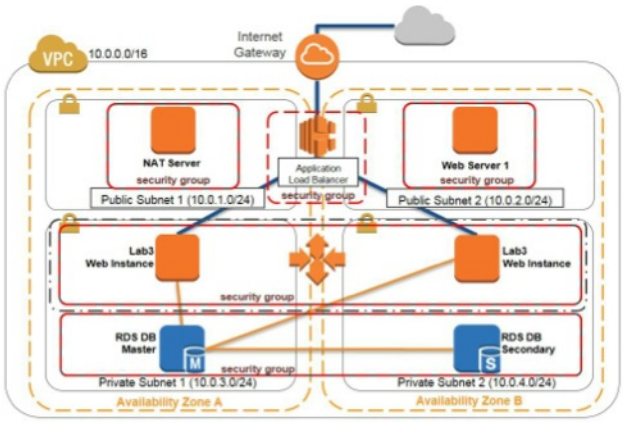
In this task you create and autoscale your infrastructure.

Scenario

You are starting with the following infrastructure:



The final state of the infrastructure is:



Task 1.1: Create an AMI for Auto Scaling

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

1.1.1 In the **AWS Management Console**, on the **Services** menu, click **EC2**.

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

1.1.3 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.

1.1.4 Right-click on **Web Server 1**, and then click **Image** > **Create Image**.

1.1.5 Do the following and leave the other values with their default:

a. For **Image name**, type **Web Server AMI**

b. For **Image description**, type **Lab 3 AMI for Web Server**

1.1.6 Click **Create Image**.

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

Task 1.2: Create a Load Balancer

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

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

1.2.2 Click **Create Load Balancer**.

1.2.3 Select **Application Load Balancer** and click **Continue**.

1.2.4 Do the following and leave the remaining values with their default:

a. For **Name**, type **Lab3ELB**

b. For **VPC**, click **My Lab VPC**.

c. For **Select Subnets**, click the **+** to select **Public Subnet 1** and **Public Subnet 2**.

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

1.2.6 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**.

1.2.7 Select the security group that contains **WebSecurityGroup** in the **Name** and a **Description** of **Enable HTTP access** and clear the **default** security group check box.

1.2.8 Click **Next: Configure Routing**.

1.2.9 Under **Target group**, for **Name**, type **Lab3Group**.

1.2.10 Expand **Advanced health check settings**, do the following and leave the remaining values with their default:

a. For **Healthy threshold**, type **2**

b. For **Timeout**, type **8**

c. For **Interval**, type **10**

1.2.11 Click **Next: Register Targets**.

1.2.12 Auto Scaling will automatically add instances later. Click **Next: Review**.

1.2.13 Review the configuration of your load balancer and click **Create**.

1.2.14 Click **Close**.

Task 1.3: Create a Launch Configuration and an Auto Scaling Group

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.

1.3.1 In the navigation pane, click **Launch Configurations**.

1.3.2 Click **Create Auto Scaling group**.

1.3.3 Click **Create launch configuration**.

1.3.4 In the navigation pane, click **My AMIs**.

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

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

1.3.7 Do the following and leave the remaining values with their default:

a. For **Name**, type **Lab3Config**

b. For **Monitoring**, click **Enable CloudWatch detailed monitoring**.

1.3.8 Click **Next: Add Storage**.

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

1.3.10 Click **Select an existing security group** and select the security group that contains **WebSecurityGroup** in the **Name** and a **Description** of **Enable HTTP access**.

1.3.11 Click **Review**.

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

1.3.13 Click **Choose an existing key pair**, select the **qwikLABS** key pair, select the acknowledgement check box, and click **Create launch configuration**.

1.3.14 Do the following for your auto scaling group and leave the remaining values with their default:

a. For **Group name**, type **Lab3 AS Group**

b. For **Group size Start with**, type **2** (instances)

c. For **Network**, click **My Lab VPC**.

d. For **Subnet**, click **Private Subnet 1 (10.0.3.0/24)** and **Private Subnet 2 (10.0.4.0/24)**.

1.3.15 Expand **Advanced Details**, do the following, and leave the remaining values with their default:

a. For **Load Balancing**, click **Receive traffic from one or more load balancers**.

b. For **Target Groups**, click **Lab3Group**.

c. For **Health Check Type**, click **ELB**.

d. For **Monitoring**, click **Enable CloudWatch detailed monitoring**.

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

1.3.17 Click **Use scaling policies to adjust the capacity of this group**.

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

1.3.19 In **Increase Group Size**, for **Execute policy when**, click **Add new alarm**.

1.3.20 Verify that **Send a notification to:** is not selected.

1.3.21 Do the following and leave the remaining with their default values:

a. For **Whenever**, click **Average**, and then click **CPU Utilization**

b. For **Is**, click **>=**, and then type **65 (**Percent).

c. For **For at least**, type **1 (**consecutive period(s) of) and then click **1 Minute**.

d. For **Name of alarm**, type **High CPU Utilization**

1.3.22 Click **Create Alarm**.

1.3.23 In **Increase Group Size**, do the following:

a. For **Take the action**, click **Add**, type **1**, click **instances**, and then type **65**

b. For **Instances need**, type **60** (seconds to warm up after each step).

1.3.24 In **Decrease Group Size**, for **Execute policy when**, click **Add new alarm**.

1.3.25 Verify that **Send a notification to:** is not selected.

1.3.26 Do the following and leave the remaining with their default values:

a. For **Whenever**, click **Average**, then click **CPU Utilization**

b. For **Is**, click **<=**, and then type **20** (Percent).

c. For **For at least**, type **1 (**consecutive period(s) of) and then click **1 Minute**.

d. For **Name of alarm**, type **Low CPU Utilization**

1.3.27 Click **Create Alarm**.

1.3.28 In **Decrease Group Size**, for **Take the action**: click **Remove**, type **1**, click **instances**, and then type **20**

1.3.29 Click **Next: Configure Notifications**.

1.3.30 Click **Next: Configure Tags**.

1.3.31 Do the following and leave the other values with their default:

a. For **Key**, type **Name**

b. For **Value**, type **Lab 3 Web Instance**

1.3.32 Click **Review**.

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

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

Task 1.4: Verify Auto Scaling is Working

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

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

Four instances are displayed: **Web Server 1**, **NAT Server**, and two new instances labeled as **Lab 3 Web Instance.**

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

1.4.3 Select **Lab3Group** and click the **Targets** tab. You see two **Lab 3 Web Instance** instances listed for this target group.

1.4.4 Wait until the instance displays a **Status** of *healthy*. Use the refresh icon in the upper right corner to check for updates.

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

1.4.6 Select **Lab3ELB** and on the **Description** tab in the lower pane, copy the **DNS name** of your load balancer, making sure to omit "(A Record)".

Task 2: Monitor Your Infrastructure

Overview

You have 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 ≥ 65% and ≤ 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.

Task 2.1: Test Auto Scaling

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

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

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

The two alarms **High CPU Utilization** and **Low CPU Utilization** are displayed. **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).

2.1.3 Paste the load balancer's DNS name that you copied in step 1.4.6 in a new browser window or tab and press **ENTER**.

2.1.4 Click **LOAD TEST** under 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.

2.1.5 Navigate back 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*. Use the refresh icon to see the changes.

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

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

More than two instances labeled **Lab 3 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 2.2: Optional: Terminate Web Server 1

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

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

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

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

2.2.4 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:

1. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
2. Return to the **qwikLABS** page where you launched your lab from and click **End**.