AWS Technical Essentials
Lab Guide
Version 4.2

100-ESS-42-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 1: Build Your VPC and Launch a Web Server

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.

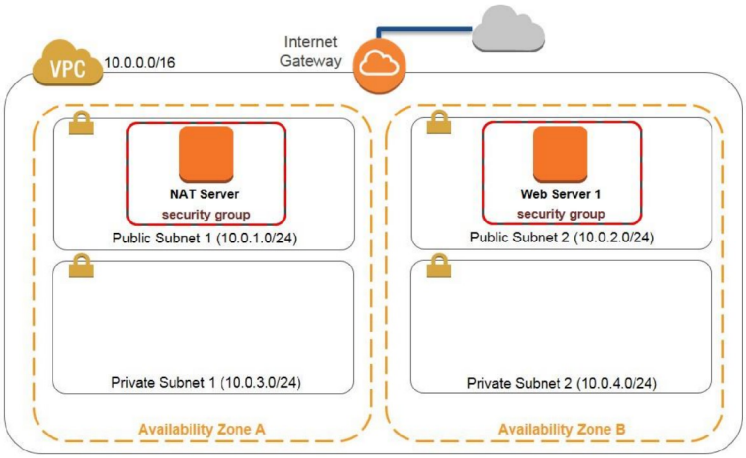
Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure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

Scenario

In this lab you build the following infrastructure:



Objectives

After completing this lab, you can:

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

Duration

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

Accessing the AWS Management Console

1. To the right of the lab title, click **Start Lab** to launch your Qwiklabs.

2. On the **Connect** tab of the Qwiklabs page, copy the **Password** to the clipboard and then click **Open Console**.

3. Sign in to the AWS Management Console using the following steps:

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

   b. For **Password**, paste the password copied from the clipboard.

   c. Click **Sign In**.

Task 1: Create Your VPC

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

4. In the **AWS Management Console**, on the **Services** menu, click **VPC**.
5. Click **Start VPC Wizard**.
6. In the navigation pane, click **VPC with Public and Private Subnets**.
7. Click **Select**.
8. 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.
9. Click **Create VPC**.
10. In the success message, click **OK**.

Task 2: Create Additional Subnets

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

11. In the navigation pane, click **Subnets**.
12. Click **Create Subnet**.
13. 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`
14. Click **Yes, Create**.
15. Click **Create Subnet**.
16. 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`
17. Click **Yes, Create**.
18. In the navigation pane, click **Route Tables**.
19. Select the route table with the VPC **My Lab VPC** and **Yes** under **Main**.
20. Double-click the empty **Name** for this route table, type `Private Route Table`, and click the checkmark to save.
21. 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.
22. Click **Subnet Associations**, and then click **Edit**.

23. Select **Private Subnet 1** and **Private Subnet 2**.
24. Click **Save**.
25. Select the route table with the VPC **My Lab VPC** and **No** under **Main**.
26. Double-click the empty **Name** for this route table, type `Public Route Table`, and click the checkmark to save.
27. 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.
28. Click **Subnet Associations**, and then click **Edit**.
29. Select **Public Subnet 1** and **Public Subnet 2**.
30. Click **Save**.

Task 3: Create a VPC Security Group

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

31. In the navigation pane, click **Security Groups**.
32. Click **Create Security Group**.
33. 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
34. Click **Yes, Create**.
35. Select **WebSecurityGroup**.
36. Click the **Inbound Rules** tab.
37. Click **Edit**.
38. For **Type**, click `HTTP (80).`
39. Click in the **Source** box and type `0.0.0.0/0`
40. Click **Save**.

Task 4: Launch Your First Web Server Instance

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

41. On the **Services** menu, click **EC2**.
42. Click **Launch Instance**.
43. In the row for **Amazon Linux AMI**, click **Select**. If you receive a warning, click **Continue**.
44. On the **Step 2: Choose an Instance Type** page, confirm that **t2.micro** is selected and then click **Next: Configure Instance Details**.
45. 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."
46. Expand the **Advanced Details** section.
47. 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.

48. Click **Next: Add Storage**.
49. Click **Next: Add Tags**.
50. 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`

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

52. 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**).

53. 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**.

54. 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.

55. Click **Choose an existing key pair**, click the **Qwiklabs** key pair, select the acknowledgment check box, and then click **Launch Instances**.

56. Scroll down and click **View Instances**. You will see two instances – **Web Server 1** and the NAT instance launched by the VPC Wizard.

57. 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.

58. Select Web Server 1 and copy the **Public DNS** value on the **Description** tab.

59. 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:

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

61. Return to the **Qwiklabs** page where you launched your lab and click **End**.

Lab 2: Build Your DB Server and Interact With Your DB Using an App

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 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).

Objectives

After completing this lab, you can:

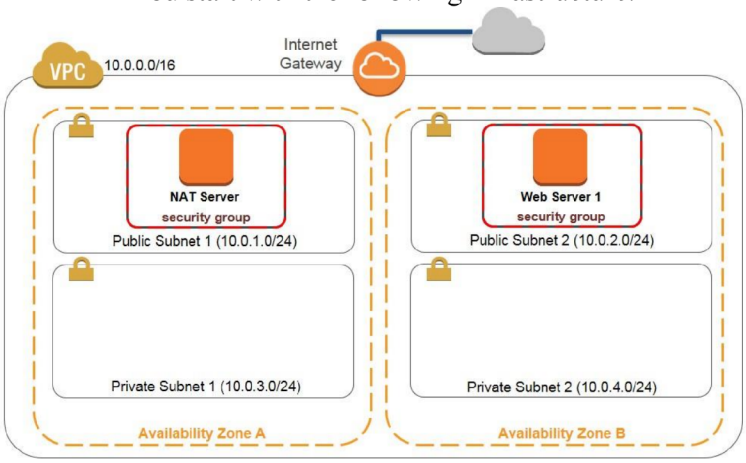
- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

Duration

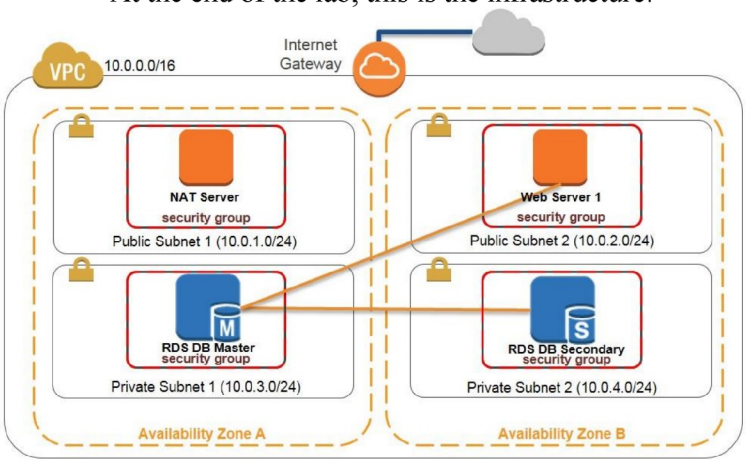
This lab takes approximately **45 minutes**.

Scenario

You start with the following infrastructure:



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



Accessing the AWS Management Console

1. To the right of the lab title, click **Start Lab** to launch your Qwiklabs.

2. On the **Connect** tab of the Qwiklabs page, copy the **Password** to the clipboard and then click **Open Console**.

3. Sign in to the AWS Management Console using the following steps:

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

   b. For **Password**, paste the password copied from the clipboard.

   c. Click **Sign In**.

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.

4. In the **AWS Management Console,** on the **Services** menu, click **VPC**.
5. In the navigation pane, click **Security Groups**.
6. Click **Create Security Group**.
7. In the **Create Security Group** dialog box, configure the following settings (and ignore any settings that aren't listed):
   a. **Name tag**: type `DBSecurityGroup`
      Ignore errors about description being required.
      That will be fixed below.
   b. **Group name**: type `DBSecurityGroup`
      It will be filled automatically.
   c. **Description**: type `DB Instance Security Group`
   d. **VPC**: Click **My Lab VPC**.
8. Click **Yes, Create**.
9. Select the security group you just created (**DBSecurityGroup**) and ensure that all other security groups are not selected.
10. Click the **Inbound Rules** tab, and then click **Edit**.
11. Configure the following settings (and ignore any settings that aren't listed):
    a. **Type**: Click `MySQL/Aurora (3306).`
    b. **Protocol**: Click **TCP(6)**.
    c. **Source**: Click **WebSecurityGroup**.
12. 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.

13. On the **Services** menu, click **RDS**.
14. In the navigation pane, click **Subnet Groups**.
15. Click **Create DB Subnet Group**.
16. On the **Create DB Subnet Group** page, configure the following settings (and ignore any settings that aren't listed):
    a. **Name**: type `dbgroup`
    b. **Description**: type `DB Instance Subnet Group`
    c. **VPC ID**: Click **My Lab VPC**.
17. For **Availability Zone**, select the first Availability Zone.
18. For **Subnet ID**, select **10.0.3.0/24**.
19. Click **Add**.
20. For **Availability Zone**, click the second Availability Zone. This adds another subnet to the DB subnet group.
21. For **Subnet ID**, click **10.0.4.0/24**.
22. Click **Add**.
23. Click **Create**.

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

Task 3: Create an RDS DB Instance

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

24. In the navigation pane, click **Instances**.
25. Click **Launch DB Instance**.
26. Click **MySQL** > **Select**.
27. Under **Production**, click **MySQL**.
28. Click **Next Step**.
29. On the **Specify DB Details** page, configure the following settings (and ignore any settings that aren't listed):
    a. **DB Instance Class**: Click the first option in the list.
    (for example: **db.t2.micro**)
    b. **Multi-AZ Deployment**: Click **Yes**.
    c. **DB Instance Identifier**: type `Lab5DB`
    d. **Master Username**: type `labuser`
    e. **Master Password**: type `labpassword`
    f. **Confirm Password**: type `labpassword`
30. Click **Next Step**.
31. On the **Configure Advanced Settings** page, configure the following settings (and ignore any settings that aren't listed):
    a. **VPC**: Click **My Lab VPC**.
    b. **Subnet Group**: Click **dbgroup**.
    c. **Publically Accessible**: Click **No**.
    d. **VPC Security Group(s)**: Click **DBSecurityGroup (VPC)**.
    e. **Database Name**: type `sampleDB`
    f. **Enable Enhanced Monitoring**: Click **No**.
32. Click **Launch DB Instance**.
33. Click **View Your DB Instances**.

34. Select **DB1** and wait until the endpoint is *available* or *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.

35. Copy and save the value of the endpoint in a text file, ommiting **:3306**.
    Your endpoint should look similar to the following example:
    lab5db.cnrczgvaxtw8.us-west-2.rds.amazonaws.com

Task 4: Interact with Your Database

In this task, you interact with your database through a PHP web application that was deployed to the web server in Lab 1. You will open a web application running on your web server.

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

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

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

39. Copy the **IPv4 Public IP** address of **Web Server 1** that appears in the lower pane.

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

41. Click the **RDS** link.

42. Configure the following settings (and ignore any settings that aren't listed):

    a. **Endpoint**: Paste the endpoint you copied previously.
       Omit the **:3306** at the end.

    b. **Database**: type `sampleDB`

    c. **Username**: type `labuser`

    d. **Password**: type `labpassword`
       Ignore any message saying the connection is not secure. This is expected.

43. Click **Submit**.
    The connection string is displayed and then the page is redirected.
    (Alternately click on the **here** link to force redirection)
    Two new records are added to the address table and displayed.

44. Test whether the PHP web application can communicate with the RDS DB database, by adding another contact.
    Click **Add Contact** and enter a **Name**, **Phone**, and **Email**, and then click **Submit Query**.

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

46. 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:

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**.

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.

Lab 3: Scale and Load Balance Your Architecture

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 automatically scale your infrastructure.

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

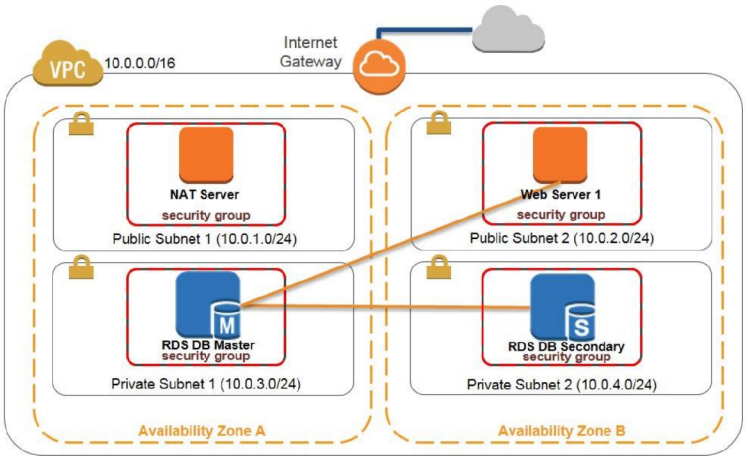
- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

Duration

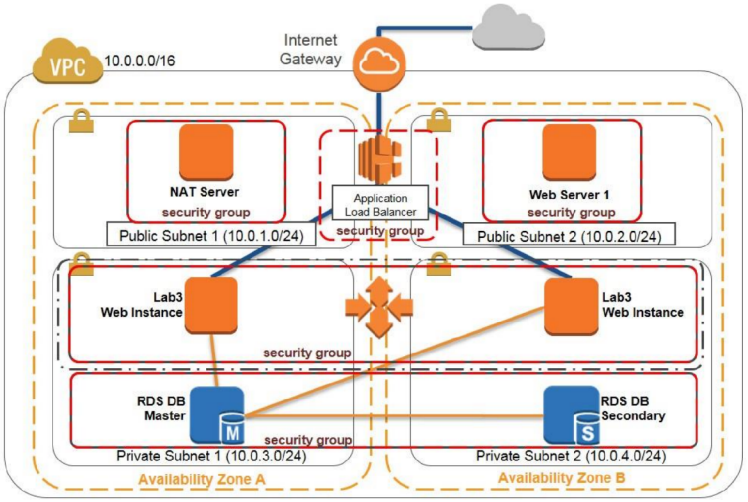
This lab takes approximately **45 minutes**.

Scenario

You start with the following infrastructure:



The final state of the infrastructure is:



Accessing the AWS Management Console

1. To the right of the lab title, click **Start Lab** to launch your Qwiklabs.

2. On the **Connect** tab of the Qwiklabs page, copy the **Password** to the clipboard and then click **Open Console**.

3. Sign in to the AWS Management Console using the following steps:

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

   b. For **Password**, paste the password copied from the clipboard.

   c. Click **Sign In**.

Task 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.

4. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
5. In the navigation pane, click **Instances**.
6. 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.
7. Right-click on **Web Server 1**, and then click **Image** > **Create Image**.
8. Configure the following settings (and ignore any settings that aren't listed):
   a. **Image name**: type `Web Server AMI`
   b. **Image description**: type `Lab 6 AMI for Web Server`
9. Click **Create Image**.

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

Task 2: Create a Load Balancer

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

10. In the navigation pane, click **Load Balancers**.
11. Click **Create Load Balancer**.
12. Select **Application Load Balancer**, and click **Continue**.
13. Configure the following settings (and ignore any settings that aren't listed):
    a. **Name**: type `Lab6ELB`
    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**
14. Click **Next: Configure Security Settings**.
15. 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**.
16. 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).
17. Click **Next: Configure Routing**.
18. Under **Target group**, for **Name**, type `Lab6Group`.
19. 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`
20. Click **Next: Register Targets**.

    Auto Scaling will automatically add instances later. Click **Next: Review**.
21. Review the configuration of your load balancer and click **Create**.
22. On the "Successfully created load balancer" message, click **Close**.

Task 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.

23. In the navigation pane, click **Launch Configurations**.
24. Click **Create Auto Scaling group**.
25. Click **Create launch configuration**.
26. In the navigation pane, click **My AMIs**.
27. In the row for **Web Server AMI**, click **Select**.
28. Accept the **t2.micro** selection and click **Next: Configure details**.
29. Configure the following settings (and ignore any settings that aren't listed):
    a. **Name**: type `Lab6Config`
    b. **Monitoring**: Click **Enable CloudWatch detailed monitoring**.
30. Click **Next: Add Storage**.
31. Click **Next: Configure Security Group**.
32. Click **Select an existing security group**, and select the security group that contains **WebSecurityGroup** in the **Name** and a **Description** of **Enable HTTP access**.
33. Click **Review**.
34. Review the details of your launch configuration and click **Create launch configuration**. Ignore the "Improve security..." warning; this is expected.
35. Click **Choose an existing key pair**, select the **Qwiklabs** key pair, select the acknowledgement check box, and click **Create launch configuration**.
36. Configure the following settings (and ignore any settings that aren't listed):
    a. **Group name**: type `Lab6 AS 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)**.

37. Expand **Advanced Details**, configure the following settings (and ignore any settings that aren't listed):

    a. **Load Balancing**: Click **Receive traffic from one or more load balancers**.

    b. **Target Groups**: Click **Lab6Group**.

    c. **Health Check Type**: Click **ELB**.

    d. **Monitoring**: Click **Enable CloudWatch detailed monitoring**.

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

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

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

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

42. Clear **Send a notification to:** .

43. 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**

44. Click **Create Alarm**.

45. 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)

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

47. Clear **Send a notification to:** .

48. 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**

49. Click **Create Alarm**.

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

51. Click **Next: Configure Notifications**.

52. Click **Next: Configure Tags**.

53. Configure the following settings (and ignore any settings that aren't listed):

    a. **Key**: type `Name`

    b. **Value**: type `Lab 6 Web Instance`

54. Click **Review**.

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

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

Task 4: Verify Auto Scaling is Working

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

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

    Four instances are displayed: **Web Server 1**, **NAT Server**, and two new instances labeled as **Lab 6 Web Instance.**

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

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

59. Select **Lab6Group**, and click the **Targets** tab.

    Two **Lab 6 Web Instance** instances should be listed for this target group.

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

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

62. Select **Lab6ELB** 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

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 ≥ 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.

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

63. On the **Services** menu, click **CloudWatch**.
64. 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).
65. Paste the load balancer's DNS name that you copied in Task 4 in a new browser window or tab and press *ENTER*.
66. 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.
67. 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.
68. On the **Services** menu, click **EC2**.
69. In the navigation pane, click **Instances**.

    More than two instances labeled **Lab 6 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

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.

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

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

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

73. 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:

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

75. Return to the **Qwiklabs** page where you launched your lab from and click **End**.