* **Lab Instructions**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **lab1-webapp-guide.md** |  |

* lab1-webapp-guide.md
* Lab01-webapp-CommandRef.txt

©   
for the use of: shweta\_misra@fanniemae.com

* [Lab 1 - Deploying a Web Application on AWS](https://globalknowledge.qwiklab.com/focuses/17007#lab-1-deploying-a-web-application-on-aws)
  + [Accessing the AWS Management Console](https://globalknowledge.qwiklab.com/focuses/17007#accessing-the-aws-management-console)
  + [Task 1 - Create IAM Users and Roles](https://globalknowledge.qwiklab.com/focuses/17007#task-1-create-iam-users-and-roles)
    - [Task 1.1: Create an IAM User](https://globalknowledge.qwiklab.com/focuses/17007#task-1-1-create-an-iam-user)
    - [Task 1.2: Set Permissions for Lab1User](https://globalknowledge.qwiklab.com/focuses/17007#task-1-2-set-permissions-for-lab1user)
    - [Task 1.3: Create an IAM Role](https://globalknowledge.qwiklab.com/focuses/17007#task-1-3-create-an-iam-role)
    - [Task 1.4: Switch to your IAM user and create a bucket](https://globalknowledge.qwiklab.com/focuses/17007#task-1-4-switch-to-your-iam-user-and-create-a-bucket)
  + [Task 2: Create the Network Layer](https://globalknowledge.qwiklab.com/focuses/17007#task-2-create-the-network-layer)
    - [Task 2.1: Create the VPC](https://globalknowledge.qwiklab.com/focuses/17007#task-2-1-create-the-vpc)
    - [Task 2.2: Attach an Internet Gateway (IGW)](https://globalknowledge.qwiklab.com/focuses/17007#task-2-2-attach-an-internet-gateway-igw)
  + [Task 2.3: Create the Public Subnet](https://globalknowledge.qwiklab.com/focuses/17007#task-2-3-create-the-public-subnet)
  + [Task 2.4: Configuring the Route Table](https://globalknowledge.qwiklab.com/focuses/17007#task-2-4-configuring-the-route-table)
  + [Task 3: Build the Compute, Storage and Database Layer](https://globalknowledge.qwiklab.com/focuses/17007#task-3-build-the-compute-storage-and-database-layer)
    - [Task 3.1: Create the Web Server](https://globalknowledge.qwiklab.com/focuses/17007#task-3-1-create-the-web-server)
  + [Task 3.2: Connect to WebServer (Windows only)](https://globalknowledge.qwiklab.com/focuses/17007#task-3-2-connect-to-webserver-windows-only)
  + [Task 3.3: Connect to the WebServer (Linux/OSX only)](https://globalknowledge.qwiklab.com/focuses/17007#task-3-3-connect-to-the-webserver-linux-osx-only)
  + [Task 3.4: Configure Your Server and the DynamoDB Table](https://globalknowledge.qwiklab.com/focuses/17007#task-3-4-configure-your-server-and-the-dynamodb-table)
  + [Challenge](https://globalknowledge.qwiklab.com/focuses/17007#challenge)
  + [Lab Complete](https://globalknowledge.qwiklab.com/focuses/17007#lab-complete)

Lab 1 - Deploying a Web Application on AWS

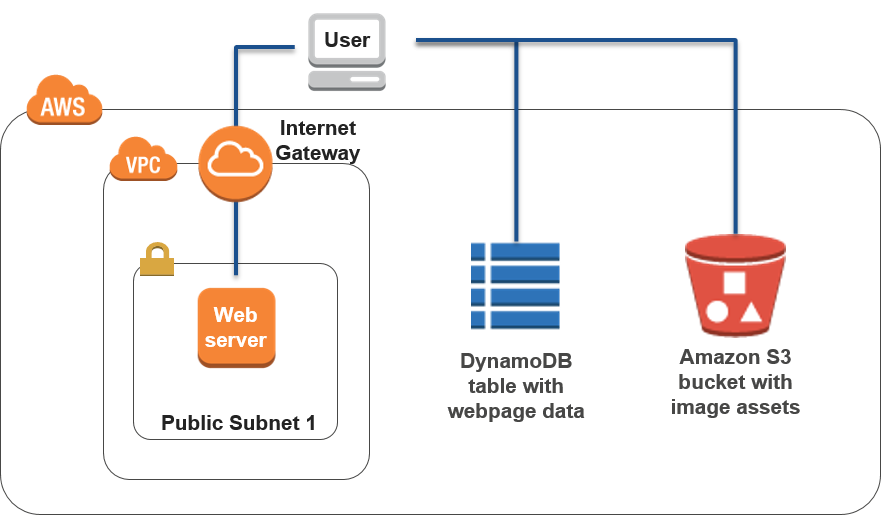
In this lab you will use IAM, EC2, S3, VPC and DynamoDB to deploy a scalable web application.

**Objectives**

After completing this lab, you will be able to:

* Create an IAM User and attach a permission policy for controlling access to services
* Create a Virtual Private Cloud (VPC) with an Internet Gateway (IGW) and a Public Subnet
* Create an S3 Bucket and upload static objects into the bucket
* Create a DynamoDB table and populate with items
* Deploy a web application in a cost optimized and scalable manner utilizing EC2, S3, and DynamoDB

The final product of your lab will be the following application environment:



**Prerequisites**

This lab requires:

* Access to a notebook 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.
* For Microsoft Windows users: Administrator access to the computer
* An Internet browser such as Chrome, Firefox, or IE9 (previous versions of Internet Explorer are not supported)
* An SSH client such as PuTTY

**Duration**

This lab will require around **60 minutes** to complete.

Accessing the AWS Management Console

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

C:\Users\student\Desktop\architect\Architecting on AWS - Lab 1 - Deploying a Simple Web Application _ Qwiklabs + globalknowledge_files\start.png

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

C:\Users\student\Desktop\architect\Architecting on AWS - Lab 1 - Deploying a Simple Web Application _ Qwiklabs + globalknowledge_files\open.png

1. Sign in to the AWS Management Console using the following steps:
   * For **User Name**, type **awsstudent**
   * For **Password**, paste the password copied from the clipboard.
   * Click **Sign In**.

Task 1 - Create IAM Users and Roles

In this part of the lab, you will create a new IAM User, assign permissions based on the least privilege principle, and create a role for your EC2 instance which will allow the web application to reach out to multiple AWS services.

**Task 1.1: Create an IAM User**

In this part of the lab, you will create a new IAM User.

1. In the **AWS Management Console**, on the **Services** menu, click **IAM**.
2. In the navigation pane, click **Users**.
3. Click **Add user**.
4. For the **User name** box, type 
5. Select **Programmatic access** and **AWS Management Console access**.

When selecting the **AWS Management Console access**, you have the option to use either an **Autogenerated password**, which will generate a random password retrievable after the user has been created or a **Custom password**, which you can define yourself.

1. Click **Autogenerated password**.
2. Clear the **Require password reset** check box.
3. Click **Next: Permissions**.
4. Click **Next: Review**.

You will see a warning message telling you that the user has no permissions. You can safely ignore it because you will grant the appropriate permissions to your Lab1User IAM user in the next task.

This policy grants the selected read-only access to all Amazon IAM resources.

1. Click **Create user**.
2. Click **Download.csv** to download the user's credentials.
3. Click **Close**.

**Task 1.2: Set Permissions for Lab1User**

In this section of the lab, you will grant the appropriate permissions to your Lab1User IAM user.

1. From the **Users** menu in the **IAM Management Console** (if you are not already there), click the name of **Lab1User** to open the **Summary** page.
2. Click **Add permissions**.
3. Click **Attach existing policies directly**.
4. Locate the entry marked **AmazonS3FullAcess** under the **Policy name** column.

This policy grants the selected user full access to all Amazon S3 functions.

**Tip** To find a policy in the list of policies, you can use the **Search** box at the top of the list to locate any policy more easily. You can use partial names like "S3" to find any policies with "S3" in the name.

1. Locate the entry marked **AmazonEC2FullAccess** under the **Policy name** column. Select the check box for this entry.

This policy grants the selected user full access to all Amazon EC2 functions.

1. Locate the entry marked **AmazonVPCFullAccess** under the **Policy name** column. Select the check box for this entry.

This policy grants the selected user full access to all Amazon VPC functions.

1. Locate the entry marked **AmazonDynamoDBFullAccess** under the **Policy name** column. Select the check box for this entry. This policy grants the selected user full access to all Amazon DynamoDB functions.
2. Locate the entry marked **IAMReadOnlyAccess** under the **Policy name** column. Select the check box for this entry.

This policy grants the selected read-only access to all Amazon IAM resources.

1. Click **Next: Review**.
2. Verify that **AmazonS3FullAccess**, **AmazonEC2FullAccess**, **AmazonVPCFullAccess** , and **AmazonDynamoDBFullAccess**, **IAMReadOnlyAccess** are listed in the **Permissions summary**.
3. Click **Add permissions**.
4. Next we need to add an Inline Policy to allow the IAM Role to be assumed by a resource. At the bottom of the page, click **Add inline policy**.
5. You will use the **Policy Generator** to create a Custom Policy to attach to your IAM Role. Verify that the **Policy Generator** is selected and in the row for Policy Generator, click **Select**.
6. Configure the following attributes for the policy using the Policy Generator:

| **Effect** | **Allow** |
| --- | --- |
| AWS Service | AWS Identity and Access Management |
| Actions | PassRole |
| Amazon Resource Name (ARN) | \* |

1. Click **Add Statement** to create the policy.
2. Click **Next Step**.
3. Review the **Policy Document** that was generated and click **Apply Policy**.

**Task 1.3: Create an IAM Role**

In this part of the lab, you will create an **IAM Role** within **AWS IAM**. An IAM Role is an IAM entity that defines a set of permissions for making AWS service requests. IAM Roles are not associated with a specific user or group. Instead, roles are assumed by trusted entities such as IAM users, applications, or AWS services such as Amazon EC2.

You are going to create an ***AWS Service Role*** in **AWS IAM** and attach two **AWS Managed Policies** to the role. The role will be used to delegate permissions for a **Trusted Entity, Amazon EC1**. You will assign the role to an **Amazon EC2 Instance** so that the applications running on the instance can access **Amazon S3** and **Amazon DynamoDB**. This means that the Amazon EC2 Instance resource can access and manipulate objects in Amazon S3, and create tables and add items to Amazon DynamoDB.

1. On the AWS Management Console, on the **Services** menu, click **IAM**.
2. In the navigation pane, click **Roles**.
3. Click **Create New Role**.
4. On the **Select role type** page, under **AWS Service Roles, in the row for Amazon EC2**, click **Select**.
5. On the **Attach Policy** page, select the check box corresponding to the **AmazonS3FullAccess** policy
6. On the **Attach Policy** page, select the check box corresponding to the **AmazonDynamoDBFullAccess** policy
7. Click **Next Step**.
8. On the **Set Role Name** page, in the **Role Name** box, type 
9. Click **Create role**.

**Task 1.4: Switch to your IAM user and create a bucket**

In this section of the lab, you will switch over to Lab1User and verify that it has the appropriate permissions by creating the Amazon S3 bucket you will use for this lab.

1. In the navigation pane, click **Dashboard**.
2. Copy the entire URL displayed below **IAM users sign-in link**.

Paste the URL into a text file and save the file on your local computer so you can reference it later in the lab.

1. Open a new browser window or tab and navigate to the AWS Account Alias URL that you copied in the previous step.
2. For **User Name**, type 
3. For **Password**, paste the password for **Lab1User** that you saved in your **credentials** file.
4. Click **Sign In**.
5. Once the **AWS Management Console** opens, you will be logged out automatically from the first account you logged in with. You can now close your previous window or tab and use your **Lab1User** account for the rest of this lab.
6. On the **Services** menu, click **S3**.
7. Click **Create bucket**.
8. In the **Create a Bucket – Select a Bucket Name and Region** dialog box:

* For **Bucket name** , type a unique bucket name (e.g., a name with today's date or your initials inside of it, with no uppercase letters).
* For **Region**, click the region you want to create the bucket in. We recommend leaving it as the default region. For the region for your S3 bucket, you can use any region EXCEPT Ohio (us- east-2) or Mumbai (ap-south-1). Ohio and Mumbai are our two newest regions. As of January 30, 2014, all new AWS regions require Sigv4 signing for S3 requests and are not compatible with some of the commands you will use later in this lab.

Details on [S3 Regions Sig v4](http://docs.aws.amazon.com/AmazonS3/latest/API/sig-v4-authenticating-requests.html)

**Note** Bucket names must:

* Be unique across all existing bucket names in Amazon S3
* Be at least 3 characters and no more than 63 characters long
* Contain only lowercase letters, numbers, and hyphens.

1. Click **Create**. Verify that the bucket is created. This validates that your permissions for Amazon S3 are set properly.
2. Click the bucket you just created. Verify that the bucket is currently empty.

Task 2: Create the Network Layer

In this part of the lab, you will create a basic Virtual Private Cloud (VPC) with a single Public Subnet. You will attach an Internet Gateway (IGW) to the VPC to allow resources in the VPC to communicate across the Internet.

When you first sign in to the AWS Management Console and launch VPC Dashboard, you will notice that there is an existing VPC; this is the default VPC. A default VPC is a logically isolated virtual network in the AWS cloud that is automatically created for your AWS account the first time you provision Amazon EC2 resources. When you launch an instance without specifying a subnet ID, your instance will be launched in your default VPC.

In this part of the lab, you will create a VPC with a public subnet and a user-specified IP address range.

**Task 2.1: Create the VPC**

In this part of the lab, you will create a basic Virtual Private Cloud (VPC) with a single Public Subnet and a user-specified IP address range. You will attach an Internet Gateway (IGW) to the VPC to allow resources in the VPC to communicate across the Internet.

When you first sign in to the AWS Management Console and launch VPC Dashboard, you will notice that there is an existing VPC; this is the default VPC. A default VPC is a logically isolated virtual network in the AWS cloud that is automatically created for your AWS account the first time you provision Amazon EC2 resources. When you launch an instance without specifying a subnet ID, your instance will be launched in your default VPC.

1. On the **Services** menu, click **VPC**.

**Note:** You can select your desired region from the drop-down list on the navigation bar. For now, let it remain as the default.

1. In the navigation pane, click **Your VPCs**.
2. Click **Create VPC**.
3. In the **Create VPC** dialog box, enter the following settings:
   * For **Name tag**, type 
   * For **IPv4 CIDR block**, type 
   * For **Tenancy**, type 
4. Click **Yes, Create**.

You should see a new VPC named **LabVPC** with a **VPC ID** assigned to it (e.g., *vpc-530de336*).

**Task 2.2: Attach an Internet Gateway (IGW)**

In this section of the lab, you will create an Internet gateway and attach it to your VPC so that your VPC will be accessible via the Internet.

1. In the navigation pane, click **Internet Gateways**.
2. Click **Create Internet Gateway**.
3. For **Name tag**, type 
4. Click **Yes, Create**.  
   At this point, the newly created **LabVPCGateway** is not attached to your VPC. Note the ID (e.g., *igw-912a31f3*).
5. If it is not already selected, select the newly created **LabVPCGateway** , and then click **Attach to VPC**.
6. In the **Attach to VPC** dialog box, for **VPC**, click the **LabVPC** that you created in **Task 5**.
7. Click **Yes, Attach**.

The **State** for the **LabVPCGateway** should change to *attached*, and the VPC ID in the **VPC** column should match the ID for **LabVPC**.

Task 2.3: Create the Public Subnet

You have complete control over your virtual networking environment, including selection of your own IP address range and subnets. A **Subnet** is a segment of a VPCs IP address range where you can place groups of isolated resources.

In this task, you are going to configure your **Amazon VPC** with a single subnet so you can provision resources in the subnet and be able to provide access to the **Internet Gateway**. The direct attachment of a resource that connects to the Internet makes this subnet a **Public Subnet**.

1. In the navigation pane, click **Subnets**.
2. Click **Create Subnet**.
3. In the **Create Subnet** dialog box
   * For **Name tag**, type 
   * For **VPC**, click the VPC that includes the name **LabVPC.**
   * For **Availability Zone**, click the first AZ (e.g., *us-west-2a*).
   * For **CIDR block**, type 
4. Click **Yes, Create**.

Task 2.4: Configuring the Route Table

A route table contains a set of rules called routes that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can be associated with only one route table at a time, but you can associate multiple subnets with the same route table.

When you create a VPC, it automatically has a main route table. Initially, the main route table contains only a single route: a local route that enables communication within the VPC. If you don't explicitly associate a subnet with a route table, the subnet is implicitly associated with the main route table.

In this section of the lab, you will create a route table that allows incoming and outgoing traffic through the Internet gateway you created earlier.

1. In the navigation pane, click **Route Tables**.
2. Click **Create Route Table**.
3. In the **Create Route Table** dialog box:
   * For **Name tag**, type 
   * For **VPC**, type 
4. Click **Yes, Create**.
5. If it is not already selected, select the **PublicRoute** route table you just created, and then click the **Routes** tab in the lower pane of the console.
6. Click **Edit**.
7. Click **Add another route**.
8. For **Destination** , type 
9. Click in the **Target** box, and then click the **LabVPCGateway** that you created earlier (the ID starts with *igw-*).
10. Click **Save**.
11. With **PublicRoute** still selected, click the **Subnet Associations** tab.
12. Click **Edit**.
13. Select **PublicSubnet1** (CIDR range of 10.200.10.0/24).
14. Click **Save**.

Task 3: Build the Compute, Storage and Database Layer

In this part of the lab, you will create an **Amazon Elastic Compute Cloud (EC2)** Instance that will function as the web server for your web application. This box will communicate with the **S3 Bucket** you built in Task 1 and with an **Amazon DynamoDB** table that will be created in this task. These three services will provide the web application with compute, storage, and database services.

**Task 3.1: Create the Web Server**

In this section of the lab, you will create an Amazon EC2 instance in PublicSubnet1

1. On the **Services** menu, click **EC2**.
2. Click **Launch Instance**.
3. To launch a new instance, you first need to select an Amazon Machine Image (AMI), which is a preconfigured template for an instance in the cloud.
4. From the **Quick Start** menu, in the row for the first **Amazon Linux AMI**, click **Select**.
5. On the **Choose an Instance Type** page, you can select the family for your image, which determines how much RAM, storage, and processing speed your instance will have.
6. To accept the default (**t2.micro**), click **Next: Configure Instance Details**.
7. On the **Configure Instance Details** page, make these selections
   * For **Network**, type 
   * For **Subnet**, type 
   * For **Auto-assign Public IP**, click **Enable**.
   * For **IAM role**, click **WebServerRole** (which you created in Task 1.3)
8. Click **Advanced Details** to expand it. Copy the contents of the user data script given below, and paste the script into the **User data** box.

Copy Code Block

#!/bin/bash

yum remove -y httpd php

yum install -y httpd24 php56

chkconfig httpd on

wget https://d2lrzjb0vjvpn5.cloudfront.net/AWS-100-ARC/v5.0/lab-1-webapp/scripts/lab1src.zip

unzip lab1src.zip -d /home/ec2-user/

mv /home/ec2-user/lab1src/index.php /var/www/html/index.php

mv /home/ec2-user/lab1src/challenge-me.php /var/www/html/challenge-me.php

wget https://github.com/aws/aws-sdk-php/releases/download/3.15.9/aws.zip

unzip aws -d /var/www/html

service httpd start

This Linux script installs a specific httpd and PHP package, downloads the source code, images, and scripts for the web application, pulls down the AWS-SDK for PHP and starts the httpd service.

1. Click **Next: Add Storage**.  
   You won't be using the storage on this instance, so you are leaving the instance's storage settings as their default.
2. Click **Next: Add Tags**.
3. Click **Click to add a Name tag**.
4. For **Value**, type 
5. Click **Next: Configure Security Group**.
6. For **Assign a security group**, verify that the **Create a new security group** option is selected.
   * For **Security group name**, type 
   * For **Description**, type 
7. There should be an existing **SSH** rule. Leave that rule as-is.
8. Click **Add Rule**.
9. In the new row that appears, specify the following:
   * For **Type**, click **HTTP**.
   * For **Source**, click **Anywhere**.
10. Click **Review and Launch**.
11. Review the settings and then click **Launch**.
12. When prompted, accept the *qwikLABS* keypair, select the acknowledgement check box, and then click **Launch Instances**.
13. Click **View Instances**.
14. Select the **WebServer** instance you just created.
15. On the **Description** tab in the lower pane, note the **IPv4 Public IP** of the instance.
16. Wait for the **WebServer** to reach **Instance State:** *running* and **Status Checks:** *2/2 checks passed.*

Task 3.2: Connect to WebServer (Windows only)

**Note** This section is for **Windows** users only. If you are running OSX or Linux, skip to **Task 3.3**.

In this section of the lab, you will download your keypair and use it to connect to your Amazon EC2 instance with PuTTY.

1. From the qwikLABS page in your browser, in the **Connect** section, for **Download PEM/PPK** , click **Download PPK**.
2. Save the file to your **\Downloads** folder or any other easy to access location on your local computer.
3. Download **PuTTY** from <https://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>
4. Launch **PuTTY** by running the **putty.exe** file you just downloaded.
5. For **Host Name** , enter the **Public IP** address from your Web Server instance which you copied into a text editor earlier in the lab.
6. In the **Connection** list, expand **SSH**.
7. Click **Auth**.
8. For **Private key file for authentication** , browse to the **.ppk** file that you downloaded earlier, then click **Open**.
9. In the **PuTTY Security Alert** dialog box that opens, click **Yes** to add the key to **PuTTY**'s cache.
10. For **login as:** type and press ENTER. You are now logged in to your **Web Server** instance.

Task 3.3: Connect to the WebServer (Linux/OSX only)

**Note** This section is for **Linux** and **Mac OSX** users only. If you are running **Windows** but have not yet connected to your instance, go back to **Task 3.2**. If you have already connected to your instance, skip ahead to **Task 3.4**.

In this section of the lab, you will download your keypair and use it to connect to your Amazon EC2 instance.

1. From the qwikLABS page in your browser, in the **Connect** section, for **Download PEM/PPK** , click **Download PEM**.
2. Save the file to your local computer in a place where you can easily access it.
3. To connect to your EC2 instance, run the following commands in Terminal

Copy Code Block

chmod 400 <path and name of pem>

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

1. For **<path and name of pem>**, substitute the path/filename to the .pem file you downloaded.
2. For **<Public IP>**, substitute the public IP address for your **Web Server** instance which you copied into a text editor earlier in the lab.

Task 3.4: Configure Your Server and the DynamoDB Table

In this part of the lab, you will use commands to set the bucket and region variables in your site's index.php file, and then create and populate a DynamoDB table which will hold your Items. This table will include the text content for services and acronyms and an object key for the image that is stored in S3.

The images and scripts have been downloaded to the **/home/ec2-user/lab1src** directory.

The web page (index.php) is located in **/var/www/html**.

**Commands usage**

You will find commands that need to be used in this task. To make sure they are configured correctly, you will need to replace some of the text according to the following steps:

* Replace all instances of with the name of the bucket you just created. **Be sure to remove the angle brackets (< >) as well**.
* Replace every instance of with your region code. Be sure to remove the angle brackets (< >) as well. In the **AWS Management Console**, in the upper-right corner, to the right of your username, you will see the name of the region where your lab is running. Use this table to find your region's code:

| **Region Name** | **Region Code** |
| --- | --- |
| N. Virginia | us-east-1 |
| N. California | us-west-1 |
| Oregon | us-west-2 |
| Ireland | eu-west-1 |
| Tokyo | ap-northeast-1 |
| Seoul | ap-northeast-2 |
| Singapore | ap-southeast-1 |
| Sydney | ap-southeast-2 |
| São Paulo | sa-east-1 |

1. Verify that your S3 bucket is currently empty and that your Instance is able to access the bucket (through the IAM Role) by executing the following command from the AWS CLI:

Copy Code Block

aws s3 ls <bucket\_name> --region <region>

1. Verify that you are working out of the /home/ec2-user/lab1src directory by executing the following command:

Copy Code Block

cd /home/ec2-user/lab1src

1. Copy all static website resources (images, JavaScript files, CSS stylesheets) to the S3 bucket by executing the following command:

Copy Code Block

aws s3 cp /home/ec2-user/lab1src/jquery/ s3://<bucket\_name>/jquery/ --recursive --acl public-read --region <region>

1. Copy all static icon images up to the S3 bucket by executing the following command:

Copy Code Block

aws s3 cp /home/ec2-user/lab1src/images/ s3://<bucket\_name>/images/ --recursive --acl public-read --region <region>

1. Verify that all objects were copied up to the S3 bucket by executing the following command:

Copy Code Block

aws s3 ls s3://<bucket\_name>/ --region <region> --recursive

1. Execute the following commands one at a time to set the values for the region and bucket variables in your index.php file.

Copy Code Block

sudo sed -i '2s/%region%/<region>/g' /var/www/html/\*.php

sudo sed -i '3s/%bucket%/<bucket\_name>/g' /var/www/html/\*.php

1. Return to the AWS console and navigate to **DynamoDB** using the **Services** menu at the top of the page.
2. Click **Create table**.
3. For **Table name** , type 

**Note** Make sure you type the table name exactly as it appears above, including case.

1. In the text box below **Partition key**, type 

**Note** Make sure you type the table name exactly as it appears above, including case.

1. Select **Add sort key**.
2. In the new text box that appears, type 

**Note** Make sure you type the table name exactly as it appears above, including case.

1. Ensure **Use default settings** is selected and click **Create**.
2. Return to your SSH session. There you will run the commands listed in the next steps in order to execute a bulk write of items to your DynamoDB table.
3. Populate the DynamoDB table by running the following three commands one at a time:

Copy Code Block

aws dynamodb batch-write-item --request-items file:///home/ec2-user/lab1src/scripts/services1.json --region <region>

aws dynamodb batch-write-item --request-items file:///home/ec2-user/lab1src/scripts/services2.json --region <region>

aws dynamodb batch-write-item --request-items file:///home/ec2-user/lab1src/scripts/services3.json --region <region>

After running each command, you should receive the following output:

Copy Code Block

{

"UnprocessedItems": {}

}

1. Return to the console and refresh the **AWS-Services** table's **Items** page to verify that the items have populated.

**TIP** You can also scan the table using the command line by executing the following:

Copy Code Block

aws dynamodb scan --table-name AWS-Services --region <region>

1. Return to your web browser and reload the web page using your web server's public IP as the URL. You should see the listing of services from DynamoDB, the service icons from S3, and a formatted page using the javascript and css resources that we also stored in S3.

Challenge

In the top right corner of the webpage you created, click the **challenge me** button. You will be redirected to a page that is also hosted by your webserver and uses the resources that you loaded into your S3 bucket and the DynamoDB items you wrote to your table. Take the challenge by dragging the Services listed in the left container and drop them in the correct Service category in the right container. Good luck!

Lab Complete

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

1. To sign out of the **AWS Management Console** click **Lab1User** in the navigation bar, and then click **Sign Out**.
2. Return to the **qwikLABS** page where you launched your lab and click **End**.

© for the use of: shweta\_misra@fanniemae.com

* Connect
* Lab Details
* Addl. Info
* **Lab Connection**   
  Please follow the lab instructions to connect to your lab

**Warning:** Do not transmit data into the AWS Console that is not related to Qwiklabs or the lab you are taking.

* + **AWS Console Details**
  + **Access Key Details**
  + [Show Access Keys](https://globalknowledge.qwiklab.com/focuses/17007#accessKeyIdModal)
  + **Key Pair Details**

|  |  |
| --- | --- |
| **Setup Time (min.)** | 0 |
| **Duration (min.)** | 180 |
| **Access (min.)** | 180 |

|  |
| --- |
| **Tags:** [**AWS-100-ARC-51-lab-1-webapp**](https://globalknowledge.qwiklab.com/tags/AWS-100-ARC-51-lab-1-webapp)  **Lab Description:** In this lab you will use IAM, EC2, S3, VPC and DynamoDB to deploy a scalable web application.  **Lab Creator:** qwikLABS publisher  **Date Created:** May 13, 2017 02:23  **AWS Region::** [us-west-2] **US West (Oregon)** |

Additional Lab Information:

×

**Access Key Details**

Use access keys to make secure REST or Query protocol requests to any AWS service API.

Close