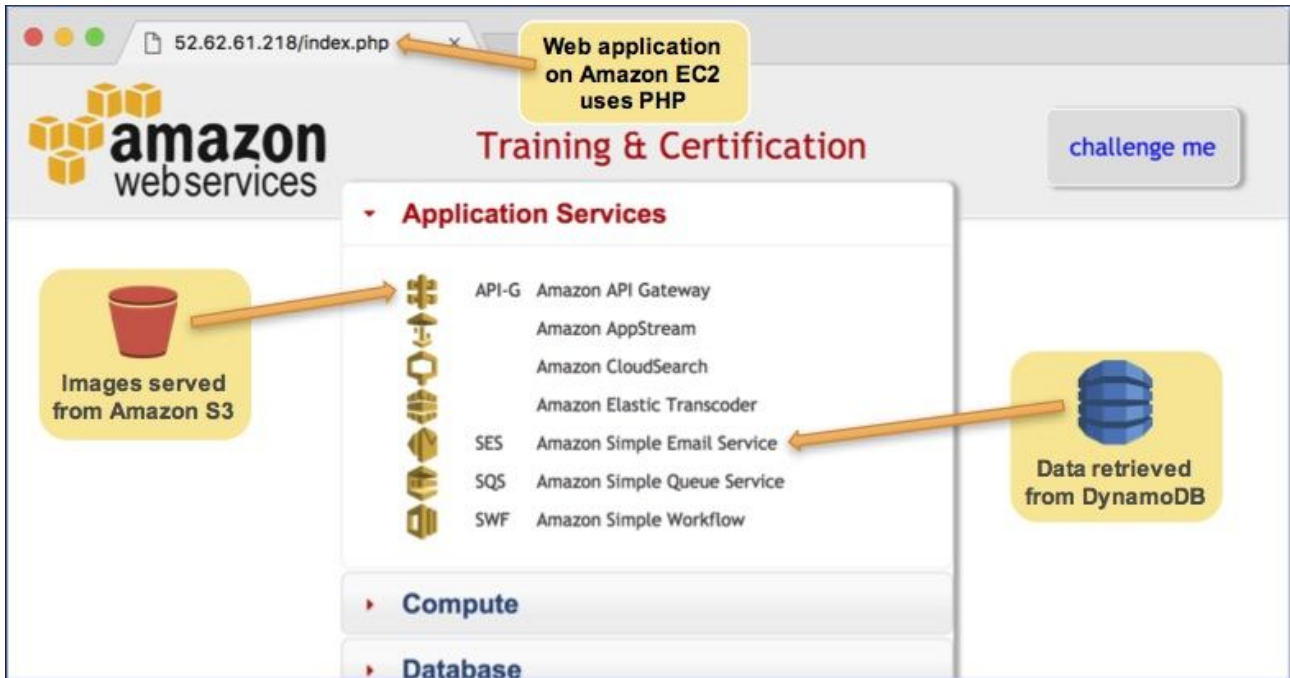


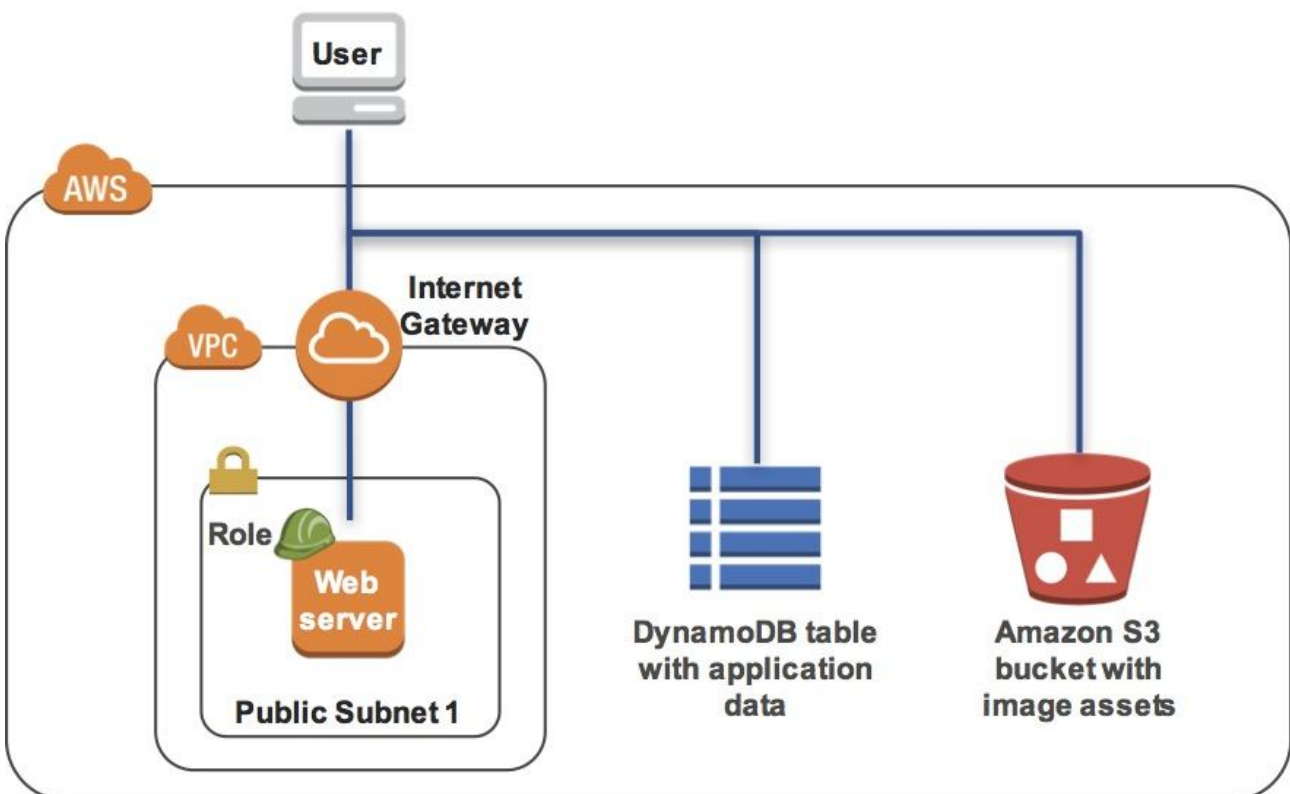
# AWS 상에 웹 애플리케이션 배포

다중 AWS 서비스를 사용하여 웹 애플리케이션을 배포하는 실습입니다.

- 웹 애플리케이션은 PHP를 사용하여 **Amazon EC2** 인스턴스에서 호스팅 됩니다.
- 데이터는 **Amazon DynamoDB** 테이블에 저장됩니다.
- 이미지는 **Amazon S3**에서 제공됩니다.



다음의 애플리케이션 환경을 배포합니다.



본 실습을 완료하면 다음과 같은 작업을 수행할 수 있습니다.

- Amazon EC2 인스턴스에서 사용할 **IAM Role** 생성
- **Amazon S3** 버킷 생성
- **Amazon DynamoDB** 테이블 생성
- 인터넷 게이트웨이와 퍼블릭 서브넷을 갖춘 **Amazon Virtual Private Cloud (VPC)** 생성
- **Amazon EC2** 상에 웹 애플리케이션 배포

## 소요 시간

본 실습에는 약 **30분**이 소요됩니다.

---

# AWS Management Console 액세스

---

1. [1]실습 제목 오른쪽에서 **Start Lab**을 클릭하여 Qwiklabs를 시작합니다.

Start Lab

1. [2]Qwiklabs 페이지의 **Connect** 탭에서 **Password**를 클립보드로 복사한 후 **Open Console**을 클릭합니다.

Open Console

1. [3]다음 단계에 따라 AWS Management Console에 로그인합니다.

- **User Name**에 'awsstudent'를 입력합니다.
- **Password**에 클립보드에서 복사한 암호를 붙여넣습니다.
- **Sign In**을 클릭합니다.

---

## 작업 1: IAM 역할 생성

---

이번 작업에서는 **IAM Role**을 생성하고 Amazon S3 및 Amazon DynamoDB에 액세스할 수 있는 권한을 부여합니다. 이 역할은 추후 웹 애플리케이션을 실행할 Amazon EC2 인스턴스에 할당됩니다.

AWS 서비스를 호출할 권한이 있다는 점에서 **IAM Role**은 사용자와 비슷합니다. 그렇지만, 1인 전용이 아니기 때문에 인증된 사용자나 서비스, 애플리케이션도 역할을 *맡을* 수 있습니다. 이 작업이 완료되면 AWS 서비스에 대한 API 호출에 사용할 수 있는 **액세스 자격 증명**이 동적으로 생성됩니다. 이것이 Amazon EC2 인스턴스에서 실행되는 애플리케이션이 AWS 서비스(예: Amazon S3 및 Amazon DynamoDB)를 호출할 수 있는 자격 증명을 획득하는 방식입니다.

1. [4]**AWS Management Console**의 **Services** 메뉴에서 **IAM**을 클릭합니다.
2. [5]왼쪽 탐색 창에서 **Roles**를 클릭합니다.
3. [6]**Create new role**을 클릭합니다.
4. [7]**Amazon EC2** 행에서 **Select**를 클릭합니다.

이는 Amazon EC2 인스턴스가 역할을 사용한다는 것을 나타냅니다.

1. [8]목록(필터 상자를 사용하여 빠르게 찾을 수 있음)에서 이 두 역할을 선택(체크)합니다.

- 'AmazonS3FullAccess'
- 'AmazonDynamoDBFullAccess'(첫 번째로 표시되는 것을 선택)

1. [9]**Next Step**을 클릭합니다.
2. [10]**Role name**에 'WebServerRole'을 입력합니다.
3. [11]**Create role**을 클릭합니다.

이 역할은 추후 실습에서 Amazon EC2 인스턴스를 시작할 때 사용합니다. 그다음에는 해당 인스턴스에서 실행 중인 웹 애플리케이션에 Amazon S3 및 Amazon DynamoDB 사용 권한이 부여됩니다.

---

## 작업 2: Amazon S3 버킷 생성

이번 작업에서는 **Amazon S3 버킷**을 생성합니다. 웹 애플리케이션에서 이 버킷을 사용하여 웹 페이지의 이미지 및 기타 정적 파일을 제공합니다.

1. [12]**Services** 메뉴에서 **S3**를 클릭합니다.
2. [13]**Create bucket**을 클릭합니다.
3. [14]**Bucket name**에 'webapp-'으로 시작하는 고유한 버킷 이름을 입력합니다.  
예를 들어, *webapp-cb35* 처럼 머리글자와 숫자를 조합하여 덧붙입니다.

버킷 이름은 반드시 'webapp-'로 시작해야 합니다. 그렇지 않으면 애플리케이션이 버킷을 찾지 못합니다.

1. [15]**Create**를 클릭합니다.

추후 웹 애플리케이션이 이 버킷에 파일을 복사하게 됩니다.

---

## 작업 3: Amazon DynamoDB 테이블 생성

이번 작업에서는 웹 애플리케이션 데이터를 보유하게 될 **Amazon DynamoDB 테이블**을 생성합니다.

1. [16]**Services** 메뉴에서 **DynamoDB**를 클릭합니다.
2. [17]**Create table**을 클릭합니다.
3. [18]**Table name**에 'AWS-Services'를 입력합니다.

반드시 위의 테이블 이름을 대소문자까지 정확하게 입력하십시오!

1. [19]**Partition key** 아래의 텍스트 상자에 'Category'를 입력합니다.

반드시 위의 이름을 대소문자까지 정확하게 입력하십시오!

1. [20]**Add sort key**를 선택합니다.
2. [21]새로운 텍스트 상자가 표시되면, 'Name'을 입력합니다.

반드시 위의 이름을 대소문자까지 정확하게 입력하십시오!

1. [22]**Create**를 클릭합니다.

이제 DynamoDB 테이블이 생성되었습니다. 추후 웹 애플리케이션이 이 테이블에 데이터를 복사하게 됩니다.

다음 작업으로 진행할 수 있습니다. 테이블이 백그라운드에서 생성됩니다.

## 작업 4: Amazon VPC 생성

이번 작업에서는 단일 퍼블릭 서브넷이 있는 기본 **Amazon Virtual Private Cloud(VPC)**를 생성합니다. VPC의 리소스가 인터넷과 통신할 수 있도록 인터넷 게이트웨이를 VPC에 연결합니다.

### 작업 4.1: VPC 생성

1. [23]**Services** 메뉴에서 **VPC**를 클릭합니다.
2. [24]왼쪽 탐색 창에서 **Your VPCs**를 클릭합니다.
3. [25]**Create VPC**를 클릭합니다.
4. [26]**Create VPC** 대화 상자에서 다음과 같이 설정합니다(목록에 없는 설정은 모두 무시).

<b>Name tag</b>	'Lab VPC'
<b>IPv4 CIDR block</b>	'10.200.0.0/16'

1. [27]**Yes, Create**를 클릭합니다.

**10.200.0.0/16** CIDR("Classless Inter-Domain Routing") 블록은 VPC의 IP 주소 범위를 정의합니다. 끝부분의 **/16**은 블록에 **10.200.\*.\*** 주소와 일치하는 모든 IP 주소가 포함된다는 의미입니다.

### 작업 4.2: 서브넷 생성

**Subnet**은 VPC의 세그먼트로 Amazon EC2 인스턴스와 같은 리소스를 포함합니다. 각 서브넷은 VPC에 할당된 IP 주소 범위의 하위 집합을 수신합니다.

1. [28]탐색 창에서 **Subnets**를 클릭합니다.
2. [29]**Create Subnet**을 클릭합니다.
3. [30]**Create Subnet** 대화 상자에서 다음과 같이 설정합니다(목록에 없는 설정은 모두 무시).

<b>Name tag</b>	'Public Subnet 1'
<b>VPC</b>	<b>Lab VPC</b>
<b>Availability Zone</b>	목록의 첫 번째 영역 선택
<b>IPv4 CIDR block</b>	'10.200.10.0/24'

1. [31]**Yes, Create**를 클릭합니다.

**퍼블릭 서브넷**은 인터넷과 통신할 수 있는 서브넷입니다. 공개적으로 액세스할 수 있어야 하는 리소스를 배치할 수 있는 영역입니다. 서브넷 이름이 **Public Subnet 1**이지만 아직 퍼블릭 서브넷은 아닙니다. 퍼

블릭 서브넷으로 만들려면 VPC를 인터넷 게이트웨이에 연결해야 합니다.

## 작업 4.3: 인터넷 게이트웨이 생성

VPC와 인터넷 간의 인터페이스인 **Internet Gateway**를 생성합니다.

1. [32]탐색 창에서 **Internet Gateways**를 클릭합니다.
2. [33]**Create Internet Gateway**를 클릭합니다.
3. [34]**Name tag**에 'Lab Gateway'를 입력합니다.
4. [35]**Yes, Create**를 클릭합니다.
5. [36]새로 생성된 **Lab Gateway**를 선택하고 **Attach to VPC**를 클릭합니다.
6. [37]**Attach to VPC** 대화 상자의 **VPC**에서 **Lab VPC**를 선택합니다.
7. [38]**Yes, Attach**를 클릭합니다.

이제 인터넷 게이트웨이가 VPC에 연결되었지만, 아직 사용할 수는 없습니다. 라우팅 테이블을 구성하여 네트워크 트래픽 경로를 인터넷 게이트웨이로 지정해야 합니다.

## 작업 4.4: 퍼블릭 라우팅 테이블 생성

**Route Table**에는 **Routes**라는 규칙 세트가 포함되어 있습니다. **Routes**는 서브넷으로 들어오고 나가는 네트워크 트래픽 경로를 지정합니다. 새로운 VPC는 VPC 내부의 네트워크 트래픽만 라우팅하는 라우팅 테이블을 자동으로 수신합니다. 이를 *프라이빗 라우팅 테이블*이라 합니다.

인터넷 게이트웨이를 통해 리소스를 인터넷과 통신하도록 해주는 *퍼블릭 라우팅 테이블*을 생성합니다.

1. [39]탐색 창에서 **Route Tables**를 클릭합니다.
2. [40]**Create Route Table**을 클릭합니다.
3. [41]**Create Route Table** 대화 상자에서 다음과 같이 설정합니다(목록에 없는 설정은 모두 무시).

<b>Name tag</b>	'Public Route Table'
<b>VPC</b>	<b>Lab VPC</b>

1. [42]**Yes, Create**를 클릭합니다.
2. [43]새로 생성된 **Public Route Table**을 선택하고 콘솔 아래쪽 창에서 **Routes** 탭을 클릭합니다. (아래쪽 창을 위로 드래그하여 확대할 수 있습니다.)
3. [44]**Edit**를 클릭합니다.
4. [45]**Add another route**를 클릭합니다.

인터넷 게이트웨이를 통해 인터넷 트래픽(와일드 카드 **0.0.0.0/0**으로 식별)을 보내도록 라우팅 테이블을 구성합니다.

1. [46]**Destination**에 '0.0.0.0/0'을 입력합니다.
2. [47]**Target** 상자를 클릭하고 이전에 생성한 **Lab Gateway**(*igw-*로 시작하는 ID)를 클릭합니다.
3. [48]**Save**를 클릭합니다.

이제 이전에 생성한 서브넷에 *퍼블릭 라우팅 테이블*을 연결할 수 있습니다.

1. [49]**Public Route Table**이 선택된 상태에서 화면 아래쪽의 **Subnet Associations** 탭을 클릭합니다.
2. [50]**Edit**를 클릭합니다.
3. [51]**Public Subnet 1** 옆의 체크 박스를 선택(체크)합니다.
4. [52]**Save**를 클릭합니다.

이제 서브넷이 *퍼블릭 서브넷*이 되었습니다. 즉, 서브넷의 리소스가 인터넷과 통신할 수 있다는 뜻입니다.

## 작업 5: Amazon EC2 인스턴스 시작

이번 작업에서는 퍼블릭 서브넷에서 **Amazon EC2 인스턴스**를 생성합니다. *인스턴스*라는 용어는 클라우드에서 실행되는 가상 머신을 나타냅니다. 시작 시 웹 애플리케이션을 설치하도록 인스턴스를 구성합니다.

애플리케이션은 **Amazon S3 Bucket** 및 **Amazon DynamoDB table**과 통신하게 됩니다. 이 세 가지 서비스는 웹 애플리케이션에 컴퓨팅, 스토리지 및 데이터베이스 서비스를 제공합니다.

1. [53]**Services** 메뉴에서 **EC2**를 클릭합니다.
2. [54]**Launch Instance**를 클릭합니다.

새 인스턴스를 시작할 때, 클라우드 내에 있는 인스턴스용 사전 구성 템플릿인 Amazon 머신 이미지(AMI)를 선택해야 합니다. Linux 운영 체제가 포함된 이미지를 선택합니다.

1. [55]첫 번째 **Amazon Linux AMI** 행에서 **Select**를 클릭합니다.

**Choose an Instance Type** 페이지에서 인스턴스에 할당할 RAM과 CPU의 양을 결정하는 *인스턴스 유형*을 선택할 수 있습니다.

1. [56]**Next: Configure Instance Details**를 클릭하여 기본값(**t2.micro**)을 적용합니다.
2. [57]**Configure Instance Details** 페이지에서 다음과 같이 설정합니다(목록에 없는 설정은 모두 무시).

<b>Network</b>	<b>Lab VPC</b>
<b>Subnet</b>	<b>Public Subnet 1</b>
<b>Auto-assign Public IP</b>	<b>Enable</b>
<b>IAM role</b>	<b>WebServerRole</b>

위의 설정을 모두 정확하게 선택했는지 확인하십시오. 설정이 잘못되면 웹 애플리케이션이 제대로 작동하지 않습니다.

1. [58]**Advanced Details**를 클릭하여 확장합니다(아래로 스크롤해야 할 수 있음).

**User Data** 필드에 스크립트가 제공되면 Amazon EC2 인스턴스가 처음 시작될 때 자동으로 스크립트를 실행합니다. 소프트웨어를 설치하고 인스턴스를 구성하는 간편한 방법입니다.

웹 애플리케이션을 자동으로 로드하고 구성하는 스크립트를 사용하게 됩니다. 이런 유형의 스크립트는 정확한 버전 추적 및 저장을 위해 일반적으로 Amazon CodeCommit과 같은 소스 코드 리포지토리에서 유지 관리됩니다.

1. [59]아래의 **Copy Code Block**을 클릭하고 **User Data** 필드에 붙여넣습니다.

```
#!/bin/bash
# Apache 웹 서버 및 PHP 설치
yum remove -y httpd php
yum install -y httpd24 php56
# 실습 파일 다운로드
wget https://us-west-2-aws-staging.s3.amazonaws.com/awsu-ilt/AWS-100-ARC/v5.2/lab-1-webapp/script
unzip lab1src.zip -d /tmp/
mv /tmp/lab1src/*.php /var/www/html/
# PHP용 AWS SDK 다운로드 및 설치
wget https://github.com/aws/aws-sdk-php/releases/download/3.15.9/aws.zip
unzip aws -d /var/www/html
# 리전 결정
AZ=$(curl --silent http://169.254.169.254/latest/meta-data/placement/availability-zone/)
REGION=${AZ::-1}
# 이름이 webapp-*인 Amazon S3 버킷에 파일 복사
BUCKET=$(aws s3api list-buckets --query "Buckets[?starts_with(Name, 'webapp-')].Name | [0]" --outp
aws s3 cp /tmp/lab1src/jquery/ s3://$BUCKET/jquery/ --recursive --acl public-read --region $REGIO
aws s3 cp /tmp/lab1src/images/ s3://$BUCKET/images/ --recursive --acl public-read --region $REGIO
aws s3 ls s3://$BUCKET/ --region $REGION --recursive
# 사용할 리전 및 버킷 구성
sed -i "2s/%region%/$REGION/g" /var/www/html/*.php
sed -i "3s/%bucket%/$BUCKET/g" /var/www/html/*.php
# DynamoDB 테이블에 데이터 복사
aws dynamodb batch-write-item --request-items file:///tmp/lab1src/scripts/services1.json --region
aws dynamodb batch-write-item --request-items file:///tmp/lab1src/scripts/services2.json --region
aws dynamodb batch-write-item --request-items file:///tmp/lab1src/scripts/services3.json --region
# 웹 서버 실행
chkconfig httpd on
service httpd start
```

스크립트를 검사합니다. 다음 단계를 수행합니다.

- Apache 웹 서버(httpd) 및 PHP 언어 설치
- 웹 애플리케이션용 스크립트가 포함된 파일 다운로드 및 압축 해제
- PHP용 AWS SDK 다운로드 및 설치
- 이름이 *webapp*-으로 시작하는 Amazon S3 버킷에 파일 복사
- DynamoDB 테이블에 데이터 복사
- 웹 서버 실행

사용자 데이터 스크립트는 인스턴스를 처음 시작할 때만 실행되므로 초기 데이터를 로드하고 인스턴스를 구성하기에 이상적인 방법입니다.

1. [60]**Next: Add Storage**를 클릭합니다.

이 인스턴스에서는 추가 스토리지가 필요 없으므로, 설정은 기본값으로 유지합니다.

1. [61]**Next: Add Tags**를 클릭합니다.

태그는 인스턴스와 관련된 *메타데이터*입니다. 태그는 인스턴스 이름 지정 및 비용 센터 연결, 목적 식별에 사용할 수 있습니다.

1. [62]**Add Tag**를 클릭합니다.
2. [63]**Key** 아래에 'Name'을 입력합니다.
3. [64]**Value** 아래에 'Web Server'를 입력합니다.
4. [65]**Next: Configure Security Group**을 클릭합니다.

이제 *보안 그룹*을 정의합니다. 보안 그룹은 인스턴스로 가는 트래픽을 제어하는 가상 방화벽 역할을 합니다. 인스턴스는 웹 서버가 되므로 포트 80(HTTP)을 통한 액세스를 허용해야 합니다.

1. [66]다음과 같이 설정합니다.

<b>Security group name</b>	'Web Server'
<b>Description</b>	'Web Server Security Group'

기존 **SSH** 규칙이 있을 것입니다. 해당 규칙은 그대로 둡니다.

1. [67]**Add Rule**을 클릭합니다.
2. [68]표시되는 *새로운 행*에서 다음과 같이 설정합니다.

<b>Type</b>	<b>HTTP</b>
<b>Source</b>	<b>Anywhere</b>

1. [69]**Review and Launch**를 클릭합니다.
2. [70]설정을 확인한 후 **Launch**를 클릭합니다.
3. [71]메시지가 표시되면, *qwikLABS* 키 페어를 수락하고, 승인 확인란을 선택한 후, **Launch Instances**를 클릭합니다.
4. [72]**View Instances**를 클릭합니다(아래로 스크롤해야 할 수 있음).

Amazon EC2 인스턴스를 시작하는 데 몇 분 정도 소요됩니다. 시작 후 *사용자 데이터 스크립트*를 실행하여 소프트웨어를 설치하고 웹 애플리케이션을 활성화합니다.

1. [73]**Status Checks: 2/2 checks passed**가 표시될 때까지 **Web Server**를 기다립니다.

이 작업은 몇 분 정도 소요됩니다. 가끔 오른쪽 위 모서리의 새로 고침 아이콘을 클릭할 수 있습니다.

이제 인스턴스에 액세스할 준비가 되었습니다!

## 작업 6: 리소스 검사

이번 작업에서는 웹 애플리케이션에 액세스하고 사용자 데이터 스크립트가 로드한 리소스를 검사합니다.

1. [74]**Web Server** 인스턴스를 선택합니다.



2. [75]아래쪽 창에 있는 **Description** 탭에서 **IPv4 Public IP**를 클립보드로 복사합니다.
3. [76]새로운 브라우저 탭을 열고 클립보드의 IP 주소를 붙여넣은 다음 엔터를 누릅니다.

웹 애플리케이션이 표시될 것입니다.

1. [77]웹 애플리케이션에서 아무 카테고리나 클릭하면 서비스 아이콘이 나타납니다.

다음과 같은 내용이 보일 것입니다.

- 기존에 생성한 **DynamoDB 테이블**에서 검색된 서비스 목록
- 기존에 생성한 **Amazon S3 버킷**에서 제공되는 서비스 아이콘

웹 애플리케이션이 작동하지 않거나 이미지가 정상적으로 표시되지 않으면 강사의 지원을 받아 구성을 진단하십시오.

웹 애플리케이션에 표시되는 이미지는 Amazon S3 버킷에서 직접 제공됩니다.

1. [78]아이콘을 우클릭하고 "이미지 주소 복사"(브라우저에 따라 이름이 다를 수 있음)를 선택합니다.
2. [79]**new browser tab**에 링크를 붙여넣고 URL이 Amazon S3 버킷을 참조하는지 확인합니다.  
URL은 다음과 같은 형식이어야 합니다. `webapp-cb35.s3.amazonaws.com/images/Developer-Tools_AWSCodeDeploy.png`

서비스 이름은 Amazon DynamoDB 테이블에서 검색되었습니다.

1. [80]브라우저에서 AWS Management Console 탭으로 돌아갑니다.
2. [81]**Services** 메뉴에서 **DynamoDB**를 클릭합니다.
3. [82]탐색 창에서 **Tables**를 클릭합니다.
4. [83]**AWS-Services** 테이블을 클릭합니다.
5. [84]**Items** 탭을 클릭합니다.

DynamoDB 테이블의 항목들이 표시됩니다. 인스턴스를 시작했을 때 사용자 데이터 스크립트가 로드한 항목들입니다.

## 선택 사항: 도전 퀴즈를 풀어보세요!

1. [85]웹 애플리케이션으로 돌아갑니다. 웹 페이지의 오른쪽 위 모서리에서 **challenge me** 버튼을 클릭합니다.

웹 애플리케이션에는 Amazon S3 버킷의 리소스와 DynamoDB 테이블의 항목을 사용한 퀴즈가 포함되어 있습니다. 왼쪽 컨테이너에 나열된 서비스를 오른쪽 컨테이너에 있는 올바른 서비스 범주로 끌어다 놓는 것이 도전 과제입니다. 행운을 빕니다!

(**힌트:** 색깔별로 분류할 수 있습니다!)

---

## 실습 완료

축하합니다! 실습을 완료했습니다.

본 실습용 qwikLABS 페이지로 돌아가 **End**를 클릭하여 실습 환경을 종료해 주십시오.

