

고가용성 환경 만들기

중요 비즈니스 시스템은 고가용성 애플리케이션으로 배포되어야 합니다. 즉, 일부 구성 요소에 오류가 발생해도 작동해야 한다는 의미입니다. AWS에서 고가용성을 실현하려면 여러 가용 영역에 걸쳐 서비스를 실행하는 것이 좋습니다.

로드 밸런서처럼, 많은 AWS 서비스가 기본적으로 고가용성이거나, 여러 가용 영역에 Amazon EC2 인스턴스를 배포하는 식으로 고가용성 구성이 가능합니다.

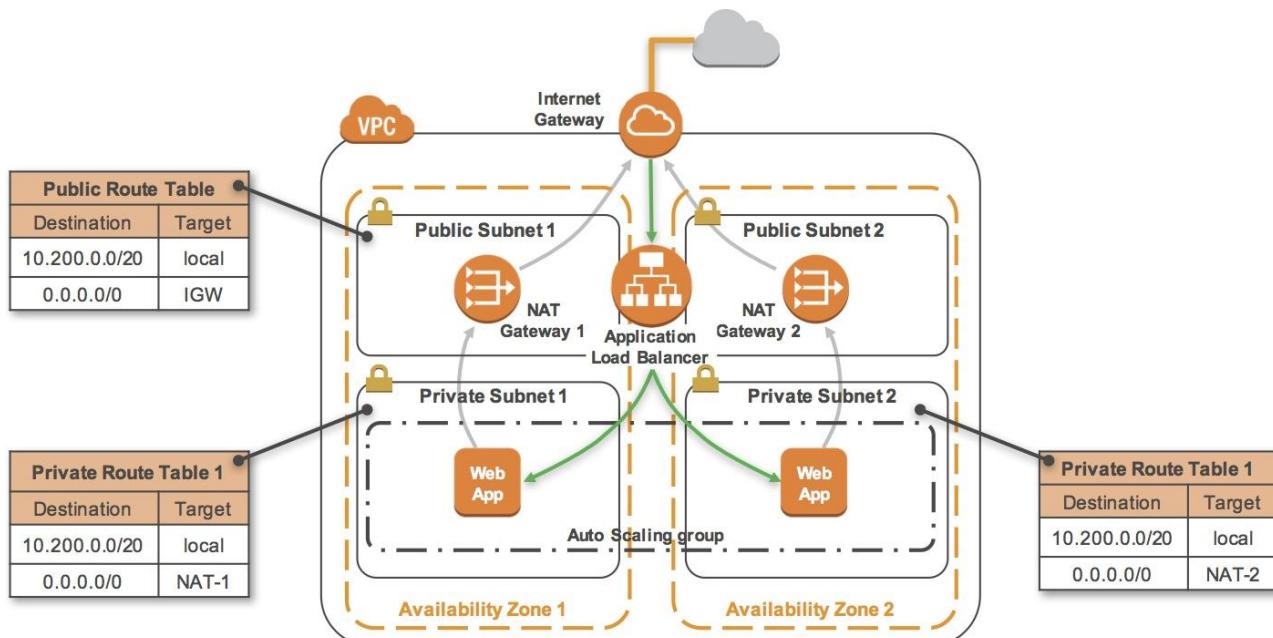
이번 실습에서는 단일 Amazon EC2 인스턴스에서 애플리케이션을 실행한 뒤 고가용성으로 전환할 것입니다.

목표

본 실습을 완료하면 다음과 같은 작업을 수행할 수 있습니다.

기존 Amazon EC2 인스턴스의 이미지를 생성하고, 이를 사용하여 새로운 인스턴스를 시작합니다. - 추가 가용 영역으로 Amazon VPC를 확장합니다. - VPC 서브넷과 라우팅 테이블을 생성합니다. - AWS NAT 게이트웨이를 생성합니다. - 로드 밸런서를 생성합니다. - Auto Scaling 그룹을 생성합니다.

실습의 최종 결과는 다음과 같습니다.



소요 시간

본 실습에는 약 60분이 소요됩니다.

AWS Management Console 액세스

- [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 클릭 합니다.

Open Console

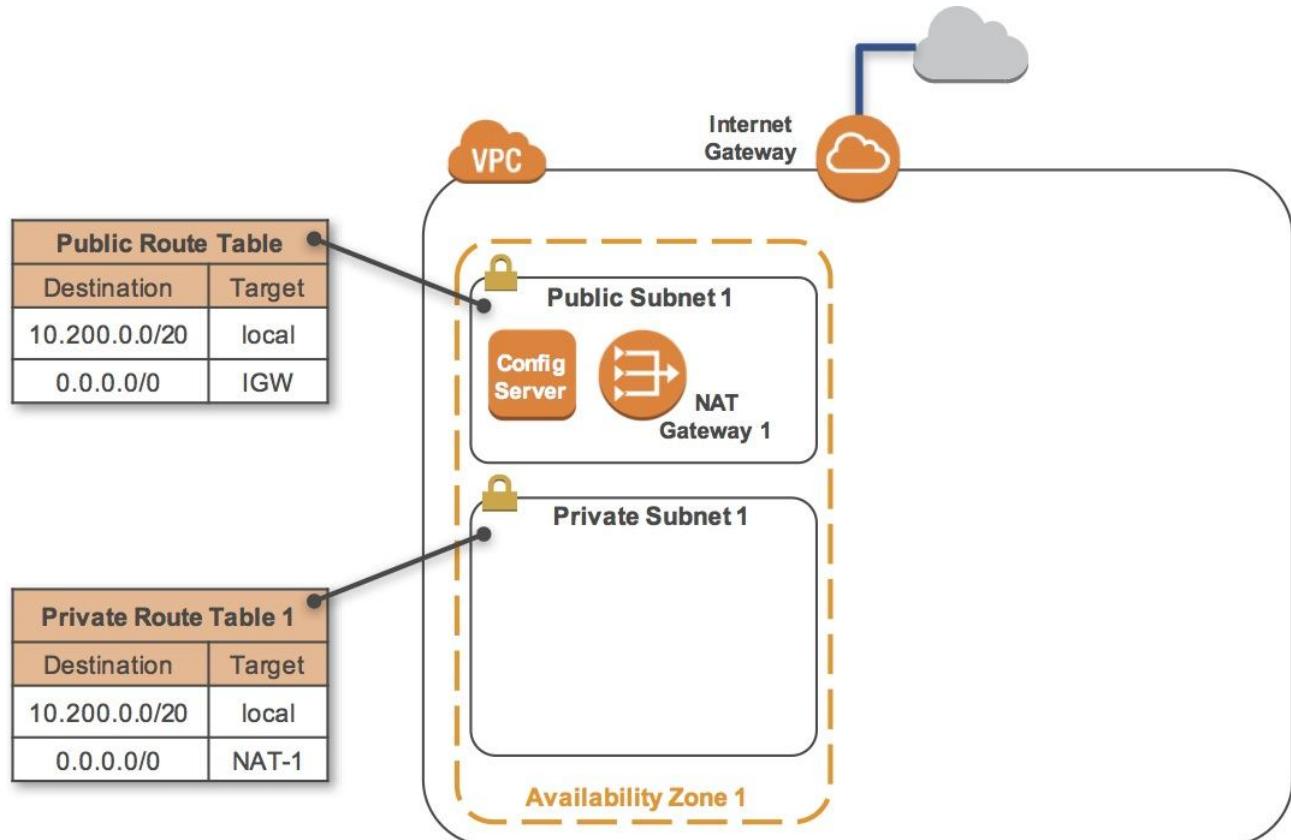
1. [3] 다음 단계에 따라 AWS Management Console에 로그인합니다.

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

작업 1: 환경 검사

이번 실습은 이미 Amazon CloudFormation을 통해 배포된 환경으로 시작하며, 다음과 같은 요소가 포함되어 있습니다.

- Amazon VPC
- 하나의 가용 영역에 대한 퍼블릭 서브넷과 프라이빗 서브넷
- 퍼블릭 서브넷과 연결된 인터넷 게이트웨이
- 퍼블릭 서브넷 내의 NAT 게이트웨이
- 퍼블릭 서브넷 내의 Amazon EC2 인스턴스



작업 1.1: VPC 검사

이 작업에서는 기존에 생성된 VPC 구성을 검토합니다.

1. [4]AWS Management Console의 **Services** 메뉴에서 **VPC**를 클릭합니다.
2. [5]왼쪽 탐색 창에서 **Your VPCs**를 클릭합니다.

이전에 생성한 **Lab VPC**를 확인할 수 있습니다.

- **CIDR** 열에서 ** 10.200.0.0/20** 값을 볼 수 있습니다. 즉, 이 VPC에 포함된 IP는 10.200.0.0에서 10.200.15.255까지 4,096개(일부 예약되거나 사용 불가능한 IP 포함)입니다.
- 이 값은 **Route Table**과 **Network ACL**에도 연결됩니다.
- 또한, 이 VPC는 **기본 Tenancy** 값이 있어 이 VPC에서 시작되는 인스턴스는 기본적으로 공유 테넌시 하드웨어를 사용합니다.

1. [6]탐색 창에서 **Subnets**를 클릭합니다.

Public Subnet 1 서브넷을 확인할 수 있습니다.

- VPC 열에서 이 서브넷이 **Lab VPC** 내부에 존재하는 것을 확인할 수 있습니다.
- **IPv4 CIDR** 열에서 **10.200.0.0/24** 값을 볼 수 있습니다. 이는 해당 서브넷의 IP가 10.200.0.0에서 10.200.0.255까지 256개(이 중 5개는 예약되었거나 사용 불가능)라는 의미입니다.
- **Availability Zone** 열에서 이 서브넷이 있는 가용 영역을 확인할 수 있습니다.

1. [7]**Public Subnet 1**이 있는 행을 클릭하면 페이지 아래쪽에 자세한 내용이 표시됩니다.
2. [8]창의 하반부에 있는 **Route Table** 탭을 클릭합니다.

이 서브넷의 상세한 라우팅을 확인할 수 있습니다.

- 첫 번째 항목은 VPC의 CIDR(**10.200.0.0/20**) 대역을 목적지로 하는 트래픽이 VPC 내에서 라우팅 되도록 지정합니다(**local**).
- 두 번째 항목은 인터넷(**0.0.0.0/0**)으로 나가는 트래픽이 인터넷 게이트웨이(*igw*-)로 라우팅되도록 지정합니다. 이렇게 설정하면 **퍼블릭 서브넷**이 됩니다.

3. [9]창의 하반부에 있는 **Network ACL** 탭을 클릭합니다.

여기에서 서브넷과 관련된 네트워크 액세스 제어 목록(ACL)을 확인할 수 있습니다. 현재 규칙은 **모든 트래픽**이 서브넷으로 들어가고 나올 수 있도록 허용되어 있지만, 보안 그룹을 사용하여 더 제한할 수 있습니다.

1. [10]왼쪽 탐색 창에서 **Internet Gateways**를 클릭합니다.

인터넷 게이트웨이가 이미 Lab VPC와 연결되어 있습니다.

1. [11]탐색 창에서 **Security Groups**를 클릭합니다.
2. [12]**Configuration Server SG**를 클릭합니다.

구성 서버에서 사용하는 보안 그룹입니다.

1. [13]창의 하반부에 있는 **Inbound Rules** 탭을 클릭합니다.

여기에서 이 보안 그룹이 SSH(TCP 포트 22) 및 HTTP(TCP 포트 80)를 통한 트래픽만 허용하는 것을 확인할 수 있습니다.

1. [14]**Outbound Rules** 탭을 클릭합니다.

여기에서 이 보안 그룹이 모든 아웃바운드 트래픽을 허용하는 것을 확인할 수 있습니다.

작업 1.2: Amazon EC2 인스턴스 검사

이 실습에서는 시작된 Amazon EC2 인스턴스를 검사합니다.

1. [15]Services 메뉴에서 **EC2**를 클릭합니다.
2. [16]왼쪽 탐색 창에서 **Instances**를 클릭합니다.

이미 실행 중인 **Configuration Server**를 확인할 수 있습니다. 창 하반부의 **Description** 탭에서 퍼블릭 및 프라이빗 IP 주소와 가용 영역, VPC, 서브넷, 보안 그룹 등, 이 인스턴스의 세부 사항을 확인할 수 있습니다.

1. [17]**IPv4 Public IP** 값을 복사하여 메모장과 같은 텍스트 편집기에 붙여넣습니다. 이 값은 나중에 사용할 것입니다.
2. [18]Actions 메뉴에서 **Instance Settings** → **View/Change User Data**를 차례로 클릭합니다.

표시되는 사용자 데이터가 없습니다! 이는 인스턴스가 아직 웹 애플리케이션을 실행하도록 구성되지 않았다는 의미입니다. Amazon EC2 인스턴스 시작 시 실행되는 **User Data script**를 제공하여 인스턴스를 구성할 수 있습니다. 하지만, 이 실습에서는 직접 인스턴스를 구성할 것입니다!

1. [19]**Cancel**을 클릭하여 사용자 데이터 대화 상자를 닫습니다.

작업 2: Amazon EC2 인스턴스에 로그인

Amazon EC2 인스턴스가 이미 시작되었지만, 아직 웹 애플리케이션을 실행하지는 않았습니다. 웹 애플리케이션을 설치하려면 SSH로 인스턴스에 로그인하여 응용 프로그램 설치 및 구성 명령을 실행해야 합니다.

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

작업 2.1: Windows SSH

Windows 사용자 전용 섹션입니다. Mac OS나 Linux에서 실행 중이시면 위의 "Mac/Linx Users" 링크를 클릭합니다.

이번 작업에서는 키 페어를 다운로드하고 이를 사용하여 PuTTY를 통해 Amazon EC2 인스턴스에 연결합니다.

1. [20]브라우저의 qwikLABS 페이지에 있는 **Connect** 섹션에서 **Download PEM/PPK** → **Download PPK**를 차례로 클릭합니다. (사용자 이름과 암호는 아래 섹션에 있습니다. 아래로 스크롤해야 할 수 있습니다.)
2. [21]로컬 컴퓨터의 **Downloads** 폴더나 접근하기 쉬운 위치에 파일을 저장합니다.

PuTTY를 사용하여 Amazon EC2 인스턴스에 연결합니다. 컴퓨터에 PuTTY가 설치되지 않았다면, [여기에서 다운로드](#)하십시오.

1. [22]다운로드한 putty.exe를 실행하여 **PuTTY**를 시작합니다.

2. [23] 실습 앞부분에서 텍스트 편집기에 복사해 둔 **Configuration Server**의 퍼블릭 IP 주소를 **Host Name**에 입력합니다.
3. [24] **Connection** 목록에서 SSH를 확장합니다.
4. [25] **Auth**를 클릭합니다.
5. [26] 앞에서 다운로드한 .ppk 파일을 **Private key file for authentication** 상자에서 찾아 Open을 클릭합니다.
6. [27] **PuTTY Security Alert** 대화 상자가 열리면, Yes를 클릭하여 PuTTY 캐시에 키를 추가합니다.
7. [28] **login as:**에 'ec2-user'를 입력하고 **Enter**를 누릅니다. 이제 **Web Server** 인스턴스에 로그인하였습니다.

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Wed Sep 10 05:55:27 2014 from 205.251.233.48

              _\   _ )     Amazon Linux AMI
             __| \__|_|_|

https://aws.amazon.com/amazon-linux-ami/2014.03-release-notes/
9 package(s) needed for security, out of 28 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-200-10-152 ~]$ █
```

(실제 텍스트는 위의 텍스트와 약간 다를 수 있습니다.)

1. [29] [Windows 사용자는 여기를 클릭하여 다음 작업으로 건너뜁니다.](#)

작업 2.2: Mac/Linux SSH

*Linux*와 Mac OS 사용자 전용 섹션입니다. [Windows 사용자는 여기를 클릭하여 다음 작업으로 건너뜁니다.](#)*

이번 작업에서는 키 페어를 다운로드하고 이를 사용하여 Amazon EC2 인스턴스에 연결합니다.

1. [30] 브라우저의 qwikLABS 페이지에 있는 **Connect** 섹션에서 **Download PEM/PPK → Download PEM**을 차례로 클릭합니다. (사용자 이름과 암호는 아래 섹션에 있습니다. 아래로 스크롤해야 할 수 있습니다.)
2. [31] 컴퓨터의 접근이 쉬운 위치에 파일을 저장합니다.
3. [32] 컴퓨터의 터미널 애플리케이션을 엽니다.
4. [33] EC2 인스턴스에 연결하려면, 터미널에서 다음 명령을 실행합니다.

```
chmod 400 <path and name of pem file>
ssh -i <path and name of pem> ec2-user@<Public IP>
```

- <path and name of pem file>의 경로/파일 이름을 다운로드한 .pem 파일로 대체합니다.
- <Public IP>의 실습 앞부분에서 텍스트 편집기에 복사해 둔 **Configuration Server**를 퍼블릭 IP 주소로 대체합니다.

작업 3: 웹 서버의 PHP 애플리케이션 다운로드, 설치 및 시작

이 작업에서는 일반적인 시스템 관리자 작업을 수행하여 웹 애플리케이션을 설치 및 구성합니다. 다음 작업에서는 이 머신의 이미지를 생성하고 더 많은 인스턴스로 애플리케이션을 자동 배포하여 고가용성을 구성합니다.

이 작업에 나온 명령을 사용하여 PHP 웹 애플리케이션을 다운로드하고 설치하며 실행할 수 있습니다. 이번 작업을 완수하기 위해 무엇을 수행하는지 정확히 이해할 수 있도록 한 번에 하나씩 각 명령을 설명하겠습니다.

1. [34]인스턴스에 설치된 기본 소프트웨어를 업데이트하려면 다음 명령을 실행합니다.

```
sudo yum -y update
```

Amazon Linux 인스턴스에서 사용할 수 있는 업데이트를 확인하고 업데이트를 다운로드하여 설치하는 명령입니다.

PuTTY 사용자 팁: 우클릭하면 간단하게 붙여넣기가 실행됩니다.

1. [35]웹 서버를 생성하는 패키지를 설치하려면 다음 명령을 실행합니다.

```
sudo yum -y install httpd php
```

Apache 웹 서버(httpd)와 PHP 언어 해석기를 설치하는 명령입니다.

1. [36]다음 명령을 실행합니다.

```
sudo chkconfig httpd on
```

Apache 웹 서버가 인스턴스가 시작할 때 자동으로 시작하도록 구성하는 명령입니다.

1. [37]다음 명령을 실행합니다.

```
wget https://us-west-2-aws-staging.s3.amazonaws.com/awsume-it/AWS-100-ARC/v5.2/lab-2-ha/scripts/phpapp.zip
```

PHP 웹 애플리케이션이 포함된 zip 파일을 다운로드하는 명령입니다.

1. [38]다음 명령을 실행합니다.

```
sudo unzip phpapp.zip -d /var/www/html/
```

기본 Apache 웹 서버 디렉터리에 PHP 애플리케이션을 압축 해제하는 명령입니다.

1. [39]다음 명령을 실행합니다.

```
sudo service httpd start
```

Apache 웹 서버를 시작하는 명령입니다.

"Could not reliably determine..."로 시작하는 경고는 무시해도 됩니다.

웹 애플리케이션을 구성했습니다! 이제 애플리케이션에 액세스하여 작동 상태를 확인할 수 있습니다.

1. [40]새로운 웹 브라우저 탭을 열고, 주소 표시줄에 인스턴스용 **Public IP** 주소를 붙여넣은 뒤 Enter 를 누릅니다. (텍스트 편집기에 복사하고, ssh/PuTTY에 사용했던 IP와 같은 IP 주소입니다.)

웹 애플리케이션이 나타나고 사용자 위치(Amazon EC2 인스턴스의 위치) 정보가 표시되어야 합니다. 이 정보는 freegeoip.net에서 가져왔습니다.

애플리케이션이 나타나지 않으면 강사의 지원을 받아 구성을 확인하십시오.

1. [41]이전 단계에서 열었던 웹 애플리케이션 브라우저 탭을 닫습니다.
2. [42]SSH 세션으로 돌아가서 다음 명령을 실행합니다.

```
exit
```

SSH 세션이 종료됩니다.

작업 4: Amazon 머신 이미지(AMI) 생성

이제 웹 애플리케이션이 인스턴스에 구성되어 웹 애플리케이션의 **Amazon 머신 이미지(AMI)**를 만들 수 있습니다. AMI는 Amazon EC2 인스턴스에 연결된 디스크 볼륨의 복사본입니다. 새 인스턴스가 AMI에서 시작되면 디스크 볼륨은 원본 인스턴스와 정확히 같은 데이터를 포함하게 됩니다.

이것은 여러 인스턴스와 가용 영역에서 애플리케이션을 실행할 수 있도록 인스턴스를 ~~복제~~하는 훌륭한 방법입니다.

이 작업에서는 Amazon EC2 인스턴스로부터 **AMI를 생성합니다**. 추후 이 이미지로 완전히 구성된 추가 인스턴스를 실행하여 고가용성 솔루션을 제공할 것입니다.

1. [43]EC2 Management Console을 보여주는 웹 브라우저 탭으로 돌아갑니다.
2. [44]Configuration Server가 선택된 것을 확인한 뒤, **Actions → Image → Create Image**를 차례로 클릭합니다.

현재 루트 볼륨이 인스턴스와 연결되었음을 확인할 수 있습니다. 이 볼륨은 AMI로 복사됩니다.

1. [45]**Image name**에 'Web application'을 입력합니다.
2. [46]**Create Image**를 클릭합니다.
3. [47]다른 값은 기본 설정으로 두고 **Close**를 클릭합니다.

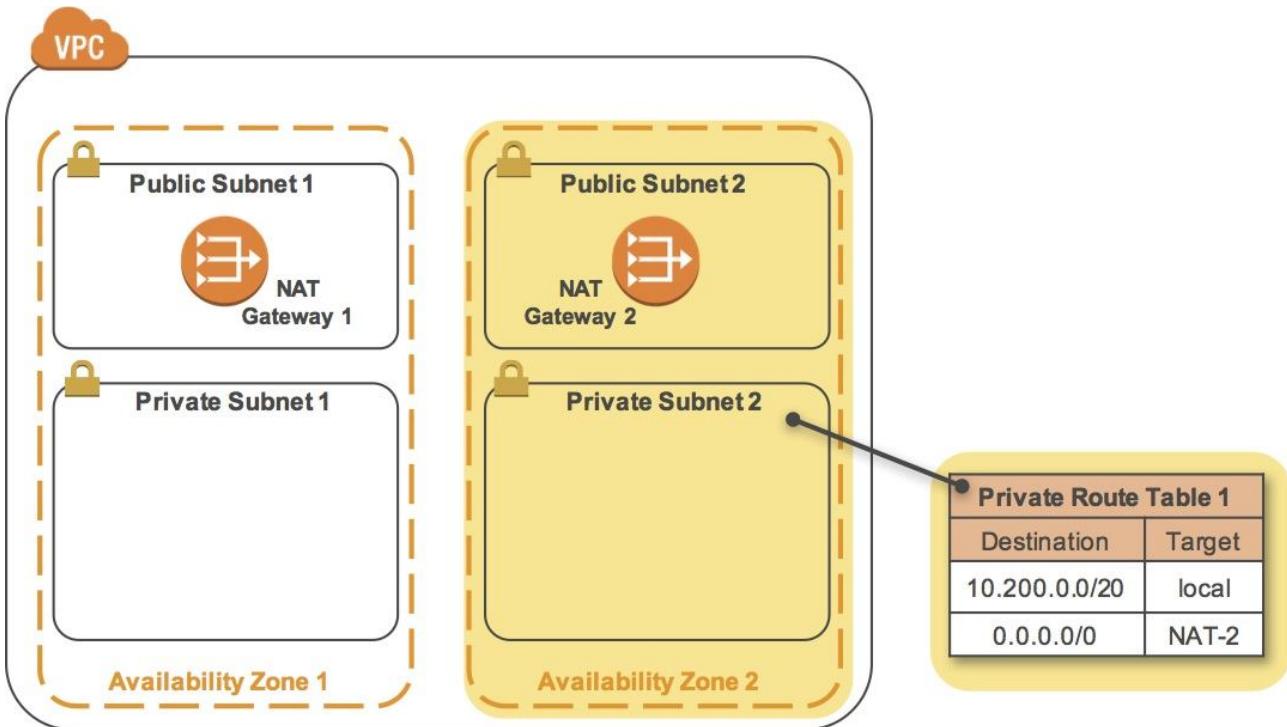
AMI는 백그라운드에서 생성되며 이후 단계에서 사용할 것입니다. AMI가 생성될 때까지 기다릴 필요는 없습니다.

작업 5: 두 번째 가용 영역 구성

고가용성 애플리케이션을 구축하려면 여러 가용 영역에서 리소스를 시작하는 것이 최선입니다. 가용 영역은 같은 리전 내의 물리적으로 분리된 데이터 센터(또는 데이터 센터 그룹)입니다. 여러 가용 영역에 걸쳐 애플리케이션을 실행하면 데이터 센터에 문제가 발생해도 더 많은 가용성을 제공합니다.

이 작업에서는 두 번째 가용 영역으로 네트워크 환경을 복제합니다. 다음과 같은 환경을 생성합니다.

- 두 번째 퍼블릭 서브넷
- 두 번째 프라이빗 서브넷
- 두 번째 NAT 게이트웨이
- 두 번째 프라이빗 라우팅 테이블



작업 5.1: 두 번째 퍼블릭 서브넷 생성

1. [48]Services 메뉴에서 VPC를 클릭합니다.
2. [49]왼쪽 탐색 창에서 Subnets를 클릭합니다.
3. [50]Public Subnet 1 행의 Availability Zone 값을 메모합니다. (값을 확인하기 위해 옆으로 스크롤 해야 할 수 있습니다.)

가용 영역 이름은 리전 이름(예: us-west-2)에 영역 식별자(예: a)를 더해 구성합니다. 조합하면, 이 가용 영역의 이름은 us-west-2a가 됩니다.

1. [51]Create Subnet을 클릭합니다.
2. [52]Create Subnet 대화 상자에서 다음과 같이 설정합니다.

Name tag	'Public Subnet 2'
VPC	Lab VPC
Availability Zone	기존 서브넷과 다른 different Availability Zone 선택(예: 기존 서브넷이 a였다면 b를 선택)
IPv4 CIDR block	'10.200.1.0/24'

이렇게 하면 다른 가용 영역에 두 번째 서브넷을 생성하면서도 여전히 **Lab VPC** 내에 있게 됩니다. 이 서브넷의 IP 범위는 10.200.1.0 ~ 10.200.1.255 사이입니다.

1. [53]**Yes, Create**를 클릭합니다.
2. [54]**Public Subnet 2**를 선택한 상태에서 창 아래쪽의 **Route Table** 탭을 클릭합니다. (왼쪽 탐색 창에서 Route Tables 링크를 클릭하지 마십시오.)

새 서브넷에 기본 라우팅 테이블이 함께 제공된 것을 확인할 수 있습니다. 하지만 이 라우팅 테이블은 인터넷 게이트웨이에 연결되지 않은 상태입니다.

1. [55]**Edit**를 클릭합니다.
2. [56]**Change to:** 드롭다운 목록에서 **Public Route Table**을 클릭합니다.
3. [57]**Save**를 클릭합니다.

이제 퍼블릭 서브넷 2는 인터넷과 직접 통신할 수 있는 **퍼블릭 서브넷**이 되었습니다.

작업 5.2: 두 번째 프라이빗 서브넷 생성

보안을 향상하기 위해 애플리케이션이 **프라이빗 서브넷**에 배포됩니다. 이렇게 하면, 인터넷에서 인스턴스에 직접 액세스할 수 없습니다. 고가용성을 구성하려면 두 번째 프라이빗 서브넷이 필요합니다.

1. [58]**Create Subnet**을 클릭합니다.
2. [59]**Create Subnet** 대화 상자에서 다음과 같이 설정합니다.

Name tag	'Private Subnet 2'
VPC	Lab VPC
Availability Zone	퍼블릭 서브넷 2와 같은 가용 영역을 선택하십시오.
IPv4 CIDR block	'10.200.4.0/23'

1. [60]**Yes, Create**를 클릭합니다.

이 서브넷의 IP 범위는 10.200.4.0 ~ 10.200.5.255 사이입니다.

작업 5.3: 두 번째 NAT 게이트웨이 생성

NAT 게이트웨이(네트워크 주소 변환)는 공용 서브넷에 프로비저닝되며 프라이빗 서브넷 리소스에 **외부 인터넷 연결**을 제공합니다. 지리 정보를 얻으려면 웹 애플리케이션이 인터넷에 연결해야 하므로 인터넷

트래픽이 NAT 게이트웨이를 지나가도록 경로를 지정해야 합니다.

고가용성 상태를 유지하려면 첫 번째 가용 영역에 문제가 발생해도 두 번째 가용 영역의 리소스에 영향이 없도록 웹 애플리케이션을 구성해야 합니다. 반대의 경우에도 마찬가지입니다. 따라서, 두 번째 가용 영역에 두 번째 NAT 게이트웨이를 생성해야 합니다.

1. [61]왼쪽 탐색 창에서 **NAT Gateways**를 클릭합니다.
2. [62]**Create NAT Gateway**를 클릭합니다.
3. [63]**Subnet**에 **Public Subnet 2**를 선택합니다.
4. [64]**Create New EIP**를 클릭합니다.

탄력적 IP 주소(EIP)는 이 NAT 게이트웨이와 연결될 고정 IP 주소입니다. 탄력적 IP 주소는 NAT 게이트웨이가 유지되는 동안 변경되지 않습니다.

1. [65]**Create a NAT Gateway**를 클릭합니다.
2. [66]**View NAT Gateway**를 클릭합니다.

이제 두 개의 NAT 게이트웨이가 보일 것입니다. (하나밖에 보이지 않는다면 둘 다 보일 때까지 오른쪽 위의 새로 고침 아이콘을 클릭합니다.)

1. [67]방금 생성한 NAT 게이트웨이를 찾습니다. 게이트웨이는 **Pending** 상태이며 프라이빗 IP 주소는 **10.200.1.1**로 시작할 것입니다.
2. [68]첫 번째 열에 *nat-*로 시작하는 **NAT Gateway ID**를 복사합니다. 복사한 ID를 다음 작업을 위해 텍스트 문서에 붙여넣습니다.

이제 두 번째 NAT 게이트웨이를 사용할 수 있도록 네트워크를 구성해야 합니다.

작업 5.4: 두 번째 프라이빗 라우팅 테이블 생성

라우팅 테이블을 서브넷으로 들어오고 나가는 트래픽 경로를 정의합니다. 이제 방금 생성했던 NAT 게이트웨이를 통해 인터넷 트래픽을 보내도록 프라이빗 서브넷 2에 대한 라우팅 테이블을 생성합니다.

1. [69]탐색 창에서 **Route Tables**를 클릭합니다.
2. [70]**Create Route Table**을 클릭합니다.
3. [71]**Create Route Table** 대화 상자에서 다음과 같이 설정합니다.

Name tag	'Private Route Table 2'
VPC	Lab VPC

1. [72]**Yes, Create**를 클릭합니다.
2. [73]창의 하반부에 있는 **Routes** 탭을 클릭합니다.

현재 라우팅 테이블은 로컬의 대상에 의해 식별되는 VPC 내에서만 트래픽을 전송합니다. 이제 두 번째 NAT 게이트웨이를 통해 인터넷 트래픽(와일드카드 **0.0.0.0/0**으로 식별)을 보내도록 라우팅 테이블을 구성합니다.

1. [74]**Edit**를 클릭합니다.
2. [75]**Add another route**를 클릭합니다.
3. [76]**Destination**에 '**0.0.0.0/0**'을 입력합니다.

4. [77]Target 상자를 클릭하고 이전에 복사한 ID의 NAT 게이트웨이를 클릭합니다. (텍스트 편집기에 서 이전에 저장한 *nat*-로 시작하는 ID를 확인하십시오.)
5. [78]Save를 클릭합니다.

이제 전에 생성한 두 번째 프라이빗 서브넷 2와 라우팅 테이블(프라이빗 라우팅 테이블 2)을 연결할 수 있습니다.

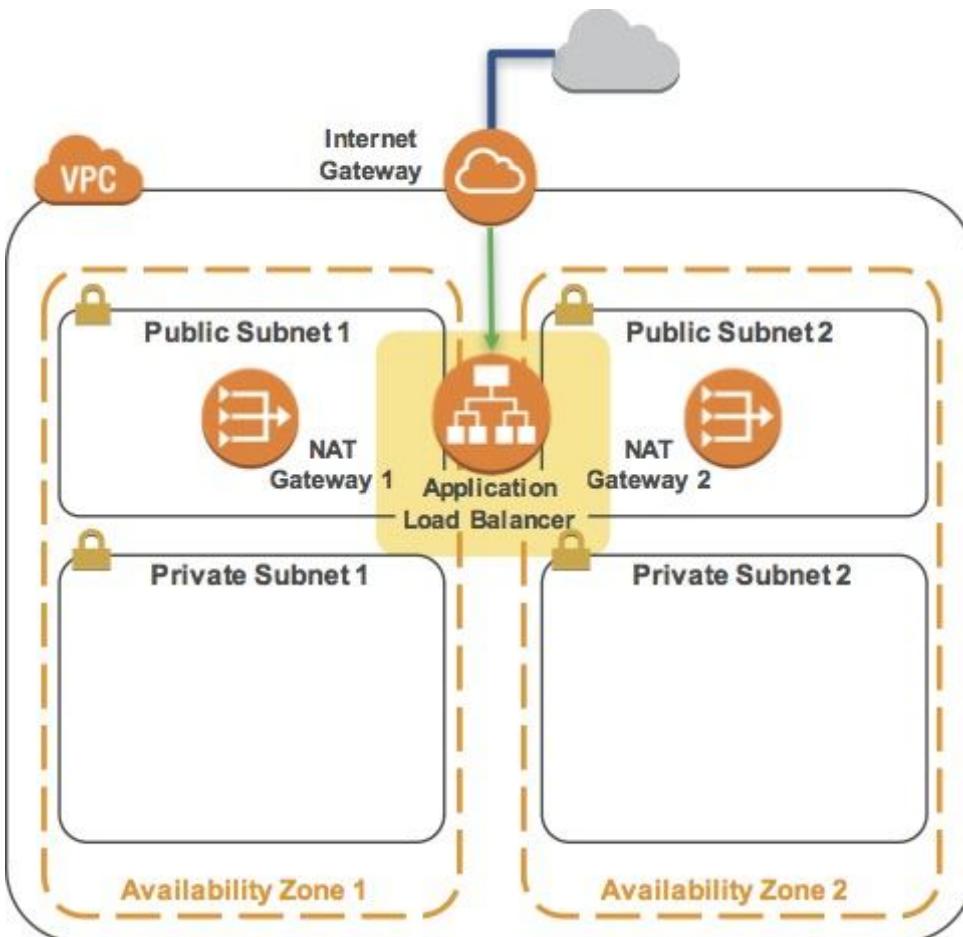
1. [79]Private Route Table 2가 선택된 상태에서 화면 아래쪽의 **Subnet Associations** 탭을 클릭합니다.
2. [80]Edit를 클릭합니다.
3. [81]Private Subnet 2 옆의 체크 박스를 선택(체크)합니다.
4. [82]Save를 클릭합니다.

이제 프라이빗 서브넷 2의 인터넷 트래픽이 두 번째 NAT 게이트웨이를 지나게 됩니다.

작업 6: 애플리케이션 로드 밸런서 생성

이번 작업에서는 여러 Amazon EC2 인스턴스에 요청을 분산하는 **Application Load Balancer**를 생성합니다. 로드 밸런서는 인스턴스 상태 확인을 수행하고 정상 인스턴스로만 요청을 전송하기 때문에 고가용성 구조의 핵심 구성 요소입니다.

아직 아무 인스턴스도 없습니다. 인스턴스는 다음 작업에서 Auto Scaling 그룹이 생성할 것입니다.



1. [83]Services 메뉴에서 **EC2**를 클릭합니다.
2. [84]왼쪽 탐색 창에서 **Load Balancers**(아래로 스크롤해야 보일 수 있음)를 클릭합니다.

3. [85]Create Load Balancer를 클릭합니다.

두 가지 유형의 로드 밸런서가 표시됩니다. 고전적인 로드 밸런서보다 더 기능이 많은 애플리케이션 로드 밸런서를 사용할 것입니다.

1. [86]반드시 **Application Load Balancer**를 선택합니다.
2. [87]Continue를 클릭합니다.
3. [88]Name에 'LB1'을 입력합니다.
4. [89]아래로 스크롤하여 **Availability Zones** 섹션을 찾습니다.
5. [90]VPC에서 **Lab VPC**를 선택합니다.

이제 로드 밸런서를 사용할 서브넷을 지정합니다. 로드 밸런서는 인터넷 연결 부하를 분산하므로 퍼블릭 서브넷 두 개를 모두 선택합니다.

1. [91]처음 표시되는 가용 영역을 클릭하고 아래쪽에 표시된 **Public Subnet**을 클릭합니다.
2. [92]두 번째 표시되는 가용 영역을 클릭하고 아래쪽에 표시된 **Public Subnet**을 클릭합니다.

이제 **Public Subnet 1**과 **Public Subnet 2**, 두 개의 서브넷이 선택되어 있어야 합니다. (아니라면, 돌아가서 다시 구성합니다.)

1. [93]Next: **Configure Security Settings**를 클릭합니다.

보안 향상을 위해 HTTPS 사용을 권장한다는 경고가 표시됩니다. 옳은 말이지만, 이 실습에서는 필요 없습니다.

1. [94]Next: **Configure Security Groups**를 클릭합니다.
2. [95]Security group for the web servers 설명에 있는 보안 그룹을 선택합니다.

이 보안 그룹은 들어오는 HTTP 트래픽만 허용하므로 로드 밸런서와 웹 서버 양쪽에 사용할 수 있습니다.

1. [96]Next: **Configure Routing**을 클릭합니다.

대상 그룹은 로드 밸런서로 들어오는 트래픽을 전송할 위치를 정의합니다. 애플리케이션 로드 밸런서는 수신 요청 URL을 기준으로 트래픽을 여러 대상 그룹으로 전송할 수 있습니다. 여러분의 웹 애플리케이션은 하나의 대상 그룹만 사용할 것입니다.

1. [97]Name에 'Group1'을 입력합니다.
2. [98]Advanced health check settings를 클릭하여 확장합니다.

애플리케이션 로드 밸런서는 자동으로 모든 인스턴스의 상태 확인을 수행하여 상태가 정상이고 요청에 응답하는지 확인합니다. 기본 설정이 권장되지만, 이 실습을 위해 약간 빠르게 설정하겠습니다.

1. [99]Healthy threshold에 '2'를 입력합니다.
2. [100]Interval에 '10'을 입력합니다.

이렇게 하면 10초마다 상태 확인을 수행하고 인스턴스가 연속 두 번 정상적으로 응답하면 정상으로 간주합니다.

1. [101]Next: **Register Targets**를 클릭합니다.

대상은 로드 밸런서의 요청에 응답할 인스턴스입니다. 아직 아무런 웹 애플리케이션 인스턴스가 없으므로 이 단계는 건너뛰어도 됩니다.

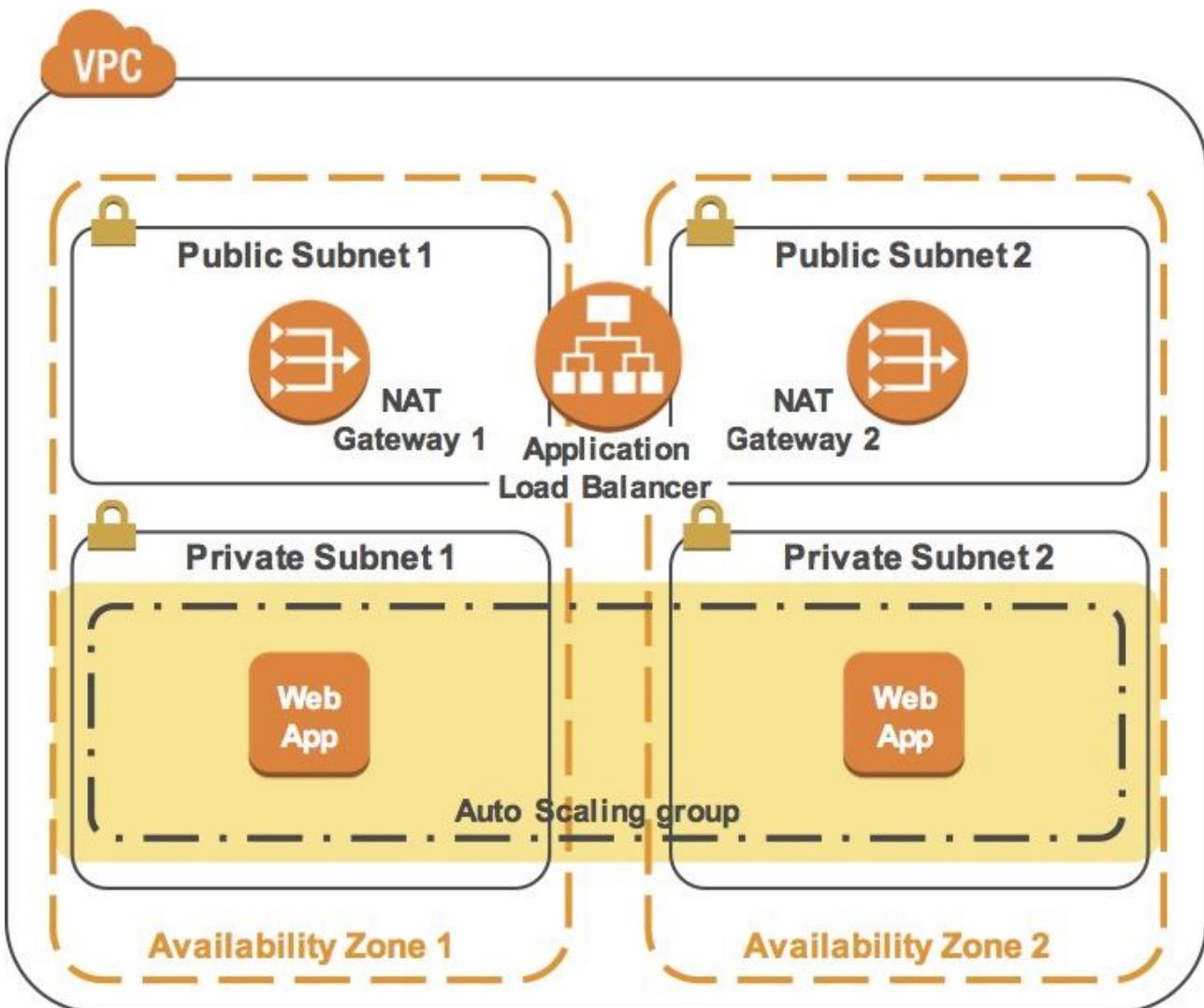
1. [102] **Next: Review**를 클릭합니다.
2. [103] 설정을 확인한 후 **Create**를 클릭합니다.
3. [104] **Close**를 클릭합니다.

이제 로드 밸런서가 백그라운드에서 프로비저닝됩니다. 이제 Auto Scaling 그룹을 생성하여 Amazon EC2 인스턴스를 시작할 수 있습니다.

작업 7: Auto Scaling 그룹 생성

Auto Scaling은 사용자가 정의한 정책, 일정 및 상태 확인에 따라 자동으로 Amazon EC2 인스턴스를 시작하거나 종료하도록 설계된 웹 서비스입니다. 또한, 인스턴스를 여러 가용 영역에 자동으로 분산하여 고가용성 애플리케이션을 만듭니다.

이번 작업에서는 Amazon EC2 인스턴스를 프라이빗 서브넷에 배포하는 Auto Scaling 그룹을 생성합니다. 이렇게 하면 프라이빗 서브넷의 인스턴스가 인터넷에 접속할 수 없으므로 애플리케이션 배포 시 최상의 보안을 확보할 수 있습니다. 대신, 사용자는 요청을 로드 밸런서로 보내고 밸런서는 요청을 프라이빗 서브넷의 Amazon EC2 인스턴스로 전달합니다.



1. [105] 왼쪽 탐색 창에서 **Auto Scaling Groups**(아래로 스크롤해야 보일 수 있음)를 클릭합니다.
2. [106] **Create Auto Scaling group**을 클릭합니다.
3. [107] **Create launch configuration**을 클릭합니다.

시작 구성은 Auto Scaling이 시작할 인스턴스 유형을 정의합니다. 인터페이스가 Amazon EC2 인스턴스를 시작하는 것과 비슷하게 보이지만, 인스턴스를 시작하기보다는 나중에 사용할 수 있도록 구성을 저장합니다.

이전에 생성한 AMI를 사용하도록 시작 구성을 설정할 것입니다. 여기에는 구성 서버에 설치한 소프트웨어 목록이 포함됩니다.

1. [108] 왼쪽 탐색 창에서 **My AMIs**를 클릭합니다.
2. [109] **Web application** 행에서 **Select**를 클릭합니다.
3. [110] **Next: Configure Details**를 클릭하여 기본값(t2.micro)을 적용합니다.
4. [111] **Name**에 'Web application'을 입력합니다.
5. [112] **Next: Add Storage**를 클릭합니다.

이 인스턴스에서는 추가 스토리지가 필요 없으므로, 설정은 기본값으로 유지합니다.

1. [113] **Next: Configure Security Group**을 클릭합니다.
2. [114] **Select an existing security group**을 클릭합니다.
3. [115] **Security group for the web servers** 설명에 있는 보안 그룹을 선택합니다.
4. [116] **Review**를 클릭합니다.

SSH를 통해 인스턴스에 연결할 수 없다는 경고 메시지가 표시될 수 있습니다. 서버 구성이 이미 AMI에 정의되었으며 인스턴스에 로그인할 필요가 없으므로 경고 메시지는 무시해도 됩니다.

1. [117] **Continue**를 클릭하여 경고 메시지를 닫습니다.
2. [118] 설정을 확인한 후 **Create launch configuration**을 클릭합니다.
3. [119] 메시지가 표시되면, qwikLABS 키 페어를 수락하고, 승인 확인란을 선택한 후, **Create launch configuration**을 클릭합니다.

이제 Auto Scaling 그룹을 생성하라는 메시지가 나타납니다. 여기에는 인스턴스 수와 인스턴스가 시작되어야 하는 위치가 정의되어 있습니다.

1. [120] **Create Auto Scaling Group** 페이지에서 다음과 같이 설정합니다.

Group Name	'Web application'
Group Size	Start with '2' instances
Network	Lab VPC
Subnet	상자를 클릭하고 Private Subnet 1 및 Private Subnet 2 모두 선택

Auto Scaling은 선택된 각기 다른 가용 영역에 있는 서브넷으로 인스턴스를 자동 분산합니다. 이렇게 하면 하나의 가용 영역에 문제가 발생해도 애플리케이션이 작동하기 때문에 탁월한 고가용성을 유지할 수 있습니다.

1. [121] **Advanced Details**를 클릭하여 확장합니다.
2. [122] **Load Balancing** 체크 박스를 선택(체크)합니다.
3. [123] **Target Groups**를 클릭하고 **Group1**을 선택합니다.
4. [124] **Next: Configure scaling policies**를 클릭합니다.

5. [125]Keep this group at its initial size를 선택했는지 확인합니다.

이 구성은 Auto Scaling이 항상 Auto Scaling 그룹에 두 개의 인스턴스를 유지할 것을 지시합니다. 이렇게 하면 하나의 인스턴스에 문제가 발생해도 애플리케이션이 계속 운영되므로 고가용성에 이상적입니다. 인스턴스에 문제가 발생하면 Auto Scaling이 자동으로 대체 인스턴스를 시작합니다.

1. [126]Next: Configure Notifications를 클릭합니다.

어떤 알림도 설정하지 마십시오.

1. [127]Next: Configure Tags를 클릭합니다.

Auto Scaling 그룹에 배치된 태그도 Auto Scaling이 시작한 인스턴스에 자동으로 전파할 수 있습니다.

1. [128]Key에 'Name'을 입력합니다.
2. [129]Value에 'Web application'을 입력합니다.
3. [130]Review를 클릭합니다.
4. [131]설정을 확인한 후 **Create Auto Scaling group**을 클릭합니다.
5. [132]Close를 클릭합니다.

처음에는 Auto Scaling 그룹이 0개의 인스턴스를 보여줍니다. 곧 두 개의 인스턴스로 업데이트됩니다. (오른쪽 위에 있는 새로 고침 아이콘을 클릭하여 최신 상태로 업데이트합니다.)

곧 두 가용 영역에서 애플리케이션이 실행되며 Auto Scaling은 하나의 인스턴스나 가용 영역에 문제가 발생해도 구성을 유지할 것입니다.

작업 8: 애플리케이션 테스트하기

이번 작업에서는 웹 애플리케이션 실행을 확인하고 고가용성을 테스트합니다.

1. [133]왼쪽 탐색 창에서 **Target Groups**를 선택합니다.
2. [134]창의 하반부에 있는 **Targets** 탭을 클릭합니다.

두 개의 등록된 인스턴스를 확인할 수 있어야 합니다. 상태 열에는 인스턴스에 대한 로드 밸런서 상태 확인 결과가 표시됩니다.

1. [135]두 인스턴스의 **Status**가 모두 **healthy**로 표시될 때까지 오른쪽 위의 새로 고침 아이콘을 흔들어 클릭합니다.

끝까지 상태가 정상으로 바뀌지 않으면, 강사의 지원을 받아 구성을 확인하십시오. 상태 열의 아이콘 위로 마우스를 가져가면 더 상세한 상태 정보를 확인할 수 있습니다.

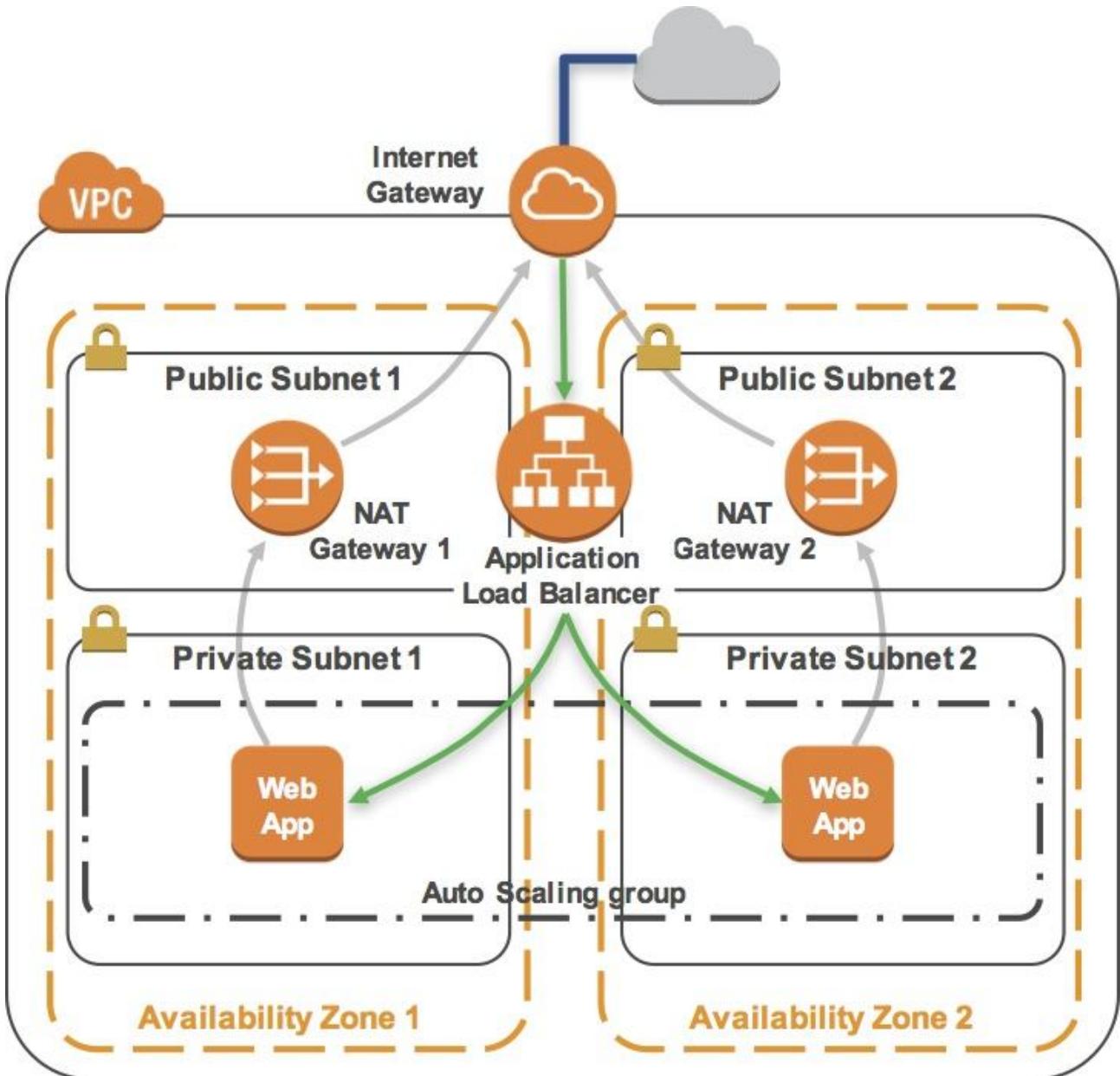
로드 밸런서에 연결하여 애플리케이션을 테스트하고 Amazon EC2 인스턴스 중 하나에 요청을 전송할 것입니다. 로드 밸런서의 DNS 이름을 검색해야 합니다.

1. [136]왼쪽 탐색 창에서 **Load Balancers**를 클릭합니다.
2. [137]창 하반부의 **Description** 탭에서 **DNS Name**을 클립보드로 복사합니다. "(A Record)"는 복사하지 마십시오. DNS 이름은 *LB1-xxxx.elb.amazonaws.com*과 비슷한 형식이어야 합니다.
3. [138]새로운 웹 브라우저 탭을 열고 클립보드의 DNS 이름을 붙여넣은 다음 엔터를 누릅니다.

로드 밸런서가 요청을 Amazon EC2 인스턴스 중 하나에 전달했습니다. 인스턴스 ID와 가용 영역은 웹 애플리케이션 아래쪽에 표시됩니다.

- [139] 웹 브라우저에서 페이지를 새로 고칩니다. 인스턴스 ID와 가용 영역이 가끔 두 인스턴스 간에 변경되는 것이 확인될 것입니다.

이 웹 애플리케이션이 표시될 때 정보 흐름은 다음과 같습니다.



- 인터넷과 연결된 퍼블릭 서브넷에 있는 로드 밸런서에 요청을 보냈습니다.
- Load Balancer**는 프라이빗 서브넷에 있는 Amazon EC2 인스턴스 중 하나를 선택하여 요청을 전달했습니다.
- Amazon EC2** 인스턴스는 freegeoip.net에서 지리 정보를 요청했습니다. 이 요청은 인스턴스와 같은 가용 영역에 있는 **NAT Gateway**를 통해 인터넷으로 나갔습니다.
- 그런 다음 Amazon EC2 인스턴스가 로드 밸런서로 웹 페이지를 돌려줬고 로드 밸런서는 웹 브라우저로 페이지를 전달했습니다.

단계 8: 고가용성 테스트

애플리케이션은 고가용성으로 구성되어 있습니다. Amazon EC2 인스턴스 중 하나를 중지시켜 고가용성을 확인할 수 있습니다.

1. [140] 웹 브라우저의 EC2 Management Console 탭으로 돌아갑니다(곧 돌아가야 하므로 웹 애플리케이션 탭을 닫지는 마십시오).
2. [141] 왼쪽 탐색 창에서 **Instances**를 클릭합니다.

먼저, 구성 서버가 더는 필요하지 않으므로 종료해도 무관합니다.

1. [142] **Configuration Server**를 선택합니다.
2. [143] **Actions** → **Instance State** → **Terminate**를 차례로 클릭한 뒤 **Yes, Terminate**를 클릭합니다.

그런 다음, 웹 애플리케이션 인스턴스 중 하나를 중지시켜 오류 상황을 시뮬레이션합니다.

1. [144] **Web application** 이름의 인스턴스 중 하나를 선택합니다(어떤 인스턴스를 선택하든 관계없음).
2. [145] **Actions** → **Instance State** → **Stop**을 차례로 클릭한 뒤 **Yes, Stop**을 클릭합니다.

잠시 후, 로드 밸런서가 해당 인스턴스가 응답하지 않는 것을 감지하고 자동으로 모든 요청을 남은 인스턴스로 전달되도록 경로를 지정합니다.

1. [146] 웹 브라우저의 웹 애플리케이션 탭으로 돌아가 페이지를 여러 번 새로 고칩니다.

페이지 아래쪽에 표시되는 **Availability Zone**이 바뀌지 않는 것이 확인될 것입니다. 인스턴스 하나가 중단되었지만, 애플리케이션은 가용 상태로 남아 있습니다.

몇 분 후, Auto Scaling도 인스턴스 오류를 감지합니다. Auto Scaling은 두 개의 인스턴스가 실행되도록 구성되었으므로 Auto Scaling이 **자동으로 대체 인스턴스를 시작합니다**.

1. [147] 웹 브라우저의 EC2 Management Console 탭으로 돌아갑니다. 새로운 Amazon EC2 인스턴스가 나타날 때까지 투或多 투 위의 새로 고침 아이콘을 클릭합니다.

몇 분 후, 새로운 인스턴스에 대한 상태 확인이 정상이 되고 로드 밸런서가 두 가용 영역으로 트래픽을 전달할 것입니다. 웹 애플리케이션 탭을 다시 고침하여 진행 상황을 확인할 수 있습니다.

이렇게 애플리케이션이 고가용성 상태인 것을 확인합니다.

실습 완료

축하합니다! 실습을 완료했습니다.

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