



training and
certification

Architecting on AWS (KO)
Lab Guide
버전 7.2.6
200-ARCHIT-72-KO-LG

© 2022 Amazon Web Services, Inc. 및 자회사. All rights reserved.

본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 임대 또는 판매는 금지됩니다.

본 과정에 대한 수정 사항이나 피드백, 문의사항이 있으면
<https://support.aws.amazon.com/#/contacts/aws-training>

을 통해 연락해 주십시오.

모든 상표는 해당 소유자의 자산입니다.

목차

실습 1: AWS 관리 콘솔 및 AWS CLI 살펴보기 및 사용	4
실습 2: Amazon VPC 인프라 구축	21
실습 3: Amazon VPC 인프라에 데이터베이스 계층 생성	55
실습 4: Amazon VPC에서 고가용성 구성	72
실습 5: 서버리스 아키텍처 구축	93
실습 6: Amazon S3 오리진으로 Amazon CloudFront 배포 구성	112
실습 7: 캡스톤 실습	136



실습 1: AWS 관리 콘솔 및 AWS CLI 살펴보기 및 사용

© 2022 Amazon Web Services, Inc. 또는 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다.
상업적인 복제, 임대 또는 판매는 금지됩니다. 모든 상표는 해당 소유자의 자산입니다.

참고: 개인 정보, 개인을 식별할 수 있는 정보 또는 기밀 정보는 실습 환경에 입력하지 마십시오.
입력한 정보가 다른 사용자에게 공개될 수도 있습니다.

수정 사항이나 피드백 또는 기타 질문이 있으십니까? *AWS Training and Certification*에서 문의해 주십시오.

실습 개요

Amazon Web Services 환경은 리소스를 신속하고 저렴하게 사용할 수 있도록 설계된 하드웨어 및 소프트웨어 서비스의 통합된 모음입니다. AWS 환경에서 AWS API가 제공됩니다. API는 리소스와 통신하는 방법을 나타냅니다. 다양한 방법으로 AWS 리소스와 상호 작용할 수 있지만 모든 상호 작용에서는 AWS API가 사용됩니다. AWS 관리 콘솔은 간편한 AWS용 웹 인터페이스를 제공합니다. AWS Command Line Interface(AWS CLI)는 명령줄을 통해 AWS 서비스를 관리하는 통합 도구입니다. AWS 관리 콘솔을 통해 AWS에 액세스하는 경우와 명령줄 도구를 사용하는 경우 모두 AWS API를 호출하는 도구를 사용하게 됩니다.

이 실습은 아키텍팅 기본 사항 모듈의 내용을 토대로 진행됩니다. 해당 모듈에서는 AWS에서 워크로드를 생성하려면 충족해야 하는 핵심 요구 사항을 중점적으로 설명합니다. 이 실습에서는 해당 모듈의 내용, 즉 AWS 워크로드를 구축하는 도구, 방법, 위치를 더욱 명확하게 파악합니다. 먼저 AWS 관리 콘솔의 기능을 살펴본 후에, Amazon Simple Storage Service(Amazon S3) API를 사용해 두 가지 방법으로 Amazon S3 버킷을 배포하고 해당 버킷에 대한 연결을 테스트합니다.

- AWS 관리 콘솔
- AWS CLI

목표

이 실습을 마치면 다음을 수행할 수 있습니다.

- AWS 관리 콘솔 살펴보기 및 사용
- AWS 관리 콘솔을 사용하여 리소스 생성

- AWS CLI 살펴보기 및 사용
- AWS CLI를 사용하여 리소스 생성

선행 조건

이 실습을 진행하려면 다음 항목이 필요합니다.

- Microsoft Windows, macOS 또는 Linux(Ubuntu, SuSE 또는 Red Hat)가 실행되는 Wi-Fi 지원 노트북 컴퓨터
- 인터넷 브라우저(예: Chrome, Firefox 또는 Microsoft Edge)
- 일반 텍스트 편집기

소요 시간

이 실습을 완료하려면 약 35분이 소요됩니다.

이 실습에서 사용되지 않는 AWS 서비스

이 실습에서 사용하지 않는 AWS 서비스는 실습 환경에서 사용 중지 상태로 설정되어 있습니다. 또한 이 실습에 사용되는 서비스의 기능은 실습에 필요한 것으로 제한됩니다. 다른 서비스에 액세스하거나 이 실습 가이드에서 제공하는 것 외의 작업을 수행하는 경우 오류가 발생할 수 있습니다.

실습 환경

작업을 쉽게 시작할 수 있도록 이 실습 환경에서 제공되는 리소스는 Amazon Virtual Private Cloud(Amazon VPC), 필요한 기본 네트워크 구조, 포트 80을 통해 HTTP 프로토콜을 허용하는 보안 그룹, Amazon CLI가 설치된 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 그리고 관련 Amazon EC2 인스턴스 프로파일입니다. 인스턴스 프로파일에는 Session Manager의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 액세스를 허용하는 데 필요한 권한이 포함되어 있습니다. Session Manager는 AWS Systems Manager의 기능입니다.

AWS 관리 콘솔과 AWS CLI를 통해 실습에 사용되는 AWS 서비스와 리소스를 생성하는 AWS API의 대화형 흐름이 다음 다이어그램에 나와 있습니다.



실습 시작

1. 실습을 시작하려면 페이지 상단에서 실습 시작을 선택합니다.

그러면 실습 리소스를 프로비저닝하는 프로세스가 시작됩니다. 실습 리소스를 프로비저닝하는 데 걸리는 예상 시간이 표시됩니다. 계속 진행하기 전에 사용할 리소스가 프로비저닝될 때까지 기다려야 합니다.

- ① 토큰을 입력하라는 메시지가 표시되면 여러분에게 배포된 토큰(또는 구매한 크레딧)을 사용하세요.

2. 실습을 열려면 콘솔 열기를 선택합니다.

새 웹 브라우저 탭에서 **AWS Management Console** 로그인 페이지가 열립니다.

3. **Sign in as IAM user** 페이지에서

- **IAM user name**에 awsstudent를 입력합니다.
- **Password**에 이 지침의 왼쪽에 나열된 **Password** 값을 복사하여 붙여넣습니다.
- **Sign in**을 선택합니다.

△ 별다른 지시가 없는 한 리전을 변경하지 마십시오.

일반적인 로그인 오류

오류: 우선 로그 아웃 필요

Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

You must first log out before logging into a different AWS account라는 메시지가 표시된다면 다음을 수행합니다.

- [click here](#)의 링크를 선택합니다.
- **Amazon Web Services Sign In** 웹 브라우저 탭을 닫고 초기 실습 페이지로 돌아갑니다.
- 콘솔 열기를 다시 선택합니다.

경우에 따라 일부 팝업 또는 스크립트 차단 웹 브라우저 확장 프로그램 때문에 **실습 시작** 버튼이 제대로 작동하지 않을 수 있습니다. 실습을 시작하는 데 문제가 있는 경우 다음을 수행합니다.

- 팝업 또는 스크립트 차단 프로그램의 허용 목록에 실습 도메인 이름을 추가하거나 차단 프로그램을 끕니다.
- 페이지를 새로 고친 후 다시 시도하십시오.

과제 1: AWS 관리 콘솔 살펴보기 및 구성

이 작업에서는 AWS 관리 콘솔과 통합 검색 도구를 살펴봅니다. 그런 다음 리전, 위젯, 서비스를 구성합니다.

- **추가 정보:** AWS 관리 콘솔에서는 AWS 계정 루트 사용자 자격 증명이나 AWS Identity and Access Management(IAM) 계정 자격 증명을 사용하는 보안 로그인 기능을 제공합니다. 처음 로그인하면 사용자 자격 증명 인증이 진행되며 홈 페이지가 표시됩니다. 홈 페이지에서 각 서비스 콘솔에 액세스할 수 있으며, AWS 관련 작업을 수행하는 데 필요한 정보를 한곳에서 액세스할 수 있습니다.

과제 1.1: AWS 리전 선택

이 작업에서는 리소스를 관리할 위치를 지정하는 AWS 리전을 선택합니다. 리전은 같은 지리적 영역에 있는 AWS 리소스 집합입니다.

4. 탐색 모음에서 콘솔 오른쪽 위에 표시된 **Region** 선택기를 선택하고 전환하려는 리전을 선택합니다.

그러면 콘솔 홈 페이지의 리전이 선택한 리전으로 변경됩니다.

△ **주의:** 특정 리전 선택 시 콘솔 홈 페이지가 아닌 다른 웹 페이지가 열리면 **Cancel**을 선택하고 다른 리전을 선택해 봅니다.

다음으로는 기본 리전을 구성합니다.

5. 탐색 모음에서 본인의 계정 이름을 선택합니다.

계정 이름의 예로는 **AWSLabUser-xxxxxx** 등이 있습니다.

6. Unified Settings 페이지를 열려면 **Settings**를 선택합니다.

Unified Settings 페이지가 표시됩니다.

7. **Localization and default Region** 섹션에서 **Edit** 버튼을 클릭합니다.

8. **Default Region**의 드롭다운 메뉴에서 **US East (N Virginia) us-east-1** 리전을 선택합니다.

9. **Save settings**를 선택합니다.

페이지에 다음 메시지가 표시됩니다.

- Successfully updated localization and Region settings

△ **주의:** 콘솔 오른쪽 위의 **Region** 선택기에 표시된 현재 리전과 같은 리전을 기본 **Region** 드롭다운 메뉴에서 선택하면 저장 성공 메시지와 **Go to new default Region** 링크가 표시되지 않습니다. 이 경우 드롭다운 메뉴에서 다른 리전을 선택해야 이 메시지가 표시되며, 다음 단계를 완료할 수 있습니다.

10. **Go to new default Region**을 선택합니다.

리전이 **US East (N Virginia) us-east-1** 또는 선택한 리전으로 설정된 **General settings** 페이지가 표시됩니다.

참고: 기본 리전을 선택하지 않으면 마지막으로 방문한 리전이 기본 리전으로 설정됩니다.

11. 페이지 왼쪽 위에 표시된 **AWS 로고**를 선택하여 콘솔 홈 페이지로 돌아갑니다.

△ **주의:** "The new AWS Console Home will replace your existing experience soon,"라는 메시지가 포함된 배너가 표시되면 **Switch now**를 선택합니다.

12. 탐색 모음에서 콘솔 오른쪽 위에 표시된 **Region** 선택기를 선택하고 이 지침 왼쪽에 나와 있는 *LabRegion* 값과 일치하는 **Region**을 선택합니다.

참고: 리소스를 생성하기 전에 리전이 올바른지 확인해야 합니다.

과제 1.2: AWS 관리 콘솔의 통합 검색

이 작업에서는 탐색 모음의 검색 상자를 살펴봅니다. 검색 상자에서는 AWS 서비스와 기능, 서비스 설명서, AWS Marketplace 등을 찾을 수 있는 통합 검색 도구가 제공됩니다.

13. 서비스용 콘솔을 열려면 AWS 관리 콘솔 위쪽 가운데 **통합 검색 창**으로 이동합니다. 검색 창에는 *Search for services, features, marketplace products, and docs* 레이블이 표시되어 있습니다. 검색 창에 *cloud*를 입력합니다.

문자를 많이 입력할수록 더 자세한 검색 결과가 제공됩니다.

14. 원하는 콘텐츠 유형만 결과에 포함되도록 하려면 왼쪽 탐색 창에서 범주 중 하나를 선택합니다.

15. 특정 서비스나 서비스에서 많이 사용되는 기능으로 빠르게 이동하려는 경우 **Services** 섹션에서 결과의 **AWS Cloud Map** 서비스 이름 위에 마우스를 놓고 링크를 선택합니다.

그러면 **AWS Cloud Map console** 페이지가 표시됩니다.

① **추가 정보:** 설명서 검색 결과나 AWS Marketplace 검색 결과 관련 추가 세부 정보를 확인하려면 결과 제목 위에 마우스를 놓고 링크를 선택합니다.

16. 페이지 왼쪽 위에 표시된 **AWS 로고**를 선택하여 콘솔 홈 페이지로 돌아갑니다.

과제 1.3: 즐겨찾기 추가 및 제거

이 작업에서는 AWS 관리 콘솔에서 Favorites 목록에 AWS 서비스를 추가하고, 추가한 서비스를 Favorites 목록에서 제거합니다.

17. 탐색 모음에서 **Services**를 선택하여 전체 서비스 목록을 엽니다.

18. 왼쪽 탐색 메뉴에서 **All services** 또는 **Recently visited**를 선택한 다음 목록에서 즐겨찾기로 추가할 서비스를 선택합니다.

19. 서비스 이름 왼쪽에서 **별표**를 선택합니다.

참고: Favorites 목록에 서비스를 더 추가하려면 이전 단계를 반복합니다.

20. 즐겨찾기 서비스 목록을 확인하려면 왼쪽 탐색 메뉴에서 **Favorites**를 선택합니다.

참고: 즐겨찾기는 콘솔 창 위쪽 탐색 모음에도 고정되어 표시됩니다.

21. 탐색 모음에서 Services를 선택하여 전체 서비스 목록을 엽니다.

22. **Favorites** 목록에서 제거할 서비스 이름 옆의 별표 선택을 취소합니다.

참고: Recently visited 목록이나 All services 목록에서 Favorites 목록에 있는 서비스 이름 옆의 별표 선택을 취소해도 됩니다.

과제 1.4: 서비스용 콘솔 열기

23. 탐색 모음에서 Services를 선택하여 전체 서비스 목록을 엽니다.

24. 특정 서비스로 빠르게 이동하려면 Favorites, Recently visited 또는 All services 아래에서 해당 서비스를 선택합니다.

선택한 **service console** 페이지가 표시됩니다.

25. 페이지 왼쪽 위에 표시된 AWS 로고를 선택하여 AWS 관리 콘솔 홈 페이지로 돌아갑니다.

과제 1.5: 위젯 생성 및 사용

이 작업에서는 위젯에 대해 알아봅니다. AWS 환경 관련 중요한 정보가 표시되는 위젯은 서비스 바로 가기를 제공합니다. 위젯을 추가/제거/다시 정렬하거나 위젯 크기를 변경하여 환경을 사용자 정의할 수 있습니다.

26. 위젯을 추가하려면 다음을 구성합니다.

- 페이지 오른쪽 아래에서 + Add widgets를 선택합니다.

팁: 페이지 오른쪽 위에서 Actions 를 선택하고 드롭다운 메뉴에서 Add widgets를 선택해도 됩니다.

Add widgets 창이 표시됩니다.

27. Add widgets 메뉴에서 콘솔에 추가할 위젯을 선택하고 Add를 선택합니다.

페이지에 다음 메시지가 표시됩니다.

- Added the widget "Favorites". Find it at the bottom of your Console Home. Click and drag to reorder your widgets, or change a widget's size using the widget menu.

28. 위젯을 다시 정렬하려면 다음을 구성합니다.

- 위젯 위쪽에서 Favorites와 같은 **제목 표시줄**을 선택한 후 콘솔 페이지의 새 위치로 위젯을 드래그합니다.

29. 위젯 크기를 조정하려면 다음을 구성합니다.

- **Recently Visited** 위젯을 선택합니다.
- 위젯 오른쪽 위의 위젯 작업 줄임표 아이콘 선택합니다.
- **Change size**를 선택합니다.

Change size 창이 표시됩니다.

- **Change size** 메뉴에서 이 위젯에 사용할 View를 선택하고 Change size를 선택합니다.

참고: Welcome to AWS, Explore AWS, AWS Health 위젯의 크기는 조정할 수 없습니다.

30. 위젯을 제거하려면 다음을 구성합니다.

- **Welcome to AWS** 위젯을 선택합니다.
- 위젯 오른쪽 위의 위젯 작업 줄임표 아이콘을 선택합니다.
- **Remove widget**를 선택합니다.

축하합니다. AWS 관리 콘솔을 살펴보았으며 콘솔 홈 화면을 사용자 정의하는 방법을 알아보았습니다.

과제 2: AWS 관리 콘솔을 사용하여 Amazon S3 버킷 생성

이 작업에서는 AWS 관리 콘솔을 사용하여 *LabRegion*에서 새 Amazon S3 버킷을 생성 및 구성합니다.

- **추가 정보:** Amazon S3는 업계 최고의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다. 고객은 Amazon S3를 사용하여 데이터 레이크, 웹 사이트, 모바일 애플리케이션, 백업 및 복원, 아카이브, 엔터프라이즈 애플리케이션, 사물 인터넷(IoT) 디바이스, 빅 데이터 분석 등과 같은 다양한 사용 사례에서 원하는 만큼의 데이터를 저장하고 보호할 수 있습니다.



31. Services 메뉴에서 **All Services**를 선택합니다.
32. 왼쪽 탐색 메뉴에서 목록 아래쪽으로 스크롤한 다음 **Storage**를 선택합니다.
33. **Storage** 목록에서 **S3**를 선택합니다.

참고: 콘솔 위쪽의 통합 검색 창에서 **s3**를 검색해도 됩니다. 검색 창에는 *Search for services, features, marketplace products, and docs* 레이블이 표시되어 있습니다.

34. 콘솔 왼쪽에 있는 탐색 창에서 **Buckets** 선택합니다.
 35. **Create bucket**을 선택합니다.
- Create bucket** 페이지가 표시됩니다.
36. General configuration 섹션에서 **Bucket name**을 **labbucket-NUMBER**로 지정합니다.
- 버킷 이름 안의 **NUMBER**는 임의의 숫자로 바꿉니다. 그러면 고유한 이름이 생성됩니다.
- labbucket-987987 등의 버킷 이름을 사용할 수 있습니다.

Amazon S3 버킷 이름은 전역적으로 고유해야 하며 도메인 이름 시스템(DNS)을 준수해야 합니다. 전체 버킷 이름 지정 규칙은 공식 [버킷 이름 지정 규칙](#) 설명서를 참조하십시오.

37. **AWS Region**은 이 실습 지침 왼쪽에 있는 *LabRegion* 값과 일치해야 합니다.

38. 이 페이지의 나머지 설정은 모두 기본 구성 그대로 두십시오.

39. 화면 아래쪽에서 *Create bucket*을 선택합니다.

① **추가 정보:** 구현하려는 버킷은 Amazon S3 API를 사용하여 생성할 수도 있습니다. 하지만 이 작업에서는 API 대신 Amazon S3 콘솔을 사용하여 같은 작업을 수행했습니다. 콘솔은 Amazon S3 API를 사용하여 Amazon S3로 요청을 전송합니다.

페이지에 다음 메시지가 표시됩니다.

- Successfully created bucket "labbucket-xxxxx".

S3 콘솔이 표시됩니다. 계정의 모든 버킷 목록에 새로 생성된 버킷이 표시됩니다.

축하합니다. 기본 구성으로 새 Amazon S3 버킷을 생성했습니다.

과제 3: S3 콘솔을 사용하여 Amazon S3 버킷에 객체 업로드

이 작업에서는 S3 콘솔을 사용하여 앞에서 생성한 S3 버킷에 객체를 업로드합니다.

40. 컨텍스트 메뉴를 열려면 이 [이미지 링크](#)를 마우스 오른쪽 버튼으로 클릭하고 컴퓨터에 이미지를 저장하는 옵션을 선택합니다.

- *SampleFile.jpg*와 비슷한 파일 이름을 지정합니다.

참고: 파일 저장 방법은 웹 브라우저별로 다릅니다. 컨텍스트 메뉴에서 적절한 이름이 지정된 옵션을 선택하면 됩니다.

41. S3 콘솔에서 **labbucket-xxxxx** 버킷을 선택합니다.

42. Upload를 선택합니다.

Upload 페이지가 표시됩니다.

43. Add files를 선택합니다.

44. 앞에서 다운로드한 **SampleFile.jpg** 그림으로 이동하여 해당 그림을 선택합니다.

45. Upload를 선택합니다.

페이지에 다음 메시지가 표시됩니다.

- Upload succeeded

46. Close를 선택합니다.

축하합니다. Amazon S3 버킷에 객체를 업로드했습니다.

과제 4: AWS CLI를 사용하여 Amazon S3 버킷을 생성한 후 객체 업로드

이 작업에서는 AWS CLI를 사용하여 Amazon S3 버킷을 생성합니다. AWS CLI는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용하는 데 사용할 수 있는 오픈 소스 도구입니다.

과제 4.1: Session Manager를 사용하여 Command Host에 대한 연결 생성

이 실습에서 사용할 수 있도록 AWS CLI로 미리 구성된 Amazon EC2 인스턴스가 제공되었습니다. 해당 인스턴스의 이름은 *Command Host*입니다.



47. AWS 관리 콘솔에서 AWS 검색 창을 사용하여 EC2를 검색한 다음 결과 목록에서 해당 서비스를 선택합니다.

참고: Services 메뉴 오른쪽에 있는 통합 검색 창에는

Search for services, features, marketplace products, and docs 레이블이 표시되어 있습니다.

⚠️ 주의: 이 실습에서는 새 EC2 콘솔을 사용해야 합니다. 화면 왼쪽 위에 **New EC2 Experience**가 표시되면 **New EC2 Experience**가 선택되었는지 확인합니다.

48. 콘솔 왼쪽에 있는 탐색 창에서 **Instances**를 선택합니다.

49. **Command Host**를 선택합니다.

50. **Connect**를 선택합니다.

Connect to instance 페이지가 표시됩니다.

51. **Session Manager** 탭을 선택합니다.

- ① **추가 정보:** Session Manager를 사용하면 방화벽이나 Amazon Virtual Private Cloud(Amazon VPC) 보안 그룹에 SSH 포트를 표시하지 않고도 Amazon EC2 인스턴스에 연결할 수 있습니다. 이 기능에 대한 더 자세한 내용은 [AWS Systems Manager Session Manager](#)를 참조하십시오.

52. **Connect**를 선택합니다.

참고: 이 실습 지침 왼쪽의 **CommandHostSessionUrl** 값을 복사하여 새 브라우저 탭에 붙여넣어도 됩니다. Command Host 인스턴스의 터미널이 열립니다.

새 브라우저 탭이나 창이 열리고 Command Host 인스턴스에 대한 연결이 표시됩니다.

과제 4.2: AWS CLI에서 고급 S3 명령 사용

이 작업에서는 AWS CLI를 사용하여 Amazon S3의 고급 기능에 액세스합니다.

53. Command Host 세션에서 다음 명령을 입력합니다.

● **팁:** 명령을 복사하려면 명령 위에 마우스를 놓고 복사() 아이콘을 선택합니다. Command Host 세션에 명령을 붙여넣습니다.

- ① 다음 **ls** 명령은 사용자가 소유한 모든 버킷의 목록을 표시합니다.

```
aws s3 ls
```

54. 다음 명령을 텍스트 편집기에 복사하고 **NUMBER**를 버킷용으로 선택한 임의의 숫자로 바꾼 다음 Command Host 세션에 명령을 붙여넣습니다.

- ① 다음 **mb** 명령은 버킷을 생성합니다.

```
aws s3 mb s3://labclibucket-NUMBER
```

- 버킷 이름의 예로는 *labclibucket-787787* 등이 있습니다.

55. Command Host 세션에서 수정한 명령을 실행하려면 Enter 키를 누릅니다.

- 샘플 출력은 *make_bucket: labclibucket-xxxxx*와 같습니다.

참고: 이 실습에서는 지침을 쉽게 설명하기 위해 이 단계에서 실제로 선택하는 버킷 이름에 상관없이 지침 나머지 부분에서는 새로 생성된 이 버킷을 **labclibucket-**NUMBER****로 지칭합니다.

56. Command Host 세션에서 다음 명령을 입력합니다.

```
aws s3 ls
```

새로 생성한 버킷이 출력 목록에 표시됩니다.

57. 다음 명령을 텍스트 편집기에 복사하고 **labclibucket-**NUMBER****를 이전 단계에서 생성한 S3 버킷의 이름으로 바꾼 다음 Command Host 세션에 명령을 붙여넣습니다.

- ① 다음 **cp** 명령은 지정된 버킷에 파일 하나를 복사합니다.

```
aws s3 cp /home/ssm-user/HappyFace.jpg s3://labclibucket-NUMBER
```

58. Command Host 세션에서 수정한 명령을 실행하려면 Enter 키를 누릅니다.

- 샘플 출력은 *upload: ./HappyFace.jpg to s3://labclibucket-xxxxx/HappyFace.jpg*와 같습니다.

59. 다음 명령을 텍스트 편집기에 복사하고 *labclibucket-NUMBER*를 이전 단계에서 생성한 S3 버킷의 이름으로 바꾼 다음 Command Host 세션에 명령을 붙여넣습니다.

- 다음 **ls** 명령은 지정한 버킷에 있는 객체의 목록을 표시합니다.

```
aws s3 ls s3://labclibucket-NUMBER
```

새로 생성한 버킷에 업로드한 객체가 출력 목록에 표시됩니다. 브라우저 탭을 닫아도 됩니다.

이 작업에서 살펴보았듯이, 고급 Amazon S3 명령을 실행하면 Amazon S3 객체를 간편하게 관리할 수 있습니다. 이러한 명령을 사용하면 Amazon S3 내에서, 그리고 로컬 디렉터리에서 S3 콘텐츠를 관리할 수 있습니다. S3 명령은 S3 API 명령에 포함된 작업을 토대로 구축되었습니다.

축하합니다. AWS CLI를 사용하여 Amazon S3 버킷을 생성하고 해당 목록을 표시한 후 버킷에 객체를 복사했습니다.

완료

▶ 축하합니다!

이 실습에서는 다음 작업을 수행하는 방법을 알아보았습니다.

- AWS 관리 콘솔 살펴보기 및 사용
- AWS 관리 콘솔을 사용하여 리소스 생성
- AWS CLI 살펴보기 및 사용
- AWS CLI를 사용하여 리소스 생성

실습 완료

▣ 축하합니다! 실습을 완료했습니다.

실습 종료

다음 단계를 따라 콘솔을 닫고 실습을 종료한 후 실습 경험을 평가해 주십시오.

60. AWS Management Console로 돌아갑니다.

61. 탐색 모음에서 **awsstudent@<AccountNumber>** 를 선택한 다음 **Sign Out**을 선택합니다.

62. 실습 종료를 선택합니다.

63. OK를 선택합니다.

64. (선택 사항):

- 해당하는 별 개수를 선택합니다.
- 의견을 입력합니다.
- **Submit**을 선택합니다.
- 별 1개 = 매우 불만족
- 별 2개 = 불만족
- 별 3개 = 보통
- 별 4개 = 만족
- 별 5개 = 매우 만족

피드백을 제공하지 않으려면 그냥 창을 닫으면 됩니다.

AWS Training and Certification에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다.

피드백, 제안 사항 또는 수정 요청 사항을 제공하려면 AWS Training and Certification 문의 양식에 세부 정보를 입력해 주시기 바랍니다.

부록

참고 자료

- [AWS 관리 콘솔이란 무엇인가요?](#)
- [AWS Command Line Interface란 무엇인가요?](#)
- [AWS Systems Manager Session Manager](#)

DO NOT COPY
pink0569@naver.com



실습 2: Amazon VPC 인프라 구축

© 2022 Amazon Web Services, Inc. 또는 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 임대 또는 판매는 금지됩니다. 모든 상표는 해당 소유자의 자산입니다.

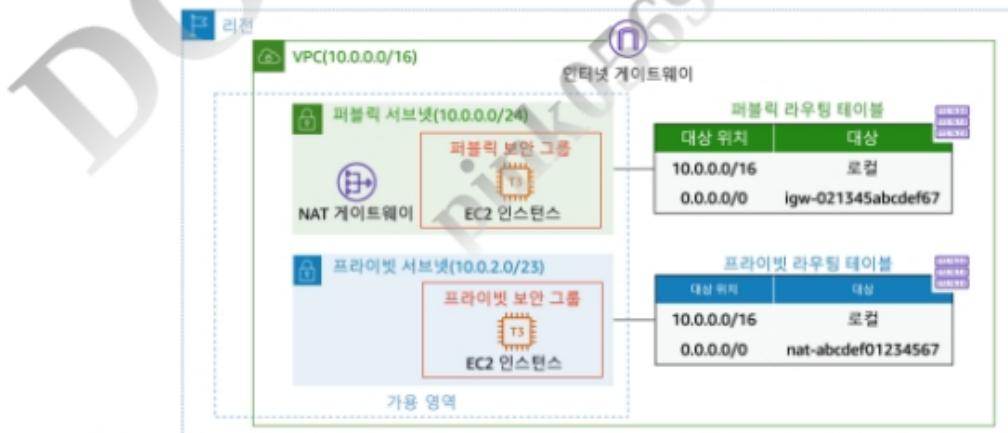
참고: 개인 정보, 개인을 식별할 수 있는 정보 또는 기밀 정보는 실습 환경에 입력하지 마십시오. 입력한 정보가 다른 사용자에게 공개될 수도 있습니다.

수정 사항이나 피드백 또는 기타 질문이 있으십니까? AWS Training and Certification에서 문의해 주십시오.

실습 개요

AWS 솔루션 아키텍트는 AWS의 전반적 기능과 AWS 네트워킹 구성 요소 간의 관계를 이해해야 합니다. 이 실습에서는 Amazon Virtual Private Cloud(VPC), 여러 사용 영역(AZ)에 걸친 서브넷, 퍼블릭 및 프라이빗 경로, NAT 게이트웨이, 인터넷 게이트웨이를 생성합니다. 이러한 서비스는 AWS 내부의 네트워킹 아키텍처의 기반입니다. 이 설계에는 인프라, 설계, 라우팅, 보안 개념이 포함됩니다.

다음 이미지는 이 실습 환경의 최종 아키텍처를 보여 줍니다.



목표

이 실습을 완료하면 다음을 할 수 있게 됩니다.

- Amazon VPC 생성
- 퍼블릭 및 프라이빗 서브넷 생성

- 인터넷 게이트웨이 생성
- 라우팅 테이블 구성 및 서브넷에 연결
- Amazon EC2 인스턴스를 생성하고 퍼블릭 액세스가 가능하도록 설정
- 프라이빗 서브넷에서 Amazon EC2 인스턴스 격리
- 보안 그룹을 생성하고 Amazon EC2 인스턴스에 할당
- 세션 관리자 도구를 사용하여 Amazon EC2 인스턴스에 연결

수강 전 권장 사항

본 실습에는 다음이 필요합니다.

- Microsoft Windows, macOS 또는 Linux(Ubuntu, SuSE, Red Hat)가 실행되는 Wi-Fi 지원 노트북
- 인터넷 브라우저(예: Chrome, Firefox 또는 Microsoft Edge)
- 일반 텍스트 편집기

소요 시간

이 실습을 완료하려면 최대 45분이 필요합니다.

시나리오

팀이 새로운 웹 기반 애플리케이션의 아키텍처 프로토타입 생성 업무를 맡았습니다. 아키텍처를 정의하려면 퍼블릭 및 프라이빗 서브넷, 라우팅, Amazon EC2 인스턴스 옵션에 대한 이해를 높여야 합니다.

실습 시작

- 실습을 시작하려면 페이지 상단에서 실습 시작을 선택합니다.

그러면 실습 리소스를 프로비저닝하는 프로세스가 시작됩니다. 실습 리소스를 프로비저닝하는데 걸리는 예상 시간이 표시됩니다. 계속 진행하기 전에 사용할 리소스가 프로비저닝될 때까지 기다려야 합니다.

① 토큰을 입력하라는 메시지가 표시되면 여러분에게 배포된 토큰(또는 구매한 크레딧)을 사용하세요.

- 실습을 열려면 콘솔 열기를 선택합니다.

새 웹 브라우저 탭에서 **AWS Management Console** 로그인 페이지가 열립니다.

3. Sign in as IAM user 페이지에서

- IAM user name에 awsstudent를 입력합니다.
- Password에 이 지침의 왼쪽에 나열된 Password 값을 복사하여 붙여넣습니다.
- Sign in을 선택합니다.

△ 별다른 지시가 없는 한 리전을 변경하지 마십시오.

일반적인 로그인 오류

오류: 우선 로그 아웃 필요

Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

You must first log out before logging into a different AWS account라는 메시지가 표시된다면 다음을 수행합니다.

- click here의 링크를 선택합니다.
- Amazon Web Services Sign In 웹 브라우저 탭을 닫고 초기 실습 페이지로 돌아갑니다.
- 콘솔 열기를 다시 선택합니다.

경우에 따라 일부 팝업 또는 스크립트 차단 웹 브라우저 확장 프로그램 때문에 실습 시작 버튼이 제대로 작동하지 않을 수 있습니다. 실습을 시작하는데 문제가 있는 경우 다음을 수행합니다.

- 팝업 또는 스크립트 차단 프로그램의 허용 목록에 실습 도메인 이름을 추가하거나 차단 프로그램을 끕니다.
- 페이지를 새로 고친 후 다시 시도하십시오.

과제 1: 리전에 Amazon VPC 생성

이 과제에서는 AWS Cloud에 새 Amazon VPC를 생성합니다.

① **추가 정보:** Amazon VPC를 사용하면 Amazon Web Services(AWS) Cloud에서 논리적으로 격리된 공간을 프로비저닝하고, 정의한 가상 네트워크에서 AWS 리소스를 시작할 수 있습니다. 자체 IP 주소 범위 선택, 서브넷 생성, 라우팅 테이블 및 네트워크 게이트웨이 구성 등 가상 네트워킹 환경을 완벽하게 제어할 수 있습니다. 또한 Amazon VPC의 향상된 보안 옵션을 활용하여 가상 네트워크의 Amazon EC2 인스턴스에 더 세분화된 액세스를 제공할 수 있습니다.



4. AWS 관리 콘솔의 AWS 검색 창에서 VPC를 검색하고 검색 결과 리스트에서 선택합니다.

주의: 콘솔의 오른쪽 상단에 표시된 리전이 실습 페이지 왼쪽에 있는 **Region** 값과 동일한지 확인합니다.

주의: 이 실습은 새로운 VPC 콘솔을 사용하도록 설계되었습니다. 화면 왼쪽 상단에 **New VPC Experience**가 표시되는 경우 **New VPC Experience**가 선택되었는지 확인합니다.

참고 VPC 관리 콘솔에는 몇 가지 VPC 아키텍처를 자동으로 생성할 수 있는 VPC 마법사가 있습니다. 그러나 이 실습에서는 VPC 구성 요소를 수동으로 생성합니다.

5. 왼쪽 탐색 창에서 **Your VPCs**를 선택합니다.

VPC 목록이 표시됩니다. 기본 VPC가 제공되므로 AWS 사용을 시작하자마자 리소스를 시작할 수 있습니다.

6. Create VPC를 선택하고 다음을 구성합니다.

- **Resources to create:** VPC only 선택
- **Name tag - optional:** Lab VPC
- **IPv4 CIDR block:** IPv4 CIDR manual input 선택
- **IPv4 CIDR:** 10.0.0.0/16

7. Create VPC 버튼을 선택합니다.

VPC Details 페이지가 보여집니다.

8. Lab VPC 상태를 확인합니다. 다음이 표시되어야 합니다.

- **State:** Available

① Lab VPC의 CIDR 범위는 **10.0.0.0/16**이며, **10.0.x.x**로 시작하는 모든 IP 주소가 포함됩니다. 이 범위에는 65,000개 이상의 주소가 포함됩니다. 나중에 이 주소를 별도의 서브넷으로 분할합니다.

9. 같은 페이지에서 Actions 를 선택하고 *Edit VPC settings*를 선택합니다.

이 옵션은 VPC의 Amazon EC2 인스턴스에 다음과 같은 친숙한 DNS 이름을 할당합니다.

`ec2-52-42-133-255.us-west-2.compute.amazonaws.com`

10. **Enable DNS hostnames** 옆의 체크박스를 선택합니다.

11. Save 버튼을 선택합니다.

이제 이 Amazon VPC에서 시작되는 모든 Amazon EC2 인스턴스는 DNS 호스트 이름을 자동으로 수신합니다. Amazon Route 53을 사용하여 나중에 좀 더 의미 있는 DNS 이름(예: `app.company.com`)을 추가할 수도 있습니다.

VPC가 성공적으로 생성되었고, 이제 정의된 이 가상 네트워크에서 AWS 리소스를 시작할 수 있습니다.

과제 2: 퍼블릭 및 프라이빗 서브넷 생성

이 과제에서는 Lab VPC에 퍼블릭 서브넷과 프라이빗 서브넷을 생성합니다. VPC에 새 서브넷을 추가하려면 VPC의 범위에서 서브넷의 IPv4 CIDR 블록을 지정해야 합니다. 서브넷을 위치할 가용 영역을 지정할 수 있습니다. 동일한 가용 영역에 여러 서브넷을 위치할 수 있습니다.



참고 서브넷은 네트워크 내 IP 주소의 하위 범위입니다. 지정된 서브넷에서 AWS 리소스를 시작할 수 있습니다. 인터넷에 연결해야 하는 리소스에는 퍼블릭 서브넷을 사용하고, 인터넷과 격리된 상태를 유지해야 하는 리소스에는 프라이빗 서브넷을 사용합니다.

과제 2.1: 퍼블릭 서브넷 생성

퍼블릭 서브넷은 인터넷과 연결되는 리소스에 사용됩니다.

12. 왼쪽 탐색 창에서 **Subnets**를 선택합니다.
13. Create subnet을 선택하고 다음을 구성합니다.
 - **VPC:** Lab VPC 선택
 - **Subnet name:** Public Subnet
 - **Availability Zone:** 목록에서 첫 번째 가용 영역을 선택합니다. *No Preference*를 선택하지 마십시오.
 - **IPv4 CIDR block:** 10.0.0.0/24
14. Create subnet 버튼을 선택합니다.
15. 상태를 확인합니다. 다음이 표시되어야 합니다.
 - **State:** Available

참고: VPC의 CIDR 범위는 **10.0.0.0/16**이며, 여기에는 모든 **10.0.x.x** IP 주소가 포함됩니다. 방금 생성한 서브넷의 CIDR 범위는 **10.0.0.0/24**이며, 모든 **10.0.0.x** IP 주소가 포함되어 있습니다. 이러한 범위는 서로 비슷해 보이지만 서브넷은 CIDR 범위가 /24이기 때문에 VPC보다 작습니다.

이제 서브넷 안에서 시작되는 모든 인스턴스에 퍼블릭 IP 주소가 자동으로 할당되도록 서브넷을 구성할 것입니다.

16. **Public Subnet**을 선택합니다.

17. Actions 를 선택하고 **Edit subnet settings**를 선택합니다.

18. Auto-assign IP settings 섹션에서 **Enable auto-assign public IPv4 address**의 체크 박스를 체크합니다.

19. Save 버튼을 선택합니다.

참고: 이 서브넷은 이름이 **Public Subnet**이지만 아직 퍼블릭 상태는 아닙니다. 퍼블릭 서브넷에는 인터넷 게이트웨이와 인터넷 게이트웨이까지의 경로가 있어야 합니다. 이 실습에서는 인터넷 게이트웨이와 라우팅 테이블을 생성하고 연결합니다.

과제 2.2: 프라이빗 서브넷 생성

프라이빗 서브넷은 인터넷과 격리된 상태를 유지해야 하는 리소스에 사용됩니다.

20. Create subnet을 선택하고 다음을 구성합니다.

- **VPC:** Lab VPC 선택
- **Subnet name:** Private Subnet
- **Availability Zone:** 목록에서 첫 번째 가용 영역을 선택합니다. *No Preference*를 선택하지 마십시오.
- **IPv4 CIDR block:** 10.0.2.0/23

21. Create subnet 버튼을 선택합니다.

22. 상태를 확인합니다. 다음이 표시되어야 합니다.

- **State:** Available

참고: CIDR 블록 **10.0.2.0/23**에는 **10.0.2.x** 및 **10.0.3.x**로 시작하는 모든 IP 주소가 포함되어 있습니다. 인터넷에서 액세스할 수 있어야 하는 특별한 경우를 제외하고 프라이빗 서브넷은 대부분의 리소스를 프라이빗으로 유지해야 하기 때문에 크기가 퍼블릭 서브넷의 두 배입니다.

이제 VPC에 서브넷이 2개 있습니다. 하지만 이러한 서브넷은 격리되어 있고 VPC 밖의 리소스와 통신할 수 없습니다. 다음으로 인터넷 게이트웨이를 통해 인터넷에 연결되도록 퍼블릭 서브넷을 구성합니다.

과제 3: 인터넷 게이트웨이 생성

이 과제에서는 인터넷 트래픽이 퍼블릭 서브넷에 액세스할 수 있도록 인터넷 게이트웨이를 생성합니다. VPC의 서브넷에 있는 인스턴스의 인터넷 액세스가 가능하도록 하려면 인터넷 게이트웨이를 생성하여 VPC에 연결합니다. 그런 다음 인터넷 바운드 트래픽을 인터넷 게이트웨이로 보내는 경로를 서브넷의 라우팅 테이블에 추가합니다.

① **추가 정보:** 인터넷 게이트웨이는 VPC 라우팅 테이블에서 인터넷 라우팅 트래픽에 대한 대상을 제공하고 퍼블릭 IPv4 주소가 할당된 인스턴스에 대해 네트워크 주소 변환(NAT)을 수행하는 두 가지 용도로 사용됩니다.

23. 왼쪽 탐색 창에서 **Internet Gateways**를 선택합니다.

24. Create internet gateway를 선택하고 다음을 구성합니다.

- **Name tag:** Lab IGW

25. Create internet gateway 버튼을 선택합니다.

이제 Lab VPC에 인터넷 게이트웨이를 연결할 수 있습니다.

26. 같은 페이지에서 Actions ▾를 선택하고 **Attach to VPC**를 선택합니다.

27. Available VPCs에서 **Lab VPC**를 선택합니다.

28. Attach internet gateway 버튼을 선택합니다.

29. 상태를 확인합니다. 다음이 표시되어야 합니다.

- **State: Attached**

이제 인터넷 게이트웨이가 Lab VPC에 연결되었습니다. 인터넷 게이트웨이를 생성하여 VPC에 연결했어도 퍼블릭 서브넷 라우팅 테이블도 인터넷 게이트웨이를 사용하도록 구성해야 합니다.

과제 4: 퍼블릭 서브넷의 인터넷 트래픽을 인터넷 게이트웨이로 라우팅

이 과제에서는 라우팅 테이블을 생성하고 인터넷 바운드 트래픽을 인터넷 게이트웨이로 보내는 경로를 라우팅 테이블에 추가한 다음 퍼블릭 서브넷을 라우팅 테이블과 연결합니다. VPC에 있는 각 서브넷은 라우팅 테이블에 연결되어 있어야 합니다. 라우팅 테이블이 서브넷에 대한 라우팅을 제어합니다. 서브넷은 한 번에 하나의 라우팅 테이블에만 연결할 수 있지만, 여러 서브넷을 같은 라우팅 테이블에 연결할 수 있습니다.

- **추가 정보:** 라우팅 테이블은 네트워크 트래픽이 전달되는 위치를 결정하는 데 사용되는 경로라는 규칙 세트를 포함합니다.

인터넷 게이트웨이를 사용하려면 인터넷 바운드 트래픽을 인터넷 게이트웨이로 보내는 경로가 서브넷의 라우팅 테이블에 포함되어야 합니다. 라우팅 테이블에 명시적으로 알려지지 않은 모든 대상으로 경로의 범위를 지정하거나(IPv4는 0.0.0.0/0, IPv6는 ::/0) 이보다 좁은 범위의 IP 주소로 경로의 범위를 지정할 수 있습니다. 인터넷 게이트웨이로 가는 경로가 있는 라우팅 테이블에 연결된 서브넷을 퍼블릭 서브넷이라고 합니다.

30. 왼쪽 탐색 창에서 **Route Tables**를 선택합니다.

현재 **Lab VPC**와 연결된 기본 라우팅 테이블이 하나 있습니다. 이 라우팅 테이블은 트래픽을 로컬로 라우팅합니다. 이제 퍼블릭 트래픽을 인터넷 게이트웨이로 라우팅하는 추가 라우팅 테이블을 생성합니다.

31. Create route table을 선택하고 다음을 구성합니다.

- **Name:** Public Route Table
- **VPC:** Lab VPC 선택

32. Create route table 버튼을 선택합니다.

33. 페이지 하단에서 **Routes** 탭을 선택합니다.

라우팅 테이블에는 10.0.0.0/16 네트워크 내의 트래픽이 네트워크 내에서 흐르도록 허용하는 경로가 하나 있지만 이 경로는 트래픽을 네트워크 외부로 라우팅하지는 않습니다. 이제 새 경로를 추가하여 퍼블릭 트래픽을 활성화합니다.

34. Edit routes를 선택합니다.

35. Add route를 선택하고 다음을 구성합니다.

- **Destination:** 0.0.0.0/0
- **Target:** 드롭다운에서 **Internet Gateway**를 선택한 다음 표시된 **Internet Gateway ID**를 선택합니다.

36. Save changes 버튼을 선택합니다.

37. **Subnet Associations** 탭을 선택합니다.

38. Edit subnet associations 버튼을 선택합니다.

39. **Public Subnet**체크 박스를 선택합니다.

40. Save associations 버튼을 선택합니다.

라우팅 테이블이 구성되었습니다. 이 서브넷은 인터넷 게이트웨이를 통한 인터넷 경로가 있기 때문에 이제 **퍼블릭**입니다.

DO NOT COPY
pink0569@naver.com

과제 5: 퍼블릭 보안 그룹 생성

이 과제에서는 사용자들이 Amazon EC2 인스턴스에 액세스할 수 있도록 보안 그룹을 생성합니다. VPC의 보안 그룹은 Amazon EC2 인스턴스에 허용되는 트래픽을 지정합니다.

① **추가 정보:** Amazon EC2 보안 그룹을 사용하면 Amazon VPC 내 인스턴스의 보안을 유지할 수 있습니다. VPC의 보안 그룹을 통해 각 Amazon EC2 인스턴스에 허용되는 인바운드 및 아웃바운드 네트워크 트래픽을 지정할 수 있습니다. 인스턴스에 명시적으로 허용되지 않은 트래픽은 자동으로 거부됩니다.

보안: HTTPS 프로토콜이 보안 강화를 위해 권장되지만 실습을 간단히 하기 위해 HTTP 프로토콜을 사용합니다.

41. 왼쪽 탐색 창에서 **Security Groups**를 선택합니다.

42. Create security group을 선택하고 다음을 구성합니다.

- **Security group name:** Public SG
- **Description:** Allows incoming traffic to public instance
- **VPC:** X를 선택하여 텍스트 상자를 지운 다음 드롭다운 메뉴에서 *Lab VPC*를 선택합니다.

43. **Inbound rules** 섹션에서 다음을 수행합니다.

- Add rule 버튼을 선택합니다.
- **Type:** HTTP
- **Source:** Anywhere-IPv4

44. **Tags** 섹션에서 다음을 수행합니다.

- Add new tag 버튼을 선택합니다.
- **Key:** Name
- **Value:** Public SG

45. Create security group 버튼을 선택합니다.

HTTP 트래픽을 허용하는 보안 그룹을 성공적으로 생성했습니다. 이 보안 그룹은 퍼블릭 서브넷에서 Amazon EC2 인스턴스를 시작하는 다음 과제에 필요합니다.

과제 6: 퍼블릭 서브넷에서 Amazon EC2 인스턴스 시작

이 과제에서는 퍼블릭 서브넷에서 Amazon EC2 인스턴스를 시작합니다. 인터넷을 통한 IPv4 통신이 가능하려면 인스턴스에 프라이빗 IPv4 주소와 연결된 퍼블릭 IPv4 주소가 있어야 합니다. 기본적으로 인스턴스는 VPC와 서브넷 안에서 정의된 프라이빗(내부) IP 주소 공간만 인식합니다.



○ **추가 정보:** 논리적으로 생성된 인터넷 게이트웨이는 인스턴스를 위한 일대일 NAT를 제공하므로 트래픽이 VPC 서브넷에서 나가 인터넷으로 갈 때 응답 주소 필드는 프라이빗 IP 주소가 아니라 인스턴스의 퍼블릭 IPv4 주소 또는 탄력적 IP 주소로 설정됩니다.

46. AWS 관리 콘솔의 AWS 검색 창에서 EC2를 검색하고 검색 결과 리스트에서 선택합니다.

△ 주의: 이 실습은 새로운 EC2 콘솔을 사용하도록 설계되었습니다. 화면 원쪽 상단에 **New EC2 Experience**가 표시되는 경우 **New EC2 Experience**가 선택되었는지 확인합니다.

Amazon EC2 관리 콘솔이 보여집니다.

과제 6.1: 인스턴스 구성 시작하기

47. 왼편의 콘솔 네비게이션 메뉴에서 **EC2 Dashboard**를 선택합니다.

48. **Launch instance** 섹션의 드롭다운 버튼 **Launch instance ▾**를 선택합니다.

49. 리스트에서 **Launch instance**를 선택합니다.

Launch an instance 페이지가 표시됩니다.

과제 6.2: 인스턴스에 태그 추가

태그는 AWS 리소스를 목적, 소유자, 환경에 따라서 카테고리로 나눌 수 있게 해줍니다. 태그는 대부분의 AWS 클라우드 리소스에 적용 가능합니다. 각 태그는 *Key*와 *_Value_*를 가지고 있고,

둘 다 사용자가 정의합니다. 태그는 같은 유형의 여러개 리소스를 관리해야 할 때 사용할 수 있습니다. 적용한 태그를 이용해서 특정 리소스를 빠르게 검색하고 식별할 수 있습니다.

이번 작업에서 EC2 인스턴스에 태그를 추가합니다.

50. **Name and tags** 섹션으로 이동합니다.

51. **Name**의 값을 `Public Instance`으로 입력합니다.

이 이름은 추후 Amazon EC2 관리 콘솔의 인스턴스에 표시됩니다.

이 실습을 위한 추가적인 인스턴스 태그는 필요하지 않습니다.

과제 6.3: Amazon Machine Image 선택

이 작업에서 Amazon Machine Image (AMI)를 선택합니다. AMI는 인스턴스를 시작하기 위한 디스크 볼륨 사본을 포함합니다.

52. **Application and OS Images (Amazon Machine Image)** 섹션으로 이동합니다.

53. **Amazon Linux** 운영 체제가 선택되었는지 확인합니다.

54. 드롭다운 메뉴에서 **Amazon Linux 2 AMI**가 선택되었는지 확인합니다.

과제 6.4: Amazon EC2 인스턴스 유형 선택

각 인스턴스 유형은 가상 CPU, 메모리, 디스크 스토리지, 네트워크 성능의 조합으로 할당되어 있습니다.

이 실습은 **t3.micro** 인스턴스 유형을 사용합니다. 이 인스턴스 유형은 2 vCPU 와 1 GiB 메모리를 가지고 있습니다.

55. **Instance type** 섹션으로 이동합니다.

56. **Instance type** 드롭다운 메뉴에서 **t3.micro** 를 선택합니다.

과제 6.5: 로그인을 위한 key pair 구성

57. **Key pair (login)** 섹션으로 이동합니다.

58. **Key pair name - required** 드랍 다운 메뉴에서

`Proceed without a key pair (Not recommended) ▾` 을 선택합니다.

과제 6.6: 인스턴스 네트워크 구성

59. **Network settings** 섹션으로 이동합니다.

60. **Edit** 버튼을 선택합니다.

61. 아래와 같이 설정합니다:

- **VPC - required:** Lab VPC
- **Subnet:** Public Subnet
- **Auto-assign public IP:** Enable

과제 6.7: 인스턴스의 보안 그룹 구성

보안 그룹을 이용해 Elastic Network Interface (ENI)로 향하는 인바운드/아웃바운드 트래픽을 허용하거나 거부하도록 정의할 수 있습니다. ENI는 인스턴스에 연결됩니다. 80번 포트는 HTTP 트래픽의 기본 포트이며, 이 실습에서 시작할 웹서버가 정상 작동하기 위해 필요합니다.

62. Select existing security group 버튼을 선택합니다.

63. Common security groups 드롭다운 메뉴에서 **Public SG** 와 같은 이름을 가진 보안그룹을 선택합니다.

과제 6.8: 스토리지 추가

Configure storage 섹션에서 인스턴스에 연결될 Amazon Elastic Block Store (EBS) 디스크 볼륨과 instance storage를 수정할 수 있습니다. EBS 볼륨은 성능과 사이즈 모두를 설정 가능합니다.

이 실습에서는 기본적 스토리지 설정으로 진행합니다. 변경이 필요하지 않습니다.

과제 6.9: 사용자 데이터 구성

64. ▶ Advanced Details 섹션으로 이동해서 확장합니다.

65. IAM instance profile 드롭다운 메뉴에서 **EC2InstProfile** 역할을 선택합니다.

참고: 웹 서버로 사용할 새 인스턴스를 설치하고 설정하기 위해서 사용자 데이터 스크립트를 작성하고 인스턴스 런치시에 자동으로 실행되도록 할 수 있습니다.

66. User data 항목에 다음을 붙여 넣습니다:

```
#!/bin/bash

# To connect to your EC2 instance and install the Apache web server with PHP

yum update -y &&
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2 &&
yum install -y httpd &&
systemctl enable httpd.service
systemctl start httpd
cd /var/www/html
wget https://us-west-2-tcprod.s3.amazonaws.com/courses/ILT-TF-200-ARCHIT/v7.2.6/lab-2-VPC/scripts/instanceData.zip
unzip instanceData.zip
```

나머지 설정은 기본 설정으로 그대로 둡니다.

과제 6.10: instance launch 리뷰

시작할 Amazon EC2 인스턴스가 제대로 설정되었는지 확인합니다.

67. **Summary** 섹션으로 이동합니다.

68. Launch instance 버튼을 선택합니다.

Launch an instance 페이지가 보여집니다.

이제 Amazon EC2 인스턴스가 시작되었고 지정한 대로 구성되었습니다.

69. View all instances 버튼을 선택합니다.

Amazon EC2 관리 콘솔이 보여집니다.

70. 새로 고침 버튼을 클릭하고 **Public Instance**의 **Instance State** 가 **Running**일 때까지 기다립니다.

참고 **Public Instance** Amazon EC2 인스턴스는 *pending* 상태로 시작됩니다. 인스턴스의 부팅이 완료되면 인스턴스의 상태가 **Running**로 기다린 후 **2/2 checks passed** 상태 점검이 통과하기를 기다립니다.

축하합니다. 퍼블릭 서브넷에서 Amazon EC2 인스턴스를 성공적으로 시작했습니다.

과제 7: HTTP를 통해 퍼블릭 인스턴스에 연결

이 과제에서는 퍼블릭 인스턴스에 연결하고 기본 Apache 웹 서버 페이지를 시작합니다. 앞서 추가한 HTTP 액세스(포트 80)를 허용하는 인바운드 규칙은 Apache를 실행하는 웹 서버에 연결하는 것을 허용합니다.

71. 왼쪽 탐색 창에서 **Instances**를 선택합니다.
72. **Public Instance**를 선택합니다.
73. 아래 쪽에서 **Networking** 탭을 선택합니다.
참고: 콘솔의 섹션을 더 크게 만들어야 하는 경우 콘솔에 표시되는 컨테이너의 수평 가장자리 크기를 조정할 수 있습니다.
74. **Public IPv4 DNS**로 이동합니다.
75. public DNS 값을 복사합니다. open address 를 선택하지 마세요. 이 실습 환경에서는 HTTPS이 설정되어 있지 않습니다.
76. 새 웹 브라우저 탭을 열고 *Public Instance*의 public DNS 값을 주소 표시줄에 붙여넣습니다. Amazon EC2 인스턴스에서 호스트되는 웹 페이지가 표시됩니다. 이 페이지에는 인스턴스 ID와 Amazon EC2 인스턴스가 있는 AWS 가용 영역이 표시됩니다.

퍼블릭 서브넷에서 Apache 웹 서버를 성공적으로 시작하고 HTTP 연결을 테스트했습니다.

과제 8: 퍼블릭 서브넷의 Amazon EC2 인스턴스에 연결

이 과제에서는 세션 관리자를 사용하여 퍼블릭 서브넷에 있는 Amazon EC2 인스턴스에 연결합니다.

① **추가 정보:** 세션 관리자는 대화형 월클릭 브라우저 기반 셀 또는 AWS CLI를 통해 Amazon EC2 인스턴스를 관리할 수 있는 완전관리형 AWS Systems Manager 기능입니다. 세션 관리자를 사용하면 계정에서 Amazon EC2 인스턴스로 세션을 시작할 수 있습니다. 세션이 시작된 후 다른 연결 유형을 통해 실행하는 것처럼 bash 명령을 실행할 수 있습니다.

77. 왼쪽 탐색 창에서 **Instances**를 선택합니다.

78. **Public Instance**를 선택한 다음 **Connect** 버튼을 선택합니다.

Connect to instance 페이지가 표시됩니다.

79. **Connection method**에서 **Session Manager** 탭을 선택합니다.

① 세션 관리자를 사용하면 방화벽 또는 Amazon Virtual Private Cloud(Amazon VPC) 보안 그룹에서 SSH 포트를 노출하지 않고도 Amazon EC2 인스턴스에 연결할 수 있습니다. 자세한 내용은 [AWS Systems Manager 세션 관리자](#)를 참조하십시오.

80. **Connect** 버튼을 선택합니다.

Public Instance로 연결되는 새 브라우저 탭 또는 창이 열립니다.

참고: 세션 관리자 서비스는 실시간으로 업데이트되지 않습니다. 방금 시작한 Amazon EC2 인스턴스에 연결할 때 세션 관리자 오류가 발생하는 경우 인스턴스가 시작되고 상태 확인을 통과하여 세션 관리자 서비스와 통신할 때까지 몇 분 정도 기다린 다음 세션 연결을 다시 열어 보십시오.

81. 다음 명령을 입력하여 홈 디렉터리(/home/ssm-user/)로 이동하고 cURL 명령을 사용하여 웹 연결을 테스트합니다.

```
cd ~
curl -I https://aws.amazon.com/training/
```

샘플 출력:

```
HTTP/2 200
content-type: text/html; charset=UTF-8
server: Server
date: Fri, 14 May 2021 14:30:15 GMT
x-amz-rid: 3WNTZRMGBGMP8AP6HT4K7
set-cookie: aws-priv=eyJ2IjoxLCJldSI6MCwic3QiObj9; Version=1; Comment="Anonymous
cookie for privacy regulations"; Domain=.aws.amazon.com; Max-Age=31536000;
Expires=Sat, 14-May-2022 14:30:15 GMT; Path=/
set-cookie: aws_lang=en; Domain=.amazon.com; Path=/
x-frame-options: SAMEORIGIN
x-content-type-options: nosniff
x-amz-id-1: 3WNTZRMGBGMP8AP6HT4K7
last-modified: Tue, 11 May 2021 17:39:32 GMT
```

```
vary: accept-encoding,Content-Type,Accept-Encoding,X-Amzn-CDN-Cache,X-Amzn-AX-Treatment,User-Agent
x-cache: Miss from cloudfront
via: 1.1 86561b4243b7d0478ca4582dd013e00e.cloudfront.net (CloudFront)
x-amz-cf-pop: ATL52-C1
x-amz-cf-id: 3VxbxlST1HIeEd4sdFtNepfu7jnEjQB-4gqet8mmAL5mbU0sQolzJw==
```

세션 관리자를 사용하여 성공적으로 퍼블릭 인스턴스에 연결했습니다. 탭을 닫고 콘솔로 돌아가도 됩니다.

DO NOT COPY
pink0569@naver.com

과제 9: NAT 게이트웨이를 생성하고 프라이빗 서브넷에서 라우팅 구성

이 과제에서는 NAT 게이트웨이를 생성한 다음 라우팅 테이블을 생성하여 비로컬 트래픽을 NAT 게이트웨이로 라우팅합니다. 그런 다음 라우팅 테이블을 프라이빗 서브넷에 연결합니다. NAT 게이트웨이를 사용하여 프라이빗 서브넷의 인스턴스를 인터넷 또는 기타 AWS 서비스에 연결하는 한편, 인터넷에서 해당 인스턴스와의 연결을 시작하지 못하도록 할 수 있습니다.

참고: NAT 게이트웨이를 생성하려면 NAT 게이트웨이가 속할 퍼블릭 서브넷을 지정해야 합니다. NAT 게이트웨이를 생성할 때 NAT 게이트웨이와 연결할 탄력적 IP 주소도 지정해야 합니다. 탄력적 IP 주소는 NAT 게이트웨이에 연결한 후 변경할 수 없습니다. NAT 게이트웨이를 생성한 후에는 인터넷 바운드 트래픽이 NAT 게이트웨이를 가리키도록 하나 이상의 프라이빗 서브넷과 연결된 라우팅 테이블을 업데이트해야 합니다. 그러면 프라이빗 서브넷의 인스턴스가 인터넷과 통신할 수 있습니다.

82. AWS 관리 콘솔이 열려 있는 탭으로 이동합니다.
83. AWS 관리 콘솔의 AWS 검색 창에서 VPC를 검색하고 검색 결과 리스트에서 선택합니다.

84. 왼쪽 탐색 창에서 **NAT Gateways**를 선택합니다.

85. Create NAT gateway를 선택하고 다음을 구성합니다.

- **Name:** Lab NGW
- **Subnet:** Public Subnet을 선택합니다.
- **Allocate Elastic IP:** 선택합니다.

86. Create NAT gateway를 선택합니다.

다음 단계에서는 비로컬 트래픽을 NAT 게이트웨이로 리디렉션하는 프라이빗 서브넷의 새 라우팅 테이블을 생성합니다.

87. 왼쪽 탐색 창에서 **Route Tables**를 선택합니다.

88. Create route table를 선택하고 다음을 구성합니다.

- **Name:** Private Route Table
- **VPC:** Lab VPC를 선택합니다.
- **Create route table:** 버튼을 선택합니다.

프라이빗 라우팅 테이블이 생성되고 프라이빗 라우팅 테이블의 세부 정보 페이지가 표시됩니다.

89. **Routes** 탭을 선택합니다.

현재는 모든 트래픽을 로컬로 보내는 경로 하나가 있습니다.

이제 NAT 게이트웨이를 통해 인터넷 바운드 트래픽을 보내는 경로를 추가합니다.

90. Edit routes 버튼을 선택한 후 다음을 수행합니다.

- Add route 버튼을 선택합니다.
- **Destination:** 0.0.0.0/0
- **Target:** 드롭다운에서 **NAT Gateway**를 선택한 다음 표시된 **NAT Gateway ID**를 선택합니다.
- 그런 다음 Save changes 버튼을 선택합니다.

91. **Subnet Associations** 탭을 선택합니다.

92. Edit subnet associations 버튼을 선택합니다.

93. **Private Subnet**을 선택합니다.

94. Save associations 버튼을 선택합니다.

이 경로는 프라이빗 서브넷의 인터넷 바운드 트래픽을 동일한 가용 영역에 있는 NAT 게이트웨이로 보냅니다.

성공적으로 NAT 게이트웨이를 생성하고 프라이빗 라우팅 테이블을 구성했습니다.

과제 10: 프라이빗 리소스용 보안 그룹 생성

이 과제에서는 퍼블릭 보안 그룹에 할당된 리소스에서 들어오는 HTTPS 트래픽을 허용하는 보안 그룹을 생성합니다.

① **추가 정보:** 보안 그룹을 규칙의 소스로 지정하면 지정된 프로토콜과 포트의 경우, 소스 보안 그룹과 연결된 네트워크 인터페이스에서 오는 트래픽이 허용됩니다. 들어오는 트래픽은 퍼블릭 IP 주소 또는 탄력적 IP 주소가 아닌 소스 보안 그룹과 연결된 네트워크 인터페이스의 프라이빗 IP 주소를 기반으로 허용됩니다. 보안 그룹을 소스로 추가해도 소스 보안 그룹의 규칙이 추가되지는 않습니다.

95. 왼쪽 탐색 창에서 **Security Groups**를 선택합니다.

96. Create security group을 선택하고 다음을 구성합니다.

- **Security group name:** Private SG
- **Description:** Allows incoming traffic to private instance using public security group
- **VPC:** X를 선택하여 텍스트 상자를 지운 다음 드롭다운 메뉴에서 *Lab VPC*를 선택합니다.

97. **Inbound rules** 섹션에서 다음을 수행합니다.

- Add rule 버튼을 선택합니다.
- **Type:** HTTPS

- **Source type:** Custom
- **Source:**
 - Custom 오른쪽에 있는 상자에 sg를 입력합니다.
 - 표시되는 목록에서 Public SG 를 선택합니다.

98. Tags 섹션에서 다음을 수행합니다.

- Add new tag 버튼을 선택합니다.
- **Key:** Name
- **Value:** Private SG

99. Create security group 버튼을 선택합니다.

성공적으로 프라이빗 보안 그룹을 생성했습니다.

DO NOT COPY
pink0569@naver.com

과제 11: 프라이빗 서브넷에서 Amazon EC2 인스턴스 시작

이 과제에서는 프라이빗 서브넷에서 Amazon EC2 인스턴스를 시작합니다.

① **추가 정보:** 프라이빗 인스턴스는 NAT 게이트웨이 또는 NAT 인스턴스를 통해 트래픽을 라우팅하여 인터넷에 액세스할 수 있습니다. 프라이빗 인스턴스는 인터넷을 통과하기 위해 NAT 게이트웨이 또는 NAT 인스턴스의 퍼블릭 IP 주소를 사용합니다. NAT 게이트웨이 또는 NAT 인스턴스는 아웃바운드 통신을 허용하지만, 인터넷상에서 시스템이 프라이빗 주소가 지정된 인스턴스에 연결을 시작하는 것은 허용하지 않습니다.

100. AWS 관리 콘솔의 AWS 검색 창에서 EC2를 검색하고 검색 결과 리스트에서 선택합니다.

△ 화면 왼쪽 상단에 **New EC2 Experience**가 표시되는 경우 **New EC2 Experience**가 선택되었는지 확인합니다. 이 실습은 새로운 EC2 콘솔을 사용하도록 설계되었습니다.

Amazon EC2 콘솔이 보여집니다.

과제 11.1: 인스턴스 구성 시작하기

101. 원편의 콘솔 네비게이션 메뉴에서 **EC2 Dashboard**를 선택합니다.

102. **Launch instance** 섹션의 Launch instance ▼ 드롭다운 버튼을 선택합니다.

103. 리스트에서 Launch instance를 선택합니다.

Launch an instance 페이지가 표시됩니다.

과제 11.2: 인스턴스에 태그 추가

이번 작업에서 EC2 인스턴스에 태그를 추가합니다.

104. **Name and tags** 섹션으로 이동합니다.

105. **Name**의 값을 Private Instance으로 입력합니다.

Amazon EC2 관리 콘솔에 인스턴스의 이 이름이 나타날 것입니다.

이 실습을 위한 추가적인 인스턴스 태그는 필요하지 않습니다.

과제 11.3: Amazon Machine Image 선택

이 작업에서 Amazon Machine Image (AMI)를 선택합니다. AMI는 인스턴스를 시작하기 위한 디스크 볼륨 사본을 포함합니다.

106. **Application and OS Images (Amazon Machine Image)** 섹션으로 이동합니다.

107. **Amazon Linux** 운영 체제가 선택되었는지 확인합니다.
108. 드랍 다운 메뉴에서 **Amazon Linux 2 AMI**가 선택되었는지 확인합니다.

과제 11.4: Amazon EC2 인스턴스 유형 선택

각 인스턴스 유형은 가상 CPUs, 메모리, 디스크 스토리지, 네트워크 성능의 조합으로 할당되어 있습니다.

이 실습은 **t3.micro** 인스턴스 유형을 사용합니다. 이 인스턴스 유형은 2 vCPUs 와 1 GiB 메모리를 가지고 있습니다.

109. **Instance type** 섹션으로 이동합니다.
110. *Instance type* 드롭다운 메뉴에서 **t3.micro** 를 선택합니다.

과제 11.5: 로그인을 위한 key pair 구성

111. **Key pair (login)** 섹션으로 이동합니다.
112. **Key pair name - required** 드롭다운 메뉴에서 *Proceed without a key pair (Not recommended) ▾* 을 선택합니다.

과제 11.6: 인스턴스 네트워크 구성

113. **Network settings** 섹션으로 이동합니다.
114. **Edit** 버튼을 선택합니다.
115. 드롭다운 메뉴에서 아래와 같이 설정합니다:
 - **VPC - required:** *Lab VPC*
 - **Subnet:** *Private Subnet*
 - **Auto-assign public IP:** *Disable*

과제 11.7: 인스턴스의 보안 그룹 구성

116. **Select existing security group** 버튼을 선택합니다.
117. **Common security groups** 드랍 다운 메뉴에서 **Private SG** 와 같은 이름을 가진 보안그룹을 선택합니다.

과제 11.8: Add storage

Configure storage 섹션에서 인스턴스에 연결될 Amazon Elastic Block Store (EBS) 디스크 볼륨과 instance storage를 수정할 수 있습니다. EBS 볼륨은 성능과 사이즈 모두를 설정 가능합니다.

이 실습에서는 기본적 스토리지 설정으로 진행합니다. 변경이 필요하지 않습니다.

과제 11.9: 사용자 데이터 구성

118. ► Advanced Details 섹션으로 이동해서 확장합니다.
119. IAM instance profile 드롭다운 메뉴에서 **EC2InstProfile** 역할을 선택합니다.

나머지 설정은 기본 설정으로 그대로 둡니다.

과제 11.10: instance launch 리뷰

시작할 Amazon EC2 인스턴스가 제대로 설정되었는지 확인합니다.

120. Summary 섹션으로 이동합니다.
121. Launch instance 버튼을 선택합니다.

Launch an instance 페이지가 보여집니다.

이제 Amazon EC2 인스턴스가 시작되었고 지정한 대로 구성되었습니다.

122. View all instances 버튼을 선택합니다.

Amazon EC2 콘솔이 보여집니다

참고 *Private Instance* Amazon EC2 인스턴스는 *pending* 상태로 시작됩니다. 인스턴스의 부팅이 완료되면 인스턴스의 상태가 **Running**로 변경됩니다.

123. 새로 고침 버튼을 간간히 누르며 *Public Instance*의 **Instance State** 가 **Running**일 때까지 기다립니다.

축하합니다. 프라이빗 서브넷에서 Amazon EC2 인스턴스를 성공적으로 시작했습니다.

과제 12: 프라이빗 서브넷의 Amazon EC2 인스턴스에 연결

이 과제에서는 세션 관리자를 사용하여 프라이빗 서브넷에 있는 Amazon EC2 인스턴스에 연결합니다.

124. 왼쪽 탐색 창에서 **Instances**를 선택합니다.
125. **Private Instance**를 선택한 다음 **Connect**를 선택합니다.

Connect to instance 페이지가 표시됩니다.

126. **Connection method**에서 **Session Manager** 탭을 선택합니다.
127. **Connect** 버튼을 선택합니다.

프라이빗 인스턴스로 연결되는 새로운 브라우저 탭 또는 창이 열립니다.

참고: 세션 관리자 서비스는 실시간으로 업데이트되지 않습니다. 방금 시작한 Amazon EC2 인스턴스에 연결할 때 세션 관리자 오류가 발생하는 경우 인스턴스가 시작되고 상태 확인을 통과하여 세션 관리자 서비스와 통신할 때까지 몇 분 정도 기다린 다음 세션 연결을 다시 열어 보십시오.

128. 다음 명령을 입력하여 홈 디렉터리(/home/ssm-user/)로 이동하고 cURL 명령을 사용하여 웹 연결을 테스트합니다.

```
cd ~
curl -I https://aws.amazon.com/training/
```

샘플 출력:

```
HTTP/2 200
content-type: text/html; charset=UTF-8
server: Server
date: Fri, 14 May 2021 14:30:15 GMT
x-amz-rid: 3WNTZRMGBGMP8AP6HT4K7
set-cookie: aws-priv=eyJ2IjoxLCJldSI6MCwic3QiOjB9; Version=1; Comment="Anonymous
cookie for privacy regulations"; Domain=.aws.amazon.com; Max-Age=31536000;
Expires=Sat, 14-May-2022 14:30:15 GMT; Path=/
set-cookie: aws_lang=en; Domain=.amazon.com; Path=/
x-frame-options: SAMEORIGIN
x-content-type-options: nosniff
x-amz-id-1: 3WNTZRMGBGMP8AP6HT4K7
last-modified: Tue, 11 May 2021 17:39:32 GMT
vary: accept-encoding,Content-Type,Accept-Encoding,X-Amzn-CDN-Cache,X-Amzn-AX-
Treatment,User-Agent
x-cache: Miss from cloudfront
via: 1.1 86561b4243b7d0478ca4582dd013e00e.cloudfront.net (CloudFront)
x-amz-cf-pop: ATL52-C1
x-amz-cf-id: 3VxbxlST1HIeEd4sdFtNepfu7jnEjQB-4gqet8mmAL5mbU0sQolzJw==
```

세션 관리자를 사용하여 성공적으로 프라이빗 인스턴스에 연결했습니다. 탭을 닫고 콘솔로 돌아가도 됩니다.

DO NOT COPY
pink0569@naver.com

선택적 과제 1: 퍼블릭 인스턴스에서 프라이빗 인스턴스 연결 테스트

이 선택적 과제에서는 ICMP(Internet Control Message Protocol)를 사용하여 퍼블릭 인스턴스에서 프라이빗 인스턴스의 네트워크 연결성을 검증합니다.

참고: 이 과제는 선택 사항이며 실습 시간이 남는 경우에 제공됩니다. 이 과제를 완료하거나 여기를 선택하여 실습 끝으로 건너뛸 수 있습니다.

129. AWS 관리 콘솔이 열려 있는 탭으로 이동합니다.

130. 왼쪽 탐색 창에서 **Instances**를 선택합니다.

131. **Private Instance**를 선택합니다.

132. **Details** 탭에서 Private IPv4 address를 클립보드에 복사합니다.

▶ **팁** *Private IPv4 addresses*를 복사하려면 마우스로 가리키고 복사 아이콘을 선택합니다.

133. **Private Instance** 선택을 취소합니다.

134. **Public Instance**를 선택합니다.

135. **Connect** 버튼을 선택합니다.

Connect to instance 페이지가 표시됩니다.

136. **Session Manager** 탭을 선택합니다.

137. **Connect** 버튼을 선택합니다.

Public Instance로 연결되는 새 브라우저 탭 또는 창이 열립니다.

138. 다음 명령을 메모장에 복사합니다. <private_ip>를 **Private IPv4 addresses**의 값으로 바꿉니다.

`ping <private_ip>`

139. 업데이트된 명령을 복사해 터미널에 붙여넣고 Enter 키를 누릅니다.

샘플 명령: 다음 명령은 사용하지 마십시오.

`ping 10.0.2.131`

140. 몇 초 후 CTRL+C를 눌러 ICMP ping 요청을 중지합니다.

프라이빗 인스턴스에 대한 ping 요청이 실패합니다. 도전 과제는 콘솔을 사용하여 프라이빗 인스턴스에 성공적으로 ping을 수행하기 위해 **Private SG**에 필요한 인바운드 규칙을 파악하는 것입니다.

선택적 과제를 완료하는 데 문제가 있다면 실습 마지막 부분의 선택적 과제 해법 섹션을 참조하십시오.

DO NOT COPY
pink0569@naver.com

선택적 과제 2: 인스턴스 메타데이터 검색

이 선택적 과제에서는 cURL과 같은 도구를 사용하여 AWS CLI에서 인스턴스 메타데이터 명령을 실행합니다. 인스턴스 메타데이터는 Amazon EC2 인스턴스를 실행하면 사용할 수 있습니다. 이는 Amazon EC2 인스턴스에서 실행할 스크립트를 작성할 때 유용합니다.

참고: 이 과제는 선택 사항이며 실습 시간이 남는 경우에 제공됩니다. 이 과제를 완료하거나 여기를 선택하여 실습 끝으로 건너뛸 수 있습니다.

141. AWS 관리 콘솔이 열려 있는 탭으로 이동합니다.
142. 왼쪽 탐색 창에서 **Instances**를 선택합니다.
143. **Public Instance**를 선택합니다.
144. **Connect** 버튼을 선택합니다.

Connect to instance 페이지가 표시됩니다.

145. **Session Manager** 탭을 선택합니다.
146. **Connect** 버튼을 선택합니다.

Public Instance로 연결되는 새 브라우저 탭 또는 창이 열립니다.

147. 실행 중인 인스턴스 내에서 인스턴스 메타데이터의 모든 범주를 보려면 다음 명령을 실행합니다.

```
curl http://169.254.169.254/latest/meta-data/
```

148. 다음 명령을 실행하여 public-hostname(이전 명령에서 가져온 최상위 메타데이터 항목 중 하나)을 검색합니다.

```
curl http://169.254.169.254/latest/meta-data/public-hostname
```

참고 IP 주소 169.254.169.254는 링크-로컬 주소이며, 이 인스턴스에서만 유효합니다.

실행 중인 Amazon EC2 인스턴스에서 인스턴스 메타데이터를 검색하는 방법을 배웠습니다.

결론

퍼블릭 서브넷과 프라이빗 서브넷이 모두 있는 VPC를 생성하면 퍼블릭 서브넷이나 프라이빗 서브넷에서 작업과 서비스를 시작할 수 있는 유연성을 얻을 수 있습니다. 프라이빗 서브넷의 작업과 서비스는 NAT 게이트웨이를 통해 인터넷에 액세스할 수 있습니다.

이 실습에서는 다음을 수행하는 방법을 배웠습니다.

- Amazon VPC 생성
- 퍼블릭 및 프라이빗 서브넷 생성
- 인터넷 게이트웨이 생성
- 라우팅 테이블 구성 및 서브넷에 연결
- Amazon EC2 인스턴스를 생성하고 퍼블릭 액세스가 가능하도록 설정
- 프라이빗 서브넷에서 Amazon EC2 인스턴스 격리
- 보안 그룹을 생성하고 Amazon EC2 인스턴스에 할당
- 세션 관리자 도구를 사용하여 Amazon EC2 인스턴스에 연결

DO NOT COPY
pink0569@naver.com

실습 종료

다음 단계를 따라 콘솔을 닫고 실습을 종료한 후 실습 경험을 평가해 주십시오.

149. AWS Management Console로 돌아갑니다.

150. 탐색 모음에서 **awsstudent@<AccountNumber>** 를 선택한 다음 **Sign Out**을 선택합니다.

151. 실습 종료를 선택합니다.

152. OK를 선택합니다.

153. (선택 사항):

- 해당하는 별 개수를 선택합니다.
- 의견을 입력합니다.
- **Submit**을 선택합니다.
- 별 1개 = 매우 불만족
- 별 2개 = 불만족
- 별 3개 = 보통
- 별 4개 = 만족
- 별 5개 = 매우 만족

피드백을 제공하지 않으려면 그냥 창을 닫으면 됩니다.

선택적 과제 해법

154. AWS 관리 콘솔이 열려 있는 탭으로 이동합니다.
155. AWS 관리 콘솔의 AWS 검색 창에서 EC2를 검색하고 검색 결과 리스트에서 선택합니다.
156. 왼쪽 탐색 창에서 **Security Groups**를 선택합니다.
157. Private SG를 선택합니다.
158. Actions를 선택한 다음 **Edit inbound rules**를 선택합니다.
159. **Edit inbound rules** 페이지의 *Inbound rules*에서 다음을 수행합니다.
 - Add rule 버튼을 선택합니다.
 - Type: Custom ICMP - IPV4
 - Source:
 - Custom 오른쪽에 있는 상자에 sg를 입력합니다.
 - 표시되는 목록에서 Public SG를 선택합니다.
160. Save rules 버튼을 선택합니다.
161. 다음 링크 (여기)를 선택하여 **Optional Task**로 이동하고 단계를 다시 실행합니다. 이제 퍼블릭 인스턴스가 프라이빗 인스턴스에 성공적으로 ping을 수행할 수 있습니다.

추가 리소스

- [VPC 소개](#)
- [서브넷](#)
- [인터넷 게이트웨이](#)
- [라우팅 테이블](#)
- [VPC 보안 그룹](#)
- [NAT 게이트웨이](#)
- [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름](#)
- [AWS에서 IPv6 네트워킹의 기본 사항 이해](#)

DO NOT COPY
pink0569@naver.com

AWS Training and Certification에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다.

피드백, 제안 사항 또는 수정 요청 사항을 제공하려면 AWS Training and Certification 문의 양식에 세부 정보를 입력해 주시기 바랍니다.



실습 3: Amazon VPC 인프라에 데이터베이스 계층 생성

© 2022 Amazon Web Services, Inc. 또는 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다.
상업적인 복제, 임대 또는 판매는 금지됩니다. 모든 상표는 해당 소유자의 자산입니다.

참고: 개인 정보, 개인을 식별할 수 있는 정보 또는 기밀 정보는 실습 환경에 입력하지 마십시오.
입력한 정보가 다른 사용자에게 공개될 수도 있습니다.

수정 사항이나 피드백 또는 기타 질문이 있으십니까? AWS Training and Certification에서 문의해
주십시오.

실습 개요

어떤 환경에서나 백엔드 데이터베이스는 중요한 역할을 하며, 이 중요한 리소스의 보안과
액세스는 어떤 아키텍처에서도 매우 중요합니다. 이 실습에서는 Amazon Aurora DB 클러스터를
생성하여 MySQL 데이터베이스와 Application Load Balancer(ALB)를 관리합니다. AWS 보안
핵심 요소에서는 사람들을 데이터에서 떨어뜨려 놓을 것을 권장하므로 데이터베이스는 ALB를
사용하여 프런트엔드에서 분리됩니다. ALB는 프런트엔드 애플리케이션을 호스트하는 정상 EC2
인스턴스로 트래픽을 라우팅함으로써 고가용성을 제공하고 ALB 뒤의 프라이빗 서브넷에서
데이터베이스와의 통신이 이루어지도록 합니다.

목표

이 실습을 마치면 다음을 수행할 수 있습니다.

- Amazon RDS 데이터베이스 생성
- Application Load Balancer 생성
- Application Load Balancer용 HTTP 리스너 생성
- 대상 그룹 생성
- 대상 그룹에 대상 등록
- 로드 밸런서 테스트 및 데이터베이스와 애플리케이션의 커넥션 테스트
- 콘솔을 통해 Amazon RDS DB 인스턴스 메타데이터 검토
- 과제(옵션): 다른 AWS 리전에 Amazon RDS 읽기 전용 복제본 생성

수강 전 권장 사항

본 실습에는 다음이 필요합니다.

- Microsoft Windows, macOS 또는 Linux(Ubuntu, SuSE, Red Hat)가 실행되는 Wi-Fi 지원 노트북
- 인터넷 브라우저(예: Chrome, Firefox 또는 Microsoft Edge)
- 일반 텍스트 편집기

소요 시간

이 실습을 완료하는 데는 약 **45분**이 소요됩니다.

본 실습에서 사용하지 않는 AWS 서비스

이 실습에서 사용하지 않는 AWS 서비스는 실습 환경에서 비활성화됩니다. 또한 이 실습에 사용되는 서비스의 기능은 실습에 필요한 작업으로 제한됩니다. 다른 서비스에 액세스하거나 실습 안내서에서 제공하는 것 외의 작업을 수행하는 경우 오류가 발생할 수 있습니다.

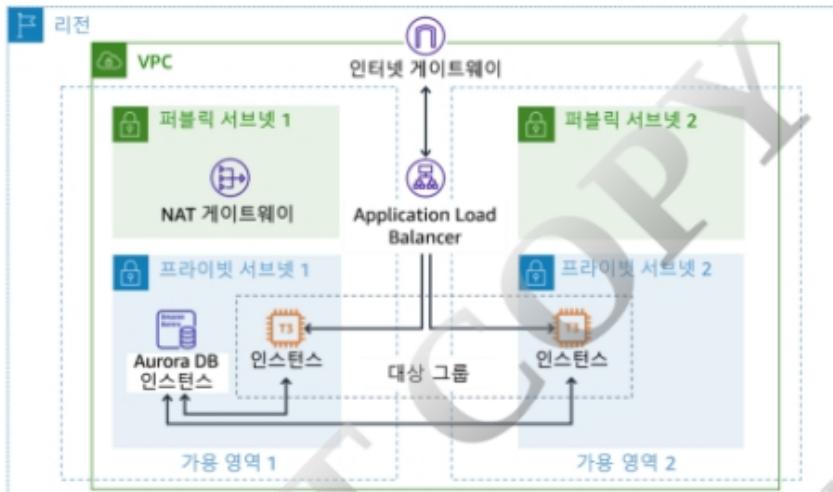
시나리오

팀이 새로운 웹 기반 애플리케이션의 아키텍처 프로토타입 생성 업무를 맡았습니다. 아키텍처를 정의하려면 로드 밸런서와 Amazon RDS 같은 관리형 데이터베이스에 대한 이해를 높여야 합니다.

실습 환경

시작할 수 있도록 실습 환경에서 다음 리소스가 제공됩니다. Amazon Virtual Private Cloud(Amazon VPC), 필요한 기본 네트워크 구조, HTTP 포트를 허용하는 보안 그룹, Amazon EC2 인스턴스, 연결된 Amazon EC2 인스턴스 프로파일. 인스턴스 프로파일에는 AWS Systems Manager 세션 관리자 기능이 Amazon EC2 인스턴스에 액세스하도록 허용하는 데 필요한 권한이 포함되어 있습니다.

다음 다이어그램은 구축할 중요 실습 리소스의 예상 아키텍처와 실습이 끝날 때 이러한 리소스가 어떻게 연결되어야 하는지 보여 줍니다.



실습 시작

1. 실습을 시작하려면 페이지 상단에서 실습 시작을 선택합니다.

그러면 실습 리소스를 프로비저닝하는 프로세스가 시작됩니다. 실습 리소스를 프로비저닝하는데 걸리는 예상 시간이 표시됩니다. 계속 진행하기 전에 사용할 리소스가 프로비저닝될 때까지 기다려야 합니다.

- ① 토큰을 입력하라는 메시지가 표시되면 여러분에게 배포된 토큰(또는 구매한 크레딧)을 사용하세요.

2. 실습을 열려면 콘솔 열기를 선택합니다.

새 웹 브라우저 탭에서 **AWS Management Console** 로그인 페이지가 열립니다.

3. **Sign in as IAM user** 페이지에서

- **IAM user name**에 awsstudent를 입력합니다.
- **Password**에 이 지침의 왼쪽에 나열된 **Password** 값을 복사하여 붙여넣습니다.
- **Sign in**을 선택합니다.

△ 별다른 지시가 없는 한 리전을 변경하지 마십시오.

일반적인 로그인 오류

오류: 우선 로그 아웃 필요

Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

You must first log out before logging into a different AWS account라는 메시지가 표시된다면 다음을 수행합니다.

- **click here**의 링크를 선택합니다.
- **Amazon Web Services Sign In** 웹 브라우저 탭을 닫고 초기 실습 페이지로 돌아갑니다.
- 콘솔 열기를 다시 선택합니다.

경우에 따라 일부 팝업 또는 스크립트 차단 웹 브라우저 확장 프로그램 때문에 **실습 시작** 버튼이 제대로 작동하지 않을 수 있습니다. 실습을 시작하는데 문제가 있는 경우 다음을 수행합니다.

- 팝업 또는 스크립트 차단 프로그램의 허용 목록에 실습 도메인 이름을 추가하거나 차단 프로그램을 끕니다.
- 페이지를 새로 고친 후 다시 시도하십시오.

과제 1: Amazon RDS 데이터베이스 생성

이 실습에서는 MySQL과 호환되는 Amazon Aurora DB 클러스터를 생성합니다. Amazon Aurora DB 클러스터는 하나 이상의 DB 인스턴스와 해당 DB 인스턴스에 대한 데이터를 관리하는 클러스터 볼륨으로 구성됩니다.

① Amazon Aurora(Aurora)는 MySQL 및 PostgreSQL과 호환되는 완전관리형 관계형 데이터베이스 엔진입니다. Aurora는 관리형 데이터베이스 서비스 Amazon Relational Database Service(Amazon RDS)에 속합니다. Amazon RDS는 클라우드에서 관계형 데이터베이스를 보다 쉽게 설치, 운영, 크기 조정할 수 있는 웹 서비스입니다.

4. AWS 관리 콘솔의 AWS 검색 창에서 RDS를 검색하고 검색 결과 리스트에서 선택합니다.

참고: 페이지 중앙 상단의 통합 검색 창에서 이름으로 검색하여 AWS 관리 콘솔에서 서비스를 찾을 수도 있습니다. 통합 검색 창은 Services 메뉴 오른쪽에 있으며,

다음과 같이 레이블이 지정되어 있습니다.

Search for services, features, marketplace products, and docs

5. 왼쪽 탐색 창에서 Databases를 선택합니다.

6. Create database 버튼을 선택합니다.

7. Choose a database creation method에서 Standard Create를 선택합니다.

8. Engine options 주 섹션에서 다음을 구성합니다.

- Engine type에서 Amazon Aurora를 선택합니다.
- Edition에서 Amazon Aurora MySQL-Compatible Edition를 선택합니다.

9. Templates 주 섹션에서 Dev/Test를 선택합니다.

10. Settings 주 섹션에서 다음을 구성합니다.

- DB cluster identifier: aurora
- Master username: dbadmin
- Master password: admin123
- Confirm password: admin123

11. Instance configuration 주 섹션에서 다음을 구성합니다.

- DB instance class: Burstable classes를 선택합니다.
- db.t3.small

12. Availability & durability 섹션의 Multi-AZ deployment에서 Don't create an Aurora Replica를 선택합니다.

① Amazon RDS 다중 AZ 배포는 데이터베이스(DB) 인스턴스의 가용성 및 내구성을 높여주므로 프로덕션 데이터베이스 워크로드에 적합합니다. 다중 AZ DB 인스턴스를 프로비저닝하면 Amazon RDS는 기본 DB 인스턴스를 자동 생성함과 동시에 다른 가용 영역(AZ)에 있는 예비 인스턴스에 데이터를 복제합니다.

참고 이 실습에서 중요한 것은 멀티 티어 아키텍처 구축에 필요한 리소스를 아는 것이므로 다중 AZ 배포를 수행할 필요는 없습니다. 다중 AZ 아키텍처 배포 방법은 다음 실습에서 배웁니다.

13. Connectivity 주 섹션에서 다음을 구성합니다.

- **Virtual Private Cloud (VPC): LabVPC**
- **Subnet group: labdbsubnetgroup**
- **Public access: No**
- **VPC security group: Choose existing**
- **Existing VPC security groups:**
 - **Existing VPC security groups** 필드에서 **default** 보안 그룹을 제거하기 위해 **X**를 선택합니다.
 - **Existing VPC security groups** 드롭다운 메뉴에서 **LabDBSecurityGroup**을 선택합니다.

① 서브넷은 보안 및 운영상 필요에 따라 리소스를 그룹화하기 위해 지정하는 Amazon VPC IP 주소 범위의 세그먼트입니다. DB 서브넷 그룹은 사용자가 Amazon VPC에서 생성한 다음 DB 인스턴스에 대해 지정하는 서브넷(일반적으로 프라이빗)의 모음입니다. DB 서브넷 그룹에서는 CLI 또는 API를 사용하여 DB 인스턴스를 생성할 때 Amazon VPC를 지정할 수 있습니다. 콘솔을 사용하는 경우 사용할 Amazon VPC와 서브넷만 선택할 수 있습니다.

② Amazon Virtual Private Cloud(Amazon VPC)를 사용하면 정의한 가상 네트워크에서 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

14. ▶ Additional configuration 섹션을 클릭해서 확장합니다.

- **Database port:** 기본 값으로 구성합니다.

15. Monitoring 섹션에서 ▶ **Enable Enhanced monitoring**을 선택 취소합니다.

16. 페이지 끝에 있는 ▶ Additional configuration 섹션을 확장합니다.

17. Database options 섹션에서 다음을 구성합니다.

- **Initial database name:** `inventory`
- **DB cluster parameter group:** 페이지 왼쪽의 **DBClusterParameterGroup** 값과 일치하는 값을 드롭다운 메뉴에서 선택합니다.

⚠ 주의 드롭다운 메뉴에서 올바른 **DB cluster parameter group** 값을 선택해야 합니다. 잘못된 값을 선택하면 데이터베이스 복제본을 구축할 때 오류가 발생합니다.

18. **Encryption** 섹션에서 **Enable encryption**을 선택 취소합니다.

① Amazon RDS DB 인스턴스에 대해 암호화 옵션을 활성화하여 저장 중 Amazon RDS 인스턴스와 스냅샷을 암호화할 수 있습니다. 저장 중 암호화되는 데이터에는 DB 인스턴스의 기본 스토리지, 자동 백업, 읽기 전용 복제본, 스냅샷이 포함됩니다.

19. **Maintenance** 섹션에서 **Enable auto minor version upgrade**를 선택 취소합니다.

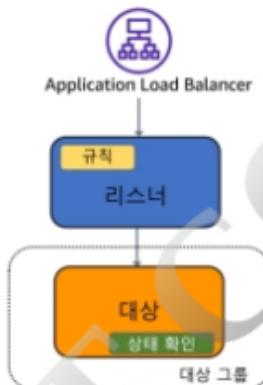
20. 화면 맨 아래로 스크롤한 다음 **Create database** 버튼을 선택합니다.

Aurora MySQL DB 클러스터 시작 프로세스가 진행됩니다. Amazon RDS 인스턴스가 시작되려면 최대 5분이 걸릴 수 있습니다. 하지만 기다리지 않고 다음 과제를 계속할 수 있습니다.

과제 2: Application Load Balancer를 생성하고 설정하기

이 과제에서는 퍼블릭 서브넷에 Application Load Balancer를 생성하여 브라우저에서 애플리케이션에 액세스합니다. Amazon EC2 콘솔로 이동하여 기존 Amazon VPC 인프라에 Application Load Balancer를 생성하고 프라이빗 Amazon EC2 인스턴스를 대상으로 추가합니다.

일반적으로 Amazon EC2 인스턴스 앞에 로드 밸런서가 배포되어 웹 애플리케이션에 액세스하는 단일 위치를 사용자에게 제공하고 사용자 요청의 부하를 분산합니다.



과제 2.1 : target group 생성하기

이번 과제에서, target group을 생성하고 target group에서 대상들을 등록합니다. 기본적으로, 로드 밸런서는 받은 요청을 target group을 위해 설정한 포트와 프로토콜을 기반으로 등록된 대상에게 전송합니다.

21. AWS 관리 콘솔의 Services 메뉴에서 **EC2**를 선택합니다.
 22. 왼쪽 탐색 창에서 **Target Groups**를 선택합니다.
 23. Create target group 버튼을 선택합니다.
- Specify group details** 페이지가 표시됩니다.
24. **Basic configuration** 섹션에서 다음을 구성합니다.
 - **Choose a target type: Instances**
 - **Target group name:** ALBTtargetGroup
 - **VPC:** LabVPC
- 페이지의 나머지 설정은 기본값 그대로 두어도 됩니다.
25. Next 버튼을 선택합니다.

Register targets 페이지가 표시됩니다.

26. **Available Instances** 섹션에서 다음을 구성합니다.

- 이름이 **AppServer1** 그리고 **AppServer2**인 Amazon EC2 인스턴스를 선택합니다.
- **Include as pending below** 버튼을 선택합니다.

페이지의 **Targets** 섹션 아래에 인스턴스가 표시됩니다.

27. **Create target group** 버튼을 선택합니다.

28. 다음과 같은 메시지가 표시됩니다.

- Successfully created target group: ALBTargetGroup

과제 2.2 : Application Load Balancer 생성

이번 과제에서, Application Load Balancer를 생성하고 이름, 스키마, IP 주소 유형과 같이 로드 밸런서에 필요한 기본적인 환경 구성은 설정합니다. 그 다음 네트워크, 리스너 정보를 등록합니다.

29. 왼쪽 탐색 창에서 **Load Balancers**를 선택합니다.

30. **Create Load Balancer** 버튼을 선택합니다.

31. **Application Load Balancer** 섹션에서 **Create** 버튼을 선택합니다.

Create Application Load Balancer 페이지가 표시됩니다.

32. **Basic Configuration** 섹션에서 다음을 구성합니다.

- **Load balancer name**에 LabAppALB를 입력합니다.

33. **Network mapping** 섹션에서 다음을 구성합니다.

- **VPC**: LabVPC를 선택합니다.

- **Mappings:**

- 나열된 첫 번째 가용 영역의 확인란을 선택하고 서브넷 목록에서 Public Subnet1을 선택합니다.
 - 나열된 두 번째 가용 영역의 확인란을 선택하고 서브넷 목록에서 Public Subnet2를 선택합니다.

34. **Security groups** 섹션에서 다음을 구성합니다.

- **default** 보안 그룹을 제거합니다. (X를 클릭합니다.)
- 드롭다운 메뉴에서 **LabALBSecurityGroup**을 선택합니다.

35. **Listeners and routing** 섹션에서 다음을 구성합니다.

- **Listener HTTP:80:** Default action 섹션의 드롭다운 메뉴에서 **ALBTargetGroup**을 선택합니다.

36. Create load balancer 버튼을 선택합니다.

37. 다음과 같은 메시지가 표시됩니다.

- Successfully created load balancer:LabAppALB

로드 밸런서 **LabAppALB**가 성공적으로 생성되었습니다.

38. **View load balancer** 버튼을 선택합니다.

로드 밸런서가 몇 분 동안 *provisioning* 상태였다가 *active*로 바뀝니다.

① 이 과제에서는 단일 Amazon EC2 인스턴스를 로드 밸런서에 대상으로 추가했습니다. 이 과제에서는 대상을 로드 밸런서에 등록하는 방법을 보여줍니다. 개별 Amazon EC2 인스턴스 외에 Auto Scaling 그룹도 로드 밸런서의 대상으로 등록할 수 있습니다. Auto Scaling 그룹을 로드 밸런싱의 대상으로 사용하는 경우 Auto Scaling 그룹이 시작하는 인스턴스는 로드 밸런서에 자동으로 등록됩니다. 마찬가지로 Auto Scaling 그룹이 종료하는 Amazon EC2 인스턴스는 로드 밸런서에서 자동으로 등록 취소됩니다. 로드 밸런서와 함께 Auto Scaling 그룹을 사용하는 방법은 다음 실습에서 설명합니다.

성공적으로 로드 밸런서를 생성하고 프라이빗 서브넷의 Amazon EC2 인스턴스를 대상으로 추가했습니다.

과제 3: 콘솔을 통해 Amazon RDS DB 인스턴스 메타데이터 검토

이 과제에서는 Amazon RDS 콘솔을 탐색하여 과제 1에서 생성한 인스턴스가 완료되었고 활성 상태인지 확인합니다. AWS 관리 콘솔을 탐색하여 DB 인스턴스의 연결 정보를 찾는 방법을 알아봅니다. DB 인스턴스의 연결 정보에는 엔드포인트, 포트, 유효한 데이터베이스 사용자가 포함됩니다.

39. AWS 관리 콘솔의 Services 메뉴에서 **RDS**를 선택합니다.
40. 탐색 창에서 **Databases**를 선택합니다.
41. DB 식별자 목록에서 이름이 **aurora**인 클러스터의 링크를 선택합니다.
데이터베이스 세부 정보가 포함된 페이지가 표시됩니다.
42. **Connectivity & security** 탭에서 데이터베이스 클러스터의 엔드포인트 및 포트 번호를 찾을 수 있습니다. 일반적으로 데이터베이스에 연결하려면 엔드포인트와 포트 번호가 모두 필요합니다.
43. **writer** 인스턴스의 **Endpoint** 값을 복사한 후, 텍스트 편집기를 열어서 붙여넣기 합니다. 이후 Endpoint 값이 필요합니다.

Endpoint는 aurora.crwxbgqad61a.rds.amazonaws.com과 비슷할 것입니다.

▶ Tip writer 인스턴스의 *Endpoint*를 복사하기 위해서, 복사 아이콘을 클릭하십시오.

엔드포인트의 상태는 **Available**입니다.

44. **Configuration** 탭에서 데이터베이스의 현재 구성에 관한 세부 정보를 찾을 수 있습니다.
45. **Monitoring** 탭에서 데이터베이스의 다음 항목에 대한 지표를 모니터링할 수 있습니다.
 - 데이터베이스 인스턴스에 대한 연결 수
 - 데이터베이스 인스턴스에 대한 읽기 및 쓰기 작업의 양
 - 데이터베이스 인스턴스가 현재 사용 중인 스토리지의 양
 - 데이터베이스 인스턴스에 사용 중인 메모리 및 CPU의 양
 - 데이터베이스 인스턴스에 들어오고 나가는 네트워크 트래픽 양

과제 4: 로드 밸런서 테스트

이 과제에서는 Application Load Balancer URL을 식별하고, 로드 밸런서를 통해 간단한 HTTP 요청을 실행하여 Amazon EC2 인스턴스에 설치된 웹 애플리케이션을 시작합니다.

46. AWS 관리 콘솔의 Services 메뉴에서 **EC2**를 선택합니다.

47. 왼쪽 탐색 창에서 **Target Groups**를 선택합니다.

48. **ALBTargetGroup**을 선택합니다.

49. **Targets** 탭에서 인스턴스 상태가 **healthy**로 표시될 때까지 기다립니다.

① ELB는 웹 서비스 인스턴스의 ping 경로를 주기적으로 테스트하여 상태를 확인합니다. 200 HTTP 응답 코드는 정상 상태를 나타내고, 그 밖의 응답 코드는 비정상 상태를 나타냅니다. 인스턴스가 비정상이고 일정 횟수의 연속 검사(비정상 임계값)에서 해당 상태가 지속되는 경우 로드 밸런서는 복구될 때까지 해당 인스턴스를 서비스에서 제거합니다.

50. 왼쪽 탐색 창에서 **Load Balancers**를 선택합니다.

51. 이름이 **LabAppALB**인 로드 밸런서의 확인란을 선택합니다.

52. **Description** 탭에서 **DNS Name**을 복사하고 새 브라우저 탭에 값을 붙여넣어 로드 밸런서를 호출합니다.

● 팁 *DNS name*을 복사하기 위해서, 마우스를 *DNS name*에 올려 놓고 복사 아이콘을 선택합니다.

다음과 같은 웹 페이지가 표시됩니다.



53. ② **Settings** 탭을 선택하고 설정합니다 :

- **Endpoint:** 이전에 복사한 *writer instance endpoint*을 붙여넣습니다.
- **Database:** *inventory*
- **Username:** *dbadmin*

- **Password:** admin123

54. **Save** 버튼을 선택합니다.

애플리케이션이 데이터베이스에 연결되어, 기본 데이터를 가져오고 정보를 화면에 나타냅니다. 이 애플리케이션을 통해서 재고 항목을 추가, 수정, 삭제할 수 있습니다.

재고 정보는 이 실습의 앞부분에서 생성한 Amazon RDS MySQL 데이터베이스에 저장됩니다. 웹 애플리케이션 서버에 장애가 발생하더라도 해당 데이터를 잃지 않습니다. 또한, 다른 애플리케이션 서버에서 이 데이터에 접근할 수 있습니다.

Amazon EC2 인스턴스에 설치된 웹 애플리케이션에 로드 밸런서를 통해 성공적으로 액세스했습니다.

선택적 과제: 다른 AWS 리전에 Amazon RDS 읽기 전용 복제본 생성

이 도전 과제에서는 소스 DB 인스턴스에서 교차 리전 읽기 전용 복제본을 생성합니다. 재해 복구 기능을 개선하거나, 읽기 작업을 사용자와 더욱 가까운 AWS 리전으로 확장하거나, 한 리전의 데이터 센터에서 다른 리전의 데이터 센터로 보다 쉽게 마이그레이션하기 위해 다른 AWS 리전에 읽기 전용 복제본을 생성합니다.

참고: 이 도전 과제는 선택 사항이며 실습 시간이 남는 경우에 제공됩니다. 이 과제를 완료하거나 여기를 선택하여 실습 끝으로 건너뛸 수 있습니다.

55. AWS 관리 콘솔의 AWS 검색 창에서 RDS를 검색하고 검색 결과 리스트에서 선택합니다.

56. 왼쪽 탐색 창에서 Databases를 선택합니다.

57. aurora DB 인스턴스를 선택합니다.

58. Actions ▾ 버튼을 선택하고 Create cross region read replica를 선택합니다.

59. Instance specifications 섹션에서 다음을 구성합니다.

- Multi-AZ deployment에서 No를 선택합니다.
- 이 섹션의 나머지 설정은 기본값 그대로 두어도 됩니다.

60. Network & Security 섹션에서 다음을 구성합니다.

- Destination region에서 실습 지침 왼쪽의 RemoteRegion 값과 일치하는 리전을 선택합니다.
- Publicly accessible에서 No를 선택합니다.
- VPC security groups에서
 - default 보안그룹을 제거하기 위해, X를 선택합니다.
 - 드롭다운 메뉴에서 LabDBSecurityGroup을 선택합니다.
- 이 섹션의 나머지 설정은 기본값 그대로 두어도 됩니다.

61. Settings 섹션에서 다음을 구성합니다.

- DB instance identifier에 LabDBreplica를 입력합니다.
- 이 섹션의 나머지 설정은 기본값 그대로 두어도 됩니다.

62. Performance Insights 섹션에서 다음을 구성합니다.

- Disable Performance Insights를 선택합니다.
- 이 섹션의 나머지 설정은 기본값 그대로 두어도 됩니다.

참고 페이지의 나머지 설정은 기본값 그대로 두어도 됩니다.

63. Create 버튼을 선택합니다.

페이지에 다음 메시지가 표시됩니다.

Your Read Replica creation has been initiated.

64. 대상 리전의 교차 리전 읽기 전용 복제본을 검토하려면 같은 페이지에서 *here*로 레이블이 지정된 하이퍼링크를 선택합니다.

65. 검토하지 않으려면 Close 버튼을 선택합니다.

선택적 과제를 성공적으로 완료하고 Amazon RDS 데이터베이스의 교차 리전 복제본 생성을 시작했습니다.

DO NOT COPY
pink0569@naver.com

결론

이 실습에서는 다음을 수행하는 방법을 배웠습니다.

- Amazon RDS DB 인스턴스 생성
- Application Load Balancer 생성
- Application Load Balancer용 HTTP 리스너 생성
- 대상 그룹 생성
- 대상 그룹에 대상 등록
- 로드 밸런서 테스트 및 데이터베이스와 애플리케이션의 커넥션 테스트
- 콘솔을 통해 Amazon RDS DB 인스턴스 메타데이터 검토

① 이 실습에서는 프로토타입 웹 애플리케이션에 필요한 다양한 리소스를 Amazon VPC에 배포하는 방법을 배웠습니다. 하지만 이 실습에서 생성한 아키텍처는 탄력적이고 내구성과 가용성이 뛰어난 설계가 아니므로 AWS Cloud 모범 사례를 충족하지 못합니다. 아키텍처에서 단일 가용 영역에만 의존하기 때문에 단일 장애 지점이 있습니다. 중복성, 장애 조치, 고가용성을 위해 아키텍처를 구성하는 방법은 다음 실습에서 배웁니다.

실습 완료

▣ 축하합니다! 실습을 마치셨습니다.

실습 종료

다음 단계를 따라 콘솔을 닫고 실습을 종료한 후 실습 경험을 평가해 주십시오.

66. AWS Management Console로 돌아갑니다.
67. 탐색 모음에서 **awsstudent@<AccountNumber>** 를 선택한 다음 **Sign Out**을 선택합니다.
68. 실습 종료를 선택합니다.
69. OK를 선택합니다.
70. (선택 사항):
 - 해당하는 별 개수를 선택합니다.
 - 의견을 입력합니다.
 - **Submit**을 선택합니다.
 - 별 1개 = 매우 불만족
 - 별 2개 = 불만족
 - 별 3개 = 보통
 - 별 4개 = 만족
 - 별 5개 = 매우 만족

피드백을 제공하지 않으려면 그냥 창을 닫으면 됩니다.

AWS Training and Certification에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다.

피드백, 제안 사항 또는 수정 요청 사항을 제공하려면 AWS Training and Certification 문의 양식에 세부 정보를 입력해 주시기 바랍니다.



실습 4: Amazon VPC에서 고가용성 구성

© 2022 Amazon Web Services, Inc. 또는 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 임대 또는 판매는 금지됩니다. 모든 상표는 해당 소유자의 자산입니다.

참고: 개인 정보, 개인을 식별할 수 있는 정보 또는 기밀 정보는 실습 환경에 입력하지 마십시오. 입력한 정보가 다른 사용자에게 공개될 수도 있습니다.

수정 사항이나 피드백 또는 기타 질문이 있으십니까? AWS Training and Certification에서 문의해 주십시오.

실습 개요

AWS는 클라우드에서 안정적이고 내결함성이 있으며 가용성이 뛰어난 시스템을 구축하기 위한 서비스와 인프라를 제공합니다. 내결함성은 시스템을 구축하는데 사용되는 일부 구성 요소에 장애가 발생해도 계속 작동할 수 있는 시스템의 능력입니다. 고가용성은 시스템 장애를 예방하는 것이 아니라 장애에서 빠르게 복구하는 시스템의 능력입니다. AWS 솔루션 아키텍트는 가용성이 뛰어나고 필요할 경우 내결함성을 갖춘 시스템을 설계하고 이러한 설계의 이점과 비용을 이해해야 합니다. 이 실습에서는 두 가지 강력한 AWS 서비스인 Elastic Load Balancing과 Auto Scaling 그룹을 통합합니다. 애플리케이션 서버로 작동하는 EC2 인스턴스의 Auto Scaling 그룹을 생성한 다음 Auto Scaling 그룹 내 인스턴스 간에 부하가 균형있게 전달되도록 Application Load Balancer를 구성합니다. 계속해서 다중 AZ를 활성화하고, 읽기 전용 복제본을 생성하며, 읽기 전용 복제본을 승격하여 Amazon Relational Database Service(RDS) 작업을 수행합니다. 읽기 전용 복제본을 사용할 경우 프라이머리 데이터베이스에 쓰고 읽기 전용 복제본에서 읽습니다. 읽기 전용 복제본은 프라이머리 데이터베이스로 승격할 수 있으므로 고가용성과 재해 복구에 유용한 도구입니다.

다음 이미지는 최종 아키텍처를 보여 줍니다.



목표

이 실습을 마치면 다음을 수행할 수 있습니다.

- EC2 Auto Scaling 그룹을 생성하여 여러 가용 영역에 걸친 Application Load Balancer에 등록
- 가용성이 뛰어난 Amazon Aurora DB 클러스터 생성
- 고가용성을 갖추도록 Amazon Aurora 데이터베이스 클러스터 수정
- 중복 NAT 게이트웨이를 사용하여 고가용성을 갖추도록 Amazon VPC 구성 수정
- 장애를 시뮬레이션하여 애플리케이션과 데이터베이스의 고가용성 확인

수강 전 권장 사항

본 실습에는 다음이 필요합니다.

- Microsoft Windows, macOS 또는 Linux(Ubuntu, SuSE, Red Hat)가 실행되는 Wi-Fi 지원 노트북
- 인터넷 브라우저(예: Chrome, Firefox 또는 Microsoft Edge)
- 일반 텍스트 편집기

소요 시간

이 실습은 완료하는 데 약 **45분**이 소요됩니다.

실습 시작

1. 실습을 시작하려면 페이지 상단에서 실습 시작을 선택합니다.

그러면 실습 리소스를 프로비저닝하는 프로세스가 시작됩니다. 실습 리소스를 프로비저닝하는데 걸리는 예상 시간이 표시됩니다. 계속 진행하기 전에 사용할 리소스가 프로비저닝될 때까지 기다려야 합니다.

① 토큰을 입력하라는 메시지가 표시되면 여러분에게 배포된 토큰(또는 구매한 크레딧)을 사용하세요.

2. 실습을 열려면 콘솔 열기를 선택합니다.

새 웹 브라우저 탭에서 **AWS Management Console** 로그인 페이지가 열립니다.

3. **Sign in as IAM user** 페이지에서

- **IAM user name**에 awsstudent를 입력합니다.
- **Password**에 이 지침의 왼쪽에 나열된 **Password** 값을 복사하여 붙여넣습니다.
- **Sign in**을 선택합니다.

△ 별다른 지시가 없는 한 리전을 변경하지 마십시오.

일반적인 로그인 오류

오류: 우선 로그 아웃 필요

Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

You must first log out before logging into a different AWS account라는 메시지가 표시된다면 다음을 수행합니다.

- [click here](#)의 링크를 선택합니다.
- **Amazon Web Services Sign In** 웹 브라우저 탭을 닫고 초기 실습 페이지로 돌아갑니다.
- 콘솔 열기를 다시 선택합니다.

경우에 따라 일부 팝업 또는 스크립트 차단 웹 브라우저 확장 프로그램 때문에 **실습 시작** 버튼이 제대로 작동하지 않을 수 있습니다. 실습을 시작하는 데 문제가 있는 경우 다음을 수행합니다.

- 팝업 또는 스크립트 차단 프로그램의 허용 목록에 실습 도메인 이름을 추가하거나 차단 프로그램을 끕니다.
- 페이지를 새로 고친 후 다시 시도하십시오.

과제 1: 기존 실습 환경 검사

기존 환경의 구성을 검토합니다. AWS CloudFormation을 통해 다음 리소스가 프로비저닝되었습니다.

- Amazon Virtual Private Cloud(Amazon VPC)
- 2개의 가용 영역에 있는 퍼블릭 및 프라이빗 서브넷
- 퍼블릭 서브넷에 연결된 인터넷 게이트웨이(다이어그램에 표시되지 않음)
- 퍼블릭 서브넷 중 하나에 있는 NAT 게이트웨이
- 들어오는 애플리케이션 트래픽을 수신하고 전달하기 위해 2개의 퍼블릭 서브넷에 배포된 Application Load Balancer
- 간단한 재고 추적 애플리케이션을 실행하고 프라이빗 서브넷 중 하나에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스
- 재고 데이터를 저장하고 프라이빗 서브넷 중 하나에 있는 단일 DB 인스턴스가 포함된 Amazon Aurora DB 클러스터

다음 이미지는 최초 아키텍처를 보여 줍니다.



과제 1.1: 네트워크 인프라 검사

이 과제에서는 실습 환경의 네트워크 구성 세부 정보를 검토합니다.

- AWS 관리 콘솔의 AWS 검색 창을 사용해 VPC를 검색하고 검색 결과 리스트에서 선택합니다.

주의: 이 실습은 새로운 VPC 콘솔을 사용하도록 설계되었습니다. 화면 왼쪽 상단에서 **New VPC Experience**가 선택되었는지 확인합니다.

① **추가 정보:** 실습 환경에 Lab VPC가 생성되었으며, 이 실습에서 사용되는 모든 애플리케이션 리소스는 이 VPC 안에 있습니다.

5. 왼쪽 탐색 창에서 **Your VPCs**를 선택합니다.

기본 VPC와 함께 Lab VPC가 목록에 표시됩니다.

6. 왼쪽 탐색 창에서 **Subnets**를 선택합니다.

Lab VPC에 속한 서브넷이 목록에 표시됩니다. **Public Subnet 1**의 열에 나열된 세부 정보를 검토합니다.

- **VPC** 열에서 이 서브넷이 연결된 VPC를 파악할 수 있습니다. 이 서브넷은 **Lab VPC** 안에 존재합니다.
- **IPv4 CIDR** 열에서 **10.0.0.0/24** 값은 이 서브넷이 10.0.0.0과 10.0.0.255 사이의 IP 256개(이 중 5개는 예약되었으며 사용할 수 없습니다)를 포함한다는 뜻입니다.
- **Availability Zone** 열에서 이 서브넷이 있는 가용 영역을 파악할 수 있습니다. 이 서브넷은 가용 영역 'A'에 있습니다.

7. **Public Subnet 1**을 선택하면 페이지 하단에 추가 세부 정보가 표시됩니다.

팁: 구분선을 위아래로 끌어 하단 창을 확장할 수 있습니다. 또는 세 가지 사각형 아이콘  중 하나를 선택하여 미리 설정된 하단 창 크기를 선택할 수 있습니다.

8. 페이지 하단에서 **Route Table** 탭을 엽니다.

이 탭에는 이 서브넷의 라우팅에 대한 세부 정보가 표시됩니다.

- 첫 번째 항목은 VPC의 CIDR 범위(**10.0.0.0/20**) 내로 향하는 트래픽이 VPC(로컬) 내에서 라우팅되도록 지정합니다.
- 두 번째 항목은 인터넷(**0.0.0.0/0**)으로 향하는 트래픽이 인터넷 게이트웨이(**igw-xxxx**)로 라우팅되도록 지정합니다. 이 구성이 서브넷을 퍼블릭 서브넷으로 만듭니다.

9. **Network ACL** 탭을 엽니다.

이 탭에는 서브넷과 연결된 네트워크 액세스 제어 목록(ACL)이 표시됩니다. 규칙은 현재 **모든 트래픽이 서브넷 안팎으로 흐르도록 허용합니다.** 네트워크 ACL을 수정하거나 보안 그룹을 사용하여 트래픽을 추가로 제한할 수 있습니다.

10. 왼쪽 탐색 창에서 **Internet Gateways**를 선택합니다.

Lab IG라는 인터넷 게이트웨이가 이미 Lab VPC에 연결되어 있습니다.

11. 왼쪽 탐색 창에서 **Security Groups**를 선택합니다.

12. **Inventory-ALB** 보안 그룹을 선택합니다.

이것은 Application Load Balancer에 들어오는 트래픽을 제어하는 데 사용되는 보안 그룹입니다.

13. 페이지 하단에서 **Inbound rules** 탭을 엽니다.

이 보안 그룹은 어디서나*(0.0.0.0/0)* 오는 인바운드 웹 트래픽을 허용합니다.

14. **Outbound rules** 탭을 엽니다.

기본적으로 보안 그룹은 모든 아웃바운드 트래픽을 허용합니다. 하지만 필요에 따라 이러한 규칙을 수정할 수 있습니다.

15. **Inventory-App** 보안 그룹을 선택합니다. 이 보안 그룹만 선택해야 합니다.

이것은 AppServer Amazon EC2 인스턴스로 들어오는 트래픽을 제어하는 데 사용되는 보안 그룹입니다.

16. 페이지 하단에서 **Inbound rules** 탭을 엽니다.

이 보안 그룹은 Application Load Balancer 보안 그룹*(Inventory-ALB)*에서 오는 인바운드 웹 트래픽(포트 80)만을 허용합니다.

17. **Outbound rules** 탭을 엽니다.

기본적으로 보안 그룹은 모든 아웃바운드 트래픽을 허용합니다. Application Load Balancer 보안 그룹의 아웃바운드 규칙과 마찬가지로 필요에 따라 이러한 규칙을 수정할 수 있습니다.

18. **Inventory-DB** 보안 그룹을 선택합니다. 이 보안 그룹만 선택해야 합니다.

이것은 데이터베이스로 들어오는 트래픽을 제어하는 데 사용되는 보안 그룹입니다.

19. 페이지 하단에서 **Inbound rules** 탭을 엽니다.

이 보안 그룹은 애플리케이션 서버 보안 그룹*(Inventory-App)*에서 오는 인바운드 MySQL/Aurora 트래픽(포트 3306)만 허용합니다.

20. **Outbound rules** 탭을 엽니다.

기본적으로 보안 그룹은 모든 아웃바운드 트래픽을 허용합니다. 이전 보안 그룹의 아웃바운드 규칙과 마찬가지로 필요에 따라 이러한 규칙을 수정할 수 있습니다.

과제 1.2: Amazon EC2 인스턴스 검사

Amazon EC2 인스턴스가 제공되어 있습니다. 이 인스턴스는 데이터베이스에서 재고를 추적하는 간단한 PHP 애플리케이션을 실행합니다. 이 과제에서는 인스턴스 세부 정보를 검사합니다.

21. AWS 관리 콘솔의 AWS 검색 창에서 EC2를 검색하고 검색 결과 리스트에서 선택합니다.

22. 왼쪽 탐색 창에서 **Instances**를 선택합니다.

23. 이름이 **AppServer**인 인스턴스 옆의 확인란 을 선택하면 페이지 하단에 추가 세부 정보가 표시됩니다.

24. 인스턴스 세부 정보를 검토한 후 Actions 메뉴 아래에서 **Instance settings**를 선택한 다음 **Edit user data**를 선택합니다.
25. **Edit user data** 페이지에서 **Current user data** 텍스트 필드의 전체 내용을 클립보드에 복사합니다.
참고 - 복사 아이콘(*Current user data* 옆)을 선택하면 사용자 데이터를 클립보드로 쉽게 복사할 수 있습니다.
26. 방금 복사한 사용자 데이터를 텍스트 편집기에 붙여넣습니다. 이후 과제에서 사용하게 됩니다.

과제 1.3: 로드 밸런서 구성 검사

Application Load Balancer와 대상 그룹이 제공되어 있습니다. 이 과제에서는 그 구성을 검토합니다.

27. AWS 관리의 AWS 검색 창에서 EC2를 검색하고 검색 결과 리스트에서 선택합니다.
28. 왼쪽 탐색 창에서 **Target Groups**를 선택합니다.
29. 이름이 **Inventory-App**인 대상 그룹 옆의 확인란 을 선택하면 페이지 하단에 추가 세부 정보가 표시됩니다.
30. 아래쪽 창에서 **Targets** 탭을 엽니다.

Application Load Balancer는 들어오는 요청을 목록의 모든 대상에 전달합니다. 앞서 검사한 AppServer EC2 인스턴스는 이미 대상으로 등록되어 있습니다.

31. 왼쪽 탐색 창에서 **Load Balancers**를 선택합니다.
32. 이름이 **Inventory-LB**인 로드 밸런서를 선택하면 페이지 하단에 추가 세부 정보가 표시됩니다.

과제 1.4: 웹 브라우저에서 간단한 PHP 재고 애플리케이션 열기

재고 애플리케이션이 올바르게 작동 중인지 확인하려면 재고 애플리케이션 설정 페이지의 URL을 검색해야 합니다.

33. 이 실습 지침 왼쪽의 **InventoryAppSettingsPageURL**을 클립보드에 복사합니다.

참고 URL은 <http://Inventory-LB-xxxx.elb.amazonaws.com/settings.php>와 비슷합니다.

34. 새 웹 브라우저 탭을 열어 이전 단계에서 복사한 URL을 붙여넣고 Enter 키를 누릅니다.

재고 애플리케이션 설정 페이지가 표시됩니다. 데이터베이스 엔드포인트, 데이터베이스 이름, 로그인 세부 정보는 이미 Amazon Aurora 데이터베이스의 값으로 미리 채워져 있습니다.

35. 재고 앱 설정 페이지의 모든 설정은 기본 구성 그대로 두십시오.

36. Save 버튼을 선택합니다.

설정을 저장하면 재고 애플리케이션이 기본 페이지로 리디렉션되고, 다양한 항목의 재고가 표시됩니다. 재고에 항목을 추가하거나 기존 재고 항목의 세부 정보를 수정해도 됩니다. 이 애플리케이션과 상호 작용할 때 로드 밸런서는 로드 밸런서의 대상 그룹에서 이전에 본 AppServer로 요청을 전달합니다. AppServer는 Amazon Aurora 데이터베이스의 재고 변경 사항을 기록합니다. 웹 페이지 하단에는 인스턴스 ID와 인스턴스가 있는 가용 영역이 표시됩니다.

주의: 나머지 실습 과제를 수행하는 동안 이 재고 애플리케이션 웹 브라우저 탭을 열어 두십시오. 이후 과제에서 이 탭으로 다시 돌아옵니다.

이제 실습 환경에서 생성된 모든 리소스 검사를 완료하고 제공된 재고 애플리케이션에 성공적으로 액세스했습니다. 다음으로 재고 애플리케이션을 고가용성으로 만들기 위해 EC2 Auto Scaling과 함께 사용할 시작 템플릿을 생성합니다.

과제 2: 시작 템플릿 생성

Amazon EC2 Auto Scaling 그룹을 생성하려면 먼저 Amazon Machine Image(AMI)의 ID 및 인스턴스 유형 등 Amazon EC2 인스턴스를 시작하는 데 필요한 파라미터가 포함된 시작 템플릿을 생성해야 합니다.

이 과제에서는 시작 템플릿을 생성합니다.

37. AWS 관리 콘솔의 AWS 검색 창에서 EC2를 검색하고 검색결과 리스트에서 선택합니다.

38. 왼쪽 탐색 창의 Instances 아래에서 Launch Templates를 선택합니다.

39. Create launch template을 선택합니다.

40. Launch template name and description 섹션에서 다음을 구성합니다.

- Launch template name: Lab-template-NUMBER

참고 NUMBER를 아래 나온 무작위 번호로 바꿉니다.

예: Lab-template-98469549

참고 템플릿 이름이 이미 존재하는 경우 다른 번호로 다시 시도하십시오.

- Template version description: version 1

Amazon Machine Image(AMI)를 선택해야 합니다. AMI는 인스턴스를 시작하는 데 필요한 정보를 제공합니다. 인스턴스를 시작할 때 AMI를 지정해야 합니다. AMI에는 인스턴스 루트 볼륨의 템플릿(예: 운영 체제, 애플리케이션 서버, 애플리케이션)이 포함되어 있습니다.

다양한 운영 체제의 AMI를 사용할 수 있습니다. 이 실습에서는 Amazon Linux 2 OS를 실행하는 인스턴스를 시작합니다.

41. Application and OS Images (Amazon machine Image) Info에서 Quick Start 탭을 선택합니다.

42. Amazon Linux를 선택합니다.

43. AMI에서 _Amazon Linux 2 AMI_를 선택합니다.

!참고 Amazon Linux 2 AMI의 Arm 버전이 아니라 x86 버전을 선택해야 합니다.

44. Instance type의 경우 드롭다운 메뉴에서 t3.micro를 선택합니다.

인스턴스를 시작할 때 인스턴스 유형에 따라 인스턴스에 할당된 하드웨어가 결정됩니다. 각 인스턴스 유형은 서로 다른 컴퓨팅, 메모리, 스토리지 용량을 제공하며, 이 용량에 따라 서로 다른 인스턴스 패밀리로 분류됩니다.

45. Security groups에서 Inventory-App을 선택합니다.

46. Advanced Details 섹션까지 아래로 스크롤합니다.

47. ▶ **Advanced details**를 확장합니다.
48. **IAM instance profile**: *Inventory-App-Role*을 선택합니다.
49. **User data** 섹션에서 과제 1.2 도중 텍스트 편집기에 저장한 사용자 데이터를 붙여넣습니다.
50. Create launch template 버튼을 선택합니다.
51. View launch templates 버튼을 선택합니다.

DO NOT COPY
pink0569@naver.com

과제 3: Auto Scaling 그룹 생성

이 과제에서는 프라이빗 서브넷에 Amazon EC2 인스턴스를 배포하는 Auto Scaling 그룹을 생성합니다. 프라이빗 서브넷의 인스턴스는 인터넷에서 액세스할 수 없기 때문에 애플리케이션을 배포할 때는 이것이 보안 모범 사례입니다. 대신 사용자가 Application Load Balancer에 요청을 보내면 다음 디어그램과 같이 해당 요청이 프라이빗 서브넷에 있는 Amazon EC2 인스턴스에 전달됩니다.



① **추가 정보:** Amazon EC2 Auto Scaling은 사용자가 정의한 정책, 일정, 상태 확인에 따라 자동으로 Amazon EC2 인스턴스를 ①시작 또는 종료하도록 설계된 서비스입니다. 또한 여러 가용 영역에 인스턴스를 자동으로 분산하여 고가용성 애플리케이션을 생성할 수 있습니다.

52. 왼쪽 탐색 창의 **Auto Scaling** 아래에서 **Auto Scaling Groups**를 클릭합니다.

53. Create Auto Scaling group 버튼을 선택합니다.

54. 다음을 구성합니다.

- **Name:** Inventory-ASG
- **Launch template:** 생성한 시작 템플릿을 선택합니다.
- **Next** 버튼을 선택합니다.

55. **Network** 섹션에서 다음을 구성합니다.

- **VPC:** Lab VPC
- **Subnets:** Private Subnet 1과 Private Subnet 2를 선택합니다.

56. **Next** 버튼을 선택합니다.

57. **Configure advanced options** 페이지에서 다음을 구성합니다.

- **Attach to an existing load balancer**를 선택합니다.
- **Select Choose from your load balancer target groups**를 선택합니다.
- **Existing load balancer target groups**의 드롭다운 메뉴에서 **Inventory-App|HTTP**를 선택합니다.

그러면 앞서 검사한 *Inventory-App* 대상 그룹의 일부로 새 EC2 인스턴스를 등록하라고 Auto Scaling 그룹에 지시합니다. 로드 밸런서가 이 대상 그룹에 있는 인스턴스로 트래픽을 전송합니다.

- **Health check grace period:** 300
- **Monitoring:** **Enable group metrics collection within CloudWatch**

기본적으로 상태 확인 유예 기간은 300으로 설정됩니다.

58. Next 버튼을 선택합니다.

59. **Configure group size and scaling policies** 페이지에서 다음을 구성합니다.

- **Desired capacity:** 2
- **Minimum capacity:** 2
- **Maximum capacity:** 2

60. Next 버튼을 선택합니다.

이 실습에서는 고가용성을 위해 항상 2개의 인스턴스를 유지합니다. 애플리케이션이 다양한 트래픽 부하를 수신할 것으로 예상되는 경우 인스턴스를 시작/종료할 시기를 정의하는 크기 조정 정책을 생성할 수도 있습니다. 하지만 이 실습의 재고 애플리케이션에는 필요하지 않습니다.

61. **Tags** 페이지가 표시될 때까지 Next 버튼을 선택합니다.

62. Add tag를 선택하고 다음을 구성합니다.

- **Key:** Name
- **Value:** Inventory-App

그러면 Auto Scaling 그룹에 이름으로 태그가 지정되고 Auto Scaling 그룹에서 시작한 Amazon EC2 인스턴스에도 적용됩니다. 그러면 어떤 애플리케이션 또는 비용 센터 같은 비즈니스 개념에 어떤 EC2 인스턴스가 연결되어 있는지 더 쉽게 식별할 수 있습니다.

63. Next 버튼을 선택합니다.

64. Auto Scaling 그룹 구성이 정확한지 검토한 다음 Create Auto Scaling group 버튼을 선택합니다.

곧 2개의 가용 영역에서 애플리케이션이 실행됩니다. 인스턴스 또는 가용 영역 하나에 장애가 발생하더라도 Auto Scaling은 구성을 유지합니다.

Auto Scaling 그룹을 생성했으므로 이 그룹에서 EC2 인스턴스를 시작했는지 확인할 수 있습니다.

65. 생성한 Auto Scaling Group을 선택합니다.

Group Details 섹션을 검사하여 Auto Scaling 그룹에 대한 정보를 검토합니다.

66. **Activity** 탭을 엽니다.

Activity History 섹션은 Auto Scaling 그룹에서 발생한 이벤트의 레코드를 유지합니다. Status 열에는 인스턴스의 현재 상태가 포함됩니다. 인스턴스를 시작하면 상태 열에 *PreInService* 가 표시됩니다. 인스턴스가 시작된 후 상태가 *Successful*로 변경됩니다.

67. **Instance management** 탭을 클릭합니다.

Auto Scaling 그룹이 2개의 Amazon EC2 인스턴스를 시작했고, 이 인스턴스는 *InService* 수명 주기 상태에 있습니다. Health Status 열에 인스턴스에 대한 Amazon EC2 인스턴스 상태 확인 결과가 표시됩니다.

① **추가 정보:** 인스턴스가 아직 *InService* 상태에 도달하지 못한 경우 몇 분 정도 기다려야 합니다. 새로 고침 버튼을 선택하여 인스턴스의 현재 수명 주기 상태를 검색할 수 있습니다.

68. **Monitoring** 탭을 엽니다. 여기에서 Autoscaling 그룹의 모니터링 관련 정보를 검토할 수 있습니다.

① **추가 정보:** 이 페이지는 Auto Scaling 그룹에서의 활동뿐 아니라 인스턴스의 사용률과 상태에 관한 정보를 제공합니다. **Auto Scaling** 탭에는 Auto Scaling 그룹에 대한 Amazon CloudWatch 지표가 표시되고, **EC2** 탭에는 Auto Scaling 그룹이 관리하는 Amazon EC2 인스턴스의 지표가 표시됩니다.

이제 애플리케이션의 가용성을 유지하고 인스턴스 또는 가용 영역 장애에 대해 복원력을 갖도록 하는 Auto Scaling 그룹을 성공적으로 생성했습니다. 다음으로 애플리케이션이 고가용성을 테스트합니다.

과제 4: 애플리케이션 테스트

이 과제에서는 웹 애플리케이션이 실행 중이고 가용성이 높은지 확인합니다.

69. 왼쪽 탐색 창에서 **Target Groups**를 클릭합니다.

70. **Name** 아래에서 **Inventory-App**을 선택합니다.

71. 페이지 하단에서 **Targets** 탭을 엽니다.

Registered targets 섹션에 3개의 인스턴스가 있습니다. 여기에는 이름이 Inventory-App인 Auto Scaling 인스턴스 2개와 과제 1에서 검사한 AppServer라는 원래 인스턴스가 포함됩니다. **Health Status** 열에는 인스턴스에 대해 수행한 로드 밸런서 상태 확인 결과가 표시됩니다. 이 과제에서는 대상 그룹에서 원래 AppServer 인스턴스를 제거하고 EC2 Auto Scaling이 관리하는 2개의 인스턴스만 남깁니다.

72. 이름이 AppServer인 인스턴스 옆의 확인란 을 선택합니다.

73. **Deregister**를 선택하여 로드 밸런서의 대상 그룹에서 인스턴스를 제거합니다.

인스턴스가 등록 취소되는 즉시 로드 밸런서는 대상으로의 요청 라우팅을 중지합니다.

AppServer 인스턴스의 **Health status** 열에는 *Draining* 상태가 표시되고, **Health Status Details** 열에는 진행 중인 요청이 완료될 때까지 *Target deregistration is in progress* 가 표시됩니다. 몇 분이 지나면 AppServer 인스턴스 등록 취소가 완료되고, 등록된 대상 목록에는 2개의 Auto Scaling 인스턴스만 남습니다.

참고 인스턴스 등록을 취소하면 로드 밸런서에서 인스턴스가 분리될 뿐입니다. AppServer 인스턴스는 사용자가 종료할 때까지 계속 무기한 실행됩니다.

74. Inventory-App 인스턴스의 **Health status** 열에 아직 **healthy**가 표시되지 않으면 2개의 Inventory-App 인스턴스 모두 **Health status** 열에 **healthy**가 표시될 때까지 페이지 오른쪽 상단의 새로 고침 버튼을 사용하여 30초마다 인스턴스 목록을 업데이트하십시오. 인스턴스 초기화가 완료되려면 몇 분 정도 걸릴 수 있습니다.

상태가 끝까지 **healthy**로 변경되지 않을 경우 강사에게 구성 진단 지원을 요청하십시오. **Health Status** 열의 정보 ○ 아이콘 위에 마우스 커서를 놓으면 상태에 대한 세부 정보가 표시됩니다.

애플리케이션을 테스트할 준비가 되었습니다. Application Load Balancer에 연결하여 애플리케이션을 테스트합니다. Application Load Balancer는 EC2 Auto Scaling이 관리하는 Amazon EC2 인스턴스 중 하나로 사용자의 요청을 전송합니다.

75. 웹 브라우저의 재고 애플리케이션 탭으로 돌아갑니다.

참고 브라우저 탭을 닫은 경우 다음 방법으로 재고 애플리케이션을 다시 열 수 있습니다.

- 왼쪽 탐색 창에서 **Load Balancers**를 선택합니다.
- 창 하단에 있는 **Description** 탭에서 **DNS 이름**을 클립보드로 복사합니다.

*Inventory-LB-xxxx.elb.amazonaws.com*과 유사할 것입니다.

- 새 웹 브라우저 탭을 열고 클립보드에서 DNS 이름을 붙여넣고 Enter 키를 누릅니다.

로드 밸런서가 Amazon EC2 인스턴스 중 하나로 사용자의 요청을 전달합니다. 인스턴스 ID와 가용 영역은 웹 페이지 하단에 표시됩니다.

76. 웹 브라우저에서 페이지를 몇 번 새로 고칩니다. 인스턴스 ID와 가용 영역은 두 인스턴스 사이에서 변경되는 경우가 있다는 점에 유의해야 합니다.

다음 이미지는 이 웹 애플리케이션의 정보 흐름을 표시합니다.



정보의 흐름은 다음과 같습니다.

- 퍼블릭 서브넷에 있는 Application Load Balancer로 요청을 전송합니다. 퍼블릭 서브넷은 인터넷에 연결되어 있습니다.
- Application Load Balancer가 프라이빗 서브넷에 있는 Amazon EC2 인스턴스 중 하나를 선택해 요청을 전달합니다.
- 그런 다음 Amazon EC2 인스턴스가 Application Load Balancer에 웹 페이지를 반환하고, Application Load Balancer가 웹 페이지를 웹 브라우저에 반환합니다.

이제 EC2 Auto Scaling이 2개의 새 Inventory-App 인스턴스를 2개의 가용 영역에서 성공적으로 시작한 것을 확인하고, 원래 AppServer 인스턴스를 로드 밸런서에서 등록 취소했습니다. Auto Scaling 그룹은 장애가 발생하는 경우 애플리케이션의 고가용성을 유지합니다. 다음으로 EC2 Auto Scaling이 관리하는 Inventory-App 인스턴스 중 하나를 종료하여 장애를 시뮬레이션합니다.

과제 5: 애플리케이션 티어의 고가용성 테스트

이 과제에서는 Amazon EC2 인스턴스 중 하나를 종료하여 애플리케이션의 고가용성 구성을 테스트합니다.

77. **EC2 관리 콘솔**로 돌아갑니다. 단, 애플리케이션 탭은 닫지 마십시오. 이후 과제에서 이 탭으로 다시 돌아옵니다.

78. 왼쪽 탐색 창에서 **Instances**를 선택합니다.

이제 웹 애플리케이션 인스턴스 중 하나를 종료하여 장애를 시뮬레이션합니다.

79. **Inventory-App** 인스턴스 중 하나를 선택합니다. (어느 것을 선택해도 상관이 없습니다.)

80. Instance State 를 선택한 다음 **Terminate instance**를 선택합니다.

81. Terminate 버튼을 선택합니다.

잠시 후 로드 밸런서 상태 확인이 인스턴스가 응답하지 않는 것을 감지하고 들어오는 모든 요청을 나머지 인스턴스로 자동으로 라우팅합니다.

82. AWS 관리 콘솔을 열어 두고 웹 브라우저의 Inventory 애플리케이션 탭으로 전환해 페이지를 여러 번 새로 고칩니다.

페이지 하단에 표시된 가용 영역은 동일하게 유지됩니다. 인스턴스에 장애가 발생한 경우에도 애플리케이션은 계속 사용할 수 있습니다.

몇 분 후 Auto Scaling도 인스턴스 장애를 확인합니다. 2개의 인스턴스가 계속 실행되도록 Auto Scaling을 구성했기 때문에 Auto Scaling이 자동으로 대체 인스턴스를 시작합니다.

83. **EC2 관리 콘솔**로 돌아갑니다. 이름이 *Inventory-App*인 새 Amazon EC2 인스턴스가 표시될 때까지 30초마다 새로 고침 버튼을 사용하여 인스턴스 목록을 다시 로드합니다.

새로 시작된 인스턴스는 Health check 열 아래에 *_Initializing_*을 표시합니다. 몇 분 후 새 인스턴스의 상태 점검이 *_healthy_*이 되고 로드 밸런서가 두 가용성 영역 간에 트래픽을 다시 분배합니다.

84. Inventory 애플리케이션 탭으로 돌아가 페이지를 여러 번 새로 고칩니다. 페이지를 새로 고치면 인스턴스 ID와 가용 영역이 바뀝니다.

이를 통해 애플리케이션이 현재 고가용성 상태임을 알 수 있습니다.

과제 6: 데이터베이스 티어의 고가용성 구성

이전 과제에서 애플리케이션 티어의 고가용성을 확인했습니다. 하지만 Amazon Aurora 데이터베이스는 여전히 하나의 데이터베이스 인스턴스에서만 작동하고 있습니다.

과제 6.1: 여러 가용 영역에서 실행되도록 데이터베이스 구성

이 과제에서는 Amazon Aurora 데이터베이스를 여러 가용 영역에서 실행되도록 구성하여 고가용성으로 만듭니다.

85. AWS 관리 콘솔의 AWS 검색 창에서 RDS를 검색하고 검색 결과 리스트에서 선택합니다.

86. 왼쪽 탐색 창에서 **Databases**를 선택합니다.

87. Amazon Aurora 데이터베이스 클러스터 중 식별자 *inventory-primary*인 행을 확인합니다.

88. 세번째 컬럼의 **Region & AZ** 항목의 값을 기록해둡니다. 이 값은 기본 인스턴스가 배포된 가용 영역입니다.

주의: 다음 단계에서 데이터베이스 클러스터를 위한 추가 인스턴스를 생성합니다. 추가로 생성하는 인스턴스는 고가용성 구조를 위해 기본 인스턴스와 다른 가용 영역에 위치해야 합니다.

89. Amazon Aurora 데이터베이스 클러스터 중 식별자 **inventory-cluster**를 선택합니다.

90. Actions 버튼을 선택한 다음 **Add reader**를 선택합니다.

91. **Settings** 섹션에서 다음을 구성합니다.

- **DB instance identifier:** *inventory-replica*

92. **Connectivity** 섹션의 Availability Zone에 아래에 있는 드롭다운 목록에서 두 번째 옵션을 선택합니다.

93. **inventory-primary**의 Region & AZ 값인 배포된 가용 영역과 다른 가용 영역을 선택합니다.

94. 페이지 하단에서 Add reader를 선택합니다.

목록에 이름이 *inventory-replica*인 새 DB 식별자가 표시되며, 상태는 *Creating*입니다. 이것이 사용할 Aurora 복제본 인스턴스입니다. 기다리지 않고 다음 작업을 계속할 수 있습니다.

① Aurora 복제본 시작이 완료되면 데이터베이스가 고가용성 구성으로 여러 가용 영역에 배포됩니다. 이는 데이터베이스가 여러 인스턴스에 분산된다는 뜻은 아닙니다. 프라이머리 DB 인스턴스와 Aurora 복제본 모두 동일한 공유 스토리지에 액세스하지만 프라이머리 DB 인스턴스만 쓰기에 사용할 수 있습니다. Aurora 복제본의 주 용도는 두 가지입니다. Aurora 복제본에 쿼리를 실행하여 애플리케이션에 대한 읽기 작업 크기를 조정할 수 있습니다. 그러려면 일반적으로 클러스터의 Reader 엔드포인트에 연결합니다. 이렇게 하면 Aurora가 읽기 전용 연결의 부하를 클러스터에 있는 여러 Aurora 복제본에 분산시킬 수 있습니다. Aurora 복제본은 가용성을 높이는 데도 도움이 됩니다. 클러스터의 Writer 인스턴스를 사용할 수 없게 되면 Aurora는 자동으로 Reader 인스턴스 중 하나를 새 Writer 인스턴스로 승격합니다.

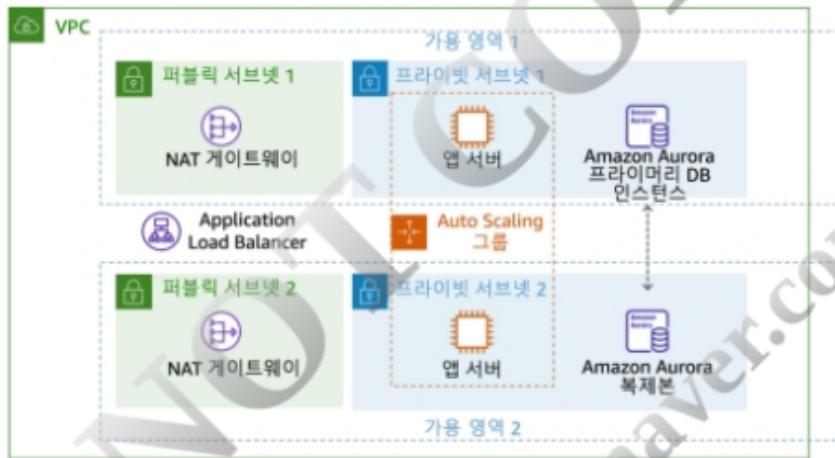
Aurora 복제본이 시작되는 동안 다음 과제를 계속하여 NAT 게이트웨이의 고가용성을 구성하고, Amazon RDS 콘솔로 돌아와 복제본 생성 완료 후 최종 과제에서 데이터베이스의 고가용성을 확인합니다.

과제 7: NAT 게이트웨이를 고가용성으로 만들기

이 과제에서는 두 번째 가용 영역에 있는 또 다른 NAT 게이트웨이를 시작하여 NAT 게이트웨이를 고가용성으로 만듭니다.

2개의 가용 영역에 걸친 프라이빗 서브넷에 Inventory-App 서버가 배포되어 있습니다. 인터넷에 액세스해야 하는 경우(예: 데이터 다운로드) 요청은 퍼블릭 서브넷에 있는 NAT 게이트웨이를 통해 리디렉션되어야 합니다. 현재 아키텍처에는 Public Subnet 1에 NAT 게이트웨이 하나만 있고, 모든 Inventory-App 서버가 이 NAT 게이트웨이를 사용하여 인터넷에 연결합니다. 즉, 가용 영역 1에서 장애가 발생하면 어느 애플리케이션 서버도 인터넷과 통신할 수 없습니다. 가용 영역 2에 두 번째 NAT 게이트웨이를 추가하면 가용 영역 1에 장애가 발생하더라도 프라이빗 서브넷의 리소스가 인터넷에 연결할 수 있습니다.

다음 다이어그램에 표시된 그 결과 아키텍처는 가용성이 높습니다.



과제 7.1: 두 번째 NAT 게이트웨이 생성

95. AWS 관리 콘솔의 AWS 검색 창에서 VPC를 검색하고 검색 결과 리스트에서 선택합니다.

96. 왼쪽 탐색 창에서 **NAT Gateways**를 선택합니다.

기존 NAT 게이트웨이가 표시됩니다. 이제 다른 가용 영역에 NAT 게이트웨이를 생성합니다.

97. Create NAT Gateway를 선택하고 다음을 구성합니다.

- **Name:** my-nat-gateway
- **Subnet:** Public Subnet 2 를 선택합니다.

98. Allocate Elastic IP address 버튼을 선택합니다.

99. Create NAT gateway 버튼을 선택합니다.

과제 7.2: 새 라우팅 테이블 생성 및 구성

이제 트래픽을 새 NAT 게이트웨이로 리디렉션하는 새 라우팅 테이블을 Private Subnet 2에 생성합니다.

100. 왼쪽 탐색 창에서 **Route Tables**를 선택합니다.
101. **Create route table**을 선택하고 다음을 구성합니다.

- **Name tag:** Private Route Table 2
- **VPC:** Lab VPC

102. **Create route table** 버튼을 선택합니다.

새로 생성된 라우팅 테이블의 세부 정보가 표시됩니다. 현재는 모든 트래픽을 **로컬로 보내는 경로** 하나가 있습니다. 이제 새 NAT 게이트웨이를 통해 인터넷 바운드 트래픽을 보내는 경로를 추가합니다.

103. **Edit routes** 버튼을 선택합니다.
104. **Add route**를 선택하고 다음을 구성합니다.
 - **Destination:** 0.0.0.0/0
 - **Target:** **NAT Gateway** 선택 후 *my-nat-gateway*를 선택합니다.
105. **Save changes** 버튼을 선택합니다.

라우팅 테이블을 생성하고 새 NAT 게이트웨이를 통해 인터넷 바운드 트래픽을 라우팅하도록 구성했습니다. 다음으로 라우팅 테이블을 Private Subnet 2에 연결합니다.

과제 7.3: Private Subnet 2 라우팅 구성

106. **Subnet Associations** 탭을 엽니다.
107. **Edit subnet associations** 버튼을 선택합니다.
108. **Private Subnet 2**를 선택합니다.
109. **Save associations** 버튼을 선택합니다.

그러면 이제 Private Subnet 2의 인터넷 바운드 트래픽을 동일한 가용 영역에 있는 NAT 게이트웨이로 보냅니다.

사용자의 NAT 게이트웨이는 이제 고가용성입니다. 한 가용 영역의 장애는 다른 가용 영역의 트래픽에 영향을 미치지 않습니다.

과제 8: Amazon Aurora 데이터베이스의 고가용성 테스트

이 과제에서는 프라이머리 DB 인스턴스의 장애를 시뮬레이션하여 데이터베이스의 고가용성을 확인합니다. 이 시뮬레이션은 이전 과제에서 생성한 Aurora 복제본에 강제로 장애 조치를 적용합니다.

110. AWS 관리 콘솔의 AWS 검색 창에서 RDS를 검색하고 검색 결과 리스트에서 선택합니다.
111. 왼쪽 탐색 창에서 **Databases**를 선택합니다.
112. Amazon Aurora 프라이머리 DB 인스턴스에 연결된 **inventory-primary** DB 식별자를 선택합니다.

참고 DB 식별자가 **inventory-primary**인 프라이머리 DB 인스턴스는 현재 Role 열 아래에 **Writer**가 표시됩니다. 이것은 현재 쓰기에 사용할 수 있는 클러스터의 유일한 데이터베이스 노드입니다.

113. Actions 를 선택한 다음 **Delete**를 선택합니다.
114. 다음 페이지에서 메시지가 표시되면 상자에 `delete me`를 입력하고 **Delete**를 선택하여 확인합니다.

inventory-primary DB 인스턴스가 *Deleting* 상태가 됩니다. 몇 분 후 Amazon RDS는 프라이머리 인스턴스가 더 이상 응답하지 않음을 감지하고 자동으로 프라이머리 인스턴스에서 Aurora 복제본으로 전환합니다.

115. **inventory-replica** DB 인스턴스의 Role 열 아래에 **Writer**가 표시될 때까지 30초마다 새로 고침 버튼을 사용하여 DB 인스턴스 목록을 다시 로드하십시오.

이 상태 변화는 Aurora 복제본으로의 장애 조치가 완료되었음을 나타냅니다. 다음으로 장애 조치 후에 재고 애플리케이션이 여전히 작동하는지 확인합니다.

116. 웹 브라우저의 재고 애플리케이션 탭으로 돌아가 페이지를 새로 고칩니다.

주의: 이전에 이 탭을 닫은 경우 이 실습 지침 왼쪽에 있는 **InventoryAppURL**을 방문하면 재고 애플리케이션에 액세스할 수 있습니다.

장애 조치 후에도 애플리케이션이 계속 올바로 작동하는 것을 관찰합니다. 이로써 데이터베이스의 고가용성이 확인됩니다.

결론

축하합니다! 다음 작업을 성공적으로 완료했습니다.

- Amazon EC2 Auto Scaling 그룹을 생성하여 여러 가용 영역에 걸친 Application Load Balancer에 등록
- 가용성이 뛰어난 Amazon Aurora DB 클러스터 생성
- 고가용성을 갖추도록 Amazon Aurora 데이터베이스 클러스터 수정

- 중복 NAT 게이트웨이를 사용하여 고가용성을 갖추도록 Amazon VPC 구성 수정
- 장애를 시뮬레이션하여 애플리케이션과 데이터베이스의 고가용성 확인

실습 종료

다음 단계를 따라 콘솔을 닫고 실습을 종료한 후 실습 경험을 평가해 주십시오.

117. AWS Management Console로 돌아갑니다.
118. 탐색 모음에서 **awsstudent@<AccountNumber>** 를 선택한 다음 **Sign Out**을 선택합니다.
119. 실습 종료를 선택합니다.
120. OK를 선택합니다.
121. (선택 사항):
 - 해당하는 별 개수를 선택합니다.
 - 의견을 입력합니다.
 - Submit**을 선택합니다.
 - 별 1개 = 매우 불만족
 - 별 2개 = 불만족
 - 별 3개 = 보통
 - 별 4개 = 만족
 - 별 5개 = 매우 만족

피드백을 제공하지 않으려면 그냥 창을 닫으면 됩니다.

AWS Training and Certification에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다.

피드백, 제안 사항 또는 수정 요청 사항을 제공하려면 AWS Training and Certification 문의 양식에 세부 정보를 입력해 주시기 바랍니다.



실습 5: 서비스 아키텍처 구축

© 2022 Amazon Web Services, Inc. 또는 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 임대 또는 판매는 금지됩니다. 모든 상표는 해당 소유자의 자산입니다.

참고: 개인 정보, 개인을 식별할 수 있는 정보 또는 기밀 정보는 실습 환경에 입력하지 마십시오. 입력한 정보가 다른 사용자에게 공개될 수도 있습니다.

수정 사항이나 피드백 또는 기타 질문이 있으십니까? AWS Training and Certification에서 문의해 주십시오.

실습 개요

분산 애플리케이션 분리를 위해 이벤트 중심 아키텍처를 채택하는 AWS 솔루션 아키텍트가 늘고 있습니다. 이러한 이벤트는 엄격한 순서에 따라 모든 구독 애플리케이션에 전파되어야 하는 경우가 많습니다. Amazon SNS 주제와 Amazon SQS 대기열을 사용하면 엔드-투-엔드 메시지 정렬, 중복 제거, 필터링, 암호화가 필요한 사용 사례를 해결할 수 있습니다. 이 실습에서는 객체가 추가될 때마다 Amazon SNS 알림을 호출하도록 S3 버킷을 구성합니다. SQS 대기열을 생성하고 상호 작용하는 방법과 SQS를 사용하여 Lambda 함수를 호출하는 방법을 배웁니다. 이 시나리오는 SNS, AWS Lambda, SQS 같은 서비스를 사용하여 S3 버킷 이벤트에 응답하도록 애플리케이션을 아키텍팅하는 방법을 이해하는데 도움이 됩니다.

목표

이 실습을 마치면 다음을 수행할 수 있습니다.

- 리소스 분리의 가치 이해
- EC2 인스턴스를 Lambda 함수로 교체하는 경우의 잠재적 가치 이해
- Amazon SNS 주제 생성
- Amazon SQS 대기열 생성
- Amazon S3에서 이벤트 알림 생성
- 기존 코드를 사용하여 AWS Lambda 함수 생성
- SQS 대기열에서 AWS Lambda 함수 트리거
- Amazon CloudWatch 로그를 통해 AWS Lambda S3 함수 모니터링

수강 전 권장 사항

본 실습에는 다음이 필요합니다.

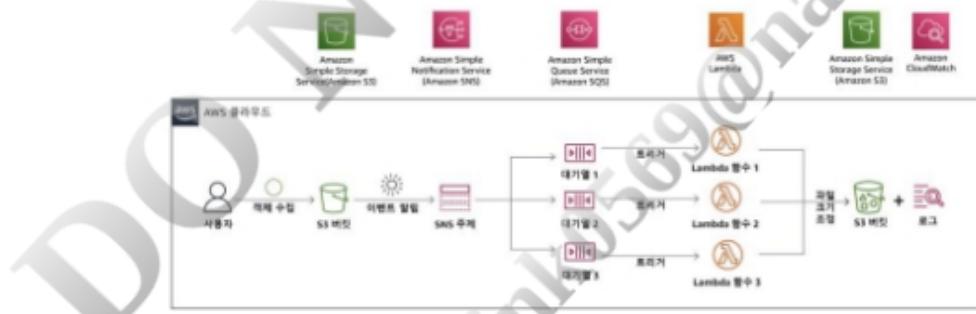
- Microsoft Windows, macOS 또는 Linux(Ubuntu, SuSE, Red Hat)가 실행되는 Wi-Fi 지원 노트북
- 인터넷 브라우저(예: Chrome, Firefox 또는 Microsoft Edge)
- 일반 텍스트 편집기

시나리오

팀이 완전한 서비스 아키텍처를 평가하는 작업을 맡았습니다. 느슨한 결합을 달성하기 위해 제안된 아키텍처를 기반으로 EC2 Auto Scaling 그룹을 보다 비용 효율적인 Lambda 함수로 교체할 수 있습니다.

고객 관리 전문가들이 제품 스냅샷을 찍어 AWS 네트워크에 업로드합니다. AWS는 이미지를 저장한 다음 Python 스크립트를 실행하여 수집 S3 버킷에 업로드된 이미지 크기를 조정합니다. 수집 버킷에 파일을 업로드하면 SNS 주제에 대한 이벤트 알림이 호출됩니다. 그러면 SNS가 3개의 별도의 SQS 대기열에 알림을 배포합니다. 초기 설계는 크기 조정 작업이 수행될 때마다 Auto Scaling 그룹에서 EC2 인스턴스를 실행하는 것이었습니다. 하지만 권장 사항에 따라 EC2 인스턴스를 Lambda 함수로 교체하려 합니다. Lambda 함수는 이미지를 세 가지 형식으로 처리하고 출력을 S3 버킷에 저장합니다.

다음 디아그램은 이 워크플로를 보여 줍니다.



시나리오 워크플로는 다음과 같습니다.

- 1 이미지 파일을 Amazon S3 버킷에 업로드합니다.
- 2 버킷의 ingest 폴더에 파일을 업로드하면 SNS 주제에 대한 이벤트 알림이 호출됩니다.
- 3 그러면 SNS가 3개의 별도의 SQS 대기열에 알림을 배포합니다.
- 4 Lambda 함수는 이미지를 세 가지 형식으로 처리하고 출력을 S3 버킷 폴더에 저장합니다.
- 5 S3 버킷 폴더의 처리된 이미지를 검증하고 Amazon CloudWatch에서 로그합니다.

소요 시간

이 실습을 완료하는 데는 약 **45분**이 소요됩니다.

DO NOT COPY
pink0569@naver.com

실습 시작

1. 실습을 시작하려면 페이지 상단에서 실습 시작을 선택합니다.

그러면 실습 리소스를 프로비저닝하는 프로세스가 시작됩니다. 실습 리소스를 프로비저닝하는데 걸리는 예상 시간이 표시됩니다. 계속 진행하기 전에 사용할 리소스가 프로비저닝될 때까지 기다려야 합니다.

- ① 토큰을 입력하라는 메시지가 표시되면 여러분에게 배포된 토큰(또는 구매한 크레딧)을 사용하세요.

2. 실습을 열려면 콘솔 열기를 선택합니다.

새 웹 브라우저 탭에서 **AWS Management Console** 로그인 페이지가 열립니다.

3. **Sign in as IAM user** 페이지에서

- **IAM user name**에 awsstudent를 입력합니다.
- **Password**에 이 지침의 왼쪽에 나열된 **Password** 값을 복사하여 붙여넣습니다.
- **Sign in**을 선택합니다.

△ 별다른 지시가 없는 한 리전을 변경하지 마십시오.

일반적인 로그인 오류

오류: 우선 로그 아웃 필요

Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

You must first log out before logging into a different AWS account라는 메시지가 표시된다면 다음을 수행합니다.

- [click here](#)의 링크를 선택합니다.
- **Amazon Web Services Sign In** 웹 브라우저 탭을 닫고 초기 실습 페이지로 돌아갑니다.
- 콘솔 열기를 다시 선택합니다.

경우에 따라 일부 팝업 또는 스크립트 차단 웹 브라우저 확장 프로그램 때문에 **실습 시작** 버튼이 제대로 작동하지 않을 수 있습니다. 실습을 시작하는 데 문제가 있는 경우 다음을 수행합니다.

- 팝업 또는 스크립트 차단 프로그램의 허용 목록에 실습 도메인 이름을 추가하거나 차단 프로그램을 끕니다.
- 페이지를 새로 고친 후 다시 시도하십시오.

과제 1: Amazon SNS 주제 생성

과제 1.1: 표준 SNS 주제 생성

이 과제에서는 SNS 주제를 생성하고 Amazon SNS 주제를 구독합니다.

4. AWS 관리 콘솔의 AWS 검색 창을 사용해 Simple Notification Service를 검색하고 검색 결과 리스트에서 선택합니다.
5. 왼쪽 상단 근처의 메뉴 아이콘 을 선택하여 탐색 메뉴를 확장합니다.
6. 탐색 메뉴에서 Topics를 선택합니다.
7. Create topic 버튼을 선택합니다.

Create topic 페이지가 표시됩니다.

8. Create topic 페이지의 Details 섹션에서 다음을 수행합니다.
 - Type에서 Standard를 선택합니다.
 - Name: 고유한 SNS 주제 이름을 입력합니다. (예: resize-image-topic 뒤에 4자리 무작위 숫자)
9. Create topic 버튼을 선택합니다.

주제가 생성되고 resize-image-topic-XXXX 페이지가 표시됩니다. 주제의 이름, ARN, (선택적) 표시 이름, 주제 소유자의 AWS 계정 ID가 Details 섹션에 표시됩니다.

10. 주제 ARN 및 Topic owner 값을 메모장에 복사합니다. 이 값은 실습 뒷부분에서 필요합니다.
 - ARN 예 - arn:aws:sns:us-east-2:123456789012:MyTopic
 - Topic owner - 123456789123(12자리 숫자)

과제 2: 3개의 Amazon SQS 대기열을 생성하고 SNS 주제 구독

과제 2.1: 썬네일용 SQS 대기열 생성

11. AWS 관리 콘솔의 AWS 검색 창을 사용해 Simple Queue Service를 검색하고 검색 결과 리스트에서 선택합니다.
 12. SQS 홈 페이지에서 Create queue 버튼을 선택합니다.
- Create queue 페이지가 표시됩니다.
13. Create queue 페이지의 Details 섹션에서 다음을 수행합니다.

- Type에서 Standard 를 선택합니다(Standard 대기열 유형이 기본적으로 설정되어 있습니다).
 - Name: thumbnail-queue
14. 대기열 Configuration 파라미터의 기본값이 콘솔에서 설정되어 있습니다. 기본값을 사용합니다.
15. Create queue 버튼을 선택합니다.

Amazon SQS가 대기열을 생성하고 대기열에 대한 세부 정보가 있는 페이지를 표시합니다.

과제 2.2: SQS 대기열을 SNS 주제에 구독

16. 대기열의 세부 정보 페이지에서 SNS subscriptions 탭을 선택합니다.
17. Subscribe to Amazon SNS topic 버튼을 선택합니다.
- 새 Subscribe to Amazon SNS topic 페이지가 열립니다.
18. Specify an Amazon SNS topic available for this queue 메뉴에서 Use existing resource 아래의 resize-image-topic을 선택합니다.
- 참고:** 메뉴에 SNS 주제가 나열되지 않으면 Enter Amazon SNS topic ARN을 선택한 다음 앞서 복사한 주제의 ARN을 입력합니다.
19. Save 버튼을 선택합니다.
- 이제 SQS 대기열이 resize-image-topic-XXXX라는 SNS 주제를 구독합니다.

과제 2.3: SQS 대기열을 2개 더 생성하고 SNS 주제 구독

과제 2.1과 2.2를 반복하여 표준 SQS 대기열을 2개 더 생성합니다. 대기열 하나의 이름은 web-queue이고 다른 하나의 이름은 mobile-queue로 지정하여 이름이 resize-image-topic-XXXX인 기존 SNS 주제를 구독합니다. 2개의 추가 대기열에 다음 구성 사용합니다.

20. 웹 크기 이미지용 SQS 대기열 생성:
- Name: web-queue로 대기열을 생성합니다.
21. 모바일 크기 이미지용 SQS 대기열 생성:
- Name: mobile-queue로 대기열을 생성합니다.

과제 2.4: 구독 확인

구독 결과를 확인하려면 주제에 게시한 다음 주제가 대기열에 전송하는 메시지를 확인합니다.

22. AWS 관리 콘솔 브라우저 탭으로 돌아와, AWS 검색 창을 사용해 Simple Notification Service를 검색하고 검색 결과 리스트에서 선택합니다.

23. 왼쪽 탐색 창에서 **Topics**를 선택합니다.

24. **Topics** 페이지에서 **resize-image-topic-XXXX**를 선택합니다.

25. Publish message 버튼을 선택합니다.

콘솔에서 **Publish message to topic** 페이지가 열립니다.

26. Message details 섹션에서 다음을 수행합니다.

- **Subject - optional:** Hello world

27. Message body 섹션에서 다음을 수행합니다.

- **Identical payload for all delivery protocols**를 선택한 다음 Testing Hello world 메시지 또는 원하는 메시지를 입력합니다.

28. **Message attributes** 섹션에서 다음을 구성합니다.

- **Type:** String
- **Name:** Message
- **Value:** Hello World

29. Publish message 버튼을 선택합니다.

메시지가 주제에 게시되고, 콘솔에서 주제의 세부 정보 페이지가 열립니다. 게시된 메시지를 조사하려면 Amazon SQS로 이동합니다.

30. AWS 관리 콘솔의 AWS 검색 창을 사용해 Simple Queue Service를 검색하고 검색 결과 리스트에서 선택합니다.

31. 목록에서 임의의 대기열을 선택합니다.

32. Send and receive messages 버튼을 선택합니다.

33. **Send and receive messages** 페이지의 **Receive messages** 섹션에서 Poll for messages 버튼을 선택합니다.

34. **Message** 섹션을 찾습니다. 목록에서 채워진 임의의 메시지를 선택하여 메시지의 Details, Body, Attributes를 확인합니다.

Message Details 상자에는 주제에 게시한 제목 및 메시지가 포함된 JSON 문서가 있습니다.

35. Done 버튼을 선택합니다.

대기열에 알림 메시지를 전송하는 주제를 성공적으로 게시했습니다.

과제 3: SNS에 대한 Amazon S3 이벤트 알림 생성

과제 3.1: SNS 액세스 정책을 구성하여 Amazon S3 버킷이 주제에 게시하도록 허용

36. AWS 관리 콘솔의 AWS 검색 창을 사용해 Simple Notification Service를 검색하고 검색 결과 리스트에서 선택합니다.
37. 탐색 메뉴에서 **Topics**를 선택합니다.
38. **resize-image-topic-XXXX** 주제를 선택합니다.
39. **Edit** 버튼을 선택합니다.
40. **Access policy**로 이동하고 필요한 경우 확장합니다.
41. JSON Editor 섹션에 아래 코드를 복사합니다.

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
        "SNS:RemovePermission",
        "SNS>DeleteTopic",
        "SNS:Subscribe",
        "SNS>ListSubscriptionsByTopic",
        "SNS:Publish",
        "SNS:Receive"
      ],
      "Resource": "Topic-ARN",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "Trusted Owner Value"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "Topic-ARN",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "Trusted Owner Value"
        }
      }
    }
  ]
}
```

```
}
```

42. **Trusted Owner Value**에 과제 1에서 복사한 숫자 12개 계정 ID로 변경하세요. 따옴표는 그대로 두세요.
43. **Topic-ARN**에 과제 1에서 복사한 ARN으로 변경하세요. 따옴표는 그대로 두세요.
44. Save changes 버튼을 선택합니다.

과제 3.2: 수집 S3 버킷에 업로드 시 단일 이벤트 알림 생성

45. AWS 관리 콘솔의 AWS 검색 창을 사용해 s3를 검색하고 검색 결과 리스트에서 선택합니다.
46. Buckets 페이지에서 이름이 `xxxxx-labbucket-xxxxx`와 비슷한 버킷을 선택합니다.
47. Properties 탭을 선택합니다.
48. Event notifications 섹션으로 이동합니다.
49. Create event notification 버튼을 선택합니다.
50. General configuration 섹션에서 다음을 수행합니다.
 - **Event name:** `resize-image-event`
 - **Prefix - optional:** `ingest/`

참고: 이 실습에서는 특정 폴더(`ingest`)에 파일이 추가될 때만 알림을 수신하도록 접두사 필터를 설정해야 합니다.

참고: 이 실습에서는 `.jpg` 파일이 업로드될 때만 알림을 수신하도록 접미사 필터를 설정해야 합니다.

51. Event types 섹션에서 **All object create events** 확인란을 선택합니다.
52. Destination 섹션에서 다음을 수행합니다.
 - **Destination:** SNS Topic 을 선택합니다.
 - **Specify SNS topic:** Choose from your SNS topics 를 선택합니다.
 - **SNS topic** 아래에서 `resize-image-topic-XXXX` SNS 주제를 드롭다운 메뉴에서 선택합니다. 또는 ARN을 지정하려면 Enter ARN을 선택하고 앞서 복사한 SNS 주제의 ARN을 입력합니다.
53. Save changes 버튼을 선택합니다.

과제 4: 3개의 AWS Lambda 함수 생성 및 구성

과제 4.1: 썸네일을 생성하는 Lambda 함수 생성

이 과제에서는 Amazon S3에서 이미지를 읽고 이미지 크기를 조정한 다음 새 이미지를 Amazon S3에 저장하는 AWS Lambda 함수를 SQS 트리거와 함께 생성합니다.

54. AWS 관리 콘솔의 AWS 검색 창을 사용해 Lambda를 검색하고 검색 결과 리스트에서 선택합니다.
55. Create function 버튼을 선택합니다.
56. Author from scratch를 선택합니다.
57. Create function 창에서 다음을 구성합니다.
 - Function name: CreateThumbnail
 - Runtime: Python 3.7
 - ▶ Change default execution role 섹션을 확장합니다.
 - Execution role에서 Use an existing role을 선택합니다.
 - Existing role: 이름이 XXXXX-LabExecutionRole-XXXXX와 비슷한 역할을 선택합니다.

이 역할은 Amazon S3와 Amazon SQS에 액세스하는 데 필요한 권한을 Lambda 함수에 부여합니다.

△ 런타임 선택시 Other supported 아래의 Python 3.7을 선택해야 합니다. Latest supported 아래의 Python 3.8을 선택하면 이 실습의 코드가 실패합니다.

58. Create function 버튼을 선택합니다.
- 함수 구성이 있는 페이지가 표시됩니다.

과제 4.2: Lambda 함수 구성 - SQS 트리거를 추가하고 Python 배포 패키지 업로드

AWS Lambda 함수는 Amazon Kinesis에서 데이터를 수신하거나 Amazon DynamoDB 데이터베이스에서 데이터를 업데이트하는 등의 작업에 의해 자동으로 트리거될 수 있습니다. 이 실습에서는 Amazon SQS 대기열에 새 객체가 푸시될 때마다 Lambda 함수를 트리거합니다.

59. Function overview 섹션에서 + Add trigger를 선택하고 다음을 구성합니다.
 - Select a trigger: SQS
 - SQS Queue: thumbnail-queue를 선택합니다.

- **Batch Size:** 1

60. 화면 맨 아래로 스크롤한 다음 Add 버튼을 선택합니다.

SQS trigger가 Function Overview 페이지에 추가됩니다. 이제 Lambda 함수를 구성합니다.

61. 아래 표시된 것처럼 **Code** 탭을 선택합니다.



62. 다음 설정을 구성하고 나열되지 않은 설정은 무시합니다.

- 이 zip 파일을 다운로드하고 저장합니다.

파일 이름을 마우스 오른쪽 버튼으로 클릭하고 zip 파일을 컴퓨터에 다운로드합니다.

[CreateThumbnail.zip](#)

- **Upload from ▶** 메뉴를 선택하고 **.zip file**을 선택합니다.
- **Upload** 버튼을 선택하고 다운로드한 zip 파일을 업로드합니다.
- **Save** 버튼을 선택합니다.

CreateThumbnail.zip 파일에는 다음 Lambda 함수가 들어 있습니다.

△ 다음 코드를 복사하지 마십시오. Zip 파일의 내용만 보여 주는 것입니다.

```
import boto3
import os
import sys
import uuid
from PIL import Image
import PIL.Image
import json
import time

s3_client = boto3.client('s3')
s3 = boto3.resource('s3')

def resize_image(image_path, resized_path):
    with Image.open(image_path) as image:
        image.thumbnail((128, 128))
        image.save(resized_path)

def handler(event, context):
    for record in event['Records']:
```

```

payload = record["body"]
sqS_message=json.loads(str(payload))
bucket_name =
json.loads(str(sqS_message["Message"]))["Records"][0]["s3"]["bucket"]["name"]
print(bucket_name)

key=json.loads(str(sqS_message["Message"]))["Records"][0]["s3"]["object"]["key"]
print(key)
download_path = '/tmp/{}'.format(uuid.uuid4(), key.split("/")[-1])
upload_path = '/tmp/resized-{}'.format(key.split("/")[-1])

s3_client.download_file(bucket_name, key, download_path)
resize_image(download_path, upload_path)
s3.meta.client.upload_file(upload_path, bucket_name, 'thumbnail/Thumbnail-
'+key.split("/")[-1])

```

63. 위의 코드를 검사합니다. 이 코드는 다음 단계를 수행합니다.

- 수신 객체(버킷, 키)의 이름이 포함된 이벤트를 수신합니다.
- 로컬 스토리지에 이미지를 다운로드합니다.
- *Pillow* 라이브러리를 사용하여 이미지 크기를 조정합니다.
- 크기가 조정된 이미지를 생성하고 새 폴더에 업로드합니다.

64. **Runtime settings** 섹션에서 **Edit** 버튼을 선택합니다.

- **Handler** 다음을 입력합니다.

`CreateThumbnail.handler`

65. **Save** 버튼을 선택합니다.

△ **Handler** 필드를 위의 값으로 설정했는지 확인하십시오. 값이 다르면 Lambda 함수를 찾을 수 없습니다.

66. **Configuration** 탭을 선택합니다.

67. **General configuration**을 선택합니다.

68. **Edit** 버튼을 선택합니다.

- **Description** 다음을 입력합니다.

`Create a thumbnail-sized image`

다른 설정은 기본값 그대로 두되 이러한 설정에 대한 간략한 설명은 다음과 같습니다.

- **Memory**는 함수에 할당되는 리소스를 정의합니다. 메모리를 늘리면 함수에 할당되는 CPU도 증가합니다.
- **Timeout**은 최대 함수 실행 시간을 설정합니다.

69. **Save** 버튼을 선택합니다.

이제 Lambda 함수가 구성되었습니다.

과제 4.3: AWS Lambda 함수 2개 더 생성 및 구성

과제 4.1과 4.2를 반복하여 다음 구성으로 Lambda 함수를 2개 더 생성하고 구성합니다(웹 및 모바일 이미지 생성용).

70. 웹 이미지를 생성하는 Lambda 함수 생성

- **Function name:** CreateWebImage
- **Select a trigger:** SQS
- **SQS Queue:** web-queue를 선택합니다.
- **Batch Size:** 1
- 이 zip 파일을 다운로드하고 저장합니다.

파일 이름을 마우스 오른쪽 버튼으로 클릭하고 zip 파일을 컴퓨터에 다운로드합니다.

[CreateWebImage.zip](#)

- **Runtime settings** 섹션에서 Edit 버튼을 선택합니다.
- **Handler** 다음을 입력합니다.

`CreateWebImage.handler`

- **Description** 다음을 입력합니다.

`Create a web-sized image`

71. 모바일 이미지를 생성하는 Lambda 함수 생성

- **Function name:** CreateMobileImage
- **Select a trigger:** SQS
- **SQS Queue:** mobile-queue를 생성합니다.
- **Batch Size:** 1
- 이 zip 파일을 다운로드하고 저장합니다.

파일 이름을 마우스 오른쪽 버튼으로 클릭하고 zip 파일을 컴퓨터에 다운로드합니다.

[CreateMobileImage.zip](#)

- **Runtime settings** 섹션에서 Edit 버튼을 선택합니다.
- **Handler** 다음을 입력합니다.

`CreateMobileImage.handler`

- **Description** 다음을 입력합니다.

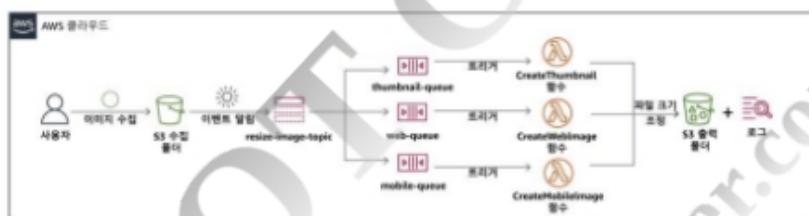
`Create a mobile-sized image`

이렇게 구성하면 3개의 Lambda 함수 모두 테스트할 준비가 됩니다.

과제 5: Amazon S3 버킷에 객체 업로드

과제 5.1: 버킷 폴더에 프로세스용 이미지 업로드

다음 다이어그램은 이 워크플로를 보여 줍니다.



구축한 것을 테스트하기 위해 그림을 업로드합니다.

72. 아래 옵션에서 이미지 하나를 다운로드하도록 선택합니다.

- 이 [링크](#)를 마우스 오른쪽 버튼으로 선택하고 그림을 컴퓨터에 다운로드합니다. [AWS.jpg](#)
- 이 [링크](#)를 마우스 오른쪽 버튼으로 선택하고 그림을 컴퓨터에 다운로드합니다. [MonaLisa.jpg](#)
- 이 [링크](#)를 마우스 오른쪽 버튼으로 선택하고 그림을 컴퓨터에 다운로드합니다. [HappyFace.jpg](#)
- **InputFile.jpg**와 비슷한 파일 이름을 지정합니다.

△ Firefox 사용자: 저장된 파일 이름이 *InputFile.jpg*인지 확인하십시오(.jpeg 아님).

73. AWS 관리 콘솔의 AWS 검색 창을 사용해 s3를 검색하고 검색 결과 리스트에서 선택합니다.

74. S3 관리 콘솔에서 **xxxxx-labbucket-xxxxx** 버킷을 선택합니다.

75. 버킷 안에서 **ingest/** 폴더를 선택합니다.

76. Upload 버튼을 선택합니다.

77. Upload 창에서 Add files를 선택합니다.

78. 다운로드한 그림 **XXXXXX.jpg**를 찾아 선택합니다.

79. Upload 버튼을 선택합니다.

80. 업로드 확인 메시지가 표시됩니다.

과제 6: 처리된 파일 확인

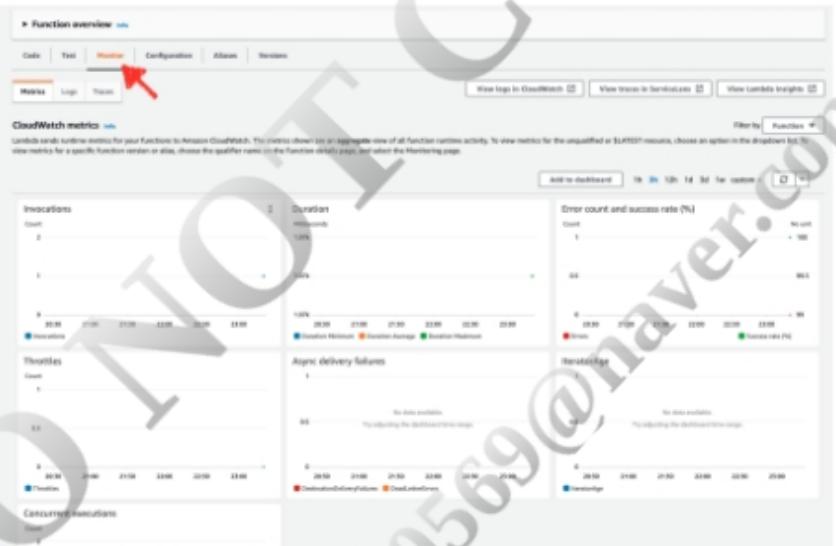
과제 6.1: Lambda 작업의 Amazon CloudWatch Logs 확인

AWS Lambda 함수를 모니터링하여 문제를 식별하고 로그 파일을 확인하여 디버그할 수 있습니다.

81. AWS 관리 콘솔의 AWS 검색 창을 사용해 Lambda를 검색하고 검색 결과 리스트에서 선택합니다.

82. **Create-** 함수 중 하나를 선택합니다.

83. 아래와 같이 **Monitor** 탭을 선택합니다.



콘솔에 다음 데이터를 보여 주는 그래프가 표시됩니다.

- **Invocations:** 함수가 호출된 횟수입니다.
- **Duration:** 평균, 최소, 최대 실행 시간입니다.
- **Error count and success rate (%):** 오류 수 및 오류 없이 완료된 실행의 비율입니다.
- **Throttles:** 너무 많은 함수가 동시에 호출되면 함수가 제한됩니다. 기본값은 1,000개 동시 실행입니다.
- **Async delivery failures:** Lambda가 대상 또는 배달 못한 편지 대기열에 쓰려고 할 때 발생한 오류의 개수입니다.

- **Iterator Age:** 스트리밍 트리거(Amazon Kinesis 및 Amazon DynamoDB Streams)에서 처리된 마지막 레코드의 경과 시간을 측정합니다.

- **Concurrent executions:** 이벤트를 처리 중인 함수 인스턴스의 개수입니다.

Lambda 함수의 로그 메시지는 **Amazon CloudWatch Logs**에 보관됩니다.

84. View logs in CloudWatch 버튼을 선택합니다.

85. 나타나는 **Log Stream**을 선택합니다.

86. ▶ 각 메시지를 확장하여 로그 메시지 세부 정보를 확인합니다.

REPORT 라인에서는 아래와 같은 자세한 사항을 제공합니다.

- **RequestId:** 함수 호출의 유일한 요청 ID입니다.

- **Duration:** 함수의 핸들러 메소드가 이벤트를 처리하는데 사용한 시간입니다.

- **Billed Duration:** 함수 호출에 관한 비용 청구 기준 시간입니다.

- **Memory Size:** 함수에 할당된 메모리 용량입니다.

- **Max Memory Used:** 함수가 사용한 최대 메모리 사용량입니다.

- **Init Duration:** 처음 요청이 처리될 때, 런타임이 함수를 읽어들이고 핸들러 메소드 외부의 코드를 실행하기 까지 걸린 시간입니다.

덧붙여서, 함수의 로깅 메시지 및 출력 상태들이 logs에 표시가 됩니다. 이것은 람다 함수의 디버깅에 도움이 됩니다.

과제 6.2: S3 버킷에서 처리된 파일 확인

87. AWS 관리 콘솔의 AWS 검색 창을 사용해 s3를 검색하고 검색 결과 리스트에서 선택합니다.

88. **xxxxx-labbucket-xxxxx**를 선택하여 버킷에 들어갑니다.

89. 이제 다음 3개의 새 폴더가 표시됩니다.

- **thumbnail**
- **web**
- **mobile**

90. 이 폴더를 탐색하여 크기 조정된 이미지(예: *Thumbnail-AWS.jpg*, *WebImage-HappyFace.jpg*, *MobileImage-MonaLisa.jpg*)를 찾습니다.

크기 조정된 이미지를 여기서 찾는다면 원본의 이미지를 세 가지 형식으로 성공적으로 크기 조정한 것입니다.

선택적 과제

참고: 도전 과제는 선택 사항이며, 실습 시간이 남는 경우에 제공됩니다. 이 선택적 과제를 완료하거나 여기를 선택하여 실습 끝으로 건너뛸 수 있습니다.

- 선택적 과제 1: 30일 후 수집 버킷에서 파일을 삭제하는 수명 주기 구성 생성

참고: 선택적 과제를 완료하는 데 문제가 있다면 실습 마지막 부분의 선택적 과제 1 해법 부록 섹션을 참조하십시오.

- 선택적 과제 2: 기존 SNS 주제에 SNS 이메일 알림 추가

참고: 선택적 과제를 완료하는 데 문제가 있다면 실습 마지막 부분의 선택적 과제 2 해법 부록 섹션을 참조하십시오.

결론

▶ 축하합니다! 성공적으로 다음을 수행했습니다.

- Amazon SNS 주제 생성
- Amazon SQS 대기열 생성
- Amazon S3에서 이벤트 알림 생성
- 기존 코드를 사용하여 AWS Lambda 함수 생성
- SQS 대기열에서 AWS Lambda 함수 트리거
- Amazon CloudWatch 로그를 통해 AWS Lambda S3 함수 모니터링

실습 종료

다음 단계를 따라 콘솔을 닫고 실습을 종료한 후 실습 경험을 평가해 주십시오.

91. AWS Management Console로 돌아갑니다.
92. 탐색 모음에서 **awsstudent@<AccountNumber>** 를 선택한 다음 **Sign Out**을 선택합니다.
93. 실습 종료를 선택합니다.
94. OK를 선택합니다.
95. (선택 사항):
 - 해당하는 별 개수를 선택합니다.
 - 의견을 입력합니다.

- **Submit**을 선택합니다.
- 별 1개 = 매우 불만족
- 별 2개 = 불만족
- 별 3개 = 보통
- 별 4개 = 만족
- 별 5개 = 매우 만족

피드백을 제공하지 않으려면 그냥 창을 닫으면 됩니다.

부록

선택적 과제 1 해법: 30일 후 수집 버킷에서 파일을 삭제하는 수명 주기 구성 생성

96. Services 메뉴에서 **S3**를 선택합니다.
97. **Buckets** 페이지에서 이름이 **xxxxx-labbucket-xxxxx**인 버킷을 선택합니다.
98. **Management** 탭을 선택합니다.
99. **Lifecycle rules** 섹션에서 **Create lifecycle rule** 버튼을 선택합니다.
100. **Lifecycle rule configuration** 섹션에서 다음을 구성합니다.
 - **Lifecycle rule name:** cleanup
 - **Choose a rule scope:** Limit the scope of this rule using one or more filters를 선택합니다.
101. **Filter type** 섹션에서 다음을 구성합니다.
 - **Prefix:** ingest/
102. **Lifecycle rule actions** 섹션에서 다음을 구성합니다.
 - **Expire current versions of objects** 및 **Permanently delete noncurrent versions of objects** 확인란을 선택합니다.
 - 열리는 새 상자에 아래 값을 입력합니다.
 - **Days after object creation:** 30
 - **Days after objects become noncurrent:** 1

103. Create rule 버튼을 선택합니다.

선택적 과제 2 해법: 기존 SNS 주제에 SNS 이메일 알림 추가

104. Services 메뉴에서 **Simple Notification Service**를 선택합니다.
105. 왼쪽 탐색 창에서 **Subscriptions**를 선택합니다.
106. **Subscriptions** 페이지에서 **Create subscription**을 선택합니다.
107. **Create subscription** 페이지의 Details 섹션에서 다음을 수행합니다.
 - **Topic ARN**에서 생성한 주제의 ARN을 선택합니다.
 - **Protocol**에서 *Email*을 선택합니다.
 - **Endpoint**에 유효한 이메일 주소를 입력합니다.
108. Create Subscription 버튼을 선택합니다.

콘솔이 구독을 생성하고 구독의 Details 페이지를 엽니다.

참고: 이메일 주소가 메시지 수신을 시작하려면 먼저 구독을 확인해야 합니다.

109. 구독을 확인하려면 받은 편지함을 확인하고 Amazon SNS가 보낸 이메일에서 구독 확인을 선택합니다.
110. Amazon SNS에서 웹 브라우저를 열고 구독 ID와 함께 구독 확인을 표시합니다

AWS Training and Certification에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다.

피드백, 제안 사항 또는 수정 요청 사항을 제공하려면 AWS Training and Certification 문의 양식에 세부 정보를 입력해 주시기 바랍니다.



실습 6: Amazon S3 오리진으로 Amazon CloudFront 배포 구성

© 2022 Amazon Web Services, Inc. 또는 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 임대 또는 판매는 금지됩니다. 모든 상표는 해당 소유자의 자산입니다.

참고: 개인 정보, 개인을 식별할 수 있는 정보 또는 기밀 정보는 실습 환경에 입력하지 마십시오. 입력한 정보가 다른 사용자에게 공개될 수도 있습니다.

수정 사항이나 피드백 또는 기타 질문이 있으십니까? AWS Training and Certification에서 문의해 주십시오.

실습 개요

AWS 솔루션 아키텍트는 콘텐츠 제공을 위해 안전하고 성능이 뛰어나며 안정적이고 효율적인 애플리케이션 및 워크로드 아키텍처를 설계 및 구축하는 일을 자주 맡게 됩니다. Amazon CloudFront는 짧은 대기 시간과 빠른 데이터 전송 속도로 콘텐츠를 간편하고 비용 효율적으로 배포할 수 있는 방법을 제공하는 웹 서비스입니다. Amazon CloudFront를 사용하면 정적 웹 사이트 콘텐츠 전송을 가속화하고, 온디맨드 비디오 또는 라이브 스트리밍 비디오를 제공하며, 엣지 로케이션에서 서비스 코드도 실행할 수 있습니다. 이 실습에서는 S3 버킷 앞에 CloudFront 배포를 구성하여 CloudFront에서 제공하는 오리진 액세스 ID(OAI)를 통해 보호합니다.

목표

이 실습을 마치면 다음을 수행할 수 있습니다.

- 기본 보안 설정으로 Amazon S3 버킷 생성
- 퍼블릭 액세스가 가능하도록 버킷 구성
- 기존 CloudFront 배포에 Amazon S3 버킷을 새 오리진으로 추가
- CloudFront 배포를 통한 액세스만 허용하도록 Amazon S3 버킷 보호
- 오리진 액세스 ID(OAI)를 구성하여 Amazon S3 버킷에 대한 보안 잠금
- 퍼블릭 또는 OAI 액세스가 가능하도록 Amazon S3 리소스 구성

수강 전 권장 사항

본 실습에는 다음이 필요합니다.

- Microsoft Windows, macOS 또는 Linux(Ubuntu, SuSE, Red Hat)가 실행되는 Wi-Fi 지원 노트북
- 인터넷 브라우저(예: Chrome, Firefox 또는 Microsoft Edge)

소요 시간

이 실습을 완료하는 데는 약 **60분**이 소요됩니다.

본 실습에서 사용하지 않는 AWS 서비스

이 실습에서 사용하지 않는 AWS 서비스는 실습 환경에서 비활성화됩니다. 또한 이 실습에 사용되는 서비스의 기능은 실습에 필요한 작업으로 제한됩니다. 다른 서비스에 액세스하거나 실습 안내서에서 제공하는 것 외의 작업을 수행하는 경우 오류가 발생할 수 있습니다.

필수 기술 지식

이 실습을 성공적으로 완료하려면 AWS 관리 콘솔에 익숙하고 AWS Cloud의 엣지 서비스에 대한 기본적 이해가 있어야 합니다.

Amazon CloudFront

Amazon CloudFront는 콘텐츠 전송 웹 서비스입니다. 다른 Amazon Web Services 제품과 통합하여 사용하면 개발자와 기업에서 최소 사용 약정 없이도 짧은 대기 시간과 빠른 데이터 전송 속도로 최종 사용자에게 쉽게 콘텐츠를 배포할 수 있습니다.

Amazon CloudFront는 엣지 로케이션의 글로벌 네트워크를 통해 동적, 정적, 스트리밍 및 대화형 콘텐츠를 비롯한 전체 웹사이트를 제공하는 데 사용할 수 있습니다. 콘텐츠에 대한 요청이 가장 가까운 엣지 로케이션으로 자동 라우팅되므로 콘텐츠 전송 성능이 아주 뛰어납니다. Amazon CloudFront는 Amazon Simple Storage Service(Amazon S3), Amazon Elastic Compute Cloud(Amazon EC2), Amazon Elastic Load Balancing 및 Amazon Route 53와 같은 다른 Amazon Web Services와 연동하도록 최적화되어 있습니다. 또한, Amazon CloudFront는 오리진 최종 파일 버전을 저장하는 AWS 오리진 서버 이외의 다른 서버에서도 원활하게 작동합니다.

Amazon S3

Amazon Simple Storage Service(Amazon S3)는 개발자와 IT 팀에 안전하고 내구성과 확장성이 뛰어난 객체 스토리지를 제공합니다. Amazon S3는 간편한 웹 서비스 인터페이스를 통해 웹상 어디서나 원하는 양의 데이터를 저장 및 검색할 수 있으며 사용이 간편합니다.

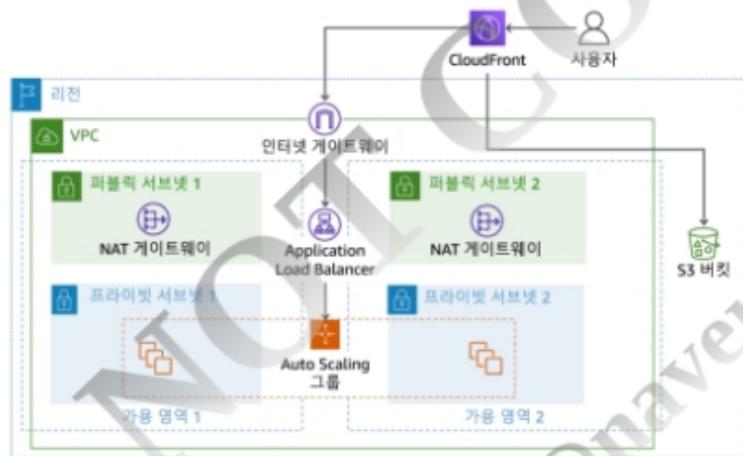
Amazon S3는 단독으로 사용하거나 Amazon Elastic Compute Cloud(Amazon EC2), Amazon Elastic Block Store(Amazon EBS) 및 Amazon Glacier와 같은 다른 AWS 서비스는 물론 서드 파티 스토리지 리포지토리 및 게이트웨이와 함께 사용할 수 있습니다. Amazon S3는 클라우드

애플리케이션, 콘텐츠 배포, 백업 및 아카이빙, 재해 복구, 빅 데이터 분석을 비롯하여 다양한 사용 사례에 적합한 비용 효율적인 객체 스토리지를 제공합니다.

실습 환경

실습 환경에서 함께 시작할 수 있는 몇 가지 리소스가 제공됩니다. 퍼블릭 액세스가 가능한 웹 서버로 활용되는 Amazon EC2 인스턴스의 Auto Scaling 그룹이 있습니다. 웹 서버 인프라가 Amazon Virtual Private Cloud(Amazon VPC)에 배포되고 다중 가용 영역에 맞게 구성되어 로드 밸런서를 활용합니다. 실습에서는 이 로드 밸런서를 오리진으로 하는 CloudFront 배포도 제공합니다.

다음 디어그램은 이 실습이 끝날 때의 일반적 아키텍처를 보여 줍니다. 이 실습에서는 기존 실습 환경을 위한 새 Amazon S3 버킷을 생성합니다. 그런 다음 기존 CloudFront 배포의 안전한 새 오리진으로 이 버킷을 구성합니다.



실습 시작

1. 실습을 시작하려면 페이지 상단에서 실습 시작을 선택합니다.

그러면 실습 리소스를 프로비저닝하는 프로세스가 시작됩니다. 실습 리소스를 프로비저닝하는데 걸리는 예상 시간이 표시됩니다. 계속 진행하기 전에 사용할 리소스가 프로비저닝될 때까지 기다려야 합니다.

① 토큰을 입력하라는 메시지가 표시되면 여러분에게 배포된 토큰(또는 구매한 크레딧)을 사용하세요.

2. 실습을 열려면 콘솔 열기를 선택합니다.

새 웹 브라우저 탭에서 **AWS Management Console** 로그인 페이지가 열립니다.

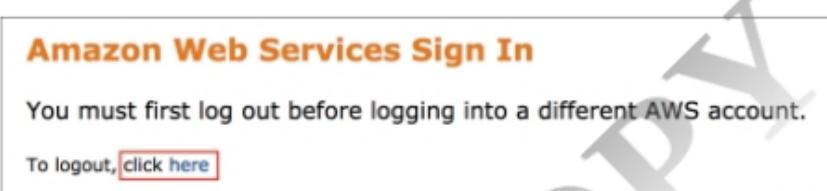
3. **Sign in as IAM user** 페이지에서

- IAM user name에 awsstudent를 입력합니다.
- Password에 이 지침의 왼쪽에 나열된 Password 값을 복사하여 붙여넣습니다.
- Sign in을 선택합니다.

△ 별다른 지시가 없는 한 리전을 변경하지 마십시오.

일반적인 로그인 오류

오류: 우선 로그 아웃 필요



You must first log out before logging into a different AWS account라는 메시지가 표시된다면 다음을 수행합니다.

- click here의 링크를 선택합니다.
- Amazon Web Services Sign In 웹 브라우저 탭을 닫고 초기 실습 페이지로 돌아갑니다.
- 콘솔 열기를 다시 선택합니다.

경우에 따라 일부 팝업 또는 스크립트 차단 웹 브라우저 확장 프로그램 때문에 실습 시작 버튼이 제대로 작동하지 않을 수 있습니다. 실습을 시작하는 데 문제가 있는 경우 다음을 수행합니다.

- 팝업 또는 스크립트 차단 프로그램의 허용 목록에 실습 도메인 이름을 추가하거나 차단 프로그램을 끕니다.
- 페이지를 새로고침 후 다시 시도하십시오.

과제 1: 기존 CloudFront 배포 살펴보기

이 과제에서는 웹 서버 콘텐츠용으로 구축된 기존 CloudFront 배포를 검사합니다. 환경을 변경하기 전에 기존 구성을 이해하는 것이 좋습니다. 개인 AWS 환경에 CloudFront 배포를 활용하려는 경우 먼저 배포 자체를 구축하고 구성해야 합니다. 이후 과제에서는 이 CloudFront 배포에 Amazon S3 버킷을 오리진으로 추가합니다.

과제 1.1: CloudFront 콘솔 열기

4. AWS 관리 콘솔을 아직 열지 않은 경우 **Start Lab** 섹션의 지침에 따라 AWS 관리 콘솔에 로그인합니다.
5. Services 메뉴에서 **CloudFront**을 선택합니다.

참고: 콘솔 상단의 통합 검색 창에서 **CloudFront**을 검색할 수도 있습니다.

과제 1.2: 기존 CloudFront 배포 열기

6. 사용할 수 있는 유일한 배포의 ID 링크를 선택합니다.

참고: 배포 목록을 찾을 수 없다면 올바른 페이지에 있는지 확인합니다. 콘솔 왼쪽에 있는 CloudFront 탐색 메뉴에서 **Distributions**를 선택합니다.

배포의 세부 정보를 보여 주는 페이지가 표시됩니다.

과제 1.3: 기존 배포의 속성 살펴보기

이 과제에서는 배포의 각 탭을 탐색하여 기존 구성을 검토합니다. 이 실습에서는 이 CloudFront 배포를 자세히 구성하지 않습니다. 하지만 CloudFront 배포 관리에 필요할 수 있는 모든 구성이 어디에 있는지 알면 유용합니다.

7. **General** 탭의 내용을 검사합니다.

이 탭에는 이 특정 CloudFront 배포의 현재 구성에 대한 세부 정보가 포함되어 있습니다. 일반적으로 가장 필요한 배포 정보가 포함되어 있습니다. 또한 배포 활성화, 로깅, 인증서 설정 등 배포의 일반적인 상위 수준 항목을 이 탭에서 구성합니다.

8. **General** 탭의 **Distribution domain name** 필드에서 배포의 Distribution domain 값을 복사합니다.

이 배포의 Distribution domain 값은 이 실습 지침 왼쪽의 *LabCloudFrontDistributionDNS* 목록에서도 찾을 수 있습니다.

9. 복사한 Distribution domain 값을 새 브라우저 탭에 붙여넣습니다.

CloudFront가 콘텐츠를 검색한 웹 서버의 정보가 표시된 간단한 웹 페이지가 로드됩니다. CloudFront 배포의 Distribution domain 값에서 콘텐츠를 요청하여 기존 캐시가 작동 중임을 확인합니다.

이 탭을 닫아도 됩니다.

10. **CloudFront 콘솔로 돌아갑니다.**

11. **Origins** 탭을 선택합니다.

이 탭에는 이 특정 CloudFront 배포를 위해 존재하는 현재 오리진에 대한 세부 정보가 포함되어 있습니다. 또한 기존 또는 새 CloudFront 오리진을 구성하는데 사용하는 콘솔이 있습니다. *CloudFront Origin*은 CloudFront 배포를 통해 전송되는 콘텐츠의 최종 오리진 버전의 위치를 정의합니다.

참고: 배포에서 유일한 오리진은 현재 Elastic Load Balancer(ELB)입니다. 이 ELB는 대상 그룹에 있는 Auto Scaling 웹 서버에 대한 웹 트래픽을 받고 보냅니다.

12. **Origin Domain**으로 레이블이 지정된 열에서 이 오리진의 ELB DNS 값을 복사합니다.

참고: 콘솔의 열 대부분은 헤더의 구분선을 끌어 너비를 조정할 수 있습니다.

13. ELB의 DNS 값을 새 브라우저 탭에 붙여넣습니다.

이 배포의 DNS 값은 이 실습 지침 왼쪽의 *LabLoadBalancerDNS* 목록에서도 찾을 수 있습니다.

Amazon EC2 인스턴스에서 호스트되는 간단한 웹 페이지가 다시 표시됩니다. 이 웹 페이지에는 CloudFront 배포가 앞서 전송한 것과 동일한 콘텐츠가 표시됩니다. 하지만 ELB 딕렉터리에서 직접 요청하면 기존 CloudFront 캐싱 시스템을 활용하지 못합니다. 트래픽이 항상 로드 밸런서 뒤에 있는 동일한 Amazon EC2 인스턴스로 라우팅되는 것은 아니므로 단일 요청에서는 페이지에 표시되는 IP 주소가 다를 수 있습니다.

이 단계는 배포를 위해 정의된 오리진이 CloudFront 배포의 프런트 엔드에 요청이 수행될 때 신규 콘텐츠를 검색하는 데 사용되는 위치임을 보여 주기 위한 단계입니다.

이 탭을 닫아도 됩니다.

14. **CloudFront 콘솔로 돌아갑니다.**

15. **Behaviors** 탭을 선택합니다.

*Behaviors*는 특정 콘텐츠를 제공할 오리진, 캐시에 있는 콘텐츠의 TTL(Time-To-Live), 다양한 헤더의 처리 방법 등 콘텐츠에 대한 요청이 있을 때 CloudFront 배포가 수행하는 작업을 정의합니다.

이 탭에는 배포에 대해 정의된 현재 동작의 목록이 포함되어 있습니다. 기존 또는 새 동작이 여기에서 구성됩니다. 배포의 동작은 이 탭에서 정의되는 명시적 순서에 따라 평가됩니다.

단일 요청의 구성을 검토하거나 편집하려면 다음을 수행합니다.

- Behaviors에서 행 옆에 있는 라디오 버튼을 선택합니다.

- **Edit** 버튼을 선택합니다.
- **Cancel**을 선택하여 페이지를 닫고 콘솔로 돌아갑니다.

이 실습 환경에는 현재 하나의 동작만 구성되어 있습니다. 이 동작은 ELB 오리진에 대한 GET 및 HEAD 요청의 HTTP 및 HTTPS를 수락합니다.

16. Error Pages 탭을 선택합니다.

이 탭에는 요청된 콘텐츠로 인해 HTTP 4xx 또는 5xx 상태 코드가 발생할 때 사용자에게 반환될 오류 페이지가 자세히 설명되어 있습니다. 특정 오류 코드에 대한 사용자 지정 오류 페이지도 여기에서 구성할 수 있습니다.

17. Geographic restrictions 탭을 선택합니다.

이 탭에는 선택한 국가의 사용자가 콘텐츠에 액세스하지 못하도록 해야 하는 경우의 배포 구성이 포함되어 있습니다. 이 실습에서는 이 기능을 사용하도록 구성되어 있지 않습니다.

18. Invalidations 탭을 선택합니다.

이 탭에는 객체 무효화에 대한 배포의 구성이 포함되어 있습니다. 무효화된 객체는 CloudFront 엣지 캐시에서 제거됩니다. 보다 빠르고 저렴한 방법은 버전 지정된 객체 또는 디렉터리 이름을 사용하는 것입니다. 기본적으로 CloudFront 배포에 대해 구성된 무효화는 없습니다.

19. Tags 탭을 선택합니다.

이 탭에는 배포에 적용되는 태그의 구성이 포함되어 있습니다. 기존 태그를 보고 편집할 수 있습니다. 새 태그도 여기에서 생성할 수 있습니다. 태그는 배포를 식별하고 구성하는 데 도움이 됩니다.

축하합니다. 기존 Amazon CloudFront 배포를 살펴봤습니다.

과제 2: Amazon S3 버킷 생성

이 과제에서는 새 Amazon S3 버킷을 생성하고 구성합니다. 이 버킷은 CloudFront 배포의 새 오리진으로 사용됩니다.

20. Services 메뉴에서 S3를 선택합니다.

참고: 콘솔 상단의 통합 검색 창에서 s3를 검색할 수도 있습니다.

21. Create bucket 버튼을 선택합니다.

참고: Create bucket 버튼을 찾을 수 없으면 올바른 페이지에 있는지 확인하십시오. 콘솔 왼쪽에 있는 탐색 메뉴에서 **Buckets**를 선택합니다.

Create bucket 페이지가 표시됩니다.

22. Bucket name 필드에 고유한 버킷 이름을 입력합니다. 예시 버킷 이름은 사용자 이니셜 다음에 4자리 무작위 숫자가 오는 것입니다.

Amazon S3 버킷 이름은 전역적으로 고유하고 DNS를 준수해야 합니다. 전체 버킷 명명 규칙은 <https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html> 링크의 공식 설명서를 참조하십시오.

참고: 이 실습의 지침을 간단히 하기 위해 이 단계에서 실제로 선택하는 버킷 이름에 상관없이 지침 나머지 부분에서는 새로 생성된 이 버킷을 *LabBucket*이라고 하겠습니다.

23. AWS Region은 이 실습 지침 원쪽에 있는 *PrimaryRegion* 값과 일치해야 합니다.

24. 이 페이지의 나머지 설정은 모두 기본 구성 그대로 두십시오.

25. Create bucket 버튼을 선택합니다.

Amazon S3 콘솔이 표시됩니다. 계정의 모든 버킷 목록에 새로 생성된 버킷이 표시됩니다.

축하합니다. 기본 구성으로 새 Amazon S3 버킷을 생성했습니다.

과제 3: 퍼블릭 액세스가 가능하도록 Amazon S3 LabBucket 구성

이 과제에서는 Amazon S3 버킷의 기본 액세스 설정을 검토합니다. 다음으로 버킷에 대한 퍼블릭 액세스를 허용하도록 권한 설정을 수정합니다.

과제 3.1: 퍼블릭 정책을 생성할 수 있도록 LabBucket 구성

26. **Buckets** 섹션에 있는 새로 생성된 *LabBucket*의 링크를 선택합니다.

모든 버킷 세부 정보가 포함된 페이지가 표시됩니다.

27. **Permissions** 탭을 선택합니다.

28. **Block public access (bucket settings)** 섹션을 찾습니다.

29. **Edit** 버튼을 선택합니다.

Edit Block public access (bucket settings) 페이지가 표시됩니다.

30. **Block all public access** 옆의 확인란 선택을 취소합니다.

31. **Save Changes** 버튼을 선택합니다.

Edit Block public access (bucket settings)라는 제목의 메시지 창이 표시됩니다.

32. 메시지 창의 필드에 `confirm`을 입력합니다.

33. **Confirm** 버튼을 선택합니다.

*LabBucket*의 모든 퍼블릭 액세스 정책에서 블록을 제거했습니다. 이제 퍼블릭 액세스를 허용하는 버킷의 액세스 정책을 생성할 수 있습니다. 버킷은 현재 퍼블릭이 아니지만 적절한 권한이 있으면 누구나 버킷 내에 저장된 객체에 대한 퍼블릭 액세스 권한을 부여할 수 있습니다.

과제 3.2: LabBucket의 퍼블릭 읽기 정책 구성

이제 이 버킷의 퍼블릭 객체 읽기 정책을 생성합니다.

34. **Permissions** 탭에서 **Bucket policy** 섹션을 찾습니다.

35. **Edit** 버튼을 선택합니다.

Edit bucket policy 페이지가 표시됩니다.

36. **Bucket ARN** 값을 복사합니다. 이 값은 *Policy* 상자 위에 있는 arn:aws:s3:::LabBucket과 같은 문자열 값입니다.

ARN 값은 이 Amazon S3 버킷을 고유하게 식별합니다. 버킷 기반 정책을 생성할 때 이 특정 ARN 값이 필요합니다.

37. 아래의 JSON 문장을 텍스트 에디터에 복사, 붙여넣기 합니다.

```
{
    "Version": "2012-10-17",
    "Id": "Policy1621958846486",
    "Statement": [
        {
            "Sid": "OriginalPublicReadPolicy",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "RESOURCE_ARN"
        }
    ]
}
```

38. JSON 문장에서 **RESOURCE_ARN** 값을 이전 작업에서 복사해둔 **Bucket ARN** 값으로 대체합니다. 그리고 **Bucket ARN** 값 마지막에 /*를 추가합니다.

① 와일드카드를 Principal 값으로 사용하면 정책 문서에 정의된 작업을 요청하는 모든 보안 주체가 작업을 수행할 수 있습니다. 또한 허용되는 Resource에 /*와일드카드를 추가하면 버킷에 있는 모든 객체에 이 정책이 적용됩니다.

다음은 생성된 JSON 정책의 예입니다.

```
{
    "Version": "2012-10-17",
    "Id": "Policy1621958846486",
    "Statement": [
        {
            "Sid": "OriginalPublicReadPolicy",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::LabBucket/*"
        }
    ]
}
```

```
    ]  
}
```

39. **Amazon S3 콘솔**로 돌아갑니다.
40. JSON을 *Policy* 상자에 붙여넣습니다.
41. **Save Changes** 버튼을 선택합니다.

⚠ 경고: 화면 하단에 오류 메시지가 표시되면 JSON 구문 오류가 원인일 수 있습니다. 정책은 JSON이 유효할 때까지 저장되지 않습니다. Amazon S3 콘솔에서 오류 메시지를 확장하여 정책 연결에 대한 자세한 내용을 볼 수 있습니다.

42. 버킷 이름 아래에 빨간색 *Public accessible* 경고가 있습니다. *Permissions overview* 섹션에도 ⚠ *Public* 경고가 있습니다.

이러한 경고는 버킷에 현재 적용되는 정책으로 인해 이 버킷의 객체에 퍼블릭 액세스가 가능하다는 것을 알리기 위한 것입니다.

이후 실습 단계에서는 CloudFront 배포에서만 액세스할 수 있도록 버킷을 구성합니다.

축하합니다. 퍼블릭 읽기 액세스가 가능하도록 Amazon S3 버킷을 구성했습니다.

과제 4: 버킷에 객체를 업로드하고 퍼블릭 액세스 테스트

이 과제에서는 단일 객체를 *LabBucket*에 업로드합니다. 나머지 실습 과제에서 이 객체를 사용하여 액세스를 테스트합니다.

과제 4.1: 버킷에 새 폴더 생성

43. **Objects** 탭을 선택합니다.
44. **Create folder** 버튼을 선택합니다.
45. *images* 값을 **Folder** 이름 필드에 입력합니다.
46. 페이지의 다른 모든 설정은 기본값으로 둡니다.
47. **Create folder** 버튼을 선택합니다.

과제 4.2: 버킷에 객체 업로드

48. 링크 [logo.png](#)를 마우스 오른쪽 버튼으로 클릭하고 로컬 디바이스에 저장을 선택하여 이 실습 지침을 위한 객체를 다운로드합니다.
49. Amazon S3 콘솔로 돌아갑니다.
50. 이전에 생성한 *images/* 폴더의 링크를 선택합니다.

51. **Upload** 버튼을 선택합니다.

Upload 페이지가 표시됩니다.

52. **Add files** 버튼을 선택합니다.

53. 로컬 스토리지 위치에서 *logo.png* 객체를 선택합니다.

54. **Upload** 버튼을 선택합니다.

Upload: status 페이지가 표시됩니다.

*LabBucket*에 파일 업로드가 완료되면 **Upload succeeded** 메시지와 함께 페이지 상단에 녹색 경계선이 표시됩니다.

과제 4.3: 객체에 대한 퍼블릭 액세스 테스트

55. **Files and folders** 섹션에서 *logo.png*의 링크를 선택합니다.

Amazon S3 객체에 대한 세부 정보가 있는 페이지가 표시됩니다.

56. **Object URL** 필드에 있는 링크를 선택합니다.

브라우저 탭에 그림이 표시됩니다.

57. 객체의 URL을 검사하여 Amazon S3 URL임을 확인합니다.

58. 객체가 있는 이 페이지를 닫습니다.

축하합니다. Amazon S3 버킷에 폴더를 생성하고, 객체를 업로드한 다음, 이 객체를 S3 URL에서 검색할 수 있는지 테스트했습니다.

과제 5: 객체를 배포에 오리진으로 추가

이 과제에서는 *LabBucket*을 기준 CloudFront 배포에 새 오리진으로 추가합니다.

과제 5.1: 새 오리진 및 오리진 액세스 ID 생성

59. 콘솔로 돌아갑니다.

60. Services 메뉴에서 **CloudFront**을 선택합니다.

참고: 콘솔 상단의 통합 검색 창에서 *cloudFront*을 검색할 수도 있습니다.

61. **CloudFront Distributions** 페이지에서 사용 가능한 유일한 배포의 ID 링크를 선택합니다.

배포의 세부 정보를 보여 주는 페이지가 표시됩니다.

62. **Origins** 탭을 선택합니다.

63. **Create origin** 버튼을 선택합니다.

Create Origin 페이지가 표시됩니다.

64. **Origin domain** 필드에서 **Amazon S3** 섹션에서 LabBucket의 이름을 선택합니다.

참고: S3 웹 사이트 엔드포인트 대신 S3 API 엔드포인트를 오리진에 사용하면 오리진 액세스 ID(OAI)를 사용하여 OAI가 CloudFront에서 GET 요청을 수행하도록 허용하는 버킷 정책을 생성할 수 있습니다. 이 실습에서는 S3 버킷이 웹 사이트로 구성된 적이 없었습니다. 단지 Amazon S3 버킷에 대해 GetObject API 요청을 수행할 수 있는 사용자에 관한 버킷 정책을 *Allow Public* 읽기 정책으로 변경했을 뿐입니다.

65. **Origin path**에 대한 항목은 비워 둡니다.

참고: 오리진 경로 필드는 선택 사항이며, 오리진 CloudFront에서 요청을 전달할 디렉터리를 구성합니다. 이 랙에서는 오리진 경로를 구성하는 대신 빈 상태로 두고 요청의 특정 패턴과 일치하는 개체만 반환하도록 동작을 구성합니다.

66. **Name**에는 My Amazon S3 Origin을 입력합니다.

67. **Origin access**는 **Legacy access identities**를 선택합니다.

68. **Create new OAI**를 클릭합니다.

Create new OAI 메시지 박스가 표시됩니다.

69. 이후 단계를 위해 기억할 수 있는 이름을 입력합니다. 이 실습 지침에서는 이 값은 'S3OAI'입니다.

70. **Create**을 클릭합니다.

참고: 이 옵션을 선택하면 새로운 OAI가 생성되고 사용자는 항상 CloudFront URL을 사용하여 Amazon S3 컨텐츠에 액세스해야 합니다. 특별한 CloudFront 사용자(오리진 액세스 ID)를 오리진에 할당합니다. 기존의 모든 QAI는 새로운 OAI를 만드는 대신 오리진에 사용될 수 있다.

OAI(Origin Access Identity)는 사용자가 Amazon S3 URL 대신 CloudFront URL을 통해 컨텐츠에 액세스하도록 요구하는 데 사용하는 가상 ID입니다. 일반적으로 CloudFront 프라이빗 컨텐츠와 함께 사용됩니다.

71. **Bucket policy**의 경우 **No, I will update the bucket policy**가 선택되었는지 확인합니다.

참고: 이후 과제에서는 OAI가 버킷에서 객체를 읽는 것이 허용되는 유일한 보안 주체가 되도록 Amazon S3 버킷을 업데이트합니다. yes를 선택하면 CloudFront는 이 오리진 생성 중에 지정된 OAI를 허용하는 Amazon S3 버킷 정책에 읽기 권한이 포함된 업데이트를 추가합니다. 이 권한은 이미 존재하는 퍼블릭 읽기 권한에 추가됩니다.

72. **Create origin** 버튼을 선택합니다.

Distribution details 페이지가 표시됩니다.

과제 5.2: Amazon S3 오리진의 새 동작 생성

이 과제에서는 오리진에 대한 들어오는 요청을 처리하는 방법에 대한 지침이 배포에 포함되도록 Amazon S3 오리진의 동작을 생성합니다.

73. **Behaviors** 탭을 선택합니다.

74. **Create behavior** 버튼을 선택합니다.

Create behavior 페이지가 표시됩니다.

75. **Path pattern** 필드에 `images/*.png`를 입력합니다.

이 필드는 오리진이 반환할 수 있는 객체 요청의 일치 패턴을 구성합니다. 구체적으로 이 동작에서는 Amazon S3 오리진의 `images` 폴더에 저장된 `.png` 객체만 반환될 수 있습니다. 구성된 동작이 없는 경우 Amazon S3 오리진에 대한 다른 모든 요청은 요청자에게 반환되는 오류를 발생시킵니다. 일반적으로 사용자들은 이런 방법으로 CloudFront 배포 URL에서 직접 객체를 요청하지 않으며, 대신 프런트 엔드 애플리케이션이 사용자에게 반환할 올바른 객체 URL 생성을 처리합니다.

76. **Origin or origin groups** 드롭다운 메뉴에서 **My Amazon S3 Origin**을 선택합니다.

77. **Cache key and origin requests** 섹션에서 **Cache policy and origin request policy(recommended)** 옵션이 선택되어 있는지 확인합니다.

78. **Cache Policy** 드롭다운 메뉴에서 **CachingOptimized** 옵션이 선택되어 있는지 확인합니다.

79. 페이지의 다른 모든 설정은 기본값으로 둡니다.

80. **Create behavior** 버튼을 선택합니다.

Distribution details 페이지가 표시됩니다.

과제 5.3: OAI Canonical ID 가져오기

81. Amazon CloudFront 탐색 메뉴의 **Security** 섹션에서 **Origin access**를 선택합니다. 먼저 **☰** 메뉴 아이콘을 선택하여 탐색 메뉴를 확장해야 할 수도 있습니다.

Origin access 페이지가 표시됩니다.

82. **Identities(legacy)** 탭을 선택합니다.

83. **Amazon S3 Canonical User ID** 열 아래에서 값을 선택합니다. (더블클릭하면 강조 표시됩니다.)

84. 텍스트 편집기에 복사해 놓습니다. 이 값은 이후 과제에서 IAM 정책을 위해 필요합니다.

축하합니다. CloudFront 배포에서 새 오리진, 오리진 액세스 ID, 배포 동작을 생성했습니다.

과제 6: CloudFront 오리진 액세스 ID로 버킷 보호

CloudFront 배포에 오리진으로 LabBucket을 추가하고 새 OAI를 생성했으며 이미지 폴더에 대한 특정 요청을 처리하기 위한 배포 동작을 설정했습니다. 이제 LabBucket 액세스 정책을 편집하여 CloudFront OAI 만 버킷에 대한 액세스가 허용되도록 해야 합니다.

과제 6.1: 버킷 정책에서 허용되는 보안 주체 편집

OAI가 유일한 보안 주체가 되도록 버킷 정책을 설정하면 GetObject API 호출에 대한 퍼블릭 액세스를 허용하지 않고 CloudFront를 통한 버킷 액세스가 허용됩니다.

85. AWS Management Console에서 Services 메뉴에서 **S3**를 선택합니다.

참고: 콘솔 상단의 통합 검색 창에서 s3를 검색할 수도 있습니다.

86. 사용할 수 있는 버킷 목록에서 **LabBucket** 링크를 선택합니다.

87. **Permissions** 탭을 선택합니다.

88. **Bucket policy** 섹션을 찾습니다.

89. **Edit** 버튼을 선택합니다.

Edit bucket policy 페이지가 표시됩니다.

90. JSON 문서 여덟 번째 줄 정도에서 **Principal** 값을 찾은 후에 **Principal**의 Amazon S3 Canonical User ID Placeholder 값을 다음과 같이 변경합니다.

이전에 작업했던 CloudFront 콘솔에서 복사한 **Amazon S3 Canonical User ID** 값으로, **Amazon S3 Canonical User ID Placeholder** 값을 변경합니다. 따옴표는 유지합니다.

```
"Principal": {"CanonicalUser": "Amazon S3 Canonical User ID Placeholder"},
```

이제 LabBucket에서 GetObject 및 GetObjectVersion 작업을 호출하는 것이 허용되는 유일한 보안 주체가 CloudFront 오리진 액세스 ID여야 합니다.

91. **Save Changes** 버튼을 선택합니다.

⚠ 경고: 화면 하단에 오류 메시지가 표시되면 JSON 구문 오류가 원인일 수 있습니다. 다시 한 번 올바른 정책을 복사해 보십시오. 정책은 JSON이 유효할 때까지 저장되지 않습니다. Amazon S3 콘솔에서 오류 메시지를 확장하여 정책 연결에 대한 자세한 내용을 볼 수 있습니다.

canonical ID 값을 사용하면 **Principal**에 대한 OAI의 현재 ARN 값이 대체됩니다.

아래의 완성된 JSON 정책 예를 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Id": "Policy1621958846486",
    "Statement": [
        {
            "Sid": "UpdatedPublicReadPolicy",
```

```

    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity
E1OOWD1GW"
    },
    "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::LabBucket/*"
}
]
}

```

메시지가 다음과 같이 페이지에 표시됩니다.

- Successfully edited bucket policy.

△ 참고: 퍼블릭 액세스에 관한 이전 경고가 이제 사라졌습니다.

과제 6.2: 퍼블릭 액세스 차단기 사용

92. Permissions 탭에서 **Block public access (bucket settings)** 섹션을 찾습니다.

93. **Edit** 버튼을 선택합니다.

Edit Block public access (bucket settings) 페이지가 표시됩니다.

94. **Block all public access** 옆의 확인란을 선택합니다.

95. **Save Changes** 버튼을 선택합니다.

Edit Block public access (bucket settings)라는 제목의 메시지 창이 표시됩니다.

96. 메시지 창의 필드에 **confirm**을 입력합니다.

97. **Confirm** 버튼을 선택합니다.

모든 버킷 세부 정보가 포함된 페이지가 표시됩니다.

축하합니다. 객체 읽기가 허용되는 유일한 보안 주체가 앞서 생성한 OAI가 되도록 Amazon S3 버킷 정책을 편집했습니다.

과제 7: S3 URL을 사용한 버킷 내 파일에 대한 직접 액세스 테스트

이 과제에서는 Amazon S3 URL을 사용하여 여전히 객체에 직접 액세스할 수 있는지 테스트합니다.

98. **Objects** 탭을 선택합니다.

99. **images/** 폴더 링크를 선택합니다.

100. logo.png 객체 링크를 선택합니다.
101. **Object URL** 필드에 있는 링크를 선택합니다.

액세스 거부됨 메시지와 함께 오류 메시지가 표시됩니다. 새 버킷 정책이 Amazon S3 URL에서 객체에 직접 액세스하는 것을 허용하지 않으므로 이는 예상된 것입니다. Amazon S3를 통한 S3 객체에 대한 직접 액세스를 거부하면 사용자들은 더 이상 CloudFront 캐시가 제공하는 제어 장치를 우회할 수 없습니다. 이러한 제어 장치에는 로깅, 동작, 서명된 URL 또는 서명된 쿠키가 포함될 수 있습니다.

축하합니다. S3 URL에서 객체에 더 이상 직접 액세스할 수 없음을 확인했습니다.

과제 8: CloudFront 배포를 사용한 버킷 내 객체에 대한 액세스 테스트

이 과제에서는 CloudFront 배포의 Amazon S3 오리진에 있는 객체에 액세스할 수 있는지 확인합니다.

102. 이 실습 지침 왼쪽의 *LabCloudFrontDistributionDNS* 목록에서 CloudFront 배포의 도메인 DNS 값을 복사합니다.
103. DNS 값을 새 브라우저 탭에 붙여넣습니다.

CloudFront가 콘텐츠를 검색한 웹 서버의 정보가 표시된 간단한 웹 페이지가 로드됩니다.

104. CloudFront 배포의 도메인 DNS 끝에 /images/logo.png를 추가하고 Enter 키를 누릅니다.

브라우저가 CloudFront 배포에 요청을 실행하고, Amazon S3 오리진에서 객체가 반환됩니다.

△ 문제 해결 팁: CloudFront URL이 S3 URL로 리디렉션하거나 객체를 즉시 사용할 수 없는 경우 CloudFront 배포가 최근 변경 내용으로 인한 업데이트 중일 수 있습니다. CloudFront 콘솔로 돌아갑니다. 탐색 메뉴에서 **Distributions**를 선택합니다. **Status** 열이 **Enabled**이고, **Last modified** 열에 타임스탬프가 있는지 확인합니다. 새 오리진과 동작을 테스트하기 전에 이를 기다려야 합니다. 배포의 상태를 확인한 후 몇 분 정도 기다린 다음 이 과제를 다시 한 번 시도합니다.

축하합니다. CloudFront 요청에서 객체가 반환되는 것을 확인했습니다.

선택 과제 9: 여러 AWS 리전에 Amazon S3 버킷 복제

이 선택적 과제는 실습 시간이 남거나 좀 더 고급 내용을 배우기 원하는 경우에 제공됩니다. 이 과제는 꼭 완료해야 하는 것은 아닙니다. 원한다면 실습 종료 단계에 따라 지금 실습을 종료할 수 있습니다. 그렇지 않다면 지침을 계속 읽으십시오.

교차 리전 복제는 버킷의 데이터를 다른 AWS 리전에 있는 다른 버킷에 자동으로 복사할 수 있는 Amazon S3의 기능입니다. 재해 복구에 유용한 기능입니다. 버킷에 교차 리전 복제 기능이 활성화되면 현재 읽기 권한이 있고 소스 버킷에서 생성되는 모든 새 객체가 사용자가 정의하는 대상 버킷에 복제됩니다. 따라서 대상 버킷에 복제되는 객체는 동일한 이름을 갖습니다. Amazon

S3 관리형 암호화 키를 사용하여 암호화되는 객체는 소스 버킷에서와 동일한 방식으로 암호화됩니다.

교차 리전 복제를 수행하려면 소스 및 대상 버킷 모두에서 객체 버전 관리가 활성화되어 있어야 합니다. 버전 관리를 활성화하여 데이터를 잘 정돈된 상태로 유지하기 위해 수명 주기 정책을 배포하여 객체를 자동으로 Amazon Simple Storage Service Glacier에 아카이빙하거나 객체를 삭제할 수 있습니다.

선택 과제 9.1: 소스 버킷에서 버전 관리 활성화

105. AWS Management Console에서 Services 메뉴에서 S3를 선택합니다.

참고: 콘솔 상단의 통합 검색 창에서 s3를 검색할 수도 있습니다.

106. Buckets 섹션에 있는 LabBucket 링크를 선택합니다.

모든 버킷 세부 정보가 포함된 페이지가 표시됩니다.

107. Properties 탭을 선택합니다.

108. Bucket Versioning 섹션을 찾습니다.

109. Edit 버튼을 선택합니다.

Edit Bucket Versioning 페이지가 표시됩니다.

110. Bucket Versioning에서 ● Enable 옵션을 선택합니다.

111. Save Changes 버튼을 선택합니다.

선택 과제 9.2: 교차 리전 복제의 대상 버킷 생성

112. Amazon S3 탐색 메뉴에서 Buckets를 선택합니다.

113. Create bucket 버튼을 선택합니다.

Create bucket 페이지가 표시됩니다.

114. Bucket name 필드에 고유한 버킷 이름을 입력합니다.

이 지침 왼쪽에는 실습 환경이 지원하는 기본 리전과 보조 리전의 값이 있습니다. 실습은 처음에 기본 리전에서 시작됩니다.

115. 이 실습 지침 왼쪽에 있는 SecondaryRegion 값과 일치하는 AWS 리전을 선택합니다.

116. Block Public Access settings for this bucket 섹션에서 Block all public access 옵션의 선택을 취소합니다.

117. 경고 메시지에서 I acknowledge that the current settings might result in this bucket and the objects within becoming public 옵션을 선택합니다.

교차 리전 복제 기능을 사용하기 위해 프라이빗 버킷에 퍼블릭 액세스를 활성화할 필요는 없습니다. 이 실습에서는 S3 URL에서 객체를 검색할 수 있는지 빠르게 테스트할 수 있도록 활성화되어 있습니다.

118. ● Bucket Versioning에서 Enable 옵션을 선택합니다.

119. Create bucket 버튼을 선택합니다.

Amazon S3 콘솔이 표시됩니다.

계정의 모든 버킷 목록 중에 새로 생성된 버킷이 표시됩니다.

△ 참고: 이 실습의 설명을 간단히 하기 위해 나머지 지침에서는 새로 생성된 이 버킷을 DestinationBucket이라고 하겠습니다.

선택 과제 9.3: 새 대상 버킷의 퍼블릭 읽기 정책 구성

이제 이 버킷의 퍼블릭 객체 읽기 정책을 생성합니다.

120. S3 탐색 메뉴 중, Buckets를 선택합니다.

121. 버킷 목록에서 DestinationBucket을 선택합니다.

122. Permissions 탭을 선택합니다.

123. Bucket policy 섹션을 찾습니다.

124. Edit 버튼을 선택합니다.

Edit bucket policy 페이지가 표시됩니다.

125. 나중에 이 정보를 사용하기 위해 Bucket ARN 값을 복사하여 텍스트 편집기에 붙여넣으세요. Policy 상자 위에 있는 arn:aws:s3::LabBucket과 같은 문자열 값입니다.

ARN 값은 이 Amazon S3 버킷을 고유하게 식별합니다. 버킷 기반 정책을 생성할 때 이 특정 ARN 값이 필요합니다.

126. 아래의 JSON 문장을 텍스트 에디터에 복사, 붙여넣기 합니다.

```
{
    "Version": "2012-10-17",
    "Id": "Policy1621958846486",
    "Statement": [
        {
            "Sid": "OriginalPublicReadPolicy",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
        }
    ]
}
```

```

        "Resource": "RESOURCE_ARN"
    }
}

```

127. JSON 문장에서 **RESOURCE_ARN** 값을 이전 작업에서 복사해 둔 **Bucket ARN** 값으로 대체합니다. 그리고 **Bucket ARN** 값 마지막에 /*을 추가합니다.

다음은 생성된 JSON 정책의 예입니다.

```

{
    "Version": "2012-10-17",
    "Id": "Policy1621958846486",
    "Statement": [
        {
            "Sid": "OriginalPublicReadPolicy",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::DestinationBucket/*"
        }
    ]
}

```

128. Amazon S3 콘솔로 돌아갑니다.

129. 완성된 JSON을 *Policy* 상자에 붙여넣습니다.

130. **Save Changes** 버튼을 선택합니다.

버킷 세부 정보 페이지가 표시됩니다.

131. 버킷 이름 아래에 빨간색 Public accessible 경고가 있습니다. *Permissions overview* 섹션에도 △ Public 경고가 있습니다.

이러한 경고는 버킷에 현재 적용되는 정책으로 인해 이 버킷의 객체에 퍼블릭 액세스가 가능하다는 것을 알리기 위한 것입니다. 실습 이후 단계에서 CloudFront 배포에서만 버킷에 액세스할 수 있도록 구성합니다.

선택 과제 9.4: 복제 규칙 생성

132. Amazon S3 탐색 메뉴에서 **Buckets**를 선택합니다.
133. **Buckets** 섹션에 있는 **LabBucket** 링크를 선택합니다.
134. **Management** 탭을 선택합니다.
135. **Replication rules** 섹션을 찾습니다.
136. **Create replication rule** 버튼을 선택합니다.

Create replication rule 페이지가 표시됩니다.

137. **Replication rule name** 필드에 **MyCrossRegionReplication**을 입력합니다.
138. **Source bucket name**에 **LabBucket**이 설정되어 있는지 확인합니다. 설정되어 있지 않다면 복제 규칙을 선택하기 전에 잘못된 버킷을 선택한 것입니다.
139. **Choose a rule scope** 섹션에서 **Apply to all objects in the bucket**을 선택합니다.
140. **Destination** 섹션을 찾습니다.
141. **Browse S3** 버튼을 선택합니다.
142. **DestinationBucket**을 선택합니다.
143. **Choose path** 버튼을 선택합니다.
144. **IAM Role** 섹션을 찾습니다.
145. 메뉴에서 **Create new role**을 선택합니다.
146. 다른 모든 옵션은 기본값이 선택된 상태로 둡니다.
147. **Save** 버튼을 선택합니다.
148. 만약 **Replicate existing objects** 항이 표시된다면, **No, do not replicate existing objects** 선택한 후 **Submit** 버튼을 선택합니다.

LabBucket에 대한 **Replication rules** 페이지가 표시됩니다.

- Replication configuration successfully updated.
Press the refresh button if changes to the configuration are not displayed

LabBucket에서 새로 생성되는 모든 객체는 **DestinationBucket**에 복제됩니다.

참고: 기존 객체의 버킷 간 복제도 가능하지만 이 실습의 범위를 벗어납니다. 이에 대한 자세한 내용은 부록 섹션의 문서 링크에서 찾을 수 있습니다.

선택 과제 9.5: 객체 복제 확인

149. Amazon S3 탐색 메뉴에서 **Buckets**를 선택합니다. ⌂ 메뉴 아이콘을 선택하여 메뉴를 확장해야 할 수도 있습니다.
150. **Buckets** 섹션에 있는 **LabBucket** 링크를 선택합니다.
151. 링크 [logo2.png](#)를 마우스 오른쪽 버튼으로 클릭하고 로컬 디바이스에 저장을 선택하여 이 실습 지침을 위한 객체를 다운로드합니다.
152. **Amazon S3** 콘솔로 돌아갑니다.
153. **images/** 폴더 링크를 선택합니다.

참고: **images** 폴더를 찾을 수 없는 경우 콘솔 왼쪽에 있는 탐색 메뉴에서 **Buckets**를 선택합니다. 그런 다음 목록에서 **LabBucket** 링크를 선택합니다. 마지막으로 **Objects** 탭을 선택하여 올바른 페이지에 있는지 확인합니다.

154. **Upload** 버튼을 선택합니다.

Upload 페이지가 표시됩니다.

155. **Add files** 버튼을 선택합니다.

156. 로컬 스토리지 위치에서 **logo2.png** 객체를 선택합니다.

157. **Upload** 버튼을 선택합니다.

Upload: status 페이지가 표시됩니다.

파일이 LabBucket에 업로드되면 다음 메시지와 함께 페이지에 메시지가 표시됩니다.

- Upload succeeded.

158. **Files and folders** 섹션에서 logo2.png의 링크를 선택합니다.

Amazon S3 객체에 대한 세부 정보가 있는 페이지가 표시됩니다.

159. **Object management overview** 섹션에서 *Replication Status*를 검사하고, 상태가 *PENDING*에서 *COMPLETED*로 바뀔 때까지 페이지를 주기적으로 새로 고칩니다.

160. Amazon S3 탐색 메뉴에서 **Buckets**를 선택합니다.

161. **Buckets** 섹션에 있는 **DestinationBucket** 링크를 선택합니다.

모든 버킷 세부 정보가 포함된 페이지가 표시됩니다.

162. **images/** 폴더 링크를 선택합니다.

163. **Files and folders** 섹션에서 logo2.png의 링크를 선택합니다.

Amazon S3 객체에 대한 세부 정보가 있는 페이지가 표시됩니다.

164. **Object management overview** 섹션에서 *Replication Status*가 **REPLICA**로 표시되는지 검사합니다.

165. **Object URL** 필드에 있는 링크를 선택합니다.

브라우저 탭에 그림이 표시됩니다.

축하합니다. LabBucket에 업로드되는 모든 새 객체의 교차 리전 복제 설정을 완료했습니다.

다음과 같은 이 선택적 과제의 후속 질문을 생각해 보십시오.

- *DestinationBucket*의 객체에 대한 액세스를 제한하려면 어떻게 해야 합니까?
- CloudFront 배포에 *DestinationBucket*을 추가하려면 어떤 단계가 필요합니까?

결론

이 실습에서는 다음을 수행했습니다.

- S3 버킷을 생성했습니다.
- 기본 S3 버킷 퍼블릭 액세스 차단기를 제거했습니다.
- 퍼블릭 액세스를 허용하는 버킷 정책을 생성했습니다.
- 기존 CloudFront 배포에서 Amazon S3 버킷의 새 오리진을 생성했습니다.
- 새 오리진 액세스 ID를 생성했습니다.
- 버킷에서 오리진 액세스 ID만 GetObject 작업을 수행할 수 있도록 기존 Amazon S3 버킷 정책을 수정했습니다.

실습 완료

▣ 축하합니다! 실습을 마치셨습니다.

추가 리소스

- Amazon S3 버킷 명명 규칙에 대해 자세히 알아보기
<https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html>
- CloudFront를 통한 프라이빗 콘텐츠 제공에 대해 자세히 알아보기
<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/PrivateContent.html>
- Amazon S3 버킷을 OAI로 제한에 대해 자세히 알아보기
<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-restricting-access-to-s3.html#private-content-granting-permissions-to-oai>
- 기존 객체의 버킷 간 복제에 대해 자세히 알아보기
<https://aws.amazon.com/blogs/storage/replicating-existing-objects-between-s3-buckets/>

실습 종료

다음 단계를 따라 콘솔을 닫고 실습을 종료한 후 실습 경험을 평가해 주십시오.

166. AWS Management Console로 돌아갑니다.
167. 탐색 모음에서 `awsstudent@<AccountNumber>`를 선택한 다음 **Sign Out**을 선택합니다.
168. 실습 종료를 선택합니다.
169. OK를 선택합니다.
170. (선택 사항):
 - 해당하는 별 개수를 선택합니다.
 - 의견을 입력합니다.
 - Submit**을 선택합니다.
 - 별 1개 = 매우 불만족
 - 별 2개 = 불만족
 - 별 3개 = 보통
 - 별 4개 = 만족
 - 별 5개 = 매우 만족

피드백을 제공하지 않으려면 그냥 창을 닫으면 됩니다.

AWS Training and Certification에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다.

피드백, 제안 사항 또는 수정 요청 사항을 제공하려면 AWS Training and Certification 문의 양식에 세부 정보를 입력해 주시기 바랍니다.



실습 7: 캡스톤 실습

© 2022 Amazon Web Services, Inc. 또는 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 임대 또는 판매는 금지됩니다. 모든 상표는 해당 소유자의 자산입니다.

참고: 개인 정보, 개인을 식별할 수 있는 정보 또는 기밀 정보는 실습 환경에 입력하지 마십시오. 입력한 정보가 다른 사용자에게 공개될 수도 있습니다.

수정 사항이나 피드백 또는 기타 질문이 있으십니까? AWS Training and Certification에서 문의해 주십시오.

실습 개요

새로운 지식을 적용하여 특정 비즈니스 사례 내에서 몇 가지 아키텍처 문제를 해결하는 작업을 맡았습니다. 먼저 설계 관련 요구 사항 목록이 제공됩니다. 그런 다음, 요구 사항을 충족하는데 필요한 서비스를 배포하고 구성하는 일련의 작업을 수행해야 합니다.

이 캡스톤 실습을 진행하면서 다음과 같은 항목을 사용할 수 있습니다.

- 다운로드 가능 CloudFormation 템플릿 및 기타 실습 파일
- 작업 시나리오, 설명 및 요구 사항
- 단계별 지침(필요한 경우 제공되는 선택 사항)

관련 배경 정보가 제공되는 작업 시나리오를 통해 요구 사항을 충족하여 기업의 실제 문제를 해결하는 방법을 파악할 수 있습니다. 템플릿과 요구 사항 목록을 사용하여 캡스톤의 모든 작업을 완료할 수 있습니다. 교육 과정에서 관련 개념과 서비스를 파악했으므로 이 실습에서 연습을 진행하여 학습 내용을 더욱 명확하게 숙지할 수 있습니다. 실제로는 잘 정의되어 있지 않거나 순서가 맞지 않는 문제에 직면하게 됩니다. 이 캡스톤이 끝날 무렵이면 실제 문제에 지식을 어떻게 적용할 수 있는지 더 잘 이해하게 될 것입니다.

목표

이 실습을 마치면 다음을 수행할 수 있습니다.

- 제공된 CloudFormation 템플릿을 사용하여 특정 리전의 여러 가용 영역에 분사나되는 가상 네트워크 배포
- Amazon Relational Database Service(Amazon RDS)를 사용하여 해당 가용 영역에 고가용성 완전관리형 관계형 데이터베이스 배포

- Amazon Elastic File System(Amazon EFS)을 사용하여 애플리케이션 티어용으로 여러 가용 영역에 공유 스토리지 계층 프로비저닝(Network File System(NFS)을 통해 지원됨)
- 로드 변화에 대응하여 자동으로 크기가 조정되는 웹 서버 그룹을 생성하여 애플리케이션 티어 완성

선행 조건

이 실습을 진행하려면 다음 항목이 필요합니다.

- Microsoft Windows, macOS 또는 Linux(Ubuntu, SuSE, Red Hat)가 실행되는 Wi-Fi 지원 노트북
- 인터넷 브라우저(예: Chrome, Firefox 또는 Microsoft Edge)
- 일반 텍스트 편집기

소요 시간

이 실습을 완료하려면 약 90분이 소요됩니다.

이 실습에서 사용되지 않는 AWS 서비스

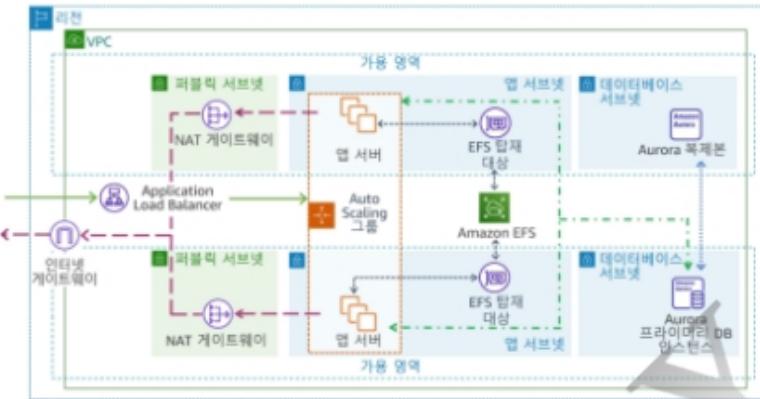
이 실습에서 사용하지 않는 AWS 서비스는 실습 환경에서 사용 중지 상태로 설정되어 있습니다. 또한 이 실습에 사용되는 서비스의 기능은 실습에 필요한 것으로 제한됩니다. 다른 서비스에 액세스하거나 이 실습 가이드에서 제공하는 것 외의 작업을 수행하는 경우 오류가 발생할 수 있습니다.

실습 시나리오

Example Corp.는 중소기업을 위한 마케팅 캠페인 제작사입니다. 최근 Example Corp.에 입사한 여러분은 엔지니어링 팀과 협력하여 사업 개념 증명을 작성해야 합니다. 현재 온프레미스 데이터 센터를 사용하여 클라이언트를 호스트하는 Example Corp.는 비용을 절약하고 클라우드 우선 방식을 통해 업무 방식을 획기적으로 개선하기 위해 운영 환경을 클라우드로 이전하기로 결정했습니다. 클라우드를 사용해 본 팀원 몇 명이 솔루션 구축용 서비스로 AWS Cloud Services를 추천했습니다.

그리고 Example Corp.는 웹 포털도 다시 설계하기로 했습니다. 고객은 포털을 사용하여 계정에 액세스하고 마케팅 계획을 만들고 마케팅 캠페인에 대한 데이터 분석을 실행하며, 2주 안에 작동하는 프로토타입을 만들려고 합니다. 이러한 애플리케이션을 지원하는 아키텍처를 설계해야 합니다. 솔루션은 빠르고 내구성이 뛰어나고 확장 가능하며 기존 온프레미스 인프라보다 비용 효율적이어야 합니다.

다음 이미지에는 설계된 솔루션의 최종 아키텍처가 나와 있습니다.



실습 시작

1. 실습을 시작하려면 페이지 상단에서 실습 시작을 선택합니다.

그러면 실습 리소스를 프로비저닝하는 프로세스가 시작됩니다. 실습 리소스를 프로비저닝하는 데 걸리는 예상 시간이 표시됩니다. 계속 진행하기 전에 사용할 리소스가 프로비저닝될 때까지 기다려야 합니다.

① 토큰을 입력하라는 메시지가 표시되면 여러분에게 배포된 토큰(또는 구매한 크레딧)을 사용하세요.

2. 실습을 열려면 콘솔 열기를 선택합니다.

새 웹 브라우저 탭에서 **AWS Management Console** 로그인 페이지가 열립니다.

3. **Sign in as IAM user** 페이지에서

- **IAM user name**에 `awsstudent`를 입력합니다.
- **Password**에 이 지침의 왼쪽에 나열된 **Password** 값을 복사하여 붙여넣습니다.
- **Sign in**을 선택합니다.

△ 별다른 지시가 없는 한 리전을 변경하지 마십시오.

일반적인 로그인 오류

오류: 우선 로그 아웃 필요

Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

You must first log out before logging into a different AWS account라는 메시지가 표시된다면 다음을 수행합니다.

- [click here](#)의 링크를 선택합니다.
- **Amazon Web Services Sign In** 웹 브라우저 탭을 닫고 초기 실습 페이지로 돌아갑니다.
- 콘솔 열기를 다시 선택합니다.

경우에 따라 일부 팝업 또는 스크립트 차단 웹 브라우저 확장 프로그램 때문에 **실습 시작** 버튼이 제대로 작동하지 않을 수 있습니다. 실습을 시작하는 데 문제가 있는 경우 다음을 수행합니다.

- 팝업 또는 스크립트 차단 프로그램의 허용 목록에 실습 도메인 이름을 추가하거나 차단 프로그램을 끕니다.
- 페이지를 새로 고친 후 다시 시도하십시오.

참고: 솔루션을 구축할 때는 2개 섹션을 참조할 수 있습니다. 섹션 1에는 요구 사항과 구성만 나와 있고, 섹션 2에서는 세부 지침을 제공합니다. 섹션 2를 참조할 수도 있지만 섹션 1을 먼저 참조한 후 단계별 지침을 확인하지 않으면 작업을 진행할 수 없을 때만 섹션 2를 참조하는 것이 좋습니다.

섹션 1: 개략적 지침

이 섹션에서는 솔루션을 구축하려면 수행해야 하는 모든 작업의 요구 사항과 구성은 제시합니다. 이 섹션만 참조하여 솔루션을 구축해 보시기 바랍니다. 작업을 진행하기가 어려울 때는 언제든지 섹션 2에서 세부 지침을 참조할 수 있습니다.

과제 1: 미리 구성된 CloudFormation 템플릿 검토 및 실행

이 작업에서는 워드프레스 애플리케이션 스택의 가용성과 확장성이 뛰어난 배포를 설계 및 배포합니다. 그리고 CloudFormation 템플릿을 사용하여 해당 애플리케이션을 지원하는데 필요한 네트워킹 리소스를 배포합니다.

과제 1: 시나리오

회사는 기존 웹 호스팅 계정의 워드프레스에 새 웹 애플리케이션을 배포하려고 합니다. 첫 번째 단계는 아키텍처 설계에 대한 요구 사항을 수집하는 것입니다. 시작하는데 도움을 주기 위해 네트워크 팀은 기존 환경에 문제가 없도록 하기 위한 요청 목록을 제공했습니다. 여기에는 Amazon Virtual Private Cloud(Amazon VPC)용 Classless Inter-Domain Routing(CIDR) 범위가 포함됩니다. 이 주소 범위를 사용하여 퍼블릭 서브넷, 프라이빗 애플리케이션 서브넷, 프라이빗 데이터베이스 서브넷이 각각 2개씩 포함된 VPC를 구축합니다. 트래픽 라우팅을 위한 NAT 게이트웨이가 있는 VPC에 인터넷 게이트웨이를 연결해야 합니다. 네트워크 팀은 탄력적 IP 주소 2개도 요청했습니다. 다음 항목을 연결해야 합니다.

- 퍼블릭 서브넷과 인터넷 게이트웨이 연결
 - 프라이빗 애플리케이션 서브넷과 프라이빗 데이터베이스 서브넷을 NAT 게이트웨이와 연결
- 네트워크 팀은 서브넷 연결에 필요한 라우팅 테이블도 제공했습니다.

클라우드 엔지니어가 CloudFormation 템플릿으로 요청을 전송했으며 이후 배포에 필요한 보안 그룹 및 출력을 설정했습니다. 클라우드 엔지니어와 함께 CloudFormation 템플릿을 검토하고 템플릿을 배포한 다음, 네트워크 팀과 다시 확인하여 빌드가 네트워크 팀의 모든 요구 사항을 충족하는지 검사하십시오.



과제 1: 하위 작업

- 템플릿 파일을 다운로드한 다음 검토하여 배포 중인 인프라를 파악합니다.
- 템플릿 파일을 업로드하여 CloudFormation 스택을 생성합니다.
- 생성된 리소스를 콘솔에서 확인합니다.

과제 1: 요구 사항 및 구성

- 이 [Task1.yaml](#) 링크에서 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭 메뉴)를 열고 컴퓨터에 CloudFormation 템플릿을 저장하는 옵션을 선택합니다.
- 다운로드한 파일을 워드 프로세서 이외의 텍스트 편집기에서 엽니다.
- CloudFormation에서 템플릿 파일을 업로드합니다.
- 이 작업에서는 기본값을 사용합니다.

참고: 나중에 사용 가능하도록 이 CloudFormation 템플릿을 저장할 수 있습니다. 나중에 본인의 AWS 계정에서 템플릿을 작성하려는 경우에는 요구 사항에 맞게 구성을 조정하면 됩니다.

과제 1: AWS 관리 콘솔로 이동

첫 번째 작업 요구 사항이 충족되었으므로 구축을 진행합니다. 작업 진행이 어려우면 섹션 2의 지침을 참조할 수 있지만, 일단은 섹션 2의 지침을 참조하지 않고 구축을 진행해 보는 것이 좋습니다.

축하합니다. 스택 구성 방법을 알아보았으며 제공된 CloudFormation 템플릿을 사용하여 모든 리소스를 생성했습니다.

과제 2: Amazon RDS 데이터베이스 생성

이 작업에서는 Amazon Aurora DB 인스턴스와 해당 인스턴스를 지원하는 서브넷 그룹을 생성합니다.

과제 2: 시나리오

네트워크 팀에서 이전 배포 요구 사항을 확인했고 모든 것이 괜찮아 보입니다. 프로젝트의 다음 단계에서는 보안 백엔드 데이터베이스를 설정하여 서비스 오픈 전 마이그레이션 작업을 위한 데이터베이스 관리자(DBA) 액세스를 부여합니다.

현재 데이터베이스에는 관리 오버헤드가 많이 필요하기 때문에 관리형 데이터베이스 서비스로 이동하는 데 회사가 동의했습니다. 회사에서 아키텍처의 가용성이 높아지기를 원하기 때문에 Multi-AZ RDS Aurora 데이터베이스 계층을 설정하도록 추천했습니다. 제안된 설계를 검토한 후 DBA는 데이터베이스 요구 사항의 요점을 언급했습니다. 이전 성능 문제로 인해 데이터베이스에 암호화가 필요하지 않으며 이것이 최선의 선택이라는 데 동의했습니다.

기존 모니터링 솔루션은 10분마다 데이터를 폴링합니다. 엔지니어링 팀에 추가 기능을 위한 예산적 여유가 없기 때문에 향상된 모니터링 기능은 필요하지 않습니다.



과제 2: 하위 작업

- 새 DB 서브넷 그룹을 생성합니다.
- 새 Aurora 데이터베이스를 생성합니다.
- 데이터베이스 메타데이터를 복사합니다.

과제 2: 요구 사항 및 구성

- 첫 단계에서는 RDS 콘솔에서 DB 서브넷 그룹을 생성합니다.

- 가용 영역 A와 가용 영역 B를 선택합니다.
- 가용 영역 A의 DB 서브넷으로는 CIDR 블록이 10.0.4.0/24인 서브넷을 선택합니다.
- 가용 영역 B의 DB 서브넷으로는 CIDR 블록이 10.0.5.0/24인 서브넷을 선택합니다.
- 데이터베이스 인스턴스는 **Amazon Aurora with MySQL compatible burstable-performance DB** 인스턴스여야 합니다.
- 데이터베이스 인스턴스 크기는 **db.t3.small**이어야 합니다.
- 데이터베이스 계층은 **LabVPC**에 배포해야 합니다.
- 서브넷 그룹은 **Database subnets**만 포함하도록 구성해야 합니다.
- 데이터베이스는 보안 팀이 설정한 **RDS Security Group**만 포함하도록 구성해야 합니다.
- **WPDatabase**와 같은 **Initial database name**을 제공해야 합니다.
- **MyDBCluster**와 같은 **DB Cluster identifier**를 제공해야 합니다.
- **Encryption**을 비활성화합니다.
- **Enhanced monitoring**을 비활성화합니다.
- **Enable auto minor version upgrade**를 비활성화합니다.
- **Enable deletion protection**을 비활성화합니다.

참고: 다음 정보를 적어 둡니다. 뒷부분에서 해당 정보가 필요합니다.

- **Master username** 및 **Master password**
- **Initial database name**
- **Writer endpoint**

참고 "Failed to turn on DevOps Guru for mydbcluster-instance-x because of missing permissions." 오류가 표시되면 무시해도 됩니다.

과제 2: AWS 관리 콘솔로 이동

요구 사항이 충족되었으므로 구축을 진행합니다. 작업 진행이 어려우면 섹션 2의 지침을 참조할 수 있지만, 일단은 섹션 2의 지침을 참조하지 않고 구축을 진행해 보는 것이 좋습니다.

축하합니다. Amazon Aurora 데이터베이스를 생성했습니다.

과제 3: Amazon EFS 파일 시스템 생성

이 작업에서는 사용자 정의 Amazon EFS 파일 시스템을 생성하고 구성합니다.

과제 3: 시나리오

Example Corp.에서는 새 하드웨어를 주문하는 데 소요되는 리드 타임 관련 문제가 발생하고 있습니다. 이로 인해 신규 고객 유치 및 사업 확장을 신속하게 진행하지 못하고 있습니다. SysOps 팀에서 클라우드용으로 구축된 스토리지 솔루션을 요청했습니다. 이 팀은 백업 정책 및 암호화 설정이 내부 요구 사항 및 규정 준수 요구 사항을 충족하는지 확인할 수 있어야 합니다. Example Corp.는 시간, 비용 및 규정 준수를 적절하게 관리함으로써 경쟁력을 높일 수 있을 것으로 예상됩니다.

여러분은 이 회사에 Amazon EFS를 추천했습니다. Amazon EFS는 한 번만 설정하면 되는 간단하고 탄력적인 서비스입니다. Amazon EFS를 사용하면 스토리지를 프로비저닝하거나 관리하지 않고도 데이터를 안전하게 공유할 수 있습니다.

여러분은 SysOps 팀의 요구 사항을 충족하는 Amazon EFS 파일 시스템을 생성해야 합니다.



과제 3: 하위 작업

- 새 EFS 파일 시스템을 생성합니다.
- EFS 파일 시스템 메타데이터를 복사합니다.

과제 3: 요구 사항 및 구성

Amazon EFS 파일 시스템 생성 과정을 사용자 정의합니다.

- 특정 리전에서 해당 시스템의 가용성과 내구성이 보장되어야 합니다.

- 그리고 비용을 제어할 수 있도록 **범용** 수준 성능을 제공해야 합니다.
- 또한 시스템 크기 조정을 위한 처리량 **버스팅** 기능도 있어야 합니다.
- Automatic backups** 및 저장 데이터 **Encryption**을 비활성화합니다.
- EFS는 **Lab VPC**에 배포해야 합니다.
- Availability Zone 2a**와 **Availability Zone 2b**를 선택합니다.
- 서브넷 ID로 **AppSubnet1** 및 **AppSubnet2**를 배정합니다.
- 기본 보안 그룹을 제거하고 **EFSMountTargetSecurityGroup**을 선택합니다.
- 지금은 파일 시스템 정책을 생성하지 않아도 됩니다.

참고: **File system ID**를 적어둡니다. 뒷부분에서 해당 ID가 필요합니다.

과제 3: AWS 관리 콘솔로 이동

요구 사항이 충족되었으므로 구축을 진행합니다. 작업 진행이 어려우면 섹션 2의 지침을 참조할 수 있지만, 일단은 섹션 2의 지침을 참조하지 않고 구축을 진행해 보는 것이 좋습니다.
축하합니다. Amazon EFS 파일 시스템을 생성했습니다.

과제 4: Application Load Balancer 생성

이 작업에서는 Application Load Balancer와 대상 그룹을 생성합니다.

과제 4: 시나리오

SysOps 팀은 새로운 EFS 구성에 큰 기대를 하고 있으며, 해당 구성을 통해 또 다른 문제점을 해결하고자 합니다. 현재 애플리케이션은 가변적이고 예상치 못한 트래픽 로드로 인해 자주 중단됩니다. SysOps 팀은 이 문제를 해결할 수 있는 서비스, 즉 애플리케이션 계층(계층 7)에서 사용 가능한 서비스가 AWS에서 제공되는지를 파악하고자 합니다. 추가 조사를 진행한 결과, 퍼블릭 서브넷의 웹 연결 리소스용 Application Load Balancer도 필요함이 확인되었습니다. 상태 확인을 사용하여 Application Load Balancer를 배포 및 구성하고 필요한 대상을 등록하십시오.



과제 4: 하위 작업

- Application Load Balancer를 생성합니다.
- Application Load Balancer 메타데이터를 복사합니다.

과제 4: 요구 사항 및 구성

LabVPC를 사용하여 다음 속성으로 인스턴스 부하를 분산하도록 Application Load Balancer 대상 그룹을 구성해야 합니다.

- Health check path:** /wp-login.php
- Advanced health check settings:**
 - Healthy threshold:** 2
 - Unhealthy threshold:** 10

- **Timeout:** 50
- **Interval:** 60
- 다른 모든 설정은 기본값으로 유지합니다.

다음 속성으로 LabVPC 및 퍼블릭 서브넷에서 **Application Load Balancer**를 구성해야 합니다.

- **Scheme:** internet-facing
- **Subnets:** 모든 퍼블릭 서브넷을 선택합니다.
- **SecurityGroups:** AppInstanceSecurityGroup을 선택합니다.
- **Listener:** 생성한 대상 그룹으로 전달합니다.

참고: Load Balancer의 **DNS name**을 적어둡니다. 뒷부분에서 해당 이름이 필요합니다.

과제 4: AWS 관리 콘솔로 이동

요구 사항이 충족되었으므로 구축을 진행합니다. 작업 진행이 어려우면 섹션 2의 지침을 참조할 수 있지만, 일단은 섹션 2의 지침을 참조하지 않고 구축을 진행해 보는 것이 좋습니다.

축하합니다. 대상 그룹과 Application Load Balancer를 생성했습니다.

과제 5: CloudFormation을 사용하여 시작 템플릿 생성

이 작업에서는 CloudFormation 템플릿을 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) Auto Scaling 시작 템플릿 내에 워드프레스 사용자 데이터를 배포합니다. 이 템플릿에는 EFS 탑재 지점과 Aurora 구성이 포함됩니다.

과제 5: 시나리오

지금까지 기본 네트워크 리소스, 데이터베이스, Amazon EFS 파일 시스템 및 Application Load Balancer를 생성했습니다. 이제 이러한 구성 요소를 모두 결합하여 아키텍처를 구축해 보겠습니다. Example Corp.에는 워드프레스 계정이 이미 있으므로 클라우드 엔지니어는 기존 환경의 설정 및構성을 사용하여 CloudFormation 템플릿을 생성합니다. 여기에는 시작 템플릿을 설정하기 위해 프로비저닝한 새 리소스가 모두 포함됩니다.

CloudFormation 템플릿을 검토하고 필요한 모든 리소스를 생성하십시오. 특히 사용자 데이터 스크립트를 자세히 살펴보시기 바랍니다. 배포에 오류가 있는지 확인하고 필요한 경우 문제를 해결합니다. 완료되면 새 환경에 대한 트래픽을 지원하기 위해 애플리케이션 서버를 생성할 수 있습니다.

과제 5: 하위 작업

- CloudFormation 템플릿을 다운로드하여 검토합니다.
- 템플릿을 업로드하여 CloudFormation 스택을 생성합니다.
- 생성된 리소스를 콘솔에서 확인합니다.
- 생성된 시작 템플릿을 확인합니다.

과제 5: 요구 사항 및 구성

- 이 [Task5.yaml](#) 링크에서 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭 메뉴)를 열고 컴퓨터에 CloudFormation 템플릿을 저장하는 옵션을 선택합니다.

참고: 나중에 사용 가능하도록 이 CloudFormation 템플릿을 저장할 수 있습니다. 나중에 본인의 AWS 계정에서 템플릿을 작성하려는 경우에는 요구 사항에 맞게 구성을 조정하면 됩니다.

- 다운로드한 파일을 워드 프로세서 이외의 텍스트 편집기에서 엽니다.
- CloudFormation에서 템플릿 파일을 업로드합니다.
- 이전 과제에서 저장한 값을 사용하여 스택 생성 시 다음 파라미터 값을 업데이트합니다.
 - DB Name:** 과제 2에서 적어 둔 *initial database name*을 붙여넣습니다.
참고: 클러스터 이름이 아니라, *initial database name*을 붙여넣습니다.
 - Database endpoint:** 과제 2에서 적어 둔 *Writer* 앤드포인트를 붙여넣습니다.

- **Database User Name:** 과제 2에서 적어 둔 **마스터 사용자 이름**을 붙여넣습니다.
- **Database Password:** 과제 2에서 적어 둔 **마스터 암호**를 붙여넣습니다.
- **WordPress admin username:** wpadmin으로 기본 설정됩니다.
- **WordPress admin password:** wpadmin123으로 기본 설정됩니다.
- **WordPress admin email address:** 유효한 이메일 주소를 입력합니다.
- **Instance Type:** t3.medium으로 기본 설정됩니다.
- **ALBDnsName:** 과제 4에서 적어 둔 **DNS name** 값을 붙여넣습니다.
- **LatestAL2Amild:** 기본값을 그대로 둡니다.
- **WPElasticFileSystemID:** 과제 3에서 적어 둔 **File System ID** 값을 붙여넣습니다.

과제 5: AWS 관리 콘솔로 이동

요구 사항이 충족되었으므로 프로젝트의 다음 단계에서 구축을 진행합니다. 작업 진행이 어려우면 섹션 2의 지침을 참조할 수 있지만, 일단은 섹션 2의 지침을 참조하지 않고 구축을 진행해 보는 것이 좋습니다.

축하합니다. 제공된 CloudFormation 템플릿을 사용하여 스택을 생성했습니다.

과제 6: Auto Scaling 그룹 및 크기 조정 정책을 구성하여 애플리케이션 서버 생성

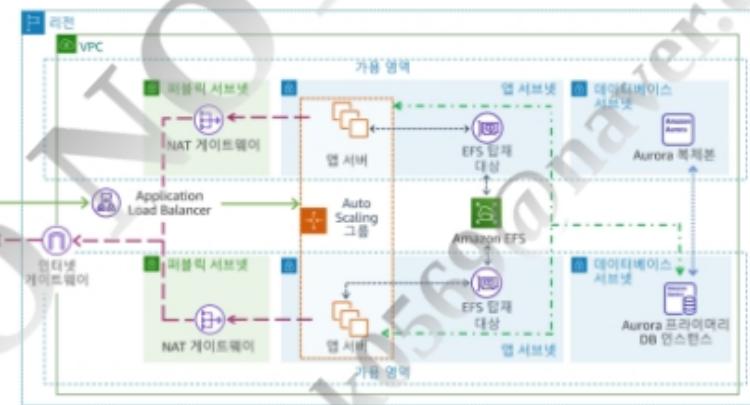
이 작업에서는 Auto Scaling 그룹 및 크기 조정 정책을 구성하여 워드프레스 애플리케이션 서버를 생성합니다.

과제 6: 시나리오

템플릿이 배포되었으므로 이제 애플리케이션 서버와 자동 크기 조정 메커니즘을 생성합니다. 이전 작업에서는 프로젝트 계획의 크기 조정 요구 사항을 충족하기 위해 앱 서버용 자동 크기 조정을 구현하도록 선택했습니다.

Auto Scaling 그룹 및 크기 조정 정책을 생성합니다. 인스턴스 상태가 정상인지 확인하고 로드 밸런서 가용성을 테스트합니다. 단위 테스트가 완료되면 환경을 엔지니어링 팀에 넘겨 전체 기능을 검증하고 피드백을 요청합니다.

엔지니어링 팀에서 환경이 적절함을 통지하면 워드프레스 사이트를 AWS 환경으로 마이그레이션하고 앱 기능을 테스트하도록 요청합니다. 일반적으로는 장애의 예를 앱에 적용해 보는 방식으로 테스트를 진행합니다. 앱이 정상 작동하는 경우 장애의 몇 가지 예를 적용해 봅니다. 가령 앱 서버를 삭제하거나 최근 백업으로 데이터베이스를 롤백합니다. 이 실습에서는 해당 단계를 수행하지 않습니다.



과제 6: 하위 작업

- Auto Scaling 그룹을 생성합니다.
- 대상 그룹을 확인하고 로드 밸런서를 통해 애플리케이션을 테스트합니다.
- 워드프레스에 로그인하여 기능을 살펴봅니다.

과제 6: 요구 사항 및 구성

- 이전 작업에서 생성한 시작 템플릿을 사용합니다.
- 네트워크 설정에는 **LabVPC**를 사용합니다.
- 두 가용 영역에서 모두 애플리케이션 서브넷을 사용합니다.
- 기존 로드 밸런서와 앞에서 생성한 대상 그룹을 사용합니다.
- Amazon ELB 상태 확인을 사용하고 상태 확인 유예 기간을 **300초**로 설정합니다.
- **Enable group metrics collection within CloudWatch**를 활성화합니다.
- 그룹 크기를 구성합니다.
 - **Desired capacity:** 2
 - **Minimum capacity:** 2
 - **Maximum capacity:** 4
- **Target tracking scaling policy**(대상 추적 조정 정책)가 포함된 **Scaling policies**를 구성하고 기본값을 사용합니다.
- 알림은 이 실습에서 사용되지 않으므로 구성하지 않아도 됩니다.
- **Auto scaling group**이 EC2 인스턴스를 시작했는지 확인합니다.
- Application Load Balancer DNS 이름을 사용하여 애플리케이션을 테스트합니다.

과제 6: AWS 관리 콘솔로 이동

최종 작업 요구 사항이 충족되었으므로 구축을 진행하여 프로젝트를 완료합니다. 작업 진행이 어려우면 섹션 2의 지침을 참조할 수 있지만, 일단은 섹션 2의 지침을 참조하지 않고 구축을 진행해 보는 것이 좋습니다.

축하합니다. Amazon Auto Scaling 그룹을 생성했으며 워드프레스 애플리케이션을 시작했습니다.

섹션 2: 상세 지침

이 섹션에는 모든 작업의 단계별 지침이 포함되어 있습니다. 솔루션을 직접 구축할 때 문제가 발생하면 이러한 지침을 참조할 수 있습니다. 먼저 **섹션 1**을 참조하여 솔루션을 구축해 보시기 바랍니다.

과제 1 지침: 미리 구성된 CloudFormation 템플릿 검토 및 실행

과제 1.1: CloudFormation 콘솔로 이동

4. AWS 관리 콘솔에서 AWS 검색 창을 사용하여 CloudFormation을 검색한 다음 결과 목록에서 해당 서비스를 선택합니다.

참고: Services 메뉴 오른쪽에 있는 통합 검색 창에는 *Search for services, features, marketplace products, and docs* 레이블이 표시되어 있습니다.

과제 1.2: CloudFormation 템플릿 다운로드 및 검토

5. 이 [Task1.yaml](#) 링크에서 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭 메뉴)를 열고 컴퓨터에 CloudFormation 템플릿을 저장하는 옵션을 선택합니다.
6. 다운로드한 파일을 워드 프로세서 이외의 텍스트 편집기에서 엽니다.
7. CloudFormation 템플릿을 검토합니다.
8. 이 템플릿에 의해 생성되는 리소스를 예측해 보십시오.

과제 1.3: CloudFormation 스택 생성

9. Create stack을 선택합니다.

참고: 콘솔이 Amazon CloudFormation 랜딩 페이지 대신 Stacks 페이지에서 시작하는 경우 두 단계를 거쳐 Create stack 페이지로 이동할 수 있습니다.

- Create stack ▼ 드롭다운 메뉴를 선택합니다.
- With new resources (standard)를 선택합니다.

Create Stack 페이지가 표시됩니다.

10. 다음을 구성합니다.
 - Template is ready를 선택합니다.
 - Amazon S3 URL을 선택합니다.

- 이 실습 지침 왼쪽에서 **Task1TemplateUrl** 값을 복사하여 **Amazon S3 URL** 텍스트 상자에 붙여넣습니다.
- **Next**를 선택합니다.

Specify stack details 페이지가 표시됩니다.

11. **Stack name**을 **VPCStack**으로 설정합니다.
12. **Parameters**는 기본값으로 설정된 상태로 유지합니다.
13. **Next**를 선택합니다.

Configure stack options 페이지가 표시됩니다. 이 페이지를 사용하여 추가 파라미터를 지정할 수 있습니다. 페이지를 탐색할 수 있지만 설정은 기본값 그대로 두십시오.

14. **Next**를 선택합니다.
15. 페이지 하단에서 **Create stack**을 선택합니다.

stack details 페이지가 표시됩니다.

이제 스택이 **CREATE_IN_PROGRESS** 상태가 됩니다.

16. **Stack info** 탭을 선택합니다.
17. 수시로 콘솔 새로 고침 버튼을 선택하십시오.
18. 스택 상태가 **CREATE_COMPLETE**로 변경될 때까지 기다립니다.

참고: 이 스택이 리소스를 배포하려면 최대 5분이 걸릴 수 있습니다.

과제 1.4: 콘솔에서 생성된 리소스 확인

19. **Resources** 탭을 선택합니다.

이 목록에는 생성 중인 리소스가 표시됩니다. CloudFormation은 리소스를 생성하는 최적의 순서를 결정합니다(예: 서브넷 전에 VPC 생성).

20. 스택에 배포된 리소스를 검토합니다.
21. **Events** 탭을 선택하고 목록을 스크롤합니다.

이 목록에는 리소스 생성 시작부터 리소스 생성 완료까지 CloudFormation이 수행하는 활동이 역순으로 표시됩니다. 스택을 생성하는 동안 발생한 모든 오류가 이 탭에 나열됩니다.

22. **Outputs** 탭을 선택합니다.

23. Outputs 섹션에서 키 값 페어를 검토합니다. 뒷부분의 실습 작업에서 이러한 값을 유용하게 활용할 수 있습니다.

축하합니다. 스택 구성 방법을 알아보았으며, 제공된 CloudFormation 템플릿을 사용하여 모든 리소스를 생성했습니다.

DO NOT COPY
pink0569@naver.com

과제 2 지침: Amazon RDS 데이터베이스 생성

과제 2.1: Amazon RDS 콘솔로 이동

24. AWS 관리 콘솔에서 AWS 검색 창을 사용하여 RDS를 검색한 다음 결과 목록에서 해당 서비스를 선택합니다.

Amazon RDS 콘솔 페이지가 표시됩니다.

과제 2.2: 새 DB 서브넷 그룹 생성

25. 왼쪽 탐색 창에서 Subnet groups를 선택합니다.

26. Create DB subnet group을 선택합니다.

Create DB subnet group 페이지가 표시됩니다.

27. Subnet group details 섹션에서 다음을 구성합니다.

- Name: AuroraSubnetGroup을 입력합니다.
- Description: A 2 AZ subnet group for my database를 입력합니다.
- VPC: 드롭다운 메뉴에서 LabVPC를 선택합니다.

28. Add subnets 섹션에서 다음을 구성합니다.

- Availability Zones 드롭다운 메뉴에서 다음을 수행합니다.
 - 이름이 **a**로 끝나는 가용 영역을 선택합니다.
 - 이름이 **b**로 끝나는 가용 영역을 선택합니다.
- Subnets 드롭다운 메뉴에서 다음을 수행합니다.
 - 이름이 **b**로 끝나는 가용 영역에서 CIDR 블록이 **10.0.5.0/24**인 서브넷을 선택합니다.
 - 이름이 **a**로 끝나는 가용 영역에서 CIDR 블록이 **10.0.4.0/24**인 서브넷을 선택합니다.

29. Create를 선택합니다.

Successfully created AuroraSubnetGroup 과 같은 메시지가 포함된 배너가 페이지 위쪽에 표시됩니다.

과제 2.3: 새 Amazon Aurora 데이터베이스 생성

30. 왼쪽 탐색 창에서 Databases를 선택합니다.

31. Create database를 선택합니다.

Create database 페이지가 표시됩니다.

32. Choose a database creation method 섹션에서 Standard Create를 선택합니다.

33. Engine options 섹션에서 다음을 구성합니다.

- Engine type에서 Amazon Aurora를 선택합니다.
- Edition에서 Amazon Aurora MySQL-Compatible Edition을 선택합니다.

34. Templates 섹션에서 Production을 선택합니다.

35. Settings 섹션에서 다음을 구성합니다.

- DB cluster identifier: MyDBCluster
- Master username: admin
- Master password: admin123
- Confirm password: admin123

참고: 자격 증명을 잘 기억해 두세요.

36. Instance configuration 섹션에서 다음을 구성합니다.

- DB instance class: Burstable classes를 선택합니다.
- instance type 드롭다운 메뉴에서 db.t3.small을 선택합니다.

37. Availability & durability 섹션의 Multi-AZ deployment에서 Create an Aurora Replica or Reader node in a different AZ를 선택합니다.

38. Connectivity 섹션에서 다음을 구성합니다.

- Virtual Private Cloud: 드롭다운 메뉴에서 LabVPC를 선택합니다.
- Subnet group: 드롭다운 메뉴에서 aurorasubnetgroup을 선택합니다.
- Public access: No를 선택합니다.
- VPC security group: Choose existing을 선택합니다.
- Existing VPC security groups:
 - 드롭다운 메뉴에서 xxxxx-RDSSecurityGroup-xxxxx를 선택합니다.
 - default 보안 그룹을 제거하려면 X를 선택합니다.
- Additional configuration 섹션을 확장하고 다음을 구성합니다.
 - Database port: 구성을 기본값으로 유지합니다.

39. **Monitoring** 섹션에서 **Enable Enhanced monitoring**을 선택 취소합니다.
40. 페이지 하단으로 스크롤하여 기본 ► **Additional configuration** 섹션을 확장합니다.
 - **Database options** 섹션에서 다음을 구성합니다.
 - **Initial database name:** WPDatabase
 - **Encryption** 섹션에서 **Enable encryption**을 선택 취소합니다.
 - **Maintenance** 섹션에서 **Enable auto minor version upgrade**를 선택 취소합니다.
 - **Deletion protection** 섹션에서 **Enable deletion protection**을 선택 취소합니다.
41. 화면 하단으로 스크롤한 다음 **Create database**를 선택합니다.

참고: Aurora MySQL DB 클러스터 시작 프로세스가 진행됩니다. 구성한 클러스터는 각각 다른 가용 영역에 있는 두 개의 인스턴스로 구성됩니다. Amazon Aurora DB 클러스터가 시작되려면 최대 5분이 걸릴 수 있습니다. mydbcluster 상태가 Available로 변경될 때까지 기다립니다. 작업을 계속 진행하기 위해 인스턴스가 사용 가능해질 때까지 기다릴 필요는 없습니다.

참고: Successfully created database mydbcluster. 메시지가 포함된 배너가 표시됩니다.

42. 성공 메시지의 끝부분에 표시된 **View connection details**를 선택하여 **mydbcluster** 데이터베이스의 연결 세부 정보를 텍스트 편집기에 저장합니다.

참고 "Failed to turn on DevOps Guru for mydbcluster-instance-x because of missing permissions" 오류가 표시되면 무시해도 됩니다.

△ 중요: 이 암호는 지금밖에 확인할 수 없습니다. 암호를 복사하여 참조용으로 저장해 두십시오. 이렇게 하지 않으면 데이터베이스를 수정해야 암호를 변경할 수 있습니다.

과제 2.4: 데이터베이스 메타데이터 복사

43. 왼쪽 탐색 창에서 **Databases**를 선택합니다.
44. **mydbcluster** 링크를 선택합니다.
45. **Connectivity & Security** 탭을 선택합니다.
46. **Writer** 인스턴스의 **endpoint** 값을 텍스트 편집기에 복사합니다.
 - **팁:** Writer 인스턴스 엔드포인트를 복사하려면 해당 엔드포인트 위에 마우스를 놓고 복사() 아이콘을 선택합니다.
47. **Master username** 값을 텍스트 편집기에 복사합니다.
48. **Master password**을 텍스트 편집기에 복사합니다. 이 값은 콘솔에서 가려집니다.
데이터베이스를 생성할 때부터 기억해야 합니다.
49. 왼쪽 탐색 창에서 **Databases**를 선택합니다.

50. **mydbcluster-instance-x Writer** 인스턴스 링크를 선택합니다.

51. **Configuration** 탭을 선택합니다.

52. **DB name** 값을 텍스트 편집기에 복사합니다.

축하합니다. Amazon Aurora 데이터베이스를 생성했습니다.

과제 3 지침: Amazon EFS 파일 시스템 생성

과제 3.1: EFS 콘솔로 이동

53. AWS 관리 콘솔에서 AWS 검색 창을 사용하여 EFS를 검색한 다음 결과 목록에서 해당 서비스를 선택합니다.

Amazon Elastic File System 콘솔 페이지가 표시됩니다.

과제 3.2: 새 파일 시스템 생성

54. Create file system을 선택합니다.

Create file system 페이지가 표시됩니다.

55. Customize를 선택합니다.

File system settings 페이지가 표시됩니다.

56. General 섹션에서 다음을 구성합니다.

- **Name:** myWPEFS
- **Enable automatic backups**를 선택 취소합니다.
- **Enable encryption of data at rest**를 선택 취소합니다.
- **Tags - optional** 섹션에서 다음을 구성합니다.
 - **Tag key:** Name
 - **Tag value – optional:** myWPEFS
- 다른 모든 설정은 기본값으로 유지합니다.

57. Next를 선택합니다.

Network access 페이지가 표시됩니다.

58. Virtual Private Cloud 드롭다운 메뉴에서 LabVPC를 선택합니다.

59. Mount targets에서 다음을 구성합니다.

- **Availability Zone:** 드롭다운 메뉴에서 이름이 “a”로 끝나는 가용 영역을 선택합니다.
- **Subnet ID:** 드롭다운 메뉴에서 AppSubnet1을 선택합니다.
- **Security group:** 드롭다운 메뉴에서 EFSMountTargetSecurityGroup을 선택합니다.
- **default 보안 그룹을 제거하려면 X**를 선택합니다.

- **Availability Zone:** 드롭다운 메뉴에서 이름이 “b”로 끝나는 가용 영역을 선택합니다.
- **Subnet ID:** 드롭다운 메뉴에서 **AppSubnet2**를 선택합니다.
- **Security group:** 드롭다운 메뉴에서 **EFSMountTargetSecurityGroup**을 선택합니다.
- **default** 보안 그룹을 제거하려면 **X**를 선택합니다.

60. Next를 선택합니다.

File system policy – optional 페이지가 표시됩니다.

참고: 이 실습에서는 이 페이지를 구성하지 않아도 됩니다.

61. Next를 선택합니다.

Review and create 페이지가 표시됩니다.

62. 페이지 하단으로 스크롤하여 **Create**를 선택합니다.

참고:

Success!

File system (fs-xxxxxx) is available 메시지가 포함된 배너가 표시됩니다.

몇 분 후에 파일 시스템 상태가 *Available*로 표시됩니다.

과제 3.3: EFS 메타데이터 복사

63. 왼쪽 탐색 창에서 **File systems**를 선택합니다.

64. *myWPEFS*으로 생성된 **File system ID**를 텍스트 편집기에 복사합니다. ID는 *fs-a1234567*과 같은 형식입니다.

축하합니다. Amazon EFS를 생성했습니다.

과제 4 지침: Application Load Balancer 생성

과제 4.1: Amazon EC2 콘솔로 이동

65. AWS 관리 콘솔에서 AWS 검색 창을 사용하여 EC2를 검색한 다음 결과 목록에서 해당 서비스를 선택합니다.

과제 4.2: 대상 그룹 생성

66. 왼쪽 탐색 창에서 **Target Groups**를 선택합니다.

67. Create target group을 선택합니다.

Specify group details 페이지가 표시됩니다.

68. **Basic configuration** 섹션에서 다음을 구성합니다.

- **Choose a target type:** Instances를 선택합니다.
- **Target group name:** myWPTargetGroup을 입력합니다.
- **VPC:** LabVPC를 선택합니다.

69. **Health checks** 섹션에서 다음을 구성합니다.

- **Health check path:** /wp-login.php를 입력합니다.
- ▶ **Advanced health check settings** 섹션을 확장하고 다음을 구성합니다.
 - **Healthy threshold:** 2를 입력합니다.
 - **Unhealthy threshold:** 10을 입력합니다.
 - **Timeout:** 50을 입력합니다.
 - **Interval:** 60을 입력합니다.

페이지의 나머지 설정은 기본값 그대로 둡니다.

70. Next를 선택합니다.

Register targets 페이지가 표시됩니다. 현재는 등록할 대상이 없습니다.

71. 페이지 하단으로 스크롤하여 Create target group을 선택합니다.

Successfully created target group: myWPTargetGroup와 같은 메시지가 표시됩니다.

과제 4.3: Application Load Balancer 생성

72. 왼쪽 탐색 창에서 **Load Balancers**를 선택합니다.
73. Create Load Balancer를 선택합니다.
74. Application Load Balancer 섹션에서 Create를 선택합니다.

Create Application Load Balancer 페이지가 표시됩니다.

75. **Basic configuration** 섹션에서 다음을 구성합니다.
 - **Load balancer name:** myWPAppALB를 입력합니다.
76. **Network mapping** 섹션에서 다음을 구성합니다.
 - **VPC:** LabVPC를 선택합니다.
 - **Mappings:**
 - 목록의 첫 번째 가용 영역을 선택하고 Subnet 드롭다운 메뉴에서 PublicSubnet1을 선택합니다.
 - 목록의 두 번째 가용 영역을 선택하고 Subnet 드롭다운 메뉴에서 PublicSubnet2를 선택합니다.
77. **Security groups** 섹션에서 다음을 구성합니다.
 - **Security groups** 드롭다운 메뉴에서 AppInstanceSecurityGroup을 선택합니다.
 - **default** 보안 그룹을 제거하려면 X를 선택합니다.
78. **Listeners and routing** 섹션에서 다음을 구성합니다.
 - **Listener HTTP:80: Default action** 드롭다운 메뉴에서 myWPTargetGroup을 선택합니다.
79. 페이지 하단으로 스크롤하여 Create load balancer를 선택합니다.
Successfully created load balancer:myWPAppALB
와 같은 메시지가 표시됩니다.
80. **View load balancers**를 선택합니다.
로드 밸런서가 몇 분 동안 Provisioning 상태였다가 Active로 바뀝니다.

작업 4.4: ELB 메타데이터 복사

81. myWPAppALB를 선택합니다.
82. **Description** 탭을 선택합니다.

83. DNS name을 텍스트 편집기에 복사합니다.

축하합니다. 대상 그룹과 Application Load Balancer를 생성했습니다.

DO NOT COPY
pink0569@naver.com

과제 5 지침: CloudFormation을 사용하여 시작 템플릿 생성

과제 5.1: CloudFormation 콘솔로 이동

84. AWS 관리 콘솔에서 AWS 검색 창을 사용하여 CloudFormation을 검색한 다음 결과 목록에서 해당 서비스를 선택합니다.

과제 5.2: CloudFormation 템플릿 다운로드 및 검토

85. 이 [Task5.yaml](#) 링크에서 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭 메뉴)를 열고 컴퓨터에 CloudFormation 템플릿을 저장하는 옵션을 선택합니다.
86. 다운로드한 파일을 워드 프로세서 이외의 텍스트 편집기에서 엽니다.
87. CloudFormation 템플릿을 검토합니다.
88. 이 템플릿에 의해 생성되는 리소스를 예측해 보십시오.

과제 5.3: CloudFormation 스택 생성

89. Create stack을 선택합니다.

참고: 콘솔이 Amazon CloudFormation 랜딩 페이지 대신 Stacks(스택) 페이지에서 시작하는 경우 두 단계를 거쳐 Create stack 페이지로 이동할 수 있습니다.

- Create stack ▼ 드롭다운 메뉴를 선택합니다.
- With new resources (standard)를 선택합니다.

Create Stack 페이지가 표시됩니다.

90. 다음을 구성합니다.

- Template is ready를 선택합니다.
- Amazon S3 URL을 선택합니다.
- 이 실습 지침 왼쪽에서 Task5TemplateUrl 값을 복사하여 Amazon S3 URL 텍스트 상자에 붙여넣습니다.
- Next를 선택합니다.

Specify stack details 페이지가 표시됩니다.

91. Stack name을 WPLaunchConfigStack으로 설정합니다.

92. **Parameters**를 구성합니다.

- **DB Name:** 과제 2에서 복사한 **DB name**을 붙여넣습니다.
참고: 클러스터 이름이 아니라, *initial database name*을 붙여넣습니다.
- **Database endpoint:** 과제 2에서 복사한 **Writer instance endpoint**를 붙여넣습니다.
- **Database User Name:** 과제 2에서 복사한 **Master username**을 붙여넣습니다.
- **Database Password:** 과제 2에서 복사한 **Database password**를 붙여넣습니다.
- **WordPress admin username:** wpadmin으로 기본 설정됩니다.
- **WordPress admin password:** wpadmin123으로 기본 설정됩니다.
- **WordPress admin email address:** 유효한 이메일 주소를 입력합니다.
- **Instance Type:** 기본값인 t3.medium을 그대로 둡니다.
- **ALBDnsName:** 작업 4에서 복사한 **DNS name** 값을 붙여넣습니다.
- **LatestAL2Amild:** 기본값을 그대로 둡니다.
- **WPElasticFileSystemID:** 작업 3에서 복사한 **File system ID** 값을 붙여넣습니다.

93. Next를 선택합니다.

Configure stack options 페이지가 표시됩니다. 이 페이지를 사용하여 추가 파라미터를 지정할 수 있습니다. 페이지를 탐색할 수 있지만 설정은 기본값 그대로 두십시오.

94. Next를 선택합니다.

Review 페이지가 표시됩니다. 이 페이지에는 모든 설정이 요약되어 있습니다.

95. 페이지 하단으로 스크롤하여 **Create stack**을 선택합니다.

stack details 페이지가 표시됩니다.

이제 스택이 ① CREATE_IN_PROGRESS 상태가 됩니다.

96. **Stack info** 탭을 선택합니다.

97. 수시로 콘솔 새로 고침 버튼을 선택하십시오.

98. 스택 상태가 CREATE_COMPLETE로 변경될 때까지 기다립니다.

참고: 이 스택이 리소스를 배포하려면 최대 5분이 걸릴 수 있습니다.

과제 5.4: 콘솔에서 생성된 리소스 확인

99. **Resources** 탭을 선택합니다.

이 목록에는 생성된 리소스가 표시됩니다.

축하합니다. 제공된 CloudFormation 템플릿을 사용하여 스택을 생성했습니다.

DO NOT COPY
pink0569@naver.com

과제 6 지침: Auto Scaling 그룹 및 크기 조정 정책을 구성하여 애플리케이션 서버 생성

과제 6.1: Auto Scaling 그룹 생성

100. AWS 관리 콘솔에서 AWS 검색 창을 사용하여 EC2를 검색한 다음 결과 목록에서 해당 서비스를 선택합니다.

101. 왼쪽 탐색 창의 Auto Scaling 섹션에서 Auto Scaling groups를 선택합니다.

102. Create an Auto Scaling group을 선택합니다.

Choose launch template or configuration 페이지가 표시됩니다.

103. 다음을 구성합니다.

- Auto Scaling group name: WP-ASG를 입력합니다.
- Launch Template: 작업 5에서 생성한 시작 템플릿을 선택합니다.

104. Next를 선택합니다.

Choose instance launch options 페이지가 표시됩니다.

105. Network 섹션에서 다음을 구성합니다.

- VPC: LabVPC를 선택합니다.
- Availability Zones and subnets: AppSubnet1 및 AppSubnet2를 선택합니다.

106. Next를 선택합니다.

Configure advanced options 페이지가 표시됩니다.

107. Configure advanced options 페이지에서 다음을 구성합니다.

- Attach to an existing load balancer를 선택합니다.
- Choose from your load balancer target groups를 선택합니다.
- Existing load balancer target groups 드롭다운 메뉴에서 myWPTargetGroup | HTTP를 선택합니다.
- Health check type: ELB를 선택합니다.
- Health check grace period: 기본값인 300 이상을 그대로 유지합니다.
- Monitoring: Enable group metrics collection within CloudWatch를 선택합니다.

108. Next를 선택합니다.

Configure group size and scaling policies 페이지가 표시됩니다.

109. **Configure group size and scaling policies** 페이지에서 다음을 구성합니다.

- **Group Size** 섹션:
 - **Desired capacity:** 2
 - **Minimum capacity:** 2
 - **Maximum capacity:** 4
- **Scaling policies** 섹션:
 - Target tracking scaling policy 옵션을 선택합니다.

이 섹션의 나머지 설정은 기본값 그대로 두어도 됩니다.

110. Next를 선택합니다.

Add notifications 페이지가 표시됩니다.

111. Next를 선택합니다.

Add tags 페이지가 표시됩니다.

112. Add tag를 선택하고 다음을 구성합니다.

- **Key:** Name을 입력합니다.
- **Value:** WP-App을 입력합니다.

113. Next를 선택합니다.

Review 페이지가 표시됩니다.

114. Auto Scaling 그룹 구성이 정확한지 검토한 다음 페이지 하단에서 Create Auto Scaling group을 선택합니다.

WP-ASG, 1 Scaling policy created successfully와 같은 메시지가 표시됩니다.

Auto Scaling groups 페이지가 표시됩니다.

Auto Scaling 그룹을 생성했으므로 이 그룹에서 EC2 인스턴스를 시작했는지 확인할 수 있습니다.

115. Auto Scaling 그룹 **WP-ASG** 링크를 선택합니다.

116. Auto Scaling 그룹에 대한 정보를 검토하려면 **Group Details** 섹션을 검사합니다.

117. **Activity** 탭을 선택합니다.

Activity History 섹션에서는 Auto Scaling 그룹에서 발생한 이벤트의 레코드가 유지됩니다. Status 열에는 인스턴스의 현재 상태가 포함됩니다. 인스턴스를 시작하면 상태 열에 *PreInService*가 표시됩니다. 인스턴스가 시작되면 상태가 *Successful*로 변경됩니다.

118. **Instance management** 탭을 선택합니다.

Auto Scaling 그룹이 2개의 Amazon EC2 인스턴스를 시작했고, 이 인스턴스는 *InService* 수명 주기 상태에 있습니다. Health Status 열에는 인스턴스의 Amazon EC2 인스턴스 상태 확인 결과가 표시됩니다.

인스턴스가 아직 *InService* 상태로 설정되지 않았다면 몇 분 기다려야 합니다. 새로 고침 버튼을 선택하여 인스턴스의 현재 수명 주기 상태를 검색할 수 있습니다.

119. **Monitoring** 탭을 선택합니다. 이 탭에서 Auto Scaling 그룹의 모니터링 관련 정보를 검토할 수 있습니다.

이 페이지는 Auto Scaling 그룹에서의 활동뿐 아니라 인스턴스의 사용량과 상태에 관한 정보를 제공합니다. **Auto Scaling** 탭에는 Auto Scaling 그룹에 대한 Amazon CloudWatch 지표가 표시되고, **EC2** 탭에는 Auto Scaling 그룹이 관리하는 Amazon EC2 인스턴스의 지표가 표시됩니다.

과제 6.2: 대상 그룹이 정상인지 확인

120. 왼쪽 탐색 창에서 **Target Groups**를 선택합니다.

121. **myWPTargetGroup** 링크를 선택합니다.

122. **Targets** 탭에서 인스턴스 상태가 *healthy*로 표시될 때까지 기다립니다.

참고: 상태 확인 결과가 *healthy*로 표시되려면 최대 5분이 걸릴 수 있습니다. 다음 작업을 계속 진행하고, 시간이 경과된 후 돌아와서 상태를 확인할 수 있습니다.

과제 6.3: 워드프레스 웹 애플리케이션에 로그인

123. 왼쪽 탐색 창에서 **Load Balancers**를 선택합니다.

124. **myWPAppALB**를 선택합니다.

125. **Description** 탭에서 DNS 이름을 텍스트 편집기에 복사하고 DNS 이름 끝에 `/wp-login.php` 값을 추가하여 워드프레스 애플리케이션 URL을 완성합니다.

완성된 워드프레스 애플리케이션 URL의 예:

`myWPAppELB-4e009e86b4f704cc.elb.us-west-2.amazonaws.com/wp-login.php`

126. 워드프레스 애플리케이션 URL 값을 새 브라우저 탭에 붙여넣습니다.

WordPress login 페이지가 표시됩니다.

127. 다음 정보를 입력합니다.

- Username or Email Address: wpadmin을 입력합니다.
- Password: wpadmin123을 입력합니다.

128. **Log in** 버튼을 선택합니다.

축하합니다. Amazon Auto Scaling 그룹을 생성했으며 워드프레스 애플리케이션을 시작했습니다.

실습 종료

다음 단계를 따라 콘솔을 닫고 실습을 종료한 후 실습 경험을 평가해 주십시오.

129. AWS Management Console로 돌아갑니다.
130. 탐색 모음에서 **awsstudent@<AccountNumber>** 를 선택한 다음 **Sign Out**을 선택합니다.
131. 실습 종료를 선택합니다.
132. OK를 선택합니다.
133. (선택 사항):
 - 해당하는 별 개수를 선택합니다.
 - 의견을 입력합니다.
 - **Submit**을 선택합니다.
 - 별 1개 = 매우 불만족
 - 별 2개 = 불만족
 - 별 3개 = 보통
 - 별 4개 = 만족
 - 별 5개 = 매우 만족

피드백을 제공하지 않으려면 그냥 창을 닫으면 됩니다.

AWS Training and Certification에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다.

피드백, 제안 사항 또는 수정 요청 사항을 제공하려면 AWS Training and Certification 문의 양식에 세부 정보를 입력해 주시기 바랍니다.