# Architecting on AWS – 실습 2
# AWS 에서 웹 애플리케이션 배포

## 실습 개요

**Windows 사용자** 이 실습용 웹 브라우저로 Google Chrome 또는 Mozilla Firefox 를 사용합니다. 실습 지침은 Amazon Relational Database(Amazon RDS) 콘솔의 차이로 인하여 **Microsoft Internet Explorer_와 호환되지 않습니다**.

기존에 웹 서버를 배포하고 보안을 구성하던 방법은 복잡하여 여러 팀이 관여해야 하고, 시간도 오래 지연되는 경우가 많았습니다. 다행히도 AWS 클라우드에 보안 인프라를 배포하는 것은 빠르고 간편합니다.

이 실습에서는 보안 그룹을 구성하고, Amazon Relational Database Service(Amazon RDS) 데이터베이스를 생성하고, Amazon Elastic Compute Cloud(Amazon EC2)로 웹 애플리케이션 서버를 시작하고, 웹 애플리케이션을 테스트합니다. 다음 이미지는 최종 아키텍처를 보여줍니다.


**목표**

이 실습을 완료하면 다음을 할 수 있게 됩니다.

- Amazon RDS 를 사용하여 데이터베이스 시작
- Amazon EC2 를 사용하여 애플리케이션 서버 시작
- EC2 인스턴스에 애플리케이션 자동 설치

소요 시간

이 실습을 완료하는 데는 약 **40 분**이 소요됩니다.

# 실습 시작

1. 이 링크를 마우스 오른쪽 버튼으로 클릭한 다음 자신의 컴퓨터로 arc_lab2_template.json 을 다운로드합니다.
2. AWS Management Console 의 서비스 메뉴에서 **Management & Governance > CloudFormation** 을 클릭합니다.
3. **Create stack** 을 클릭하고 아래 단계에 따라 스택을 생성합니다.

   **1 단계: 템플릿 지정**

   - **Template source:** **Upload a template file** 을 선택합니다.
   - **Upload a template file:** **Choose file** 을 클릭하고 다운로드한 **arc_lab2_template.json** 파일을 선택합니다.
   - **Next** 를 클릭합니다.

   **2 단계: 스택 세부 정보 지정**

   - **Stack name:** arclab2
   - **Next** 를 클릭합니다.

   **3 단계: 스택 옵션 구성**

   - **Next** 를 클릭합니다.

   **4 단계: 검토**

   - **I acknowledge that...** 의 체크박스에 체크합니다.
   - **Create stack** 을 클릭합니다.

AWS CloudFormation 에서는 이제 템플릿을 사용하여 리소스의 *스택* 을 생성합니다.

**Stack info** 탭을 클릭합니다.

- **Status** 가 CREATE_COMPLETE 로 변경될 때까지(약 3 분) 대기합니다.

  **참고** 필요한 경우 새로 고침 아이콘을 15 초마다 클릭하면 화면이 업데이트됩니다.

4. **Outputs** 탭을 클릭합니다.

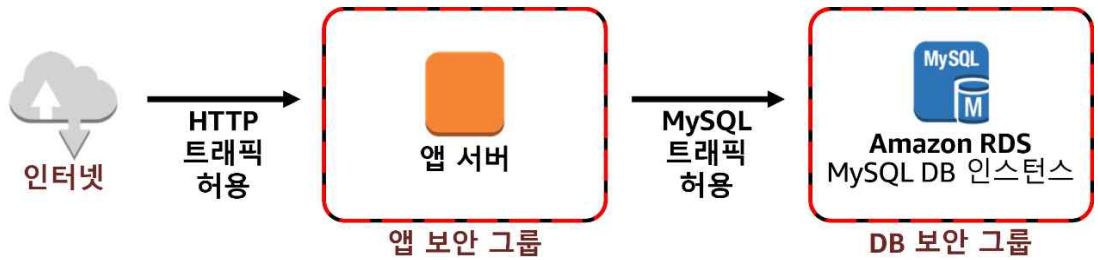
AWS CloudFormation 스택에서 지정된 리소스 ID 및 리소스 링크와 같은 *출력 정보*를 제공할 수 있습니다.

- **Region:** 생성된 리소스들의 리전 코드입니다.

# 작업 1: 보안 구성

아키텍처의 각 계층, 즉 애플리케이션, 서버, 네트워크를 비롯해 인터넷에 연결할 때도 보안을 구현해야 합니다.

이 작업에서는 Amazon EC2 애플리케이션 서버와 Amazon RDS 데이터베이스 인스턴스용 보안 그룹을 정의합니다. 다음 다이어그램은 이러한 보안 그룹과 트래픽 흐름 방식을 보여줍니다.




*보안 그룹*은 하나 이상의 인스턴스에 대한 트래픽을 제어하는 가상 방화벽 역할을 합니다. 인스턴스를 시작할 때 하나 이상의 보안 그룹을 인스턴스와 연결합니다. 각 보안 그룹에 *규칙*을 추가하고, 이러한 규칙은 그룹의 연결된 인스턴스와 트래픽을 주고받을 수 있도록 허용합니다. 언제든지 보안 그룹에 대한 규칙을 수정할 수 있습니다. 새 규칙은 보안 그룹과 연결된 모든 인스턴스에 자동으로 적용됩니다.

먼저 앱 보안 그룹을 생성하고 인터넷에서 들어오는 HTTP 연결을 허용하도록 구성합니다.

5. AWS Management Console 의 **Services** 메뉴에서 **Compute > EC2** 를 클릭합니다.

6. 화면 왼쪽 상단에 **New EC2 Experience** 가 표시되면, **New EC2 Experience** 가 선택되었는지 확인하십시오. 이 실습은 새로운 EC2 콘솔을 사용하도록 설계되었습니다.

7. 왼쪽 탐색 창에서 **Security Groups** 를 클릭합니다.

기존 보안 그룹 몇 개가 나열됩니다. 앱 서버에 대한 새 보안 그룹을 생성합니다.

8. **Create security group** 을 클릭합니다.

9. **Basic details** 에서 다음을 구성합니다.

- **Security group name:** App-SG

- **Description:** Allow HTTP access
- **VPC:** *Lab VPC*

**참고** 기존에 등록되어 있는 VPC 는 **X** 버튼을 눌러서 삭제 후 *Lab VPC* 를 선택하십시오.

10. **Inbound rules** 에서 **Add rule** 을 클릭하고 다음을 구성합니다.

- **Type:** *HTTP*
- **Source:** *Anywhere-ipv4*

11. 페이지 하단에서 **Create security group** 을 클릭합니다.

보안 그룹이 생성됩니다. 이후 실습에서 앱 서버를 시작할 때 이 보안 그룹을 사용합니다.

다음으로 데이터베이스 보안 그룹을 구성하고 앱 서버에서 수신되는 데이터베이스 연결을 허용하도록 구성합니다.

12. 왼쪽 탐색 창에서 **Security Groups** 를 클릭합니다.

13. **Create security group** 을 클릭합니다.

14. **Basic details** 에서 다음을 구성합니다.

- **Security group name:** DB-SG
- **Description:** Allow DB access
- **VPC:** *Lab VPC*

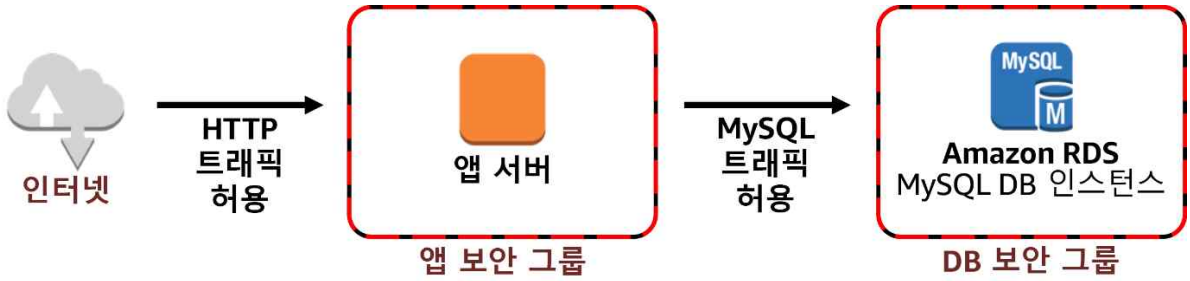
**참고** 기존에 등록되어 있는 VPC 는 **X** 버튼을 눌러서 삭제 후 *Lab VPC* 를 선택하십시오.

15. **Inbound rules** 에서 **Add rule** 을 클릭하고 다음을 구성합니다.

- **Type:** *MYSQL/Aurora*
- **Source:**
  - 돋보기 모양의 아이콘을 누르시고
  - sg 를 입력한 다음에 *App-SG* 를 선택합니다.

16. 페이지 하단에서 **Create security group** 을 클릭합니다.

다음 다이어그램과 같이 이 구성에 따라 데이터베이스 보안 그룹(DB-SG)이 애플리케이션 보안 그룹(App-SG)의 인바운드 액세스를 허용합니다.




데이터베이스 보안 그룹에 대한 인바운드 규칙을 생성할 때 애플리케이션 보안 그룹 ID 를 원본으로 사용했음을 알 수 있습니다. 한 보안 그룹이 다른 보안 그룹을 참조하는 기능은 강력한 기능입니다. 따라서 추가 EC2 인스턴스를 애플리케이션 보안 그룹과 연결하여 데이터베이스에 액세스할 수 있는 권한을 부여할 수 있습니다. 애플리케이션 보안 그룹과 연결된 인스턴스는 데이터베이스(정확히 말하면 데이터베이스 보안 그룹에 연결된 데이터베이스)와 통신할 수 있게 됩니다.

다음 작업에서는 새 데이터베이스 보안 그룹을 사용합니다.

# 작업 2: Amazon RDS 데이터베이스 생성

일반적으로 데이터베이스 생성은 데이터베이스 관리자 또는 시스템 관리자가 있어야 하는 복잡한 과정입니다. AWS 클라우드에서 Amazon Relational Database Service(Amazon RDS)를 사용하면 이 프로세스를 간소화할 수 있습니다.

이 작업에서는 Virtual Private Cloud(VPC)에서 MySQL 데이터베이스를 생성합니다. MySQL 은 널리 사용되는 오픈 소스 관계형 데이터베이스 관리 시스템(RDBMS)이기 때문에 소프트웨어 라이선스 비용이 들지 않습니다.

**Windows 사용자** 이 실습용 웹 브라우저로 Google Chrome 또는 Mozilla Firefox 를 사용합니다. 실습 지침은 Amazon Relational Database(Amazon RDS) 콘솔의 차이로 인하여 **Microsoft Internet Explorer 와 호환되지 않습니다**.

17. AWS Management Console 의 **Services** 메뉴에서 **Database > RDS** 를 클릭합니다.

18. **Create database** 를 클릭합니다.

**참고 Switch to the new database creation flow** 링크가 포함된 배너가 표시되는 경우 링크를 클릭하여 해당 워크플로로 전환하십시오.

19. **Choose a database creation method** 에서 **Standard create** 을 선택합니다.

20. **Engine options** 에서 **MySQL** 을 선택합니다.

21. **Templates** 에서 **Dev/Test** 를 선택합니다.

이제 자격 증명, 인스턴스 크기, 스토리지 유형 및 양과 같은 데이터베이스 설정을 구성할 수 있습니다. **다중 AZ 배포** 옵션은 가용성과 내구성을 제공하기 위해 다른 가용 영역에 예비 인스턴스를 생성합니다. 이 실습에서는 단일 데이터베이스 인스턴스를 사용합니다.

22. **Settings** 에서 다음을 구성합니다.

- **DB instance identifier:** inventory-db
- **Master username:** dbadmin
- [**Master password(마스터 비밀번호)**]: lab-password
- **Confirm password:** lab-password

23. **DB instance size** 에서 다음을 구성합니다.

- **Burstable classes (includes t classes)** 를 선택합니다.
- *db.t2.micro* 를 선택합니다.

  주의: 만일 메뉴에서 *db.t2.micro* 가 보이지 않는 다면 바로 아래 **Include previous generation classes** 버튼을 활성화 하면 보입니다.

24. **Connectivity** 에서 다음을 구성합니다.

- **Virtual private cloud (VPC):** *Lab VPC*
- **VPC security group: Choose existing** 을 선택합니다.
- **Existing VPC security groups:** 드롭다운 메뉴를 클릭합니다. **DB-SG** 를 선택하여 파란색으로 강조 표시한 다음 **default** 를 클릭하여 기본 보안 그룹을 제거합니다.

25. **Additional configuration** 를 확장하고 다음을 구성합니다.(주의 : Connectivity 항목의 하위 메뉴인 **Additional configuration** 이 아닙니다. 아래쪽에 **Additional configuration** 항목이 따로 존재 합니다.)

- **Initial database name:** inventory
- **Enable automated backups** 선택을 취소합니다.
- **Enable Enhanced monitoring** 선택을 취소합니다.

**참고 최초 데이터베이스 이름**은 애플리케이션에서 사용할 데이터베이스의 논리적 이름입니다.

**팁** 페이지에 표시된 다른 옵션을 자유롭게 살펴볼 수 있지만 기본값으로 설정된 상태로 두십시오. 여기에 포함된 옵션들에는 자동 백업, 로그 파일 내보내기 기능 및 자동 버전 업그레이드가 있습니다. 확인란을 클릭하여 이러한 기능들을 활성화하는 것은 데이터베이스를 직접 설치, 백업 및 유지 관리하는 대신에 완전관리형 데이터베이스 솔루션이 제공하는 기능을 보여주게 됩니다.

26. 페이지 하단에서 **Create database** 를 클릭합니다.

**참고 rds-monitoring-role** 을 언급하는 오류 메시지가 표시되면 **Additional configuration** 섹션에서 **Enable Enhanced monitoring** 을 선택하지 않았는지 확인하십시오. 그리고 다시 시도하십시오.

데이터베이스가 생성 중임을 나타내는 메시지가 표시됩니다. 몇 분 정도 걸리지만 다음 작업을 계속할 수 있습니다. 데이터베이스가 생성될 때까지 기다릴 필요는 없습니다.

# 작업 3: Amazon EC2 를 사용하여 애플리케이션 서버 시작

이제 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 시작하여 애플리케이션을 실행할 준비가 되었습니다. 시작할 때 애플리케이션을 자동으로 설치하는 *구성 스크립트* 가 제공됩니다. 그리고 앞선 실습에서 생성한 애플리케이션 보안 그룹과 인스턴스를 연결합니다. 그러면 인터넷에서 HTTP(웹) 액세스가 허용됩니다.

27. AWS Management Console 의 **Services** 메뉴에서 **Compute > EC2** 를 클릭합니다.

28. **Launch instance** 섹션으로 스크롤합니다.

29. **Launch instance** 를 클릭하고 *Launch instance* 를 선택합니다.

## 1 단계: 태그 추가

태그를 사용하여 목적, 소유자 또는 환경 등 다양한 방식으로 AWS 리소스를 분류할 수 있습니다. 이 기능은 동일한 유형의 리소스가 여러 개일 때 유용합니다. 태그로 특정 리소스를 신속하게 찾을 수 있습니다. 각 태그는 사용자가 정의하는 *키*와 *값*으로 구성됩니다.

30. **Add Tag** 를 클릭하고 다음을 구성합니다.

- **Key:** Name
- **Value:** App Server

EC2 Management Console 의 인스턴스에 **Name** 태그가 표시됩니다.

# 2 단계: AMI 선택

이제 인스턴스 시작에 사용할 디스크 볼륨 사본이 있는 *Amazon Machine Image(AMI)* 를 선택할 수 있습니다.

**참고** 여러 버전의 Microsoft Windows 및 Linux 가 포함된 AMI 목록을 확인합니다. 이러한 디스크 이미지는 정기적으로 업데이트되어 AWS 서비스 사용에 도움이 되는 보안 패치와 소프트웨어를 통합합니다. 사용자의 자체 데이터와 애플리케이션이 있는 AMI 를 생성하거나 AWS Marketplace 에서 사전에 구축된 상용 애플리케이션을 선택할 수 있습니다.

본 실습에서는 애플리케이션이 Amazon Linux 2 AMI 를 사용합니다.

31. **Amazon Linux 2 AMI** 에서 **Select** 를 클릭합니다.

# 3 단계: 인스턴스 유형 선택

이제 사용자의 EC2 인스턴스에 할당될 리소스를 결정하는 *인스턴스 유형* 을 선택할 수 있습니다. 각 인스턴스 유형은 가상 CPU, 메모리, 디스크 스토리지, 네트워크 성능 조합을 할당합니다.

인스턴스 유형은 컴퓨팅 최적화, 메모리 최적화 및 스토리지 최적화와 같은 *패밀리* 로 구분됩니다. 인스턴스 유형 이름에는 **t2** 또는 **m4** 와 같은 패밀리 식별자가 포함됩니다. 이 숫자는 인스턴스 *세대* 를 나타냅니다. 따라서 **m5** 가 **m4** 보다 최신 버전입니다.

이 애플리케이션은 **t2.micro** 인스턴스 유형을 사용하는데, 이것은 사용량이 많을 때 기본 성능을 초과할 수 있는 스몰 인스턴스입니다. 이 유형은 워크로드가 많은 애플리케이션의 개발과 테스트용으로 적합합니다.

32. **t2.micro** 인스턴스 유형을 선택합니다.

# 4 단계: 키페어

이 단계에서는 키페어를 설정합니다.

33. *Proceed without a key pair* 를 선택합니다.
34. 아래로 내려와  네트워크 설정에서 [편집] 버튼을 누른다

# 5 단계: 인스턴스 구성

이제 시작할 인스턴스의 수와 네트워크 구성 같은 인스턴스 세부 정보를 구성합니다. 각 필드에 대한 설명을 보려면 정보 아이콘에 커서를 올려놓습니다.

Lab VPC 네트워크 내 퍼블릭 서브넷에서 인스턴스를 시작할 수 있습니다.

35. 다음 설정을 구성합니다.

- **Network:** *Lab VPC*
- **Subnet:** *Public Subnet 1* (**참고 Private** 이 아닌 **Public** 이어야 합니다.)
- 퍼블릭 IP 자동 할당은 활성화로 그대로 둔다
- 방화벽(보안 그룹)을 '기존 보안 그룹 선택' 으로 선택하고
- 일반 보안 그룹을 App-SG 로 선택한다
- 맨 마지막으로 스크롤하고 **Advanced Details** 섹션을 확장합니다
- **IAM 인스턴스 프로파일 :** *Inventory-App-Role* 을 선택해준다

*(IAM Role 설명) Inventory-App-Role* 에 다음 정책이 연결되어 인스턴스에서 실행 중인 애플리케이션이 AWS 서비스로 요청할 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "ssm:*",
            "Resource": "arn:aws:ssm:*:*:parameter/inventory-app/*",
            "Effect": "Allow"
        }
    ]
}
```

이 경우 이 역할은 AWS Systems Manager Parameter Store 내에 있는 **inventory-app** 설정에 액세스할 권한을 부여합니다. 이 설정은 구성 설정을 저장할 때 사용됩니다.

맨 아래로 스크롤 합니다. User data 필드가 표시됩니다.

**참고** 인스턴스를 시작할 때 **User data** 필드를 통해 구성 스크립트를 전달할 수 있습니다. 이 스크립트를 사용하여 구성 작업을 수행하고 소프트웨어를 설치할 수 있습니다.

이 인스턴스에서는 Amazon Linux 를 실행하므로 인스턴스를 시작할 때 실행되는 *셸 스크립트*를 제공합니다.

36. **User data** 필드에 다음 스크립트를 복사하고 붙여 넣습니다.(배포 파일사용 가능)

```
#!/bin/bash
# Install Apache Web Server and PHP
yum install -y httpd mysql
amazon-linux-extras install -y php7.2
# Download Lab files
wget https://s3.us-west-2.amazonaws.com/arclab.applaycrew.com/Lab2Files/inventory-app.zip
unzip inventory-app.zip -d /var/www/html/
# Download and install the AWS SDK for PHP
wget https://s3.us-west-2.amazonaws.com/arclab.applaycrew.com/Lab2Files/aws.zip
unzip aws -d /var/www/html
# Turn on web server
chkconfig httpd on
service httpd start
```

이 스크립트는 다음 작업을 수행합니다.

- Apache 웹 서버(httpd)와 PHP 언어 설치
- Inventory 애플리케이션 및 AWS 소프트웨어 개발 키트(SDK) 다운로드
- 웹 서버를 활성화하여 부팅 시 자동으로 시작되도록 구성

이 유형의 스크립트를 사용하면 로그인하여 소프트웨어를 수동으로 구성할 필요 없이 새 인스턴스를 구성할 수 있습니다. 그리고 기술 인력의 개입 없이도 새 인스턴스를 시작하고 완벽하게 구성할 수 있기 때문에 스크립트를 통해 간편하게 자동화할 수 있습니다. Windows 인스턴스에서는 PowerShell 스크립트를 사용하여 구성할 수 있습니다.

37. **Launch Instances** 를 클릭합니다.

인스턴스가 시작되는 중입니다.

38. 페이지 하단에서 **View Instances** 를 클릭합니다.

인스턴스가 *pending* 상태로 표시될 수 있으며, 이는 시작 중임을 의미합니다. 상태가 *running* 으로 변경되는 경우 인스턴스 부팅이 시작된 것입니다. 구성 스크립트에서 애플리케이션을 설치 및 구성한 후에 인스턴스에 액세스할 수 있습니다.

다음 작업을 계속하기 전에 인스턴스 상태가 **running** 으로 표시될 때까지 대기합니다.

기다리는 동안 **Description** 탭에 표시된 정보를 검토합니다. 여기에는 인스턴스 유형, 보안 설정 및 네트워크 설정에 관한 정보가 포함되어 있습니다.

**참고** 잠시 후에 인스턴스 실행이 시작됩니다. 하지만 상태 확인의 경우 몇 분 더 소요됩니다.

# 작업 4: 애플리케이션 테스트

이제 애플리케이션이 작동하는지 테스트할 준비가 되었습니다. EC2 인스턴스의 IP 주소를 통해 웹 애플리케이션에 액세스할 수 있습니다.

39. EC2 Management Console 에서 **App Server** 를 선택합니다.

40. **Details** 탭에서 **IPv4 Public IP** 를 클립보드에 복사합니다.

**팁** IP 주소를 복사하려면 IP 주소에 커서를 가리킨 다음 복사 아이콘을 클릭합니다.

**팁** IP Address 로 오픈하기 위해 **open address** 버튼은 클릭하지 마십시요.

41. 새 웹 브라우저 탭을 열고, IP 주소를 주소 표시줄에 붙여 넣은 다음 Enter 를 누릅니다.

웹 애플리케이션이 표시됩니다. 애플리케이션이 아직 데이터베이스에 연결되지 않아 표시되는 정보가 거의 없습니다.

이제 앞에서 생성한 Amazon RDS DB 인스턴스를 사용하도록 애플리케이션을 구성할 수 있습니다. 먼저 *Database Endpoint* 를 불러와 애플리케이션이 데이터베이스에 연결하는 방법을 인식하게 합니다.

**42.** **AWS Management Console** 로 돌아갑니다. 애플리케이션 탭은 닫지 마십시오. (곧 이 탭으로 돌아올 것입니다.)

**43.** **Services** 메뉴에서 **Database > RDS** 를 클릭합니다.

**44.** 왼쪽 탐색 창에서 **Databases** 를 클릭합니다.

**45.** **inventory-db** 식별자를 클릭합니다.

**46.** **Connectivity & security** 섹션으로 스크롤한 다음 **Endpoint** 를 클립보드에 복사합니다.

*inventory-db.crwxbgqad61a.rds.amazonaws.com* 과 유사해야 합니다.

**47.** Inventory 애플리케이션이 포함된 웹 브라우저 탭으로 돌아갑니다.

**48.** **Settings** 를 클릭하고 다음을 구성합니다.

- **Endpoint:** 이전에 복사한 엔드포인트를 붙여넣습니다.
- **Database:** inventory
- **Username:** dbadmin
- **Password:** lab-password

**49.** **Save** 를 클릭합니다.

애플리케이션이 데이터베이스에 연결되어 일부 초기 데이터를 로드하고 정보를 표시합니다. 이 애플리케이션을 사용하면 매장의 재고에서 품목을 추가, 편집 또는 삭제할 수 있습니다.

재고 정보는 실습 앞 부분에서 생성한 Amazon RDS MySQL 데이터베이스에 저장됩니다. 즉, 웹 애플리케이션 서버에 장애가 발생하는 경우에도 데이터가 손실되지 않습니다. 그리고 여러 애플리케이션 서버에서 동일한 데이터에 액세스할 수 있습니다.

# 결론

축하합니다! 다음 작업이 성공적으로 완료되었습니다.

- Amazon RDS 를 사용하여 데이터베이스 시작
- Amazon EC2 를 사용하여 애플리케이션 서버 시작
- EC2 인스턴스에 애플리케이션 자동 설치

# 실습 종료

다음 순서 따라 실습 과정에서 생성 된 리소스를 정리하십시오.

1. **RDS:** Database 삭제 (오랜 시간 소요)

   Create final snapshot? 체크 해제, I acknowledge... 체크

2. **EC2:** App Server Instance 삭제

   **(1 번 RDS 삭제 완료 까지 기다린다)**

3. **EC2:** Security Group DB-SG 삭제 후 App-SG 삭제

4. **CloudFormation:** Stack 삭제

5. **S3:** Bucket(cf-templates 으로 시작하는 이름)의 파일과 Bucket 삭제

6. **끝.**