

제 목	04장 AWS RDS 연결하기	
상세내용	데이터베이스 연결하기 (RDS)	

## 1. RDS란? / RDS를 왜 사용하는걸까? / 현업에서의 RDS

### ✔ RDS(Relational Database Service)란?

#### ▶ 한 줄 요약 : 관계형 데이터베이스 서비스이다.

MySQL, MariaDB 등 여러 관계형 데이터베이스 서비스를 AWS로부터 빌려서 사용하는 형태이다.

### ✔ RDS(Relational Database Service)를 왜 사용하는걸까?

로컬 환경(ex. 내 노트북)에서 개발할 때는 로컬 환경에 설치된 MySQL과 같은 DB를 연결해서 사용한다. 하지만 서버를 배포하고 나면 서버가 내 컴퓨터에 설치된 MySQL과 연결을 할 수가 없다. DB도 외부 인터넷에서 접근할 수 있게 같이 배포해 주어야 한다.

이러한 이유 때문에 AWS RDS라는 데이터베이스를 빌려서 사용하는 것이다. 이 외에도 AWS RDS는 여러 편리한 부가기능(자동 백업, 모니터링, 다중 AZ 등)을 많이 가지고 있다.

### ✔ EC2에 MySQL을 직접 설치해서 운영하면 안 되나요?

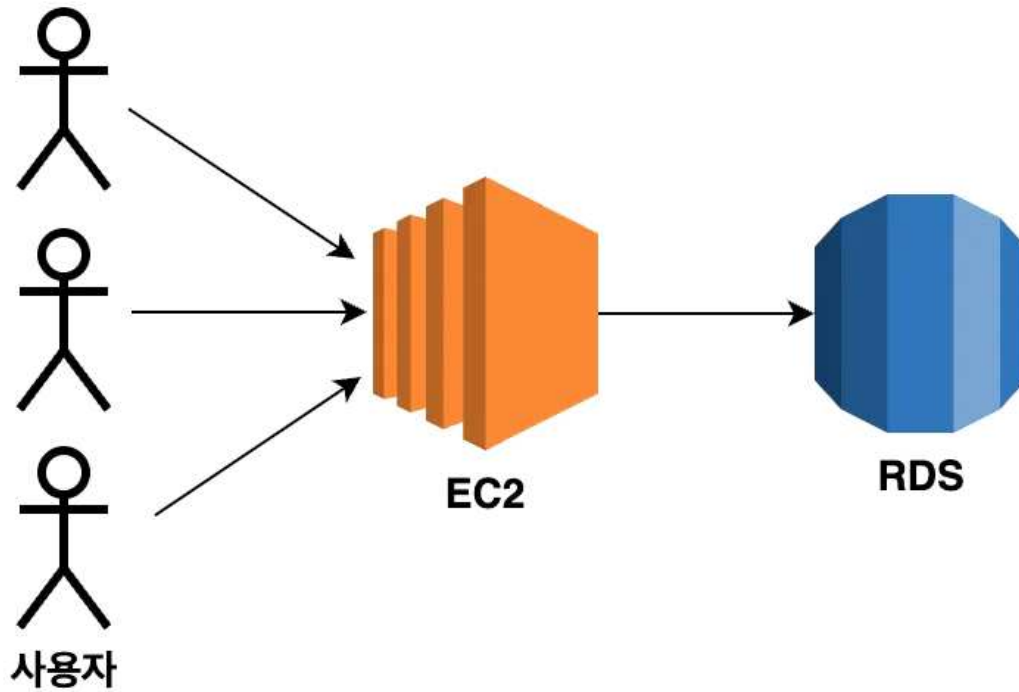
AWS에 대해 공부하면서 검색하다보면 하나의 EC2에 백엔드 서버와 MySQL을 같이 설치해서 쓰는 걸 볼 수 있다. 즉, RDS를 사용하지 않고 EC2에 MySQL을 직접 설치해서 사용하기도 한다.

EC2에 MySQL을 설치하면 별도의 RDS의 비용이 나오지 않기 때문에 비용을 절감할 수 있다. 비용의 이점 때문에 학생 때 자주 활용했던 구성 방법이다. 하지만 실제 현업에서는 하나의 EC2에 백엔드 서버와 MySQL을 같이 사용하지 않는다. 왜냐하면 백엔드 서버에 장애로 인해 EC2 컴퓨터가 죽을 경우, 애꿎은 MySQL도 같이 죽기 때문이다. 또한 RDS가 제공하는 부가적인 편리한 기능이 많아서 MySQL을 직접 설치해서 쓰지 않고 RDS를 쓰기도 한다.

정리하자면 현업에서는 EC2와 RDS를 분리해서 인프라를 구성하는 경우가 대부분이다. 하지만 비용이 부담된다면 하나의 EC2에 백엔드 서버와 MySQL을 직접 설치해서 사용해도 무방하다.

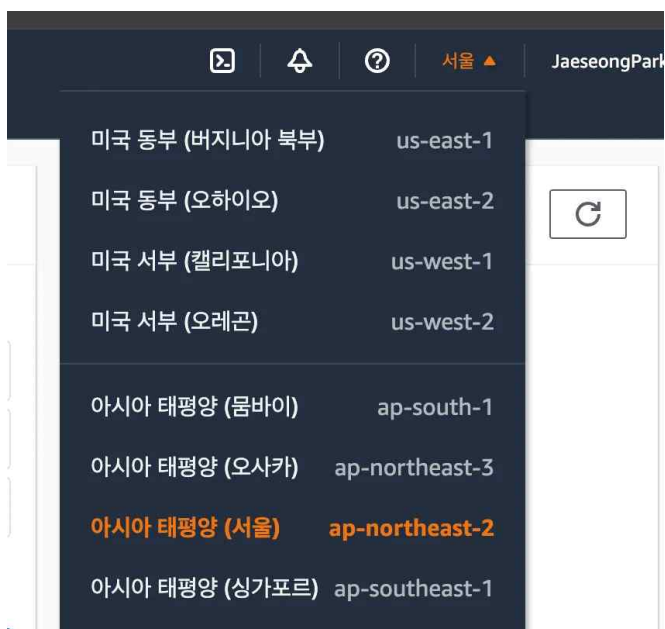
## 2. RDS를 활용한 아키텍처 구성

### ✓ RDS를 활용한 아키텍처 구성



### [실습] 1. RDS 생성하기

#### ✓ 1. 리전 선택



## ✓ 2. 데이터베이스 종류 선택

## 데이터베이스 생성 방식 선택 정보

### ☒ 표준 생성

가용성, 보안, 백업 및 유지 관리에 대한 옵션을 포함하여 모든 구성 옵션을 설정합니다.

### ☐ 손쉬운 생성

권장 모범 사례 구성을 사용합니다. 일부 구성 옵션은 데이터베이스를 생성한 후 변경할 수 있습니다.

## 엔진 옵션

### 엔진 유형 정보

#### ☐ Aurora (MySQL Compatible)

#### ☐ Aurora (PostgreSQL Compatible)

#### ☒ MySQL

#### ☐ MariaDB

#### ☐ PostgreSQL

#### ☐ Oracle

#### ☐ Microsoft SQL Server

### 에디션

#### ☒ MySQL Community

#### 알려진 문제/제한 사항

[알려진 문제/제한 사항](#)을 검토하여 특정 데이터베이스 버전과 발생할 수 있는 호환성 문제를 확인하세요.

### ▼ 필터 숨기기

#### ☒ 다중 AZ DB 클러스터를 지원하는 버전 표시 정보

기본 DB 인스턴스 1개와 읽기 가능한 대기 DB 인스턴스 2개로 다중 AZ DB 클러스터를 생성합니다. 다중 AZ DB 클러스터는 최대 2배 빠른 트랜잭션 커밋 지연 시간과 일반적으로 35초 미만의 자동 장애 조치를 제공합니다.

#### ☒ Amazon RDS 최적화된 쓰기를 지원하는 버전 표시 정보

Amazon RDS 최적화된 쓰기는 추가 비용 없이 쓰기 처리량(throughput)을 최대 2배 늘립니다.

### 엔진 버전

MySQL 8.0.33

### ✓ 3. 템플릿 선택

프리 티어를 선택하면 된다.

**템플릿**  
해당 사용 사례를 충족하는 샘플 템플릿을 선택하세요.

- ☐ **프로덕션**  
고가용성 및 빠르고 일관된 성능을 위해 기본값을 사용하세요.
- ☐ **개발/테스트**  
이 인스턴스는 프로덕션 환경 외부에서 개발 용도로 마련되었습니다.
- ☒ **프리 티어**  
RDS 프리 티어를 사용하여 새로운 애플리케이션을 개발하거나, 기존 애플리케이션을 테스트하거나 Amazon RDS에서 실무 경험을 쌓을 수 있습니다. [정보](#)

여기서 많이들 오해하는 게 프리 티어는 학습할 때나 테스트할 때만 쓰는 안 좋은 사양의 컴퓨터라고 생각한다. 하지만 실제 서비스에서 활용해도 될 정도로 나름 괜찮은 사양이다. 하루 방문자 수가 2,000명 정도였던 서비스를 운영했었는데 문제 없이 잘 돌아갔다. 성능에 문제가 직접적으로 생기기 전까지는 너무 걱정하지 말자.

### ✓ 4. 설정

**설정**

**DB 인스턴스 식별자** [정보](#)  
DB 인스턴스 이름을 입력하세요. 이름은 현재 AWS 리전에서 AWS 계정이 소유하는 모든 DB 인스턴스에 대해 고유해야 합니다.

test

DB 인스턴스 식별자는 대소문자를 구분하지 않지만 'mydbinstance'와 같이 모두 소문자로 저장됩니다. 제약: 1자~60자의 영숫자 또는 하이픈으로 구성되어야 합니다. 첫 번째 문자는 글자이어야 합니다. 하이픈 2개가 연속될 수 없습니다. 끝에 하이픈이 올 수 없습니다.

▼ **자격 증명 설정**

**마스터 사용자 이름** [정보](#)  
DB 인스턴스의 마스터 사용자에게 로그인 ID를 입력하세요.

admin

1~16자의 영숫자. 첫 번째 문자는 글자여야 합니다.

☐ **암호 자동 생성**  
Amazon RDS에서 사용자를 대신하여 암호를 생성하거나 사용자가 직접 암호를 지정할 수 있습니다.

**마스터 암호** [정보](#)

.....

제약 조건: 8자 이상의 인쇄 가능한 ASCII 문자. 다음은 포함할 수 없습니다. /(슬래시), '(작은따옴표), "(큰따옴표) 및 @(앳 기호).

**암호 확인** [정보](#)

마스터 사용자 이름과 마스터 암호는 데이터베이스에 접근하기 위한 아이디와 비밀번호와 같은 값이다. 따라서 마스터 사용자 이름과 마스터 암호는 따로 적어두자.

## ✓ 5. 인스턴스 구성, 스토리지

(기본값으로 그대로 두자.)

## ✓ 6. 연결

퍼블릭 액세스를 **예로 체크**하자. 여러 환경(로컬 환경, 개발 환경 등)에서 편하게 DB에 접근할 수 있게 된다.

연결 정보

컴퓨팅 리소스

이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정할지를 선택합니다. 연결을 설정하면 컴퓨팅 리소스가 이 데이터베이스에 연결할 수 있도록 연결 설정이 자동으로 변경됩니다.

☒ EC2 컴퓨팅 리소스에 연결 안 함  
이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정하지 않습니다. 나중에 컴퓨팅 리소스에 대한 연결을 수동으로 설정할 수 있습니다.

☐ EC2 컴퓨팅 리소스에 연결  
이 데이터베이스의 EC2 컴퓨팅 리소스에 대한 연결을 설정합니다.

Virtual Private Cloud(VPC) 정보

VPC를 선택합니다. VPC는 이 DB 인스턴스의 가상 네트워킹 환경을 정의합니다.

Default VPC (vpc-d0897abb)

4 서브넷, 4 가용 영역

해당 DB 서브넷 그룹이 있는 VPC만 나열됩니다.

① 데이터베이스를 생성한 후에는 VPC를 변경할 수 없습니다.

DB 서브넷 그룹 정보

DB 서브넷 그룹을 선택합니다. DB 서브넷 그룹은 선택한 VPC에서 DB 인스턴스가 어떤 서브넷과 IP 범위를 사용할 수 있는지를 정의합니다.

default

2 서브넷, 2 가용 영역

퍼블릭 액세스 정보

☒ 예  
RDS는 데이터베이스에 퍼블릭 IP 주소를 할당합니다. VPC 외부의 Amazon EC2 인스턴스 및 다른 리소스가 데이터베이스에 연결할 수 있습니다. VPC 내부의 리소스도 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

☐ 아니요  
RDS는 퍼블릭 IP 주소를 데이터베이스에 할당하지 않습니다. VPC 내부의 Amazon EC2 인스턴스 및 다른 리소스만 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

VPC 보안 그룹(방화벽) 정보

데이터베이스에 대한 액세스를 허용할 VPC 보안 그룹을 하나 이상 선택합니다. 보안 그룹 규칙이 적절한 수신 트래픽을 허용하는지 확인합니다.

☒ 기존 항목 선택  
기존 VPC 보안 그룹 선택

☐ 새로 생성  
새 VPC 보안 그룹 생성

기존 VPC 보안 그룹

하나 이상의 옵션 선택

default

×

가용 영역 정보

기본 설정 없음

RDS 프록시

RDS 프록시는 애플리케이션 확장성, 복원력 및 보안을 개선하는 완전관리형 고가용성 데이터베이스 프록시입니다.

☐ RDS 프록시 생성 정보  
RDS는 프록시에 대한 IAM 역할과 Secrets Manager 보안 암호를 자동으로 생성합니다. RDS 프록시에 대한 추가 비용이 있습니다. 자세한 내용은 다음을 참조하세요. [Amazon RDS 프록시 요금](#).

인증 기관 - 선택 사항 정보

서버 인증서를 사용하면 Amazon 데이터베이스에 대한 연결이 이루어지고 있는지 검증하여 추가 보안 계층을 제공합니다. 프로비저닝하는 모든 데이터베이스에 자동으로 설치되는 서버 인증서를 확인하여 이를 수행합니다.

rds-ca-2019 (기본값)

만료: Aug 23, 2024

인증 기관을 선택하지 않으면 RDS에서 대신 인증 기관을 선택합니다.

▶ 추가 구성

- 5 -

나중에 보안에 조금 더 신경써서 구성하고 싶을 때는 퍼블릭 액세스를 **\*\*아니오\*\***로 체크하고 RDS를 만들 때도 있다. 하지만 입문할 때는 불편하기 때문에 퍼블릭 액세스를 **\*\*예\*\***로 두고 많이 사용한다.

이렇게 설정하면 보안적으로 치명적인 위험이 있지 않을까 걱정하는 분들이 있다. 생각보다 그렇지 않다. 그러니 안심하고 퍼블릭 액세스를 **\*\*예\*\***로 체크하고 RDS를 만들어서 연습하자.

## ✔ 7. 데이터베이스 인증, 모니터링

(기본값으로 그대로 두자.)

## ✔ 8. 비용

프리 티어일 경우 아래에서 책정된 금액 정도의 비용이 나올 일은 없으니 걱정하지 말자. (아주 약간의 비용은 나올 수 있다.)

### Estimated Monthly costs

DB 인스턴스	18.98 USD
스토리지	2.62 USD
<b>합계</b>	<b>21.60 USD</b>

청구 예상 비용은 [Amazon RDS 요금](#)에 설명된 바와 같이 온디맨드 방식의 사용량을 기준으로 청구됩니다. 예상 비용에는 백업 스토리지, IO(해당되는 경우) 또는 데이터 전송 비용이 포함되지 않습니다.

다음을 사용하여 DB 인스턴스의 월별 청구액을 추산할 수 있습니다. [AWS 월 사용량 계산기](#).

### 월별 추정 요금

Amazon RDS 프리 티어는 12개월 동안 사용할 수 있습니다. 매월 프리 티어를 통해 아래 나열된 Amazon RDS 리소스를 무료로 사용할 수 있습니다.

- 단일 AZ db.t2.micro, db.t3.micro 또는 db.t4g.micro 인스턴스에서 Amazon RDS를 750시간 사용.
- 20GB의 범용 스토리지(SSD).
- 20GB의 자동 백업 스토리지 및 사용자가 시작한 모든 DB 스냅샷.

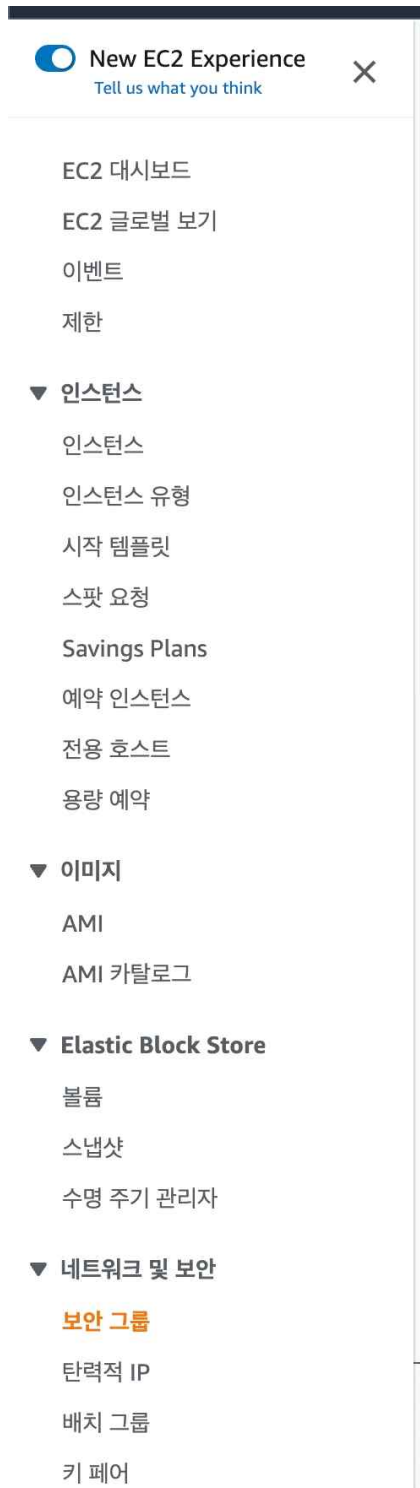
[AWS 프리 티어에 대해 자세히 알아보세요.](#)

무료 사용이 만료되었거나 애플리케이션에서 프리 티어 사용량을 초과한 경우 [Amazon RDS 요금 페이지](#)에서 설명한 대로, 표준 종량 서비스 요금이 적용됩니다.

## [실습] 2. 보안그룹 설정하기

### ✓ 1. 보안그룹 생성하기

#### 1.1. 'AWS EC2 - 보안 그룹' 메뉴 선택



별도의 설정을 하지 않았다면 RDS의 MySQL은 3306번 포트에서 실행된다. DB에 접근하기 위해 3306번 포트를 인바운드 규칙에 추가해주자. 아웃바운드 규칙에는 모든 트래픽을 허용하는 규칙을 추가해주자.

취소      **보안 그룹 생성**



## ✓ 2. 생성한 보안그룹을 RDS에 붙이기

데이터베이스 (2) 그룹 리소스 수정 작업 ▼ S3에서 복원 데이터베이스 생성

Q 데이터베이스를(들) 기준으로 필터링

DB 식별자 ▲	상태 ▼	역할 ▼	엔진 ▼	리전 및 AZ ▼	크기 ▼	작업 ▼	CPU ▼	현재 활동 ▼
jscode	사용 가능	인스턴스	MySQL Community	ap-northeast-2c	db.t3.micro	1 작업	4.80%	3 연결
practice	사용 가능	인스턴스	MySQL Community	ap-northeast-2c	db.t3.micro	-	6.56%	0 연결

### 연결

네트워크 유형 **정보**  
 듀얼 스택 모드를 사용하려면 IPv6 CIDR 블록을 지정한 VPC의 서브넷과 연결해야 합니다.

☒ **IPv4**  
 리소스는 IPv4 주소 지정 프로토콜을 통해 서만 통신할 수 있습니다.

☐ **듀얼 스택 모드**  
 리소스는 IPv4, IPv6 또는 둘 모두를 통해 통신할 수 있습니다.

**DB 서브넷 그룹**  
 default ▼

**보안 그룹**  
 이 DB 인스턴스와 연결할 DB 보안 그룹 목록입니다.  
 보안 그룹 선택 ▼

**db-security-group** ✕

**인증 기관 정보**  
 서버 인증서를 사용하면 Amazon 데이터베이스에 대한 연결이 이루어지고 있는지 검증하여 추가 보안 계층을 제공합니다. 프로비저닝하는 모든 데이터베이스에 자동으로 설치되는 서버 인증서를 확인하여 이를 수행합니다.

rds-ca-2019 (기본값) ▼  
 만료: Aug 23, 2024

▶ 추가 구성

## [실습] 3. 파라미터 그룹 추가하기

### ✔ 파라미터 그룹 생성하기



#### 1. 아래 속성 전부 **utf8mb4**로 설정하기

- `character\_set\_client`
- `character\_set\_connection`
- `character\_set\_database`
- `character\_set\_filesystem`
- `character\_set\_results`
- `character\_set\_server`

**참고)** **utf8** 대신에 **utf8mb4**를 사용하는 이유는 '한글' 뿐만 아니라 '이모티콘'도 지원이 가능하도록 하기 위해서이다.

## 2. 아래 속성 전부 `utf8mb4\_unicode\_ci`로 설정하기

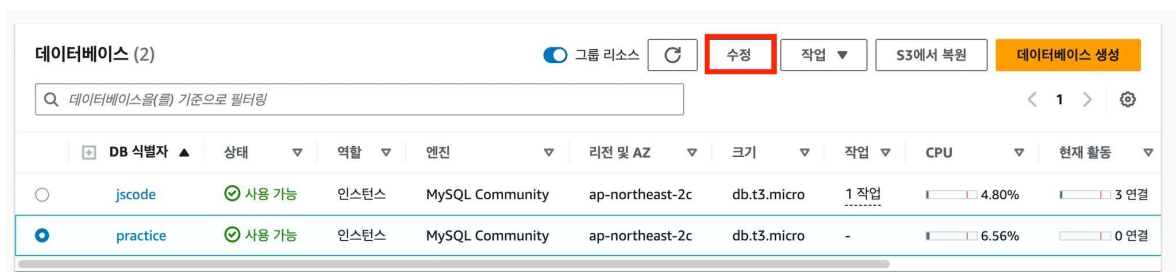
- `collation\_connection`
- `collation\_server`

참고) `utf8mb4\_unicode\_ci`은 정렬, 비교 방식을 나타낸다.

## 3. `time\_zone`을 `Asia/Seoul`로 설정하기

### ✓ RDS의 파라미터 그룹 변경하기

RDS의 DB 인스턴스 수정을 통해 DB 파라미터 그룹 변경하면 된다.



#### ▼ 추가 구성

데이터베이스 옵션, 백업 꺼짐, 향상된 모니터링 꺼짐, 유지 관리, CloudWatch Logs, 삭제 방지 꺼짐

#### 데이터베이스 옵션

DB 파라미터 그룹 정보

korean-mysql8

옵션 그룹 정보

default:mysql-8-0

주의) DB 파라미터 그룹을 변경한 뒤에는 RDS의 DB를 재부팅해야만 정상적으로 적용된다.

## [실습] 4. RDS에 접속하기

### ▶ VSCode를 이용해 RDS에 접속

DB 관리 전문 툴 사용시 어떤 툴을 써야 할 지 고민인 사람이라면, 최근 트렌드, 대중성 및 편리성을 고려했을 때 Datagrip → DBeaver → Workbench 순으로 추천한다.

### ✓ RDS 인스턴스에 접속하기

RDS > 데이터베이스 > practice

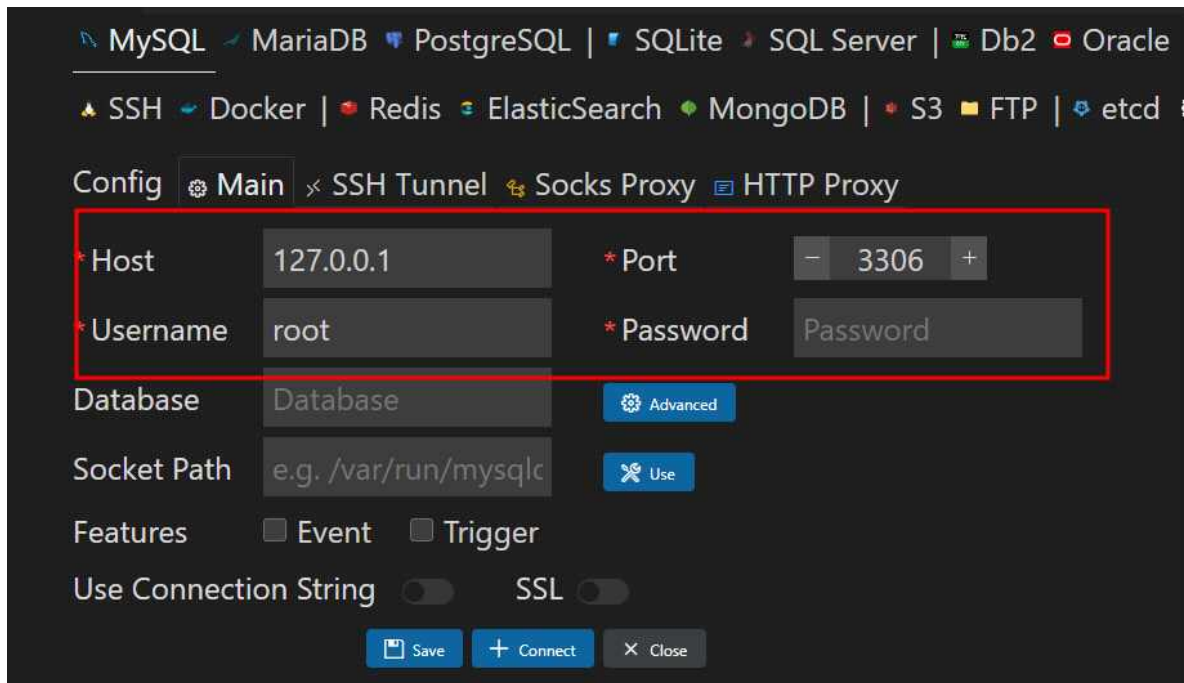
**practice** 수정 작업 ▼

요약			
DB 식별자 practice	CPU 2.44%	상태 🟢 사용 가능	클래스 db.t3.micro
역할 인스턴스	현재 활동 0 연결	엔진 MySQL Community	리전 및 AZ ap-northeast-2c

[연결 & 보안](#) | [모니터링](#) | [로그 및 이벤트](#) | [구성](#) | [유지 관리 및 백업](#) | [태그](#)

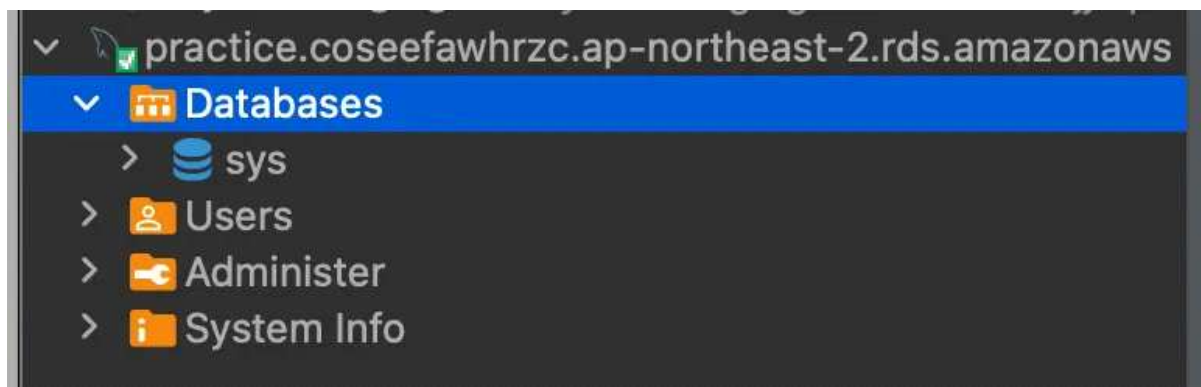
#### 연결 & 보안

엔드포인트 및 포트	네트워킹	보안
<b>엔드포인트</b> practice.coseefawhrzc.ap-northeast-2.rds.amazonaws.com	<b>가용 영역</b> ap-northeast-2c	<b>VPC 보안 그룹</b> default (sg-2287f34d) 🟢 <b>활성</b>
<b>포트</b> 3306	<b>VPC</b> vpc-d0897abb	<b>퍼블릭 액세스 가능</b> 예
	<b>서브넷 그룹</b> default	<b>인증 기관</b> <a href="#">정보</a> rds-ca-2019
	<b>서브넷</b> subnet-35db365e subnet-3a9dcc76	<b>인증 기관 날짜</b> August 23, 2024, 02:08 (UTC+09:00)
	<b>네트워크 유형</b> IPv4	<b>DB 인스턴스 인증서 만료 날짜</b> August 23, 2024, 02:08 (UTC+09:00)



- **Host**에는 **\*\*RDS의 엔드포인트\*\***를 입력하면 된다.
- **Username**에는 **\*\*마스터 사용자 이름(AWS RDS 데이터베이스 만들 때 정함)\*\***을 넣으면 된다.
- **Password**에는 **\*\*마스터 암호(AWS RDS 데이터베이스 만들 때 정함)\*\***를 넣으면 된다.

[정상 연결 화면]



✓ 데이터베이스도 생성해보기

- mydb이라는 데이터베이스를 하나 만들어보자.

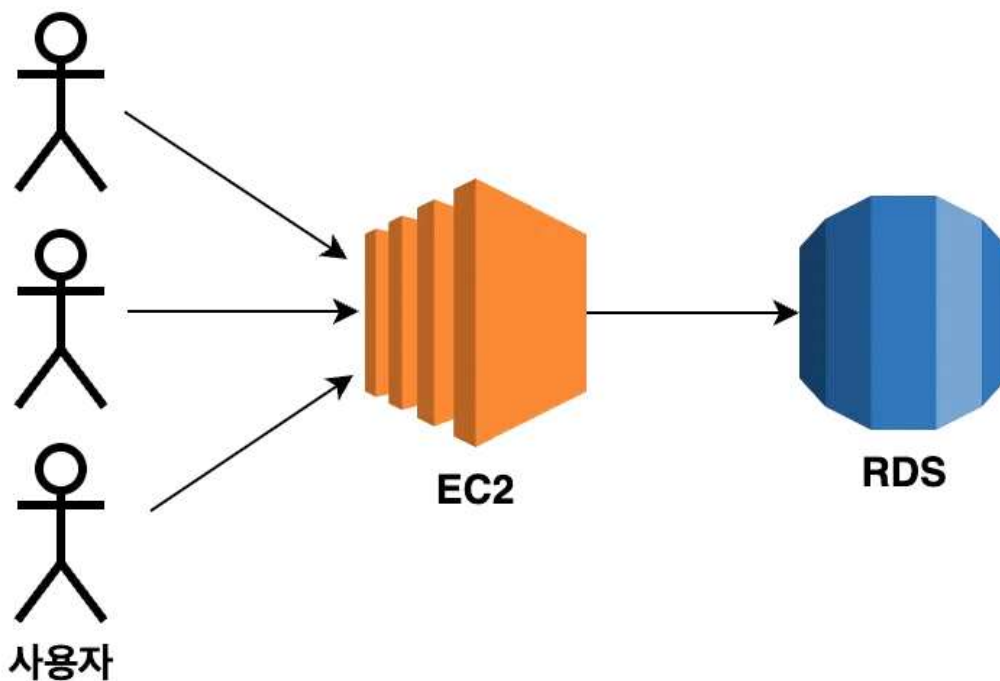
## ✓ 엔드포인트(Endpoint)란?

특정 리소스(ex. 서버, DB 등)에 접근할 수 있도록 해주는 URL을 의미한다.

### ▶ 왜 엔드포인트(Endpoint)란 이름이 붙었을까?

엔드포인트(Endpoint)를 있는 그대로 해석해보면 ‘\*\*끝점\*\*’이라는 의미이다. 어떤 게 대체 ‘\*\*끝점\*\*’이라는 의미를 가지고 있는걸까? 우리는 EC2나 RDS와 같은 리소스에 접근 할 때 특정 주소를 통해 접근한다. 그 주소가 해당 리소스의 대문같은 역할을 한다. 그리고 그 대문을 넘어서면 내부에 있는 코드나 데이터들을 자세히 들여다 볼 수 있다. 여기서 리소스에 접근할 때 가장 처음 마주치는 대문은 가장 끝 단에 위치하고 있다. 이런 관점에서 엔드포인트(Endpoint)라는 이름을 붙였지 않았을까 싶다.

(개인적인 추측일 뿐이니 위에서 설명한 유래는 엔드포인트라는 단어의 이미지를 머릿속에 떠올리는 용도로만 사용하자.)



## [실습] 5. Express 서버에 RDS 연결하기

### ✓ 1. EC2 인스턴스에서 Express 서버 clone하기

▶ 실습 깃코드: <https://github.com/JSCODE-EDU/rds-express-sequelize-sample.git>

```
$ git clone https://github.com/JSCODE-EDU/rds-express-sequelize-sample.git
$ cd rds-express-sequelize-sample
$ npm i
```

### ✓ 2. .env 파일 수정하기

내가 직접 만든 RDS의 정보에 맞게 `.env` 파일을 수정하면 된다.

.env

```
DATABASE_NAME=instagram
DATABASE_USERNAME=admin
DATABASE_PASSWORD=password
DATABASE_HOST=-----.ap-northeast-2.rds.amazonaws.com
```

주의) `.env` 파일은 반드시 `.gitignore`에 추가해서 Github에 올라가지 않게 해야 합니다.

(위 Github Repository는 편의상 .env를 Github에 올린 것 뿐입니다.)

### ✓ 3. Express 서버가 RDS와 잘 연결되는 지 확인하기

```
$ node app.js
```

[잘 연결된 경우]

```
jaeseong ~/Documents/Develop/jscode/nodejs-class/aws-express-sequelize-sample main ± node app.js
Executing (default): SELECT 1+1 AS result
DB 연결 성공!
Example app listening on port 80
```

## [중요] 비용 나가지 않게 RDS 깔끔하게 종료하기

### ✔ RDS 데이터베이스 삭제하기



database-1 인스턴스를 삭제하시겠습니까?

빠른 작업 - 신규

다중 AZ 배포로 변환

일시적으로 중지

재부팅

삭제

EC2 연결 설정

DB 인스턴스 **database-1**을(를) 영구적으로 삭제하시겠습니까? 이 작업은 실행 취소할 수 없습니다.

⚠

이 작업을 계속하면 인스턴스와 모든 해당 콘텐츠가 삭제되고 관련 리소스에 영향을 줄 수 있습니다. [자세히 알아보기](#)

☐

최종 스냅샷 생성

DB 인스턴스를 삭제하기 전에 최종 DB 스냅샷을 생성할지 여부를 결정합니다.

☐

자동 백업 보존

삭제 후 1일 동안 자동 백업을 보존할지 결정합니다.

☒

인스턴스 삭제 시 시스템 스냅샷 및 특정 시점으로 복구를 포함한 자동화된 백업을 더 이상 사용할 수 없다는 점을 인정합니다.

실수로 삭제되는 것을 방지하기 위해 추가 서면 동의를 제공하세요.

삭제를 확인하려면 필드에 *delete me*을(를) 입력하세요.

delete me

⚠

인스턴스를 삭제한 후에는 자동화된 백업을 더 이상 사용할 수 없기 때문에 인스턴스를 삭제하기 전에 최종 스냅샷을 만드는 것을 권장합니다.

취소

삭제

최종 스냅샷 생성과 자동 백업 보존을 체크하면 비용이 나간다. 따라서 실제 운영용 데이터베이스가 아니라면 체크를 해제하고 삭제를 하자.