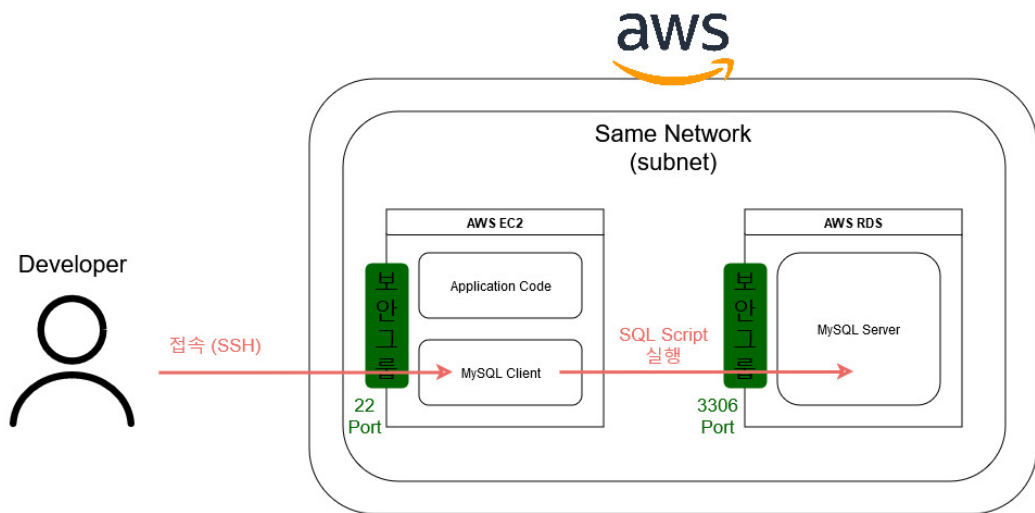


## DB Implementation and administration

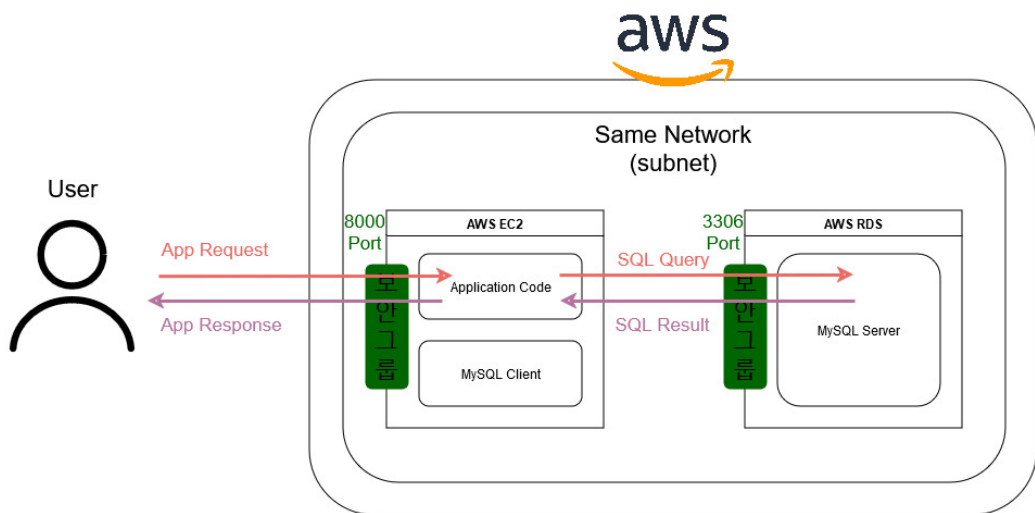
## 개요

대략적인 구조는 아래와 같습니다.

## 개발



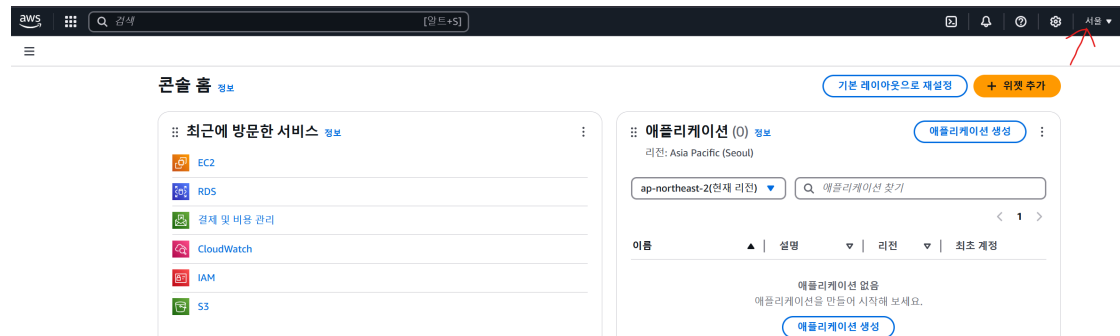
## 사용자 접속



## AWS EC2 생성

AWS Console에서 다음의 과정들을 거쳐 필요한 자원을 생성합니다.

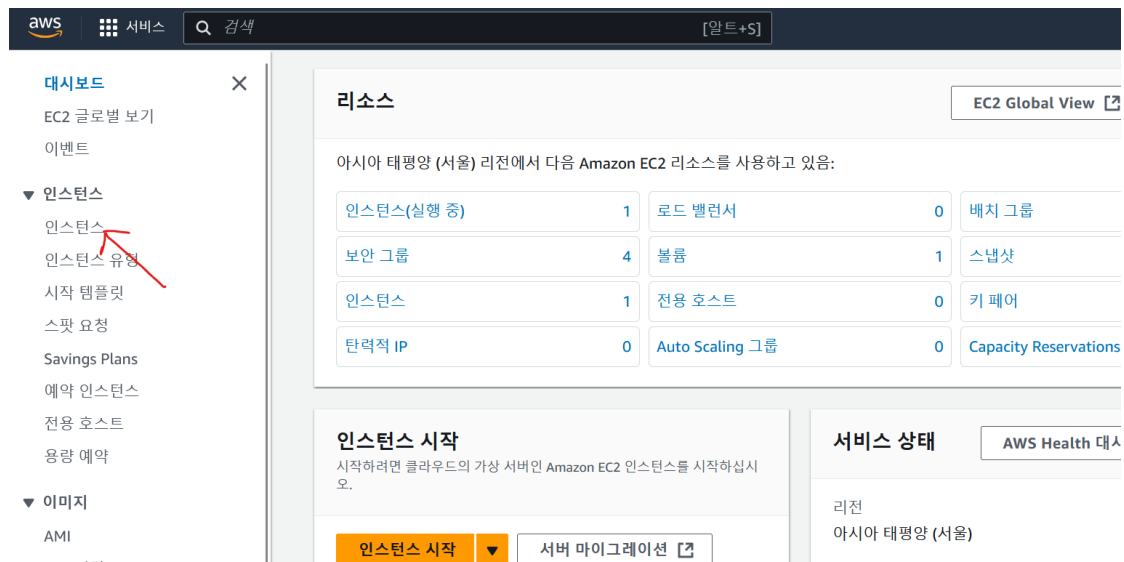
먼저 Application Server이자 Mysql Client 역할을 해줄 컴퓨터(인스턴스)를 생성하겠습니다.



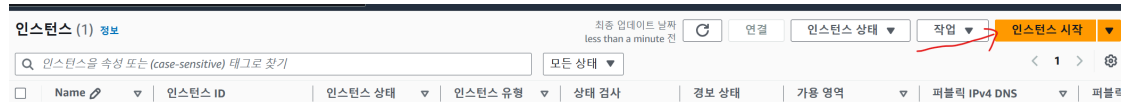
제일 먼저 지역이 서울로 잘 설정되어 있는지 확인해줍니다.



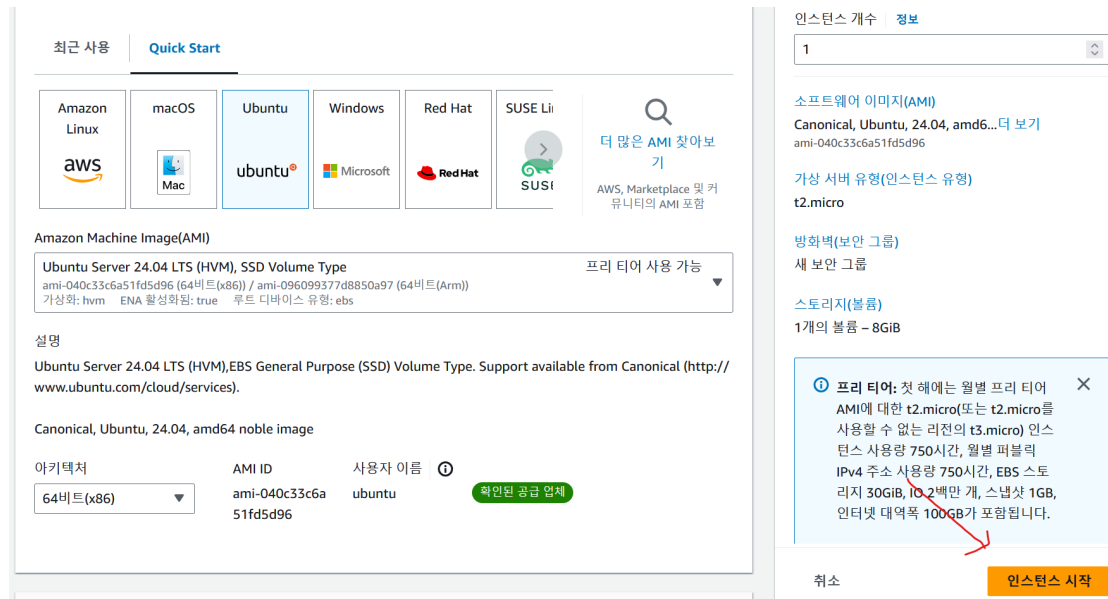
EC2를 검색한 후 들어가 줍니다.



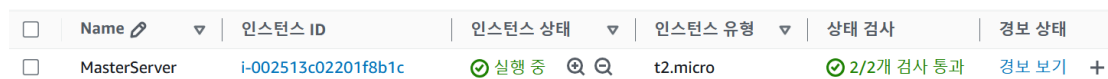
왼쪽 탭에서 인스턴스를 선택합니다.



인스턴스 시작을 눌러줍니다.



사용할 운영체제만 설정 해준 후, 인스턴스 시작을 눌러줍니다.



정상적으로 생성된 모습을 확인해볼 수 있습니다.

<input checked="" type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태
<input checked="" type="checkbox"/>	MasterServer	i-002513c02201f8b1c	실행 중	t2.micro	2/2개 검사 통과	경보 보기 +

**i-002513c02201f8b1c (MasterServer)**

세부 정보

상태 및 경보

모니터링

보안

네트워킹

스토리지

태그

▼ 인스턴스 요약 정보

인스턴스 ID

i-002513c02201f8b1c

퍼블릭 IPv4 주소

3.35.235.211 | [개방 주소법](#)

인스턴스 상태

실행 중

프리미엄 인스턴스

아니요

IPv6 주소

-

프라이빗 IP 주소

10.0.0.1

인스턴스 유형

t2.micro

태그

태그 없음

인스턴스 옆 체크박스를 선택하면 아래에 추가적인 정보가 보입니다.  
여기서 보안을 눌러줍니다.

**i-002513c02201f8b1c (MasterServer)**

세부 정보

상태 및 경보

모니터링

보안

네트워킹

스토리지

태그

▼ 보안 세부 정보

IAM 역할

-

소유자 ID

390402573868

보안 그룹

sg-063217d16fb4c1e6e (main\_sg-DEV)

그런 다음 보안 그룹(firewall)의 sg-...으로 보이는 파란 글씨를 클릭해줍니다.

인바운드 규칙

아웃바운드 규칙

공유 - 신규

VPC 연결 - 신규

태그

인바운드 규칙 (2)

태그 관리

인바운드 규칙 편집

검색

< 1 >

<input type="checkbox"/>	Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위	소스
--------------------------	------	-------------	-------	----	------	-------	----

인바운드 규칙 편집을 눌러줍니다.

**인바운드 규칙 정보**

보안 그룹 규칙 ID	유형 정보	프로토콜 정보	포트 범위 정보	소스 정보	설명
sgr-0520ba5796eb1f5d1	SSH	TCP	22	사용자 지정	Q
sgr-05a47005ebf30c9d7	사용자 지정 TCP	TCP	8000	사용자 지정	Q

규칙 추가

다음과 같이 보안그룹을 설정해줍니다.

22번 포트는 컴퓨터에 접속하기 위해, 8000번 포트는 사용자가 웹 서버에 접속하기 위해 열어줍니다.

## sg-063217d16fb4c1e6e - main\_sg-DEV

### 세부 정보

보안 그룹 이름

main\_sg-DEV

소유자

390402573868

보안 그룹 ID

sg-063217d16fb4c1e6e

인바운드 규칙 수

2 권한 항목

인바운드 규칙

아웃바운드 규칙

공유 - 신규

VPC 연결 - 신규

태그

### 인바운드 규칙 (2)

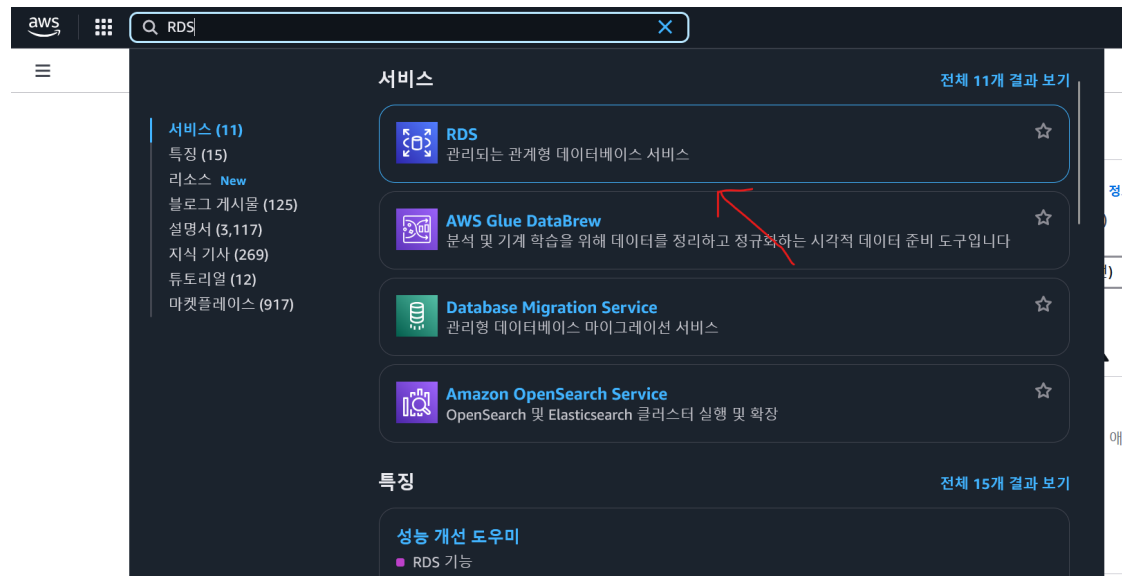
Q 검색

<input type="checkbox"/>	Name ▾	보안 그룹 규칙 ID ▾	IP 버전 ▾	유형
<input type="checkbox"/>	-	sgr-0520ba5796eb1f5d1	IPv4	SSH
<input type="checkbox"/>	-	sgr-05a47005ebf30c9d7	IPv4	사용

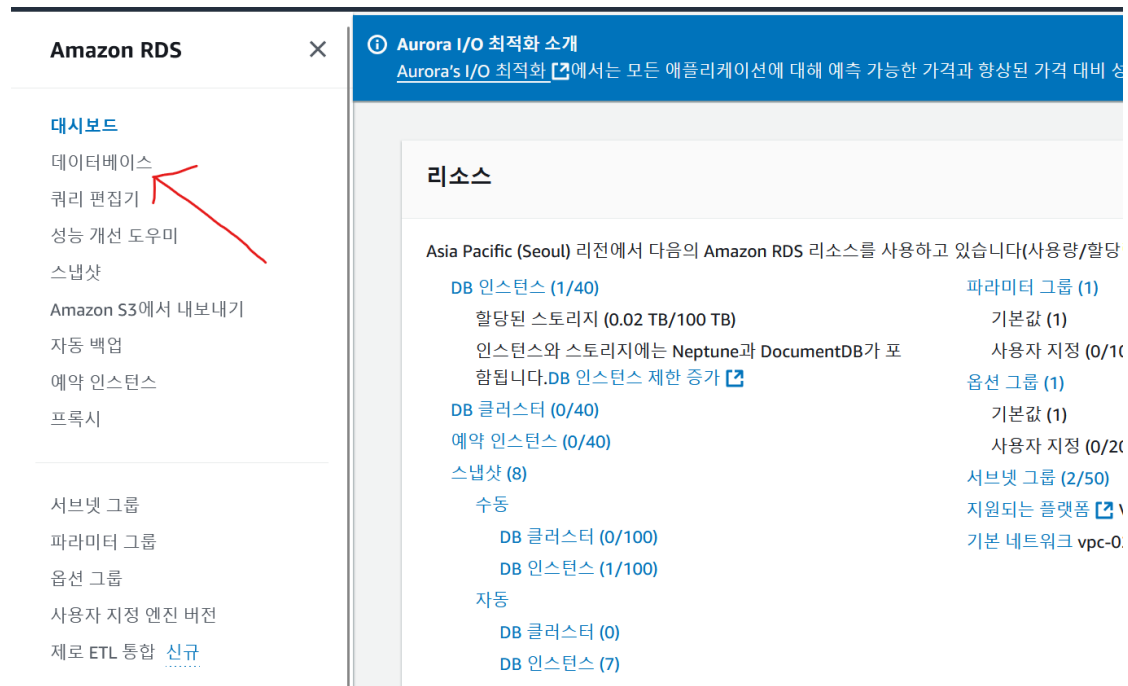
보안 그룹 ID만 다른 곳에 복사를 해둔 후, 마무리합니다.

## AWS RDS MySQL 생성

이제 MySQL Server를 생성합니다.



RDS를 검색한 후 들어가 줍니다.



왼쪽에 보이는 탭에서 데이터베이스를 선택합니다.

RDS > 데이터베이스

**업그레이드 중 가동 중지 시간을 최소화하기 위해 블루/그린 배포 생성 고려**

업그레이드하는 동안 Amazon RDS 블루/그린 배포 사용을 고려하고 가동 중지 시간을 최소화하는 것이 좋습니다. 블루/그린 배포는 프로덕션 데이터베이스 변경을 위한 스테이징 환경을 제공합니다. [RDS 사용 설명서](#) [Aurora 사용 설명서](#)

**EC2 데이터베이스에서 RDS로 간편한 경로 동종 데이터 마이그레이션**

Amazon RDS 콘솔은 AWS DMS 기반의 통합된 동종 데이터 마이그레이션을 통해 EC2 데이터베이스에서 동등한 RDS 데이터베이스로 간단하고 성능 좋은 데이터 마이그레이션을 활용합니다. 시작하려면 기존 RDS 데이터베이스를 선택하고 작업 메뉴에서 **EC2 데이터베이스에서 데이터 마이그레이션(들)** 선택하세요. 지원되는 엔진 유형과 기능 제한을 반드시 확인하세요. [자세히 알아보기](#)

데이터베이스 (1) 그룹 리소스 수정 작업 S3에서 복원 **데이터베이스 생성**

Q 데이터베이스(들) 기준으로 필터링

데이터베이스 생성을 눌러줍니다.

RDS > 데이터베이스 생성

## 데이터베이스 생성 정보

### 데이터베이스 생성 방식 선택

☒ **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**



☐ PostgreSQL



☐ MariaDB

☐ Oracle

그러면 등장하는 설정 화면. 엔진은 MySQL을 선택해줍니다.



## 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](#)

프리티어(무과금)를 기준으로 선택하겠습니다.

## 설정

### DB 인스턴스 식별자 [정보](#)

DB 인스턴스 이름을 입력하세요. 이름은 현재 AWS 리전에서 AWS 계정이 소유하는 모든 DB 인스턴스에 대해 고유해야 합니다.

soongsil-student-cafeteria-scm

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

#### 자격 증명 관리

AWS Secrets Manager를 사용하거나 마스터 사용자 자격 증명을 관리할 수 있습니다.

### ☐ AWS Secrets Manager에서 관리 - 가장 뛰어난 안정성

RDS는 자동으로 암호를 생성하고 AWS Secrets Manager를 사용하여 전체 수명 주기 동안 암호를 관리합니다.

### ☒ 자체 관리

사용자가 암호를 생성하거나 RDS에서 암호를 생성하고 사용자가 관리할 수 있습니다.

### ☐ 암호 자동 생성

Amazon RDS에서 자동으로 암호를 생성하거나 사용자가 직접 암호를 지정할 수 있습니다.

### 마스터 암호 [정보](#)

●●●●●●●●

### Password strength

Strong

최소 제약 조건: 8자 이상의 인쇄 가능한 ASCII 문자를 사용합니다. / ' " @ 기호는 포함할 수 없습니다.

master 사용자(user)의 이름과 비밀번호를 설정해줍니다.

#### 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 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 지정 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

#### 새 VPC 보안 그룹 이름

새 VPC 보안 그룹 이름 입력

#### 가용 영역 정보

ap-northeast-2c

#### 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](#)

아래로 내려가서 연결 부분에서 보안그룹을 새로 생성으로 설정해줍니다.

이름은 아무거나, 가용 영역은 ap-northeast-2c로 설정하겠습니다.

#### 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.

❗ 귀하는 AWS 서비스와 함께 사용하는 타사 제품 또는 서비스 일체에 대해 필요한 모든 권리를 보유할 책임이 있습니다.

취소

데이터베이스 생성

데이터 베이스 생성을 클릭합니다.

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

Q 데이터베이스를(들) 기준으로 필터링

DB 식별자

상태

역할

엔진

리전 및 ...

크기

권장 사항

CPU

soongsil-student-cafeteria-scm

2 생성 중

인스턴스

MySQL Co...

-

db.t4g.mi...


-

정상적으로 생성된 모습을 확인해볼 수 있습니다.

위의 보이는 화면에서 파란 글씨의 식별자를 선택합니다.

## 연결 및 보안

### 엔드포인트 및 포트

엔드포인트  
 terraform-2024111601081838430000000  
 3.c3um20ke28jo.ap-northeast-2.rds.amazonaws.com  
 포트  
 3306

### 네트워킹


가용 영역  
 ap-northeast-2c  
 VPC  
[vpc-DEV \(vpc-0fef6489f17246149\)](#)  
 서브넷 그룹  
 rds-subnet-group  
 서브넷  
[subnet-08c40b9f87c537c76](#)  
[subnet-0b087b3d2649111ed](#)  
 네트워크 유형  
 IPv4


### 보안

VPC 보안 그룹  
[rds-sg-DEV \(sg-068cdd7175246cba7\)](#)  
 **활성**  
 퍼블릭 액세스 가능  
 아니요  
 인증 기관 정보  
 rds-ca-rsa2048-g1  
 인증 기관 날짜  
 May 21, 2061, 02:28 (UTC+09:00)  
 DB 인스턴스 인증서 만료 날짜  
 November 16, 2025, 10:10 (UTC+09:00)

엔드포인트만 복사를 해줍니다.  
 그런 다음 보안 그룹을 선택합니다.

### 보안 그룹 (1) 정보

 Find resources by attribute or tag

[sg-068cdd7175246cba7](#)  Clear filters


<input type="checkbox"/>	Name	보안 그룹 ID	보안 그룹 이름
<input type="checkbox"/>	RDS-Security-Group	<a href="#">sg-068cdd7175246cba7</a>	rds-sg-DEV

보안 그룹 ID를 선택합니다.

[EC2](#) > [보안 그룹](#) > [sg-068cdd7175246cba7 - rds-sg-DEV](#)

**sg-068cdd7175246cba7 - rds-sg-DEV** 작업 ▼


**세부 정보**

보안 그룹 이름  rds-sg-DEV	보안 그룹 ID  sg-068cdd7175246cba7	설명  Managed by Terraform	VPC ID  <a href="#">vpc-0fef6489f17246149</a>
소유자  390402573868	인바운드 규칙 수 1 권한 항목	아웃바운드 규칙 수 1 권한 항목	

[인바운드 규칙](#) | [아웃바운드 규칙](#) | [공유 - 신규](#) | [VPC 연결 - 신규](#) | [태그](#)

**인바운드 규칙 (1)**

 태그 관리 **인바운드 규칙 편집**

 검색

<input type="checkbox"/>	Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위	소스
--------------------------	------	-------------	-------	----	------	-------	----

인바운드 규칙 편집을 눌러줍니다.

**인바운드 규칙 정보**

보안 그룹 규칙 ID: gr-0ca966325d9f13b60

유형: MySQL/Aurora

프로토콜: TCP

포트 범위: 3306

소스: 사용자 지정

설명 - 선택 사항

규칙 추가

규칙 목록:

- :/16
- :/32
- :/48
- :/64
- 보안 그룹
- rds-sg-DEV | sg-068cdd7175246cba7
- RDS-Security-Group
- default | sg-02d89f64998b41375
- main\_sg-DEV | sg-063217d16fb4c1e6e
- 협두사 목록
- com.amazonaws.ap-northeast-2.ec2-instance-connect |

미리 보기

규칙 저장

아까 복사해둔 보안 그룹 ID를 찾아서 선택해준 후 규칙을 저장합니다.

모든 자원할당 과정이 끝났습니다.

이제 EC2에 접속해서, 데이터베이스 설정과 application 실행 과정을 진행하겠습니다

## 데이터베이스 설정 및 Application 실행

대시보드 × 인스턴스 (1/1) 정보 최종 업데이트 날짜 less than a minute 전 연결 인스턴스 상태 ▼

EC2 글로벌 보기 이벤트

인스턴스 상태를 속성 또는 (case-sensitive) 태그로 찾기 모든 상태 ▼

인스턴스 상태 = running X 필터 지우기

<input checked="" type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역
<input checked="" type="checkbox"/>	MasterServer	i-002513c02201f8b1c	실행 중	t2.micro	2/2개 검사 통과	경보 보기 +	ap-northeast-2a

위의 보이는 화면에서 인스턴스 ID를 선택합니다.

i-002513c02201f8b1c (MasterServer)에 대한 인스턴스 요약 정보 최종 업데이트 날짜 less than a minute 전 연결 인스턴스 상태 ▼ 작업 ▼

<b>인스턴스 ID</b> i-002513c02201f8b1c <b>IPv6 주소</b> - <b>호스트 이름 유형</b> IP 이름: ip-10-0-1-98.ap-northeast-2.compute.internal <b>프라이빗 리소스 DNS 이름 응답</b> - <b>자동 할당된 IP 주소</b> 3.35.235.211 [퍼블릭 IP]	<b>퍼블릭 IPv4 주소</b> 3.35.235.211   <a href="#">개방 주소법</a> <b>인스턴스 상태</b> 실행 중 <b>프라이빗 IP DNS 이름(IPv4만 해당)</b> ip-10-0-1-98.ap-northeast-2.compute.internal <b>인스턴스 유형</b> t2.micro <b>VPC ID</b> vpc-0fef6489f17246149 (vpc-DEV)	<b>프라이빗 IPv4 주소</b> 10.0.1.98 <b>퍼블릭 IPv4 DNS</b> ec2-3-35-235-211.ap-northeast-2.compute.amazonaws.com   <a href="#">개방 주소법</a> <b>탄력적 IP 주소</b> - <b>AWS Compute Optimizer 찾기</b> 권장 사항을 위해 AWS Compute Optimizer에 옵트인합니다.   <a href="#">자세히 알아보기</a>
---	--	--

그러면 보이는 화면. 퍼블릭 IPv4 DNS만 복사한 후, 연결을 눌러줍니다.

## 인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-002513c02201f8b1c (MasterServer)에 연결

EC2 인스턴스 연결

Session Manager

SSH 클라이언트

EC2 직렬 콘솔

인스턴스 ID

i-002513c02201f8b1c (MasterServer)

연결 유형

☒ EC2 Instance Connect를 사용하여 연결

Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ EC2 인스턴스 연결 엔드포인트를 사용하여 연결

프라이빗 IPv4 주소 및 VPC 엔드포인트가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.

☒ 퍼블릭 IPv4 주소

3.35.235.211

☐ IPv6 주소

-

사용자 이름

인스턴스를 시작하는 데 사용되는 AMI에 정의된 사용자 이름을 입력합니다. 사용자 지정 사용자 이름을 정의하지 않은 경우 기본 사용자 이름인 ubuntu(들) 사용합니다.

Q ubuntu

X

**참고:** 대부분의 경우 기본 사용자 이름 ubuntu(는) 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하세요.

취소

연결

이어지는 화면에서 다시 연결을 선택합니다.

```
aws [일트45]
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1072-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Nov 21 12:20:24 UTC 2024

System load:  0.0               Processes:    99
Usage of /:   31.4% of 7.57GB    Users logged in: 0
Memory usage: 26%              IPv4 address for eth0: 10.0.1.98
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

5 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Nov 21 06:23:07 2024 from 39.118.248.251
ubuntu@ip-10-0-1-98:~$
```

EC2 인스턴스에 접속했습니다.

이제 필요한 프로그램들을 설치합니다.

```

sudo apt update
sudo apt install -y mysql-client python-is-python3 python3-pip
pip install flask pymysql

git clone https://github.com/cryscham123/db_gimal_work.git work # source code

```

그런 다음, 다음 명령어를 실행해서 데이터베이스를 설정해줍니다.

<Copied RDS Endpoint>는 복사해둔 RDS의 endpoint, 나머지 <RDS User Name>과 <RDS Password>는 RDS 생성시 설정한 것을 입력해줍니다.

```

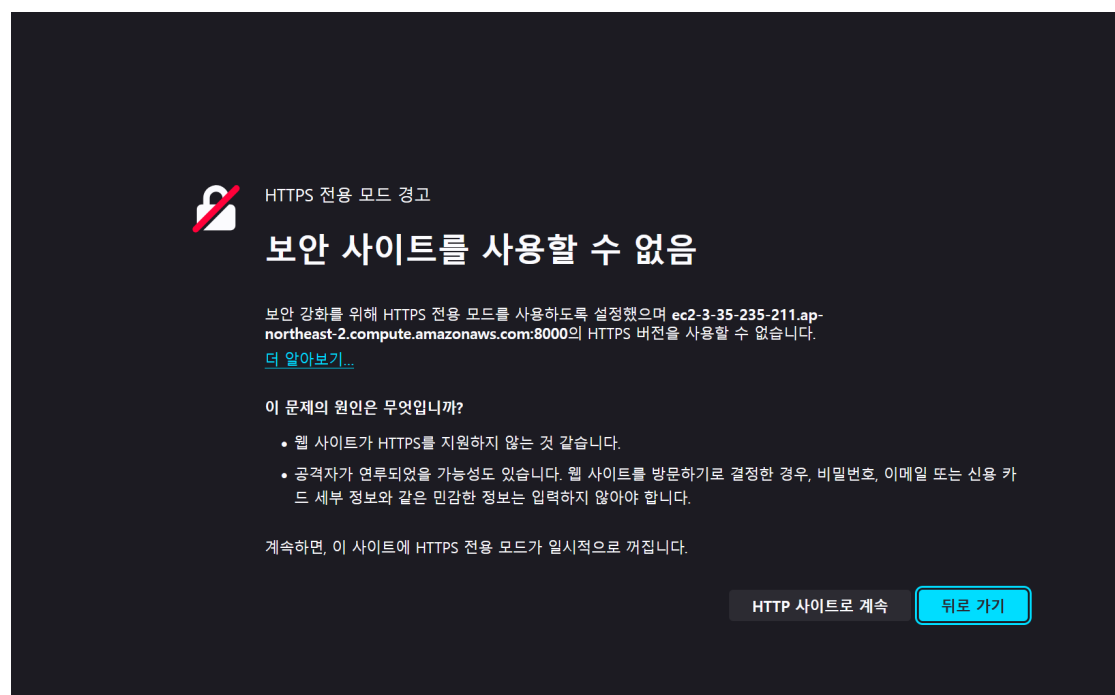
export RDS_ENDPOINT=<Copied RDS Endpoint>
export RDS_USERNAME=<RDS User Name>
export RDS_PASSWORD=<RDS Password>
mysql -h $RDS_ENDPOINT -u $RDS_USERNAME -p $RDS_PASSWORD < work/sql/create.sql
mysql -h $RDS_ENDPOINT -u $RDS_USERNAME -p $RDS_PASSWORD < work/sql/auth.sql
mysql -h $RDS_ENDPOINT -u $RDS_USERNAME -p $RDS_PASSWORD < work/sql/insert.sql

```

이제 다음 명령어를 실행해서 application을 실행합니다.

```
python work/app/index.py
```

이제 브라우저에서 복사해둔 퍼블릭 IPv4 DNS의 8000번 포트로 접속하면, 아래와 같은 화면이 나옵니다.



네트워크 암호화 설정을 안해준 관계로 해당 경고창이 뜹니다.

그냥 HTTP 사이트로 계속 진행하겠습니다.

# 송실대학교 학생식당

[메뉴 레시피](#) [재료 정보](#) / [구입 주문 목록](#) [공급자](#)

2024 . 11 . 21 . 📅

2024-11-21

메뉴 이름	시간	코너	가격		
<a href="#">두부조림</a>	아침	뚝배기 코너	1000 원	<input type="button" value="수정"/>	<input type="button" value="삭제"/>
<a href="#">뚝배기삼겹살김치찌개</a>	점심	뚝배기 코너	5000 원	<input type="button" value="수정"/>	<input type="button" value="삭제"/>
<a href="#">바베큐치킨뚝배기</a>	점심	양식 코너	5000 원	<input type="button" value="수정"/>	<input type="button" value="삭제"/>
<a href="#">등심돈까스</a>	점심	양식 코너	5000 원	<input type="button" value="수정"/>	<input type="button" value="삭제"/>

정상적으로 실행되는 모습.



# DB administration

## Backup

**백업**  
자동 백업  
활성화됨(7 Days)  
스냅샷으로 태그 복사  
비활성화됨  
백업 대상  
AWS 클라우드 (Asia Pacific (Seoul))

최근 복원 시간  
November 22, 2024, 08:40 (UTC+09:00)  
백업 기간  
07:00-09:00 UTC (GMT)

리전으로 복제  
-  
복제된 자동 백업  
-

스냅샷 (7)

복원

제거

스냅샷 생성

< 1 >

<input type="checkbox"/>	스냅샷 이름	스냅샷 생성 시간	상태	스냅샷 유형
<input type="checkbox"/>	<a href="#">rdsterraform-20241116010818384300000003-2024-11-16-01-11</a>	November 16, 2024, 10:16 (UTC+09:00)	✔ 사용 가능	자동
<input type="checkbox"/>	<a href="#">rdsterraform-20241116010818384300000003-2024-11-16-07-06</a>	November 16, 2024, 16:06 (UTC+09:00)	✔ 사용 가능	자동
<input type="checkbox"/>	<a href="#">rdsterraform-20241116010818384300000003-2024-11-17-07-06</a>	November 17, 2024, 16:06 (UTC+09:00)	✔ 사용 가능	자동
<input type="checkbox"/>	<a href="#">rdsterraform-20241116010818384300000003-2024-11-18-07-06</a>	November 18, 2024, 16:06 (UTC+09:00)	✔ 사용 가능	자동
<input type="checkbox"/>	<a href="#">rdsterraform-20241116010818384300000003-2024-11-19-07-06</a>	November 19, 2024, 16:06 (UTC+09:00)	✔ 사용 가능	자동
<input type="checkbox"/>	<a href="#">rdsterraform-20241116010818384300000003-2024-11-20-07-06</a>	November 20, 2024, 16:06 (UTC+09:00)	✔ 사용 가능	자동
<input type="checkbox"/>	<a href="#">rdsterraform-20241116010818384300000003-2024-11-21-07-06</a>	November 21, 2024, 16:06 (UTC+09:00)	✔ 사용 가능	자동

RDS 기본 설정으로 백업이 설정되어 있습니다.

만약 백업 파일을 수동으로 생성하고 싶다면 EC2 인스턴스에서 다음 명령어를 실행합니다.

```
mysqldump -h $RDS_ENDPOINT -u $RDS_USERNAME -p$RDS_PASSWORD > backup.sql
```

만약 transaction log를 수동으로 생성하고 싶다면 다음의 단계를 따릅니다.

```
CALL mysql.rds_set_configuration('binlog retention hours', 24);
```

우선 mysql에 접속해서 log파일의 보관기간을 수동으로 늘려줍니다.

```
SHOW BINARY LOGS;
```

그런 다음 log file의 파일명을 확인합니다.

```
mysqlbinlog \  
  --read-from-remote-server \  
  --host=<RDS_ENDPOINT> \  
  --port=3306 \  
  --user admin \  
  --password \  
  --raw \  
  --verbose \  
  --result-file=<path to store> \  
<name of log file>
```

위의 명령어를 통해 log file을 생성할 수 있습니다.

```
mysqlbinlog \  
  --base64-output=DECODE-ROWS \  
  --verbose \  
  <stored path> > binlog.sql
```

사용하기 좋게 sql 형식으로 변환해줍니다.

## Recovery

AWS 콘솔에서 복구를 선택할 수 있지만, 수동으로 생성한 백업 파일을 이용해서 복구할 수 있습니다. 다음의 명령어를 새로운 환경에서 실행합니다.

```
cat << EOF | mysql  
CREATE DATABASE SOONGSIL_STUDENT_CAFETERIA_SCM;  
EOF  
mysql SOONGSIL_STUDENT_CAFETERIA_SCM < backup.sql  
mysql SOONGSIL_STUDENT_CAFETERIA_SCM < binlog.sql
```

log file은 AWS RDS 환경이 아니면 에러가 발생할 수 있습니다. 하지만 AWS RDS 환경에서 log file을 직접적으로 사용할 권한이 없습니다. 결론적으로, 우리의 구현에서 log file은 사용할 수 없습니다.

## Database Security

### 1. firewall

AWS 보안 그룹을 이용해서 RDS 인스턴스에 대한 접근을 제한했습니다.

### 2. Encryption store

#### 암호화

☒ 암호화 활성화

지정된 인스턴스를 암호화하려면 이 옵션을 선택합니다. AWS Key Management Service 콘솔을 사용하여 마스터 키 ID와 별칭이 생성된 후 해당 항목이 목록에 표시됩니다. [정보](#)

AWS KMS 키 [정보](#)

(default) aws/rds

RDS 인스턴스 기본 설정으로 암호화 설정을 활성화했습니다.