



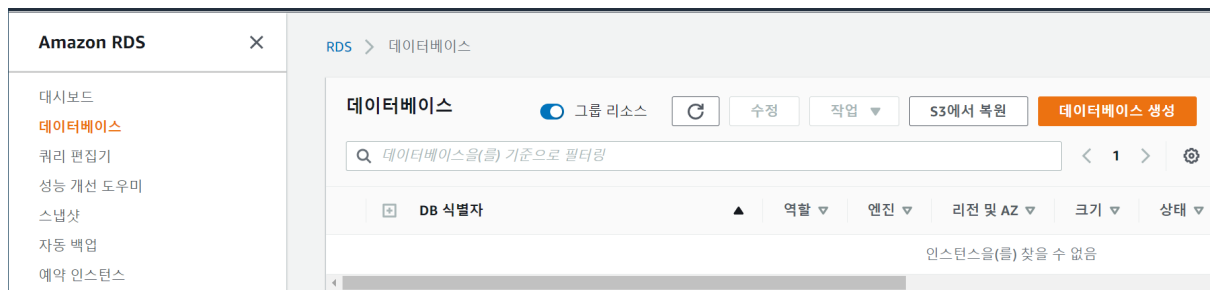
AWS RDS(DB Server)-PostgreSQL

1. RDS 생성하기

- 하단 AWS RDS 페이지

 <https://ap-northeast-2.console.aws.amazon.com/rds/home?region=ap-northeast-2>

- 데이터베이스 생성 클릭



- 설정

- 데이터베이스 이름 : 원하는 이름
- 엔진 유형
 - PostgreSQL
- 템플릿(꼭 프리 티어로 설정) → 아니면 과금 낼지도 모름
- 설정 → 나중에 다른 곳에 연동할 때 필요한 정보
 - 마스터 사용자 이름 : postgres (기본값 - 수정 가능)

- 마스터 암호 : 8자리(기억)
- 연결
 - 퍼블릭 액세스 : 예
- 생성하기

서비스

서비스, 기능, 블로그, 설명서 등을 검색합니다.

[Alt+S]

EC2

S3

RDS

PostgreSQL

Oracle

Microsoft SQL Server

버전

PostgreSQL 13.4-R1

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](#)

설정

DB 인스턴스 식별자 [정보](#)

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

database-1

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.

postgres

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. 다음을 포함할 수 없습니다. /(슬래시), '(작은따옴표), "(큰따옴표) 및 @(앳 기호).

암호 확인 [정보](#)

.....

연결

Virtual Private Cloud(VPC) [정보](#)

이 DB 인스턴스의 가상 네트워킹 환경을 정의하는 VPC.

Default VPC (vpc-05fee53b394106bb5)

해당 DB 서브넷 그룹이 있는 VPC만 나열됩니다.

데이터베이스를 생성한 후에는 VPC를 변경할 수 없습니다.

서브넷 그룹 [정보](#)

선택한 VPC에서 DB 인스턴스가 어떤 서브넷과 IP 범위를 사용할 수 있는지를 정의하는 DB 서브넷 그룹.

기본값

퍼블릭 액세스 [정보](#)

예

VPC 외부의 Amazon EC2 인스턴스 및 디바이스는 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 VPC 내부의 EC2 인스턴스 및 디바이스를 지정하는 하나 이상의 VPC 보안 그룹을 선택하세요.

아니요

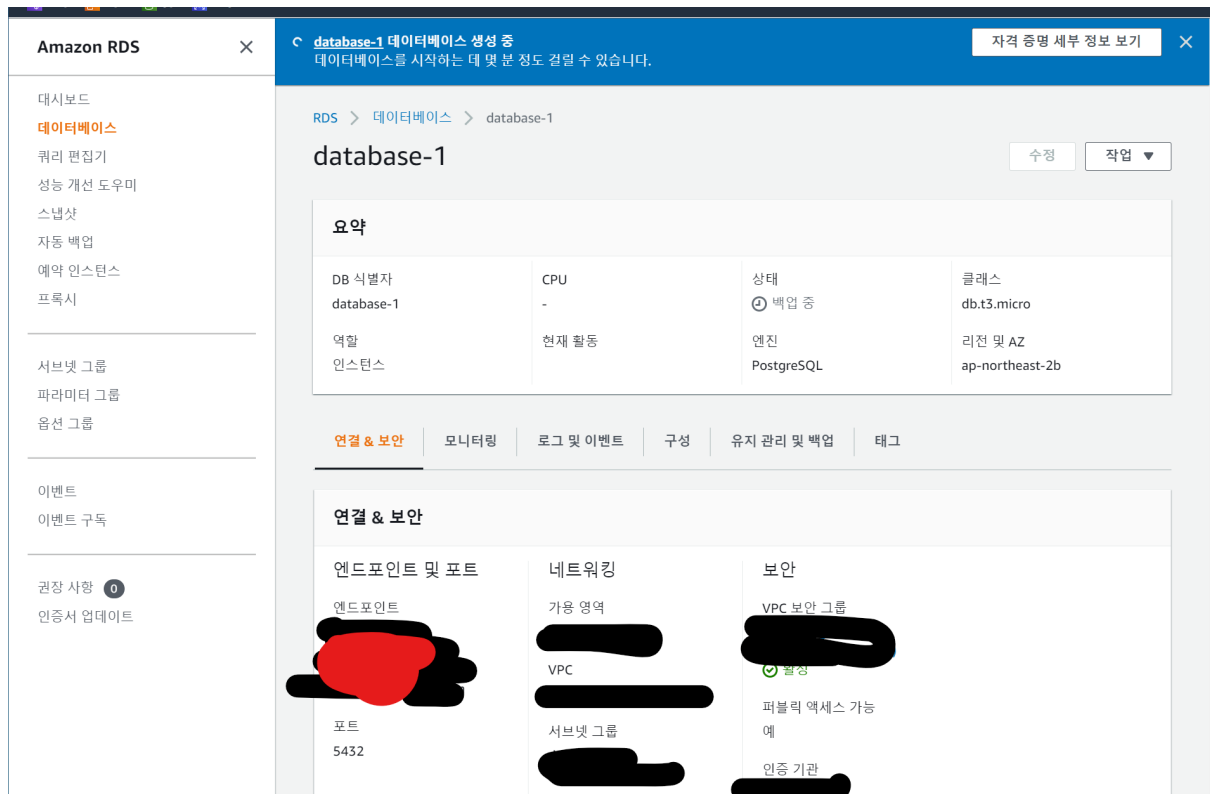
RDS는 데이터베이스에 퍼블릭 IP 주소를 할당하지 않습니다. VPC 내부의 Amazon EC2 인스턴스 및 디바이스만 데이터베이스에 연결할 수 있습니다.

2. DBeaver 에 RDS 연결하기

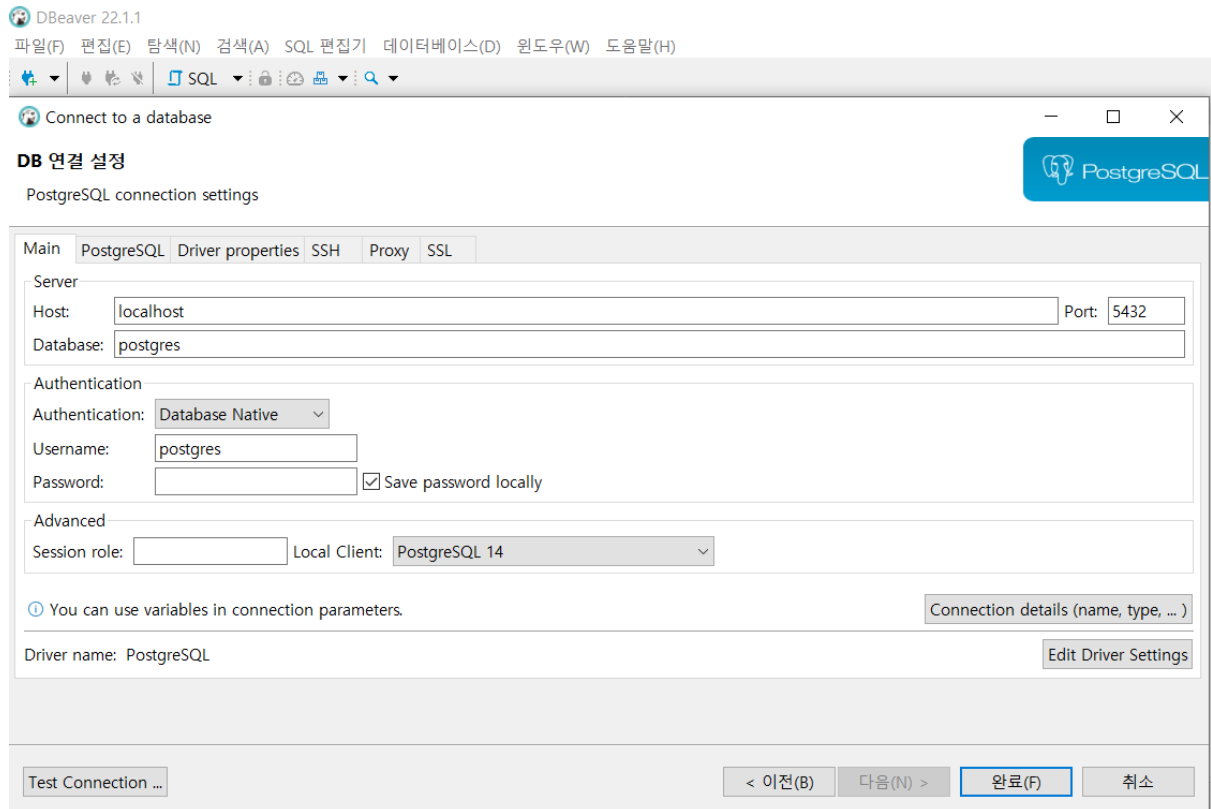
- 방금 생성한 DB 클릭
- **엔드포인트**와 **포트** 번호 기억하기

AWS RDS(DB Server)-PostgreSQL

4



- DBeaver 실행
- 데이터베이스 추가
- PostgreSQL 클릭
- Host : 위에서 저장한 엔드포인트
- Port : 위에서 저장한 포트 번호
- Username: RDS 생성할 때, Username
- Password: RDS 생성할 때, Password
- Test Connection → 성공 → 완료
- 실패 → 잘못 적은거 확인



3. Dump Data 넣기

```
// Dump Data 있는 폴더 위치까지 이동 : ex. 바탕화면 example 폴더에 있는 경우
// window terminal 실행
cd Desktop/example

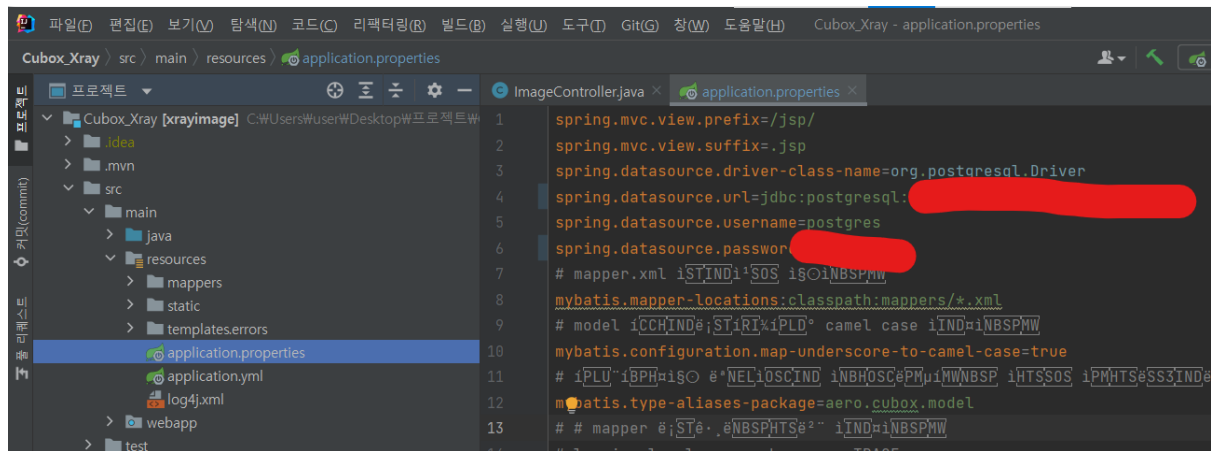
// postgresql 접속 명령어(psql)
// psql -h (접속 호스트 IP 주소) -U (설정한 유저 네임) -p (설정한 로컬 포트 번호) -f (dump 파일 이름)
psql -h (엔드포인트) -U postgres -p (포트 번호: 5432) -f (파일 이름: dump.sql)
```

4. Spring boot 에 RDS 연결하기

- Github 에서 코드 다운로드
- src → main → resources → application.properties (파일 경로)

```
// 아래 코드 수정
spring.datasource.url=jdbc:postgresql://(엔드포인트):(포트번호)/(데이터베이스 이름 : postgres)
```

```
spring.datasource.username=postgres(RDS 만들 때, 정한 username)
spring.datasource.password=(password)
```



- 하고 실행하면, 성공

참고자료

[AWS] EC2와 RDS. 연동하기

새로 시작한 JELLOgram 을 AWS의 EC2 서버로 AWS의 RDS에 배포해보자! 그러기 위해서 AWS의 EC2, RDS 무엇인지에 대해 알아보고자 한다. 요약 : EC2, RDS, S3 EC2 : AWS의 가상 server RDS :

 https://vlog.io/@c_hyun403/AWS

AWS 