

AWS Database

Data Store

❖ 데이터 저장소 종류

✓ RDBMS

- 기존 애플리케이션, 전사적 자원 관리(ERP), 고객 관계 관리(CRM), 전자 상거래
- Amazon Aurora, Amazon RDS, Amazon Redshift

✓ Key-Value Database

- 높은 트래픽의 웹 애플리케이션, 전자 상거래 시스템, 게임 애플리케이션
- Amazon DynamoDB

✓ In-Memory DB

- ✓ Caching, 세션 관리, 게임 순위표, 지리 공간 애플리케이션
- ✓ Amazon ElastiCache, Amazon Memory DB for Redis

✓ Document

- 콘텐츠 관리, 카탈로그, 사용자 프로필
- Amazon Document DB(Mongo DB 호환)

✓ Wide Column

- 장비 관리, 플릿 관리(소프트웨어 플랫폼을 통해 물리적 자산과 디지털 자산으로 구성된 엔터프라이즈의 플리트를 감독하는 것) 및 경로 최적화에 사용하는 대규모 산업용 앱
- Amazon Keyspaces

Data Store

❖ 데이터 저장소 종류

✓ Graph

- 부정 탐지, 소셜 네트워킹, 추천 엔진
- Amazon Neptune

✓ Time Series

- 사물 인터넷(IoT) 애플리케이션, DevOps, 산업용 텔레메트리
- Amazon Timestream

✓ Ledger

- 레코드 시스템, 공급망, 등록, 은행 거래
- Amazon Ledger Database Services(QLDB)

RDS

❖ Amazon RDS

- ✓ Amazon Relational Database Service(Amazon RDS)는 관계형 데이터베이스 6 종류의 제품을 클라우드에 최적화된 상태로 제공하는 서비스
- ✓ Amazon Aurora 외에도 PostgreSQL(Aurora 버전 제공) MySQL(Aurora 버전 제공), MariaDB, Oracle Database, SQL Server, IBM Db2를 지원하며 메모리, 성능, I/O 등이 최적화된 데이터베이스 인스턴스를 제공
- ✓ Aurora 버전은 기존 소스들을 AWS에 맞게 수정한 버전
- ✓ 데이터베이스의 제공 방식은 EC2와 비슷한데 VPC 상에 인스턴스 형태로 구축하며 다양한 인스턴스 클래스를 사용할 수 있음
- ✓ EC2와 다른 점은 RDS는 Managed Service이며 업데이트 등의 관리가 AWS에 의해 자동으로 이루어진다는 점
- ✓ 백업과 같이 번거로운 관리 작업도 자동화되어 있어 관리자가 수행할 필요가 없음
- ✓ AWS Database Migration Server(DMS)를 사용하면 기존 데이터베이스를 이전하거나 복제하는 것도 가능

RDS

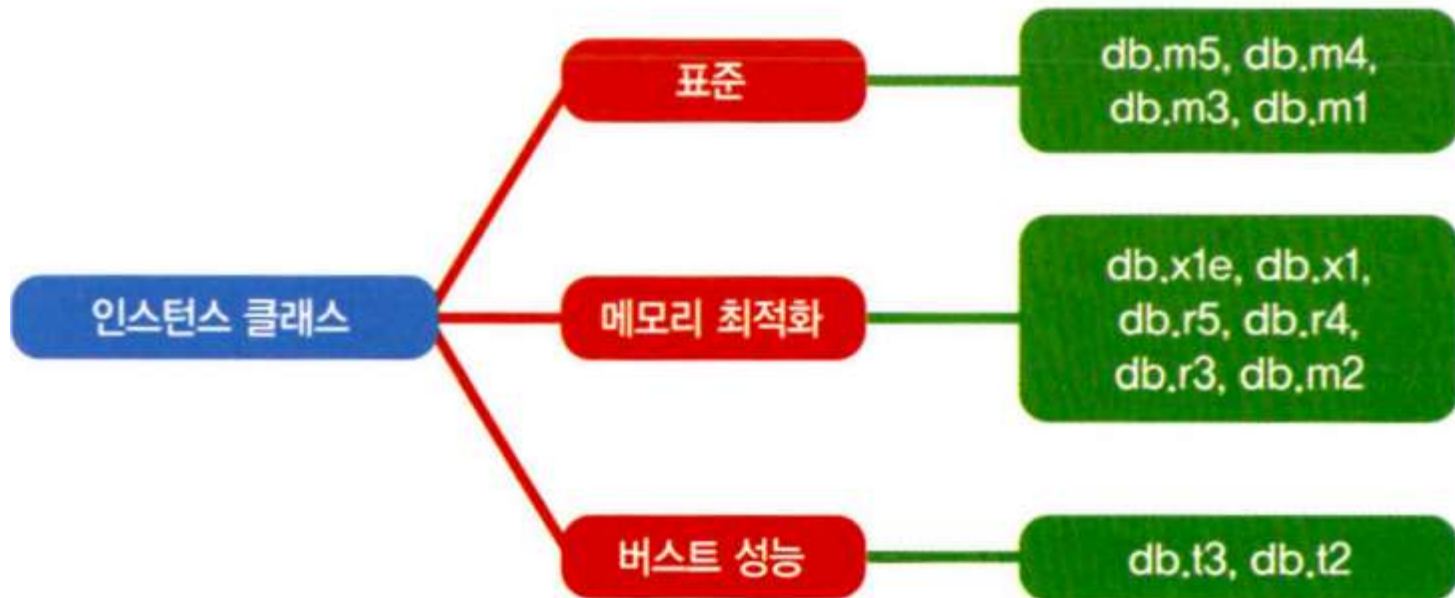
❖ 인스턴스 클래스

- ✓ RDS는 EC2와 마찬가지로 인스턴스 형식으로 사용
- ✓ 인스턴스 클래스는 표준, 메모리 최적화, 버스트 성능의 3종류가 있으며 클래스에 따라 micro, small, medium, large, xlarge, 2xlarge, 4xlarge, 8xlarge, 16xlarge 크기를 제공하는데 인스턴스 클래스에 따라 지원하지 않는 DBMS나 버전도 있음
- ✓ 데이터베이스 인스턴스는 VPC에 설치
- ✓ 주요 인스턴스 클래스

용도	인스턴스 클래스	내용
표준	db.m5 등	범용적인 인스턴스 클래스이다.
메모리 최적화	db.x1e	메모리를 많이 사용하는 애플리케이션에 최적화된 인스턴스 클래스로, 일부 리전에서만 제공된다. 한국 리전에서는 이용할 수 있다.
	db.x1	메모리를 많이 사용하는 애플리케이션에 최적화된 인스턴스 클래스이다. 메모리 최대치가 db.x1e의 절반이다.
	db.r5 등	네트워크와 EBS의 성능을 강화한다. AWS Nitro System을 사용한다.
버스트 성능	db.t3 등	CPU의 최대 사용률까지 버스트 성능으로 사용할 수 있다.

RDS

- ❖ 인스턴스 클래스
 - ✓ 인스턴스 종류



RDS

❖ RDS 요금

- ✓ RDS의 요금은 기본적으로 EC2와 비슷하지만 DBMS에 따라 인스턴스 요금이 다른 것이 특징
- ✓ DBMS는 무료와 유료와 있고 유료 DBMS는 AWS에서도 라이선스 사용료가 부과되는데 라이선스 요금은 인스턴스 요금에 포함되어 있음
- ✓ $\text{요금} = \text{스토리지(확보하고 있는 용량) 요금} + \text{데이터베이스 인스턴스(가동한 시간당으로 부과되는데 다중 AZ로 구성하여 이중화를 구축하면 2배 + 라이선스 비용) 요금} + \text{백업 스토리지(스냅샷에 부과되는 요금으로 데이터베이스 스토리지 크기의 100% 넘는 경우만 발생) 요금} + \text{통신료(VPC 내의 EC2 인스턴스 와 통신하는 경우 부과되지 않음)}$

RDS

❖ RDS 장점 과 단점

✓ 장점

- RDS의 장점은 Managed Service라는 점으로 AWS가 업데이트 등을 관리해 주기 때문에 따로 관리하지 않아도 됨
- 간단하게 데이터베이스를 생성할 수 있으며 On-premise에서 데이터를 이전하기 쉽고 소프트웨어를 수정하지 않아도 이전이 가능
- EC2와 연동하기 쉽고 같은 네트워크 내에 있다면 통신료도 무료

✓ 단점

- 사용자가 자유롭게 사용할 수 없다는 점으로 제공하고 있는 DBMS의 종류나 버전이 한정되어 있으며 AWS가 업데이트를 자동으로 수행하여 편리하지만 오히려 업데이트하면 곤란한 시점에 수행되는 경우도 있음

RDS

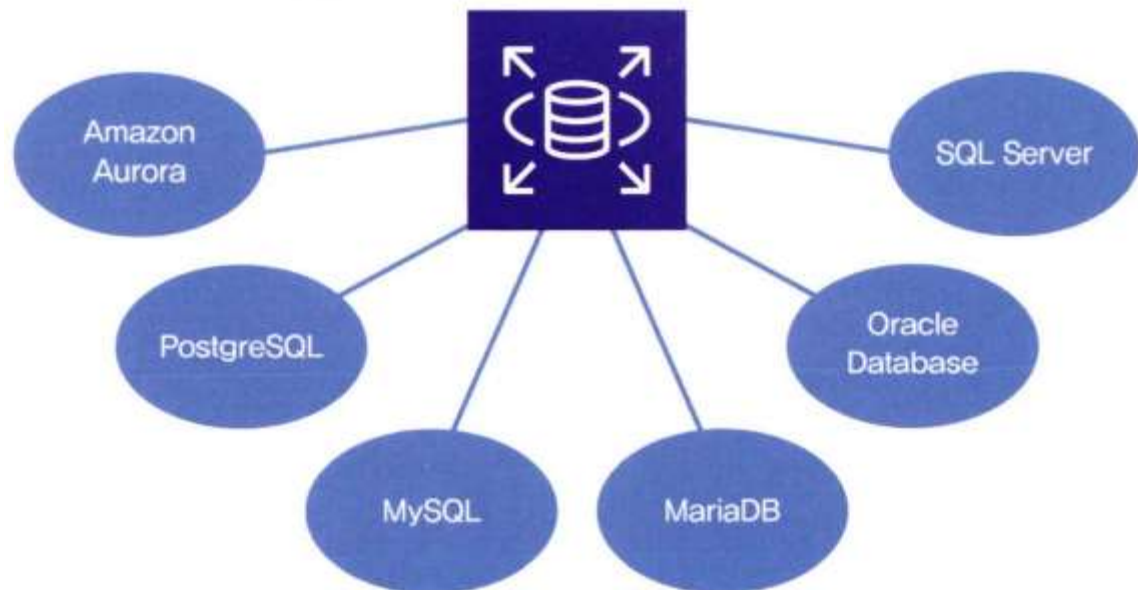
❖ RDS 자동 업데이트

- ✓ RDS는 Managed Service
- ✓ 새로운 버전이 출시되거나 소프트웨어 취약점이 발견 된 경우에는 자동으로 업데이트
- ✓ 예고없이 갑자기 업데이트한다면 당연히 곤란하기 때문에 서비스 정지가 필요할 경우에는 AWS에서 따로 공지를 하는데 사용자는 세 가지 중 하나를 선택할 수 있음
 - 자동 업데이트를 하는 방법으로 테스트 환경으로 사용할 경우는 별로 영향이 없을지도 모르지만 그래도 싫어하는 관리자가 있음
 - 관리 콘솔에서 수동으로 업데이트하거나 업데이트 시간을 지정하는 방법으로 업데이트 시에 일시적으로 데이터베이스 인스턴스를 사용할 수 없지만 관리자가 모르는 사이에 업데이트되는 것을 방지할 수 있고, 사전에 사이트 사용자와 시스템 사용자에게 공지할 수 있음
 - 업데이트를 무시하는 방법으로 치명적인 취약점에 대한 업데이트는 무시하기 힘들고 오래된 버전은 AWS가 점점 지원을 종료하므로 어느 시점에는 업데이트를 해야 하는데 업데이트를 하지 않는 선택은 문제를 미루는 것에 불과

RDS

❖ 사용 가능한 RDS 데이터베이스 엔진

- ✓ RDS는 AWS가 만든 Amazon Aurora 외에도 PostgreSQL, MySQL, MariaDB, Oracle Database, SQL Server, IBM DB2 와 같은 데이터베이스 엔진 6 종류를 사용할 수 있으며 On-premise에서 AWS로 쉽게 이전할 수 있음
- ✓ SQL Server, Oracle Database 와 같은 상용 데이터베이스도 지원
- ✓ 상용 데이터베이스의 경우 데이터베이스 라이선스 요금이 인스턴스 사용 요금에 포함되어 있다. 요금 체계가 단순하다는 점도 장점



RDS

❖ 지원하는 DBMS

DBMS(제공사)	내용
Amazon Aurora (Amazon)	MySQL이나 PostgreSQL과 호환할 수 있다. AWS가 만든 관계형 데이터베이스이다.
PostgreSQL (PostgreSQL Global Development Group)	1970년대에 개발된 Ingres라는 데이터베이스로부터 만들어진 오픈 소스 데이터베이스이다. MySQL과 마찬가지로 블로그를 비롯하여 쇼핑몰 사이트까지 폭넓게 사용되고 있다.
MySQL (Oracle)	스웨덴의 MySQL AB가 개발한 데이터베이스이다. 그 후 썬 마이크로 시스템(현재 오라클)에 매수되었다. 블로그와 같이 소규모 시스템부터 쇼핑몰 사이트와 같은 대규모 시스템까지 설계에 폭넓게 사용되고 있으며 오픈 소스 RDBMS 중에 인기가 많은 편이다.
MariaDB (Monty Program Ab)	MySQL 개발자가 스핀아웃 ¹ 하여 개발한 데이터베이스이다. 기능은 MySQL과 거의 같지만 라이선스가 GPL ² 이라는 점이 다르다.
Oracle Database (Oracle)	오라클이 제공하는 데이터베이스이다. 소규모 시스템부터 엔터프라이즈 시스템까지 폭넓게 지원하는 데이터베이스이며 증권이나 금융 시스템에 많이 사용되고 있다. 윈도우는 물론 리눅스에서도 동작한다.
Microsoft SQL Server (Microsoft)	마이크로소프트가 제공하는 데이터베이스이다. 소규모 시스템부터 엔터프라이즈 시스템까지 폭넓게 지원한다. 원래는 윈도우 서버용 소프트웨어였지만 현재는 리눅스에서도 동작한다.

- ✓ 스핀 아웃: 독립적으로 분리하는 것
- ✓ GPL: General Public License의 약자로 소프트웨어 개발자가 정해 놓은 사용 방법으로 조건 안에서 사용하는 것을 전제로 무료로 사용할 수 있는 라이선스

RDS

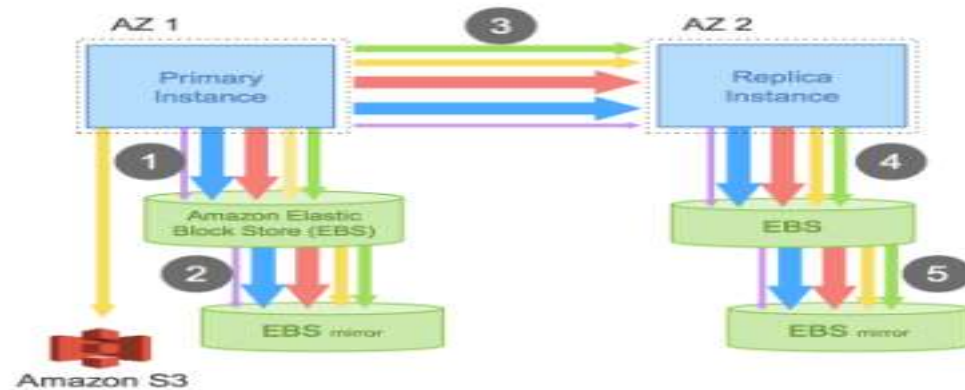
❖ Aurora

- ✓ Amazon Aurora는 MySQL이나 PostgreSQL과 호환할 수 있으며 AWS가 만든 관계형 데이터베이스
- ✓ 요금이 조금 비싼 편이지만 견고하고 성능이 뛰어남
- ✓ 호환성을 가지고 있기 때문에 일반적인 도구나 스냅샷을 사용하여 On-premise 환경의 MySQL이나 PostgreSQL을 이전할 수 있고 SQL 문도 그대로 사용할 수 있으며 코드, 애플리케이션, 드라이버 도구 등의 프로그램 및 소프트웨어를 수정할 필요도 없음
- ✓ AWS용으로 설계되어 있기 때문에 일반적인 MySQL이나 PostgreSQL 데이터베이스에 비해 속도가 빠르고 Managed Service이며 안전하게 사용할 수 있음
- ✓ AWS에서 MySQL이나 PostgreSQL을 사용하는 방법
 - EC2에 설치해서 사용
 - RDS를 사용
 - Amazon Aurora를 사용하는 방법으로 Amazon Aurora의 경우 DBMS를 변경할 때 큰 문제가 없고 높은 성능을 원할 경우에 사용을 추천하며 RDS로 이전할 때 최신 버전이나 평가판 이전 버전은 지원하지 않을 가능성이 있으므로 이럴 경우에는 EC2에 MySQL이나 PostgreSQL을 설치하여 이전

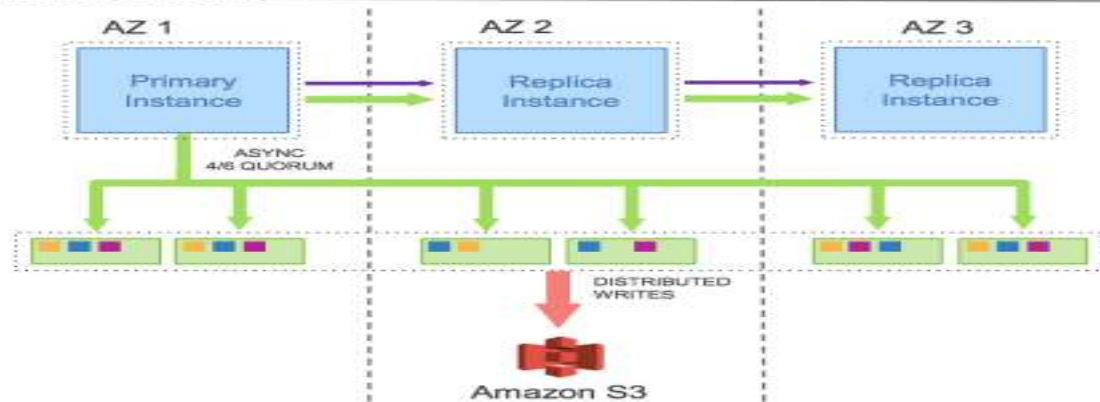
RDS

- ❖ Aurora
 - ✓ Aurora 의 차이

MYSQL WITH REPLICA



AMAZON AURORA



RDS

❖ Aurora

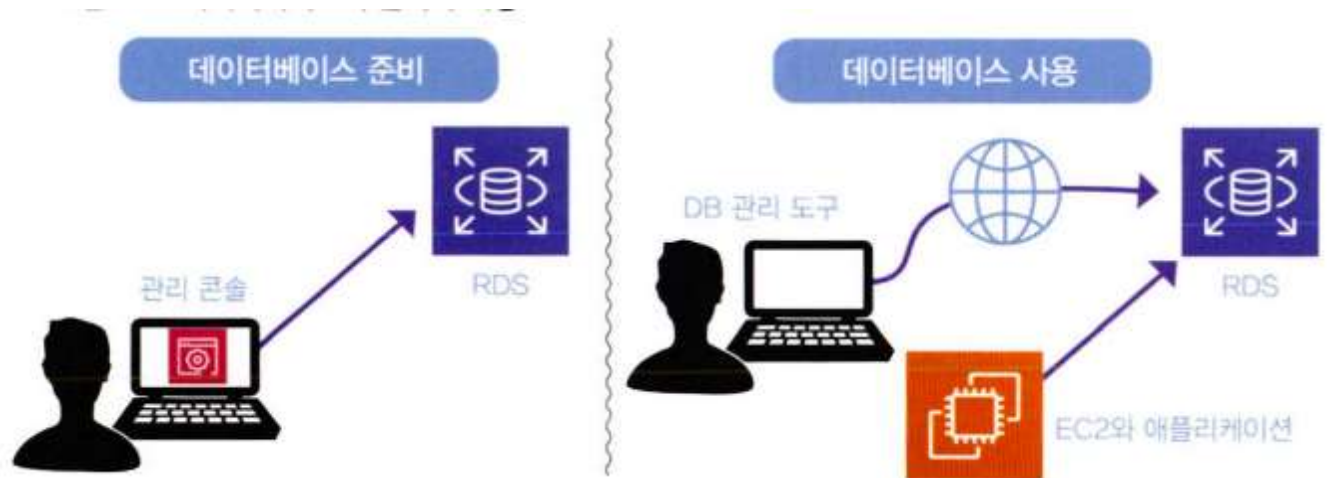
✓ Aurora 의 차이

- MySQL의 경우 자신의 EBS로 데이터를 쌓고 쌓은 데이터를 EBS로 미러링 한 다음 Replication을 통해 Replica로 데이터를 보내고 Replica는 받은 데이터를 자신의 스토리지로 쌓음
- Aurora의 경우에는 4/6 Quorum을 사용해서 스토리지에 저장하며 Replica로 보내는 것은 frm(테이블의 스키마 정보를 가진 정보) 및 redo log 이므로 network bandwidth 사용도 적으며 빠르게 변경분을 저장하고 반영할 수 있는데 이는 인스턴스와 스토리지의 영역을 나뉘기 때문에 이와 같은 아키텍처를 그릴 수 있음
- Writer에 많은 DML이 들어오는 Instance의 경우 Aurora를 사용하면 적은 Replica Lag를 가지면서 서비스 운영할 수 있음

RDS

❖ RDS 조작

- ✓ EC2와 마찬가지로 데이터베이스 작업은 데이터베이스 준비 와 데이터베이스 사용으로 나누어짐
- ✓ 데이터베이스 준비 작업은 데이터베이스를 만들거나 설정하는 등의 조작을 의미하는데 이러한 작업은 관리 콘솔에서 이루어 짐
- ✓ 데이터베이스 사용 작업은 데이터를 입력, 삭제, 변경하는 조작을 의미하는데 이러한 작업은 데이터베이스 클라이언트(관리 도구)나 소프트웨어(애플리케이션)를 통해 수행



RDS

❖ RDS 조작

- ✓ RDS는 데이터베이스 서비스이므로 설정할 수 있는 항목이 여러 가지
- ✓ 설정 항목
 - 데이터베이스 엔진 항목
 - ◆ 어떤 DBMS를 사용할 것인지 결정해야 하는데 AWS가 지원하는 버전 과 지원하지 않는 버전이 있음
 - ◆ 대부분의 데이터베이스는 소프트웨어와 연동해서 사용하기 때문에 같이 사용할 소프트웨어를 지원하는 DBMS 버전이 무엇인지 알아 두어야 함

항목	의미
데이터베이스 엔진	DBMS의 종류를 말한다. MySQL, PostgreSQL, MariaDB, Amazon Aurora 등이 있다.
템플릿	프로덕션 환경인지, 개발/테스트 환경인지를 표현한다. 프로덕션 환경을 선택하면 다중 AZ가 기본으로 활성화되는 등 해당 환경에 관한 설정이 표시된다.
라이선스 모델	GPL을 포함하여 라이선스 모델이 여러 개인 경우는 선택한다.
데이터베이스 엔진 버전	버전이 여러 개인 경우는 선택한다.

RDS

❖ RDS 조작

✓ 설정 항목

● 데이터베이스 인스턴스 항목

- ◆ On-premise 환경의 물리적인 서버에 해당하는 항목
- ◆ 사용할 수 있는 예산과 관련되기 때문에 요금과 기대 효과를 고려하여 필요한 사양을 선택
- ◆ 다중 AZ 나 스케일링은 데이터베이스 규모와 중요도에 맞게 선택

RDS

❖ RDS 조작

✓ 설정 항목

● 데이터베이스 인스턴스 항목

항목	의미
인스턴스 클래스	데이터베이스 인스턴스의 사양이다. EC2와 같이 여러 인스턴스 클래스를 제공한다.
다중 AZ 배포	두 개 이상의 AZ에 배포하여 이중화 구성 여부를 설정한다.
스토리지 유형	스토리지의 종류이다.
스토리지 할당	스토리지의 용량이다.
자동 스케일링	스토리지의 부족할 때 자동으로 늘릴지 여부를 설정한다.
스케일링 임계 값	자동 스케일링의 임계 값이다.
데이터베이스 클러스터 식별자	AWS에서 데이터베이스 클러스터를 관리하기 위한 이름이다. 데이터베이스명이 아니므로 주의해야 한다.
마스터 사용자명	데이터베이스 인스턴스의 관리자 권한을 가진 사용자명이다. 임의로 설정한다.
마스터 사용자 암호	데이터베이스 인스턴스의 관리자의 암호이다. 임의로 설정한다.

RDS

❖ RDS 조작

✓ 설정 항목

● 네트워크 항목

- ◆ RDS 인스턴스는 반드시 VPC에 설치해야 함
- ◆ EC2와 연동하려면 EC2가 사용하고 있는 서브넷 그룹 및 가용 영역을 확인해야 하고 보안 그룹도 미리 생성해 두면 좋음

항목	의미
VPC	RDS를 설치할 VPC이다. VPC의 생성은 VPC 대시보드에서 수행한다. 기본 VPC도 선택할 수 있다.
서브넷 그룹	RDS를 설치할 서브넷 그룹이다. EC2 인스턴스와 연동할 경우 동일한 곳에 설치하는 것이 일반적이다(사전에 정보를 알아 두어야 한다).
퍼블릭 액세스 가능 ³	데이터베이스 인스턴스에 공인 IP 주소를 할당할지 여부를 나타낸다. 같은 VPC 내의 서비스가 아닌 곳에서 데이터베이스에 직접 접속할 경우(회사 내 서버에서 직접 RDS에 접속해야 하는 경우를 말하지만 보안 상으로 위험하므로 실제 사용 사례는 적을 것이다)에도 필요하다.
가용 영역	RDS를 설치할 가용 영역이다. EC2 인스턴스와 연동할 경우 동일한 곳에 설치하는 것이 일반적이다(사전에 정보를 알아 두어야 한다).
보안 그룹	인스턴스 단위로 설정하는 방화벽이다. 사용할 포트를 열지 않으면 통신이 안 되므로 주의하자. ⁴ 보안 그룹이 없으면 미리 만들어 두자. 설정 하면서 생성도 가능하다.

RDS

❖ RDS 조작

✓ 설정 항목

● 데이터베이스 환경 항목

- ◆ 데이터베이스별로 환경을 설정할 수 있음
- ◆ 유지 보수와 관련된 설정이 많으므로 운영을 고려하여 설정하는 것이 좋음

항목	의미
데이터베이스명	데이터베이스 이름이다.
포트	사용할 포트이다. 1433(SQL Server), 3306(MySQL), 5432(PostgreSQL), 5439(Redshift), 1521(Oracle _Database)
데이터베이스 파라미터 그룹	환경 설정을 위한 옵션 설정이다. 대부분의 경우 기본 설정을 사용한다.
옵션 그룹	추가 기능에 대한 설정이다. 대부분의 경우 기본 설정을 사용한다.

RDS

❖ RDS 조작

✓ 설정 항목

● 데이터베이스 환경 항목

항목	의미
암호화	암호화 여부를 나타낸다. 인스턴스의 종류에 따라서 사용할 수 없는 경우도 있다.
백업	자동으로 생성된 백업의 보존 기간을 말한다.
모니터링	데이터베이스 인스턴스를 모니터링하는 방법이다. '확장 모니터링'을 활성화하면 좀 더 상세한 정보를 얻을 수 있지만 요금이 부과된다.
로그 내보내기	Amazon CloudWatch Logs를 사용하여 로그를 출력한다.
유지 관리	자동 업데이트에 대한 방법과 시간대를 설정한다.
삭제 방지	삭제되지 않도록 보호 여부를 나타낸다.

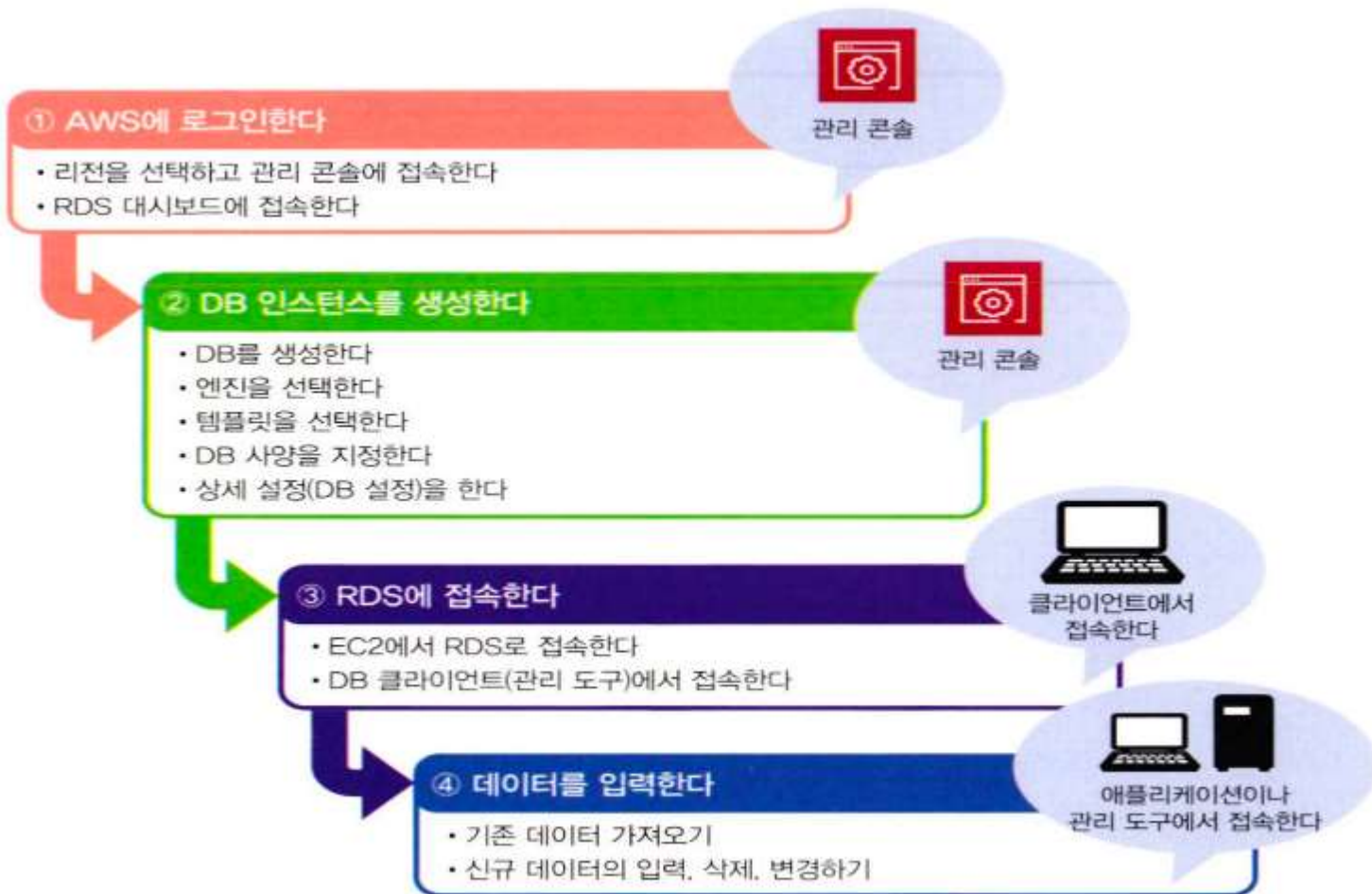
RDS

❖ RDS 사용 절차

- ✓ RDS를 사용하려면 관리 콘솔에서 RDS 대시보드에 접속하여 인스턴스를 생성
- ✓ 인스턴스가 생성되면 데이터베이스 클라이언트(관리 도구) 혹은 소프트웨어를 통해 데이터를 입력하고 갱신할 수 있음
- ✓ 대부분의 경우 RDS는 EC2에 설치된 소프트웨어와 연동하여 사용하기 때문에 EC2와 같은 네트워크에 위치해야 하며 EC2와 연동 작업이 필요
- ✓ EC2, RDS 모두 같은 VPC에 위치해야 하는데 RDS를 구축할 때 RDS와 연동할 EC2와 VPC에 대한 정보를 미리 알아 두면 좋음
- ✓ 설계 단계에서 EC2, VPC, RDS에 대한 정보를 잘 정리하여 EC2에 설치한 소프트웨어가 RDS와 연동이 잘 안 되는 상황을 피해야 함

RDS

❖ RDS 사용 절차



RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 생성
 - RDS 서비스에서 데이터베이스 생성 클릭



RDS

❖ MySQL 서버를 RDS로 생성하고 외부 접속

✓ 생성

- RDS 서비스에서 데이터베이스 생성 방식 과 종류 선택

데이터베이스 생성 방식 선택 정보

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

☐ 손쉬운 생성
원장 구성 사례 구성을 사용합니다. 일부 구성 옵션은 데이터베이스를 생성할 수 변경할 수 있습니다.

엔진 옵션

엔진 유형 정보

☐ Aurora (MySQL Compatible) 	☐ Aurora (PostgreSQL Compatible) 
☒ MySQL 	☐ MariaDB 
☐ PostgreSQL 	☐ Oracle 

RDS

❖ MySQL 서버를 RDS로 생성하고 외부 접속

✓ 생성

● 템플릿 선택

메디션

☒ 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 복원력, 인스턴트 지면 시간과 일반적으로 35% 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 높입니다.

엔진 버전

MySQL 8.0.33 ▼

템플릿

해당 사용 사례를 충족하는 샘플 템플릿을 선택하세요.

☐ 프로덕션

고가용성 및 빠른 일관된 성능을 위해 기본값을 사용하세요.

☐ 개발/테스트

이 인스턴스는 프로덕션 환경 외부에서 개발 목적으로 마틴되었습니다.

☒ 프리 티어

RDS 프리 티어를 사용하여 새로운 애플리케이션을 개발하거나, 기존 애플리케이션을 테스트하거나 Amazon RDS에서 실행 중인 애플리케이션을 실행할 수 있습니다. **정보**

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속

✓ 생성

- 접속 가능한 관리자 아이디 와 비밀번호 설정

설정

DB 인스턴스 식별자 **정보**

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

database-1

DB 인스턴스 식별자는 대소문자를 구분하지 않지만 mydbinstance와 같이 모두 소문자로 저장됩니다. 제어 T-460자의 영숫자 또는 하이픈으로 구성되어야 합니다. 첫 번째 문자는 글자이어야 합니다. 하이픈 2개가 연속될 수 없습니다. 하이픈으로 끝낼 수 없습니다.

▼ 자격 증명 설정

마스터 사용자 이름 **정보**

DB 인스턴스의 마스터 사용자에게 로그인 ID를 입력하세요.

adam

1~16자의 영숫자, 첫 번째 문자는 글자여야 합니다.

1 AWS Secrets Manager에서 마스터 보안 인증 정보 관리

Secrets Manager에서 마스터 사용자 보안 인증을 관리합니다. IAM은 사용자에 대한 암호를 생성하고 수명 주기 등과 이를 관리할 수 있습니다.

① Secrets Manager에서 마스터 사용자 보안 인증 정보를 관리하는 경우 일부 RDS 기능은 지원되지 않습니다. [자세히 알](#)
[아보기](#)

1 암호 자동 생성

Amazon RDS에서 사용자에 대한 암호를 생성하거나 사용자가 직접 암호를 지정할 수 있습니다.

마스터 암호 **정보**

지정 조건: 8자 이상의 대소 기호인 ASCII 문자. 대문자 포함할 수 있습니다. /(슬래시), !(!문자), *(*문자) 및 @(@문자).

마스터 암호 확인 **정보**

RDS

❖ MySQL 서버를 RDS로 생성하고 외부 접속

✓ 생성

● 프로세서 와 메모리 선택

인스턴스 구성

아래의 DB 인스턴스 구성 옵션은 위에서 선택한 엔진에서 지원하는 옵션으로 제한됩니다.

DB 인스턴스 클래스 [정보](#)

▼ 필터 숨기기

☒ **Amazon RDS 최적화된 쓰기를 지원하는 인스턴스 클래스 표시** [정보](#)
Amazon RDS 최적화된 쓰기는 추가 비용 없이 쓰기 처리량(throughput)을 최대 2배 늘립니다.

☐ 이전 세대 클래스 포함

☐ 스탠다드 클래스(m 클래스 포함)

☐ 메모리 최적화 클래스(r 및 x 클래스 포함)

☒ 버스터블 클래스(t 클래스 포함)

db.t3.micro

2 vCPUs 1 GiB RAM 네트워크: 2,085Mbps

RDS

❖ MySQL 서버를 RDS로 생성하고 외부 접속

✓ 생성

● 스토리지 종류 와 크기 선택

스토리지

스토리지 유형 [정보](#)

범용 SSD(gp2)

볼륨 크기에 따라 기본 성능 결정

할당된 스토리지 [정보](#)

20

GiB

최솟값은 20GiB이고, 최댓값은 6,144GiB입니다.

ⓘ DB 인스턴스의 스토리지를 수정하면 DB 인스턴스의 상태가 스토리지 최적화 상태가 됩니다. 스토리지 최적화 작업이 완료되어도 인스턴스는 계속 사용할 수 있습니다. [자세히 알아보기](#)

▶ 스토리지 자동 조정

RDS

❖ MySQL 서버를 RDS로 생성하고 외부 접속

✓ 생성

● 접속 가능 여부 설정

연결 정보

컴퓨팅 리소스
이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정할 지를 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 네트워크 환경을 정의합니다.

Default VPC {vpc-0c7db576b075e6a5d}

4 서브넷, 4 가용 영역

해당 DB 서브넷 그룹에 있는 VPC만 나열됩니다.

④ 데이터베이스를 생성한 후에는 VPC를 변경할 수 없습니다.

DB 서브넷 그룹 정보
DB 서브넷 그룹을 선택합니다. DB 서브넷 그룹은 선택한 VPC에서 DB 인스턴스가 머신 서브넷과 IP 인스턴스를 사용할 수 있는지를 정의합니다.

default-vpc-0c7db576b075e6a5d

4 서브넷, 4 가용 영역

퍼블릭 액세스 정보

☒ **예**
RDS는 데이터베이스에 연결된 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.

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 생성
 - VPC, 보안 그룹, 포트 번호 설정

VPC 보안 그룹(방화벽) 정보
데이터베이스에 대한 액세스를 허용할 VPC 보안 그룹을 하나 이상 선택합니다. 보안 그룹 규칙이 적절한 우선 처리를 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 X

가용 영역 정보

기본 설정 없음

RDS 프로кси
RDS 프로키는 애플리케이션의 복잡성, 복잡성 및 보안을 개선하는 완전관리형 고가용성 데이터베이스 프록시입니다.

☐ RDS 프로кси 생성 정보

RDS 프로кси에 대한 IAM 역할과 Secrets Manager 보안 암호를 자동으로 생성합니다. RDS 프로кси에 대한 추가 비용이 있습니다. 자세한 내용은 [AWS를 참조하십시오](#) [Amazon RDS 프로кси 요금](#)

인증 기관 - 선택 사항 정보
새 인증서를 사용하면 Amazon 데이터베이스에 대한 연결이 이루어지고 있는지 검증하여 추가 보안 계층을 제공합니다. 프로시저는 모든 데이터베이스에 자동으로 설치되는 서버 인증서를 사용하여 이를 수행합니다.

rds-ca-2019 (기본값)
만료: Aug 23, 2024

인증 기관을 선택하지 않으면 RDS에서 대한 인증 기관을 선택합니다.

추가 구성

데이터베이스 포트 정보
데이터베이스가 애플리케이션 연결에 사용할 TCP/IP 포트입니다.

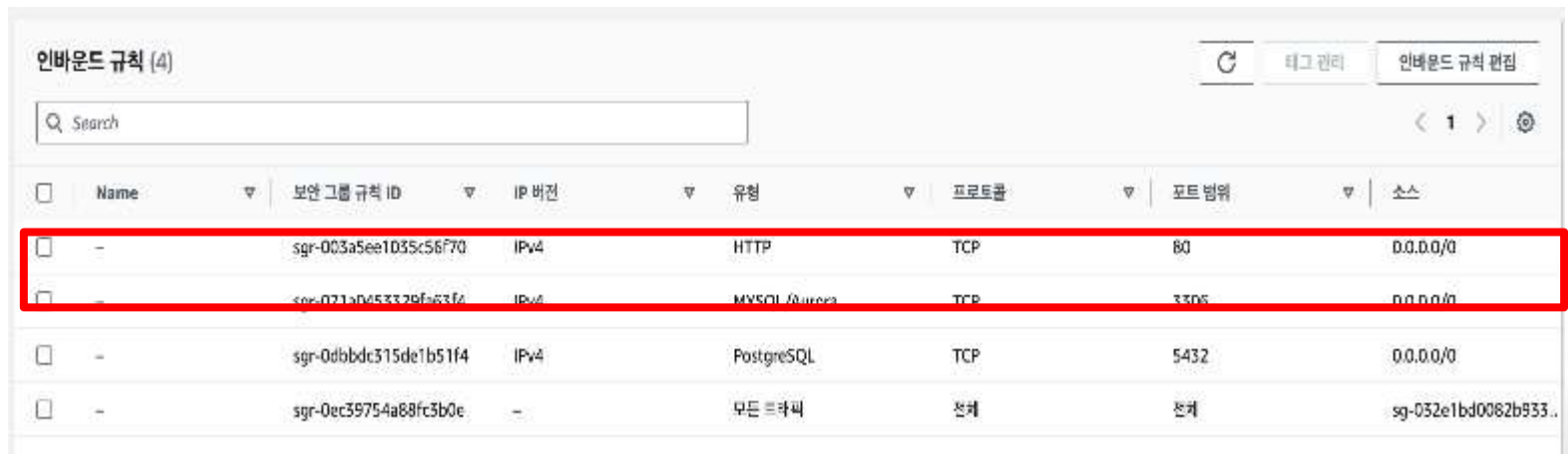
5306

RDS

❖ MySQL 서버를 RDS로 생성하고 외부 접속

✓ 외부에서 접근이 가능하도록 보안 그룹 설정

- 3306 번 포트가 외부에서 접속할 수 있도록 되어 있는지 확인
- 3306 번 포트에 대한 인바운드 규칙이 없으면 [인바운드 규칙 편집]을 눌러서 추가



인바운드 규칙 (4)

Search

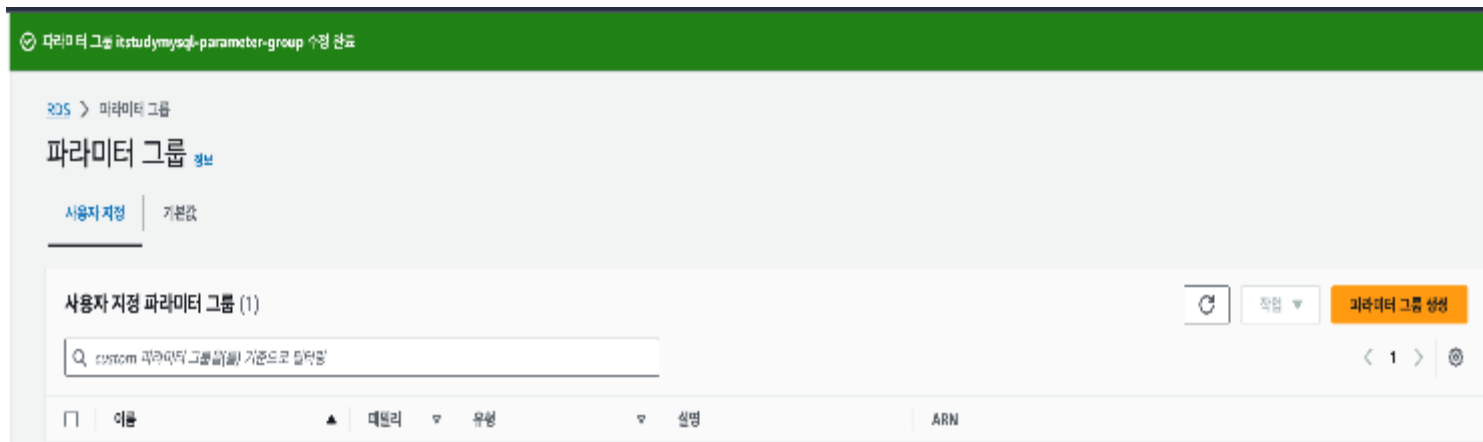
태그 관리 인바운드 규칙 편집

< 1 > ⚙

☐	Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위	소스
☐	-	sgr-003a5ee1d35c56f70	IPv4	HTTP	TCP	80	0.0.0.0/0
☐	-	sgr-021a0453328fa63fa	IPv4	MySQL/Aurora	TCP	3306	0.0.0.0/0
☐	-	sgr-0dbbdc315de1b51f4	IPv4	PostgreSQL	TCP	5432	0.0.0.0/0
☐	-	sgr-0ec39754a88fc3b0e	-	모든 트라픽	전체	전체	sg-032e1bd0082b933..

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - 파라미터 그룹을 생성



RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - 파라미터 그룹을 생성

RDS > 파라미터 그룹 > 파라미터 그룹 생성

파라미터 그룹 생성

파라미터 그룹 세부 정보

파라미터 그룹 패밀리
이 DB 파라미터 그룹을 적용할 DB 패밀리

mysql8.0 ▼

유형
DB 파라미터 그룹 유형

DB Parameter Group ▼

그룹 이름
DB 파라미터 그룹의 식별자

itstudymysql

설명
DB 파라미터 그룹에 대한 설명

itstudymysql

[취소](#) [생성](#)

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - 파라미터 설정 - 한글 인코딩

수정 가능한 파라미터 (393)

가용성으로 설정 취소 변경 사항 저장

Q character X 3자 일치 < 1 >

이름 ▲ 값 ▼ 적용 유형 ▼ 데이터 형식 ▼ 소스 ▼

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - 파라미터 설정 - 한글 인코딩

Parameter	Value	Type	Modified
character_set_database	utf8mb4	Dynamic	Modified
character_set_connection	utf8mb4	Dynamic	Modified
character_set_database	utf8mb4	Dynamic	Modified
character_set_filesystem	utf8mb4	Dynamic	Modified
character_set_results	utf8mb4	Dynamic	Modified
character_set_server	utf8mb4	Dynamic	Modified

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - 파라미터 설정 - 문자 정렬 순서

	collation_server	Dynamic	String	Engine default
	utf8mbd_unicode_ci			

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - 파라미터 설정 – time_zone



RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - 설정 변경

▼ 추가 구성

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

데이터베이스 옵션

DB 파라미터 그룹 [정보](#)

itstudymysql-parameter-group ▼

옵션 그룹 [정보](#)

default:mysql-8-0 ▼

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - 설정 변경

DB 인스턴스 수정: itstudymysql

수정 사항 요약

다음 수정 사항을 제출하려고 합니다. 변경되는 값만 표시됩니다. 변경 사항을 신중하게 확인하고 [DB 인스턴스 수정(Modify DB Instance)]을 클릭하세요.

속성	현재 값	새 값
DB 파라미터 그룹	default.mysql8.0	itstudymysql-parameter-group

수정 예약

수정 사항을 적용할 시간

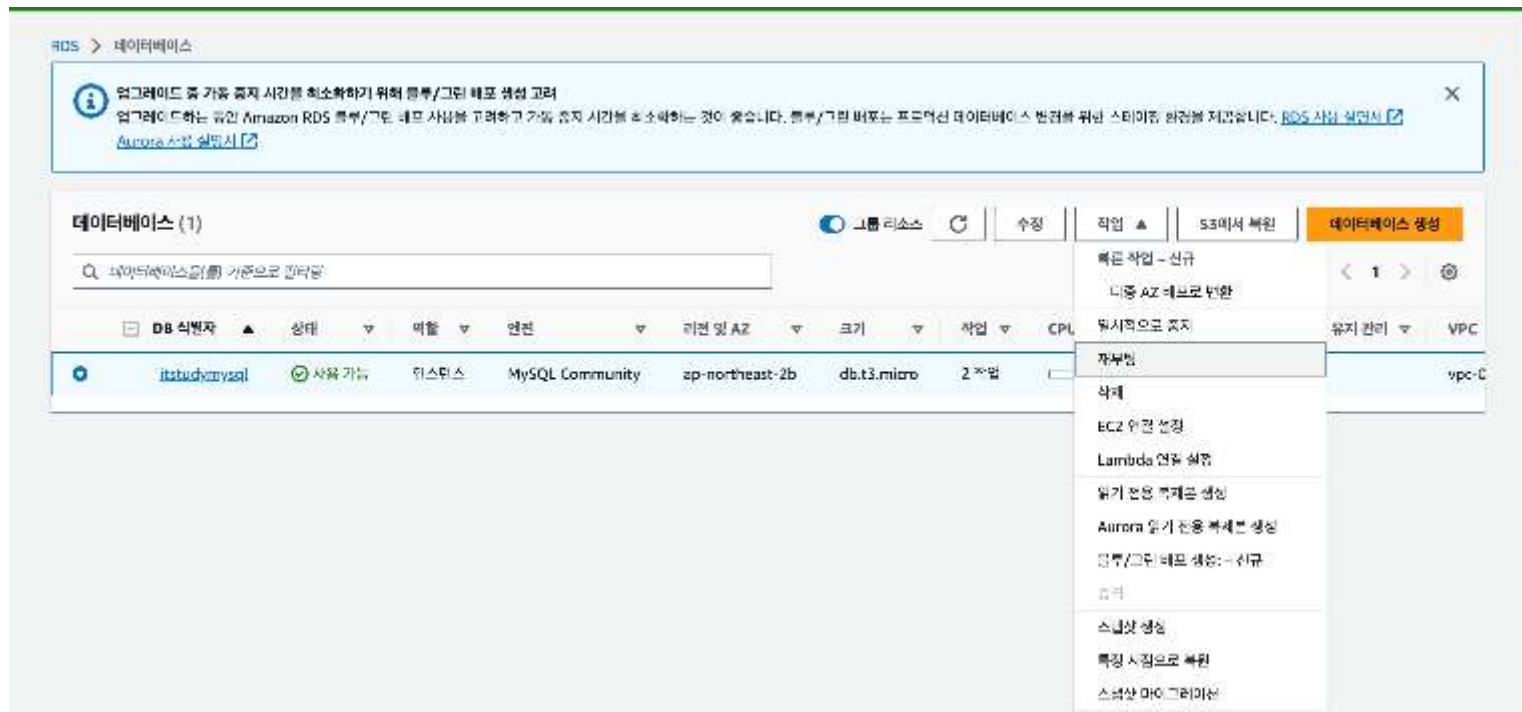
☐ 예약된 다음 유지 관리 기간에 적용
현재 유지 관리 기간: December 05, 2023 23:48 - December 06, 2023 00:18 UTC+9

☒ 즉시 적용
이 요청의 수정 사항과 보류 중인 수정 사항은 이 데이터베이스 인스턴스의 유지 관리 기간과 관계없이 가능한 한 빨리 비동기식으로 적용됩니다.

[취소](#) [뒤로](#) [DB 인스턴스 수정](#)

RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 데이터베이스 설정 변경
 - RDS 재부팅



RDS

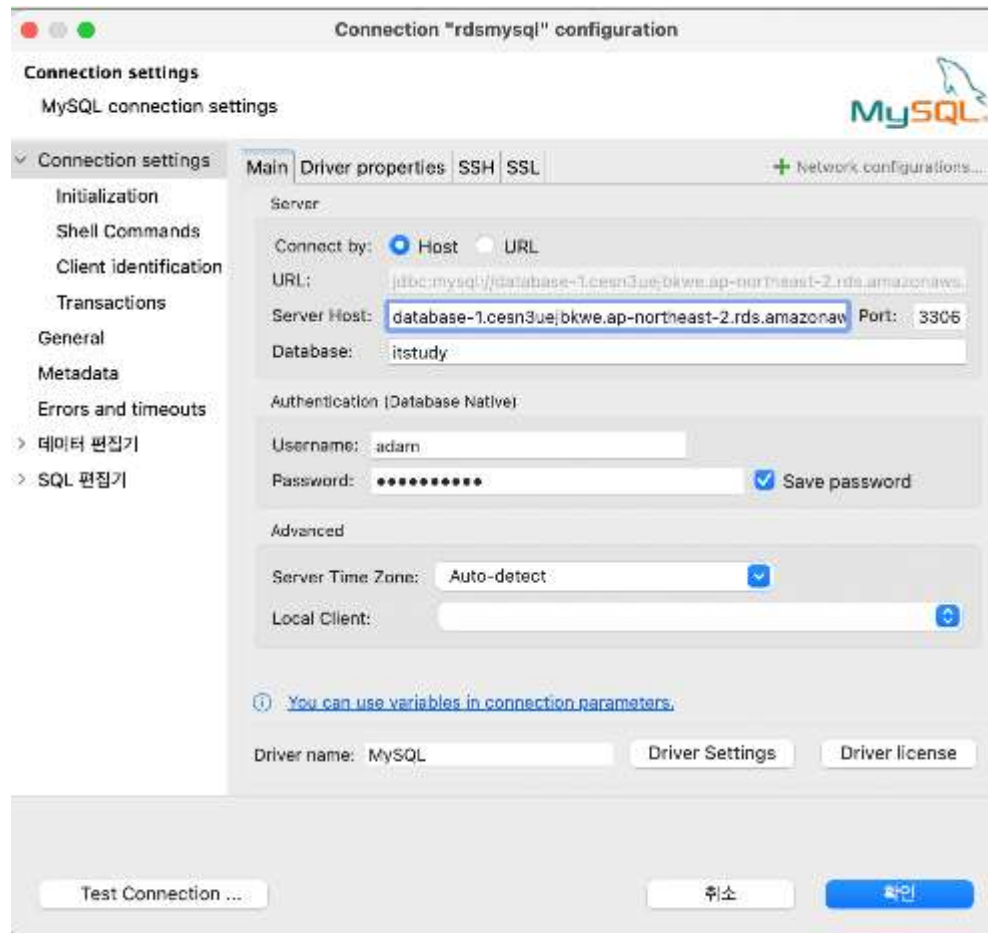
- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ End Point 확인

연결 및 보안

엔드포인트 및 포트	네트워킹	보안
엔드포인트 database-1.cesn3uejbkwe.ap-northeast-2.rds.amazonaws.com	가용 영역 ap-northeast-2c	보안 VPC 보안 그룹 default (sg-032e1bd0082b9339d) ✅ 완료
포트 3306	VPC vpc-0c7db576b075e6a5d	퍼블릭 액세스 가능 예
	서브넷 그룹 default-vpc-0c7db576b075e6a5d	인증 기관 정보 rds-ca-2019
	서브넷 subnet-0bcdf13c38a3170b4 subnet-0acde0836a0eab490 subnet-0fc7b211818954e26 subnet-0469bab0d1faaeecc	인증 기관 날짜 August 23, 2024, 02:08 (UTC+09:00)
	네트워크 유형 IPv4	D8 인스턴스 인증서 만료 날짜 ⚠ August 23, 2024, 02:08 (UTC+09:00)

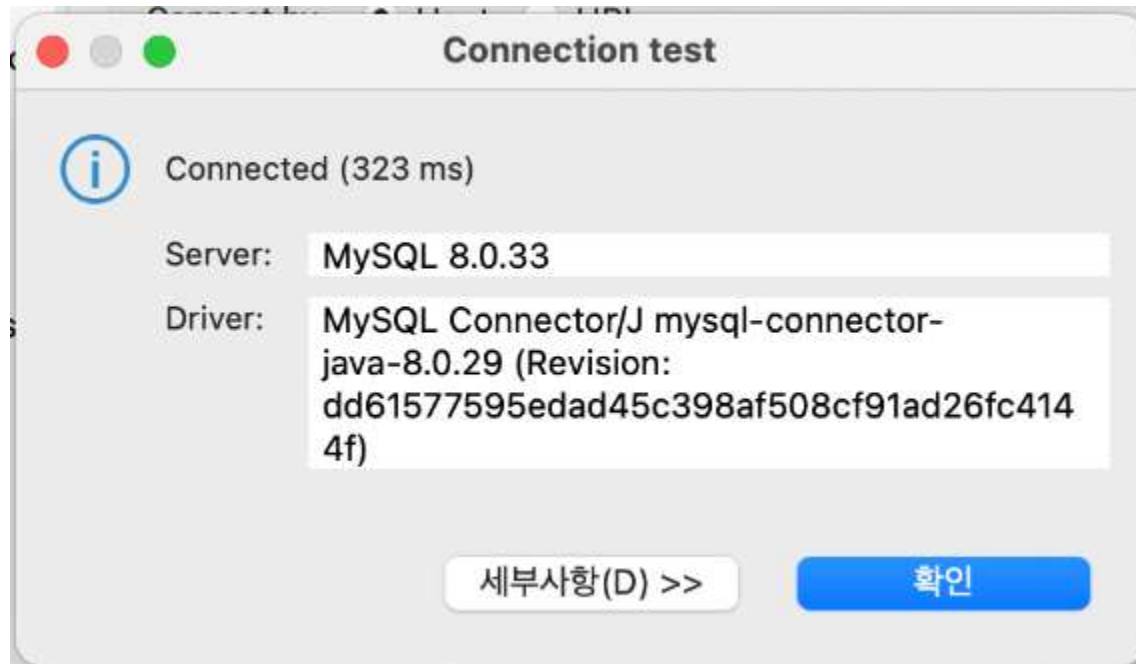
RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 접속 도구를 이용해서 접속



RDS

- ❖ MySQL 서버를 RDS로 생성하고 외부 접속
 - ✓ 접속 도구를 이용해서 접속



기타 데이터베이스

❖ Key Value Store Database

- ✓ Key Value Store 데이터베이스란 데이터의 형식에 관계없이 어떤 데이터에 대해서 특정 키가 되는 값을 결합하여 저장하는 방식의 데이터베이스
- ✓ NoSQL 데이터베이스의 대표적인 예이며 관계형 데이터베이스보다 역사가 오래 됨
- ✓ Key는 데이터를 찾기 쉽게 하기 위한 라벨을 의미
- ✓ 관계형 데이터베이스와 다르게 유연성이 높아서 형식에 상관없이 데이터를 입력할 수 있고 데이터에 대한 액세스가 빠름
- ✓ 빅데이터 나 IoT 와 같은 대용량 데이터를 빠르게 처리하는 경우에 사용되고 있어 주목받고 있음
- ✓ AWS에 Key Value Store 데이터베이스는 두 종류
 - DynamoDB - 스토리지
 - Amazon ElastiCache - 메모리

저장된 내용은 제각각이다
ID와 유형으로 검색한다

ID	유형	데이터
1	book01	로버트 A 하인라인 「여름으로 가는 문」 「별을 위한 시간」
2	music01	「알프스 교향곡」 지휘자 ○○, △△ 오케스트라 R.슈트라우스 작곡 55분
3	movie01	미나리
4	movie01	기생충 132분

기타 데이터베이스

❖ Key Value Store Database

✓ Dynamo DB

- DynamoDB는 AWS에서 제공하는 Serverless 기반 Key-Value NoSQL Database
- 높은 성능 과 비용적인 측면에서 이점을 가져옴
- NoSQL 데이터베이스
 - ◆ NoSQL 데이터베이스에서는 JOIN이라는 개념이 없는데 애플리케이션 레벨에서 구현할 수는 있지만 매우 비효율적이며 NoSQL 데이터베이스를 쓰는 이유와 맞지 않음
 - ◆ JOIN이라는 개념이 없기 때문에 정규화도 거의 불가능하기 때문에 NoSQL에서는 보통 반정규화를 수행
 - ◆ RDBMS(관계형 데이터베이스)는 종류나 특성에 따라 테이블을 나눴지만 DynamoDB에서는 모두 하나의 테이블에 표현할 수 있음
- Amazon Dynamo DB는 Key Value Store 데이터베이스
 - ◆ Key를 제외한 테이블의 속성을 미리 정의해 둘 필요가 없음
 - ◆ 데이터에 대해 미리 스키마가 만들어져 있어야 하는 RDBMS와 달리 DynamoDB는 유연하게 데이터를 처리할 수 있음
 - ◆ 관계형 데이터베이스에 맞지 않는 범용적 데이터를 저장하는 데 사용
 - ◆ Key Value 스토어 데이터베이스는 데이터를 저장하는 형식이 정해져 있지 않으며 데이터끼리 연관 관계도 없기 때문에 SQL을 사용할 수 없고 고급 검색을 할 수 없다는 단점이 있지만 관계형 데이터베이스보다 응답 속도가 빠름

기타 데이터베이스

❖ Key Value Store Database

✓ Dynamo DB

- DynamoDB는 VPC가 불필요하므로 Lambda 등과 같은 VPC를 사용하지 않는 애플리케이션 환경에 사용하기 편리
- 대규모 데이터를 처리하는 경우에 대비해서 ACID(트랜잭션에 필요한 특성) 트랜잭션, 데이터 암호화, 접근 제한 등의 서비스를 갖추고 있으며 테이블을 글로벌 테이블로 생성하면 전 세계 각 Region에 분산되어 복제되며 각 Region의 테이블에 데이터가 갱신되면 자동으로 동기화 됨
- HTTP로 통신
 - ◆ DynamoDB는 HTTP로 통신
 - ◆ 다른 DB 리소스들은 TCP Connection 기반인데 비해 DynamoDB는 Connectionless
- Serverless
 - ◆ DynamoDB를 위한 별도의 서버가 존재하지 않기 때문에 요청한 만큼만 비용을 지불
 - ◆ AWS Lambda 같은 다른 Serverless 기반 서비스와 좋은 시너지를 낼 수 있음

기타 데이터베이스

❖ Key Value Store Database

✓ Amazon ElastiCache

- Amazon ElastiCache란 인 메모리 데이터베이스(in-memory database)
- Amazon ElastiCache는 유연한 실시간 사용 사례를 지원하는 완전 관리형 인 메모리 캐싱 서비스
- 캐싱에 ElastiCache를 사용하면 애플리케이션 및 데이터베이스 성능을 가속화할 수 있으며 세션 스토어, 게임 리더보드, 스트리밍 및 분석과 같이 내구성이 필요하지 않는 사용 사례에서는 기본 데이터 스토어로 사용할 수 있음
- 인 메모리 데이터베이스란 데이터베이스를 조작할 때마다 외부 기억 장치에 읽기/쓰기를 수행하지 않고 빈번하게 읽고 쓰는 데이터는 일시적으로 메모리에 보관(캐시)하는 방식으로 메모리를 활용하여 처리 속도를 빠르게 하는 데이터베이스
- Dynamo DB는 스토리지에 저장하기 때문에 인 메모리 데이터베이스인 ElastiCache가 Dynamo DB보다 처리 속도가 더 빠르지만 인스턴스를 재 시작할 때 데이터가 삭제되기 때문에 ElastiCache는 빠른 속도를 목적으로 하는 캐시 용도로 많이 사용
- Redis 와 Memcached는 대표적인 AWS의 인 메모리형 키 밸류 스토어 DBMS
- ElastiCache는 이들과 호환할 수 있으며 현재 사용하고 있는 애플리케이션을 변경하지 않아도 사용할 수 있음

기타 데이터베이스

❖ 그 이외 데이터베이스

- ✓ 시스템의 구조가 거대화되고 복잡해지면서 기존의 관계형 데이터베이스 나 Key Value 데이터베이스로는 처리할 수 없는 데이터 유형이 생기기 시작함
- ✓ 이를 위해 등장한 것이 문서 기반 데이터베이스나 그래프 데이터베이스 등의 데이터베이스
- ✓ AWS는 이러한 데이터베이스도 사용할 수 있음



DocumentDB
(문서 기반 데이터베이스)



Neptune
(그래프 데이터베이스)



Timestream
(시계열 데이터베이스)



Quantum Ledger
Database
(장부 데이터베이스)

기타 데이터베이스

❖ Amazon Document DB(MongoDB 호환)

- ✓ Amazon Document DB는 MongoDB 호환 데이터베이스 서비스
- ✓ JSON 형식을 바이너리로 변환하여 BSON 형식으로 데이터를 저장할 수 있음
- ✓ 2000년 후반부터 등장한 빅데이터 처리, 분산 처리에 적합한 NoSQL 데이터베이스의 하나로 높은 평가를 받고 있음
- ✓ Document DB는 MongoDB가 대규모의 데이터를 안정적으로 입출력 할 수 있도록 설계되었으며 기본 MongoDB 드라이버와 도구를 사용할 수 있다는 점도 특징
- ✓ AWS Database Migration Service를 사용하여 Amazon EC2에서 가동 중인 MongoDB 데이터베이스를 Amazon Document DB로 다운타임(시스템을 이용할 수 없는 시간)없이 마이그레이션하는 것이 가능
- ✓ 기본적으로 VPC 내부에서만 접근 가능

기타 데이터베이스

❖ Amazon Neptune

- ✓ Amazon Neptune은 그래프 데이터베이스
- ✓ 그래프 데이터란 여러 가지 요소 사이의 관계 혹은 처리의 흐름을 나타내기 때문에 노드의 방향을 포함한 노드 사이의 연결을 기록한 데이터
- ✓ Amazon Neptune은 대표적인 그래프 모델인 Property Graph와 W3C의 RDF에서 사용하는 쿼리 언어인 Apache TinkerPop Gremlin과 SPARQL을 지원
- ✓ 관리가 자동화되어 있고 네트워크를 통한 데이터의 읽기/쓰기는 HTTPS로 암호화 됨
- ✓ 추천 시스템이나 소셜 네트워크, 지식 그래프(지식 기반의 발전된 형태로 정보를 축적할 뿐만 아니라 정보 간의 관련성을 가진 기술) 등에 사용되며 데이터 간의 복잡한 관계를 처리할 수 있음

기타 데이터베이스

❖ Amazon Timestream

- ✓ Amazon Timestream은 시계열 데이터베이스 서비스
- ✓ 시계열 데이터베이스란 시간의 경과에 따른 사물의 변화를 기록한 데이터로 전형적인 IoT의 데이터 취득 형태
- ✓ 시간 별로 취득한 데이터를 관계형 데이터베이스에 저장하는 것은 가능하지만 집약 및 집계 쿼리를 수행하기는 쉽지 않음
- ✓ Amazon Timestream에서 쿼리 처리 엔진이 시간 별로 처리하는 데 최적화되어 있으며 평활법(데이터의 추세나 수준 등을 분석하는 모델), 근사(선형 함수), 보간(두 점을 연결하는 방법) 등의 분석 함수를 사용할 수 있음
- ✓ 자원이 충돌하지 않도록 새로운 데이터를 삽입하고 기존 데이터의 쿼리를 다른 계층에서 처리
- ✓ 필요에 따라 스케일링되기 때문에 용량이나 부하를 고려할 필요가 없음
- ✓ 이러한 방식을 통해 IoT 데이터 처리 성능을 유지하고 데이터 관리 비용을 줄일 수 있음

기타 데이터베이스

❖ Amazon Quantum Ledger Database

- ✓ Amazon Quantum Ledger Database(QLDB)는 장부 데이터베이스
- ✓ 장부는 기업의 상거래 및 재무 기록 뿐만 아니라 감사에도 필요
- ✓ 관계형 데이터베이스는 감사에 필요한 데이터 변경 내용의 추적 및 검증에 대해서는 지원하지 않으므로 시간과 비용을 들여서 감사용 애플리케이션을 개발해야 함
- ✓ Amazon QLDB는 데이터 변경을 추적하여 이력을 남기는 저널 형식을 채용
- ✓ 저널 데이터는 변경, 삭제가 불가능하며 암호화 해시 함수(SHA- 256)에서 변경한 내용을 변조하는 것을 방지하고 있음
- ✓ SQL과 비슷한 API나 문서 기반 데이터 모델을 제공하며 조작은 쉬운 편