

06. AWS 데이터베이스 서비스

6.1. 데이터베이스와 DBMS

AWS가 제공하는 데이터베이스 서비스를 살펴보기에 앞서 데이터베이스가 무엇인지 이해해야 한다. 데이터와 데이터베이스, DBMS를 먼저 학습한 후 데이터베이스 모델 유형과 DBMS 종류를 알아본다.

6.1.1. 데이터와 데이터베이스

데이터(data)는 어떤 이론을 세우는 데 기초가 되는 자료로, 문자와 숫자, 그림, 영상 등의 형태로 된 단위이다. 데이터는 원시적인 자료에 불과할 뿐 특정한 의미와 가치는 없다. 이런 데이터를 특정 목적에 따라 가공하고 이론을 세워 의미와 가치를 부여한 결과를 정보(information)라고 한다.

데이터를 하나에 모아 두는 것을 데이터베이스(database)라고 하며, 데이터베이스는 간단히 데이터를 모아 두는 집합이라고 정의할 수 있다. 데이터베이스를 관리하는 시스템을 DBMS(DataBase Management System)라고 하며, DBMS는 데이터에 대해 사용자의 논리적 명령을 해석하고 필요한 데이터를 찾아주는 시스템이라고 정의할 수 있다.

6.1.2. 데이터베이스 유형

데이터베이스는 구조, 운영, 저장 방식에 따라 크게 계층형(hierarchical), 네트워크형, 관계형(relationship), 객체 지향형(object-oriented) 모델 유형으로 구분할 수 있다.

계층형 데이터베이스 모델

트리(tree) 구조를 기반으로 하여, 데이터는 1:N으로 상하 종속 관계 구조로 되어 있다. 데이터에 빠르게 접근이 가능하지만, 구조 변경에 대한 유연성이 부족하다는 단점이 있다.

네트워크형 데이터베이스 모델

계층형 데이터베이스의 정형화된 구조를 해결하기 위해 1:N뿐 아니라 1:1, N:N 형태의 구조를 가질 수 있다. 하위 개체는 다수의 상위 개체를 가질 수 있어 형태가 좀 더 자유롭다. 다만 다양한 개체 간 연결에 따라 복잡성은 높아지기 때문에 데이터베이스 전반의 구조를 이해하는 데 어려움이 따를 수 있다.

관계형 데이터베이스 모델

데이터의 논리적 관계를 초점으로 특정 개체 정보를 이용하여 열(column)과 행(row)으로 된 테이블(table)이라는 최소 단위로 구성된다. 계층형과 네트워크형 데이터베이스 모델은 데이터가 변화하면 상관관계에 따라 모든 데이터를 변경해야 하지만, 관계형 데이터베이스 모델은 데이터가 변화하면 쉽게 적용할 수 있다는 장점이 있다. 하지만 시스템 자원을 많이 차지해서 상대적으로 느리다는 단점이 있다.

객체 지향형 데이터베이스 모델

모든 데이터를 테이블 형태로 구성하는 관계형 데이터베이스 모델과 다르게 객체(object)라는 형태의 최소 단위로 표현한다. 객체 집합을 정의하고 연결 구조를 확립하여 복잡한 객체 구조도 즉각적으로 표현할 수 있으며, 사용자 정의 데이터, 멀티미디어 등 비정형 데이터도 지원한다. 하지만 보편적으로 사용하는 관계형 데이터베이스 호환성 문제와 복잡성에 따른 문제도 있다. 현재 객체 지향형 데이터베이스 모델은 특수한 전문 분야가 아니면 잘 쓰지 않는다.

6.1.3. 관계형 DBMS의 SQL 언어

현재 가장 보편적으로 사용하는 데이터베이스 모델은 관계형 DBMS이다. 여기에 저장된 데이터와 통신할 때 사용하는 SQL 프로그래밍 언어를 좀 더 알아보자. SQL 명령은 데이터 정의, 데이터 조작, 데이터 제어로 분류하며, 다음과 같이 정리할 수 있다.

SQL 데이터 정의

DBMS와 연계되는 다양한 응용 프로그램에서 요구하는 데이터의 형식과 구조를 정의하는 기능으로, 테이블이나 관계의 구조를 정의하는 목적에 따라 CREATE, ALTER, DROP 등 명령어가 있다.

- CREATE: 테이블을 구성하고 속성에 대한 제약을 정의하는 명령어
- ALTER: 생성된 테이블 속성을 정의하고 변경하는 명령어
- DROP: 생성된 테이블을 삭제하는 명령어

SQL 데이터 조작

데이터 검색, 삽입, 삭제 등을 처리하는 인터페이스를 제공하는 기능으로 SELECT, INSERT, UPDATE, DELETE 등 명령어가 있다.

- SELECT: 테이블에 있는 정보를 가져오는 명령어
- INSERT: 테이블에 새로운 데이터를 삽입하는 명령어
- UPDATE: 테이블에 있는 데이터를 수정하는 명령어
- DELETE: 테이블에 있는 기존 데이터를 삭제하는 명령어

SQL 데이터 제어

데이터 무결성 유지와 접근 권한 및 다수의 사용자 데이터베이스에 정확하게 접근하는 기능으로 GRANT, DENY, REVOKE 등 명령어가 있다.

- GRANT: 테이블에 대한 권한을 허용하는 명령어
- DENY: 테이블에 대한 권한을 차단하는 명령어
- REVOKE: 테이블에 대한 권한을 회수하는 명령어

6.1.4. DBMS 종류

다양한 회사에서 만든 DBMS 제품들이 있는데, 대표적으로 Oracle, MySQL, MS-SQL, PostgreSQL 등이 있다. DBMS 인기도에 따라 순위를 매기는 랭킹 사이트에서 최신 DBMS 트렌드를 확인할 수 있다.

<https://db-engines.com/en/ranking>

423 systems in ranking, January 2025							
Rank			DBMS	Database Model	Score		
Jan 2025	Dec 2024	Jan 2024			Jan 2025	Dec 2024	Jan 2024
1.	1.	1.	Oracle	Relational, Multi-model	1258.76	-5.03	+11.27
2.	2.	2.	MySQL	Relational, Multi-model	998.15	-5.61	-125.31
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	798.55	-7.14	-78.05
4.	4.	4.	PostgreSQL	Relational, Multi-model	663.41	-2.97	+14.45
5.	5.	5.	MongoDB	Document, Multi-model	402.50	+2.12	-14.98
6.	7.	9.	Snowflake	Relational	153.90	+6.54	+27.98
7.	6.	6.	Redis	Key-value, Multi-model	153.36	+3.08	-6.03
8.	8.	7.	Elasticsearch	Multi-model	134.92	+2.60	-1.15
9.	9.	8.	IBM Db2	Relational, Multi-model	122.97	+0.19	-9.43
10.	10.	11.	SQLite	Relational	106.69	+4.97	-8.51

6.2. AWS 데이터베이스 서비스

AWS에서는 사용자 요구 사항에 따라 관계형 데이터베이스, 키-값 데이터베이스, 인-메모리 데이터베이스 등 다양한 데이터베이스 서비스를 제공한다.

6.2.1. Amazon RDS(Relational Database Service)

클라우드 환경에서 관계형 데이터베이스를 간편하게 설정하고 운영할 수 있는 서비스로 원하는 성능 요구에 따라 다양한 데이터베이스 인스턴스 유형인 Amazon Aurora, PostgreSQL, Oracle Database, SQL Server 등 관계형 데이터베이스 엔진을 선택할 수 있다.

관계형 데이터베이스 모델 유형으로, 테이블 구조에 행(row)과 열(column)로 구성된다. Amazon RDS는 관계형 데이터베이스 엔진을 사용하기 때문에 SQL 언어 기반으로 데이터베이스를 손쉽게 제어할 수 있다.

사용 편의성이 높고, 모니터링 및 지표와 이벤트 알람을 이용하여 높은 관리 효율성을 보장하며, 비용도 상대적으로 매우 저렴하다. 또한 간편한 복제 기능을 이용하여 워크로드의 가용성과 확장성을 확보할 수 있다. Amazon RDS가 지원하는 데이터 복제 기능인 Multi-AZ와 Read Replica를 다시 다룬다.

6.2.2. Amazon RDS 데이터 복제

데이터베이스를 복제하는 방법으로 Multi-AZ와 Read Replica가 있다. 이런 복제 기능으로 데이터베이스에 있는 데이터를 복제하여 Primary DB의 데이터를 동기화할 수 있다.

Multi-AZ 복제 방식

기본적으로 액티브-스탠바이(active-standby) 형태로 동작한다. Primary DB가 액티브(활성) 상태이며, 보조의 Standby Replica가 스탠바이(대기) 상태입니다. Primary DB에 문제가 발생하면 Standby Replica를 Primary DB로 승격하여 동적으로 유지한다.

Read Replica 복제 방식

원본 데이터를 Primary DB에 두고, 읽기 전용의 복제 데이터를 Read Replica 데이터베이스에 생성하여 유지한다. 읽기 전용 복제 데이터가 있는 Read Replica는 확장이 가능하며, 데이터 읽기 처리 속도를 높일 수 있다.

6.2.3. Amazon Aurora

AWS 관계형 데이터베이스 서비스는 Amazon RDS를 이용하여 MySQL, PostgreSQL, Oracle 등 데이터베이스 엔진을 주로 사용하다 Amazon Aurora라는 AWS 자체의 클라우드 데이터베이스 엔진을 개발했다. Amazon Aurora는 엔터프라이즈 수준의 관계형 데이터베이스 엔진으로 안정적으로 고성능의 데이터베이스 처리가 가능하다. 또한 오픈 소스를 기반으로 다른 관계형 데이터베이스와 호환성이 우수하며, 비용 효율이 높다는 장점이 있다. 이런 Amazon Aurora 데이터베이스 엔진은 Amazon RDS에서 관리하며 프로비저닝, 패치, 백업, 복원, 장애, 복구 등 작업을 수행한다.

Amazon Aurora 복제 방식

Amazon Aurora 데이터베이스 엔진은 다른 관계형 데이터베이스 엔진보다 스토리지 내결함성이 우수하다. 예를 들면 MySQL 데이터베이스 엔진은 데이터베이스 인스턴스에 EBS 스토리지가 연결되어 서로 다른 가용 영역으로 동기식 복제가 된다. 반면 Amazon Aurora 데이터베이스 엔진은 공유 스토리지를 통해 최소 세 개의 가용 영역에서 두 개씩 총 여섯 개의 복제 데이터를 가지고 있어 더욱 안정적으로 서비스할 수 있다.

6.2.4. Amazon DynamoDB

비관계형 데이터베이스로, 키-값(key-value) 메소드를 사용하는 키-값 데이터베이스이다. 여기에서 비관계형 데이터베이스는 관계형 데이터베이스와 다르게 데이터가 서로 연결되지 않는 개별 형태로 저장하여 복잡하고 구조화되지 않은 데이터 유형에 적합한 데이터베이스이다. 앞서 관계형 데이터베이스가 SQL 언어를 사용했다면, Amazon DynamoDB는 비관계형 데이터베이스로 SQL 문을 사용하지 않는다. 이런 측면에서 NoSQL 데이터베이스라고도 한다.

Amazon DynamoDB의 데이터베이스 구조는 키와 값으로 된 비관계형 데이터베이스 유형이다. 키를 데이터의 고유한 식별자로 사용하고, 값의 유형의 제한이 없어 단순한 개체(entity)뿐 아니라 복잡한 집합체까지 무엇이든 가능한 비정형 데이터를 입력할 수 있다. 이런 키와 값을 쌍으로 집합해서 저장한다.

키-값 데이터베이스는 데이터베이스 구조가 단순하여 빠른 처리가 가능하다. 이런 특성에 따라 Amazon DynamoDB는 대규모 환경에서도 일관되게 10밀리초 미만의 처리 성능을 제공한다. 그리고 Amazon DynamoDB는 별도 서버를 구축하지 않고 운영되는 서버리스(serverless)로 동작하기 때문에 서버에 대한 프로비저닝, 패치, 소프트웨어 설치가 필요 없고 용량에 따라 데이터를 자동으로 확장 및 축소해서 관리 편의성이 높다.

6.2.5. Amazon ElastiCache

Amazon ElastiCache는 인-메모리 데이터베이스로, 데이터를 메모리에 저장하는 형태로 동작한다. 데이터가 메모리

상에 위치하여 데이터를 빠르게 처리할 수 있다는 장점이 있지만, 데이터 양이 많다면 데이터 처리가 느려질 수 있기 때문에 대용량 데이터에는 적합하지 않고 주로 데이터를 빠르게 자주 접근해야 할 때 사용한다. 이런 Amazon ElastiCache는 Memcached 방식과 Redis 방식 두 가지로 구분된다.

Amazon ElastiCache for Memcached

Memcached는 보편적으로 사용하는 메모리 객체 캐싱 시스템으로, 인-메모리 데이터베이스 서비스이다. Amazon ElastiCache for Memcached는 Memcached와 호환하여 자주 접근할 데이터를 메모리에 놓고 빠르게 처리할 수 있다.

Amazon ElastiCache for Redis

Redis는 데이터베이스, 캐시, 메시지 브로커 및 대기열 용도로 사용되는 인-메모리 데이터베이스 서비스이다. Amazon ElastiCache for Redis는 Redis 기반으로 구축되고 Redis API와 호환하여 개방형 Redis 데이터 형식으로 저장한다. 실시간 애플리케이션을 지원할 수 있도록 1밀리초 미만의 지연으로 빠른 데이터를 처리할 수 있다.

6.3. 웹 서버와 Amazon RDS 연동하기

이번 실습은 AWS의 관계형 데이터베이스인 Amazon RDS를 배포하고 웹 서버와 연동하는 것으로,고가용성 확보를 위한 Multi-AZ 기능과 성능 확장을 위한 Read Replica 기능을 알아본다.

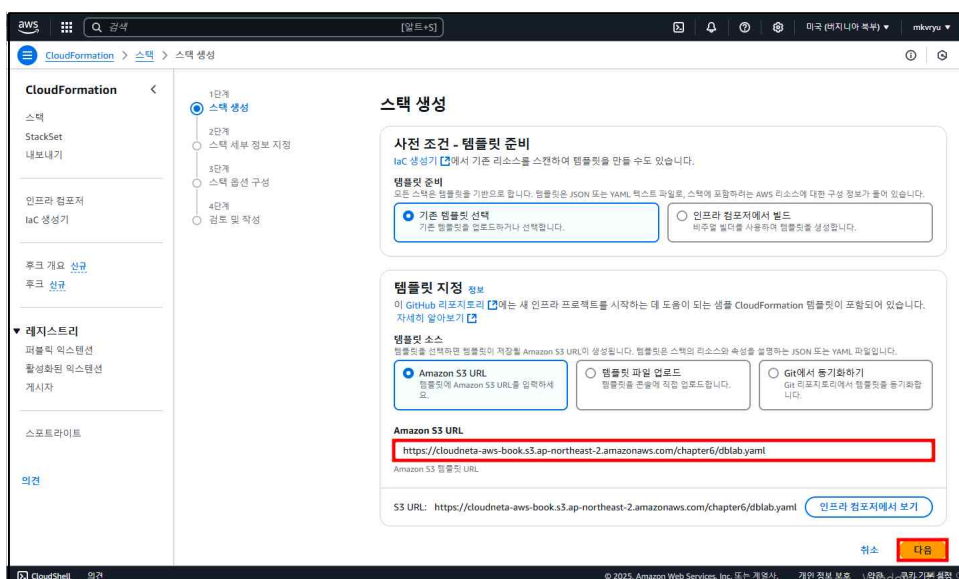
6.3.1. CloudFormation으로 기본 인프라 배포하기

AWS 관리 콘솔에서 서비스 => 관리 및 거버넌스 => CloudFormation으로 들어가 “스택 생성”을 누른다.



아래쪽 Amazon S3 URL에 다음 URL을 입력하고 “다음”을 누른다.

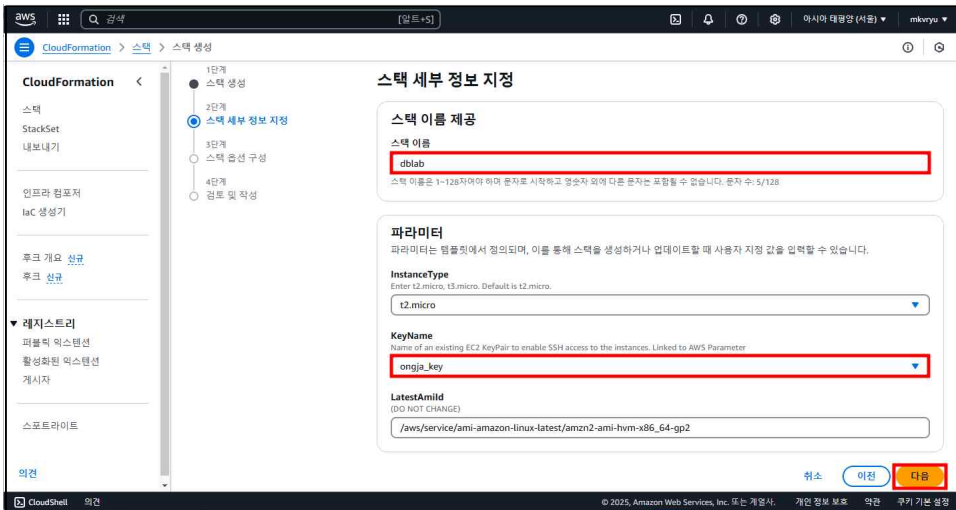
- <https://cloudneta-aws-book.s3.ap-northeast-2.amazonaws.com/chapter6/dblab.yaml>



스택 세부 정보 지정 페이지에서 다음과 같이 설정하고 “다음”을 누른다.

- 스택 이름에 “dblab” 입력

· KeyName은 사용자 키 페어 파일 선택



스택 옵션 구성에서는 별도 설정 없이 “다음”을 누른다. 검토 및 작성에서도 별도의 설정 없이 “전송”을 누른다. AWS CloudFormation 기본 인프라를 배포하고 일정 시간(5분)이 지나 스택 상태가 “CREATE_COMPLETE”가 되면 모든 인프라 배포가 정상적으로 완료된 것이다.



AWS CloudFormation으로 생성된 기본 인프라 자원 정보는 다음과 같다.

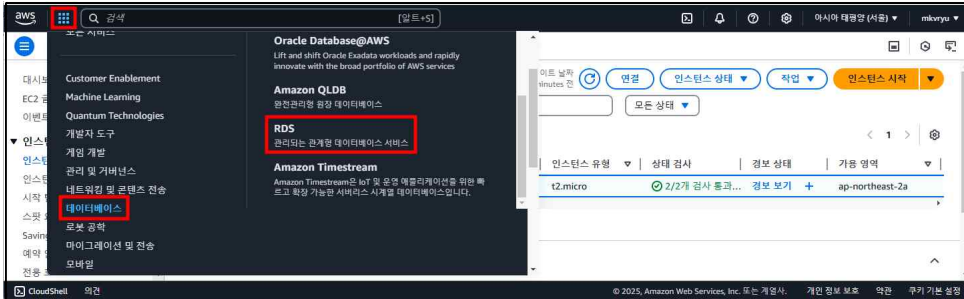
생성 자원	이름	정보
VPC	CH6-VPC	10.6.0.0/16
인터넷 게이트웨이	CH6-IGW	
퍼블릭 라우팅 테이블	CH6-PublicRT	
프라이빗 라우팅 테이블	CH6-PrivateRT	
서브넷 1(퍼블릭)	CH6-Subnet1	CH6-PublicRT 연결
서브넷 2(프라이빗)	CH6-Subnet2	CH6-PrivateRT 연결
서브넷 3(프라이빗)	CH6-Subnet3	CH6-PrivateRT 연결
EC2 인스턴스	CH6-WebSrv	10.6.1.10
보안 그룹 1	CH6-SGWebSrv	TCP 22, 88, ICMP 허용
보안 그룹 2	CH6-SGRDS	TCP 3306 허용
DB 서브넷 그룹	DBLab-Subnet2 · 3	CH6-Subnet2, CH6-Subnet3
DB 파라미터 그룹	mymysql8	

기본 인프라 자원이 생성되고 EC2 인스턴스(CH6-WebSrv)에 정상적으로 SSH 접속이 되는지 확인한다.

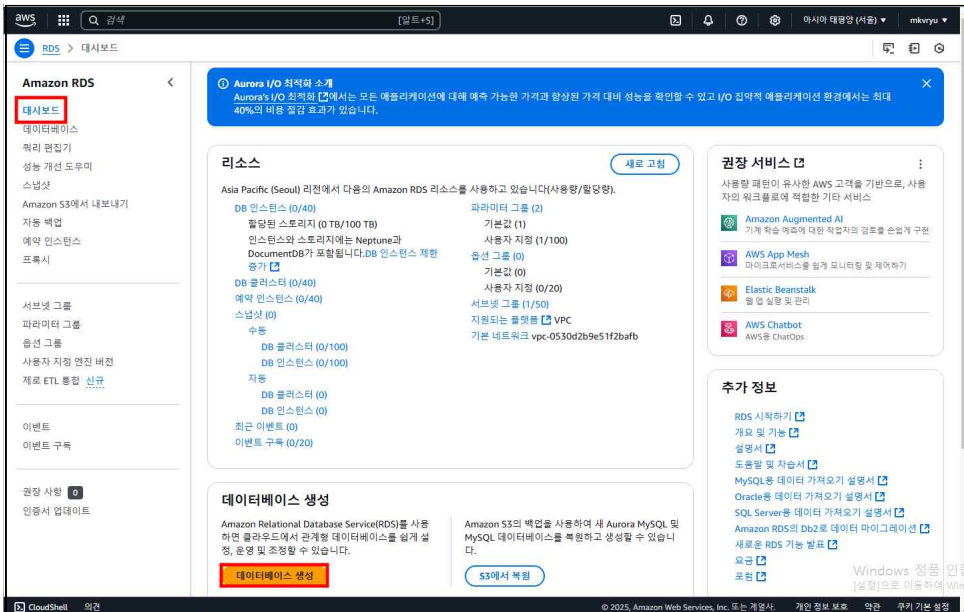
6.3.2. Amazon RDS를 생성하고 웹 서버와 연동하기

이번 실습에서는 두 개의 Amazon RDS를 생성한다. 데이터베이스 생성, 수정, 삭제 등은 각 5~15분 정도 시간이 소요되므로 실습 시간이 길어질 수 있으니 참고한다.

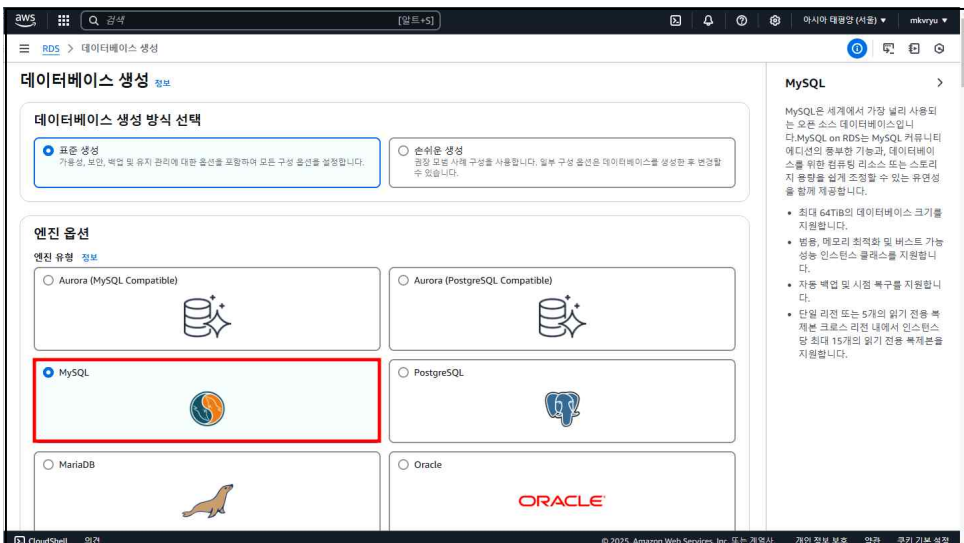
Amazon RDS1 데이터베이스를 생성하기 위해 AWS 관리 콘솔에서 서비스 => 데이터베이스 => RDS를 선택한다.



대시보드에서 “데이터베이스 생성”을 누른다.



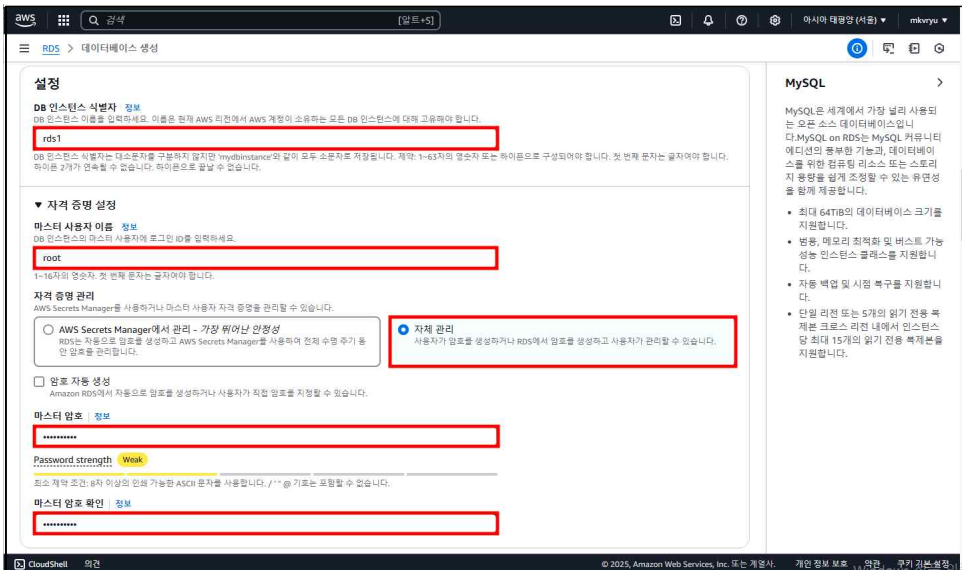
RDS 데이터베이스 생성 페이지에서 다음과 같이 설정하고 맨 아래쪽에 있는 “데이터베이스 생성” 누르기
· 엔진 옵션에서 엔진 유형은 “MySQL” 선택



- 템플릿은 “개발/테스트” 선택
- 가용성 및 내구성은 “다중 AZ DB 인스턴스” 선택



- DB 인스턴스 식별자에 “rds1” 입력
- 마스터 사용자 이름에 “root” 입력
- 자격 증명 관리는 “자체 관리” 선택
- 마스터 암호와 마스터 암호 확인에 “qwerty12345” 입력



- “이전 세대 클래스 포함” 체크
- “버스터블 클래스(t클래스 포함) 선택
- “db.t3.micro” 선택



- Virtual Private Cloud(VPC)는 “CH6-VPC” 선택
- 기존 VPC 보안 그룹은 “default”를 제거한 후 “db1ab-CH6SG2-XXXX” 선택

연결 정보

컴퓨팅 리소스
이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정할 지를 선택합니다. 연결을 설정하면 컴퓨팅 리소스가 이 데이터베이스에 연결할 수 있도록 연결이 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 데이터베이스와 EC2 컴퓨팅 리소스에 대한 연결을 설정합니다.

Virtual Private Cloud (VPC) 정보
VPC를 선택합니다. VPC는 이 DB 인스턴스의 가상 네트워킹 환경을 정의합니다.

CH6-VPC (vpc-05530e62f4c3dca8f)
3 서브넷, 2 가용 영역

해당 DB 서브넷 그룹이 있는 VPC만 나열됩니다.

☒ 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.

dblab-mydbsubnetgroup-njg21h229n7
2 서브넷, 2 가용 영역

파라미터 그룹 정보

☐ 예
RDS는 데이터베이스에 포함된 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(선택적) 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

dblab-CH6SG2-c23f0c52oaj

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

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

인증 기관 - 선택 사항 정보
서버 인증서를 사용하려면 Amazon 데이터베이스에 대한 연결이 이루어지고 있는 지 검증하여 추가 보안 계층을 제공합니다. 프로비저닝하는 모든 데이터베이스에 자동으로 설치되는 서버 인증서를 한 번에 만들 수 있습니다.

rds-ca-rs2048-g1 (기본값)
만료: May 21, 2061

인증 기관을 선택하지 않으면 RDS에서 대신 인증 기관을 선택합니다.

▶ 추가 구성

· “향상된 모니터링 활성화”에 체크 해제

모니터링

☒ 향상된 모니터링 활성화
향상된 모니터링 지표를 활성화하면 다른 프로세스 또는 스레드에서 CPU를 사용하는 방법을 확인하는 경우에 유용합니다.

- 맨 아래쪽으로 내려가 “추가 구성”을 클릭하고 초기 데이터베이스 이름에 “sample” 입력
- DB 파라미터 그룹은 “dblab-mydparametergroup-XXXX” 선택
- 백업 보존 기간은 “35”일 선택

추가 구성
데이터베이스 옵션, 암호화 커밋, 백업 커밋, 복구 커밋, 유지 관리, CloudWatch Logs, 삭제 방지 커밋.

데이터베이스 옵션
초기 데이터베이스 이름 정보

sample

데이터베이스 이름을 지정하지 않으면 Amazon RDS에서 데이터베이스를 생성하지 않습니다.

DB 파라미터 그룹 정보

dblab-mydparametergroup-k0jc6e0yqjs

옵션 그룹 정보

default-mysql-8-0

백업

☒ 자동 백업을 활성화합니다.
데이터베이스의 특정 시점 스냅샷을 생성합니다.

☒ 자동 백업 기능은 현재 InnoDB 스토리지 엔진에 대해서만 지원됩니다. MyISAM을 사용하는 경우 [여기](#)에서 자세한 정보를 참조하세요.

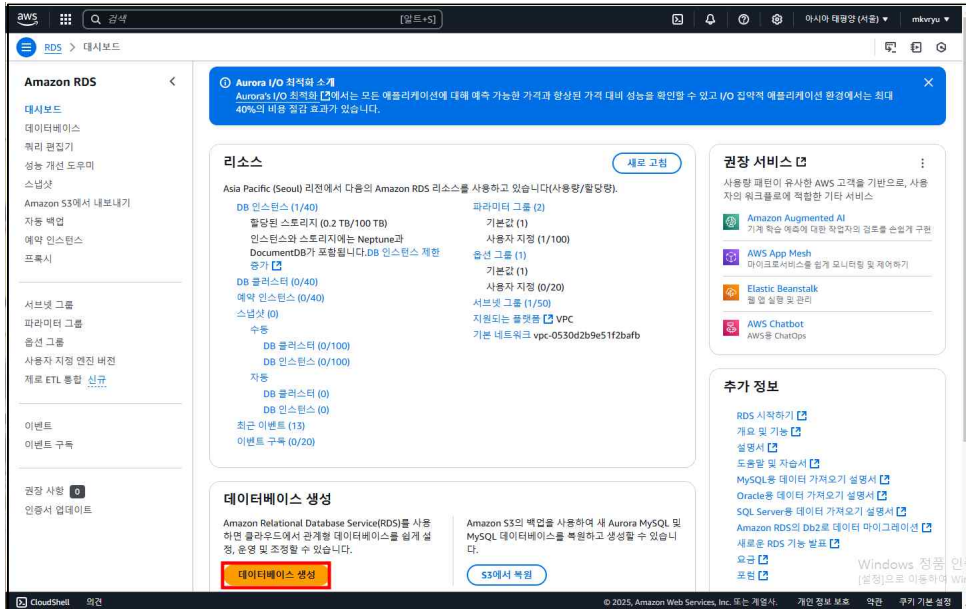
백업 보존 기간 정보
자동 백업이 유지되는 동안(1~35일)입니다.

35 일

RDS1 데이터베이스가 생성되기까지 약 12분 정도 소요된다. 이번엔 생성한 RDS1 데이터베이스의 특징은 “다중 AZ DB 인스턴스”를 선택하여 Multi-AZ 기능을 활성화했다. 참고로 Multi-AZ 기능을 활성화하면 Primary DB와 Standby Replica가 생성되기 때문에 생성 시간이 길어진다.

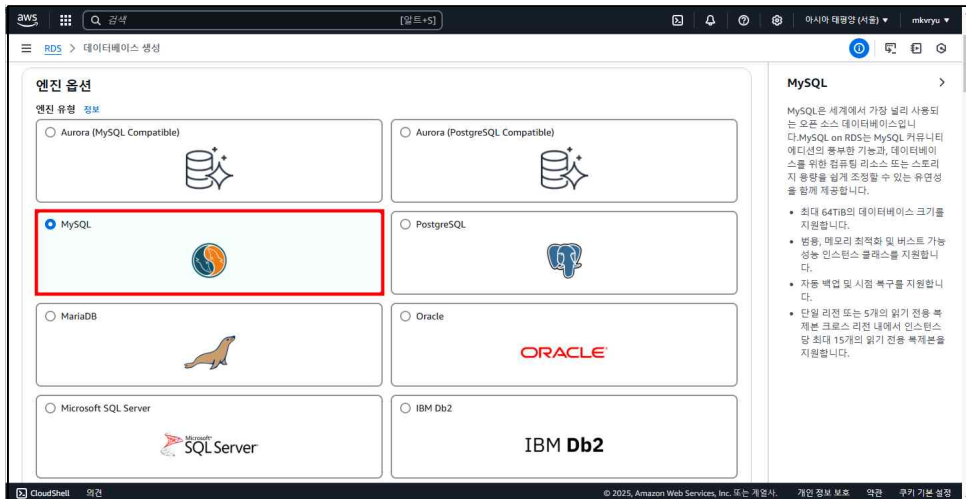
RDS1 데이터베이스가 생성되는 동안 두 번째 데이터베이스인 RDS2를 생성하는 작업도 진행한다.

Amazon RDS2 데이터베이스를 생성하기 위해 AWS 관리 콘솔에서 서비스 => 데이터베이스 => RDS를 선택하고 대시보드에서 “데이터베이스 생성”을 누른다.



RDS 데이터베이스 생성 페이지에서 다음과 같이 설정하고 맨 아래쪽에 있는 “데이터베이스 생성” 누르기

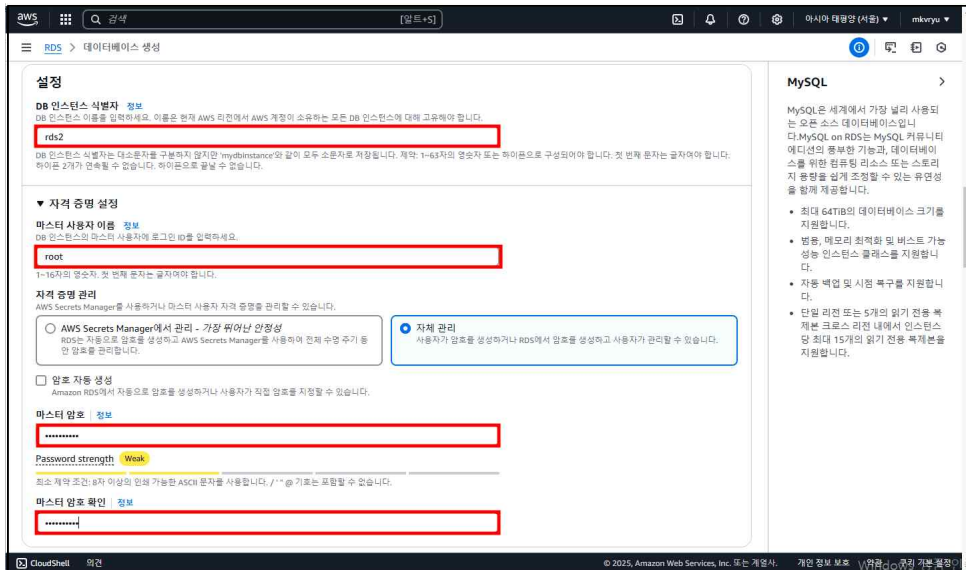
· 엔진 유형은 “MySQL” 선택



· 템플릿은 “프리 티어” 선택



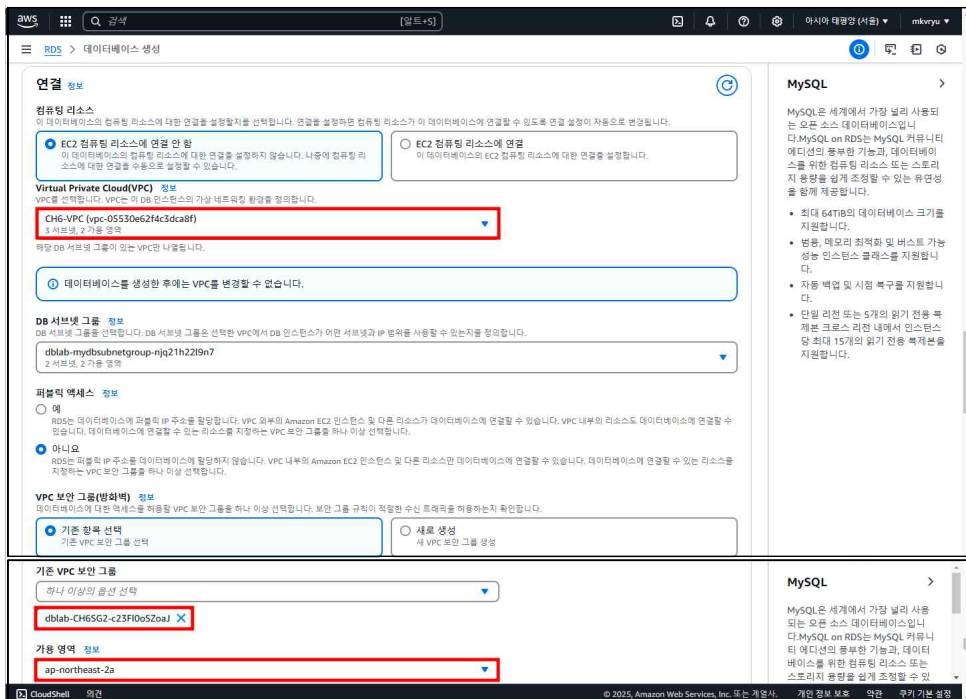
- 설정의 DB 인스턴스 식별자에 “rds2” 입력
- 마스터 사용자 이름에 “root” 입력
- 마스터 암호와 마스터 암호 확인에 “qwerty12345” 입력



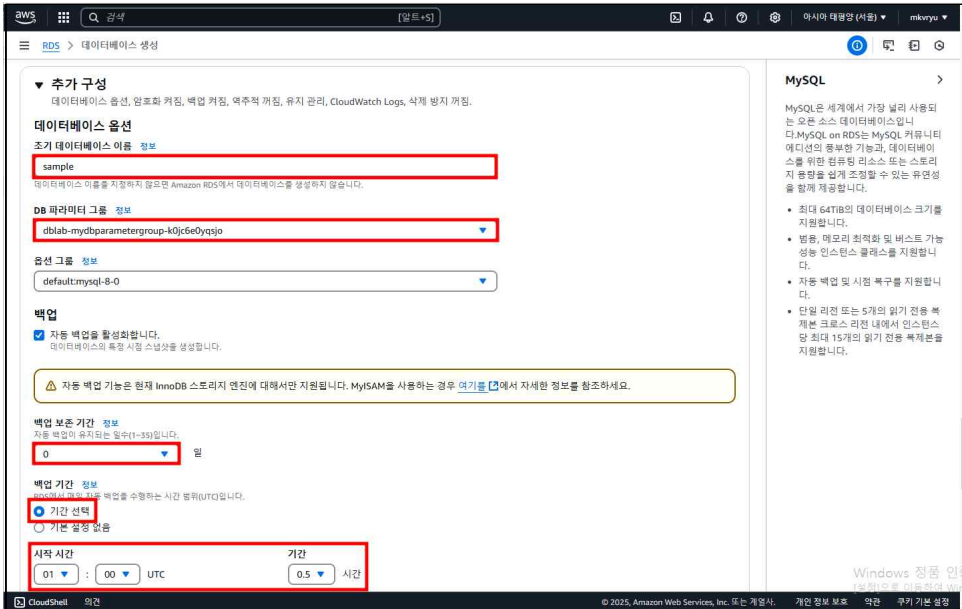
- “이전 세대 클래스 포함” 체크
- “버스터블 클래스(t클래스 포함) 선택
- “db.t3.micro” 선택



- Virtual Private Cloud(VPC)는 “CH6-VPC” 선택
- 기존 VPC 보안 그룹은 “default”를 제거한 후 “dblab-CH6SG2-XXXX” 선택
- 가용 영역은 “ap-northeast-2a” 선택



- 맨 아래쪽으로 내려가 “추가 구성”을 클릭하고 초기 데이터베이스 이름에 “sample” 입력
- DB 파라미터 그룹은 “dblab-mydbparametergroup-XXXX” 선택
- 백업 보존 기간은 “0”일 선택
- 백업 기간은 “기간 선택”을 선택하고 “01:00 UTC, 0.5시간”으로 지정



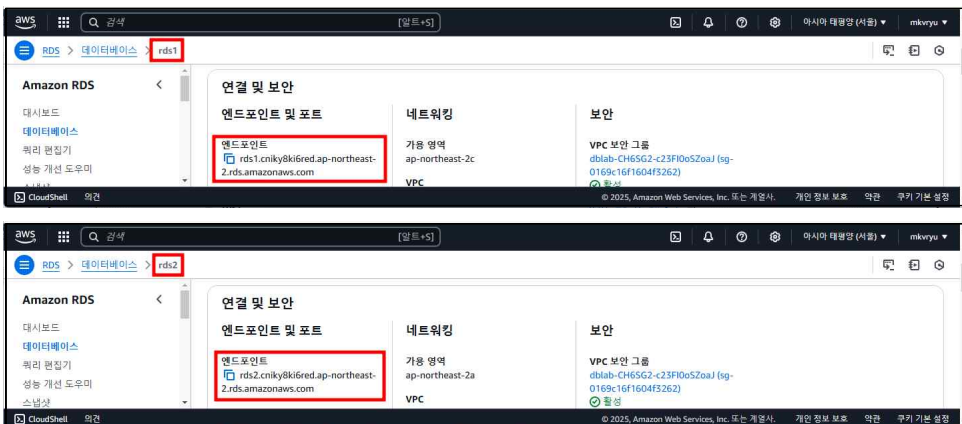
RDS 데이터베이스는 생성까지 약 4분 정도 소요된다. 이번에 생성한 RDS2 데이터베이스의 특징은 “Multi-AZ” 기능이 비활성화된 상태로, 백업 보존 기간을 0일로 설정하여 자동 백업 기능을 비활성화 했다. 아무래도 데이터베이스 하나만 단독으로 생성되므로 RDS1 보다는 빠르게 생성된다.

왼쪽 Amazon RDS => 데이터베이스 메뉴를 선택하면 생성된 데이터베이스를 확인할 수 있다. 정상적으로 데이터베이스가 생성되면 다음 그림과 같이 RDS1과 RDS2는 “사용 가능” 상태로 전환한 것을 확인할 수 있다.



이번에는 웹 서버(CH6-WebSrv) 및 Amazon RDS와 연동한다.

RDS1과 RDS2 데이터베이스의 엔드포인트 주소를 확인한다. 데이터베이스 메뉴에서 생성된 데이터베이스의 DB 식별자를 각각 클릭하면 아래쪽에 있는 “연결 & 보안 탭”에서 엔드포인트 주소를 확인할 수 있다.



CH6-WebSrv에 SSH로 접속하여 확인한 엔드포인트 주소를 변수로 선언한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1과 RDS2의 엔드포인트 주소를 변수로 선언(각자의 엔드포인트 주소로 입력)
[root@CH6-WebSrv ~]# RDS1=rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com
[root@CH6-WebSrv ~]# RDS2=rds2.cniky8ki6red.ap-northeast-2.rds.amazonaws.com

# 선언된 변수 호출
[root@CH6-WebSrv ~]# echo $RDS1
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com
[root@CH6-WebSrv ~]# echo $RDS2
rds2.cniky8ki6red.ap-northeast-2.rds.amazonaws.com
```

RDS1과 RDS2 데이터베이스에 MySQL 명령어로 접속하고 데이터베이스 정보를 확인한다. 참고로 MySQL 명령어에서 -h 옵션으로 데이터베이스 주소(\$RDS1)를 지정하고, -u 옵션으로 사용자 ID(root)를 입력하며, -p 옵션으로 암호(qwert12345)를 입력한다. 이렇게 접속한 데이터베이스에서 빠져나올 때는 Ctrl + C를 입력한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1 데이터베이스에 접속(RDS2도 동일하게 수행)
[root@CH6-WebSrv ~]# mysql -h $RDS1 -uroot -pqwe12345
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.39 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> status;
-----
mysql Ver 15.1 Distrib 5.5.68-MariaDB, for Linux (x86_64) using readline 5.1

Connection id:          17
Current database:
Current user:           root@10.6.1.10
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server:                 MySQL
Server version:         8.0.39 Source distribution
Protocol version:       10
Connection:             rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com via TCP/IP
Server characterset:    utf8mb3
Db      characterset:    utf8mb3
Client characterset:    utf8mb3
Conn.  characterset:    utf8mb3
TCP port:               3306
Uptime:                 29 min 16 sec
...
```

```
MySQL [(none)]> show databases;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sample             |
| sys                |
+-----+
5 rows in set (0.00 sec)
```

```
MySQL [(none)]> Ctrl-C -- exit!
Aborted
```

CH6-WebSrv의 index.php 파일을 수정하여 RDS2 데이터베이스와 연동한다.

```
# CH6-WebSrv에 SSH 터미널
# index.php 파일의 상위 다섯 줄만 확인
[root@CH6-WebSrv ~]# head -5 /var/www/html/index.php
<?php
define('DB_SERVER', 'dbaddress');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'qwe12345');
define('DB_DATABASE', 'sample');
```

index.php 파일에서 DB_SERVER 값은 임의로 설정되어 있어 데이터베이스 주소를 입력해야 한다. RDS2의 엔드포인트 주소를 해당 영역에 입력하기 위해 다음과 같이 명령어를 입력해야 한다.

sed의 기본 문자열 치환 사용법 => sed 's/패턴/교체/g' 파일명

```
# CH6-WebSrv에 SSH 터미널
# index.php 파일의 DB_SERVER 값을 RDS2 엔드포인트 주소로 치환
[root@CH6-WebSrv ~]# sed -i "s/dbaddress/$RDS2/g" /var/www/html/index.php
[root@CH6-WebSrv ~]# head -2 /var/www/html/index.php
<?php
define('DB_SERVER', 'rds2.cniky8ki6red.ap-northeast-2.rds.amazonaws.com');
```

DB_SERVER 주소는 변수로 선언한 RDS2 엔드포인트 주소로 변환되어 CH6-WebSrv와 RDS2 데이터베이스 간 연동 환경을 구성한다.

EC2 인스턴스(CH6-WebSrv)의 퍼블릭 IP 주소를 입력하여 인터넷 웹 브라우저로 접속한 후 데이터를 추가한다.



ID	NAME	ADDRESS
1	홍길동	서울시 여당가
2	임택정	경기도 여당가

CH6-WebSrv의 SSH 터미널에 접속하여 MySQL 명령어로 데이터베이스 테이블을 확인한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS2의 데이터베이스 EMPLOYEES 테이블 확인(1회)
[root@CH6-WebSrv ~]# mysql -h $RDS2 -uroot -pqwe12345 -e "USE sample;SELECT * FROM EMPLOYEES;"
+-----+-----+-----+
| ID | NAME      | ADDRESS          |
+-----+-----+-----+
| 1 | 홍길동    | 서울시 어딘가    |
| 2 | 임꺽정    | 경기도 어딘가    |
+-----+-----+-----+

[root@CH6-WebSrv ~]# while true; do mysql -h $RDS2 --connect-timeout=2 -uroot -pqwe12345 -e "USE sample;
SELECT * FROM EMPLOYEES;"; host $RDS2; date; sleep 1; done
+-----+-----+-----+
| ID | NAME      | ADDRESS          |
+-----+-----+-----+
| 1 | 홍길동    | 서울시 어딘가    |
| 2 | 임꺽정    | 경기도 어딘가    |
+-----+-----+-----+
rds2.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.2.67
Thu Jan  9 00:41:23 KST 2025
+-----+-----+-----+
| ID | NAME      | ADDRESS          |
+-----+-----+-----+
| 1 | 홍길동    | 서울시 어딘가    |
| 2 | 임꺽정    | 경기도 어딘가    |
+-----+-----+-----+
rds2.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.2.67
Thu Jan  9 00:41:24 KST 2025
...
```

RDS2 데이터베이스에 직접 진입하지 않고 명령어를 이용하여 EMPLOYEES 테이블을 확인할 수 있다. 1회성 명령어와 반복문 명령어가 가능하다.

RDS2가 중지될 경우 동작 확인하기

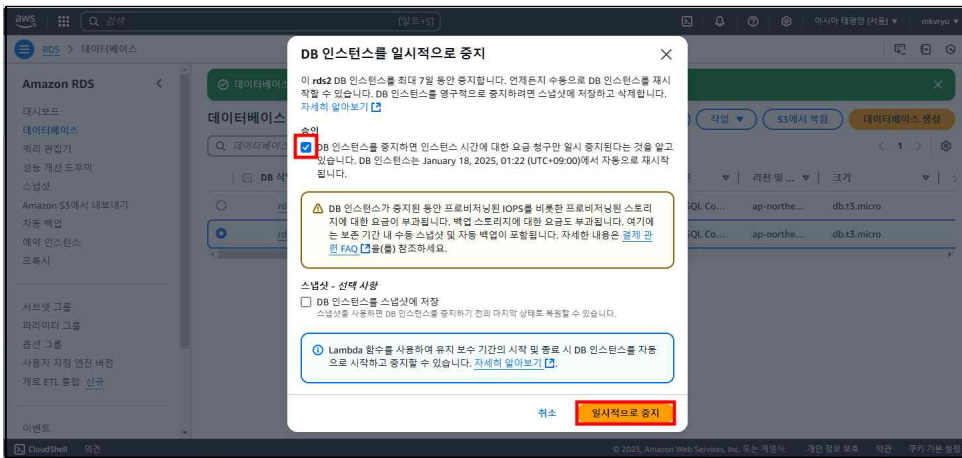
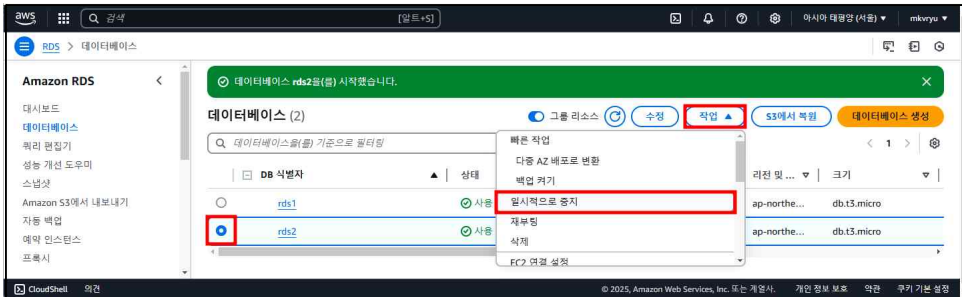
현재 RDS2는 Multi-AZ 기능이 동작하지 않고 단일 데이터베이스로 동작한다. 이 때문에 RDS2는 장애에 대한 페일오버(failover)(장애에 극복 기능)를 수행할 수 없다.

CH6-WebSrv의 SSH 터미널에 접속하여 MySQL 명령어로 테이블을 확인하는 반복문을 수행한다. 명령어가 길어 편의를 위해 스크립트(SELECT_TABLE_RDS2.sh)를 구성해 두었다.

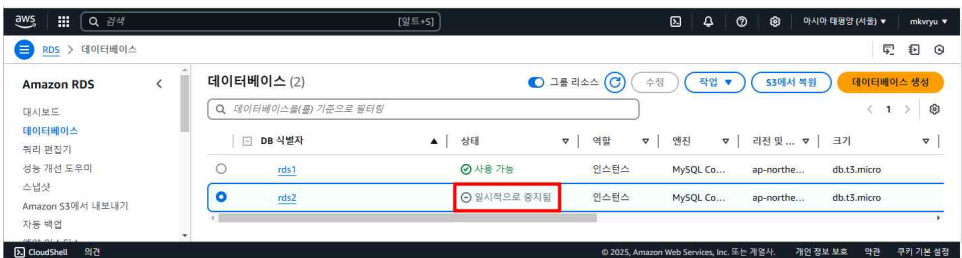
```
# CH6-WebSrv에 SSH 터미널
# RDS2의 데이터베이스 EMPLOYEES 테이블 확인 스크립트(반복문)
[root@CH6-WebSrv ~]# . /db_sh/SELECT_TABLE_RDS2.sh
+-----+-----+-----+
| ID | NAME      | ADDRESS          |
+-----+-----+-----+
| 1 | 홍길동    | 서울시 어딘가    |
| 2 | 임꺽정    | 경기도 어딘가    |
+-----+-----+-----+
```


rds2.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.2.222
Sat Jan 11 00:53:31 KST 2025
...

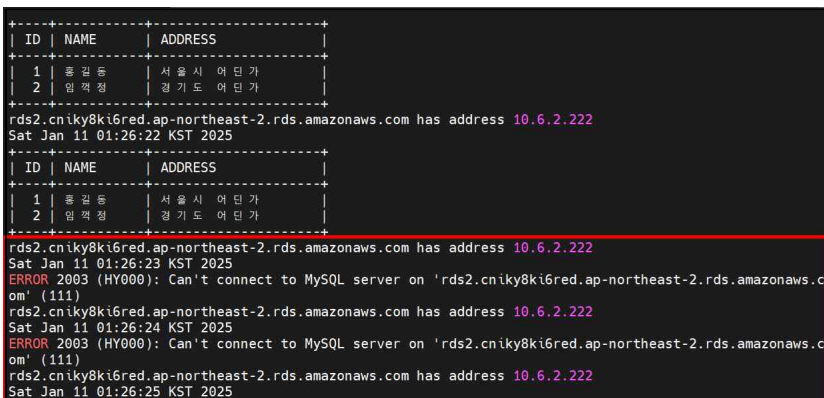
RDS2 EMPLOYEES 테이블을 확인하는 반복문을 유지한 상태에서 RDS2 데이터베이스를 중지한다. RDS2를 선택한 후 작업 => 일시적으로 중지를 선택한다. 이때 버튼을 클릭하면 팝업창이 열리는데 “7일 후 DB 인스턴스가 자동으로 다시 시작하는 것에 동의합니다.”에 체크한 후 “일시적으로 중지”를 클릭한다.



일정 시간이 지나면 RDS2 상태 정보는 “중지 중” 상태를 거쳐 “일시적으로 중지됨” 상태로 전환된다.

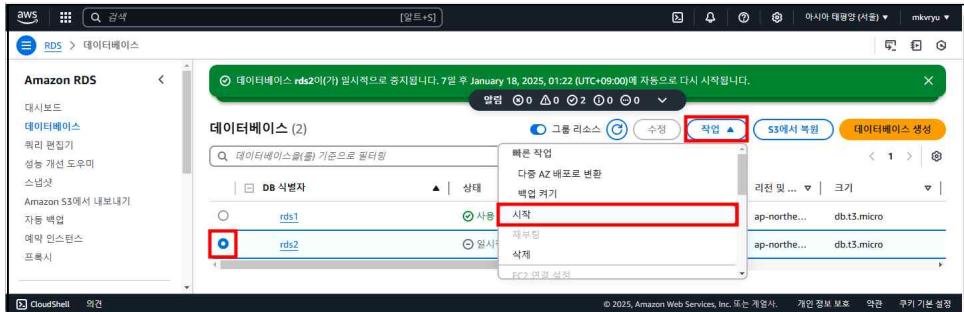


데이터베이스 테이블을 확인하는 반복문을 다시 확인한다. 다음 그림과 같이 RDS2가 중지되어 데이터베이스 테이블 정보를 가져오지 못하고 있다.



RDS2 데이터베이스는 단일로 동작하고 있기 때문에 중지되거나 재부팅이 발생하면 데이터베이스를 찾을 수 없어 문제가 발생된다.

장애에 따라 지속적인 서비스가 불가능한 문제를 해결하기 위해 Multi-AZ 기능을 확인해 본다. 다음 실습을 위해 중지된 RDS2 데이터베이스를 다시 시작한다. RDS2를 선택한 후 작업 => 시작을 선택하면 다시 시작한다.



6.3.3. Amazon RDS의 고가용성을 위한 Multi-AZ 동작 확인하기

이번 단계에서는 앞서 제기된 문제를 해결하기 위해 Multi-AZ 기능을 알아본다. CH6-WebSrv를 Multi-AZ 기능이 활성화된 RDS1로 연결한다.

CH6-WebSrv의 index.php 파일을 수정하여 RDS1 데이터베이스와 연동한다.

```
# CH6-WebSrv에 SSH 터미널
# index.php 파일의 DB_SERVER 값을 RDS1 엔드포인트 주소로 치환
[root@CH6-WebSrv ~]# sed -i "s/$RDS2/$RDS1/g" /var/www/html/index.php
[root@CH6-WebSrv ~]# head -2 /var/www/html/index.php
<?php
define('DB_SERVER', 'rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com');
```

DB_SERVER 주소를 RDS1 엔드포인트 주소로 변환하여 CH6-WebSrv와 RDS1 데이터베이스 간 연동 환경을 구성한다.

RDS1 데이터베이스의 EMPLOYEES 테이블 정보를 확인한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1의 데이터베이스 EMPLOYEES 테이블 확인 스크립트(반복문)
[root@CH6-WebSrv ~]# ./db_sh/SELECT_TABLE_RDS1.sh
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.3.193
Sun Jan 12 19:35:14 KST 2025
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.3.193
Sun Jan 12 19:35:15 KST 2025
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.3.193
Sun Jan 12 19:35:16 KST 2025
...
```

EMPLOYEES 테이블을 만든다.

```
# CH6-WebSrv에 SSH 터미널
# EMPLOYEES 테이블 생성
[root@CH6-WebSrv ~]# mysql -h $RDS1 -uroot -pqwe12345 -e "USE sample;CREATE TABLE EMPLOYEES(ID int, NAME CHAR(20), ADDRESS CHAR(20));"
```

```
ERROR 1050 (42S01) at line 1: Table 'EMPLOYEES' already exists
```

RDS1에 EMPLOYEES 테이블이 있으면 위와 같이 에러 메시지가 발생되고 EMPLOYEES 테이블이 없으면 테이블이 생성된다. 다시 RDS1의 EMPLOYEES 테이블을 확인하는 스크립트(. /db_sh/SELECT_TABLE_RDS1.sh)를 실행하면 오류 메시지는 발생되지 않는다.

INSERT 명령어로 RDS1의 EMPLOYEES 테이블에 데이터를 추가한다.

```
# CH6-WebSrv에 SSH 터미널
# EMPLOYEES 테이블에 데이터 추가
[root@CH6-WebSrv ~]# mysql -h $RDS1 -uroot -pqwe12345 -e "USE sample;INSERT INTO EMPLOYEES VALUES ('1', 'Son', 'UK');"

```

RDS1 데이터베이스의 EMPLOYEES 테이블의 정보를 확인한다. RDS1 데이터베이스의 EMPLOYEES 테이블에 데이터가 추가된 것을 확인할 수 있다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1의 데이터베이스 EMPLOYEES 테이블 확인 스크립트(반복문)
[root@CH6-WebSrv ~]# . /db_sh/SELECT_TABLE_RDS1.sh
+----+-----+-----+
| ID | NAME | ADDRESS |
+----+-----+-----+
| 1 | Son | UK      |
+----+-----+-----+
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.3.193
Sun Jan 12 19:54:01 KST 2025
+----+-----+-----+
| ID | NAME | ADDRESS |
+----+-----+-----+
| 1 | Son | UK      |
+----+-----+-----+
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.3.193
Sun Jan 12 19:54:02 KST 2025
...

```

CH6-WebSrv는 RDS1과 연동되어 SQL 질의와 응답을 이용하여 데이터를 생성하고 조회할 수 있다. 여기에서 RDS1은 Multi-AZ가 활성화되어 Primary DB와 Standby Prplica는 서로 다른 가용 영역에 위치한다.

Primary DB와 Standby Replica는 서로 동기화되어 테이블을 유지하며, Primary DB에 장애가 발생하면 Standby Replica가 Promary DB로 승격되어 페일오버를 수행할 수 있다.

RDS1 데이터베이스를 재부팅했을 때 페일오버 동작을 확인한다.

RDS1 데이터베이스의 EMPLOYEES 테이블 정보를 반복하는 SELECT_TABLE_RDS1.sh 스크립트를 실행하고 유지한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1의 데이터베이스 EMPLOYEES 테이블 확인 스크립트(반복문)
[root@CH6-WebSrv ~]# . /db_sh/SELECT_TABLE_RDS1.sh
+----+-----+-----+

```

ID	NAME	ADDRESS
1	Son	UK

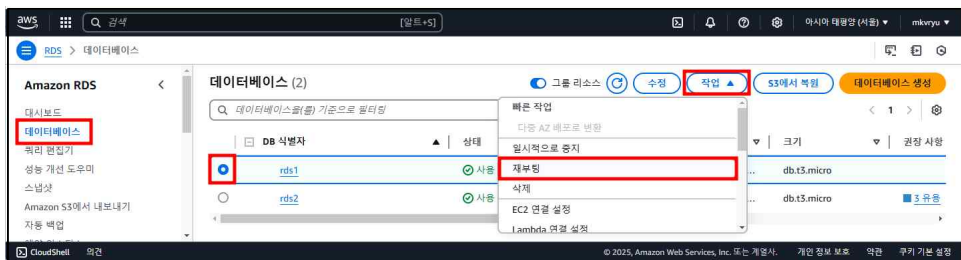
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.3.193

Sun Jan 12 20:03:12 KST 2025

...

여기에서 RDS1의 IP 주소는 10.6.3.193이다. 물론 각자 실습한 IP 주소는 다르다.

RDS1을 재부팅하기 위해 데이터베이스 메뉴에서 “RDS1”을 선택하고 작업 => 재부팅을 선택한다. 재부팅 창이 열리면 “장애 조치로 재부팅하시겠습니까?”에 체크하고 “확인”을 누른다.



앞서 유지하고 있는 스크립트를 다시 확인한다.

```
ERROR 2003 (HY000): Can't connect to MySQL server on 'rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com' (4)
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.3.193
Sun Jan 12 20:10:33 KST 2025
ERROR 2003 (HY000): Can't connect to MySQL server on 'rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com' (4)
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.3.193
Sun Jan 12 20:10:36 KST 2025
+----+-----+-----+
| ID | NAME | ADDRESS |
+----+-----+-----+
| 1 | Son | UK |
+----+-----+-----+
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.2.85
Sun Jan 12 20:10:37 KST 2025
+----+-----+-----+
| ID | NAME | ADDRESS |
+----+-----+-----+
| 1 | Son | UK |
+----+-----+-----+
rds1.cniky8ki6red.ap-northeast-2.rds.amazonaws.com has address 10.6.2.85
Sun Jan 12 20:10:38 KST 2025
```

RDS1을 재부팅하면서 Primary DB는 중지될 것이고 장애 조치를 위해 Standby Replica를 Primary DB로 승격시켜 자동으로 페일오버를 수행한다. 여기에서 RDS1의 IP 주소를 확인해 보면 최초 10.6.3.193에서 10.6.2.85로 변경된 것을 확인할 수 있다. 이 의미는 Primary DB가 변경된 것으로 이해할 수 있다.

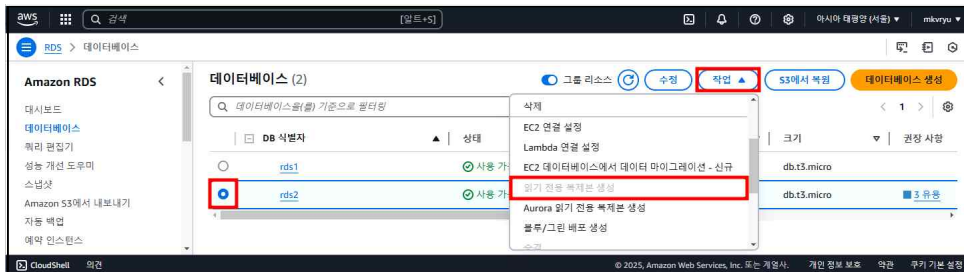
RDS1의 Primary DB에 장애가 발생하면 Multi-AZ 설정에 따라 다른 가용 영역에 생성된 Standby Replica를 자동으로 승급하기 때문에 Primary DB 역할로 서비스를 지속적으로 유지할 수 있다.

6.3.4. Amazon RDS의 성능을 확장하는 Read Replica 동작 확인하기

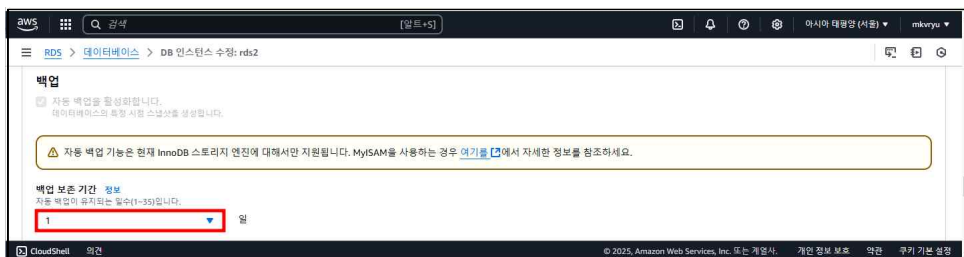
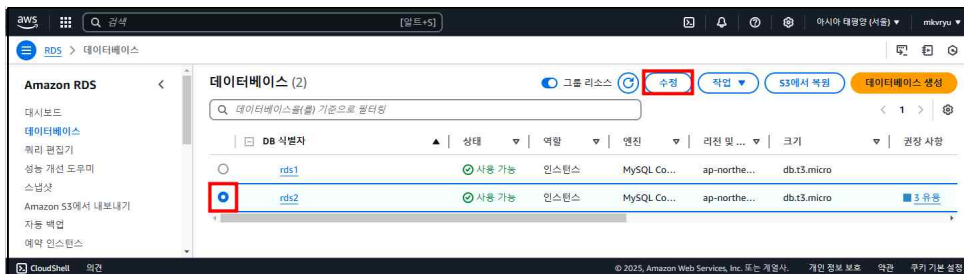
이번에는 데이터베이스의 데이터 처리 성능 확장을 위한 Read Replica 기능을 살펴본다. 참고로 Read Replica는 앞서 설명한 Multi-AZ 같은 고가용성 서비스가 아니다. 읽기 전용 데이터베이스인 Read Replica 데이터베이스를 복제

하여 데이터 처리 성능을 높이는 기능이다.

RDS2 데이터베이스의 Read Replica를 설정하기 위해 데이터베이스 메뉴에서 RDS2를 선택하고 작업 > 읽기 전용 복제본 생성을 선택한다. 하지만 RDS2의 Read Replica를 설정하는 읽기 전용 복제본 생성 메뉴가 비활성화되어 설정할 수 없다.



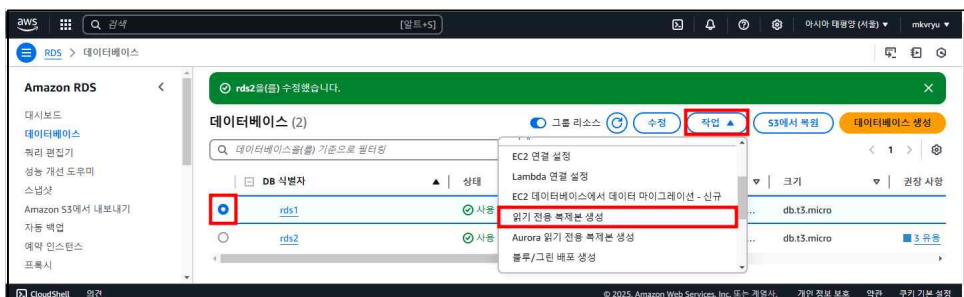
Read Replica 설정 조건 중에는 자동 백업 기능이 활성화 상태여야 한다는 조건이 있다. 처음 RDS2를 생성할 때 백업 보존 기간을 “0일”로 설정했다. 이는 자동 백업을 비활성화한다는 의미이다. 그래서 RDS2 데이터베이스는 Read Replica 설정을 할 수 없다. 설정을 원한다면 RDS2 데이터베이스를 선택한 후 “수정”을 눌러 백업 보존 기간을 1일 이상의 값으로 변경해야 한다.



데이터베이스 수정에 따른 대기 시간이 발생하는 관계로 이번에는 Read Replica의 제약 조건이 자동 백업 기능을 활성화해야 한다는 점만 이해하고 넘어간다. 실제 Read Replica 실습은 RDS1에서 진행한다.

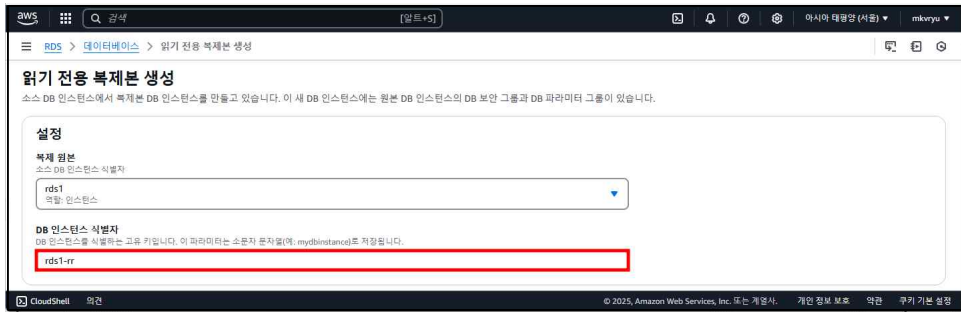
RDS1의 Read Replica 설정으로 읽기 전용의 데이터베이스 복제본을 생성한다. 최초 RDS1 설정값 중 백업 보존 기간을 35일로 설정했다. 자동 백업 기능이 활성화된 상태로 Read Replica 설정에 대한 제약이 없다.

RDS1 데이터베이스의 Read Replica 설정을 위해 데이터베이스 메뉴에서 RDS1를 선택하고 작업 => 읽기 전용 복제본 생성을 선택한다.

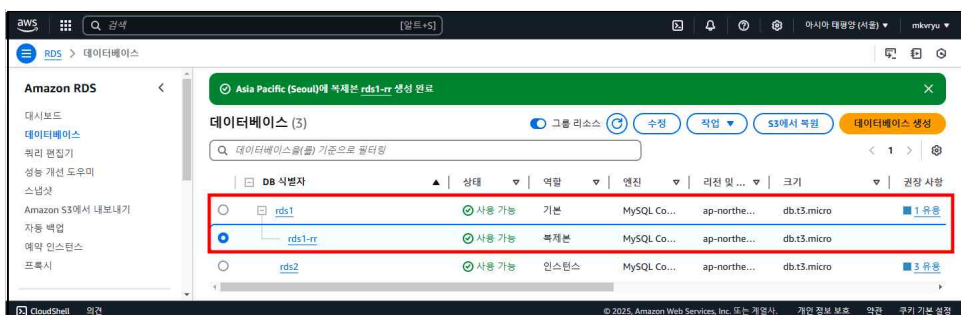


DB 인스턴스의 인스턴스 읽기 전용 복제본 생성 페이지가 나타나면 다음과 같이 설정한다.

- DB 인스턴스 식별자는 “rds1-rr”로 입력
- 가장 아래쪽에 있는 “읽기 전용 복제본 생성” 누르기



일정 시간을 대기하면 다음 그림과 같이 RDS1에 대한 Read Replica가 생성된 것을 확인할 수 있다.



RDS1-RR 생성이 완료되면 DB 식별자를 클릭하고 엔드포인트 주소를 메모해 둔다. CH6-WebSrv에 SSH로 접속하여 메모해 둔 RDS1-RR의 엔드포인트 주소를 변수로 선언한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1-RR의 엔드포인트 주소를 변수로 선언
[root@CH6-WebSrv ~]# RDS1RR=rds1-rr.cniky8ki6red.ap-northeast-2.rds.amazonaws.com
[root@CH6-WebSrv ~]# echo $RDS1RR
rds1-rr.cniky8ki6red.ap-northeast-2.rds.amazonaws.com
```

RDS1과 RDS1-RR의 데이터베이스 EMPLOYEES 테이블을 확인한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1의 EMPLOYEES 테이블 확인
[root@CH6-WebSrv ~]# mysql -h $RDS1 -uroot -pqwe12345 -e "USE sample;SELECT * FROM EMPLOYEES;"
+----+-----+-----+
| ID | NAME | ADDRESS |
+----+-----+-----+
| 1 | Son | UK |
+----+-----+-----+
```



```
# RDS1-RR의 EMPLOYEES 테이블 확인
[root@CH6-WebSrv ~]# mysql -h $RDS1RR -uroot -pqwe12345 -e "USE sample;SELECT * FROM EMPLOYEES;"
+----+-----+-----+
| ID | NAME | ADDRESS |
+----+-----+-----+
| 1 | Son  | UK      |
+----+-----+-----+
```

RDS1-RR은 RDS1의 복제본 데이터베이스이기 때문에 서로 동일한 테이블을 유지한다.

MySQL INSERT 명령어를 이용하여 RDS1의 EMPLOYEES 테이블에 데이터를 추가한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1의 EMPLOYEES 테이블에 데이터 추가
[root@CH6-WebSrv ~]# mysql -h $RDS1 -uroot -pqwe12345 -e "USE sample;INSERT INTO EMPLOYEES VALUES ('2',
'Park', 'Suwon');"

```

RDS1과 RDS1-RR의 데이터베이스 EMPLOYEES 테이블을 확인한다.

```
# CH6-WebSrv에 SSH 터미널
# RDS1의 EMPLOYEES 테이블 확인
[root@CH6-WebSrv ~]# mysql -h $RDS1 -uroot -pqwe12345 -e "USE sample;SELECT * FROM EMPLOYEES;"
+----+-----+-----+
| ID | NAME | ADDRESS |
+----+-----+-----+
| 1 | Son  | UK      |
| 2 | Park | Suwon   |
+----+-----+-----+

# RDS1-RR의 EMPLOYEES 테이블 확인
[root@CH6-WebSrv ~]# mysql -h $RDS1RR -uroot -pqwe12345 -e "USE sample;SELECT * FROM EMPLOYEES;"
+----+-----+-----+
| ID | NAME | ADDRESS |
+----+-----+-----+
| 1 | Son  | UK      |
| 2 | Park | Suwon   |
+----+-----+-----+
```

마찬가지로 RDS1과 RDS1-RR은 서로 동기화되어 테이블을 유지한다.

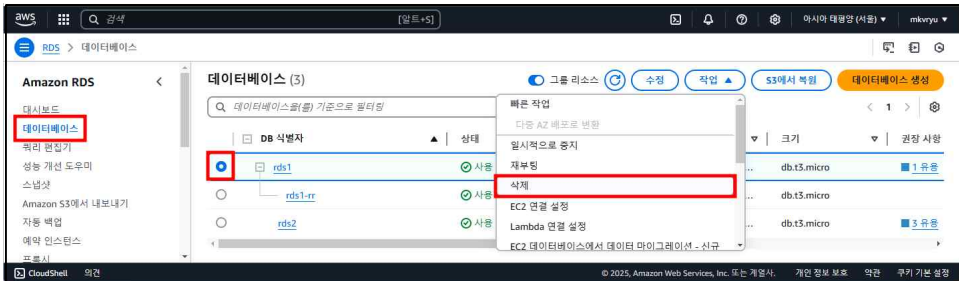
INSERT 명령어를 이용하여 RDS1-RR의 EMPLOYEES 테이블에 데이터를 추가하면 어떻게 될까?

```
# CH6-WebSrv에 SSH 터미널
# RDS1의 EMPLOYEES 테이블에 데이터 추가
[root@CH6-WebSrv ~]# mysql -h $RDS1RR -uroot -pqwe12345 -e "USE sample;INSERT INTO EMPLOYEES VALUES
('3', 'Lee', 'China');"
ERROR 1290 (HY000) at line 1: The MySQL server is running with the --read-only option so it cannot
execute this statement
```

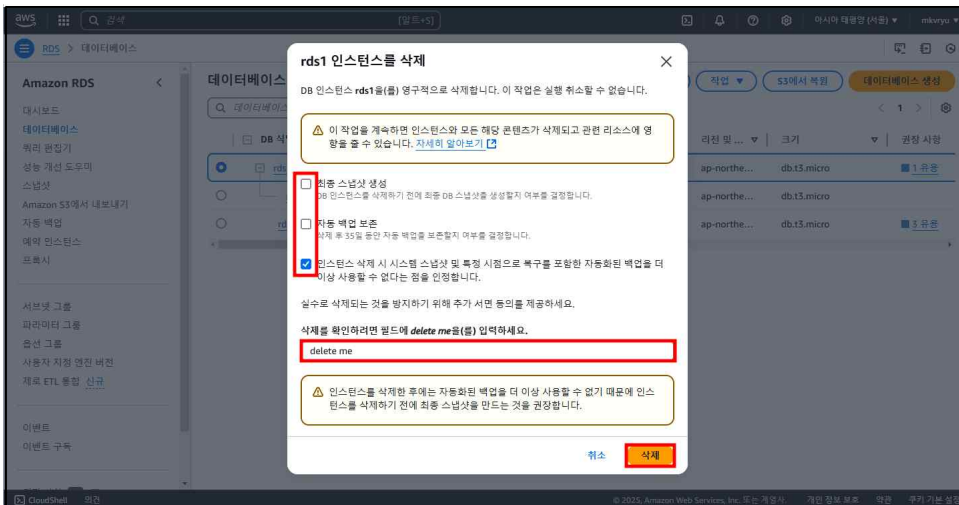
RDS1-RR은 읽기 전용(read-only) 데이터베이스라는 오류 메시지를 출력하며 데이터를 추가하지 못한다.

6.3.5. 실습을 위해 생성된 모든 자원 삭제하기

서비스 => 데이터베이스 => RDS => 데이터베이스 메뉴에 들어간다. RDS1, RDS1-RR, RDS2를 각각 선택한 후 작업 => 삭제를 선택한다.



- “최종 스냅샷 생성”, “자동 백업 보존” 체크 해제
- “인스턴스 삭제 시 시스템 스냅샷 및 특정 시점으로 복구를 ~~~~ 점을 인정합니다.”에 체크
- “삭제를 확인하려면 필드에 delete me를 입력하세요.”에 delete me 입력
- “삭제” 누르기



삭제 작업은 약 10분 내외가 소요된다. RDS가 삭제되어야 CloudFormation 스택을 삭제할 수 있다.

이제 CloudFormation 스택을 삭제한다. 서비스 => 관리 및 거버넌스 => CloudFormation => 스택 메뉴에 들어가서 “dblab” 스택을 선택한 후 “삭제”를 누른다. 이후 나타나는 창에서 “스택 삭제”를 누른다.

