

- AWS RDS(MariaDB) 인스턴스 생성하기
- 1. RDS 인스턴스 생성하기
- 2. RDS 운영환경에 맞는 파라미터 설정하기
 - 파라미터 그룹 생성하기
 - 2-1. 타임존
 - 2-2. Character Set
 - 2-3. Max Connection
- 3. 로컬에서 RDS 접속하기
- 4. EC2에서 RDS 접속하기

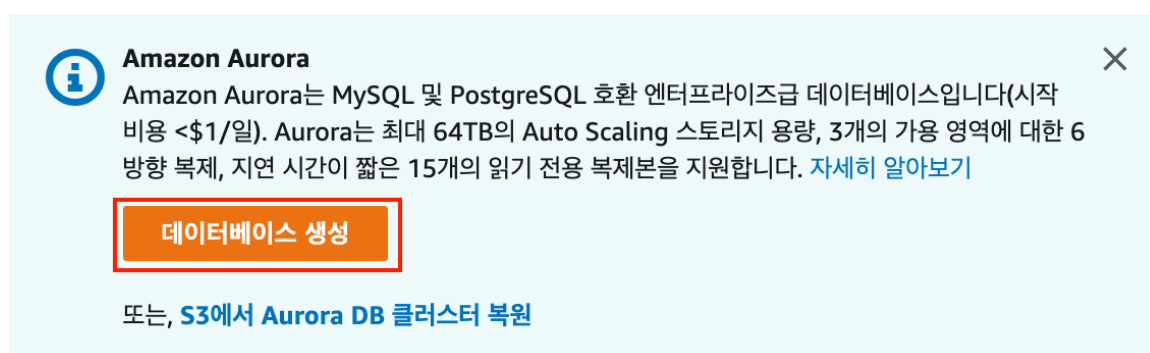
AWS RDS(MariaDB) 인스턴스 생성하기

웹 서비스의 백엔드를 다룬다고 했을 때 애플리케이션 코드를 작성하는 것 만큼 중요한 것이 데이터베이스를 다루는 일이다. 백엔드 개발자가 직접 데이터베이스를 설치해서 다루게 되면 모니터링, 알람, 백업, HA 구성 드을 모두 직접 해야만 한다. 처음 구축할 때 며칠이 걸리는 일이다.

AWS에서는 앞에서 언급한 작업을 모두 지원하는 **관리형 서비스인 RDS(Relational Database Service)**를 제공한다. RDS는 AWS에서 지원하는 클라우드 기반 관계형 데이터베이스이다. 하드웨어 프로비저닝, 데이터베이스 설정, 패치 및 백업과 같이 지원하는 서비스이다. 추가로 조정 가능한 용량을 지원하여 예상치 못한 양의 데이터가 쌓여도 비용만 추가로 내면 정상적으로 서비스가 가능한 장점도 있다.

1. RDS 인스턴스 생성하기

1) rds에 접속 > **데이터베이스 생성** 버튼 클릭



데이터베이스 생성

2) 표준 생성 > 엔진 : **MariaDB** 선택 > 템플릿 : **프리 티어** 선택 >

MariaDB선택 이유는 **다음 글**에 정리했다. 간단하게 정리하면 다음과 같다.

- 첫 번째 이유로는 MySQL은 사용 경험이 있지만 MariaDB는 아직 사용해본 적이 없기 때문이다.
- 두 번째 이유로는 MySQL과 많은 기능이 흡사하기 때문에 익숙했기 때문이다. PostgreSQL은 한번도 접해보지 않은 DB라서 뭔지 모를 거부감이 들었다.

- 세 번째 이유로 미래적인 관점으로 MariaDB가 더 발전가능성이 많아 보였기 때문이다. 프로젝트를 길게 끌고간다면 당연히 개발 커뮤니티가 활발한 DB를 택하는 것이 맞다고 생각한다. (커뮤니티가 활발하면 구글링만으로도 에러 해결이 풀리는 경우가 많아 아주 편리하다. ㅎㅎㅎ)

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

☐ Amazon Aurora


☐ MySQL


☒ MariaDB


☐ PostgreSQL


☐ Oracle


☐ Microsoft SQL Server


버전

MariaDB 10.4.13

▼

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](#)

엔진 옵션, 템플릿 설정

3) 설정 > DB 인스턴스 식별자 입력 > 마스터 사용자 정보 (마스터 사용자 이름, 암호) 등록

본인만의 DB 인스턴스 이름과 사용자 정보를 등록한다. 여기서 사용된 정보로 실제 DB에 접근하게 되니 꼭 마스터 정보는 기억해야 한다.

설정

DB 인스턴스 식별자 [정보](#)

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

spring-deploy-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.

loosie

1~16자의 영숫자. 첫 번째 문자는 글자이어야 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](#)

.....

설정

4) DB 인스턴스 **db.t2.micro** > 스토리지 > 할당된 스토리지 : 20

해당 부분은 기본으로 설정되어 있으므로 확인만하고 넘어가면 된다. db.t2.micro - 1 vCPU, 1GiB RAM/ 할당된 스토리지 : 20

DB 인스턴스 클래스

DB 인스턴스 클래스 [정보](#)

☐ 스탠다드 클래스(m 클래스 포함)

☐ 메모리 최적화 클래스(r 및 x 클래스 포함)

☒ 버스터블 클래스(t 클래스 포함)

db.t2.micro

1 vCPUs 1 GiB RAM Not EBS Optimized

☐ 이전 세대 클래스 포함

스토리지

스토리지 유형 [정보](#)

범용(SSD)

할당된 스토리지

20

GiB

(최소: 20GiB, 최대: 16,384GiB) 할당된 스토리지가 많으면 IOPS 성능이 개선될 수 있습니다.

스토리지 자동 조정 [정보](#)

애플리케이션의 필요에 따라 데이터베이스 스토리지의 동적 조정 지원을 제공합니다.

☒ 스토리지 자동 조정 활성화

이 기능을 활성화하면 지정한 임계값 초과 시 스토리지를 늘릴 수 있습니다.

최대 스토리지 임계값 [정보](#)

데이터베이스를 지정한 임계값으로 자동 조정하면 요금이 부과됩니다.

1000

GiB

최소: 21 GiB, 최대: 16,384 GiB

인스 턴스, 스토리지

5) 네트워크 설정 > 퍼블릭 액세스 : 예

이후 보안 그룹에서 지정된 IP만 접근하도록 막을 예정이다.

연결

Virtual Private Cloud(VPC) [정보](#)

이 DB 인스턴스의 가상 네트워킹 환경을 정의하는 VPC.

Default VPC (vpc-0624936d)

해당 DB 서브넷 그룹이 있는 VPC만 나열됩니다.

데이터베이스를 생성한 후에는 VPC 선택을 변경할 수 없습니다.

서브넷 그룹 [정보](#)

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

default-vpc-0624936d

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.

VPC 보안 그룹

데이터베이스에 대한 액세스를 허용할 VPC 보안 그룹을 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

VPC 보안 그룹 선택

default

가용 영역 [정보](#)

기본 설정 없음

네트워크 및 보안

6) 추가 구성 > 데이터베이스 옵션 > 초기 데이터베이스 이름 입력 > 데이터베이스 생성

추가구성은 초기 데이터베이스 이름만 입력해주고 데이터베이스 생성 버튼을 눌러서 DB 설정을 마무리한다.

추가 구성

데이터베이스 옵션, 백업 활성화됨, 역추적 비활성화됨, 향상된 모니터링 비활성화됨, 유지 관리, CloudWatch Logs, 삭제 보호 비활성화됨

데이터베이스 옵션

초기 데이터베이스 이름 [정보](#)

spring_deploy_test

데이터베이스 이름을 지정하지 않으면 Amazon RDS에서 데이터베이스를 생성하지 않습니다.

DB 파라미터 그룹 [정보](#)

default.mariadb10.4

옵션 그룹 [정보](#)

default:mariadb-10-4

추가 구성

rds 인스턴스가 생성될 때까지 기다렸다가 생성되면 다음 단계로 넘어가 운영환경에 맞는 파라미터를 설정해보도록 하자.

RDS > 데이터베이스 > spring-deploy-test

spring-deploy-test

수정

작업 ▼

요약

DB 식별자 spring-deploy-test	CPU -	상태 🕒 생성 중	클래스 db.t2.micro
역할 인스턴스	현재 활동	엔진 MariaDB	리전 및 AZ ap-northeast-2c

RDS 인스턴스 생성중

2. RDS 운영환경에 맞는 파라미터 설정하기

RDS를 처음 생성하면 몇 가지 설정을 필수로 해야 한다. 우선 다음 3개의 설정을 차례로 진행해보자. 왼쪽 카테고리에서 **[파라미터 그룹]** 탭을 클릭해서 이동하자.

파라미터 그룹 생성하기

1) **파라미터 그룹** 탭으로 이동 > **파라미터 그룹 생성** 버튼 클릭

사용자 지정 파라미터 그룹 (3)

🔄

작업 ▼

파라미터 그룹 생성

🔍

custom 파라미터 그룹 필터

파라미터 그룹 생성

2) 파라미터 그룹 세부 정보 > 파라미터 그룹 패밀리 : **maridadb10.4** 선택 > **그룹 이름, 설명** 입력 후 생성

파라미터 그룹 패밀리는 방금 RDS에서 생성한 MariaDB와 같은 버전으로 맞춰야 한다.

파라미터 그룹 세부 정보

파라미터 그룹 패밀리
이 DB 파라미터 그룹을 적용할 DB 패밀리

mariadb10.4

그룹 이름
DB 파라미터 그룹의 식별자

spring-deploy-test

설명
DB 파라미터 그룹에 대한 설명

spring-deploy-test

취소

생성

파라미터 그룹 세부 정보 설정

3) 파라미터 > 수정 버튼 클릭

생성한 파라미트 그룹을 선택하여 파라미터를 하나씩 변경해주자.

파라미터 (618) 정보

수정

파라미터

파라미터 편집

2-1. 타임존

파라미터 필드에 **time_zone**을 입력한 후 **Asia_Seoul**을 입력 후 선택한다.

수정 가능한 파라미터 (1539)

time_zone

☒

이름

값

☒

time_zone

Asia/Seoul

타임존 설정

2-2. Character Set

다음으로 Character Set을 변경한다. Character Set은 항목이 많다.총 8개 항목으로 character 항목들은 모두 **utf8mb4**로, collation 항목들은 **utf8mb4_general_ci**로 변경한다. utf8과 utf8mb4의 차이는 이모지 저장 가능 여부이다.

수정 가능한 파라미터 (6539)		
<div> <div> <div></div> <div>character_set</div> </div> </div>		
<div> <div></div> <div>이름</div> <div></div> </div>	값	
<div> <div></div> <div>character_set_server</div> </div>	<div>utf8mb4</div> <div> <div>허용된 값</div> <div>big5,dec8,cp850,hp8,koi8r,latin1,latin2,swe7,ascii,ujis,sjis,hebr</div> </div>	
<div> <div></div> <div>character_set_results</div> </div>	<div>utf8mb4</div> <div> <div>허용된 값</div> <div>big5,dec8,cp850,hp8,koi8r,latin1,latin2,swe7,ascii,ujis,sjis,hebr</div> </div>	
<div> <div></div> <div>character_set_filesystem</div> </div>	<div>utf8mb4</div> <div> <div>허용된 값</div> <div>big5,dec8,cp850,hp8,koi8r,latin1,latin2,swe7,ascii,ujis,sjis,hebr</div> </div>	
<div> <div></div> <div>character_set_database</div> </div>	<div>utf8mb4</div> <div> <div>허용된 값</div> <div>big5,dec8,cp850,hp8,koi8r,latin1,latin2,swe7,ascii,ujis,sjis,hebr</div> </div>	
<div> <div></div> <div>character_set_connection</div> </div>	<div>utf8mb4</div> <div> <div>허용된 값</div> <div>big5,dec8,cp850,hp8,koi8r,latin1,latin2,swe7,ascii,ujis,sjis,hebr</div> </div>	
<div> <div></div> <div>character_set_client</div> </div>	<div>utf8mb4</div> <div> <div>허용된 값</div> <div>big5,dec8,cp850,hp8,koi8r,latin1,latin2,swe7,ascii,ujis,sjis,hebr</div> </div>	

수정 가능한 파라미터 (2539)

이름	값
<input type="checkbox"/> rocksdb_strict_collation_exceptions	<input type="text"/>
<input type="checkbox"/> rocksdb_strict_collation_check	<input type="text"/>
<input type="checkbox"/> rocksdb_error_on_suboptimal_collation	<input type="text"/>
<input checked="" type="checkbox"/> collation_server	<input type="text" value="utf8mb4_general_ci"/> 허용된 값 big5_chinese_ci, big5_bin, dec8_swedish_ci, dec8_bin, cp
<input checked="" type="checkbox"/> collation_connection	<input type="text" value="utf8mb4_general_ci"/> 허용된 값 big5_chinese_ci, big5_bin, dec8_swedish_ci, dec8_bin, cp

collation 설정

2-3. Max Connection

마지막으로 Max Connection을 수정한다. RDS의 Max Connection은 인스턴스 사양에 따라 자동으로 정해진다. 현재 프리티어 사양으로는 약 60개의 커넥션만 가능해서 좀 더 넉넉한 값을 지정한다.

수정 가능한 파라미터 (1539)

max_connection

<input checked="" type="checkbox"/>	이름	값
<input checked="" type="checkbox"/>	max_connections	150 허용된 값 1-100000

변경 사항 검토			
<div>수정 사항 요약</div> <div>다음 수정 사항을 제출하려고 합니다. 변경되는 값만 표시됩니다.</div>			
이름	현재 값	새 값	적용 유형
character_set_client		utf8mb4	dynamic
character_set_connection		utf8mb4	dynamic
character_set_database		utf8mb4	dynamic
character_set_filesystem		utf8mb4	dynamic
character_set_results		utf8mb4	dynamic
character_set_server		utf8mb4	dynamic
collation_connection		utf8mb4_general_ci	dynamic
collation_server		utf8mb4_general_ci	dynamic
max_connections	{DBInstanceClassMemory/12582880}	150	dynamic
time_zone		Asia/Seoul	dynamic
<div>할당된 리소스 (0)</div> <div>할당된 모든 데이터베이스에 변경 사항이 적용됩니다.</div> <div> <input type="text"/> <div>할당된 리소스 필터링</div> </div> <div> <div>< 1 ></div> <div>⚙</div> </div>			
이름	버전	리전	크기
할당된 리소스(를) 찾을 수 없음			
취소		이전	변경 사항 적용

▼ 추가 구성

데이터베이스 옵션, 장애 조치, 백업 활성화됨, 역추적 비활성화됨, 성능 개선 도우미 비활성화됨, 향상된 모니터링 비활성화됨
CloudWatch Logs, 삭제 보호 비활성화됨

데이터베이스 옵션

DB 파라미터 그룹 [정보](#)

spring-deploy-test

옵션 그룹 [정보](#)

default:mariadb-10-4

DB에 파라미터 그룹 적용

수정 예약

수정 사항을 적용할 시간

☐ 예약된 다음 유지 관리 시간에 적용

현재 유지 관리 시간: August 30, 2021 05:20 - 05:50 UTC+9

☒ 즉시 적용

이 요청의 수정 사항과 보류 중인 수정 사항은 이 데이터베이스 인스턴스의 유지 관리 시간과 관계없이 가능하면 빨리 비동기식으로 적용됩니다.

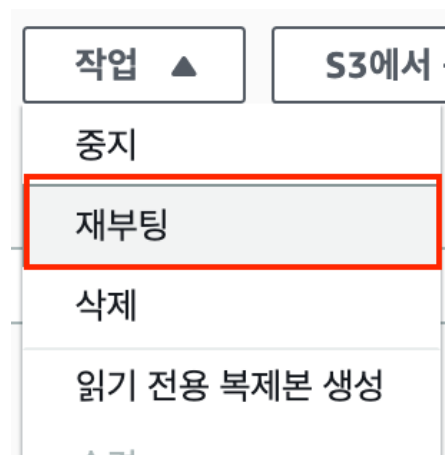
취소

뒤로

DB 인스턴스 수정

DB 인스턴스 수정

간혹 그룹이 제대로 반영되지 않는 경우도 있으므로 정상 적용을 위해 한 번 더 **재부팅**을 진행한다. 재부팅 까지 성공했다면 로컬에서 RDS에 한 번 접속해보자.

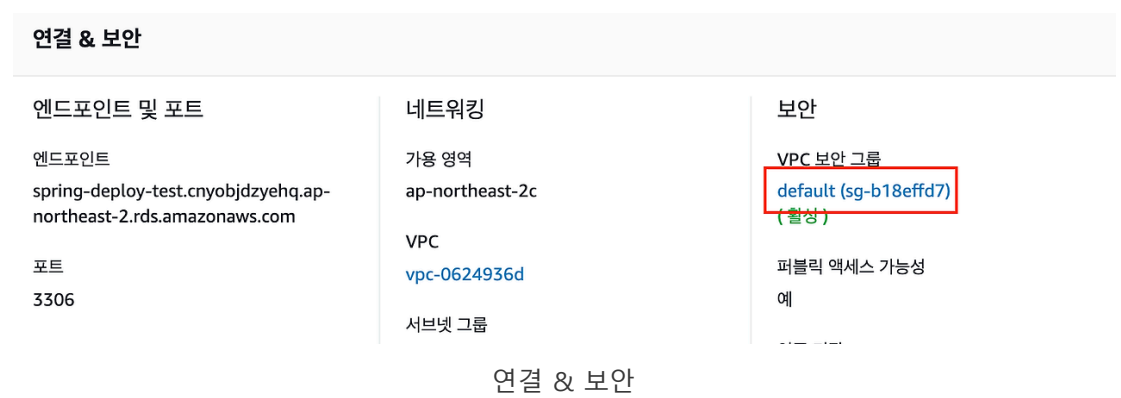


재부팅

3. 로컬에서 RDS 접속하기

1) RDS 보안그룹 클릭

로컬에 접속하기 위해서 RDS의 보안 그룹에 본인 PC의 IP를 추가해줘야 한다. RDS 세부정보 페이지에서 **[보안 그룹]** 항목을 클릭한다.



2) EC2 인스턴스 보안 그룹 ID 복사

RDS의 보안 그룹 정보를 그대로 두고, 브라우저를 새로 열어 본다. 그래서 브라우저 다른 창에서는 보안 그룹 목록 중 **EC2에 사용된 보안 그룹의 그룹 ID**를 복사한다.

<input checked="" type="checkbox"/>	Name ▾	보안 그룹 ID ▲	보안 그룹 이름 ▾
<input checked="" type="checkbox"/>	-	sg-0ffbee4702e7d33c7	spring-deploy-test-ec2

EC2의 보안 그룹 복사

3) 보안그룹 생성 > 복사된 보안 그룹 ID와 본인의 IP를 RDS 보안 그룹의 인바운드로 추가 > 생성

기존에 디폴트 보안 그룹에 적용해도 되고 만약 해당 보안 그룹이 다른 RDS에서 사용중이라면 새로 생성해도 된다. 나는 기존에 있던 보안 그룹은 다른 RDS가 사용중이기 때문에 새롭게 생성해주었다.

기본 세부 정보

보안 그룹 이름 정보

deploy-test-rds

생성 후에는 이름을 편집할 수 없습니다.

설명 정보

sprind-deploy-test-rds

VPC 정보

Q vpc-0624936d X

인바운드 규칙 정보

인바운드 규칙 1

삭제

유형 정보

MySQL/Aurora ▾

소스 유형 정보

사용자 지정 ▾

프로토콜 정보

TCP

소스 정보

Q

포트 범위 정보

3306

설명 - 선택 사항 정보

sg-0ffbee4702e7d33c7 X

인바운드 규칙 2

삭제

유형 정보

MySQL/Aurora ▾

소스 유형 정보

내 IP ▾

프로토콜 정보

TCP

소스 정보

Q

포트 범위 정보

3306

설명 - 선택 사항 정보

인바운드 규칙에 추가

새롭게 보안그룹을 생성한 경우에만 다시 RDS 인스턴스로 가서 보안 그룹을 수정해주고 즉시 적용 해준다.

연결

서브넷 그룹

default-vpc-0624936d

보안 그룹

이 DB 인스턴스와 연결할 DB 보안 그룹 목록입니다.

보안 그룹 선택

deploy-test-rds X

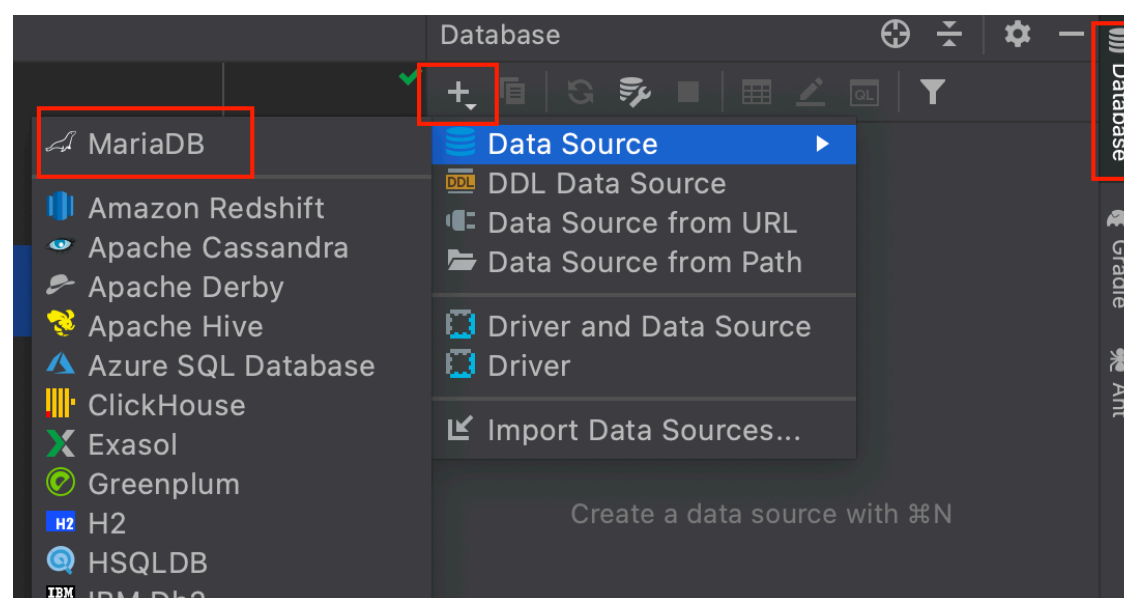
인증 기관

rds-ca-2019

보안 그룹 선택

로컬에서 원격 DB로 붙을 때 GUI 클라이언트를 많이 사용한다. MySQL의 경우 대표적으로 Workbench, Sequel Pro(Mac 전용) 등이 있다. 각각의 도구마다 큰 차이가 없으니 본인에게 맞는 툴을 사용하면 된다. 이번에는 IntelliJ에 기본적으로 내장되어있는 Database를 사용할 것이다. (해당 기능은 인텔리제이 유료 버전만 지원된다.)

4) 인텔리제이 오른쪽 탭 Database 클릭 > + 버튼 클릭 > MariaDB 선택



IntelliJ > MariaDB

5) RDS 엔드포인트 복사

연결 & 보안

엔드포인트 및 포트

엔드포인트

spring-deploy-test.cnyobjdzyehq.ap-northeast-2.rds.amazonaws.com

포트

3306

네트워킹

가용 영역

ap-northeast-2c

VPC

vpc-0624936d

서브넷 그룹

default-vpc-0624936d

보안

VPC 보안 그룹

deploy-test-rds (sg-0244f3a2afef8525a)

(활성)

퍼블릭 액세스 가능성

예

이츠 기과

RDS 엔드포인트

6) Databse 설정 > Name 입력 (spring-deploy-test) > Host에 RDS 엔드포인트 입력 > User/ Password 항목에 마스터 정보 입력 > Database 항목에 추가 구성에서 입력한 초기 데이터베이스 이름 입력

모든 입력이 끝난 후 Test Connection을 눌러서 통과가 되면 생성해준다.

Name: spring-deploy-test

Reset

Comment:

General

Options

SSH/SSL

Schemas

Advanced

Connection type: default

Driver: MariaDB

Host: /ehq.ap-northeast-2.rds.amazonaws.com

Port: 3306

User: loosie

Password: <hidden>

Save: Forever

Database: spring_deploy_test

URL: ibjdzyehq.ap-northeast-2.rds.amazonaws.com:3306/spring_deploy_test

Overrides settings above

Test Connection

✓

DBMS: MariaDB (ver. 10.4.13-MariaDB-log)

Case sensitivity: plain=exact, delimited=exact

Driver: MariaDB Connector/J (ver. 2.6.0, JDBC4.2)

Ping: 56 ms

SSL: no

Cancel

Apply

OK

DB 설정 후 생성

7) RDS 설정 정보 확인하기

RDS에서 설정한 정보들이 잘 입력됐는지 확인해보자.

```
show variables like 'c%';
```

만약 적용이 안된 부분이 있다면 직접 변경해주면 된다.

```
alter database spring_deploy_test
character set = 'utf8mb4'
collate = 'utf8mb4_general_ci';
```

	Variable_name	Value
1	character_set_client	utf8mb4
2	character_set_connection	utf8mb4
3	character_set_database	utf8mb4
4	character_set_filesystem	utf8mb4
5	character_set_results	utf8mb4
6	character_set_server	utf8mb4
7	character_set_system	utf8
8	character_sets_dir	/rdsdbbin/mariadb-10.4.13.R8/shareCharsets/
9	check_constraint_checks	ON
10	collation_connection	utf8mb4_general_ci
11	collation_database	utf8mb4_general_ci
12	collation_server	utf8mb4_general_ci

```
show variables like 'c%';
```

성공적으로 모든 항목이 적용되었으면 타임존도 확인해보자.

```
select @@time_zone, now();
```

	@@time_zone	now()
1	Asia/Seoul	2021-08-24 01:33:01

```
select @@time_zone, now();
```

마지막으로 한글명이 잘들어가는지 간단한 테이블 생성과 insert 쿼리를 실행해보자.

```
create table test(
  id bigint(20) not null auto_increment,
  content varchar(255) default null,
  primary key (id)
) engine =InnoDB;

insert into test(content) values ('테스트');
```

```
select * from test;
```

다음과 같이 한글 데이터도 잘 등록되는 것이 확인된다. 이제 모든 설정이 끝났으므로 EC2와 RDS가 잘 연동되는지 확인해보면 된다.

	id	content
1	1	테스트

select 쿼리 실행 결과

4. EC2에서 RDS 접속하기

[#0-1 AWS EC2 인스턴스 생성 후 접속하기](#)에서 진행한대로 EC2에 ssh로 접속한다. 접속되었다면 MySQL 접근 테스트를 위해 **MySQL CLI**를 설치한다.

* 실제 EC2의 MySQL을 설치해서 쓰는게 아닌, 명령어 라인만 쓰기위한 설치이다.

- **sudo yum install mysql**

설치가 다 되었으면 로컬에서 접근하듯이 계정, 비밀번호, 호스트 주소를 사용해 RDS에 접속한다.

- **mysql -u 계정 -p -h Host주소**
- 나의 경우 다음과 같이 입력하였다.
 - **mysql -u loosie -p -h spring-deploy-test.cnyobjdzyehq.ap-northeast-2.rds.amazonaws.com**

패스워드를 입력하면 다음과 같이 EC2에서 RDS에 접속되는 것을 확인할 수 있다.

```
[ec2-user@spring-deploy-test ~]$ mysql -u loosie -p -hspring-deploy-test.cnyobjdzyehq.ap-northeast-2.rds.amazonaws.com
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 22
Server version: 10.4.13-MariaDB-log 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.
```

ec2에서 rds 접속

RDS에 접속되었으면 실제로 생성한 RDS가 맞는지 간단한 쿼리를 한 번 실행해보자.

```
show databases;
use spring_deploy_test;
show tables;
select * from test;
```

IntelliJ DB 기능을 통해서 생성했던 테이블과 칼럼이 EC2에서도 잘 조회되는 것을 확인할 수 있다.

```
MariaDB [(none)]> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| spring_deploy_test |
+-----+
5 rows in set (0.01 sec)

MariaDB [(none)]> use spring_deploy_test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [spring_deploy_test]> show tables;
+-----+
| Tables_in_spring_deploy_test |
+-----+
| test                          |
+-----+
1 row in set (0.00 sec)

MariaDB [spring_deploy_test]> select * from test;
+----+-----+
| id | content |
+----+-----+
|  1 | 테스트  |
+----+-----+
1 row in set (0.00 sec)
```

쿼리 테스트