

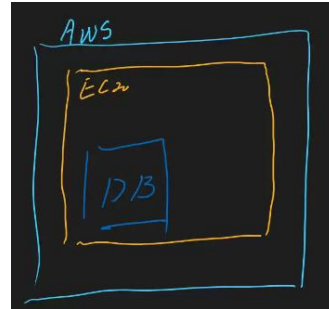
23-12-12

- WAS 인스턴스 재설치
- putty에서 마리아 DB 설치
- ubuntu에서 사용자 생성, 방화벽 설정
- heidiSQL과 Workbench에서 WAS IP로 연결

workbench와 heidiSQL 설치함

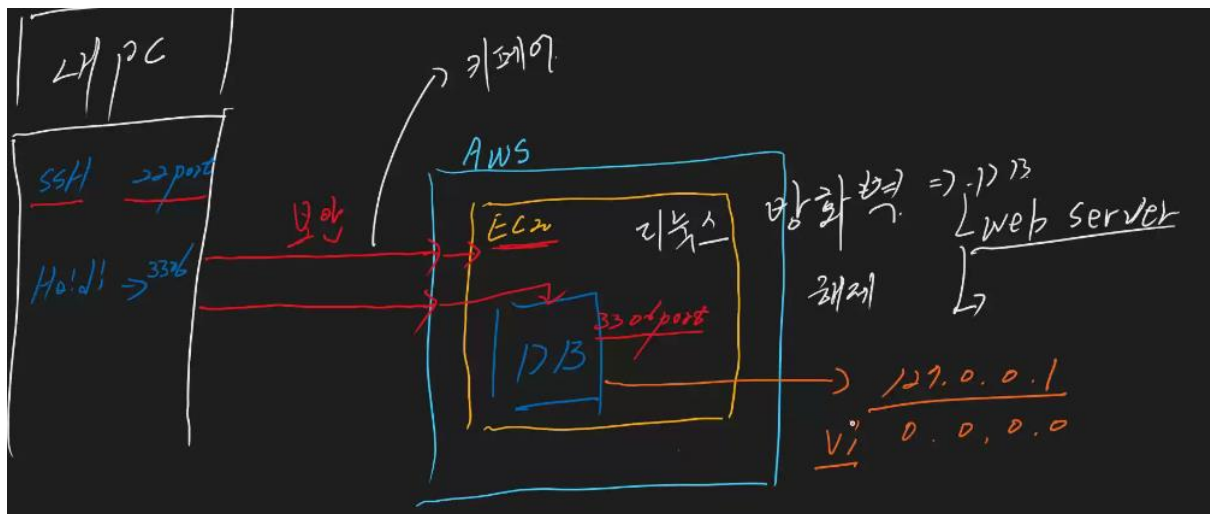
Workbench랑 heidiSQL이랑 동시에 사용하지 않기

AWS 안에 EC2 안에 MariaDB 설치했다



내 pc에서 AWS에 오려고 하니 보안규칙을 통과해야 한다

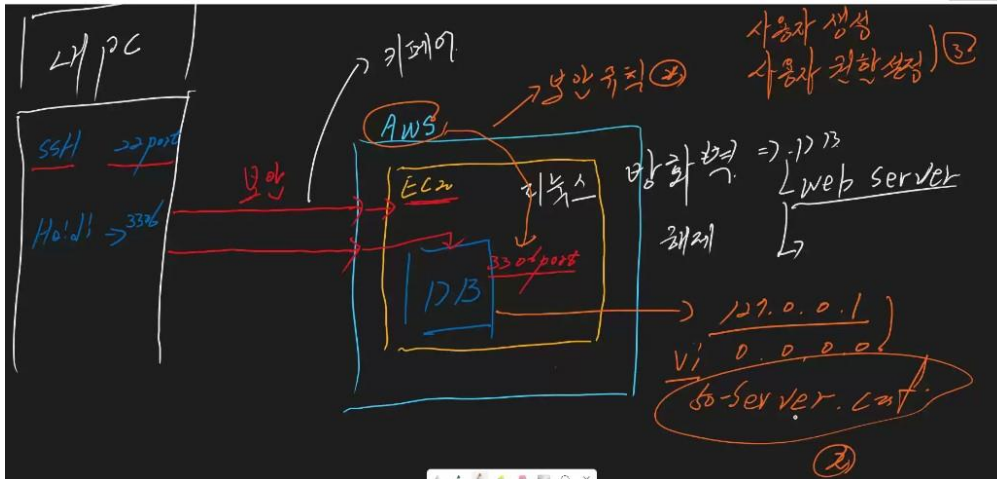
보안을 통과하면 EC2까지 도착해야 한다.(내 pc에서 ssh 22port 열어줘야하고 heidiSQL 3306)



암호화 기법은 키페어를 사용하여 보안규칙을 통과한다

리눅스에 방화벽 설정이 있다

-> 실행하려고 할 때 방화벽이 자동으로 활성화된다.



1. 보안규칙설정

2. Server cnf

3. 사용자 생성 / 사용자 권한 설정

A. 사용자 생성하는 명령어

Create User 'papawon'@'LocalHost' identified by '1234';
비밀번호도 같이 부여하겠다는 뜻이다!

B. 사용자 삭제하는 명령어

Drop User 'papawon'@'LocalHost';

C. 사용자 확인하는 명령어

User와 필드를 보이라고 명령어 써준 것
Select user, host, password from mysql.user;
show databases;

D. 사용자 권한 주는 명령어

Grant ALL PRIVILEQS on *(DB명) *(테이블명) 'papawon'@'%' with grant option;
ALL-> CRUD
(option 이렇게 주면 문제됨, 왜 문제되는지 모르겠음)
(grant all 주고싶지 않으면 grant select, update, insert, delete 이렇게 써도됨
그리고 all이 아니라 선택해서 줘도 됨)

비밀번호1234입력하고 온 애들은 모두다 상관없이 들어올 수 있는 것

```
MariaDB [(none)]> grant all on *.* to 'you'@'%' identified by '1234' with grant option;
Query OK, 0 rows affected (0.000 sec)
```

create user 'root'@'%' :

% 부분이 모든 것을 포함하는 것

특정 ip만 하고 싶으면 %자리에 ip입력하면 됨

%자리에 전체 ip를 다 대응하는 것 거기에 특정 아이피 집어넣으면 특정아이피만 접속가능

그 아이피 바뀌면 사용자는 읽지만 아이피 바뀌면 들어올 수 없다

그래서 DB는 인증 접속하는 것이 3개다 id,pw,ip

이 3개가 같아야지 같은 거임

그래서 %로 만들어준 것은 local 즉 root와 %root는 다른 것이다

```
Create User 'papawon'@'localhost' identified by '1234';
Drop User 'papawon'@'localhost';
Select User, Host from mysql.user;
localhost = 127.0.0.1

Grant ALL PRIVILEGES ON *.*
C U R D DB명 테이블명
'papawon'@'%' with grant option;
```

권한 설정하는 부분 했다

```
MariaDB [(none)]> select user, host, password from mysql.user;
+-----+-----+-----+
| user | host      | password |
+-----+-----+-----+
| root | localhost |          |
+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [(none)]> select user from mysql.user;
+-----+
| user |
+-----+
| root |
+-----+
1 row in set (0.000 sec)
```

select 해서 원하는 부분 검색하는 것

비밀번호 설정안하고 마리아디비부터해서

나는 password가 없어서

create user 'root'@'localhost' identified by '1234';를 해줘서 비밀번호 설정해줬다

```
MariaDB [(none)]> drop user 'root'@'%';
Query OK, 0 rows affected (0.000 sec) 삭제함
```

생성한 사용자 root 그리고 host를 %로 설정해 준 것을 삭제하는 명령어

질문사항 : mysql -u root -p를 먼저 설정하지 않고 create로 비밀번호 설정해주는 것 먼저해도 되는 것인가? 가능하다 mysql -u root -p와 비밀번호 설정해주는 것은 다른 것.

- 방화벽 설정

Ufw => ip port (port)

SUDO Ufw show raw => 방화벽 기본설정

SUDO Ufw default deny => 차단

SUDO Ufw default allow => 허용

SUDO Ufw allow 22 => Tcp/UDP 22접속을 다 허용한다.

SUDO Ufw allow 22/Tcp => Tcp 22접속을 허용한다.

Ufw => ip port 관련있는것??

Sudo ufw show raw => 방화벽 기본설정을 보여줌

Sudo ufw default deny => 방화벽 차단

Sudo ufw default allow => 방화벽 모두 허용

Sudo ufw allow 22 => Tcp/UDP 22접속을 다 허용한다

Sudo ufw allow 22/TCP => TCP 22접속을 허용한다

AWS는 인바운드 규칙이 먼저이다

리눅스에서 열려있지 않으면 접속할 수 없다

리눅스 열어줬다 해도 인바운드 규칙 설정하지 않으면 열어줄 수 없다

인바운드 규칙이 적용하고 -> 이 시스템에서 방화벽 허용해줘야 열린다(putty안에서 설정)

```
ubuntu@ip-172-31-32-134:~$ sudo ufw show raw;
IPv4 (raw):
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts      bytes target     prot opt in     out     source
ination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts      bytes target     prot opt in     out     source
ination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts      bytes target     prot opt out     source
```

방화벽 기본설정 보여줌

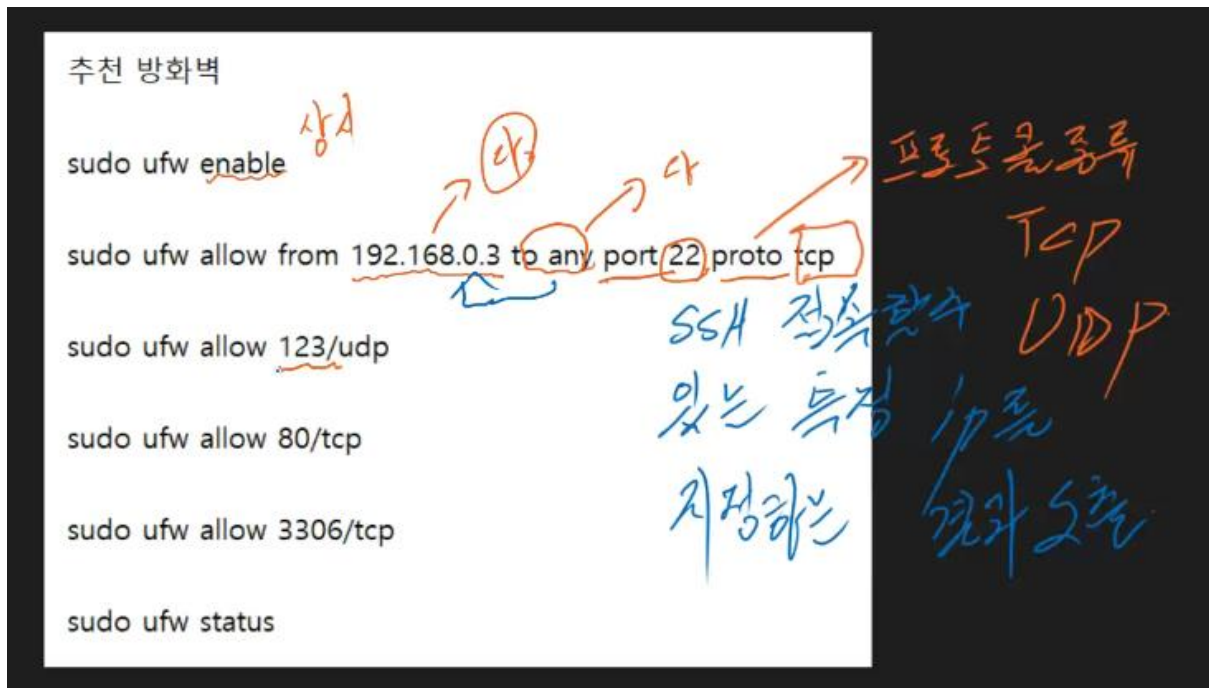
```
ubuntu@ip-172-31-32-134:~$ sudo ufw allow 22
Rules updated
Rules updated (v6)
```

22번 포트 방화벽 허용한다는 것

```
ubuntu@ip-172-31-32-134:~$ sudo ufw deny 22
```

22번 포트 방화벽 차단한다는 것(열었던 것을 닫았다)

방화벽차단



<https://webdir.tistory.com/206>

Sudo ufw enable -> 상시 열어주라는 것

192.168.0.3 에 대해서는 다 열어주라는 것 / to any (이것도 다)

프로토콜 종류 -> TCP, UDP

SSH 접속할 수 있는 특정 ip를 지정하는 결과 도출

최종적으로는 ubuntu의 권한이 먼저 적용이 됨

aws에서 모든 규칙 설정하고 ubuntu는 웬만하면 건들이지 않는 것이 안정적인 방법

aws에서 인바운드 규칙에서 적용해라(선생님 의견)

관리자 스타일마다 다름

- Password 변경방법

1.

```
mysqladmin -u root password '1234'
```

mysqladmin이라는 최고 경력자 명령어가 있다

Mysqladmin -u root password '1234' => root의 비밀번호가 바뀐다

(있다는 것만 알면됨)

2.

```
SET password for root=password('1234')
```

SET password for root=password('1234')

3.

```
update user set password=password('1234') where user='root'
```

위와 같이 사용할 수 있지만 지금 이 방법인 update, set, where로 작업할거다

Update user set password=password('1234') where user='root'

```
MariaDB [(none)]> update mysql.user set password=password('asdf1234') where user='root';
Query OK, 1 row affected (0.000 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

user는 mysql의 user이다

~~루트 비밀번호는 보안장치에만 보관한다(usb비트로커처리)~~

- 시간대 설정

```
MariaDB [(none)]> select @@global.time_zone, @@session.time_zone;
+-----+-----+
| @@global.time_zone | @@session.time_zone |
+-----+-----+
| SYSTEM             | SYSTEM               |
+-----+-----+
1 row in set (0.000 sec)
```

mariadb안에서

set global time_zone='Asia/Seoul'; set time_zone='Asia/Seoul';

<https://jwkim96.tistory.com/23> : mysql server timezone을 한국으로 설정하기(이렇게 할수도 있구나 참고만)

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

MariaDB [(none)]> select * from mysql.time_zone_name where Name = 'Asia/Seoul';
+-----+-----+
| Name      | Time_zone_id |
+-----+-----+
| Asia/Seoul | 311          |
+-----+-----+
1 row in set (0.000 sec)
```

(이거는 미리 설치한 선생님 화면이다)

```
+-----+-----+
| @@global.time_zone | @@session.time_zone |
+-----+-----+
| SYSTEM             | SYSTEM               |
+-----+-----+
1 row in set (0.000 sec)

MariaDB [(none)]> SET GLOBAL time_zone='Asia/Seoul'; SET time_zone = 'Asia/Seoul';
Query OK, 0 rows affected (0.001 sec)

Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> SELECT @@global.time_zone, @@session.time_zone
-> ;
+-----+-----+
| @@global.time_zone | @@session.time_zone |
+-----+-----+
| Asia/Seoul         | Asia/Seoul          |
+-----+-----+
1 row in set (0.000 sec)
```



```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p mysql
```

mysql_tzinfo_to_sql/ usr/share/zoneinfo | mysql -u root -p mysql

(우분투에서 써야 함 그런데 쓰는 것만으로 해결되지 않음)

Tz (time zone) info(information) usr디렉토리 안에 share안에 있다는 것

mysql -u root -p 접속하는 방법

```
MariaDB [(none)]> exit; 1
Bye
```

```
ubuntu@ip-172-31-37-141:~$ sudo systemctl stop mariadb
```

마리아 디비 잠깐 stop하는 것

```
ubuntu@ip-172-31-37-141:~$ sudo systemctl stop mariadb
ubuntu@ip-172-31-37-141:~$ sudo systemctl status mariadb
```

```
• mariadb.service - MariaDB 10.3.38 database server
```

```
Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
```

```
Active: inactive (dead) since Tue 2023-12-12 05:44:21 UTC; 14s ago
```

```
Docs: man:mysqld(8)
```

마리아 db dead라고 나올

```
https://mariadb.com/kb/en/library/systemd/
```

```
Process: 12114 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld 그냥 systemctl 사용한다
```

```
Process: 12115 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION
```

그러니까 Asia로 바뀌었는데 exit로 해도 다시 마리아디비 들어오면 Asia로 되어 있다

그런데 system을 stop하고 다시 start해주면

시간검색하면 system으로 변경되어 있다

서버시간으로 맞추는 것이 중요하다

아래 잘린 부분 -> 괜히바꿔주지 않아도 됨

⇒ 결론은 systemctl 사용한다는 것

- 마리아 디비 안에서 사용자 만들 것

Create user 'you'@'%' : you라는 user와 host를 전체로 준 것

```
MariaDB [(none)]> create user 'papawon02'@'%'
```

여기까지만 하면 user와 host만 생기는 것

```
MariaDB [(none)]> create user 'papawon02'@'%' identified by '1234';
Query OK, 0 rows affected (0.000 sec)
```

비밀번호 생성함 1

```
MariaDB [(none)]> create user 'you'@'%' identified by '1234';
Query OK, 0 rows affected (0.000 sec)
```

```
MariaDB [(none)]> select user, host, password from mysql.user;
```

| user | host | password |
|------|-----------|---|
| root | localhost | *A4B6157319038724E3560894F7F932C8886EBFCF |
| you | % | *A4B6157319038724E3560894F7F932C8886EBFCF |

%대신에 168.219...이런 ip를 쓸 수 있는데 그거는 그 아이피만 받아들이는 것

User만 쓰면 drop이 가능하다 하나만 있더라도

```
MariaDB [(none)]> select user, host, password from mysql.user;
+-----+-----+-----+
| user      | host      | password |
+-----+-----+-----+
| root      | localhost | *A4B6157319038724E3560894F7F932C8886EBFCF |
| papawon02 | %         | *A4B6157319038724E3560894F7F932C8886EBFCF |
+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [(none)]> drop user 'papawon02'@'%';
Query OK, 0 rows affected (0.000 sec)
```

가입했던 회원이 탈퇴하더라고 해서 삭제하는 것 아니고

삭제한 회원에 보관해야 한다

- 권한 설정

```
MariaDB [(none)]> grant all on *.* to 'you'@'%' identified by '1234' with grant option;
Query OK, 0 rows affected (0.000 sec)
```

모든 권한 설정해줬다

- 방화벽 설정

Sudo ufw allow 3306/tcp : 포트번호 3306 의 방화벽을 열어줌

```
MariaDB [(none)]> show global variables like 'port';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| port          | 3306  |
+-----+-----+
1 row in set (0.001 sec)
```

port 열려있는 것이 몇번인지 확인해줌 => exit;

sudo systemctl restart mariadb => 재설정

⇒ 방화벽 잡았다

- 편집기 수정 저장

```
ubuntu@ip-172-31-37-141:~$ sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf
```

sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf

vi 편집기로 들어감


```

user = mysql
pid-file = /run/mysqld/mysqld.pid
socket = /run/mysqld/mysqld.sock
port = 3306
basedir = /usr
datadir = /var/lib/mysql
tmpdir = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address = 127.0.0.1

# * Fine Tuning
key_buffer_size = 16M
max_allowed_packet = 16M
thread_stack = 192K
thread_cache_size = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
28,29 13%

```

:q! 저장하지 않고 나오는 것

A 또는 I 누르면 insert => esc => :wq : 저장하고 나오기

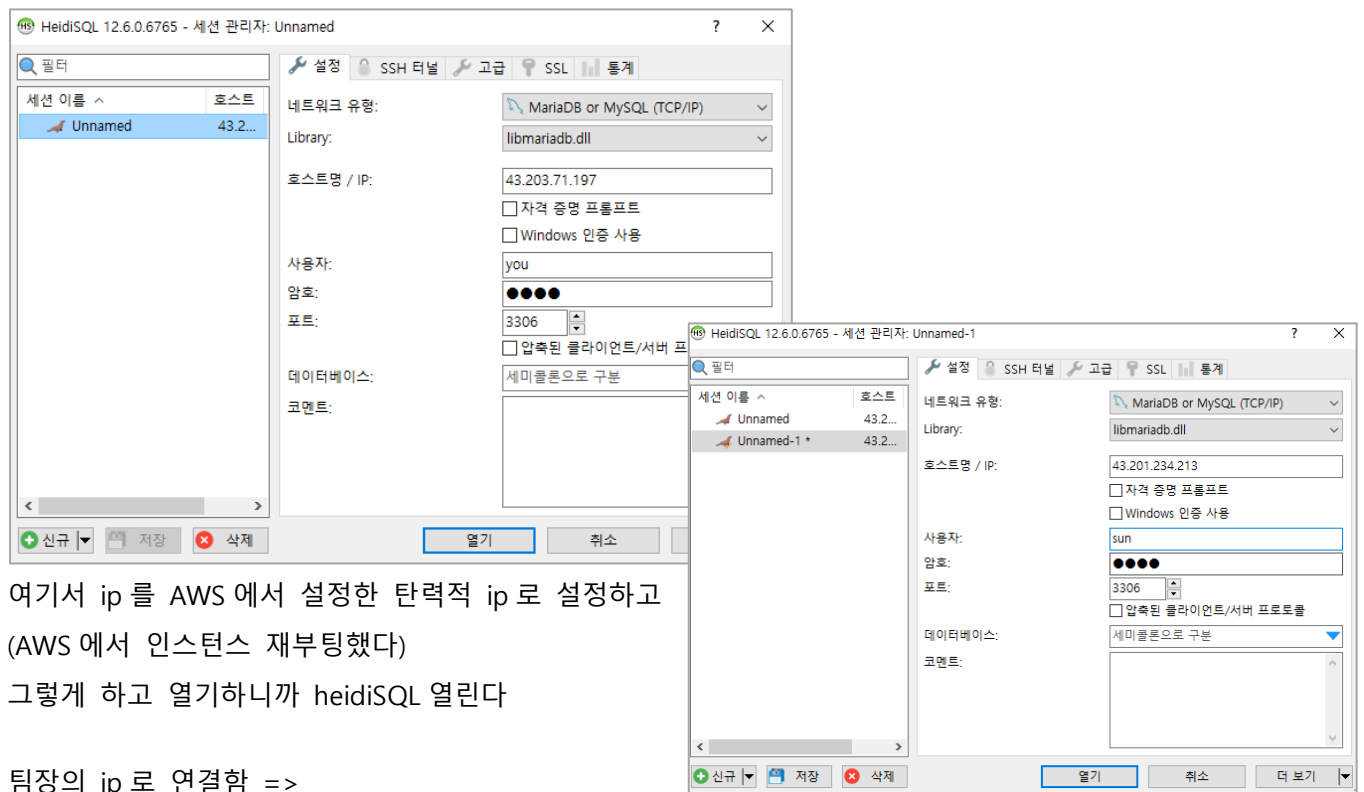
```

bind-address = 192.168.63.211, 168.211.144.53
# 이렇게 하면 지정한 ip만 허용한다는 것

```

● HeidiSQL 연결

그 다음에 heidiSQL 을 열고

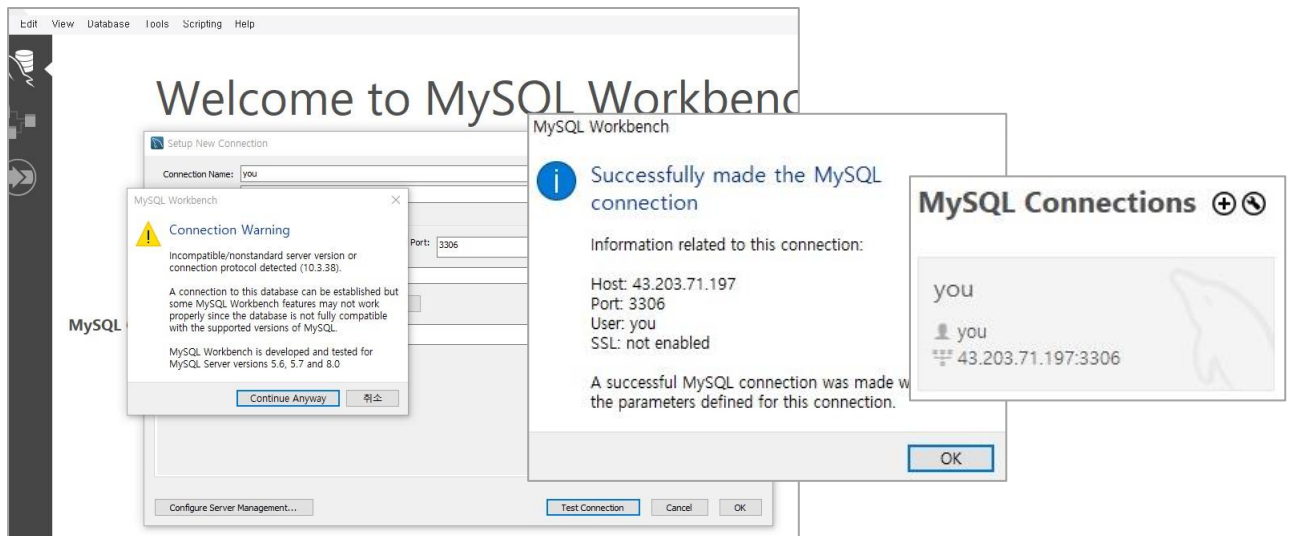


여기서 ip 를 AWS 에서 설정한 탄력적 ip 로 설정하고
(AWS 에서 인스턴스 재부팅했다)
그렇게 하고 열기하니까 heidiSQL 열린다

팀장의 ip 로 연결함 =>

팀장이 권한 설정한 아이디 모든 값을 입력하면 들어갈 수 있는 것이다.

- Workbench 연결



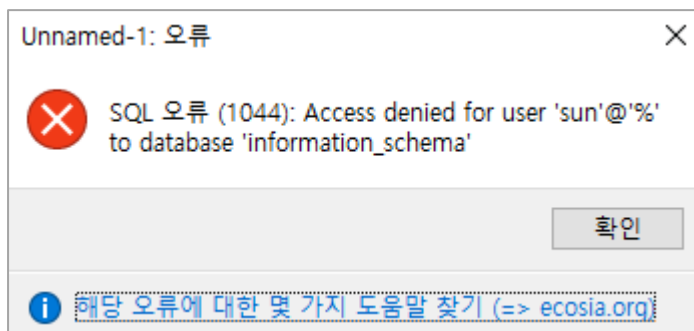
이름과 아이피, 포트번호 등 입력할 사항 입력하면 위와 같이 나오고(테스트 먼저함), ok 누르면 연결이 되는 것을 확인할 수 있다.

해보기!

권한을 팀장이 주면 접속하기

HeidiSQL 에서 팀장 ip 로 접속하기

=> 접속 가능



HeidiSQL 에서 테이블 추가하거나 변경하려고 했더니
Select 로만 권한을 줘서 변경과 추가가 가능하지 않음

권한 준 것이 아이디어가 됨

<처음부터 다시 실행>

1. 인스턴스 설치
키페어, 보안규칙(인바운드 규칙 설정), 탄력적 IP 생성
2. Putty 연결
3. putty 에서 mariadb 설치
4. 사용자 생성, 사용자 권한 설정(나만 all 주고 다른 사람들은 select 만)
5. 방화벽 열어주기 3306
6. vi 편집기에 0.0.0.0 으로 변경 후 저장

sudo systemctl restart mariadb

🟢 보안 그룹(0a0ae8e7fbf5abb2fc012-12-you)이 생성되었습니다.

▶ 세부 정보

보안 그룹 (2) 정보

🔄 작업 ▼ 보안 그룹을 CSV로 내보내기 ▼ 보안 그룹 생성

🔍 Find resources by attribute or tag

| <input type="checkbox"/> | Name ▼ | Security group ID ▼ | 보안 그룹 이름 ▼ | VPC ID |
|--------------------------|--------|--------------------------------------|------------|--------------------------------------|
| <input type="checkbox"/> | - | sg-0ae8e7fbf5abb2fc0 | 12-12-you | ypc-05318589d19bf85b |
| <input type="checkbox"/> | - | sg-0ada6bac00eb7e3a7 | default | ypc-05318589d19bf85b |

보안그룹 생성

키 페어 (1) 정보

🔄 작업 ▼ 키 페어 생성

🔍 Key Pair를 속성 또는 태그로 찾기

| <input type="checkbox"/> | 이름 ▼ | 유형 ▼ | 생성 완료 ▼ | 지문 | ID ▼ |
|--------------------------|-----------|------|------------------------|--------------------------------|-----------------------|
| <input type="checkbox"/> | 12-12-you | rsa | 2023/12/12 17:48 GMT+9 | b9:2f:ab:99:c9:30:d6:e4:85:... | key-03a7856f56fd5f... |

키페어 생성 및 저장

인스턴스 (2) 정보

🔄 연결 인스턴스 상태 ▼ 작업 ▼ 인스턴스 시작 ▼

🔍 Instance를 속성 또는 (case-sensitive) 태그로 찾기

| <input type="checkbox"/> | Name ✎ ▼ | 인스턴스 ID | 인스턴스 상태 ▼ |
|--------------------------|-----------|-------------------------------------|-----------|
| <input type="checkbox"/> | 12-12-you | i-0d1f56c6a70da7b29 | 🔴 종료됨 🔍 |
| <input type="checkbox"/> | 12-12-you | i-032e6f96f4b48d013 | 🟢 실행 중 🔍 |

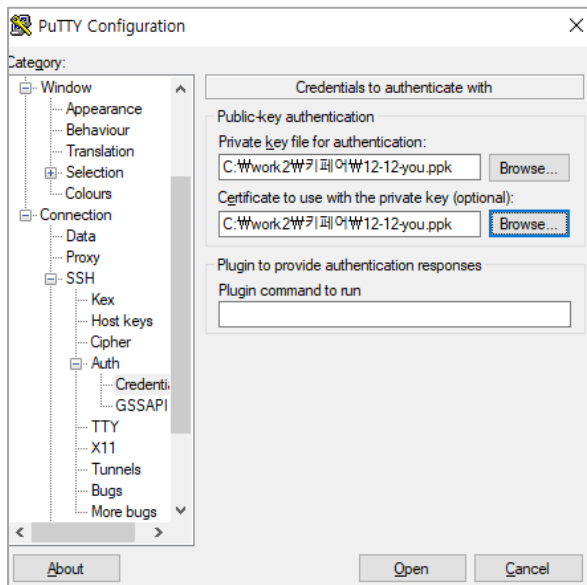
i-032e6f96f4b48d013 (12-12-you)

PublicIPs: 15.164.215.69 PrivateIPs: 172.31.37.226

인스턴스 생성 및 연결함

| 탄력적 IP 주소 (1/1) | | | | |
|--|---------------------|---------------|----------------------------|------------|
| <div> <div>탄력적 IP 주소 필터링</div> <div> <div><</div> <div>1</div> <div>></div> <div>⚙</div> </div> </div> | | | | |
| 역방향 DNS 레코드 | 연결된 인스턴스 ID | 프라이빗 IP 주소 | 연결 ID | 네트워크 인 |
| - | i-032e6f96f4b48d013 | 172.31.37.226 | eipassoc-04b3d9fb962a4e9b7 | 4814656939 |

탄력적 ip 연결

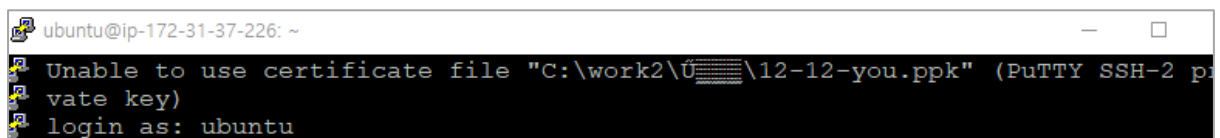


setting 을 탄력적 ip 로 설정해주고

글씨 크기는 appearance 에 change 부분에서 설정할 수 있음

그리고 Auth -> Credentials 와 GSSAPI 에 생성한 ppk 넣어줌

그리고 open 하기



로그인 ubuntu 하고

```
ubuntu@ip-172-31-37-226:~$ sudo apt-get update
```

```
ubuntu@ip-172-31-37-226:~$ sudo apt-get upgrade
```

```
ubuntu@ip-172-31-37-226:~$ sudo apt-get install mariadb-server
```

마리아 DB 설치

Sudo mysql_secure_installation

```

MariaDB [(none)]> update mysql.user set plugin='mysql_native_password' where user='root';
Query OK, 1 row affected (0.000 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit;
Bye
ubuntu@ip-172-31-37-226:~$

```

mysql -u root -p 로 마리아디비 접속할 수 있도록 변경했다

```

MariaDB [(none)]> create user 'you'@'%' identified by '1234';
Query OK, 0 rows affected (0.000 sec)

```

user 를 생성해줬고 host 는 모든 host 를 허용한다는 %를 주었고, 비밀번호는 1234 이다.

```

MariaDB [(none)]> select user, host, password from mysql.user;
+-----+-----+-----+
| user | host      | password |
+-----+-----+-----+
| root | localhost | *A4B6157319038724E3560894F7F932C8886EBFCF |
| you  | %         | *A4B6157319038724E3560894F7F932C8886EBFCF |
+-----+-----+-----+
2 rows in set (0.000 sec)

```

생성된 user, host, password 를 확인할 수 있는 명령어 select 를 사용해서 검색해줌

```

MariaDB [(none)]> grant all on *.* to 'you'@'%' identified by '1234' with grant option;
Query OK, 0 rows affected (0.000 sec)

```

grant 는 권한설정 명령어로, you 에게 all 모든 권한을 주고 비밀번호는 1234 로 주는 것

```

MariaDB [(none)]> grant select on *.* to 'sun'@'%' identified by '1234' with grant option;
Query OK, 0 rows affected (0.000 sec)      select 권한만 준 것

MariaDB [(none)]> grant select on *.* to 'ceb'@'%' identified by '1234' with grant option;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> grant select on *.* to 'eunjin'@'%' identified by '1234' with grant option;
Query OK, 0 rows affected (0.000 sec)

```

다른 팀원에게는 select 권한만 준 것

```

MariaDB [(none)]> show grants for 'sun'@'%';
+-----+
| Grants for sun@% |
+-----+
| GRANT SELECT ON *.* TO `sun`@`%` IDENTIFIED BY PASSWORD '*A4B6157319038724E3560894F7F932C8886EBFCF' WITH GRANT OPTION |
+-----+
1 row in set (0.000 sec)

```

sun 에 대한 권한 어떻게 설정되어 있는지 보여준다

select * from mysql.user where user='sun';

| 인스턴스 (2) 정보 | | | | | | | 인스턴스 시작 ▼ |
|---|-----------|---------------------|-----------|-----------|-------|-------|-----------|
| Instance를 속성 또는 (case-sensitive) 태그로 찾기 | | | | | | | < 1 > ⚙ |
| <input type="checkbox"/> | Name ↗ | 인스턴스 ID | 인스턴스 상태 ▼ | 인스턴스 유형 ▼ | 상태 검사 | 경보 상태 | |
| <input type="checkbox"/> | 12-12-you | i-0d1f56c6a70da7b29 | ⏹ 종료됨 🔍 | t2.micro | - | 경보 없음 | + |
| <input type="checkbox"/> | 12-12-you | i-032e6f96f4b48d013 | 🟢 실행 중 🔍 | t2.micro | 🕒 초기화 | 경보 없음 | + |

권한 부여 후 AWS 실행중인 인스턴스 재부팅한다

```
ubuntu@ip-172-31-37-226:~$ sudo ufw allow 3306/tcp
Rules updated
Rules updated (v6)
```

방화벽 3306 허용하는 것(AWS 에서 인바운드규칙 3306 허용도 해야함)

```
ubuntu@ip-172-31-37-226:~$ sudo ufw allow 3306/tcp
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-37-226:~$ sudo systemctl restart mariadb
ubuntu@ip-172-31-37-226:~$ sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf
ubuntu@ip-172-31-37-226:~$
```

mariadb 재시작하고

vi 편집기 열었다

```
ubuntu@ip-172-31-37-226: ~
[mysqld]
#
# * Basic Settings
#
user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
socket                  = /run/mysqld/mysqld.sock
port                    = 3306
basedir                 = /usr
datadir                 = /var/lib/mysql
tmpdir                  = /tmp
lc-messages-dir         = /usr/share/mysql
skip-external-locking
# Instead of skip-networking the default is now to listen only on
# localhost which is more secure and is not less secure.
bind-address             = 0.0.0.0
#
# * Fine Tuning
#
# 127.0.0.0 이었는데, 0.0.0.0으로 모두 들어
# 을 수 있도록 변경해줌
```

주소를 127.0.0.0 에서 0.0.0.0 으로 변경

이후 HeidiSQL 에서 탄력적 ip 로 접속이 가능하다

