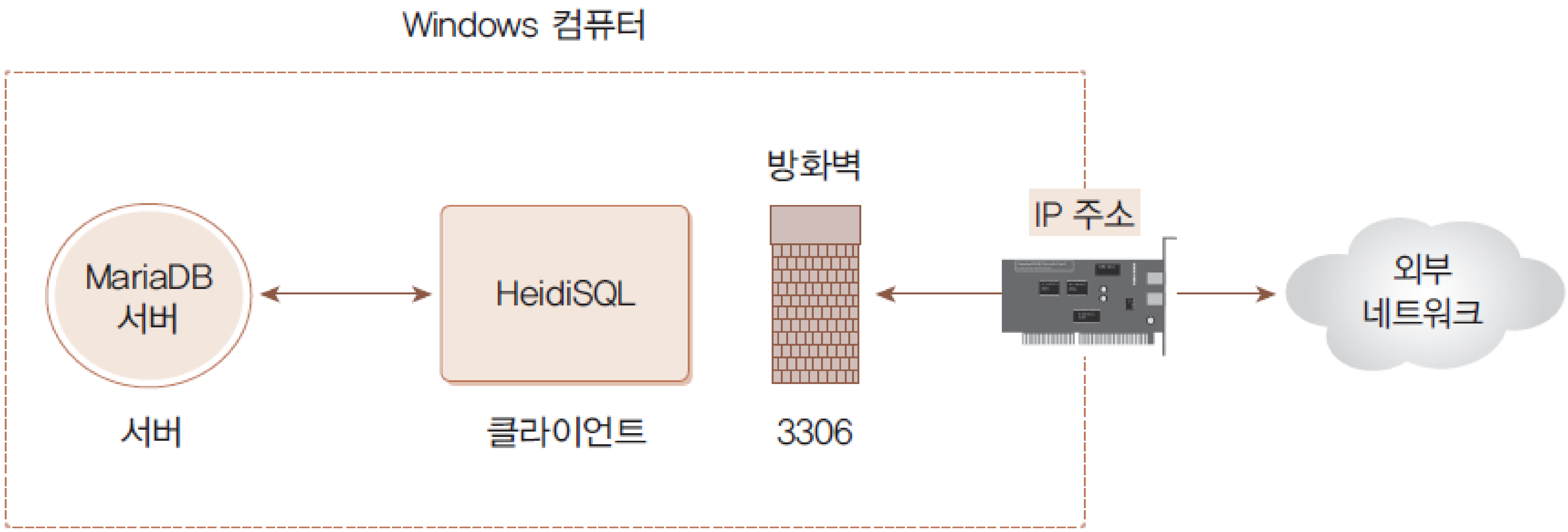
데이터베이스 서버를 활용한 빅데이터 저장

MLP LAB 임경태

DB server의 의미

- 우리가 현재까지 구축한 DB의 구조는 다음과 같다.

  - 한대의 컴퓨터에 DB를 설치하고 해당 컴퓨터에서 직접 데이터 가공

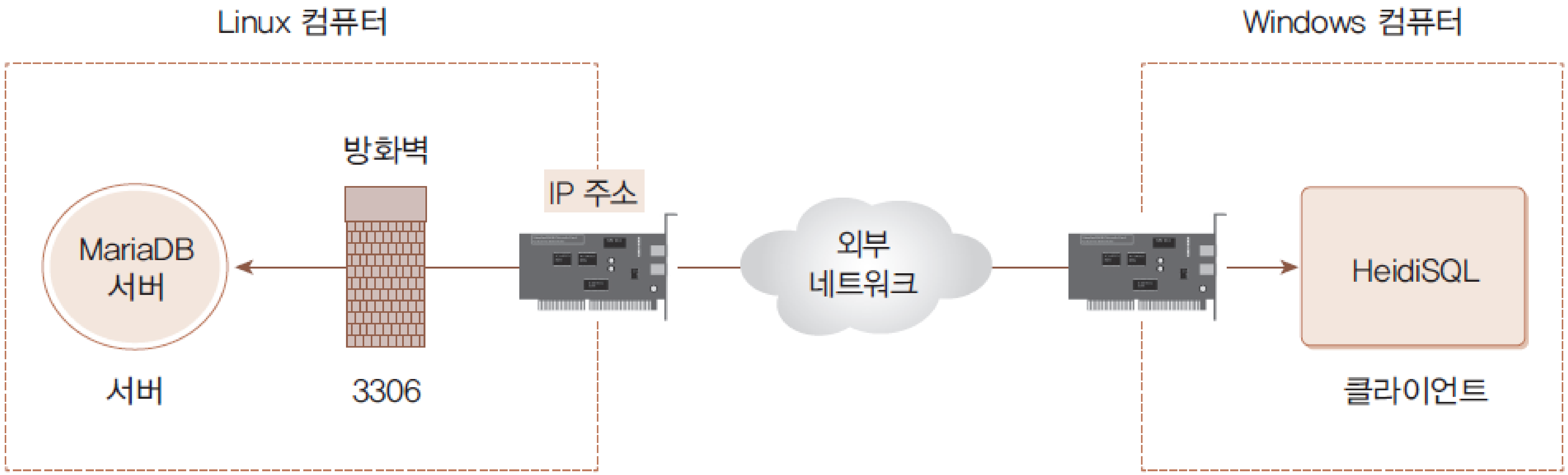


[그림 5-45] Windows 한 대에 서버와 클라이언트를 모두 설치한 상태

DB server의 의미

- 다음과 같이 DB서버를 외부 Linux 컴퓨터에 설치하고 사용하면 무슨 장점이 있나?

  - (1) 다수의 사용자가 함께 작업 가능, (2) DB를 위한 특수한 서버 구축 관리



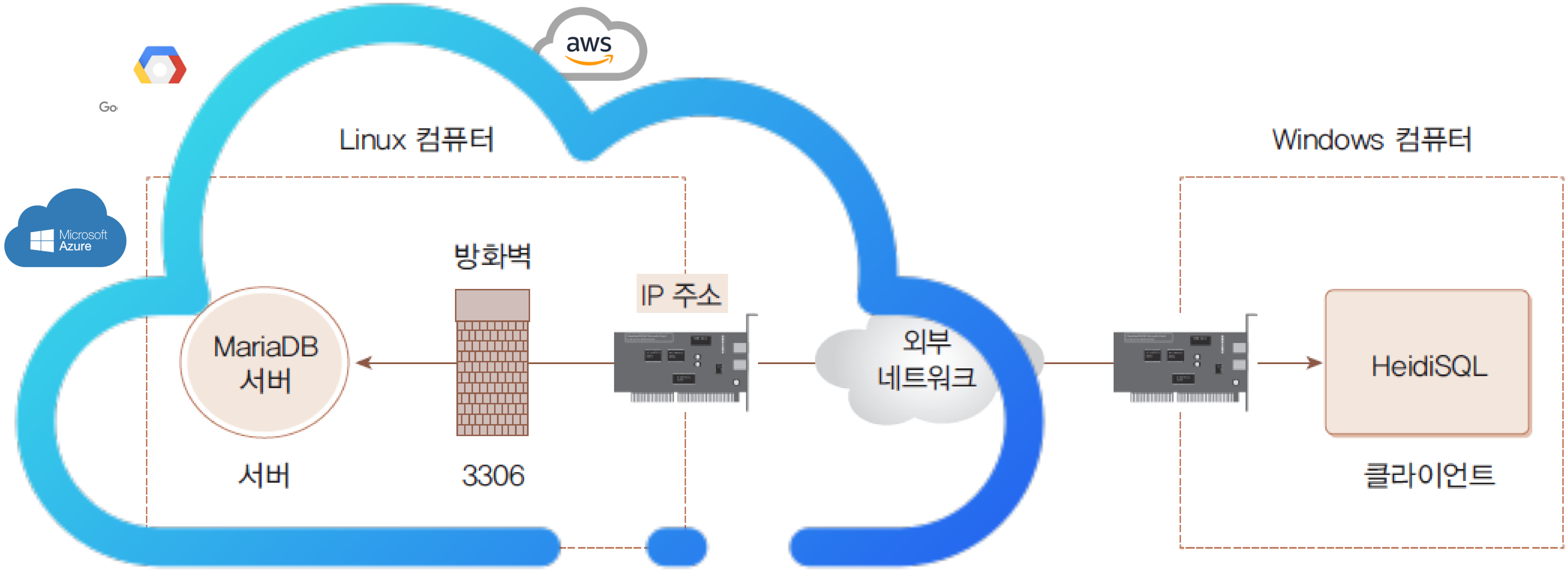
[그림 5-46] Linux 설치된 MariaDB Server에 Windows에 설치된 HeidiSQL에서 접속한 상태

01. GCP로 DB 서버 구축하기

현업에서 사용하는 DB server를 구축해 보자

- 실제 현업에서는 Cloud 서비스를 이용해 DB서버를 구축함

  - (1) 손쉽게 DB서버 구축 및 관리, (2) 필요한 만큼 자원 할당해서 사용



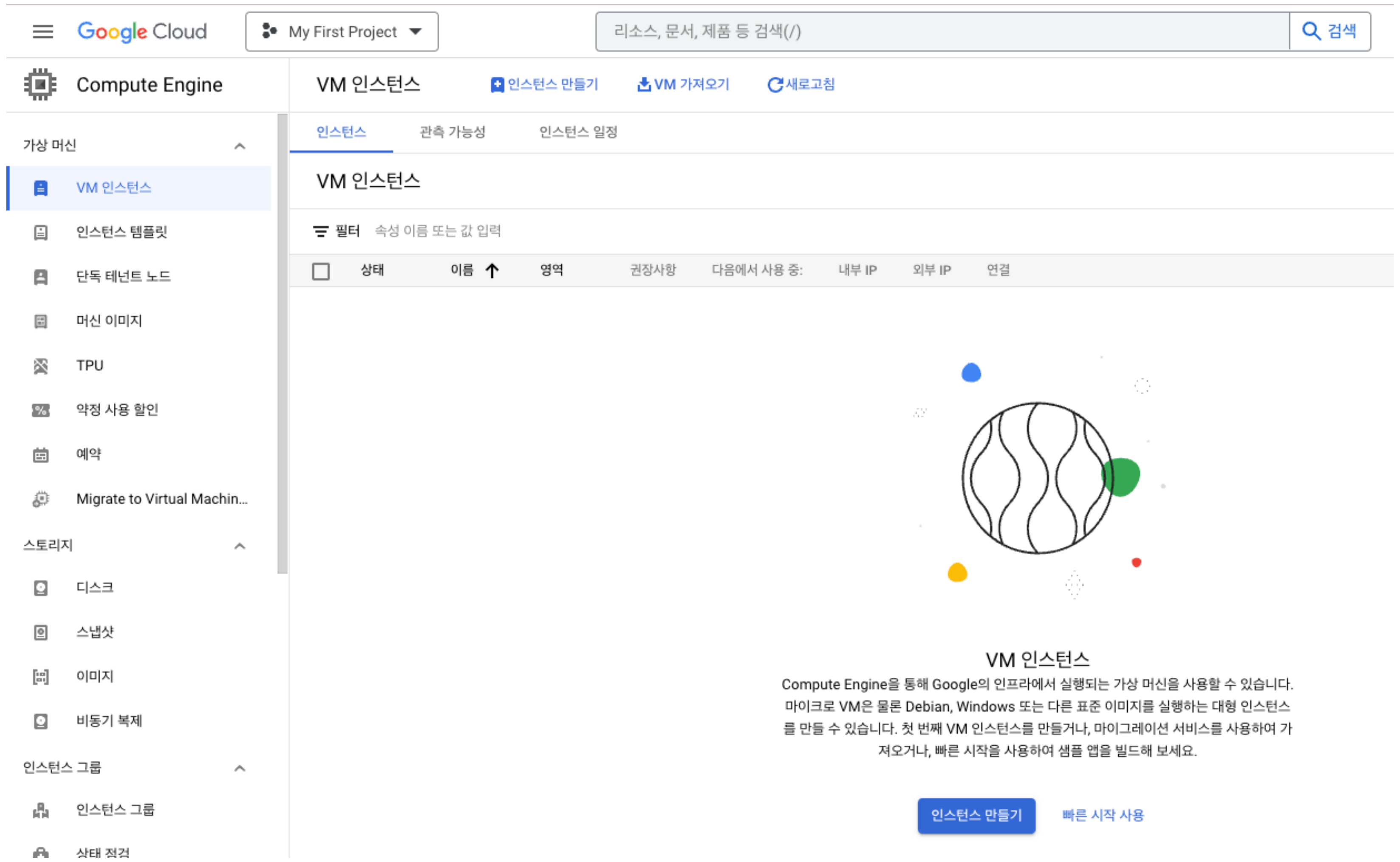
[그림 5-46] Linux 설치된 MariaDB Server에 Windows에 설치된 HeidiSQL에서 접속한 상태

그림출처: https://www.flaticon.com/kr/free-icons

GCP Server setup

- 우리가 구동하는 colab 파일은 어디 서버에서 작동되지? 바로 구글 원격서버!!

  - 그러면 DB도 구글 원격 서버로 만들 수 있겠네? 넵

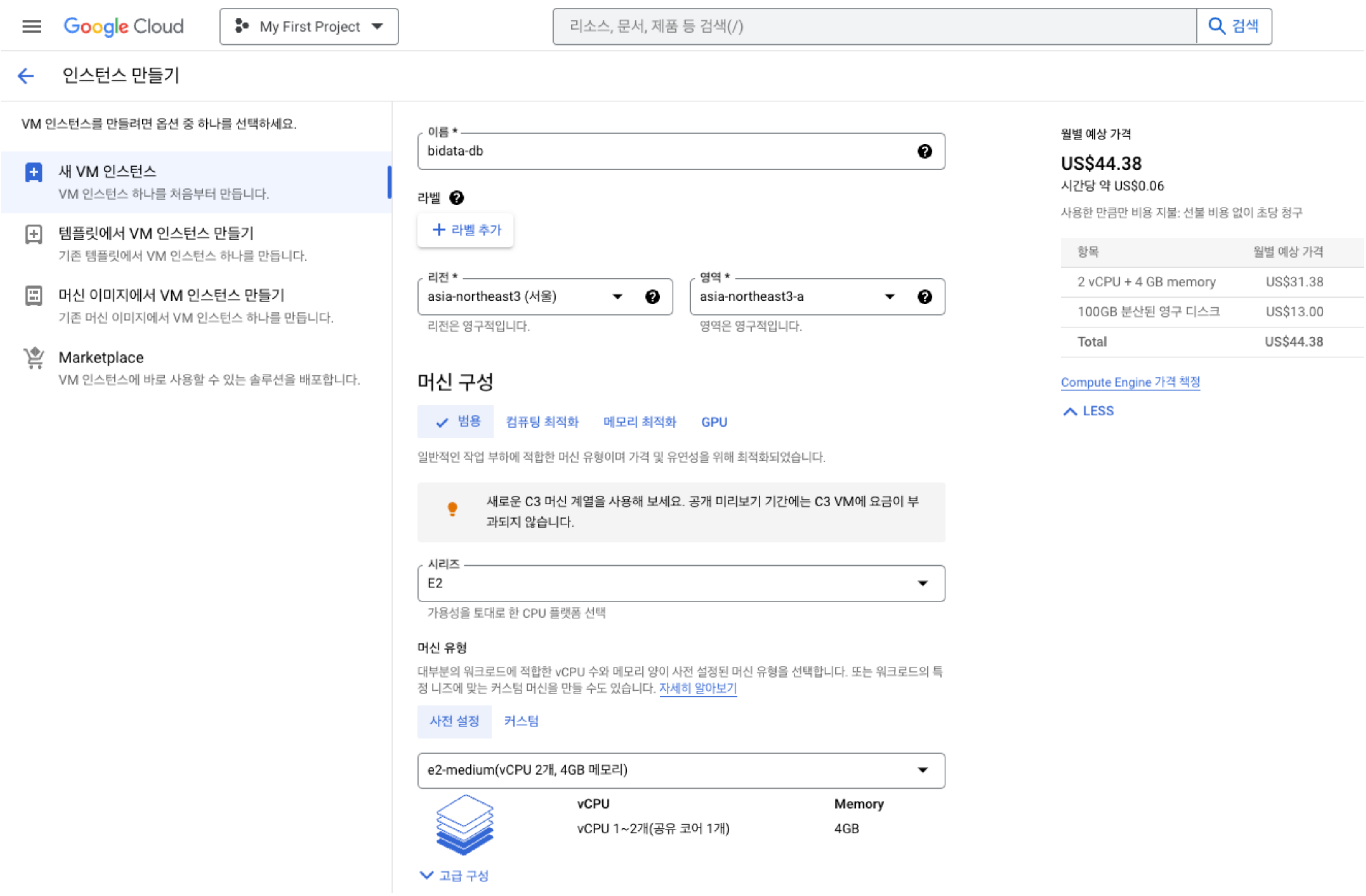


☐	상태	이름 ↑	영역	권장사항	다음에서 사용 중:	내부 IP	외부 IP	연결

VM 인스턴스

Compute Engine을 통해 Google의 인프라에서 실행되는 가상 머신을 사용할 수 있습니다. 마이크로 VM은 물론 Debian, Windows 또는 다른 표준 이미지를 실행하는 대형 인스턴스를 만들 수 있습니다. 첫 번째 VM 인스턴스를 만들거나, 마이그레이션 서비스를 사용하여 가져오거나, 빠른 시작을 사용하여 샘플 앱을 빌드해 보세요.

GCP Server setup (방화벽)



인스턴스 만들기

VM 인스턴스를 만들려면 옵션 중 하나를 선택하세요.

- 새 VM 인스턴스 — VM 인스턴스 하나를 처음부터 만듭니다.
- 템플릿에서 VM 인스턴스 만들기 — 기존 템플릿에서 VM 인스턴스 하나를 만듭니다.
- 머신 이미지에서 VM 인스턴스 만들기 — 기존 머신 이미지에서 VM 인스턴스 하나를 만듭니다.
- Marketplace — VM 인스턴스에 바로 사용할 수 있는 솔루션을 배포합니다.

이름 *: bidata-db

라벨 ❓ + 라벨 추가

리전 *: asia-northeast3 (서울) — 리전은 영구적입니다.

영역 *: asia-northeast3-a — 영역은 영구적입니다.

머신 구성

✓ 범용 | 컴퓨팅 최적화 | 메모리 최적화 | GPU

일반적인 작업 부하에 적합한 머신 유형이며 가격 및 유연성을 위해 최적화되었습니다.

새로운 C3 머신 계열을 사용해 보세요. 공개 미리보기 기간에는 C3 VM에 요금이 부과되지 않습니다.

시리즈: E2 — 가용성을 토대로 한 CPU 플랫폼 선택

머신 유형

대부분의 워크로드에 적합한 vCPU 수와 메모리 양이 사전 설정된 머신 유형을 선택합니다. 또는 워크로드의 특정 니즈에 맞는 커스텀 머신을 만들 수도 있습니다. 자세히 알아보기

사전 설정 | 커스텀

e2-medium(vCPU 2개, 4GB 메모리)

vCPU	Memory
vCPU 1~2개(공유 코어 1개)	4GB

고급 구성

월별 예상 가격

US$44.38

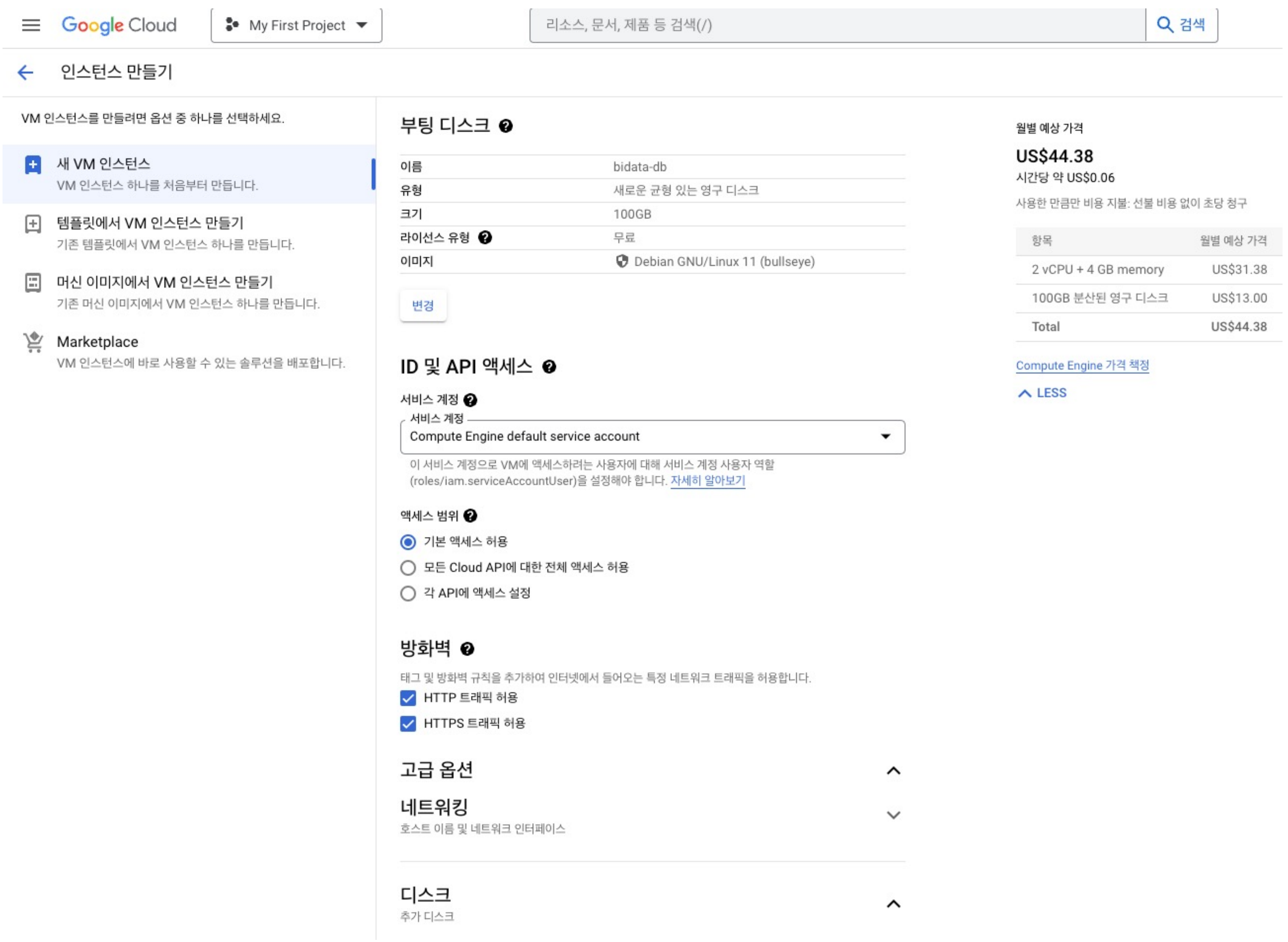
시간당 약 US$0.06

사용한 만큼만 비용 지불: 선불 비용 없이 초당 청구

항목	월별 예상 가격
2 vCPU + 4 GB memory	US$31.38
100GB 분산된 영구 디스크	US$13.00
Total	US$44.38

Compute Engine 가격 책정

LESS



인스턴스 만들기

VM 인스턴스를 만들려면 옵션 중 하나를 선택하세요.

- 새 VM 인스턴스 — VM 인스턴스 하나를 처음부터 만듭니다.
- 템플릿에서 VM 인스턴스 만들기 — 기존 템플릿에서 VM 인스턴스 하나를 만듭니다.
- 머신 이미지에서 VM 인스턴스 만들기 — 기존 머신 이미지에서 VM 인스턴스 하나를 만듭니다.
- Marketplace — VM 인스턴스에 바로 사용할 수 있는 솔루션을 배포합니다.

부팅 디스크

이름	bidata-db
유형	새로운 균형 있는 영구 디스크
크기	100GB
라이선스 유형	무료
이미지	Debian GNU/Linux 11 (bullseye)

변경

ID 및 API 액세스

서비스 계정: Compute Engine default service account

이 서비스 계정으로 VM에 액세스하려는 사용자에 대해 서비스 계정 사용자 역할 (roles/iam.serviceAccountUser)을 설정해야 합니다. 자세히 알아보기

액세스 범위

● 기본 액세스 허용

○ 모든 Cloud API에 대한 전체 액세스 허용

○ 각 API에 액세스 설정

방화벽

태그 및 방화벽 규칙을 추가하여 인터넷에서 들어오는 특정 네트워크 트래픽을 허용합니다.

☑ HTTP 트래픽 허용

☑ HTTPS 트래픽 허용

고급 옵션

네트워킹 — 호스트 이름 및 네트워크 인터페이스

디스크 — 추가 디스크

월별 예상 가격

US$44.38

시간당 약 US$0.06

사용한 만큼만 비용 지불: 선불 비용 없이 초당 청구

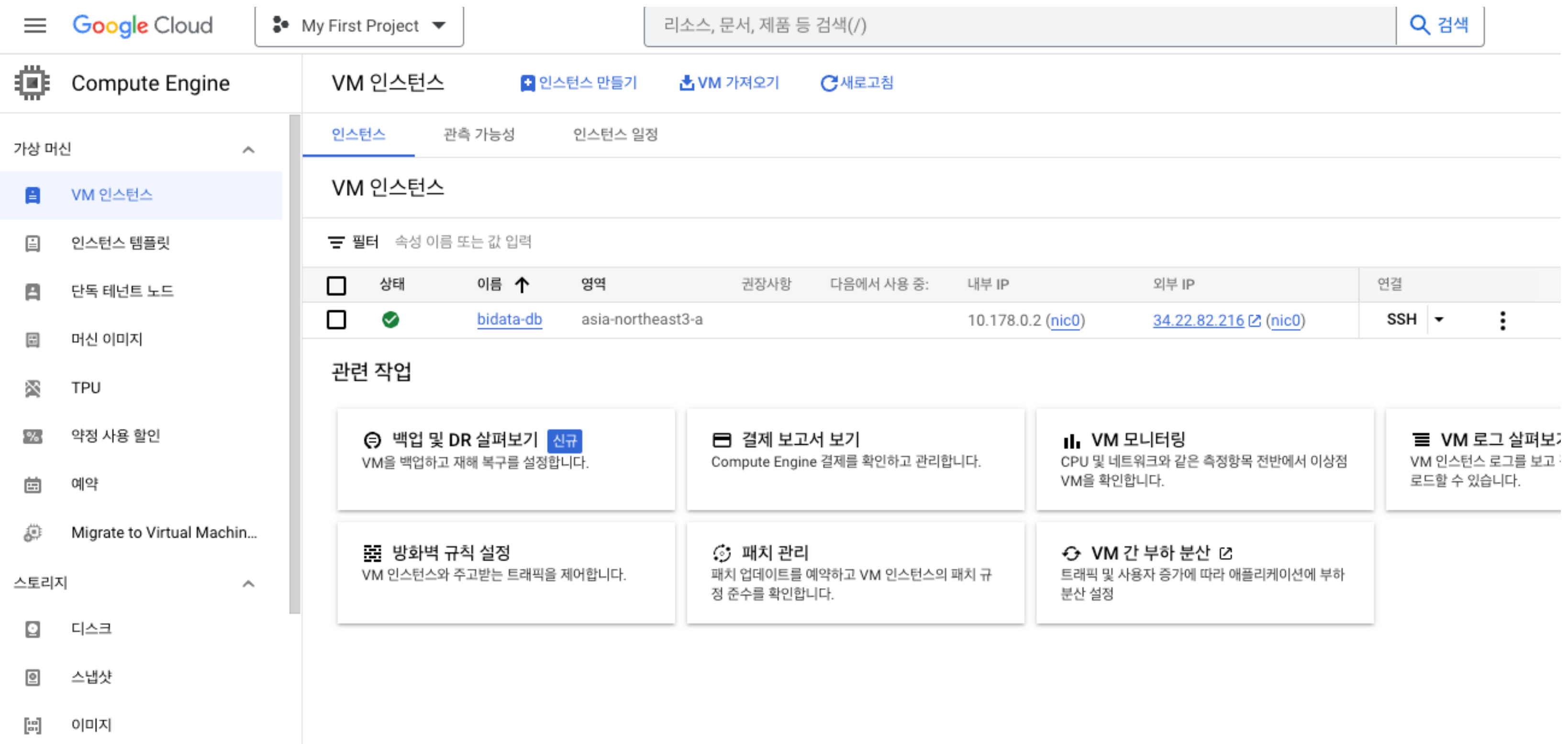
항목	월별 예상 가격
2 vCPU + 4 GB memory	US$31.38
100GB 분산된 영구 디스크	US$13.00
Total	US$44.38

Compute Engine 가격 책정

LESS

02. DB설정하기

DB setup



VM 인스턴스

□	상태	이름 ↑	영역	권장사항	다음에서 사용 중:	내부 IP	외부 IP	연결
□	✓	bidata-db	asia-northeast3-a			10.178.0.2 (nic0)	34.22.82.216 (nic0)	SSH

관련 작업

백업 및 DR 살펴보기 신규
VM을 백업하고 재해 복구를 설정합니다.

결제 보고서 보기
Compute Engine 결제를 확인하고 관리합니다.

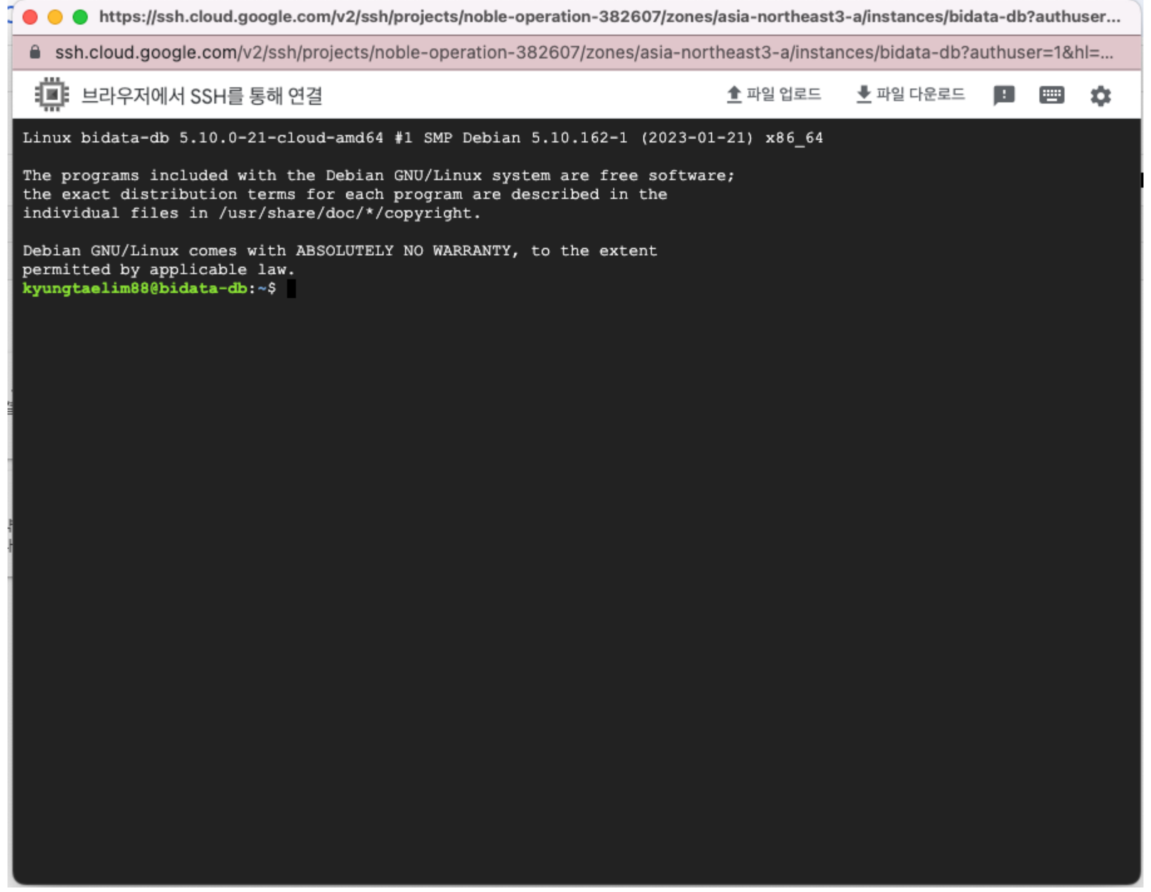
VM 모니터링
CPU 및 네트워크와 같은 측정항목 전반에서 이상점 VM을 확인합니다.

VM 로그 살펴보기
VM 인스턴스 로그를 보고 로드할 수 있습니다.

방화벽 규칙 설정
VM 인스턴스와 주고받는 트래픽을 제어합니다.

패치 관리
패치 업데이트를 예약하고 VM 인스턴스의 패치 규정 준수를 확인합니다.

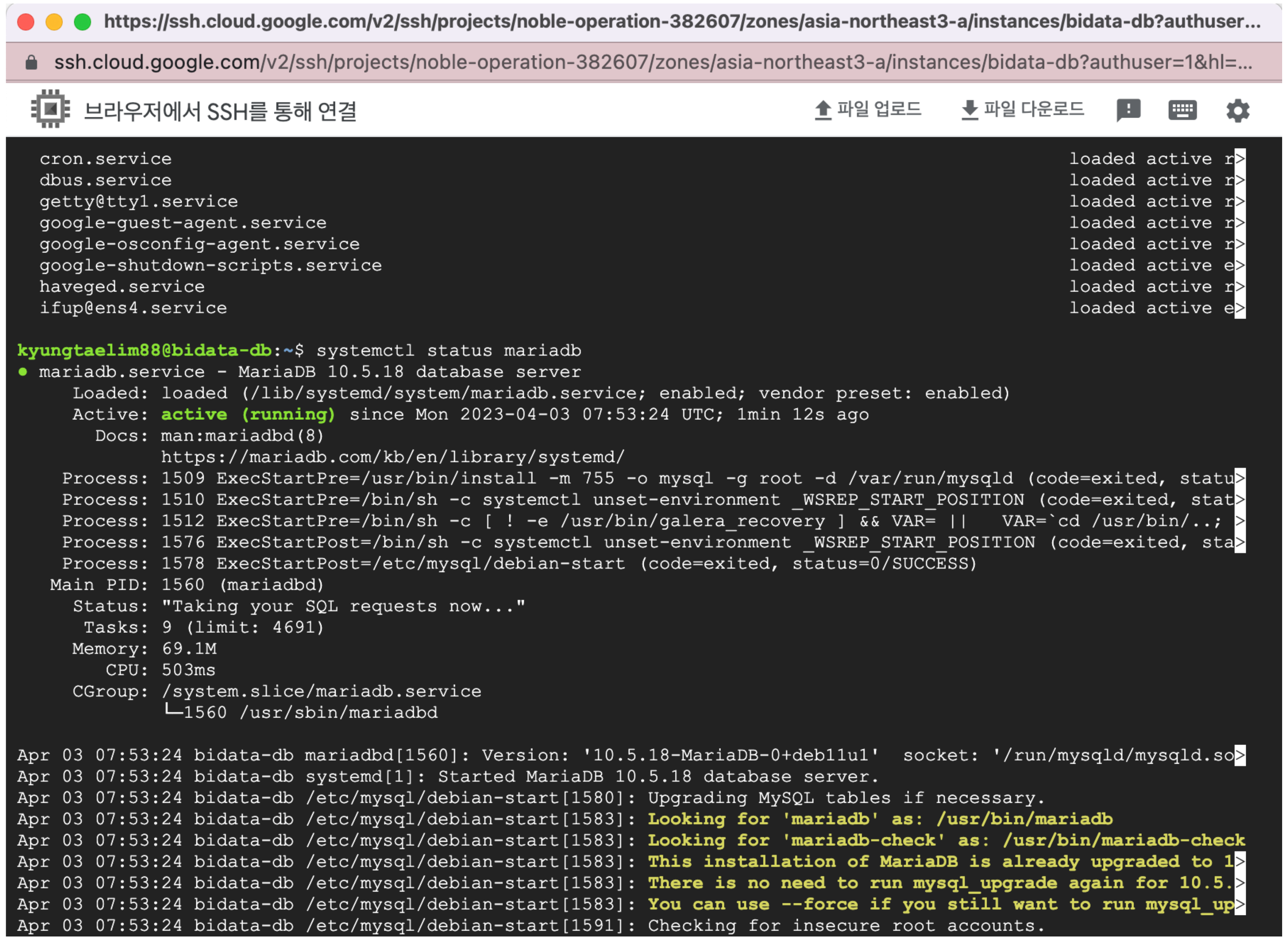
VM 간 부하 분산
트래픽 및 사용자 증가에 따라 애플리케이션에 부하 분산 설정


https://ssh.cloud.google.com/v2/ssh/projects/noble-operation-382607/zones/asia-northeast3-a/instances/bidata-db?authuser...
ssh.cloud.google.com/v2/ssh/projects/noble-operation-382607/zones/asia-northeast3-a/instances/bidata-db?authuser=1&hl=...
브라우저에서 SSH를 통해 연결
파일 업로드
파일 다운로드

```
Linux bidata-db 5.10.0-21-cloud-amd64 #1 SMP Debian 5.10.162-1 (2023-01-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
kyungtaelim88@bidata-db:~$
```

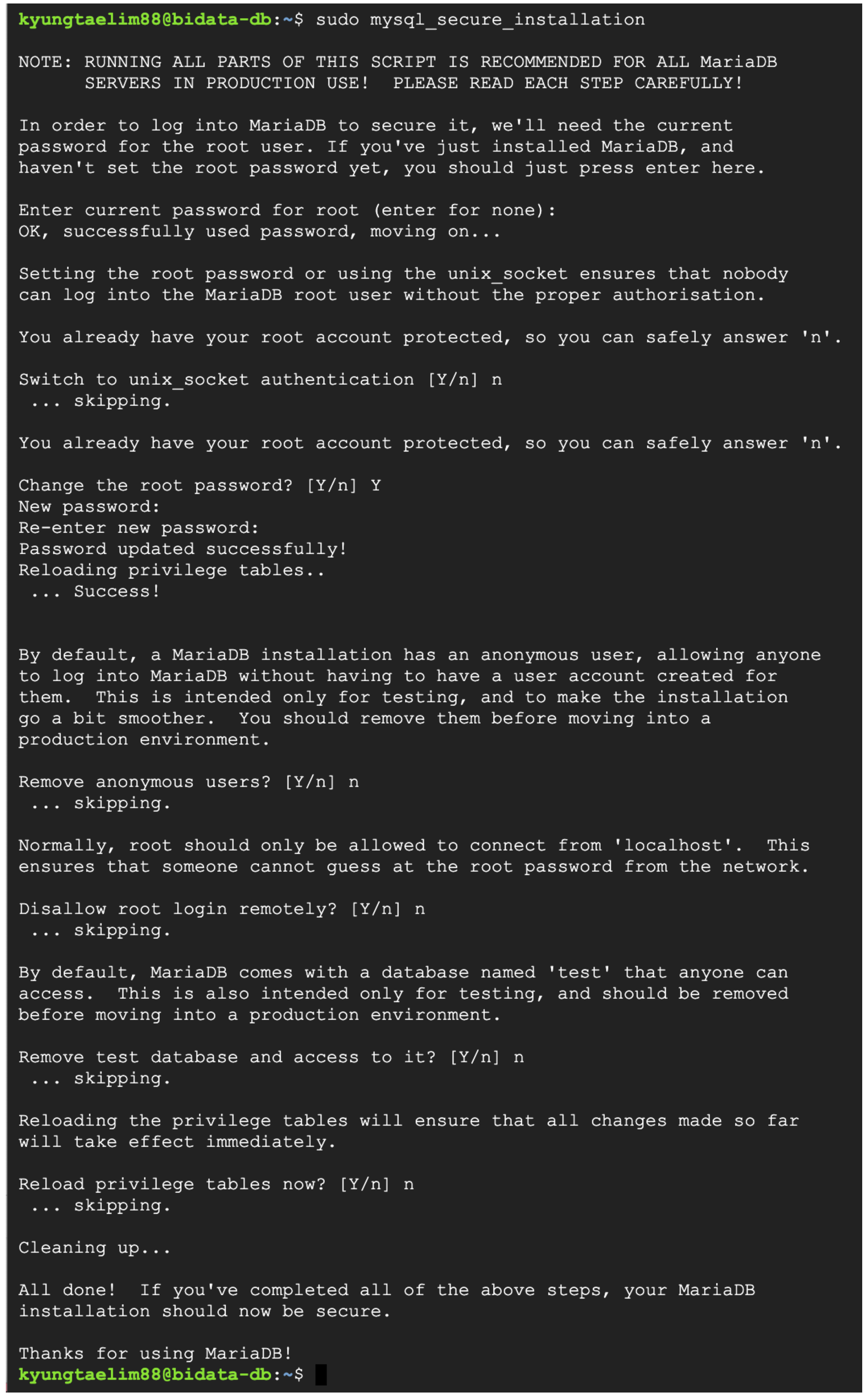
DB setup

kyungtaelim88@bidata-db:~$ sudo apt-get install mariadb-server


https://ssh.cloud.google.com/v2/ssh/projects/noble-operation-382607/zones/asia-northeast3-a/instances/bidata-db?authuser...
ssh.cloud.google.com/v2/ssh/projects/noble-operation-382607/zones/asia-northeast3-a/instances/bidata-db?authuser=1&hl=...
브라우저에서 SSH를 통해 연결
파일 업로드
파일 다운로드

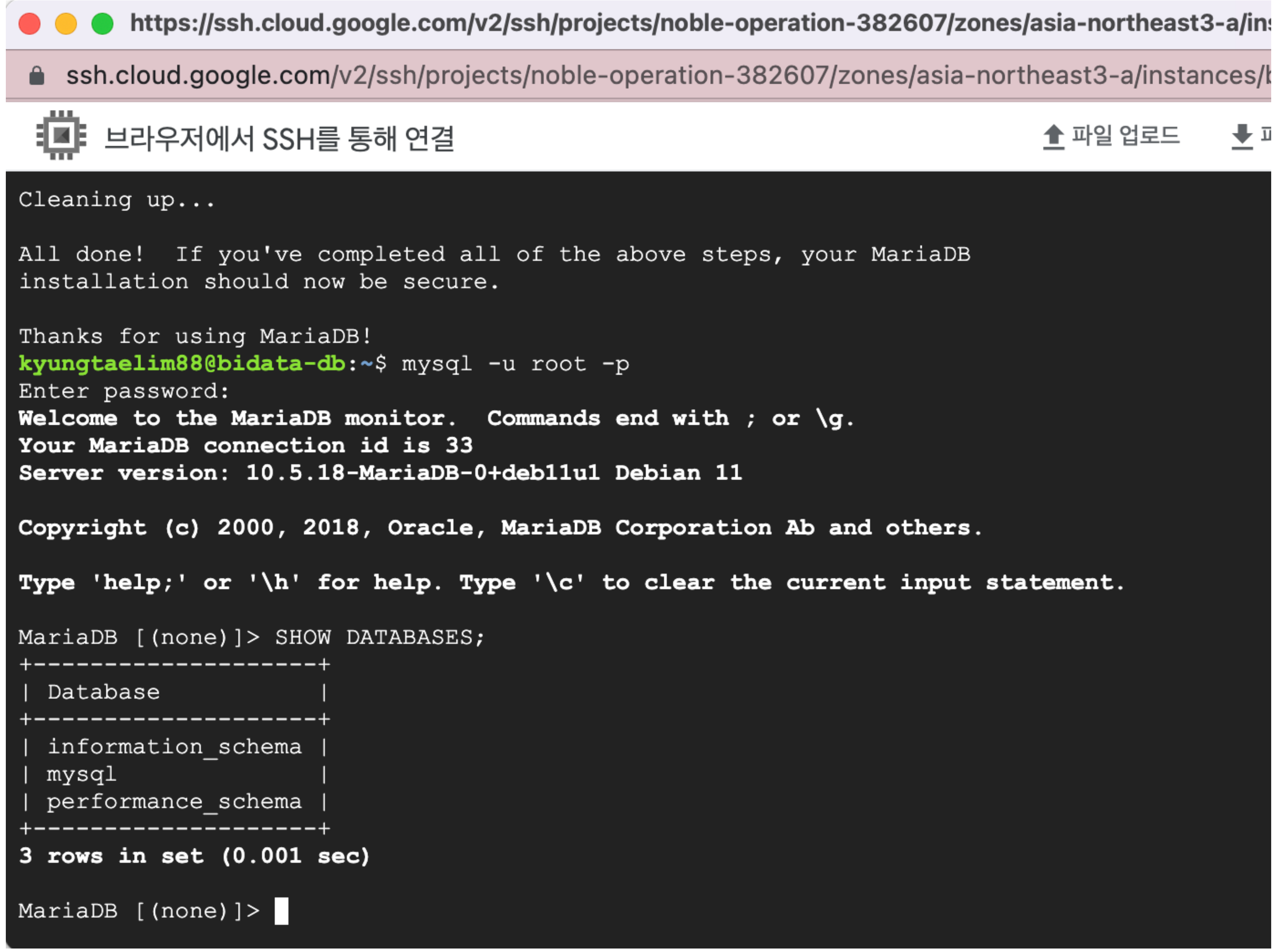
```
  cron.service                                                                               loaded active r>
  dbus.service                                                                               loaded active r>
  getty@tty1.service                                                                         loaded active r>
  google-guest-agent.service                                                                 loaded active r>
  google-osconfig-agent.service                                                              loaded active r>
  google-shutdown-scripts.service                                                            loaded active e>
  haveged.service                                                                            loaded active r>
  ifup@ens4.service                                                                          loaded active e>

kyungtaelim88@bidata-db:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.5.18 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-04-03 07:53:24 UTC; 1min 12s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 1509 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, statu>
    Process: 1510 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, stat>
    Process: 1512 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||   VAR=`cd /usr/bin/..; >
    Process: 1576 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, sta>
    Process: 1578 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 1560 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 4691)
     Memory: 69.1M
        CPU: 503ms
     CGroup: /system.slice/mariadb.service
             └─1560 /usr/sbin/mariadbd

Apr 03 07:53:24 bidata-db mariadbd[1560]: Version: '10.5.18-MariaDB-0+deb11u1'  socket: '/run/mysqld/mysqld.so>
Apr 03 07:53:24 bidata-db systemd[1]: Started MariaDB 10.5.18 database server.
Apr 03 07:53:24 bidata-db /etc/mysql/debian-start[1580]: Upgrading MySQL tables if necessary.
Apr 03 07:53:24 bidata-db /etc/mysql/debian-start[1583]: Looking for 'mariadb' as: /usr/bin/mariadb
Apr 03 07:53:24 bidata-db /etc/mysql/debian-start[1583]: Looking for 'mariadb-check' as: /usr/bin/mariadb-check
Apr 03 07:53:24 bidata-db /etc/mysql/debian-start[1583]: This installation of MariaDB is already upgraded to 1>
Apr 03 07:53:24 bidata-db /etc/mysql/debian-start[1583]: There is no need to run mysql_upgrade again for 10.5.>
Apr 03 07:53:24 bidata-db /etc/mysql/debian-start[1583]: You can use --force if you still want to run mysql_up>
Apr 03 07:53:24 bidata-db /etc/mysql/debian-start[1591]: Checking for insecure root accounts.
```

sudo mysql_secure_installation

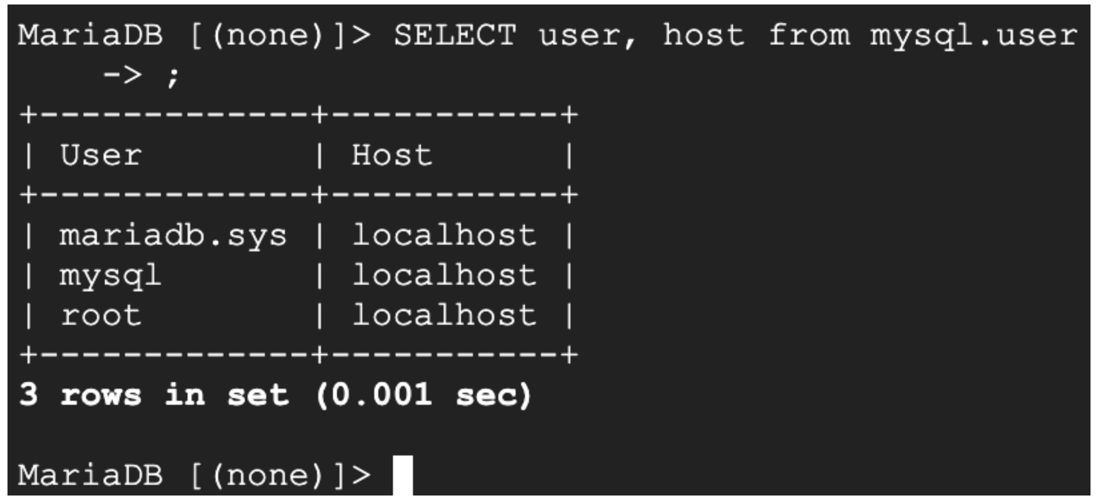


```
kyungtaelim88@bidata-db:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
 ... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] n
 ... skipping.

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
 ... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] n
 ... skipping.

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] n
 ... skipping.

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
kyungtaelim88@bidata-db:~$ 
```

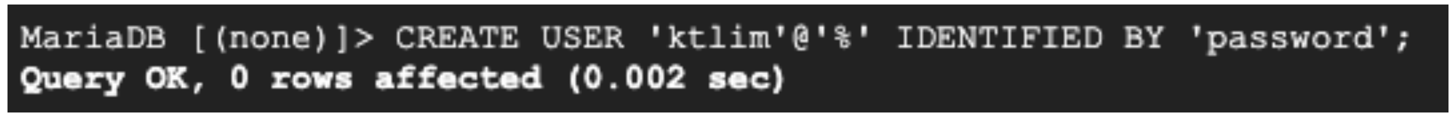
DB setup


https://ssh.cloud.google.com/v2/ssh/projects/noble-operation-382607/zones/asia-northeast3-a/ins
ssh.cloud.google.com/v2/ssh/projects/noble-operation-382607/zones/asia-northeast3-a/instances/b
브라우저에서 SSH를 통해 연결
파일 업로드
파

```
Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
kyungtaelim88@bidata-db:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.5.18-MariaDB-0+deb11u1 Debian 11

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

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

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.001 sec)

MariaDB [(none)]>
```



```
MariaDB [(none)]> SELECT user, host from mysql.user
    -> ;
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
+-------------+-----------+
3 rows in set (0.001 sec)

MariaDB [(none)]>
```



```
MariaDB [(none)]> CREATE USER 'ktlim'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.002 sec)
```

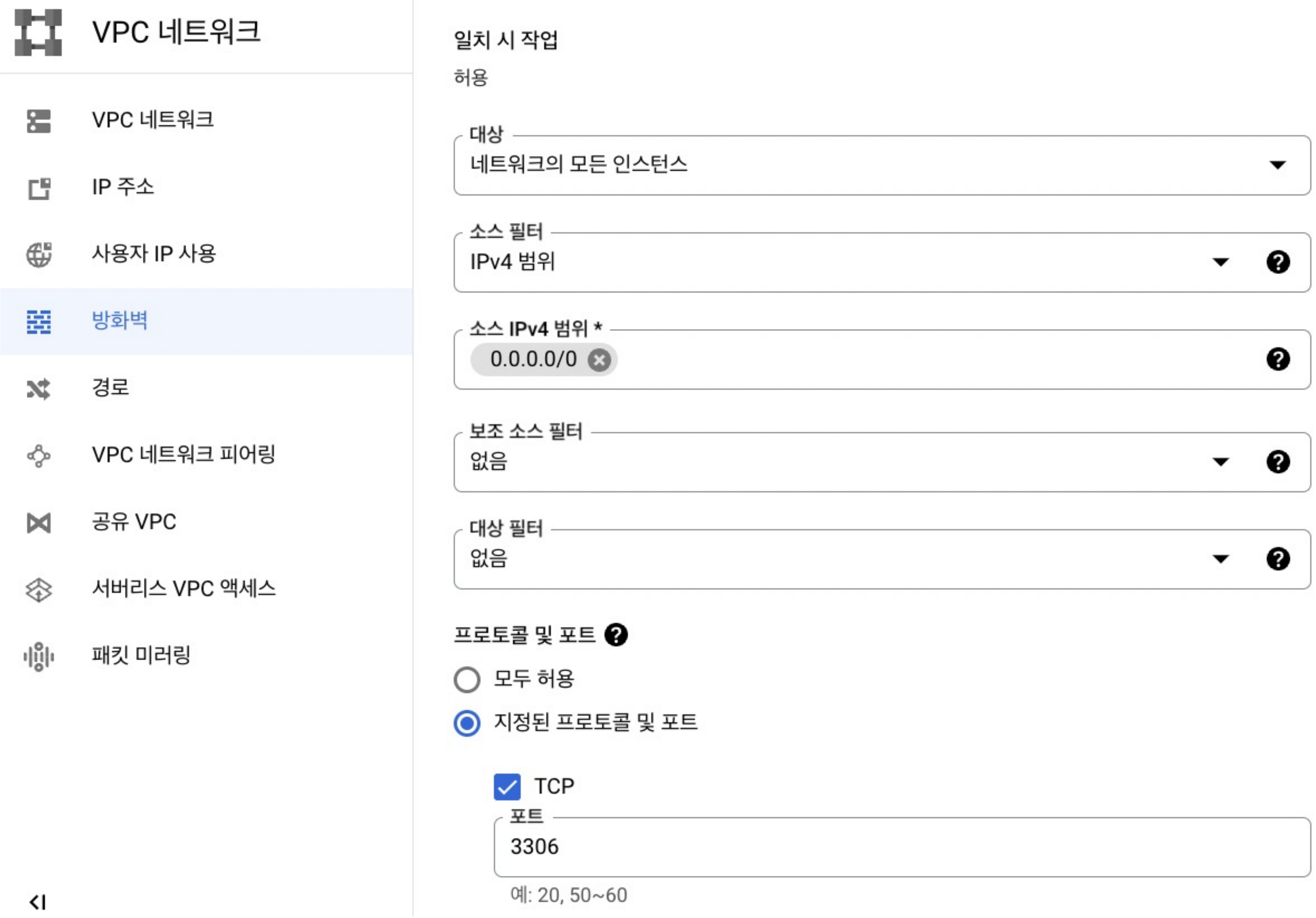
DB setup

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

```
# bind-address = 127.0.0.1
bind-address = 0.0.0.0
```

```
sudo /etc/init.d/mariadb restart
```

DB setup



일치 시 작업

허용

대상
네트워크의 모든 인스턴스

소스 필터
IPv4 범위

소스 IPv4 범위 *
0.0.0.0/0

보조 소스 필터
없음

대상 필터
없음

프로토콜 및 포트

○ 모두 허용

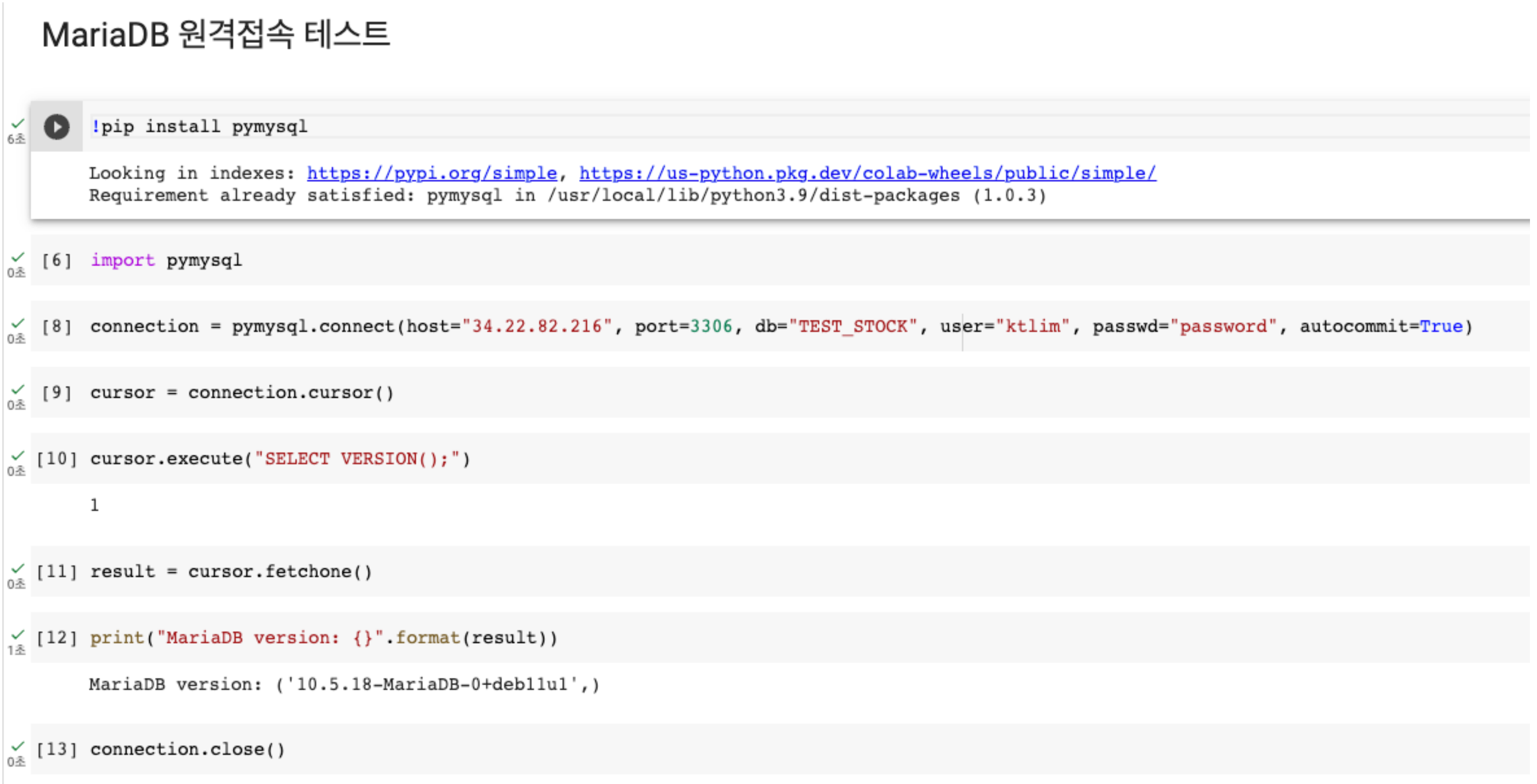
● 지정된 프로토콜 및 포트

☑ TCP

포트
3306

예: 20, 50~60

DB test in Colab



MariaDB 원격접속 테스트

```
6초  !pip install pymysql

     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
     Requirement already satisfied: pymysql in /usr/local/lib/python3.9/dist-packages (1.0.3)

0초  [6] import pymysql

0초  [8] connection = pymysql.connect(host="34.22.82.216", port=3306, db="TEST_STOCK", user="ktlim", passwd="password", autocommit=True)

0초  [9] cursor = connection.cursor()

0초  [10] cursor.execute("SELECT VERSION();")

     1

0초  [11] result = cursor.fetchone()

1초  [12] print("MariaDB version: {}".format(result))

     MariaDB version: ('10.5.18-MariaDB-0+deb11u1',)

0초  [13] connection.close()
```

```
import pymysql
connection = pymysql.connect(host="34.22.82.216", port=3306, db="TEST_STOCK", user="ktlim",
passwd="password", autocommit=True)
cursor = connection.cursor()

cursor.execute("SELECT VERSION();")
result = cursor.fetchone()
print("MariaDB version: {}".format(result))
connection.close()
```

감사합니다.