MySQL 설치형 서비스 소개

네이버 클라우드 플랫폼 MySQL 설치형 서비스의 개념과 기본 구조를 설명합니다. "MySQL"은 세계에서 가장 많이 쓰이는 오픈 소스 관계형 데이터베이스(RDBMS)인 MySQL 을 이용하고 있습니다.

MySQL

전세계에서 가장 인기 있고, 많이 사용하고 있는 오픈소스 기반의 관계형 데이터베이스 시스템(Relational Database Management System, 이하 RDBMS)입니다. 다른 RDMBS 와 마찬가지로 데이터베이스를 생성하고 테이블, 칼럼 단위의 구조를 사용하여 레코드 단위의 데이터를 조회/변경/추가/삭제합니다. 별도의 GUI 를 제공하지 않으며, 주로 CLI(Command Line Interface)를 사용하여 관리하고, GUI 기반의 관리를 위해 MySQL 에서 제공하는 Workbench 나 Third Party-Tool 을 사용합니다.

2008 년 Sun 이 MySQL AB 를 인수한 후 2010 년 Oracle 이 Sun 을 인수하면서 현재 MySQLEnterprise/Community Version 의 모든 권리는 Oracle 이 소유하고 있습니다.

네이버 클라우드에서 제공하는 MySQL 설치형 서비스에서는 MySQL Community Edition(GPL)을 제공하여 Basic Install 수준의 기 설치된 이미지를 지원합니다.

네이버 클라우드 플랫폼 MySQL 설치형 서비스 구성

설치형 서비스

네이버 클라우드에서 지원하는 MySQL 설치형 서비스의 경우 DB 설치에 대해 원활한 운영표준을 제시하는 환경파일 제공 및 DB 상태를 볼수 있는 CLI 형태의 모니터링 Binary 를 포함한 MySQL Daemon 에 Startup 상태까지 지원합니다. 이에 관련하여 **DB 설치 이외 부분에 대해서는 기술 지원하지 않고 있습니다**.

RPM/APT

MySQL 설치 방법에는 Source Compile 설치 방법, RPM/APT 설치 방법, Binary 설치 방법이 있으며, 네이버 클라우드에서는 Normal Install 방법인 RPM(APT) 설치 이미지를 제공합니다. RPM(APT)에서 사용하는 .repo 파일은 MySQL 에서 공식적으로 제공하는 파일을 사용하여 설치를 진행하며 Repo 에서 제공하는 Latest Version 을 기본으로 하여 서버 이미지에 포함하여 제공합니다.

RPM Install

CentOS 6.x/7.x with MySQL 에서 사용하는 RPM 설치되는 패키지는 아래와 같이 구성됩니다.

MySQL 설치 및 실행에 관련되어 OS Default 설정에서 추가로 설치되는 패키지입니다.

- mysql-server
- mysql-devel
- libmysqlclient-dev
- build-essential

APT Install

Ubuntu 14.X/16.X with MySQL 에서 사용하는 APT 로 설치되는 패키지는 아래와 같이 구성됩니다.

MySQL 설치 및 실행에 관련되어 OS Default 설정에서 추가로 설치되는 패키지입니다.

- mysql-server
- libmysqlclient-dev

Recommanded Parameters (5.6/5.7 공통)

네이버 클라우드 플랫폼내의 MySQL 설치형 이미지에서 표준화하여 Recommanded Parameter 는 아래와 같이 구성됩니다.

- innodb_file_per_table
- innodb_log_file_size = 512M
- innodb_log_files_in_group = 2
- default_storage_engine=InnoDB
- query_cache_type = ON
- query_cache_size = 0M
- sysdate-is-now

innodb_file_per_table

InnoDB 테이블 생성 시 InnoDB Global Tablespace 를 사용하면, 테이블 삭제 시 이미 늘어난 물리 크기는 줄어들지 않습니다. innodb_file_per_table 옵션을 사용하면 InnoDB 테이블 생성 시 테이블당 파일을 생성하고 테이블 삭제 시 해당 파일이 삭제되어 물리 공간을 확보할 수 있습니다.

Innodb_log_file_size/innodb_log_files_in_group

Heavy Traffic 이 들어오는 경우 CheckPoint 를 위한 병목 현상을 방지하기 위해 innodb_log_file_size 옵션을 늘려서 설정합니다. 해당 Variables 을 조정하기 위해서는 DB Restart 가 필요합니다.

default_storage_engine

Online-DDL 및 Transction 사용이 가능한 InnoDB 로 표준 지정합니다.

query_cache_type/query_cache_size

DB 기동 후 query_cache 를 사용자가 동적으로 ON/OFF 할 수 있도록 ON/0 으로 설정합니다. 이후 서비스 상황에 맞도록 조정합니다.

sysdate-is-now

ysdate 를 사용하는 경우 Replication 구성에서 Master-Slave 간의 데이터 처리 시간이 다를 경우 Time Type 데이터가 다를 수 있어 sysdate 의 경우 자동으로 now 를 replace 합니다.

MySQL 설치형 서비스 관리

MySQL 설치형 서비스에서 관련되는 관리 방법을 설명합니다.

MySQL 이미지 서버 생성

MySQL 이 설치된 인스턴스 서버를 생성하면 아래와 같이 네이버 클라우드에 표준화된 my.cnf 의 환경을 가진 MySQL Daemon 이 Startup 되어 생성된 후 Daemon 기동까지 완료된 후 제공됩니다.

my.cnf

my.cnf 의 Parameter 설명은 아래와 같습니다.

속성	설명
expire_logs_days = 7	log-bin으로 설정된 로그 크기가 커지는 것을 방지하기 위해 7일간 저장하도록 합니다.
innodb_file_per_table	InnoDB 테이블 생성 시 InnoDB Global Tablespace를 사용하면, 테이블 삭제 시 이미 늘어난 물리 크기는 줄어들지 않습니다. innodb_file_per_table 옵션을 사용하면 InnoDB 테이블 생성 시 테이블당 파일을 생성하고 테이블 삭제 시 해당 파일이 삭제되어 물리 공간을 확보할 수 있습니다.
innodb_log_file_size	Heavy Traffic이 들어오는 경우 CheckPoint로 인한 병목을 방지하기 위해 innodb_log_file_size 옵션을 늘립니다. Variables을 조정하기 위해서는 DB Restart가 필요함.
iinnodb_log_files_in_group	innodb_log_file의 개수를 지정합니다. Rotation을 위해 2개 이상을 권고합니다.

속성	설명
default_storage_engine	Online-DDL 및 Transction 사용이 가능한 InnoDB로 표준 지정합니다.
query_cache_type query_cache_size	DB 기동 후 query_cache를 사용자가 동적으로 ON/OFF할 수 있도록 ON/0으로 설정합니다. 이후 서비스 상황에 맞도록 조정합니다.
sysdate-is-now	sysdate를 사용하는 경우 Replication 구성에서 Master-Slave 간의 데이터 처리 시간이 다를 경우 Time Type 데이터가 다를 수 있어 sysdate의 경우 자동으로 now를 replace합니다.
log_timestamps = SYSTEM(5.7 only)	MySQL 5.7에서 LogTime이 UTC으로 되어있어 SYSTEM Local TIME으로 변경합니다.

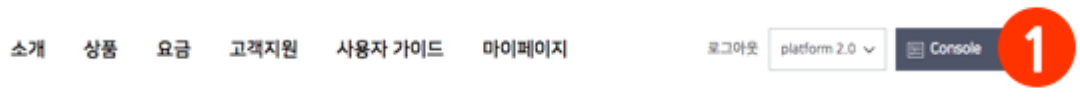
MySQL 설치된 디렉터리는 아래와 같습니다.

속성	설명
DATADIR	/var/lib/mysql/
SLOW_LOG	/var/lib/mysql/*slow.log
ERROR_LOG	/var/log/mysqld.log - CentOS /var/log/mysql/error.log - Ubuntu
my.cnf	/etc/my.cnf - CentOS /etc/mysql/my.cnf - Ubuntu.
sqlstatus_nCloud	/usr/bin/sqlstatus_nCloud

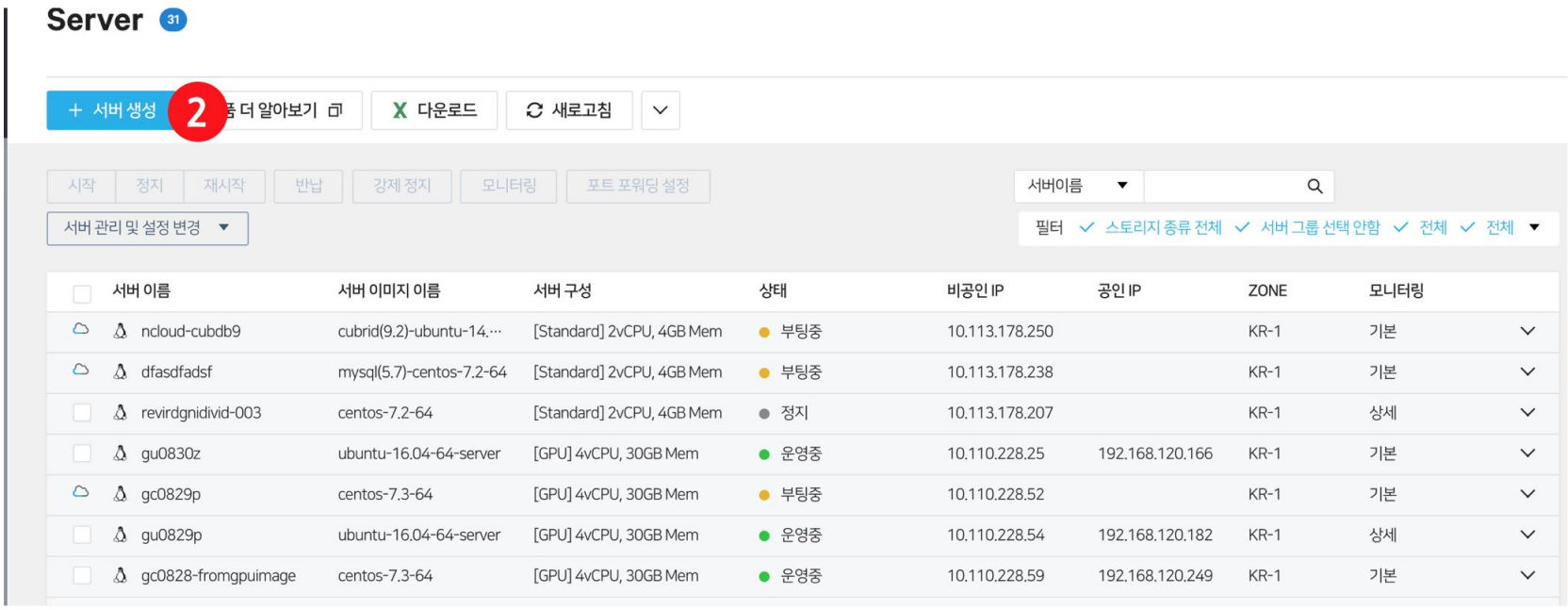
서버 생성 절차

MySQL 서버 생성 과정은 아래와 같습니다.

www.ncloud.com 접속 및 Login



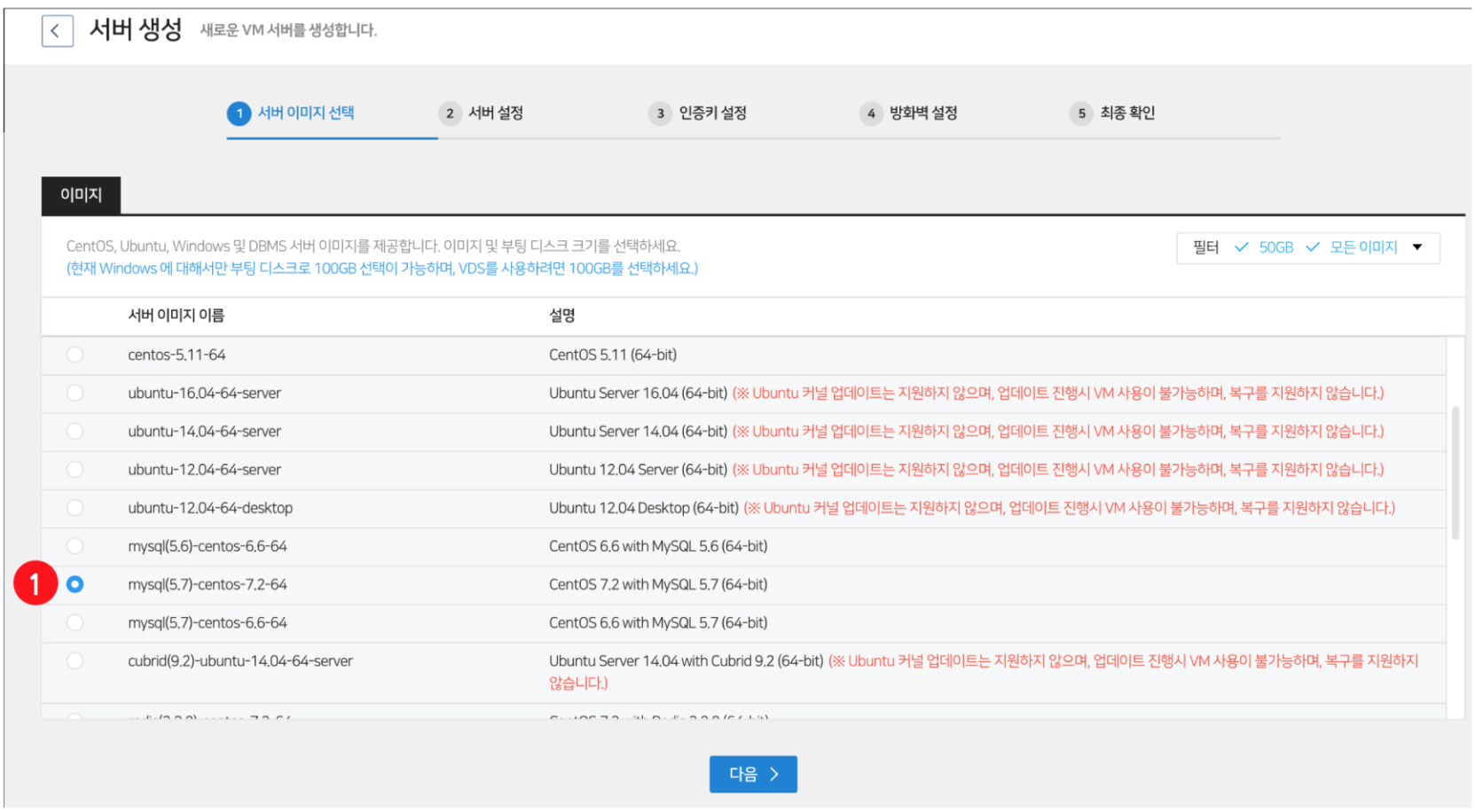
① 우측 상단 콘솔을 선택하여 콘솔에 접속합니다.



② 콘솔 화면에서 서버 생성 선택을 합니다.

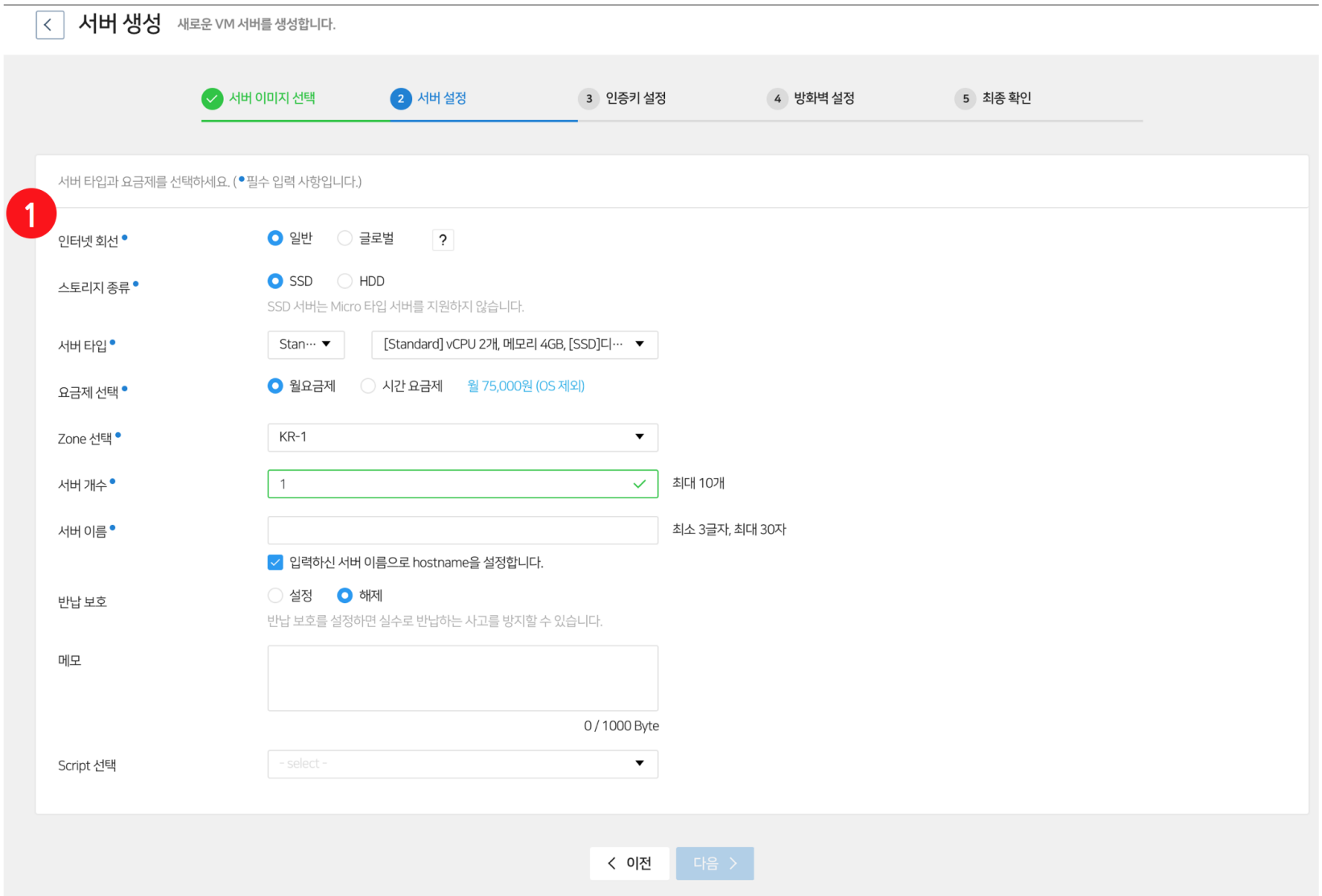
서버 생성

Step 1. 서버 이미지 선택



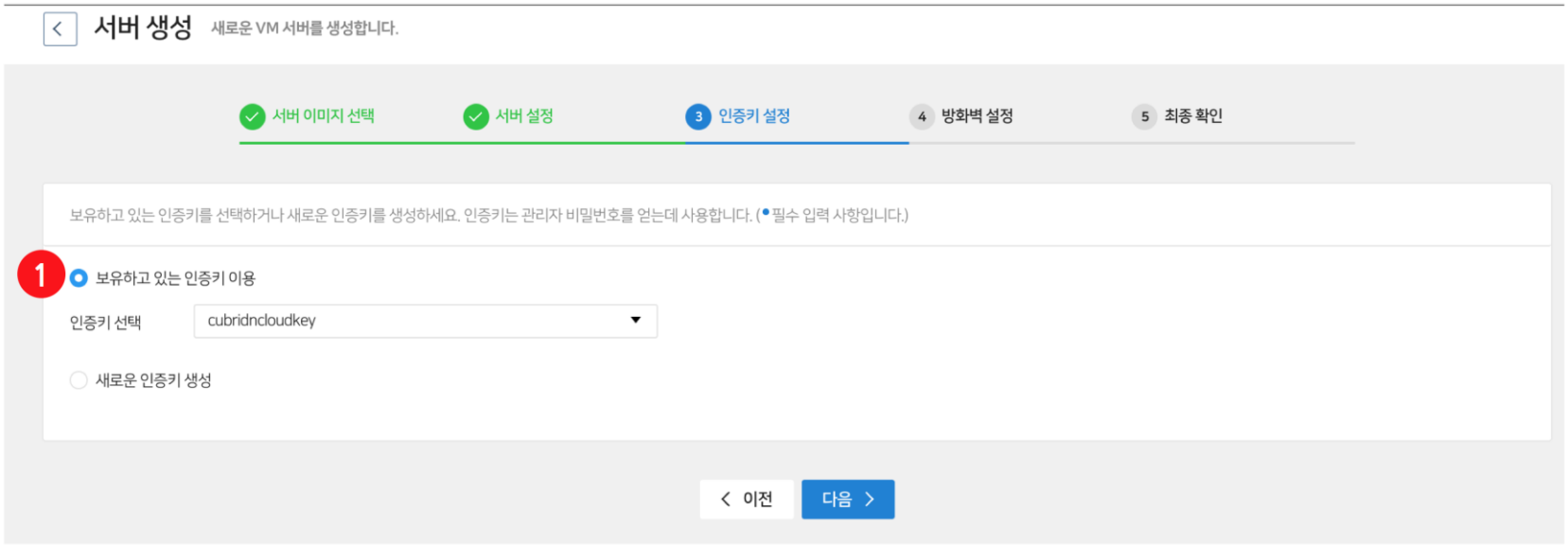
① OS Version 에 해당하는 MySQL Version 을 체크한 후 하단의 [다음] 버튼을 선택합니다.

Step 2. 서버 설정



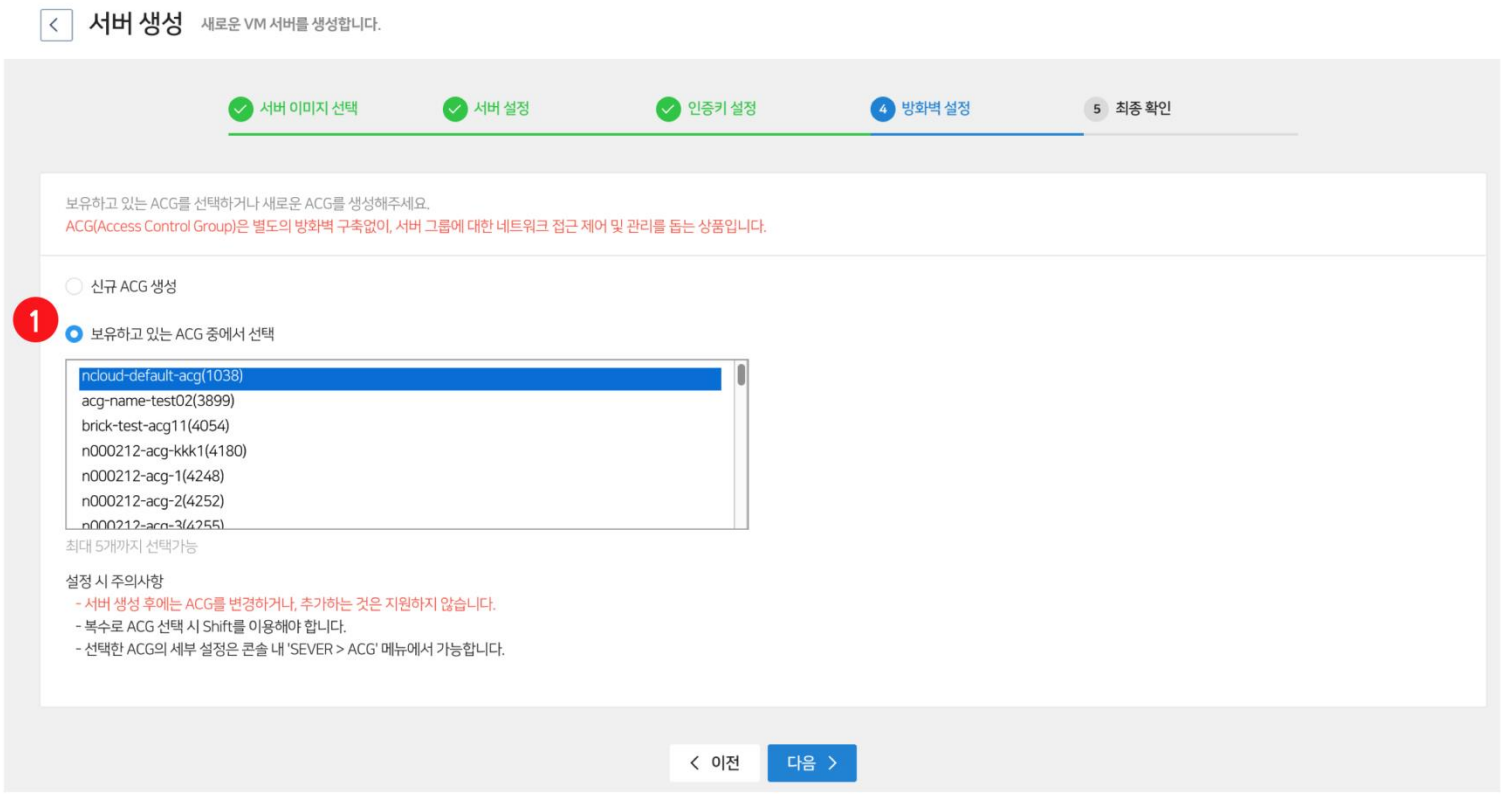
① 아래와 같이 서버타입과 원하는 요금제를 선택합니다.

Step 3. 인증키 설정



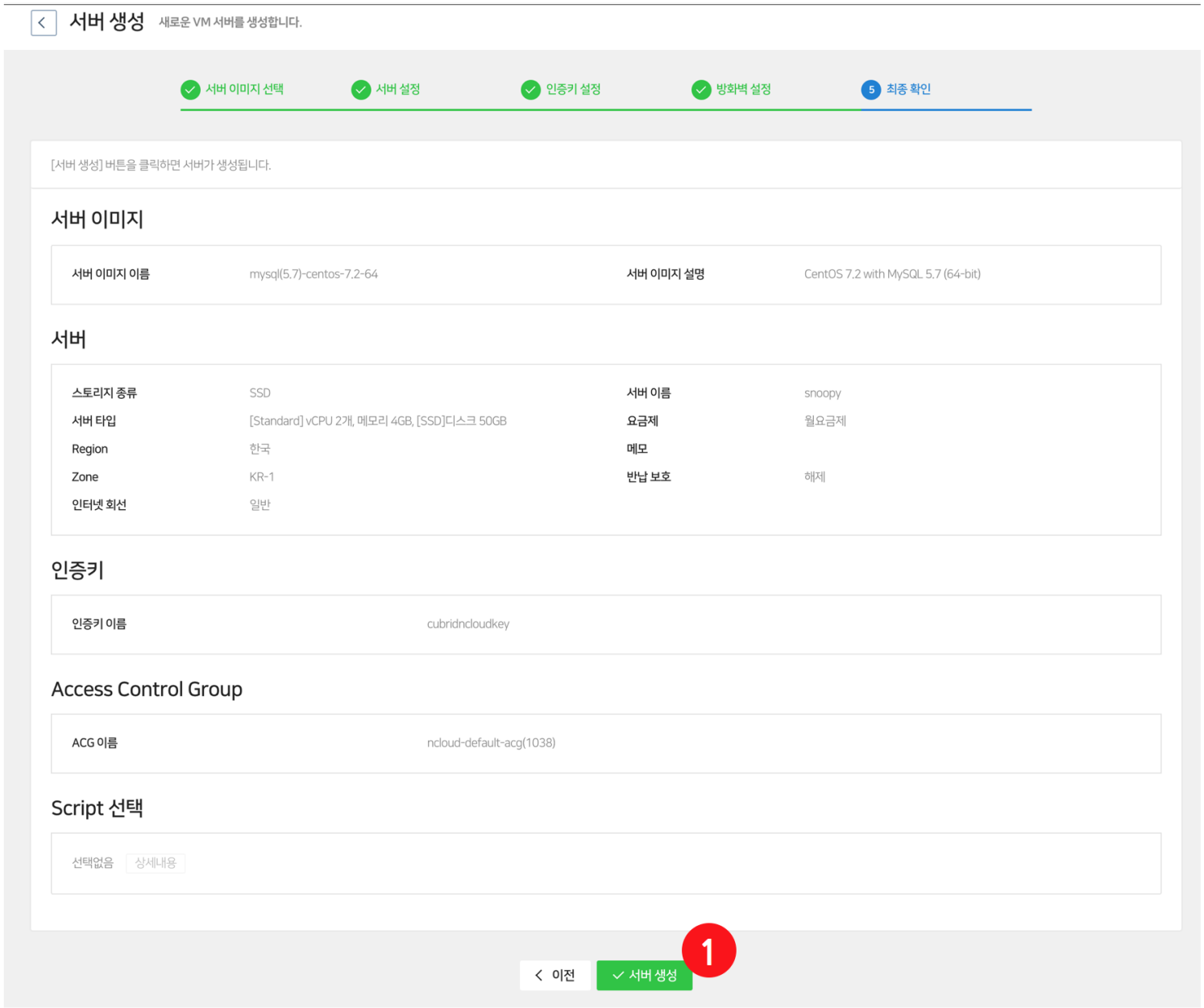
① 사용자의 인증키를 선택합니다.

Step 4. 방화벽 설정



① 보유하고 있는 ACG 를 선택하거나 신규로 생성합니다.

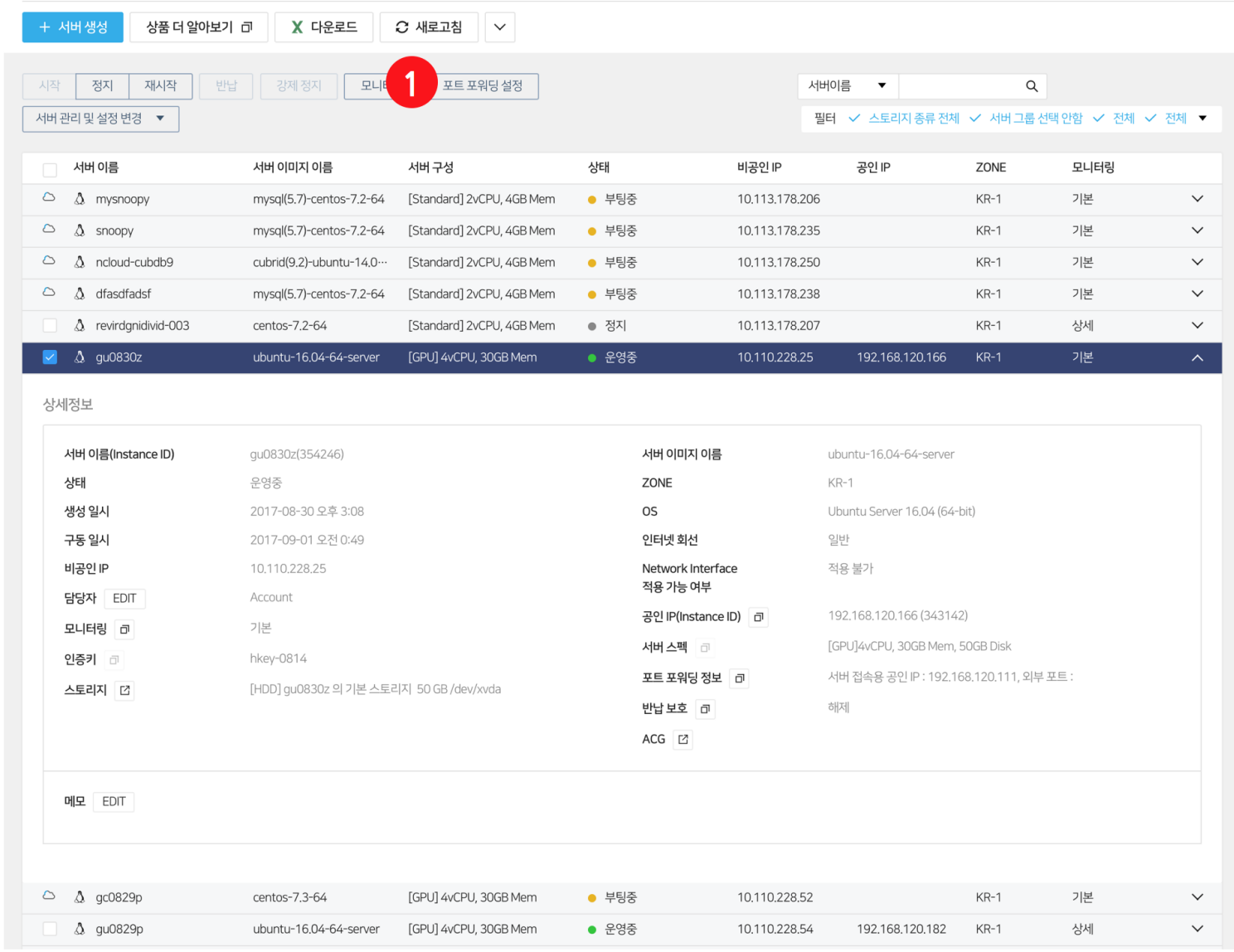
Step 5. 최종확인



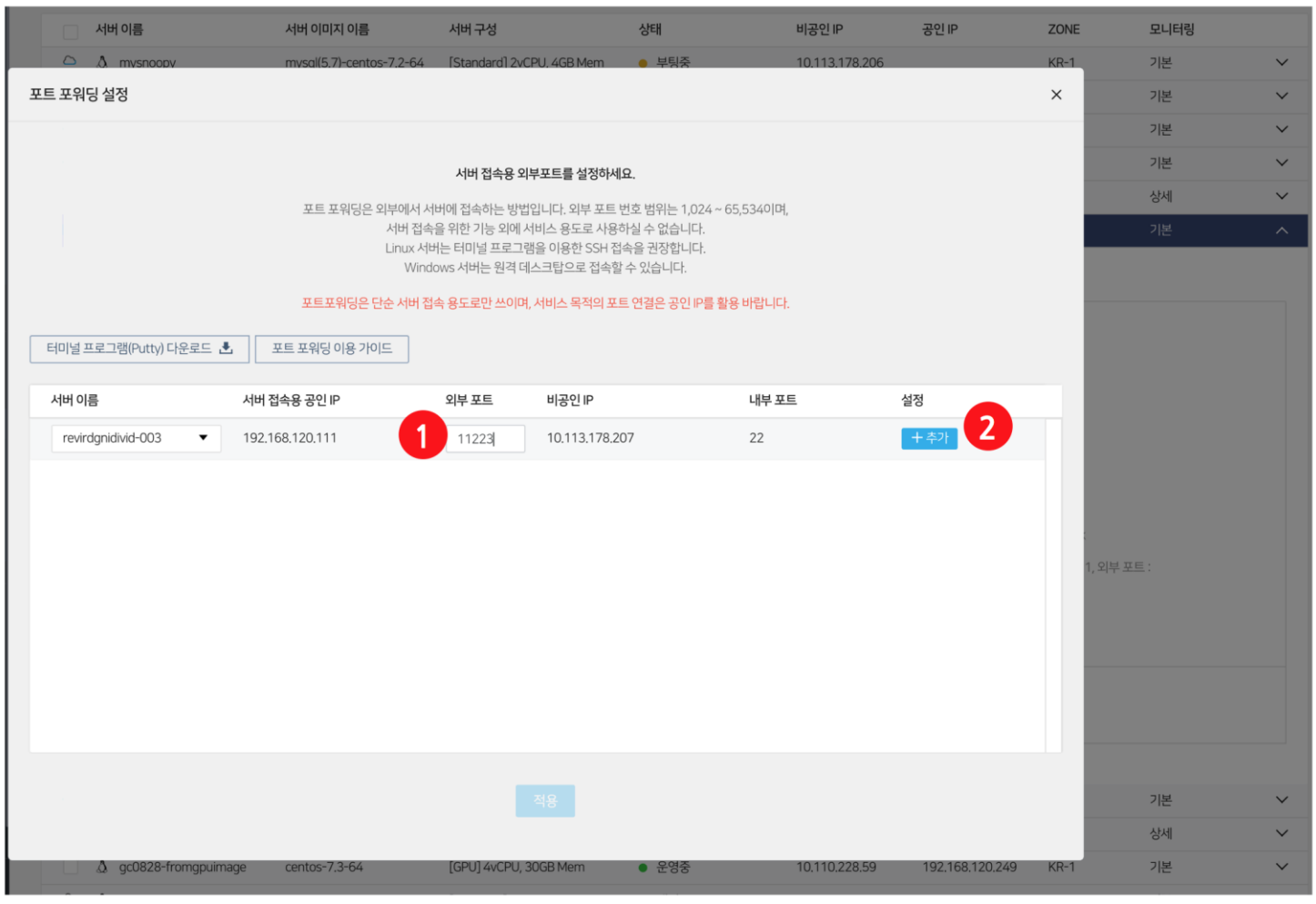
① 선택한 내용를 확인한 후에 MySQL 이 설치된 서버를 생성합니다

서버접속 설정 및 확인

포트 포워딩 설정

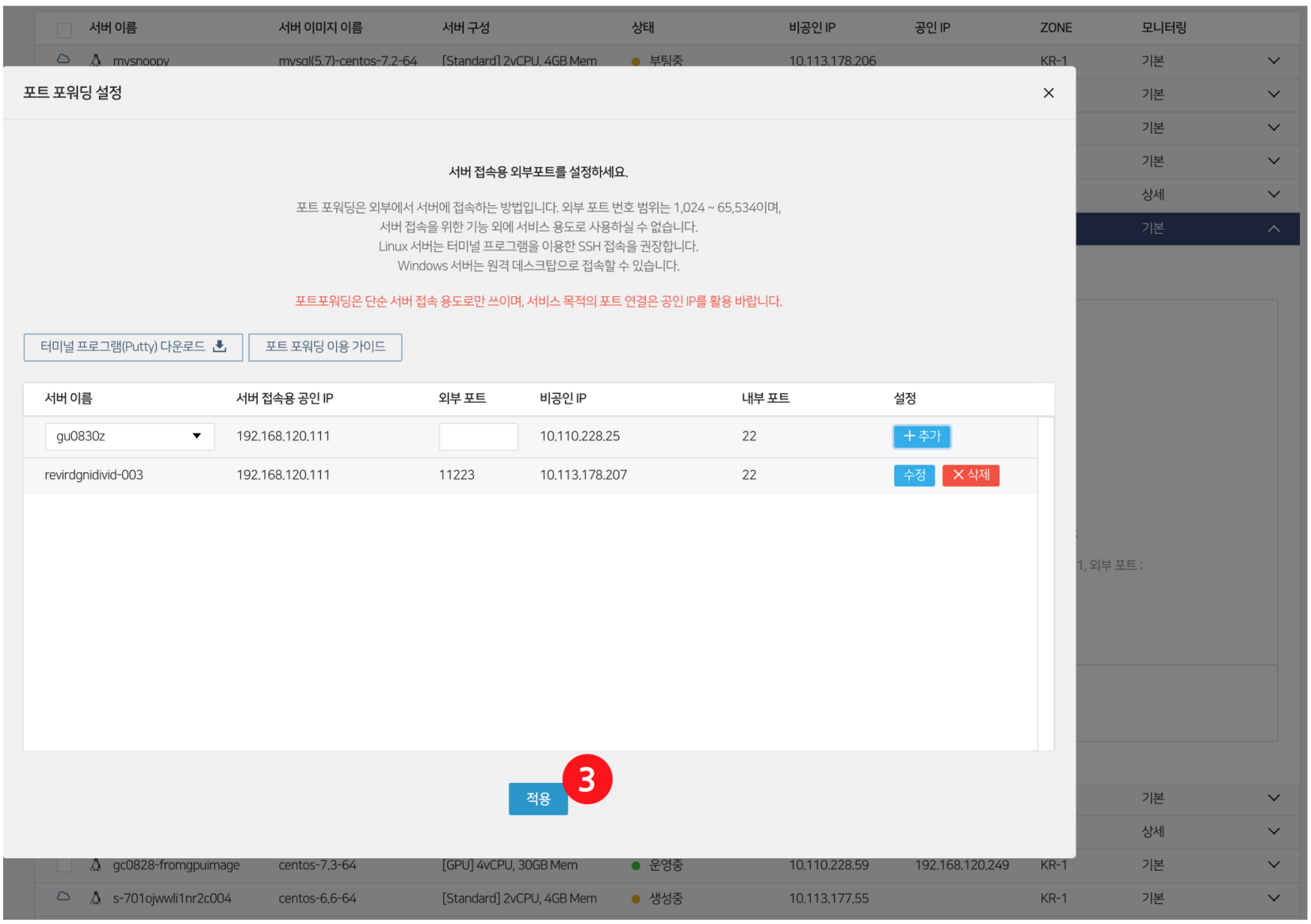


① 서버 생성 이후 상단의 [포트 포워딩 설정] 버튼을 선택합니다.



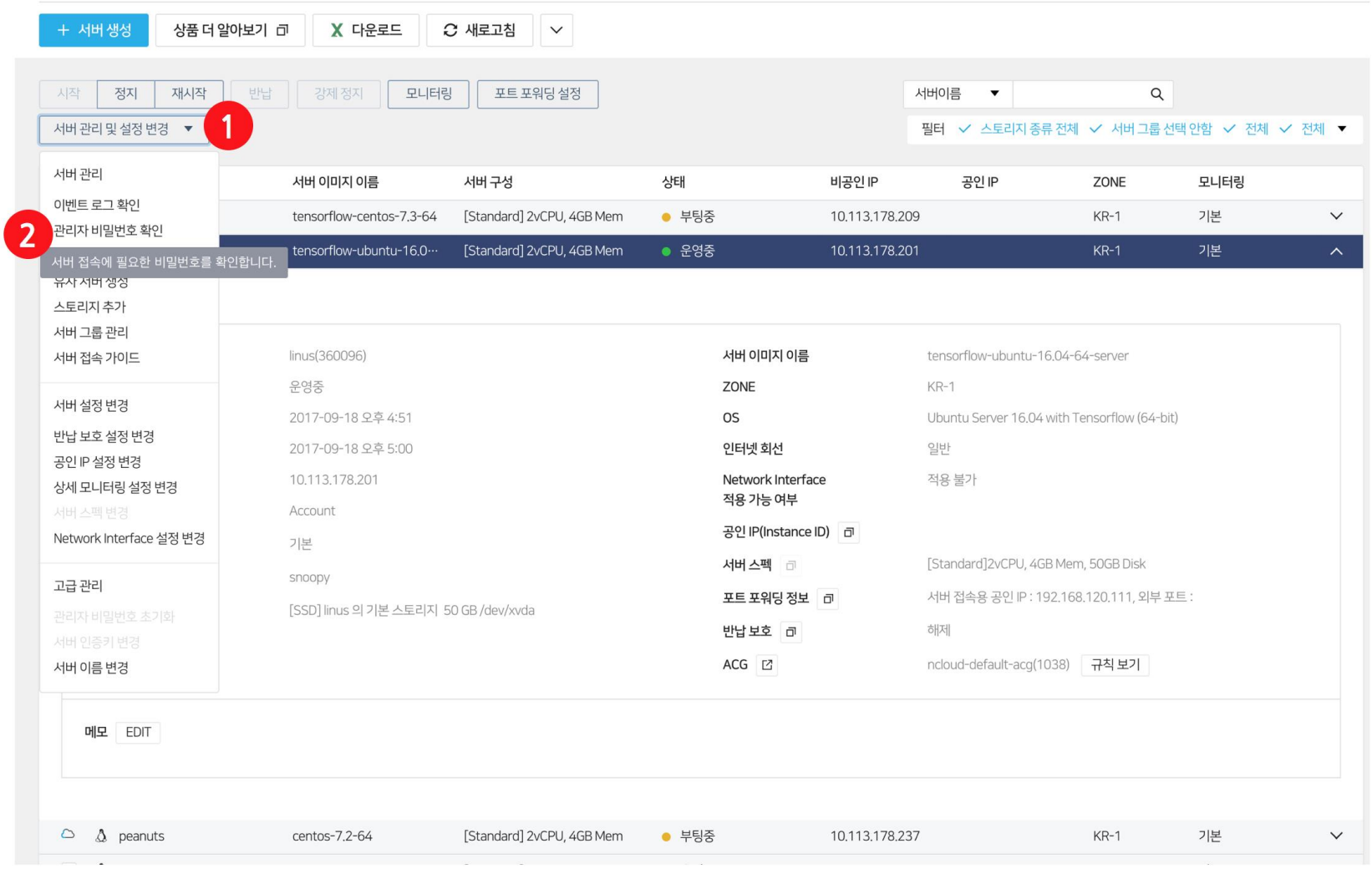
① 아래 화면의 포트 포워딩 설정에서 ssh 접속을 위한 외부포트를 등록합니다.

② 외부포트등록후 [추가]를 선택합니다.



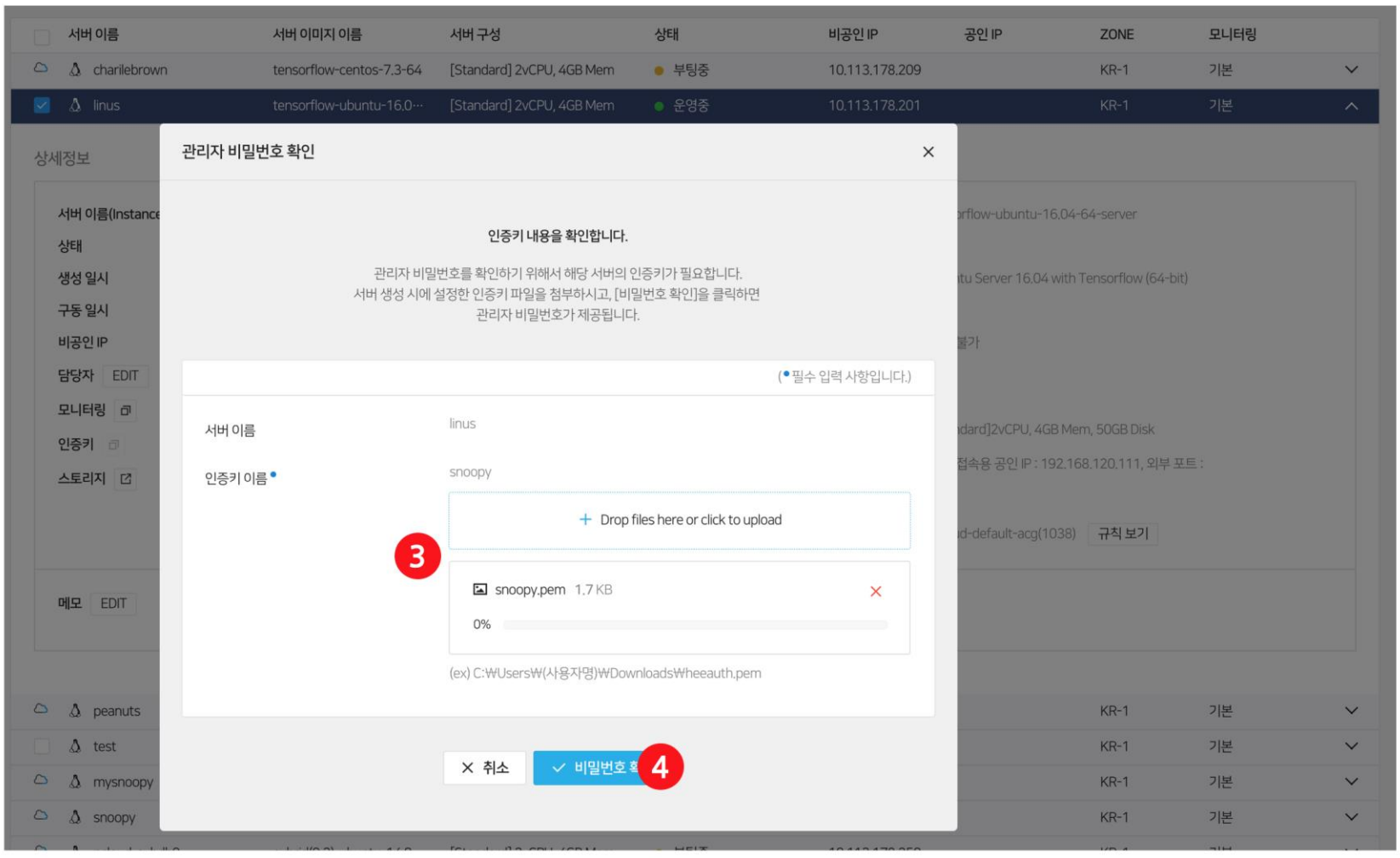
③ [적용]을 클릭합니다.

관리자 비밀번호 확인



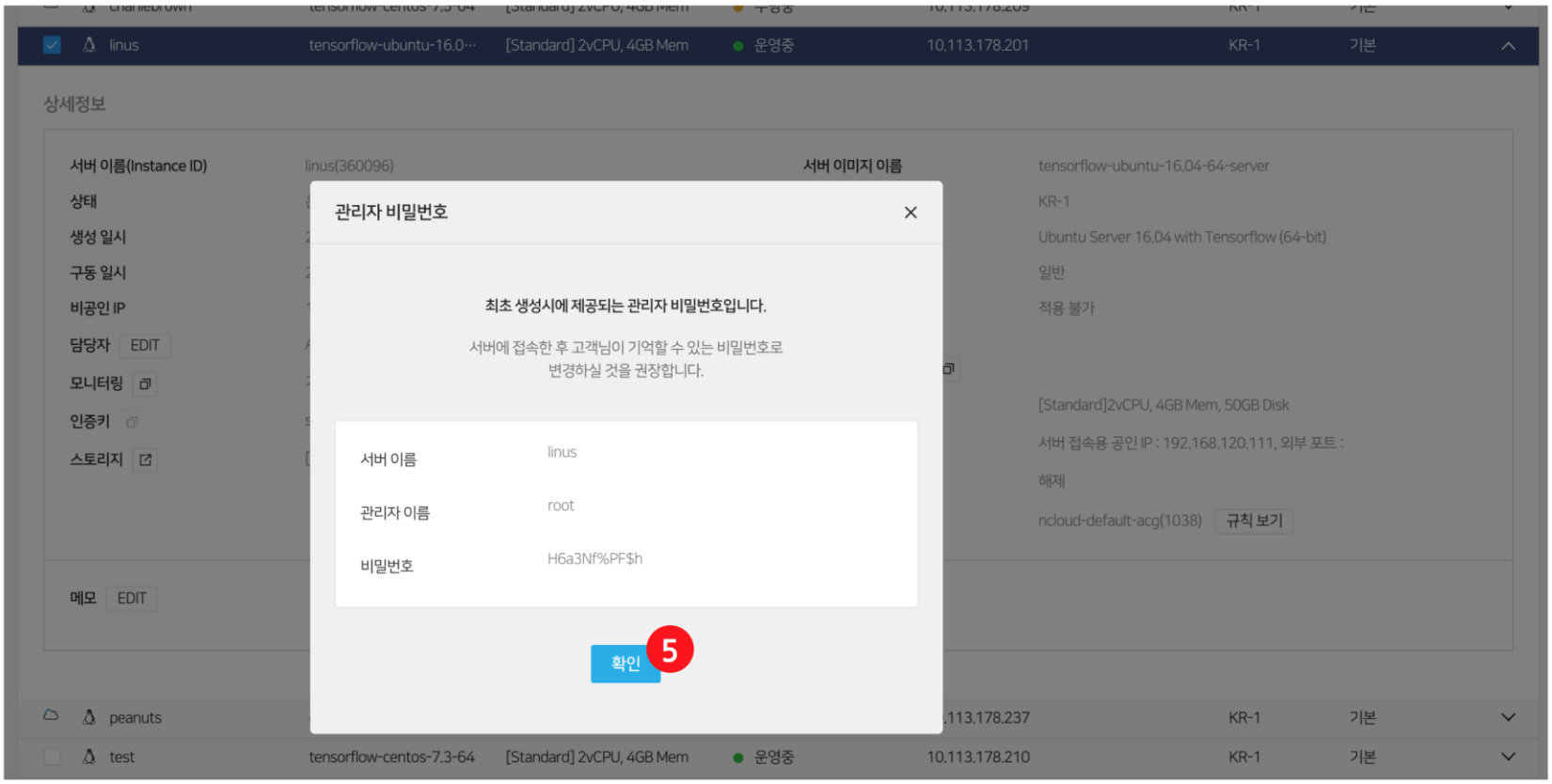
① 포트 포워딩 설정이후 상단의 [서버관리 및 설정변경] 버튼을 선택합니다.

② 이후 팝업 메뉴에 [관리자 비밀번호 확인]을 선택합니다.



③ 관리자 비밀번호 확인창에서 [파일선택]으로 인증키 파일을 선택합니다.

④ [비밀번호 확인] 버튼을 클릭합니다.



⑤ 이후 화면에 나온 비밀번호를 인지합니다.

터미널을 통한 서버접속

터미널프로그램을 사용하여 아래와 같이 접속합니다.

- PORT : 포트 포워딩으로 설정한 포트번호
- 공인 IP : 포트포워딩 화면에 안내된 서버 접속용 공인 IP
- 패스워드 : 관리자 비밀번호에 출력된 패스워드(초기 패스워드는 반드시 변경합니다.)

```
ssh -p [PORT] root@[공인 IP]
```

서버 이미지 생성이후

1. 서버 생성 이후 바로 사용하지 않을 경우 보안 위험을 예방하기 위해 **MySQL Daemon 을 ShutDown** 합니다.
2. 생성된 MySQL 의 경우 패스워드가 설정되지 않았기 때문에 **반드시 패스워드 설정**을 합니다.

MySQL 기동/정지 및 접속 절차

- MySQL Daemon 기동및 정지는 아래와 같은 명령어로 실행합니다.

  - CentOS
  - `shell> /sbin/service mysqld start`
  - `shell> /sbin/service mysqld stop`
  - Ubuntu
  - `Shell> /usr/sbin/service mysql start`
  - `Shell> /usr/sbin/service mysql stop`
- MySQL Daemon 구동 유무는 아래와 같은 명령어로 확인합니다.

```
Shell> ps -ef | grep mysqld
```

  - 결과: mysqld/mysqld 프로세스가 보여야 합니다.
- 네이버 클라우드에서 생성한 MySQL 은 아래 명령어로 접속합니다.

```
Shell> mysql -uroot
```

- MySQL Password 정책에 따라 초기 패스워드는 지정되지 않습니다.
- root Password 의 경우 설정하지 않으므로 보안 침해 방지를 위해 **최초 접속 시 반드시 패스워드 변경을 합니다.**
- root 패스워드는 아래 명령어를 사용하여 변경합니다.

  - MySQL 5.6
  - `mysql> set password=password('패스워드');`
  - MySQL 5.7
  - `mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '패스워드';`

- MySQL Built-in Password 기본 정책은 아래와 같습니다.
  - 최소 길이 8 자 이상
  - 특수문자 1 개 이상
  - 숫자 1 개 이상
  - 대소문자 조합 1 개 이상

MySQL 이미지 서버 반납

네이버 클라우드에서 MySQL 이 생성된 서버 반납 시 아래와 같은 절차 이후 반납하기를 권장합니다.

1. MySQL 에서 저장한 데이터 백업
   - 백업된 데이터의 경우 Download 이후 별도 보관
2. MySQL Daemon 정지
3. mysql 이 저장된 Data Directory 삭제
   - 예: rm -rf /var/lib/mysql
4. 네이버클라우드 콘솔에서 [정지]->[반납] 순서로 선택하여 반납을 진행합니다.

MySQL 외부접속 허용

Ubuntu 에서 MySQL 을 사용하는 경우 Default Parameter 로 인해 Localhost 에서만 접속이 가능합니다.
이에 외부접속이 가능하도록 아래와 같은 예로 가이드합니다.

1. MySQL DB 접속
2. 계정 권한 부여
   예: GRANT ALL PRIVILEGES ON *.* to '계정명'@'%' IDENTIFIED BY '비밀번호';
3. 환경파일 수정
   예: sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
4. bind-address 주석처리

```
# By default we only accept connections from localhost
# bind-address  = 127.0.0.1
```

5. MySQL 재시작

SQLSTATUS

네이버 클라우드에서 제공하는 MySQL 서비스에는 CLI(Command Line Interface) 형태의 모니터링 프로그램이 제공됩니다.

- 해당 프로그램은 **nCloud 서버 이외에 외부 유출 및 임의로 변경이나 변조하는 경우를 금지**하고 있습니다.

개요

MySQL 의 내부 Status 를 1 초 단위로 볼 수 있도록 Binary 형태로 제공합니다.
1 초간의 MySQL Status 를 Display 를 하여 Real-Time 으로 DB 의 상태 변화를 볼 수 있습니다.

사용법

sqlstatus_nCloud 에서 사용할 계정 생성(Default 계정으로 포함)

```
 GRANT PROCESS, SHOW DATABASES, SUPER, EXECUTE, REPLICATION CLIENT ON *.* TO
'sqlstatus_ncloud'@'127.0.0.1' IDENTIFIED BY PASSWORD '*AB9C7542BD3FE96F5342F2A616D030CF4494EB44';
```

설치 경로 : /usr/bin/sqlstatus_nCloud

기본 3306 포트 사용 시 : sqlstatus_nCloud

3306 이외 사용자 Port 사용 시 : sqlstatus_nCloud 포트번호

사용하지 않을 경우 Default 계정을 삭제해도 됩니다.(`DROP USER sqlstatus\_ncloud@127.0.0.1;`)

사용자 화면

```
 - Interval Time : 1 Sec
 - Port          : 3306
 > Time          : Current Time
 > Conn          : Thread Connection
 > Run           : Thread Running
 > Abort         : Aborted Connection
 > Select        : Select Count
 > Update        : Update + Update_Multi Count
 > Insert        : Insert + Insert_Select Count
 > Delete        : Delete + Delete_Multi Count
 > Replace       : Replace Count
 > Qcache        : Query Cache Hits
 > Call          : Procedure Call Count
 > Slave         : Seconds_Behind_Master
 > user          : normal processes executing in user mode
 > system        : processes executing in kernel mode
 > iowait        : waiting for I/O to complete
 > Slow_queries : Slow Query Count
 > Sum           : Sum Slow Query Count

 > CopyRight     : programed By naver

-------------------------------------------------------------------------------------------------------------
Time     Conn Run Abort Select Update Insert Delete Replace Qcache Call   QPS Slave Slow  Sum User  Sys Wait Idle
-------------------------------------------------------------------------------------------------------------
17:16:33    2   2     0      0      4      1      0       0      0    0     5     0    0    0    1    0    0   99
17:16:34    2   2     0      0     63      1      0       0      0    0    64     0    0    0    0    0    0   99
17:16:35    2   2     0      0     59      1      0       0      0    0    60     0    0    0    1    1    0   99
17:16:36    2   2     0      0     55      1      0       0      0    0    56     0    0    0    1    1    0   99
17:16:37    3   2     0      0     55      1      0       0      0    0    56     0    0    0    1    0    0   99
17:16:38    2   2     0      0     55      1      0       0      0    0    56     0    0    0    0    0    0  100
17:16:39    2   2     0      0     54      1      0       0      0    0    55     0    0    0    1    0    0   99
17:16:40    2   2     0      0     56      1      0       0      0    0    57     0    0    0    0    0    0   99
17:16:41    2   2     0      0     57      1      0       0      0    0    58     0    0    0    1    1    0   99
17:16:42    3   2     0      0     55      1      0       0      0    0    56     0    0    0    1    0    0   99
17:16:43    2   2     0      0     57      1      0       0      0    0    58     0    0    0    0    0    0  100
17:16:44    2   2     0      0     53      1      0       0      0    0    54     0    0    0    4    1    0   95
```

모니터링 항목 설명

속성	설명
Port	sqlstatus_nCloud에서 접속하는 DB 포트정보.
Time	모니터링 시간.
Conn	현재 DB에 접속하는 Connection Count
Run	DB에 Running 중인 Thread Count 현재 DB에서 쿼리를 처리하는 Thread 수를 나타냅니다.
Abort	DB에 접속 실패하는 Count max_connection 수를 초과하거나, DB 계정테이블에 등록되지 않는 사용자가 접속할 때 수를 표시합니다.
Select	DB에서 처리하는 SELECT Count
Update	DB에서 처리하는 UPDATE Count

속성	설명
Insert	DB에서 처리하는 INSERT Count
Delete	DB에서 처리하는 DELETE Count
Replace	DB에서 처리하는 REPALCE Count
Qcache	Query Cache를 통해 처리하는 쿼리 Count
Call	Procedure/Function을 사용하여 처리하는 쿼리 Count
QPS	DB에서 처리된 쿼리의 Total Count
Slave	Replication 구성된 SlaveDB의 복제지연 시간 Replication이 아닐 경우 Default 0으로 표시
user	CPU normal processes executing in user mode
system	CPU processes executing in kernel mode
iowait	CPU waiting for I/O to complete
Slow_queries	long_query_time에 지정된 수보다 오래 실행되는 쿼리 Count
Sum	sqlstatus를 실행한 이후 발생된 SlowQuery Count Default Parmeter는 10초이며 아래명령어로 변경이 가능합니다.(단위 Sec) mysql> set global long_query_time=1

연관 정보 바로가기

아래 가이드에서 연관 정보를 확인할 수 있습니다.

MSSQL 서버 이미지 사용자 가이드 MySQL 설치형 서비스 소개

네이버 클라우드 플랫폼 MySQL 설치형 서비스의 개념과 기본 구조를 설명합니다. "MySQL"은 세계에서 가장 많이 쓰이는 오픈 소스 관계형 데이터베이스(RDBMS)인 MySQL 을 이용하고 있습니다.

MySQL

전세계에서 가장 인기 있고, 많이 사용하고 있는 오픈소스 기반의 관계형 데이터베이스 시스템(Relational Database Management System, 이하 RDBMS)입니다. 다른 RDMBS 와 마찬가지로 데이터베이스를 생성하고 테이블, 칼럼 단위의 구조를 사용하여 레코드 단위의 데이터를 조회/변경/추가/삭제합니다. 별도의 GUI 를 제공하지 않으며, 주로 CLI(Command Line Interface)를 사용하여 관리하고, GUI 기반의 관리를 위해 MySQL 에서 제공하는 Workbench 나 Third Party-Tool 을 사용합니다.

2008 년 Sun 이 MySQL AB 를 인수한 후 2010 년 Oracle 이 Sun 을 인수하면서 현재 MySQLEnterprise/Community Version 의 모든 권리는 Oracle 이 소유하고 있습니다.

네이버 클라우드에서 제공하는 MySQL 설치형 서비스에서는 MySQL Community Edition(GPL)을 제공하여 Basic Install 수준의 기 설치된 이미지를 지원합니다.

네이버 클라우드 플랫폼 MySQL 설치형 서비스 구성

설치형 서비스

네이버 클라우드에서 지원하는 MySQL 설치형 서비스의 경우 DB 설치에 대해 원활한 운영표준을 제시하는 환경파일 제공 및 DB 상태를 볼수 있는 CLI 형태의 모니터링 Binary 를 포함한 MySQL Daemon 에 Startup 상태까지 지원합니다. 이에 관련하여 **DB 설치 이외 부분에 대해서는 기술 지원하지 않고 있습니다**.

RPM/APT

MySQL 설치 방법에는 Source Compile 설치 방법, RPM/APT 설치 방법, Binary 설치 방법이 있으며, 네이버 클라우드에서는 Normal Install 방법인 RPM(APT) 설치 이미지를 제공합니다. RPM(APT)에서 사용하는 .repo 파일은 MySQL 에서 공식적으로 제공하는 파일을 사용하여 설치를 진행하며 Repo 에서 제공하는 Latest Version 을 기본으로 하여 서버 이미지에 포함하여 제공합니다.

RPM Install

CentOS 6.x/7.x with MySQL 에서 사용하는 RPM 설치되는 패키지는 아래와 같이 구성됩니다.

MySQL 설치 및 실행에 관련되어 OS Default 설정에서 추가로 설치되는 패키지입니다.

- mysql-server
- mysql-devel
- libmysqlclient-dev
- build-essential

APT Install

Ubuntu 14.X/16.X with MySQL 에서 사용하는 APT 로 설치되는 패키지는 아래와 같이 구성됩니다.

MySQL 설치 및 실행에 관련되어 OS Default 설정에서 추가로 설치되는 패키지입니다.

- mysql-server
- libmysqlclient-dev

Recommanded Parameters (5.6/5.7 공통)

네이버 클라우드 플랫폼내의 MySQL 설치형 이미지에서 표준화하여 Recommanded Parameter 는 아래와 같이 구성됩니다.

- innodb_file_per_table
- innodb_log_file_size = 512M
- innodb_log_files_in_group = 2
- default_storage_engine=InnoDB
- query_cache_type = ON
- query_cache_size = 0M
- sysdate-is-now

innodb_file_per_table

InnoDB 테이블 생성 시 InnoDB Global Tablespace 를 사용하면, 테이블 삭제 시 이미 늘어난 물리 크기는 줄어들지 않습니다. innodb_file_per_table 옵션을 사용하면 InnoDB 테이블 생성 시 테이블당 파일을 생성하고 테이블 삭제 시 해당 파일이 삭제되어 물리 공간을 확보할 수 있습니다.

Innodb_log_file_size/innodb_log_files_in_group

Heavy Traffic 이 들어오는 경우 CheckPoint 를 위한 병목 현상을 방지하기 위해 innodb_log_file_size 옵션을 늘려서 설정합니다. 해당 Variables 을 조정하기 위해서는 DB Restart 가 필요합니다.

default_storage_engine

Online-DDL 및 Transction 사용이 가능한 InnoDB 로 표준 지정합니다.

query_cache_type/query_cache_size

DB 기동 후 query_cache 를 사용자가 동적으로 ON/OFF 할 수 있도록 ON/0 으로 설정합니다. 이후 서비스 상황에 맞도록 조정합니다.

sysdate-is-now

ysdate 를 사용하는 경우 Replication 구성에서 Master-Slave 간의 데이터 처리 시간이 다를 경우 Time Type 데이터가 다를 수 있어 sysdate 의 경우 자동으로 now 를 replace 합니다.

MySQL 설치형 서비스 관리

MySQL 설치형 서비스에서 관련되는 관리 방법을 설명합니다.

MySQL 이미지 서버 생성

MySQL 이 설치된 인스턴스 서버를 생성하면 아래와 같이 네이버 클라우드에 표준화된 my.cnf 의 환경을 가진 MySQL Daemon 이 Startup 되어 생성된 후 Daemon 기동까지 완료된 후 제공됩니다.

my.cnf

my.cnf 의 Parameter 설명은 아래와 같습니다.

속성	설명
expire_logs_days = 7	log-bin으로 설정된 로그 크기가 커지는 것을 방지하기 위해 7일간 저장하도록 합니다.
innodb_file_per_table	InnoDB 테이블 생성 시 InnoDB Global Tablespace를 사용하면, 테이블 삭제 시 이미 늘어난 물리 크기는 줄어들지 않습니다. innodb_file_per_table 옵션을 사용하면 InnoDB 테이블 생성 시 테이블당 파일을 생성하고 테이블 삭제 시 해당 파일이 삭제되어 물리 공간을 확보할 수 있습니다.
innodb_log_file_size	Heavy Traffic이 들어오는 경우 CheckPoint로 인한 병목을 방지하기 위해 innodb_log_file_size 옵션을 늘립니다. Variables을 조정하기 위해서는 DB Restart가 필요함.
iinnodb_log_files_in_group	innodb_log_file의 개수를 지정합니다. Rotation을 위해 2개 이상을 권고합니다.
default_storage_engine	Online-DDL 및 Transction 사용이 가능한 InnoDB로 표준 지정합니다.
query_cache_type query_cache_size	DB 기동 후 query_cache를 사용자가 동적으로 ON/OFF할 수 있도록 ON/0으로 설정합니다. 이후 서비스 상황에 맞도록 조정합니다.
sysdate-is-now	sysdate를 사용하는 경우 Replication 구성에서 Master-Slave 간의 데이터 처리 시간이 다를 경우 Time Type 데이터가 다를 수 있어 sysdate의 경우 자동으로 now를 replace합니다.
log_timestamps =	MySQL 5.7에서 LogTime이 UTC으로 되어있어 SYSTEM

속성	설명
SYSTEM(5.7 only)	Local TIME으로 변경합니다.

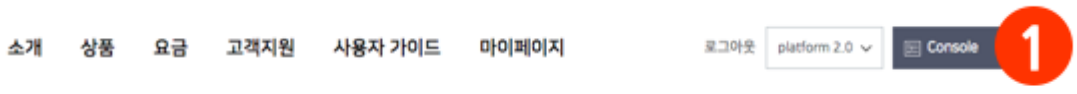
MySQL 설치된 디렉터리는 아래와 같습니다.

속성	설명
DATADIR	/var/lib/mysql/
SLOW_LOG	/var/lib/mysql/*slow.log
ERROR_LOG	/var/log/mysqld.log - CentOS /var/log/mysql/error.log - Ubuntu
my.cnf	/etc/my.cnf - CentOS /etc/mysql/my.cnf - Ubuntu.
sqlstatus_nCloud	/usr/bin/sqlstatus_nCloud

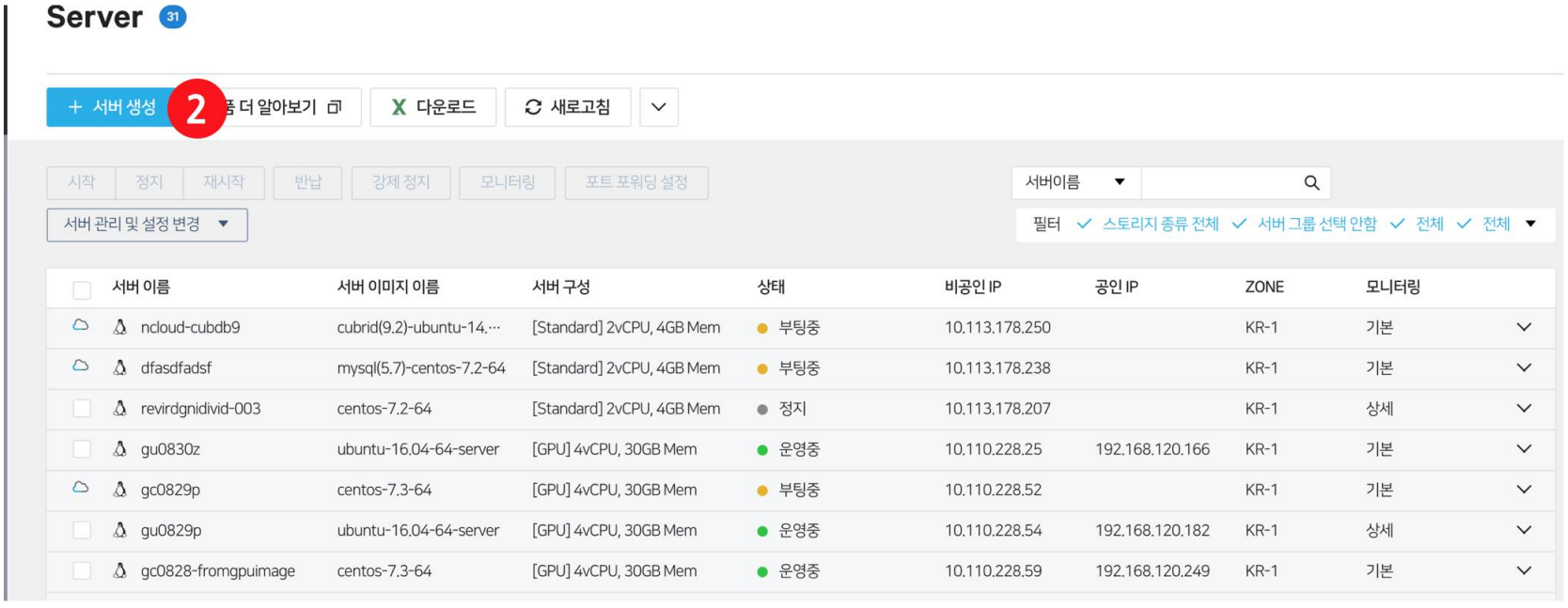
서버 생성 절차

MySQL 서버 생성 과정은 아래와 같습니다.

www.ncloud.com 접속 및 Login



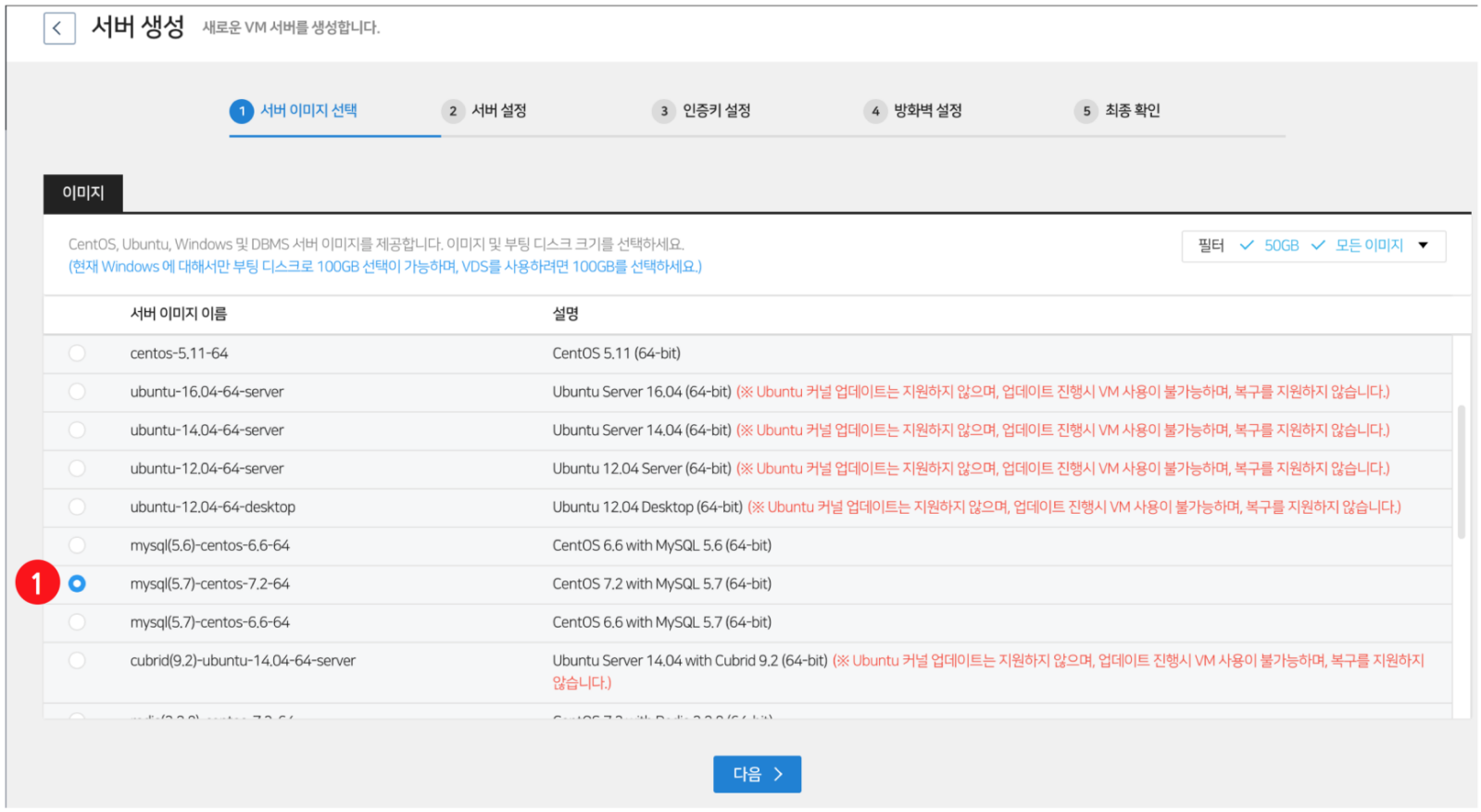
① 우측 상단 콘솔을 선택하여 콘솔에 접속합니다.



② 콘솔 화면에서 서버 생성 선택을 합니다.

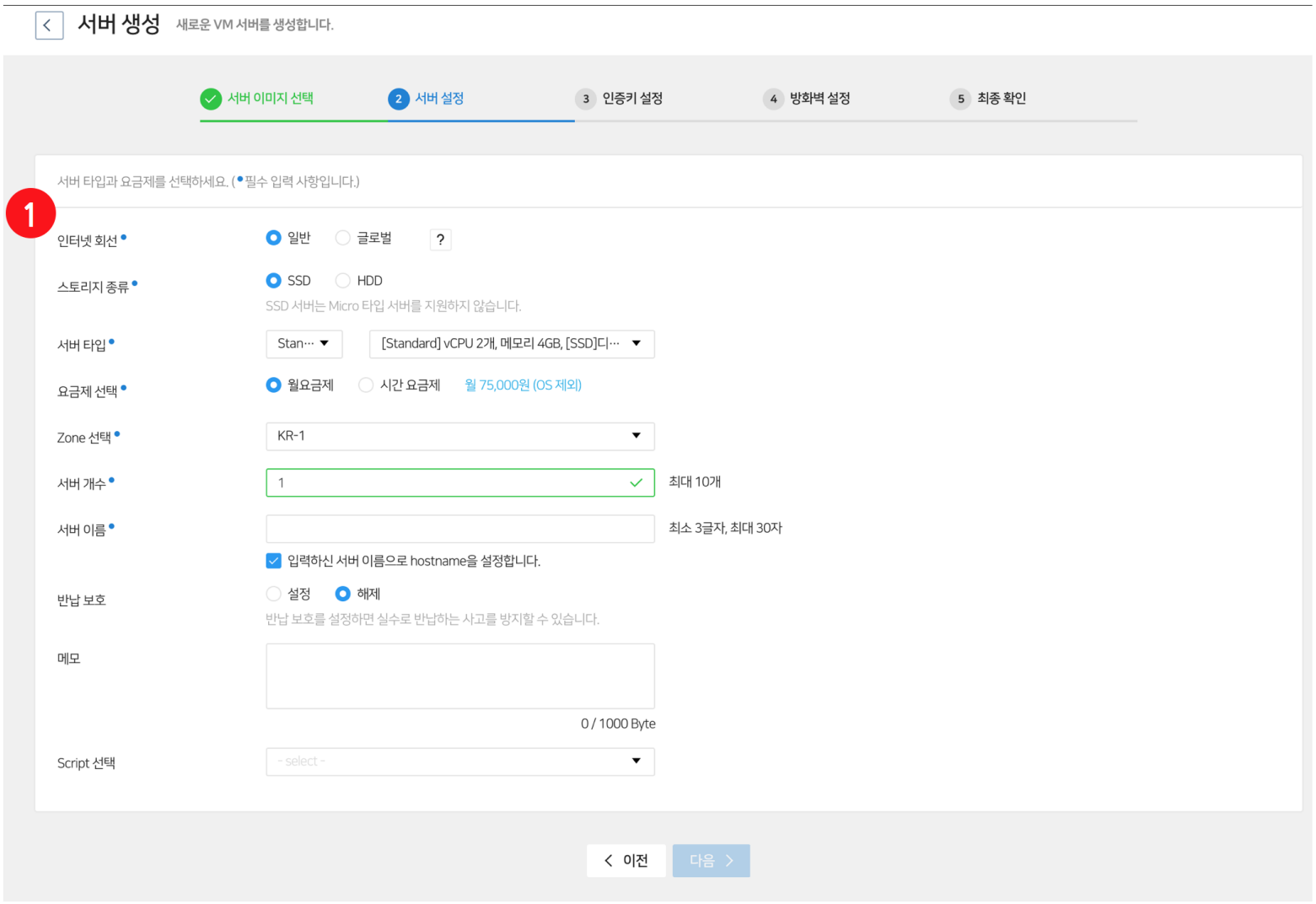
서버 생성

Step 1. 서버 이미지 선택



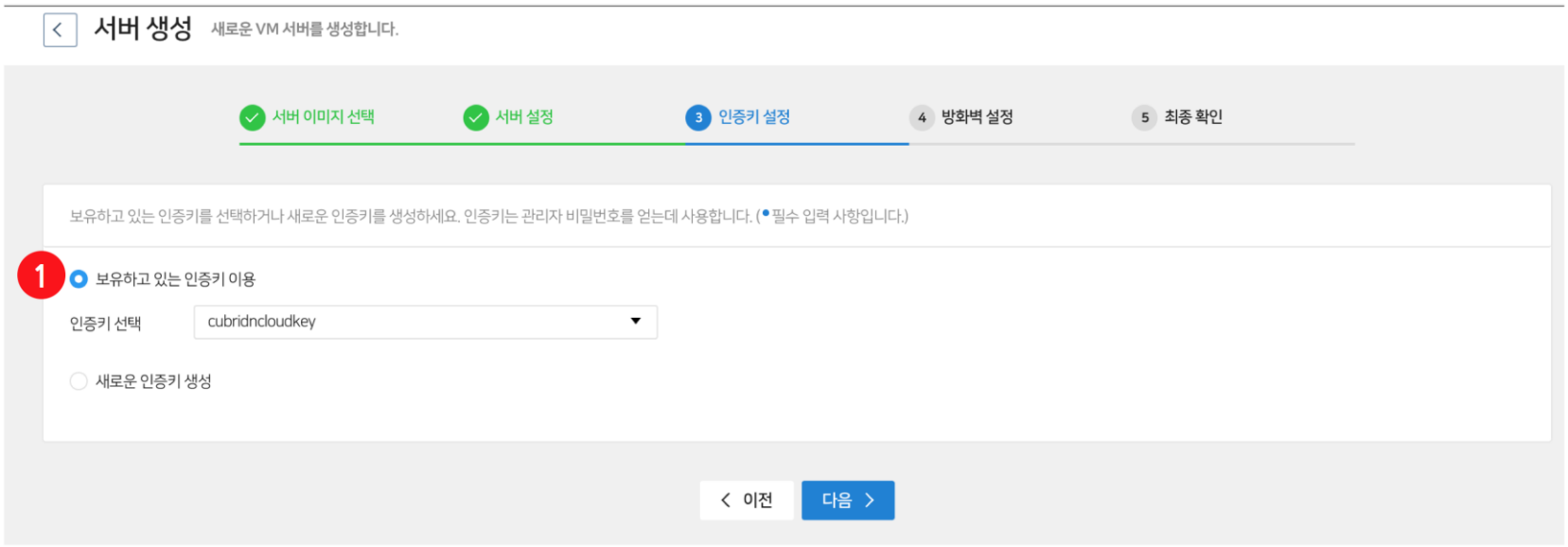
① OS Version 에 해당하는 MySQL Version 을 체크한 후 하단의 [다음] 버튼을 선택합니다.

Step 2. 서버 설정



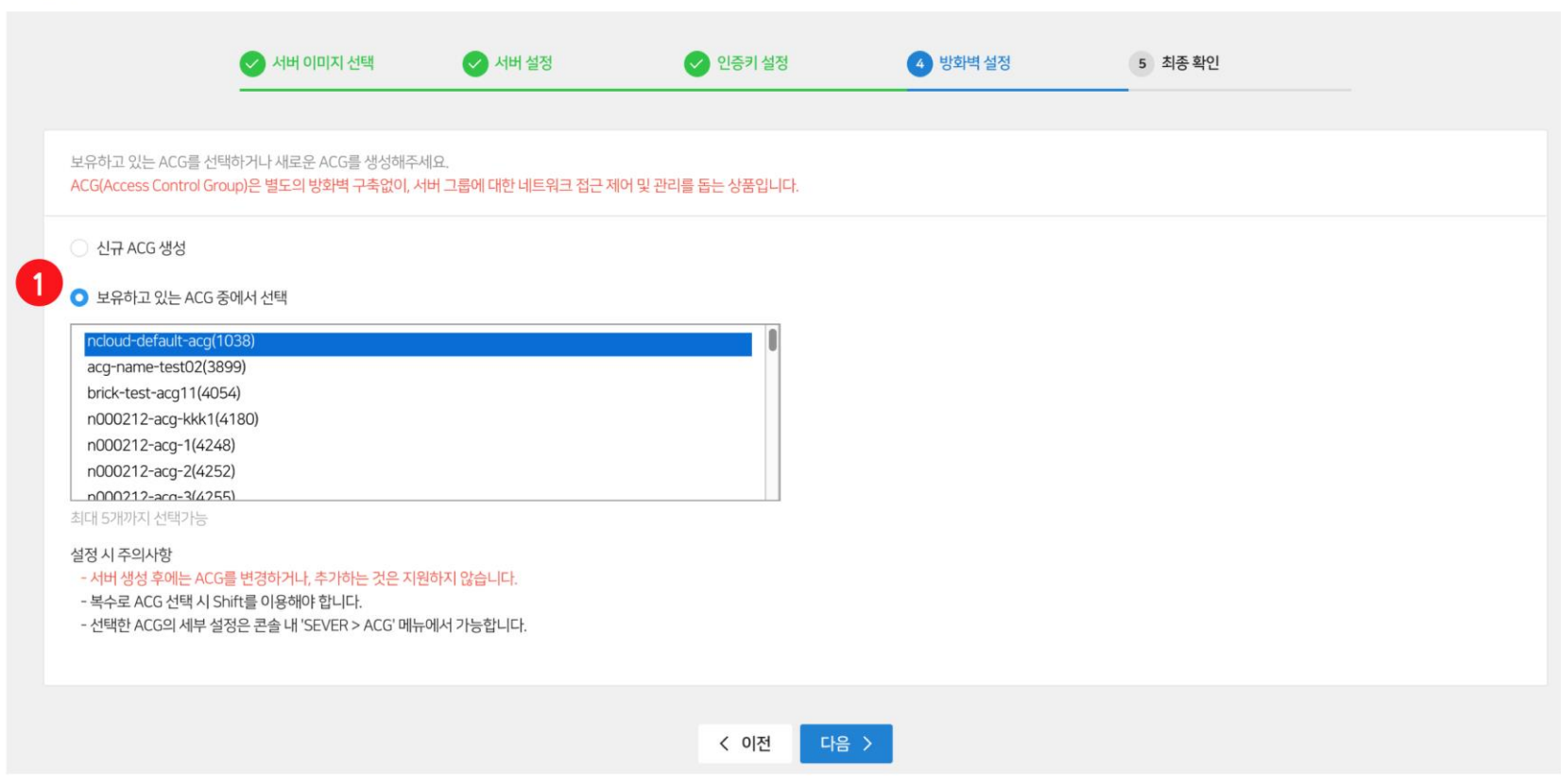
① 아래와 같이 서버타입과 원하는 요금제를 선택합니다.

Step 3. 인증키 설정



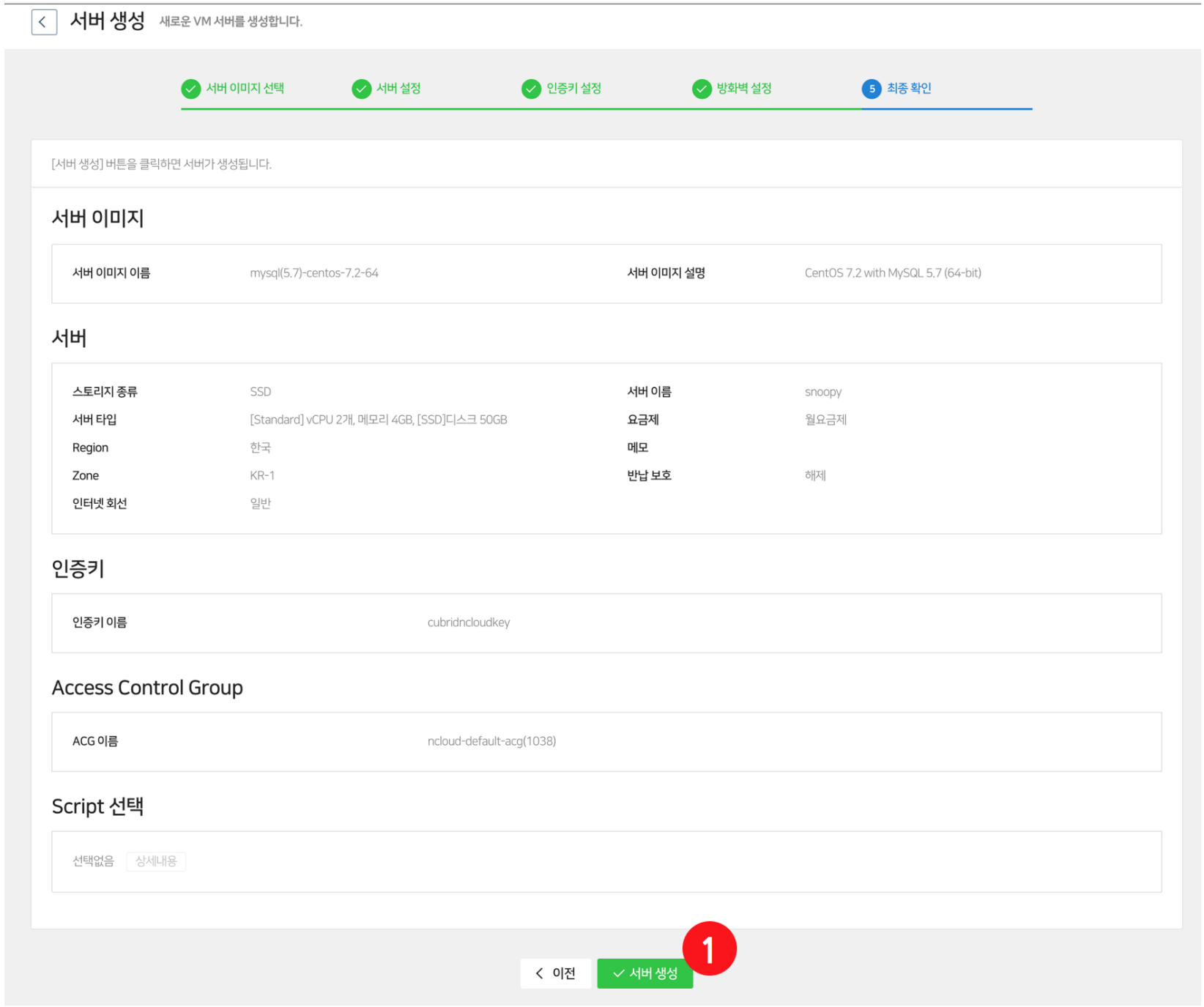
① 사용자의 인증키를 선택합니다.

Step 4. 방화벽 설정



① 보유하고 있는 ACG 를 선택하거나 신규로 생성합니다.

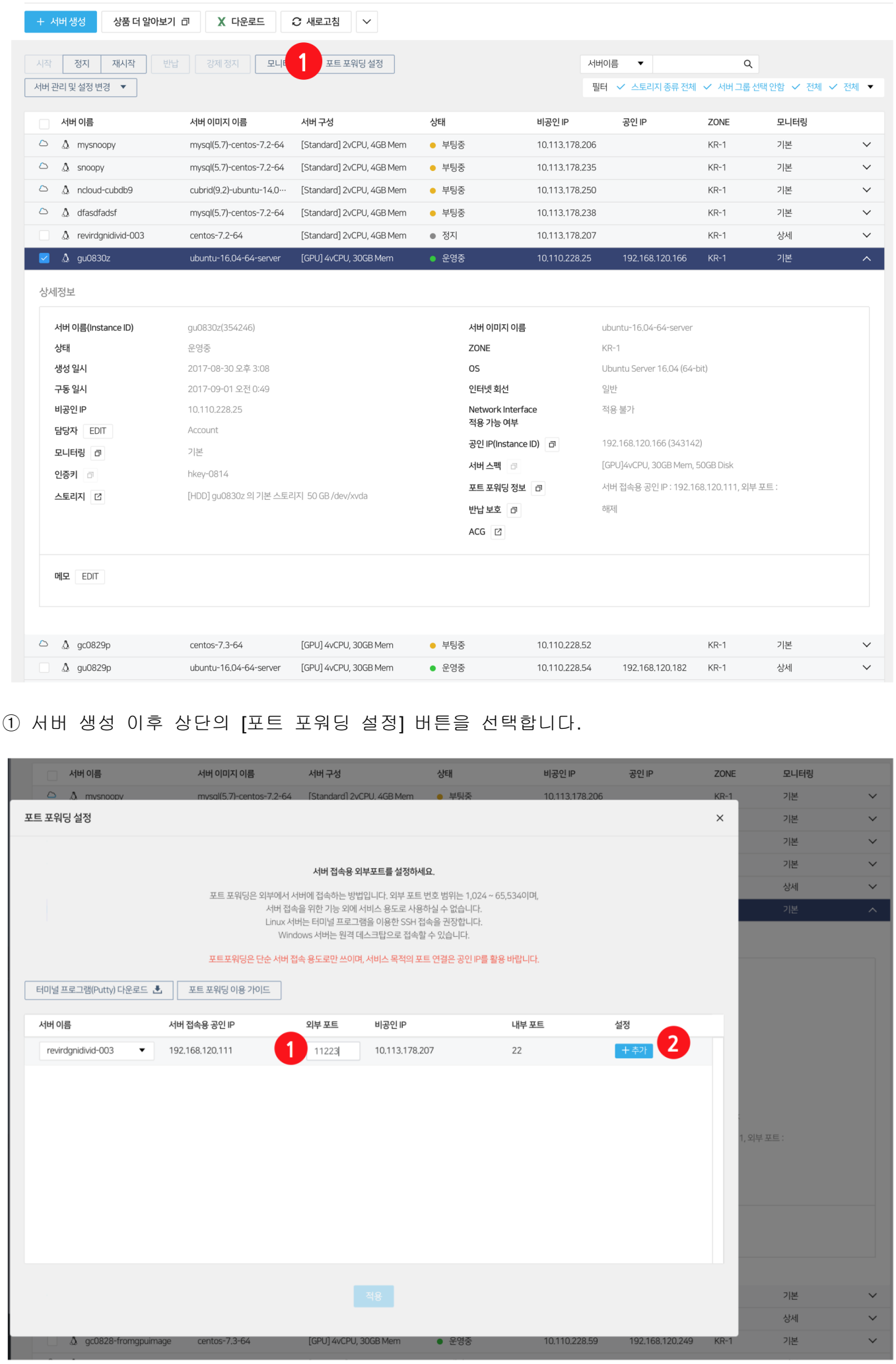
Step 5. 최종확인



① 선택한 내용를 확인한 후에 MySQL 이 설치된 서버를 생성합니다

서버접속 설정 및 확인

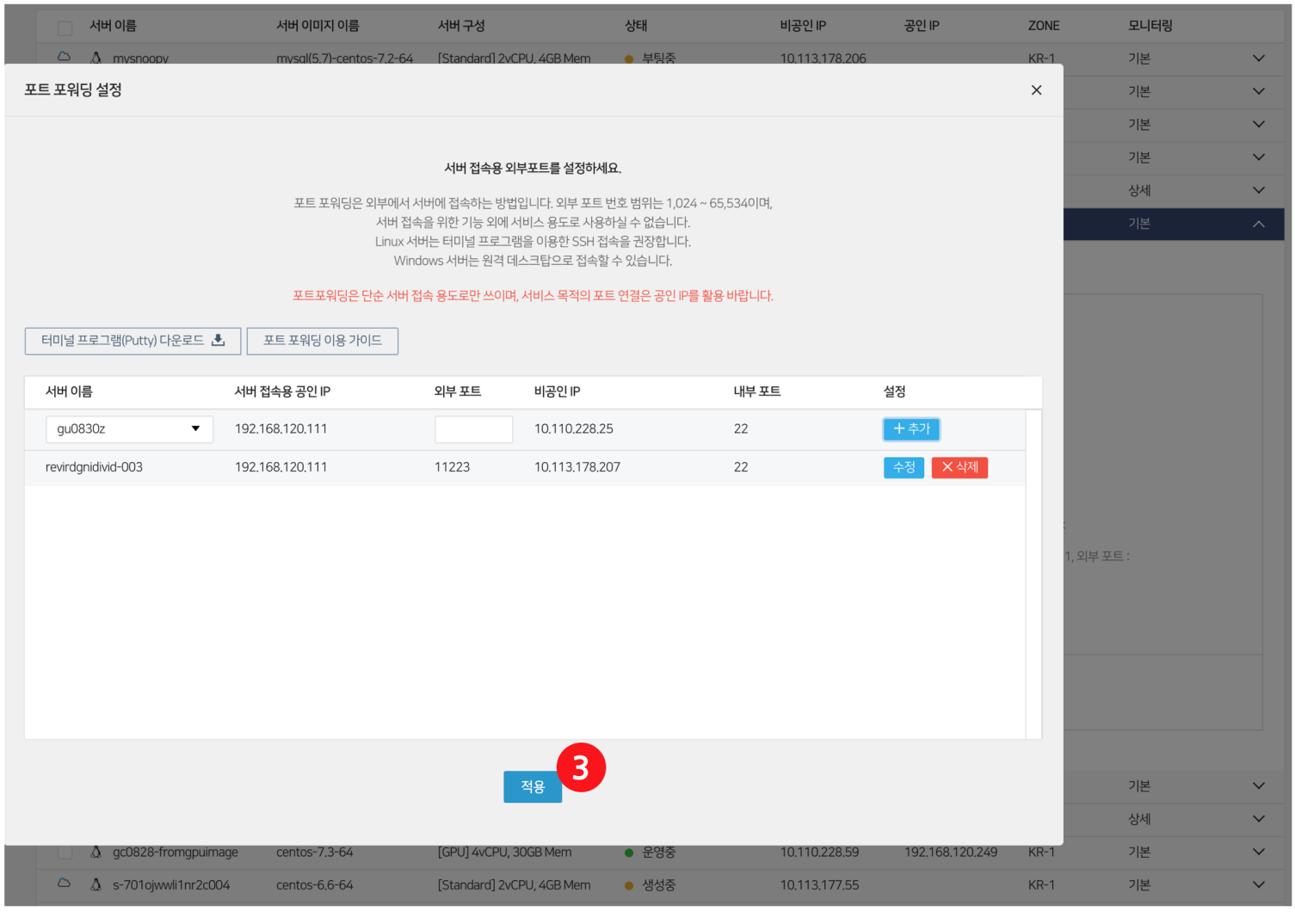
포트 포워딩 설정



① 서버 생성 이후 상단의 [포트 포워딩 설정] 버튼을 선택합니다.

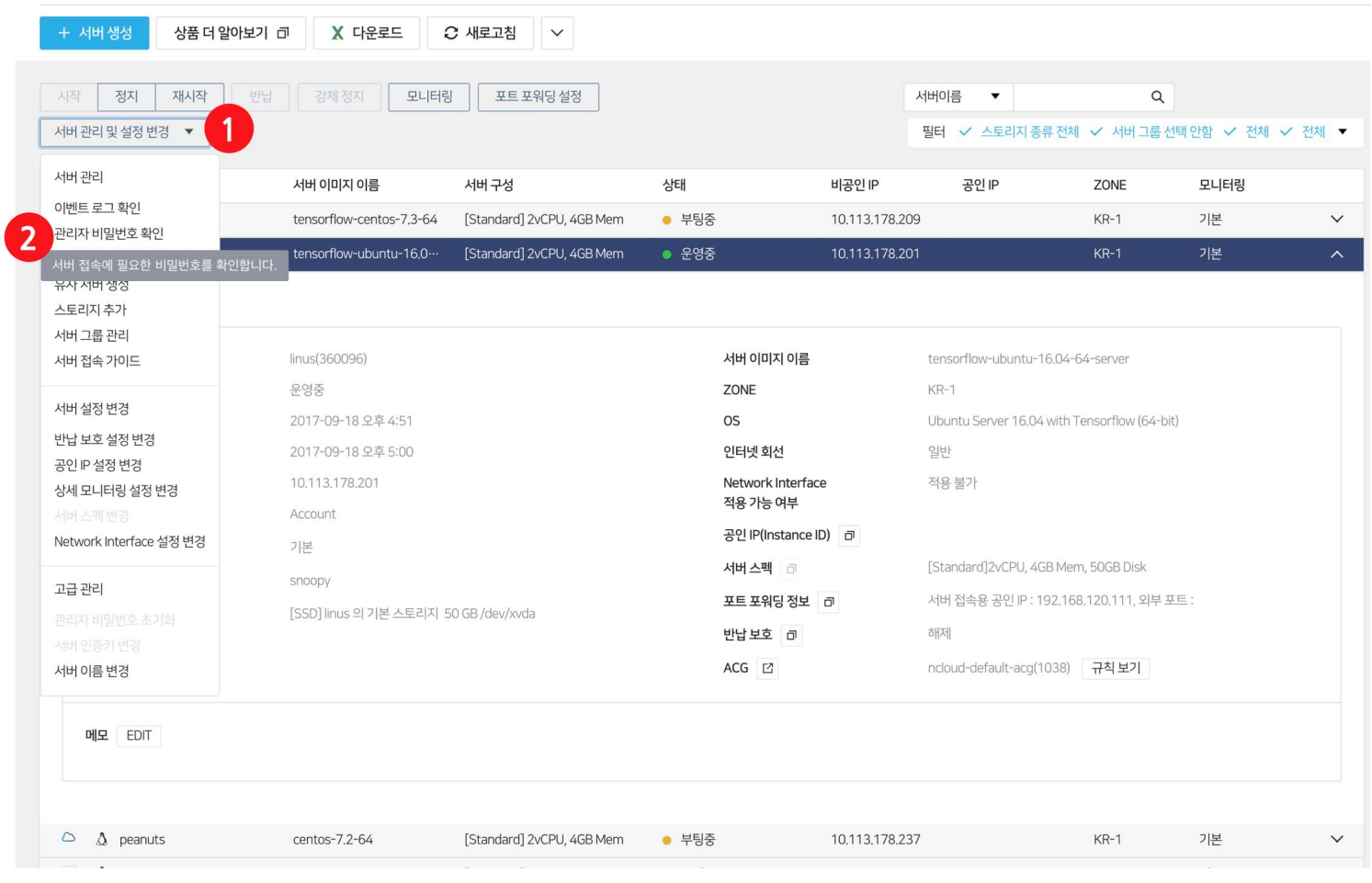
① 아래 화면의 포트 포워딩 설정에서 ssh 접속을 위한 외부포트를 등록합니다.

② 외부포트등록후 [추가]를 선택합니다.



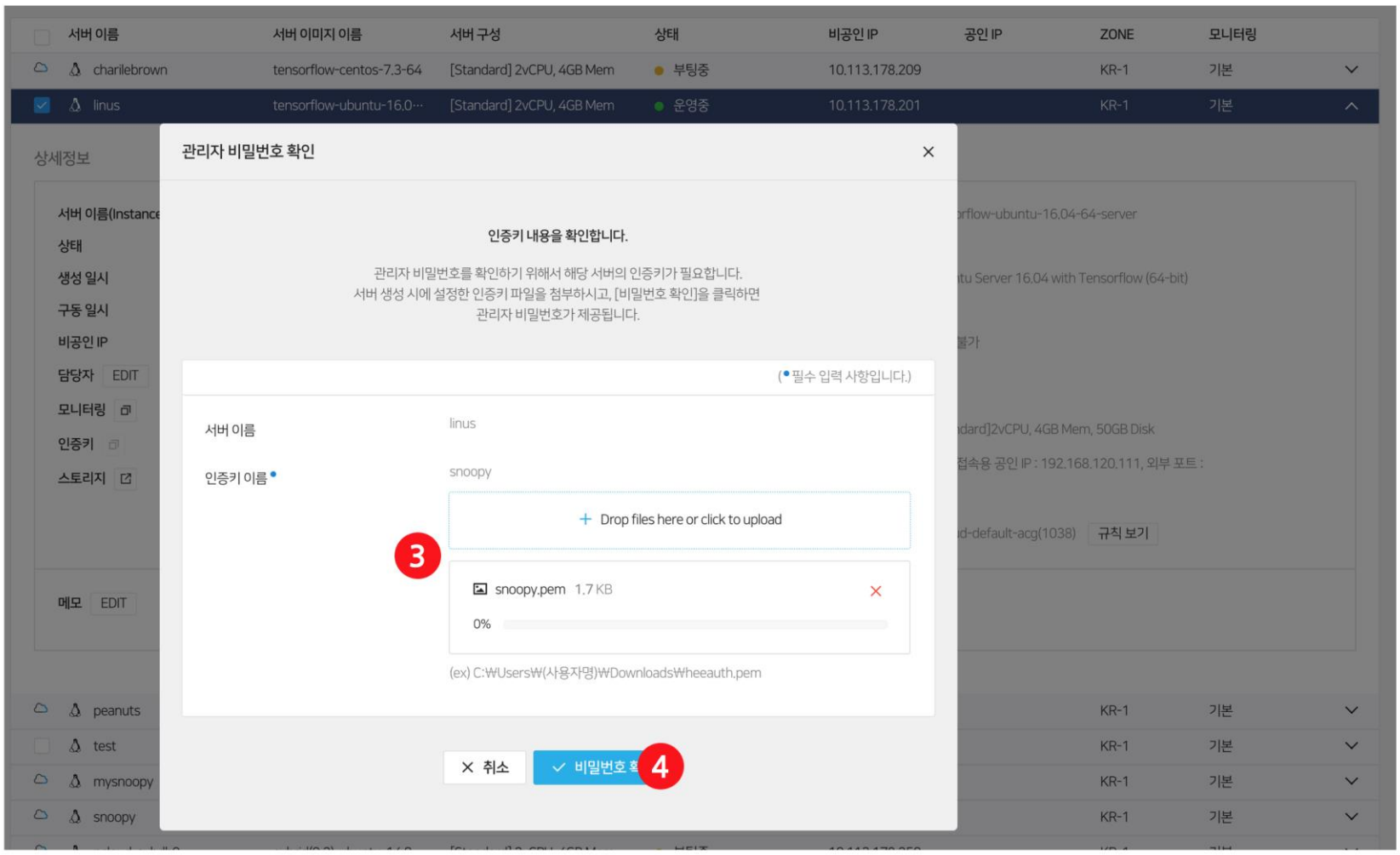
③ [적용]을 클릭합니다.

관리자 비밀번호 확인



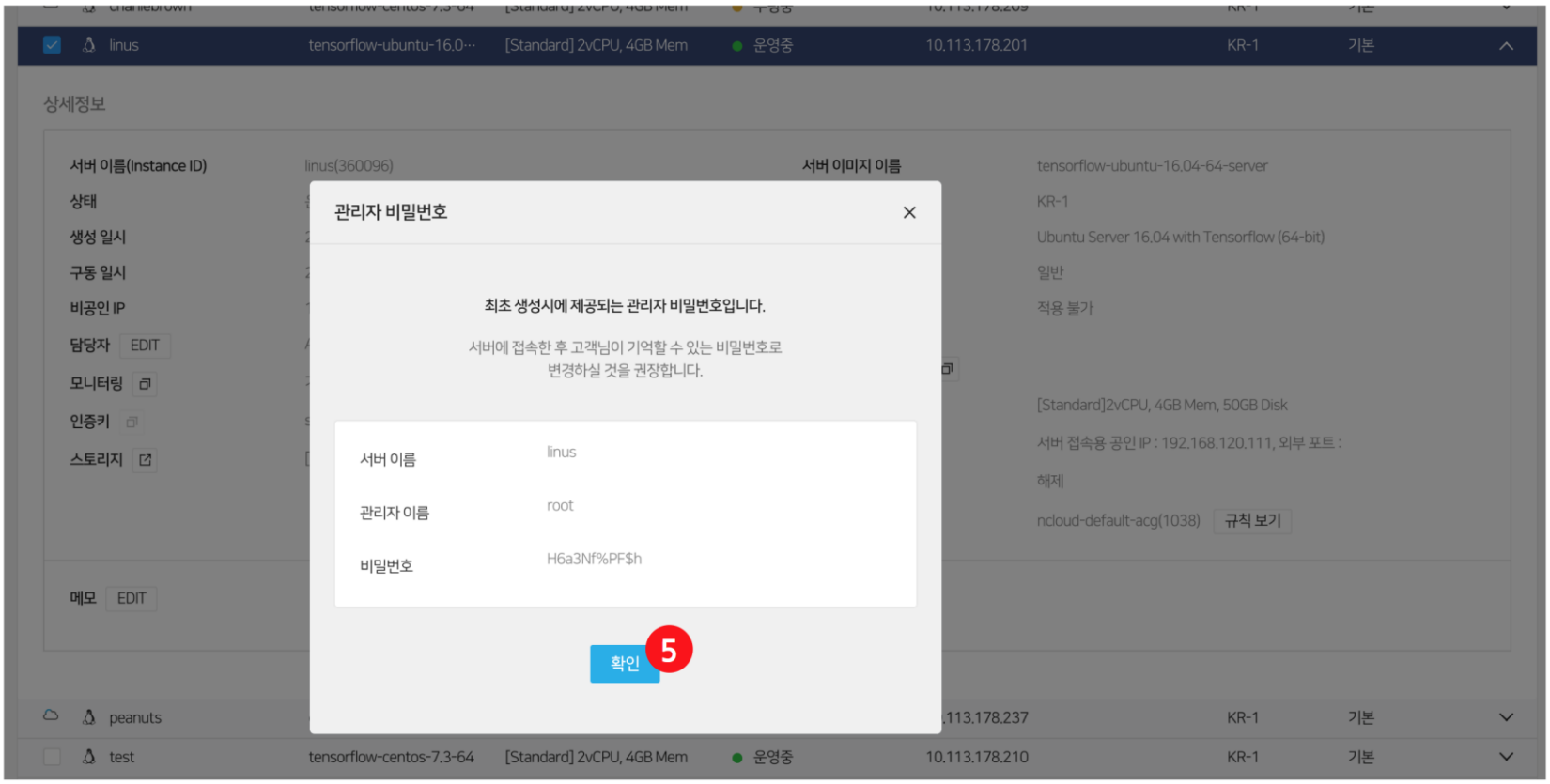
① 포트 포워딩 설정이후 상단의 [서버관리 및 설정변경] 버튼을 선택합니다.

② 이후 팝업 메뉴에 [관리자 비밀번호 확인]을 선택합니다.



③ 관리자 비밀번호 확인창에서 [파일선택]으로 인증키 파일을 선택합니다.

④ [비밀번호 확인] 버튼을 클릭합니다.



⑤ 이후 화면에 나온 비밀번호를 인지합니다.

터미널을 통한 서버접속

터미널프로그램을 사용하여 아래와 같이 접속합니다.

- PORT : 포트 포워딩으로 설정한 포트번호
- 공인 IP : 포트포워딩 화면에 안내된 서버 접속용 공인 IP
- 패스워드 : 관리자 비밀번호에 출력된 패스워드(초기 패스워드는 반드시 변경합니다.)
- `ssh -p [PORT] root@[공인 IP]`

서버 이미지 생성이후

1. 서버 생성 이후 바로 사용하지 않을 경우 보안 위험을 예방하기 위해 **MySQL Daemon 을 ShutDown** 합니다.
2. 생성된 MySQL 의 경우 패스워드가 설정되지 않았기 때문에 **반드시 패스워드 설정**을 합니다.

MySQL 기동/정지 및 접속 절차

- MySQL Daemon 기동및 정지는 아래와 같은 명령어로 실행합니다.

  - CentOS
  - `shell> /sbin/service mysqld start`
  - `shell> /sbin/service mysqld stop`
  - Ubuntu
  - `Shell> /usr/sbin/service mysql start`
  - `Shell> /usr/sbin/service mysql stop`
- MySQL Daemon 구동 유무는 아래와 같은 명령어로 확인합니다.

- `Shell> ps -ef | grep mysqld`
  - 결과: mysqld/mysqld 프로세스가 보여야 합니다.
- 네이버 클라우드에서 생성한 MySQL 은 아래 명령어로 접속합니다.

- `Shell> mysql -uroot`
- MySQL Password 정책에 따라 초기 패스워드는 지정되지 않습니다.

- root Password 의 경우 설정하지 않으므로 보안 침해 방지를 위해 **최초 접속 시 반드시 패스워드 변경을 합니다.**

- root 패스워드는 아래 명령어를 사용하여 변경합니다.

  - MySQL 5.6
  - `mysql> set password=password('패스워드');`
  - MySQL 5.7
  - `mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '패스워드';`
- MySQL Built-in Password 기본 정책은 아래와 같습니다.
  - 최소 길이 8 자 이상
  - 특수문자 1 개 이상
  - 숫자 1 개 이상
  - 대소문자 조합 1 개 이상

MySQL 이미지 서버 반납

네이버 클라우드에서 MySQL 이 생성된 서버 반납 시 아래와 같은 절차 이후 반납하기를 권장합니다.

1. MySQL 에서 저장한 데이터 백업

   - 백업된 데이터의 경우 Download 이후 별도 보관
2. MySQL Daemon 정지
3. mysql 이 저장된 Data Directory 삭제

   - 예: rm -rf /var/lib/mysql
4. 네이버클라우드 콘솔에서 [정지]->[반납] 순서로 선택하여 반납을 진행합니다.

MySQL 외부접속 허용

Ubuntu 에서 MySQL 을 사용하는 경우 Default Parameter 로 인해 Localhost 에서만 접속이 가능합니다.
이에 외부접속이 가능하도록 아래와 같은 예로 가이드합니다.

1. MySQL DB 접속
2. 계정 권한 부여
   예: GRANT ALL PRIVILEGES ON *.* to '계정명'@'%' IDENTIFIED BY '비밀번호';
3. 환경파일 수정
   예: sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
4. bind-address 주석처리

```
# By default we only accept connections from localhost
# bind-address  = 127.0.0.1
```

5. MySQL 재시작

SQLSTATUS

네이버 클라우드에서 제공하는 MySQL 서비스에는 CLI(Command Line Interface) 형태의 모니터링 프로그램이 제공됩니다.

- 해당 프로그램은 **nCloud 서버 이외에 외부 유출 및 임의로 변경이나 변조하는 경우를 금지**하고 있습니다.

개요

MySQL 의 내부 Status 를 1 초 단위로 볼 수 있도록 Binary 형태로 제공합니다.
1 초간의 MySQL Status 를 Display 를 하여 Real-Time 으로 DB 의 상태 변화를 볼 수 있습니다.

사용법

sqlstatus_nCloud 에서 사용할 계정 생성(Default 계정으로 포함)

```
 GRANT PROCESS, SHOW DATABASES, SUPER, EXECUTE, REPLICATION CLIENT ON *.* TO
'sqlstatus_ncloud'@'127.0.0.1' IDENTIFIED BY PASSWORD '*AB9C7542BD3FE96F5342F2A616D030CF4494EB44';
```

설치 경로 : /usr/bin/sqlstatus_nCloud

기본 3306 포트 사용 시 : sqlstatus_nCloud

3306 이외 사용자 Port 사용 시 : sqlstatus_nCloud 포트번호

사용하지 않을 경우 Default 계정을 삭제해도 됩니다.(`DROP USER sqlstatus\_ncloud@127.0.0.1;`)

사용자 화면

```
 - Interval Time : 1 Sec
 - Port         : 3306
 > Time         : Current Time
 > Conn         : Thread Connection
 > Run          : Thread Running
 > Abort        : Aborted Connection
 > Select       : Select Count
 > Update       : Update + Update_Multi Count
 > Insert       : Insert + Insert_Select Count
 > Delete       : Delete + Delete_Multi Count
 > Replace      : Replace Count
 > Qcache       : Query Cache Hits
 > Call         : Procedure Call Count
 > Slave        : Seconds_Behind_Master
 > user         : normal processes executing in user mode
 > system       : processes executing in kernel mode
 > iowait       : waiting for I/O to complete
 > Slow_queries : Slow Query Count
 > Sum          : Sum Slow Query Count

 > CopyRight    : programed By naver

-------------------------------------------------------------------------------------------------------------
Time     Conn Run Abort Select Update Insert Delete Replace Qcache Call   QPS Slave Slow  Sum User  Sys Wait Idle
-------------------------------------------------------------------------------------------------------------
17:16:33    2   2     0      0      4      1      0      0      0    0     5     0    0    0    1    0    0   99
17:16:34    2   2     0      0     63      1      0      0      0    0    64     0    0    0    0    0    0   99
17:16:35    2   2     0      0     59      1      0      0      0    0    60     0    0    0    1    1    0   99
17:16:36    2   2     0      0     55      1      0      0      0    0    56     0    0    0    1    1    0   99
17:16:37    3   2     0      0     55      1      0      0      0    0    56     0    0    0    1    0    0   99
17:16:38    2   2     0      0     55      1      0      0      0    0    56     0    0    0    0    0    0  100
17:16:39    2   2     0      0     54      1      0      0      0    0    55     0    0    0    1    0    0   99
17:16:40    2   2     0      0     56      1      0      0      0    0    57     0    0    0    0    0    0   99
17:16:41    2   2     0      0     57      1      0      0      0    0    58     0    0    0    1    1    0   99
17:16:42    3   2     0      0     55      1      0      0      0    0    56     0    0    0    1    0    0   99
17:16:43    2   2     0      0     57      1      0      0      0    0    58     0    0    0    0    0    0  100
17:16:44    2   2     0      0     53      1      0      0      0    0    54     0    0    0    4    1    0   95
```

모니터링 항목 설명

속성	설명
Port	sqlstatus_nCloud에서 접속하는 DB 포트정보.
Time	모니터링 시간.
Conn	현재 DB에 접속하는 Connection Count
Run	DB에 Running 중인 Thread Count 현재 DB에서 쿼리를 처리하는 Thread 수를 나타냅니다.
Abort	DB에 접속 실패하는 Count max_connection 수를 초과하거나, DB 계정테이블에 등록되지 않는 사용자가 접속할 때 수를 표시합니다.
Select	DB에서 처리하는 SELECT Count
Update	DB에서 처리하는 UPDATE Count
Insert	DB에서 처리하는 INSERT Count
Delete	DB에서 처리하는 DELETE Count
Replace	DB에서 처리하는 REPALCE Count
Qcache	Query Cache를 통해 처리하는 쿼리 Count
Call	Procedure/Function을 사용하여 처리하는 쿼리 Count
QPS	DB에서 처리된 쿼리의 Total Count
Slave	Replication 구성된 SlaveDB의 복제지연 시간 Replication이 아닐 경우 Default 0으로 표시
user	CPU normal processes executing in user mode
system	CPU processes executing in kernel mode
iowait	CPU waiting for I/O to complete
Slow_queries	long_query_time에 지정된 수보다 오래 실행되는 쿼리 Count
Sum	sqlstatus를 실행한 이후 발생된 SlowQuery Count Default Parmeter는 10초이며 아래명령어로 변경이 가능합니다.(단위 Sec)

속성	설명
	mysql> set global long_query_time=1

연관 정보 바로가기

아래 가이드에서 연관 정보를 확인할 수 있습니다.

- MSSQL 서버 이미지 사용자 가이드
-