

Lab3. Tencent for MySQL 다루기

1. 목적

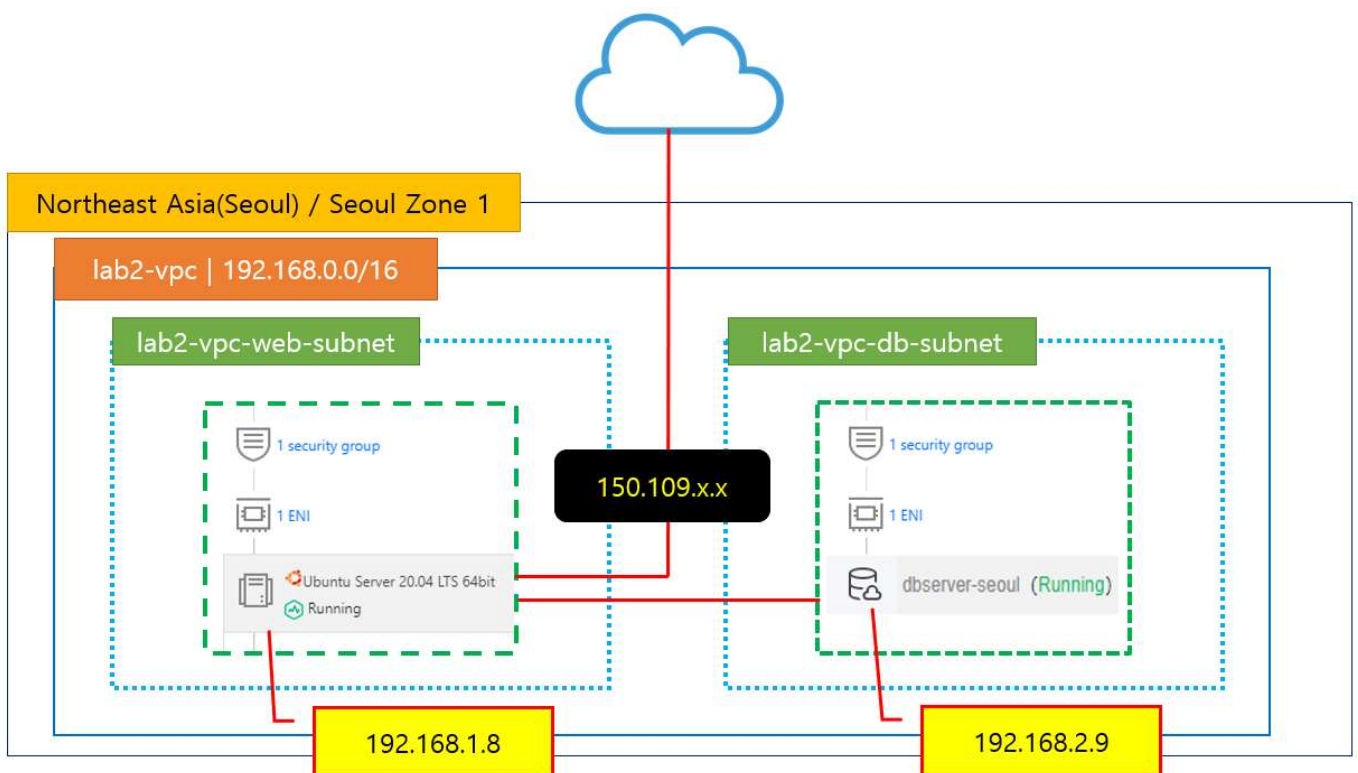
- 이번 Lab에서는 Tencent Cloud에서 제공하는 Database Service 중 Tencent for MySQL을 설치하고 이용하는 방법을 다룬다. MySQL Database Server는 새로운 Subnet 생성 후 설치한다. 설치 후 Web Server의 Private Network를 이용해서 연결하는 방법을 다룬다.

2. 사전 준비물

- Tencent Cloud Account

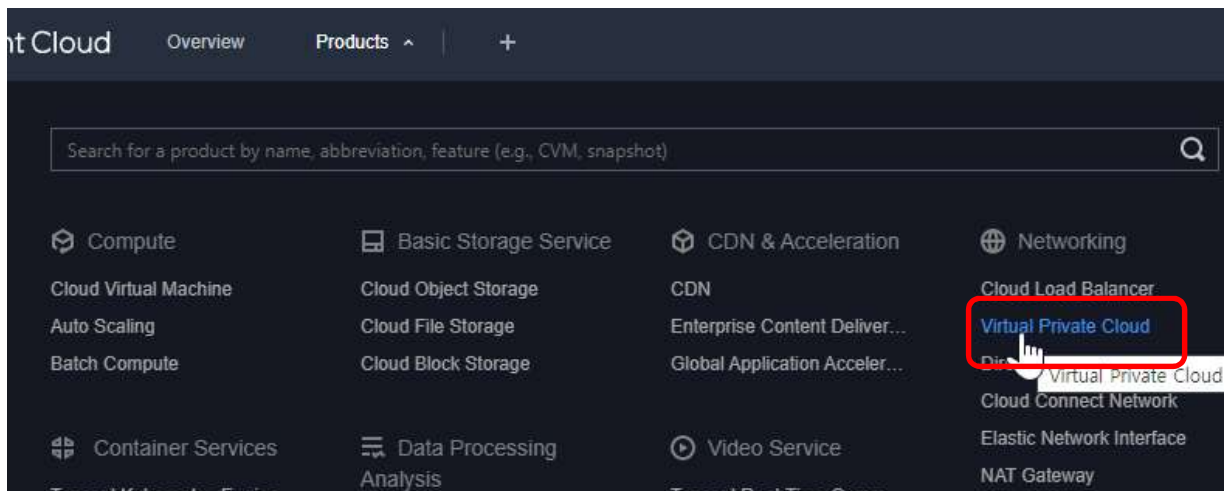
3. 목차

- Task1. Lab2에서 생성한 VPC(Virtual Private Cloud)에 새 Subnet 생성하기
- Task2. 새 Subnet에 Tencent for MySQL 설치하기
- Task3. MySQL Database Server에 직접 연결하기
- Task4. Web Server에서 Private Network으로 Database Server에 연결하기

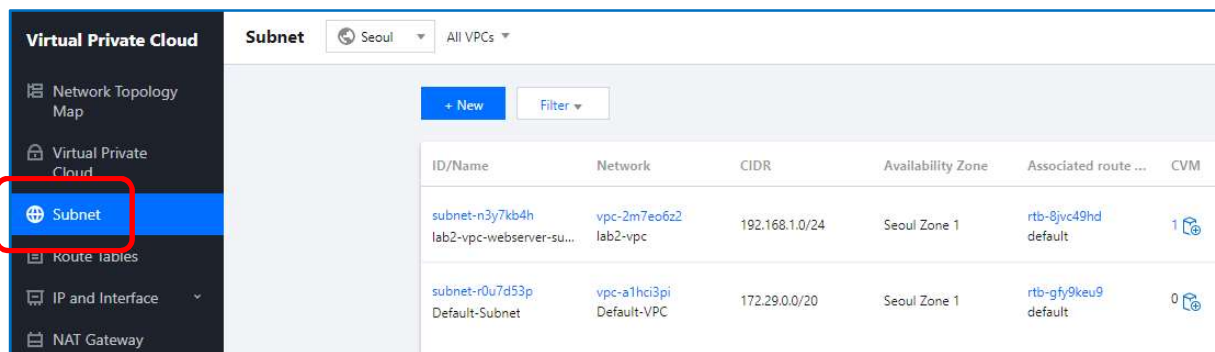


Task1. Lab2에서 생성한 VPC(Virtual Private Cloud)에 새 Subnet 생성하기

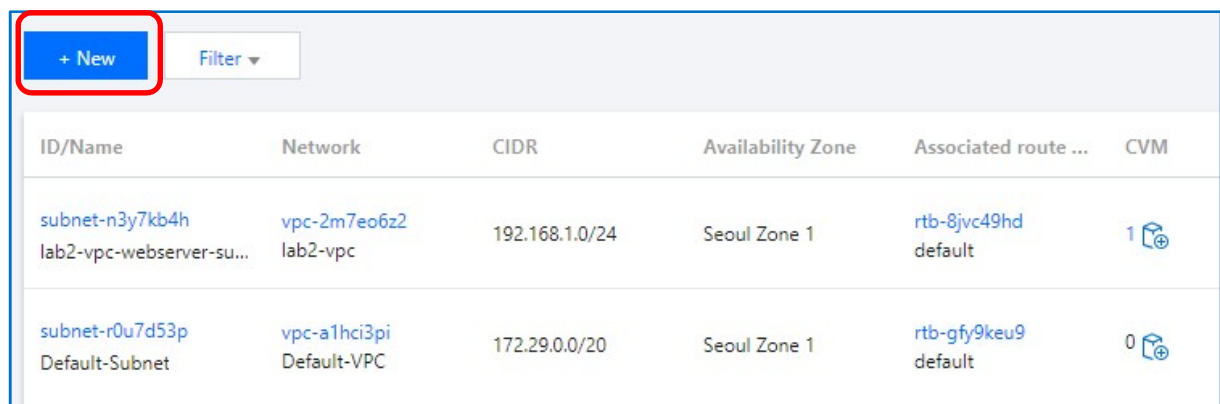
1. Subnet으로 이동하기 위해 상단 메뉴 [Products] > [Networking] > [Virtual Private Cloud]를 클릭한다.



2. [Virtual Private Cloud] 페이지로 이동했다. 좌측 메뉴에서 [Subnet]을 클릭한다.



3. Default-Subnet도 보이고, Lab2에서 생성한 lab2-vpc-webserver-subnet도 보인다. 새 Subnet을 생성하기 위해 [+New] 파란색 버튼을 클릭한다.



4. **[Create a Subnet]** 페이지가 나타난다. 먼저 Network에서 Lab2에서 생성한 VPC임을 확인한다. 그 다음 **[Subnet Name]**에 **lab2-vpc-dbserver-subnet**이라고 넣고, **[CIDR]**은 **192.168.2.0/24**를, **[Availability Zone]**은 **Seoul Zone 1**으로 맞추고 **[Create]** 파란색 버튼을 클릭하여 새로운 Subnet을 생성한다.

Create a Subnet

Network

vpc-2m7eo6z2(lab2-vpc | 192.1 ▼

1 existing subnets

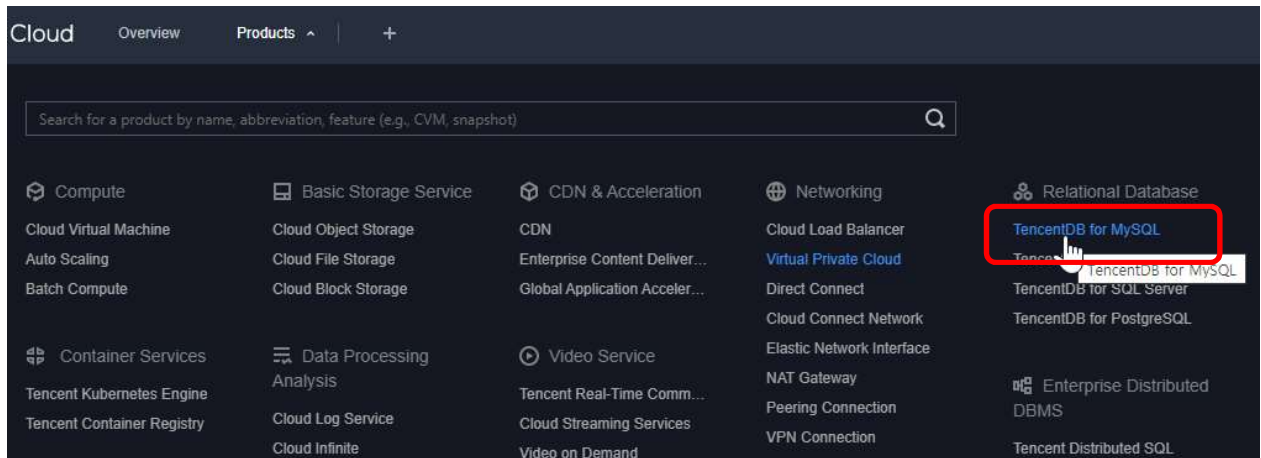
Subnet Name	VPC IP Range	CIDR ⓘ	Availability Zone ⓘ	Associated route table ⓘ	
lab2-vpc-dbserver-subnet	24/60	192.168.0.0/16 ▼	192.168.2.0/24 ▼	Seoul Zone 1 ▼	default ▼

5. **lab2-vpc-dbserver-subnet**이 만들어졌다. 아직 연결된 CVM이 없는 것을 확인할 수 있다.

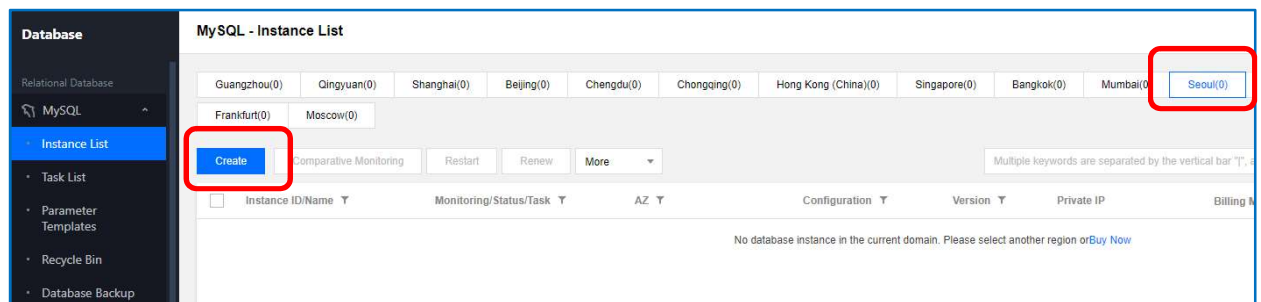
ID/Name	Network	CIDR	Availability Zone	Associated route ...	CVM
subnet-1maufq9f lab2-vpc-dbserver-sub...	vpc-2m7eo6z2 lab2-vpc	192.168.2.0/24	Seoul Zone 1	rtb-8jvc49hd default	0
subnet-n3y7kb4h lab2-vpc-webserver-su...	vpc-2m7eo6z2 lab2-vpc	192.168.1.0/24	Seoul Zone 1	rtb-8jvc49hd default	1
subnet-r0u7d53p Default-Subnet	vpc-a1hci3pi Default-VPC	172.29.0.0/20	Seoul Zone 1	rtb-gfy9keu9 default	0

Task2. 새 Subnet에 Tencent for MySQL 설치하기

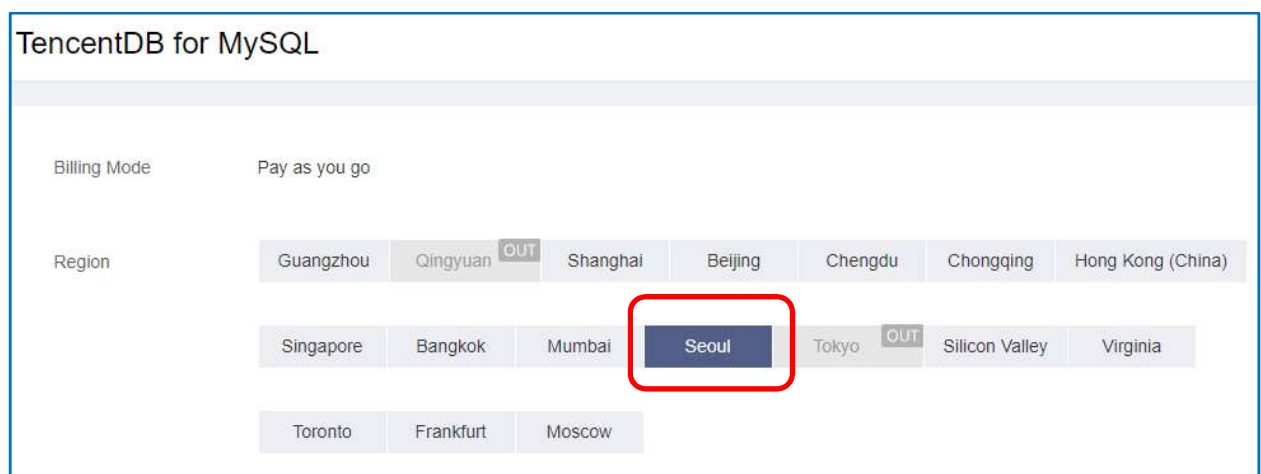
1. [Tencent for MySQL]을 생성하기 위해 상단 메뉴 [Products] > [Relational Database] > [Tencent for MySQL] 을 클릭한다.



2. [MySQL – Instance List] 페이지로 들어왔다. MySQL을 생성할 리전은 [Seoul]이다. MySQL Instance를 생성하기 위해 [Create] 파란색 버튼을 클릭한다.



3. [TencentDB for MySQL] 생성 페이지이다. [Billing Mode]는 기본적으로 Pay as you go이다. 설치 [Region]은 Seoul이다.



4. [Version]은 [MySQL5.7]로, [Architecture]는 [Two-node]로, [Source AZ]는 [Seoul Zone 1]에 원본 서버를 놓고, [Replica AZ]을 [Seoul Zone 2]에 맞춤으로 복제서버는 **Seoul Zone 2**에 놓기로 한다.

The screenshot shows the configuration page for a MySQL instance. A red rounded rectangle highlights the following settings:

- Version:** MySQL5.7 (selected)
- Architecture:** Two-node (selected)
- Source AZ:** Seoul Zone 1 (selected)
- Replica AZ:** Seoul Zone 2 (selected)

Below the highlighted settings, there is a note: "Products in the same region but different availability zones can communicate each other via private network. For example, under the same VPC, the CVM in Guangzhou Zone 2 can access the MySQL instance in Guangzhou Zone 3 via private network."

5. [Instance Specification]은 목록에서 제일 스펙이 작은 [1core1000MB]를 선택하고, [Hard Disk]는 기본값 그대로 200GB를, [Data Replication Mode]도 기본값 [Async]를 선택한다.

The screenshot shows the configuration page for the instance specification. A red rounded rectangle highlights the following settings:

- Instance Specification:** 1core1000MB (selected)
- Hard Disk:** 200 GB (Increment: 5GB) (selected)
- Data Replication Mode:** Async (selected)

Below the highlighted settings, there is a note: "Local SSD, featuring robust performance and dynamically changing IOPS capabilities for different instance specifications."

6. MySQL Database Server가 위치할 VPC와 Subnet 설정이다. 이 실습의 목적에 맞게 VPC는 lab2-vpc로, Subnet은 앞 Task1에서 생성한 lab2-vpc-dbserver-subnet을 선택한다. [Security Group]은 기존의 Security Group과 다르기 때문에 [Create Security Group] 링크를 클릭하여 새 Security Group을 생성한다.

The screenshot shows the configuration page for the network and security group. A red rounded rectangle highlights the following settings:

- Network:** lab2-vpc (selected) and lab2-vpc-dbserver-subnet (selected)

Below the highlighted settings, there is a note: "To change the network, please go to the console to Create VPC or Create Subnet. After the TencentDB instance is purchased, the private network/subnet can be switched in the console."

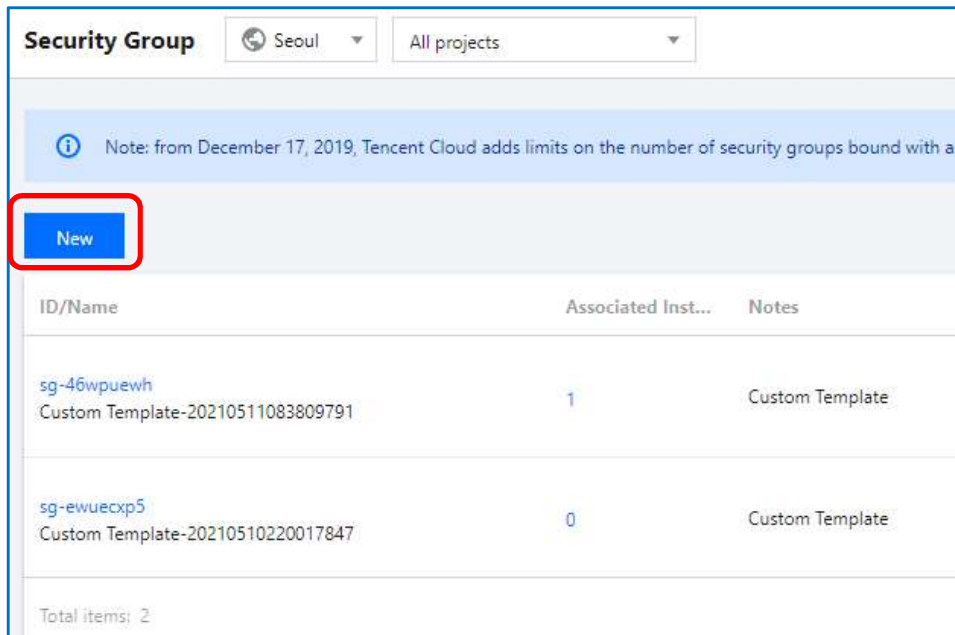
Below the network settings, there is a note: "In the current network environment, only devices in the 'lab2-vpcVPC' can access this database instance."

Below the network settings, there is a section for Security Group:

- Security Group:** Custom Template-20210511083809791 (selected)

Below the security group settings, there is a note: "To open other ports, you can Create Security Group."

7. [Security Group] 페이지로 들어왔다. 새 **Security Group**을 생성하기 위해 [New] 파란색 버튼을 클릭한다.



8. [New security group]창이다. 다음 그림과 같이 [Template]는 **Custom**으로, [Name]은 **lab3-dbserver-sg**라고 명명하고 나머지 값은 기본값 그대로 놓고, [OK] 파란색 버튼을 클릭하여 생성한다.

The screenshot shows the 'New security group' dialog box. The 'Template' dropdown is set to 'Custom', the 'Name' field is 'lab3-dbserver-sg', and the 'OK' button is highlighted with a red box.

Template: Custom

Name: lab3-dbserver-sg

Project: DEFAULT PROJECT

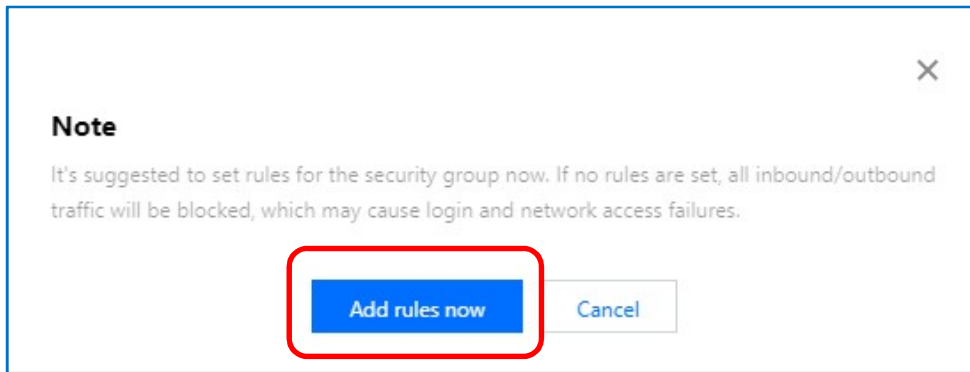
Notes: Custom

Advanced

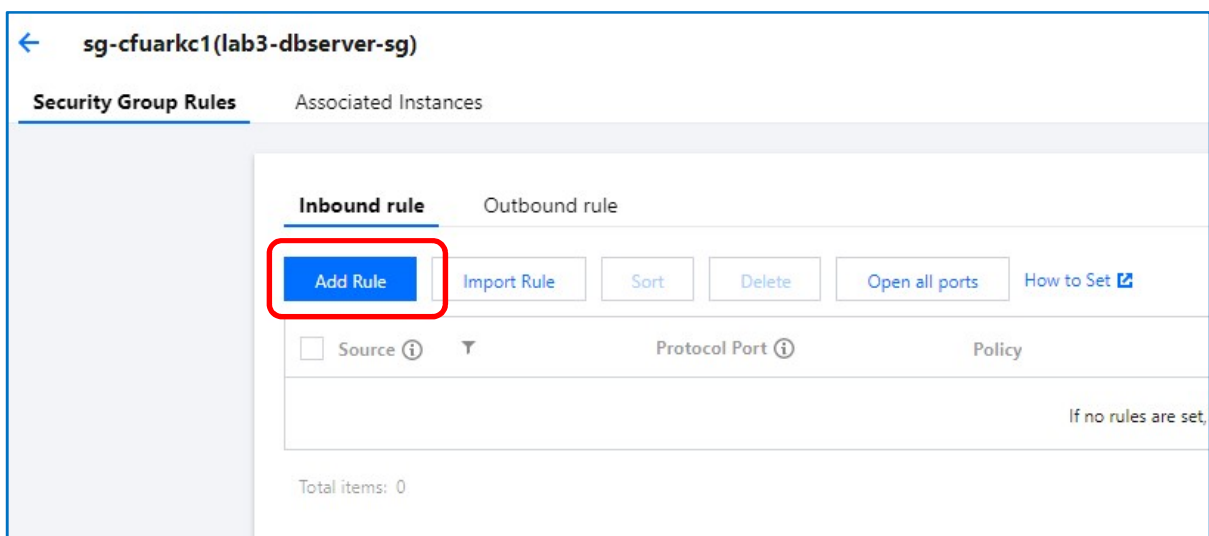
Display template rule

OK Cancel

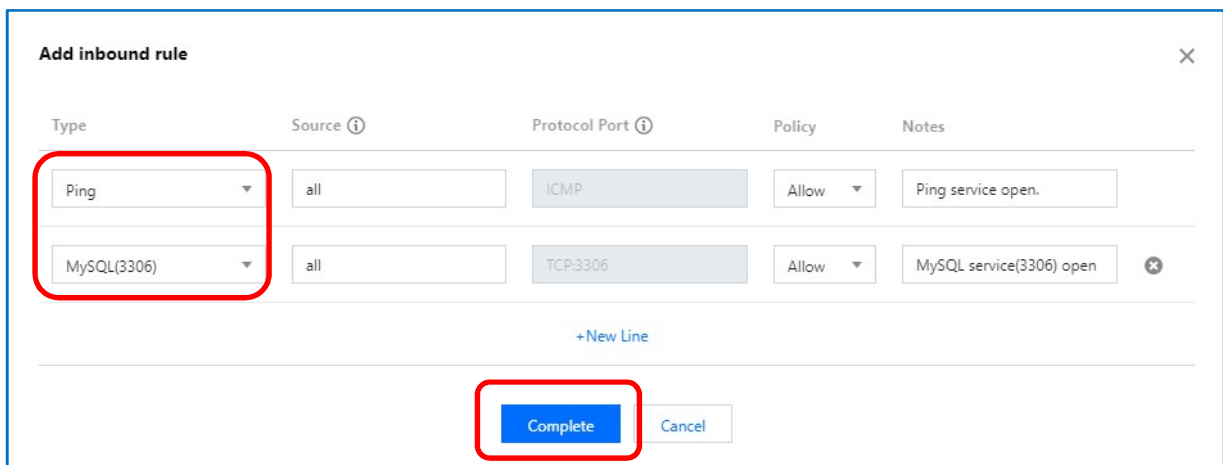
9. [Add rules now] 파란색 버튼을 클릭하여 새로 생성하는 **Security Group**에 Port를 추가하자.



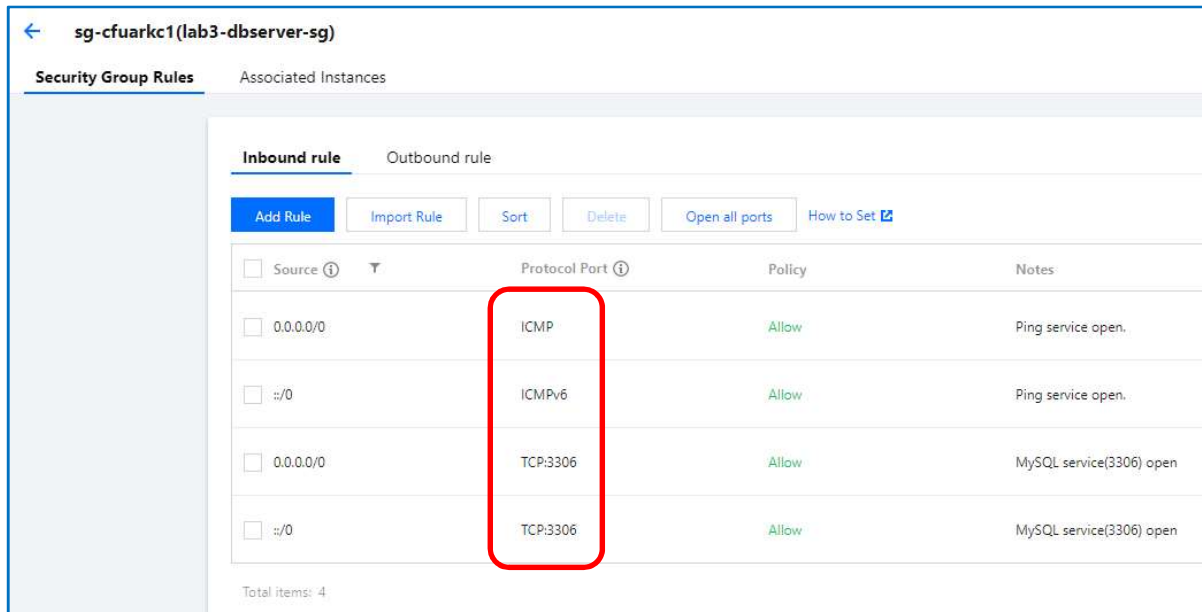
10. 아직 **Inbound rule**이 없다. [Add Rule] 파란색 버튼을 클릭하여 **Inbound rule**을 추가한다.



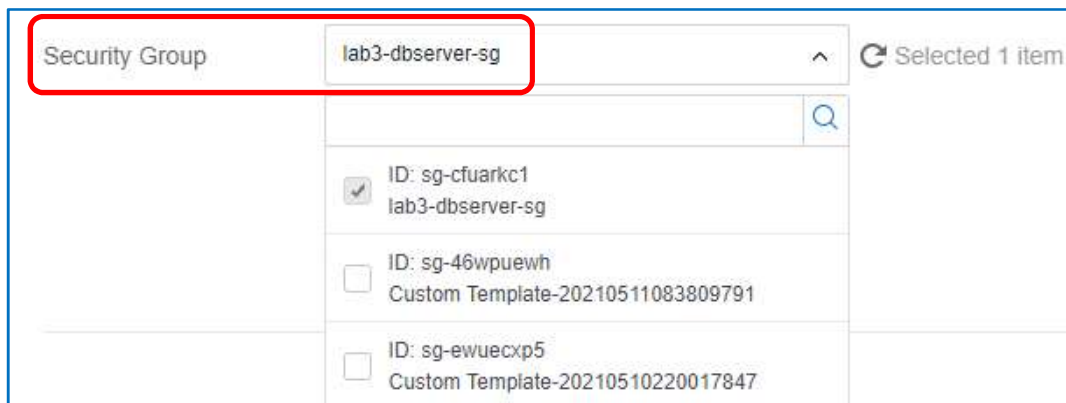
11. MySQL Database Server Instance는 **Ping**과 **MySQL port(3306)**만 추가한다. 더 포트를 추가하려면 [+New Line] 링크를 클릭하여 추가하면 된다. 필요한 포트를 추가했으면 [Complete] 파란색 버튼을 클릭하여 페이지를 닫는다.



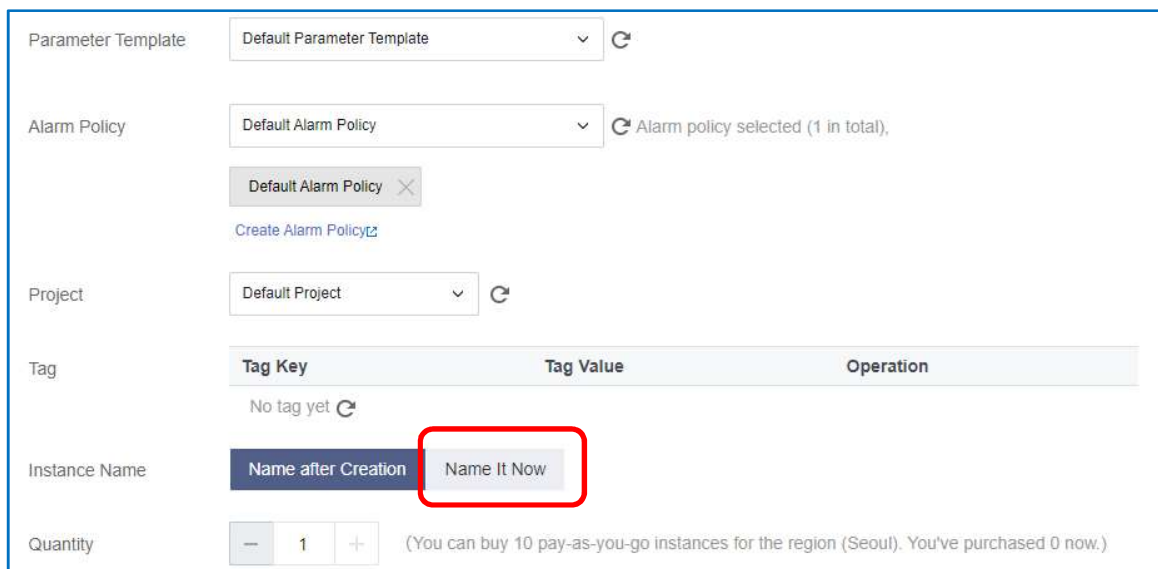
12. 방금 추가한 포트 목록이다. **Ping**을 위한 **ICMP**와 **MySQL Port**인 **TCP:3306**을 확인할 수 있다.



13. 다시 MySQL 설정페이지로 돌아와서 방금 생성한 **Security Group**으로 설정을 맞춘다.



14. 나머지 설정 값은 그대로 기본값을 사용하기로 한다. 다만, [Instance Name]은 [Name It Now] 즉 지금 설정하기로 한다. [Name It Now]를 클릭한다.



15. [Instance Name]은 lab3-mysql로 넣는다. 이제 모든 설정을 마쳤다. [Buy Now] 주황색 버튼을 클릭한다.

Instance Name

Name after Creation Name It Now

lab3-mysql Up to 60 characters comprised of letters, digits and special characters (-_./()[])

When you purchase instances in batch, a numeric sequence number will be added to the end of the custom instance name.

Quantity - 1 + (You can buy 10 pay-as-you-go instances for the region (Seoul). You've purchased 0 now.)

Fees

Configuration Fees

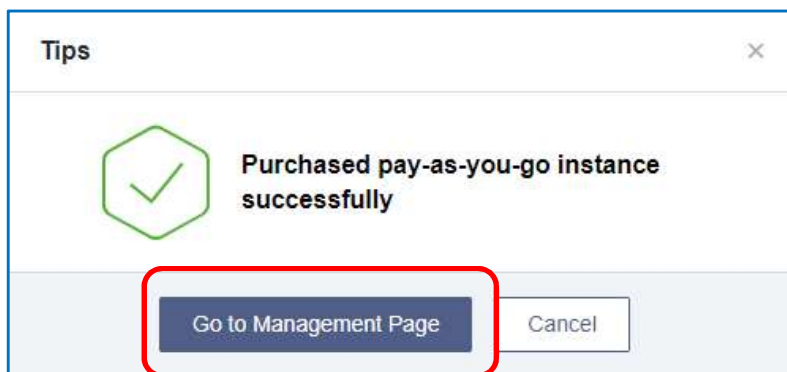
0.10277956 USD/hour (After 15 days of use, it will be reduced to 0.07500178 USD/hour ?)

Backup Fees ? 0.000127 USD/GB/hour (Excess space beyond free limit will be charged)

Traffic Fees 0.00 USD/GB ?

Buy Now

16. 잠시 후, 아래 그림과 같이 설치 성공 다이얼로그가 나타나면 [Go to Management Page] 버튼을 클릭하여 MySQL - Instance List 페이지로 이동하자.



17. Seoul 리전에 MySQL Instance가 생성되었다.

MySQL - Instance List

Guangzhou(0)

Qingyuan(0)

Shanghai(0)

Beijing(0)

Chengdu(0)

Chongqing(0)

Hong Kong (China)(0)

Singapore(0)

Bangkok(0)

Mumbai(0)

Seoul(1)

Tokyo(0)

Frankfurt(0)

Moscow(0)

Create


Comparative Monitoring

Restart

Renew

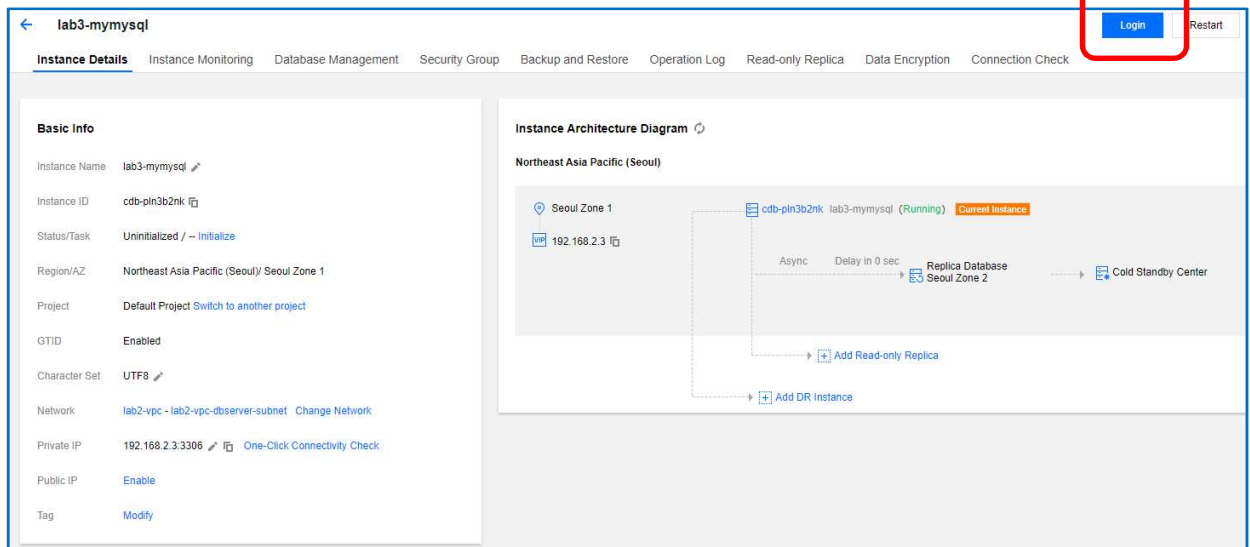
More

Multiple keywords are separated by the vertical bar "|", and multiple keywords are separated by the vertical bar "|".

<input type="checkbox"/>	Instance ID/Name	Monitoring/Status/Task	AZ	Configuration	Version	Private IP	Billing Mode
<input type="checkbox"/>	<div><div>cdb-pln3b2nk</div><div>lab3-mysql</div></div>	<div><div></div><div>Uninitialized</div></div>	Seoul Zone 1	Two-node 1core1000MB/200GB Network: lab2-vpc-lab2-vpc-dbsvr-subnet	MySQL5.7	192.168.2.3306	Pay as you go

Task3. MySQL Database Server에 직접 연결하기

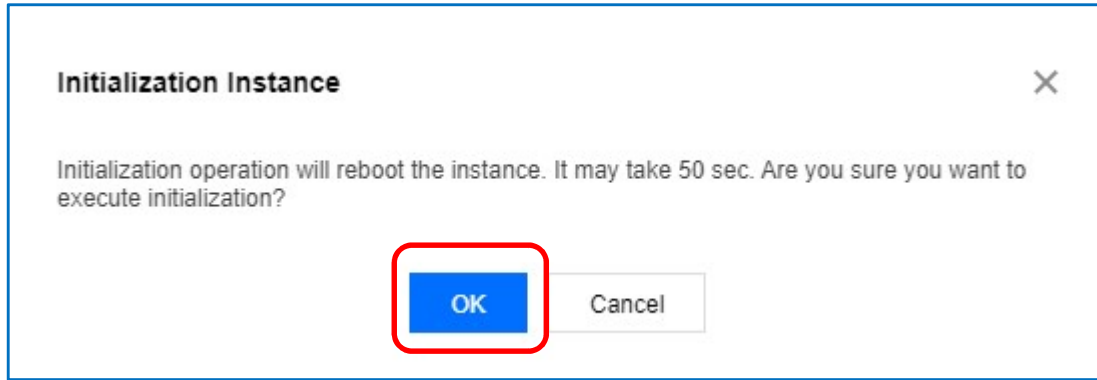
1. 먼저 방금 생성한 MySQL Database Server에 연결하기 위해 **[MySQL – Instance List]**에서 해당 DB Server를 클릭한다. 해당 Database Server의 정보를 알 수 있다. 우측 상단의 **[Login]** 파란색 버튼을 클릭해보자.



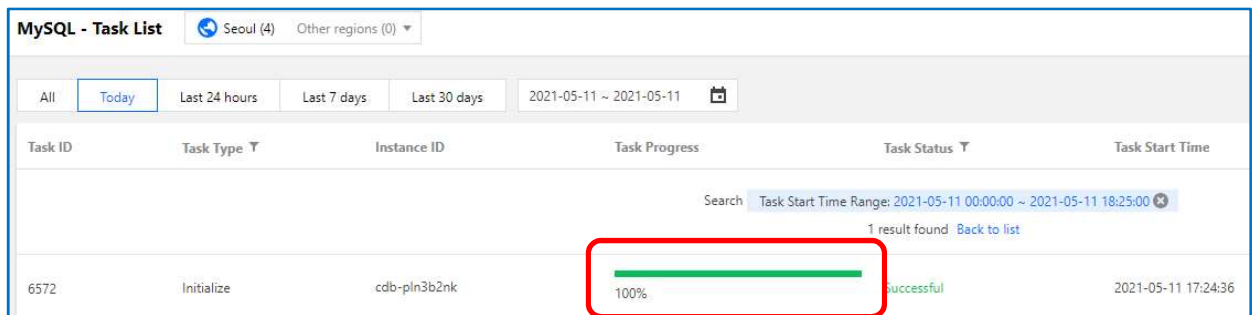
2. 원래는 Database Server를 생성한 후, 초기화 과정을 거쳐야 하는데, 아직 초기화과정을 거치지 않았다면 아래 그림과 같이 초기화 페이지로 넘어온다. 여기서 Login을 위한 비밀번호를 설정하게 된다. 이번 실습에서는 비밀번호를 Lab1에서 Linux Server 관리자 비밀번호였던 **P@\$W0rd1234**를 그대로 사용하기로 한다. **[OK]** 파란색 버튼을 클릭하여 초기화를 마친다.

The screenshot shows the 'Initialize' dialog box for the MySQL instance. It includes options for 'Support Character Set' (LATIN1, UTF8, GBK, UTF8MB4), 'Table Name Case Sensitivity' (toggle), 'Custom Port' (3306), 'Set Password of Root Account' (password field), and 'Confirm Password' (password field). The 'Custom Port' field is highlighted with a red box. The 'Set Password of Root Account' field is also highlighted with a red box. The 'Confirm Password' field is also highlighted with a red box. The 'OK' button is highlighted with a red box.

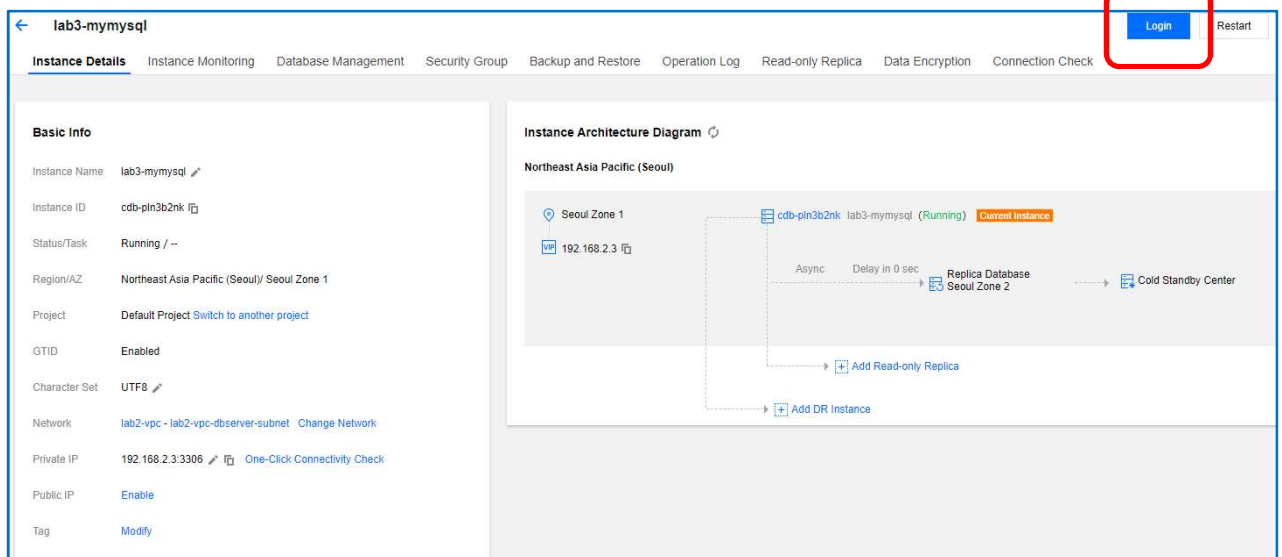
3. [OK] 파란색 버튼을 클릭하여 초기화를 시작한다. 초기화 과정은 해당 Database Instance를 재부팅하는데, 약 50초 정도의 시간이 필요하다.



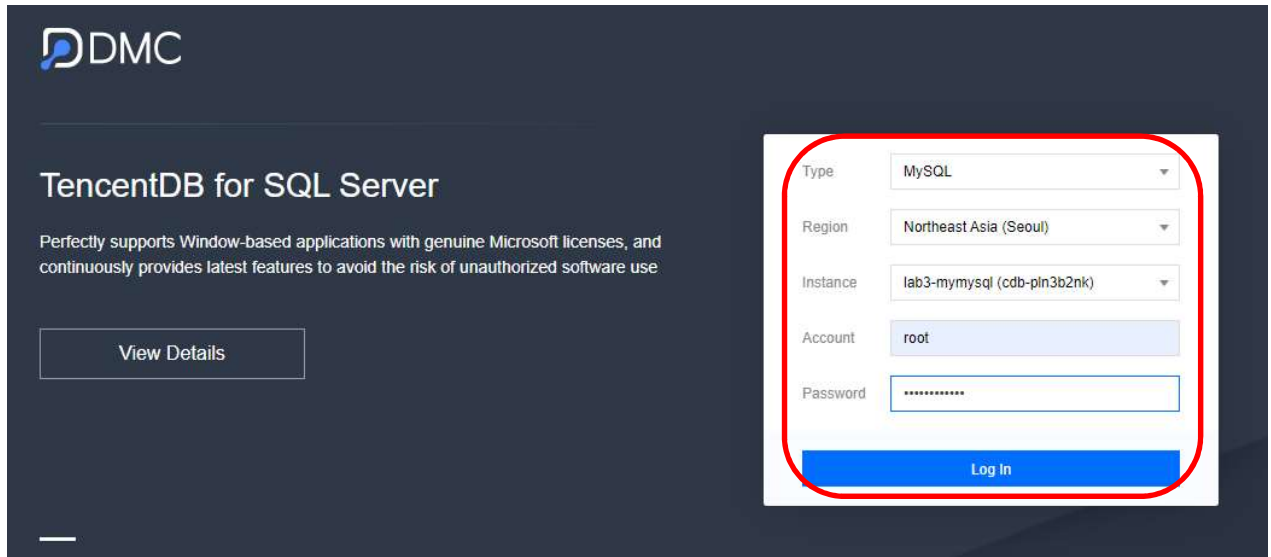
4. [MySQL – Task List] 페이지로 넘어간다. [Task Progress]가 100%임을 확인하여 초기화 과정이 모두 끝났음을 알 수 있다.



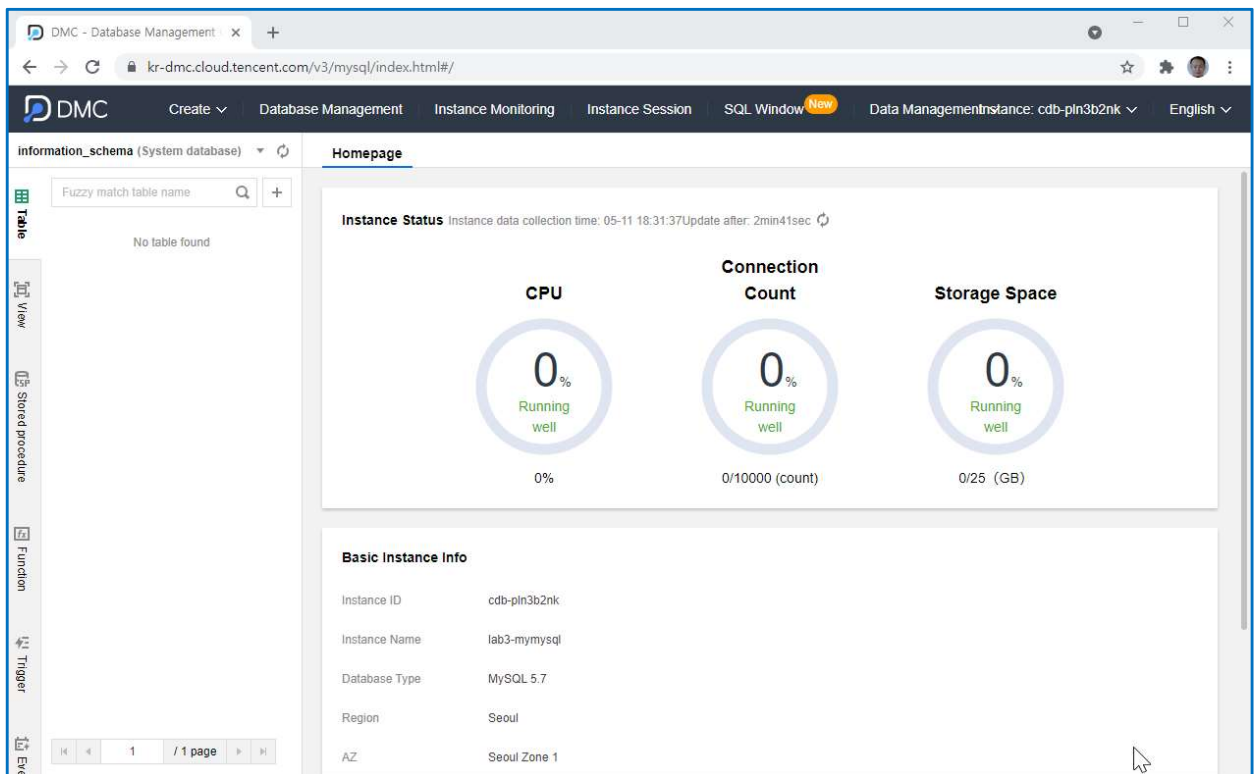
5. [MySQL - Instance List]에서 [lab3-mysql]을 클릭하여 해당 Database Server 정보페이지로 들어오자. 다시 우측 상단의 [Login] 파란색 버튼을 클릭하자.



6. [DMC]창이 나타난다. [Account]는 root로, [Password]는 위에서 이미 설정한 P@\$W0rd1234를 입력하고 [Log In] 파란색 버튼을 클릭하여 로그인하자.

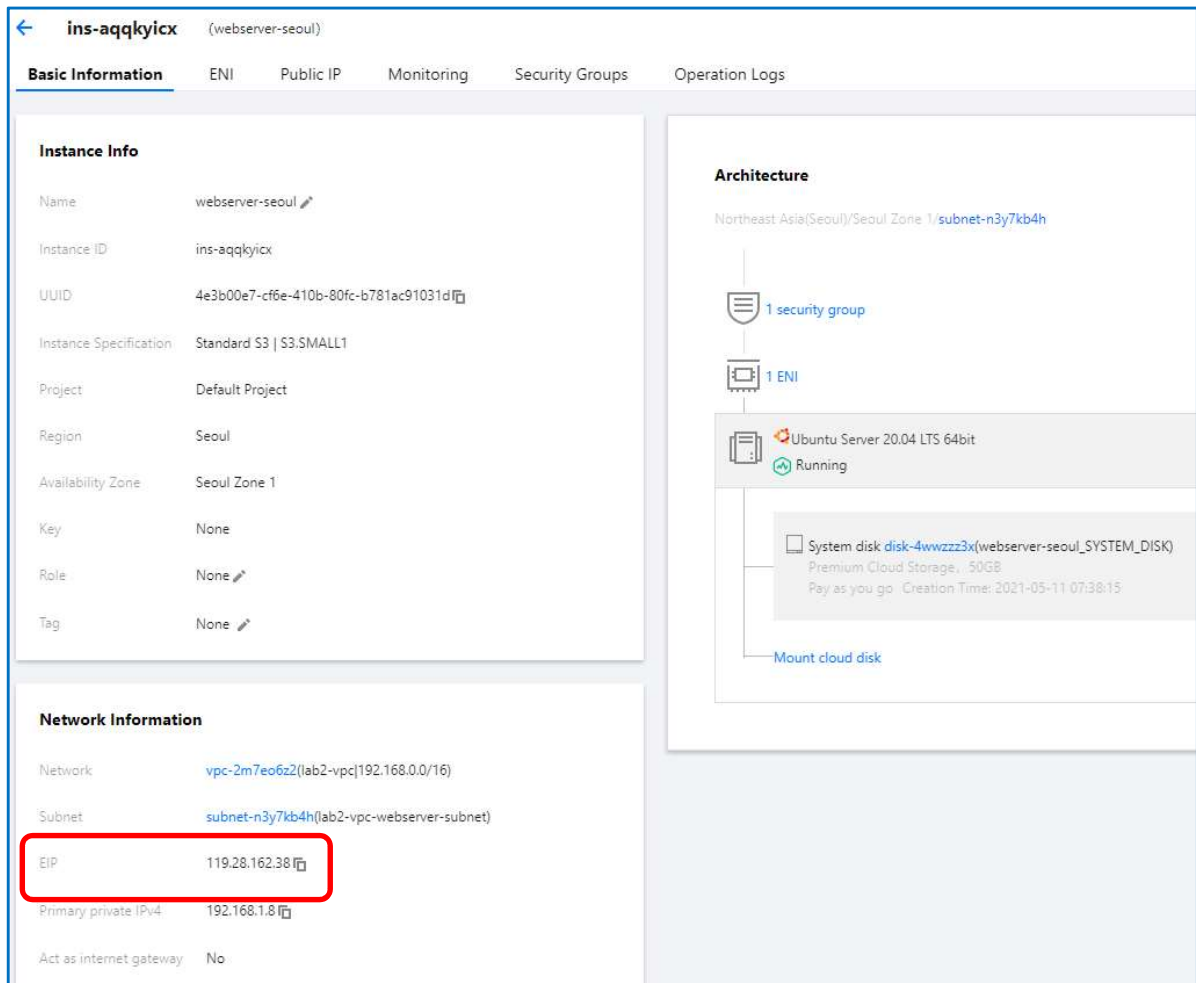


7. 연결에 성공했다. 아직 어떤 Table도 어떤 Data도 없는 것을 확인할 수 있다.

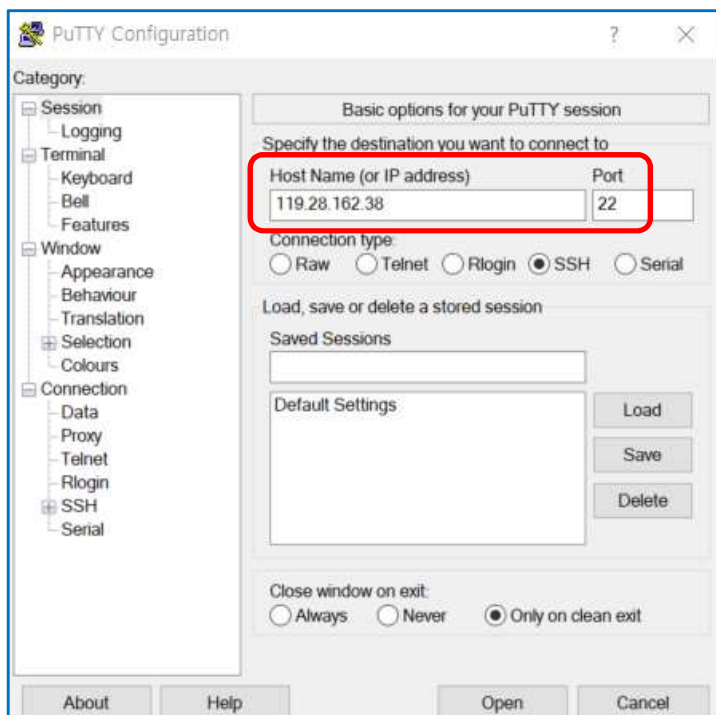


Task4. Web Server에서 Private Network으로 Database Server에 연결하기

1. Lab1에서 생성한 **webserver-seoul**에 연결하기 위해 **EIP** 주소를 복사한다.



2. **PUTTY**를 열고 **SSH**를 통해 **webserver-seoul Instance**에 연결하자.



3. 로그인에 성공했다.

```
ubuntu@webserver-seoul: ~  
login as: ubuntu  
ubuntu@119.28.162.38's password:  
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-72-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Tue 11 May 2021 05:42:01 PM CST  
  
System load:  0.0           Processes:            111  
Usage of /:   6.8% of 49.16GB Users logged in:          0  
Memory usage: 25%          IPv4 address for eth0: 192.168.1.8  
Swap usage:   0%  
  
* Pure upstream Kubernetes 1.21, smallest, simplest cluster ops!  
  
https://microk8s.io/  
  
Last login: Tue May 11 07:41:19 2021 from 211.60.50.190  
ubuntu@webserver-seoul:~$
```

4. **webserver-seoul**에 연결 후, 먼저 **apt list update**부터 시작한다.

\$ sudo apt update

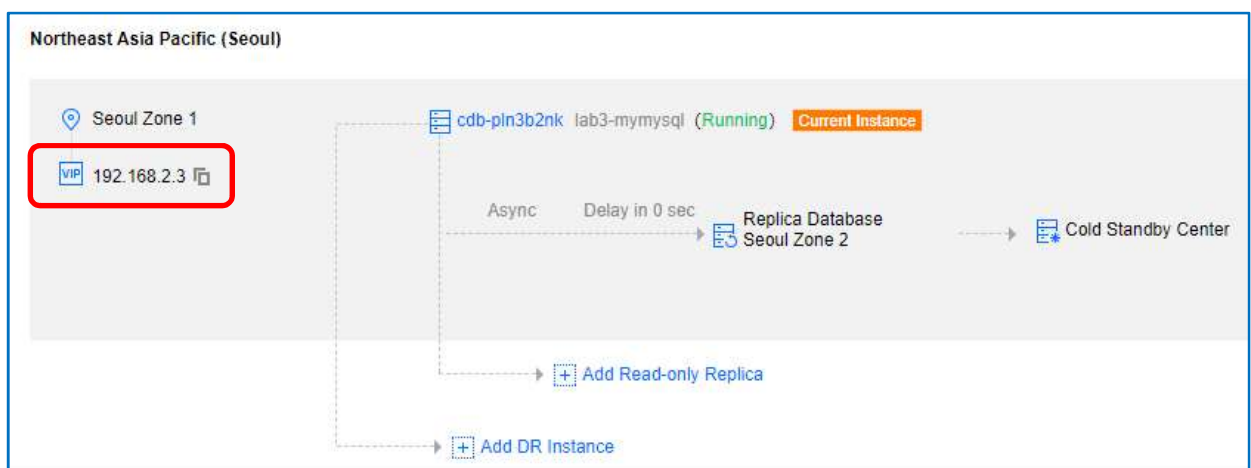
```
ubuntu@webserver-seoul: ~  
  
* Pure upstream Kubernetes 1.21, smallest, simplest cluster ops!  
  
https://microk8s.io/  
  
Last login: Tue May 11 07:41:19 2021 from 211.60.50.190  
ubuntu@webserver-seoul:~$ sudo apt update  
Hit:1 http://mirrors.tencentyun.com/ubuntu focal InRelease  
Get:2 http://mirrors.tencentyun.com/ubuntu focal-security InRelease [109 kB]  
Get:3 http://mirrors.tencentyun.com/ubuntu focal-updates InRelease [114 kB]  
Get:4 http://mirrors.tencentyun.com/ubuntu focal-updates/main amd64 Packages [971 kB]  
Get:5 http://mirrors.tencentyun.com/ubuntu focal-updates/main amd64 c-n-f Metadata [13.2 kB]  
Get:6 http://mirrors.tencentyun.com/ubuntu focal-updates/universe amd64 Packages [772 kB]  
Get:7 http://mirrors.tencentyun.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [17.3 kB]  
Fetched 1,997 kB in 1s (1,812 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
166 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@webserver-seoul:~$
```

5. **webserver-seoul**에서 MySQL Database server에 연결하기 위해 **MySQL Client Tool**을 설치한다.

\$ sudo apt install -y mysql-client


```
ubuntu@webserver-seoul: ~  
Selecting previously unselected package mysql-client-core-8.0.  
(Reading database ... 124663 files and directories currently installed.)  
Preparing to unpack .../mysql-client-core-8.0_8.0.23-0ubuntu0.20.04.1_amd64.deb  
...  
Unpacking mysql-client-core-8.0 (8.0.23-0ubuntu0.20.04.1) ...  
Selecting previously unselected package mysql-common.  
Preparing to unpack .../mysql-common_5.8+1.0.5ubuntu2_all.deb ...  
Unpacking mysql-common (5.8+1.0.5ubuntu2) ...  
Selecting previously unselected package mysql-client-8.0.  
Preparing to unpack .../mysql-client-8.0_8.0.23-0ubuntu0.20.04.1_amd64.deb ...  
Unpacking mysql-client-8.0 (8.0.23-0ubuntu0.20.04.1) ...  
Selecting previously unselected package mysql-client.  
Preparing to unpack .../mysql-client_8.0.23-0ubuntu0.20.04.1_all.deb ...  
Unpacking mysql-client (8.0.23-0ubuntu0.20.04.1) ...  
Setting up mysql-common (5.8+1.0.5ubuntu2) ...  
update-alternatives: using /etc/mysql/my.cnf.fallback to provide /etc/mysql/my.cnf (my.cnf) in auto mode  
Setting up mysql-client-core-8.0 (8.0.23-0ubuntu0.20.04.1) ...  
Setting up mysql-client-8.0 (8.0.23-0ubuntu0.20.04.1) ...  
Setting up mysql-client (8.0.23-0ubuntu0.20.04.1) ...  
Processing triggers for man-db (2.9.1-1) ...  
ubuntu@webserver-seoul:~$
```

6. 이제 **MySQL Client Tool**을 통해 **webserver-seoul**에서 **lab3-mymysql**에 연결해 보자. 현재 이 실습에서의 **lab3-mymysql** 머신의 IP는 **192.168.2.3**이다.



```
$ mysql -h 192.168.2.3 -u root -p
```

```
ubuntu@webserver-seoul: ~  
ubuntu@webserver-seoul:~$ mysql -h 192.168.2.3 -u root -p  
Enter password: 
```

7. 비밀번호 **P@\$W0rd1234**를 입력하면 **webserver-seoul**에서 **lab3-mymysql Database server**에 로그인되는 것을 볼 수 있다.

```
ubuntu@webserver-seoul: ~  
ubuntu@webserver-seoul:~$ mysql -h 192.168.2.3 -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 1407  
Server version: 5.7.18-txsq1-log 20201231  
  
Copyright (c) 2000, 2021, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> 
```

8. Database를 조회하니깐 root권한으로 볼 수 있는 Database 목록이 보인다.

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql> 
```