

GCP

인스턴스 생성 및 실행

1 프로젝트 생성

프로젝트 선택 > 새 프로젝트 > 프로젝트 이름 입력 후 만들기

2. Compute Engine

Compute Engine - VM 인스턴스 선택 > Compute Engine API 사용 > 인스턴스 만들기

3. 인스턴스 만들기

1. VM 이름 지정

2. 영역

리전: asia-northeast3 (서울)

영역: asia-northeast3-a

3. 머신 구성(기본 설정에서 변경 x)

일반 용도

시리즈: E2

머신 유형: e2-medium

CPU 플랫폼: 자동

4. 부팅 디스크

운영체제: CentOS

버전: CentOS 7

부팅 디스크 유형: 균형 있는 영구 디스크

크기(GB): 20

5. 액세스 범위

모든 Cloud API에 대한 전체 액세스 허용

6. 방화벽

HTTP 트래픽 허용, HTTPS 트래픽 허용 둘 다 체크

4. 네트워크

VPC 네트워크 - 외부 IP 주소 - 해당 VM의 가장 오른쪽에 예약 - 이름 설정

해당 VM의 유형이 임시에서 고정으로 바뀐 것을 볼 수 있다!

인스턴스가 재실행 될 경우 IP가 변경될 가능성이 있으므로 이를 막도록 고정 IP 등록하는 것?

5. VM 인스턴스 연결(PuTTY 사용)

1. PuTTY 설치

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

2. PuTTYgen 실행

Generate 선택 후 key 생성 - Save private key(개인 키 저장), Save public key(공개 키 저장)

Conversions - Export OpenSSH Key(PuTTY 외에서 사용)

3. SSH 키 등록

<input type="checkbox"/>	Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/>	<input checked="" type="checkbox"/> instance-1	us-east1-b		10.142.0.2 (nic0)	35.231.114.114 ↗	SSH ▾ ⋮

사용할 VM 행의 SSH 선택

```
# root 비밀번호 설정
sudo passwd

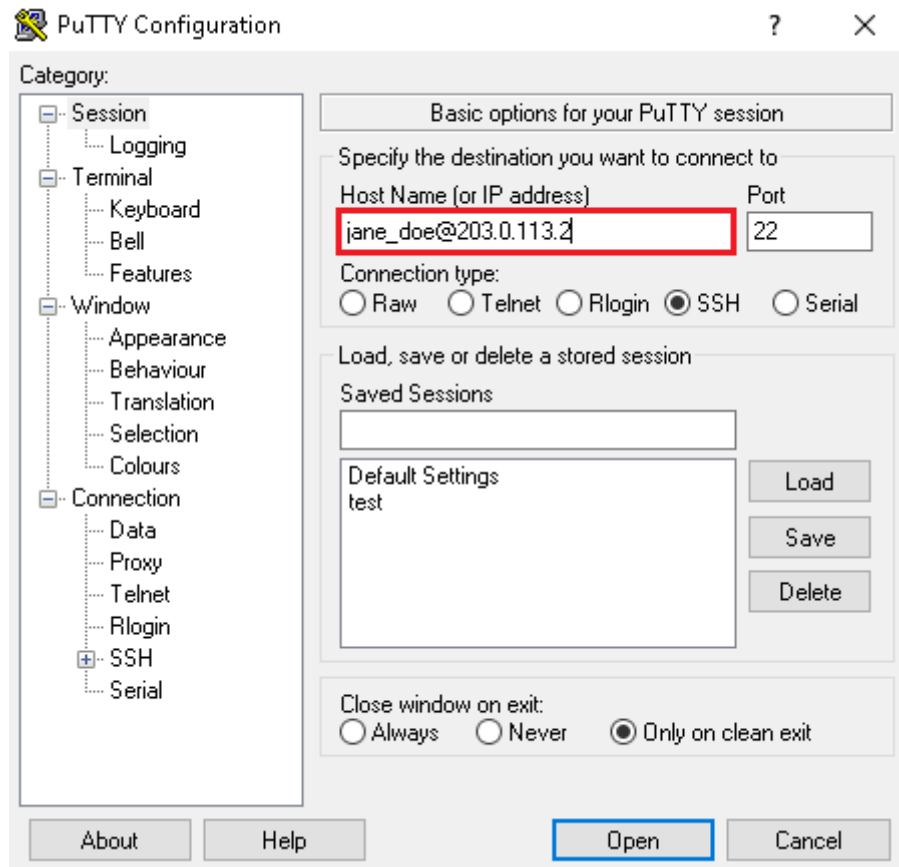
# root 로그인
su -

# 계정 생성
# useradd -g {그룹명} -d {계정 홈 디렉토리} {계정이름}
useradd -g wheel -d /home/yum yum

# 계정 디렉토리에서 작업 (/home/{계정})
# .ssh 폴더 생성
mkdir .ssh

# puttygen에서 생성된 authorized_keys file 내용 붙여넣기
vi .ssh/authorized_keys
# i 누리고 붙여넣기(shift+insert), esc 누리고 ':wq' 입력하여 파일 저장해서 나오기
```

4. PuTTY 실행



Session

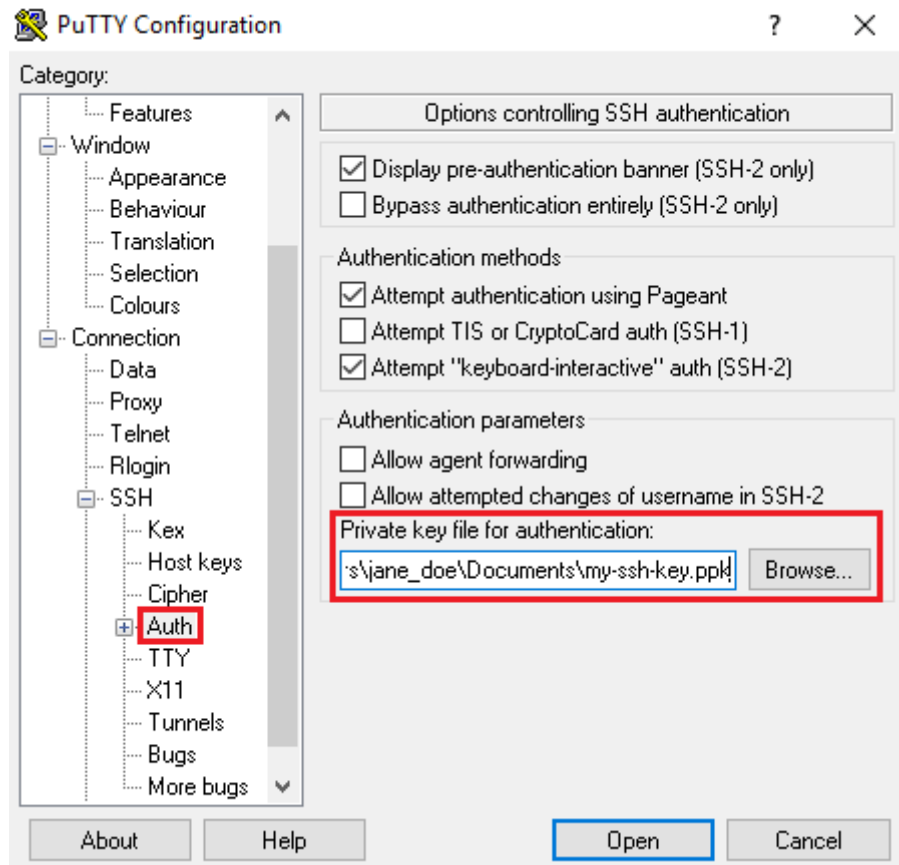
Host Name

: 인스턴스 연결하는 사용자 이름 or 작업하는 계정 이름@외부 IP 주소

- IP 주소만 입력하고 싶으면 Connection > Data > Auto-login username에서 사용자 이름 치면 된당

Port: 22

Connection type: SSH



Connection > Auth

Private key file for authentication > PuTTYgen에서 저장했던 private key(.ppk) 불러오기

다시 Session

Saved Sessions에서 session 이름 입력 후 Save

Open!!!

다음에 다시 접속 할 때는 저장한 session Load 후 Open하면 된다!

PostgreSQL

1. 설치

<https://www.postgresql.org/ftp/source/v14.1/>

tar.gz 파일 우클릭 - 링크 주소 복사

```
# root 로그인
su -
# /root에서 작업
cd /root

# 사전 패키지 설치
yum -y install readline readline-devel zlib zlib-devel openssl openssl-devel
python python-devel perl systemd-devel gcc gcc-c++ systemd systemd-devel gettext
gettext-devel autoconf wget flex flex-devel

# 웹 서버에서 콘텐츠(.tar.gz) 가져오기
wget https://ftp.postgresql.org/pub/source/v14.1/postgresql-14.1.tar.gz
```

```
# 압축풀기
tar xvfz postgresql-14.1.tar.gz
cd postgresql-14.1

# 빌드 환경 구성
./configure --prefix=/usr/local/yum/pgsql --with-python --with-openssl --with-systemd --enable-nls=ko

# 빌드 및 설치
make
make install

# 확인
cd /usr/local/yum
ls -la
```

2. postgres 계정 만들기, 권한

```
# 그룹 생성
groupadd postgres
useradd -M -g postgres postgres

# 권한 설정
chown -R postgres:postgres /usr/local/yum/pgsql

# 권한 확인
cd /usr/local/yum
ls -la
```

```
[root@yum yum]# ls -la
total 0
drwxr-xr-x. 3 root root 19 Nov 28 02:33 .
drwxr-xr-x. 13 root root 142 Nov 28 02:22 ..
drwxr-xr-x. 6 root root 56 Nov 28 17:18 pgsql
[root@yum yum]# ls -la
total 0
drwxr-xr-x. 3 root root 19 Nov 28 02:33 .
drwxr-xr-x. 13 root root 142 Nov 28 02:22 ..
drwxr-xr-x. 6 postgres postgres 56 Nov 28 17:18 pgsql
```

권한이 root에서 postgres로 바뀐 것을 볼 수 있다!!

3. 환경 설정

```
vi /etc/profile
```

아래 내용 입력

```
# postgres
export POSTGRES_HOME=/usr/local/yum/pgsql
export PGLIB=$POSTGRES_HOME/lib
export PGDATA=$POSTGRES_HOME/data
export PATH=$PATH:$POSTGRES_HOME/bin
```

esc 누른 후 ':wq' 입력하고 저장하여 빠져나오기

```
source /etc/profile
```

4. 기본 데이터베이스 생성

```
sudo -u postgres /usr/local/yum/pgsql/bin/initdb -E UTF8 --locale=ko_KR.UTF-8  
/usr/local/yum/pgsql/data
```

5. 실행

```
# 실행  
sudo -u postgres /usr/local/yum/pgsql/bin/pg_ctl -D /usr/local/yum/pgsql/data -l  
/usr/local/yum/pgsql/data/logfile start  
  
# 실행 확인  
ps -ef | grep postgres  
cat logfile      # /usr/local/yum/pgsql/data에서 확인
```

+ systemd 사용을 위한 제어 스크립트

--with-systemd 컴파일 옵션을 추가 했을 시 사용

systemctl start/stop/restart postgresql 을 사용할 수 있음

```
vi /etc/systemd/system/postgresql.service
```

```
[Unit]  
Description=PostgreSQL database server  
Documentation=man:postgres(1)  
  
[Service]  
Type=notify  
User=postgres  
  
ExecStart=/usr/local/yum/pgsql/bin/postgres -D /usr/local/yum/pgsql/data -p  
5432 -c config_file=/usr/local/yum/pgsql/data/postgresql.conf  
ExecReload=/bin/kill -HUP $MAINPID  
KillMode=mixed  
KillSignal=SIGINT  
TimeoutSec=0  
  
[Install]  
WantedBy=multi-user.target
```

```
systemctl enable postgresql
```

```
# systemctl을 이용하여 실행  
systemctl start postgresql
```

6. postgres 비밀번호 설정

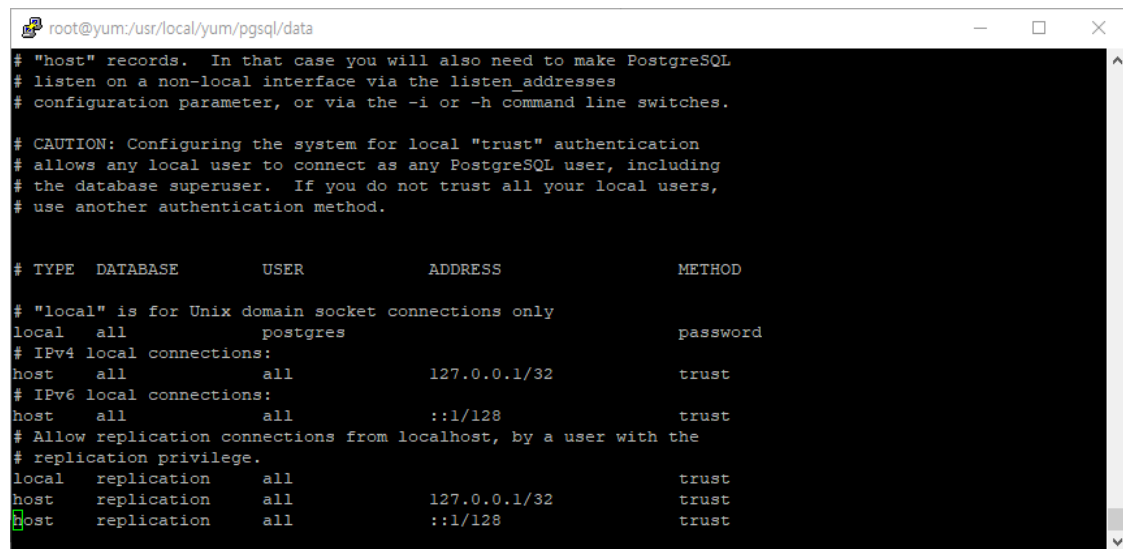
```
psql -U postgres
```

```
# alter user {계정} with
alter user postgres with password 'postgres';

# postgres 나오기
\q
```

```
# "local" is for Unix domain socket connections only
local  all          all          trust
```

아래와 같이 all -> postgres, trust -> password로 변경



```
root@yum:/usr/local/yum/pgsql/data
# "host" records. In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.

# CAUTION: Configuring the system for local "trust" authentication
# allows any local user to connect as any PostgreSQL user, including
# the database superuser. If you do not trust all your local users,
# use another authentication method.

# TYPE  DATABASE  USER  ADDRESS  METHOD
# "local" is for Unix domain socket connections only
local   all           postgres          password
# IPv4 local connections:
host    all           all          127.0.0.1/32      trust
# IPv6 local connections:
host    all           all          ::1/128          trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication  all           trust
host    replication  all          127.0.0.1/32      trust
host    replication  all          ::1/128          trust
```

```
# postgres 종료
killall postgres
# 안 될 경우
pkill postgres
# 또는
systemctl stop postgresql

# postgres 실행
sudo -u postgres /usr/local/yum/pgsql/bin/pg_ctl -D /usr/local/yum/pgsql/data -l
/usr/local/yum/pgsql/data/logfile start
# 또는
systemctl start postgresql

# restart도 가능
systemctl restart postgresql

# postgres 계정으로 접속
psql -U postgres
```

비밀번호를 입력해야 들어가진다!

+ 기본 명령어

```
\q # 나가기
\! {명령어} # 리눅스 명령어 사용가능 ex) \! clear
select version(); # 버전 확인
select current_date; # 오늘 날짜 확인
\? # help
```

7. PostgreSQL 사용해보기

- 데이터베이스 생성

```
create database webdb;
# 확인
\l
# 연결
\c webdb
```

- 테이블 생성

```
create table pet(
    name varchar(20),
    owner varchar(20),
    species varchar(20),
    gender char(1),
    birth date,
    death date);

# 테이블 확인
\dt
# 스키마 확인
\d users
```

- 테이블 삭제

```
drop table pet;
```

- 값 넣기

```
insert into pet values('별이', '주이', '포메', 'f', '2016-03-13', null);

# 확인
select * from pet;
```

- text 파일 복사 -> 데이터 복사

```
# postgresql 나오기
\q

# 파일 생성
vi /home/yum/pet.txt
```


Fluffy	Harold	cat	f	1993-02-04	\N
Claws	Gwen	cat	m	1994-03-17	\N
Buffy	Harold	dog	f	1989-05-13	\N
Fang	Benny	dog	m	1990-08-27	\N
Bowser	Diane	dog	m	1979-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	\N
Whistler	Gwen	bird	f	1997-12-09	\N
Slim	Benny	snake	m	1996-04-29	\N

```
# postgresql 접속
psql -U postgres
```

```
\c webdb

# 복사
\copy pet from '/home/yum/pet.txt';

# 확인
select * from pet;
```

8. webdb 계정 생성 및 사용하기

- 사용자 추가

```
psql -U postgres
```

```
create user webdb with password 'webdb';
```

- 테이블 권한 추가

```
# 스키마 확인
\dn

# 권한 추가
grant all privileges on all tables in schema public to webdb;
# 테이블 소유자 변경
\c webdb
alter table pet owner to webdb;

# 나가기
\q
```

- 연결 설정

```
vi /usr/local/yum/pgsql/data/pg_hba.conf
```

local	all	postgres	password	아래
local	webdb	webdb	password	추가

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all postgres password
local webdb webdb password
```

```
# 재시작
systemctl restart postgresql
```

- webdb 계정 연결

```
psql -U webdb;
```

```
select * from pet;
```

9. 외부 연결

```
vi /usr/local/yum/pgsql/data/postgresql.conf
```

```
#listen_addresses = 'localhost' -> listen_addresses = '*'

# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
listen_addresses = 'localhost' # what IP address(es) to listen on;

# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
listen_addresses = '*' # what IP address(es) to listen on;
```

esc 누르고 `:wq` 저장 후 나가기

```
vi /usr/local/yum/pgsql/data/pg_hba.conf
```

IPv4 local connections: 밑에 아래 내용 추가

host	all	postgres	127.0.0.1/32	password
host	webdb	webdb	127.0.0.1/32	password
host	webdb	webdb	{자신의 내부 IP}/32	password

`create user 'webdb'@'{내부 IP}'` 같은 의미

- 내부 IP는 GCP 또는 `ifconfig`를 입력하여 알 수 있다!

```
root@yum:/usr/local/yum/pgsql/data

# CAUTION: Configuring the system for local "trust" authentication
# allows any local user to connect as any PostgreSQL user, including
# the database superuser. If you do not trust all your local users,
# use another authentication method.

# TYPE  DATABASE  USER        ADDRESS      METHOD
# "local" is for Unix domain socket connections only
local   all      postgres    password
local   webdb   webdb       password
# IPv4 local connections:
host    all      postgres    127.0.0.1/32 password
host    webdb   webdb       127.0.0.1/32 password
host    webdb   webdb       [REDACTED]/32 password
# IPv6 local connections:
host    all      all         ::1/128      trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication  all      trust
host    replication  all      127.0.0.1/32 trust
host    replication  all      ::1/128      trust
```

esc 누르고 `:wq` 저장 후 나가기

```
# 재시작
systemctl restart postgresql
```

10. TablePlus

- 설치

<https://tableplus.com/windows> 다운로드

- 방화벽 풀기

GCP

VPN 네트워크 > 방화벽 > 방화벽 규칙 만들기

이름: 알아서
트래픽 방향: 수신
대상: 네트워크의 모든 인스턴스
소스 필터: IPv4
범위: 0.0.0.0/0
프로토콜 및 포트: 지정된 프로토콜 및 포트

- tcp: 5432

- connection

Menu



Welcome to TablePlus

Build 182



Backup



Restore



Register

You are using the free trial

Create a new connection...

No connection! Right click or click the "Create a new connection..." button to create a new connection

Group...

MySQL...

MariaDB...

SQLite...

Cassandra...

Cockroach...

Snowflake...

PostgreSQL...

Greenplum...

Amazon Redshift...

Microsoft SQL Server...

Redis (Beta)...

MongoDB (Beta)...

Import from URL...

Import from file...

PostgreSQL Connection

×

Name

webdb

Status color

Tag

development

▼

Host

Port

5432

User

webdb

Password

●●●●●●

Database

webdb

Bootstrap commands...

SSL mode

PREFERRED

▼

Clear keys

Key...

Cert...

CA Cert...

☒ Over SSH

Server

Port

User

yum

Password

☒ Use SSH Key

yum_open_key

Passphrase

Save

Test

Connect

Name: TablePlus에서 사용할 database 별칭

Tag: development

Host: 자신의 내부 IP

Port: 5432

User: webdb(postgres에서 만든 user name)

Password: webdb(postgres에서 만든 user의 password)

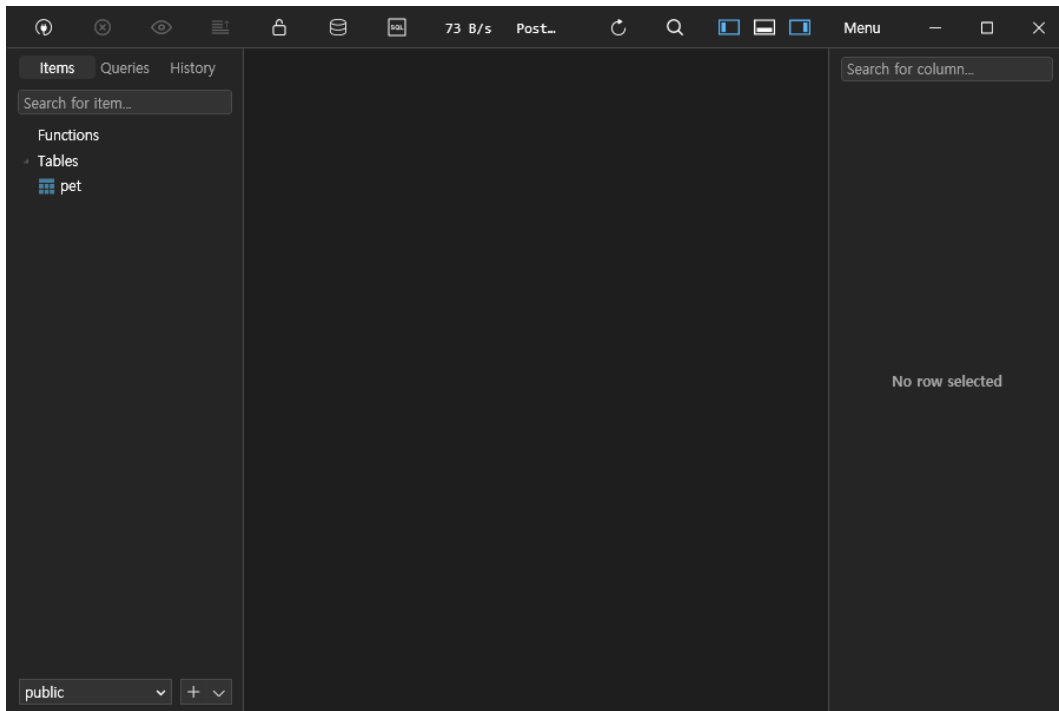
Database: webdb(postgres에서 만든 데이터베이스)

Over SSH - 체크

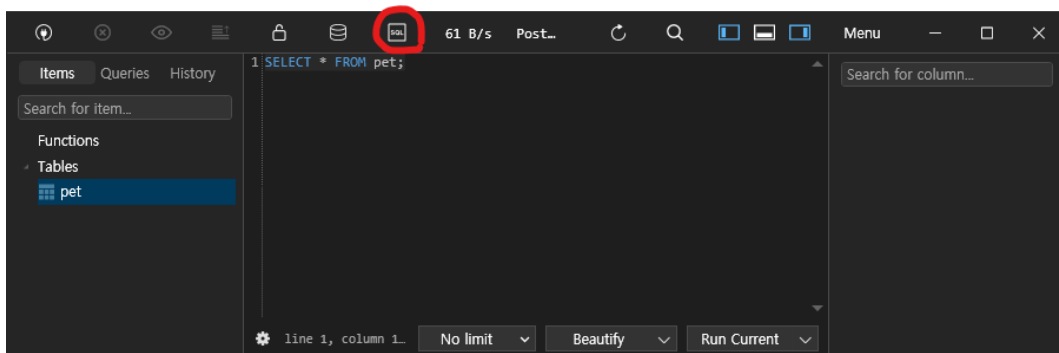
Server: 자신의 외부 IP

User: key를 등록했던 계정

Use SSH Key - 체크 후 Puttygen에서 OpenSSH key 파일 불러오기



드디어 됐다.....



상단 네모 박스 안에 SQL이 써져 있는 버튼을 클릭하면 쿼리문을 작성할 수 있는 탭이 나온다!

Java

1. JDK 설치

- 파일 옮기기
GCP SSH를 사용하여 업로드 하는 방법도 있지만 업로드 시간이 오래 걸려서 이 방법을 사용!
만약 GCP를 이용한다고 하면 GCP SSH로 접속하여 오른쪽 상단 설정 아이콘을 누르고 파일 업로드 하면 되당!

1. XShell 실행

```
sftp {key 등록된 계정명}@{외부 IP 주소}
```

수락

2. SSH 사용자 인증

Public Key

- 사용자 키: 찾아보기 > 가져오기 > PuTTYgen으로 생성한 OpenSSH key

3. 옮기기

<https://www.oracle.com/kr/java/technologies/javase/javase8u211-later-archive-downloads.html>

위 사이트에서 jdk-8u291-linux-x64.tar.gz 다운로드

```
put C:\jdk-8u291-linux-x64.tar.gz
```

put 뒤에는 파일 경로 써주면 됨!

2. 압축 풀기

```
# 다운로드 된 디렉토리로 이동
cd /home/{위 단계에서 접근한 계정}v

# 압축 풀기
tar xvfz jdk-8u291-linux-x64.tar.gz
```

3. 설치 및 링크 파일 생성

```
mv jdk1.8.0_291 /usr/local/yum/java1.8
ln -s /usr/local/yum/java1.8 /usr/local/yum/java
```

4. 설정

```
vi /etc/profile
```

가장 아래에 추가

```
# java
export JAVA_HOME=/usr/local/yum/java
export CLASSPATH=.:$JAVA_HOME/lib/tools.jar
export PATH=$PATH:$JAVA_HOME/bin
```

```
# 현재 shell 환경에 적용
source /etc/profile
```

5. 확인

```
java -version
```

```
Java(TM) SE Runtime Environment (build 1.8.0_291-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.291-b10, mixed mode)
```

위와 같이 출력 되면 성공적으로 설치된 것!

Tomcat

1. 다운로드

```
# 작업 디렉토리 /root
su -
cd /root

# 다운로드
wget https://d1cdn.apache.org/tomcat/tomcat-8/v8.5.73/bin/apache-tomcat-8.5.73.tar.gz
```

wget 뒤 링크는

<https://tomcat.apache.org/download-80.cgi> 에 접속하여 Core의 tar.gz을 우클릭하여 링크 주소 복사!

2. 압축 풀기

```
tar xvfz apache-tomcat-8.5.73.tar.gz
```

3. 설치 및 링크 파일 생성

```
mv apache-tomcat-8.5.73 /usr/local/yum/tomcat8.5
ln -s /usr/local/yum/tomcat8.5 /usr/local/yum/tomcat
```

4. 포트 설정

```
vi /usr/local/yum/tomcat/conf/server.xml
```

/ 입력 후 8080 검색

n 입력하면 next

8080 되어 있는 부분을 80으로 변경

- 변경 전

```
<Connector port="8080" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" />
```

- 변경 후

```
<Connector port="80" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" />
```

port를 8080으로 설정해놓고 작업할 경우!!!

- 방화벽 추가
GCP에서 VPC 네트워크 > 방화벽 > 방화벽 규칙 만들기
 - 트래픽 방향: 수신
 - 대상: 네트워크의 모든 인스턴스
 - 소스 IPv4 범위: 0.0.0.0/0
 - 지정된 프로토콜 및 포트 > tcp : 8080

5. 실행

```
/usr/local/yum/tomcat/bin/catalina.sh start
```

실행 확인

```
ps -ef | grep tomcat
```

6. 브라우저로 접근

- 포트를 80으로 설정했을 경우
http://{외부 IP 주소}
- 포트를 80 외 다른 포트로 설정했을 경우
http://{외부 IP 주소}:8080

7. 중지

```
/usr/local/yum/tomcat/bin/catalina.sh stop
```

8. 서비스 등록하기

```
vi /usr/lib/systemd/system/tomcat.service
```

```
[Unit]
Description=tomcat8
After=network.target syslog.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/local/douzone/java
User=root
Group=root

ExecStart=/usr/local/douzone/tomcat/bin/startup.sh
ExecStop=/usr/local/douzone/tomcat/bin/shutdown.sh

UMask=0007
RestartSec=10
Restart=always
```

```
[Install]
WantedBy=multi-user.target
```

```
systemctl enable tomcat
```

9. tomcat 서비스 실행/중지/재실행

```
# 실행
systemctl start tomcat
# 중지
systemctl stop tomcat
# 재실행
systemctl restart tomcat
```

10. tomcat manager 설정

```
# tomcat-users.xml 수정
vi /usr/local/yum/tomcat/conf/tomcat-users.xml
```

마지막 라인 `</tomcat-users>` 바로 위에 아래 내용 추가

```
<role rolename="manager"/>
<role rolename="manager-gui" />
<role rolename="manager-script" />
<role rolename="manager-jmx" />
<role rolename="manager-status" />
<role rolename="admin"/>
<user username="{manager ID}" password="{manager Password}" roles="admin,
manager, manager-gui, manager-script, manager-jmx, manager-status"/>
```

```
# context.xml 수정
vi /usr/local/yum/tomcat/webapps/manager/META-INF/context.xml
```

가장 아래 `<Context ...> ... </Context>` 부분을 `<!-- -->` 로 주석 처리 후, 아래 내용 추가

```
<Context antiResourceLocking="false" privileged="true"
docBase="${catalina.home}/webapps/manager">
  <valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="^.*$" />
</Context>
```

- 수정 전

```

...
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor
className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|string)|org\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>

```

- 수정 후

```

...
<!--
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor
className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|string)|org\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>
-->
<Context antiResourceLocking="false" privileged="true"
docBase="\${catalina.home}/webapps/manager">
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="\^.*$" />
</Context>

```

11. 재시작

```
systemctl restart tomcat
```

12. 브라우저로 접근

- 포트 80
http://{외부 IP 주소}/manager
- 포트 80 외
http://{외부 IP 주소}:{포트}/manager

Git

1. 사전 패키지 설치

```
su -  
cd /root  
yum -y install curl-devel expat-devel gettext-devel openssl-devel zlib-devel  
perl-devel asciidoc xmlto
```

2. 다운로드

```
wget https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.9.5.tar.gz
```

3. 압축 풀기

```
tar xvfz git-2.9.5.tar.gz
```

4. 소스 디렉토리

```
cd git-2.9.5
```

5. configure

```
./configure --prefix=/usr/local/yum/git
```

6. 빌드 및 설치

```
make  
make install
```

7. 설정

```
vi /etc/profile
```

아래에 내용 추가

```
# git  
export PATH=$PATH:/usr/local/yum/git/bin
```

```
source /etc/profile
```

Maven

1. 다운로드

```
su -  
cd /root  
wget http://mirror.apache-kr.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
```

2. 압축 풀기

```
tar xvfz apache-maven-3.8.4-bin.tar.gz
```

3. 설치

```
mv apache-maven-3.8.4 /usr/local/yum/maven3.8  
ln -s /usr/local/yum/maven3.8 /usr/local/yum/maven
```

4. 설정

```
vi /etc/profile
```

```
# maven  
export PATH=$PATH:/usr/local/yum/maven/bin
```

```
source /etc/profile
```

5. 확인

```
mvn -v
```

Jenkins

1. 다운로드

```
# root에서 작업  
su -  
cd /root  
  
# 다운로드  
wget https://get.jenkins.io/war-stable/2.303.3/jenkins.war
```

2. 설치

```
# tomcat 실행 확인
systemctl status tomcat

mv jenkins.war /usr/local/yum/tomcat/webapps
```

3. 확인

```
cd /usr/local/douzone/tomcat/webapps/
ls -la
```

jenkins.war 파일이 자동으로 풀려있는 것을 볼 수 있음! -> tomcat이 해줌!!

```
drwxr-x---. 11 root root    4096 Nov 29 04:02 jenkins
-rw-r--r--.  1 root root 72292947 Aug 25 20:45 jenkins.war
drwxr-x---.  6 root root    114 Nov 29 02:52 manager
```

4. 브라우저로 접근

http://{외부 IP 주소}/jenkins

5. Jenkins 시작 전 설정

- Administrator password

```
cd /root/.jenkins/secrets
cat initialAdminPassword
```

출력 되는 Password를 브라우저에 입력 후 Continue

- Customize Jenkins

Install suggested plugins 선택

- Create First Admin User

계정명, 암호 jenkins로 통일해도 되고... 알아서 ...

설정하고 Save and Continue!

6. Jenkins 관리 설정

Jenkins 관리 > Global Tool Configuration

JDK

Add JDK > Install automatically 체크 해제

Name: java1.8

JAVA_HOME: /usr/local/yum/java1.8

-- 앞에 띄어쓰기 안 되게 조심!

Git

Add Git > Install automatically 체크 해제

Name: git

Path to Git executable: /usr/local/yum/git/bin/git