

# AWS 사용방법

AWS 계정 생성 방법

AWS EC2 인스턴스 생성

AWS EC2 서버 접속

EC2 서버 Setting

git 설치

Java 설치

MySql 설치

Mysql 접속 후 계정 생성

외부 접속 허용을 위한 mysqld.cnf 파일 수정

AWS 인바운드 규칙에 mysql 포트 번호 추가

우분투 방화벽(ufw) 설정

git clone

프로젝트 build

Maven 설치

tomcat9 설치

war 파일 이름 바꾸기

war 파일을 webapp 밑으로 이동

war 파일 옮겨졌는지 확인

Tomcat9 실행

포트 변경 80


Tomcat9 재실행

이슈 해결(권한 부여-파일 업로드)

## AWS 계정 생성 방법

무료 클라우드 컴퓨팅 서비스 - AWS 프리 티어

AWS 플랫폼, 제품 및 서비스를 무료로 체험해 보세요 AWS Activate 는 자격이 있는 스타트업에 AWS 서비스에서 사용 가능한 무료 AWS Credit과 AWS Support를 비롯한 리소스 호스팅을 제공합니다. 지금

 [https://aws.amazon.com/ko/free/?trk=fa2d6ba3-df80-4d24-a453-bf30ad163af9&sc\\_channel=ps&s\\_kwcid=AL!4422!3!563761819834!e!!g!!aws&ef\\_id=Cj0KCQiA7bucBhCeARIsAlOwr-\\_vlyW6bhICa97VN5ddGE9wXlwt4jwQPAtsyNBSG0F\\_FDUfWCLZjU0aAv6KEALw\\_wcB:G:s&s\\_kwcid=AL!4422!3!563761819834!e!!g!!aws&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=\\*all&awsf.Free%20Tier%20Categories=\\*all](https://aws.amazon.com/ko/free/?trk=fa2d6ba3-df80-4d24-a453-bf30ad163af9&sc_channel=ps&s_kwcid=AL!4422!3!563761819834!e!!g!!aws&ef_id=Cj0KCQiA7bucBhCeARIsAlOwr-_vlyW6bhICa97VN5ddGE9wXlwt4jwQPAtsyNBSG0F_FDUfWCLZjU0aAv6KEALw_wcB:G:s&s_kwcid=AL!4422!3!563761819834!e!!g!!aws&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all)





## 로그인

### ☒ 루트 사용자

무제한 액세스 권한이 필요한 작업을 수행하는 계정 소유자입니다. [자세히 알아보기](#)

### ☐ IAM 사용자

일일 작업을 수행하는 계정 내 사용자입니다. [자세히 알아보기](#)

루트 사용자 이메일 주소

username@example.com

다음

계속 진행하는 경우 [AWS 고객 계약](#) 또는 [AWS 서비스에 대한 기타 계약 및 개인 정보 보호 정책](#)에 동의하게 됩니다. 이 사이트는 필수 쿠키를 사용합니다. 자세한 내용은 [쿠키 고지](#)를 참조하세요.

————— AWS를 처음 사용하십니까? —————

[AWS 계정 새로 만들기](#)



© 2022, Amazon Web Services, Inc. 또는 자회사. All rights reserved.

한국어 ▼

**주의 사항** 주소 및 우편번호는 영어로 입력해야 한다!

## 프리 티어 혜택

모든 AWS 계정은 사용하는 제품에 따라 3가지 유형의 무료 혜택을 탐색할 수 있습니다.



**항상 무료**  
만료되지 않음



**12개월 무료**  
최초 가입 날짜부터 시작



**평가판**  
서비스 활성화 날짜부터 시작

## AWS에 가입

### 연락처 정보

AWS를 어떻게 사용할 계획이신가요?

- ☐ 비즈니스 - 업무, 학교 또는 조직의 경우  
☐ 개인 - 자체 프로젝트의 경우

이 계정에 대해 누구에게 문의해야 하나요?

전체 이름

전화 번호

 +1 ▼

222-333-4444

국가 또는 리전

미국 ▼

주소

아파트, 동, 호수, 빌딩, 층 등

시

시, 도 또는 리전

우편 번호

☐ [AWS 이용약관](#)을 읽었으며 이에 동의합니다



## AWS에 가입

### Support 플랜 선택

비즈니스 또는 개인 계정에 대한 Support 플랜을 선택합니다. [플랜 및 요금 예시를 비교](#) 해 보세요. 언제든지 AWS Management Console에서 플랜을 변경할 수 있습니다.

#### ☒ 기본 지원 - 무료

- AWS를 처음 시작하는 신규 사용자에게 권장
- AWS 리소스에 대한 연중무휴 24시간 셀프 서비스 액세스
- 계정 및 청구 문제 전용
- Personal Health Dashboard 및 Trusted Advisor에 대한 액세스



#### ☐ 개발자 지원 - 시작가는 29 USD/월

- AWS를 체험해보는 개발자에게 권장
- 업무 시간 중 AWS Support에 대한 이메일 액세스
- 12시간(업무 시간 기준) 이내의 응답 시간



#### ☐ 비즈니스 지원 - 시작가는 100 USD/월

- AWS 기반 프로덕션 워크로드 실행에 추천
- 이메일, 전화 및 채팅을 통한 연중무휴 24시간 기술 지원
- 1시간 이내의 응답 시간
- Trusted Advisor 모범 사례 권장 사항 전체 세트



엔터프라이즈 수준의 지원이 필요하신가요?

최저 월 15,000 USD로 15분 이내에 응답을 받을 수 있으며 기술 지원 관리자가 배정된 컨시어지 스타일의 서비스를 이용할 수 있습니다. [자세히 알아보기](#)

가입 완료



## 로그인

### ☒ 루트 사용자

무제한 액세스 권한이 필요한 작업을 수행하는 계정 소유자입니다. [자세히 알아보기](#)

### ☐ IAM 사용자

일일 작업을 수행하는 계정 내 사용자입니다. [자세히 알아보기](#)

루트 사용자 이메일 주소

username@example.com

다음

계속 진행하는 경우 [AWS 고객 계약](#) 또는 [AWS 서비스에 대한 기타 계약 및 개인 정보 보호 정책](#)에 동의하게 됩니다. 이 사이트는 필수 쿠키를 사용합니다. 자세한 내용은 [쿠키 고지](#)를 참조하세요.

— AWS를 처음 사용하십니까? —

[AWS 계정 새로 만들기](#)

## AWS로 백업 및 복원

AWS를 사용하여 확장 가능하고 내구성이 있으며 안전한 데이터 보호 솔루션을 구축하세요.

[자세히 알아보기 >](#)



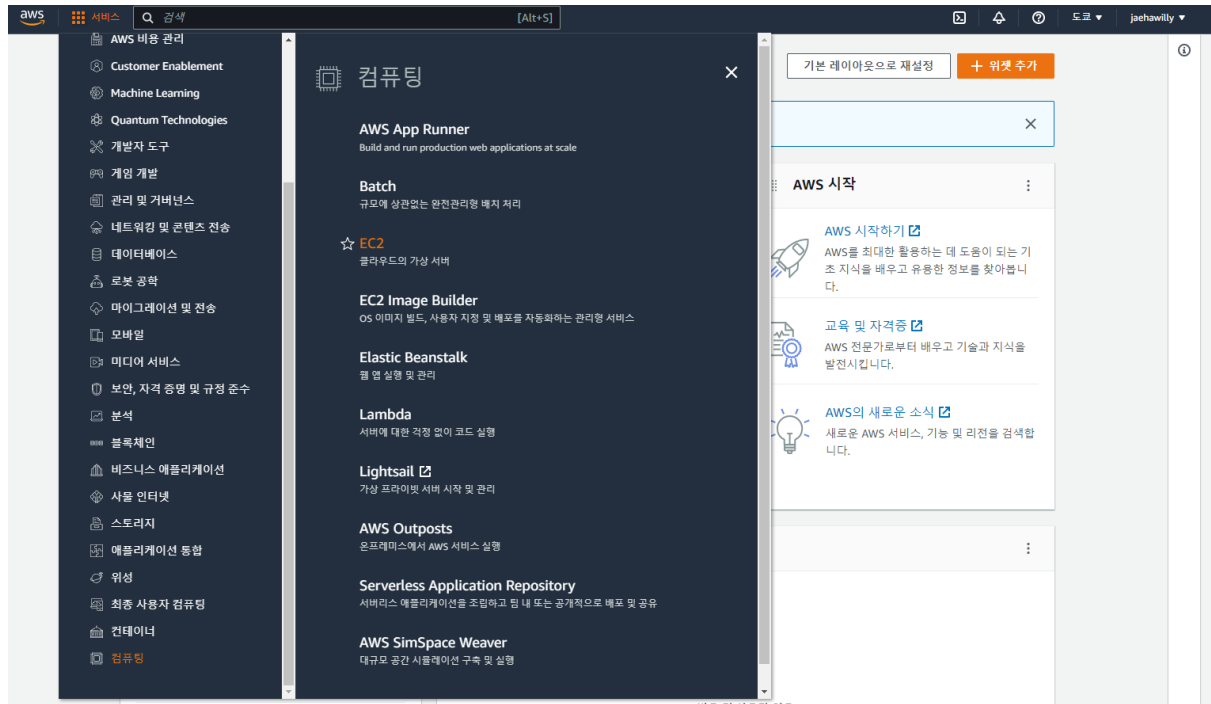
© 2022, Amazon Web Services, Inc. 또는 자회사. All rights reserved.

한국어 ▼

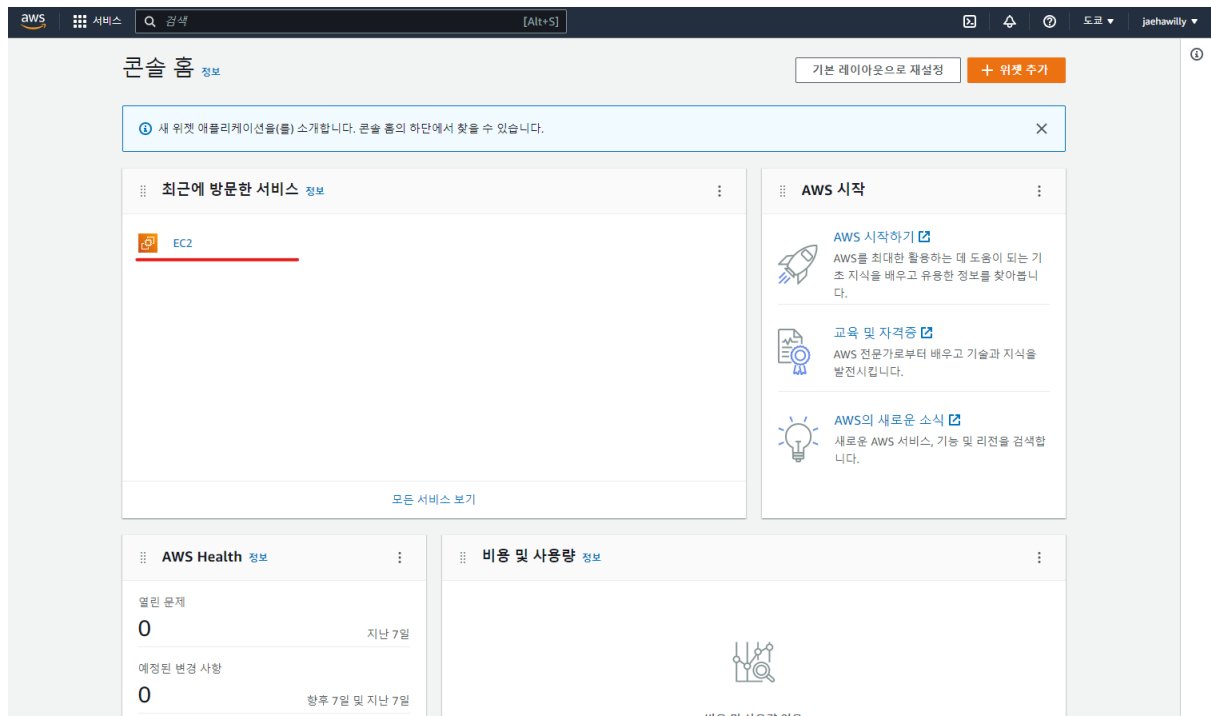
로그인하게 되면 이렇게 콘솔 홈이 보이게 된다!

# AWS EC2 인스턴스 생성

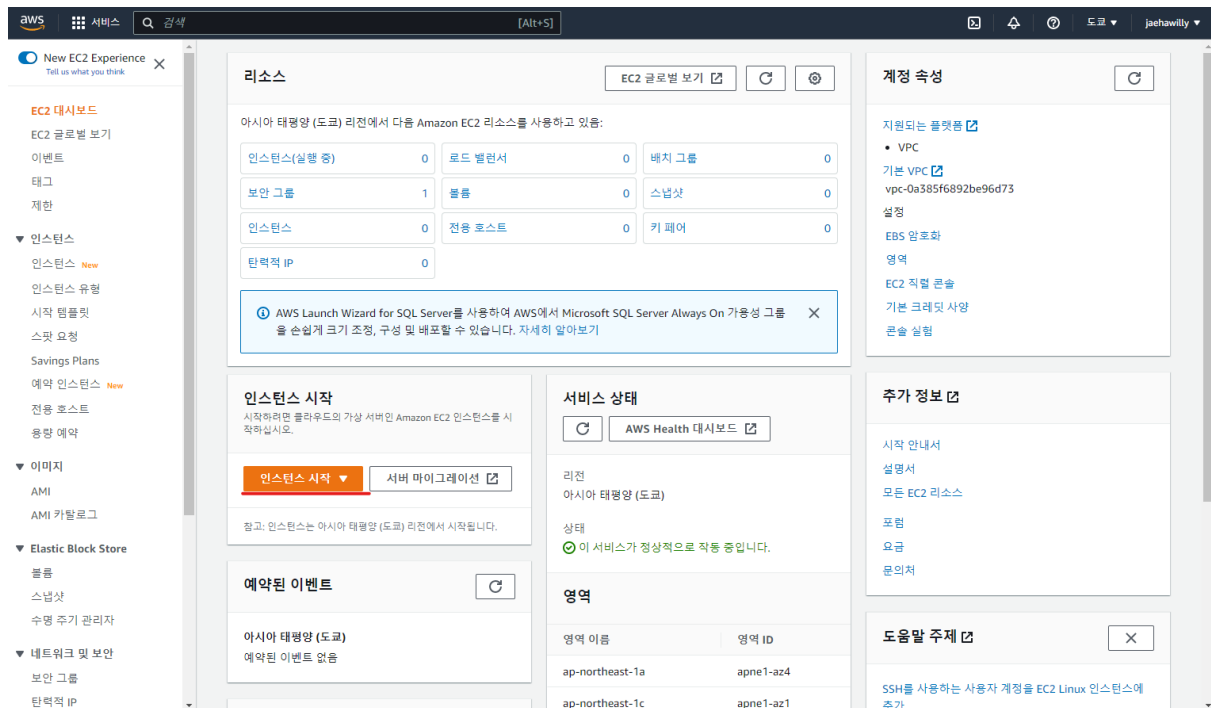
참고 : <https://sookmyunglion.tistory.com/8>



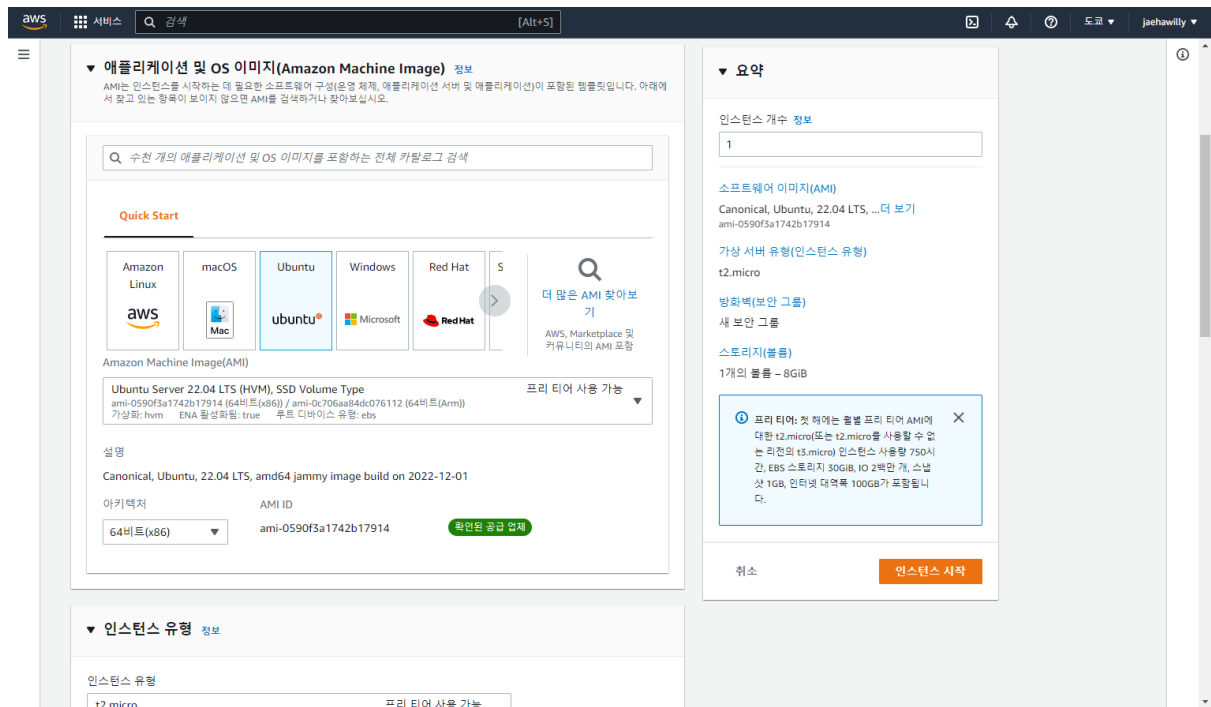
서비스 → 컴퓨팅 → EC2 클릭



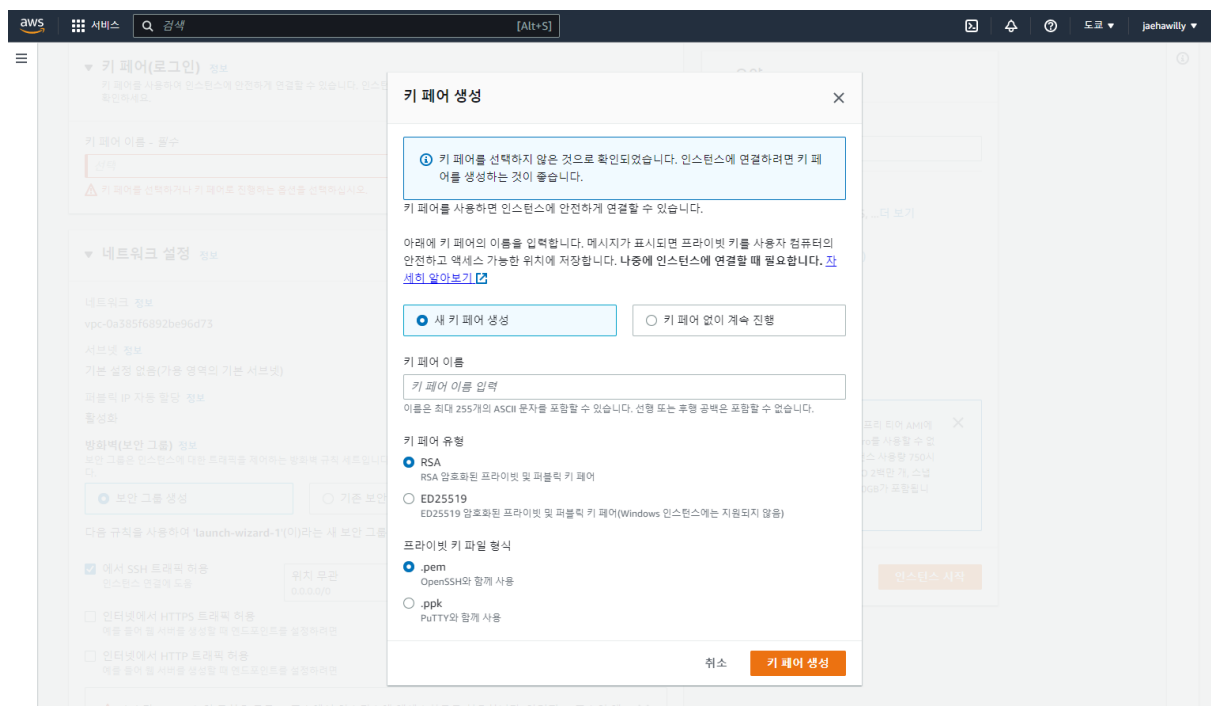
## EC2 클릭하면



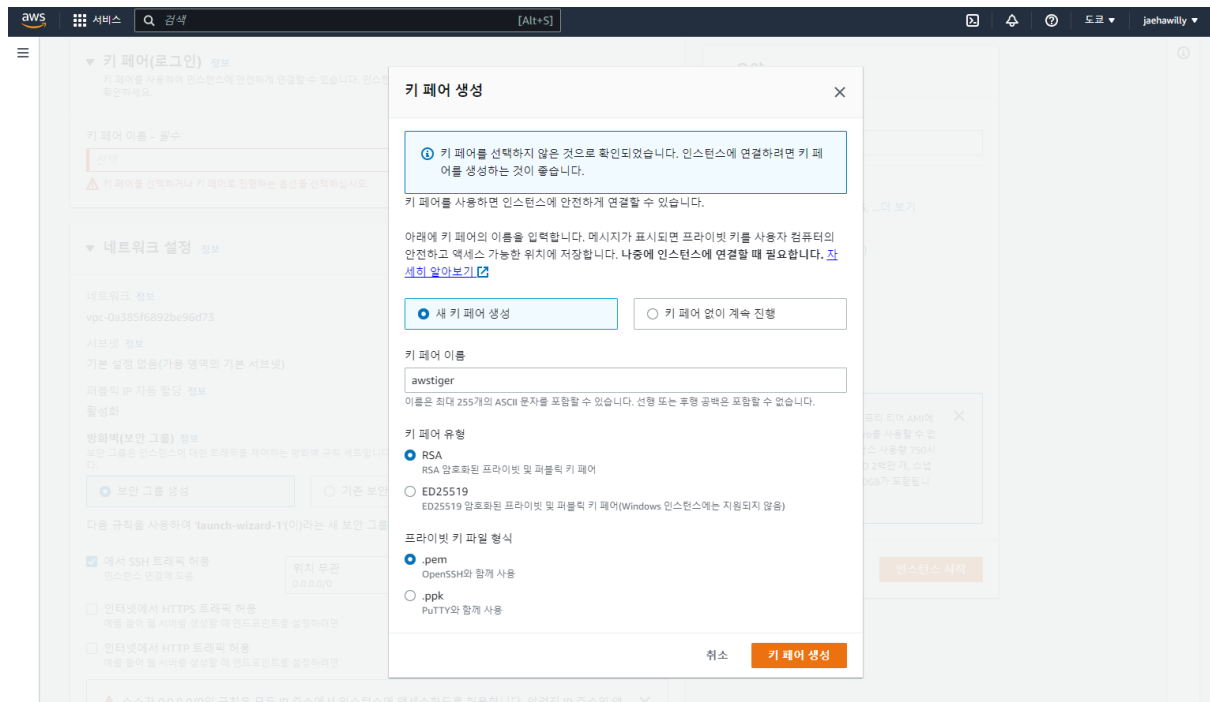
## 인스턴스 시작 클릭!



## Ubuntu 선택하고 인스턴스 시작 누르기!

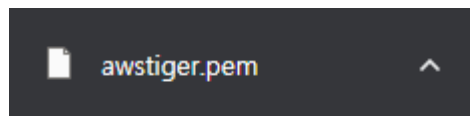






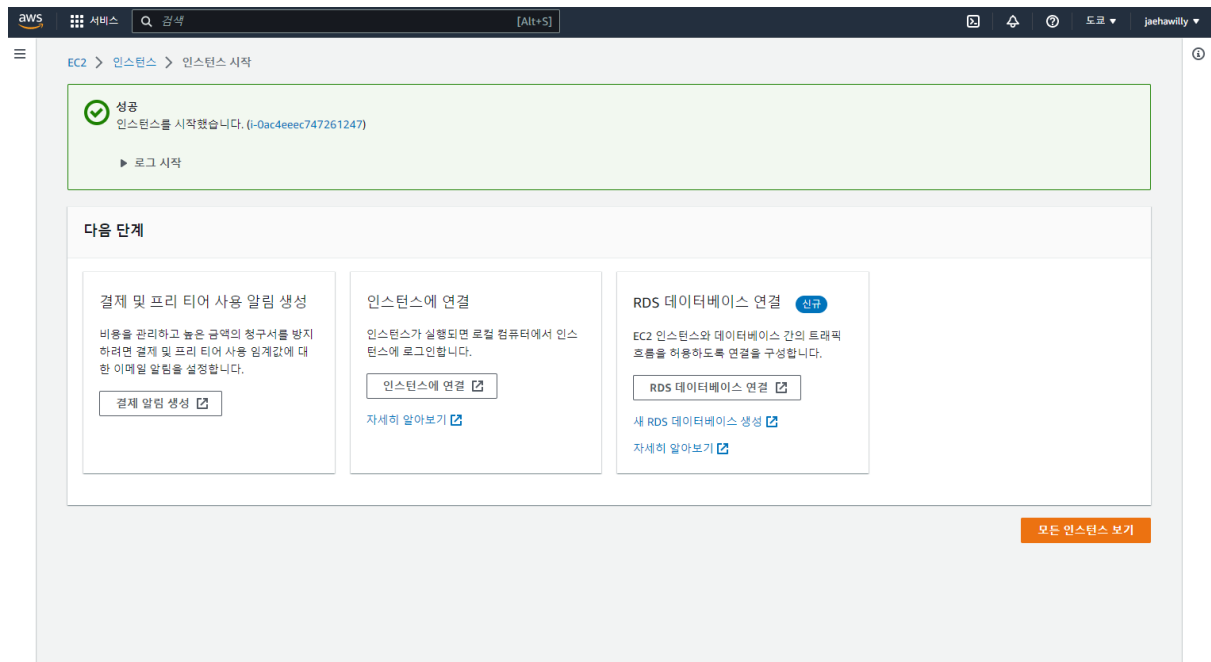
키페어 생성!

(awstiger)



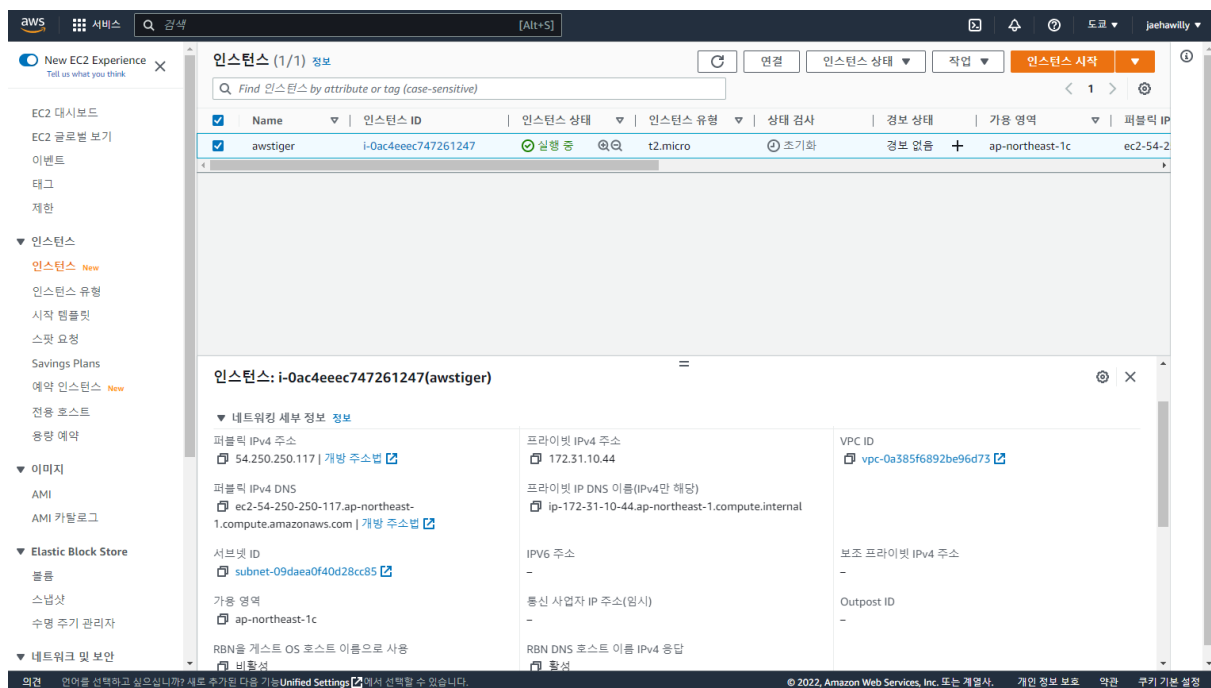
awstiger.pem이 다운로드 되었다

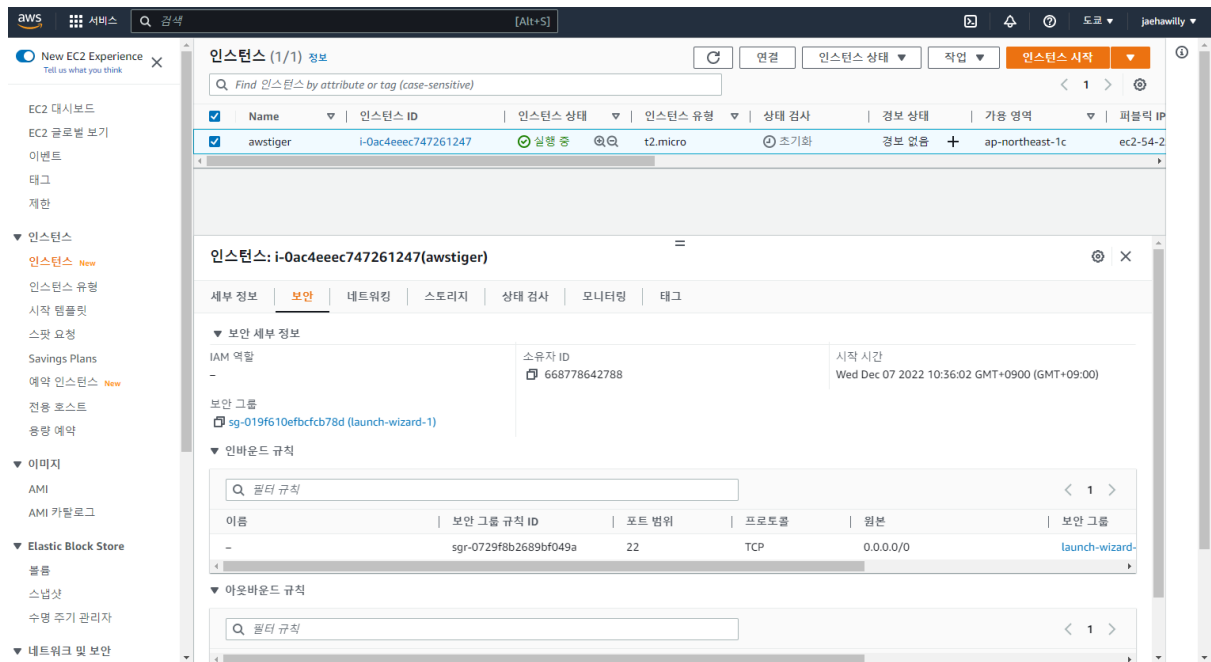
이름 지정해서 인스턴스 시작을 했더니



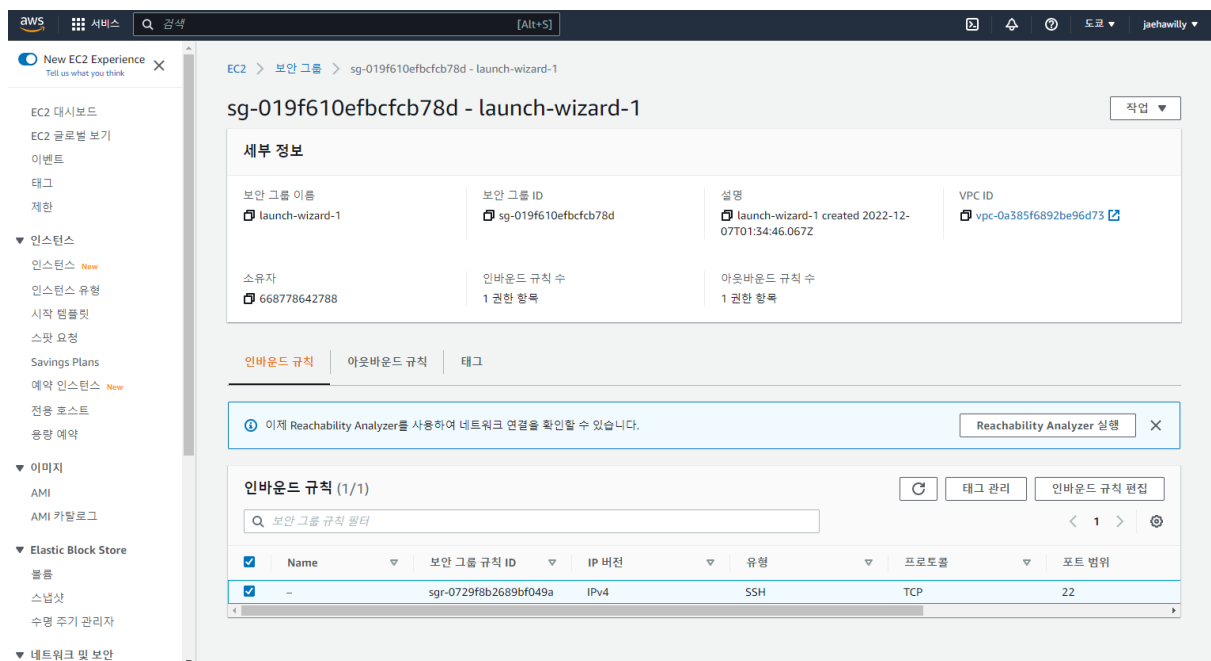
인스턴스 시작되었다는 창이 뜨게 되었다

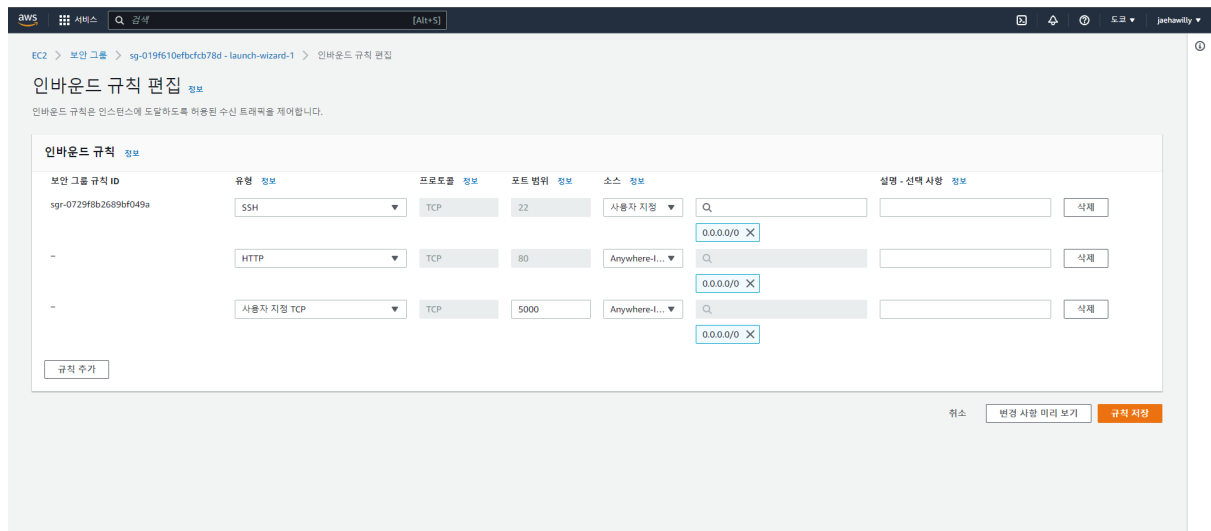
인스턴스에 가서 만든 인스턴스에 체크를 하게 되면 이렇게 뜬다





보안그룹에 있는 링크를 클릭하면 다음과 같이 뜨게 되고, 인바운드 규칙 편집을 누르면 된다

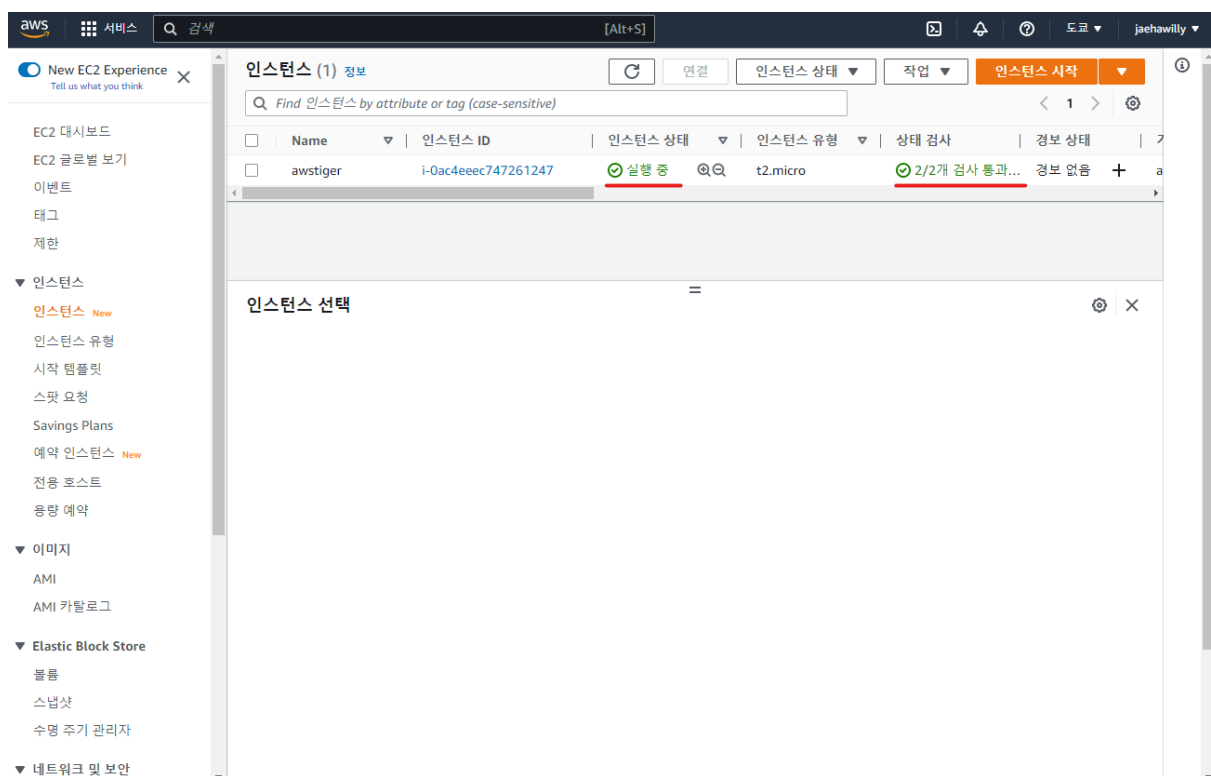




이렇게 포트 포워딩한다

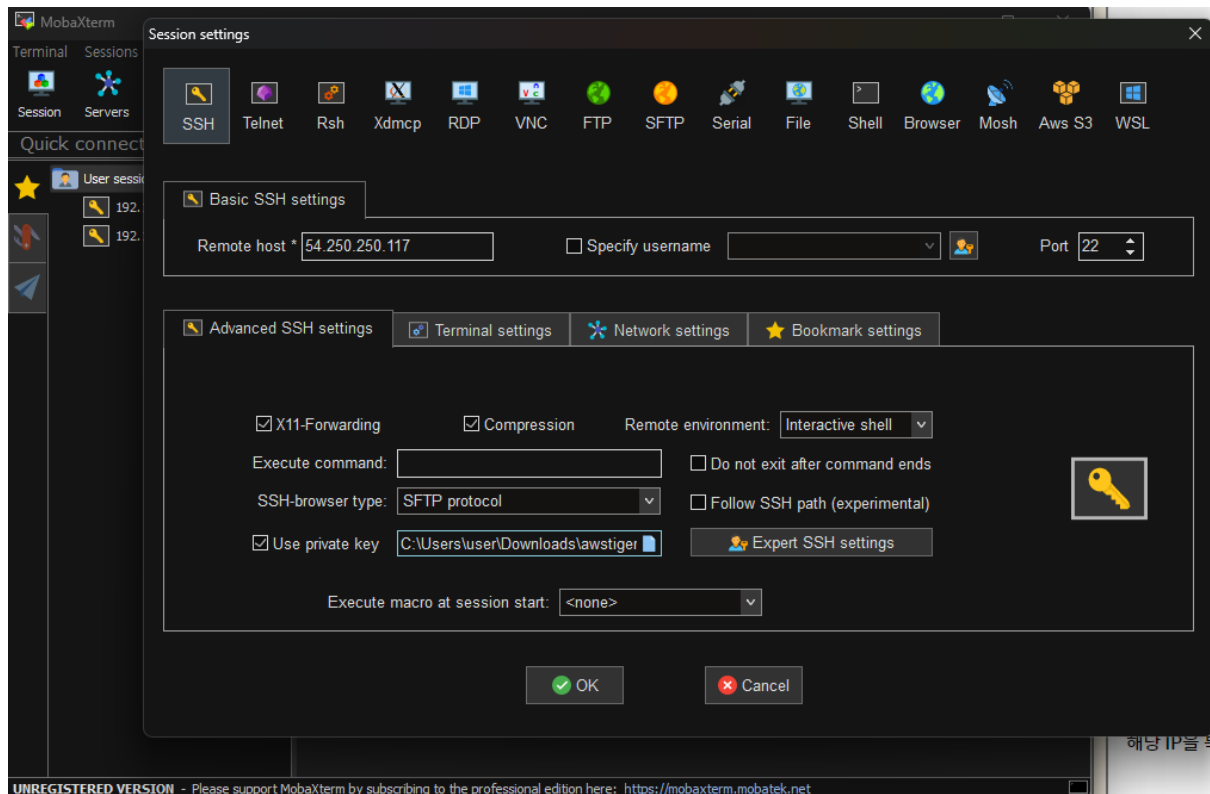
참고 : <https://velog.io/@green9930/AWS로-사이트-배포하기-1-AWS-서버-세팅하기>

인스턴스 확인하기!(인스턴스 상태, 상태 검사) 초기화해서 확인해보기



참고 : <https://victorydntmd.tistory.com/62>

## AWS EC2 서버 접속

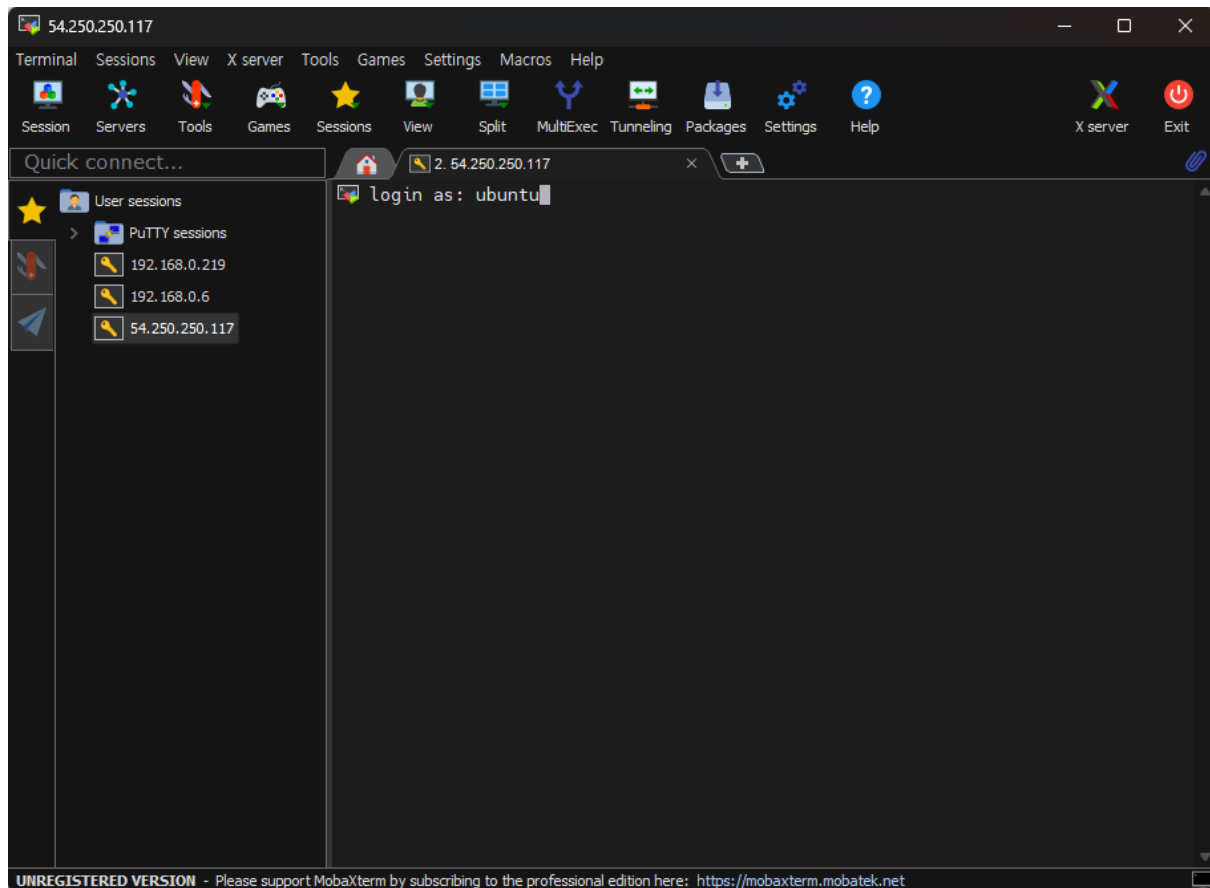


포트 번호에 "IPv4 퍼블릭 IP" 추가하고, Use private key에 아까 다운로드 받았던 pem 파일을 넣는다

세션 접속 후 username 입력

인스턴스 생성 시 ubuntu로 생성한 경우 default username은 **ubuntu**이다.

cf. linux로 생성한 경우 ec2-user



참고 : <https://cococon1787.tistory.com/850>

## EC2 서버 Setting

### 계정 이동 명령어

root 이동 → `sudo su`

ubuntu 이동 → `sudo su ubuntu`

### git 설치

`sudo apt-get install -y git`

### Java 설치

`java --version`

`sudo apt install [open jdk]`

### MySql 설치

```
sudo apt-get install mysql-server
```

## Mysql 접속 후 계정 생성

```
create user '[계정명]'@[호스트 ip 주소] identified by '[비밀번호]';  
grant [할당 권한] on [데이터베이스명].[테이블명] to '[계정명]'@[호스트 ip 주소] with grant option;  
show grants for '[계정명]'@[호스트 ip 주소];
```

```
$ sudo mysql -u root -p
```

```
mysql> create user 'tiger'@'%' identified by '1234';  
mysql> grant all privileges on *.* to 'tiger'@'%' with grant option;  
mysql> show grants for 'tiger'@'%';
```

```
mysql> flush privileges;
```

```
mysql> show databases;  
mysql> use mysql;  
mysql> select user, host from user;
```

## 외부 접속 허용을 위한 mysqld.cnf 파일 수정

```
$ cd /etc/mysql/mysql.conf.d  
$ sudo vi mysqld.cnf
```

```

#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 13306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir             = /tmp
#
# 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
mysqlx-bind-address = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack       = 256K
#
# thread_cache_size  = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
mysam-recover-options = BACKUP

# max_connections    = 151
#
# table_open_cache    = 4000

#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
# general_log_file    = /var/log/mysql/query.log
:wq!

```

위의 그림처럼 주석 처리되어 있던 port 설정 부분을 원하는 port로 변경한다. 그리고 bind-address 를 주석 처리 하거나 0.0.0.0으로 변경한다.

이때 port는 mysql 기본 포트 3306이 아닌 포트로 설정하는 것이 좋다. 윈도우 로컬 환경에서 3306 포트를 사용하고 있을 확률이 높기 때문이다.

bind-address란 데이터베이스 서버가 어떤 주소로의 요청을 허용할 것인지 설정하는 것이다. 그래서 0.0.0.0/0을 설정하게 되면 네트워크 전체 대역, 즉 모든 트래픽에 대해서 요청을 허용한다는 뜻이기에 문제없이 원격 커넥션이 이루어진다.

## AWS 인바운드 규칙에 mysql 포트 번호 추가

mysql 포트 번호 추가

## 우분투 방화벽(ufw) 설정



```
sudo service mysql restart
sudo ufw allow out 13306/tcp
sudo ufw allow in 13306/tcp
sudo service mysql restart
```

## git clone

- clone 하기 전 aws 서버 내 폴더 생성

```
mkdir apps
```

```
cd apps/
```

- git clone 과정

```
54.250.250.117
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ubuntu/
Name
..
.cache
.ssh
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.xauthority
Cloning into 'AWS-Project' ...
Username for 'https://github.com': jaehawilly
Password for 'https://jaehawilly@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended mode
s of authentication.
fatal: Authentication failed for 'https://github.com/PoscoAWS/AWS-Project.git/'
ubuntu@ip-172-31-10-44:~/apps$ ^C
ubuntu@ip-172-31-10-44:~/apps$ git clone https://github.com/PoscoAWS/AWS-Project.git
fatal: destination path 'AWS-Project' already exists and is not an empty directory.
ubuntu@ip-172-31-10-44:~/apps$ git clone https://github.com/PoscoAWS/AWS-Project.git
fatal: destination path 'AWS-Project' already exists and is not an empty directory.
ubuntu@ip-172-31-10-44:~/apps$ ls
Command 'lsd' not found, but can be installed with:
sudo snap install lsd
ubuntu@ip-172-31-10-44:~/apps$ ls
AWS-Project
ubuntu@ip-172-31-10-44:~/apps$ rmdir AWS-Project
rmdir: failed to remove 'AWS-Project': Directory not empty
ubuntu@ip-172-31-10-44:~/apps$ ls
ubuntu@ip-172-31-10-44:~/apps$ cd ../
ubuntu@ip-172-31-10-44:/$ cd ..
ubuntu@ip-172-31-10-44:/$ cd apps
bash: cd: apps: No such file or directory
ubuntu@ip-172-31-10-44:/$ cd ..
ubuntu@ip-172-31-10-44:/$ cd apps
bash: cd: apps: No such file or directory
ubuntu@ip-172-31-10-44:/$ sudo su
root@ip-172-31-10-44:/# sudo su ubuntu
ubuntu@ip-172-31-10-44:/$ ls
bin  dev  home  lib32  libx32  media  opt  root  sbin  srv  tmp  var
boot  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
ubuntu@ip-172-31-10-44:/$ cd ~
ubuntu@ip-172-31-10-44:~$ ls
apps
ubuntu@ip-172-31-10-44:~$ cd apps
ubuntu@ip-172-31-10-44:~/apps$ git clone https://github.com/PoscoAWS/AWS-Project.git
Cloning into 'AWS-Project' ...
Username for 'https://github.com': jaehawilly
Password for 'https://jaehawilly@github.com':
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 78 (delta 15), reused 66 (delta 3), pack-reused 0
Receiving objects: 100% (78/78), 9.13 KiB | 667.00 KiB/s, done.
Resolving deltas: 100% (15/15), done.
ubuntu@ip-172-31-10-44:~/apps$
```

# 프로젝트 build

Maven 프로젝트이므로 Maven 설치가 필요하다.

## Maven 설치

```
sudo apt update
```

```
sudo apt install maven
```

```
mvn -v
```

ubuntu 계정에서

```
cd /apps/AWS-Project/ 에서 pom.xml 있는 경로 확인하고
```

```
mvn package(dependency 다운로드)
```

```
cd /apps/AWS-Project/target/ 으로 가서 war 파일 생성된 것 확인
```

## tomcat9 설치

```
sudo apt install tomcat9
```

## war 파일 이름 바꾸기

```
mv [기존war파일 이름] [바꾸고 싶은 war파일 이름]
```

## war 파일을 webapp 밑으로 이동

```
mv [war파일] /var/lib/tomcat9/webapps
```

## war 파일 옮겨졌는지 확인

```
cd /var/lib/tomcat9/webapps
```

## Tomcat9 실행

```
systemctl start tomcat9
```

```
systemctl status tomcat9
```

## 포트 변경 80

```
cd /var/lib/tomcat9/conf
```

```
vi server.xml
```

포트 번호를 8080에서 80으로 변경

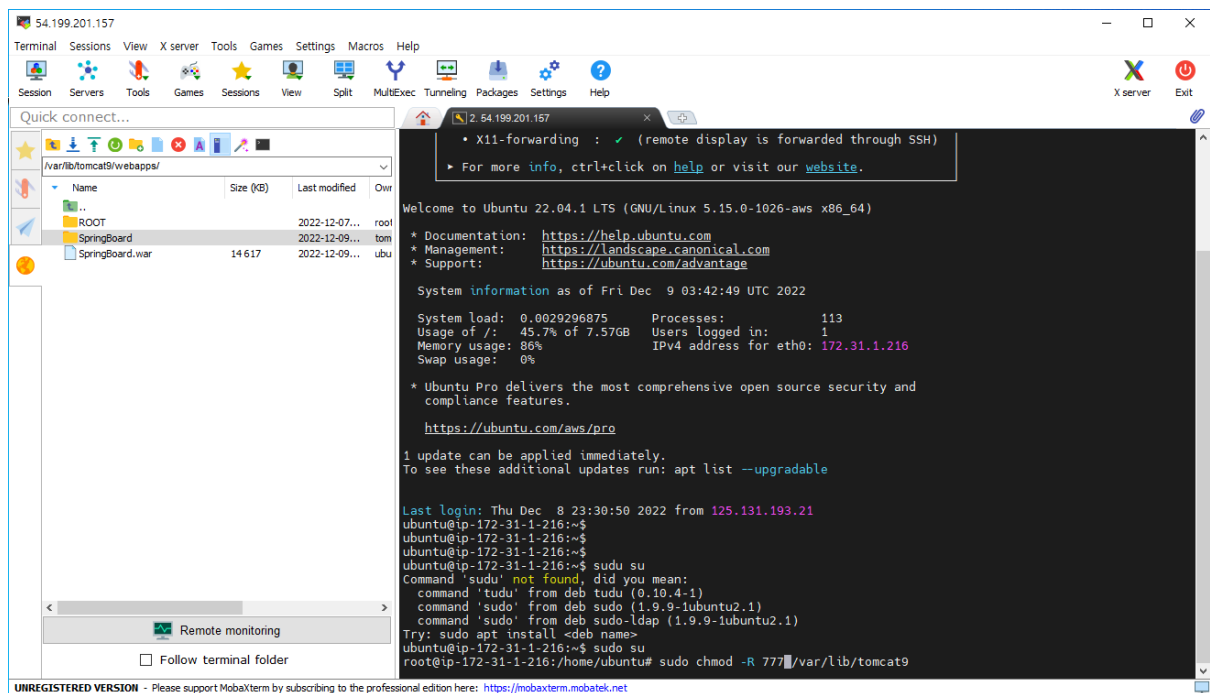
## Tomcat9 재실행

systemctl restart tomcat9

systemctl status tomcat9

최종 확인 : <http://54.250.250.117/Spring01-0.0.1-SNAPSHOT/> 이동

## 이슈 해결(권한 부여-파일 업로드)



[공유 파일]

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e265512f-6595-46ba-95c0-853085158071/awstiger.pem>