

aws.amazon.com 접속 후 무료계정으로 가입!

프리티어 EC2 인스턴스 생성

1. putty 로 우분투 접속

아이디 : ubuntu

2. root 패스워드 설정

sudo passwd root

1111

.. 한번 더...^^

3. root 로 로그인 하자

su root

1111

4. apt-get update

5. apt-get upgrade

6. 시간설정

dpkg-reconfigure tzdata

gui 환경이 나오면 Asia 선택 후 seoul 선택

7. 자바 설치

```
apt-get install openjdk-17-jre-headless
```

```
apt-get install openjdk-17-jdk
```

8. 자바 버전확인

```
java -version
```

9. 자바 어디설치 됐는지 확인해보자

```
which java
```

/usr/bin/java 라고 나타남!.

10. 톰캣 설치

```
apt-get install tomcat10
```

11. 확인해보자

```
http://엘라스틱 ip:8080
```

안되면 AWS 콘솔에서 8080 포트를 열어야 한다.

아마존 클라우드 콘솔창으로 들어가서 아래와 같이 설정해준다.

해당 인스턴스 선택한 후 화면 아래쪽에 보면 [보안 그룹] 항목에 있는 파랑글씨 클릭한다.

그리고 [인바운드] 탭을 클릭한 후 편집을 선택! 그리고 [사용자 지정 TCP 규칙]을 선택한 후 포트를 [8080]으로 지정함!

그리고 다시 브라우저에서 인스턴스 IP:8080 으로 접속!

12. 톰캣이 설치된 곳을 찾아보자!

```
find / -name tomcat10
```

이렇게 하면 찾아지는데 너무 많이 나와서 보기 불편해서 정리했다.

?

톰캣 기본 위치

HOME : /usr/share/tomcat10

CONF : /etc/tomcat10

LOG : /var/log/tomcat10

ROOT : /var/lib/tomcat10/webapps/ROOT

```
# cd /etc/tomcat10
```

```
Tomcat10/# ls
```

```
... server.xml, context.xml ....
```

```
tomcat8/# vi server.xml
```

13. 톰캣 재시작

```
service tomcat10 restart
```

14. 톰캣 중지

```
service tomcat10 stop
```

15. 톰캣 시작

```
service tomcat10 start
```

16. 확인

```
apt-get install net-tools
```

```
netstat -ntl
```

17. mysql 설치하자

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

가장 최신 버전이 설치됨

```
apt-get install mysql-client
```

가장 최신 버전이 설치됨.

만약에 원하는 버전이 있으면 뒤에 apt-get install mysql-client-8

이런식으로 버전 적어주면 된다.

18. mysql 버전 확인

```
mysql -V
```

:: mysql 완전 삭제하는 법

```
apt-get remove --purge mysql*  
rm -rf /etc/mysql /var/lib/mysql  
rm -rf /var/log/mysql  
rm -rf /var/log/mysql.*  
rm /var/lib/dpkg/info/*  
apt-get autoremove  
apt-get autoclean
```

mysql 재 설치(그냥 위에서 설치하는 명령어를 그대로 입력하면 된다.)

```
apt-get install mysql-server --fix-missing --fix-broken
```

19. mysql 언어셋과 원격접속 허용을 해주자.

```
vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

다음의 ip 가 명시된 곳을 수정해야 한다.

```
bind-address = 127.0.0.1 이부분을 #으로 주석처리
```

파일 열어서 가장 아래에 적어주면 된다.

```
character-set-server = utf8mb4
```

```
collation-server = utf8mb4_unicode_ci
```

MySQL 서버 다시 시작

```
service mysql restart
```

vi 첨 써보신다면 i 를 누르면 입력모드가 되는데

저걸 젤 밑에 붙여넣고

esc 를 누르고 입력모드를 해제한뒤

: 를 누르고

wq 를 누르고 enter

w 누르고 enter 하면 저장

q!누르고 enter 하면 그냥 나가는거

wq 누르고 enter 하면 저장하고 나가는거임.

만약에 뭔가 잘못쳐서 자꾸 오류나서 :q! 로 강제종료 하면 됨.

20. JDBC 드라이버 설치해주자.

```
apt install libmariadb-java
```

/usr/share/java/mysql-connector-java.jar 위치에 connector 가 설치된다.

21. jdk 와 tomcat 을 jdbc 드라이버로 연결해주자.

```
ln -s /usr/share/java/mysql-connector-java.jar /usr/share/tomcat10/lib/mysql-connector-java.jar
```

22. 톰캣 재시작해준다.

```
service tomcat10 restart
```

23. mysql db 와 테이블 생성하고 데이터 한개 insert 하기

```
mysql -u root -p
```

```
1111
```

```
show databases;
```

데이터베이스 생성

```
mysql> create database test_db default character set utf8;
```

```
Query OK, 1 row affected (0.00 sec)
```

데이터베이스 변경

```
mysql> use test_db;
```

특정 계정에 특정 DB 권한 부여
예)

```
mysql> create user test@'%' identified by '1111';(뒤는 패스워드임)
```

```
mysql> grant all privileges on test_db.* to test_admin@'%' with grant option;
```

```
mysql> flush privileges;
```

빠져나오기

```
mysql> quit
```

Git 설치

```
# sudo apt-get install git
```

다음은 cd 명령으로 톰캣의 ROOT 로 이동한다.

```
# cd /var/lib/tomcat10/webapps/ROOT
```

이제 이동한 폴더에 GitLab 의 응용프로그램을 배포받자!

```
# git clone https://github.com/deopard/fc-sample-project-a.git
```

이때 파일들을 상위 폴더로 이동하려면

```
# mv * ../
```

특정 폴더를 삭제할 때는

```
# rm -r {폴더명}
```

WinSCP설치 / mobaXterm

이건 로컬에 있는 파일을 서버(원격)로 복사할 때 편하게 해주는 툴이다.

우선 <http://winscp.net/eng/download.php> 에서 다운 받아 압축해제 한다.

그리고 PuTTY로 서버에 접속하여 ssh터미널의 환경설정을 변경하자! 이렇게 하는 이유는 WinSCP툴은 ubuntu계정으로만 로그인 되어 파일을 복사하거나 삭제할 수 없으므로 root계정으로 접속할 수 있도록 하기 위해서다.

root계정으로 외부에서 접속할 수 있도록 하는 설정 -----

```
$ vi /etc/ssh/sshd_config 또는 nano /etc/ssh/sshd_config
```

PermitRootLogin without-password 라인을 아래와 같이 수정

PermitRootLogin yes

StrictModes yes 라인을 아래와 같이 수정(주석처리로 비활성화. 앞에 #만 붙이면 됨)

#StrictModes yes

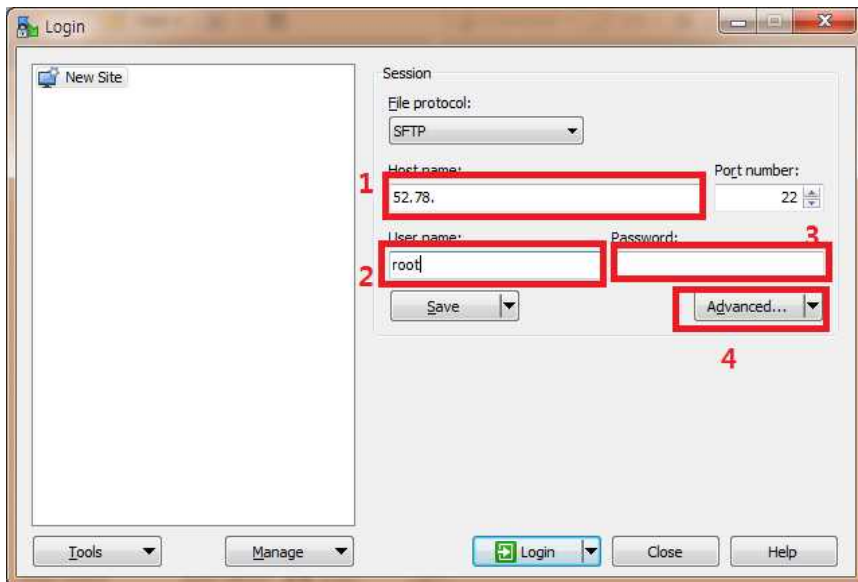
아마존 AWS 로 부터 받은 일반사용자 암호화키를 루트관리자 계정도 복사(cp 명령어 사용 : cp 는 copy 를 의미)

```
$ sudo cp /home/ubuntu/.ssh/authorized_keys /root/.ssh
```

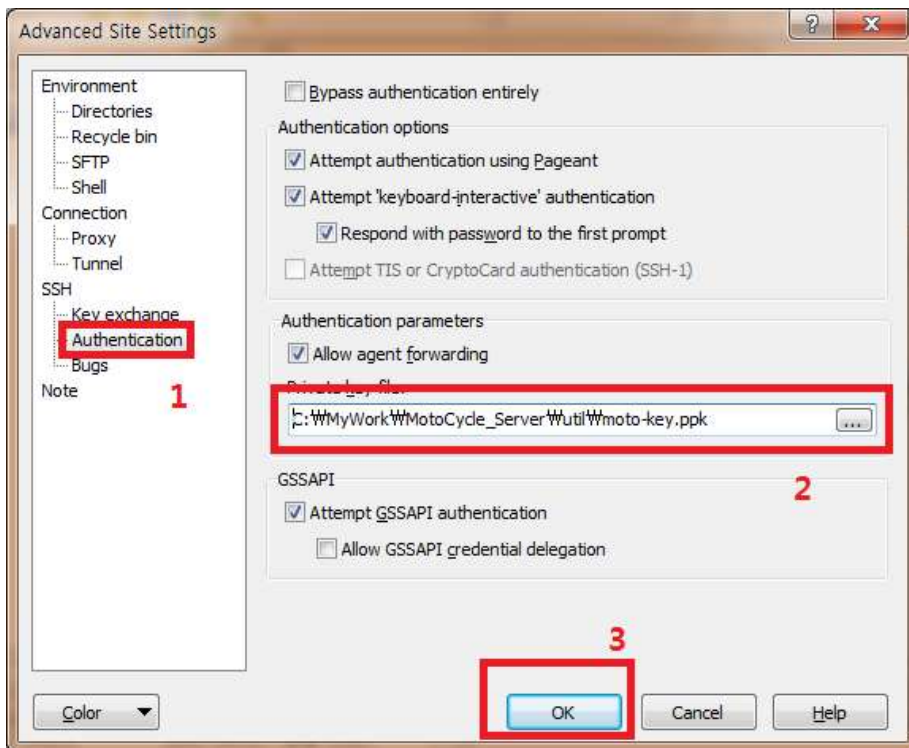
ssh 서비스 재시작

```
$ sudo service ssh restart
```

WinSCP 실행

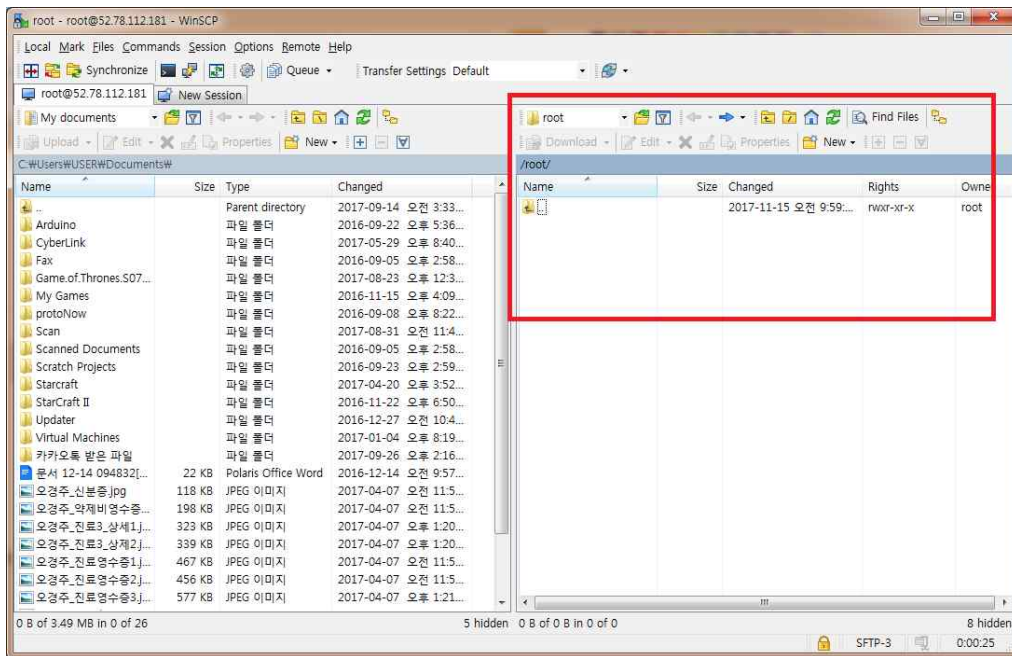


다음은 위의 4번째 선택한 Advanced...창의 내용이다.



2번에서 PuTTY-gen으로 만든 키를 지정하고 OK버튼을 선택!(PuTTY-gen 이 pem 파일을 ppk파일로 컨버팅하는 프로그램임)

그리고 뒤의 화면으로 돌아갔을 때 [Login]버튼을 클릭하여 접속한다.



리눅스 폴더 권한 부여하기

WinSCP로 우리가 소스를 배포하면 생성되는 디렉토리에 대한 권한은 root가 가지고 있을 것이다.

이대 게시판등에서 파일 업로드할 때 권한이 없어서 오류가 발생할 수 있으며 이것을 다음의 문장으로 리눅스에서 권한을 부여할 수 있다.

```
chmod -R a+rw /var/lib/tomcat10/webapps/ROOT/resources
```

nginx 설치(아파치2대신)

```
apt-get install nginx
```

설정파일 수정

```
vi /etc/nginx/sites-available/default
```

```
location / {
```

```
# First attempt to serve request as file, then
```

```
# as directory, then fall back to displaying a 404.
```

```
# try_files $uri $uri/ =404;
```

```
proxy_pass http://localhost:8080;

proxy_set_header X-Real-IP $remote_addr;

proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

proxy_set_header Host $http_host;

}
```

빠져나온 후 재 시작

service nginx restart

AWS인스턴스 가상으로 메모리 증설법:

블럭사이즈가 1MB, 블럭 갯수가 2000K 인 빈파일을 만듭니다. (2000000 = 2GB)

```
dd if=/dev/zero of=/root/swapfile bs=1k count=2000000 conv=excl
```

swapfile 의 권한을 생성합니다.

```
chmod 600 /root/swapfile
```

그리고 스왑 파일로 설정을 해줍니다.

```
mkswap /root/swapfile
```

스왑파일을 실행해줍니다.

```
swapon /root/swapfile
```

```
free -h
```

다음과 같이 swap 에 메모리가 증설된 것을 확인할 수 있다.

	total	used	free	shared	buff/cache	available
Mem:	965M	356M	89M	276K	519M	476M
Swap:	1.9G	11M	1.9G			

인스턴스를 재인식할때마다 swap 파일을 기동시켜줘야하는데 이 작업은 너무 번거롭기 때문에 fstab 파일에 값을 추가해주면 인스턴스를 재시작할때 자동으로 해당 swap 을 실행시켜준다.

```
vi /etc/fstab
```

vi 명령어를 이용하여 파일을 연 후 다음의 내용을 추가해 준다.

```
/root/swapfile swap swap auto 0 0
```

수동으로 NginX에 SSL적용!

우선 Certbot를 위한 repository(저장소)를 추가합니다.

```
$ sudo apt update
$ sudo apt upgrade
$ sudo add-apt-repository ppa:certbot/certbot
```

Cerbot 의 Nginx 패키지를 설치합니다.

```
$ sudo apt install -y certbot python3-certbot-nginx
```

NginX Configuration 을 설정

```
$ sudo vim /etc/nginx/sites-available/default
```

```
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
```

변경 전

76,0-1

67%

```
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name coiting.shop www.coiting.shop;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
```

변경 후

82,44-51

67%

그림처럼 server_name 항목을 찾아 뒤에 도메인명을 넣어준다.

이제 확인해보자!

```
$ sudo nginx -t
```

테스트 성공인 경우 다음과 같이 나타남!

```
[ubuntu@ip-1 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-1 ~$
```

이제 NginX와 Tomcat을 재 가동하자!

```
$ sudo service nginx restart
$ sudo service tomcat9restart
```

SSL인증서 받기

원하는 도메인 등록과 인증서 획득! 그리고 이메일 입력

```
$ sudo certbot --nginx -d 도메인 -d www.도메인
```

```
ubuntu@ip-10-10-10-10:~$ sudo certbot --nginx -d rising-bunjang.store -d www.rising-bunjang.store
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to cancel)
@gmail.com
```

이메일 입력

먼저 승인 Yes를 선택!

```
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: a
```

Let's Encrypt 이메일 리스트에 추가할거냐? 아니(No 선택)

```
-----
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: n
```

잘 마무리된 화면

```
[Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 1
Deploying certificate
Successfully deployed certificate for coiting.shop to /etc/nginx/sites-enabled/default
Successfully deployed certificate for www.coiting.shop to /etc/nginx/sites-enabled/default
Congratulations! You have successfully enabled HTTPS on https://coiting.shop and
https://www.coiting.shop

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
```

마지막으로 AWS콘솔로 로그인 하여 해당 인스턴스의 보안 그룹에서 [인바운드 규칙]을 하나 추가해야 한다. 추가할 항목을 HTTPS를 추가하면 끝~~~!