aws.amazon.com접속 후 무료계정으로 가입!

1. putty로 우분투 접속

아이디 : ubuntu

2. root패스워드 설정

sudo passwd root

1111

.. 한번 더...^^

3. root로 로그인 하자

su root

1111

4. apt-get update

5. apt-get upgrade

6. 시간설정

dpkg-reconfigure tzdata

gui환경이 나오면 Asia선택 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. 톰켓 설치

apt-get install tomcat9

11. 확인해보자

http://엘라스틱ip:8080

안되면 AWS콘솔에서 8080포트를 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 TCP규칙]을 선택한 후 포트를 [8080]으로 지정함!

그리고 다시 브라우저에서 인스턴스IP:8080으로 접속!

12. 톰켓이 설치된 곳을 찾아보자!

find / -name tomcat9

이렇게 하면 찾아지는데 너무 많이 나와서 보기 불편해서 정리했다.

?

톰켓 기본 위치

HOME : /usr/share/tomcat9

CONF : /etc/tomcat9

LOG : /var/log/tomcat9

ROOT : /var/lib/tomcat9/webapps/ROOT

# cd /etc/tomcat9

Tomcat9/# ls

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

tomcat8/# vi server.xml

13.톰켓 재시작

service tomcat9 restart

14.톰켓 중지

service tomcat9 stop

15.톰켓 시작

service tomcat9 start

16.확인

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

19. mysql 언어셋 해주자.

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

다음의 ip가 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

vi첨 써보신다면 i 를 누르면 입력모드가 되는데

저걸 젤 밑에 붙여넣고

esc를 누르고 입력모드를 해제한뒤

: 를 누르고

wq를 누르고 enter

w누르고 enter하면 저장

q누르고 enter하면 그냥 나가는거

wq누르고 enter하면 저장하고 나가는거임.

만약에 먼가 잘못쳐서 자꾸 오류나서 :q! 로 강제종료 하면 됨.

20. JDBC드라이버 설치해주자.

apt install libmariadb-java

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

21. jdk와 tomcat을 jdbc드라이버로 연결해주자.

ln -s /usr/share/java/mysql-connector-java.jar /usr/share/tomcat9/lib/mysql-connector-java.jar

22. 톰켓 재시작해준다.

service tomcat9 restart

23. mysql db와 테이블 생성하고 데이터 한개 insert하기

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 계정에게 특정 DB권한 부여  
예)

mysql> create user test\_admin@’%’ identified by 'test1111';(뒤는 패스워드임)

mysql> grant all privileges on test\_db.\* to test\_admin@’%’ with grant option;

mysql>flush privileges;

-------------------------------------------------------------------------------------------------------------------

빠져나오기   
mysql> quit

Git설치

# sudo apt-get install git

다음은 cd명령으로 톰켓의 ROOT로 이동한다.

# cd /var/lib/tomcat9/webapps/ROOT

이제 이동한 폴더에 GitLab의 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.

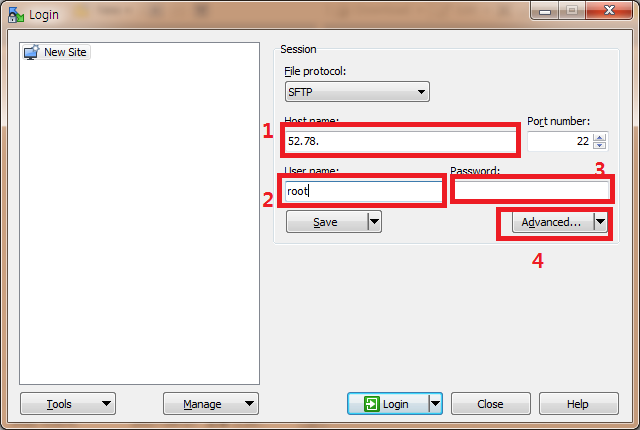
root계정으로 외부에서 접속할 수 있도록 하는 설정 ---------------------------------------------------------

$ 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

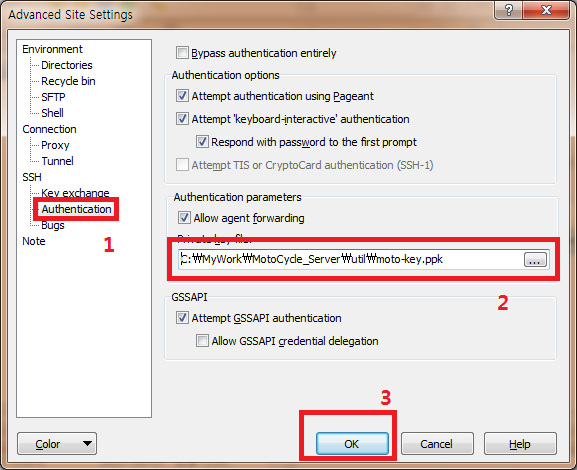
아마존 AWS로 부터 받은 일반사용자 암호화키를 루트관리자 계정에도 복사(cp 명령어 사용 : cp는 copy를 의미)  
 $ sudo cp /home/ubuntu/.ssh/authorized\_keys /root/.ssh

ssh 서비스 재시작  
 $ sudo service ssh restart

WinSCP실행

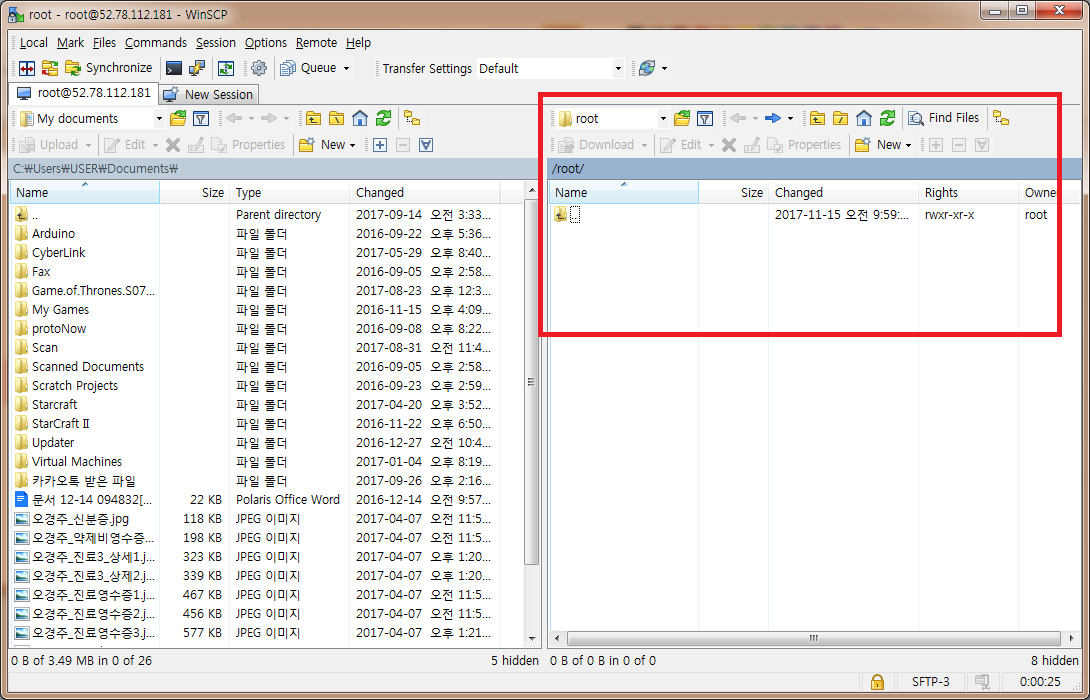


다음은 위의 4번째 선택한 Advanced...창의 내용이다.



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 배표하면 생성되는 디렉토리에 대한 권한은 root가 가지고 있을 것이다.

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

chmod –R a+rw /var/lib/tomcat8/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 |

**swapfile의 권한을 생성합니다.**

|  |
| --- |
| chmod 600 /root/swapfile |

#### 그리고 스왑 파일로 설정을 해줍니다.

|  |
| --- |
| mkswap /root/swapfile |

**스왑파일을 실행해줍니다.**

|  |
| --- |
| swapon /root/swapfile  free -h |

다음과 같이 swap에 메모리가 증설된 것을 확인할 수 있다.  


**인스턴스를 재인식할때마다 swap 파일을 기동시켜줘야하는데 이 작업은 너무 번거롭기 때문에**

**fstab파일에 값을 추가해주면 인스턴스를 재시작할때 자동으로 해당 swap을 실행시켜준다.**

|  |
| --- |
| vi /etc/fstab |

vi명령어를 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.

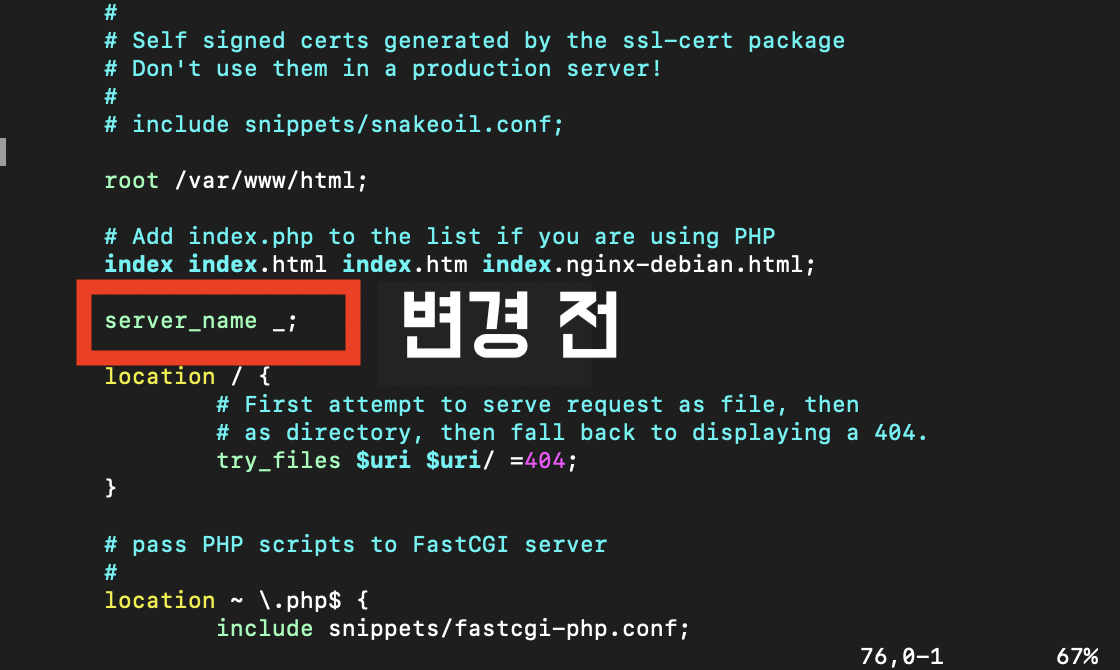
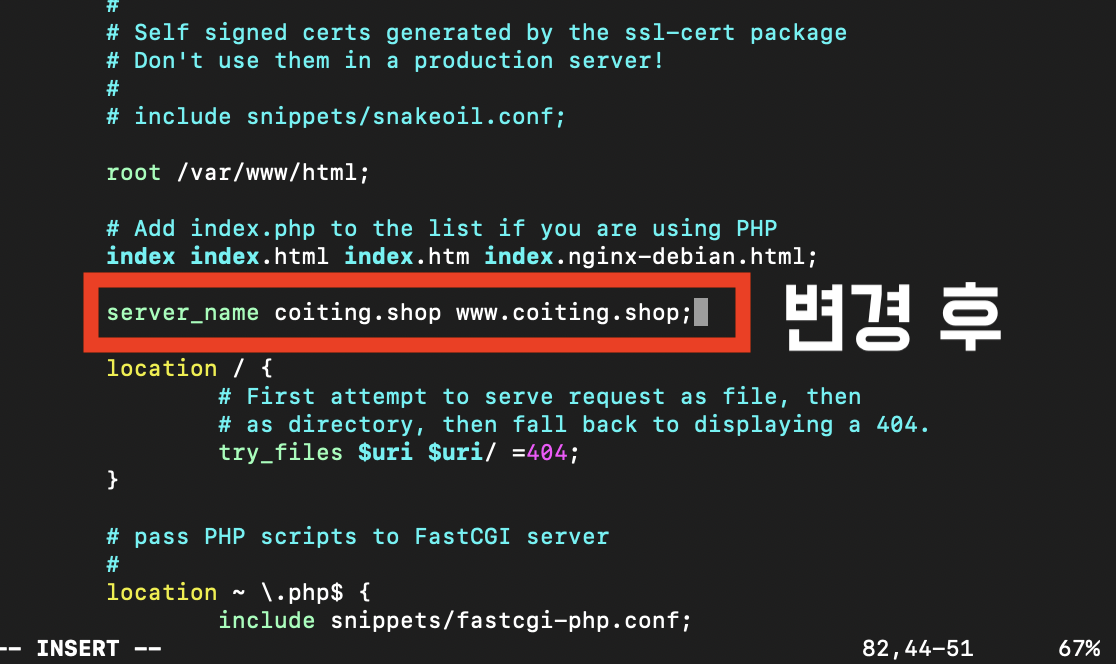
|  |
| --- |
| $ sudo apt update  $ sudo apt upgrade  $ sudo add-apt-repository ppa:certbot/certbot |

### Cerbot의 Nginx 패키지를 설치합니다.

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

### NginX Configuration을 설정

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

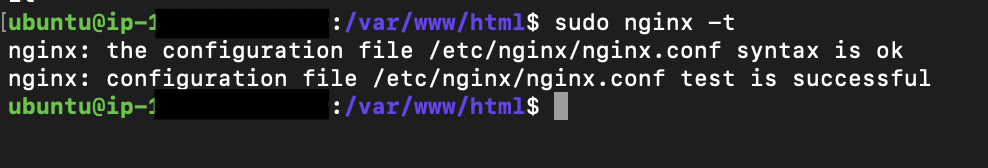
  
  


그림처럼 server\_name 항목을 찾아 뒤에 도메인명을 넣어준다.

이제 확인해보자!

|  |
| --- |
| $ sudo nginx -t |

테스트 성공인 경우 다음과 같이 나타남!

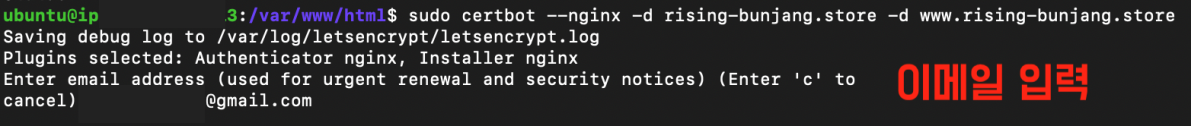


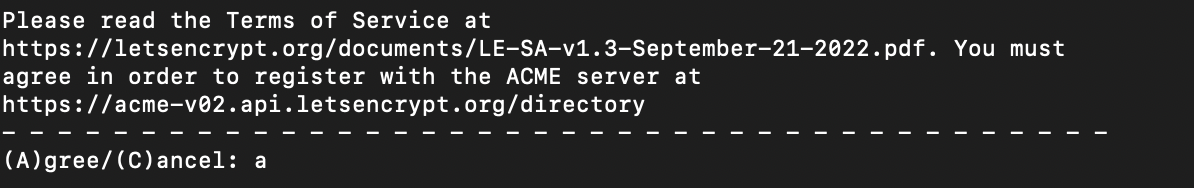
이제 NginX와 Tomcat을 재 가동하자!

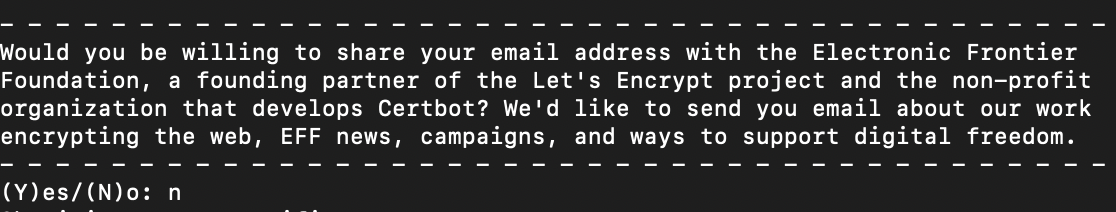
|  |
| --- |
| $ sudo service nginx restart  $ sudo service tomcat9restart |

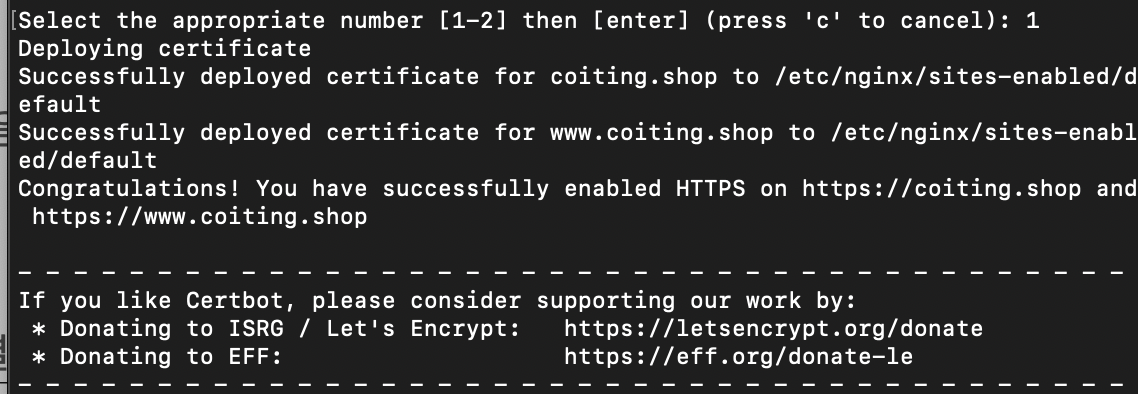
SSL인증서 받기   
원하는 도메인 등록과 인증서 획득! 그리고 이메일 입력

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



먼저 승인 Yes를 선택!  


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


잘 마무리된 화면  


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