
2024년도 본사 직원 모의 프로젝트
AWS 호스팅 환경 구성 가이드

버전	1.0
작성일자	2024.01.04.
작성자	기태성 책임

<목차>

1. Amazon EC2	1
1.1. AWS 계정 생성 및 로그인	1
1.1.1. 계정 생성	1
1.1.2. 1단계	2
1.1.3. 2단계	2
1.1.4. 3단계	2
1.1.5. 4단계	3
1.1.6. 5단계	3
1.1.7. 가입 완료 후 로그인	3
2. 인스턴스 생성	4
2.1. EC2 서비스 생성	4
2.2. 인스턴스 시작	4
2.2.1. 운영체제 선택	4
2.2.2. 인스턴스 유형	5
2.2.3. 키 페어 생성	5
2.2.4. 네트워크 설정 확인	6
2.2.5. 스토리지 구성 확인	6
2.2.6. 인스턴스 생성 요청	6
2.3. 인스턴스 IP 확인	7
2.4. 인스턴스 연결 정보 확인	7
3. 인스턴스 접속	8
3.1. SSH	8
3.1.1. PuTTY	8
3.2. FTP	10
3.2.1. FileZilla	10
4. Java	11
4.1. Oracle JDK (1.8.0_202)	11
4.2. Open JDK (Eclipse Temurin 빌드)	11
4.3. 서버에 설치	11

5. Tomcat	12
5.1. 버전별 스펙	12
5.2. 다운로드	12
5.3. 서버에 설치	13
5.4. 환경 설정	13
5.4.1. bin/catalina.sh	13
5.4.2. war 파일 배포	13
5.4.3. conf/server.xml	14
5.5. 서버 구동	15
5.5.1. 시작	15
5.5.2. 종료	15
5.5.3. 로깅	15
6. Database	16
6.1. MariaDB	16
6.1.1. 패키지 최신화 및 업데이트	16
6.1.2. install	16
6.1.3. 기본 명령어	16
6.1.4. 외부 접속 설정	17
6.1.5. 계정 설정	17
6.1.6. 접근 확인	17
7. GIT	18
7.1. install	18
8. 기타 서버 환경 설정	19
8.1. hostname 변경	19
8.2. timezone 변경	19
8.3. AWS 포트 열기	20

1. Amazon EC2

<https://aws.amazon.com/ko/ec2/>

※ **해외결제** 가능한 **신용카드** 필요

1.1. AWS 계정 생성 및 로그인

1.1.1. 계정 생성

Amazon EC2

거의 모든 워크로드에 적합한 안전하고 크기 조정 가능한 컴퓨팅 용량

[Amazon EC2 시작하기](#)[Amazon EC2 전문가와 연결](#)

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

[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](#)

STC (주)에스티크리에이티브
smarttomorrow creative

- 1 -

1.1.2. 1단계

AWS에 가입

루트 사용자 이메일 주소
계정 복구 및 일부 관리 기능에 사용

AWS 계정 이름
계정의 이름을 선택합니다. 이름은 가입 후 계정 설정에서 변경할 수 있습니다.

이메일 주소 확인

또는

기존 AWS 계정에 로그인

AWS에 가입

사용자 확인

보안 유지 - AWS가 하는 일입니다.

확인 코드가 포함된 이메일을
tski@stcreative.kr(으)로 보냈습니다. (귀하의
계정이 아닙니까?)

이메일을 확인하려면 아래에 입력하세요.

확인 코드

확인

AWS에 가입

암호 생성

암호는 AWS에 대한 로그인 액세스를 제공하므로 올바른 정보를 얻는 것이 중요합니다.

루트 사용자 암호

루트 사용자 암호 확인

계속(1/5단계)

또는

1.1.3. 2단계

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

국가 또는 리전

주소

메타모르포빌딩 10층 1006호

시

시, 도 또는 리전

우편 번호

☒ AWS 이용약관 및 개인정보의 수집 및 이
용에 대한 사항에 동의합니다.

계속(2/5단계)

※ 예시를 위해 회사주소로 작성함.

각 개인 정보에 맞추어 작성

1.1.4. 3단계

AWS에 가입

결제 정보

신용카드 번호



AWS는 현지에서 발급된 대부분의 신용카드를 허용합니다. 결제 옵션에 대해 자세히 알아보려면 FAQ를 참조하세요.

만료 날짜

보안 코드

보안 코드

카드 소유자 이름

청구지 주소

☒ 내 연락처 주소 사용

☐ 새 주소 사용

이메일 주소

이메일 주소는 AWS와의 거래를 위해 VAT 영수증을 발송하는 데 사용됩니다.

이메일 주소

이메일 주소는 AWS와의 거래를 위해 VAT 영수증을 발송하는 데 사용됩니다.

확인 및 계속(3/5단계)

확인 요금을 승인하기 위해 은행의 웹 사이트로 리디렉션될 수 있습니다.

※ 이후 카드 인증 진행

1.1.5. 4단계

AWS에 가입

자격 증명 확인

AWS 계정을 사용하려면 먼저 전화번호를 확인해야 합니다. 계속하면 AWS 자동 시스템이 확인 코드 전송을 위해 연락합니다.

확인 코드를 어떻게 보내 드릴까요?

- ☒ 문자 더 메시지(SMS)
☐ 음성 통화

국가 또는 리전 코드

대한민국 (+82)

후대전화 번호

01000601030

보안 검사



위에 보이는 문자를 입력하세요.

SMS 전송(4/5단계)

AWS에 가입

자격 증명 확인

코드 확인

계속(4/5단계)

문제가 있으신가요? 때로는 확인 코드를 가져오는 데 최대 10분이 걸립니다. 이보다 오래 걸리면 이전 페이지로 돌아가서 다시 시도하세요.

1.1.6. 5단계

Support 플랜 선택

비즈니스 또는 개인 계정에 대한 Support 플랜을 선택합니다. 플랜 및 요금 예시를 비교해 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

1.1.7. 가입 완료 후 로그인



축하합니다!

AWS에 가입해 주셔서 감사합니다.

지금 계정을 활성화하고 있으며 몇 분 이내에 완료됩니다. 활성화가 완료되면 이메일을 받게 됩니다.

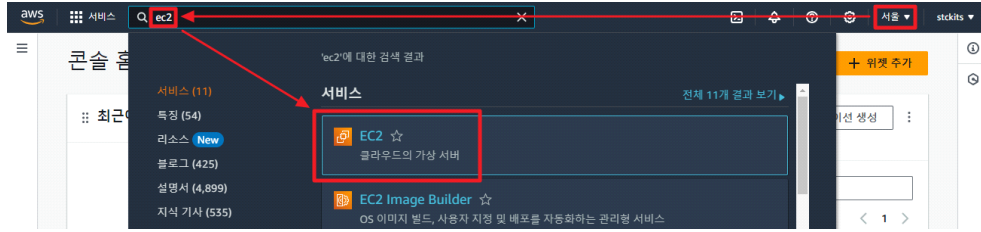
AWS Management Console로 이동

[다른 계정을 등록하거나 영업 팀에 문의](#)

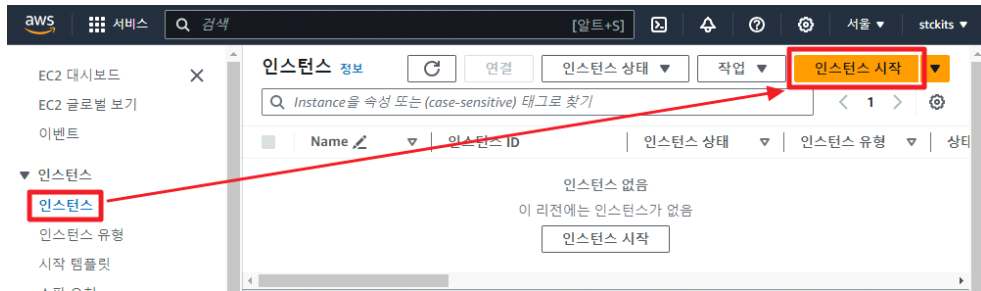
관리 콘솔로 이동 클릭하여, 생성한 계정으로 로그인

2. 인스턴스 생성

2.1. EC2 서비스 생성



서울 리전 확인 > ec2 검색 > 서비스의 “EC2” 선택



2.2. 인스턴스 시작

인스턴스 시작 정보

Amazon EC2를 사용하면 AWS 클라우드에서 실행되는 가상 머신 또는 인스턴스를 생성할 수 있습니다. 아래의 간단한 단계에 따라 빠르게 시작할 수 있습니다.

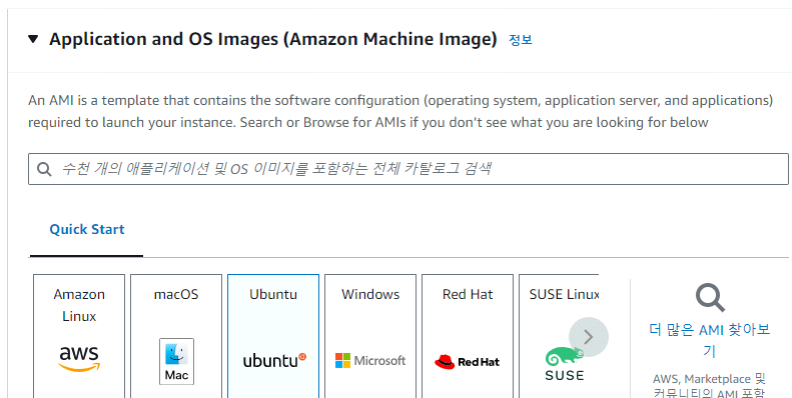
이름 및 태그 정보

이름

예: 내 웹 서버

추가 태그 추가

2.2.1. 운영체제 선택



※ 가이드에서는 Ubuntu 22.04로 진행

2.2.2. 인스턴스 유형

▼ 인스턴스 유형 정보 | [Get advice](#)

인스턴스 유형

t2.micro
 패밀리: t2 1 vCPU 1 GiB 메모리 현재 세대: true
 온디맨드 RHEL 기본 요금: 0.0744 USD 시간당
 온디맨드 Linux 기본 요금: 0.0144 USD 시간당
 온디맨드 SUSE 기본 요금: 0.0144 USD 시간당
 온디맨드 Windows 기본 요금: 0.019 USD 시간당

프리 티어 사용 가능

☒ 모든 세대

[인스턴스 유형 비교](#)

소프트웨어가 사전 설치된 AMI에는 추가 비용이 적용됩니다.

※ 가이드에서는 t2.micro 프리 티어로 진행

2.2.3. 키 페어 생성

▼ 키 페어(로그인) 정보

키 페어를 사용하여 인스턴스에 안전하게 연결할 수 있습니다. 인스턴스를 시작하기 전에 선택한 키 페어에 대한 액세스 권한이 있는지 확인하세요.

키 페어 이름 - 필수

선택

[새 키 페어 생성](#)

키 페어 생성

키 페어 이름
키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

이름에는 최대 255개의 ASCII 문자를 포함할 수 있습니다. 앞 또는 뒤에 공백을 포함할 수 없습니다.

키 페어 유형

☒ RSA
RSA 암호화된 프라이빗 및 퍼블릭 키 페어

☐ ED25519
ED25519 암호화된 프라이빗 및 퍼블릭 키 페어

프라이빗 키 파일 형식

☐ .pem
OpenSSH와 함께 사용

☒ .ppk
PuTTY와 함께 사용

⚠ 메시지가 표시되면 프라이빗 키를 사용자 컴퓨터의 안전하고 액세스 가능한 위치에 저장합니다. 나중에 인스턴스에 연결할 때 필요합니다. [자세히 알아보기](#)

취소 [키 페어 생성](#)

“키 페어 이름”.ppk로 다운로드되며, 해당 파일은 서버 접근 시 사용되므로 별도 폴더에 저장해 둘 것

2.2.4. 네트워크 설정 확인

▼ 네트워크 설정 정보

네트워크 정보

vpc-088f7faa65de95912

서브넷 정보

기본 설정 없음(가용 영역의 기본 서브넷)

퍼블릭 IP 자동 할당 정보

활성화

방화벽(보안 그룹) 정보

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 특정 트래픽이 인스턴스에 도달하도록 허용하는 규칙을 추가합니다.

※ 퍼블릭 IP가 할당되도록 되어있는지 확인

2.2.5. 스토리지 구성 확인

▼ 스토리지 구성 정보

Advanced

1x 30 GiB gp3

루트 볼륨 (암호화되지 않음)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

×

※ 프리 티어는 30GB 사용 가능

2.2.6. 인스턴스 생성 요청

▼ 요약

인스턴스 개수 정보

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-0f3a440bbcf3d043

Virtual server type (instance type)

t2.micro

Firewall (security group)

새 보안 그룹

Storage (volumes)

1 volume(s) - 30 GiB

프리 티어: 첫 해에는 월별 프리 티어 AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다.

×

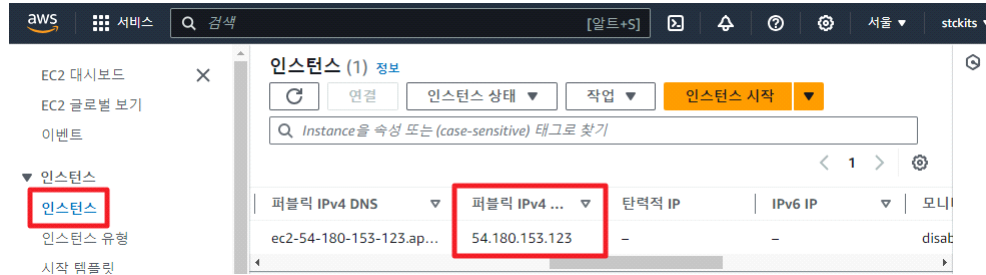
취소

인스턴스 시작

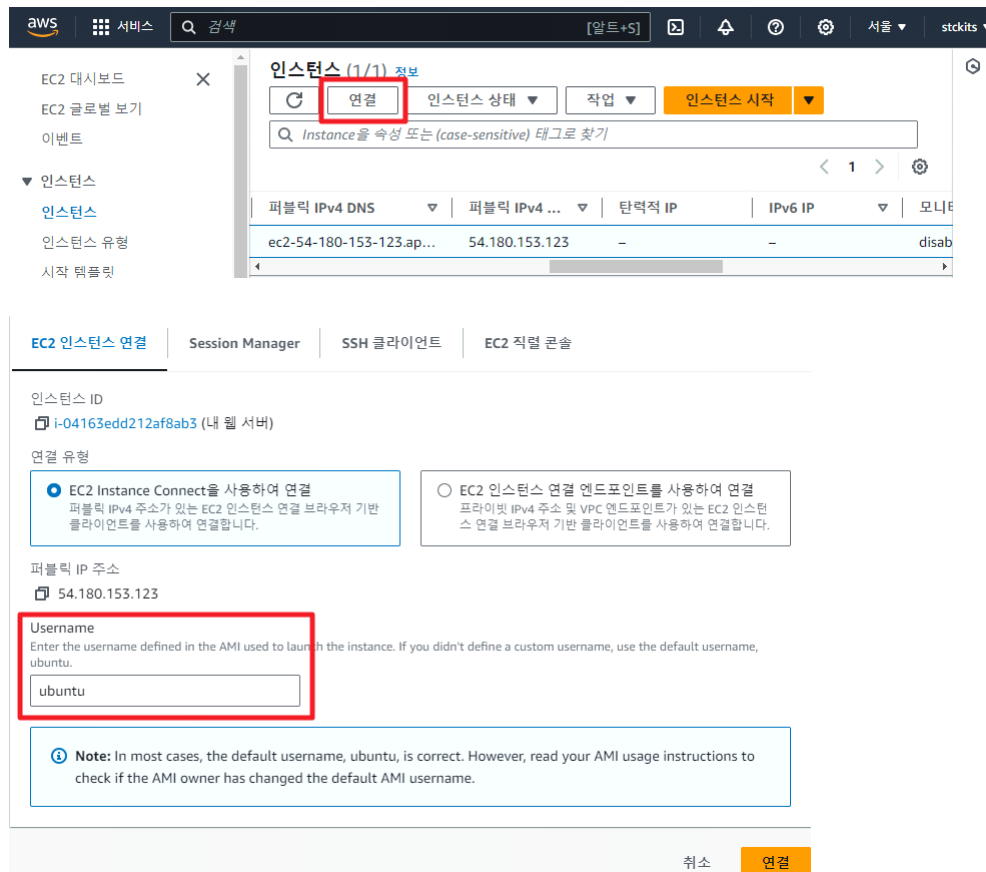
명령 검토

인스턴스 시작 버튼 클릭 후,
약간의 시간 소요 후 생성됨

2.3. 인스턴스 IP 확인



2.4. 인스턴스 연결 정보 확인

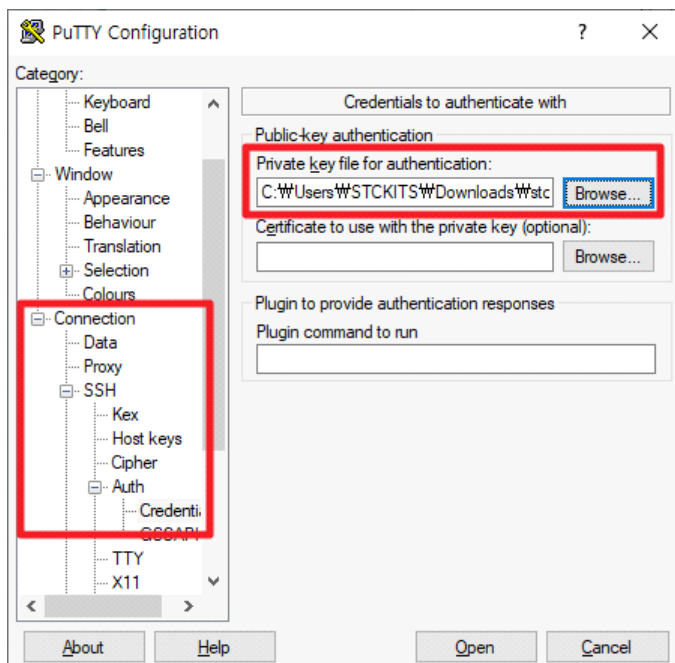
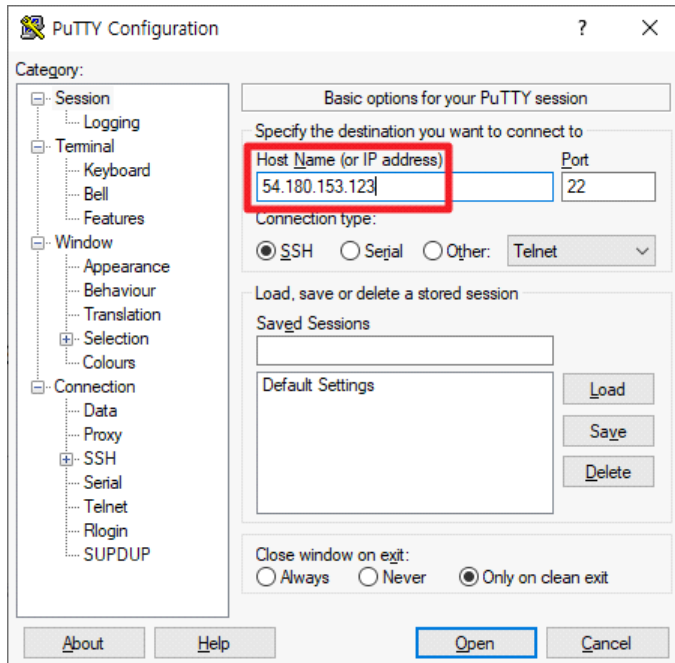


3. 인스턴스 접속

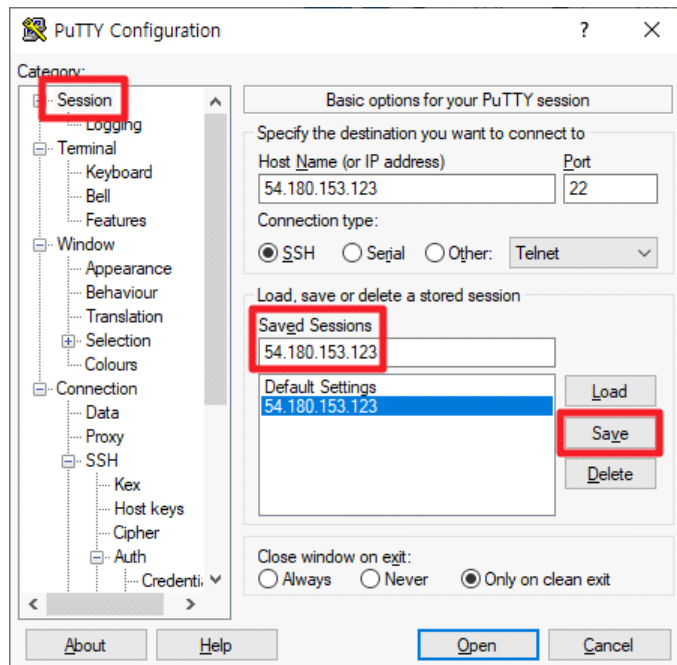
3.1. SSH

3.1.1. PuTTY

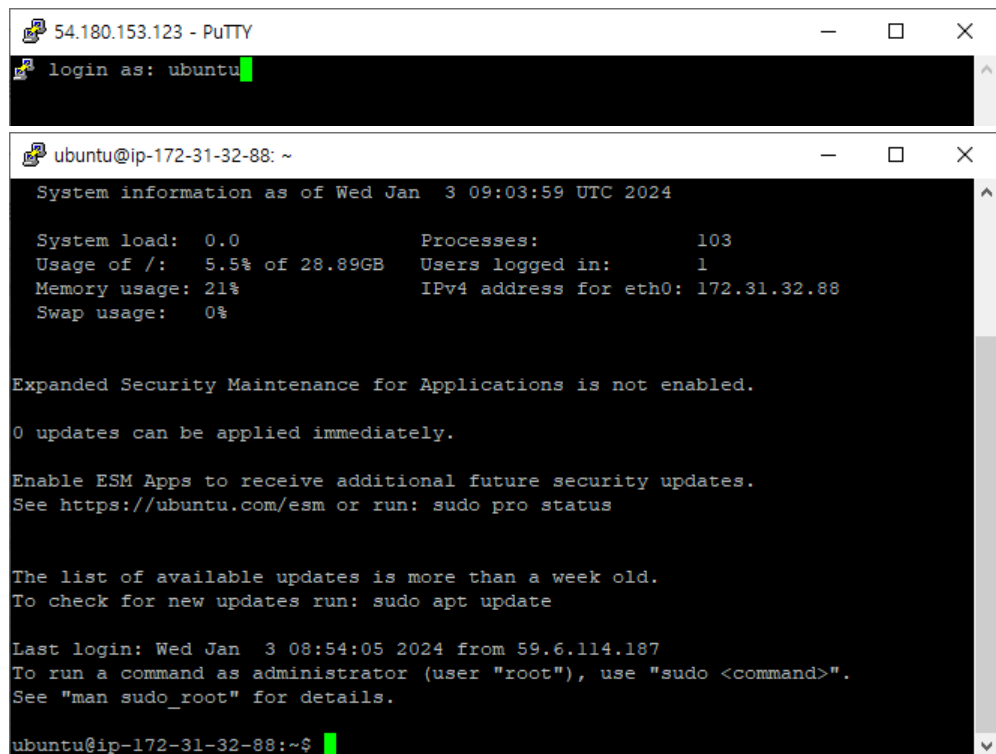
<https://www.putty.org/>



Connection > SSH > Auth > Credentials



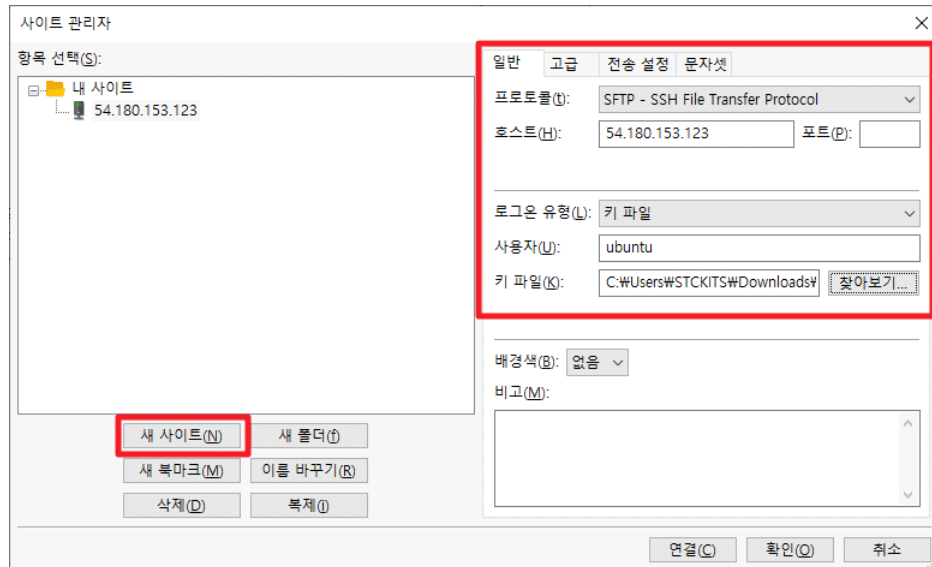
Session > Saved Sessions 항목에 저장할 이름 입력 후, Save버튼 클릭



3.2. FTP

3.2.1. FileZilla

<https://filezilla-project.org/>



4. Java

- ※ 가이드에서는 JDK 8 (Oracle JDK 8u202) 버전을 사용
- ※ 패키지 매니저를 통해 설치하지 않고, 별도 파일을 받아서 사용하는 방법으로 안내

4.1. Oracle JDK (1.8.0_202)

<https://www.oracle.com/kr/java/technologies/javase/javase8-archive-downloads.html>
jdk-8u202-linux-x64.tar.gz 파일 다운로드 (※ Oracle 홈페이지 계정 필요)

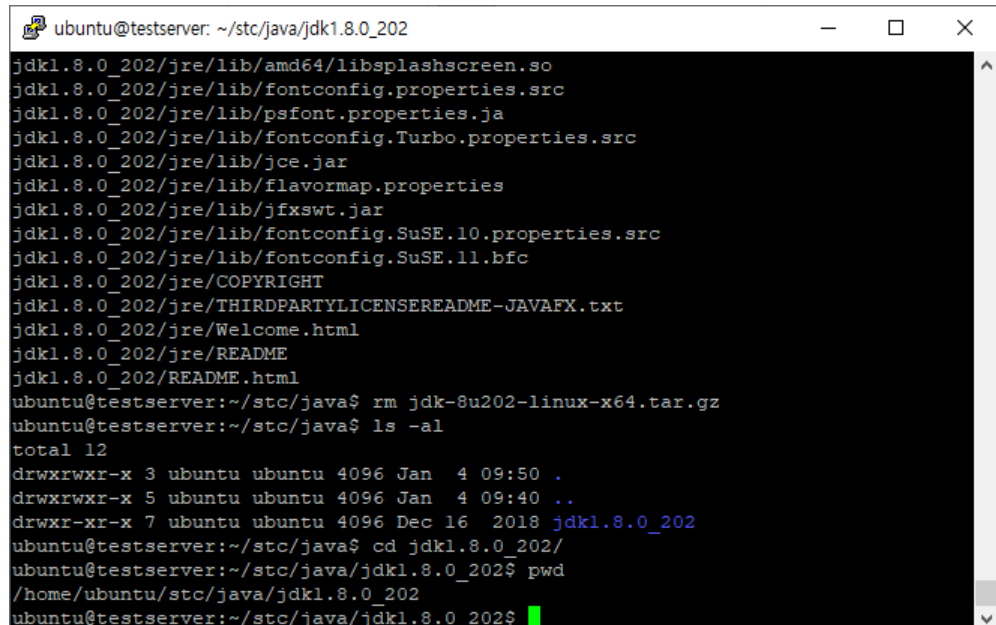
4.2. Open JDK (Eclipse Temurin 빌드)

<https://adoptium.net/temurin/releases/?version=8>

4.3. 서버에 설치

다운받은 압축파일을 서버에 업로드 후, tar 명령어로 압축 해제

```
tar -zxvf jdk-8u202-linux-x64.tar.gz
```



```
ubuntu@testserver: ~/stc/java/jdk1.8.0_202
jdk1.8.0_202/jre/lib/amd64/libsplashscreen.so
jdk1.8.0_202/jre/lib/fontconfig.properties.src
jdk1.8.0_202/jre/lib/psfont.properties.ja
jdk1.8.0_202/jre/lib/fontconfig.Turbo.properties.src
jdk1.8.0_202/jre/lib/jce.jar
jdk1.8.0_202/jre/lib/flavormap.properties
jdk1.8.0_202/jre/lib/jfxswt.jar
jdk1.8.0_202/jre/lib/fontconfig.SuSE.10.properties.src
jdk1.8.0_202/jre/lib/fontconfig.SuSE.11.bfc
jdk1.8.0_202/jre/COPYRIGHT
jdk1.8.0_202/jre/THIRDPARTYLICENSEREADME-JAVAFX.txt
jdk1.8.0_202/jre/Welcome.html
jdk1.8.0_202/jre/README
jdk1.8.0_202/README.html
ubuntu@testserver:~/stc/java$ rm jdk-8u202-linux-x64.tar.gz
ubuntu@testserver:~/stc/java$ ls -al
total 12
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan  4 09:50 .
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan  4 09:40 ..
drwxr-xr-x 7 ubuntu ubuntu 4096 Dec 16 2018 jdk1.8.0_202
ubuntu@testserver:~/stc/java$ cd jdk1.8.0_202/
ubuntu@testserver:~/stc/java/jdk1.8.0_202$ pwd
/home/ubuntu/stc/java/jdk1.8.0_202
ubuntu@testserver:~/stc/java/jdk1.8.0_202$
```

5. Tomcat

※ 가이드에서는 Tomcat 8.5 버전을 사용

5.1. 버전별 스펙

Servlet Spec	JSP Spec	EL Spec	WebSocket Spec	Authentication (JASPIC) Spec	Apache Tomcat Version	Latest Released Version	Supported Java Versions
6.1	4.0	6.0	TBD	TBD	11.0.x	11.0.0-M15 (alpha)	21 and later
6.0	3.1	5.0	2.1	3.0	10.1.x	10.1.17	11 and later
5.0	3.0	4.0	2.0	2.0	10.0.x (superseded)	10.0.27 (superseded)	8 and later
4.0	2.3	3.0	1.1	1.1	9.0.x	9.0.84	8 and later
3.1	2.3	3.0	1.1	1.1	8.5.x	8.5.97	7 and later
3.1	2.3	3.0	1.1	N/A	8.0.x (superseded)	8.0.53 (superseded)	7 and later
3.0	2.2	2.2	1.1	N/A	7.0.x (archived)	7.0.109 (archived)	6 and later (7 and later for WebSocket)
2.5	2.1	2.1	N/A	N/A	6.0.x (archived)	6.0.53 (archived)	5 and later
2.4	2.0	N/A	N/A	N/A	5.5.x (archived)	5.5.36 (archived)	1.4 and later
2.3	1.2	N/A	N/A	N/A	4.1.x (archived)	4.1.40 (archived)	1.3 and later
2.2	1.1	N/A	N/A	N/A	3.3.x (archived)	3.3.2 (archived)	1.1 and later

<https://tomcat.apache.org/whichversion.html>

5.2. 다운로드

<https://tomcat.apache.org/>

Download

Which version?

- Tomcat 11 (alpha)
- Tomcat 10
- Tomcat 9
- Tomcat 8**
- Tomcat Migration Tool for Jakarta EE
- Tomcat Connectors
- Tomcat Native
- Taglibs
- Archives

Documentation

- Tomcat 11.0 (alpha)
- Tomcat 10.1

Mirrors

You are currently using <https://dlcdn.apache.org/>. If you encounter

Other mirrors:

8.5.97

Please see the [README](#) file for packaging information. It explains w

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)**
 - [32-bit Windows zip \(pgp, sha512\)](#)
- Full docu
 - [tar](#)
- Deployer
 - [zip](#)

※ 서버에서 바로 다운로드

wget <https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.97/bin/apache-tomcat-8.5.97.tar.gz>

5.3. 서버에 설치

```
tar -zxvf apache-tomcat-8.5.97.tar.gz
```

```
ubuntu@testserver: ~/stc/was/apache-tomcat-8.5.97
apache-tomcat-8.5.97/bin/version.sh
ubuntu@testserver:~/stc/was$ rm apache-tomcat-8.5.97.tar.gz
ubuntu@testserver:~/stc/was$ cd apache-tomcat-8.5.97/
ubuntu@testserver:~/stc/was/apache-tomcat-8.5.97$ pwd
/home/ubuntu/stc/was/apache-tomcat-8.5.97
ubuntu@testserver:~/stc/was/apache-tomcat-8.5.97$ ls -al
total 156
drwxrwxr-x 9 ubuntu ubuntu 4096 Jan  4 09:59 .
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan  4 09:59 ..
-rw-r----- 1 ubuntu ubuntu 19992 Dec  8 02:57 BUILDING.txt
-rw-r----- 1 ubuntu ubuntu 6210 Dec  8 02:57 CONTRIBUTING.md
-rw-r----- 1 ubuntu ubuntu 57011 Dec  8 02:57 LICENSE
-rw-r----- 1 ubuntu ubuntu 1726 Dec  8 02:57 NOTICE
-rw-r----- 1 ubuntu ubuntu 3398 Dec  8 02:57 README.md
-rw-r----- 1 ubuntu ubuntu 7139 Dec  8 02:57 RELEASE-NOTES
-rw-r----- 1 ubuntu ubuntu 16505 Dec  8 02:57 RUNNING.txt
drwxr-x--- 2 ubuntu ubuntu 4096 Jan  4 09:59 bin
drwx----- 2 ubuntu ubuntu 4096 Dec  8 02:57 conf
drwxr-x--- 2 ubuntu ubuntu 4096 Jan  4 09:59 lib
drwxr-x--- 2 ubuntu ubuntu 4096 Dec  8 02:57 logs
drwxr-x--- 2 ubuntu ubuntu 4096 Jan  4 09:59 temp
drwxr-x--- 7 ubuntu ubuntu 4096 Dec  8 02:57 webapps
drwxr-x--- 2 ubuntu ubuntu 4096 Dec  8 02:57 work
ubuntu@testserver:~/stc/was/apache-tomcat-8.5.97$
```

5.4. 환경 설정

5.4.1. bin/catalina.sh

※ 가이드에서는 시스템상에 JAVA 환경변수를 설정하지 않고, Tomcat 실행 시 사용할 JDK를 지정하여 구동하도록 안내

```
vi catalina.sh
JAVA_HOME="자바 위치한 경로"
```

```
#
#   USE_NOHUP      (Optional) If set to the string true the start command will
#                   use nohup so that the Tomcat process will ignore any hangup
#                   signals. Default is "false" unless running on HP-UX in which
#                   case the default is "true"
# -----
#
# OS specific support.  $var _must_ be set to either true or false.
cygwin=false
darwin=false
os400=false
hpux=false

JAVA_HOME="/home/ubuntu/stc/java/jdk1.8.0_202"
```

5.4.2. war 파일 배포

webapps/ 경로 하위에 배포할 war파일을 위치시키면, 톰캣 구동시 deploy됨

```
ubuntu@testserver: ~/stc/was/apache-tomcat-8.5.97/webapps
ubuntu@testserver:~/stc/was/apache-tomcat-8.5.97/webapps$ ls -al
total 27796
drwxr-x--- 3 ubuntu ubuntu 4096 Jan  4 11:10 .
drwxrwxr-x 9 ubuntu ubuntu 4096 Jan  4 09:59 ..
drwxr-x--- 8 ubuntu ubuntu 4096 Jan  4 11:10 sample
-rw-rw-r-- 1 ubuntu ubuntu 28449543 Jan  4 11:09 sample.war
```


※ webapps/ 경로 하위에 있는 디폴트 폴더들은 삭제하는 것을 권장

```
ubuntu@testserver: ~/stc/was/apache-tomcat-8.5.97/webapps
ubuntu@testserver:~/stc/was/apache-tomcat-8.5.97/webapps$ ls -al
total 28
drwxr-x--- 7 ubuntu ubuntu 4096 Jan  4 11:06 .
drwxrwxr-x 9 ubuntu ubuntu 4096 Jan  4 09:59 ..
drwxr-x--- 3 ubuntu ubuntu 4096 Jan  4 09:59 ROOT
drwxr-x--- 16 ubuntu ubuntu 4096 Jan  4 09:59 docs
drwxr-x--- 7 ubuntu ubuntu 4096 Jan  4 09:59 examples
drwxr-x--- 6 ubuntu ubuntu 4096 Jan  4 09:59 host-manager
drwxr-x--- 6 ubuntu ubuntu 4096 Jan  4 09:59 manager
ubuntu@testserver:~/stc/was/apache-tomcat-8.5.97/webapps$ rm -rf *
ubuntu@testserver:~/stc/was/apache-tomcat-8.5.97/webapps$ ls -al
total 8
drwxr-x--- 2 ubuntu ubuntu 4096 Jan  4 11:08 .
drwxrwxr-x 9 ubuntu ubuntu 4096 Jan  4 09:59 ..
ubuntu@testserver:~/stc/was/apache-tomcat-8.5.97/webapps$
```

5.4.3. conf/server.xml

※ webapps/ 경로에 war파일을 배포하지 않고, 별도 경로에 배포하여 사용하는 경우에만 수정

별도의 배포 경로에 war파일 위치

```
ubuntu@testserver: ~/stc/deploy
ubuntu@testserver:~/stc/deploy$ pwd
/home/ubuntu/stc/deploy
ubuntu@testserver:~/stc/deploy$ ls -al
total 27792
drwxrwxr-x 2 ubuntu ubuntu 4096 Jan  4 11:33 .
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan  4 09:40 ..
-rw-rw-r-- 1 ubuntu ubuntu 28449543 Jan  4 11:09 sample.war
ubuntu@testserver:~/stc/deploy$
```

unzip 명령어로 압축 해제

unzip sample.war -d sample

```
ubuntu@testserver: ~/stc/deploy/sample
creating: sample/js/
inflating: sample/js/common.js
inflating: sample/js/jquery-1.12.4.min.js
ubuntu@testserver:~/stc/deploy$ ls -al
total 27796
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan  4 12:30 .
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan  4 09:40 ..
drwxrwxr-x 8 ubuntu ubuntu 4096 Jan  4 12:30 sample
-rw-rw-r-- 1 ubuntu ubuntu 28449543 Jan  4 11:09 sample.war
ubuntu@testserver:~/stc/deploy$ cd sample/
ubuntu@testserver:~/stc/deploy/sample$ ls -al
total 36
drwxrwxr-x 8 ubuntu ubuntu 4096 Jan  4 12:30 .
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan  4 12:30 ..
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan  4 12:30 META-INF
drwxrwxr-x 6 ubuntu ubuntu 4096 Jan  4 10:56 WEB-INF
drwxrwxr-x 2 ubuntu ubuntu 4096 Jan  4 10:56 common
drwxrwxr-x 2 ubuntu ubuntu 4096 Dec 20 2022 css
drwxrwxr-x 2 ubuntu ubuntu 4096 Dec 20 2022 images
-rw-rw-r-- 1 ubuntu ubuntu 232 Jan  4 2024 index.jsp
drwxrwxr-x 2 ubuntu ubuntu 4096 Dec 20 2022 js
ubuntu@testserver:~/stc/deploy/sample$ pwd
/home/ubuntu/stc/deploy/sample
ubuntu@testserver:~/stc/deploy/sample$
```

컨텍스트 정보 추가

```
<Context path="/sample" docBase="/home/ubuntu/stc/deploy/sample" reloadable="false"></Context>
```

```
ubuntu@testserver: ~/stc/was/apache-tomcat-8.5.97/conf

<Host name="localhost"  appBase="webapps"
      unpackWARs="true" autoDeploy="true">

    <!-- SingleSignOn valve, share authentication between web applications
      Documentation at: /docs/config/valve.html -->
    <!--
    <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
    -->

    <!-- Access log processes all example.
      Documentation at: /docs/config/valve.html
      Note: The pattern used is equivalent to using pattern="common" -->
    <Valve className="org.apache.catalina.valves.AccessLogValve" directory="
logs"
           prefix="localhost_access_log" suffix=".txt"
           pattern="%h %l %u %t &quot;%r&quot; %s %b" />

    <Context path="/sample" docBase="/home/ubuntu/stc/deploy/sample" reloada
ble="false"></Context>
  </Host>
</Engine>
</Service>
</Server>
-- INSERT --
```

5.5. 서버 구동

5.5.1. 시작

```
bin/startup.sh
```

5.5.2. 종료

```
bin/shutdown.sh
```

5.5.3. 로깅

```
tail -f logs/catalina.out
```

6. Database

※ 가이드에서는 MariaDB를 설치

특정 버전으로 설치가 필요할 시 해당 버전의 설치파일을 내려받아 진행하지만,
DB서버 구축이 아닌 모의 프로젝트에서 사용할 DB를 설치하는 것이므로
패키지 매니저를 통해 설치하는 방법으로 작성함

6.1. MariaDB

6.1.1. 패키지 최신화 및 업데이트

```
sudo apt update
sudo apt upgrade -y
```

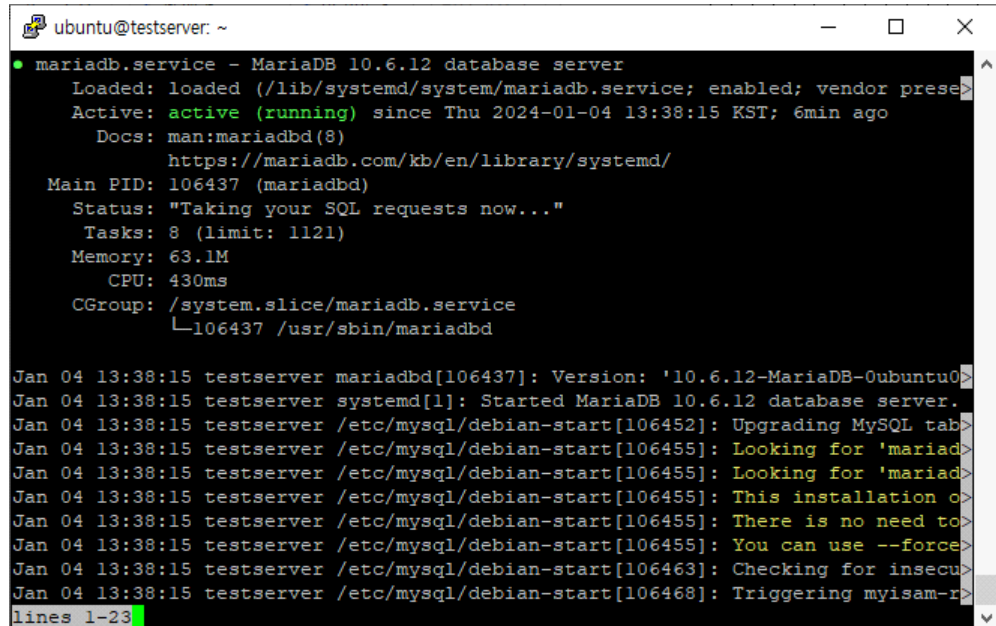
6.1.2. install

```
sudo apt install mariadb-server
```

6.1.3. 기본 명령어

상태확인

```
sudo systemctl status mysqld
```



```

ubuntu@testserver: ~
● mariadb.service - MariaDB 10.6.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
   Active: active (running) since Thu 2024-01-04 13:38:15 KST; 6min ago
     Docs: man:mariadbd(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 106437 (mariadbd)
    Status: "Taking your SQL requests now..."
     Tasks: 8 (limit: 1121)
    Memory: 63.1M
       CPU: 430ms
    CGroup: /system.slice/mariadb.service
            └─106437 /usr/sbin/mariadbd

Jan 04 13:38:15 testserver mariadbd[106437]: Version: '10.6.12-MariaDB-0ubuntu0
Jan 04 13:38:15 testserver systemd[1]: Started MariaDB 10.6.12 database server.
Jan 04 13:38:15 testserver /etc/mysql/debian-start[106452]: Upgrading MySQL tab
Jan 04 13:38:15 testserver /etc/mysql/debian-start[106455]: Looking for 'mariad
Jan 04 13:38:15 testserver /etc/mysql/debian-start[106455]: Looking for 'mariad
Jan 04 13:38:15 testserver /etc/mysql/debian-start[106455]: This installation c
Jan 04 13:38:15 testserver /etc/mysql/debian-start[106455]: There is no need to
Jan 04 13:38:15 testserver /etc/mysql/debian-start[106455]: You can use --force
Jan 04 13:38:15 testserver /etc/mysql/debian-start[106463]: Checking for insecu
Jan 04 13:38:15 testserver /etc/mysql/debian-start[106468]: Triggering myisam-r
lines 1-23

```

서비스 활성화

```
sudo systemctl enable mysqld
```

※ ubuntu 재시작 후에도 자동으로 MariaDB 실행되도록 설정

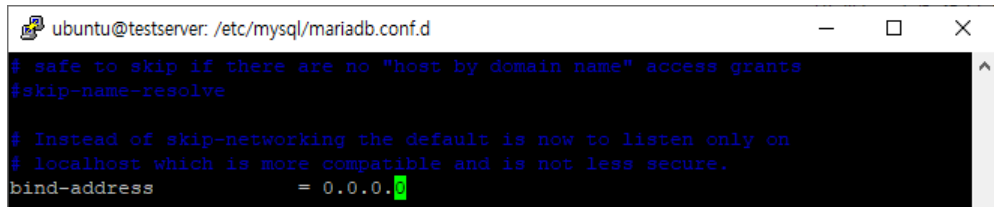
서비스 구동 관련

```
sudo systemctl start mysqld
sudo systemctl stop mysqld
sudo systemctl restart mysqld
```

6.1.4. 외부 접속 설정

◎ bind-address 수정

```
sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf
bind-address = 127.0.0.1를 0.0.0.0 으로 수정
```



MariaDB restart

6.1.5. 계정 설정

◎ root 계정 비밀번호 설정

```
sudo mysql
use mysql
set password for 'root'@'localhost' = PASSWORD('Passw0rd!');
flush privileges;
```

◎ root 계정으로 접속

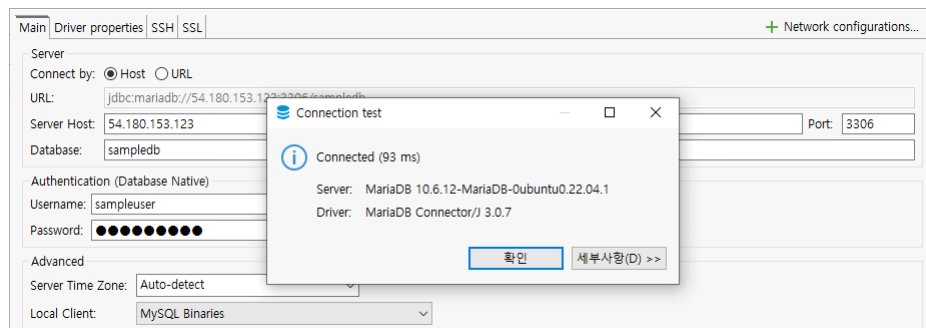
```
mysql -uroot -p
```

◎ database/user 생성 및 권한 부여

```
create database sampledb;
create user 'sampleuser'@'%' identified by 'Passw0rd!';
create user 'sampleuser'@'localhost' identified by 'Passw0rd!';
grant all privileges on sampledb.* to 'sampleuser'@'%';
grant all privileges on sampledb.* to 'sampleuser'@'localhost';
flush privileges;
※ 볼드 처리한 부분 개인별 설정
```

6.1.6. 접근 확인

※ DBeaver, HeidiSQL 등 DB tool 활용하여 Database 접속 확인



7. GIT

※ git 사용방법에 대한 내용은 포함하지 않음

git에서 commit 혹은 push 발생 시 별도 script를 실행할 수 있어 서버에 배포되도록 환경을 구성할 수 있으나, 전자정부(spring) 프로젝트의 경우 별도의 build환경 구축이 필요하므로 이번 가이드에서는 제외함

7.1. install

AWS의 ubuntu 배포판에 기본 탑재되어 있는 것으로 보이며, 설치가 안되어 있다면
sudo apt install git 명령어로 설치

7.2. GitHub

<https://github.com/>

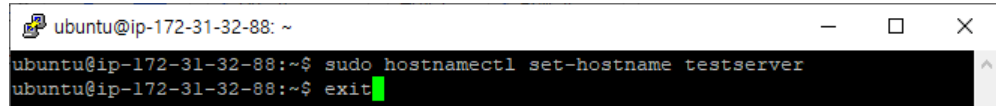
깃허브에 repository 생성하여 형상관리 가능

GitHub Desktop 프로그램 설치하여 로컬환경에서 관리 가능

8. 기타 서버 환경 설정

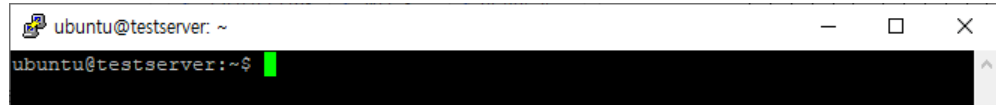
8.1. hostname 변경

sudo hostnamectl set-hostname 서버이름



```
ubuntu@ip-172-31-32-88: ~  
ubuntu@ip-172-31-32-88:~$ sudo hostnamectl set-hostname testserver  
ubuntu@ip-172-31-32-88:~$ exit
```

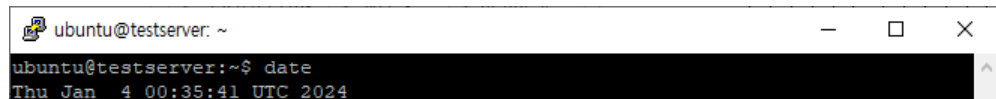
hostnamectl 명령어 수행 후, 재접속하면 반영 확인 가능



```
ubuntu@testserver: ~  
ubuntu@testserver:~$
```

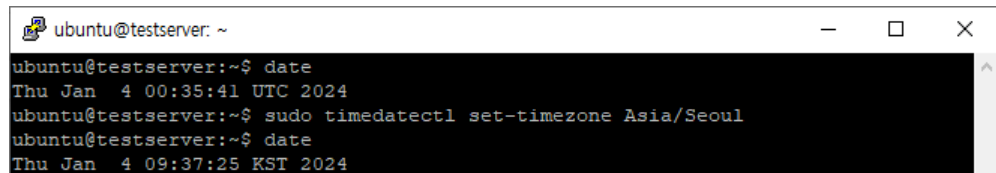
8.2. timezone 변경

date 명령어로 시간 확인해 보면 UTC로 되어있음



```
ubuntu@testserver: ~  
ubuntu@testserver:~$ date  
Thu Jan 4 00:35:41 UTC 2024
```

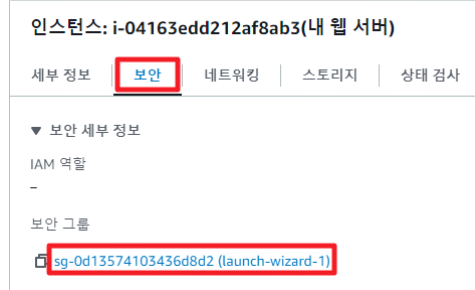
sudo timedatectl set-timezone Asia/Seoul



```
ubuntu@testserver: ~  
ubuntu@testserver:~$ date  
Thu Jan 4 00:35:41 UTC 2024  
ubuntu@testserver:~$ sudo timedatectl set-timezone Asia/Seoul  
ubuntu@testserver:~$ date  
Thu Jan 4 09:37:25 KST 2024
```

8.3. AWS 포트 열기

인스턴스 > 하단 보안 탭 > 보안 그룹 클릭



보안 그룹 > 하단 인바운드 규칙 탭 > 인바운드 규칙 편집



규칙 추가 버튼 클릭 > 오픈할 포트정보 입력 > 규칙 저장 클릭

