

# 시스템프로그래밍 레포트

## 리눅스에 SSH 서버 설치하기



한국기술교육대학교  
**KOREATECH**

이름	곽재우
학번	2019136009
분반	01
작성일자	2023/09/03

## 1. SSH란 무엇일까?

SSH는 Secure Shell Protocol로 네트워크 프로토콜 중 하나이다. 네트워크 프로토콜이니 컴퓨터끼리 통신할 때 가지는 규칙이라고 이해하며 시작해보자. 그런데 SSH의 Secure라는 단어를 통해 보안과 관련된 네트워크 프로토콜인 것을 알 수 있다. 즉 **SSH는 컴퓨터끼리 통신할 때 보안이 되어 안전하게 통신하기 위해 사용하는 네트워크 프로토콜이다. SSH라는 프로토콜로 네트워크를 통해 다른 컴퓨터로 접근하거나 해당 컴퓨터에서 어떤 명령 실행을 할 수 있다.**

그렇다면 SSH는 어디서 사용될까? 한번쯤은 GitHub에 코드를 push 해본 경험 있을 것이다. 로컬 환경(내 컴퓨터)에서 원격 저장소로 코드가 전송되는데, 여기에 SSH가 사용된다. 즉, SSH는 데이터 전송에 대표적으로 사용된다. 이외에도 동일한 네트워크에 존재하는 다른 컴퓨터에 접속하는 원격 제어에도 사용된다. 오늘은 원격 제어에 사용되는 SSH에 대해 자세하게 알아보자.

SSH라는 프로토콜로 통신하려면 당연히도 Server와 Client가 필요하다. 즉, SSH Server와 SSH Client의 구성이 필요하다. 먼저 SSH Server를 설치해보자.

## 2. 리눅스에 SSH Server 설치하기

터미널에서 아래 명령어로 Open SSH Server를 설치한다.

```
jaewoo@ubuntu ~$ sudo apt install openssh-server
```

## 3. SSH Server 실행하기

이제 SSH Server를 실행해보자. SSH Server를 실행하는 명령어는 아래와 같다.

```
jaewoo@ubuntu ~$ sudo systemctl start ssh
```

아래 명령어로 SSH가 실행 중인지 확인할 수 있다. **active (running)** 상태로 나타난다.

```
jaewoo@ubuntu ~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor pre>
   Active: active (running) since Sat 2023-09-02 08:48:43 UTC; 1min 21s>
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 39438 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/>
   Main PID: 39439 (sshd)
     Tasks: 1 (limit: 9374)
    Memory: 1.2M
     CGroup: /system.slice/ssh.service
             └─39439 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 start>

Sep 02 08:48:43 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
Sep 02 08:48:43 ubuntu sshd[39439]: Server listening on 0.0.0.0 port 22.
Sep 02 08:48:43 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
Sep 02 08:48:43 ubuntu sshd[39439]: Server listening on :: port 22.
lines 1-16/16 (END)
```

만약 SSH가 실행 중이 아니라면 아래와 같이 (inactive) 나타난다.

```
jaewoo@ubuntu ~$ sudo systemctl stop ssh
jaewoo@ubuntu ~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor pre>
   Active: inactive (dead) since Sat 2023-09-02 08:50:57 UTC; 1s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 39438 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/>
   Process: 39439 ExecStart=/usr/sbin/sshd -D $SSHD_OPTS (code=exited, s>
   Main PID: 39439 (code=exited, status=0/SUCCESS)

Sep 02 08:48:43 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
Sep 02 08:48:43 ubuntu sshd[39439]: Server listening on 0.0.0.0 port 22.
Sep 02 08:48:43 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
Sep 02 08:48:43 ubuntu sshd[39439]: Server listening on :: port 22.
Sep 02 08:50:57 ubuntu systemd[1]: Stopping OpenBSD Secure Shell server...
Sep 02 08:50:57 ubuntu sshd[39439]: Received signal 15; terminating.
Sep 02 08:50:57 ubuntu systemd[1]: ssh.service: Succeeded.
Sep 02 08:50:57 ubuntu systemd[1]: Stopped OpenBSD Secure Shell server.
lines 1-17/17 (END)
```

SSH를 비활성화 하는 명령어는 SSH를 실행하는 명령어와 거의 비슷하다.

```
jaewoo@ubuntu ~$ sudo systemctl stop ssh
```

#### 4. Client에서 접속하기

우분투에 SSH Server를 설치하였다. 이제 SSH Client를 설치해서 접속해보자.  
우분투에는 기본적으로 SSH Client가 설치되어 있다.

설치되어 있지 않는 경우에는 아래 명령어로 설치할 수 있다.

```
jaewoo@ubuntu ~$ sudo apt-get install openssh-client
```

SSH Client를 설치하였다면 아래와 같은 명령어 형식으로 SSH Server에 접속할 수 있다.

```
jaewoo@ubuntu ~$ ssh username@ip_address
```

우선 현재 우분투가 SSH Server와 Client 역할을 모두 수행하므로 터미널에서 나의 우분투 컴퓨터에 접속해보자.

나의 컴퓨터 username은 jaewoo이다. 그렇다면 IP address는 어떻게 확인할까?

IP address는 ifconfig나 ip a 명령어로 확인할 수 있다.

```
jaewoo@ubuntu ~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 32:f6:ad:56:08:54 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.211/24 brd 192.168.0.255 scope global dynamic noprefixr
oute enp0s1
        valid_lft 7131sec preferred_lft 7131sec
    inet6 fe80::30f6:adff:fe56:854/64 scope link
        valid_lft forever preferred_lft forever
```

IP address가 192.168.0.211인 것을 확인할 수 있다.

위의 형식대로 username과 IP address를 입력하여 SSH Client에서 서버로 접속해 보자.

```
✗ jaewoo@ubuntu ~$ ssh jaewoo@192.168.0.211
```

위와 같이 서버에 접속하려고 하니 아래와 같은 메시지가 출력되었다.

```
✗ jaewoo@ubuntu ~$ ssh jaewoo@192.168.0.211
The authenticity of host '192.168.0.211 (192.168.0.211)' can't be establis
hed.
ECDSA key fingerprint is SHA256:io+DS1T2WTUjRVG08jV7QvG+joteAvb2uEMbid+7ZS
0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

이는 특정 호스트에 **최초로 SSH 접속 시** RSA Key fingerprint로 접속 여부를 물 어보는 것이다. 처음 접속하기 때문에 "들어갈거야?" 라고 물어보는 것이다.

yes를 입력하고 비밀번호를 입력하면 접속할 수 있다.

비밀번호를 입력하니 성공적으로 접속한 모습이다.

```
jaewoo@192.168.0.211's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-159-generic aarch64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Sun 03 Sep 2023 01:05:07 PM UTC

System load:  0.39               Processes:           219
Usage of /:   40.8% of 29.82GB   Users logged in:    1
Memory usage: 10%               IPv4 address for enp0s1: 192.168.0.211
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how Micro
K8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

15 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

37 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Sat Sep 2 08:28:33 2023 from 172.30.1.25
```

상단에 Welcome to Ubuntu 메시지와 함께 SSH Server에 접속한 것을 환영해주는 모습이다. 컴퓨터 서버가 나를 환영해주니 기분이 좋다.

또한 하단에는 최근 로그인 시간을 보여주는데, 172.30.1.25는 우리 대전 집 주소이다. 현재는 한기대 기숙사 다솔관에서 접속하였기에 IP address가 바뀐 것을 알 수 있다. 이렇게 SSH Server에 접속을 간단하게 할 수 있다.

그러나 이렇게 끝내기엔 아쉽다. 정말 SSH Client에서 SSH Server로 잘 접속한 것일까? 내 맥북의 UTM 가상 환경 위에서 동작하는 우분투에서 자기 자신에게 접속한 꼴이니 뭔가 와 닿지 않는다.. 이번에는 컴퓨터 2대로 직접 실습해보자.

## 5. SSH Server 접속 실습

환경 구성은 다음과 같다.

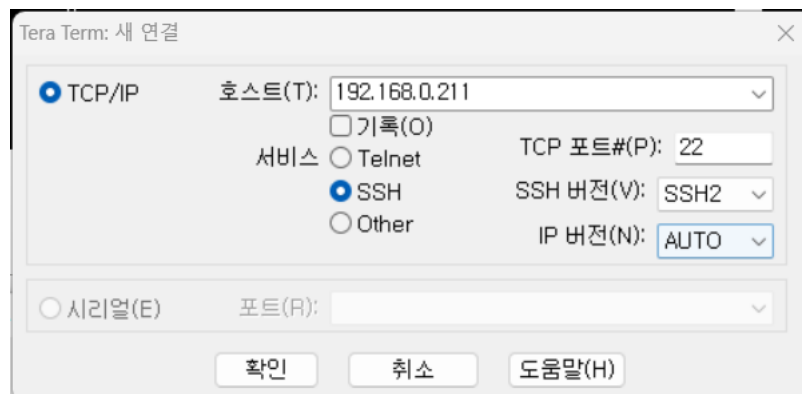
왼쪽은 MacBook Air M1에서 구동하는 Ubuntu이고 오른쪽은 LG Gram이다.



전자는 SSH Server가 될 것이고, 후자는 SSH Client가 될 것이다.

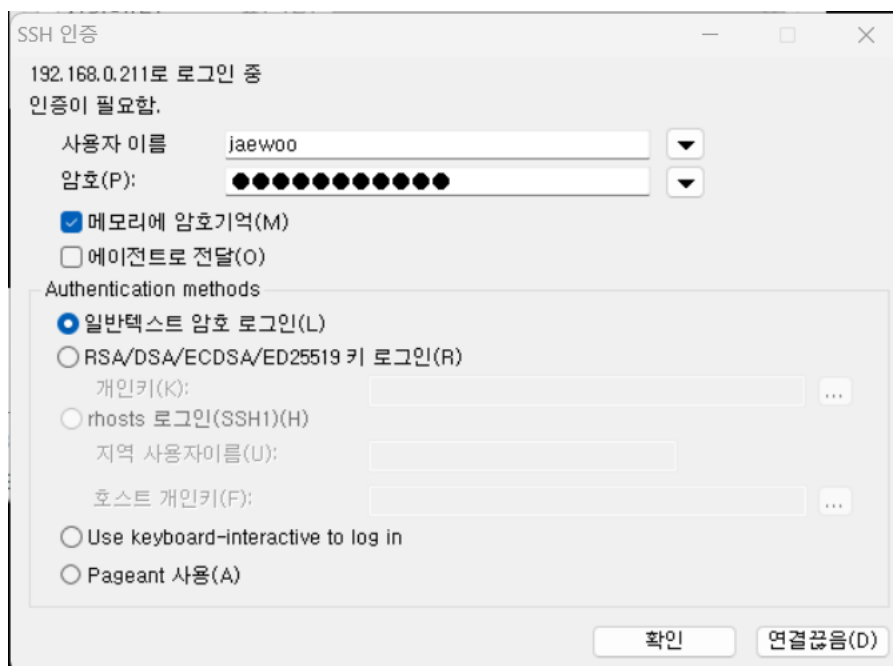
SSH Client는 Windows에서 진행하기 때문에 teraterm을 사용하였다.

teraterm을 설치하고 호스트 설정을 해 준다.



Server IP address는 192.168.0.211이니 호스트에 그대로 입력해준다.

그리고 확인을 클릭하면 아래와 같은 창이 나타난다.



서버의 사용자 이름과 암호를 입력해주면 된다.

그리고 확인을 누르면,



```
192.168.0.211 ~ VT
메뉴(F) 수정(E) 설정(S) 제어(O) 창(W) 도움말(H)
System load: 0.01      Processes: 220
Usage of /: 40.8% of 29.82GB    Users logged in: 1
Memory usage: 10%      IPv4 address for enp0s1: 192.168.0.211
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

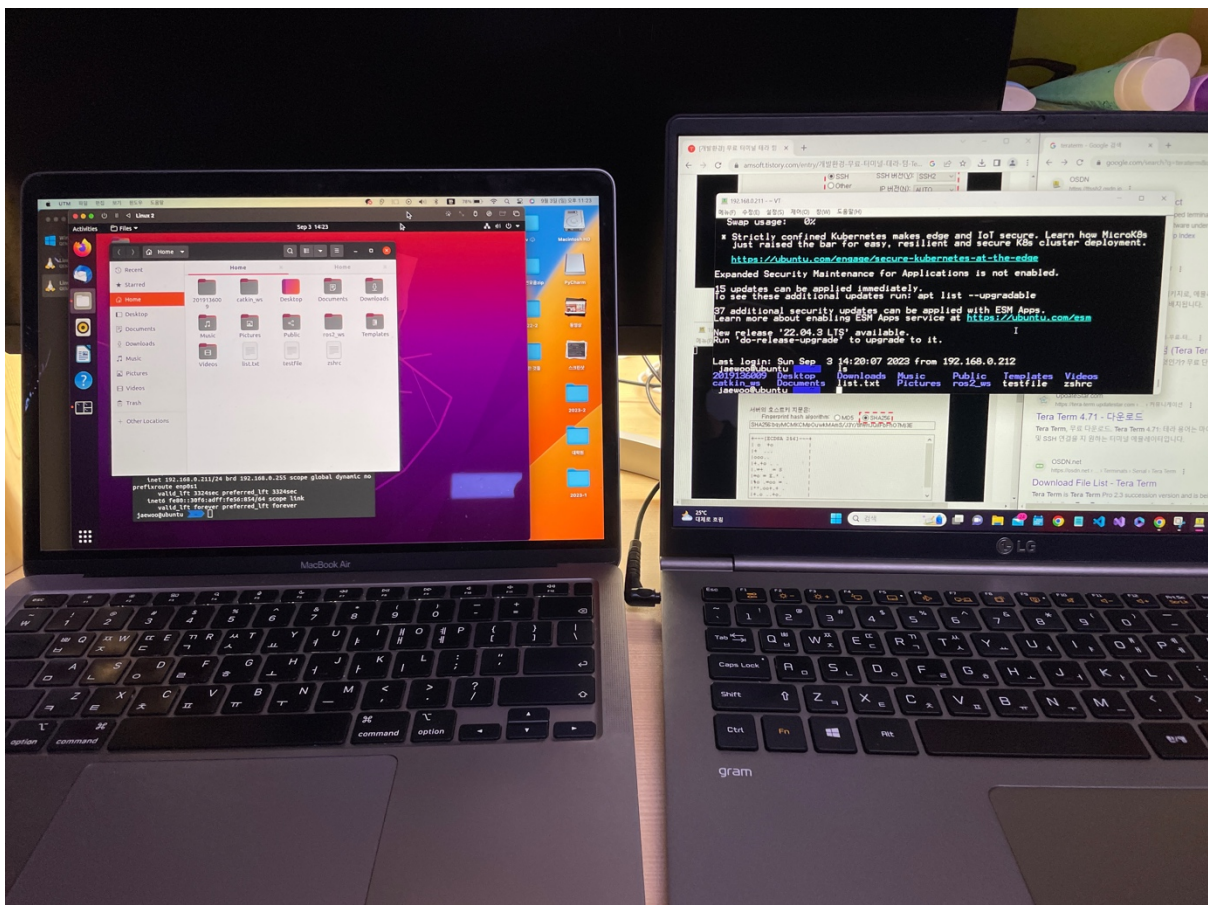
Expanded Security Maintenance for Applications is not enabled.
15 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
37 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Sep  3 14:20:07 2023 from 192.168.0.212
jaewoo@ubuntu
```

위와 같이 성공적으로 로그인 한 것을 볼 수 있다.

LG Gram이 내 맥북의 가상머신 위에서 돌아가는 우분투와 통신할 수 있게 된 것이다. LG Gram의 teraterm에서 SSH Server에 로그인한 기념으로 ls 명령어를 입력하여 보았다.



정말 신기하게도 LG Gram의 Teraterm에서 우분투의 파일 구조를 출력하는 것을 볼 수 있다. 위 사진에서 맥북에서는 우분투의 홈 디렉터리 구조를 아이콘으로 보여주고 있고, LG Gram에서는 Teraterm에서 커맨드 라인으로 우분투의 홈 디렉터리 구조를 보여주고 있다. 중요한 사실은 어떤 컴퓨터에서 다른 컴퓨터에 원격으로 접속하여 통신한다는 것이다.

## 6. 마무리

이렇게 SSH가 무엇인지, 그리고 SSH Server와 Client를 구성하여 서로 통신하는 실습을 진행해보았다. 사실 이전까지는 SSH라는 단어를 몇번 보았지만 의미는 몰랐는데 이번에 공부하면서 SSH가 무엇인지 잘 알게 되었다. 더 나아가 두 대의 PC로 SSH Server와 Client를 구성하여 통신해보면서 SSH 통신의 강력함을 느낄 수 있는 좋은 기회였다.

## 7. Reference

<https://codechacha.com/ko/ubuntu-install-openssh/>

<https://baked-corn.tistory.com/52>

<https://wlsvud84.tistory.com/12>

<https://chancoding.tistory.com/165>

<https://jootc.com/p/201808031462>