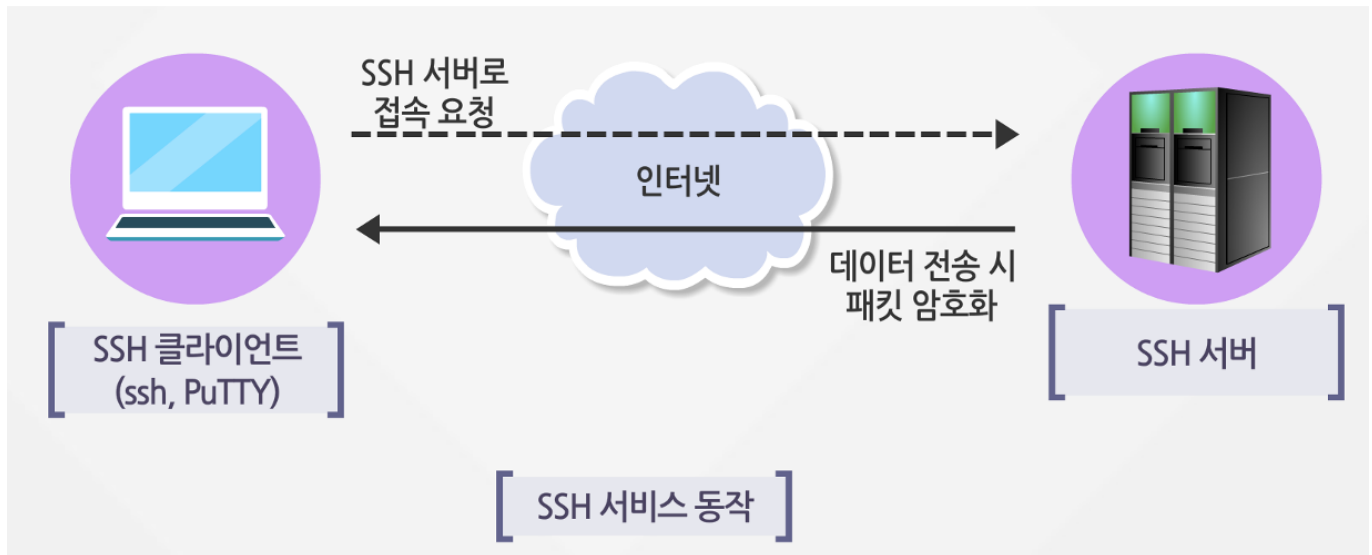


SSH 서비스

#01. SSH의 이해

- 전송되는 데이터가 암호화 처리되는 보안이 강화된 원격 접속 서비스
- 과거에 사용되던 텔넷(Telnet) 방식의 경우 모든 데이터가 암호화 되지 않고 전송되어 악의적인 목적을 갖는 공격자가 중간에 패킷을 가로채어 분석 후 해킹을 시도할 수 있으므로 현재는 사용되지 않음 (ex: PC통신)



#02. SSH 서비스 작동 확인

우분투 리눅스를 설치하는 과정에서 이미 SSH 서비스를 설치하여 사용하고 있으므로 정상 동작 여부만 확인하면 된다.

1. SSH 서비스 동작 여부 확인하기

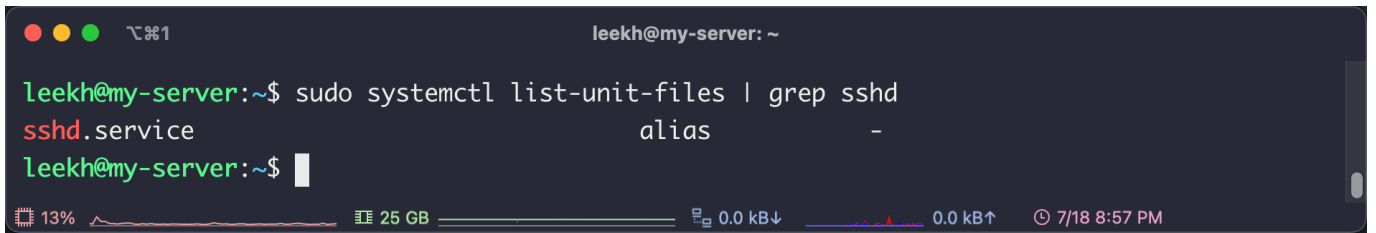
```
$ sudo systemctl status sshd
```

```
leekh@my-server: ~
leekh@my-server:~$ sudo systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-07-15 05:39:44 UTC; 3 days ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 753 (sshd)
      Tasks: 1 (limit: 2192)
     Memory: 6.6M
        CPU: 421ms
    CGroup: /system.slice/ssh.service
            └─753 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

lines 1-11...skipping...
```

2. 부팅시 자동 실행 여부 확인하기

```
$ sudo systemctl list-unit-files | grep sshd
```



```
leekh@my-server: ~
leekh@my-server:~$ sudo systemctl list-unit-files | grep sshd
sshd.service
alias -
leekh@my-server:~$
```

#03. SSH 보안 설정

- SSH는 기본 포트번호 22번을 사용하고 있기 때문에 이를 임의의 포트로 변경한다.
- 시스템의 모든 권한을 갖는 root계정에 대한 SSH 직접 접속을 차단하고 일반 사용자로만 접근이 가능하도록 해야 한다.

1. 환경설정파일 열기

```
$ sudo vi /etc/ssh/sshd_config
```

2. 환경설정파일에 아래의 내용 추가

대소문자 구분과 이름과 값 사이에 =표시가 없음에 주의한다.

```
# 기본 접속 포트 설정
Port 9901

# root 계정의 직접 로그인 차단
PermitRootLogin no
```

3. SSH 서비스 재시작

```
$ sudo systemctl restart sshd
```

4. 방화벽 포트 설정

기본 포트로 사용중이던 22번을 삭제하고 새로 설정한 9901을 허용한다.

```
$ sudo ufw delete allow 22
$ sudo ufw allow 9901/tcp
```

```
leekh@my-server:~$ sudo ufw delete allow 22
Rule deleted
Rule deleted (v6)
leekh@my-server:~$ sudo ufw allow 9901/tcp
Rule added
Rule added (v6)
leekh@my-server:~$
```

5. 방화벽 규칙 다시 로드하기

```
$ sudo ufw reload
```

6. 방화벽 규칙 확인

```
$ sudo ufw status
```

```
leekh@my-server:~$ sudo ufw status
Status: active

To Action From
--
9902/tcp ALLOW Anywhere
10100:10200/tcp ALLOW Anywhere
9901/tcp ALLOW Anywhere
9902/tcp (v6) ALLOW Anywhere (v6)
10100:10200/tcp (v6) ALLOW Anywhere (v6)
9901/tcp (v6) ALLOW Anywhere (v6)
```

#04. 외부에서의 접속 확인

Window, Mac 등의 호스트 운영체제에서 Ubuntu로 SSH 접속을 시도한다. 이 때 변경된 포트번호를 명시해야 한다.

```
$ ssh 계정명@리눅스아이피 -p포트번호
```

root로 로그인 되지 않음을 확인

비밀번호를 올바르게 입력하더라도 권한이 없다는 에러 메시지가 표시된다.

```

ssh root@10.211.55.8 -p9901 - 87x7

www on main via v8.1.5
→ ssh root@10.211.55.8 -p9901
root@10.211.55.8's password:
Permission denied, please try again.
root@10.211.55.8's password:
Permission denied, please try again.
root@10.211.55.8's password:

```

일반 사용자 계정으로 정상 접속 확인

```

leekh@my-server: ~ - 87x15

www on main via v8.1.5
→ ssh leekh@10.211.55.8 -p9901
leekh@10.211.55.8's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-41-generic aarch64)

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

System information as of Thu Jul 21 08:45:11 AM UTC 2022

System load:          0.04833984375
Usage of /:            19.6% of 29.82GB
Memory usage:         13%
Swap usage:           0%

```

#05. SSH 인증키를 사용한 로그인 설정

SSH 키는 공개 키 암호화 방식 및 인증 확인 응답 인증을 사용하는 SSH 서버에 대해 자체 식별하는 방식이다.

비공개 서버에 접속하기 위해서는 인증절차를 거쳐야 하는데, 기존에 비밀번호를 네트워크를 통해 보내는 비밀번호 인증은 네트워크 상에서 ID/비밀번호가 그대로 노출되는 문제가 있고, 접속할 때마다 입력해야 하는 번거로움이 있다.

SSH 키는 이와 달리 공개키 암호 방식을 사용하여 서버에서 인증받을 수 있으며, 암호를 생략하고 원격 호스트로 접속할 수 있다.

인증키는 서버에서 생성하거나 접속할 클라이언트 컴퓨터에서 모두 생성할 수 있다.

1) 인증키 관련 파일

인증키 이름이 **helloworld** 라고 가정할 경우 인증키와 공개키가 일치해야만 로그인이 이루어진다.

파일	설명
helloworld	클라이언트 운영체제의 ~/.ssh 디렉토리에 보관되어야 할 인증키 파일

파일	설명
helloworld.pub	서버 운영체제에서 접속할 사용자 계정 홈 디렉토리의 <code>~/.ssh</code> 위치에 보관되어야 할 공개키 파일
authorized_keys	서버 운영체제에서 접속할 사용자 계정 홈 디렉토리의 <code>~/.ssh</code> 위치에 존재해야 하며 공개키 파일의 내용을 이 파일에 등록해야 한다. 인증키나 공개키 파일이름에 상관 없이 고유한 파일명을 갖는다.

2) 서버 운영체제에서 인증서 생성하기

서버 운영체제에 인증키 생성을 원하는 계정으로 로그인한 후 아래의 명령어를 수행한다.

```
$ ssh-keygen
```

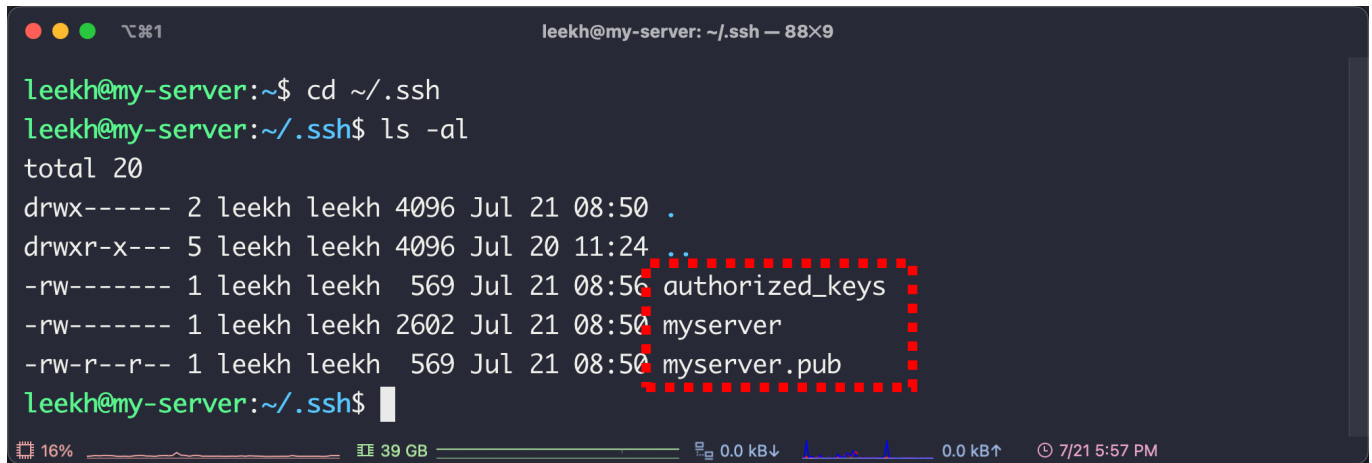
```
leekh@my-server: ~ — 88x22
leekh@my-server:~$ ssh-keygen 1
Generating public/private rsa key pair.
Enter file in which to save the key (/home/leekh/.ssh/id_rsa): /home/leekh/.ssh/myserver 2
Enter passphrase (empty for no passphrase): 3
Enter same passphrase again: 4
Your identification has been saved in /home/leekh/.ssh/myserver
Your public key has been saved in /home/leekh/.ssh/myserver.pub
The key fingerprint is:
SHA256:FakMnd4DTwiXKIYGNt5fdAIAtA+es16WrUsdBhxEqvU leekh@my-server
The key's randomart image is:
+---[RSA 3072]-----+
|o*+*..o*..=..|
|o 0 + +.0 o.|
| *. = . = *.|
|o.+o . +.+|
|. + .E+ S .|
| o = .|
| . = o|
|. + .|
|. o.|
+---[SHA256]-----+
leekh@my-server:~$
```

1. 명령어 입력
2. 인증키 파일의 경로 입력 (따로 입력하지 않을 경우 id_rsa로 파일명이 고정된다. 다른 인증키와 구분하기 위해서는 가급적 파일이름을 지정해 주는 것이 좋다.)
3. 인증키 비밀번호 (입력하지 않고 엔터)
4. 인증키 비밀번호 확인

인증키 파일을 클라이언트가 보관하고 인증키를 접속하려는 대상(서버 운영체제, github등)에 보관하는 원리이다.

3) 생성된 인증키 확인

~/**.ssh** 폴더 안에 인증키가 생성된 것을 확인한다.



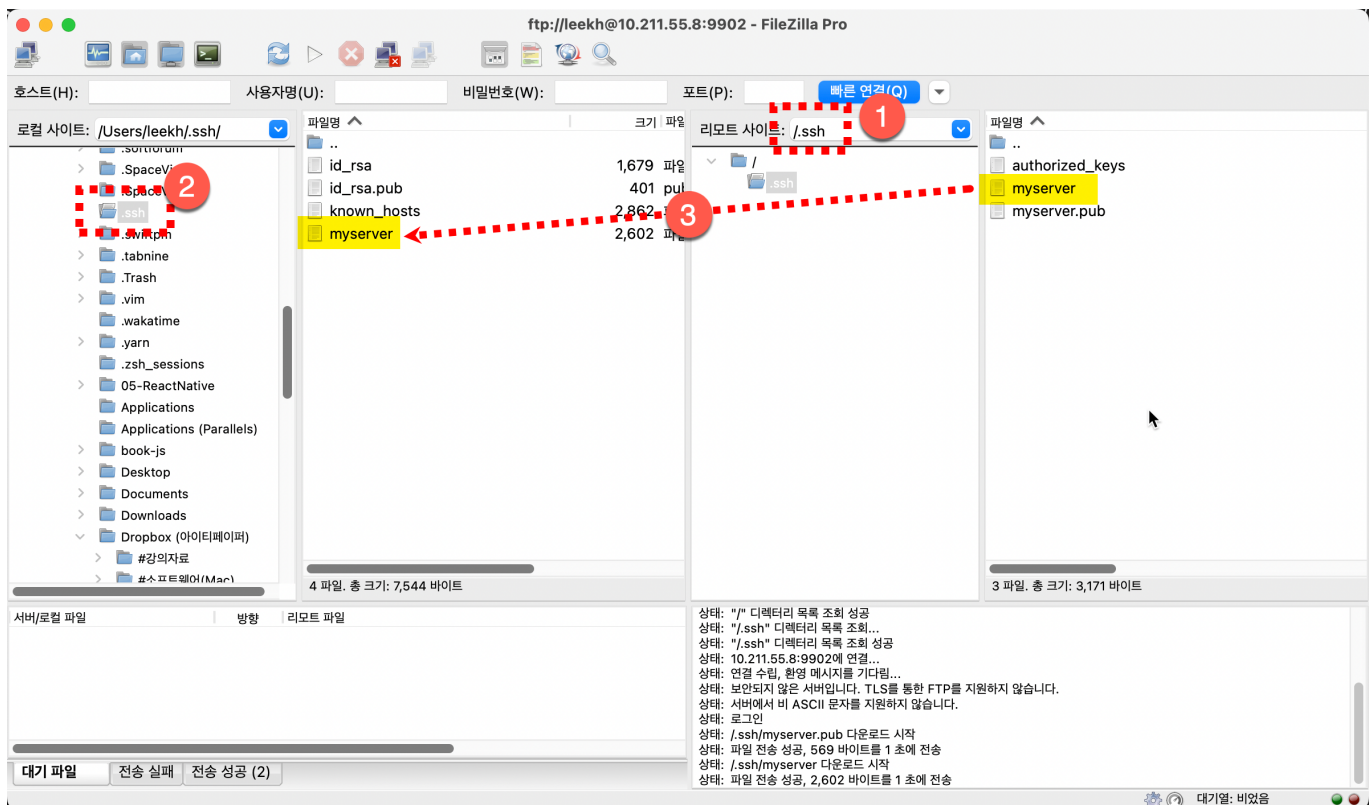
```
leekh@my-server: ~/$ cd ~/.ssh
leekh@my-server: ~/.ssh$ ls -al
total 20
drwx----- 2 leekh leekh 4096 Jul 21 08:50 .
drwxr-x--- 5 leekh leekh 4096 Jul 20 11:24 ..
-rw----- 1 leekh leekh 569 Jul 21 08:50 authorized_keys
-rw----- 1 leekh leekh 2602 Jul 21 08:50 myserver
-rw-r--r-- 1 leekh leekh 569 Jul 21 08:50 myserver.pub
leekh@my-server: ~/.ssh$
```

4) 공개키 파일의 내용을 **authorized_keys**에 등록

```
$ cat ~/.ssh/myserver.pub >> ~/.ssh/authorized_keys
```

5) 생성된 인증키 파일을 클라이언트 운영체제에 내려받기

FTP 클라이언트를 사용하여 생성된 인증키 파일을 내려받는다.



1. ~/**.ssh** 디렉토리는 숨김 상태이므로 FTP 클라이언트에 표시되지 않는다. 직접 폴더 경로를 입력해서 이동해야 한다.
2. 클라이언트 운영체제의 사용자 홈 디렉토리내에 있는 **.ssh** 폴더에 다운로드 받도록 FTP 클라이언트를 조정한다.
3. 인증키 파일을 내려받는다.

6) Mac 운영체제에서 인증키를 내려받은 경우

만약 **.ssh** 폴더가 없다면 직접 생성한다.

```
$ mkdir ~/.ssh
```

폴더 생성 후 이 폴더의 접근 퍼미션을 **0700**으로 설정해야 한다. (윈도우에서는 퍼미션 설정을 생략한다.)

```
$ chmod 0700 ~/.ssh
```

~/.ssh/myserver 위치로 인증키를 내려받았다면 인증키 파일의 접근 퍼미션을 **0600**으로 설정해야 한다. (윈도우에서는 퍼미션 설정을 생략한다.)

```
$ chmod 0600 ~/.ssh/myserver
```

```
leekh@leekh-MacBookPro:~/.ssh — 88x23
→ cd .ssh

~/.ssh
→ ls -l
total 32
-rw-----@ 1 leekh  staff  1679  9 26  2018 id_rsa
-rw-----@ 1 leekh  staff   401  9 26  2018 id_rsa.pub
-rw-----  1 leekh  staff  2862  7 11  15:31 known_hosts
-rw-r--r--  1 leekh  staff  2602  7 21  17:52 myserver

~/.ssh
→ chmod 600 myserver

~/.ssh
→ ls -l
total 32
-rw-----@ 1 leekh  staff  1679  9 26  2018 id_rsa
-rw-----@ 1 leekh  staff   401  9 26  2018 id_rsa.pub
-rw-----  1 leekh  staff  2862  7 11  15:31 known_hosts
-rw-----  1 leekh  staff  2602  7 21  17:52 myserver

~/.ssh
→
```

13% 39 GB 1.0 kB↓ 1.0 kB↑ 7/21 5:59 PM

7) 클라이언트 운영체제에서 접속 확인

아래의 명령어로 ssh 접속을 시도한다. 인증서가 정상적으로 식별되었다면 비밀번호 입력 없이 바로 로그인이 진행된다.

```
$ ssh 계정명@리눅스아이피 -p포트번호 -i인증키경로
```

```
leekh@my-server: ~ — 88x29
~
→ ssh leekh@10.211.55.8 -p9901 -i ~/.ssh/myserver
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-41-generic aarch64)

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

System information as of Thu Jul 21 09:00:12 AM UTC 2022

System load:            0.0
Usage of /:              19.8% of 29.82GB
Memory usage:           14%
Swap usage:             0%
Processes:              110
Users logged in:        0
IPv4 address for enp0s5: 10.211.55.8
IPv6 address for enp0s5: fdb2:2c26:f4e4:0:21c:42ff:fe50:80c1

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.

Last login: Thu Jul 21 09:00:12 2022 from 10.211.55.2
leekh@my-server:~$
```

클라이언트 운영체제가 Mac인 경우 이 명령어를 쉘 초기화파일에 alias로 등록하면 더욱 간편하게 사용할 수 있다.