

# Linux 보충 자료

# 여러 서버에 접속할 수 있는 SSH Key 생성하기

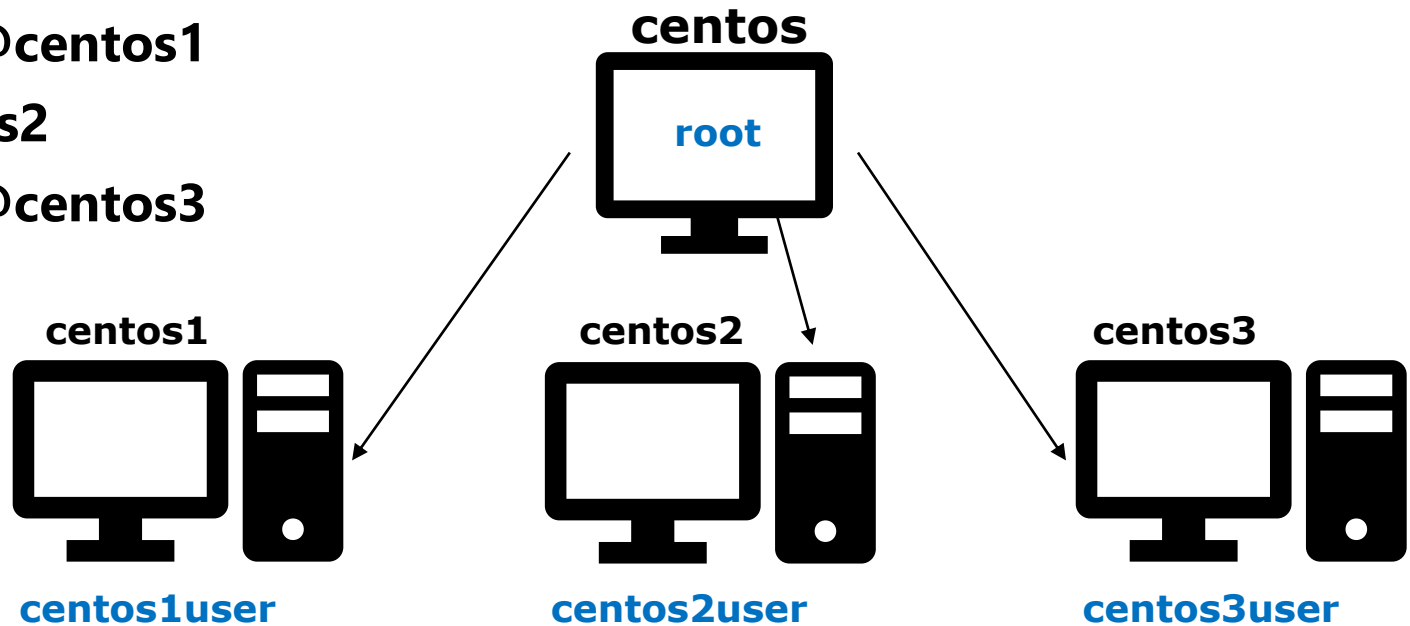
- 암호 대신 SSH Key로 인증하기
  - 원격 컴퓨터에 접속하기 위해서는 인증 단계를 통과해야 하는데, 보통은 id와 password로 인증하게 된다
  - 암호화 되지 않은 password를 사용하는 경우에는 암호 유출 위험이 있어서 암호화된 인증키(certificate)를 사용하는 것이 보안상 유리하다
  - 또한 로컬 컴퓨터에 인증키(Private Key)를 저장해 놓고 원격 서버에 접속하면 다른 컴퓨터에서는 인증키가 저장되어 있지 않기 때문에 원격 서버 접속을 할 수 없으므로 암호 보다는 보안성이 높다
  - 보통 원격 서버들의 id와 password는 다르다(단, root 계정 이름은 같지만 암호는 당연히 다르다)
  - 로컬 컴퓨터와 원격 컴퓨터의 id와 암호가 모두 같은 경우에는 ssh key 인증 구현이 쉽다. 예를 들면 root 계정에 대한 암호가 모두 1이면 다음과 같이 설정하면 된다.
    - **ssh-keygen**
    - **ssh-copy-id root@centos1**
    - **ssh centos1**

# 여러 서버에 접속할 수 있는 SSH Key 생성하기

- 암호 대신 SSH Key로 인증하기
  - 그런데 로컬 컴퓨터와 원격 컴퓨터의 root 계정의 암호가 다르거나 심지어 사용자 계정까지 다를 경우에는 ssh key로 인증하기 위해서는 몇가지 추가적인 작업이 필요하다
  - 무엇보다 로컬 컴퓨터에 원격 컴퓨터용의 public key와 private key를 생성해 두고, 각 원격 컴퓨터에 각 사용자에게 대한 public key를 복사해 두어야 한다
    - **ssh-keygen -C "centos1 user"**
    - **ssh-copy-id -i ~/.ssh/centos1user.pub centos1user@centos1**
  - 그리고 원격 접속을 할 때는 원격 컴퓨터의 이름과 사용자 계정과 Private key 정보를 입력해야 한다
    - **ssh -i ~/.ssh/centos1user centos1user@centos1**
  - 원격 컴퓨터의 사용자에게 대한 Private key 파일 정보를 입력하지 않는 것이 제일 편리한데, 그렇게 하기 위해서는 ~/.ssh/config 파일을 만들어서 해당 Private key 정보를 미리 입력을 해두면 된다
    - 각각 다른 원격 컴퓨터의 다른 사용자에게 대하여 인증키로 접속하기 위해서는 이 방법으로 해야 한다

# 여러 서버에 접속할 수 있는 SSH Key 생성하기

- 원격 컴퓨터와 동일한 root 계정과 암호 사용으로 접속하기
  - 로컬 컴퓨터에 root 계정으로 로그인 되어 있다
  - **ssh-keygen**
  - **ssh-copy-id** root@centos1
  - **ssh-copy-id** root@centos2
  - **ssh-copy-id** centos3
- **ssh root@centos1**
- **ssh centos2**
- **ssh root@centos3**



# 여러 서버에 접속할 수 있는 SSH Key 생성하기

- 원격 컴퓨터와 다른 계정과 암호 사용으로 접속하기-1
  - 원격 컴퓨터의 각 사용자에게 대한 Public key와 Private key를 생성한다
    - **ssh-keygen -C "centos1 user"**
      - `/root/.ssh/centos1user` 입력
    - **ssh-keygen -t ed25519 -C "centos2 user"**
      - `/root/.ssh/centos2user` 입력
      - 암호 알고리즘을 rsa가 아니 ed25519 사용
    - **ssh-keygen -t ed25519 -C "centos3 user"**
      - `/root/.ssh/centos3user` 입력
    - **ls -l /root/.ssh**

**default 암호화 기술: rsa**  
**ssh-keygen -t rsa**

```
[root@centos7 ~]# ls -la ~/.ssh/
total 40
drwx----- 2 root root 197 May 24 09:48 .
dr-xr-x---. 4 root root 159 May 24 08:38 ..
-rw----- 1 root root 1675 May 24 09:28 centos1user
-rw-r--r-- 1 root root 394 May 24 09:28 centos1user.pub
-rw----- 1 root root 399 May 24 09:29 centos2user
-rw-r--r-- 1 root root 94 May 24 09:29 centos2user.pub
-rw----- 1 root root 399 May 24 09:30 centos3user
-rw-r--r-- 1 root root 94 May 24 09:30 centos3user.pub
```

# 여러 서버에 접속할 수 있는 SSH Key 생성하기

- 원격 컴퓨터와 다른 계정과 암호 사용으로 접속하기-1
  - 원격 컴퓨터의 각 사용자에게 대한 Public key를 원격 컴퓨터에 복사한다
    - **ssh-copy-id -i ~/.ssh/centos1user.pub centos1user@centos1**
    - **ssh-copy-id -i ~/.ssh/centos2user.pub -p 22 centos2user@centos2**
    - **ssh-copy-id -i ~/.ssh/centos3user.pub centos3user@centos3**
  - 원격 컴퓨터의 각 사용자에게 대한 private key를 가지고 접속한다
    - **ssh -i ~/.ssh/centos1user centos1user@centos1**
    - whoami
    - exit
    - **ssh -i ~/.ssh/centos2user centos2user@centos2**
    - **ssh -i ~/.ssh/centos3user centos3user@centos3**
    - ##암호를 입력하지 않고 접속이 된다

# 여러 서버에 접속할 수 있는 SSH Key 생성하기

- 원격 컴퓨터와 다른 계정과 암호 사용으로 접속하기-2
  - 원격 컴퓨터의 각 사용자에게 대한 **Private key**를 사용하지 않고 접속하려고 한다
  - **ssh-add** 명령어를 사용하여 Private key 정보 없이 로그인이 되지만 나중에 다른 세션으로 접속할 때는 안되는 단점이 있다

- **eval \$(ssh-agent)**

- **ps aux | grep 9189**

- **ssh-add ~/.ssh/centos**

- **ssh-add ~/.ssh/centos2user**

- **ssh-add ~/.ssh/centos3user**

- **ssh centos1user@centos1**

- **ssh centos2user@centos2**

- **ssh centos3user@centos3**

- **su - root** (##다시 root 계정으로 로그인한다)

- **eval \$(ssh-agent)** (##9189가 아니다)

- **ssh centos1user@centos1** (##접속 실패)

```
[root@centos7 ~]# eval $(ssh-agent)
Agent pid 9189
[root@centos7 ~]# ps aux | grep 9189
root      9189  0.0  0.0  72552  784 ?        Ss   09:38   0:00 ssh-agent
root      9191  0.0  0.0  112812  980 pts/0    R+   09:38   0:00 grep --col
```

# 여러 서버에 접속할 수 있는 SSH Key 생성하기

- 원격 컴퓨터와 다른 계정과 암호 사용으로 접속하기-3
  - 원격 컴퓨터에 접속할 때 명령어에 영구적으로 Private key 정보를 입력하지 않고 로그인할 필요가 있다
  - 그렇게 하려면 Private key 정보가 들어 있는 구성 파일(~/.ssh/config)을 만들어야 한다
  - ~/.ssh/config 파일 생성하기
    - **cd ~/.ssh/**
    - **vi config**

<config 파일 다운로드하기>

**cd** ~/.ssh/

**wget** <http://down.cloudshell.kr/linux/config>

```
Host centos1
  Hostname centos1
  User centos1user
  IdentityFile ~/.ssh/centos1user

Host centos2
  Hostname centos2
  User centos2user
  Port 22
  IdentityFile ~/.ssh/centos2user

Host centos3
  Hostname centos3
  User centos3user
  IdentityFile ~/.ssh/centos3user
```



# 여러 서버에 접속할 수 있는 SSH Key 생성하기

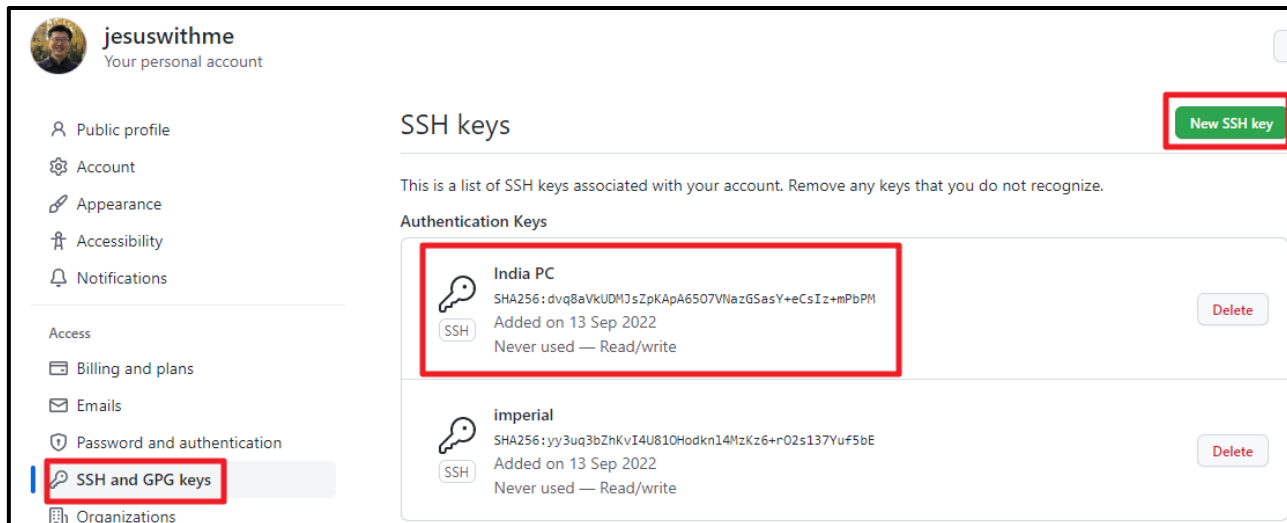
- 원격 컴퓨터와 다른 계정과 암호 사용으로 접속하기-3
  - 원격 컴퓨터의 각 사용자로 원격 컴퓨터에 접속한다
    - **eval \$(ssh-agent)**
    - **ssh centos1user@centos1**
    - **ssh centos2user@centos2**
    - **ssh centos3user@centos3**
      - ##접속이 잘 된다
    - ## ~/.ssh/config 파일을 만들면 private key 정보를 입력하지 않고서도 저장된 정보를 이용하여 로컬에 있는 private key와 원격의 public key를 비교하여 암호대신 인증에 사용하여 로그인한다
    - **su - root** (##다시 로컬 컴퓨터에 root 계정으로 로그인한다)
    - **eval \$(ssh-agent)** (##위의 것과 다르다)
    - **ssh centos1user@centos1**
    - **ssh centos2user@centos2**
    - **ssh centos3user@centos3**
      - ##여전히 접속이 잘 된다

# SSH Key로 접속하는 구체적인 서비스

- Private Key와 Public Key로 접속하는 서비스

## 1. <https://github.com>

- 2021년 8월 경부터 github.com에 파일을 업로드할 때 인증 처리는 **SSH key**로만 하도록 변경되었다
- 로컬 컴퓨터에서 Public Key와 Private Key를 만든 후에 **Public Key** 내용을 복사하여 **github.com**에 추가해야 한다



- `git remote add origin git@github.com:jesuswithme/pr.git`
- `git push origin master` (##이렇게 파일을 업로드할 때 인증 절차를 거친다)

# SSH Key로 접속하는 구체적인 서비스

- Private Key와 Public Key로 접속하는 서비스

## 2. Azure Linux VM 생성

- Azure Cloud에서 Linux VM을 생성할 때 인증을 암호 방식과 SSH Key 방식이 있다
- 보안성을 높이기 위해서는 SSH Key 방식을 사용하는데 VM을 만들 때 Private key와 Public Key를 생성하게 되고, 그 중에서 Private Key를 로컬 컴퓨터로 다운로드하여 SSH Client로 접속할 때 사용한다
- SSH Client는 다양한 것이 있는데 Linux나 Windows10/11에 기본 내장된 것을 사용하거나 MobaXterm, Putty 같은 프로그램에서 Private Key를 등록해서 Azure Cloud에 있는 Linux VM에 접속하게 된다

Administrator account

Authentication type ⓘ

☒ SSH public key

☐ Password

ⓘ Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username \* ⓘ

azureuser

SSH public key source

Generate new key pair

Key pair name \*

linuxsshserver\_key

Generate new key pair

ⓘ An SSH key pair contains both a public key and a private key. **Azure doesn't store the private key.** After the SSH key resource is created, you won't be able to download the private key again. [Learn more](#)

Download private key and create resource

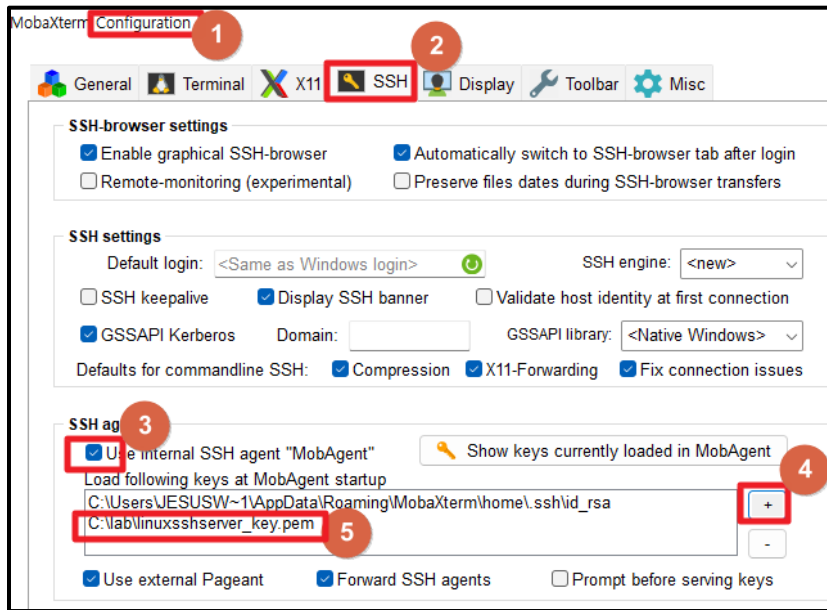
Return to create a virtual machine

# SSH Key로 접속하는 구체적인 서비스

- Private Key와 Public Key로 접속하는 서비스

## 2. Azure Linux VM 생성

- Private Key 파일을 C:\Lab\ 폴더에 다운로드함
  - linuxsshserver\_key.pem
- MobaXterm 프로그램에서 SSH Private Key 등록하기



암호 대신 ssh key로 로그인하기

```
[jesuswithme.imperial] > ssh azureuser@20.63.156.70
Warning: Permanently added '20.63.156.70' (RSA) to the list of known hosts.
X11 forwarding request failed on channel 0
[azureuser@linuxsshserver ~]$
```