SSH 구성 및 활용
보충 자료

SSH 구성 및 활용

- SSH Server 개요
- SSH Client 설치 및 SSH Server에 접속하기
- 암호 대신 Key를 사용하여 SSH Server에 접속하기
- 여러 서버에 접속할 수 있는 SSH Key 생성하기
- SSH Key로 접속하는 구체적인 서비스
- SSH Server의 default port 변경하기
- SSH Server에 접속하자 마자 작업하고 빠져 나오기

SSH Server 개요

- SSH 기술 이해
  - SSH(Secure Shell)는 네트워크 보안에 있어 대중적이고 강력한 접근 방식을 제공하기 위해 만들어진 프로토콜이다.
  - SSH에 기반한 제품은 서버와 클라이언트의 한 쌍으로 구성되어 있는데, 보통 사용자들은 SSH 클라이언트를 rsh (remote shell) 대용이나 telnet 대용으로 사용하고 있다.
  - 로컬 컴퓨터와 원격 컴퓨터간의 통신할 때 Secure Channel(Tunnel)을 만들어 그곳으로 패킷을 전달하므로 안전한 통신을 보장한다

- SSH가 제공하는 4가지 기능
  - 인증(Authentication)
  - 암호화(Encryption)
  - 무결성(Integrity)
  - 압축(Compression)

SSH Server 개요

- SSH Server 실행 여부 확인
  - service sshd status
- SSH Server 설치하기
  - yum install openssh-server
- SSH client 설치하기
  - yum install openssh-clients

SSH Client 설치 및 SSH Server에 접속하기

- Windows에 SSH Client 설치하기
  - MobaXterm(또는 Putty)
  - WinSCP

- SSH Server에 접속하기
  - Linux 및 Mac에는 SSH Client가 내장되어 있음
    - **ssh** ipaddress_of_RemoteServer
    - **ssh** ipaddress_of_RemoteServer **-l adminuser** (##소문자 L)
    - **ssh adminuser@**ipaddress_of_RemoteServer
  - Windows에서는 PowerShell 및 MobaXterm(또는 Putty)에서 접속하기

암호 대신 Key를 사용하여 SSH Server에 접속하기

- 암호 대신 Key를 사용하여 SSH Server에 접속하기
  - Key를 이용한 인증의 특징
    - 암호 방식보다 더 안전하다
    - Private key와 Public key 사용하여 인증 및 데이터 암호화 처리
    - SSH Client에서 Public key와 Private key를 생성하여 접속하는 모든 SSH Server들에게 동일한 Public Key를 복사해두어 편리하게 관리
    - Public Key는 ~/.ssh/authorized_keys안에 들어 있다
    - SSH Server에 복사한 Public Key에 대해서는 현재 로그인하여 작업중인 사용자인 adminuser만 액세스하는 권한만 부여한다. 왜냐하면 Public key를 복사할 때 adminuser의 Home Directory 밑에 위치한 ~/.ssh/authorized_keys가 위치가 정해져 있기 때문이다.
    - 인증서는 주로 관리자가 사용하는 일반 계정(adminuser)만 사용하는 경향이 있다. 일단 리눅스 서버에 접속할 때 쉽게 접속한 후 관리 작업할 때는 su root를 하여 root 계정에 대한 암호를 입력하는 것이 보안에 좋다

암호 대신 Key를 사용하여 SSH Server에 접속하기

- Public key 파일을 생성하여 원격 컴퓨터에 복사하여 두면 ssh server에 접속할 때 private key와 public key의 matching을 통하여 인증 절차를 통과하여 접속하게 한다
  - 사전에 원격 컴퓨터에 로컬 사용자와 **동일한 사용자가 존재해야 한다**
  - 로컬 사용자와 원격 사용자의 암호가 **동일할 필요는 없다**
  - 사실, 원격 컴퓨터에 존재하는 사용자 계정과 동일한 이름의 key 파일을 생성하여 원격으로 복사만 하면 된다
- 암호를 사용하지 않고 접속하는 것의 이점
  - 관리자가 사용하는 일반 계정(adminuser)으로 쉽게 접속 후 관리 작업할 때만 su를 사용하여 root에 대한 암호를 입력하면 편의성과 보안성을 모두 해결할 수 있다
  - rsync를 사용하여 백업을 할 때 ssh로 접속하여 원격 서버에 백업을 하게 되는 경우에 인증 문제를 쉽게 해결할 수 있다

암호 대신 Key를 사용하여 SSH Server에 접속하기

- 다음 절차대로 작업한다
  - 로컬 컴퓨터에서 ssh-keygen 명령을 사용하여 public key와 Private key를 생성한다
    - **ssh-keygen**
  - ssh-copy-id 명령을 사용하여 public key를 원격 컴퓨터에 복사한다
    - **ssh-copy-id root@192.168.219.250**
      - **(또는 ssh-copy-id -I ~/.ssh/id_rsa.pub 192.168.219.250)**
    - 이것은 원격 컴퓨터의 ~/.ssh/authorized_keys 파일에 public key 내용을 appen하는 것이다
    - -i는 identity 파일을 입력하라는 뜻
  - 암호를 입력하지 않고 원격 컴퓨터에 로그인하기
    - **ssh 192.168.219.250**

암호 대신 Key를 사용하여 SSH Server에 접속하기

- Windows의 Git이나 Linux에서 로컬에 없지만 원격에 있는 사용자 계정에 대한 Private key, Public key를 생성할 수 있다
  - 이것의 장점은 로컬 컴퓨터의 사용자는 전혀 신경 쓸 필요가 없고 단지 원격 컴퓨터의 사용자 계정에만 신경을 쓰면 된다
  - Windows azure의 vm에 접속할 때 매우 유용하다
- 다음 절차대로 작업한다
  - 로컬 컴퓨터에서 ssh-keygen 명령을 사용하여 원격 컴퓨터에 존재하는 사용자 계정의 public key와 Private key를 생성한다(-f: --filename)
    - **ssh-keygen -f** *azureuser*
  - ssh-copy-id 명령을 사용하여 원격 컴퓨터의 사용자 계정에게 public key를 복사한다
    - **ssh-copy-id azureuser@centos9191.cloudapp.net**
    - 이것은 원격 컴퓨터의 ~/.ssh/authorized_keys 파일에 public key 내용을 append하는 것이다
  - 암호를 입력하지 않고 원격 컴퓨터에 로그인하기
    - **ssh azureuser@centos9191.cloudapp.net**

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**

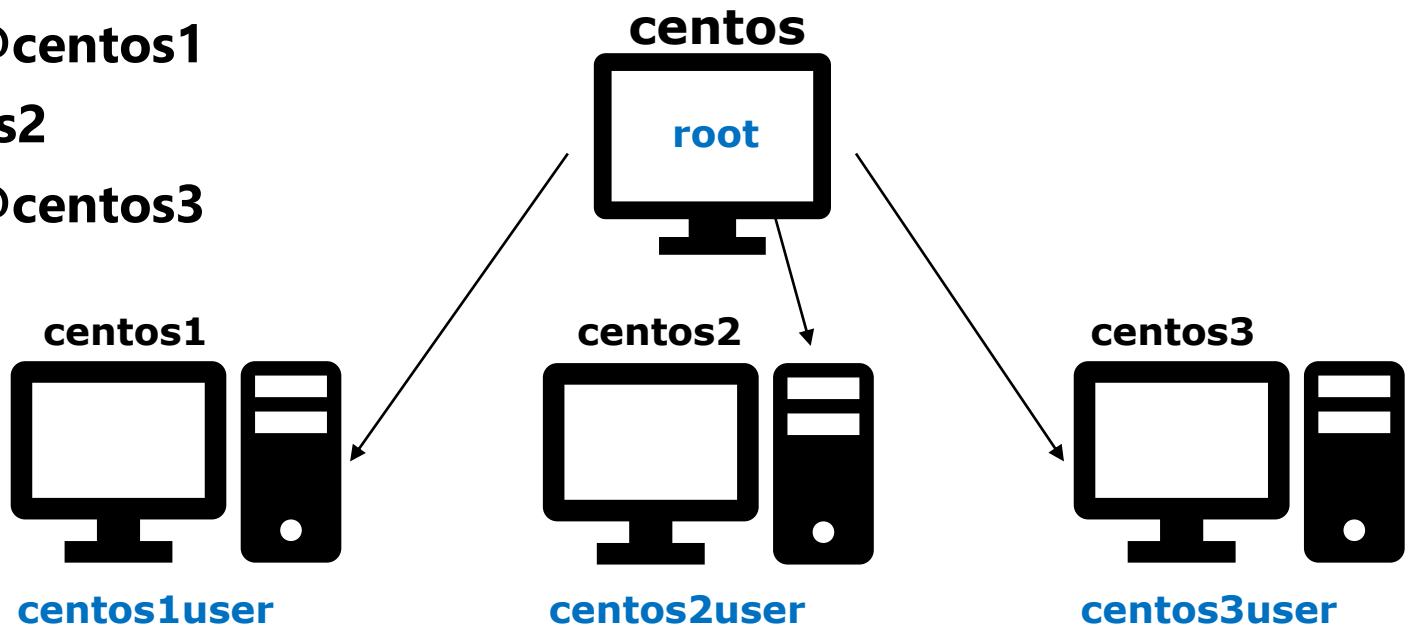
여러 서버에 접속할 수 있는 SSH Key 생성하기

- 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 사용자에 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 사용자에 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 사용자에 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 사용자에 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 알고리듬을 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 사용자에 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 사용자에 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 사용자에 대한 **Private key를 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
[root@centos7 ~]#
```

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 로그인을 할 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

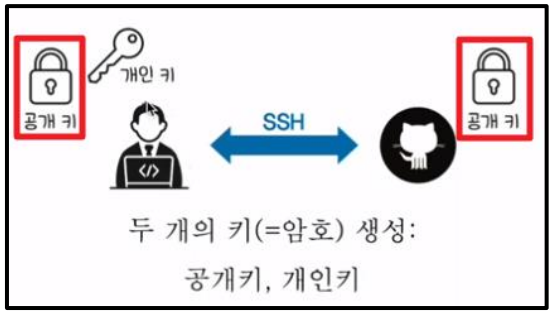
```
Host centos1
    Hostname centos1
    User centos1user
    IdentityFile ~/.ssh/centos1user

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

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

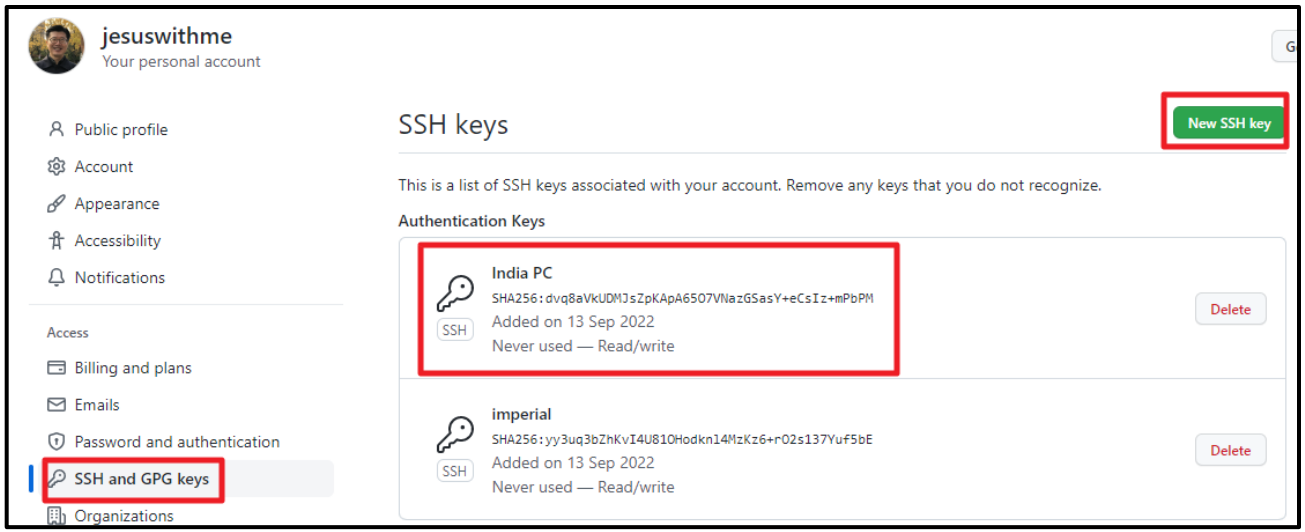
여러 서버에 접속할 수 있는 SSH Key 생성하기

- 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 key로만** 하도록 변경되었다
     - 로컬 컴퓨터에서 Public Key와 Private Key를 만든 후에 **Public Key 내용을 복사하여 github.com에 추가해야 한다**





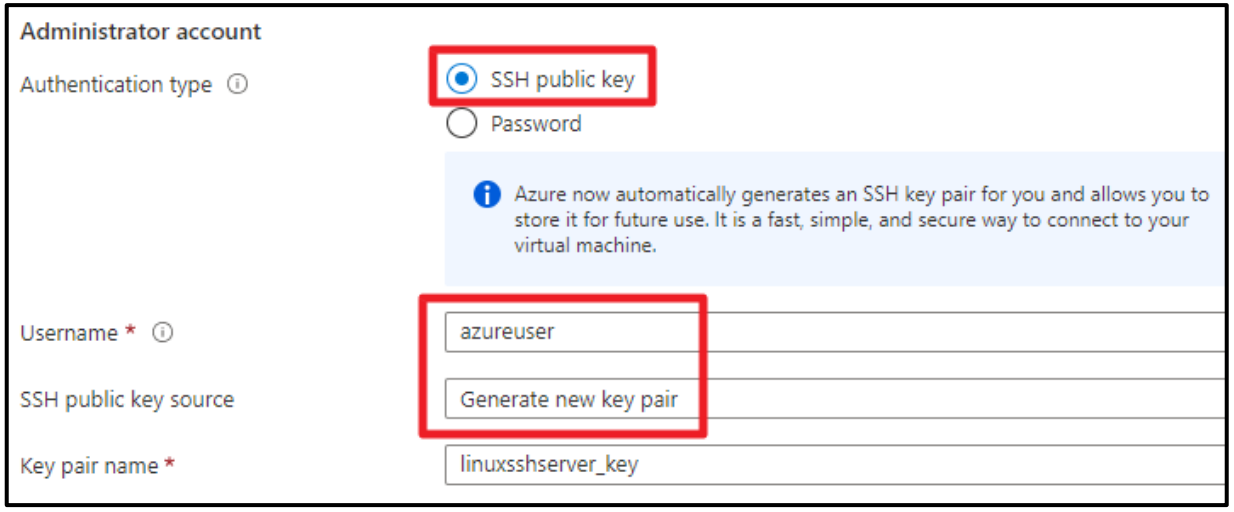
- **git remote add origin git@github.com:jesuswithme/pr.git**
- **git push origin master** (##이렇게 파일을 업로드할 때 인증 절차를 거친다)

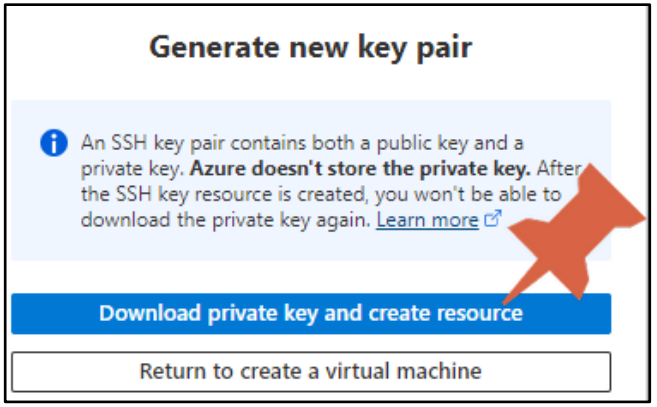
SSH Key로 접속하는 구체적인 서비스

- Private Key와 Public Key로 접속하는 서비스

2. Azure Linux VM 생성

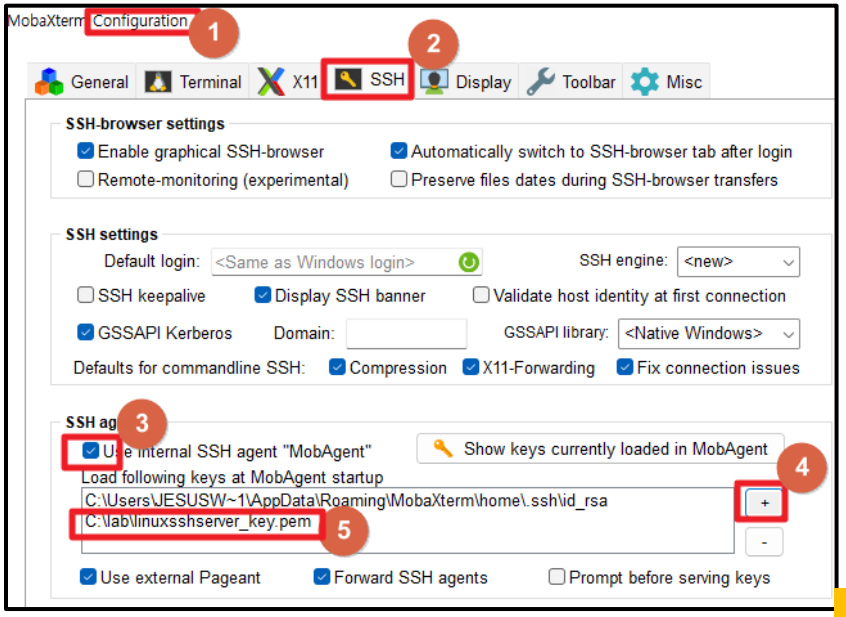
- Azure Cloud에서 Linux VM을 생성할 때 인증을 암호 방식과 SSH Key 방식이 있다
- 보안성을 높이기 위해서는 SSH Key 방식을 사용하는데 VM을 만들 때 Private key와 Public Key를 생성하게 되고, 그 중에서 Private Key를 로컬 컴퓨터로 다운로드하여 SSH Client로 접속할 때 사용한다
- SSH Client는 다양한 것이 있는데 Linux나 Windows10/11에 기본 내장된 것을 사용하거나 MobaXterm, Putty 같은 프로그램에서 Private Key를 등록해서 Azure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





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
X11 forwarding request failed on channel 0
[azureuser@linuxsshserver ~]$
```

SSH Server의 default port 변경하기

- 22번 포트를 사용하는 openssh-server를 보안을 위해서 포트 번호를 변경할 수 있다
- 다음과 같이 진행한다
  - **su root**
  - **rpm -qc openssh-server** (##특정 package의 설정 파일 찾기)
  - **vi /etc/ssh/sshd_config**
    - ##여기서 **#Port 22**를 **Port 2222**변경한다
  - **systemctl restart sshd** (##설정 파일이 변경되었기 때문에 서비스 재시작한다)
  - **원격 컴퓨터에서** 로컬로 다음과 같이 접속한다
    - **ssh 192.168.219.100 -l adminuser -p 2222**
  - 방화벽 때문에 접속이 되지 않으면 로컬에서 2222번 포트를 허용한다
    - **firewall-cmd --permanent --add-port=2222/tcp**
    - **firewall-cmd --reload** (## 수정된 설정 파일을 적용하기)
  - 원격에서 로컬로 다시 접속을 해본다
    - **ssh 192.168.219.100 -l adminuser -p 2222**

SSH Server에 접속하자 마자 작업하고 빠져 나오기

- 파일 및 Directory 생성하기
  - 파일 생성하기
    - **ssh** adminuser@ipaddress **‘touch myfile.txt‘**
  - 디렉터리 생성하기
    - **ssh** adminuser@ipaddress **'mkdir lab/‘**
  - 파일 및 디렉터리의 생성여부 확인하기
    - **ssh** adminuser@ipaddress **‘ls -l‘**

- 파일 내용 수정하기
  - /etc/resolv.conf에 DNS Server IP 주소 추가하기
    - ssh **root**@ipaddress **‘echo “nameserver 8.8.8.8” > /etc/resolv.conf’**
  - /etc/resolv.conf 파일 내용 수정 여부 확인하기
    - ssh **root**@ipaddress **‘cat /etc/resolv.conf’**

SSH Server에 접속하자 마자 작업하고 빠져 나오기

- 마운트 된 디스크 확인 및 디스크 사용량 확인하기
  - 마운트 된 디스크 확인하기
    - **ssh root**@ipaddress 'fdisk –l | grep /dev'
  - 디스크 사용량 확인하기
    - **ssh** root@ipaddress 'du –sh /home'
      ssh root@ipaddress 'du –sh /'
      ssh root@ipaddress 'du –sh ~adminuser'
      ssh root@1ipaddress 'du /home --max-depth=1 | sort –n –r'

SSH Server에 접속하자 마자 작업하고 빠져 나오기

- **scp**를 사용하여 파일 복사하기
  - 로컬 파일을 원격으로 복사하기
    - **touch** myfile.txt
      scp myfile.txt **adminuser@ipaddress:**
      (##만약 command not found라는 메시지가 나오면서 복사를 실패하면 원격 컴퓨터에서 yum install openssh-clients –y를 실행한 후 다시 작업하면 된다)
    - **scp** myfile.txt **root@ipaddress:/var/tmp**
  - 원격에 존재하는 파일을 로컬로 복사하기
    - **scp** adminuser@ipaddress:myfile.txt **./myfile2.txt**
      ls –l ~adminuser

SSH Server에 접속하자 마자 작업하고 빠져 나오기

- **scp**를 사용하여 디렉터리 내용 몽땅 복사하기
  - 디렉터리 내용 몽땅 복사하기
    - 디렉터리 생성하기
      mkdir -p ~adminuser/lab/sales
    - 생성된 디렉터리에 파일 생성하기
      fallocate -l 10m ~adminuser/lab/sales/10MB.img (##-l은 length)
      ls –l lab/sales/
    - 디렉터리 내용 몽땅 원격 서버에 복사하기
      scp **-r ~adminuser/lab/** adminuser@ipaddress:/home/adminuser/
      (##-r은 recursive)
      ssh adminuser@ipaddress '**ls –l** ~adminuser/lab/'
      ssh adminuser@ipaddress '**tree** ~adminuser/lab/'
      (##원격 컴퓨터에 tree 명령어가 없으면 yum install tree -y로 설치한 후 다시 작업을 한다)

Root 계정 사용 못하게 막기

- 보안상 이유로 root 계정을 사용하지 못하게 할 수 있다
- 대신 wheel 그룹에 넣은 계정으로 sudo 명령으로 관리자 권한을 행사하게 한다
  - **sudo passwd -dl root**
    - -d: root 계정에 대한 암호 삭제하기
    - -l: root 계정 잠그기