원격 SSH 접속 및 MySQL Database 활용

Park Cheon Bok

Client

- Windows 11

- Python 3.12

Server

- Ubuntu 24.04

- MySQL Server 8.0.39

1. Windows 환경변수에 존재하는 값으로 SSH 접속 후 MYSQL\_ID, MYSQL\_PW 값 가져오기.

- 구현 Module은 “Connection\_ssh.py” 파일명.

- SSH 접속 필수 값

① HOSTNAME : IP (192.168.3.4)

② PORT : 22 (SSH 기본 포트)

③ USERNAME : 접속 Server Name

④ PASSWORD : 접속 Server Password ( 이 값을 Windows 환경변수에서 가져옴 )

- 환경 변수 ‘ssh\_pw’에 기록된 값을 가져와서 접속할 때 사용.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure 1. 환경변수 값 확인

- Python에서 SSH 접속 및 명령 수행을 위해 ‘paramiko’ Library를 준비한다.

ㆍ python -m pip install --upgrade pip

ㆍ pip install paramiko

1) Windows의 환경변수에서 ssh\_pw의 값 가져오기.

- import os

- os Library의 getenv(“value”) Function을 적용해 환경변수 값을 가져올 수 있다.

- 환경변수 값을 가져오는 function을 정의해 사용할 수 있도록 작성.

|  |
| --- |
| def get\_ssh\_password():  pw = os.getenv('ssh\_pw')  return pw |

2) Ubuntu Server에 접속할 때 필요한 매개변수 받기.

- “Connection\_ssh.py” 모듈 파일에서 Ubuntu Server에 접근하기 위한 고정 값을 가지고 있다.

- Client가 모듈을 통해 Function call로 값들을 받을 수 있도록 작성한다.

|  |
| --- |
| def get\_ssh\_info() :  info = {'ID':'pcb', 'IP':'192.168.3.4', 'PORT':22}  return info |

3) Ubuntu Server에 접속 후 Mysql Server Login에 필요한 ID와 PW를 가져온다.

- import paramiko

- connect( Hostname, Port, Username, Password ) Function으로 Ubuntu Server와 SSH 연결 시도.

- Session 연결 상태에서 exec\_command(“Command”) Function으로 Terminal 명령 수행 가능.

ㆍ exec\_command()를 사용할 때마다 새로운 Session으로 생성되어 의도하지 않은 결과가 발생할 수 있다.

ㆍ exec\_command() 한 번의 수행으로 원하는 명령을 모두 처리할 수 있도록 작성해야 한다.

- Session 연결된 시점에서 Ubuntu의 환경변수에는 ID와 PW 값이 export된 상태가 아니다.

- Shell Script를 실행해야 Ubuntu 환경변수에 Mysql Server Login에 필요한 ID와 PW를 export할 수 있다.

ㆍShell Script에는 환경변수를 등록하는 코드를 작성한다.

ㆍexec\_command()의 매개변수로 넣을 명령어는 Shell Script 실행과 환경변수를 확인하는 명령어를 합친 것이다.

- 환경변수는 “MYSQL\_ID”와 “MYSQL\_PW” key로 export한다.

- 환경변수 목록에서 key 값을 찾았다면 value를 추출함으로써 ID와 PW를 획득한다.

▶ Ubuntu Server에서 Shell Script 작성.

- 환경변수 ‘MYSQL\_ID’, ‘MYSQL\_PW’를 export.

- sh 배치 경로 “~/working” Directory.

- 파일명 : set\_mysql\_env.sh

- Shell Script를 실행하는 ‘pcb’ 계정만 파일을 다룰 수 있도록 접근 제한 설정.

ㆍ sudo chown pcb:pcb set\_mysql\_env.sh

ㆍ sudo chmod 700 set\_mysql\_env.sh

- 아래 Script에서 ID와 PW 부분은 빨간 마크 처리했으므로 위치 확인.

|  |
| --- |
| #!/bin/bash  #1 MYSQL\_ID 값이 있는지 확인 후 없으면 환경변수 추가  if [ "$MYSQL\_ID" != '[ID]' ]; then  export MYSQL\_ID=[ID]  echo "MYSQL\_ID is created!!"  else  echo "MYSQL\_ID is already exists...!"  fi  #2 MYSQL\_PW 값이 있는지 확인 후 없으면 환경변수 추가  if [ "MYSQL\_PW" != '[PW]' ]; then  export MYSQL\_PW=[PW]  echo "MYSQL\_PW is created!!"  else  echo "MYSQL\_PW is already exists...!"  fi  source ~/.bashrc  #환경변수 설정 확인  echo "MYSQL Environment Setting Complete" |

Figure 2. Ubuntu Server의 set\_mysql\_env.sh code

▶ Host Key 등록.

- SSH Protocol은 Client가 Server에 접속할 때 Host Key를 확인해 ‘신뢰’할 수 있는 Server인지 검사한다.

- Server에서 Host Key는 “/etc/ssh” Directory 위치에서 확인할 수 있다.

ㆍ RSA key : ssh\_host\_rsa\_key.pub

ㆍ ECDSA key : ssh\_host\_ecdsa\_key.pub

ㆍ ED25519 key : ssh\_host\_ed25519\_key.pub

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure 3. Ubuntu Server Host Key

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure 4. ssh\_host\_rsa\_key.pub 확인 (RSA Host key)

- Client는 신뢰할 수 있는 Host의 Key를 사전에 등록해야 한다.

ㆍ Windows에서 “~/.ssh/known\_hosts”에 등록함으로써 Host Key를 관리할 수 있다.

- Windows에서 Server의 Host key를 확인하려면 “ssh-keygen” 명령을 사용한다.

ㆍ ssh-keygen -F [ Server IP 또는 HostName ]

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure 5. Client에서 Ubuntu Server(192.168.3.4)의 Host key 확인

- ssh-keygen 명령으로 확인한 Server의 Host key를 Client의 ~/.ssh/known\_hosts 파일에 추가(복사)한다.

ㆍ WSL 환경이 아니라면 직접 경로를 찾아 메모장으로 열어 수정한다.

ㆍ WSL 환경이라면 편집기로 수정한다 ( nano ~/.ssh/known\_hosts )

ㆍ “#”이 포함된 주석 부분은 추가하지 않아도 된다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

Figure 6. Client의 known\_hosts 파일에 Host key 추가

- (번외) paramiko 모듈로 자동 Host key 등록이 가능하다.

ㆍ 검증 절차를 생략하고 “새로 연결된 서버를 전적으로 신뢰”하기 때문에 보안 취약점이 될 수 있다.

|  |
| --- |
| **import paramiko**   # pip install paramiko  **SSH = paramiko.SSHClient()** # ssh 연결 객체 생성  # Client 위치에서 Host key가 등록되어 있지 않을 때, 자동으로 추가.  **SSH.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())** |

Figure 7. Host key가 등록되지 않았을 때, 자동으로 추가해주는 paramiko Function

▶ SSH 연결 및 명령어 수행.

- known\_hosts 파일에 Ubuntu server의 Host key가 추가되었다면 SSH 연결 Code 작성을 진행한다.

- 처리 절차는 다음과 같다.

① SSH Connection을 위한 SSH Object를 paramiko Module을 통해 생성한다.

② Server 검증에 필요한 Host key를 known\_hosts 파일에서 불러와 SSH Object에 Load.

③ Host Key Load가 되었는지 확인 후, SSH 연결을 시도한다.

④ 연결을 완료했다면 Server에서 MYSQL\_ID, MYSQL\_PW 환경변수를 추가하는 shell script를 실행하고,

추가된 이후의 환경변수 목록에서 “MYSQL\_ID”, “MYSQL\_PW”와 매칭되는 키워드를 찾으면 해당 값을 반환한다.

⑤ 모든 처리가 완료되면 처리 상태 코드(status), MYSQL\_ID값(mysqlid), MYSQL\_PW값(mysqlpw)을 return.

-

|  |
| --- |
| Import paramiko  def connect\_ssh(hname, hport, husername, hpassword):      SSH = paramiko.SSHClient()   # SSH Object 생성      mysqlid = None   # MYSQL\_ID 값 변수      mysqlpw = None   # MYSQL\_PW 값 변수      status  = None   # 처리 상태 코드 (200:OK, 400:명령어오류, 500:서버연결오류)      HOST\_KEY = r'C:\Users\phant\.ssh\known\_hosts'   # Windows의 known\_hosts 파일 경로        # Automatic Host key registration (호스트 키 자동 Load는 아래의 코드 주석 해제 후 사용)      #SSH.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())    #known\_hosts 파일에 있는 Host key Load      SSH.load\_host\_keys(HOST\_KEY)        # known\_hosts 파일에 없는 Host key에 대한 Server 접속은 차단하는 정책 반영      SSH.set\_missing\_host\_key\_policy(paramiko.RejectPolicy())        try :          # SSH Object를 통해 SSH 연결          SSH.connect(hostname=hname, port=hport, username=husername, password=hpassword)            # Server에서 수행할 명령어 작성 (shell script 실행 + 환경변수 목록 확인)          COMMAND = "source ~/working/set\_mysql\_env.sh && env"          # 명령어 내용 전달 및 결과 받기          stdin, stdout, stderr = SSH.exec\_command(COMMAND)            output = stdout.read().decode().strip() #명령 후 prompt에서 나온 결과 text 받음          errors = stderr.read().decode().strip() #명령 후 에러가 발생하면 받음            if errors: # 명령어 수행 중 오류 발생 유무에 따른 분기             sys.stdout.write(f"Error: {errors}\n")             status = '400'          else:              # Server의 환경변수 목록에서 키워드와 매칭된 경우 값 추출              for line in output.splitlines():                  if line.startswith('MYSQL\_ID='):                      mysqlid = line.split('=')[1]                  elif line.startswith('MYSQL\_PW='):                      mysqlpw = line.split('=')[1]              status = '200'        except Exception as e :          sys.stdout.write(f"Connection Fail.\nError:{e}")          status = '500'        finally :          SSH.close()          return status, mysqlid, mysqlpw |

Figure 8. SSH 연결 connect\_ssh() Function 구현 Code

- Test 수행 ( 다른 .py 코드에서 Connection\_ssh를 import )

|  |
| --- |
| import Connection\_ssh as sshc  if \_\_name\_\_ == "\_\_main\_\_" :      svpw = sshc.get\_ssh\_password()      svinfo = sshc.get\_ssh\_info()       status, MysqlId, MysqlPw = sshc.connect\_ssh(svinfo['IP'],                                          svinfo['PORT'],                                          svinfo['ID'],                                          svpw)     print(status, MysqlId, MysqlPw)  출력: |

Figure 9. Mysql Server Login에 필요한 ID와 PW 획득 Test 결과