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 가져오기.**

- 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

2. MySQL Server SSL 관련 파일 관리.

**1) MySQL SSL 연결 구성 파일 확인.**

- Ubuntu Server에 Mysql Server가 설치되어 있다는 전제로 시작한다.

|  |
| --- |
| # MySQL Server 설치  sudo apt update && sudo apt upgrade -y  sudo apt install mysql-server -y |

- mysql Query로 SSL value 값에 대해 확인한다.

ㆍ mysql -u root -p -e "SHOW VARIABLES LIKE '%ssl%';"

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Figure 10. Mysql Server의 SSL Key 구성 확인

- SSL 인증서 관련 파일

ㆍ CA 인증서 (ssl\_ca) : 인증기관(CA)에서 개인 키로 서명한 인증서.

ㆍ 서버 인증서 (ssl\_cert) : CA가 서명한 Server의 인증서

ㆍ 서버 키 (ssl\_key) : 서버(Mysql)의 개인 키.

- MySQL에서 사용 중인 CA 인증서와 서버 인증서가 저장된 경로 확인.

ㆍClient가 외부에서 SSL로 접속하기 위해서는 CA 인증서와 키(key)가 필요하다.

ㆍ경로 찾기가 어려운 경우에는 find 명령을 통해 파일명을 통한 탐색을 수행한다.

|  |
| --- |
| **sudo find / -name “ca.pem”** |

ㆍ파일명은 [Figure 10.]에서 mysql Query를 통해 알 수 있었다.

ㆍ여기에서는 “/var/lib/mysql/” 경로에 인증서 파일이 존재하는 것을 확인할 수 있었다.

- sudo ls /var/lib/mysql 명령 수행 결과

ㆍ ca-key.pem : 인증기관(CA)의 개인 키 파일.

ㆍ ca.pem : 인증기관(CA)이 개인 키로 서명한 인증서.

ㆍ server-cert.pem : 서버(Mysql)가 인증기관(CA)의 서명으로 발급받은 인증서.

ㆍ server-key.pem : 서버(Mysql)의 개인 키 파일.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Figure 11. MySQL Server에 생성된 인증기관(CA)의 인증서와 서버 인증서

3. Client의 SSL 연결에 필요한 인증서 파일 생성.

ㆍSSL 통신의 필수 절차는 ‘신뢰할 수 있는 인증기관(CA)’이 서명한 인증서를 제시할 수 있는지에 있다.

ㆍMySQL Server의 경우 CA의 서명을 통해 server-cert.pem 인증서를 발급 받았다.

ㆍClient의 경우, CA의 서명으로 발급받은 인증서가 아직 없다. 따라서 CA의 개인 키(ca-key.pem)로 서명한 인증서

를 발급받는 절차를 진행한다.

**1) Client 요구 파일**

ㆍ개인 키 (client-key.pem) : Client임을 입증하는 역할과 함께 인증서 요청에 필요.

ㆍCSR (Certificate Signing Request) 생성 (client-cert.csr)

🡪 인증서 발급을 위해 필요한 정보를 포함한 “인증서 신청 형식”의 데이터 파일.

ㆍ인증서 (client-cert.pem) : CA의 서명으로 발급받은 클라이언트의 인증서.

**2) openssl Package Install**

- 인증서 생성 및 관리 목적의 openssl 패키지 확인.

ㆍ보편적으로 openssl이 설치되어 있으므로 먼저 설치 상태를 확인 후, 설치되지 않았으면 진행.

|  |
| --- |
| $ **openssl version** 🡪 openssl 설치 확인 (버전이 나오면 설치된 상태)  $ **sudo apt update** 🡪 openssl 설치가 되어 있지 않다면 패키지 업데이트 후 설치  $ **sudo apt install openssl** |

**3) 클라이언트 개인 키, 인증서 생성**

- MySQL Server의 인증서 파일들은 /var/lib/mysql directory에 위치하므로 해당 경로로 이동해 작업을 진행한다.

- 관리자 권한으로 접근이 가능한 위치이므로 cd 명령을 수행하기 위해서는 root shell로 변경한다.

ㆍcd는 shell 내부 명령어서 sudo와 함께 사용이 불가하므로 ‘sudo -i’로 root shell 변경 후 명령 처리.

|  |
| --- |
| $ **sudo -i** 🡪 root shell  $ **cd /var/lib/mysql** |

- Client 개인 키 생성

ㆍ 클라이언트 인증서 발급에 필요한 클라이언트 개인 키를 rsa 방식으로 생성.

|  |
| --- |
| **openssl genrsa 2048 > client-key.pem** |

ㆍ genrsa : RSA 비대칭 암호화 방식의 키 생성

ㆍ 2048 : 2048 bit 크기의 값으로 설정.

ㆍ > client-key.pem : 생성 결과를 저장할 파일.

- Client가 CA에 인증서 생성 요청하기 위해 필요한 CSR 파일 생성

ㆍ 일반적인 조직(비즈니스) 내의 정보, 비대칭 키 암/복호화에 사용될 ‘공개 키’ 값 등이 기록된다.

ㆍ SSL/TLS 인증서를 발급받기 위해 CA(인증기관)에 제출하는 파일.

|  |
| --- |
| **openssl req -new -key client-key.pem -out client-cert.csr** |

ㆍreq : 새 인증서 서명 요청(CSR)

ㆍ-new : 새로운 CSR 만들기.

ㆍ-key client-key.pem : 클라이언트 개인 키를 사용해 생성.

ㆍ-out client-cert.csr : 생성 결과 파일명.

ㆍ명령을 수행하면 조직 관련 정보를 입력하는 Command가 나온다. 입력을 희망하지 않으면 온점( ‘.’ ) 입력.

텍스트, 스크린샷, 폰트, 흑백이(가) 표시된 사진

자동 생성된 설명

Figure 12. CSR 파일 생성 시 나타나는 Command 확인

- Client 인증서 생성

ㆍ클라이언트의 CSR 파일로 인증서를 요청하고, CA(인증기관)의 인증서와 개인 키를 통해 서명한 클라이언트의

인증서를 생성한다.

|  |
| --- |
| **openssl x509 -req -in client-cert.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out client-cert.pem -days 3650** |

ㆍx509 -req : X.509 인증서 표준으로 새 인증서를 생성.

ㆍ-in client-cert.csr : CSR 파일을 통해 CA는 인증서를 발급한다 (CSR에는 Client의 공개 키 등이 포함).

ㆍ-CA ca.pem : 인증기관의 인증서를 사용.

ㆍ-Cakey ca-key.pem : 인증기관의 개인 키 파일 사용.

ㆍ-CAcreateserial : 발급한 인증서 관리를 위해 서명된 인증서의 일련번호 부여.

ㆍ-out client-cert.pem : 생성된 클라이언트 인증서의 파일명.

ㆍ-days 3650 : 생성된 인증서의 유효기간 10년.



Figure 13. 인증서 생성 완료 시 출력 내용

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure 14. ls 명령으로 확인한 클라이언트 인증서 파일 확인

**4) Mysql Server와 SSL 연결에 필요한 파일 복사**

- DBeaver Tool에서 SSL 연결을 수행하며 필요한 파일은 다음과 같다.

ㆍ CA Certificate : 인증기관(CA)의 인증서

ㆍ Client Certificate : 클라이언트 인증서

ㆍ Client Private Key : 클라이언트 개인 키

- Client는 외부 Windows이므로 Server 내부에 있는 클라이언트 인증서 파일을 복사해서 옮겨야 한다.

ㆍ 외장 메모리 등을 이용해 파일을 옮길 수 있도록 한다.

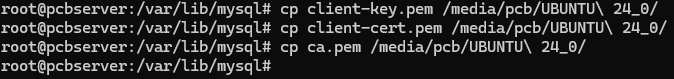


Figure 15. 클라이언트의 MySQL SSL 연결을 위한 필수 파일 복사

- Windows(Client 위치)에서 적절한 경로에 인증서 파일을 보관 후 DBeaver Tool에서 인증서 경로를 추가한다.

- 필수 연결 정보(IP, Port, MySQL Server 로그인 값)가 모두 입력 되었다는 전제로 수행된다.

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

Figure 16. 복사한 인증서 경로를 Parameters 부분에 추가

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Figure 17. 정상 연결 시 Test Connection 결과

- Advanced Option 설명

ㆍRequire SSL : SSL 연결을 강제(필수)로 수행한다. 암호화 통신을 보장한다.

ㆍVerify server certificate : 서버의 SSL 인증서를 검증한다. 서버가 인증기관(CA)으로부터 신뢰할 수 있는 인증서를

발급 받았는지 확인하며 아직 유효한지 검사한다.

ㆍAllow public key retrieval : 클라이언트가 서버로부터 공개 키를 추출해 암호화 통신을 하는 것을 허용한다.

**번외) Windows(Client) 위치에서 openssl Package Install**

- Client에 OpenSSL 설치.

ㆍ Client 위치에서 CA 인증서(ca-cert)와 개인 인증서(client-cert) 및 키(client-key)를 생성할 수 있도록

OpenSSL 설치를 수행한다.

ㆍ Windows용 패키지는 “Shining Light Productions”에서 설치 파일을 다운받아 사용한다.

URL : https://slproweb.com/products/Win32OpenSSL.html

ㆍ 필수 기능만 사용할 목적이라면 Light Version으로 설치한다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

Figure 18. OpenSSL 패키지 exe 파일로 설치



ㆍ설치 마법사에서 OpenSSL DLL 파일을 복사하는 위치 설정은 사용할 환경에 따라 선택한다.

1. “모든 프로그램”에서 OpenSSL을 사용하기 위해 “Windows System Directory”에 DLL 파일을 복사한다.

* C:\Windows\System32 경로에 DLL 파일이 복사된다.
* Windows System 경로에 있으면 모든 프로그램이 접근할 수 있으므로 편리하게 OpenSSL을 적용 가능.
* 여러 Version의 OpenSSL이 존재할 경우에는 버전 충돌에 의한 문제가 발생할 수 있다.

1. The OpenSSL binaries (/bin) directory.

* DLL 파일을 OpenSSL Directory 내의 bin Directory에 복사한다.
* OpenSSL을 사용하려면 수동으로 경로를 직접 지정해야 한다.
* 특정한 프로그램에서만 OpenSSL을 사용하거나 Version 충돌 위험이 있다면 적용한다.
* 본 실습에서는 이 옵션을 선택.

텍스트, 스크린샷, 폰트, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

Figure 19. OpenSSL DDL 파일 복사 옵션 선택

ㆍ Donation(기부) 선택 마법사에서는 아무런 행동을 하지 않으려면 선택을 해제 후 “finish”로 진행한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

ㆍ 설치가 완료되었다면 환경 변수 등록 후 cmd에서 openssl 명령을 사용할 수 있도록 한다.

설치 마법사에서 “The OpenSSL binaries (/bin) directory”를 선택했다면 “C:\Program Files\OpenSSL-Win64\bin” 경로에 OpenSSL DLL 구성 파일이 존재한다. 해당 경로를 시스템 환경변수 ‘Path’에 추가한다.

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Figure 20. windows 환경 변수에 OpenSSL bin path 추가

ㆍcmd에서 “openssl version” 명령을 통해 정상적으로 수행되는지 확인한다.

