User Guide - How to Connect to MS SQL Database Server from Data Enclave Linux VDI using Python

Overview

In this user facing document, we have a tested solution for connecting to SQL server from Python in the Linux VDI.

This solution is to connect from a Linux machine to a windows SQL Server with pyodbc.

Requirements

- Need to log on with a windows domain account
- Need to use python3
- You will need a Terminal (Command Line) and be familiar with basic linux commands

The solution is provided as a minimum set of the instructions for the user to make the connection.

Step 1. Install python library – pyodbc and other modules

You will need to install the python library – pyodbc

A. Open a Terminal window on your Linux Virtual Desktop

B. Install pyodbc python package

When installing pyodbc on Linux, pip will download and compile the pyodbc source code. Some related components and source files must be available for the compile to succeed.

On Ubuntu systems, all you need to do is run

```
pip3 install --user pyodbc
```

C. Check FreeTDS configuration

You can use **pyodbc via FreeTDS**. We already installed and configured the Free TDS in odbcinst.ini. You can see the attached odbcinst.ini in the /etc folder.

Check the configuration of FreeTDS

```
cat /etc/odbcinst.ini
```

It will look like this:

```
[ODBC Driver 17 for SQL Server]
Description=Microsoft ODBC Driver 17 for SQL Server
Driver=/opt/microsoft/msodbcsql17/lib64/libmsodbcsql-17.8.so.1.1
UsageCount=1

[FreeTDS]
Description=FreeTDS Driver
Driver = /usr/lib/x86_64-linux-gnu/odbc/libtdsodbc.so
```

D. Install panda modules

```
pip install pandas
```

Step 2. Configuring the User DSN in ~/.odbc.ini

In the odbc.ini file, you can configure your DSN name, SQL Server and port with Free TDS Driver information.

You can copy the attached example odbc.ini to your home folder as ~/.odbc.ini and modify it as needed.

```
cp odbc.ini ~/.odbc.ini
```

Make sure to include the following lines to configure ***~/.odbc.ini***

```
[i2b2]
#
# Use TDS driver
Driver = FreeTDS

# Server = 10.162.34.52
Server=mssql01-t.analyticsenclave.org
Port = 1433
Database = i2b2
TDS_Version = 8.0
use ntlmv2 = yes
```

Both odbc.ini and odbcinst.ini are to be copy to /etc folder in the Linux VDI.

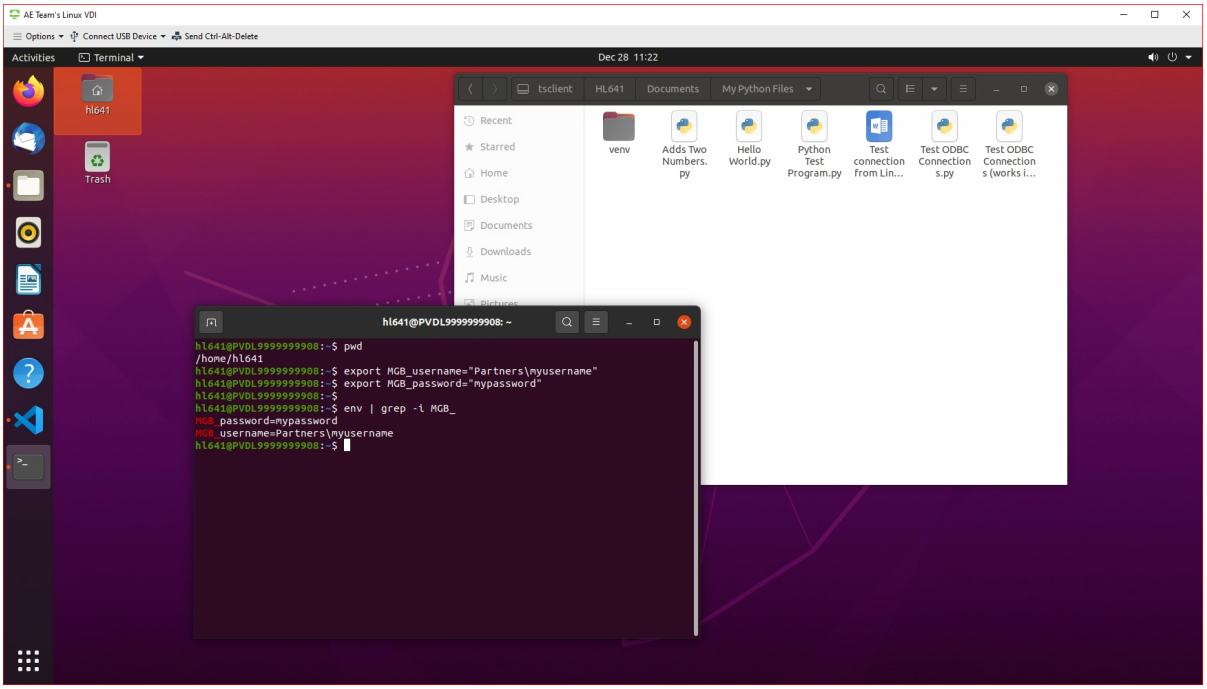
Step 3. Programming Examples using Pyodbc

The following examples uses the environment variables MGB_username and MGB_password. You can create your own environment variables as the steps below.

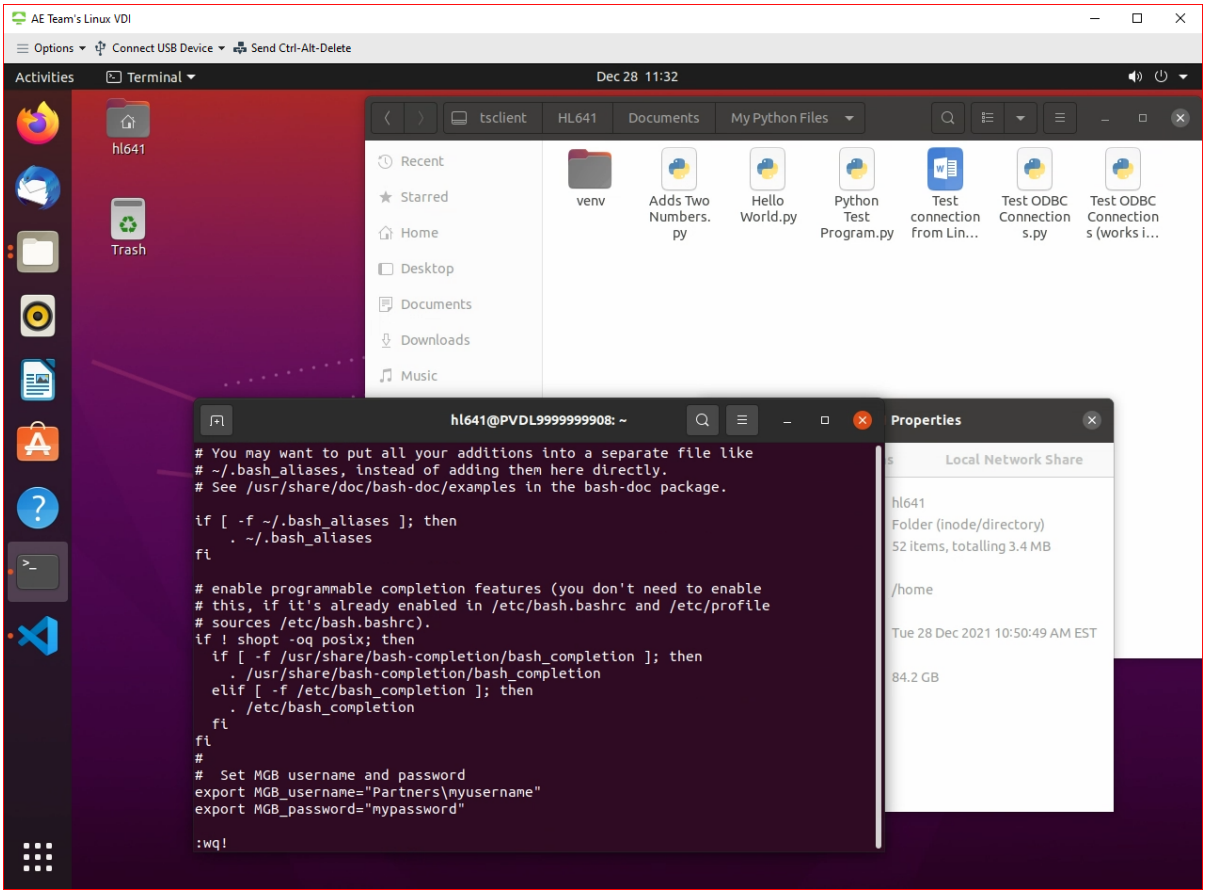
```
export MGB_USERNAME="PARTNERS\\myusername"
export MGB_PASSWORD="mypassword"
```

You can varify your environment variables, MGB_USERNAME and MGB_PASSWORD, by the following step. You should see your MGB username and password as you entered.

```
env | grep -i MGB_
```

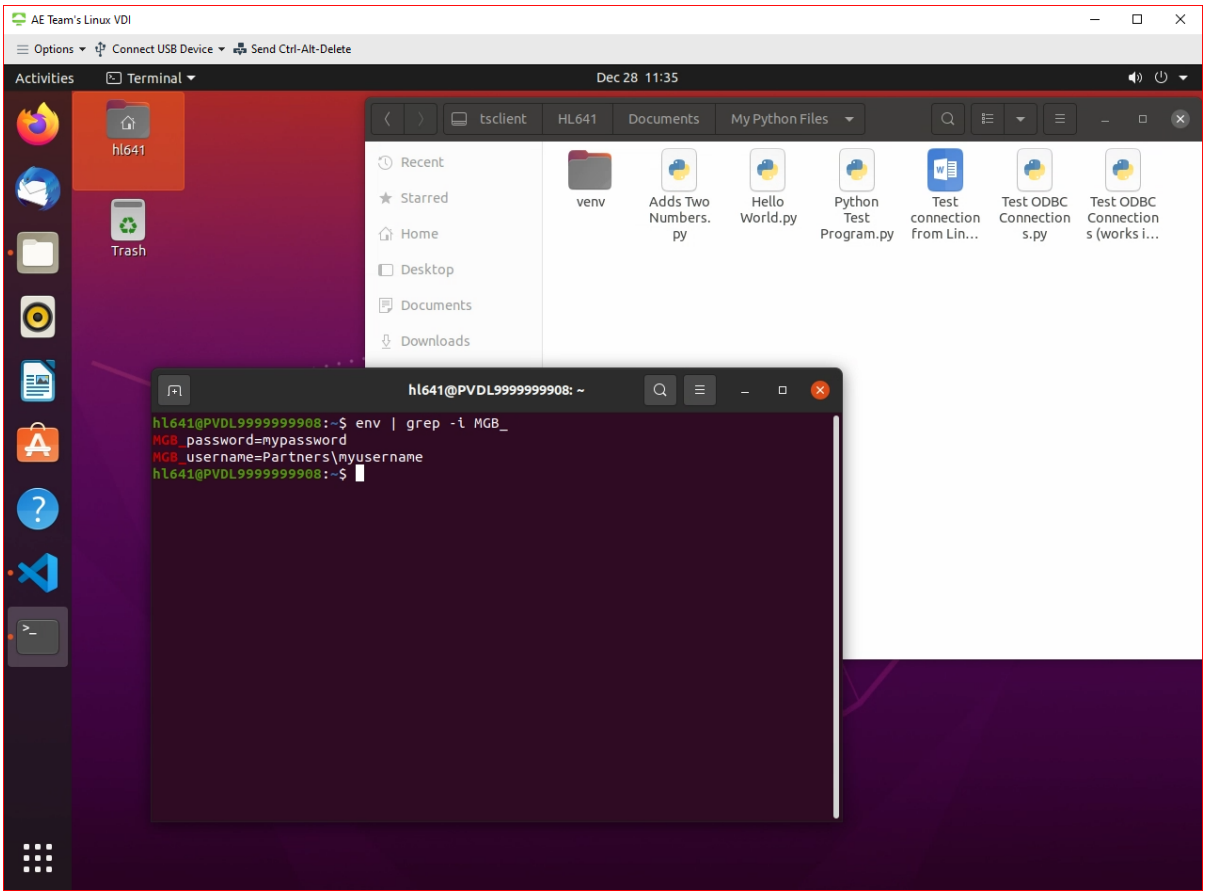


Alternatively, you can enter the two export commands to set up the environment variables MGB_USERNAME and MGB_PASSWORD at the end of your ***~/.bashrc*** file.



And you can veriy them using the same step above after you open the bash terminal windows

```
source ~/.bashrc
env | grep -i MGB_
```



Example #1. Test ODBC Connections to i2b2 Database.py

Open **VS Code** to paste and run the python code snippet below

```
import pandas as pd
import os
import pyodbc

dsn = 'i2b2'
username = os.getenv('MGB_USERNAME')
password = os.getenv('MGB_PASSWORD')

CONN = pyodbc.connect(DSN=dsn,UID=username,PWD=password,)

#Sample select query
sql = "SELECT * from i2b2.dbo.ACT_COVID"
df = pd.read_sql(sql, CONN)
print(df)
```