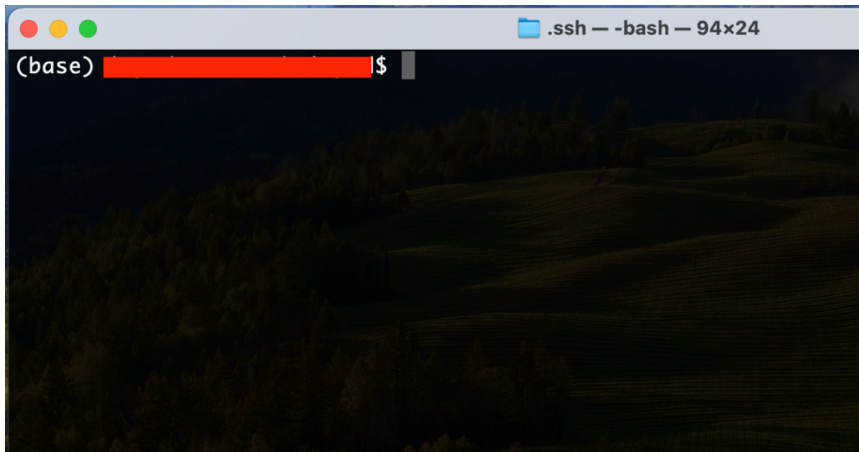
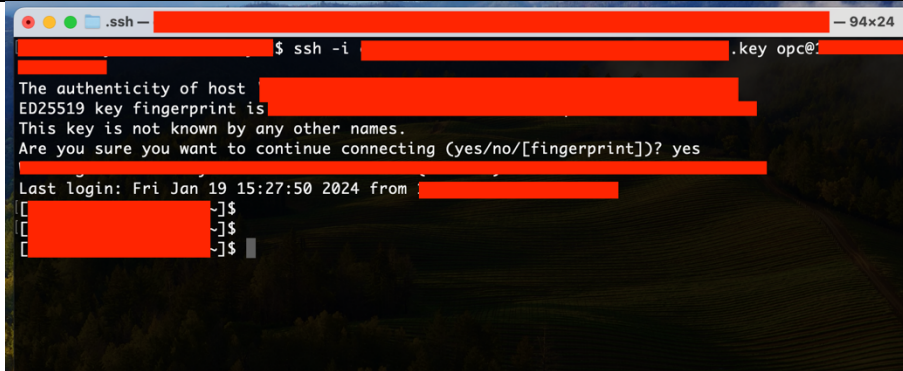


Connect to Oracle Database via OCI Data Science

Option 1: Disable Encryption and use Thin Client Connection.

Description: The purpose of this guide is to connect to your Oracle Database Cloud Service 23ai instance via OCI Data Science with NNE Encryption disabled via the Thin Client.

Prerequisites: OCI Data Science Notebook Session created with a pre-installed Conda Environment. Ensure the Notebook Session is in a Network which is able to talk to the Oracle Database VCN. Connection details to your Oracle Database (IP Address, Port, Pluggable DB Name, Host Domain Name, DB User, DB Password, DB Private Key File).

Step	Description
<p>We need to disable NNE encryption to be able to connect via Python oracledb thin client.</p> <p>Open up a Terminal.</p> <p>Navigate to the folder which has our Database SSH Keys</p>	
<p>SSH into your Database Node:</p> <p>ssh -i <location-to-private-key> opc@<node-ip-address></p>	 <p>If prompted on whether you want to continue connecting enter yes</p>



<p>Switch to root: <i>sudo su</i></p> <p>Switch to Oracle user: <i>su - oracle</i></p>	<pre> Last login: Fri Jan 19 15:27:50 2024 from [REDACTED] [REDACTED]\$ [REDACTED]\$ [REDACTED]\$ pwd /home/opc [REDACTED]\$ sudo su [REDACTED]# su - oracle Last login: Wed May 22 16:30:05 UTC 2024 [oracle@REDACTED]\$ [oracle@REDACTED]\$ </pre>
<p>Navigate to our network admin directory:</p> <p><i>cd \$ORACLE_HOME/network/admin</i></p>	<pre> [REDACTED]\$ [REDACTED]\$ [REDACTED]\$ cd \$ORACLE_HOME/network/admin [REDACTED]\$ [REDACTED]\$ pwd /u01/app/oracle/product/23.0.0/dbhome_1/network/admin [REDACTED]\$ </pre>
<p>Edit the <i>sqlnet.ora</i> file.</p> <p><i>vi sqlnet.ora</i></p>	<pre> [REDACTED]\$ [REDACTED]\$ [REDACTED]\$ vi sqlnet.ora </pre>
<p><i>Comment out all lines with a #</i></p> <p><i>Save and close the file.</i></p>	<pre> # ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/opt/oracle/dcs/commonstore/wallets/tde/\$ORACLE_UNQNAME))) #ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/opt/oracle/dcs/commonstore/wallets/\$ORACLE_UNQNAME/tde))) #SQLNET.ENCRYPTION_SERVER=REQUIRED #SQLNET.CRYPTO_CHECKSUM_SERVER=REQUIRED #SQLNET.ENCRYPTION_TYPES_SERVER=(AES256,AES192,AES128) #SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER=(SHA256,SHA384,SHA512,SHA1) #SQLNET.ENCRYPTION_CLIENT=REQUIRED #SQLNET.CRYPTO_CHECKSUM_CLIENT=REQUIRED #SQLNET.ENCRYPTION_TYPES_CLIENT=(AES256,AES192,AES128) #SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT=(SHA256,SHA384,SHA512,SHA1) #SQLNET.EXPIRE_TIME=10 </pre>

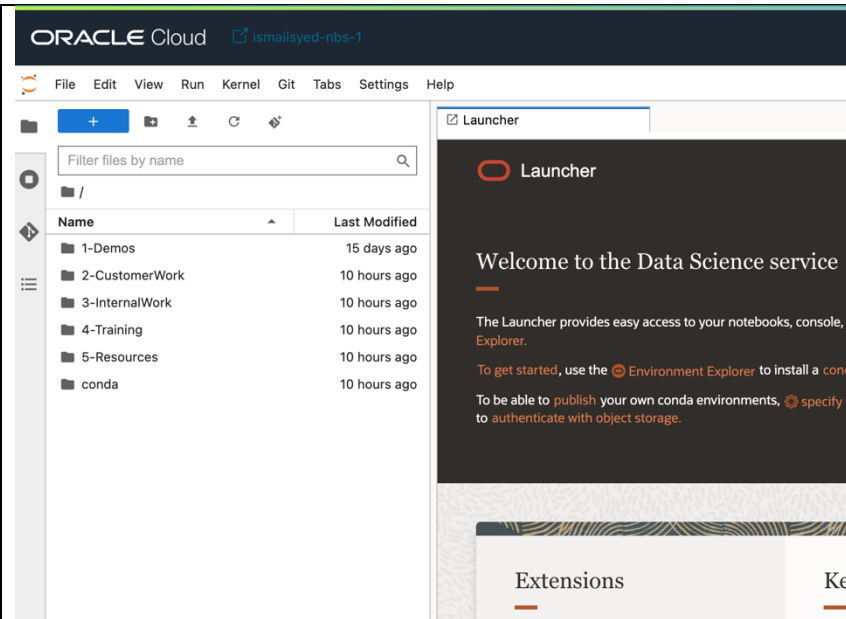


<p>Let's restart the database.</p> <p>Enter: sqlplus</p> <p>Username: sys as sysdba</p> <p>Password: <db-admin-password-created-earlier></p>	<pre> [redacted]\$ [redacted]\$ [redacted]\$ sqlplus SQL*Plus: Release 23.0.0.0.0 - Production on Wed May 22 16:30:48 2024 Version 23.4.0.24.05 Copyright (c) 1982, 2024, Oracle. All rights reserved. Enter user-name: sys as sysdba Enter password: Connected to: Oracle Database 23ai EE High Perf Release 23.0.0.0.0 - Production Version 23.4.0.24.05 SQL> SQL> SQL> SQL> </pre>
<p>We will shutdown the Container DB:</p> <p>shutdown;</p> <p>We will then startup the Container DB:</p> <p>startup;</p>	<pre> SQL> SQL> shutdown; Database closed. Database dismounted. ORACLE instance shut down. SQL> SQL> SQL> startup; ORACLE instance started. Total System Global Area 3.2694E+10 bytes Fixed Size 7697192 bytes Variable Size 3959422976 bytes Database Buffers 2.8588E+10 bytes Redo Buffers 138514432 bytes Database mounted. Database opened. SQL> SQL> </pre>
<p>We can exit from SQLPlus, the Oracle User, the Root User and the SSH Connection by running:</p> <p>exit;</p>	<pre> SQL> SQL> SQL> exit Disconnected from Oracle Database 23ai EE High Perf Release 23.0.0.0.0 - Production Version 23.4.0.24.05 [redacted]\$ exit logout [redacted]# exit exit [redacted]\$ exit logout Connection to [redacted] closed. [redacted]\$ [redacted]\$ [redacted]\$ </pre>

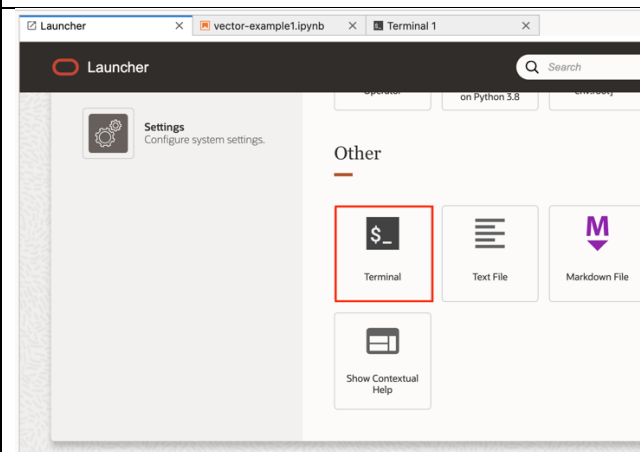


Now login to your OCI Data Science Notebook Session.

Ensure your Notebook Session allows you to 'see' the Oracle DB from a networking perspective.



Scroll down and **open up a Terminal**.



Activate the Conda environment you are planning on using and upgrade the oracledb library.

conda activate
<location-of-
conda-env>

pip install
oracledb --
upgrade

```
(base) bash-4.2$ pwd
/home/datascience
(base) bash-4.2$ conda activate conda/generalml_p38_cpu_v1/
/home/datascience/conda/generalml_p38_cpu_v1 bash-4.2$
/home/datascience/conda/generalml_p38_cpu_v1 bash-4.2$
/home/datascience/conda/generalml_p38_cpu_v1 bash-4.2$
/home/datascience/conda/generalml_p38_cpu_v1 bash-4.2$ pip install oracledb --upgrade
Requirement already satisfied: oracledb in ./conda/generalml_p38_cpu_v1/lib/python3.8/site-packages (1.1.1)
Collecting oracledb
  Downloading oracledb-2.2.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
    11.5/11.5 MB 115.0 MB/s eta 0:00:00
Requirement already satisfied: cryptography>=3.2.1 in ./conda/generalml_p38_cpu_v1/lib/python3.8/site-packages (from oracledb) (37.0.2)
Requirement already satisfied: cffi>=1.12 in ./conda/generalml_p38_cpu_v1/lib/python3.8/site-packages (from cryptography>=3.2.1->oracledb) (1.14.6)
Requirement already satisfied: pycparser in ./conda/generalml_p38_cpu_v1/lib/python3.8/site-packages (from cffi>=1.12->cryptography>=3.2.1->oracledb) (2.21)
Installing collected packages: oracledb
  Attempting uninstall: oracledb
    Found existing installation: oracledb 1.1.1
    Uninstalling oracledb-1.1.1:
      Successfully uninstalled oracledb-1.1.1
Successfully installed oracledb-2.2.0
/home/datascience/conda/generalml_p38_cpu_v1 bash-4.2$
/home/datascience/conda/generalml_p38_cpu_v1 bash-4.2$
/home/datascience/conda/generalml_p38_cpu_v1 bash-4.2$
```



Open a Notebook Session and run the following script:

```
# Imports
import oracledb

# Establish Connect to Oracle DB
connection = oracledb.connect(
    user=<user_name>,
    password=<password>,
    dsn=<host:port/pluggable_db_name.host_domain_name>)

# Establish Cursor
cursor = connection.cursor()

# Drop Table if exists
cursor.execute("""
    begin
        execute immediate 'drop table demo_setup';
        exception when others then if sqlcode <> -942 then raise; end if;
    end;""")

# Create new table with ID column and Vector Type Column
cursor.execute("""
    create table demo_setup (
        id number,
        v vector(3, float32),
        primary key (id))""")

# Bind variable values
id_val = 1
vector_val = [5.3, 2.4, 3.1412]

# Insert Record into Table
cursor.setinputsizes(None, oracledb.DB_TYPE_VECTOR)
cursor.execute("insert into demo_setup values (:1, :2)", [id_val,
vector_val])

# Commit Insert
connection.commit()

# Query table and print result of query
cursor.execute('select * from demo_setup')

for row in cursor:
    print(row)

# Close Connection
connection.close()
```

