Practice 20: Preparing Oracle Database 19c Practice Environment

Practice Overview

In this practice, you will create two Oracle VirtualBox appliances or virtual machines (VMs). Each of them will have Oracle Database 19c software installed in it. You will use those VMs in the incoming practice lectures to exercise Oracle database 18c and 19c new multitenant features.

In high level, you will perform the following:

- Download an Oracle VirtualBox appliance import file from my website.
- Import the downloaded file into Oracle VirtualBox as a machine named `srv1`
- Import the downloaded file again into Oracle VirtualBox as a different machine named `srv2`

Note

The VM that you used in the course will not be used in the incoming lectures because it has Oracle database version 12c.

Note

Each vm will take 6 GB from the hosting PC memory. Therefore, make sure that the hosting PC has at least 12 GB free memory.

A. Download the Oracle VirtualBox appliance import file

In this section of the practice, you will download an Oracle VirtualBox appliance import file (.ova) from my website. It will be used as a seed to create Oracle VirtualBox appliances with Oracle database 19c software installed in it.

1. Download the Oracle VirtualBox import file from the following link.

   https://www.ahmedbaraka.com/a019-oracle-database-19c-on-linux-7/

   It is an Oracle VirtualBox appliance sized nearly 9.7 GB with Oracle Database 19c software (19.8) installed in it and Oracle Linux 7.8. It has an Oracle 19c database installed in it. For further information, including the passwords, refer to the readme file in the download page.

   After the download is finished, you should have a file named "**Oracle 19c DB.ova**"

Note: Take a note of the readme file from the link above. It has the password of the `root`.

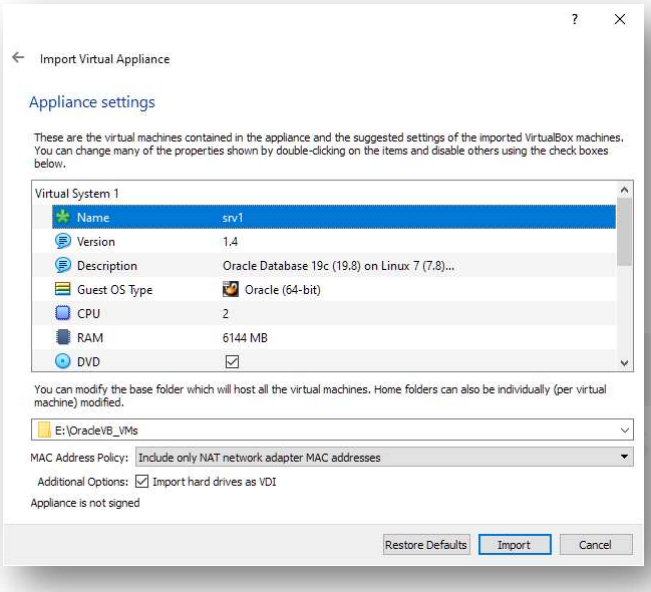
B. Create Oracle VirtualBox Appliance srv1

In this section of the practice, you will create a new Oracle VirtualBox appliance and name it as `srv1`. This is different from the VM that you has used so far in the course.

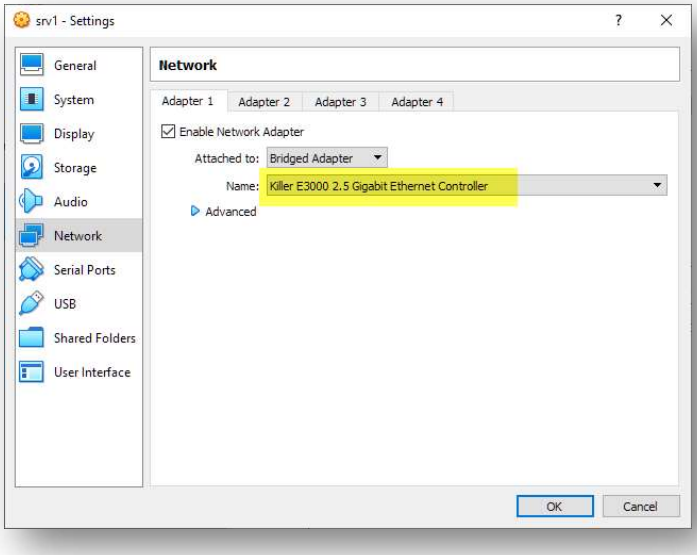
2. In Oracle VirtualBox, import the downloaded file.

Select **File** > **Import Appliance** > select the downloaded file > in the **Appliance Settings** window, set the **name** to **srv1** > make sure the Base Folder is set to the directory where the VMs are saved in your PC > click on **Import** button

Note: When you import an `ova` file, Oracle VirtualBox automatically creates a sub-directory under the **Base Folder** and name it with the same name as the Name given to the vm.

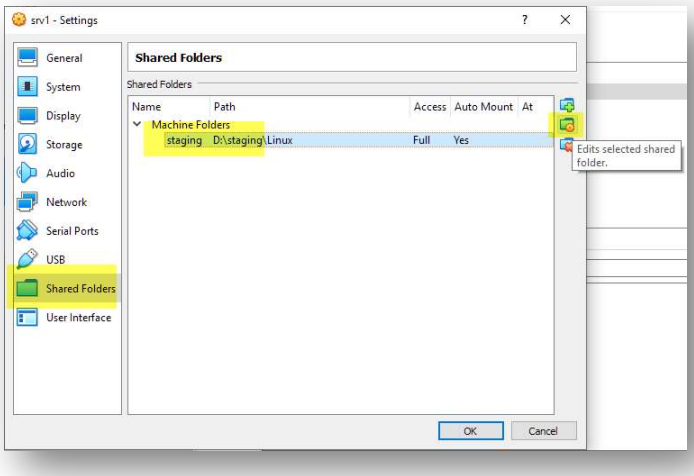


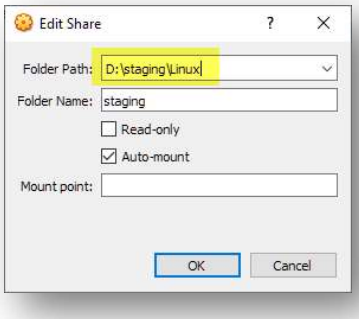
3. After the appliance is fully imported, open the VM settings window. On the left panel, click on the **Network** option > select the **Name** of the network adapter that matches the name of the network adapter in your PC



4. Click on the **Shared Folders** option, select the **staging** option, click on **Edit** button, change the **Folder Path** value to reflect the available folder in your environment > click on **OK** button

   Shared folder is used to easily exchange files between the vms and the hosting PC.





5. Start `srv1` and login to it as `root`. Its password must be provided in the vm `readme` file.

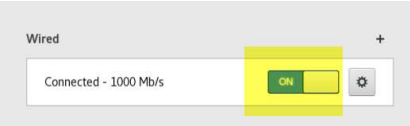
6. Open the Network settings

   Applications > **System Tools** > **Settings** > **Network** > in the **Wired** panel click on the **Gear** button > click on **IPv4**

7. Make sure the IP address of `srv1` works with your network subnetting. If not, either change its IP address or change the Network setting to "**Automatic DHCP**" so that the vm takes its new IP address from the router in your network. Click on **Apply** button.

8. If you made changes on the vm IP address, perform the following:

8.1. Disable the Wired network and enable it again by turning its switch off and then on again. This action makes the changes take effect.



8.2. Open a Terminal window, and issue the following command to obtain the new IP address assigned to `srv1`

```
ifconfig
```

8.3. open the `/etc/hosts` file with the `vi` editor and fix `srv1` IP address in the file to reflect the new IP address.

```
vi /etc/hosts
```

9. Open Putty and configure a connection to `srv1`

10. Open a Putty connection session to `srv1`

11. In the Putty session, change the current user to `oracle`, open SQL*Plus, and connect to the local database as `sysdba`. Display the existing PDBs to verify that `PDB1` is running in the database.

```
su - oracle
sqlplus / as sysdba

show pdbs
```

12. Issue the following query to verify that standard HR schema is installed in `pdb1`.

```
col CON_ID format 999
col USERNAME format A15

SELECT CON_ID, USERNAME FROM CDB_USERS WHERE ORACLE_MAINTAINED='N';
```

13. Exit from SQ*Plus and shutdown the vm.

C. Create Oracle VirtualBox Appliance srv2

In this section of the practice, you will create another Oracle VirtualBox appliance and name it as `srv2`.

14. In Oracle VirtualBox, import the downloaded file and name it as `srv2`

Select **File** > **Import Appliance** > select the downloaded file > in the **Appliance Settings** window, set the **name** to **srv2** > make sure the Base Folder is set to the directory where the VMs are saved in your PC > click on **Import** button

15. After the appliance is fully imported, open the VM settings window. On the left panel, click on the **Network** option > select the **Name** of the network adapter that matches the name of the network adapter in your PC

16. Click on the **Shared Folders** option, select the **staging** option, click on **Edit** button, change the **Folder Path** value to reflect the available folder in your environment > click on **OK** button

17. Start `srv2` and login to it as `root`

18. Open a Terminal window and issue the following command to rename the hostname of the vm to `srv2`

```
# display the current hostname:
hostnamectl status

# change the hostname
hostnamectl set-hostname srv2.localdomain

# verify:
hostnamectl status
```

19. Open the Network settings

Applications > **System Tools** > **Settings** > **Network** > in the **Wired** panel click on the **Gear** button > click on **IPv4**

20. Change the IP address of `srv2` to an address different from the `srv1` IP address. Alternatively, change the Network setting to "**Automatic DHCP**" so that the vm takes its new IP address from the router in your network. Click on **Apply** button.

21. Disable the Wired network and enable it again by turning its switch off and then on again. This action makes the changes take effect.

22. In the Terminal window, issue the following command to obtain the new IP address assigned to `srv2`

```
ifconfig
```

23. Open the `/etc/hosts` file with the `vi` editor and fix `srv2` IP address and hostname in the file to reflect the new IP address.

```
vi /etc/hosts
```

24. Open Putty and configure a connection to `srv2`

25. Open a Putty connection session to `srv2`

Note: I recommend changing the font color of `srv2` to light green color to easily distinguish `srv1` Putty sessions from `srv2` sessions.

26. In the Putty session, change the current user to `oracle`

```
su - oracle
```

27. Open the `tnsnames.ora` file in `vi` editor and fix the hostname in the descriptor `LISTENER_ORADB`

```
vi $TNS_ADMIN/tnsnames.ora
```

28. Open the `listener.ora` file in `vi` editor and fix the hostname in it. Reload the file afterwards.

```
vi $TNS_ADMIN/listener.ora

lsnrctl reload
```


29. Issue the following command to drop the database in `srv2`

```
cd ${ORACLE_HOME}/bin/
dbca -silent -deleteDatabase -sourceDB ${ORACLE_SID} -sysDBAUserName sys -
sysDBAPassword oracle
```

30. In `oracle` profile file, change `ORACLE_SID` to `oradb2`. Source the file afterwards.

```
vi ~/.bash_profile

source ~/.bash_profile
```

31. Issue the following code to create a new database in `srv2` named as `oradb2`

The created CDB database contains one PDB named as `pdb21`.

```
mkdir /u01/app/oracle/oradata/ORADB2

./dbca -silent -createDatabase \
   -templateName General_Purpose.dbc \
   -gdbName ${ORACLE_SID} \
   -sid ${ORACLE_SID} \
   -sysPassword oracle \
   -systemPassword oracle \
   -emConfiguration NONE \
   -datafileDestination '/u01/app/oracle/oradata/ORADB2' \
   -recoveryAreaDestination '/u01/app/oracle/oradata/ORADB2' \
   -storageType FS \
   -automaticMemoryManagement FALSE \
   -createAsContainerDatabase true \
   -pdbAdminPassword oracle \
   -numberOfPDBs 1 \
   -pdbName pdb21 \
   -enableArchive TRUE  \
   -ignorePreReqs \
   -initparams sga_target=2048M,DB_RECOVERY_FILE_DEST_SIZE=20G
```


32. Open the `oratab` file in `vi` editor

```
vi /etc/oratab
```

33. Change the last field for the database line to Y

This setting makes the database automatically restarts when the system is rebooted or started.

```
oradb2:/u01/app/oracle/product/19.0.0/db_1:Y
```

34. Open `tnsnames.ora` file and change the `PDB1` setting in it to `pdb21` as follows:

Observe that `ORADB2` setting is automatically added by the `dbca`

```
vi $TNS_ADMIN/tnsnames.ora

PDB21 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = srv2.localdomain)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = pdb21)
    )
  )
```

35. Add to the `tnsnames.ora` file the `oradb` connection descriptor.

```
ORADB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = srv1.localdomain)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = oradb.localdomain)
    )
  )
```

36. Test connecting to `oradb2` and `pdb21`

```
sqlplus /nolog

conn system/oracle@oradb2

conn system/oracle@pdb21
```

37. In `srv2` session, preserve the startup state of `pdb21` through the CDB restart.

```
conn / as sysdba

# make sure pdb21 is open in read/write mode:
SELECT NAME, OPEN_MODE FROM V$PDBS WHERE NAME=UPPER('PDB21');

ALTER PLUGGABLE DATABASE pdb21 SAVE STATE;

# verify:
col CON_NAME for a20
col INSTANCE_NAME for a20
SELECT CON_NAME, INSTANCE_NAME, STATE FROM DBA_PDB_SAVED_STATES;
```

38. Exit from `oracle` session so that the current user becomes `root`

39. Start `srv1`

40. Open a Putty session to `srv1` as `root`

41. In each server, add the setting to the `hosts` file so that each machine can see the other. The following code is the case in my environment.

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
#::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.124 srv1.localdomain srv1
192.168.1.125 srv2.localdomain srv2
```

42. In `srv1` session, open the `tnsnames.ora` file and add the connection descriptor to `oradb2`

```
su - oracle
vi $TNS_ADMIN/tnsnames.ora
```

```
ORADB2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = srv2.localdomain)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = oradb2)
    )
  )
```

43. Make sure each database can connect to the other database.

```
sqlplus system/oracle@oradb
sqlplus system/oracle@oradb2
```

44. In `srv1`, enable the archivelog mode in `oradb` database.

If the archivelog mode is not enabled in a database, the dbca automatically shuts down the source PDB when it is used to clone a PDB.

```
-- verify the archivelog is disabled:
sqlplus / as sysdba
archive log list;

-- mount the database
SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST'
SCOPE=SPFILE;

-- enable the archivelog mode
ALTER DATABASE ARCHIVELOG;

-- restart the database
SHUTDOWN IMMEDIATE
STARTUP OPEN

-- switch the log file
ALTER SYSTEM SWITCH LOGFILE;
-- checkout the generated archive log file
SELECT NAME FROM V$ARCHIVED_LOG;
```

Summary

By the end of this practice, we end up with having two vms named: srv1 and srv2. The first one has an Oracle database 19c CDB named as `oradb` with one PDB named as `pdb1`. The other vm is named as `srv2` with an Oracle database 19c CDB named as `oradb2` and one PDB named as `pdb21`. Those vms will be used for the incoming practice lectures.