

Practice

Creating Oracle Database

Practice Target

In this practice you will create an Oracle database in the Linux platform `srv1` and another Oracle database in Windows platform `winsrv`.

Practice Overview

In high level, in this practice, you will perform the following tasks:

- Create an Oracle non-CDB Database in the Linux Platform
- Create an Oracle non-CDB Database in the Windows Platform



Creating an Oracle non-CDB Database in the Linux Platform

In the following steps, you will use the dbca silent mode to create an Oracle 19c non-CDB database named `oradb`.

Note: In production systems (unless it is specifically required by the system), always support creating a multitenant database (CDB). In this practice, you will create a non-CDB database because so many systems still running with non-CDB databases. Later in the course, you will learn about creating CDBs.

1. If `srv1` is shutdown, start it up.
2. Open a new Putty session to `srv1` as `oracle`
3. Display the values of the basic Oracle database environment variables.

Make sure the values of those variables are correctly set before invoking the `dbca`.

```
echo $ORACLE_BASE  
echo $ORACLE_HOME  
echo $ORACLE_SID
```

4. Verify that no listener is running in the system.

```
lsnrctl status  
ps -ef | grep lsn
```

5. Run the following command to create a new Listener in silent mode:

```
netca -silent -responsefile  
/u01/app/oracle/product/19.0.0/db_1/assistants/netca/netca.rsp
```

6. Login as `oracle` to the GUI interface of `srv1` (Oracle VirtualBox window)

7. Open a terminal window.

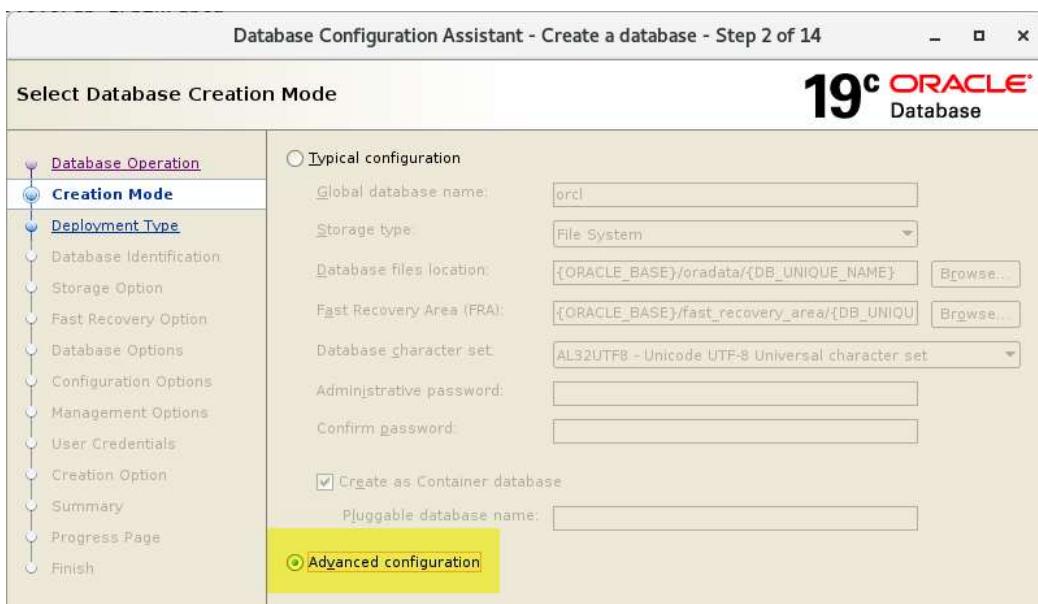
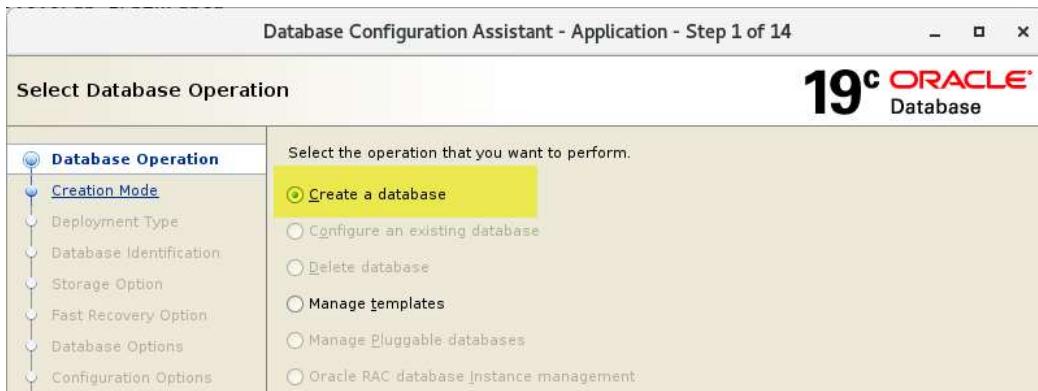
8. Verify that the `dbca` is invoked from the require Oracle home directory.

```
which dbca
```

9. Invoke the `dbca` utility

```
dbca
```

10. Respond to dbca windows as follows:



Database Configuration Assistant - Create a database - Step 3 of 14

Select Database Deployment Type

19c ORACLE Database

Database type: Oracle Single Instance database

Configuration type: Admin Managed

Select a template for your database.

Templates that include datafiles contain pre-created databases. They allow you to create a new database quickly. Use templates without datafiles only when necessary, such as when you need to change attributes like block size that cannot be altered after database creation.

Template name	Include datafiles	Details
Data Warehouse	Yes	View details
Custom Database	No	View details
General Purpose or Transaction Processing	Yes	View details

Template location: /u01/app/oracle/product/19.0.0/db_1/assistants/dbca/templates [Change...](#)

Database Configuration Assistant - Create a database - Step 4 of 14

Specify Database Identification Details

19c ORACLE Database

Global database name: oradb.localdomain

SID: oradb

Service name:

Create as Container database

A Container database can be used for consolidating multiple databases into a single database, and it enables database virtualization. A Container database (CDB) can have zero or more pluggable databases (PDB).

Database Configuration Assistant - Create 'oradb' database - Step 5 of 14

Select Database Storage Option

19c ORACLE Database

Use template file for database storage attributes
Storage type and location for database files will be picked up from the specified template (Custom Database).

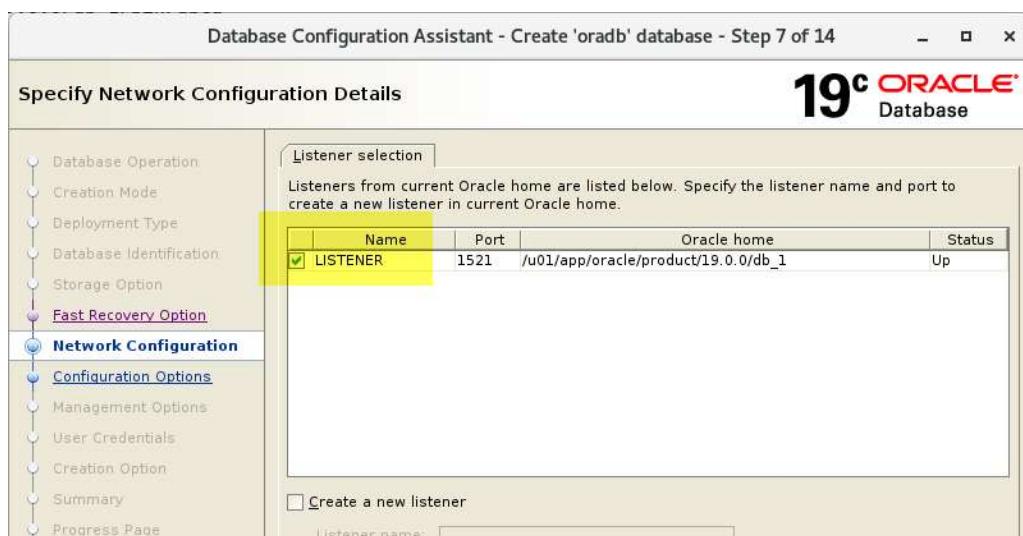
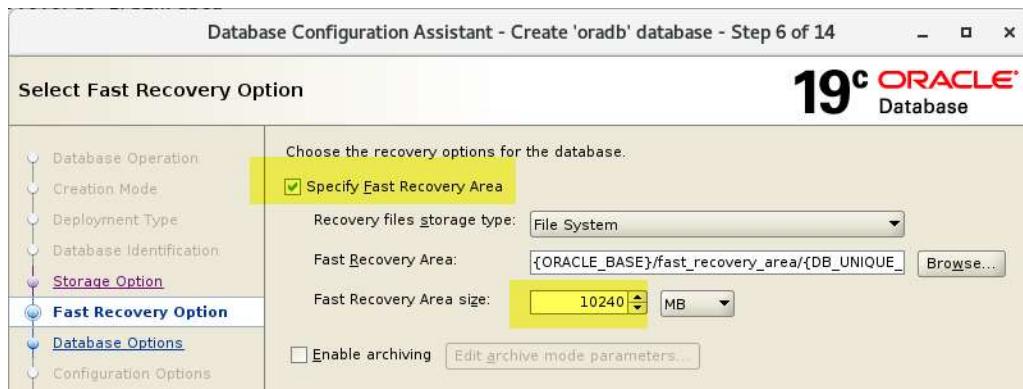
Use following for the database storage attributes
All the database files will be put at the specified location below. You can customize the name and location of each datafile in the subsequent screen.

Database files storage type: File System

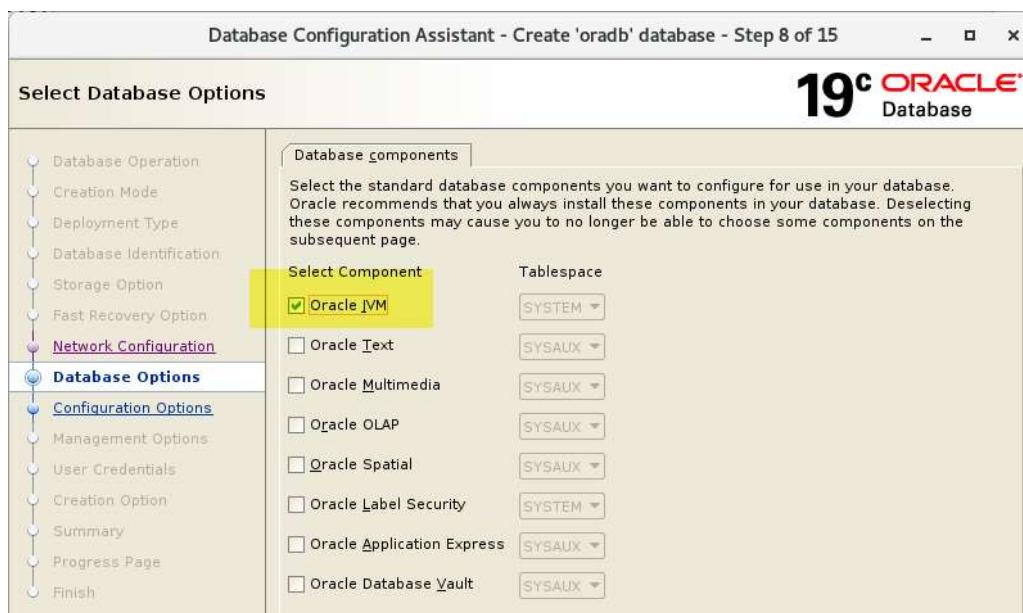
Database files location: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME} [Browse...](#)

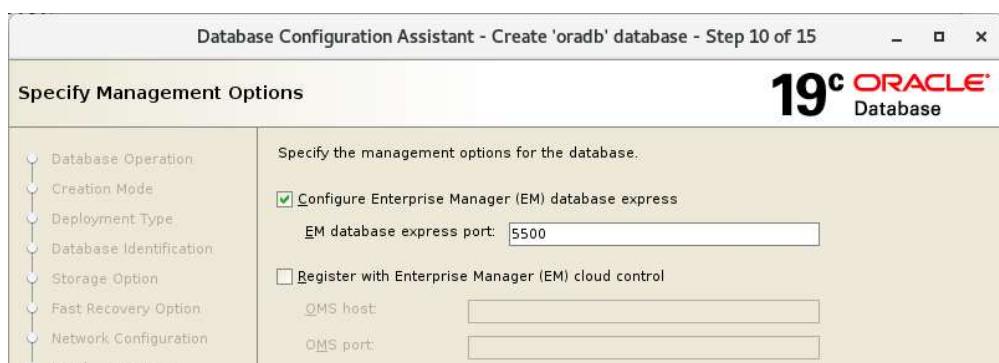
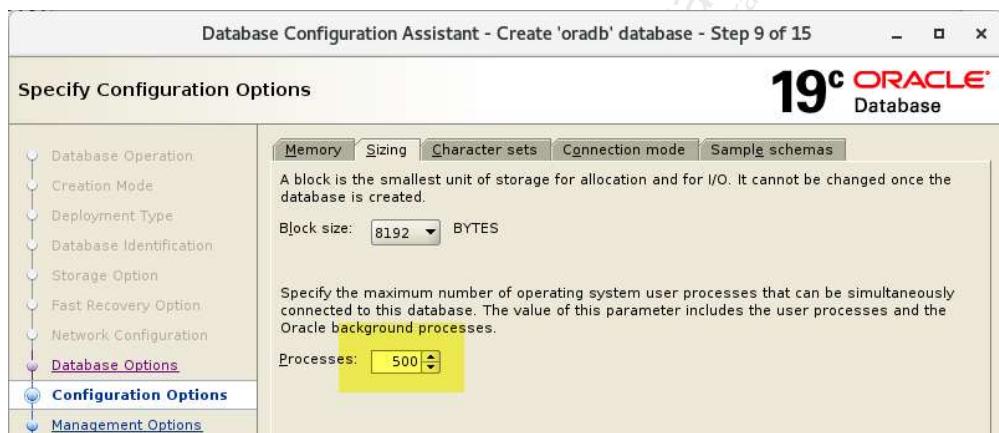
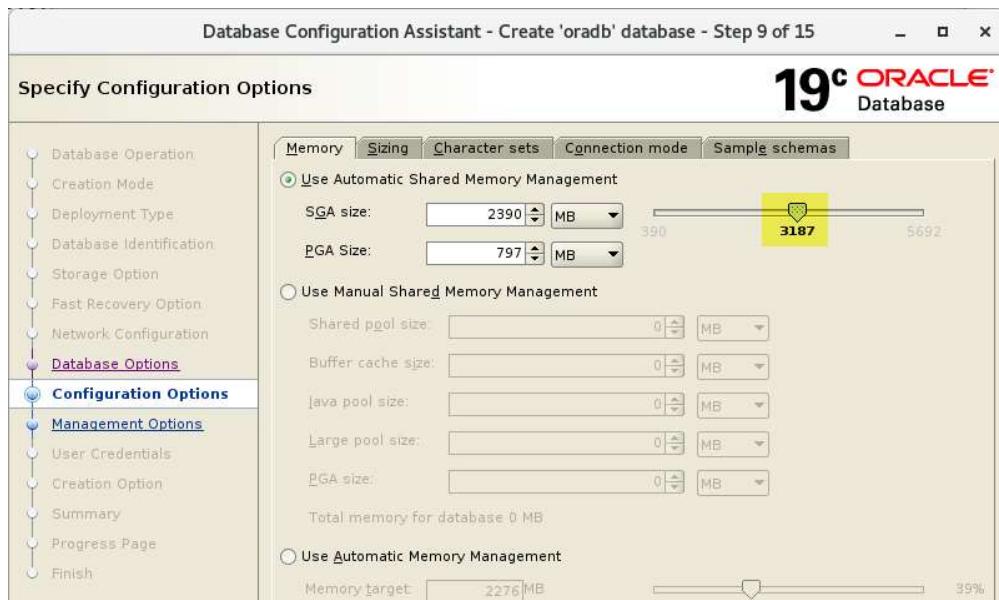
Oracle Managed files option will enable Oracle to automatically generate the names of the datafiles for simplified database management.

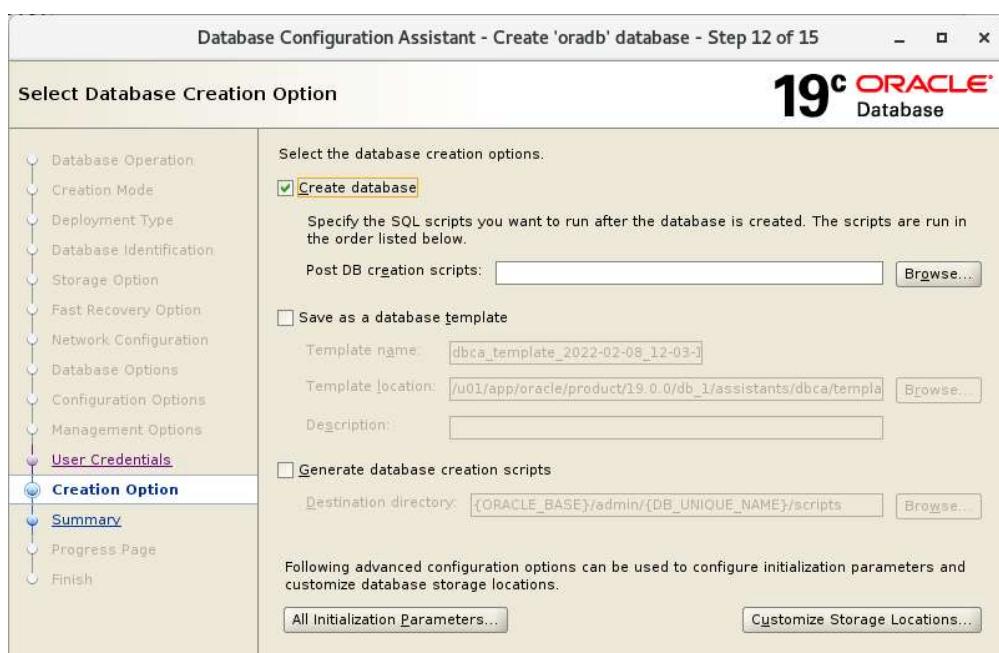
Use Oracle-Managed Files (OMF) [Multiplex redo logs and control files...](#)

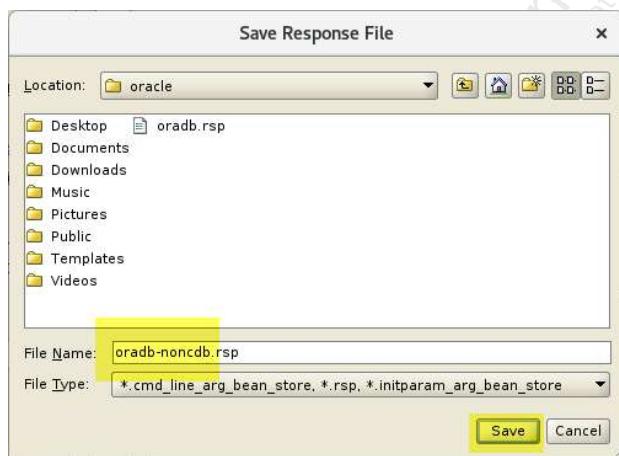
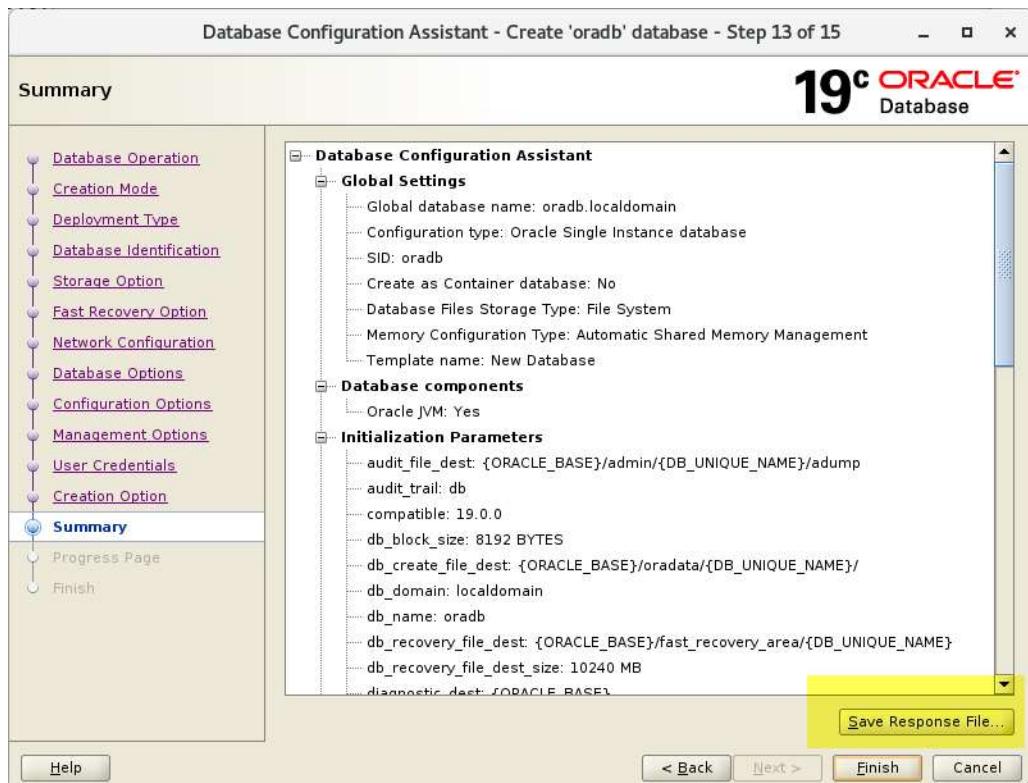


Note: the options selected in this window is not saved in the generated response file.









11. Check if the response file contains the parameter dbOptions

The response file does not contain the `dbOptions` parameter. This means if we create a database using the response file only, all the database options will be created, which we do not want to do. To resolve this issue, we will pass the `dbOptions` to the `dbca` command line.

Note: The response file does not contain the password that we entered in the `dbca` windows either. When we use this response file to create a database, it will prompt for the password.

```
cat /home/oracle/oradb-noncdb.rsp | grep dbOptions
```

12. In the Putty session, invoke `dbca` in silent mode to create the database as defined in the created response file.

The utility will prompt you to enter the passwords for `SYS` and `SYSTEM` users. In the course practices, we use the password **ABcd##1234**. Observe that the assistant does not confirm the entered password.

```
dbca -createDatabase -silent -responseFile /home/oracle/oradb-noncdb.rsp -  
dbOptions  
JSERVER:true,DV:false,APEX:false,OMS:false,SPATIAL:false,IMEDIA:false,ORACLE_TEX  
T:false,CWMLITE:false -sampleSchema true
```

13. After the database creation are finished, verify the database is up and running by verifying that the `pmon` process is up and running or by logging to the database using SQL*Plus as `sys`

```
ps -ef | grep pmon  
sqlplus / as sysdba
```

14. Submit the following query to retrieve the options installed in the database.

Observe that the JVM is installed. APEX is not installed. The components come with nearly every database.

```
set linesize 180  
col COMP_NAME for a40  
col STATUS for a15  
col VERSION for a10  
SELECT COMP_NAME, STATUS, VERSION FROM DBA_REGISTRY ORDER BY 1;
```

15. Check if the `HR` schema is created.

The `HR` schema is not created, although we set the parameter `sampleSchema` to `true` to the `dbca` command line.

```
SELECT USERNAME FROM DBA_USERS WHERE USERNAME='HR';
```

16. Run the following script to create the `HR` schema.

When you are prompted, enter the password **ABcd##1234**, the default tablespace as `users`, the temporary tablespace as `temp`, and the log file as `hr.log`.

In production systems, we do not need the sample schemas.

```
@ $ORACLE_HOME/demo/schema/human_resources/hr_main_new.sql
```

- 17.** After the script is finished, verify that the `HR` schema is created.

```
SELECT USERNAME FROM DBA_USERS WHERE USERNAME='HR';
SELECT COUNT(*) FROM HR.EMPLOYEES;
```

- 18.** Exit from SQL*Plus then check if a connection entry to `oradb` is added to the `tnsnames.ora` file

The `tnsnames.ora` file was created and a connection descriptor to `oradb` is added to it. This was done by the `dbca`. You will learn about the `tnsnames.ora` file later in the course.

```
exit
cat $ORACLE_HOME/network/admin/tnsnames.ora
```

- 19.** In the VirtualBox window of `srv1`, start the Firefox browser and open the EM Express using the following URL. Accept the warning displayed by the browser. Enter the `sys` username, its password, then click on Login button.

EM Express is a simple web-based GUI interface that is used for monitoring and troubleshooting the local database performance.

```
firefox https://srv1:5500/em &
```

Configuring Auto-restart script

When Oracle Restart is not configured, Oracle database does not automatically startup when rebooting the machine. In the following steps, you will create and deploy a script that automatically starts up the database when the machine is rebooted.

- 20.** In the Putty session, change the current user to `root`

- 21.** Edit the `oratab` file

```
vi /etc/oratab
```

- 22.** Change the last field for the database line to `y`

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

23. Create the file /etc/init.d/dbora and add the following code in it:

```
vi /etc/init.d/dbora

#!/bin/sh
# description: Oracle auto start-stop script.
ORA_HOME=/u01/app/oracle/product/19.0.0/db_1
ORA_OWNER=oracle

case "$1" in
'start')
    # Start the Oracle databases:
    # Remove "&" if you don't want startup as a background process.
    su - $ORA_OWNER -c "$ORA_HOME/bin/dbstart $ORA_HOME" &
    touch /var/lock/subsys/dbora
    ;;

'stop')
    # Stop the Oracle databases:
    su - $ORA_OWNER -c "$ORA_HOME/bin/dbshut $ORA_HOME" &
    rm -f /var/lock/subsys/dbora
    ;;
esac
```

24. Change the group of the dbora file to oinstall, and set its permissions to 750

```
chgrp oinstall /etc/init.d/dbora
chmod 750 /etc/init.d/dbora
```

25. Create symbolic links to the dbora script in the appropriate run-level script directories

```
ln -s /etc/init.d/dbora /etc/rc.d/rc0.d/K01dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/S99dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/S99dbora
```

26. Restart srv1 and wait for a few minutes to allow the database to automatically start up.

```
reboot
```

27. Login as oracle to srv1 and verify that the database has automatically started.

```
ps -ef | grep pmon
sqlplus / as sysdba
```

28. As root, shut down srv1

```
shutdown -h now
```

Note: In real life, after creating the database, we normally set the database to operate in archivelog mode. You will learn about this procedure later in the course.

Creating an Oracle non-CDB Database in the Windows Platform

In the following steps, you will use the dbca to create an Oracle 19c non-CDB database named orawindb.

29. Start up winsrv

30. Login to winsrv as oracle

31. Open a command prompt window **as administrator** and issue the following command to verify that no listener is running in the system.

```
lsnrctl status
```

32. Run the following command to invoke Network Configuration Assistant utility.

We created the Listener using the silent mode in the previous section. Now, you will create the Listener using the netca utility.

```
netca
```

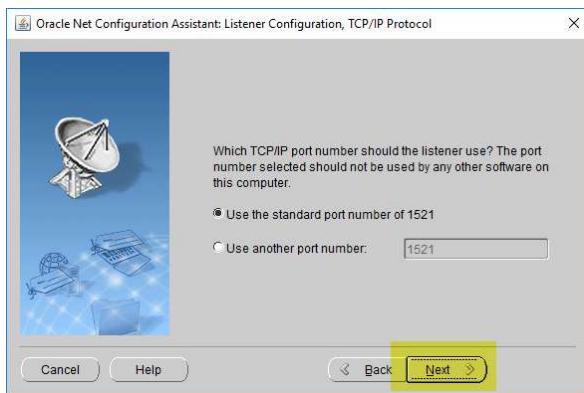
33. Respond to the utility windows as follows:





Observe in the following window, you should provide the password of `oraclesvc` user. This is a service user and its password should never expire. If you, for any reason, change its password in the future, you must update the password of the running listener service as well.





- 34.** In the command line window, type the following command to open the services window.

Observe the **OracleOraDB19Home1TNSListener** is created and running.

```
services.msc
```

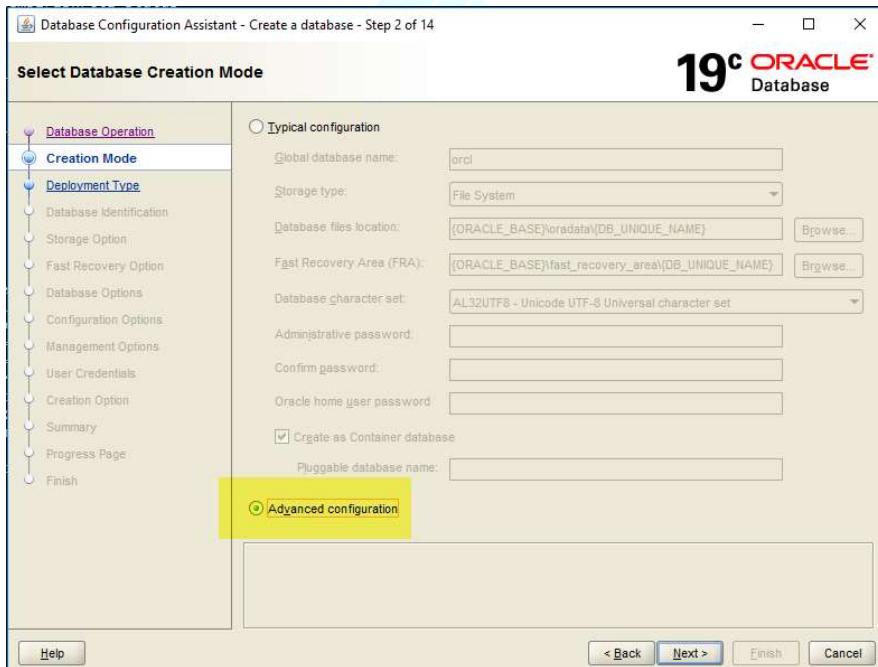
- 35.** Open the properties of the service > click on the "Log On" tab > observe that the service is running as `oraclesvc`

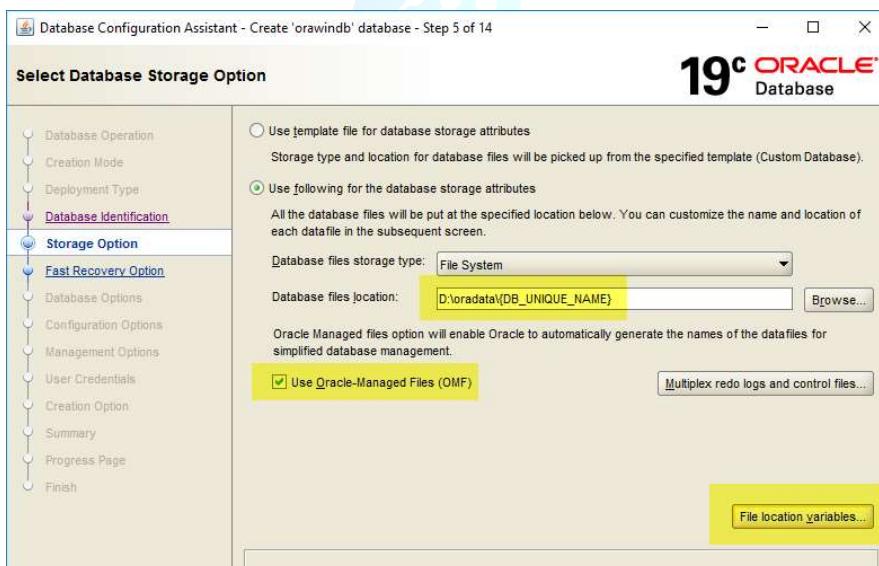
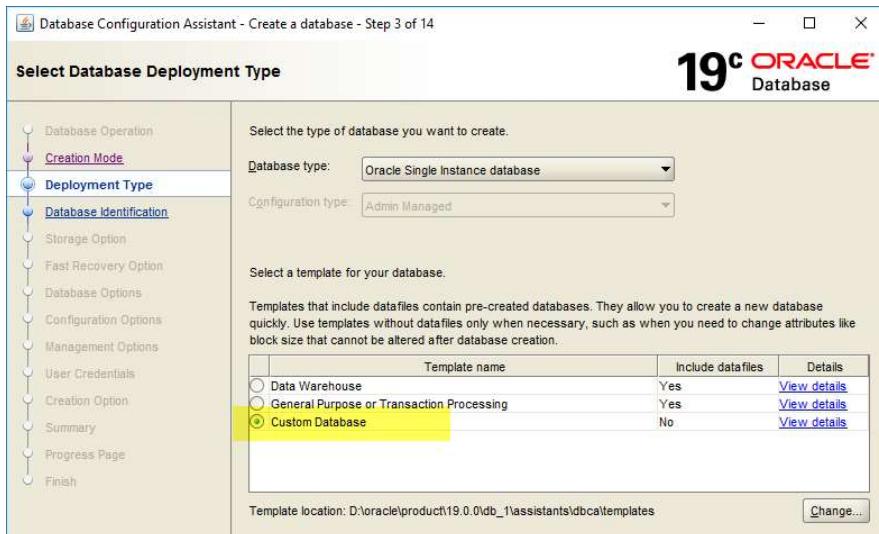
- 36.** In the command prompt window, issue the following command to verify that the listener is running.

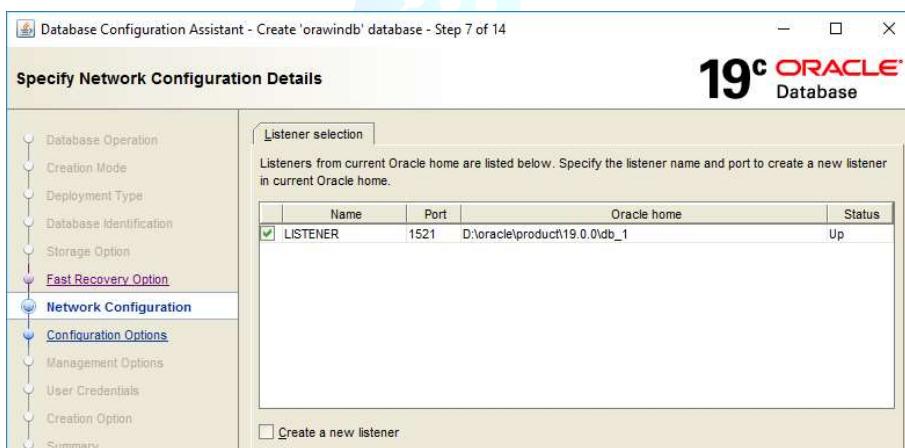
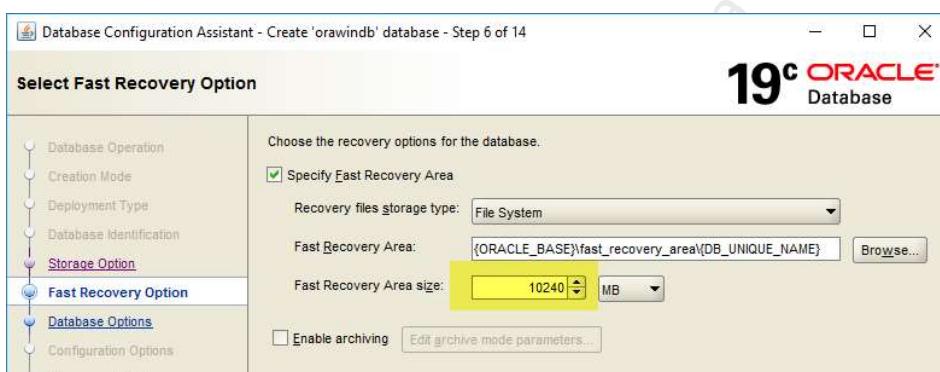
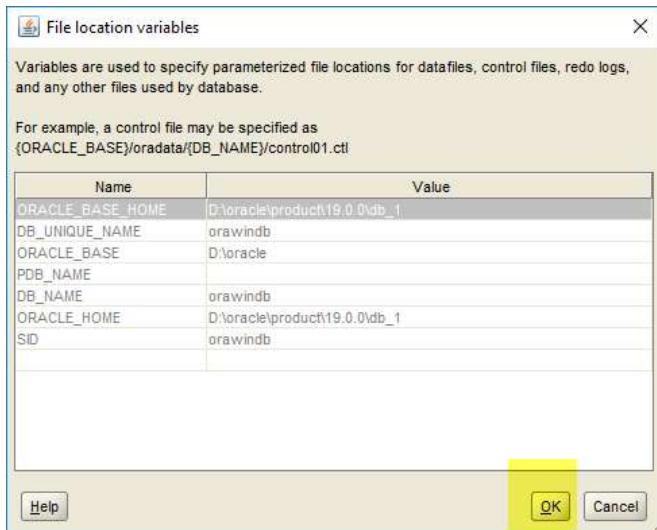
```
lsnrctl status
```

- 37.** Invoke the `dbca` utility and respond to its windows as follows. It is pretty much similar to our responses in Linux platform.

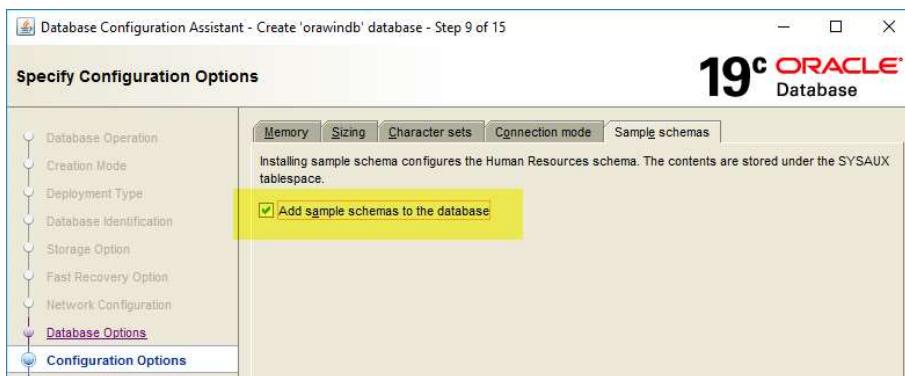
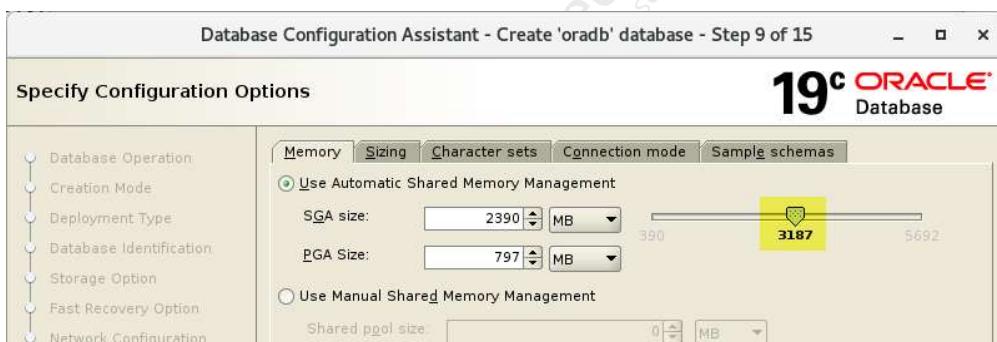
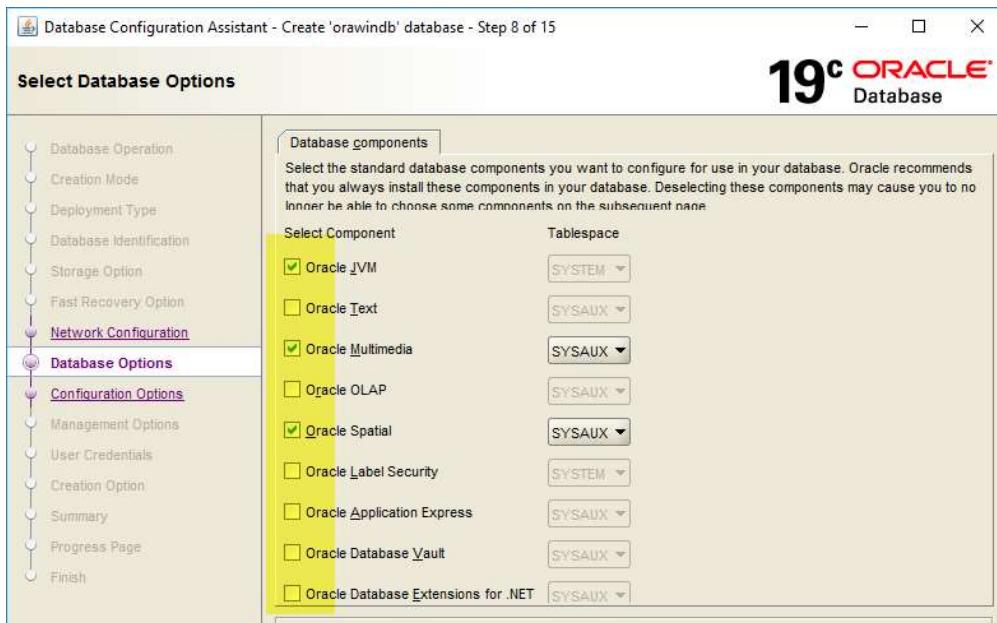
```
dbca
```

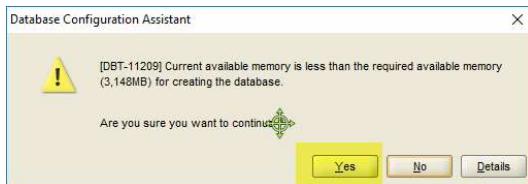




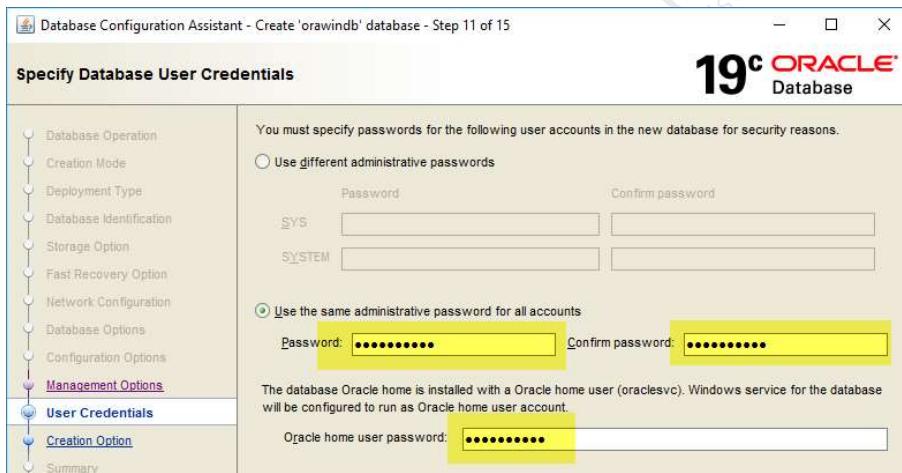


You need to select the database options as follows because they are needed by the sample schema that you will install in the database.

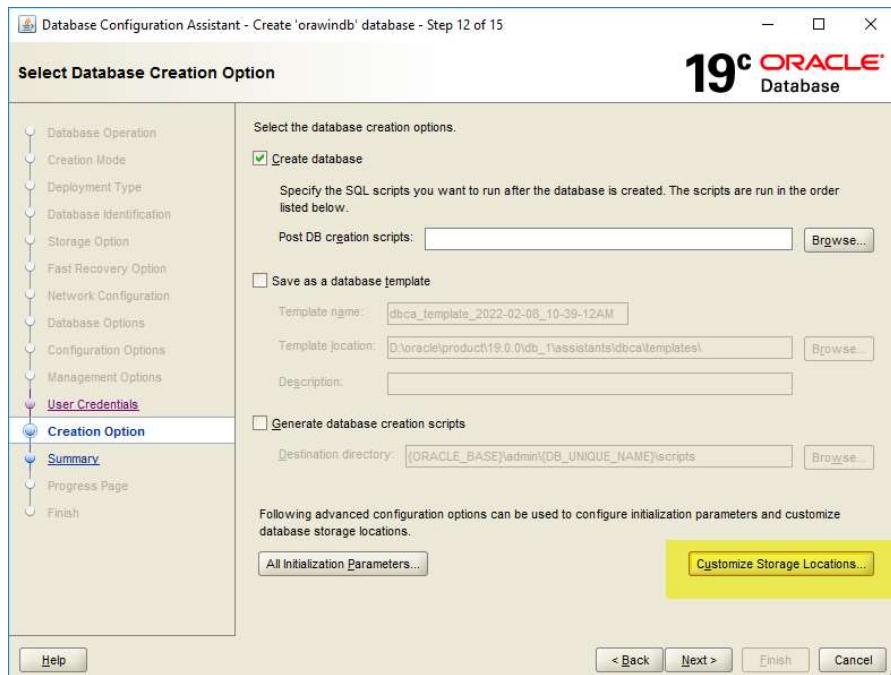




For the following window, we do not need to configure the EM Express in this vm because we will not need it in this vm in the course. In real life scenario, you may consider configuring it.

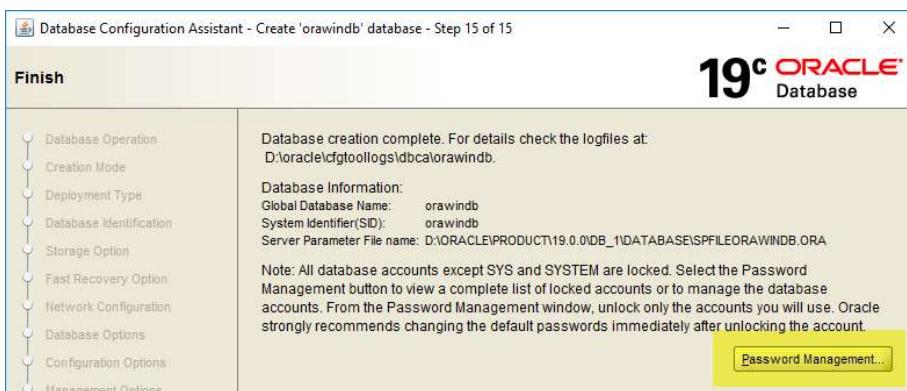
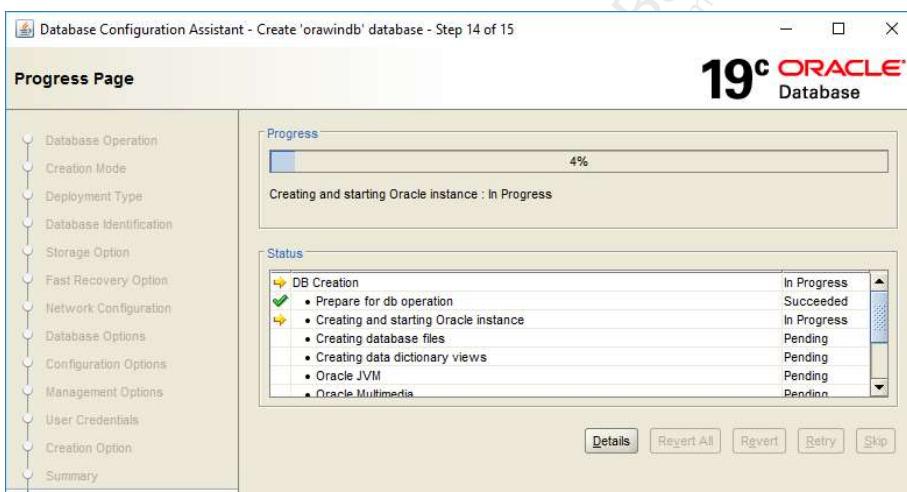
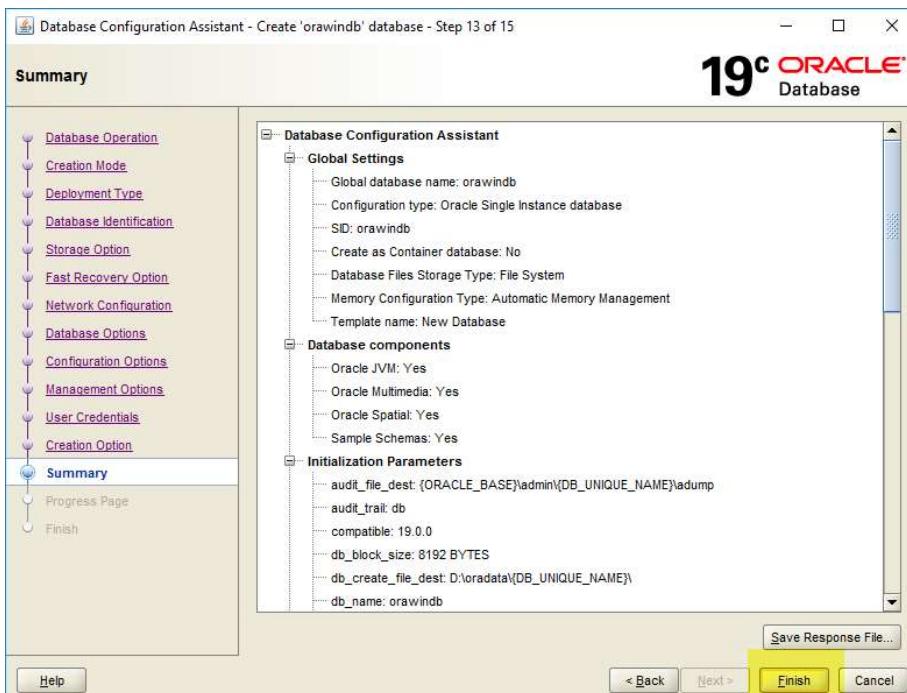


In the following window, click on “**Customize Storage Locations**” button. Explore the contents of the popup window.

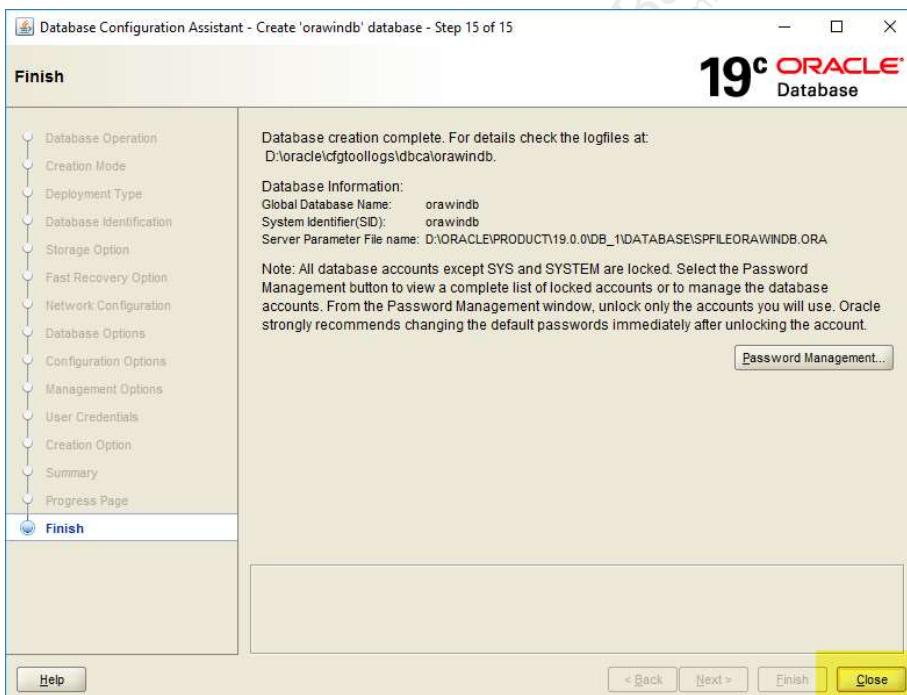
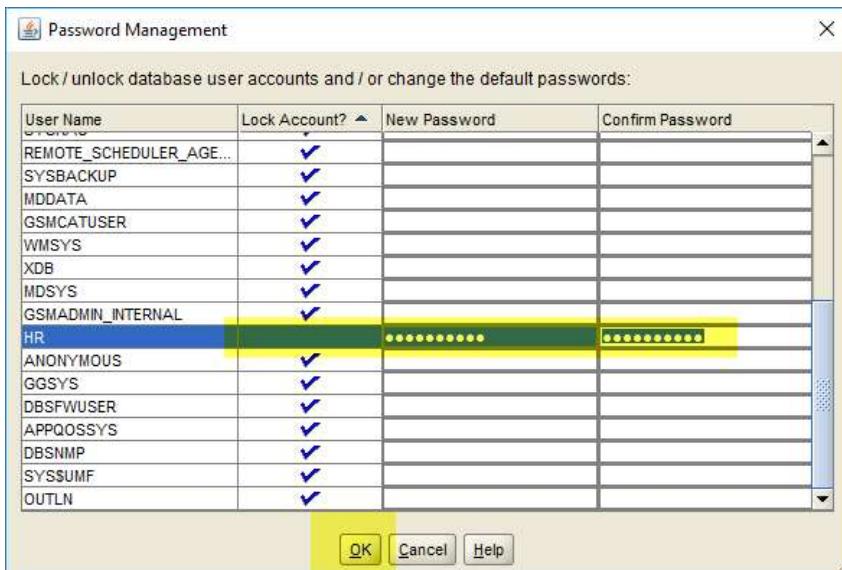


Observe the following in the window:

- The maximum data files in the database is set by default to 100. For some large databases, this might not be enough and you may need to increase it. This value can hardly be changed after the database is created.
- The size of each redo log group member is 204800 KB, which is equivalent to 200 MB. We have 3 groups. Therefore, 600 MB will be occupied by redo log files. For real life databases, this size could be fair enough or we may need to increase it depending on the transaction generation rate.



In the following window, we want to enable the HR account and set its password. Untick the "Lock Account" flag beside the HR user. Set its password. Then click on OK button.



38. Invoke SQL*Plus and login to the database as sysdba.

```
sqlplus / as sysdba
```

- 39.** List the sample schemas created by the dbca

Sample schemas are considered by the database that same as user schemas. Therefore, their ORACLE_MAINTAINED is set to false.

Only HR schema is created.

```
SELECT USERNAME FROM DBA_USERS WHERE ORACLE_MAINTAINED='N' ;
```

- 40.** Invoke the following command to login as HR user to the database to verify the database creation was successful.

```
conn hr/ABcd##1234
```

- 41.** Submit the following query.

```
SELECT COUNT(*) FROM EMPLOYEES ;
```

- 42.** Exit from SQL*Plus.

```
exit
```

- 43.** Open the file explorer. Navigate the D:\oracle directory and explore its contents.

- 44.** Invoke the Services window.

```
services.msc
```

- 45.** Observe that the following two services were added:

- OracleServiceORAWIBDB
- OracleVssWriterORAWIBDB

The first service startups the database. Without this service, we cannot have Oracle database up and running. However, normally, we do not control the status for the database from this service. We use the standard Oracle commands to change the status of the database. You will learn about those commands later in the course.

Thanks to the first service, we do not have to configure anything in Windows to make the database automatically startup when the system boots up.

The second service is used to integrate Oracle database with VSS-based backup systems. Because we are not going to use this backup type in the course, we do not need to have this service running.

- 46.** Stop the service OracleVssWriterORAWIBDB and disable it.

- 47.** Shutdown the machine.

Summary

- In this practice, we used the silent mode in `dbca` to create a non-CDB Oracle 19c database in a Linux-based machine `srv1`.
- We used the interactive mode in `dbca` to create a non-CDB Oracle 19c database in the Windows-based machine `winsrv`.

