Sri Lanka Institute of Information Technology

Database Systems and Storage Systems Assignment Report

Duneesha Samarakoon – IT20457952

30th September 2023

Contents

1. Pre-Requisites 2

2. Question 1 - Install the latest Oracle database version (19.X c or 21.X c) on the UNIX platform. 3

2. Question 2 - Create CDB in non-archive log mode named <your name> using DBCA and create a PDB called PDBDASS17

3. Question 3 - Answer the following questions 22

A. The database you created in question number 2 above uses a binary-type parameter file. When a parameter file is corrupted or missing how can you recover it and start the database? Demonstrate your answer using your database. 22

B. Increase the processes parameter by 10. 28

C. Create a new listener called LISTNER2 with port number 1522. 30

Question 4 - Create a new tablespace EXAMPLE1 of size 5 MB with one data file. After that, expand the tablespace size to 8MB by adding a new data file 35

Question 5 - Write a report about database security features and their use in Oracle 19c or 21c. The word count is 300 words 37

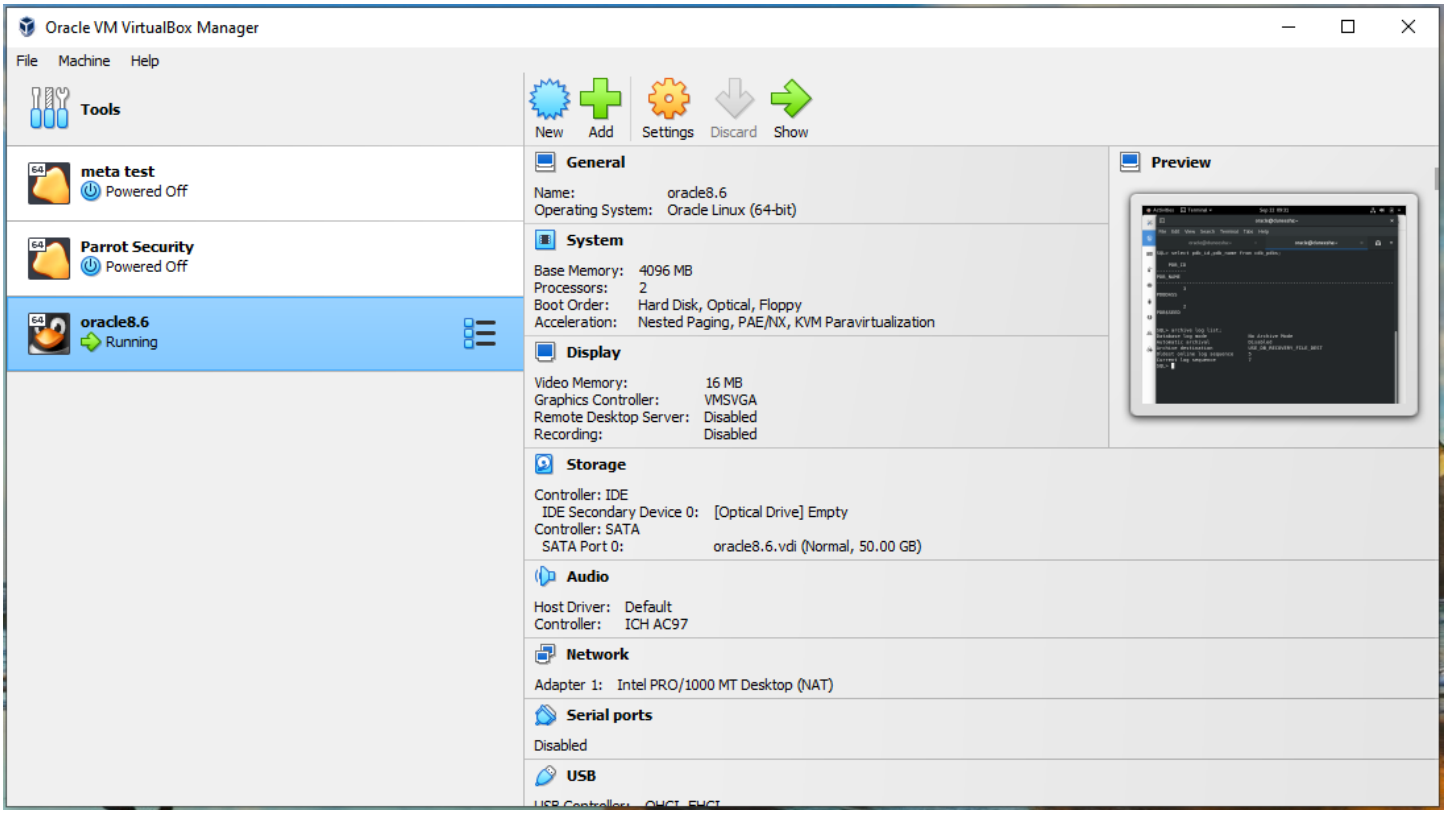
1. Pre-Requisites

To do this tasks it is important have running oracle Linux operating system and I have set up oracle Linux operating system in oracle virtual machine.

For the oracle machine I have allocated,

- 50GB hard drive space
- 4GB memory space.

And installed the operating system.



2. Question 1 - Install the latest Oracle database version (19.X c or 21.X c) on the UNIX platform.

1. Check host files and hostname as it is a must contain a fully qualified name for the server.

```
[duneesha@duneesha ~]$ cat /etc/hostname
duneesha
[duneesha@duneesha ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.122.1 duneesha.localdomain duneesha
[duneesha@duneesha ~]$
```

2. Check Memory and Swap space is of adequate amounts for oracle 19c installation. "Mem total" represents the physical RAM installed in the system, while "Swap total" represents the maximum amount of virtual memory provided by swap space on the disk. Physical RAM is faster than swap space, so the goal is to keep as much data in physical RAM as possible for optimal system performance.

```
[oracle@duneesha ~]$ grep MemTotal /proc/meminfo
MemTotal:        3730332 kB
[oracle@duneesha ~]$ grep SwapTotal /proc/meminfo
SwapTotal:       4149244 kB
[oracle@duneesha ~]$
```

3. Since there is no issue in memory allocated and swap space, we are good to move on to the next step which is to run the pre installation for oracle 19c which would Automatically downloads and installs any additional RPM packages needed for installing Oracle Grid Infrastructure and Oracle Database and resolves any dependencies.

```
[duneesha@duneesha ~]$ sudo dnf install -y oracle-database-preinstall-19c
Oracle Linux 8 BaseOS Latest (x86_64)           3.4 kB/s | 3.6 kB     00:01
Dependencies resolved.
================================================================================
 Package         Arch   Version                          Repository         Size
================================================================================
Installing:
 oracle-database-preinstall-19c
                 x86_64 1.0-2.el8                        ol8_appstream      31 k
Installing dependencies:
 glibc-devel     x86_64 2.28-189.1.0.1.el8               ol8_baseos_latest  80 k
 ksh             x86_64 20120801-257.0.1.el8             ol8_appstream     929 k
 libaio-devel    x86_64 0.3.112-1.el8                    ol8_baseos_latest  19 k
 libnsl          x86_64 2.28-189.1.0.1.el8               ol8_baseos_latest 103 k
 libstdc++-devel x86_64 8.5.0-10.0.2.el8                 ol8_appstream     2.1 M
 libxcrypt-devel x86_64 4.1.1-6.el8                      ol8_baseos_latest  25 k
 lm_sensors-libs x86_64 3.4.0-23.20180522git70f7e08.el8  ol8_baseos_latest  59 k
 make            x86_64 1:4.2.1-11.el8                   ol8_baseos_latest 498 k
 sysstat         x86_64 11.7.3-9.0.1.el8                 ol8_appstream     427 k

Transaction Summary
================================================================================
Install  10 Packages
```

```
Total download size: 4.2 M
Installed size: 18 M
Downloading Packages:
(1/10): libaio-devel-0.3.112-1.el8.x86_64.rpm    28 kB/s |  19 kB     00:00
(2/10): libxcrypt-devel-4.1.1-6.el8.x86_64.rpm  126 kB/s |  25 kB     00:00
(3/10): glibc-devel-2.28-189.1.0.1.el8.x86_64.r  86 kB/s |  80 kB     00:00
(4/10): libnsl-2.28-189.1.0.1.el8.x86_64.rpm    104 kB/s | 103 kB     00:00
(5/10): lm_sensors-libs-3.4.0-23.20180522git70f 230 kB/s |  59 kB     00:00
(6/10): make-4.2.1-11.el8.x86_64.rpm            695 kB/s | 498 kB     00:00
(7/10): oracle-database-preinstall-19c-1.0-2.el 224 kB/s |  31 kB     00:00
(8/10): sysstat-11.7.3-9.0.1.el8.x86_64.rpm     975 kB/s | 427 kB     00:00
(9/10): libstdc++-devel-8.5.0-10.0.2.el8.x86_64 730 kB/s | 2.1 MB     00:02
(10/10): ksh-20120801-257.0.1.el8.x86_64.rpm    269 kB/s | 929 kB     00:03
--------------------------------------------------------------------------------
Total                                           940 kB/s | 4.2 MB     00:04
Oracle Linux 8 BaseOS Latest (x86_64)           190 kB/s | 3.1 kB     00:00
Importing GPG key 0xAD986DA3:
 Userid     : "Oracle OSS group (Open Source Software group) <build@oss.oracle.c
om>"
 Fingerprint: 76FD 3DB1 3AB6 7410 B89D B10E 8256 2EA9 AD98 6DA3
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
Key imported successfully
Running transaction check
Transaction check succeeded.
```

```
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : libxcrypt-devel-4.1.1-6.el8.x86_64                     1/10
  Installing       : glibc-devel-2.28-189.1.0.1.el8.x86_64                  2/10
  Running scriptlet: glibc-devel-2.28-189.1.0.1.el8.x86_64                  2/10
  Installing       : libstdc++-devel-8.5.0-10.0.2.el8.x86_64                3/10
  Installing       : ksh-20120801-257.0.1.el8.x86_64                        4/10
  Running scriptlet: ksh-20120801-257.0.1.el8.x86_64                        4/10
  Installing       : make-1:4.2.1-11.el8.x86_64                             5/10
  Running scriptlet: make-1:4.2.1-11.el8.x86_64                             5/10
  Installing       : lm_sensors-libs-3.4.0-23.20180522git70f7e08.el8.x8     6/10
  Running scriptlet: lm_sensors-libs-3.4.0-23.20180522git70f7e08.el8.x8     6/10
  Installing       : sysstat-11.7.3-9.0.1.el8.x86_64                        7/10
  Running scriptlet: sysstat-11.7.3-9.0.1.el8.x86_64                        7/10
  Installing       : libnsl-2.28-189.1.0.1.el8.x86_64                       8/10
  Installing       : libaio-devel-0.3.112-1.el8.x86_64                      9/10
  Running scriptlet: oracle-database-preinstall-19c-1.0-2.el8.x86_64       10/10
  Installing       : oracle-database-preinstall-19c-1.0-2.el8.x86_64       10/10
  Running scriptlet: oracle-database-preinstall-19c-1.0-2.el8.x86_64       10/10
  Verifying        : glibc-devel-2.28-189.1.0.1.el8.x86_64                  1/10
  Verifying        : libaio-devel-0.3.112-1.el8.x86_64                      2/10
  Verifying        : libnsl-2.28-189.1.0.1.el8.x86_64                       3/10
```

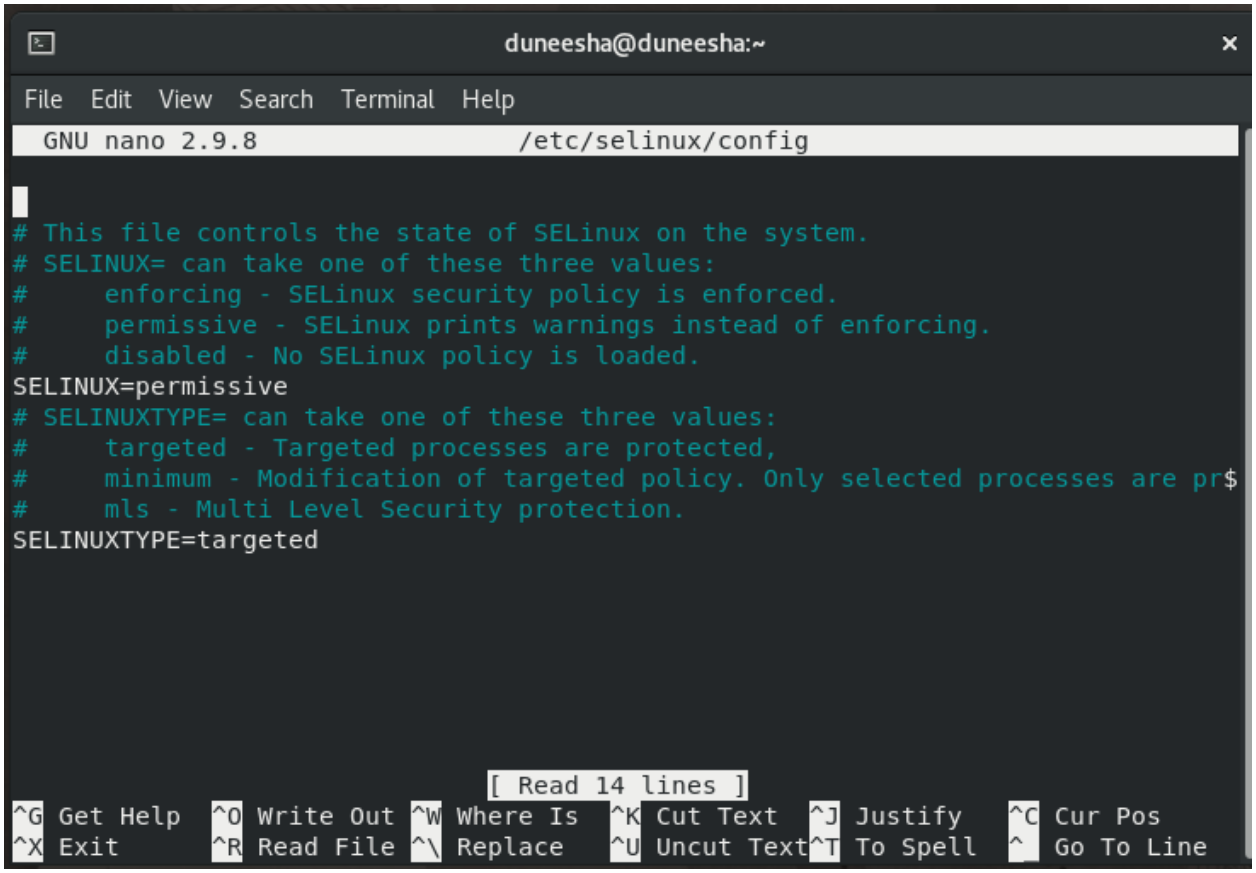
```
  Verifying        : libaio-devel-0.3.112-1.el8.x86_64                      2/10
  Verifying        : libnsl-2.28-189.1.0.1.el8.x86_64                       3/10
  Verifying        : libxcrypt-devel-4.1.1-6.el8.x86_64                     4/10
  Verifying        : lm_sensors-libs-3.4.0-23.20180522git70f7e08.el8.x8     5/10
  Verifying        : make-1:4.2.1-11.el8.x86_64                             6/10
  Verifying        : ksh-20120801-257.0.1.el8.x86_64                        7/10
  Verifying        : libstdc++-devel-8.5.0-10.0.2.el8.x86_64                8/10
  Verifying        : oracle-database-preinstall-19c-1.0-2.el8.x86_64        9/10
  Verifying        : sysstat-11.7.3-9.0.1.el8.x86_64                       10/10

Installed:
  glibc-devel-2.28-189.1.0.1.el8.x86_64
  ksh-20120801-257.0.1.el8.x86_64
  libaio-devel-0.3.112-1.el8.x86_64
  libnsl-2.28-189.1.0.1.el8.x86_64
  libstdc++-devel-8.5.0-10.0.2.el8.x86_64
  libxcrypt-devel-4.1.1-6.el8.x86_64
  lm_sensors-libs-3.4.0-23.20180522git70f7e08.el8.x86_64
  make-1:4.2.1-11.el8.x86_64
  oracle-database-preinstall-19c-1.0-2.el8.x86_64
  sysstat-11.7.3-9.0.1.el8.x86_64

Complete!
[duneesha@duneesha ~]$
```

4. The pre-installation creates a new user as oracle. It is essential to that a password is set for the newly created user.

```
[duneesha@duneesha ~]$ sudo passwd oracle
Changing password for user oracle.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[duneesha@duneesha ~]$
```

5. Next, SELinux is set to permissive in which permission denials are logged but not enforced. This would allow the installation to run smoothly as setting SELinux to enforcing may cause some features to not run or install.

```
[duneesha@duneesha ~]$ sudo nano /etc/selinux/config
[duneesha@duneesha ~]$
```



6. After set enforce is set to permissive to switch the mode SELinux is running in from enforcing to permissive without requiring rebooting.

```
[duneesha@duneesha ~]$ sudo setenforce
usage:  setenforce [ Enforcing | Permissive | 1 | 0 ]
[duneesha@duneesha ~]$ sudo setenforce Permissive
[duneesha@duneesha ~]$
```

7. If Linux firewall is enabled, it is required to disable it to continue with the installation without any issue.

```
[duneesha@duneesha ~]$ sudo systemctl stop firewalld
[duneesha@duneesha ~]$ sudo systemctl disable firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[duneesha@duneesha ~]$
```

8. Afterwards we require to create the directories which are required for oracle installation. dbhome_1 will act as the oracle home directory. After creating directories read and write permissions should be set for oracle user as well as the current user who is the administrator.

```
[duneesha@duneesha ~]$ sudo mkdir -p /u01/app/oracle/product/19.0.0/dbhome_1
[sudo] password for duneesha:
[duneesha@duneesha ~]$ sudo mkdir -p /u02/oradata
[duneesha@duneesha ~]$ sudo chown -R oracle:oinstall /u01 /u02
[duneesha@duneesha ~]$ ls -ld /u01
drwxr-xr-x. 3 oracle oinstall 17 Sep 22 03:53 /u01
[duneesha@duneesha ~]$ ls -ld /u02
drwxr-xr-x. 3 oracle oinstall 21 Sep 22 03:53 /u02
[duneesha@duneesha ~]$ sudo chmod -R 775 /u01 /u02
[duneesha@duneesha ~]$ ls ld /u01
ls: cannot access 'ld': No such file or directory
/u01:
app
[duneesha@duneesha ~]$ ls -ld /u01
drwxrwxr-x. 3 oracle oinstall 17 Sep 22 03:53 /u01
[duneesha@duneesha ~]$ ls -ld /u02
drwxrwxr-x. 3 oracle oinstall 21 Sep 22 03:53 /u02
[duneesha@duneesha ~]$
```

9. For the rest of the configurations change user to oracle user.

```
[duneesha@duneesha ~]$ su - oracle
Password:
[oracle@duneesha ~]$
```

10. Create a scripts directory and create a file named “setEnv.sh”. This file contains the environmental variable that needs to be configured for oracle database installation.

```
[oracle@duneesha ~]$ mkdir /home/oracle/scripts
[oracle@duneesha ~]$ cat > /home/oracle/scripts/setEnv.sh <<EOF
> export TMP=/tmp
> export TMPDIR=\$TMP
>
> export ORACLE_HOSTNAME=duneesha.localdomain
> export ORACLE_UNIQNAME=mydb
> export ORACLE_BASE=/u01/app/oracle
> export ORACLE_HOME=\$ORACLE_BASE/product/19.0.0/dbhome_1
> export ORA_INVENTORY=/u01/app/oraInventory
> export ORACLE_SID=mydb
> export PDB_NAME=pdb1
> export DATA_DIR=/u02/oradata
>
> export PATH=/usr/sbin:/usr/local/bin:\$PATH
> export PATH=\$ORACLE_HOME/bin:\$PATH
>
> export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:/lib:/usr/lib
> export CLASSPATH=\$ORACLE_HOME/jlib:\$ORACLE_HOME/rdbms/jlib
> EOF
[oracle@duneesha ~]$
```

11. Check if the configurations are successfully done, run an echo command to check the oracle home directory. If there is no output, do a source to apply the changes done as below.

```
[oracle@duneesha ~]$ echo $ORACLE_HOME

[oracle@duneesha ~]$ source scripts/setEnv.sh
[oracle@duneesha ~]$ echo $ORACLE_HOME
/u01/app/oracle/product/19.0.0/dbhome_1
[oracle@duneesha ~]$
```

12. Add a reference to the "setEnv.sh" file at the end of the "/home/oracle/.bash_profile" file.

```
[oracle@duneesha ~]$ echo ". /home/oracle/scripts/setEnv.sh" >> .bash_profile
[oracle@duneesha ~]$ cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
. /home/oracle/scripts/setEnv.sh
[oracle@duneesha ~]$
```

13. Download the oracle installation setup zip file directly to the VM or download to your host machine and then copy to your VM using a shared folder.

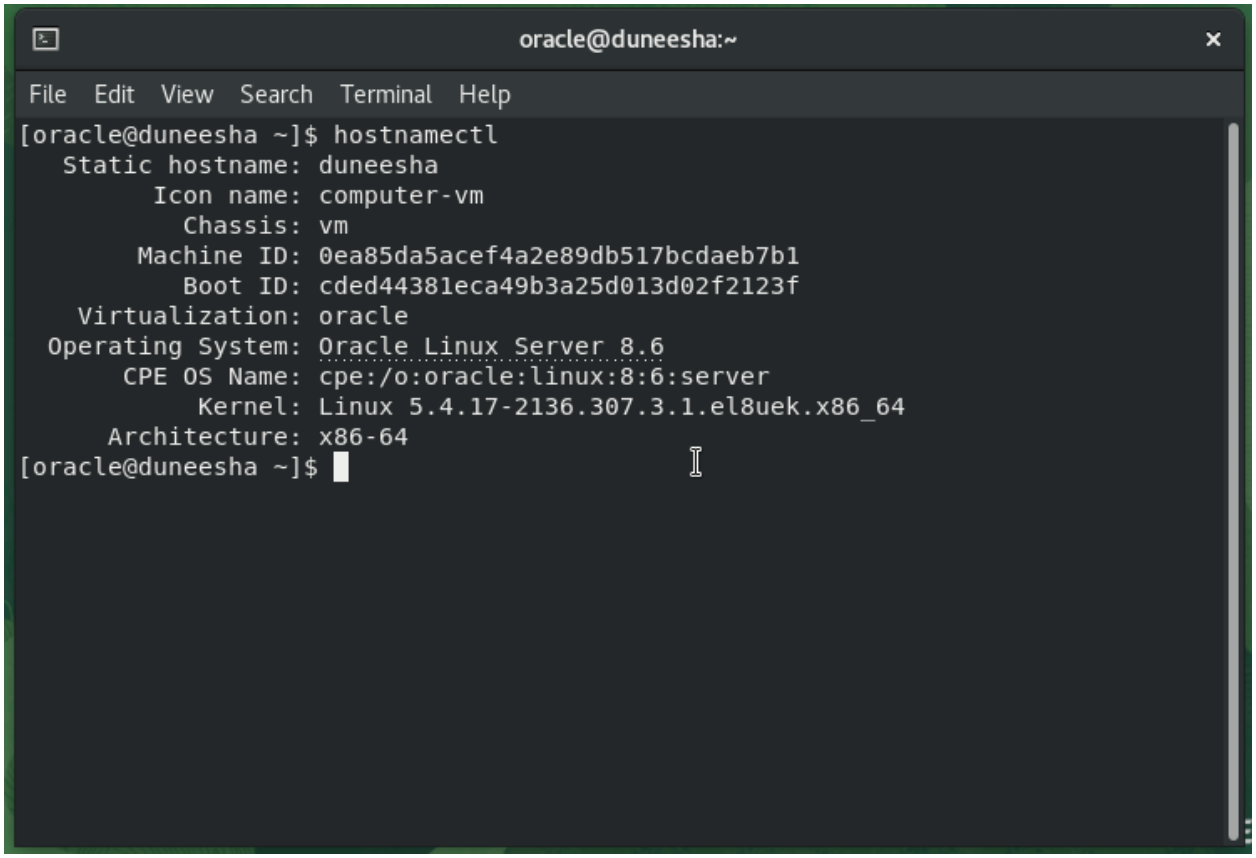
```
[root@duneesha duneesha]# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[root@duneesha duneesha]# cd Downloads
[root@duneesha Downloads]# ls
LINUX.X64_193000_db_home.zip
[root@duneesha Downloads]# cp LINUX.X64_193000_db_home.zip /u01/app/oracle/produ
ct/19.0.0/dbhome_1
[root@duneesha Downloads]# su oracle
[oracle@duneesha Downloads]$ ls
LINUX.X64_193000_db_home.zip
[oracle@duneesha Downloads]$
```

14. After downloading change your directory to ORACLE_HOME and then unzip the downloaded file. Use -oq to avoid printing all the files that are unzipping on your terminal.

```
[oracle@duneesha dbhome_1]$ unzip -qo LINUX.X64_193000_db_home.zip
[oracle@duneesha dbhome_1]$ ls
addnode      env.ora                      olap      root.sh.old
apex         has                          OPatch    root.sh.old.1
assistants   hs                           opmn      runInstaller
bin          install                      oracore   schagent.conf
clone        instantclient                ord       sdk
crs          inventory                    ords      slax
css          javavm                       oss       sqldeveloper
ctx          jdbc                         oui       sqlj
cv           jdk                          owm       sqlpatch
data         jlib                         perl      sqlplus
dbjava       ldap                         plsql     srvm
dbs          lib                          precomp   suptools
deinstall    LINUX.X64_193000_db_home.zip QOpatch   ucp
demo         md                           R         usm
diagnostics  mgw                          racg      utl
dmu          network                      rdbms     wwg
drdaas       nls                          relnotes  xdk
dv           odbc                         root.sh
[oracle@duneesha dbhome_1]$
```

15. Reboot your system.

16. Check your system details and note down the oracle Linux version.

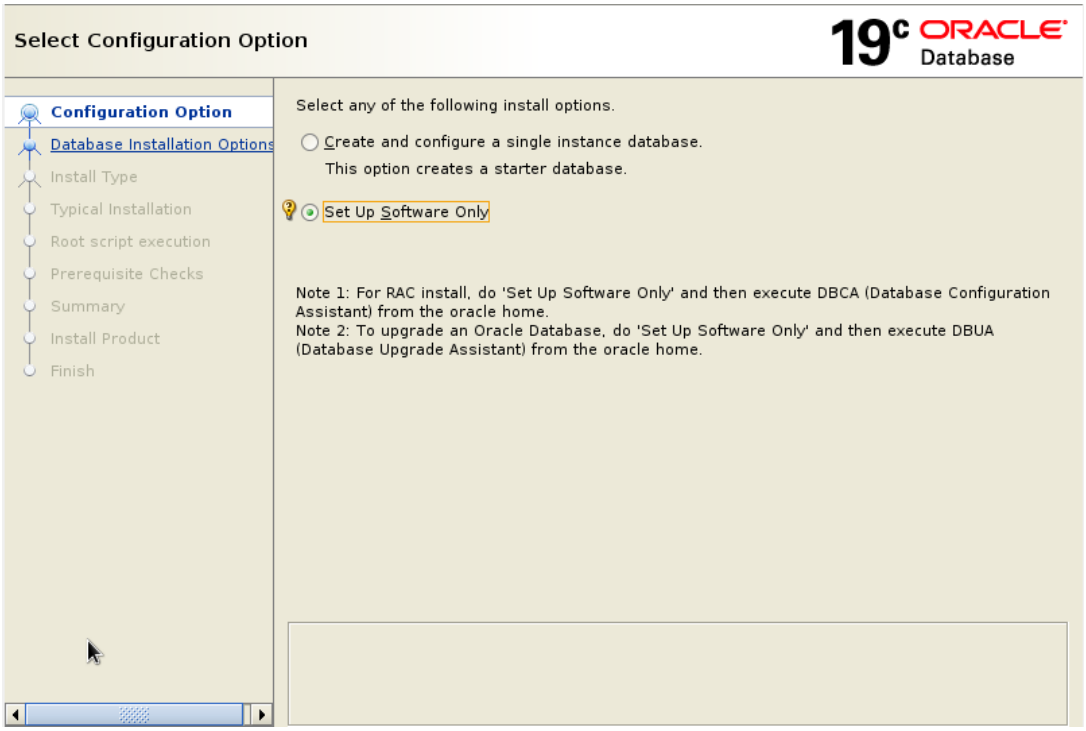


17. Set oracle version as OEL8 and change directory to ORACLE_HOME.

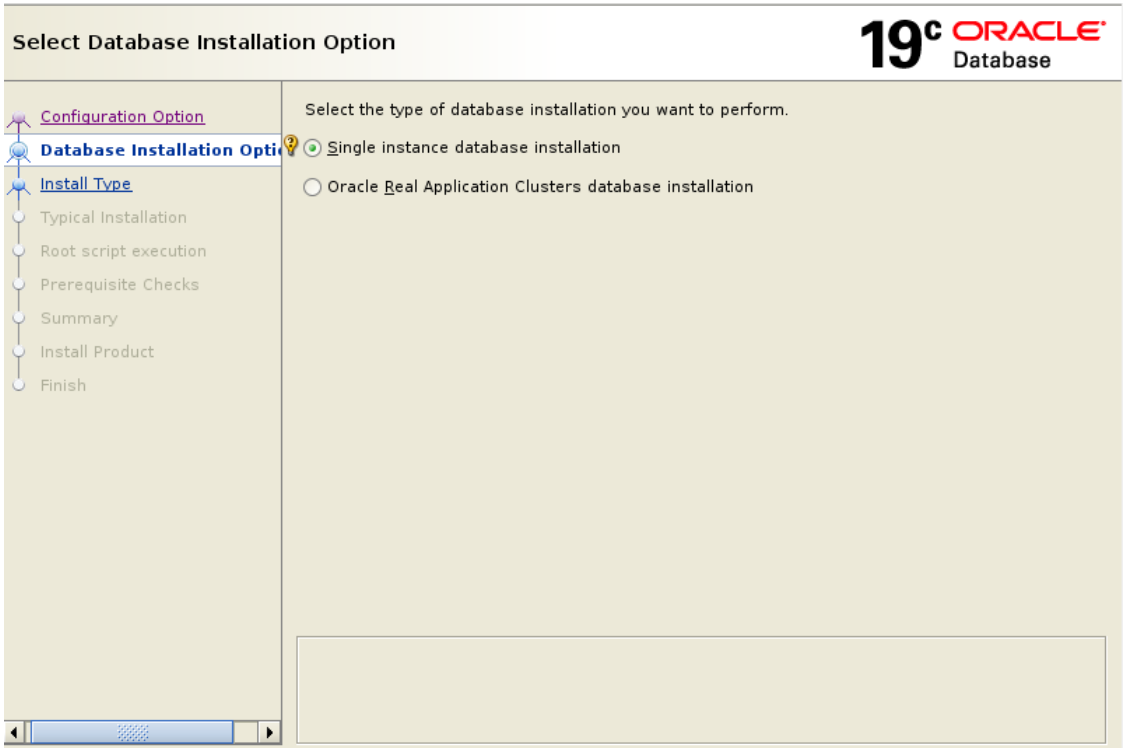
18. Run installation file to open the setup wizard GUI.

19. Proceed with the following steps to install oracle database 19c on wizard.

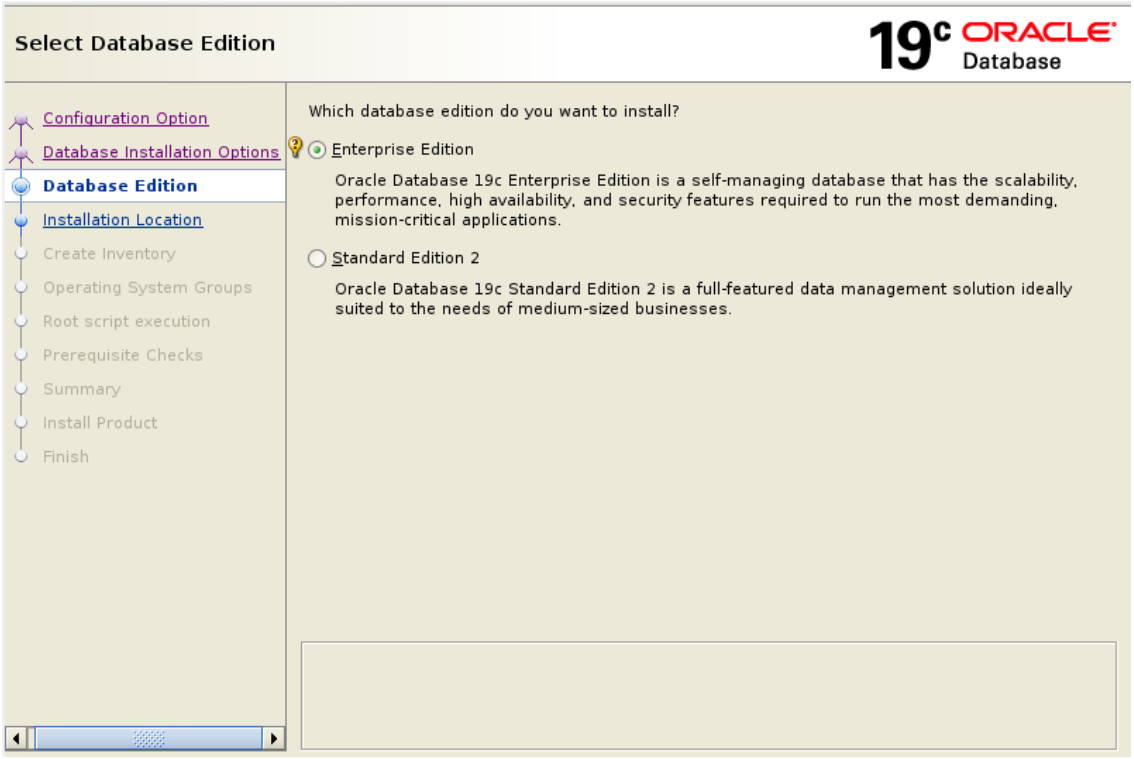
a. Step 1 – From Configuration option select software only.



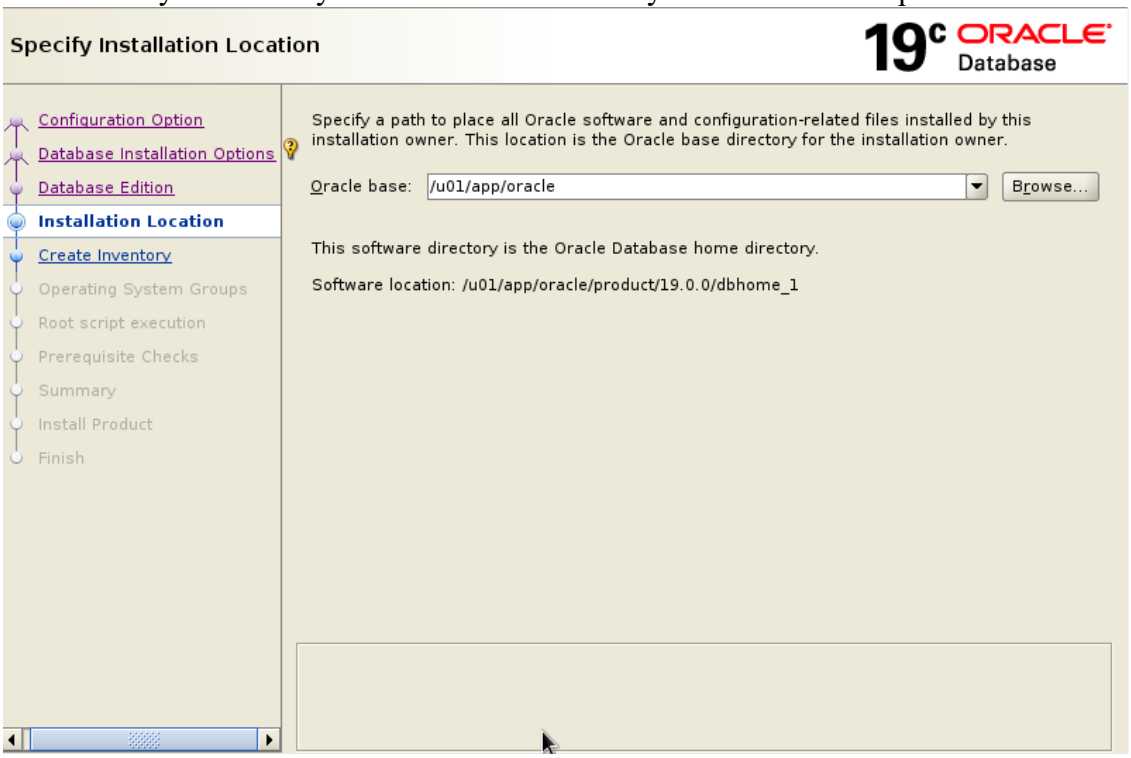
b. Step 2 – Select single instance database installation.



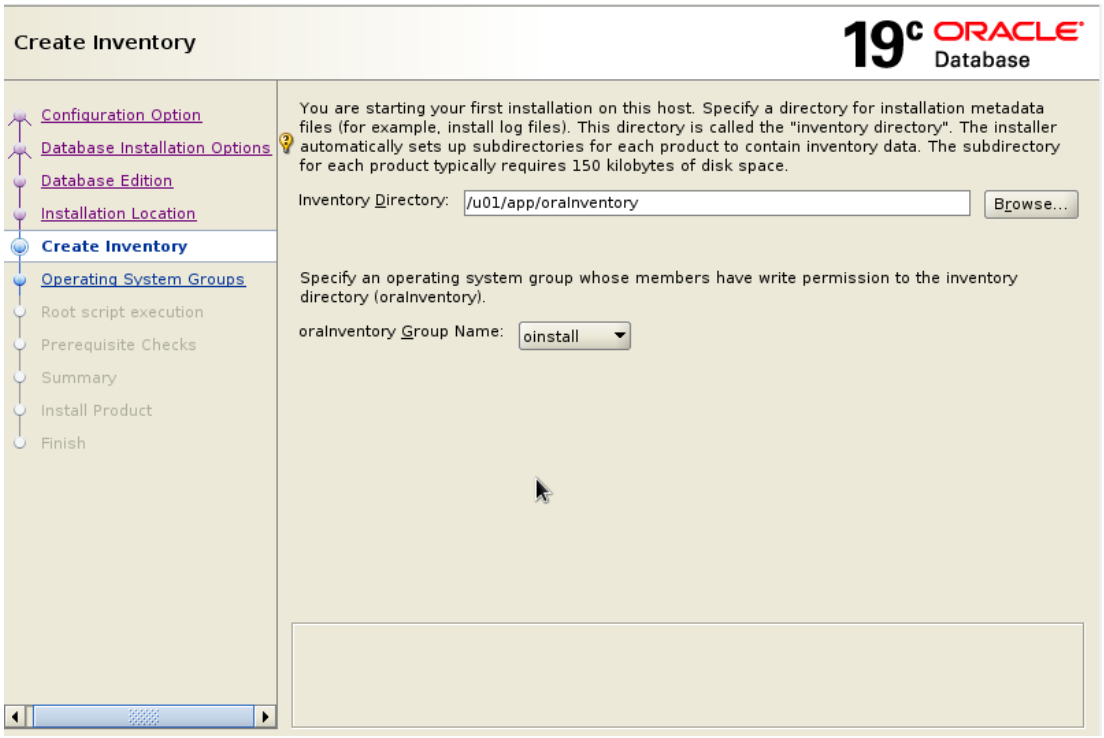
c. Step 3 – Select Enterprise edition.



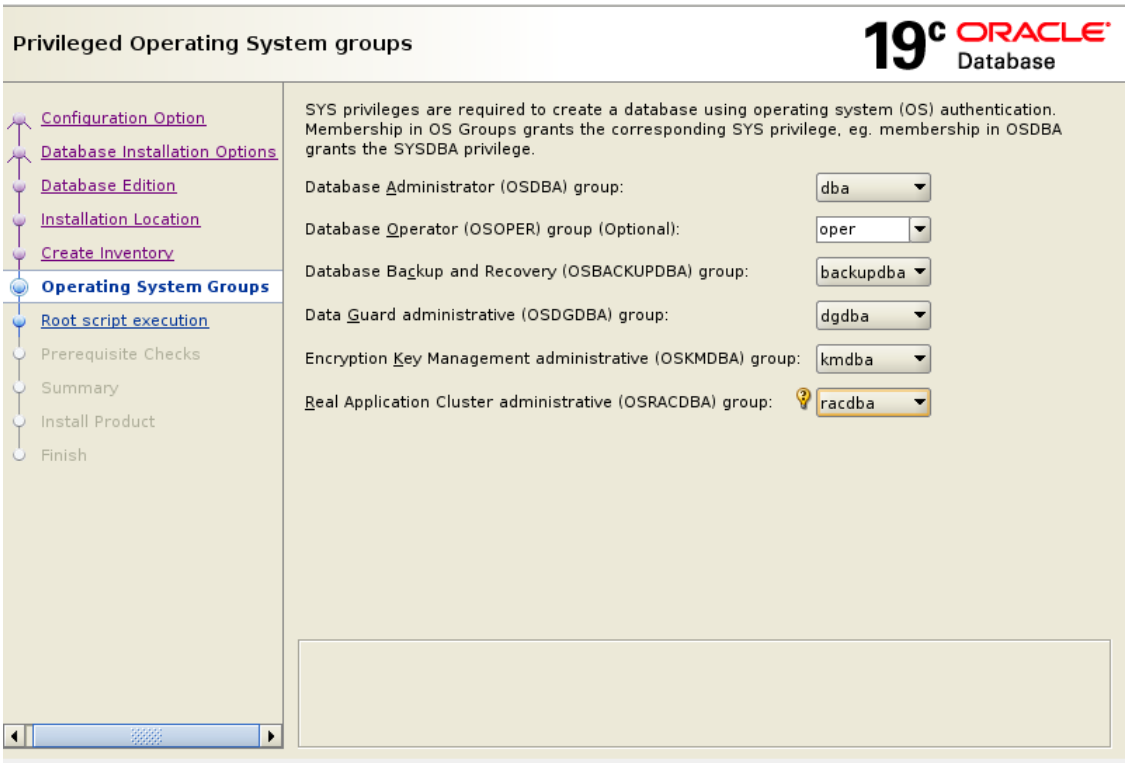
d. Step 4 – By default your oracle home directory will be set as oracle base path. In any case if it is not set manually browse to your oracle home directory to set oracle base path.



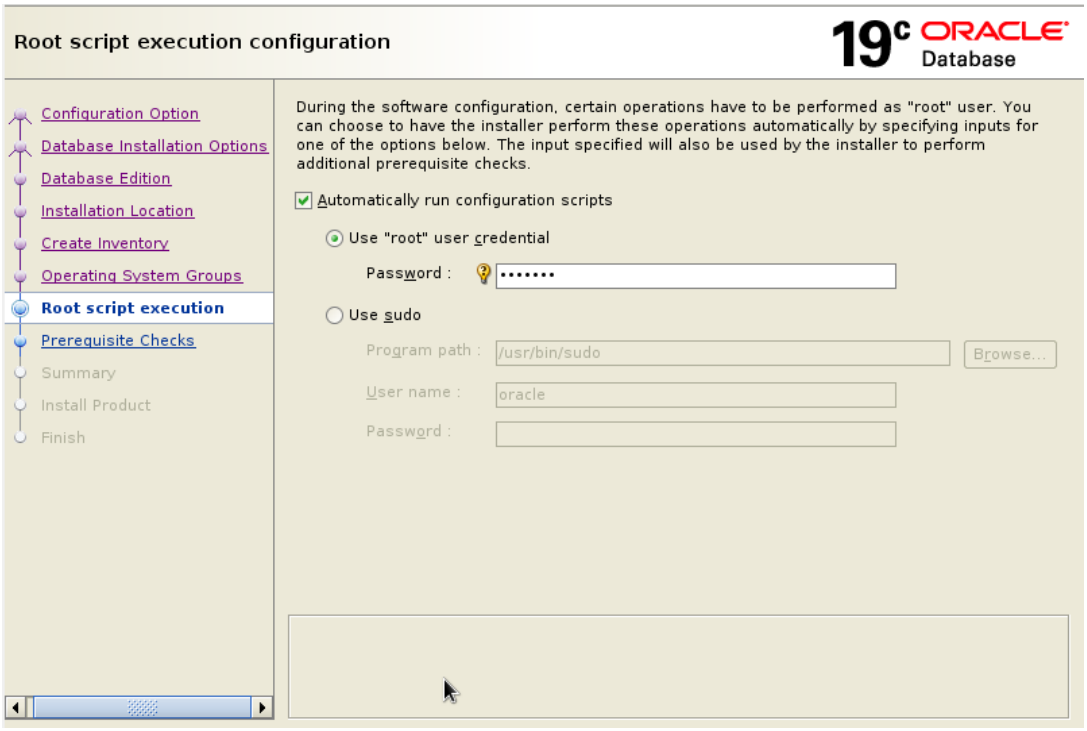
e. Step 5 – Leave create inventory as default.



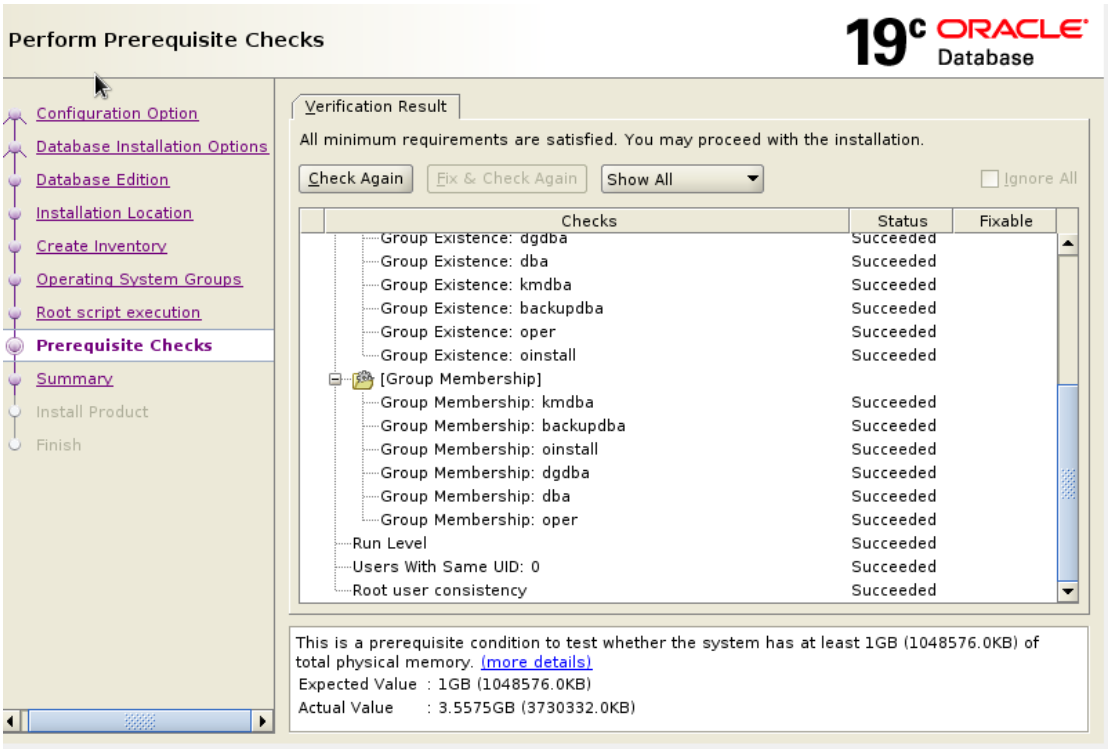
f. Step 6 - Leave the operating system group permissions in default settings.



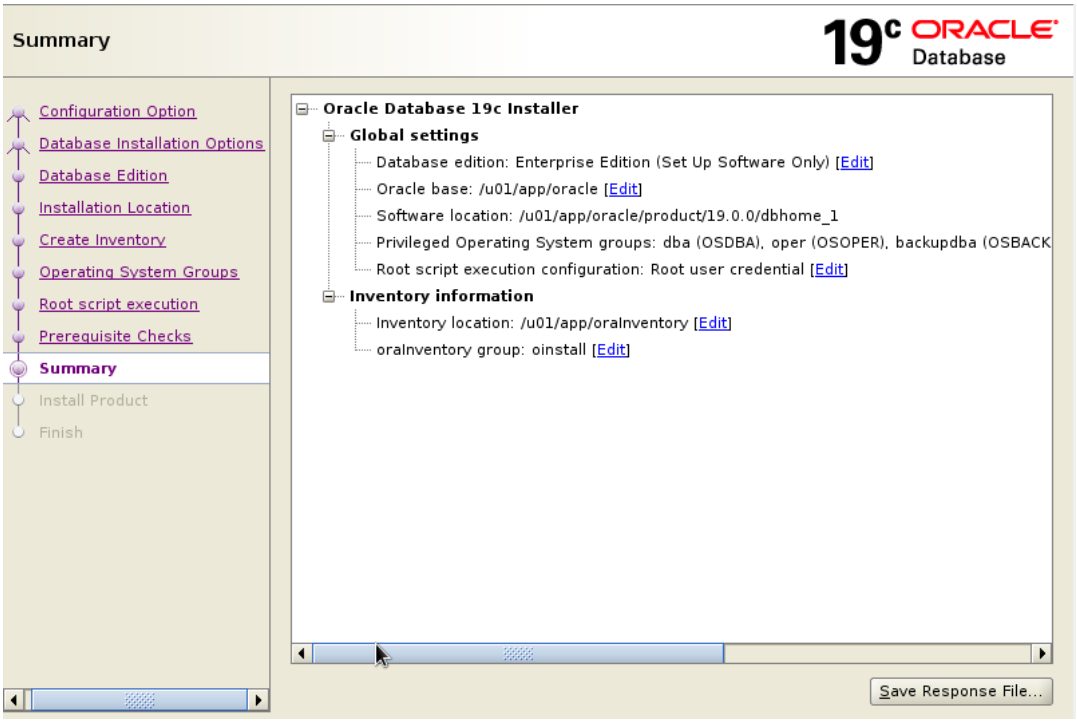
g. Step 7 – Since you have created root user and know root password add the root password at this step.



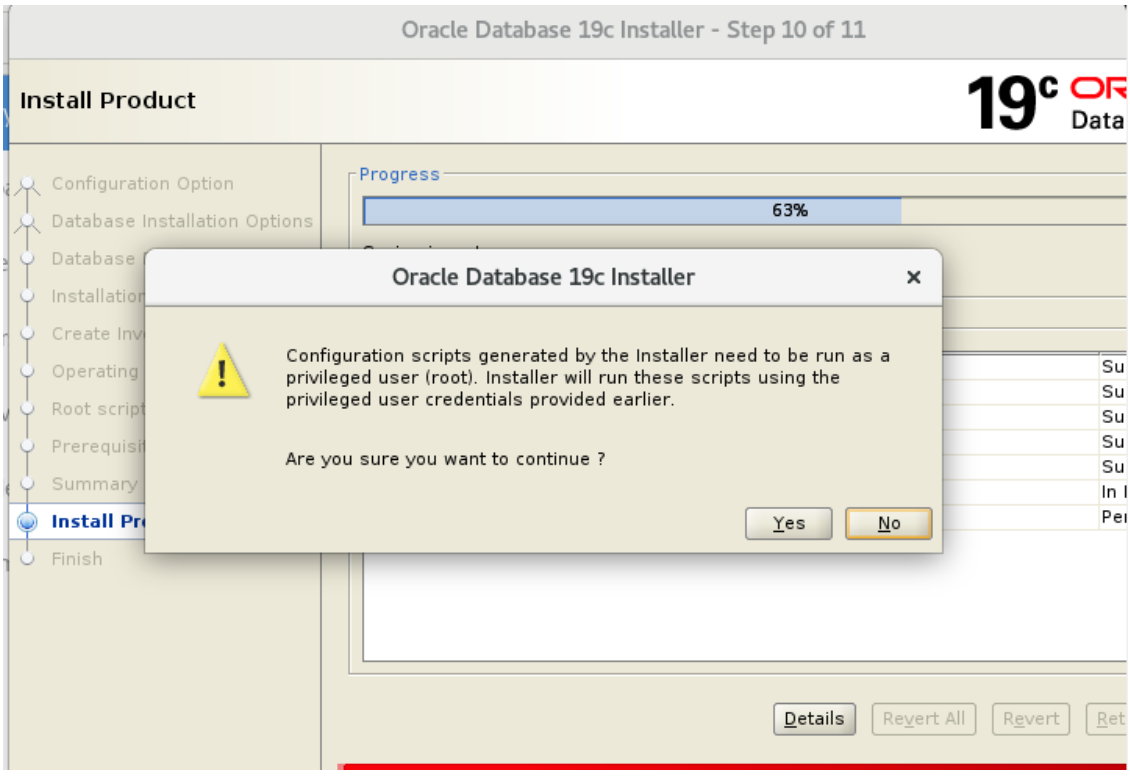
h. Step 8 – This step would check to see if all the requirements such as adequate swap total is available for the installation and such before installation begins and will give a warning if at least one such check fails.



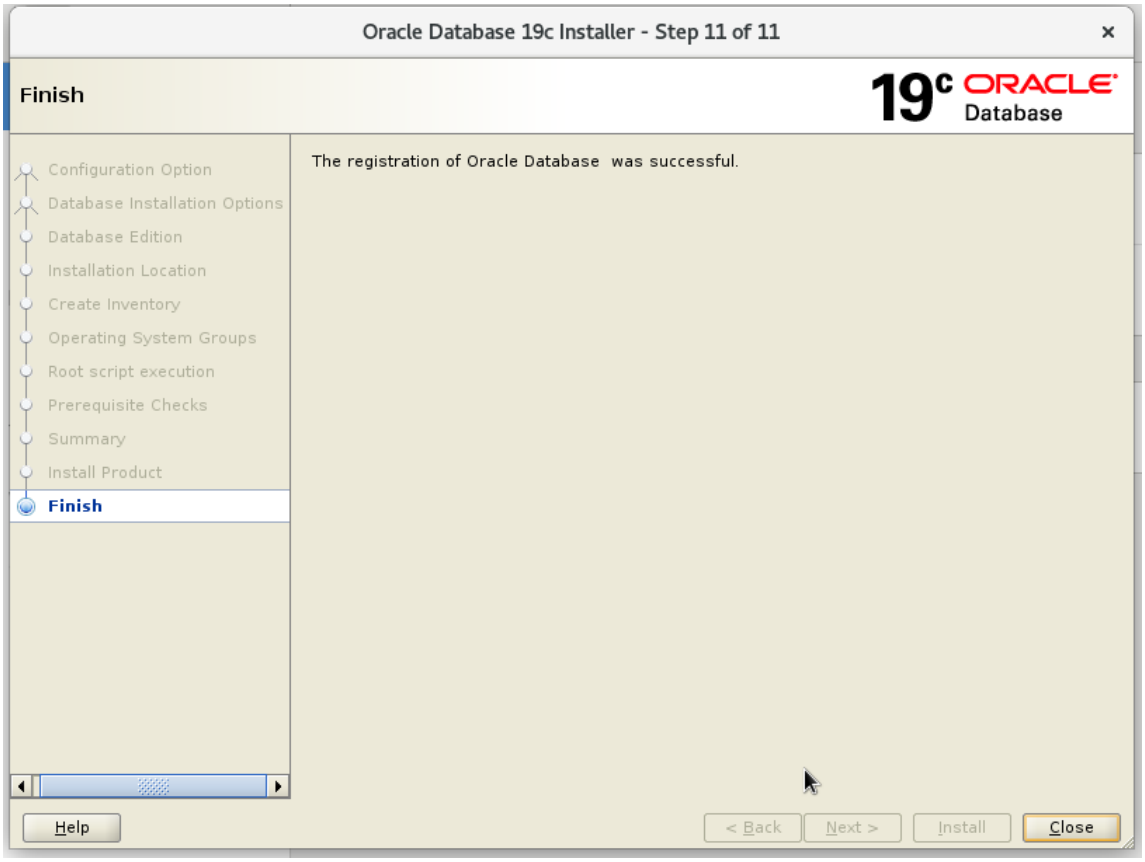
i. Step 9 – If all prerequisites are passed then check the summary and start installation.



j. Step 10 – At 63% completion of the installation a message will be prompt requiring your consent to run scripts as root user. Select yes to continue with the installation.



k. Step 11- Installation Successful



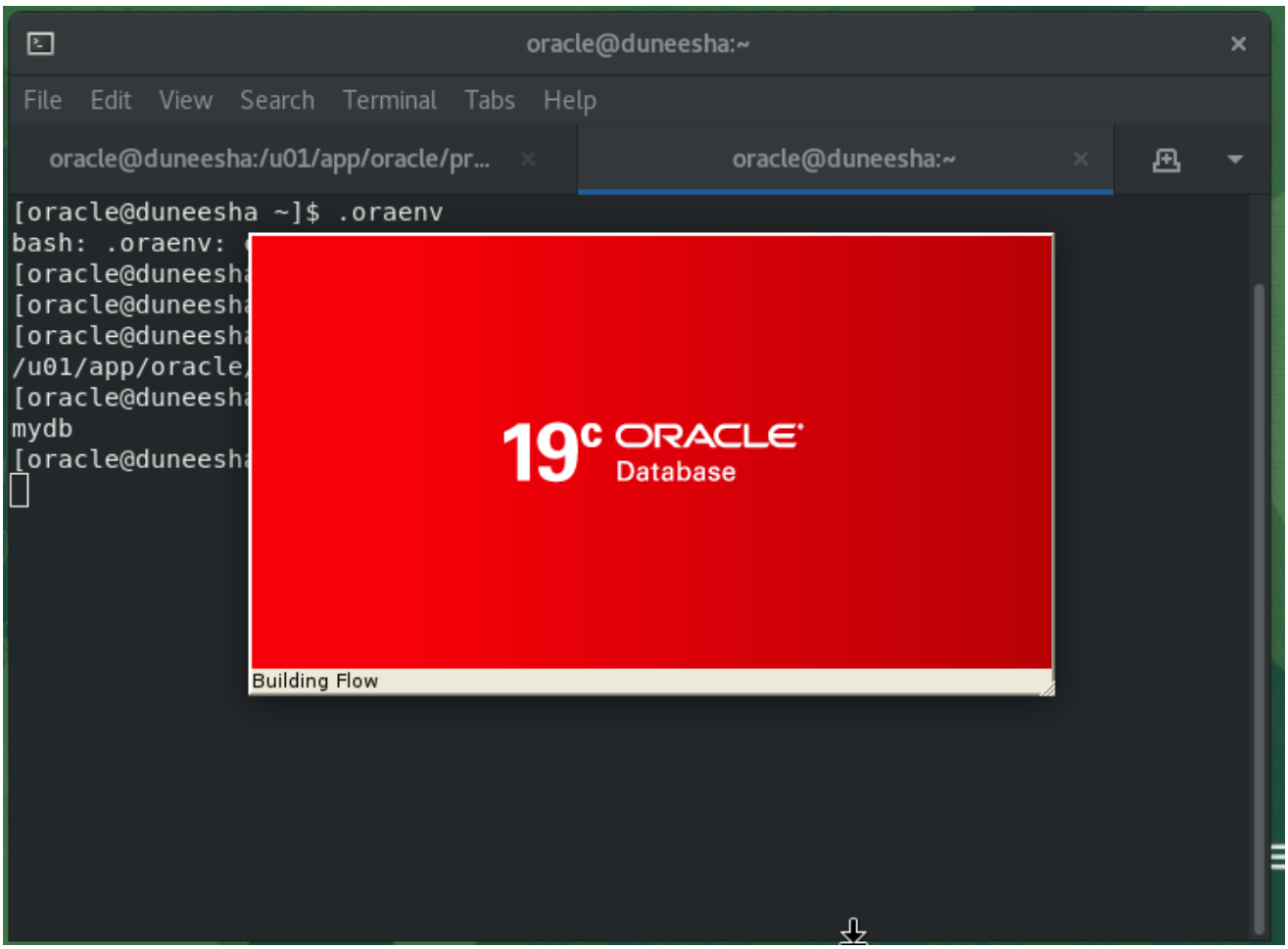

```
[oracle@duneesha dbhome_1]$ ./runInstaller
Launching Oracle Database Setup Wizard...

The response file for this session can be found at:
 /u01/app/oracle/product/19.0.0/dbhome_1/install/response/db_2023-09-22_06-05-19
AM.rsp

You can find the log of this install session at:
 /tmp/InstallActions2023-09-22_06-05-19AM/installActions2023-09-22_06-05-19AM.lo
g
Moved the install session logs to:
 /u01/app/oraInventory/logs/InstallActions2023-09-22_06-05-19AM
[oracle@duneesha dbhome_1]$
```

2. Question 2 - Create CDB in non-archive log mode named <your name> using DBCA and create a PDB called PDBDASS

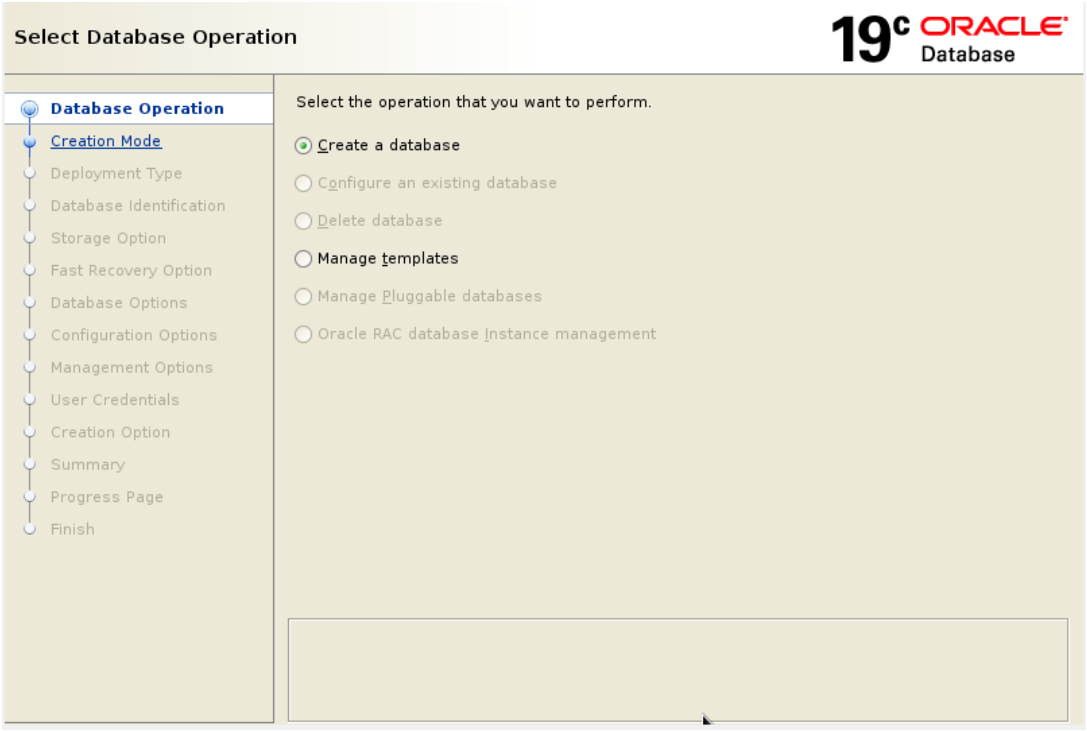
1. To create a new CDB and a PDB first check if you can go to your oracle home by using the command “cd $ORACLE_HOME”. If this command gives an error as no directory found it is required to set the environment variables again as below.

```
[oracle@duneesha dbhome_1]$ cd
[oracle@duneesha ~]$ echo $ORACLE_HOME
/u01/app/oracle/product/19.0.0/dbhome_1
[oracle@duneesha ~]$ echo $ORACLE_SID
mydb
[oracle@duneesha ~]$
```

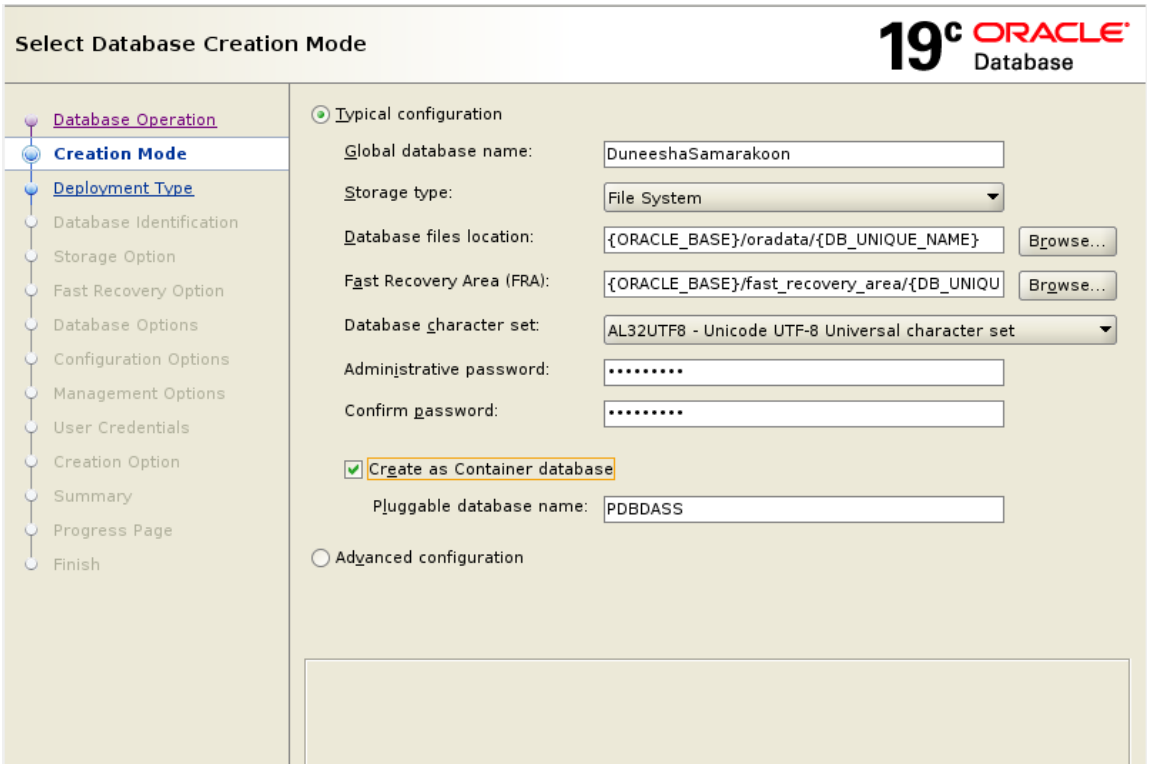
2. After setting environment run the “dbca” command (Database Configuration Assistant) which is an automated approach and preferred way to create a database.



3. Proceed with the following steps to create a CDB and PDB.
    a. Step 1 – Select create database and go to the next window.

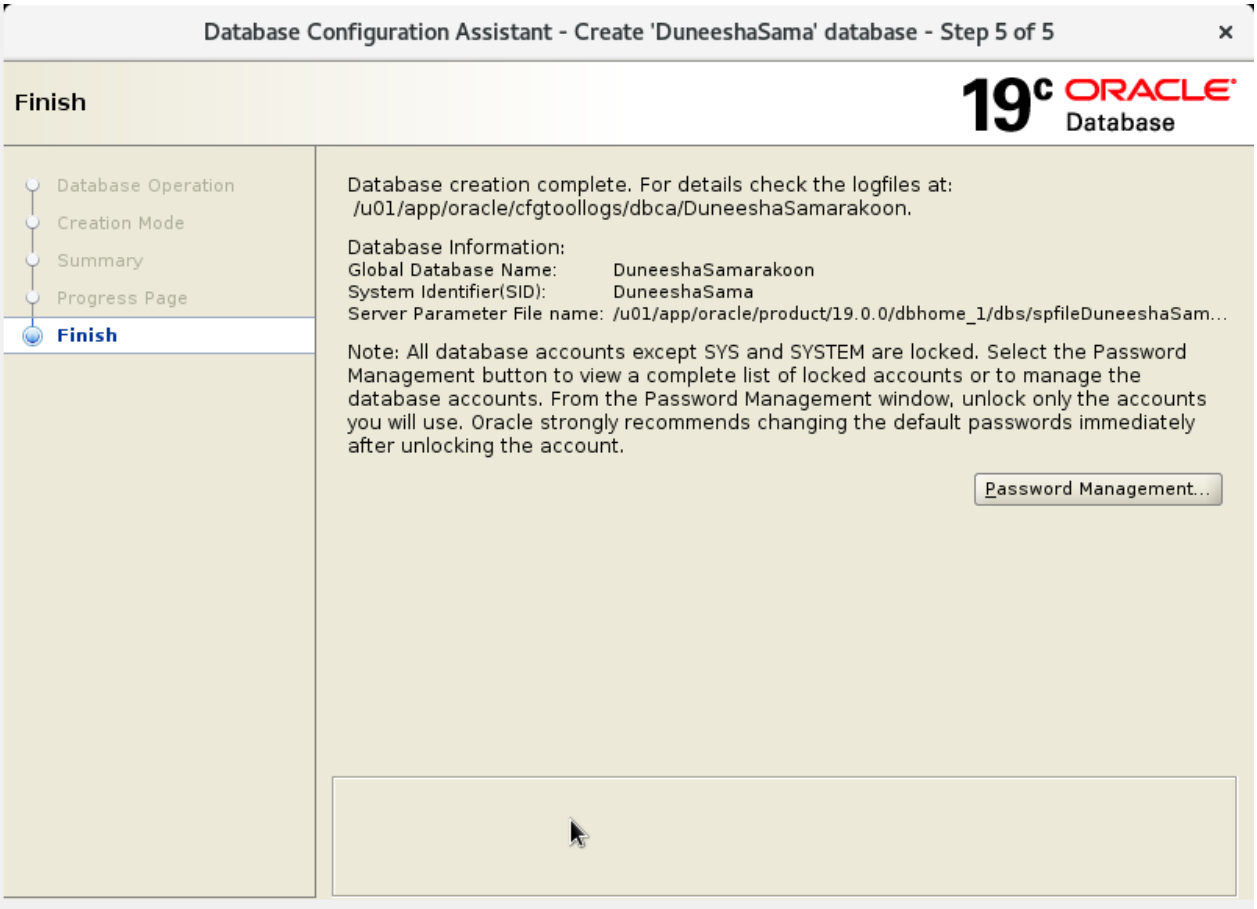


    b. Step 2 – Enter your name as the Global database name and set an administrative password of your choice with capital, simple letters along with numbers. Check the create as container database to create a PDB and set PDB name as PDBDASS.



c. Step 3 – Check the summary and click finish to create the database.

d. Step 4 and 5 – The database creation will take throughout step 4 and at after the creation is completed step 5 will display the summary of the database creation with the system identifier (SID).



4. Next, we connect the newly created instance. If the connection status is displayed as “connected to an idle instance” this means that the instance is currently in shutdown state and needs to be startup. In this case just use the command “startup” to start the instance.

```
[oracle@duneesha ~]$ export ORACLE_SID=DuneeshaSama
[oracle@duneesha ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 22 08:55:08 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1526723568 bytes
Fixed Size                  9135088 bytes
Variable Size             889192448 bytes
Database Buffers          620756992 bytes
Redo Buffers                7639040 bytes
Database mounted.
Database opened.
SQL>
```

5. After connecting the successfully created CDB, it specifies the server instance as below.

```
[oracle@duneesha ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 22 08:42:25 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

6. Check the permissions to check read and write access are allowed and the log mode of the database created to ensure that it is created in non-archive mode.

```
SQL> select name , open_Mode from v$database;

NAME      OPEN_MODE
--------- --------------------
DUNEESHA  READ WRITE
```

```
SQL> select name , log_Mode from v$database;

NAME      LOG_MODE
--------- ------------
DUNEESHA  NOARCHIVELOG
```

7. Check information related to the PDB created.

```
SQL> select pdb_id,pdb_name from cdb_pdbs;

    PDB_ID
----------
PDB_NAME
--------------------------------------------------------------------------------
         3
PDBDASS

         2
PDB$SEED


SQL>
```

8. Further to confirm the database is created in non-archive mode you can use the command “archive log list”.

```
SQL> archive log list;
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     5
Current log sequence           7
SQL>
```

3. Question 3 - Answer the following questions

A. The database you created in question number 2 above uses a binary-type parameter file. When a parameter file is corrupted or missing how can you recover it and start the database? Demonstrate your answer using your database.

1. start up the Database.

```
[oracle@duneesha ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 29 19:30:57 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1526723568 bytes
Fixed Size                  9135088 bytes
Variable Size             889192448 bytes
Database Buffers          620756992 bytes
Redo Buffers                7639040 bytes
Database mounted.
Database opened.
SQL>
```

2. Check open mode and the name of the Database.

```
SQL> select name,open_mode from v$database;

NAME      OPEN_MODE
--------- --------------------
DUNEESHA  READ WRITE

SQL>
```

3. Check the background dump test location.

```
SQL> show parameter background_dump_dest;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
background_dump_dest                 string      /u01/app/oracle/product/19.0.0
                                                 /dbhome_1/rdbms/log
SQL>
```

4. Shutdown the Database.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

5. Goto spfile location.

```
SQL> !
[oracle@duneesha ~]$ cd $ORACLE_HOME/dbs
[oracle@duneesha dbs]$ ls
hc_DuneeshaSama.dat  lkDUNEESHA            orapwDuneeshaSama
init.ora             lkDUNEESHASAMARAKOON  spfileDuneeshaSama.ora
[oracle@duneesha dbs]$
```

6. Here we have to remove spfile. According to my case (spfileDuneeshaSama.ora) is spfile.

```
[oracle@duneesha dbs]$ rm spfileDuneeshaSama.ora
[oracle@duneesha dbs]$ ls
hc_DuneeshaSama.dat  lkDUNEESHA            orapwDuneeshaSama
init.ora             lkDUNEESHASAMARAKOON
[oracle@duneesha dbs]$
```

7. now we are going to login as sysdba from dbs location.

```
[oracle@duneesha dbs]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 29 21:12:47 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file '/u01/app/oracle/product/19.0.0/dbhome_
1/dbs/initDuneeshaSama.ora'
SQL>
```

8. now we are going to find and open the trace files.

```
dbhome_1
[oracle@duneesha 19.0.0]$ cd ..
[oracle@duneesha product]$ ls
19.0.0
[oracle@duneesha product]$ cd ..
[oracle@duneesha oracle]$ ls
admin  cfgtoollogs  diag                oradata
audit  checkpoints  fast_recovery_area  product
[oracle@duneesha oracle]$ /diag
bash: /diag: No such file or directory
[oracle@duneesha oracle]$ cd /diag
bash: cd: /diag: No such file or directory
[oracle@duneesha oracle]$ cd diag
[oracle@duneesha diag]$ cd rdbms
[oracle@duneesha rdbms]$ ls
duneeshasamarakoon
[oracle@duneesha rdbms]$ cd duneeshasamarakoon
[oracle@duneesha duneeshasamarakoon]$ ls
DuneeshaSama  i_1.mif
[oracle@duneesha duneeshasamarakoon]$ cd DuneeshaSama
[oracle@duneesha DuneeshaSama]$ ls
alert  hm        incpkg  lck  metadata       metadata_pv  sweep
cdump  incident  ir      log  metadata_dgif  stage        trace
[oracle@duneesha DuneeshaSama]$ cd trace
```

9. See that logs in trace.

```
[oracle@duneesha trace]$ cat alert_DuneeshaSama.log
2023-09-22T07:31:20.809027-04:00
Starting ORACLE instance (normal) (OS id: 10975)
2023-09-22T07:31:20.871795-04:00
************************************************
 /dev/shm will be used for creating SGA
Large pages will not be used. Only standard 4K pages will be used
************************************************
2023-09-22T07:31:20.880654-04:00
************************************************************************
2023-09-22T07:31:20.880811-04:00
Dump of system resources acquired for SHARED GLOBAL AREA (SGA)

2023-09-22T07:31:20.881073-04:00
 Domain name: user.slice/user-54321.slice/user@54321.service
2023-09-22T07:31:20.881191-04:00
 Per process system memlock (soft) limit = 64K
2023-09-22T07:31:20.881383-04:00
 Expected per process system memlock (soft) limit to lock
 instance MAX SHARED GLOBAL AREA (SGA) into memory: 1456M
2023-09-22T07:31:20.881614-04:00
```

10. since we don't have pfile and spfile so we 'll get back into the alert log file select system parameter with non-default values. So will find the system parameters with non-default values which was used in last successful database login.

```
System parameters with non-default values:
  processes                = 300
  memory_target            = 1456M
  control_files            = "/u01/app/oracle/oradata/DUNEESHASAMARAKOON/controlfile/o1
_mf_ljtyzgtd_.ctl"
  control_files            = "/u01/app/oracle/fast_recovery_area/DUNEESHASAMARAKOON/con
trolfile/o1_mf_ljtyzhml_.ctl"
  db_block_size            = 8192
  compatible               = "19.0.0"
  db_create_file_dest      = "/u01/app/oracle/oradata"
  db_recovery_file_dest    = "/u01/app/oracle/fast_recovery_area"
  db_recovery_file_dest_size= 12732M
  undo_tablespace          = "UNDOTBS1"
  remote_login_passwordfile= "EXCLUSIVE"
  dispatchers              = "(PROTOCOL=TCP) (SERVICE=DuneeshaSamaXDB)"
  audit_file_dest          = "/u01/app/oracle/admin/DuneeshaSamarakoon/adump"
  audit_trail              = "DB"
  db_name                  = "Duneesha"
  db_unique_name           = "DuneeshaSamarakoon"
  open_cursors             = 300
  diagnostic_dest          = "/u01/app/oracle"
  enable_pluggable_database= TRUE
2023-09-22T08:55:23.315628-04:00
```

11. From trace location into go dbs location.

```
[oracle@duneesha trace]$ cd $ORACLE_HOME/dbs
[oracle@duneesha dbs]$ ls
hc_DuneeshaSama.dat  init.ora  lkDUNEESHA  lkDUNEESHASAMARAKOON  orapwDuneeshaSama
[oracle@duneesha dbs]$
```

12. After moving to dbs location create a new file in the name of 'initDuneeshaSama.ora'.

```
[oracle@duneesha dbs]$ vi initDuneeshaSama.ora
```

13. Past the non-default values found from trace file.

```
processes                = 300
  memory_target            = 1456M
  control_files            = "/u01/app/oracle/oradata/DUNEESHASAMARAKOON/controlfile/o1
_mf_ljtyzgtd_.ctl"
  control_files            = "/u01/app/oracle/fast_recovery_area/DUNEESHASAMARAKOON/con
trolfile/o1_mf_ljtyzhml_.ctl"
  db_block_size            = 8192
  compatible               = "19.0.0"
  db_create_file_dest      = "/u01/app/oracle/oradata"
  db_recovery_file_dest    = "/u01/app/oracle/fast_recovery_area"
  db_recovery_file_dest_size= 12732M
  undo_tablespace          = "UNDOTBS1"
  remote_login_passwordfile= "EXCLUSIVE"
  dispatchers              = "(PROTOCOL=TCP) (SERVICE=DuneeshaSamaXDB)"
  audit_file_dest          = "/u01/app/oracle/admin/DuneeshaSamarakoon/adump"
  audit_trail              = "DB"
  db_name                  = "Duneesha"
  db_unique_name           = "DuneeshaSamarakoon"
  open_cursors             = 300
  diagnostic_dest          = "/u01/app/oracle"
  enable_pluggable_database= TRUE

~
~
~
~
~
```

14. Once we done to create a pfile, login from dbs location to log into database using default values in 'initDuneeshaSama.ora' file

```
[oracle@duneesha dbs]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 29 21:42:01 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1526723568 bytes
Fixed Size                  9135088 bytes
Variable Size             889192448 bytes
Database Buffers          620756992 bytes
Redo Buffers                7639040 bytes
Database mounted.
Database opened.
SQL>
```

15. check the name and mode of the database.

```
SQL> select name,open_mode from v$database;

NAME      OPEN_MODE
--------- --------------------
DUNEESHA  READ WRITE

SQL>
```

16. Create spfile using pfile.

```
SQL> create spfile from pfile;

File created.

SQL>
```

17. check the parameter by using which file database parameter instance is working.

```
SQL> show parameter spfile

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
spfile                               string
SQL>
```

18. we found our database instance is working. If go to the spfile location we can find the spfile in list which was deleted before. (spfileDuneeshaSama.ora)

```
SQL> !
[oracle@duneesha dbs]$ cd $ORACLE_HOME/dbs
[oracle@duneesha dbs]$ ls
hc_DuneeshaSama.dat   init.ora    lkDUNEESHASAMARAKOON  spfileDuneeshaSama.ora
initDuneeshaSama.ora  lkDUNEESHA  orapwDuneeshaSama
[oracle@duneesha dbs]$
```

B. Increase the processes parameter by 10.

1. Show parameter sessions.

```
SQL> show parameter sessions;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
java_max_sessionspace_size           integer     0
java_soft_sessionspace_limit         integer     0
license_max_sessions                 integer     0
license_sessions_warning             integer     0
sessions                             integer     472
shared_server_sessions               integer
SQL>
```

2. Show parameter processes.

```
SQL> show parameter processes;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
aq_tm_processes                      integer     1
db_writer_processes                  integer     1
gcs_server_processes                 integer     0
global_txn_processes                 integer     1
job_queue_processes                  integer     40
log_archive_max_processes            integer     4
processes                            integer     300
SQL>
```

3. Show parameter transactions.

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
transactions                         integer     519
transactions_per_rollback_segment    integer     5
SQL>
```

4. Increase the parameter by 10.

```
SQL> alter system set processes=310 scope=spfile;

System altered.

SQL>
```

5. Shutdown abort and startup.

```
SQL> shutdown abort
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 1526723592 bytes
Fixed Size                  9135112 bytes
Variable Size             889192448 bytes
Database Buffers          620756992 bytes
Redo Buffers                7639040 bytes
Database mounted.
Database opened.
SQL> show parameter processes;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
aq_tm_processes                      integer     1
db_writer_processes                  integer     1
gcs_server_processes                 integer     0
global_txn_processes                 integer     1
job_queue_processes                  integer     40
log_archive_max_processes            integer     4
processes                            integer     310
SQL>
```

So, at last the in-process parameter the values is changed to 310 which was 300 before.

C. Create a new listener called LISTNER2 with port number 1522.

1. To add listener,

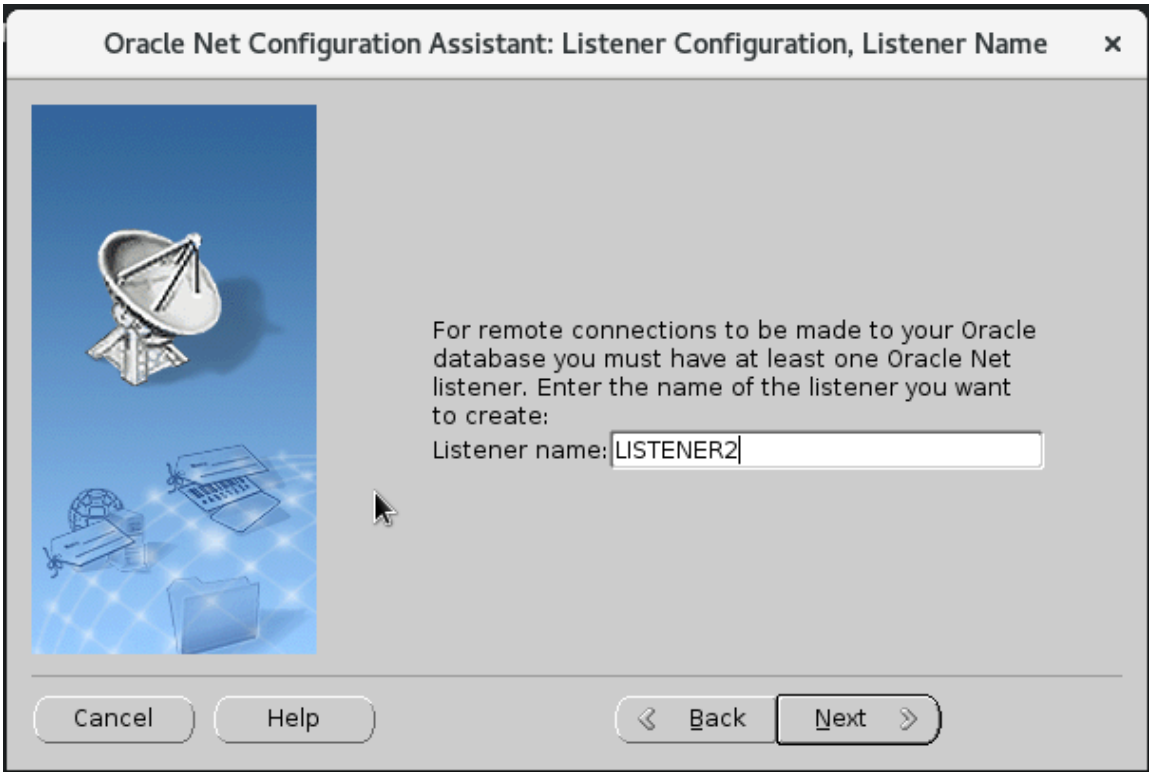
```
[oracle@duneesha ~]$ netca
```

2. Follow the wizard.

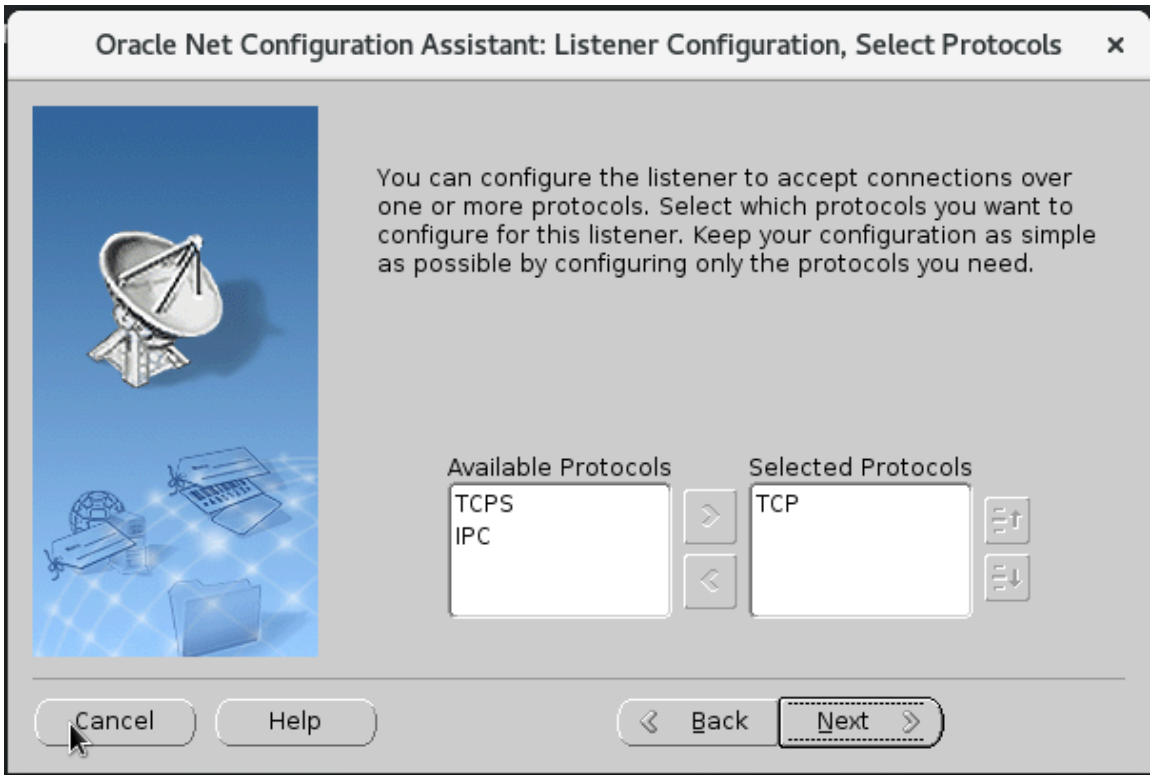
a. Start the setup listener configuration.

b. Adding the listener.

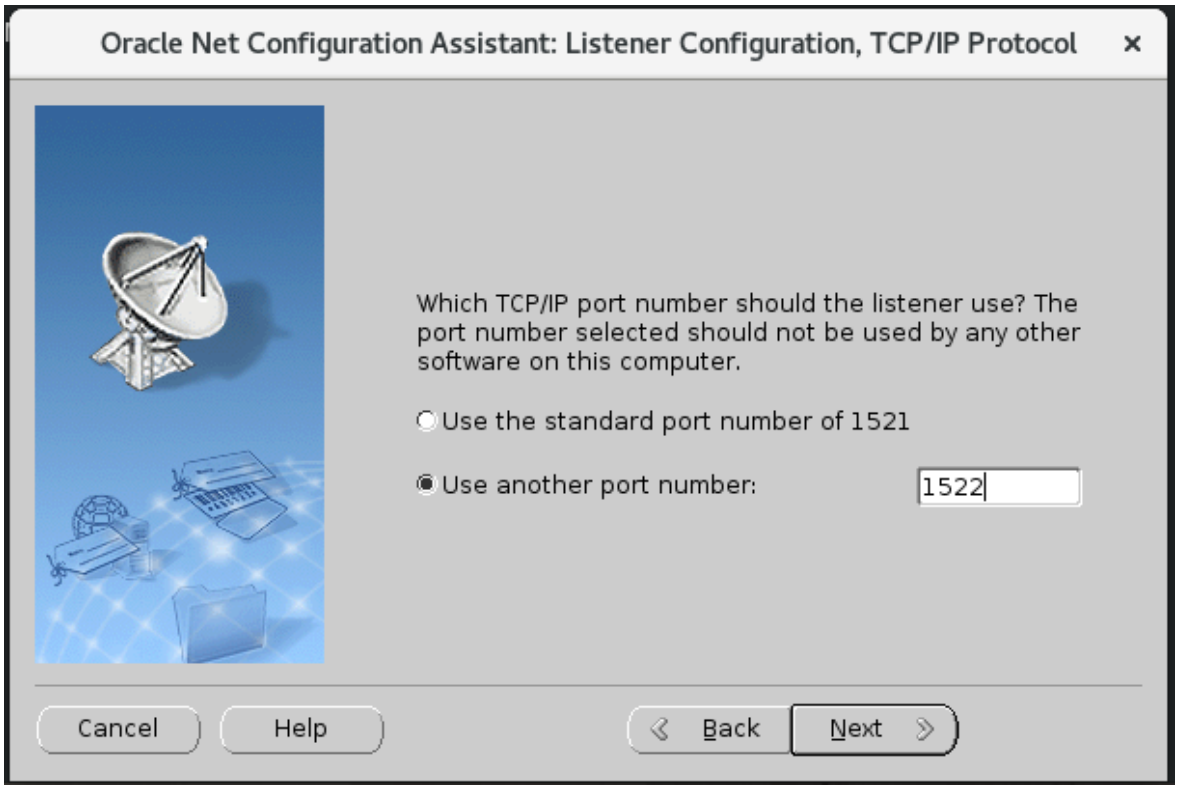
c. Creating listener called LISTNER2.



d. Accepted protocols.

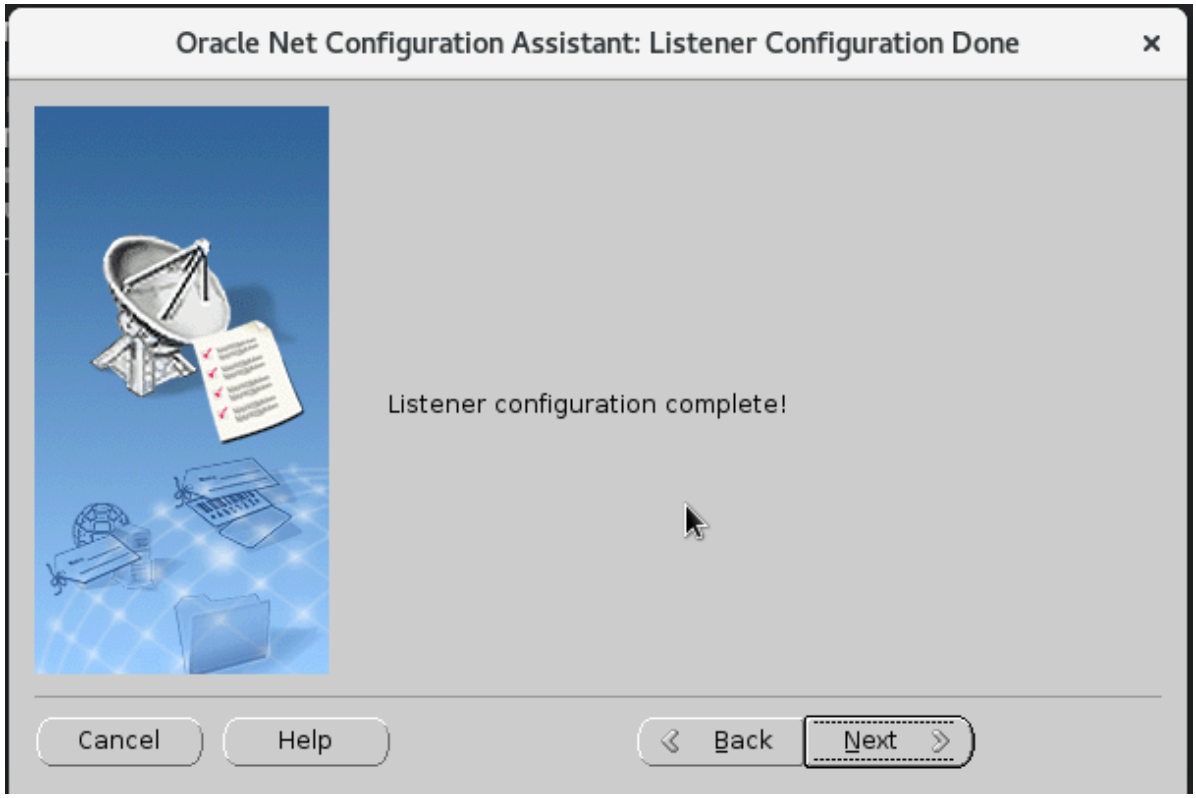


e. Adding Port number.



f. Select ‘NO’ option here.

g. Listener configuration done.



```
[oracle@duneesha ~]$ netca

Oracle Net Services Configuration:
Configuring Listener:LISTENER2
Listener configuration complete.
Oracle Net Listener Startup:
    Running Listener Control:
      /u01/app/oracle/product/19.0.0/dbhome_1/bin/lsnrctl start LISTENER2
    Listener Control complete.
    Listener started successfully.
Oracle Net Services configuration successful. The exit code is 0
[oracle@duneesha ~]$
```

3. check the lister.

```
[oracle@duneesha ~]$ ss -napt | grep 1522
LISTEN 0      128                *:1522             *:*    users:(("tnslsnr",pid
=7408,fd=8))
[oracle@duneesha ~]$
```

Question 4 - Create a new tablespace EXAMPLE1 of size 5 MB with one data file. After that, expand the tablespace size to 8MB by adding a new data file

1. Startup the database.

```
[oracle@duneesha ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Sat Sep 30 00:02:20 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

2. Create tablespace with 5MB space.

```
SQL> create tablespace EXAMPLE1 datafile '/u01/app/oracle/oradata/DUNEESHASAMARA
KOON/EXAMPLE1_1.dbf' size 5M;

Tablespace created.

SQL>
```

3. Check table space.

```
SQL> select tablespace_name, bytes / 1024 / 1024 MB from dba_free_space;

TABLESPACE_NAME                        MB
------------------------------ ----------
SYSTEM                              .4375
SYSAUX                             35.0625
UNDOTBS1                                3
USERS                              2.3125
EXAMPLE1                                4

SQL>
```

4. expand the tablespace size to 8MB by adding 3MB a new data file

```
SQL> alter tablespace EXAMPLE1 add datafile
  2  '/u01/app/oracle/oradata/DUNEESHASAMARAKOON/EXAMPLE1_2.dbf' size 3M
  3  autoextend on;

Tablespace altered.

SQL>
```

5. Check table space.

```
SQL> select tablespace_name, bytes / 1024 / 1024 MB from dba_free_space;

TABLESPACE_NAME                        MB
------------------------------ ----------
SYSTEM                              .4375
SYSAUX                           35.0625
UNDOTBS1                               3
USERS                             2.3125
EXAMPLE1                               2
EXAMPLE1                               4

6 rows selected.

SQL>
```

Question 5 - Write a report about database security features and their use in Oracle 19c or 21c. The word count is 300 words.

A comprehensive set of security capabilities are available in Oracle Database 19c and 21c to address the difficulties of contemporary cybersecurity. These features are intended to protect against threats and safeguard data. Here is a quick summary of several significant database security features in Oracle 19c and 21c and how they are used:

TDE: Transparent Data Encryption TDE encrypts data while it is at rest, guaranteeing that secret data kept in the database is kept that way even if the physical storage media are hacked. Compliance with data privacy laws depends on it.

To prevent unauthorized access to sensitive data, data redaction enables sensitive material to be displayed in query results partially hidden. It assists in preventing the use of applications to expose sensitive data.

Even from users with DBA capabilities, Database Vault limits access to highly privileged accounts. By limiting who may carry out privileged acts, it offers an extra degree of security.

Data transferred between the client and the database server can be secured with network encryption and strong authentication from Oracle Advanced Security, preventing data in transit from being intercepted.

The combination of Audit Vault and Database Firewall enables thorough auditing, monitoring, and firewall capabilities to keep track of database activity and deny unauthorized access. It aids businesses in meeting audit standards.

The Database Security Assessment Tool (DBSAT) checks databases for possible security flaws, configuration errors, and exposed sensitive data. It aids in spotting security flaws and proactively fixing them.

Real-time database traffic monitoring by Oracle Database Firewall allows for the detection and prevention of dangerous actions like SQL Injection. It shields the database against attackers.

Oracle offers a variety of authentication procedures, such as multi-factor authentication and password complexity guidelines, to make sure that only authorized users may access the database. Granular privileges are granted through techniques for fine-grained access control.

Database Data updates, SQL statements, and login attempts are just a few examples of the actions that auditing records and tracks within the database. It supports forensic analysis, compliance reporting, and user behavior monitoring.

Database Vault Realms give businesses the ability to compartmentalize data and limit access to certain user groups, ensuring that sensitive information is only accessible to those who need it.

Oracle Database 19c and 21c's security features are essential components of a defense-in-depth strategy, providing multiple layers of protection for critical data assets. These features help organizations meet regulatory requirements, defend against evolving cyber threats, and maintain the confidentiality, integrity, and availability

of their databases. Properly configuring and utilizing these security features is crucial for safeguarding sensitive information in the Oracle Database environment.

End.