

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

# Oracle Database 19c Installation on Linux



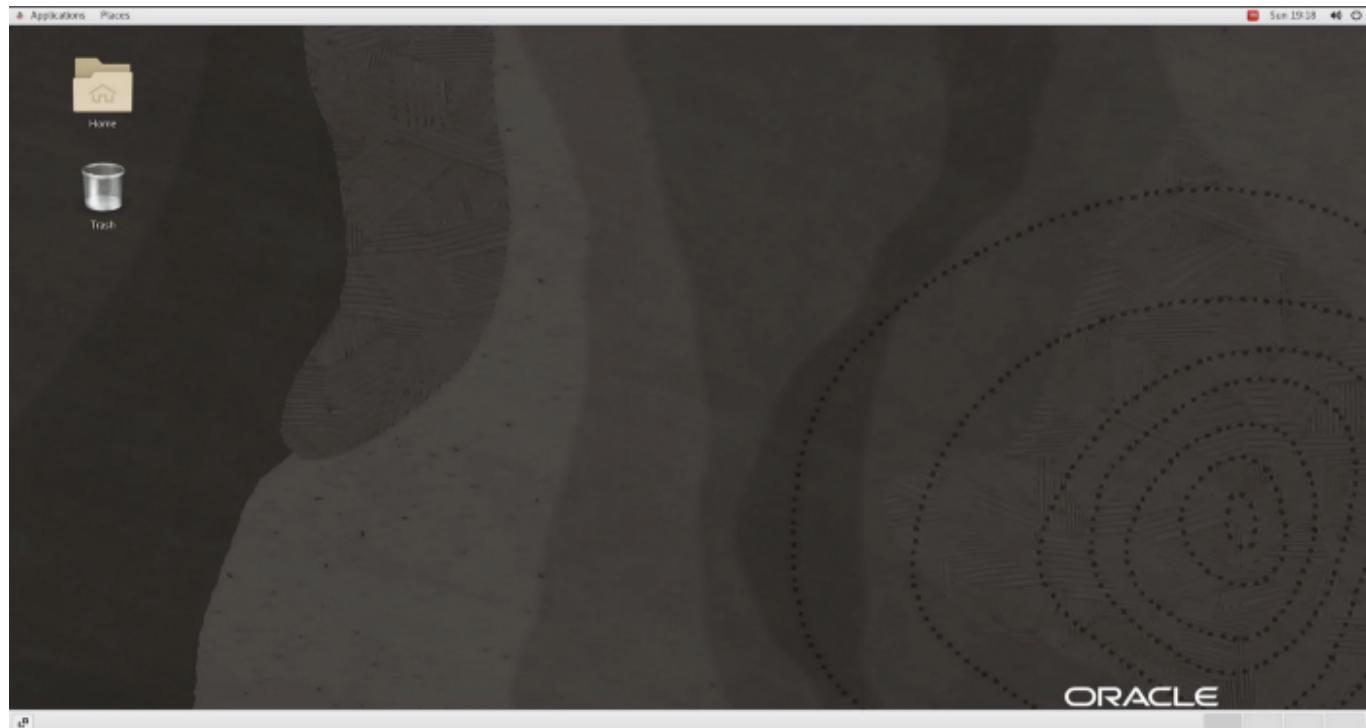
Jared Bach

[Follow](#)

Sep 13, 2021 · 16 min read



Today we are going to be installing Oracle Database 19c on Oracle Linux 7.9 in OCI. If you do not already have a working Linux VM in OCI, first complete the steps in [this blog](#) that I also wrote. If you have not already done so, log-into your Linux instance's GUI. Your screen should look like this.

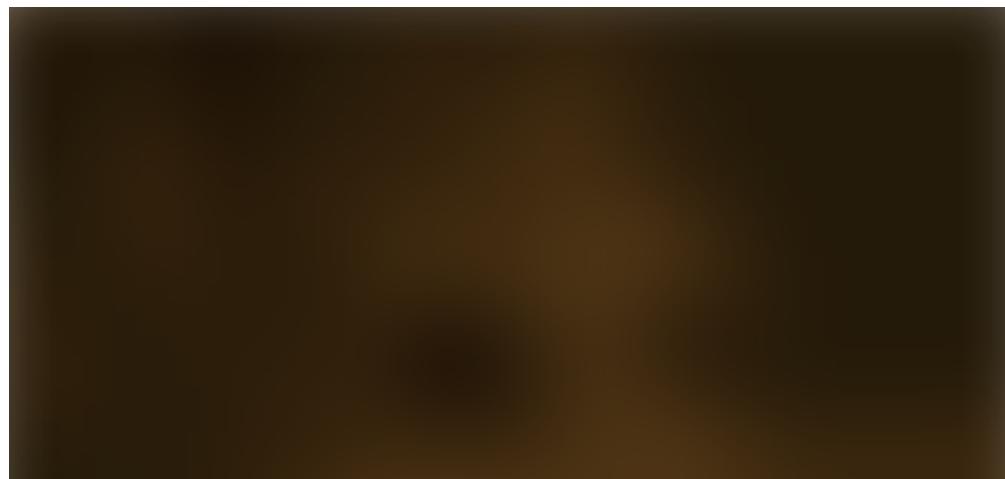


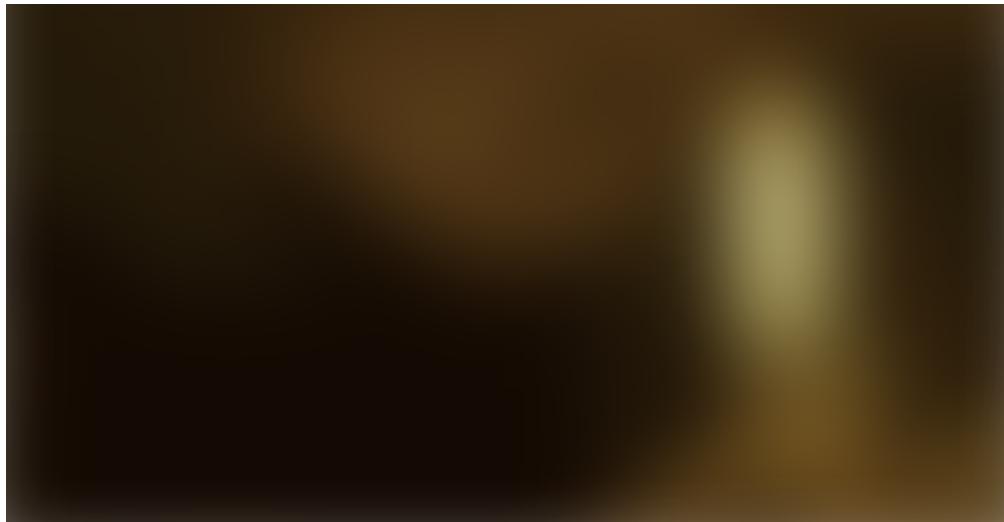
Once inside your Linux VM instance, open a new terminal window. Your terminal screen should look like this.



Next, we are going to log-in as the root user. To do this, copy and paste the following command and if prompted, enter your password.

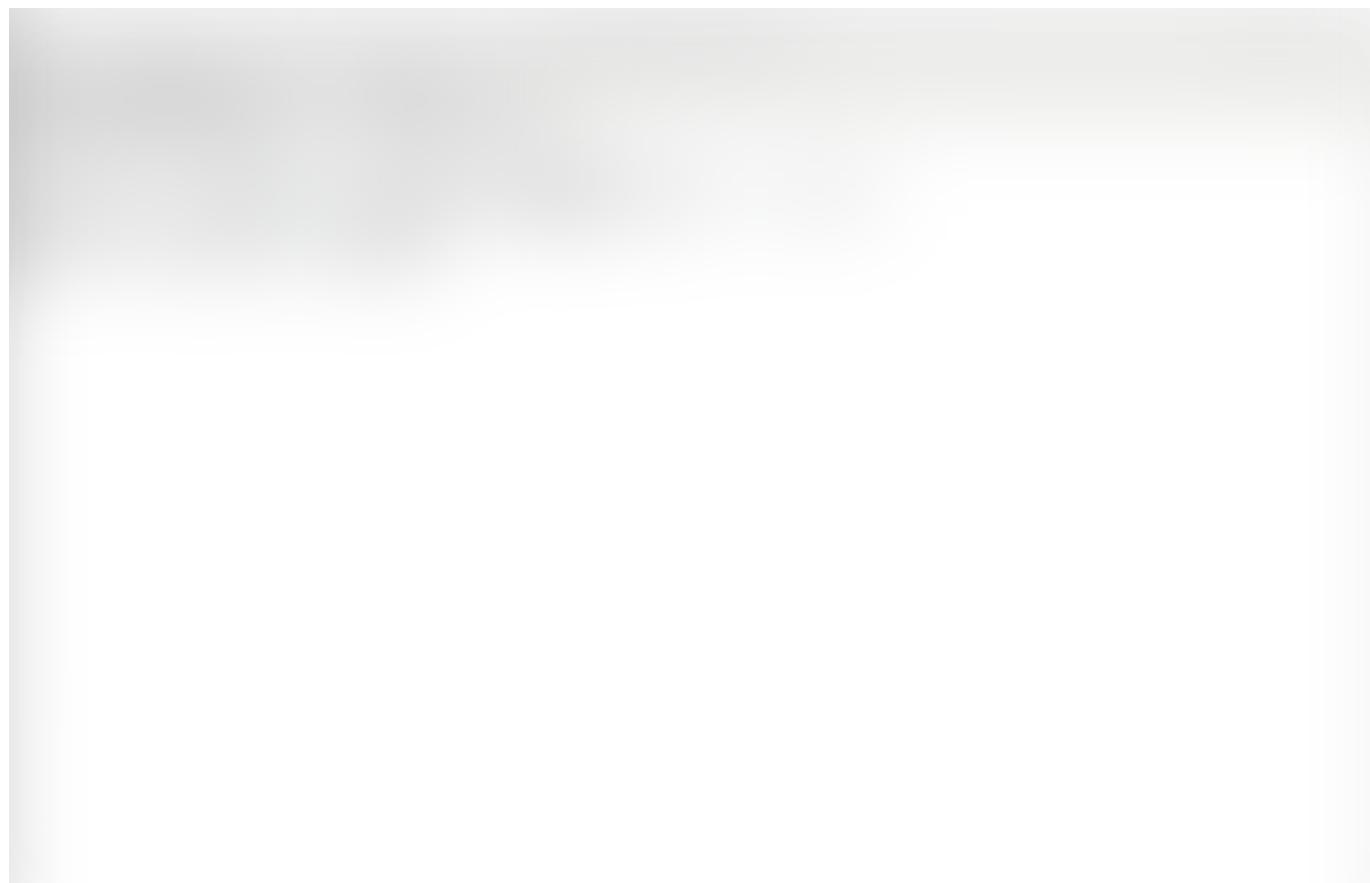
```
sudo su -
```





Not *Groot* — Root. But close enough.

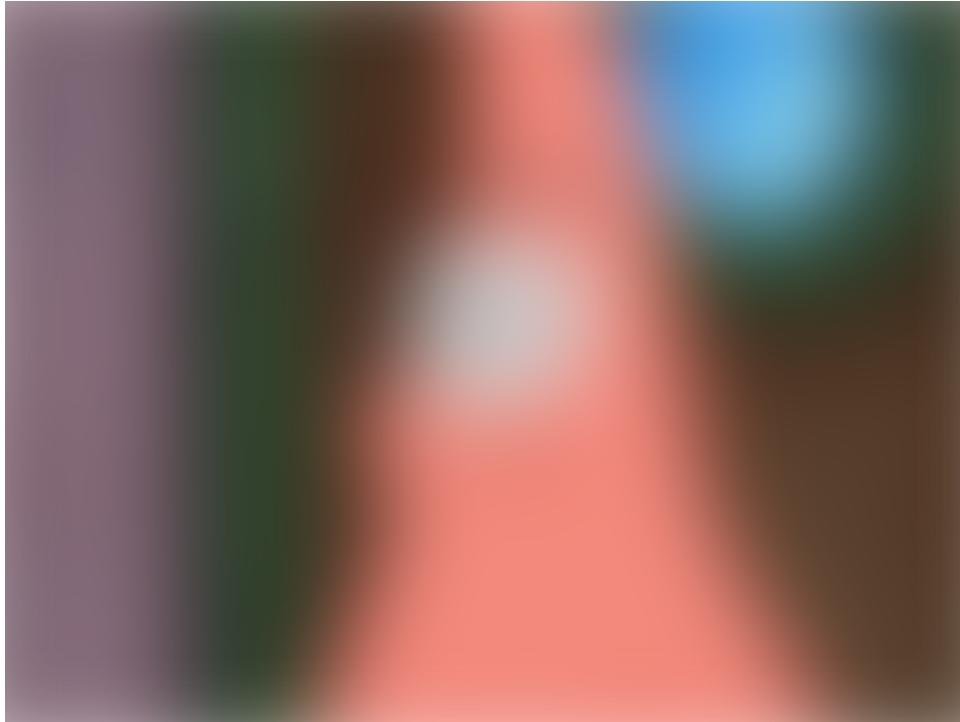
Your terminal screen should now look like this.



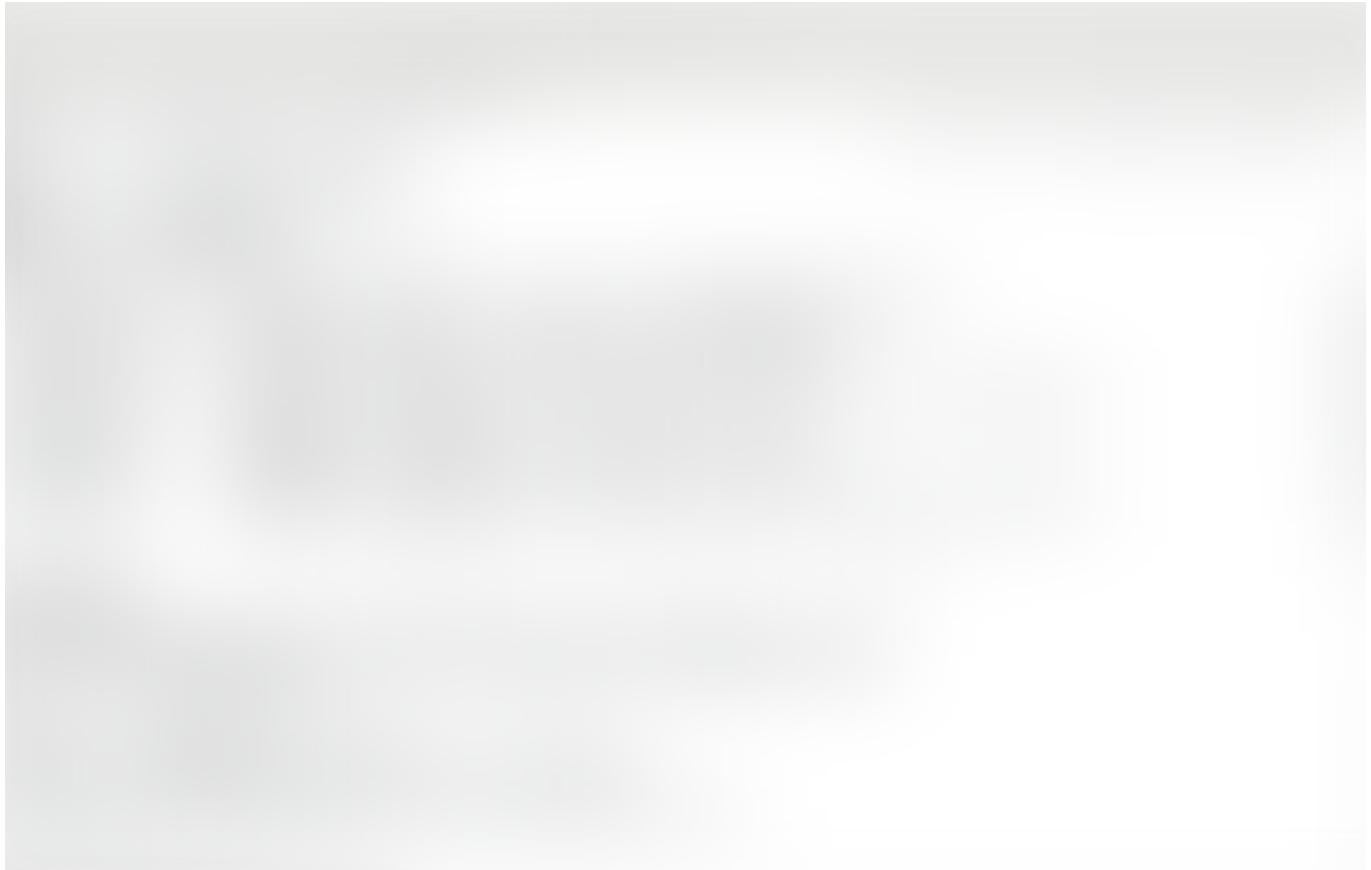
## The pre-installation

Before downloading and installing Oracle Database 19c, we need to do some preinstallations. We are going to be using the yum repository to complete all of our preinstallations. In your terminal, paste this line of code to complete the preinstallation.

```
yum install -y oracle-database-preinstall-19c
```



Once the preinstallation is completed, your terminal screen should look like this.



## Setting-up our oracle user account

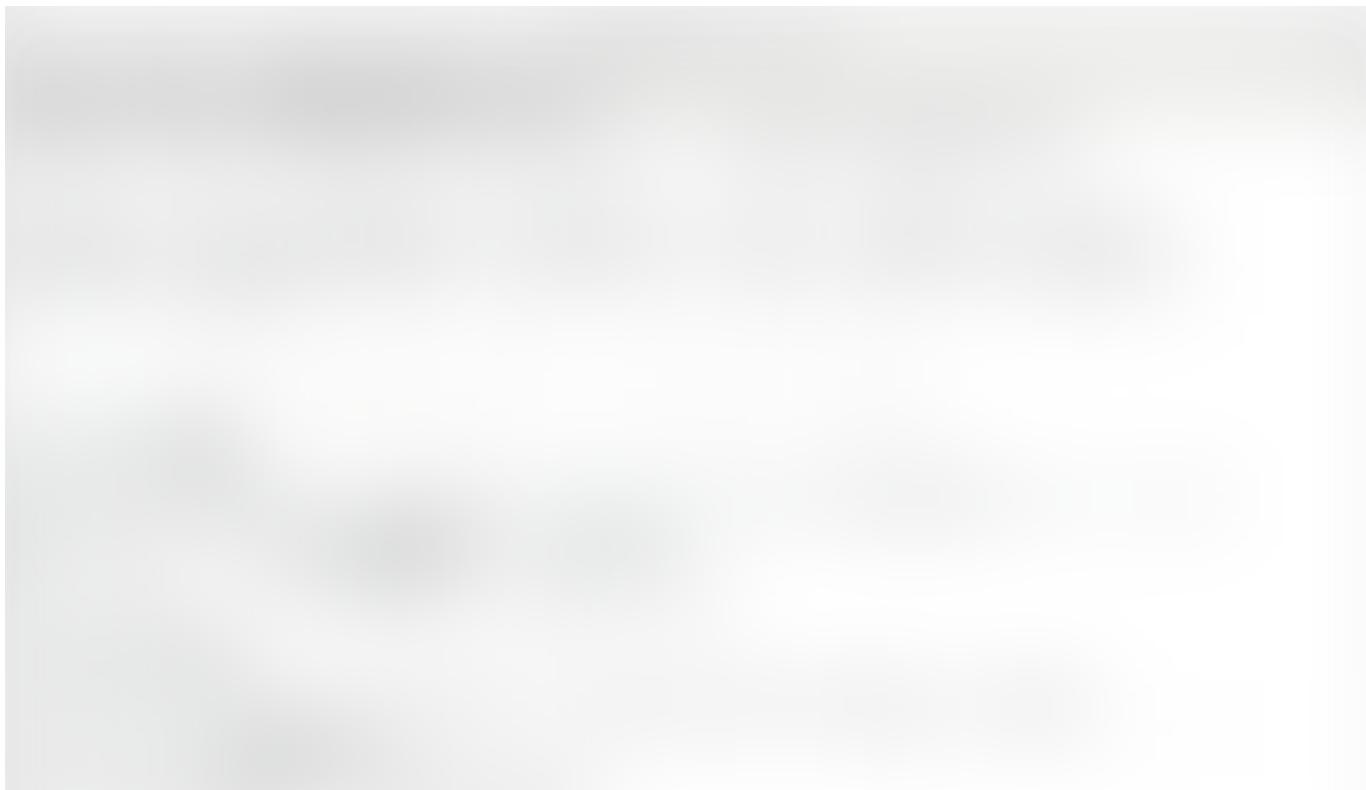
Next, we are going to set the password for the oracle user that was created in the pre-installation. We will be working out of the oracle account when we are working on our database installation. First we need to log-in as a root user. Then, we can change the password. We can do this by running the following command.

```
passwd oracle
```

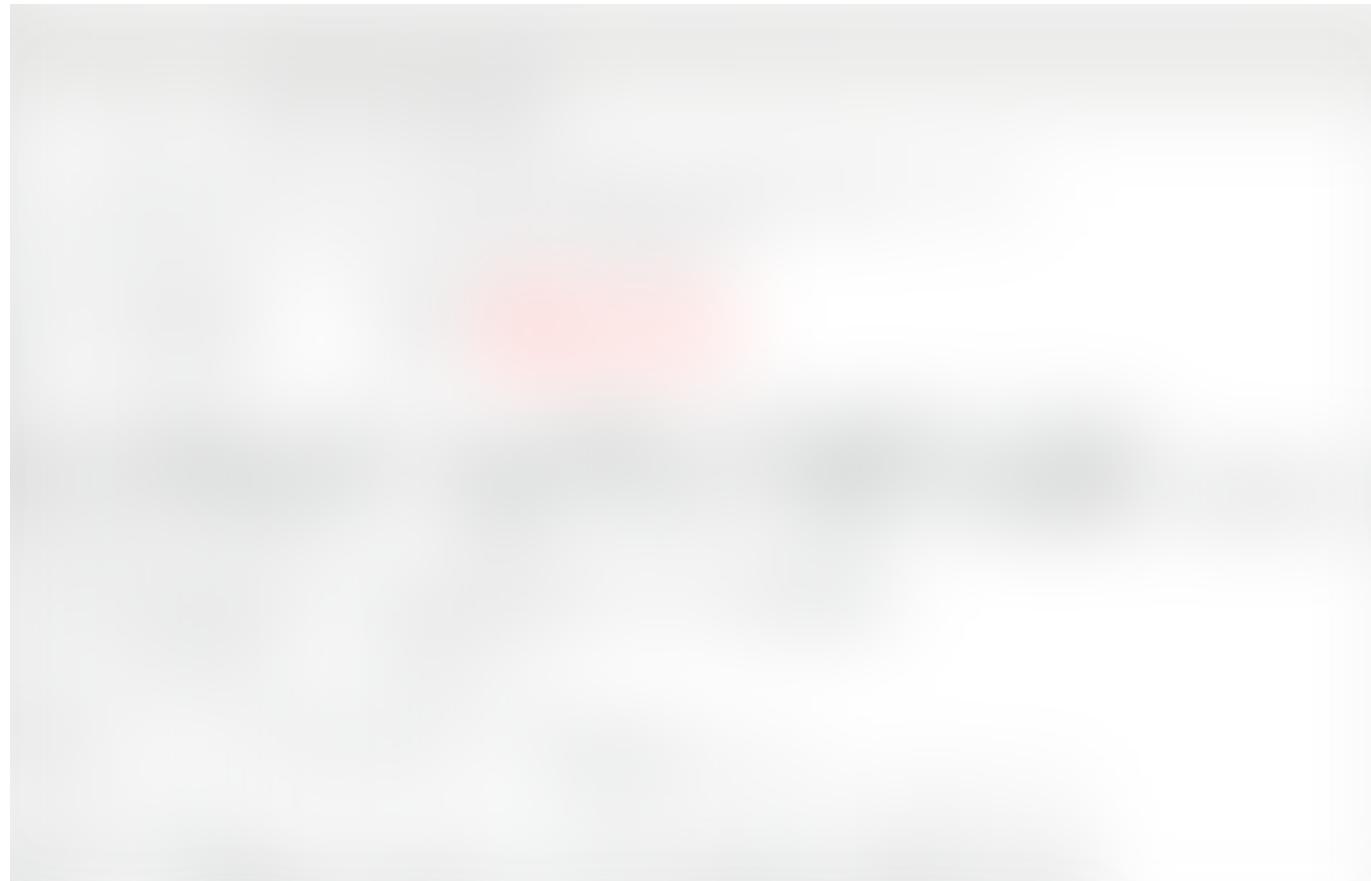
Now, we are going to give the oracle user root privileges. To do this, we need to edit the /etc/sudoers file. Never edit the sudoers file in a normal text editor. This could lead to simultaneous editing and corrupt files. We can edit the sudoers files by using the visudo command.

```
sudo visudo
```

Your terminal should look something like this.



Press Shift+G to navigate to the bottom of the document. Press the letter “i” on your keyboard to begin editing the document. You will then want to navigate up in the document to this snippet of text.

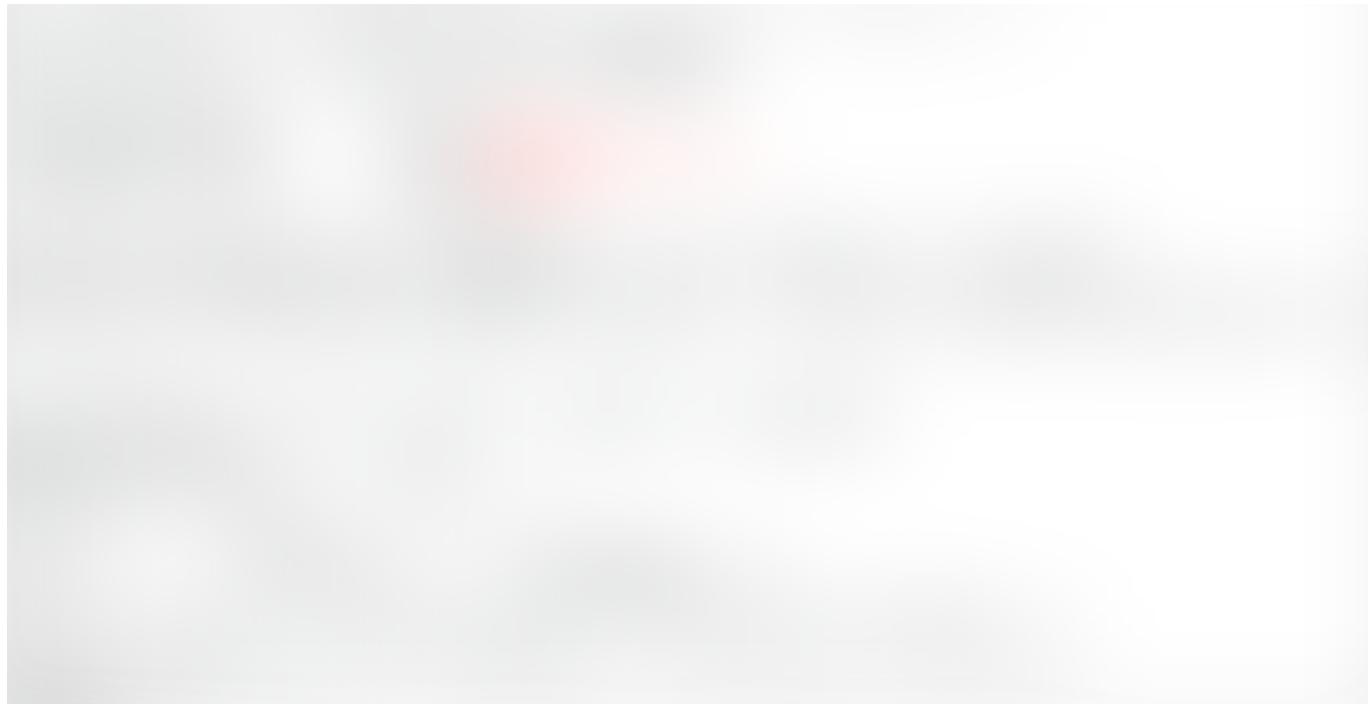


Underneath where it says root and/or jaredb, paste this snippet of text.

```
oracle ALL=(ALL)      ALL
```

This snippet of text will give oracle root privileges. Your terminal should now look something like this.





Once you complete this task, hit the ESC button, scroll down to the bottom of the document and type the following text. Then, hit enter.

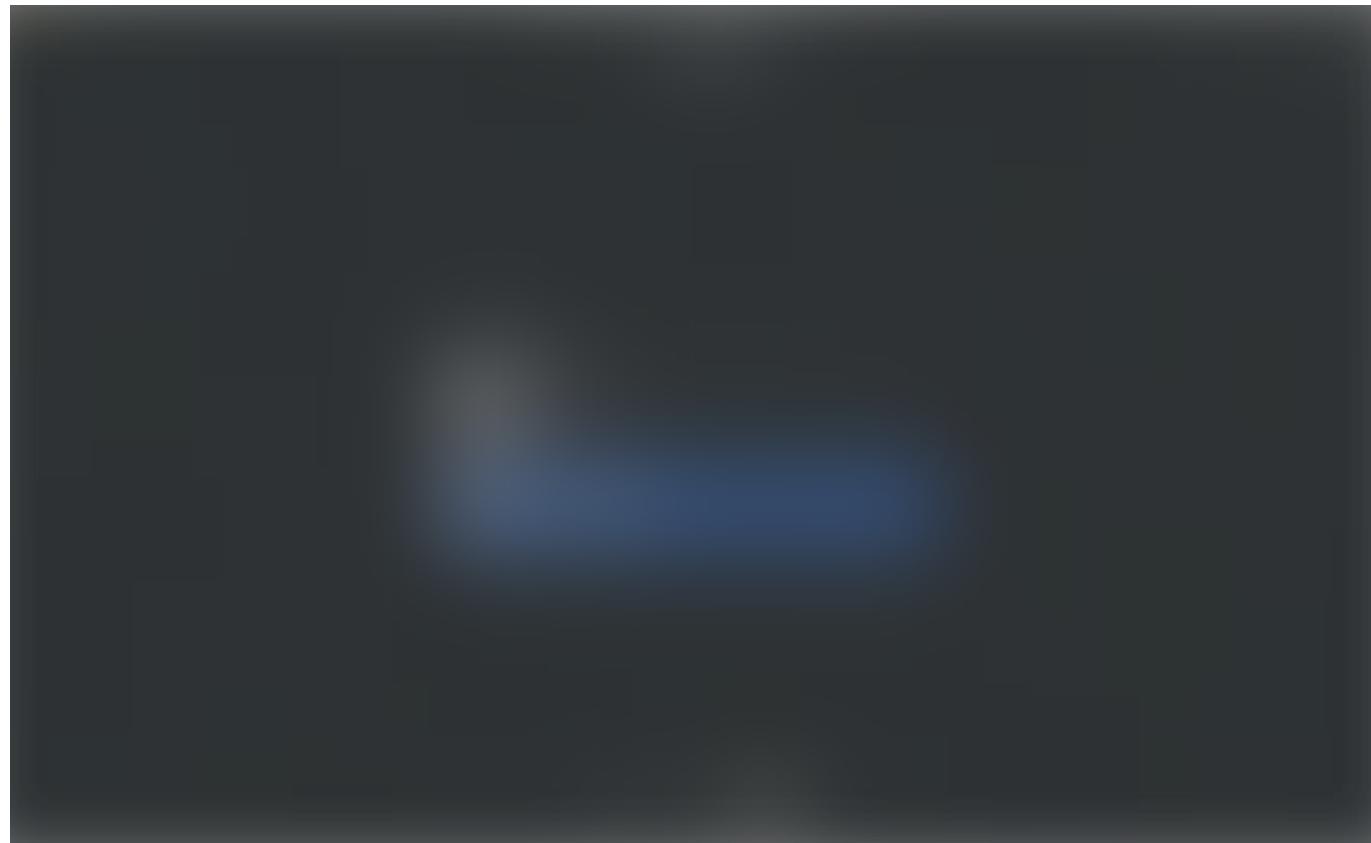
```
:wq
```

This sequence will save the file and exit the editor. We have now granted sudo privileges to oracle. Let's now switch users in our GUI, so that we are working under oracle and not jaredb. Close your terminal and press the power button in the top right-hand corner of the screen. Where it says the name of the user that you are logged-on under, select that.





Then, select “Switch User” and log-in as the oracle user that we created in the previous step.



Once you are logged into your oracle account, open a new terminal window. Your terminal screen should look like this.



## Creating an Oracle home directory

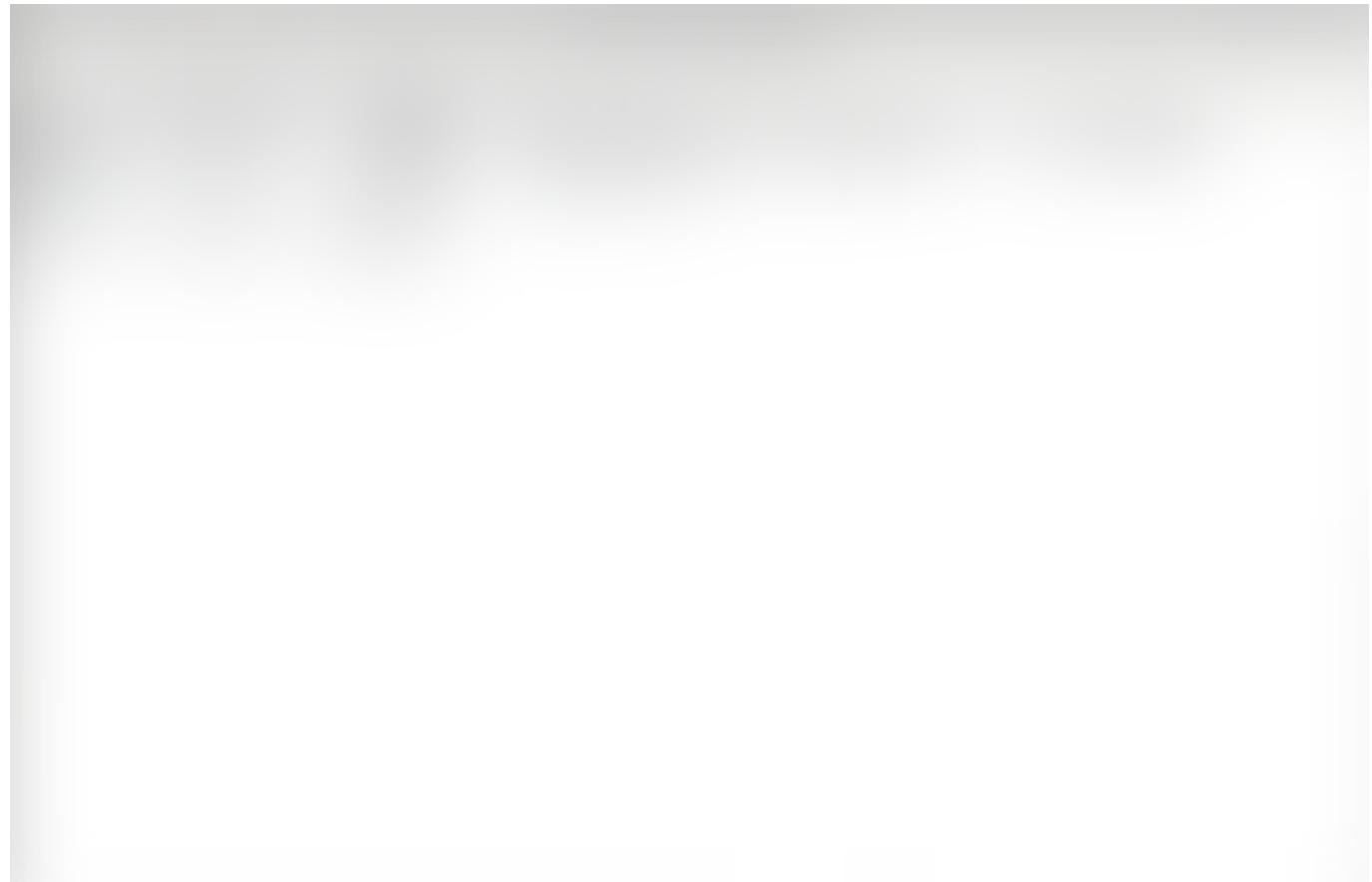
Next, we are going to be creating an Oracle home directory and giving ownership of that directory to the user, oracle. First we need to log-in as the root user. We can do this with the following command. Enter your password when prompted.

```
sudo su -
```

Your terminal screen should now look like this.

Now, we are going to be creating our home directory. Copy and paste the following code into your terminal to accomplish this.

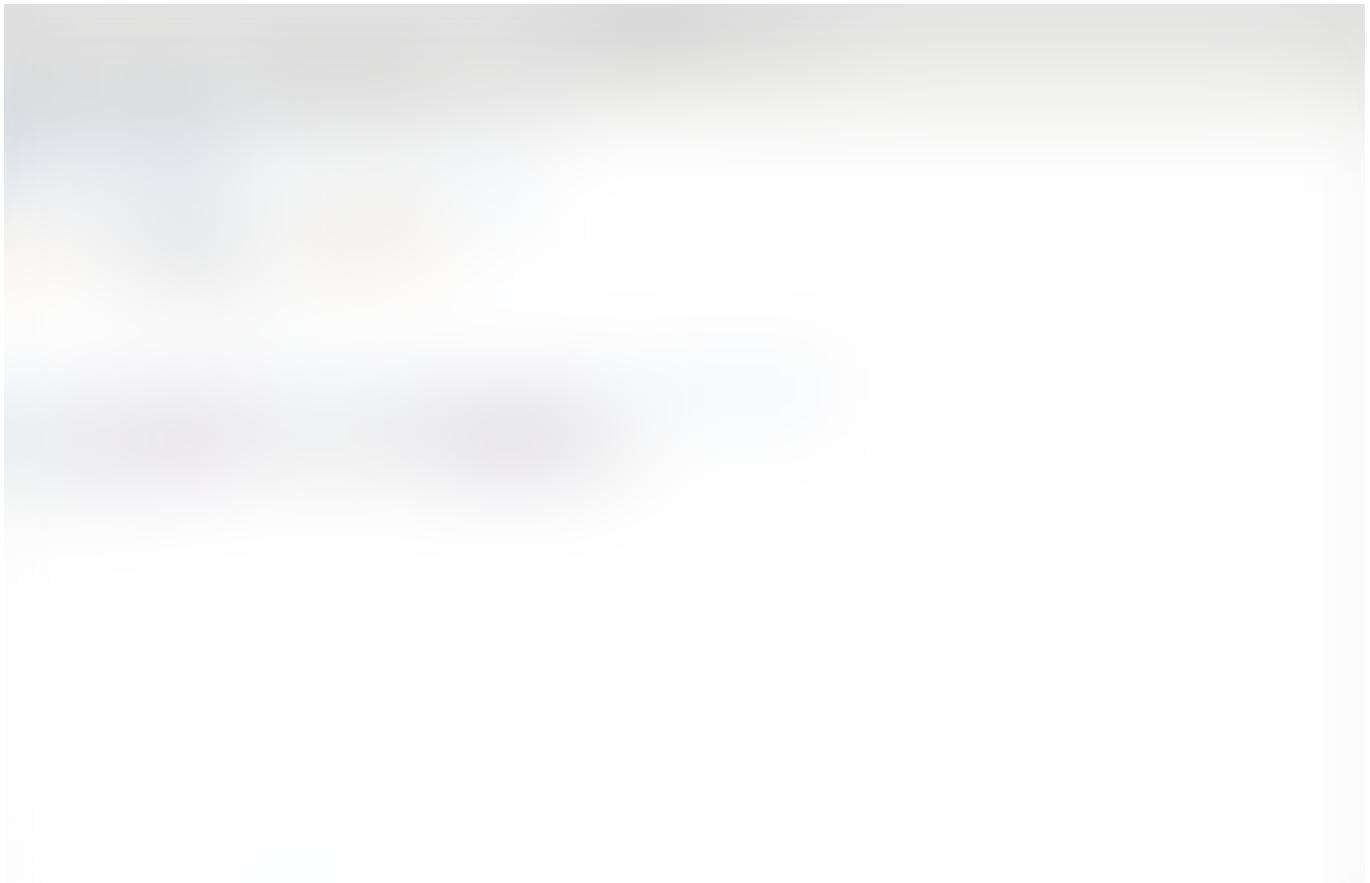
```
mkdir -p /u01/app/oracle/product/19.3/db_home  
chown -R oracle:oinstall /u01  
chmod -R 775 /u01
```



Now, we are going to be switching back to the oracle user and setting-up a bash\_profile for them. Copy and paste the code below into your terminal.

```
su - oracle  
vi .bash_profile
```

After running the above line of code in your terminal, your terminal screen should look like this.



Press Shift+G to navigate to the bottom of the document. Press the letter “i” on your keyboard to begin editing the document. Delete everything in the document by holding down the backspace button on your keyboard. After everything is deleted, your terminal screen should look like this.



Once everything in the file is deleted, copy and paste the code below into your terminal. If necessary, change the environment variables in the block of code below, so that they match your environment. If you have been following along this tutorial and have not changed any of the environment and/or directory names and variables, you should not need to change anything in the code below.

```
# .bash_profile

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

# User specific environment and startup programs

export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.3/db_home

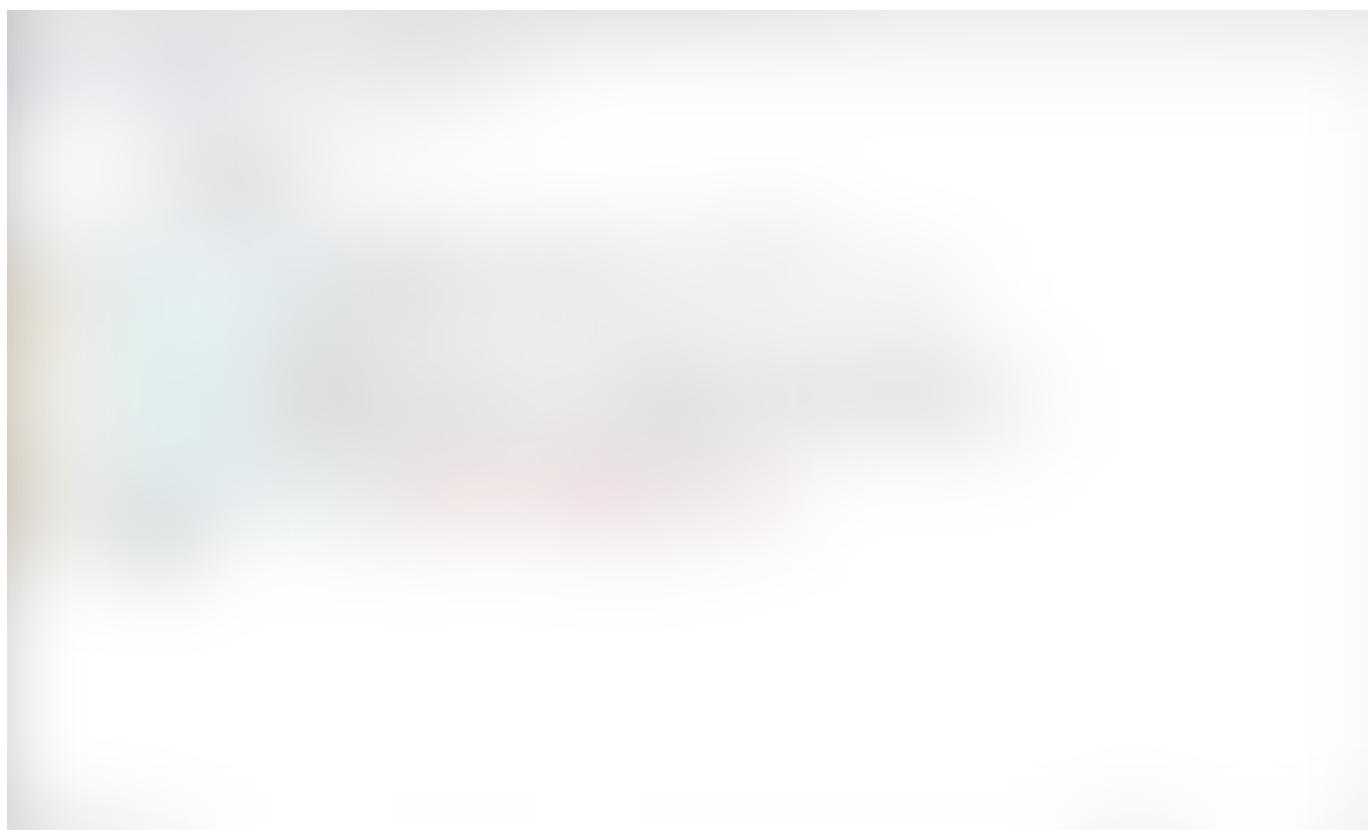
export ORACLE_SID=CDB

export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=\$ORACLE_HOME/jlib:\$ORACLE_HOME/rdbms/jlib
export NLS_LANG=american_america.al32utf8
export NLS_DATE_FORMAT="yyyy-mm-dd:hh24:mi:ss"

PATH=$PATH:$HOME/.local/bin:$ORACLE_HOME/bin

export PATH
```

After copying and pasting the block of code from above into the bash\_profile, your terminal should look like this.



When you are done, hit the ESC button, scroll down to the bottom of the document and type the following text. Then, hit enter.

```
:wq
```

This sequence will save the file and exit the editor. We now need to export the bash profile. We can do this with the following command.

```
. .bash_profile
```

Your terminal screen should now look like this.



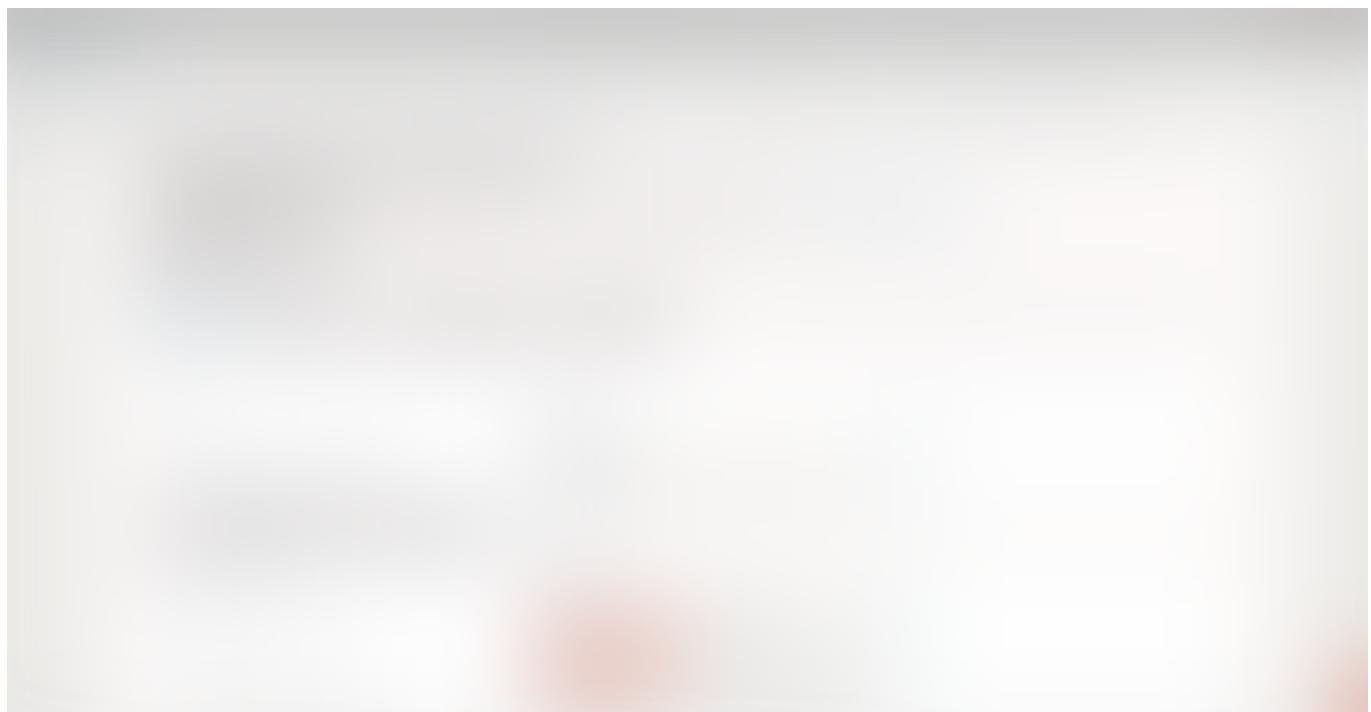


We are now going to begin installing Oracle Database 19c.

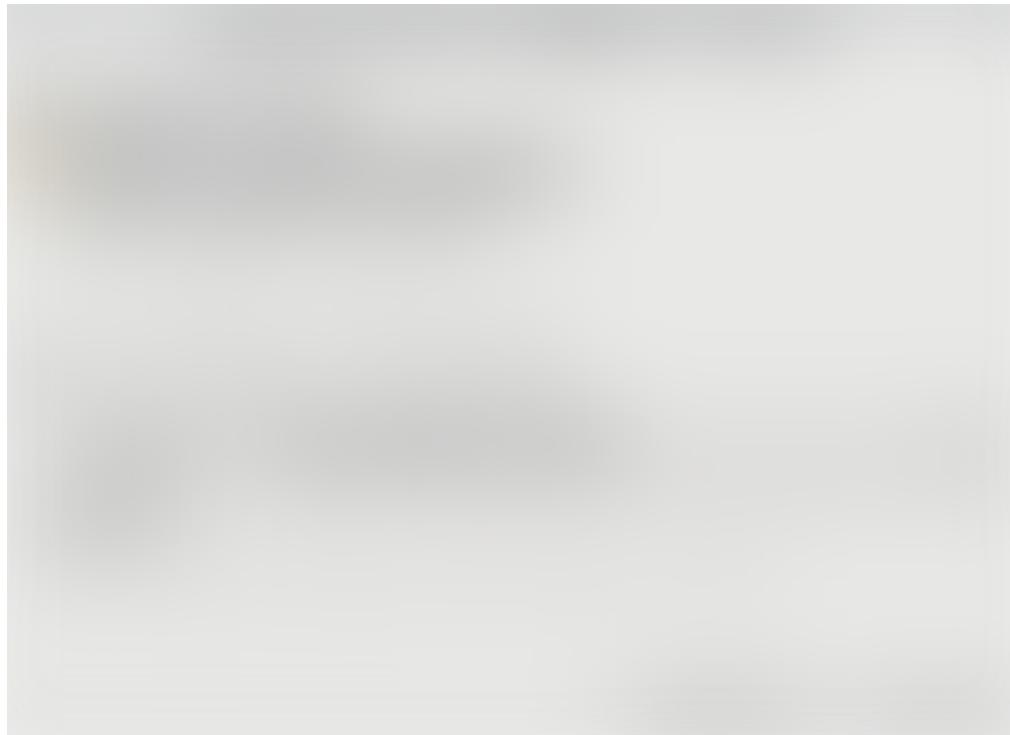
## Installing Oracle Database 19c



Navigate to [this webpage](#) inside your Linux VM to download Oracle 19c. When you arrive at the webpage, you are going to want to download the 19.3 zip file for Linux x86–64, like I have circled in red below.



Before the download can commence, you will be prompted to log into your Oracle SSO account. After successfully logging-in, you will be asked if you want to open the download file with Archive Manager (default) or if you want to save the file. Select “Save File” and then hit “OK.” Make a mental note of where this zip file was saved. It should have been saved, by default, in your downloads folder.



We are now going to copy the installation files to the ORACLE\_HOME location that we created in the previous step and unzip them. Copy and paste the code below into your terminal to begin this process. The second line of code calls the location path of where the zip file was downloaded . That is why you see /Downloads. That is also why I told you to make a mental note of where this file was being downloaded to. If your file downloaded to a different location, make sure you change this accordingly in the code. Also, if you downloaded a slightly different version or a file with a slightly different name, make sure that you are calling the right name when putting this code into your terminal.

```
cd $ORACLE_HOME  
unzip -qo /home/oracle/Downloads/LINUX.X64_193000_db_home.zip
```

This may take a few minutes, so be patient. When the unzipping completes, there will not be any output on your screen. You can check whether the unzipping was successful by entering the \$ORACLE\_HOME directory and looking at the contents of the directory with the following command.

```
ls
```

Your terminal screen should look like this.



We are now going to officially install the software. If you wish to run the installation without the GUI, copy and paste this code into your terminal.

```
# For Silent Installation
./runInstaller -ignorePrereq -waitForCompletion -silent      \
-responseFile ${ORACLE_HOME}/install/response/db_install.rsp \
oracle.install.option=INSTALL_DB_SWONLY                         \
ORACLE_HOSTNAME=${HOSTNAME}                                     \
UNIX_GROUP_NAME=oinstall                                      \
```

```
INVENTORY_LOCATION=/u01/app/oraInventory
SELECTED_LANGUAGES=en, en_GB
ORACLE_HOME=${ORACLE_HOME}
ORACLE_BASE=${ORACLE_BASE}
oracle.install.db.InstallEdition=EE
oracle.install.db.OSDBA_GROUP=dba
oracle.install.db.OSBACKUPDBA_GROUP=dba
oracle.install.db.OSDGDBA_GROUP=dba
oracle.install.db.OSKMDBA_GROUP=dba
oracle.install.db.OSRACDBA_GROUP=dba
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
```

If you wish to use the GUI installer, copy and paste this code instead into your terminal.

```
# For GUI installation
./runInstaller
```

I ran the silent installation and had the following output on my terminal once the installation completed.



For now, I am going to ignore the failed prerequisites. However, I am being prompted to run the scripts circled in red as a root user. I will also put the names of these scripts below.

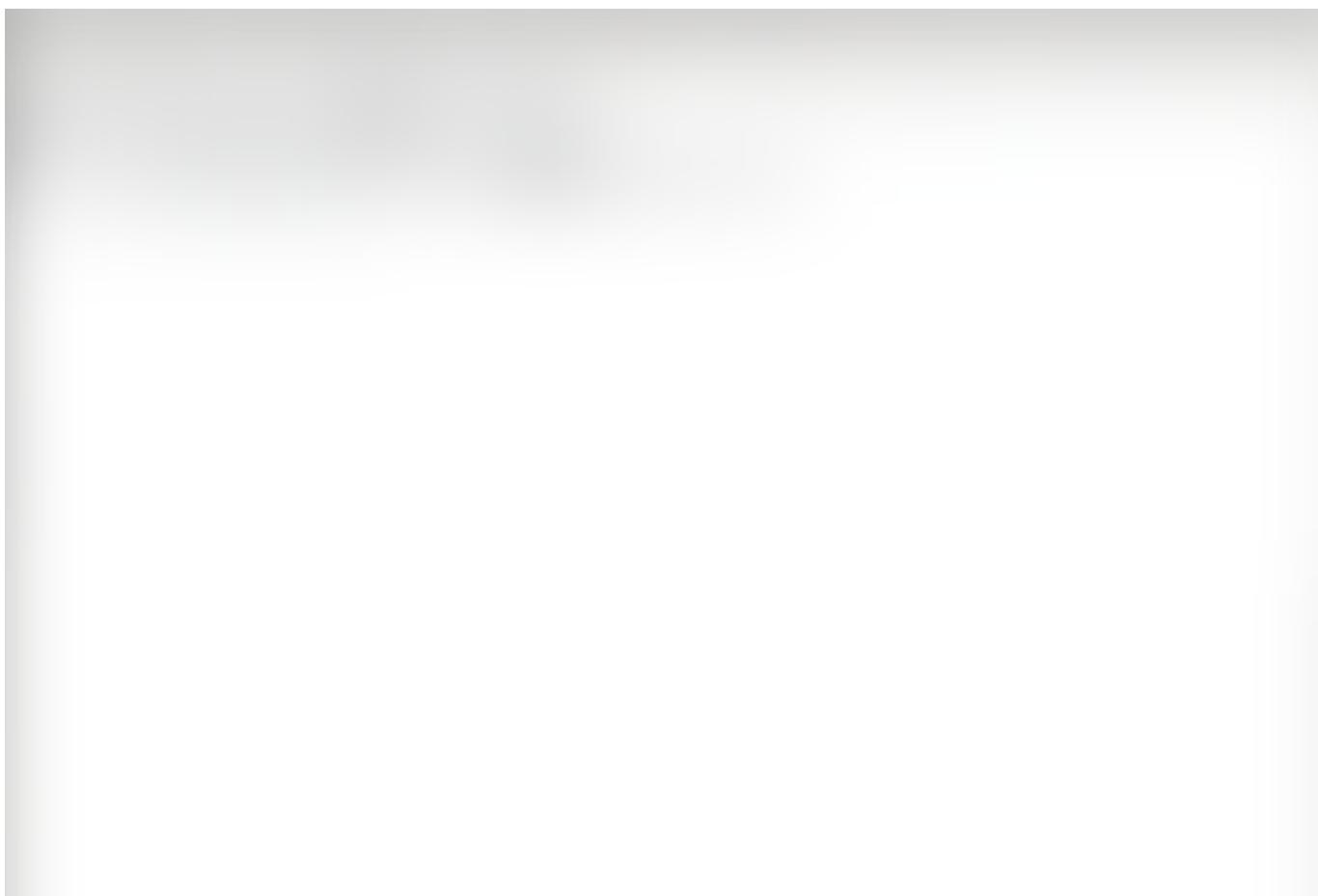
As a root user, execute the following script(s):

1. /u01/app/oraInventory/orainstRoot.sh
2. /u01/app/oracle/product/19.3/db\_home/root.sh

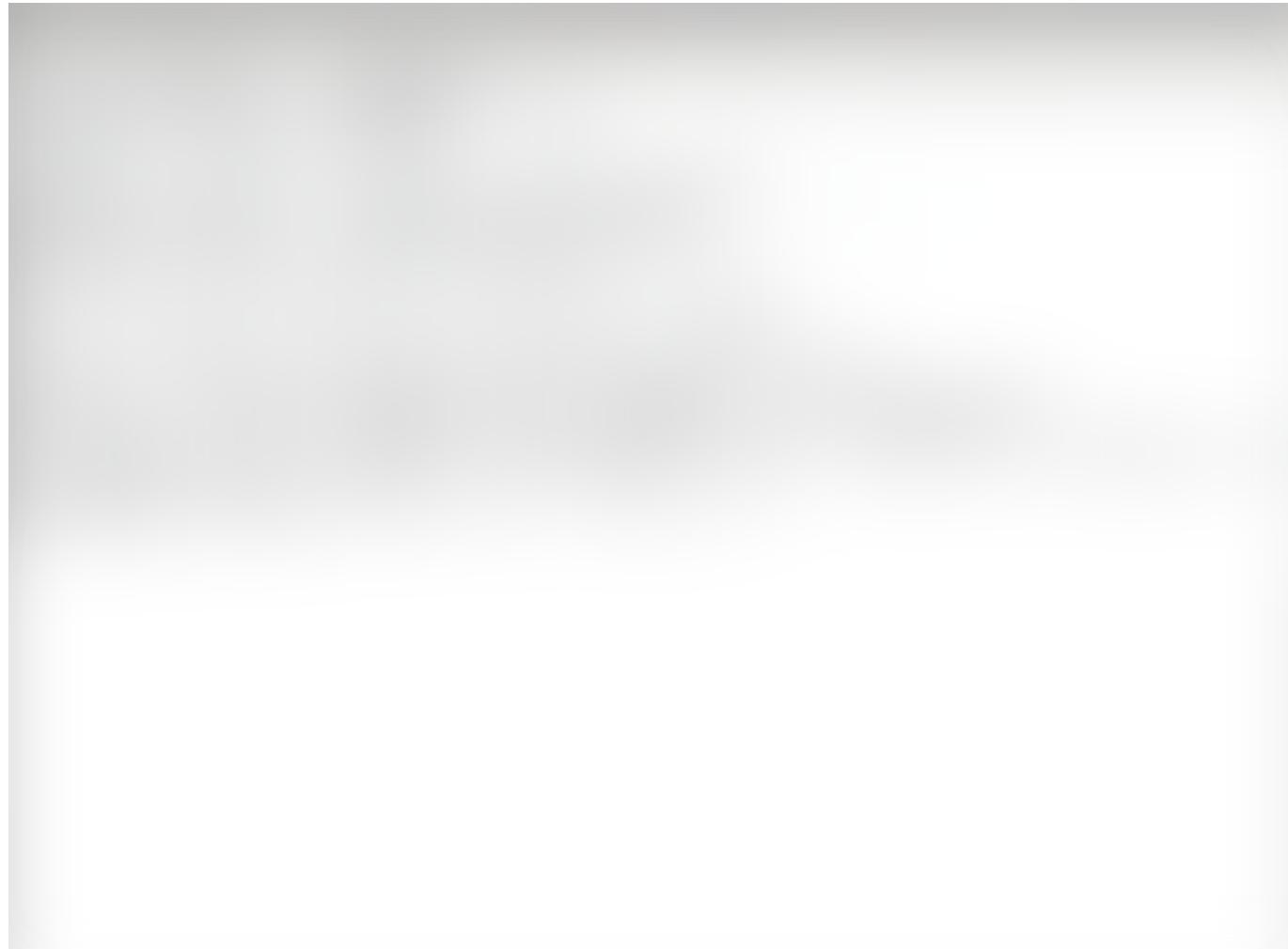
Inside my terminal, I am going to log-in as a root user with the following command.

```
sudo su -
```

Your terminal should now look like this.



I am now going to run the scripts that I had circled in red. The output should look like this.



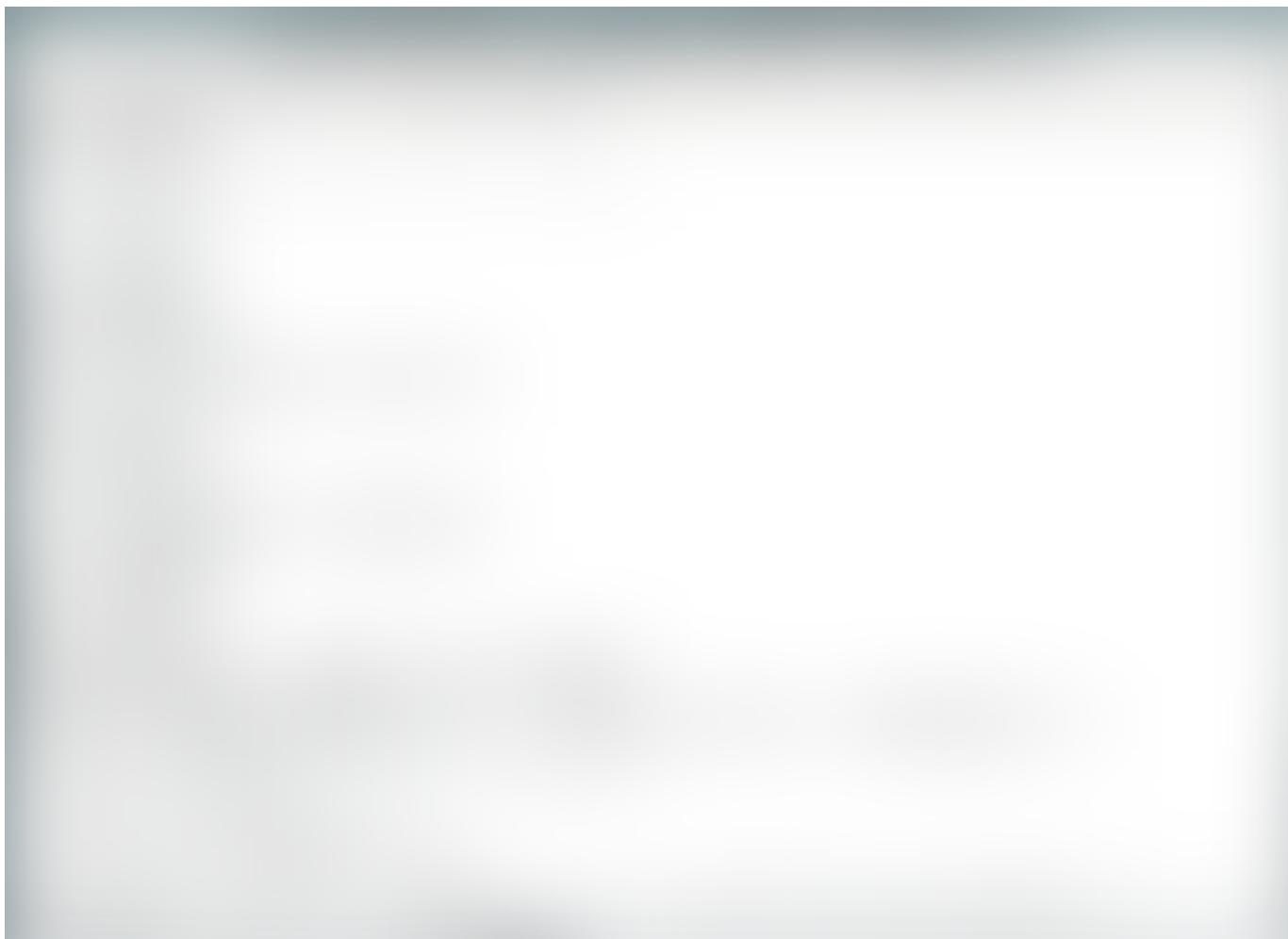
We have now successfully installed Oracle Database 19c. Let's now create a DBCA 19c container database. Run the following in your terminal.

```
dbca -silent -createDatabase
      -templateName General_Purpose.dbc
      -gdbname ${ORACLE_SID} -sid ${ORACLE_SID}
      -characterSet AL32UTF8
      -sysPassword enterCDB#123
      -systemPassword enterCDB#123
      -createAsContainerDatabase true
      -totalMemory 2000
      -storageType FS
      -datafileDestination /u01/CDB
```

```
-emConfiguration NONE  
-numberOfPDBs 2  
-pdbName PDB  
-pdbAdminPassword enterPDB#123  
-ignorePreReqs
```

— — — —

Your terminal should look like this when the process has been completed. Be patient. It takes some time.



Now, let's check all the containers inside the database. First, log-in to SQL Plus.

```
sqlplus "/ AS SYSDBA"
```

If you need to reconnect to the ORACLE instance (for example, you just started up a new virtual machine and are connecting to this boot volume for the first time in a new

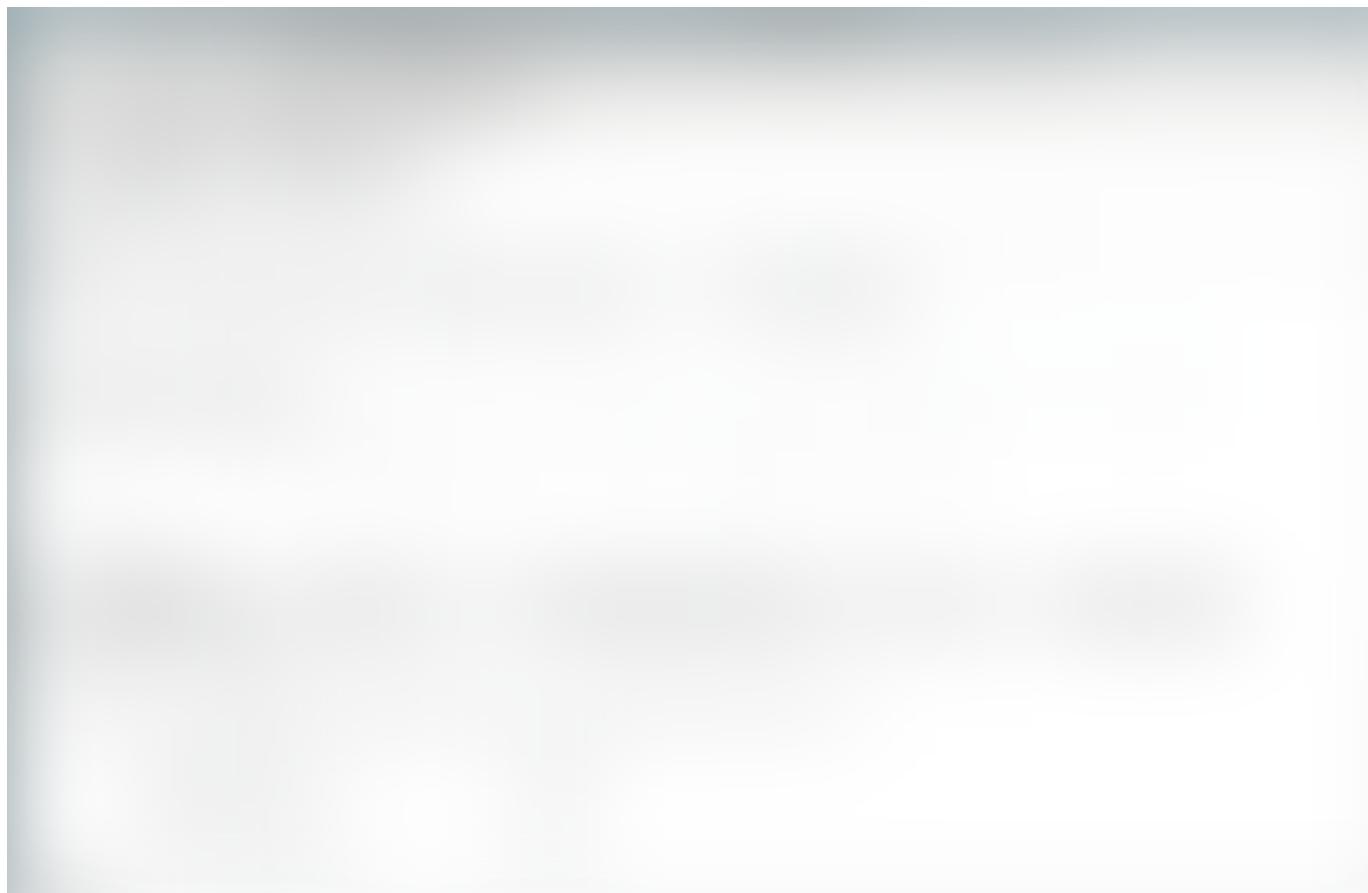
instance), run this command.

```
STARTUP FORCE;
```

Then, run the following query.

```
SELECT NAME, OPEN_MODE, CDB FROM V$DATABASE;
```

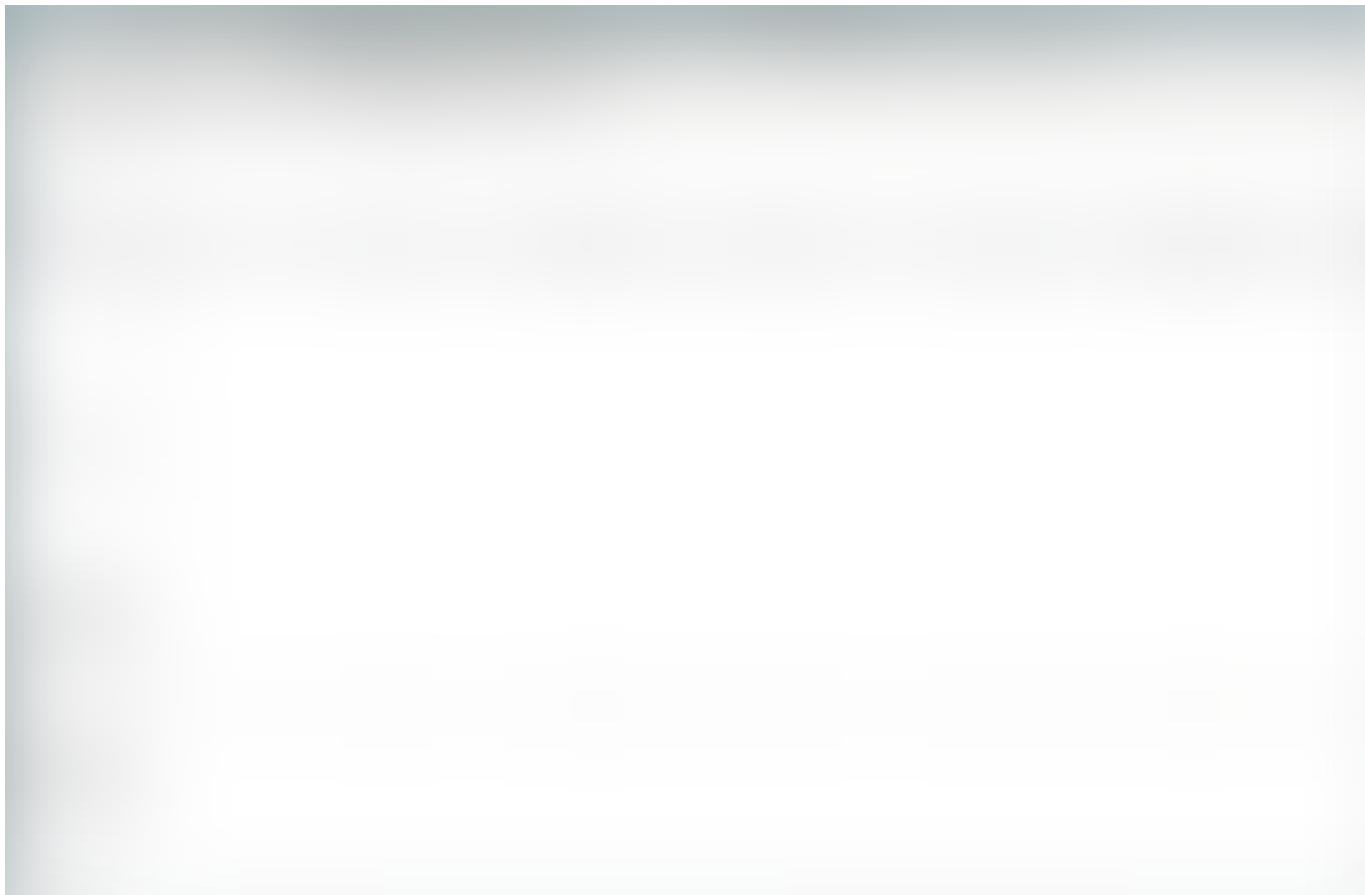
The terminal should look like this.



Then, run this following query.

```
SELECT CON_ID, NAME, OPEN_MODE FROM V$CONTAINERS;
```

The terminal should look like this.



It looks like everything is running. We are not done yet, however. We need to check to see that our connection to the database works. We can do this with SQL Developer.

Type *exit* in your terminal and press enter to exit SQL Plus and to return to the normal terminal.

## Installing SQL Developer on Linux



## Step 1: Installing JDK

Before we install SQL Developer, we need to make sure that we have the correct version of Java JDK installed. For SQL Developer to work properly, we need either JDK 8 or 11. Navigate to [this page](#) to download JDK 8 for your machine. On the page, scroll down to Linux x64 RPM Package and download it.

Before the download can commence, you will be prompted to log into your Oracle SSO account. After successfully logging-in, you will be asked if you want to open the download file with Archive Manager (default) or if you want to save the file. Select “Save File” and then hit “OK.” Make a mental note of where this zip file was saved. It should have been saved, by default, in your downloads folder. After the download is completed, navigate back to your terminal and make sure that you are logged in as root. Again, you can log-in as root with the following command.

```
sudo su -
```

Your terminal screen should look like this.



Navigate to where you downloaded the RPM. By default, it should have saved to the Downloads folder. Navigate to the Downloads folder with the following command.

```
cd /home/oracle/Downloads
```

Your terminal should now look like this.



To install the RPM package, run the following command.

```
rpm -ivh jdk-8u301-linux-x64.rpm
```

## Step 2: Installing SQL Developer

We now need to install SQL Developer. To install SQL Developer, [navigate to this page](#) and download the Linux RPM file. Again, make a mental note of where you download this file.



Navigate to where you downloaded the RPM. By default, it should have saved to the Downloads folder. Navigate to the Downloads folder with the following command.

```
cd /home/oracle/Downloads
```

Your terminal should now look like this. Again, remember to be logged-in as root.





To install the RPM package, run the following command.

```
rpm -ivh sqldeveloper-21.2.1-204.1703.noarch.rpm
```

We now need to navigate to the SQL Developer folder. We can do that with this command.

```
cd /opt/sqldeveloper
```

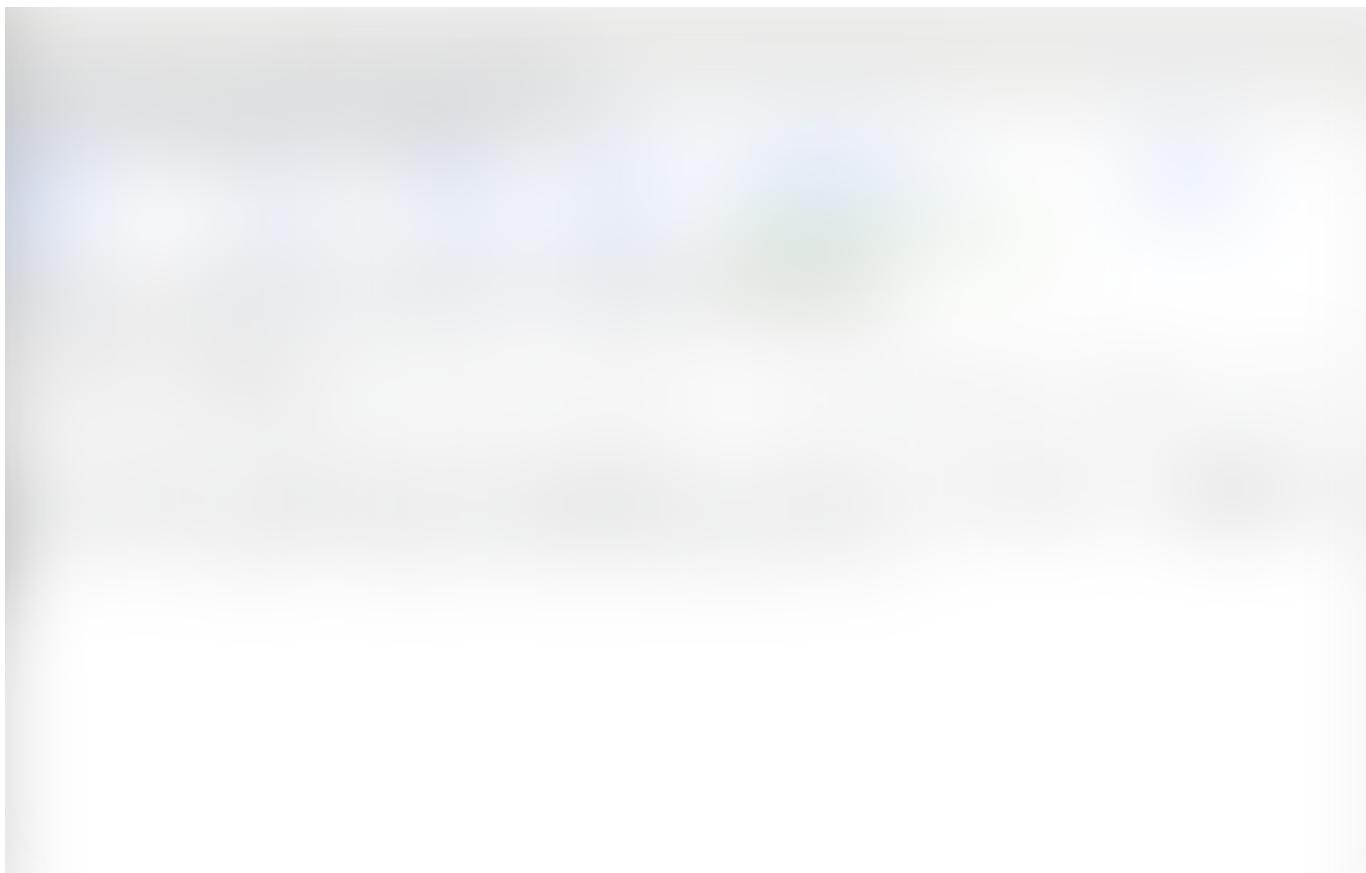
With the `ls` command we can see the contents of the `sqldeveloper` folder. Your terminal should look like this.



We need to run the sqldeveloper.sh file to complete the installation and start-up SQL Developer. Run the following command to do this.

```
./sqldeveloper.sh
```

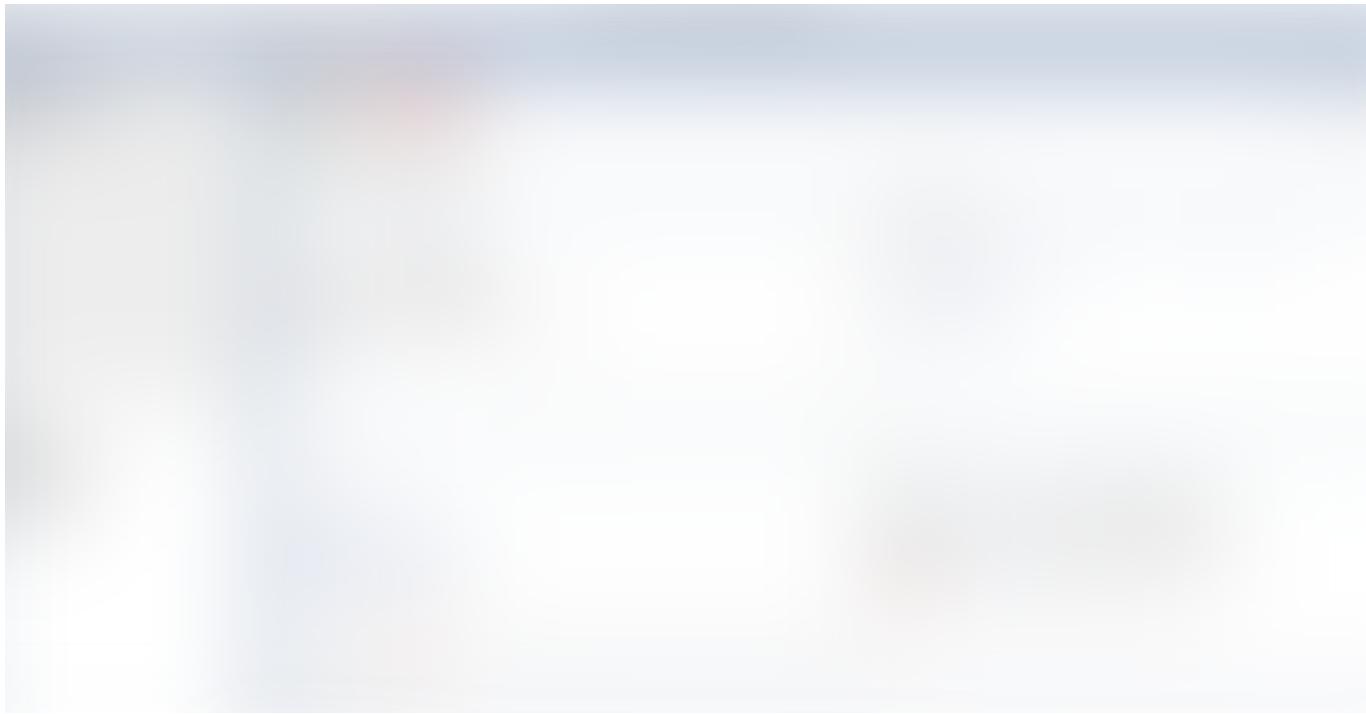
Your screen should now look like this.



You are being asked for the path of the JDK that we installed. Paste this into your terminal

```
/usr/java/jdk1.8.0_301-amd64
```

Oracle SQL Developer should now have successfully loaded and opened. Your screen should look like this. Minimize SQL Developer for now. We will be coming back to this.



In the future, when you want to start a new SQL Developer session, you can run the following command in your terminal to open the application.

```
/opt/sqldeveloper/sqldeveloper.sh
```

If you are prompted for it again, this is the JDK path.

```
/usr/java/jdk1.8.0_301-amd64
```

Let's now connect to a sample database. Minimize SQL Developer for now.

## Accessing a sample dataset

We are now almost done. The final step in this installation is testing our connection to a database in Oracle 19c. To test our connection, we are going to tap into a sample dataset that comes with Oracle Database 19c: the HR dataset. We first need to unlock the sample HR user for this database.



Follow the steps below to do this. Run the following command in your terminal to log-in to SQL Plus as a SYS user.

```
sqlplus "/ AS SYSDBA"
```

## Step 1: Find PDB name

Execute the following command in SQL.

```
SELECT name, con_id, open_mode FROM v$pdbs;
```

Your output should look something like this.

The PDB base is PDB1. Additionally, we can see that PDB1 is mounted but it does not have read and write privileges — yet. We are going to fix this.

## 2. Create and update tnsnames.ora file

The tnsnames.ora file contains client side network configuration parameters. Execute the line of code below in a new terminal tab, outside of SQL Plus. This will bring you to the location where your tnsnames.ora file is located, or should be located. If there is no tnsnames.ora file in this location — do not fret. We are going to create one. Open a new terminal tab and input the following command to change the directory.

```
cd $ORACLE_HOME/network/admin
```

To create and edit the tnsnames.ora file, run the line of code below.

```
sudo vi tnsnames.ora
```

Your terminal screen should now look like this.

Press the letter “i” on your keyboard to begin editing the document. Add the following text.

```
LISTENER_CDB =
  (ADDRESS = (PROTOCOL = TCP)(HOST = LOCALHOST)(PORT = 1521))

CDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = LOCALHOST)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = CDB)
    )
  )

PDB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = LOCALHOST)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PDB1)
    )
  )
```

Once you complete this task, hit the ESC button, scroll down to the bottom of the document and type the following text. Then, hit enter.

```
:wq
```

This sequence will save the file and exit the editor.

### 3. Create and update the listener.ora file

The listerner.ora file contains server side network configuration parameters. To create and configure the listener.ora file, run the line of code below inside of the directory that we were just working out of.

```
sudo vi listener.ora
```

Your terminal screen should now look like this.

Press the letter “i” on your keyboard to begin editing the document. Add the following text.

```

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = CDB)
      (ORACLE_HOME = /u01/app/oracle/product/19.3/db_home)
      (SID_NAME = CDB)

(ENVS="TNS_ADMIN=/u01/app/oracle/product/19.3/db_home/network/admin")
  )
  (SID_DESC =
    (GLOBAL_DBNAME = PDB1)
    (ORACLE_HOME = /u01/app/oracle/product/19.3/db_home)
    (SID_NAME = PDB1)

(ENVS="TNS_ADMIN=/u01/app/oracle/product/19.3/db_home/network/admin")
  )
  )

ADR_BASE_LISTENER = /u01/app/oracle

```

Once you complete this task, hit the ESC button, scroll down to the bottom of the document and type the following text. Then, hit enter.

```
:wq
```

This sequence will save the file and exit the editor. Stop and then start the listener with this command

```
lsnrctl stop
lsnrctl start
```

## 4. Open PDB1

We now want to connect to PDB1. Navigate back to your terminal that had SQL Plus open. If you need to re-open a terminal with SQL Plus, run this command.

```
sqlplus "/ AS SYSDBA"
```

Your terminal screen should look like this.

Run the command below to open-up our container database.

```
ALTER PLUGGABLE DATABASE ALL OPEN;
```

## 5. Connect to container PDB1

Run the command below to connect to PDB1.

```
ALTER SESSION SET container=PDB1;
```

## 6. Unlock HR schema

Finally, we can unlock the HR schema with the command below.

```
ALTER USER HR IDENTIFIED BY HR ACCOUNT UNLOCK;
```

If you get this error, like I did, do not fret. Continue to the next step.

## 7. Checking the tnsnames.ora configuration

In a new terminal outside of SQL Plus, change directories by executing the command below.

```
cd $ORACLE_HOME/bin
```

Once you are in this directory, execute the command below to check that the tnsnames.ora file is configured correctly.

```
./tnsping pdb1
```

If it is, then your output should look like this.

Let's move onto the next step, where we will finally connect to the HR dataset.

## Finalize the HR dataset connection

Next, we will need to run the `hr_main.sql` file to create all the objects and load the data. The following steps will provide a summary of this installation process.

Run this command in your terminal inside SQL PLUS.

```
@?/demo/schema/human_resources/hr_main.sql
```

Follow the prompts on the screen. Use the criteria below to guide you.

1. Enter *hr* here. This is for the user HR. This is what we will use to access the database.
2. Enter an appropriate tablespace, for example, *users* as the default tablespace for HR.
3. Enter *temp* as the temporary tablespace for HR
4. Enter the directory path for your log directory, for example:

```
$ORACLE_HOME/demo/schema/log/
```

After script *hr\_main.sql* runs successfully and schema HR is installed, you are connected as user HR. To verify that the schema was created, use the following command.

```
SELECT table_name FROM user_tables;
```

You should have gotten a result that looked like this.

We are onto something. Let's test the connection in SQL Developer.

## Accessing the HR database in SQL Developer

Open up SQL Developer on your local virtual machine. If you need to re-open it, use the following command in your terminal.

```
/opt/sqldeveloper/sqldeveloper.sh
```

If you are prompted for it again, this is the JDK path.

/usr/java/jdk1.8.0\_301-amd64

Once you are inside SQL Developer, select the “Create a Connection Manually” button, circled in red below.

A window will pop-up that looks like this.

Fill it out with the following criteria:

- **Name:** *HR*
- **Username:** *hr*
- **Password:** *hr*
- **Service name:** check this box and put *PDB1*

The window should now look like this.

When you are ready, click “Test” — the moment of truth. You will know if it was successful if it says “Success” in the bottom left hand corner next to “Status:”. Once the test is successful, click the “Save” button and then click “Connect.” If you are prompted for a password again, enter *hr*.

Your screen should now look like this.

Let's run a quick query to see our database in-action. Let's see how many employees there are in the database with the following SQL command.

```
select count(*) from employees;
```

I got 107 — it worked.

## Starting the database after a computer reboot

One last note — if you reboot your computer you are going to need to follow the steps below to get your database up-and-running again. First, start the listener with the command below.

```
lsnrctl start
```

We now need to log back into SQL Plus. Use the command below to log back into SQL Plus.

```
sqlplus "/ AS SYSDBA"
```

Run this command to startup the database.

```
STARTUP FORCE;
```

Run the command below to open-up the container database, if it is not already open.

```
ALTER PLUGGABLE DATABASE ALL OPEN;
```

Run the command below to connect to PDB1, if you are not already connected to it.

```
ALTER SESSION SET container=PDB1;
```

Enjoy your database.

---

**Get an email whenever Jared Bach publishes.**

Your email

Subscribe

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

[Oracle Cloud](#)    [Oracle Database](#)    [Oracle 19c](#)    [Sql](#)    [Sql Developer](#)

[About](#)   [Write](#)   [Help](#)   [Legal](#)

Get the Medium app

