

Set up Oracle 12c on Linux VM in Azure

Step-by-step walkthrough

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

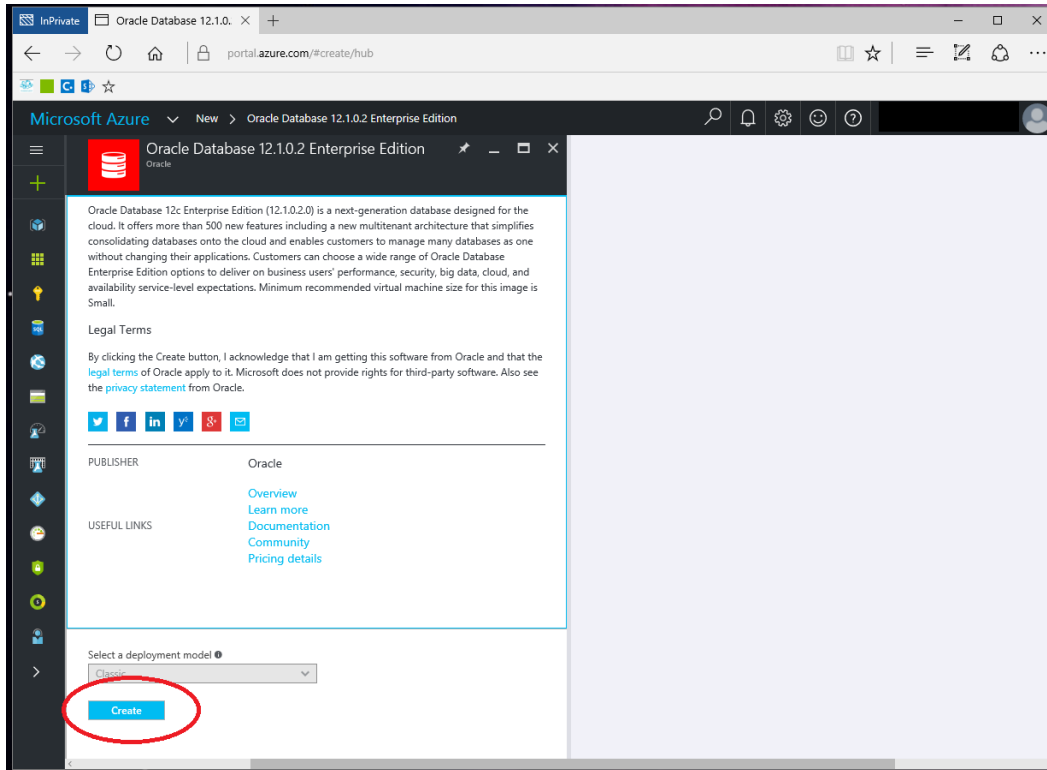
Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2018 Microsoft. All rights reserved.

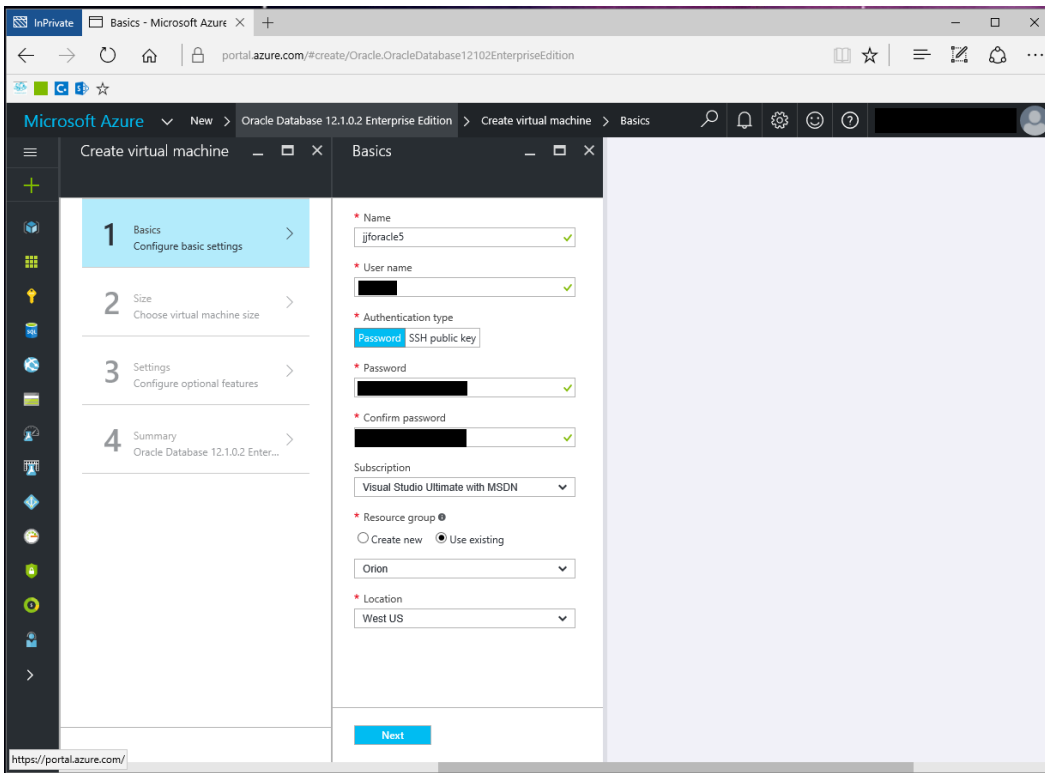
1.) Create Azure VM Instance using Oracle 12c Template

Log in to your Azure subscription, go to the Azure Marketplace, search for 'Oracle' and find the Marketplace Template named 'Oracle Database 12.1.0.2 Enterprise Edition' (shown below).



2.) Fill out the settings on the 'Basics' blade.

Note: This procedure uses a 'Password' for the authentication type rather than the SSH public key.

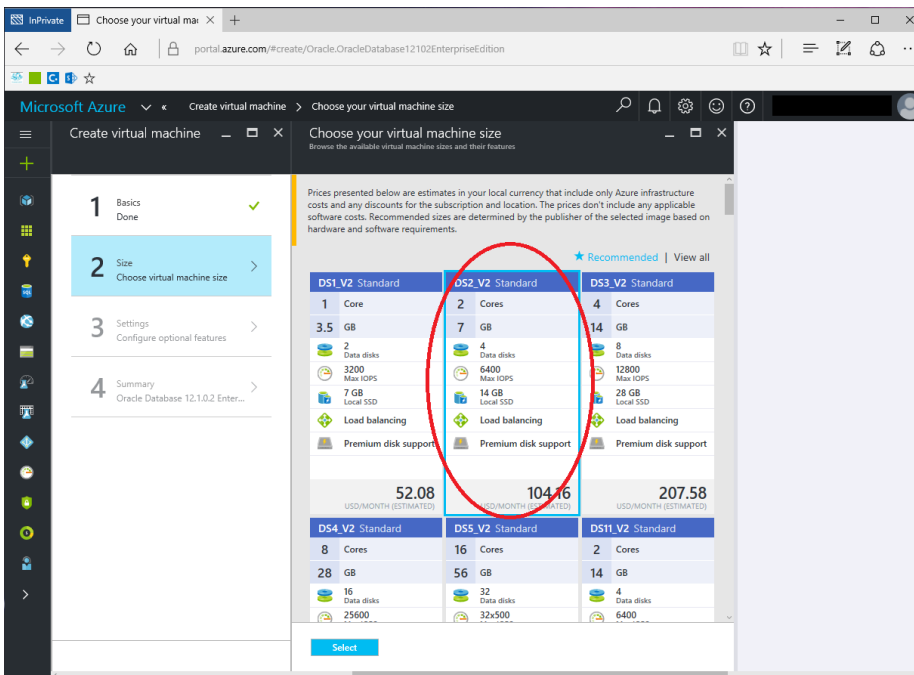


3.) Fill out the settings on the 'Size' blade

At the time of writing this document, I found the DS2_V2 VM size to be the best value for performance (this may change). (**Note:** Oracle 12c requires at least 2 cores to be in a supportable state) (**Note:** To find the DS2_V2 VM size configuration, you must click 'View All' link on upper right)

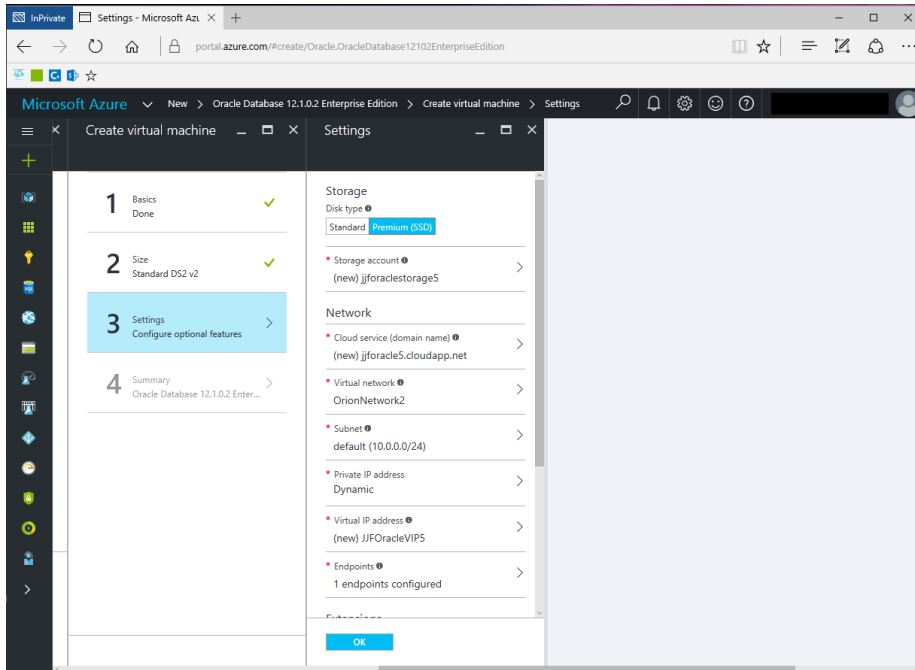
[Oracle 12c Hardware Requirements:

http://docs.oracle.com/cd/E25178_01/install.1111/e22624/preinstall_req_hw.htm]



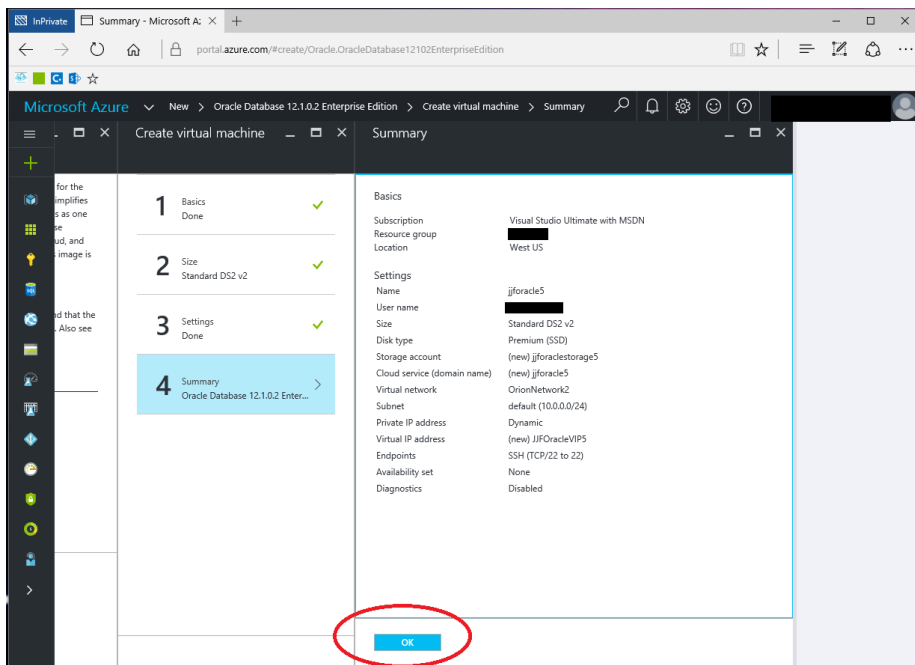
4.) Fill out the settings on the 'Settings' blade.

Fill out settings based on your existing network in your Azure subscription. If creating isolated VM, then create new Virtual Network and Subnet. Be sure to create an assigned Virtual IP Address for easy SSH access later. Note that Port 22 is already added for SSH access.



5.) Review the settings specified on the 'Summary' blade

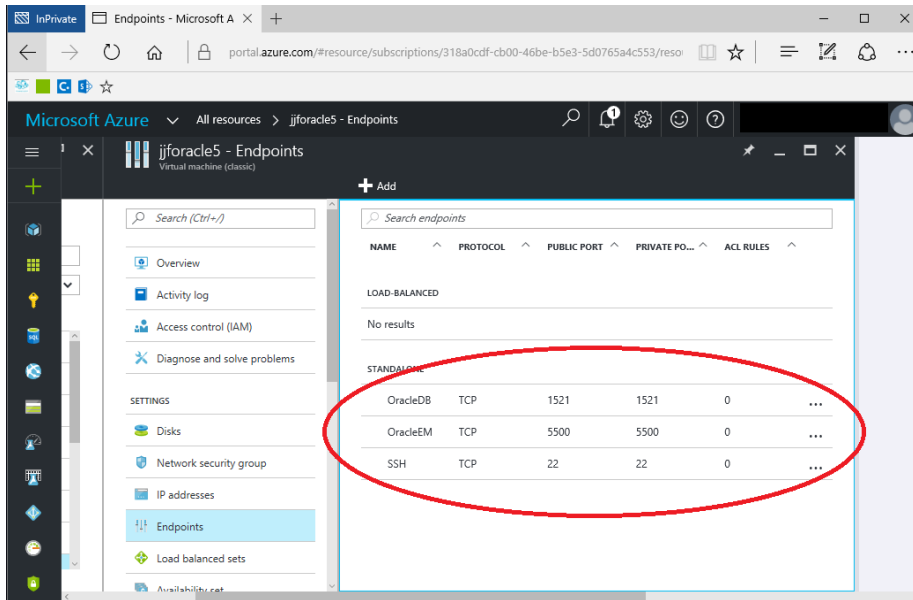
Review the summary of the settings, and then click 'OK' to kick off the deployment of this instance.



6.) Open Endpoints for Oracle DB Access and Oracle Enterprise Manager

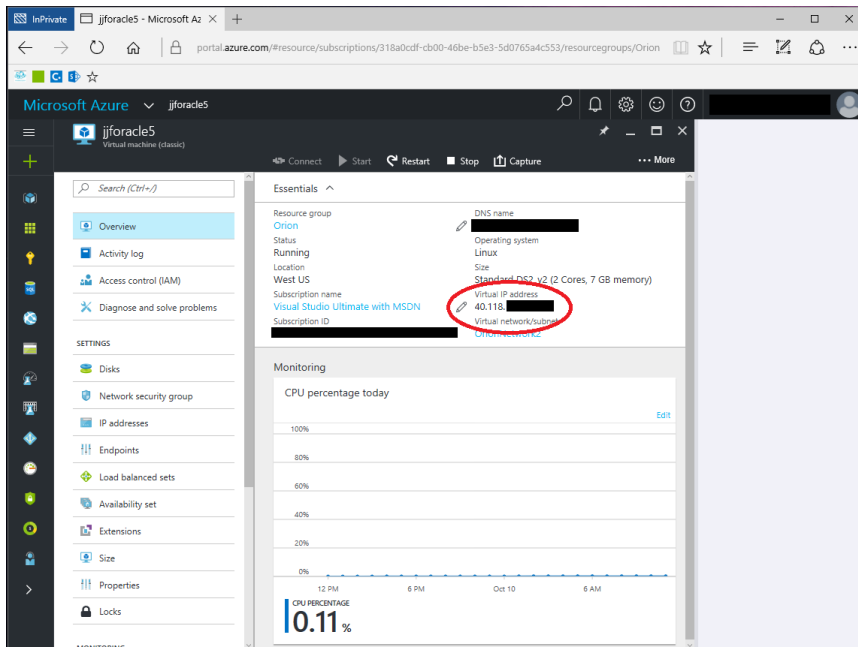
After the VM is deployed, click on the VM instance, and then navigate to Settings > Endpoints. Add the ports 1521 (For Oracle DB Access) and 5500 (For Oracle Enterprise Manager) both on TCP Protocol.

(**Note:** The Oracle 12c Template provisions a Classic-style VM, so the Endpoint addition process is different than with a new Resource Manager-style VM).



7.) Find the Virtual IP Address for the VM

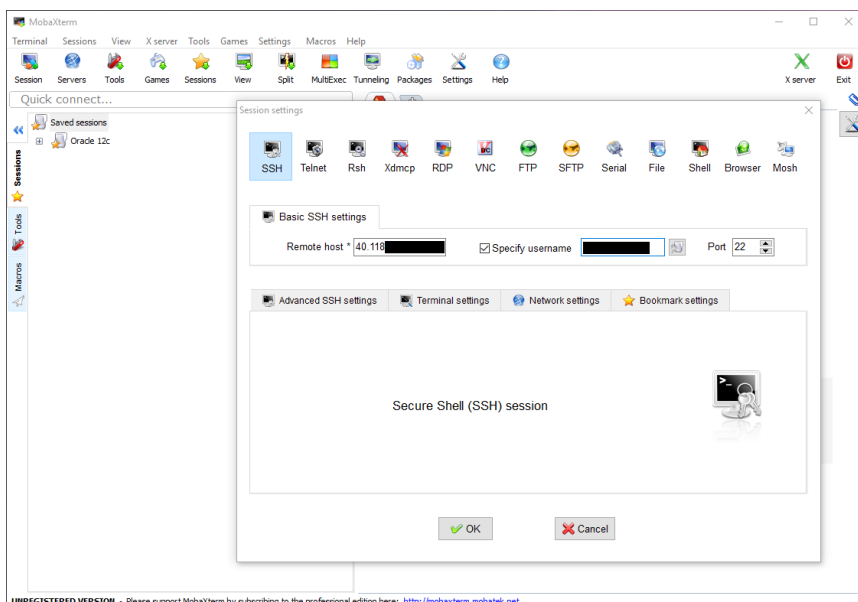
Click on the VM Instance and in the 'Overview' tab, the Virtual IP Address is located under the 'Essentials' heading. Note down this IP address to use it next. You could also use the DNS Hostname alternatively.



8.) Remote into the VM using SSH

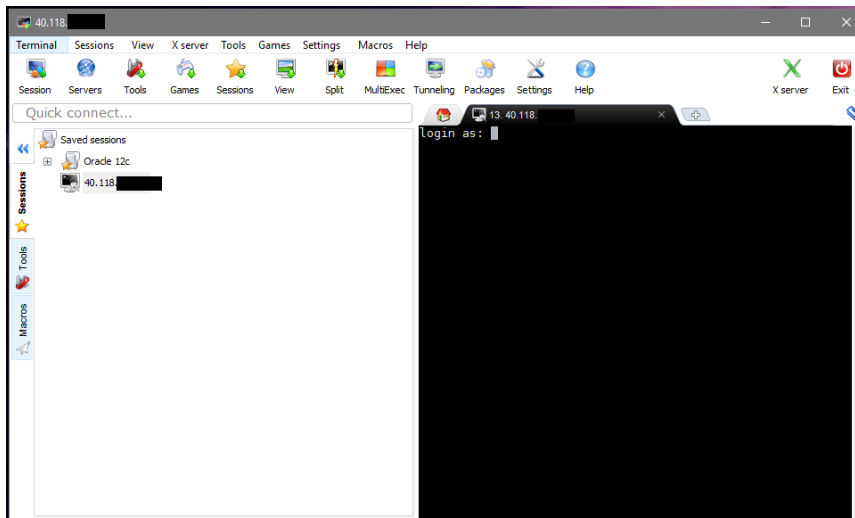
There are many different tools to use to SSH into a Linux VM. A very popular one is Putty, which [can be downloaded here](#). I like to use MobaXTerm, which [can be found here](#). MobaXTerm has similarities to the tool 'Remote Desktop Connection Manager' in that you can create saved profiles of connections, save passwords and organize them in folders. For this walkthrough, I use MobaXTerm, but a tool such as Putty could just as easily be used.

After downloading MobaXTerm, go 'Sessions > New session' to create a new session and enter the Virtual IP Address obtained (or DNS Hostname). After you click 'OK' on the 'Session settings' dialog, the session activates and you can login into the VM.



9.) Login to VM over SSH

When prompted the 'login as', provide the username you selected when you created the VM and hit enter. Then for the password, provide the password you selected when you create the VM.



10.) Edit Bash Profile for your user account and for 'oracle' account

Now that we are logged into the VM over SSH, the first thing we need to do is edit the Bash profile for both the user account you signed in as plus an account named 'oracle' that was created as part of the Oracle 12c VM provisioning process. We will use the 'oracle' account shortly.

Editing the Bash profile will automatically set some Oracle related environmental variables going forward.

Note that the Oracle 12c bits are installed during the VM provisioning process, but there are some post-install setup steps that need to be done to configure the database and provide access to the database.

a. Navigate to the home directory for your user account and edit the .bash_profile file in vi

```
$ cd /home/{username}
$ vi .bash_profile
```

b. Inside of vi, add the following lines at the bottom of the file to set these environmental variables:

(Note: vi is a text editor on UNIX and Linux, for a guide on how to use vi, see this [link](#) and here is [another](#))

(Also note: we are setting the Oracle SID to 'orcl', we will use this later on in the next steps)

```
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
ORACLE_SID=orcl
```

```
export ORACLE_HOME
export ORACLE_SID
```

c. Save and exit the file in vi (ESC, colon, wq, enter).

d. Log in as the 'oracle' user and edit the .bash_profile for that account

```
$ sudo -u oracle -i
{enter password}
$ cd /home/oracle
$ vi .bash_profile
```

- e. Inside of vi, add the following lines at the bottom of the file to set these environmental variables:**

```
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
ORACLE_SID=orcl

export ORACLE_HOME
export ORACLE_SID
```

- f. Save and exit the file in vi (ESC, colon, wq, enter)**

- g. Lastly, exit from the session in and re-login**

This will ensure the Bash profile will take effect and set the environmental variables.

11.) Create Database using dbca

In this step we will create the database for this Oracle instance. This command below will create a main database plus one 'pluggable' database. Pluggable databases (multitenant architecture) are introduced in Oracle 12c and follow a similar architecture to SQL Server, where now, an Oracle server instance can contain multiple databases where previously there was only a one to one relationship between Oracle instance and Oracle database.

- a. Login as your user account, then switch to the 'oracle' user**

```
$ sudo -u oracle -i
{enter password}
```

- b. Invoke the dbca command with the following parameters.**

(**Note:** the passwords below in yellow need to be added to whatever you prefer.)

```
$ dbca -silent -createDatabase -templateName General_Purpose.dbc -gdbname orcl -sid orcl -
responseFile NO_VALUE -characterSet AL32UTF8 -sysPassword {password} -systemPassword
{password} -createAsContainerDatabase true -numberOfPDBs 1 -pdbName pdb1 -
pdbAdminPassword Password1 -databaseType MULTIPURPOSE -automaticMemoryManagement
false -storageType FS -ignorePreReqs
```

```
$ dbca -silent -createDatabase -templateName General_Purpose.dbc -gdbname orcl -sid orcl -
responseFile NO_VALUE -characterSet AL32UTF8 -sysPassword {password} -systemPassword
{password} -createAsContainerDatabase false -databaseType MULTIPURPOSE -storageType FS -
ignorePreReqs -sampleSchema true
```

This step above will run for a while to create the databases. When this step is complete, you should see something similar to the output below:

```
Copying database files
1% complete
2% complete
8% complete
13% complete
27% complete
Creating and starting Oracle instance
29% complete
32% complete
33% complete
34% complete
38% complete
42% complete
```


43% complete
 45% complete
 Completing Database Creation
 48% complete
 51% complete
 53% complete
 62% complete
 70% complete
 72% complete
 Creating Pluggable Databases
 78% complete
 100% complete
 Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/orcl/orcl.log" for further details.

12.)Start up the Oracle Listener

a. Ensure you are still logged in with user 'oracle', if not follow command below

```
$ sudo -u oracle -i
{enter password}
```

b. Invoke the lsnrctl command with the 'start' parameter

(Note: the passwords below in yellow need to be added to whatever you prefer.)

```
$ lsnrctl start
```

You should see some output similar to the below:

```
LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 10-OCT-2016 15:26:45
Copyright (c) 1991, 2014, Oracle. All rights reserved.
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...
TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Log messages written to /u01/app/oracle/diag/tnslsnr/jjforacle5/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=jjforacle5.jjforacle5.d7.internal.cloudapp.net)(
PORT=1521
)))
Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date           10-OCT-2016 15:26:45
Uptime                0 days 0 hr. 0 min. 0 sec
Trace Level           off
Security              ON: Local OS Authentication
SNMP                  OFF
Listener Log File    /u01/app/oracle/diag/tnslsnr/jjforacle5/listener/alert/log.xml
Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=jjforacle5.jjforacle5.d7.internal.cloudapp.net)(
PORT=1521)))
The listener supports no services
The command completed successfully
```

13.) Perform a test query in SQL*Plus

SQL*Plus (sqlplus command) is a command line program in Oracle to query the database, perform configuration changes, etc.

a. Invoke sqlplus and login as sys user

(**Note:** that the password for the 'sys' account was specified during the dbca database creation process)

```
$ sqlplus sys/{password} as sysdba
```

b. At the SQL*Plus command prompt, perform the query below:

(**Note:** that the password for the 'sys' account was specified during the dbca database creation process)

```
SQL> select name, dbid from v$pdb;
```

You should now see a result set of the portable databases just as an example query:

NAME	DBID
-----	-----
PDB\$SEED	4131105394
PDB1	3978492103

14.) Install a DB management tool such as Toad or SQL Developer to access the Oracle instance remotely

Now that we have the Database up and running, we can now access the database from a management tool such as Toad or SQL Developer. SQL Developer is a tool provided by Oracle and has similarities to SQL Server Management Studio. Toad is a 3rd Party Tool and also has similarities to SQL Server Management Studio.

You can [download Toad here](#). You can [download SQL Developer here](#).

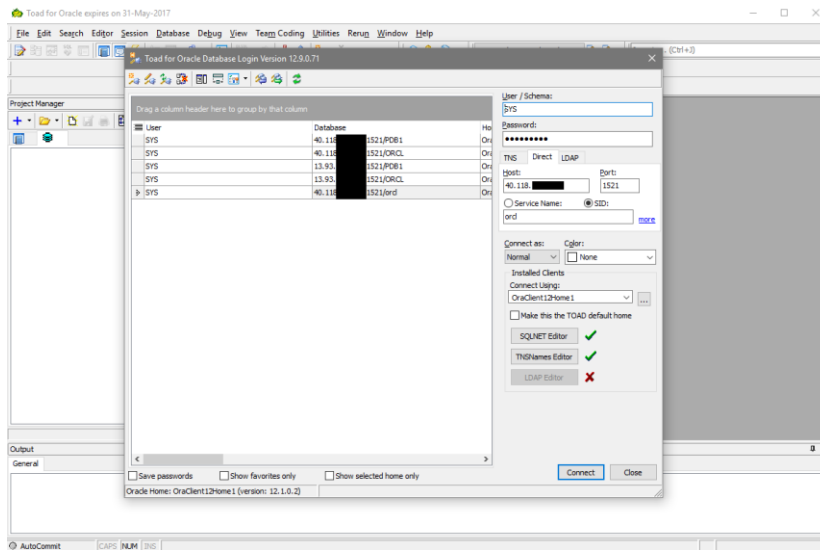
For these steps below, I used Toad for Oracle (on Windows 10) version 12.10, 64bit.

I used SQL Developer version 4.1.5, Windows 64bit with JDK 8 included.

To add a connection in Toad to both the main container database (CDB) and the pluggable database (PDB) deployed on the Oracle, these steps are below:

Add connection to main container database (CDB):

- Open Toad
- Go Session > New Connection
- For User / Schema you can use "SYS" (Oracle system user created automatically)
- For Password, enter the password provided during the dbca step earlier
- Choose the 'Direct' tab
- For Host, enter the Virtual IP Address of the Azure Oracle VM
- For Port, enter 1521
- To access the main container database, choose 'SID' option and enter 'orcl' as the SID name defined during the dbca step
- For Connect as, select 'SYSDBA'
- Click OK, and you should now have a connection to your container database.



Add connection to pluggable database (PDB):

To add a connection in Toad to the pluggable database (PDB) deployed on the Oracle, perform the following steps:

- a. Open Toad.
- b. Go Session > New Connection.
- c. For User / Schema you can use "SYS" (Oracle system user created automatically).
- d. For Password, enter the password provided during the dbca step earlier.
- e. Choose the 'Direct' tab.
- f. For Host, enter the Virtual IP Address of the Azure Oracle VM.
- g. For Port, enter 1521.
- h. To access the main container database, choose 'Service Name' option and enter 'pdb1' as the Service Name defined during the dbca step.
- i. For Connect as, select 'SYSDBA'.
- j. Click OK, and you should now have a connection to your pluggable database.

15.)Configure Oracle Enterprise Manager

In this step, we will configure Oracle Enterprise Manager, a built-in web application to Oracle that provides a visual interface into the Oracle database and configurations:

<TBD>

Feedback and suggestions

If you have feedback or suggestions for improving this data migration asset, please contact the Data Migration Jumpstart Team (askdmjfordmtools@microsoft.com). Thanks for your support!

Note: For additional information about migrating various source databases to Azure, see the [Azure Database Migration Guide](#).