Introduction to Oracle 12c Multitenant Container and Pluggable Databases

Aman Sharma

@amansharma81

http://blog.aristadba.com
What We Are Going to Talk About?

- Introduction to pre 12.1 Databases
- Introduction to 12c Multitenant database
- Creation of CDB and PDB
- Q & A
Who Am I?

- Aman Sharma
- About 12+ years using Oracle Database
- **Oracle ACE**
- Frequent Contributor to OTN Database forum(*Aman....*)
- Oracle Certified
- Sun Certified

* @amansharma81
* http://blog.aristadba.com

Sangam-2014
Recap-Oracle Database 11g

SALES App

SALES Database

System Metadata

User Metadata

User Data

Instance

Shared Pool

Buffer Cache

LGWR

DBWR

PMON

ORL

ARCH

CTL

SYSTEM

SYSAUX

UNDO

FLBLOG

BKUP

TEMP

Sangam-2014
The More The Merrier?

SALES App
- Shared Pool
- Buffer Cache
- LGWR
- DBWR
- PMON
- System Metadata
- User Metadata
- User Data

HR App
- Shared Pool
- Buffer Cache
- LGWR
- DBWR
- PMON
- System Metadata
- User Metadata
- User Data

SALES
- ORL
- ARCH
- CTL
- SYSTEM
- SYSAUX
- SALES
- UNDO
- TEMP

HR
- ORL
- ARCH
- CTL
- SYSTEM
- SYSAUX
- HR
- UNDO
- TEMP

Sangam-2014
Not really!!

• Ideally, one server should host just one database
  — But world != ideal
• In real-world-server to database ratio = 1:N
• Individual databases need individual resource allocations
• Resource requirements are never same
• Consolidation, with 1:N-becomes a major challenge
• Un-necessary system metadata duplication
• Provisioning remains a challenge
• Patching remains a challenge
Which one Is Better?

11g

12c
Say Hello To Oracle **Database 12c**

- Newest release-came out in **June, 2013**
- About **500** new features
- Introduced the concept of **Multi-tenancy**
- Forget **g** as **c** is in the house
- Brings **clouds** a little more closer 😊
Is my data safe in the Cloud?

Yeah, until it rains.

Brainstuck.com
First Oracle database release for cloud and consolidation

- Allows multiple tenant databases under one main container
- Tenants share the SGA, background processes & data-dictionary
- Reduced resource sharing
  - Less storage space
  - Less RAM
  - Lesser cost
- Rapid
  - Provisioning
  - Migration
- Much faster Patching and Upgrading
- Database privacy remains intact
• Full support for RAC & Data Guard features
• Compatible with non-CDB databases working
• Easy GUI management using
  – Cloud Control 12c
  – EM Express
  – SQL Developer
• Ease in Database Administration
• DBaaS
• Resource Manager for both CDB and PDB
• Manage many as one!!
Multitenant Database Architecture (simplified)

CDB Instance

CDB Database  \text{CON\_ID}=0

ROOT container

\text{CON\_ID}=1

SEED  \text{CON\_ID}=2

SALES PDB  \text{CON\_ID}=3

HR PDB  \text{CON\_ID}=4

FINANCE PDB  \text{CON\_ID}=5

Sangam-2014
Containers in a Multitenant Database

- A multitenant CDB database always contains
  - ROOT container
  - SEED container
- A CDB may also contain
  - PDB container
- An Instance is allocated for the whole CDB database
- \textbf{V$\text{CONTAINERS}$}
CDB Instance

- Maintained by **SYS** user
- Allocated for the CDB *only*
- Maintained by CDB specific **SPfile/PFILE**
- No PDB specific **SPFILE/PFILE(PDB_SP FILES)**
- Parameter(s) changed at PDB level don’t go up to their CDB database

CDB Instance

- **SGA PDB1**
- **SGA PDB2**
- **LGWR**
- **DBWR**
- **PMON**

Alert Log

**SPFILE**

Sangam-2014
Root Container **CDB$ROOT**

**CDB Database**
- **CON_ID=0**
- **CDB$ROOT**
  - **CON_ID=1**
  - **RSRCMGR**
  - **OBJ$**
  - **TAB$**
  - **USER$**

**System Metadata**

**PSDB$SEED**
- **CON_ID=2**
  - **ORL**
  - **ARCH**
  - **CTL**
  - **SYSTEM**
  - **SYSAUX**
  - **UNDO**
  - **FLBLOG**
  - **BKUP**
  - **TEMP**

Sangam-2014
Root Container CDB$ROOT

- The very first tenant within a multitenant database
- **CDB$ROOT**
- Contains SGA & PGA-\textit{shared} among all the PDB’s
- Contains data files, online redo logs, control files, Undo
- Contains \textit{system} metadata for Oracle supplied objects
- Contains \textit{system} schemas \& roles
- Contains information about underlying PDB’s
- Contains \textit{common users}
- Should \textit{not} contain user data
- Backup support by \textbf{RMAN}
- Managed by a Container DBA(CDBA)
Seed Container **PDB$SEED**

- **Default** Pluggable database
- Created at the time of creation of the multitenant database
- Named as **PDB$SEED**
- Acts as a template for creating pluggable databases in the future
- Users/DBA’s can’t create and modify objects in the seed database
- Can’t be deleted
- One CDB contains only one SEED container
- Backup support by RMAN
### Meet The Tenants via SQL

SQL> `select con_id, dbid, name from V$database;`

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>DBID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4244090852</td>
<td>ACDB</td>
</tr>
</tbody>
</table>

SQL> `select con_id, dbid, name from v$containers;`

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>DBID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4244090852</td>
<td>CDB$ROOT</td>
</tr>
<tr>
<td>2</td>
<td>4063311135</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>3</td>
<td>2577910186</td>
<td>APDB</td>
</tr>
</tbody>
</table>
Pluggable Containers (PDB’s)

CDB Database

CDB$ROOT

System Metadata

RSRCMGR

Metadata Link

USER Metadata

SALES PDB

PDB

OBJ$ TAB$ USER$

SYSTEM SYSAUX SALES USERS TEMP

SALES

Customers

Country

USER Data

SYSTEM SYSAUX UNDO FLBLOG TEMP

PSDB$SEED

SYSTEM SYSAUX TEMP

ORL ARCH CTL
Pluggable Containers (PDB’s)

- User defined and created
- Explicitly named
- Stores user defined objects e.g. tables, indexes etc
- Contains data dictionary holding information about user data
- **Pointers** are maintained to the system metadata container
- **Service Naming** is used for connections
- Contains **local users**-specific to exactly one PDB
- PDB specific Resource Manager
- **253** PDB’s (including **SEED**) maximum are allowed in one CDB
- Managed by PDB DBA (PDBA)
- Linked with other PDB’s using **inter-PDB DB links**
- **V$PDBS, DBA_PDBS**

Sangam-2014
There are several ways

✓ Using **OUI** (at the time of the installation)
✓ Using **DBCA**
✓ Using **CREATE DATABASE** command
Creating CDB using DBCA

Database Configuration Assistant - Create Database - Step 2 of 5

Creation Mode

- Database Operation
- Creation Mode
- Pre Requisite Checks
- Summary
- Progress Page

Create a database with default configuration
- Global Database Name:
- Storage Type: File System
- Database Files Location: (ORACLE_BASE)/oradata
- Fast Recovery Area: (ORACLE_BASE)/fast_recovery_area
- Database Character Set: WEBMSWIN1252 - MS Windows Code Page 1252 8-bit Wes...
- Administrative Password:
- Confirm Password:
- Create As Container Database
- Pluggable Database Name:
- Advanced Mode

Sangam-2014
The **CREATE DATABASE** command is *almost* the same

- **ENABLE PLUGGABLE DATABASE** clause must be used
- **SEED FILE_NAME_CONVERT** clause (only if *not* using OMF)
- **DB_CREATE_FILE_FILE_DEST** initialization parameter if using OMF

or

- **PDB_FILE_NAME_CONVERT** initialization parameter
CREATE DATABASE acdb
USER SYS IDENTIFIED BY ORCL
USER SYSTEM IDENTIFIED BY ORCL
EXTENT MANAGEMENT LOCAL
DEFAULT TABLESPACE users
DEFAULT TEMPORARY TABLESPACE temp
UNDO TABLESPACE undotbs1
ENABLE PLUGGABLE DATABASE
SEED
SYSTEM DATAFILES SIZE 300M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED SYSAUX DATAFILES SIZE 200M;

SQL> @/?/rdbms/admin/catcdbc.sql*

*17033183.8
CREATE DATABASE *without OMF*

```sql
SQL> CREATE DATABASE acdb
USER SYST IDENTIFIED BY ORACLE
USER SYSTEM IDENTIFIED BY ORACLE

....

ENABLE PLUGGABLE DATABASE SEED FILE_NAME_CONVERT = ('/u01/app/oracle/oradata/acdb/',
'/u01/app/oracle/oradata/pdbseed/') SYSTEM
DATAFILES SIZE 300M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
SYSAUX DATAFILES SIZE 200M USER_DATA TABLESPACE usertbs DATAFILE
'/u01/app/oracle/oradata/pdbseed/usertbs01.dbf'
SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

SQL> @~/rdbms/admin/catcdbc.sql
```
Successfully executed **CREATE DATABASE**?

- One **ROOT** container, one **SEED** container
- One database service per container
- **Local** & **Common** Users/Roles
- *New* Data Dictionary views
- Tablespaces
Common vs Local

- **Common user**
  - *Must* start from C## or c##
  - Are available in *all* the containers
  - Is defined within the **CDB$ROOT**
  - Works for *both* Root and underlying PDB’s
  - Default schemas SYS, SYSTEM are common to all containers
  - `CREATE USER....CONTAINER=ALL`

- **Local user**
  - Doesn’t need any prefix
  - Are available only in the PDB in which they are created
  - Is defined within the **specific PDB** itself only
  - Can *connect* and *work* for only their specific PDB
  - `CREATE USER....CONTAINER=<PDB>`
  - `CREATE USER(from PDB)`
CDB_*

- Dictionary views showing data from Root and all **Pluggable** databases

DBA_*

- Dictionary views showing data from a **container** or from **Pluggable** databases

ALL_*

- Dictionary views showing data **accessible** by a users

USER_*

- Dictionary views showing data **owned** by a user
Which Container We Are Into Now?

- SQL> show con_id
  CON_ID
  -----------------------------
  1
- SQL> show con_name
  CON_NAME
  -----------------------------
  APDB

SQL> conn sys/oracle@apdb as sysdba
Connected.
SQL> show con_id
  CON_ID
  -----------------------------
  3

SQL> conn sys/oracle@acdb as sysdba
Connected.
SQL> alter session set container=apdb;
Session altered.
SQL> show con_id
  CON_ID
  -----------------------------
  3
Look What You Ask For

- **SQL> conn / as sysdba**
  Connected.

- **SQL> show con_id**

  CON_ID
  1

- **SQL> select count(username) from dba_users;**

  COUNT(username)
  36

- **SQL> select count(username) from cdb_users;**

  COUNT(username)
  119

*contd.*
Look What You Ask For

- **SQL> connect sys/oracle@apdb as sysdba**
Connected.

- **SQL> show con_id**

  CON_ID
  ---------------------------
  3

- **SQL> select count(username) from dba_users;**

  COUNT(username)
  ----------------
  48

- **SQL> select count(username) from cdb_users;**

  COUNT(username)
  ----------------
  48
### CDB_SERVICES

```sql
SQL> select name from cdb_services;
NAME
-------------------
SYS$BACKGROUND
SYS$USERS
acdbXDB
acdb
apd
```

512 maximum services
```sql
SQL> select con_id, pdb_id, pdb_name, tablespace_name  
  2  from dba_pdb$ full outer join cdb_tablespaces  
  3  on dba_pdb$.pdb_id=cdb_tablespaces.con_id  
  4  order by con_id;

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>PDB_ID</th>
<th>PDB_NAME</th>
<th>TABLESPACE_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>SYSTEM</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>USERS</td>
<td>USERS</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>SYSAUX</td>
<td>SYSAUX</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>UNDOTBS1</td>
<td>UNDOTBS1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>TEMP</td>
<td>TEMP</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>PDB$SEED</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>PDB$SEED</td>
<td>SYSAUX</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>PDB$SEED</td>
<td>TEMP</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>APDB</td>
<td>USERS</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>APDB</td>
<td>TEMP</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>APDB</td>
<td>SYSAUX</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>APDB</td>
<td>EXAMPLE</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>APDB</td>
<td>SYSTEM</td>
</tr>
</tbody>
</table>

13 rows selected.
```
Managing & Provisioning PDB’s

✓ Using SQL commands
✓ Using SQL Developer
✓ Using Multi-tenant Self Service Provisioning application (Beta)
✓ Using Database Configuration Assistant (DBCA)
Managing PDB’s using SQL

- SQL> **alter pluggable database all open**;
  Pluggable database altered.
- SQL> **alter pluggable database all close**;
  Pluggable database altered.
- SQL> **select name, open_mode from v$pdbs**;

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDB$SEED</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>APDB</td>
<td>READ WRITE</td>
</tr>
</tbody>
</table>
- SQL> **alter pluggable database apdb close**;
  Pluggable database altered.
- SQL> **alter pluggable database apdb open read only**;
  Pluggable database altered.
- SQL> **select name, open_mode from v$pdbs**;

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDB$SEED</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>APDB</td>
<td>READ ONLY</td>
</tr>
</tbody>
</table>
## Querying Info About The PDB’s

1. **SQL**>
   ```sql
   select con_id, con_uid, name
   2   from V$pdbs;
   ```

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_UID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4063311135</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>3</td>
<td>2577910186</td>
<td>APDB</td>
</tr>
</tbody>
</table>

2. **SQL**>
   ```sql
   column pdb_name format a20
   ```

   **SQL**>
   ```sql
   select pdb_id, pdb_name, con_uid
   2   from dba_pdbs;
   ```

<table>
<thead>
<tr>
<th>PDB_ID</th>
<th>PDB_NAME</th>
<th>CON_UID</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>APDB</td>
<td>2577910186</td>
</tr>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>4063311135</td>
</tr>
</tbody>
</table>
Manually Provisioning PDB’s

I. By using **SEED** container
II. By **cloning** an existing PDB
III. By **plugging-in** an unplugged PDB
IV. By plugging-in a **non-CDB** into a **CDB**
CREATE PLUGGABLE DATABASE APDBCOPY ADMIN USER AMAN IDENTIFIED BY AMAN;

Pluggable database created.

SQL> SELECT CON_ID, NAME FROM V$PDBS;

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>3</td>
<td>APDB</td>
</tr>
<tr>
<td>4</td>
<td>APDBCOPY</td>
</tr>
</tbody>
</table>
Creating PDB using existing PDB

SQL> alter pluggable database apdb open read only;
Pluggable database altered.

SQL> create pluggable database apdb2 from apdb;
select con_id, name from V$pdbs;

Pluggable database created.
SQL> select con_id, name from V$pdbs;

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>3</td>
<td>APDB</td>
</tr>
<tr>
<td>4</td>
<td>APDB2</td>
</tr>
</tbody>
</table>

Sangam-2014
Creating PDB using an Unplugged PDB

SQL> alter pluggable database all close;
Pluggable database altered.

SQL> alter pluggable database apdbcpy unplug into 'apdbcpy.xml';
Pluggable database altered.

SQL> drop pluggable database apdbcpy keep datafiles;
Pluggable database dropped.

SQL> create pluggable database apdbcpy using 'apdbcpy.xml';
Pluggable database created.

SQL> select con_id, name from V$pdbs;

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>3</td>
<td>APDB</td>
</tr>
<tr>
<td>4</td>
<td>APDBCPY</td>
</tr>
</tbody>
</table>
Creating PDB using an non-CDB

In non-CDB(12c) database:

SQL> alter database open read only;
Database altered.
SQL> exec dbms_pdb.describe('orcl12.xml');
PL/SQL procedure successfully completed.

In 12c CDB database:

SQL> create pluggable database orcl12 using '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/orcl12.xml';
Pluggable database created.
SQL> alter session set container=orcl12;
Session altered.
SQL>@?/rdbms/admin/noncdb_to_pdb.sql
Dropping a PDB

• **SQL> drop pluggable database apdbcpy including datafiles;**
  
  Pluggable database dropped.

  *or*

• **SQL> alter pluggable database apdbcpy unplug into 'apdbcpy.xml';**
  
  Pluggable database altered.

• **SQL> drop pluggable database apdbcpy including datafiles;**
  
  Pluggable database dropped
PDB Management using SQLDeveloper
PDBSS-Multitenant Self Service App (Beta)

• Self Service application for rapid provisions of PDB’s
• Based over APEX (Application Express) technology
• Easy to use and very feature-rich
• Still in Beta
• Not installed by default! 😞
Installation & Configuration of PDBSS

- Install Oracle database release12c (12.1.0.1 or above) database software
- Create a new Container database (CDB)
- Check the APEX release (required is 4.2.3)
  - SQL> select VERSION_NO from APEX_RELEASE;
    VERSION_NO
    --------------
    4.2.5.00.08   *12.1.0.2
- For 12.1.0.1 software, APEX release is 4.2.0
- If APEX version=4.2.0, apply patch #17347169
- Configure RESTful (Representational State Transfer) web services
  - $cd patch
  - As SYS user from Root container run @apex_rest_config_con.sql
Installation & Configuration of PDBSS\textsuperscript{(contd)}

- When prompted, supply passwords for
  - APEX\_LISTENER
  - APEX\_REST\_PUBLIC\_USER
- Unlock and open APEX\_PUBLIC\_USER
- Download and configure \textbf{Oracle Rest Data Services\textsuperscript{}}(ORDS) (formerly known as APEX Listener)
- Unzip the downloaded file and issue,

\$java –jar ords.war

- Issue an available port number and follow the instructions
- Download the installation zip of PDBSS from \textbf{OTN\textsuperscript{}}(pdbss\_1030.zip)
- Unzip the file and install the PDBSS application using the script \textbf{@pdbss\_ins.sql}
PDBSS is installed!

http://<host>:<port>/ords/f?p=600

Login using ADMIN user
PDB’s Administration Using DBCA

Manage Pluggable Databases

Select an operation that you want to perform in container database:

- Create a Pluggable Database
- Unplug a Pluggable Database
- Delete a Pluggable Database
- Configure a Pluggable Database
What’s New for Multitenants in 12.1.0.2

- PDB **CONTAINERS** clause
- PDB File Placement in OMF using **CREATE_FILE_DEST**
- PDB Tablespace Logging
  - PDB must be opened in **Restricted** mode
- PDB Metadata Clone using **NO DATA** clause
  - Source PDB must NOT contain
    - Index-Organized tables
    - Advanced Queue tables
    - Clustered tables
    - Table clusters
- PDB Subset Cloning Using **USER_TABLESPACES** clause
- PDB State Management across CDB restart using **SAVE/DISCARD STATE** clause
  - New view to check the state- **DBA_PDB_SAVED_STATES**
- PDB Remote Cloning
  - A non-CDB database is added to a CDB database as a PDB container using DB links
Oracle database release 12c has brought up very major changes in almost all the aspects of the database administration.

Multitenant database has been created to solve a lot of issues faced by the DBA’s in the day-to-day management of large data centers.

This was just an introduction to 12c Multi-tenant databases.

For more deeper-dive into the multi-tenancy, bookmark

- http://www.oracle.com/goto/oll

Good luck!!