Oracle Database 12c Multitenant Administration Scenarios

Presented to UNYOUG June 10, 2016

Kevin Gilpin, Advizex
kgilpin@advizex.com

Agenda

- Brief Description and Justification of Multitenant
- CDB/PDB Creation
- Container Users and Security
- CDB/PDB Upgrade
- Backup and Recovery
- Data Guard

ROLTA ADVIZEX

Brief Description of Multitenant

Brief Description and Justification of Multitenant

- Today's "C" buzzword (besides "Cloud") is … *Converged*. Actually, *Converged Infrastructure*, but that would be a buzz*term*, not a buzz*word*. Is there such a thing as a buzzterm? I guess there is now.
- VM usage is hot and will only accelerate (Docker anyone?).
- Finding ways to run many applications and databases on fewer physical servers is a popular pursuit for infrastructure, license and support cost savings.

ROLTA ADVIZEX

Brief Description and Justification of Multitenant

- Converged Infrastructure options for Oracle databases?
  - Run many databases on fewer physical servers (migrate to appliance and/or hypervisor platforms).
  - Converge application schemas from multiple databases into fewer databases.
- Do these options definitively solve the problem of converging the infrastructure?
- In some ways yes, but they are not ideal solutions.
- Alternative?

ROLTA ADVIZEX

Brief Description and Justification of Multitenant

- Enter the Oracle Database Multitenant option.
- The Multitenant option (let's abbreviate it "MT") implements an Oracle database as a *container database ("CDB")*.
- In a CDB, there is a root container and a collection of "sub-containers" that are either a *pluggable database* (PDB) or a *seed pluggable database*.
- The root container is not intended to store application objects and data.
- A PDB is intended to store only application objects and data.

ROLTA ADVIZEX

Brief Description and Justification of Multitenant

- The seed PDB is used as a creation template for PDBs.
- A CDB has all of the familiar components of a non-CDB – SGA, tablespaces (system, sysaux, undo, temp), redo/archived log files, controlfiles.
- A PDB has a system tablespace and application tablespaces.
- The PDB is intended to contain RDBMS-version-independent structures and data. This facilitates unplug/plug upgrade.
- Besides unplug/plug upgrade, there are other features/benefits.

ROLTA ADVIZEX

Brief Description and Justification of Multitenant

- Non-CDB/PDB compatibility guarantee.
- PDB-specific upgrade via unplug/plug/clone (upgrade in minutes).
- PDB-specific security (solves the problem of the “any” privileges in converged schema non-CDB databases).
- PDB-specific recovery (cannot do that with schema convergence).
- PDB-specific resource management (contrast that with converged schema resource management).

ROLTA ADVIZEX

Brief Description and Justification of Multitenant

- The features are great, but note that the MT option does have a licensing cost to fully utilize all MT features (it's a Database option in the Oracle Price List, just like RAC or Partitioning).
- Note this tidbit also (from the 12.1 Oracle Upgrade Guide):
  "Section 8.1.1 Deprecation of Non-CDB Architecture:
  The non-CDB architecture is *deprecated* in Oracle Database 12c, and may be desupported and unavailable in a release after Oracle Database 12c Release 2. Oracle recommends use of the CDB architecture".

ROLTA ADVIZEX

Brief Description and Justification of Multitenant

- Whoa! Huh?!? What?!? The initial reaction might be one of panic and/or a furrowed brow at Oracle and the raising of the question “How can Oracle Corporation do this?”.
- Answer – here’s how:
  - A CDB can be created with one PDB (besides the included seed PDB) with no requirement to license the MT option (i.e., it’s free). This is referred to as the “singletenant configuration.
  - Some MT functionality such as unplug/plug/clone of the single PDB can be used in singletenant without MT licensing.
  - There is a non-CDB/PDB compatibility guarantee.
  - Oracle testing has shown no performance disadvantage of singletenant compared to non-CDB.

ROLTA ADVIZEX

Brief Description and Justification of Multitenant

- OK, I'm sweating less about this now, but why would I do that instead of just running a "regular" non-CDB database as I do now pre-12.x?
- Answer:
  - The benefit of unplug/plug/clone upgrade in minutes or less for singletenant can be used without MT licensing.
  - PDB clone in singletenant has some advantages over the transportable tablespace feature.

ROLTA ADVIZEX

Brief Description and Justification of Multitenant

- Eureka! Multitenant here I come!
- The fine print – Careful planning is still urged when considering this because not all database features are implemented yet in MT.
  - Some flashback features.
  - Data Recovery Advisor.
  - Others – Consult README Guide in doc set.

ROLTA ADVIZEX

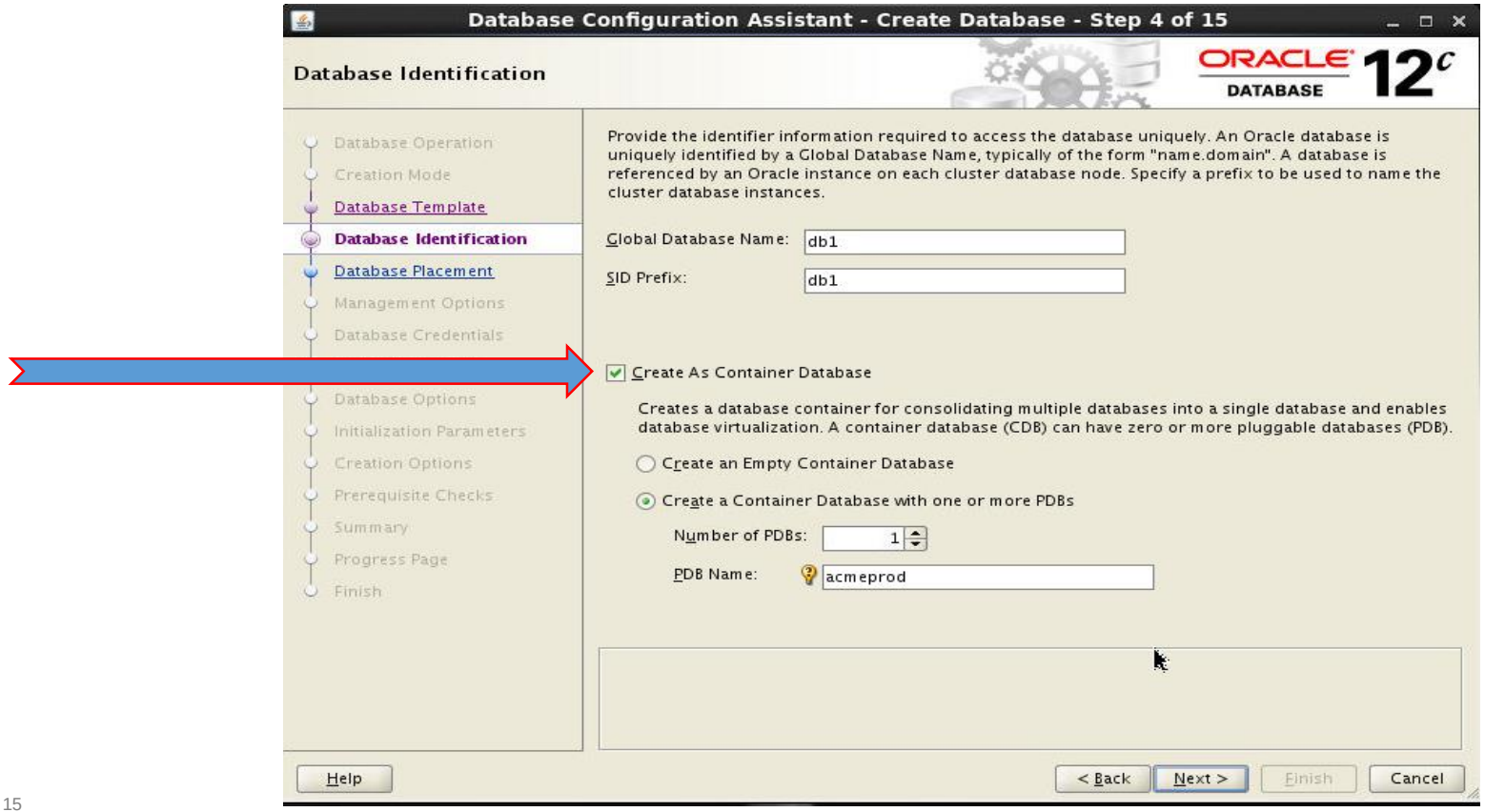
CDB/PDB Creation

CDB/PDB Creation

- The CDB is created first – that gives you a root container and a seed PDB.
- Invoke dbca.
- Choose “Create a database”, Advanced Configuration. After specifying RAC/non-RAC and admin or policy managed, the next page will ask you the database name and offer a checkbox to “Create as a container database”.
- The rest of the dbca dialog is as you would expect for a non-CDB. There is no additional MT configuration in the rest of the dbca.

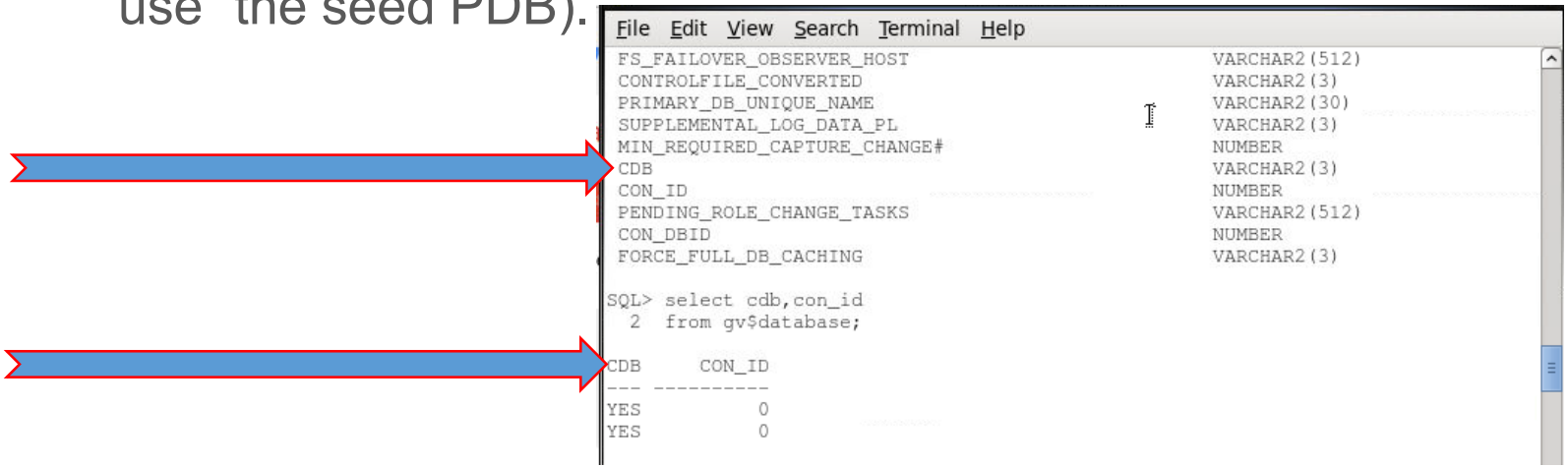
ROLTA ADVIZEX

CDB/PDB Creation



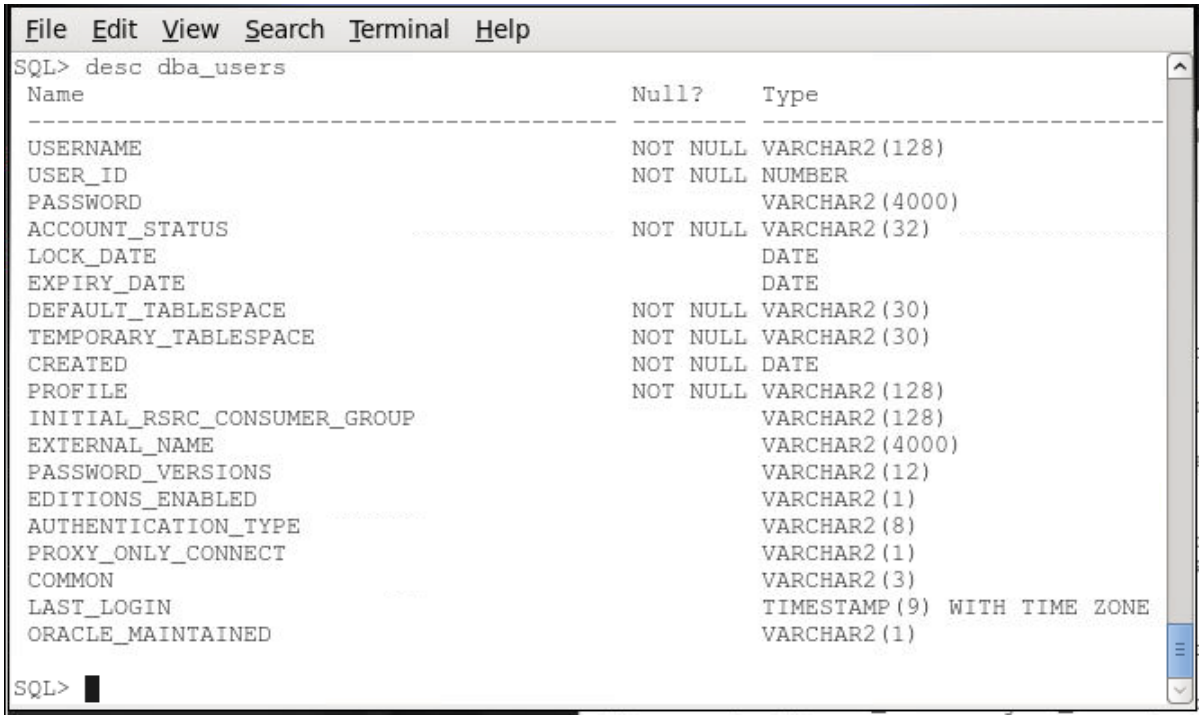
CDB/PDB Creation

- That's it! Presto magic – you have a CDB.
- It has a containerized dictionary, a root container, a seed PDB and, since I specified to create one PDB, a single usable PDB (I mean "usable" to contrast with seed because you cannot "use" the seed PDB).



CDB/PDB Creation

- No con_id column in dba_users.



CDB/PDB Creation

- con_id column in cdb_users.

```
File  Edit  View  Search  Terminal  Help

SQL> desc cdb_users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                  NOT NULL VARCHAR2(128)
 USER_ID                                   NOT NULL NUMBER
 PASSWORD                                           VARCHAR2(4000)
 ACCOUNT_STATUS                            NOT NULL VARCHAR2(32)
 LOCK_DATE                                          DATE
 EXPIRY_DATE                                        DATE
 DEFAULT_TABLESPACE                        NOT NULL VARCHAR2(30)
 TEMPORARY_TABLESPACE                      NOT NULL VARCHAR2(30)
 CREATED                                   NOT NULL DATE
 PROFILE                                   NOT NULL VARCHAR2(128)
 INITIAL_RSRC_CONSUMER_GROUP                        VARCHAR2(128)
 EXTERNAL_NAME                                      VARCHAR2(4000)
 PASSWORD_VERSIONS                                  VARCHAR2(12)
 EDITIONS_ENABLED                                   VARCHAR2(1)
 AUTHENTICATION_TYPE                                VARCHAR2(8)
 PROXY_ONLY_CONNECT                                 VARCHAR2(1)
 COMMON                                             VARCHAR2(3)
 LAST_LOGIN                                         TIMESTAMP(9) WITH TIME ZONE
 ORACLE_MAINTAINED                                  VARCHAR2(1)
 CON_ID                                             NUMBER

SQL>
```

CDB/PDB Creation

- Creating additional PDBs:
  - Create from seed PDB.
  - Clone from another PDB.
  - Plug in a non-CDB.
  - Plug in from unplugged PDB.

ROLTA ADVIZEX

CDB/PDB Creation

- Creating from seed PDB:
  - Connect to the root container (connect system/password or connect system/password@cdbroot).
  - SQL> create pluggable database [options…];
  - Options:
    - Local admin user
    - Storage quota
    - Default tablespace
    - File location options

ROLTA ADVIZEX

CDB/PDB Creation

- Cloning from another PDB:
  - From a PDB in the same CDB.
  - From a PDB in a different CDB.
  - From a non-CDB.
  - Snap clone.

ROLTA ADVIZEX

CDB/PDB Creation

- From a PDB in the same CDB:

  SQL> create pluggable database hrqa from hrprod;

- Note some optional parameters omitted:
  - path_prefix
  - file_name_convert
  - maxsize
  - max_shared_temp_size

ROLTA ADVIZEX

CDB/PDB Creation

- From a PDB in a different CDB:
  - Prerequisites:
    - Create a database link from the target CDB to the source PDB/non-CDB.
    - Source database (PDB/non-CDB) must be open in read only mode.
    - Same database options.
    - Same endianness if cross platform.

ROLTA ADVIZEX

CDB/PDB Creation

- From remote PDB source:

```
SQL> create pluggable database hrqa from
  hrprod@HRREMOTE;
```

- From remote non-CDB source:

```
SQL> create pluggable database hrqa from non$cdb@HRREMOTE;
```

- Note optional clauses omitted from above examples.

ROLTA ADVIZEX

CDB/PDB Creation

- From an unplugged PDB:
- Execute dbms_pdb.describe to create an XML file of the source PDB structure.
- Execute dbms_pdb.check_plug_compatibility in target CDB.
- Execute alter pluggable database [pdbname] unplug into [an xml file]
- Execute create pluggable database [pdbname] as clone using

[thexmlfile] copy;

- Open the PDB in the target CDB.

ROLTA ADVIZEX

CDB/PDB Creation

- Once you have created the PDB, you can connect to it with a username/password@PDB1 connect string.
- You can also connect to the CDB and then execute this:

SQL> alter session set container=PDB1;

ROLTA ADVIZEX

Container Users and Security

Container Users and Security

- There are two types of database users – “common” and “local”.
- A common database user is defined in the root container and has access to all current and *future* PDBs.
- A local user is defined in a particular PDB and has access to only that PDB.
- Common user names must start with the characters “c##”.
- Local user names follow the same naming rules as for non-CDB databases.
- Local user names in PDBs in the same CDB may have the same name.

ROLTA ADVIZEX

Container Users and Security

- User SCOTT in one PDB may have different privilege and role grants than user SCOTT in another PDB in the same CDB.
- Grants to public in one PDB may be different than grants to public in another PDB in the same CDB.
- Contrast this with the schema consolidation method of Converged Infrastructure (can't have two SCOTTs in the same database).
- Role/privilege inheritance on clone operations.
- Common user creation (from the CDB):

```
SQL> create user c##hradmin container=all identified by hrpwd;
```


ROLTA ADVIZEX

CDB/PDB Upgrade

PDB/CDB Upgrade

- A CDB can be upgraded normally. This upgrades all PDBs in the CDB along with the CDB.
- In singletenant or multitenant, a PDB can be upgraded without executing *all* (like the catupgrd script, which is the most time consuming) of the traditional upgrade steps on it by executing an unplug/plug operation.
- Overview of steps:

1. Source PDB: Execute preupgrd.sql and preupgrade_fixups.sql script.
2. Source PDB: Gather dictionary stats.

ROLTA ADVIZEX

PDB/CDB Upgrade

3. Close the source PDB.
4. Execute the unplug command (recall that this creates the xml file).
5. Execute “alter pluggable database drop [pdbname] keep datafiles;

   Actually, recommended to do this later!
6. Execute dbms_pdb.check_plug_compatibility using the xml file.
7. Check pdb_plug_in_violations view.
8. Execute the create pluggable database command using the xml file.
9. Execute “alter pluggable database [pdbname] open upgrade;”.

ROLTA ADVIZEX

PDB/CDB Upgrade

10. Execute perl catctl.pl -c “PDBNAME” –l upgrade_catupgrd.sql
11. Gather optimizer stats on the upgraded PDB.
12. Back up the upgraded PDB.
13. Execute the postupgrade_fixups.sql.
14. Now, drop the source PDB from the source CDB.
15. Note that you can do this for multiple PDBs in one group – use a comma-separated list of PDB names in the Perl script catctl.pl call.

ROLTA ADVIZEX

Backup and Recovery

Backup and Recovery

- It is recommended to back up the whole CDB, but there are times when it will make sense to perform backup (and restore/recovery) of a PDB only.
- While connected to the container in an RMAN session:

  RMAN> backup pluggable database [pdbname1,pdbname2…];
- While connected to the PDB in an RMAN session:

  RMAN> backup database;
- Note that you can only back up archive log files while connected to the CDB!

ROLTA ADVIZEX

Backup and Recovery

- While connected to a CDB:
- RMAN> restore pluggable database [PDBNAME] …
- RMAN> recover pluggable database [PDBNAME] …
- While connected to a PDB:
- RMAN> restore database …
- RMAN> recover database …

ROLTA ADVIZEX

Data Guard

Data Guard

- A PDB may optionally be included in the standby(s) of a CDB – specified at PDB creation time.
- A create or clone PDB operation is automatically replicated by the standby (CDB) database.
- A clone PDB operation in the primary database must be handled similarly to a transportable tablespace operation – the database files must be made available at the standby database.
- Note that logging/nologging operations can occur in PDBs as in non-CDBs. Account for these as in a non-CDB.

ROLTA ADVIZEX

Thank You!

Kevin Gilpin, Advizex
kgilpin@advizex.com