Multitenant Architecture and RAC

By Ahmed Baraka

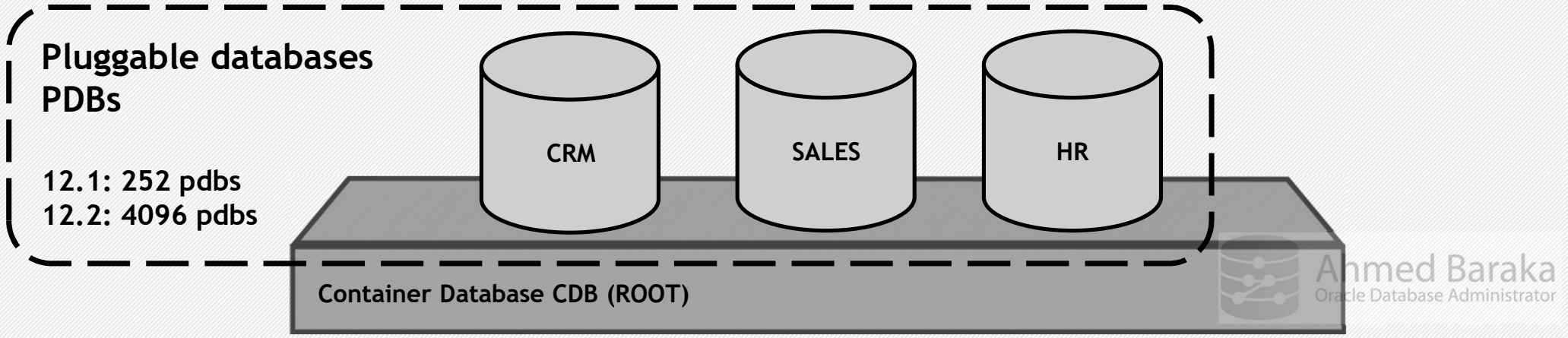
Objectives

In this lecture, you will learn how to perform the following:

- Describe the architecture of Oracle Database Multitenant
- Create a Container Database (CDB)
- Describe the difference between common and local users
- Connect to a CDB and PDBs
- Startup and shutdown CDB and PDBs
- Understand the CDB-level performance views
- Clone a PDB online
- Drop a PDB

What is the Oracle Database Multitenant Architecture?

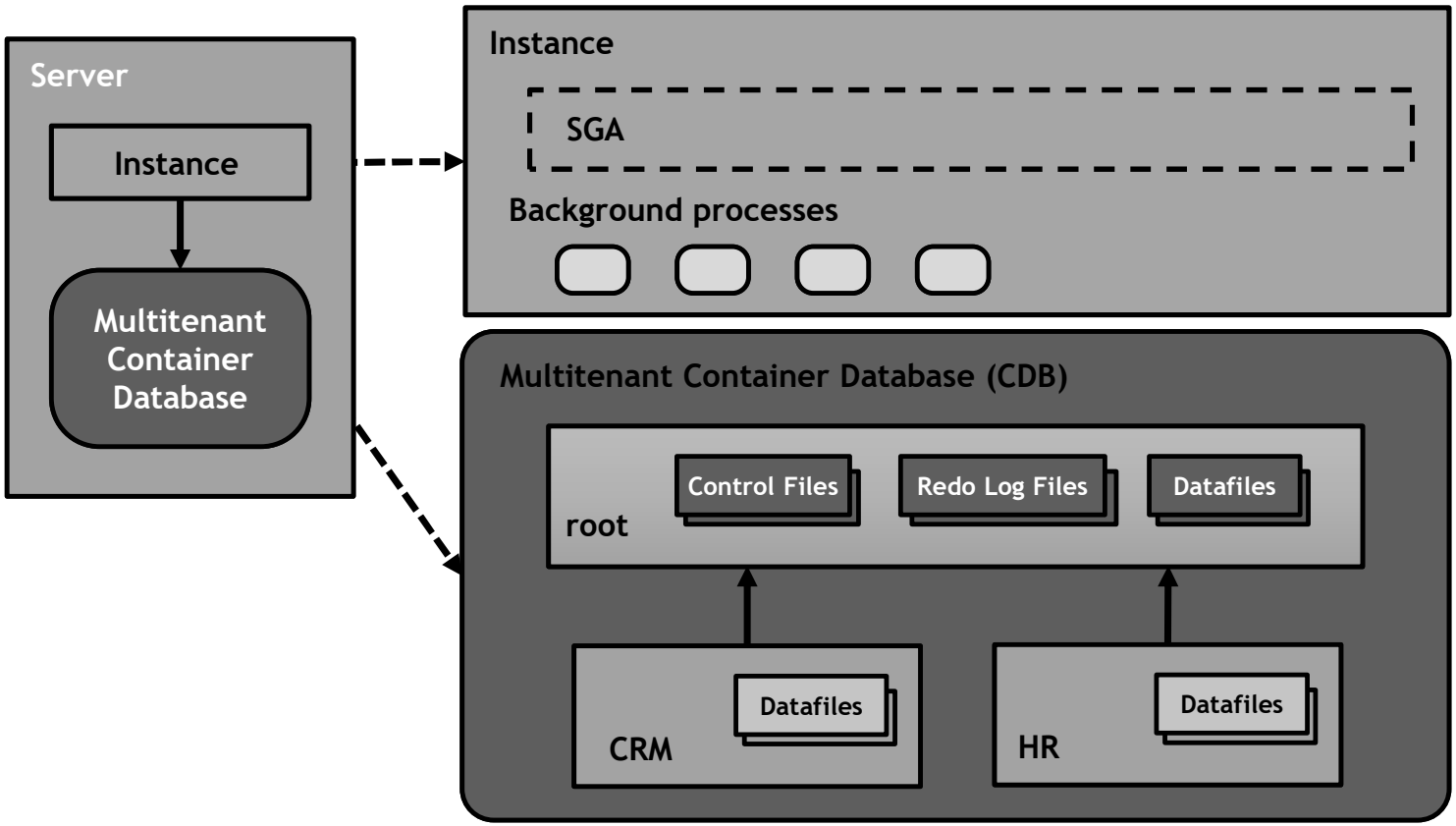
- Consolidate one or more databases into a single housing container database.
- CDB database vs non-CDB database



Oracle Database Multitenant Advantages

- Addresses the consolidation challenges:
  - Resources utilization
  - DBA resource costs
- Existing applications and schema do not need to change
- Tenant isolation
- Easy and efficient database cloning and moving

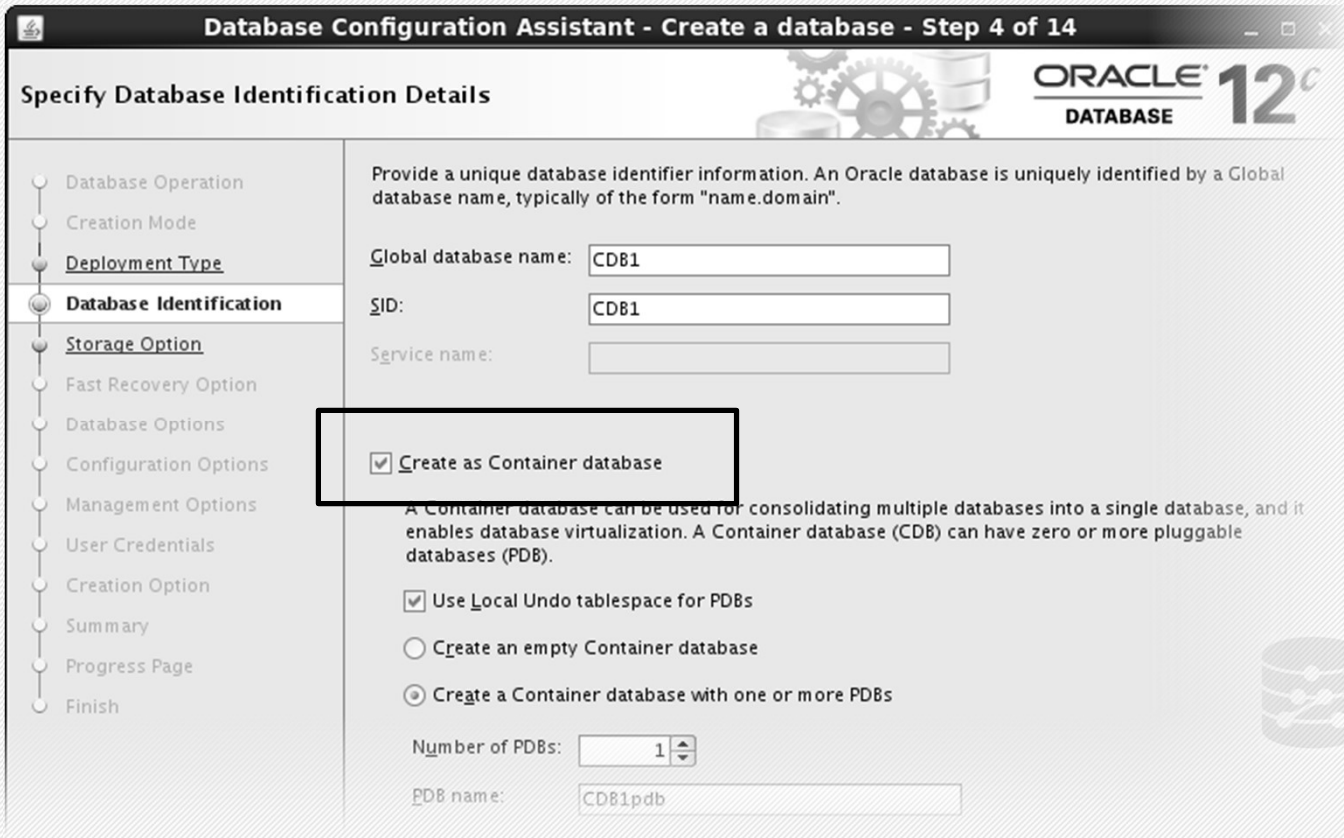
Multitenant Architecture



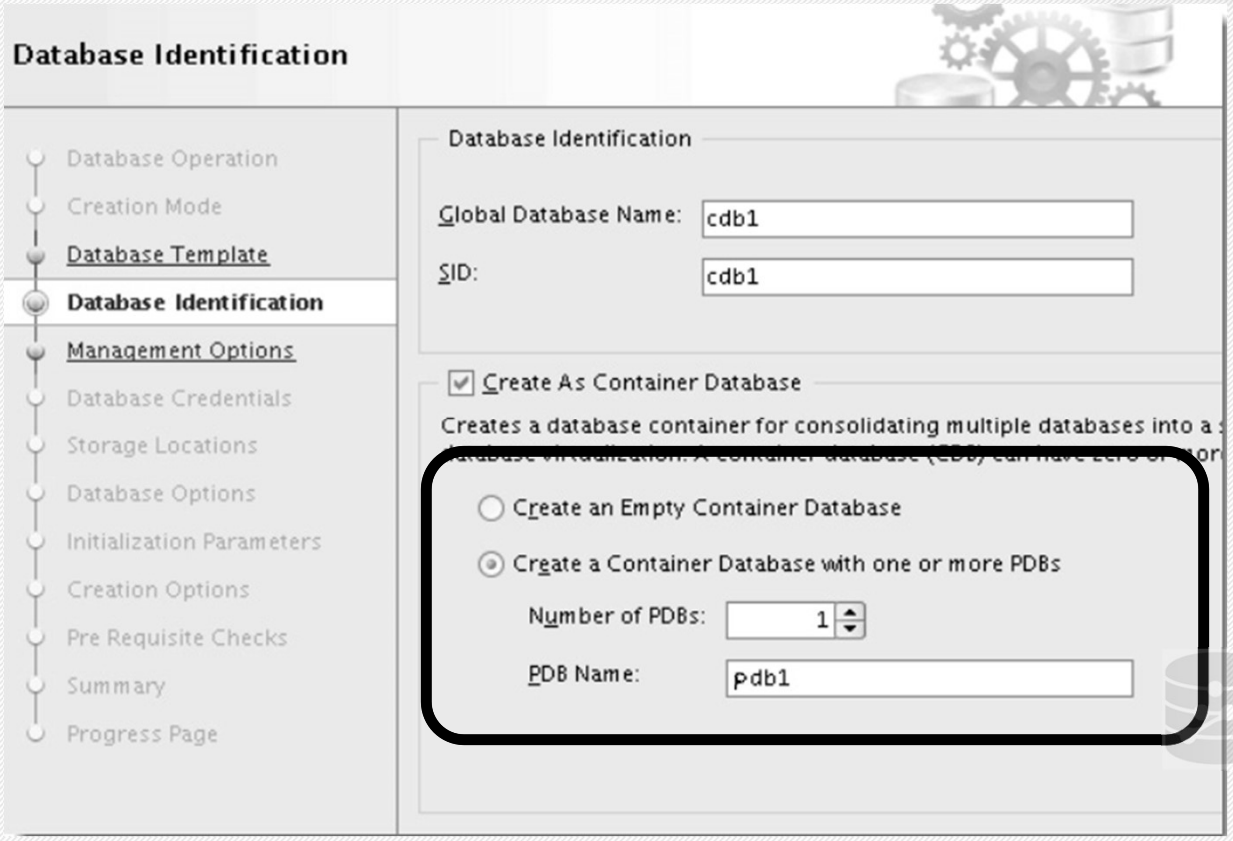
Multitenant Architecture Terms

Term	Definition
Container Database (CDB)	A multitenant container database that houses zero or more pluggable databases.
non-CDB	A database that was created without using the multitenant architecture. All the pre-12c databases are non-CDBs.
Pluggable Database (PDB)	Portable collection of schemas, schema objects, and nonschema objects that appears to an Oracle Net client as a non-CDB.
Root container (CDB$ROOT)	A master set of data files and metadata (data dictionary tables, Oracle-supplied packages, system users) and containing information regarding all containers within a CDB.

Creating a CDB using DBCA



Creating a CDB using DBCA including PDBs



Common and Local Users

- **Common user:**
  - Created in the root
  - Can login to other PDBs in the CDB
  - Usually used for CDB-level administrators
  - SYS and SYSTEM are default common users
- **Local user:**
  - Defined in the container level
  - Can login only to its container
  - Usually used for data-owners and PDB-level administrators

Connecting to a CDB and PDBs

```
SQL> CONNECT / as sysdba
SQL> CONNECT sys@//hostname:1525/CDB1 as sysdba
SQL> CONNECT sys@//hostname:1525/PDBHR as sysdba
SQL> CONNECT scott@//hostname/PDBHR
SQL> SHOW CON_NAME
```

Startup and Shutdown a PDB

- Affecting the current container:

```
ALTER PLUGGABLE DATABASE OPEN [READ ONLY];
ALTER PLUGGABLE DATABASE CLOSE [IMMEDIATE];
```

- Affecting a named container (from root):

```
ALTER PLUGGABLE DATABASE pdb1 OPEN;
ALTER PLUGGABLE DATABASE ALL OPEN;
ALTER PLUGGABLE DATABASE ALL EXCEPT pdb1 OPEN;
```

Startup and Shutdown a PDB (cont)

- Using the **STARTUP** command (from root):

```
STARTUP PLUGGABLE DATABASE pdb1 [OPEN] [READ ONLY]
                                [RESTRICT] [FORCE]
```

- After connecting as PDB admin:

```
STARTUP
SHUTDOWN
```

- To view the state of the PDBs (from root):

```
SELECT NAME, OPEN_MODE, RESTRICTED FROM V$PDBS
```

Startup and Shutdown a PDB in RAC

- To start a PDB in specific instances:

```
ALTER PLUGGABLE DATABASE pdb1 OPEN INSTANCES=('rac1','rac2');
ALTER PLUGGABLE DATABASE pdb1 OPEN INSTANCES=ALL;
ALTER PLUGGABLE DATABASE pdb1 OPEN INSTANCES=ALL EXCEPT ('rac1');
```

- To view the state of the PDBs (from root):

```
SELECT INST_ID, NAME, OPEN_MODE, RESTRICTED FROM GV$PDBS
```

- The same applies in shutting down the PDBs

```
ALTER PLUGGABLE DATABASE pdb1 CLOSE INSTANCES=('rac1','rac2');
```

Startup and Shutdown a CDB

- Traditional command supported:

```
sqlplus / as sysdba
STARTUP
SHUTDOWN
```

- If registered in clusterware or Oracle Restart:

```
srvctl start database -d cdb
srvctl stop database -d cdb
```

Data Dictionary Views in CDB level

View Category within a PDB	Description
USER_***	Objects owned by the current user
ALL_***	Objects accessible by the current user
DBA_***	All the objects in within the current container (root or pdb)
CDB_***	All objects in the CDB (container identified by CON_ID) Examples: CDB_PDBS, CDB_TABLESPACES, CDB_USERS
V$***	SGA accessed by all the containers (container identified by CON_ID)

Cloning a PDB Online

- You can clone a PDB within a CDB online:

```
CREATE PLUGGABLE DATABASE pdb2 FROM pdb1;
ALTER PLUGGABLE DATABASE pdb2 OPEN INSTANCES=ALL;
```

- In RAC, to register the cloned PDB in the clusterware:

```
srvctl add service -db cdbrac -pdb pdb2 -s pdb2srv
 -preferred cdbrac1 -available cdbrac2

srvctl start service -db cdbrac -s pdb2srv
```

Dropping a PDB

- Remove its service:

```
srvctl stop service -db cdbrac -s pdb2srv
srvctl remove service -db cdbrac -s pdb2srv
```

- Drop the PDB including its datafiles:

```
ALTER PLUGGABLE DATABASE  pdb2 CLOSE INSTANCES=ALL;
DROP PLUGGABLE DATABASE  pdb2 INCLUDING DATAFILES;
```

Oracle Multitenant Architecture: Further Learning

- Non-CDB is obsolete
- Information sources:
  - Courses
  - Books
  - Blogs

Summary

In this lecture, you should have learnt how to perform the following:

- Describe the architecture of Oracle Database Multitenant
- Create a Container Database (CDB)
- Describe the difference between common and local users
- Connect to a CDB and PDBs
- Startup and shutdown CDB and PDBs
- Understand the CDB-level performance views
- Clone a PDB online
- Drop a PDB