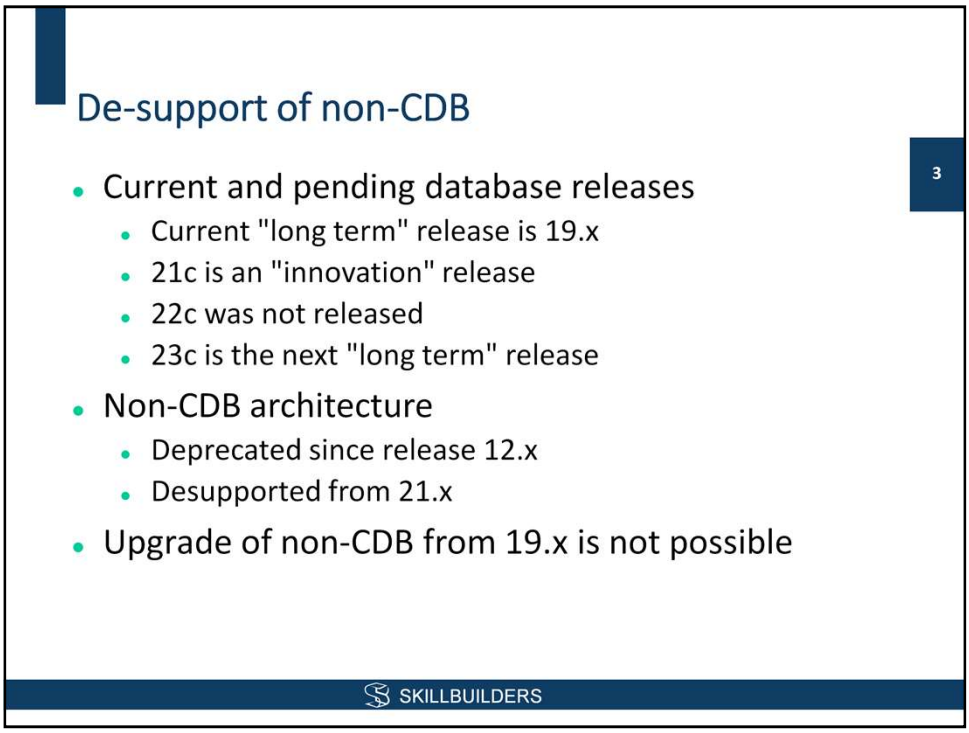
Lesson 8
Some New Features

Objectives

- Appreciate the significance of upgrade for non-CDB
- New features in release 21c
  - Replay Upgrade
  - DbNest
- New features in release 23c
  - Hybrid read-only mode
  - PDB priority

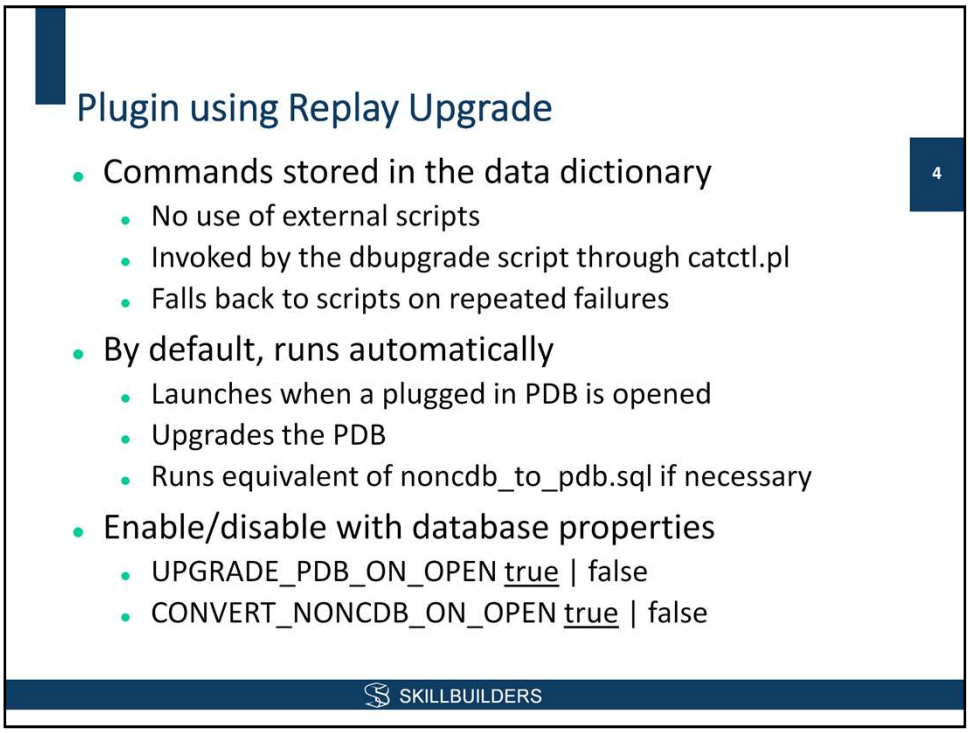


From 12.2, the version numbering and terminology has changed. In principal, there is a new release each year. However, only some of these are "long term support" releases. These releases are guaranteed to have premium support for at least five years and extended support for at least three more years. Interim releases are "innovation" releases, with shorter premium support and no extended support.

It may not be advisable to upgrade to an innovation release, because another upgrade will always be necessary in the not-so-distant future. Even if the new features are essential, it may be that if one is patient they will be back-ported into the current long term support release.

Direct upgrade to 21c is supported from 12.2.0.1, 18.x, and 19.x. Direct upgrade to 23c is supported only from 21.x and 19.x.

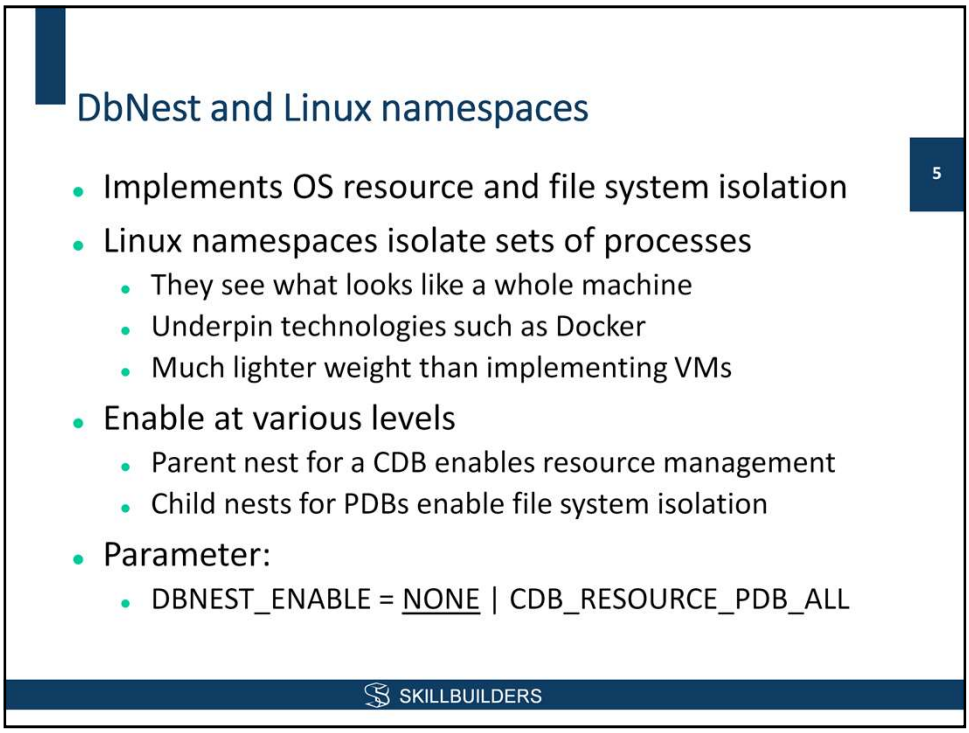
The traditional non-CDB architecture has been deprecated since the first 12c release. From 21c, it is no longer supported. Furthermore, various DB admin tools will no longer work with it. For example, DBCA cannot create a non-CDB. It follows that upgrading a non-CDB from 19.x is not possible: it must be converted into a PDB.



Previously available techniques for launching an upgrade are still available. One can run the catctl.pl script, either directly or through the dbupgrade wrapper script. To do this, open the CDB and PDBs in upgrade mode off the new Oracle Home. However, catctl will in fact upgrade PDBs by running the Replay commands stored in data dictionary following upgrading the root container.

Invoking upgrade with other utilities (AutoUpgrade, srvctl, DBUA, the REST APIs) is also possible, but they all default to using the Replay Upgrade method rather than the older script based method.

Plugging in a non-CDB to a 21c or 23c CDB will always involve an upgrade, since a non-CDB, by definition, must be 19.x or earlier. There will therefore be two steps: upgrade followed by conversion to PDB. Plugin the non-CDB using the usual method. Then, by default, the first time an attempt is made to open the PDB Replay Upgrade will launch. It will upgrade the new PDB, and if the source was a non-CDB it will run the commands necessary to convert the data dictionary to PDB.

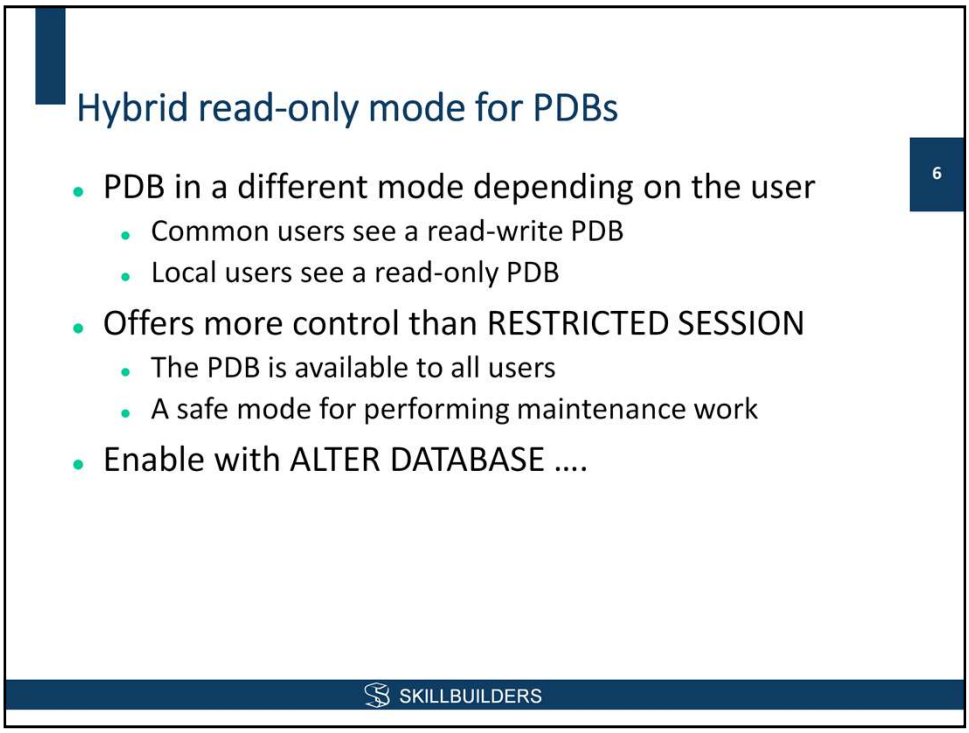


A Linux namespaces restricts a set of processes to seeing only a part of the machine's resources. This can control the amount of CPU available to the namespace, and also present a set of virtual file systems that are visible only to the namespace. Perhaps most importantly, it is absolutely impossible for a process in one namespace to access a process in another.

A dbNest exploits this:

- Operating system resource isolation means that a process belonging to one PDB is not visible to other PDBs or to the CDB root.
- File system isolation controls which directories are visible to each nest, using the chroot mechanism.
- Resource management (CPU and memory) for a nest is based on the availability of the same resources from the parent nest.

DbNest enables a database instance to run in a protected virtualized environment without the need to run instances in separate VMs. This is the parent nest of the CDB. At the next level, processes for PDBs in their child nest have no access to any processes or files assigned to another nest. This provides a last level of defense against a security breach where a malicious user compromises a process.



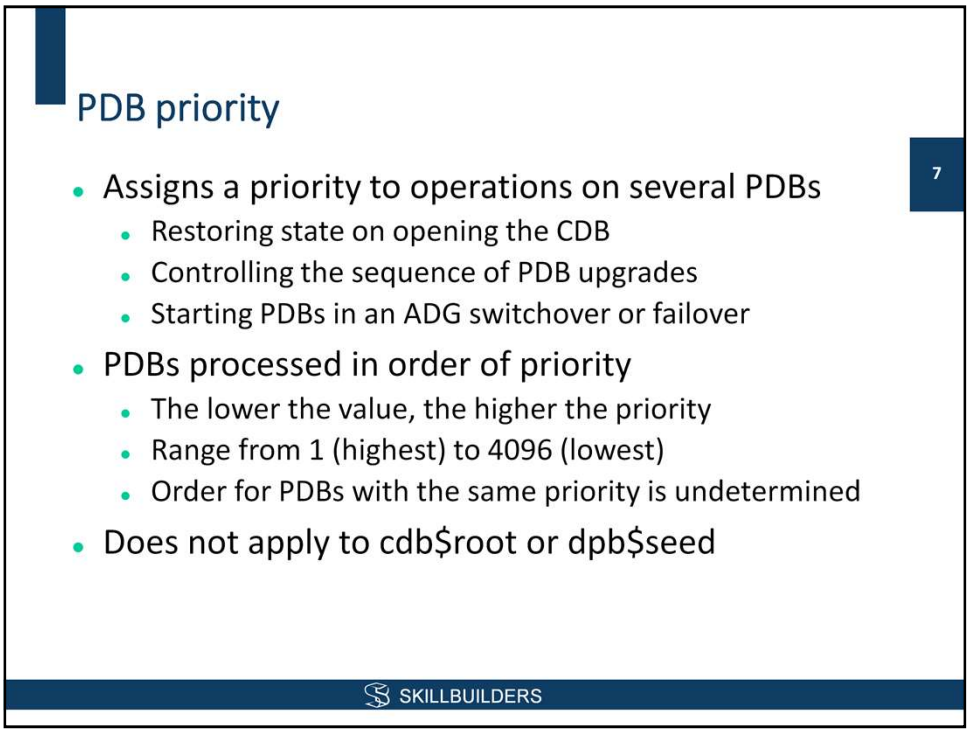
A not uncommon issue in large systems is a need to prevent any DML actions by regular users while maintenance work is being carried out. A simple example would be transactions on a table preventing index creation, or refreshing the content of a PDB used for running queries.

In hybrid read only mode, all users can connect (unlike RESTRICTED SESSION) and can continue to run SELECT type statement (unlike QUIESCE RESTRICTED). However, only common users can run code that will write to the database. In this mode database and application administrators can perform patching or maintenance activities against an open PDB in a without interference from local users, even those that have been granted very high privileges.

To enable:

ALTER PLUGGABLE DATABASE OPEN HYBRID READ ONLY;

The above command will open a closed PB, or change the open mode of an already open PDB.



Without configuring priorities, when executing these commands

ALTER PLUGGABLE DATABASE … OPEN … ;

ALTER PLUGGABLE DATABASE … CLOSE ... ;

If the command is applied to a list of PDBs (or to ALL or to ALL EXCEPT …) the order in which each PDB is affected is non-deterministic. While this is unlikely to be significant for small environments, in a CDB with many PDBs it makes predictable service levels impossible. A workaround is to write scripts that open each PDB in the desired sequence, but that is not practical for a large scale, dynamic, environment.

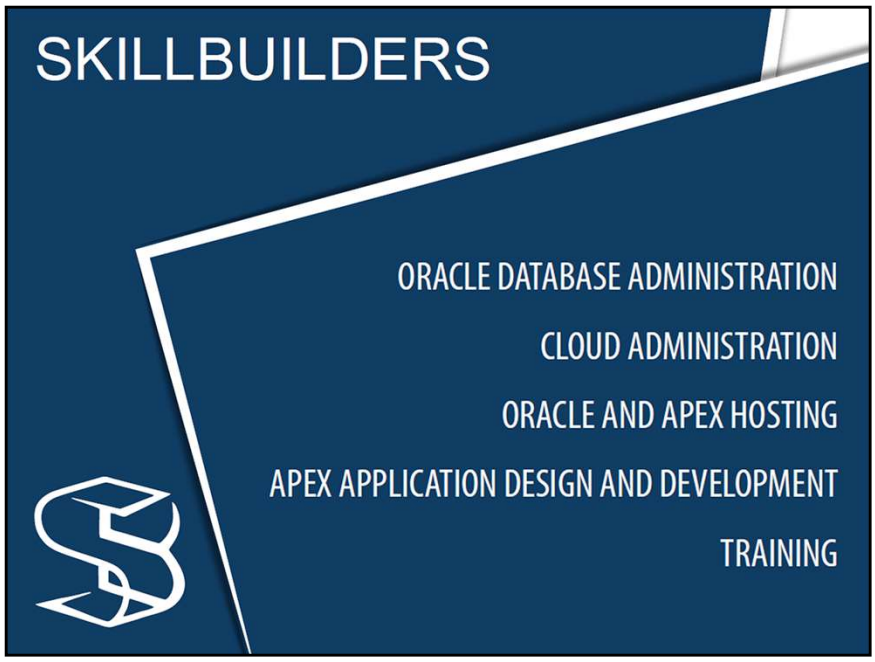
Set priorities with:

ALTER PLUGGABLE DATBASE … PRIORITY n ;

PDBs for which no priority has been set are considered to be the lowest priority.

New Features: Exercise

- Plug a 19c non-CDB into a 21c CDB
- Observe the Replay Upgrade mechanism


SKILLBUILDERS
ORACLE DATABASE ADMINISTRATION
CLOUD ADMINISTRATION
ORACLE AND APEX HOSTING
APEX APPLICATION DESIGN AND DEVELOPMENT
TRAINING