

SE2 goes Enterprise – Hidden Treasures in Oracle 12.2

Mag. Dr. Thomas Petrik
Sphinx IT Consulting
Wien

Keywords

Oracle Database, SE2, Standard Edition, Multitenant, Single Tenant, refreshable PDB, Hot Clone, Relocation, Application Container, LXC

Introduction

The Oracle 12c CDB architecture is not any more an option – in fact it is the default for 12c databases. Of course, we are talking about the single tenant architecture which is available in the Standard Edition (SE2) as well. A closer look into this infrastructure reveals a lot of features which are commonly attributed as “enterprise features” or “multitenant functionality” but can be utilized in an SE2 in nearly the same way: A farm of SE2 installations is operated as an enterprise grade infrastructure. The most prominent example is the 12.2 feature of the refreshable PDB which now introduces an asynchronous but near real time standby solution for all SE2 users without extra costs, third party products or additional effort for individual solutions – it is simply available, just out of the box. Separation of duties, delegation of DBA responsibilities and advanced resource management within a Standard Edition 2 are further topics which can be solved in most simple and elegant ways.

The Data Guard for an SE2

A refreshable PDB can be built between 2 CDBs (where the CDBs may be of any type – SE2 or EE) in two simple steps (with exactly 2 simple statements):

1. create a DB link from the target CDB to the source PDB
2. create the refreshable PDB in the desired refresh mode (automatic or manual)

Having executed these steps the target PDB fetches periodically (secondly, minutely, hourly, etc.) all committed transactions from the source in an incremental way. There is no need any more to transport archivelogs manually. All in all such a refreshable PDB behaves like a physical standby in Maximum Performance mode (if you go for a sub-minute refresh interval). It is also possible to setup multiple destinations for the same source – one for near real time refresh, another for reporting which is open read only and refreshed just once a day, etc.

Building an Online Clone History

Hot Cloning is another 12.2 feature which is clearly not restricted to an Enterprise Edition. It is now possible to create an online clone of some production PDB into another container. By using this method you can build a full clone history of a source. Of course, in case of an SE2 the clones reside in dedicated CDBs while you could use a single CDB if you had licensed an EE with the Multitenancy Option. There is just one thing which does not work in an SE2 environment: snapshot cloning. However, by using OS container technologies like LXC in combination with BTRFS as a copy on write capable filesystem you can also workaround this limitation.

Relocate your PDB Online

Online relocation of a PDB from one container to another is based on the same technology as the refreshable PDB feature. A PDB may be moved from one CDB to another from EE to SE2 and back, from cloud to an on-premises service or vice versa. Be aware that if the PDB is moved between an SE2 and an EE no EE options or features must be used. Furthermore, a clear patching strategy within your 12c environment is key for a successful operation in this way. However, online relocation may also be used for upgrade scenarios, since the PDB can be implicitly patched during this procedure. Online relocation may as well be used as a replacement for VM-based live migration scenarios which are nowadays mostly forbidden in many virtualized environments due to license restrictions. However, it should be noted that such a relocation is never 100% online but depending on the method used it may be almost online.

3 relocation scenarios can be distinguished:

1. Source and target CDB share the same listener: In this case all new connections are directed immediately to the new CDB.
2. Source and target CDB have 2 separate listeners but can be registered in the same listener network: Also in this case all new connections are directed immediately to the new CDB.
3. The relocation is done in maximum availability mode: a so called tombstone PDB remains after the successful relocation as a place holder and all incoming connections are forwarded to the relocated PDB. This tombstone PDB is basically a proxy PDB which still has SYSTEM and SYSAUX tablespaces which are fully in sync with the relocated PDB. This scenario turns out to be rather useful if you want to redirect your applications completely transparently into the cloud without changing all firewall routes.

In any case current connections are killed during the relocation process.

The Application DBA: Separation of Duties I

Working in a CDB/PDB environment means that the system is automatically enabled for delegation scenarios even if the configuration is single tenant only. Lockdown profiles are the key component in 12.2 in order to restrict a PDB-DBA (i.e. application DBA) although he may have been granted the sysdba-privilege. The application DBA should not access the OS or change any external structure by running external jobs under the credentials of the Oracle user or by defining directories somewhere on the machine. Also the directory structure for database files should be maintained by the head DBA only. Parameters like `create_file_dest` or `patch_prefix` on PDB level now enable the head DBA to restrict the PDB to dedicated filesystem structures. Lockdown profiles restrict the application DBA from misusing commands like “alter system” without the need to completely revoke the whole alter system privilege. Even if the application DBA is granted the sysdba role with a proper lockdown profile he cannot work around these restrictions.

The Application DBA: Separation of Duties II

An application DBA should have full control over the PDB he is managing. He may be responsible for creating users, granting privileges and managing resources but he should not be able to change the structure of the application itself especially if the application deployment is automated and centrally managed. Application containers are a great 12.2 feature for controlling mass deployments in multitenant environments but can also be successfully used in SE2 architectures. A closer look into the license terms reveals that an SE2 may have 1 normal PDB, 1 proxy PDB and 1 application root. This means that even in an SE2 one application root PDB may be deployed together with exactly one application PDB. Of course, such an architecture is not meant to be used for mass deployments but it serves in a perfect way as a basis for a controlled and reproducible deployment scenario managed by the

head DBA while the application DBA cannot interfere with the application structure (and even common data) while he still has full control over the application PDB itself.

Conclusion

Using the Oracle 12.2 CDB/PDB architecture in a single tenant configuration within an SE2 environment opens up a wide range of methods which have been originally designed for enterprise infrastructures. The availability of these features for Standard Edition 2 environments puts new light on a broad range of use cases which can now be covered with standard technologies instead of facing the need to implement third party products or other proprietary solutions.

Contact:

Mag. Dr. Thomas Petrik

Sphinx IT Consulting

Aspernbrückengasse 2

A-1020 Wien

Phone: +43 664 155 8304

Fax: +43 (1) 599 31-99

E-Mail Thomas.Petrik@sphinx.at

Internet: www.sphinx.at