

Oracle Real Application Clusters (RAC) 12c Best Practices

Markus Michalewicz
Oracle
Redwood Shores, CA, USA

Keywords: Oracle RAC 12c, Grid Infrastructure, Best Practices

Introduction

It is no secret that standardization can lead to significant cost savings by re-utilizing the same parts as well as by reducing individual workflows in favor of standardized workflows that are executed repeatedly. The same ideas apply to the installation and management of Oracle Real Application Clusters (RAC) and ensuring that best practices are applied to a given system. The installation flow of the Oracle Universal Installer (OUI) which is used to install the Oracle Grid Infrastructure as well as an Oracle RAC-enabled database home attempts to cover most of the common installation scenarios assuming that best practices during the system setup (hardware and software) are followed. This article describes the steps that will allow OUI to work efficiently. It will point to the critical steps that need special considerations depending on the future use of the system and it will list the post-installation steps that may be required.

Standardization and Oracle RAC

Oracle RAC has become a synonym for a stack that includes more than just an Oracle RAC database. When it comes to standardization however, distinguishing the various layers in the stack allows for further optimization. The first component that needs to be installed in order to use an Oracle RAC database is the Oracle Grid Infrastructure (GI).

Oracle Grid Infrastructure comes in two variants 1) Oracle Grid Infrastructure for Standalone Deployment, 2) Oracle Grid Infrastructure for a Cluster. For both deployment types the same binaries are used. It is the configuration that decides whether an Oracle RAC database can be installed or whether Oracle GI can only be used to host an Oracle Single Instance Database.

Deploying an Oracle Grid Infrastructure in a standalone environment using a standalone deployment enables an Oracle Single Instance Database to use Oracle Automatic Storage Management (ASM) for storage management. Using standard tools such as the Oracle Database Configuration Assistant (DBCA) or the OUI will also allow the Single Instance Oracle Database to automatically benefit from the Oracle Restart feature. The Oracle Restart feature provides the ability to start, stop and restart an Oracle Database or Oracle ASM instance (if used) on the same server as part of the management provided by Oracle Clusterware, which is an integral part of Oracle Grid Infrastructure.

The Oracle Grid Infrastructure for Standalone Deployment is therefore designed for standalone environments that are unlikely to require horizontal growth (adding more servers to a system) in the foreseeable future. If there is a chance that horizontal growth is required, an Oracle Grid Infrastructure for a Cluster deployment should be used.

Oracle Grid Infrastructure for a Cluster requires a cluster setup on hardware level and thereby slightly increases the pre-installation requirements in terms of storage and network setup. However, realistically, those requirements are mostly met by most of today's systems by default. All that may be required is a subtle change in how these systems are configured.

Oracle Grid Infrastructure for a Cluster requires shared storage (SAN or NAS devices) as well as a so called interconnect (dedicated connection) between the servers in addition to the public network (the network that is used to connect to the server from a client). Most of today's systems probably use a SAN / NAS for non-local storage. Hence, meeting shared storage requirements should rather be a matter of software configuration, since the physical setup has already been established. As a matter of fact, systems that are set up to either function as a failover cluster or use a virtualization solution that allows for inter-server failover or migration of virtual instances between servers are most likely already prepared to host an Oracle Grid Infrastructure for a Cluster deployment.

It is therefore Oracle's recommendation to standardize on a cluster setup for any deployment; may it be a Single Instance Database or an Oracle RAC Database. This recommendation is the standard on Oracle's Engineered Systems. The Oracle Database Appliance (ODA) as well as Oracle Exadata systems use a 2-node cluster setup as their minimum configuration. It needs to be noted in this context, that the cluster setup itself does not impose any additional licensing, as Oracle GI is free of charge. The licensing of such a system is determined by the software being hosted in the cluster.

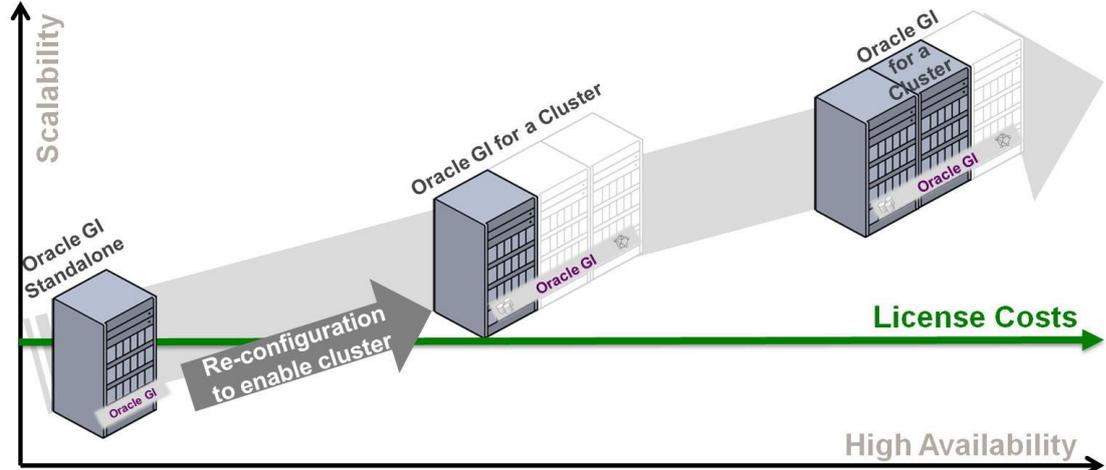


Illustration 1: Standardization on Clusters

Standardizing on a cluster allows for scalability in future and immediately improves High Availability (HA) as permitted by such architectures. With respect to an Oracle Grid Infrastructure deployment and despite the fact that the Oracle Grid Infrastructure for Standalone Deployments and the Oracle Grid Infrastructure for a Cluster use the same binaries, it is recommended to standardize on the cluster setup (even on a 1-node cluster). The main reason is that changing from a standalone setup to a clustered configuration requires a re-configuration as shown in illustration 1.

Whether or not the Oracle Database can immediately benefit from the cluster setup is determined by the database type that is deployed in the cluster. Potential database types are: 1) an Oracle Single Instance Database, 2) an Oracle RAC One Node Database, 3) an Oracle RAC Database.

Oracle RAC databases, either Oracle RAC One Node or Oracle RAC provide the highest level of flexibility¹. Oracle RAC One Node utilizes multiple features that make it the ideal standard database deployment. From an application point of view, Oracle RAC One Node appears – for all practical purposes – as an Oracle Single Instance Database towards the application. Therefore, there is hardly ever any need to certify an application with Oracle RAC One Node explicitly, as long as the application is certified to run against an Oracle Database.

From an administration point of view, Oracle RAC One Node shares and benefits from the same architecture that has been used for Oracle RAC for more than a decade. This design allows Oracle RAC One Node to use the Online Database Relocation-feature for uninterrupted database service in case of planned downtime, typically used during patching the OS or the database home for example. A Single Instance database, even when deployed on a cluster, needs to be taken down in this case.

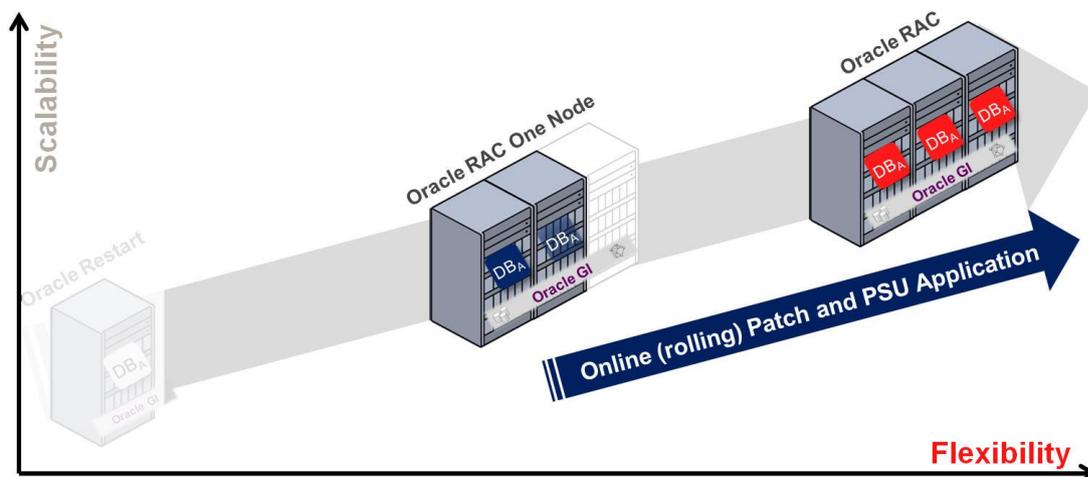


Illustration 2: Standardization on Oracle RAC

Preparation of the System and the Servers

Assuming that a standardized cluster setup is used, the main preparation task will be to set up the network connections as well as the storage infrastructure. In case of Oracle RAC, the network requirements are as such that an interconnect (a dedicated connection between the servers) as well as a certain setup for the public network need to be configured, which includes providing a number of IP addresses. The official Oracle documentation will provide detailed information on this matter. Similar attention needs to be paid to the setup of the storage. A shared storage setup either based on a NAS or SAN needs to be provided in a suitable manner and following the guidelines outlined in the official Oracle documentation.

As soon as the hardware setup is put into place, setting up the Operating System (OS) is a rather simple task and utilizing the appropriate tools will significantly simplify the deployment of an Oracle RAC cluster. For the purpose of this paper, an Oracle Linux based cluster is assumed, in which case, additional tools provided by Oracle, such as the pre-install packages, can be used.

¹ For more information regarding Oracle RAC One Node and Oracle RAC, see <http://www.oracle.com/goto/rac>, especially: <http://www.oracle.com/technetwork/products/clustering/rac-one-node-wp-12c-1896130.pdf> and <http://www.oracle.com/technetwork/products/clustering/rac-wp-12c-1896129.pdf>

On Linux, Oracle provides pre-install-packages, one for Oracle Database 11g Rel. 2 and one for Oracle Database 12c, which can be downloaded using yum (if configured) and are named accordingly:

```
[root@dasher Desktop]# yum list oracle-*
oracle-rdbms-server-11gR2-preinstall.x86_64 1.0-7.el6          ol6_latest
oracle-rdbms-server-12cR1-preinstall.x86_64 1.0-8.el6          ol6_latest
```

The pre-installation packages will configure the OS to host an Oracle Single Instance database of the respective version. The deployment of the packages will configure OS groups and user structures as well as certain Kernel parameters in accordance to Oracle Database Single Instance best practices. As these best practices do not include Oracle Grid Infrastructure, deploying those packages is only recommended as a first step; additional steps need to be performed to optimize the OS for the deployment of an Oracle Grid Infrastructure for a Cluster configuration. Those additional steps include, but are not limited to:

- Adding a “Grid” software owner and group as required
- Creating Oracle Base / Oracle Home structures for Oracle GI
- Check / enable Huge Pages as required²

While the Oracle RAC installation is designed to integrate best practices into the installation flow, spending a bit more time on the preparation of the system can go a long way. In order to set up a system as homogenous as possible, cloning the OS once the first server has been configured properly can simplify the deployment.

Regardless of the exact deployment model for the OS, choosing an appropriate hostname for each server to be deployed in the cluster is important, as changing the hostname is not supported once Oracle GI is configured. A hostname-change will require a re-configuration of the cluster stack on this server. Each server in the cluster needs to maintain a unique hostname; ideally all lowercase hostnames are used (not required by design, but due to BUGs by experience). It is arguable whether or not uniqueness should be enforced by numeric values. This paper recommends using a name-scheme for hostnames, flexible enough to allow for cluster expansion in multiple ways. For the purpose of this paper, a 4-node cluster with the server names “dasher, dancer, comet and vixen” is used.

Installation of Oracle Grid Infrastructure 12c

Oracle has spent the last two releases of Oracle Grid Infrastructure improving and simplifying the installation of Oracle GI using the Oracle Universal Installer. As a consequence, OUI offers various options to deploy the Oracle GI as well as the Oracle Database software, of which all consider best practices as part of the installation flow.

Installation options are provided to allow for OUI to cover most of the common deployment and installation use cases, while best practices are implemented in various ways; one way is by means of the “pre-selection” or “default”. Although, OUI offers options, in each dialog, a pre-selection is made and unless user information is required, this pre-selection leads to a recommended deployment.

² Using Huge Pages is a standing recommendation on high performance Oracle RAC systems. Configuring the system for the use of Huge Pages can be performed either as part of the OS setup or later. More information:

- My Oracle Support note 361323.1 – HugePages on Linux: What It Is... and What It Is Not...
- My Oracle Support note 401749.1 – Shell Script to Calculate ... Values for Linux HugePages

Example: using the OUI to install Oracle Grid Infrastructure 12c and retaining the pre-selection in the first four dialogs that appear after the OUI has been started will lead to an Oracle GI for a Cluster Deployment (recommendation), using a Standard Cluster (as opposed to an Oracle Flex Cluster, discussed later in this paper) and a Typical Installation. With the exception of the “Typical Installation”, this is the recommended deployment for Oracle GI 12c.

Why Standard Cluster?

In order to understand why a “Standard Cluster” deployment is recommended, it first needs to be considered what additional benefits Oracle Flex Cluster provides. Oracle Clusterware, in its original design, is a tightly coupled clustering solution which requires shared storage and shared network between all servers in the cluster. This deployment model is referred to as “Standard Cluster” with Oracle RAC 12c going forward. This model is useful when Oracle Clusterware is installed for the use with Oracle RAC Databases. In this case, a tightly coupled cluster deployment is required to preserve data integrity of multi-instance, shared cache and clustered databases.

Oracle Flex Cluster relaxes the tightly coupled nature of Oracle Clusterware and permits the collocation of applications and the Oracle Database in a common Oracle Flex Cluster. In the Oracle Flex Cluster, the database tier, composed of Hub-servers, preserves the tightly coupled configuration, while the application tier, composed of Leaf-servers, is loosely coupled and does not require shared storage and peer-to-peer communication with other servers in the cluster.

This innovative tightly coupled/loosely coupled clustering solution provides far greater scalability without compromising the availability and resource management expected in high performance, and very large scale consolidation environments. In the initial release of Oracle Grid Infrastructure 12c Oracle Flex Cluster is designed for collocating applications and the Oracle Database in a common cluster using the Hub-Leaf model described above³.

The collocation of applications and the database in a common cluster offers ease of management between application components and the database, where start/stop order-dependencies, and placement may be defined and automated. Specifically, applications may depend on components outside of the respective application management domain.

However, as the primary use cases for an Oracle Flex Cluster with Oracle Grid Infrastructure 12c Release 1 are outside of the database domain and because a conversion from an Oracle Flex Cluster back to a Standard Cluster is not supported, the default deployment remains to be a “Standard Cluster” deployment with this release. It needs to be noted in this context that a conversion from a Standard Cluster to a Flex Cluster can be performed at any time and an all-Hub Flex Cluster virtually behaves like a Standard Cluster using Oracle Flex ASM.

Why not Typical Installation, but Advanced Installation?

While the installation default as defined above (based on pre-selection) will foresee to perform a “Typical Installation” of Oracle Grid Infrastructure, this paper deviates from this recommendation and suggests to perform an “Advanced Installation”, as it allows to perform an installation suitable for the most common configurations.

³ For more information regarding Oracle Flex Cluster, see:

<http://www.oracle.com/technetwork/products/clusterware/overview/oracle-clusterware-12c-overview-1969750.pdf>

However, the main reason for recommending the Advanced Installation option is the choice to install Oracle Flex ASM as well as the Oracle Grid Infrastructure Management Repository (GIMR), which are both new features available with Oracle Grid Infrastructure 12c.

Oracle Flex ASM is a new Oracle Automatic Storage Management (ASM) deployment model that increases database instance availability and reduces the Oracle ASM related resource consumption. Oracle Flex ASM facilitates cluster based database consolidation, as it ensures that Oracle Database 12c instances running on a particular server will continue to operate, should the Oracle Flex ASM instance on that server fail.

The Grid Infrastructure Management Repository is a Single Instance Oracle Database 12c which will be installed on one of the nodes in the cluster. It is managed as a failover database and contains the Cluster Health Monitor (CHM) data. The GIMR is stored in the first ASM disk group that is created as part of the configuration. In addition, Quality of Service Management (QoS) required resources are added to the cluster configuration (e.g. OC4J) as part of the GIMR creation. Therefore and because creating the GIMR as a post-installation step requires additional steps and is currently not supported, it is strongly recommended to configure the GIMR upfront.

In addition, the Advanced Installation option allows for making additional choices regarding:

- SCAN and Cluster naming
- GNS configuration
- Network Interface Usage

For those entities the recommendations are simple and summarized as follows:

- SCAN⁴ remains to be required for the cluster
 - Choose cluster-specific unique SCAN names
 - OUI will try to enforce it as much as possible
- Choose to “Configure GNS” if:
 - You want to use dynamic IP management
 - DHCP assigned / IPv6 Stateless Address Auto configuration for client connectivity
 - In this case, GNS requirements remain as with previous version for standard cluster
 - You chose to install an Oracle Flex Cluster
- On the “Network Interface Usage” dialog, make sure to
 - Use the “Redundant Interconnect Usage Feature” by selecting multiple interfaces from different subnets for private communication. If Flex ASM is used, align the ASM network with the private interconnect.

Prerequisite Checks – *The Best Practices Guidelines Incorporated into OUI*

Oracle has spent the last two releases of Oracle Grid Infrastructure improving and simplifying the installation of Oracle GI using the Oracle Universal Installer. The main result of these efforts is the integration of the so called “Prerequisite Checks” performed as part of the Oracle GI installation:

⁴ For more information on SCAN, see:

<http://www.oracle.com/technetwork/products/clustering/overview/scan-129069.pdf>

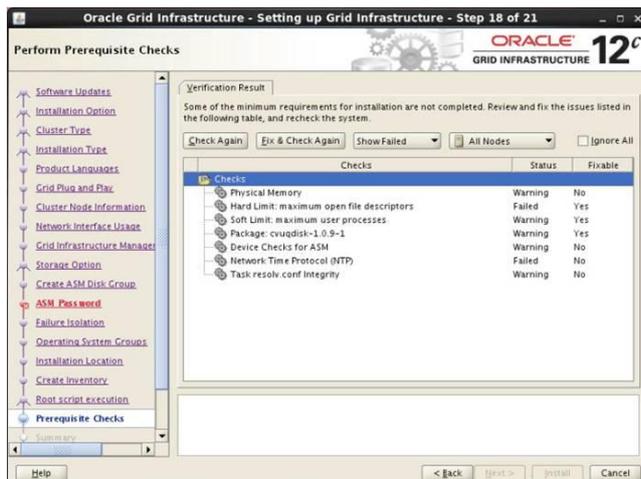


Illustration 3: Prerequisite Checks

The Prerequisite Checks performed as part of the Oracle GI installation attempt to ensure that the system is configured properly to host an Oracle GI for a cluster installation and in extension an Oracle Real Application Clusters database. Following the various instructions provided as part of the dialog is vital for a successful cluster operation.

Cluster Verification Utility (CVU), will be started by OUI in the background and will check the system, providing detailed feedback on the state of the servers. For each negative check result, it will list the check performed, its status and whether or not the issue is fixable by CVU, as shown in illustration 3. The status can vary between “Failed” and “Warning”, while a particular issue is either “fixable” (Yes) or not (No).

While all checks listed in this dialog can be ignored (checking the “Ignore All” option in the right top corner as shown in illustration 3), it is strongly recommended to tackle all issues prior to proceeding with the installation and configuration of an Oracle RAC Database.

One check that was introduced with Oracle GI 12c is the check for physical memory, which tries to ensure that at least 4GB of physical memory are available on the server and returns negatively, if the server does not provide this amount of memory. This check does not intend to indicate that Oracle GI will use this amount of memory. It attempts to ensure a stable environment once the installation is complete. Memory pressure has proven to be one of the main reasons for server failures.

Depending on the choices made during the Oracle GI installation, up to three pre-configured Oracle instances (the GIMR management database, an Oracle ASM instance and an APX instance used for ACFS when Flex ASM is enabled) could be running on a given server in the cluster (most likely the one, which was used to install the stack) in addition to the Oracle Database instances that is are meant to be hosted on the cluster.

That said, the main reason for requiring 4GB of physical memory may not be so obvious: shared memory. In Linux shared memory is managed as a tmpfs (mounted under /dev/shm typically). The default size for this tmpfs is roughly 50% of the physical memory available on the server, which on a 4GB server means approximately 2GB. These 2GB will just be enough to run the three preconfigured instances plus a small database instance in addition. Less than 4GB of physical memory may lead to instances not being able to start, unless these values are manually adapted.

Installation of the Database Home(s)

Once the Oracle Grid Infrastructure is installed, the database home deployment can be tackled. This is a comparably easy task, especially when a “Software Only” installation is performed (step 3 into the OUI installer flow for the database home). This is the only deviation recommended from the “default flow” which for step 4 should foresee to deploy an Oracle Real Application Clusters Database (now that Oracle Grid Infrastructure in the clustered version is used).

Step 5 should then be used to deploy the database home on all nodes in the cluster, which is particularly useful when policy-managed databases are meant to be used. Do not forget to establish the SSH connectivity one more time, assuming that the Oracle Grid Infrastructure installation was performed using a different user.

In step 7, an Oracle Enterprise Edition is chosen for this paper, while Oracle RAC functionality is also available in the Standard Edition. However, the Oracle Enterprise Edition provides additional features that might be useful in an Oracle RAC environment and is therefore the recommended choice. The subsequent steps should be self-explanatory.

Before Creating the Database

Before creating the database, it is recommended to review and adjust the current configuration, which by now should run an Oracle Grid Infrastructure 12c-based cluster and should have the Oracle database homes provisioned on either a per node basis (the default) or on a shared files system such as Oracle ACFS, if a respective manual configuration, not covered in this paper, was performed prior to deploying the database homes.

Adaption of the current configuration before proceeding is most likely required to adjust the number of Oracle ASM disk groups and / or to add Server Pools, if policy-managed databases are meant to be used on this cluster. Unfortunately, the Oracle Grid Infrastructure installation flow only allows for creating one Oracle ASM disk group during the initial deployment and configuration, limiting the options that can be configured for this disk group.

The recommendation is to configure the Oracle ASM disk group created during the initial deployment of Oracle Grid Infrastructure assuming that this disk group is being used as the “infrastructure disk group” (called INFRA therefore), hosting the Voting Files as well as the Oracle Cluster Registry (OCR). Normal redundancy is recommended for this disk group, unless a sufficient level of redundancy can be provided externally. Note in this context that the INFRA disk group will also be used for the GIMR management DB by default and hence the size requirements for this disk group have increased with Oracle GI 12c, especially when the GIMR is deployed as recommended. Enhancement requests have been filed to improve flexibility in this area in future.

Oracle RAC Database Creation using DBCA

Similar to the approach taken by the OUI, the DBCA has a default flow and going with the pre-selection in steps one to four would lead to a policy-managed, Oracle RAC-based, non-CDB Oracle Database, at least assuming that the DBCA is started from an Oracle RAC-enabled database home.

The only deviation from this default should be the choice regarding the non-CDB. With Oracle Database 12c Oracle introduced Oracle Multitenant, which is a new option to the Oracle Database 12c Enterprise Edition that helps customers reduce IT costs by simplifying consolidation, provisioning, upgrades, and more. It is based on a new architecture that allows a multitenant container database to

hold many pluggable databases (PDBs). The idea is that an existing database can be simply adopted, with no changes in the application tier, as a pluggable database. In this architecture, Oracle RAC provides the local high availability that is required when consolidating various business critical applications on one system.

When using PDBs with Oracle RAC, the multitenant container database (CDB) is based on Oracle RAC. Each pluggable database can be made available on either every instance of the RAC CDB or a subset of instances. In any case, access to and management of the PDBs are regulated using Dynamic Database Services, which will also be used by applications to connect to the respective PDB – as they would in a Single Instance Oracle Database using Oracle Net Services for connectivity.

The remaining steps are rather self explanatory and the official Oracle documentation can provide additional details as needed. Attention needs to be paid to the configuration of the disk groups used by the database, as DBCA does not automatically enforce separating the data and Fast Recovery Area (FRA) disk group depending on the configuration used on the system.

Milestone – What has been configured so far?

Following the steps outlined in this paper, the installation of the new Oracle RAC 12c system has reached a critical milestone with Oracle Grid Infrastructure 12c installed and configured and at least one database created. Depending on the choices made during the installation process, the system at this point in time could look like the one outlined in illustration 4.

The system outlined in illustration 4 consists of four nodes (dasher, dancer, comet and vixen), of which all use Oracle Linux 6.4 with an UEK Kernel (choice of the author) and an Oracle Grid Infrastructure 12c for a Cluster deployed on each server. The servers also share an Oracle RAC-enabled Database 12c Home, which in this particular installation is hosted on an ACFS-based cluster file system. Two user-created Server Pools have been created in this cluster (frontoffice and backoffice). The multitenant container database (called “raccdb1”) is an Oracle RAC-based, policy-managed database, currently running on nodes dasher and dancer.

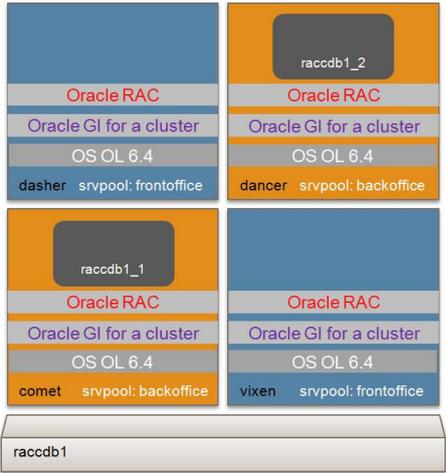


Illustration 4: Milestone - What has been configured so far?

Post-Creation and -Installation Steps

Post-database-creation steps include creating PDBs as required as well as the creation of cluster-managed dynamic database services. Post-installation steps include configuring regular system checks.

CVU runs automatically and regularly as part of the cluster configuration. In addition, Oracle recommends running RACcheck⁵ regularly and setting up Trace File Analyzer (TFA)⁶ to simplify the collection of diagnostic information should the need arise.

PDBs can be created in many ways, of which DBCA is probably the most convenient one, especially if PDBs are created as part of a new database deployment as assumed for this article. When using DBCA to create PDBs for a policy-managed Oracle RAC-based CDB, it needs to be ensured that the DBCA is started on a server that actually runs a CDB instance. DBCA will not proceed with the PDB creation until this requirement is met.

The last step of creating cluster-managed dynamic database services can be performed on any node in the cluster. Dynamic Database Services can be registered on either the CDB-level (using a service to connect to the CDB) as well as the PDB-level. Creating a service on CDB-level is optional, as CDBs should not be used for client connectivity and administrators can find different ways to connect to it. Creating a service on each PDB is highly recommended, as it not only simplifies connectivity to the PDB, but also automates the startup of the PDB in the cluster in a way that the service startup will pull-up and open the PDB in the instance(s) on which the respective workload is meant to run.

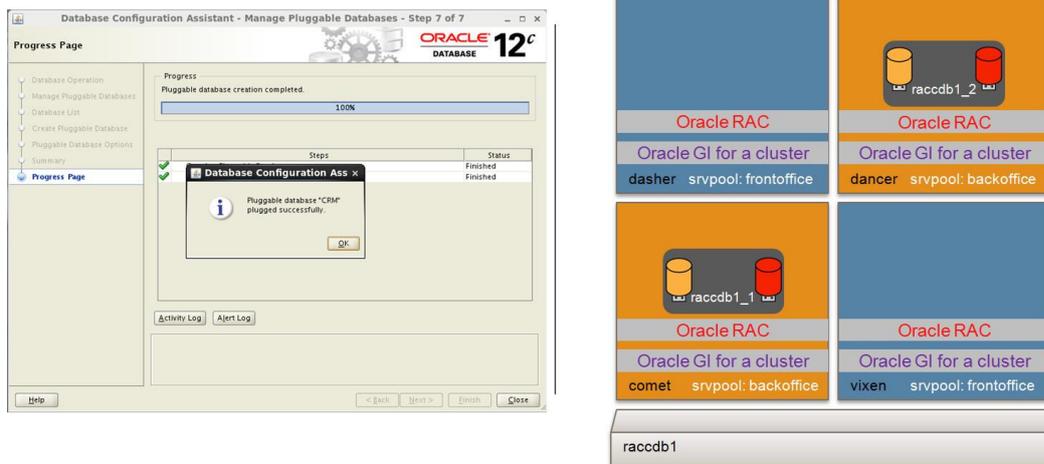


Illustration 5: A 4-node Cluster Example with 1 CDB and 2 PDBs (HR / CRM)

Contact address:

Markus Michalewicz, Oracle
500 Oracle Parkway M/S 4OP840
Redwood Shores, CA, 94065, USA

Phone: +1 650 506 5444
Email: Markus.Michalewicz@oracle.com
Internet: <http://www.oracle.com/goto/rac>

⁵ For more information see MOS note 1268927.1 - RACcheck - RAC Configuration Audit Tool

⁶ For more information see "The Oracle Trace File Analyzer" –
<http://www.oracle.com/technetwork/products/clustering/overview/tracefileanalyzer-2008420.pdf>