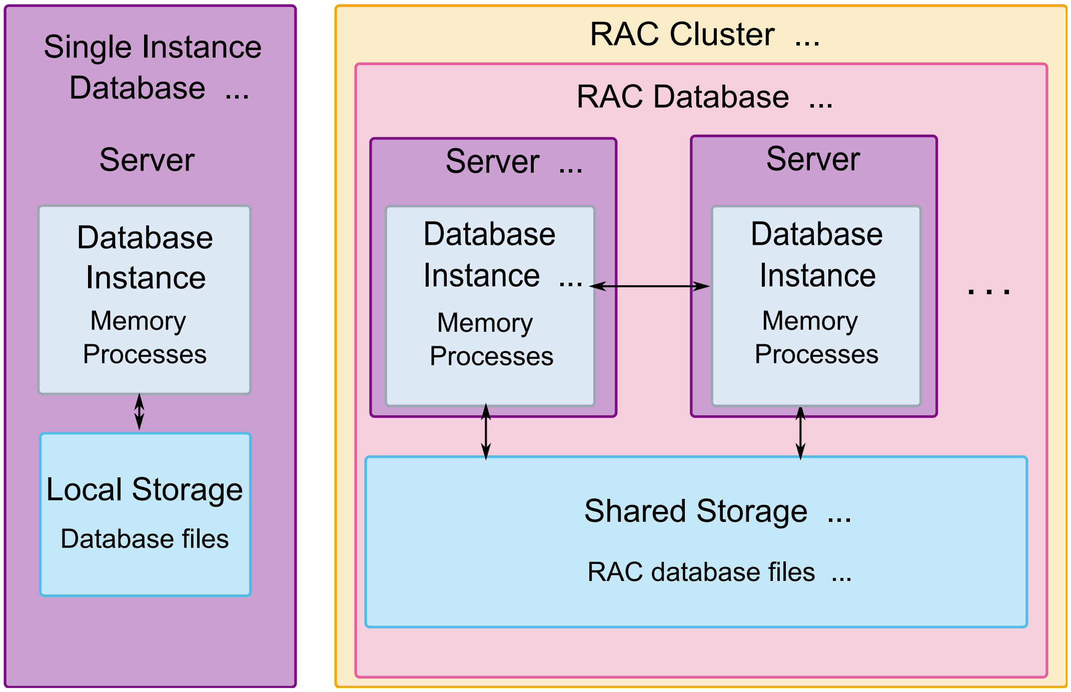
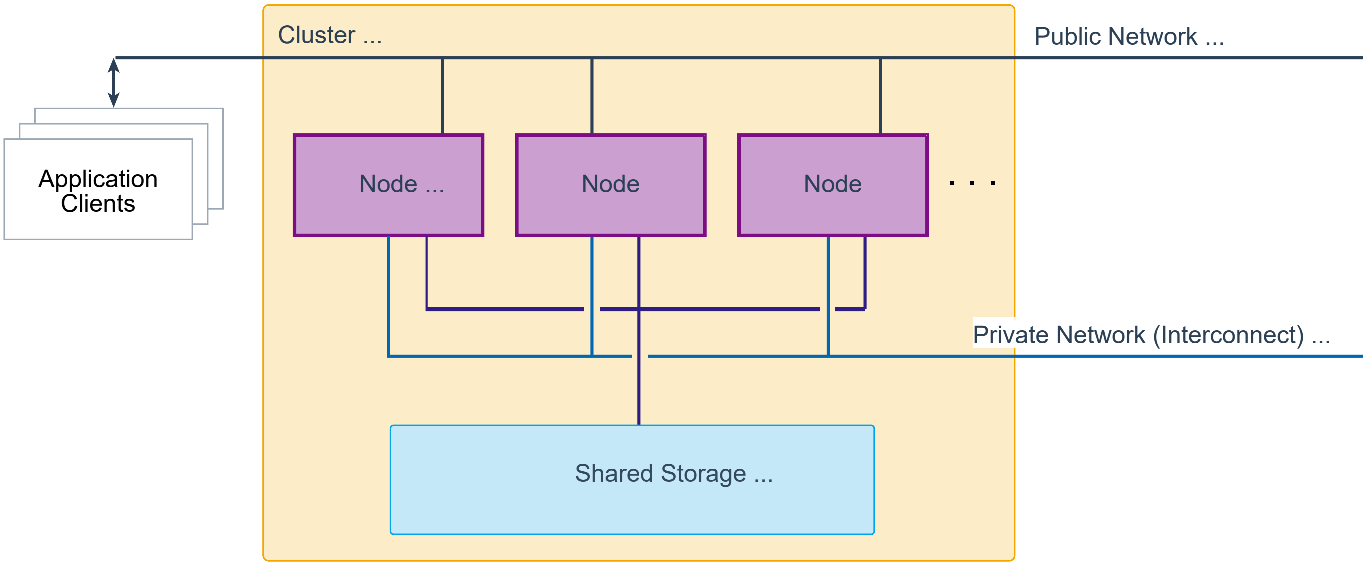
Oracle Real Application Clusters 19c Technical Architecture

Oracle Database Configurations



Oracle Database can be configured in two basic ways: as a single instance (SI) database (non-cluster) or a Real Application Clusters (RAC) database. A single instance database has the Oracle software, database files, memory and processes all on one server. (link to database technical architecture diagram). A single instance database has a one-to-one relationship between the instance (memory and processes) and the database (database files). Oracle RAC environments, however, have a one-to-many relationship between the database and instances. An Oracle RAC database can have up to 100 instances, all of which access one database. All the servers for a single database in the Oracle RAC environment must belong to the same Oracle Real Application Clusters.

Real Application Clusters (RAC) Overview



All Real Application Clusters have certain things in common, each cluster has Grid Infrastructure (GI) installed locally on each node. Grid Infrastructure includes several products: Automatic Storage Management, Oracle Clusterware, and others.

Each Instance in the Oracle RAC cluster accesses the database files stored on shared storage. Each node also requires local storage which is used to store the GI and Oracle Database software.

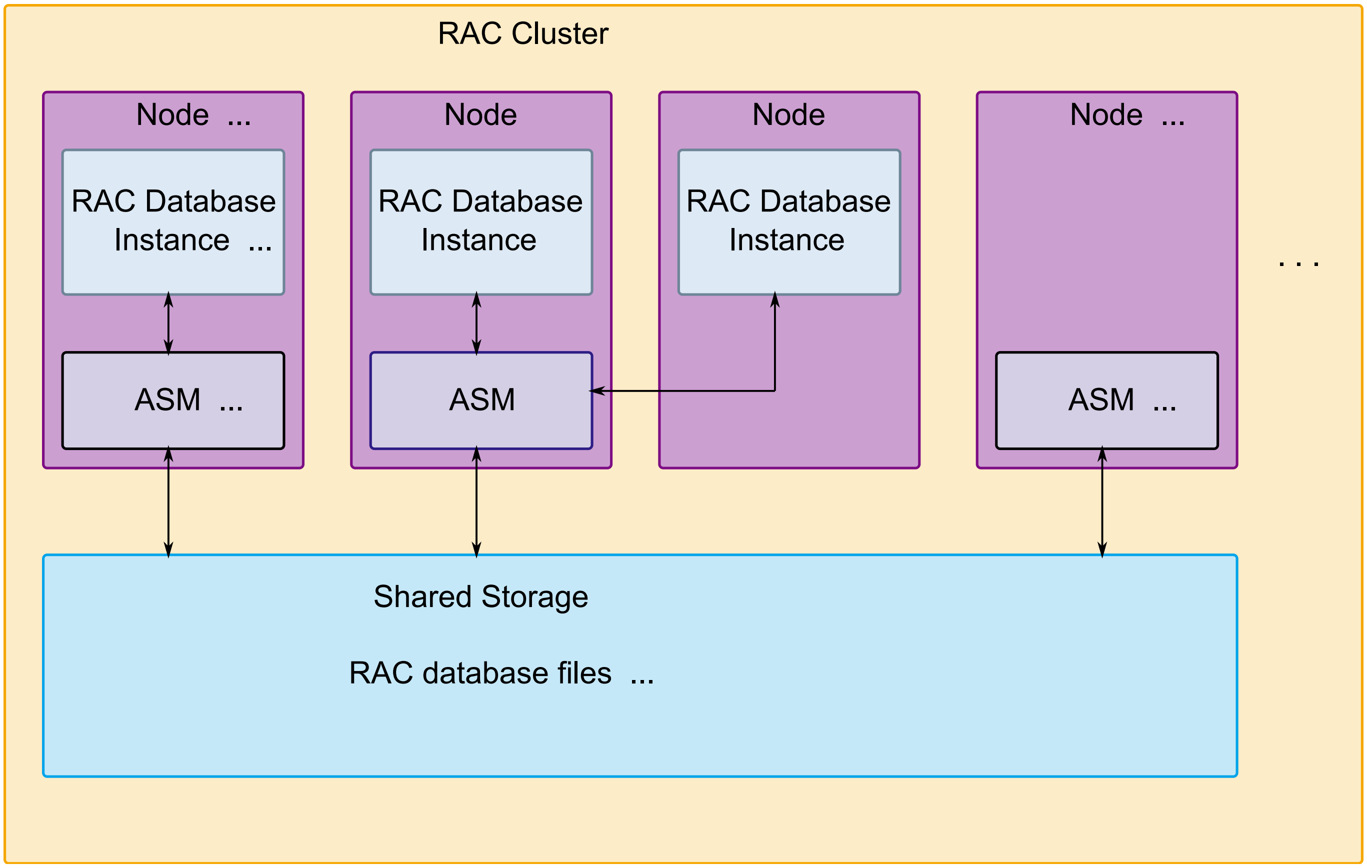
Each node needs at least a public interface and a private interface. Each network may use multiple network interface cards to increase bandwidth, availability or fault tolerance. Oracle Corp. recommends that each network use multiple network interface cards (NICs) per node and multiple network switches in each network to avoid a single point of failure.

A cluster comprises multiple interconnected computers or servers that appear as if they are one server to end users and applications. The Oracle RAC option with Oracle Database enables you to cluster Oracle databases. Oracle RAC uses Oracle Clusterware for the infrastructure to bind multiple servers so they operate as a single system.

Oracle Clusterware is a portable cluster management solution that is integrated with Oracle Database. Oracle Clusterware is a required component for using Oracle RAC that provides the infrastructure necessary to run Oracle RAC. Oracle Clusterware also manages resources, such as Virtual Internet Protocol (VIP) addresses, databases, listeners, services, and so on. In addition, Oracle Clusterware enables both noncluster Oracle databases and Oracle RAC databases to use the Oracle high-availability infrastructure. Oracle Clusterware along with Oracle Automatic Storage Management (Oracle ASM) (the two together comprise the Oracle Grid Infrastructure) enables you to create a clustered pool of storage to be used by any combination of noncluster and Oracle RAC databases.

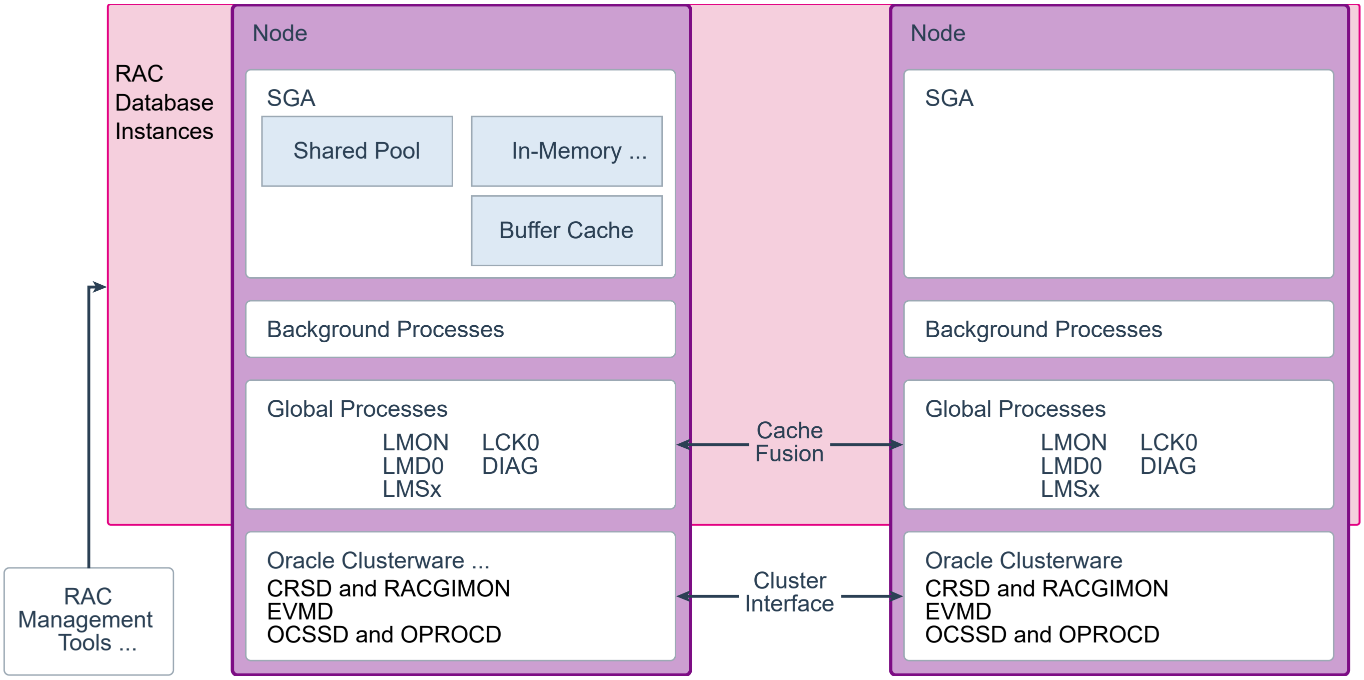
Oracle Clusterware is the only clusterware that you need for most platforms on which Oracle RAC operates. If your database applications require vendor clusterware, then you can use such clusterware in conjunction with Oracle Clusterware if that vendor clusterware is certified for Oracle RAC.

RAC Database



A Real Application Clusters (RAC) Database has multiple database Instances hosted on a RAC Cluster. These instances access a single set of Database files on Shared Storage. Frequently the Shared storage is provided by an ASM instance local to the node hosting the RAC database instance. In Oracle ASM 19c there is only one configuration option, that is Flex ASM, Flex ASM can have a variety of placements for the ASM instances in the cluster that is controlled by setting the cardinallity, the number of ASM instances, in the cluster.

RAC Database Instances

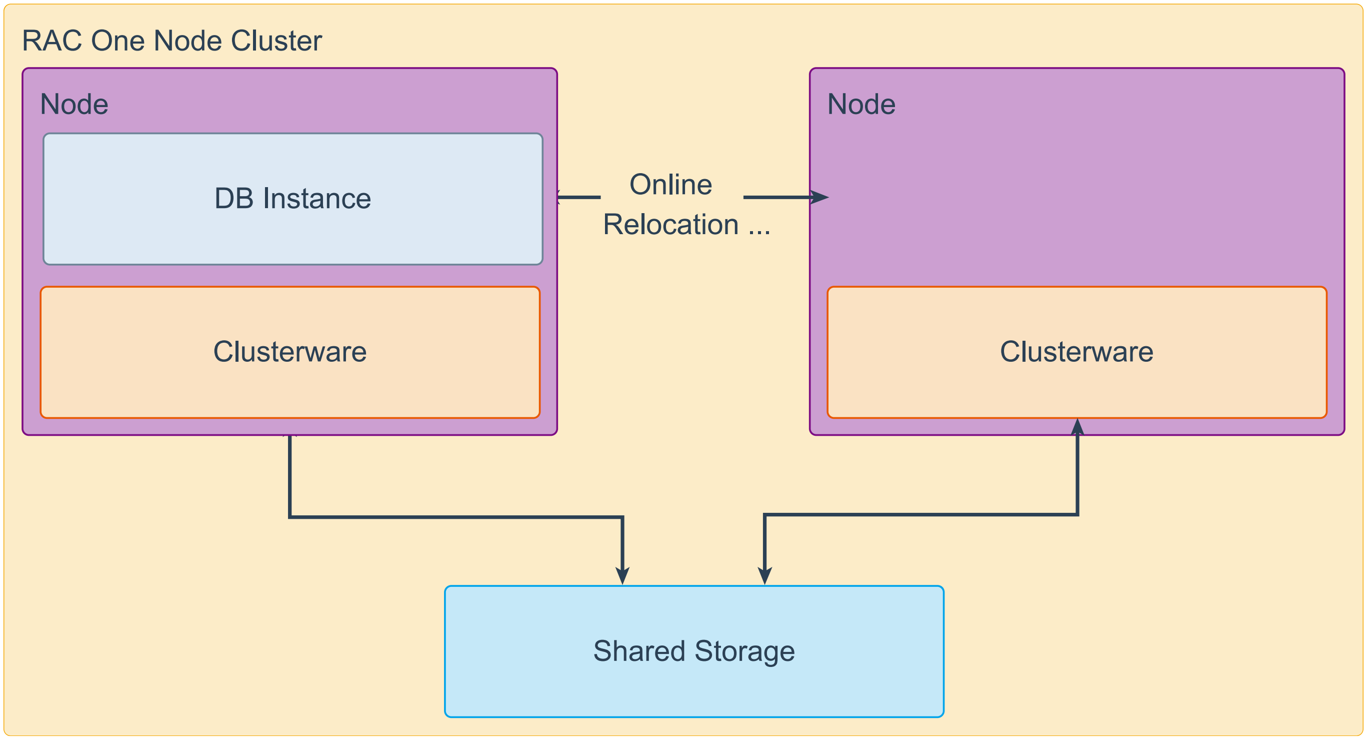


A RAC database is a set of files on shared storage and can span multiple nodes.

A RAC database instance is a set of processes and memory that operates on the database files. A RAC database instance resides on one node. It communicates through the global processes and clusterware with other RAC database instances in the same cluster.A RAC database instance has the same memory and background processes as a single instance database, with the addition of Cache Fusion, and Global Processes. Cache Fusion is a diskless cache coherency mechanism in Oracle Real Application Clusters that provides copies of blocks directly from a holding instance's memory cache to a requesting instance's memory cache. This provides a logical merging of the SGA’s across the DB instances in the cluster. Global Processes (not all are shown) provide the necessary services to maintain the cache coherency, and recovery in the event of a node failure.

Cache Fusion and Cluster Interface both use the Private Network (Interconnect) for communication between cluster nodes.

RAC One Node Overview 1



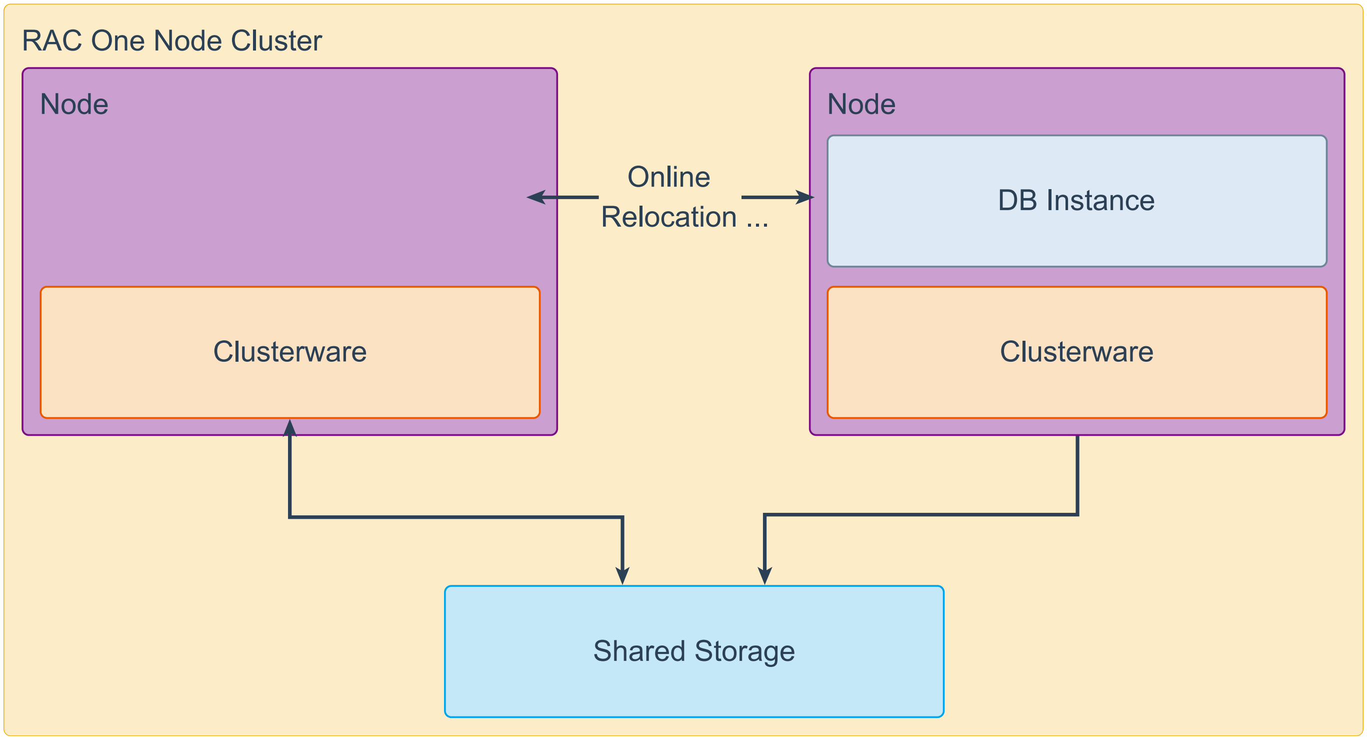
Oracle Real Application Clusters One Node (Oracle RAC One Node) is a single instance of an Oracle Real Application Clusters (Oracle RAC) database that runs on one node in a cluster. This option adds to the flexibility that Oracle offers for database consolidation. You can consolidate many databases into one cluster with minimal overhead while also providing the high availability benefits of failover protection, online rolling patch application, and rolling upgrades for the operating system and Oracle Clusterware.

The database instance can be relocated to another node in the cluster while maintaining application continuity. RAC One node instances may co-exist with other RAC database instances in the same cluster, assuming there are enough nodes with sufficient resources to service all the instances.

The main difference between Oracle RAC One Node and Oracle RAC is that with an Oracle RAC One Node Database, there is only one instance running at a time under normal operation. Should this one instance be impacted by unplanned downtime, there is a short period of time during which no database instance is running.

The stack used for Oracle RAC One Node ensures that the database service will be resumed as fast as possible, but technically and as with any other failover cluster solution, there is a moment of time, during which no instance of the Oracle RAC One Node Database is available to perform any work. This would not be the case in a multi-node Oracle RAC system, because multiple instances are already running concurrently, accessing the same database

RAC One Node Overview 2



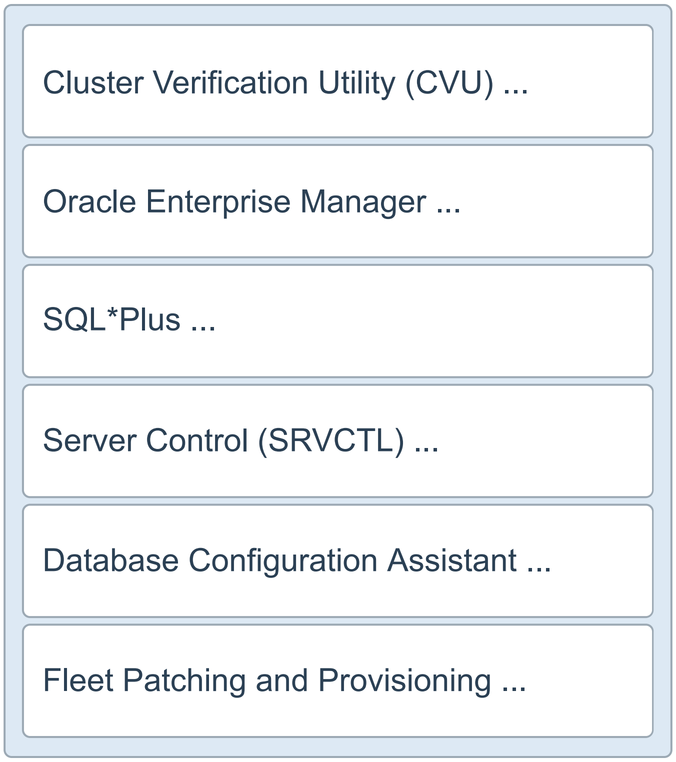
Oracle Real Application Clusters One Node (Oracle RAC One Node) is a single instance of an Oracle Real Application Clusters (Oracle RAC) database that runs on one node in a cluster. This option adds to the flexibility that Oracle offers for database consolidation. You can consolidate many databases into one cluster with minimal overhead while also providing the high availability benefits of failover protection, online rolling patch application, and rolling upgrades for the operating system and Oracle Clusterware.

The database instance can be relocated to another node in the cluster while maintaining application continuity. RAC One node instances may co-exist with other RAC database instances in the same cluster, assuming there are enough nodes with sufficient resources to service all the instances.

The main difference between Oracle RAC One Node and Oracle RAC is that with an Oracle RAC One Node Database, there is only one instance running at a time under normal operation. Should this one instance be impacted by unplanned downtime, there is a short period of time during which no database instance is running.

The stack used for Oracle RAC One Node ensures that the database service will be resumed as fast as possible, but technically and as with any other failover cluster solution, there is a moment of time, during which no instance of the Oracle RAC One Node Database is available to perform any work. This would not be the case in a multi-node Oracle RAC system, because multiple instances are already running concurrently, accessing the same database

RAC Tools


Cluster Verification Utility (CVU) ...

Oracle Enterprise Manager ...

SQL*Plus ...

Server Control (SRVCTL) ...

Database Configuration Assistant ...

Fleet Patching and Provisioning ...

You administer a cluster database as a single-system image using the Server Control Utility (SRVCTL), Oracle Enterprise Manager, SQL*Plus, and other utilities.

Cluster Verification Utility (CVU): CVU is a command-line tool that you can use to verify a range of cluster and Oracle RAC components, such as shared storage devices, networking configurations, system requirements, and Oracle Clusterware, in addition to operating system groups and users. You can use CVU for preinstallation checks and for postinstallation checks of your cluster environment. CVU is especially useful during preinstallation and during installation of Oracle Clusterware and Oracle RAC components. Oracle Universal Installer runs CVU after installing Oracle Clusterware and Oracle Database to verify your environment. Install and use CVU before you install Oracle RAC to ensure that your configuration meets the minimum Oracle RAC installation requirements. Also, use CVU for verifying the completion of ongoing administrative tasks, such as node addition and node deletion.

Oracle Enterprise Manager: Oracle Enterprise Manager Cloud Control GUI interface for managing both noncluster database and Oracle RAC database environments. Oracle recommends that you use Oracle Enterprise Manager to perform administrative tasks whenever feasible.

You can use Oracle Enterprise Manager Cloud Control to also manage Oracle RAC One Node databases.

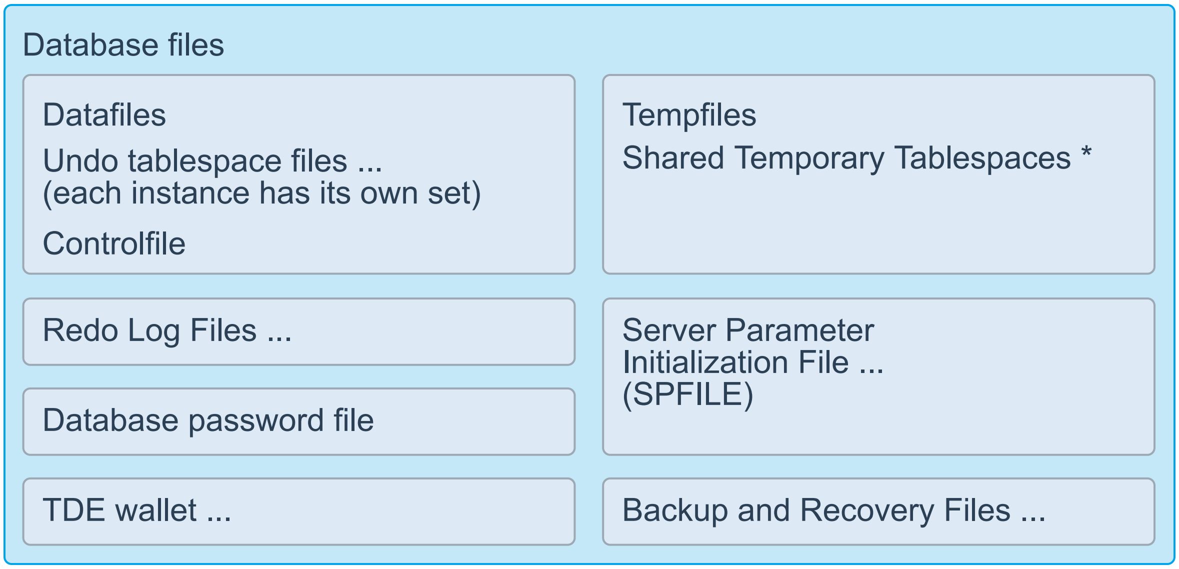
SQL*Plus: SQL*Plus commands operate on the current instance. The current instance can be either the local default instance on which you initiated your SQL*Plus session, or it can be a remote instance to which you connect with Oracle Net Services.

Server Control Utility (SRVCTL): SRVCTL is a command-line interface that you can use to manage an Oracle RAC database from a single point. You can use SRVCTL to start and stop the database and instances and to delete or move instances and services. You can also use SRVCTL to manage configuration information, Oracle Real Application Clusters One Node (Oracle RAC One Node), Oracle Clusterware, and Oracle ASM.

DBCA: The recommended utility for creating and initially configuring Oracle RAC, Oracle RAC One Node, and Oracle noncluster databases.

Fleet Patching and Provisioning: Use Fleet Patching and Provisioning to patch, upgrade, and provision Oracle RAC databases.

Database Files in RAC



Database files

Datafiles

Undo tablespace files ...
(each instance has its own set)

Controlfile

Tempfiles

Shared Temporary Tablespaces *

Redo Log Files ...

Server Parameter
Initialization File ...
(SPFILE)

Database password file

TDE wallet ...

Backup and Recovery Files ...

Every instance must have access to all the datafiles, including the datafiles associated with Undo tablespaces, and controlfiles. Each Instance must have an Undo Tablespace dedicated to it. All the Undo Tablespaces must be on Shared Storage. All redo log files must be accessible to all instances, each redo log file should be multiplexed as in a single instance. When using ASM with normal redundancy each redo log member is mirrored, and a second Multiplexed member is placed in a different disk group. Each instance must have at least two redo log groups also called a thread.

* The tempfiles associated with the shared temporary tablespaces are in shared storage and can spill to local storage when using local temporary tablespaces. For more information, see About Temporary Tablespaces.

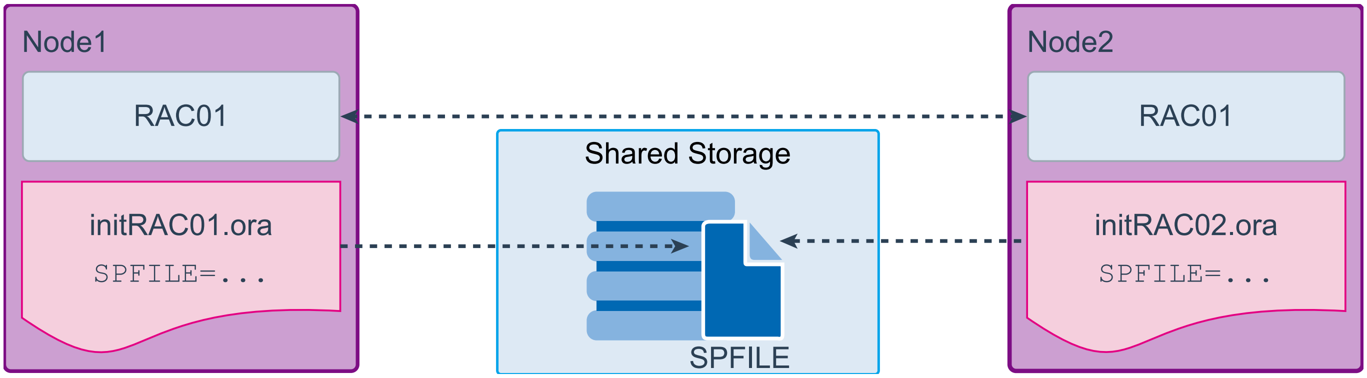
You can create a shared temporary tablespace by specifying the TEMPORARY TABLESPACE clause. A shared temporary tablespace stores tempfiles on shared disk, so that the temporary space is accessible to all database instances. Starting with Oracle Database 12c Release 2 (12.2), you can create a local temporary tablespace by specifying the LOCAL TEMPORARY TABLESPACE clause. Local temporary tablespaces are useful in an Oracle Clusterware environment. They store a separate, nonshared temp files for each database instance, which can improve I/O performance. A local temporary tablespace must be a BIGFILE tablespace

There should be at least 2 multiplexed controlfiles accessible to all instances, as with redo log files each multiplexed controlfile should be placed in a different disk group.

A common (default) SPFILE with both the common parameters and instance specific parameters is highly recommended in shared storage. The SPFILE can be stored in ASM. Some initialization parameters must be the same across all instances in a cluster, and some must be unique per instance, most parameters should be the same to maintain consistent behavior.

The Database Password file (orapwd*) may be placed in shared storage, or may be on each local node. This file contains the passwords of privileged user and must be available when the database in NOT open.

Server Parameter File in RAC



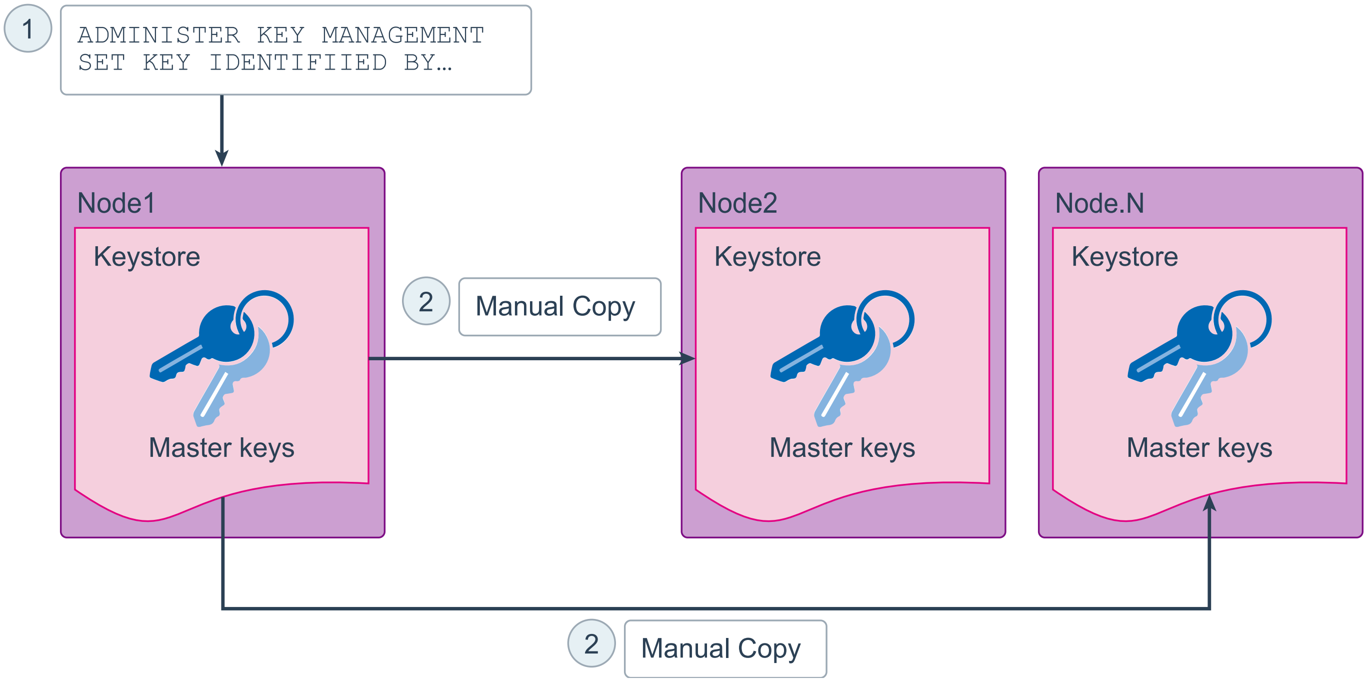
When you create the database, DBCA creates an SPFILE in the file location that you specify. This location can be an ASM disk group, cluster file system file, or a shared raw device. If you manually create your database, it is recommended that you create an SPFILE from a PFILE. All instances in the cluster database use the same SPFILE. Because the SPFILE is a binary file, do not edit it. You can change the SPFILE settings by using EM or ALTER SYSTEM SQL statements. RAC uses a traditional PFILE only if an SPFILE does not exist or if you specify PFILE in your STARTUP command. Using SPFILE simplifies administration, maintaining consistent parameter settings, and guaranteeing parameter settings persistence across database shutdown and startup. In addition, you can configure RMAN to back up your SPFILE.

You can modify the value of your initialization parameters by using the ALTER SYSTEM SET command. This is the same as with a single-instance database except that you have the possibility to specify the SID clause in addition to the SCOPE clause. By using the SID clause, you can specify the SID of the instance where the value takes effect. Specify SID='*' if you want to change the value of the parameter for all instances. Specify SID='sid' if you want to change the value of the parameter only for the instance sid. This setting takes precedence over previous and subsequent ALTER SYSTEM SET statements that specify SID='*'. If the instances are started up with an SPFILE, then SID='*' is the default if you do not specify the SID clause. If you specify an instance other than the current instance, then a message is sent to that instance to change the parameter value in its memory if you are not using the SPFILE scope. The combination of SCOPE=MEMORY and SID='sid' of the ALTER SYSTEM RESET command allows you to override the precedence of a currently used <dparam> entry. This allows for the current *.<dparam> entry to be used, or for the next created *.<dparam> entry to be taken into account on that particular SID. Using the last example, you can remove a line from your SPFILE.

- CLUSTER_DATABASE: Enables a database to be started in cluster mode. Set this to TRUE.
- CLUSTER_DATABASE_INSTANCES: Sets the number of instances in your RAC environment. A proper setting for this parameter can improve memory use.
- CLUSTER_INTERCONNECTS: Specifies the cluster interconnect when there is more than one interconnect. Refer to your Oracle platform–specific documentation for the use of this parameter, its syntax, and its behavior. You typically do not need to set the CLUSTER_INTERCONNECTS parameter. For example, do not set this parameter for the following common configurations:
    - If you have only one cluster interconnect
    - If the default cluster interconnect meets the bandwidth requirements of your RAC database, which is typically the case
    - If NIC bonding is being used for the interconnect

    - When OIFCFG’s global configuration can specify the right cluster interconnects. It only needs to be specified as an override for OIFCFG.
- DB_NAME: If you set a value for DB_NAME in instance-specific parameter files, the setting must be identical for all instances.
- DISPATCHERS: Set this parameter to enable a shared-server configuration, that is, a server that is configured to allow many user processes to share very few server processes.

TDE Keystores in RAC

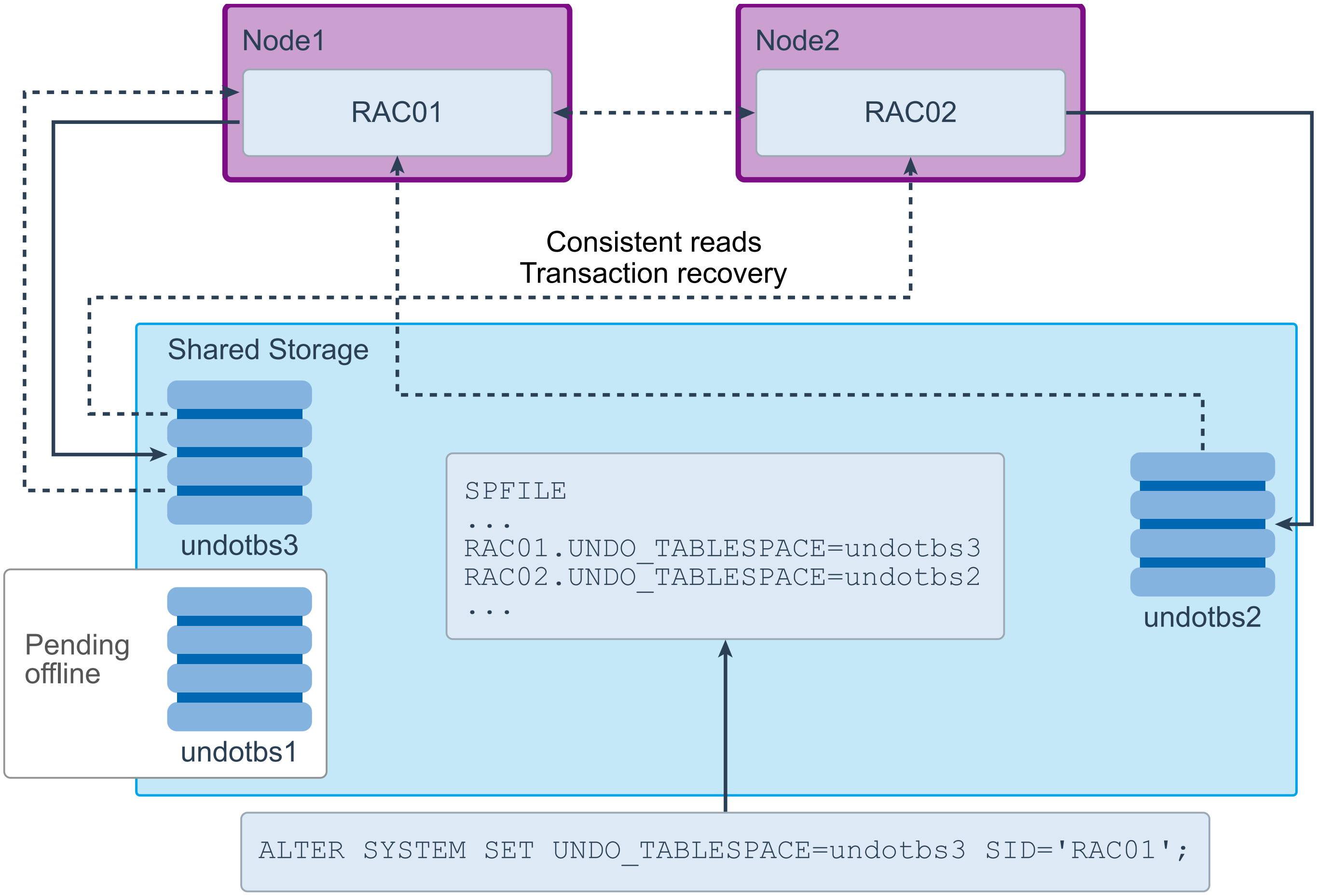


Wallets used by RAC instances for Transparent Database Encryption (TDE) may be a local copy of a common wallet shared by multiple nodes, or a shared copy residing on shared storage that all of the nodes can access.

A deployment with a single wallet on a shared disk requires no additional configuration to use Transparent Data Encryption.

If you want to use local copies, you must copy the wallet and make it available to all of the other nodes after initial configuration. For systems using Transparent Data Encryption with encrypted wallets, you can use any standard file transport protocol. For systems using Transparent Data Encryption with obfuscated wallets, file transport through a secured channel is recommended. The wallet must reside in the directory specified by the setting for the WALLET_LOCATION or ENCRYPTION_WALLET_LOCATION parameter in sqlnet.ora. In 19c, WALLET_ROOT initialization parameter superceddes the settings in sqlnet.ora. The local copies of the wallet need not be synchronized for the duration of Transparent Data Encryption usage until the server key is rekeyed through the ALTER SYSTEM SET KEY SQL statement. Each time you run the ALTER SYSTEM SET KEY statement at a database instance, you must again copy the wallet residing on that node and make it available to all of the other nodes. To avoid unnecessary administrative overhead, reserve rekeying for exceptional cases where you are certain that the server master key is compromised and that not rekeying it would cause a serious security problem.

Undo Tablespaces in RAC



```
SPFILE
...
RAC01.UNDO_TABLESPACE=undotbs3
RAC02.UNDO_TABLESPACE=undotbs2
...
```

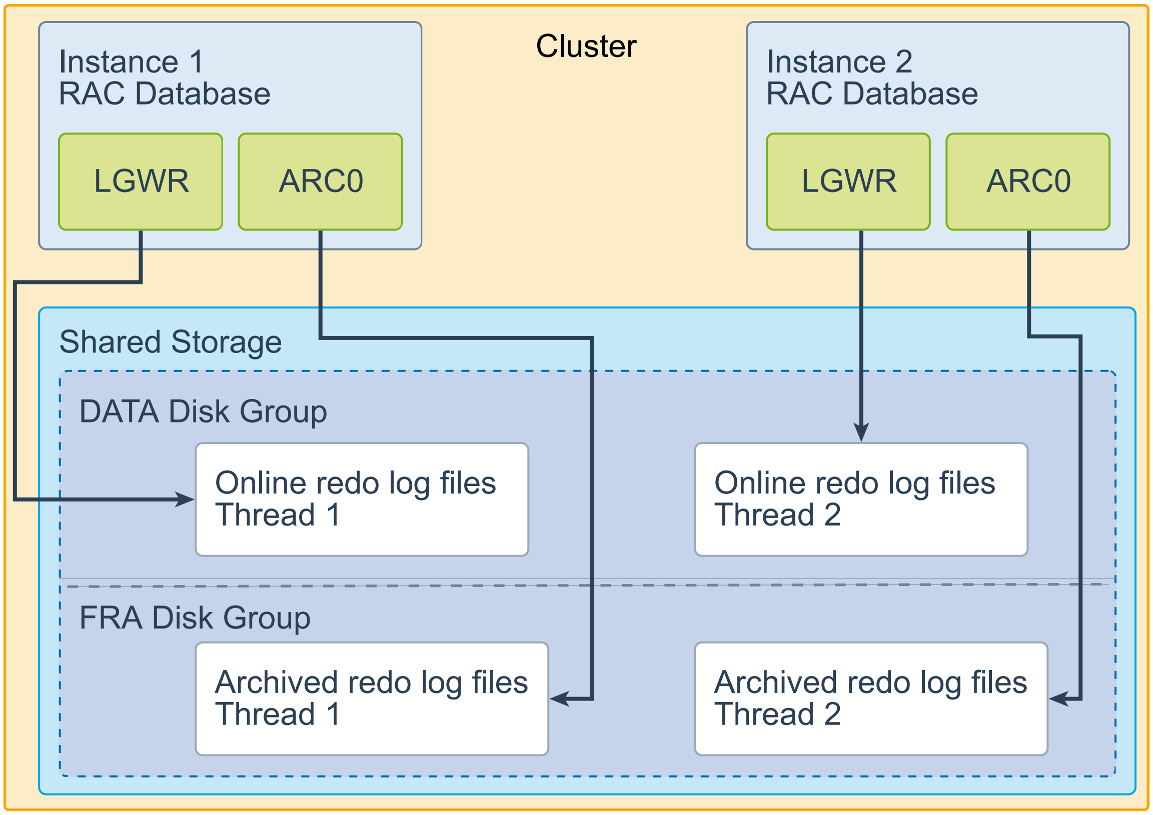
```
ALTER SYSTEM SET UNDO_TABLESPACE=undotbs3 SID='RAC01';
```

The Oracle database automatically manages undo segments within a specific undo tablespace that is assigned to an instance. Under normal circumstances, only the instance assigned to the undo tablespace can modify the contents of that tablespace. However, all instances can always read all undo blocks for consistent-read purposes. Also, any instance can update any undo tablespace during transaction recovery, as long as that undo tablespace is not currently used by another instance for undo generation or transaction recovery.

You assign undo tablespaces in your RAC database by specifying a different value for the UNDO_TABLESPACE parameter for each instance in your SPFILE or individual PFILEs. If you do not set the UNDO_TABLESPACE parameter, each instance uses the first available undo tablespace. For policy-managed databases, Oracle automatically allocates the undo tablespace when the instance starts if you have OMF enabled.

You can dynamically switch undo tablespace assignments by executing the ALTER SYSTEM SET UNDO_TABLESPACE statement. You can run this command from any instance. In the example above, the previously used undo tablespace, unodtbs1, assigned to the RAC01 instance remains assigned to it until RAC01's last active transaction commits. The pending offline tablespace may be unavailable for other instances until all transactions against that tablespace are committed. You cannot simultaneously use Automatic Undo Management (AUM) and manual undo management in a RAC database. It is highly recommended that you use the AUM mode.

Redo and Redo Threads



Each instance's redo thread must contain at least two redo log groups. Each redo log group should contain at least two members: a redo log and its mirrored copy. If you create your Oracle RAC database using DBCA, then your Oracle RAC database automatically implements a configuration that meets the Oracle recommendations.

You should create redo log groups only if you are using administrator-managed databases. For policy-managed databases, if an instance starts due to a change in server pool cardinality, then Oracle Database automatically creates redo log files, enables a redo thread for the instance if there is not a redo thread allocated to that instance, and creates the undo tablespace if there is not an undo tablespace allocated to that instance. The database must be using Oracle Managed Files and Oracle ASM in this situation.

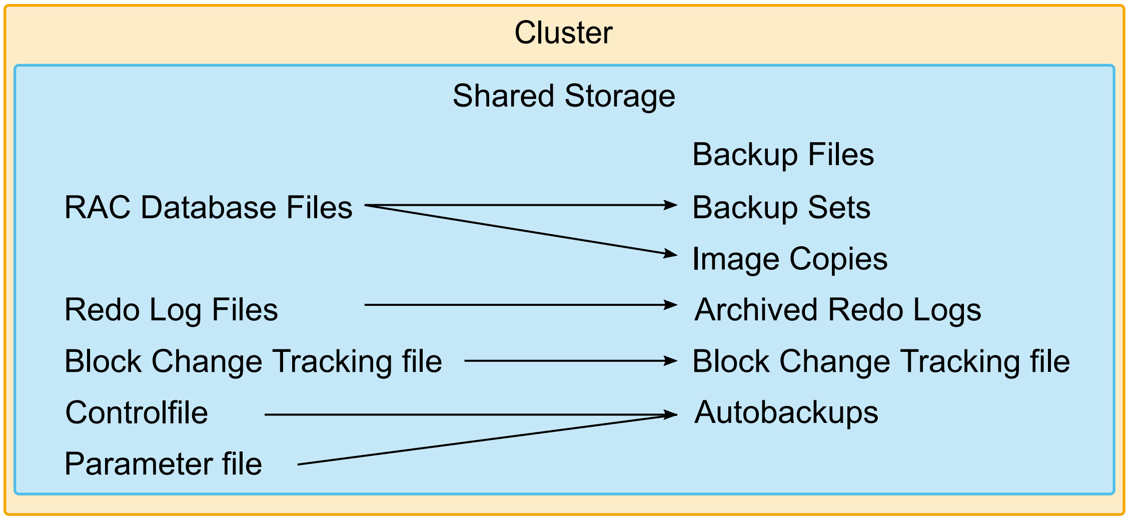
In an Oracle RAC database, all the redo log files reside on shared storage. In addition, each instance must have access to the redo log files of all the other instances in the cluster. If your Oracle RAC database uses Oracle ASM, then Oracle ASM manages the shared storage for the redo log files and the access to those files.

In an Oracle RAC database, each instance writes and archives the redo log groups in its redo thread in the same manner that single-instance Oracle databases do. However, in recovery mode, the instance performing the recovery can read and process all the redo threads for the database, regardless of which instance generated the redo thread. Being able to read all the redo threads enables a running instance to recover the work completed by one or more failed instances.

In case of instance failure, a surviving instance can read the redo logs of the failed instance. Users can continue to access and update the database without waiting for the failed instance to be restarted. For example, assume that you have an Oracle RAC database with two instances, instance A and instance B. If instance A is down, then instance B can read the redo log files for both instance A and B to ensure a successful recovery.

The diagram assumes the use of ASM Diskgroups

Backup Files in RAC



Backup files in RAC have the same characteristics as in a single instance database. But with the added consideration of multiple instances having access to all the datafiles. Because any instance in the RAC cluster may perform restore and recovery operations, the backup files must be accessible to that instance at the time of the operation. It is highly recommended that the backup files and archive log files be placed in shared storage, so that they are available to all instances at all times.

RAC and Database Features

Single Instance Feature	RAC Implementation
Multitenant Database	Whether a PDB is available on more than one instance of an Oracle RAC CDB, the CDB is typically managed by the services running on the PDB. You can manually enable PDB access on each instance of an Oracle RAC CDB by starting the PDB manually on that instance.
Database In-Memory	Each node in an Oracle RAC environment has its own In-Memory column store (IMCS). By default, populated objects are distributed across all IM column stores in the cluster.

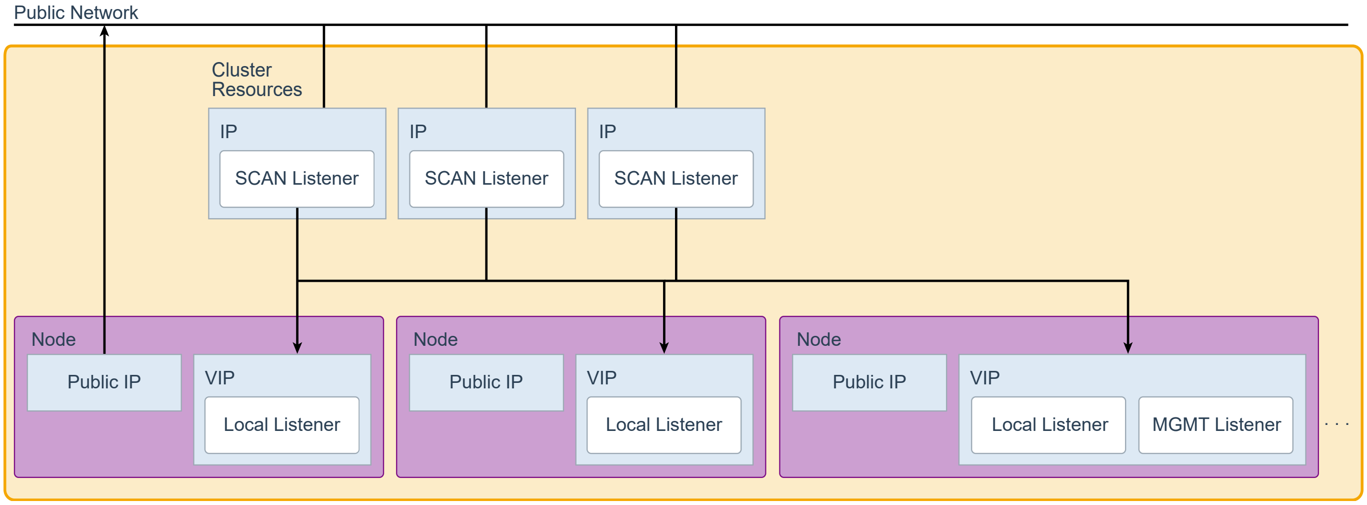
When using PDBs with Oracle RAC, the multitenant CDB is an Oracle RAC database. This means for the CDB database, there is an instance running on multiple nodes in the cluster. You can make each PDB available on every instance of the Oracle RAC database or a subset of instances. In either case, access to and management of the PDBs are regulated using dynamic database services, which are also used by applications to connect to the respective PDB.

If you create an Oracle RAC database as a CDB and plug one or more PDBs into the CDB, then, by default, a PDB is not started automatically on any instance of the Oracle RAC CDB. When you assign the first dynamic database service to the PDB (other than the default database service which has the same name as the database name), then the PDB is made available on those instances on which the service runs.

Whether a PDB is available on more than one instance of an Oracle RAC CDB, is automatically managed by the services running on the PDB. You can manually enable PDB access on each instance of an Oracle RAC CDB by starting the PDB manually on that instance.

Note: Services must be enabled to use PDBs with RAC.

Listeners in RAC



SCAN - Single Client Access Name is a single name assigned to the cluster, DNS resolves this name to one of 3 or more IP address in a round robin fashion. SCAN listeners and SCAN addresses are optional, but recommended.

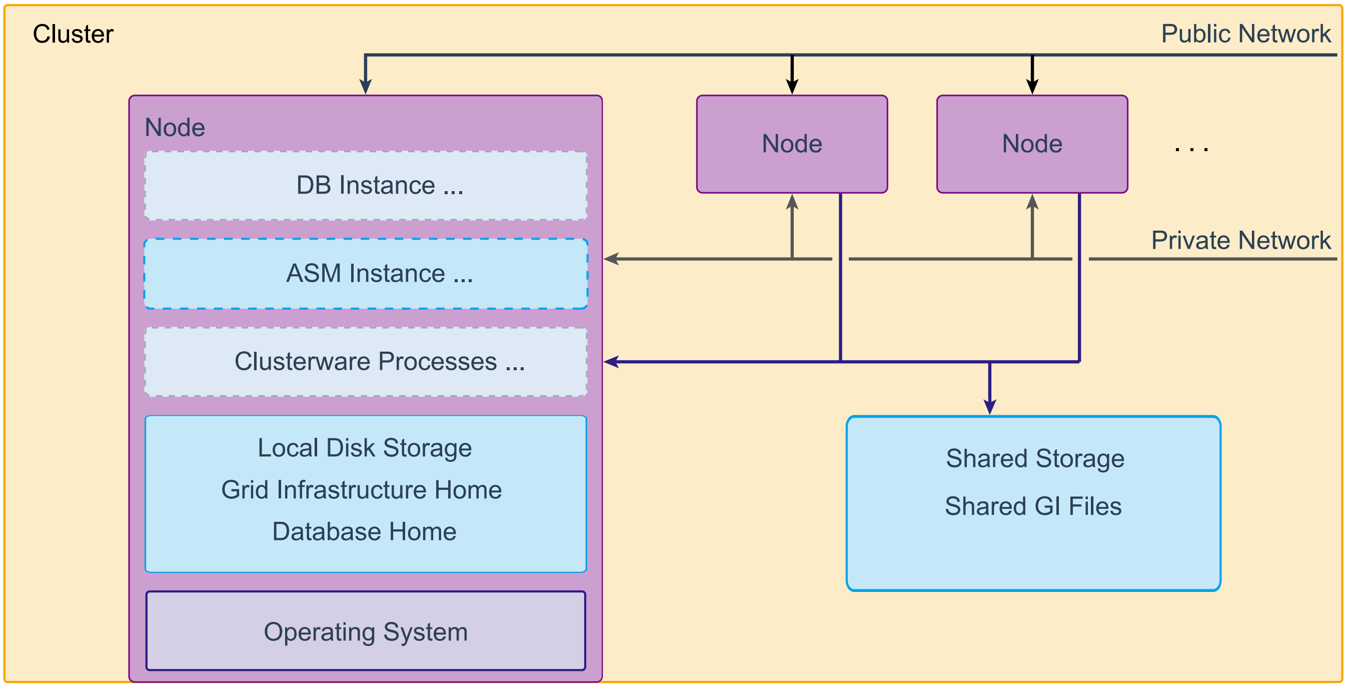
The IP addresses assigned to SCAN, are relocatable, each SCAN IP will move to a active node in the cluster if the current node fails. A SCAN listener is assigned to each SCAN IP. The SCAN name should be configured in DNS so that three SCAN addresses may be assigned to the same SCAN name.

Each database registers with the SCAN listener and updates it with the current load metric. The SCAN listener receives requests and routes the requests to the Local listener on the least loaded node. The local listener is using a virtual IP (VIP) address.

Each node will have a public IP address assigned to the network interface for that particular node.

The MGMTLISTENER allows the Cluster Health Monitor to communicate with the Grid Infrastructure Management Repository (GIMR). In the Cluster Domain configuration the GIMR must be configured in the Domain Services Cluster (DSC). In a standalone cluster configuration Each cluster may have a GIMR, on one of the nodes, by default it is created on the first node created in the cluster. In Oracle Database Release 19c the GIMR is optional, but with reduced functionality.

Clustered Node



A Clustered Node is a server that is a member or a planned member of a Cluster. There may be one or many nodes in a cluster.

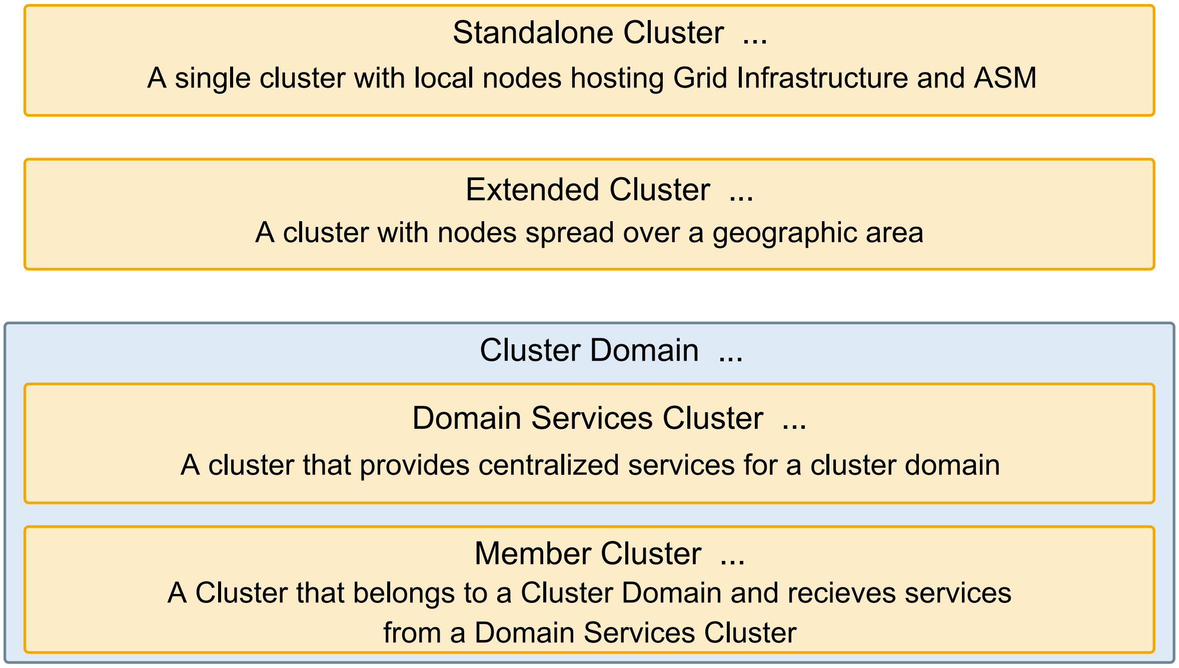
Each node has local storage. Local storage provides for the OS needs and the Grid Infrastructure software installation. Every clustered node has an instance of Clusterware running. The Automatic Diagnostics Repository (ADR) and a local copy for the Oracle Cluster Registry(OLR) are both stored in the node local storage.

Each node must have access to shared storage. Shared storage is available from all the nodes of the cluster.

A node may have one or more other instances running of a RAC database or an application. It is possible to install the software for these instances on shared storage, but it is recommended that these be installed on local storage to allow rolling upgrades, or relocation. However, using local storage also increases your software maintenance costs, because each Oracle home must be upgraded or patched separately. This cost can be mitigated by using Fleet Patching and Provisioning.

Each Node has a network connection to what is called the public network. This network provides the connection applications that use the database or clustered application. Each Node has a private network connection also called the interconnect that provides communication between the nodes of a cluster. Shared storage must be accessible from each node.

Cluster Configuration Options



Standalone Cluster ...

A single cluster with local nodes hosting Grid Infrastructure and ASM

Extended Cluster ...

A cluster with nodes spread over a geographic area

Cluster Domain ...

Domain Services Cluster ...

A cluster that provides centralized services for a cluster domain

Member Cluster ...

A Cluster that belongs to a Cluster Domain and recieves services from a Domain Services Cluster

The Standalone Cluster consists of one or more cluster nodes configured with locally available shared storage, a private interconnect, local instances of Automatic Storage Management (ASM) for managing that shared storage, and the GI Management Repository (GIMR) for storing cluster health and diagnostic information.

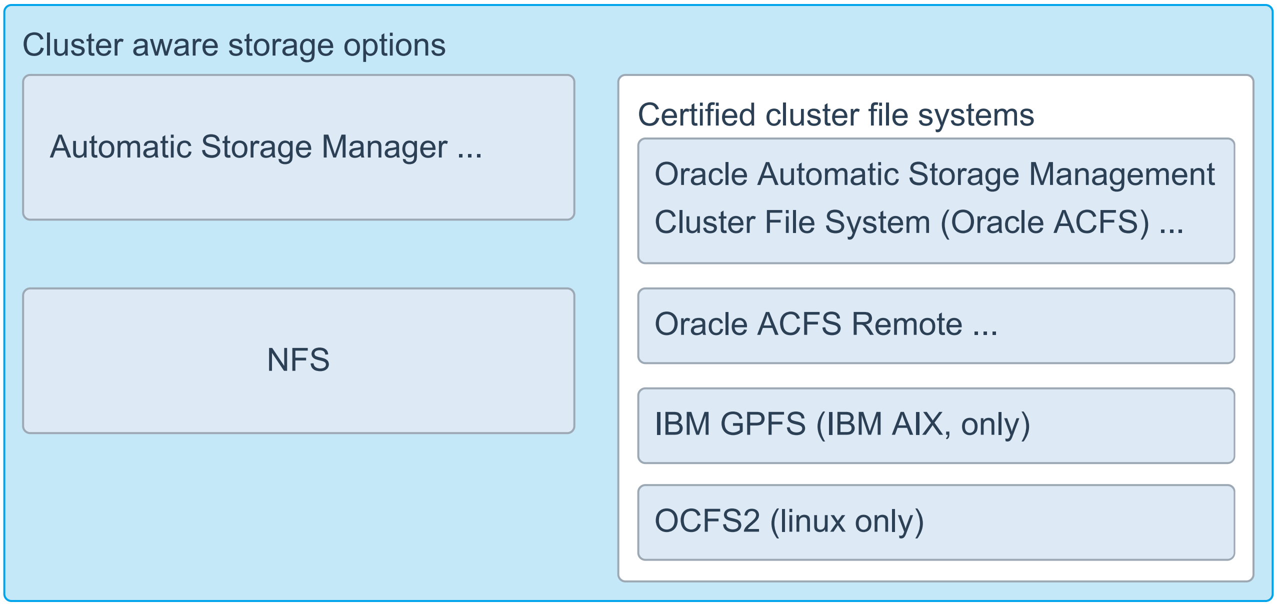
An Extended Cluster is a special case of a standalone cluster where the nodes of the cluster are in two or more sites at different locations.

A Cluster Domain is a grouping of clusters. A Cluster Domain consists of a single Domain Services Cluster and a number of Member Clusters (hosting applications or databases) that utilize services offered on the Domain Services Cluster.

A Member Cluster receives services such ASM or ASM Cluster File System (ACFS) from a Domain Services Cluster. A standalone cluster may be converted to a member cluster.

Domain Services Cluster provides centralized and consolidated services, that are consumed by the Member Clusters registered with that Domain Services Cluster.

Cluster Storage Options


Cluster aware storage options
Automatic Storage Manager ...
NFS
Certified cluster file systems
Oracle Automatic Storage Management Cluster File System (Oracle ACFS) ...
Oracle ACFS Remote ...
IBM GPFS (IBM AIX, only)
OCFS2 (linux only)

A cluster requires a shared storage area that is accessible to all the cluster members. The supported types of shared storage depend upon the platform you are using, for example:

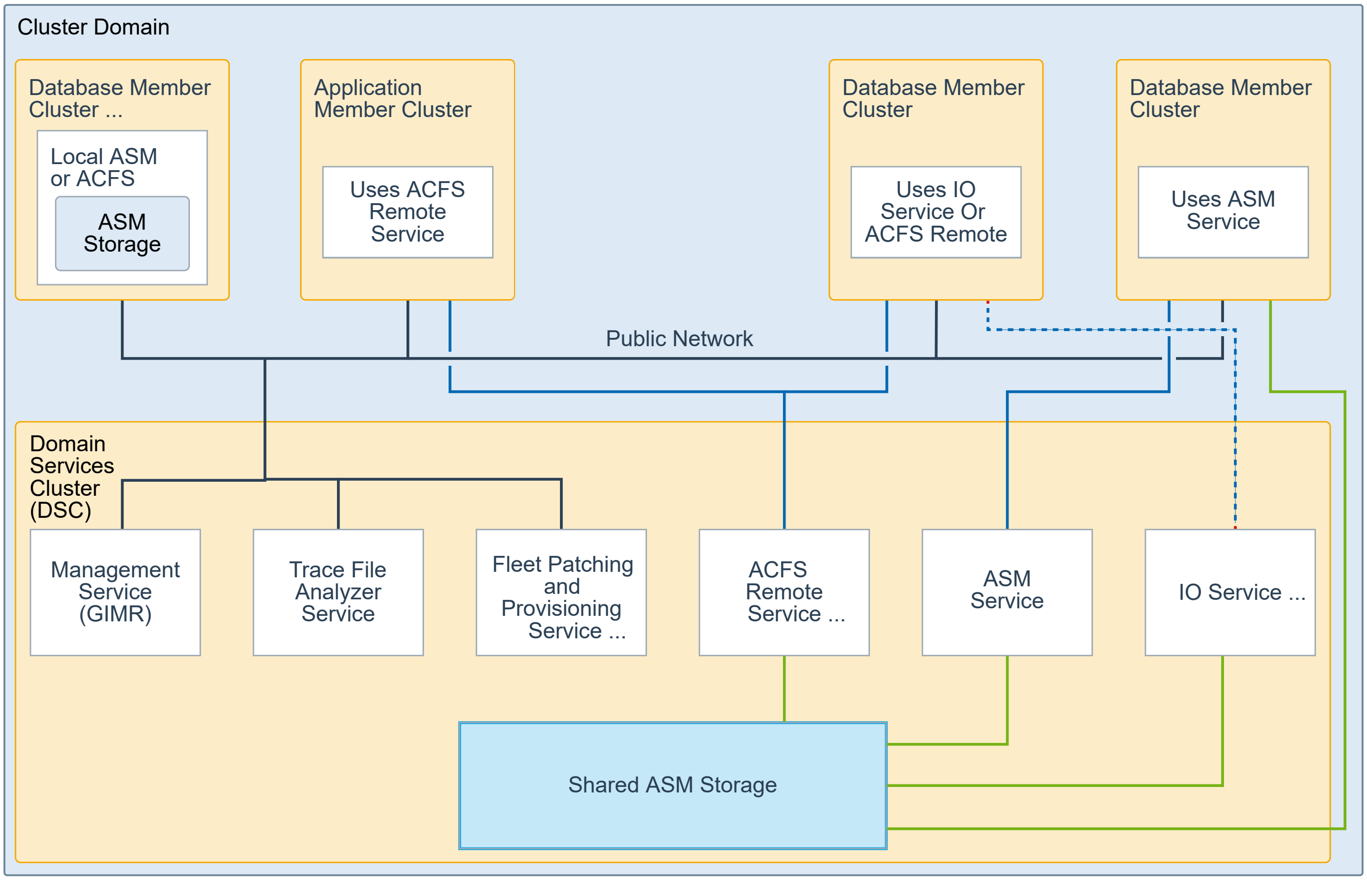
- Oracle Automatic Storage Management (strongly recommended).
- Network file system (NFS), which is not supported on Linux on POWER or on IBM zSeries Based Linux.
- OCFS2 for Linux, or General Parallel File System (GPFS) on IBM platforms.
- Oracle also provides ASM Cluster File system (ACFS), which is a general purpose cluster file system built on a ASM cluster

The storage is presented to the Operating system as block storage such as a partition, or as a file system such as typical mount point and directory structure as with Oracle ASM Cluster File system.

Any shared storage must be cluster aware and certified for use with Oracle RAC Database.

Note: You cannot use OUI to install Oracle Grid Infrastructure binaries on block or raw devices. You cannot put Oracle Grid Infrastructure binaries and files on Oracle Automatic Storage Management Cluster File System (Oracle ACFS).

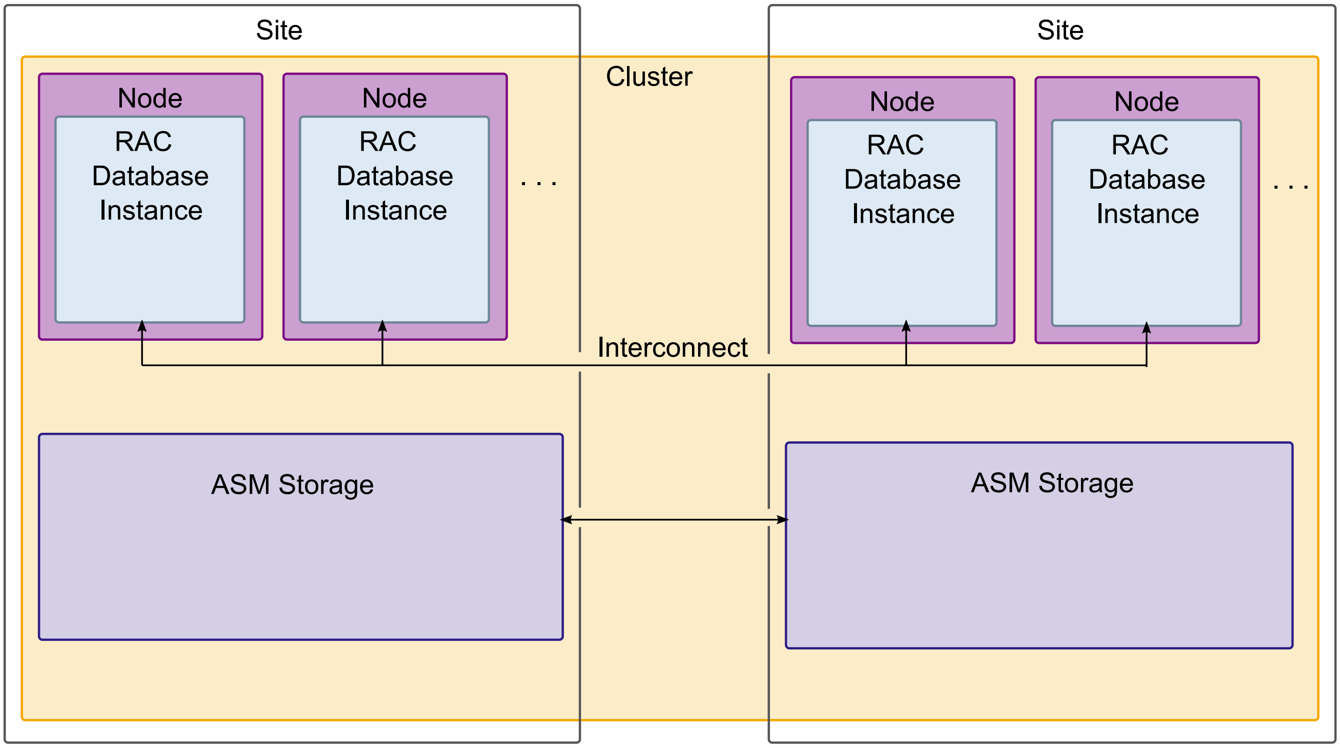
Cluster Domain



The Standalone Cluster consists of one or more cluster nodes configured with locally available shared storage, a private interconnect, local instances of Automatic Storage Management (ASM) for managing that shared storage, and optionally the GI Management Repository (GIMR) for storing cluster health and diagnostic information.

A Cluster Domain is a grouping of clusters. A Cluster Domain consists of a single Domain Services Cluster and a number of Member Clusters (hosting applications or databases) that utilize services offered on the Domain Services Cluster. Centralized and consolidated services are hosted by the Domain Services Cluster, and consumed by the Member Clusters that are registered with that Domain Services Cluster.

Extended Cluster Overview



An Oracle Extended Cluster consists of nodes that are located in multiple locations called sites. For more information see Oracle Extended Clusters.

When you deploy an Oracle Standalone Cluster, you can also choose to configure the cluster as an Oracle Extended Cluster. You can extend an Oracle RAC cluster across two, or more, geographically separate sites, each equipped with its own storage. In the event that one of the sites fails, the other site acts as an active standby. The Storage is replicated, so that each site may function when the other is unavailable.

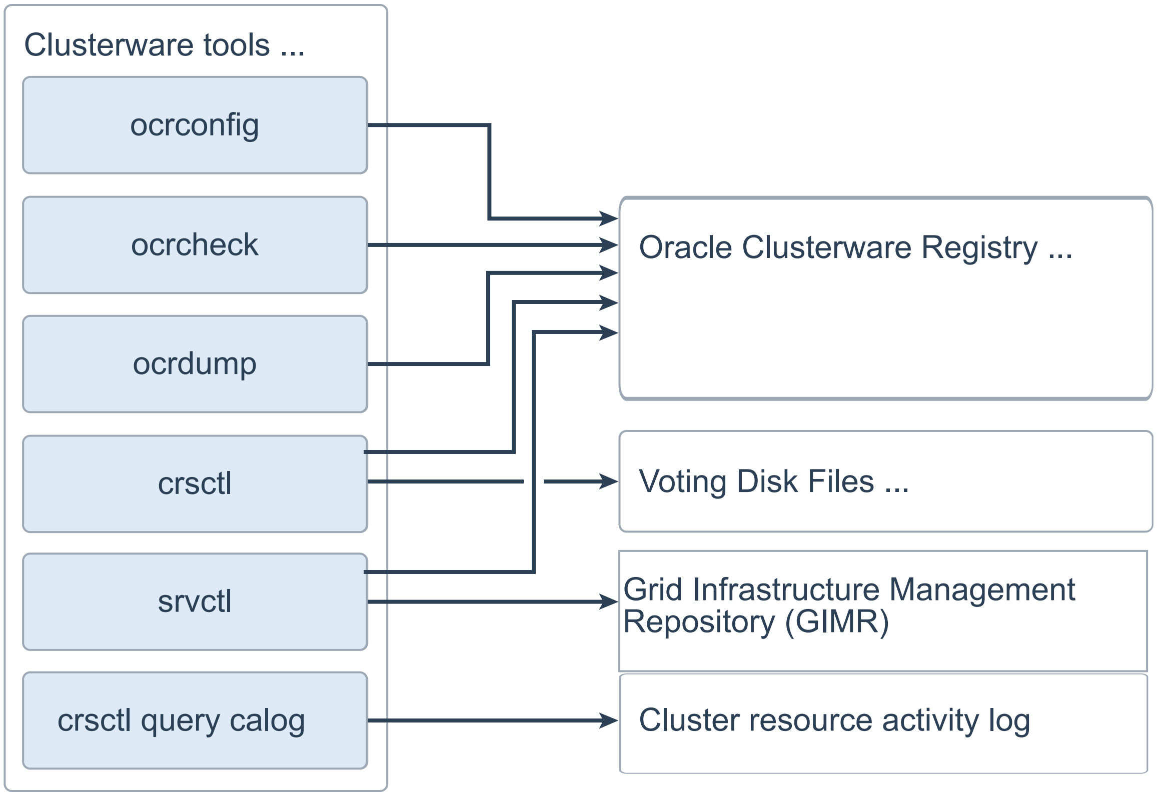
Both Oracle ASM and the Oracle Database stack, in general, are designed to use enterprise-class shared storage in a data center. Fibre Channel technology, however, enables you to distribute compute and storage resources across two or more data centers, and connect them through Ethernet cables and Fibre Channel, for compute and storage needs, respectively.

You can configure an Oracle Extended Cluster when you install Oracle Grid Infrastructure. You can also do so post installation using the ConvertToExtended script. You manage your Oracle Extended Cluster using CRSCTL.

Oracle recommends that you deploy Oracle Extended Clusters with normal redundancy disk groups. You can assign nodes and failure groups to sites. Sites contain failure groups, and failure groups contain disks. For normal redundancy disk groups, a disk group provides one level of failure protection, and can tolerate the failure of either a site or a failure group.

Note: Recommended distance between sites is less than 100km.

Grid Infrastructure Files and Management Database

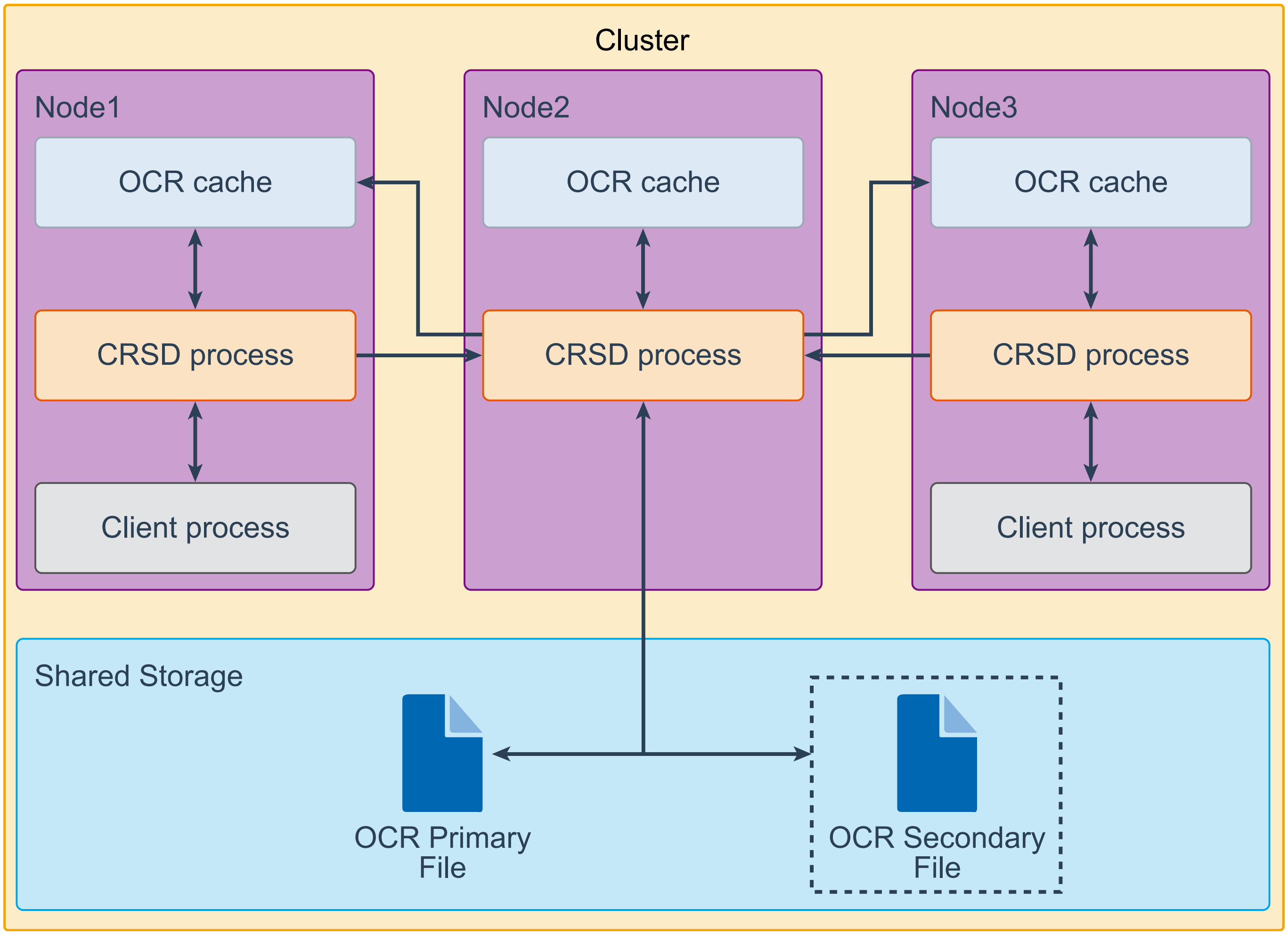

Clusterware tools ...
ocrconfig
ocrcheck
ocrdump
crsctl
srvctl
crsctl query calog
Oracle Clusterware Registry ...
Voting Disk Files ...
Grid Infrastructure Management Repository (GIMR)
Cluster resource activity log

Oracle 19c Grid Infrastructure requires certain files and provides tools to access each. Each Standalone Cluster should have a Grid Infrastructure Management Repository (GIMR) installed or be a part of a Domain Service Cluster having a GIMR. In Oracle 19c Grid Infrastructure, the GIMR is not required for a Standalone Cluster but recommended. If the GIMR is not installed and configured several features are not available, such as: Cluser Health Advisor, and Fleet Patching and Provisioning.

Oracle Clusterware uses voting files to provide node fencing using the disk storage heartbeat and cluster node membership information. Oracle Cluster Registry (OCR) provides cluster configuration and resource information. Both CRSVTL and SVRCTL will update the cluster resource information in the OCR, and the OLR's as necessary. Collectively, voting files and OCR are referred to as Oracle Clusteware files.

Oracle Clusterware files must be stored on Oracle ASM. If the underlying storage for the Oracle ASM disks is not hardware protected, such as RAID, then Oracle recommends that you configure multiple locations for OCR and voting files.

Oracle Clusterware Registry (OCR)

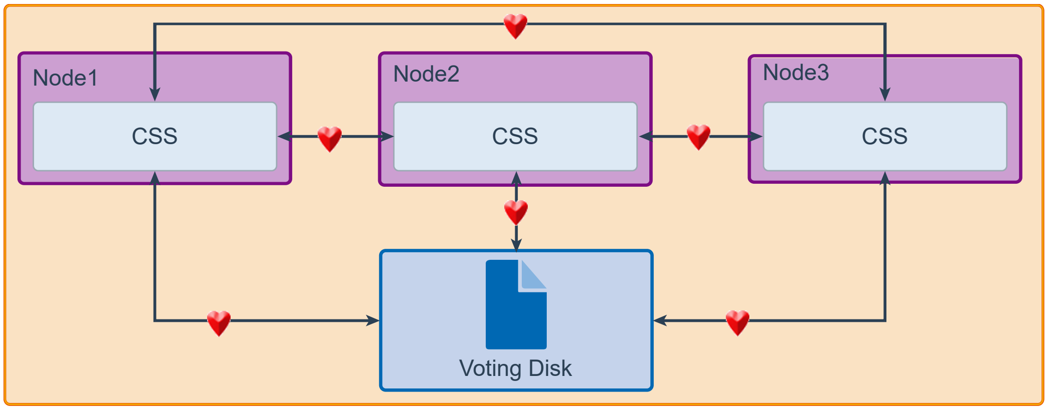


Cluster configuration information is maintained in the OCR. You can have up to five OCR locations. Each OCR location must reside on shared storage that is accessible by all nodes in the cluster. The OCR relies on distributed shared cache architecture for optimizing queries, and clusterwide atomic updates against the cluster repository. Each node in the cluster maintains an in-memory copy of OCR, along with the CRSD that accesses its OCR cache. Only one of the CRSD processes actually reads from and writes to the OCR file on shared storage. This process is responsible for refreshing its own local cache, as well as the OCR cache on other nodes in the cluster. For queries against the cluster repository, the OCR clients communicate directly with the local CRS daemon (CRSD) process on the node from which they originate. When clients need to update the OCR, they communicate through their local CRSD process to the CRSD process that is performing input/output (I/O) for writing to the repository on disk.

The main OCR client applications are CRSCTL, OUI, SRVCTL, Enterprise Manager (EM), the Database Configuration Assistant (DBCA), the Database Upgrade Assistant (DBUA), Network Configuration Assistant (NETCA), and the ASM Configuration Assistant (ASMCA). Furthermore, OCR maintains dependency and status information for application resources defined within Oracle Clusterware, specifically databases, instances, services, and node applications.

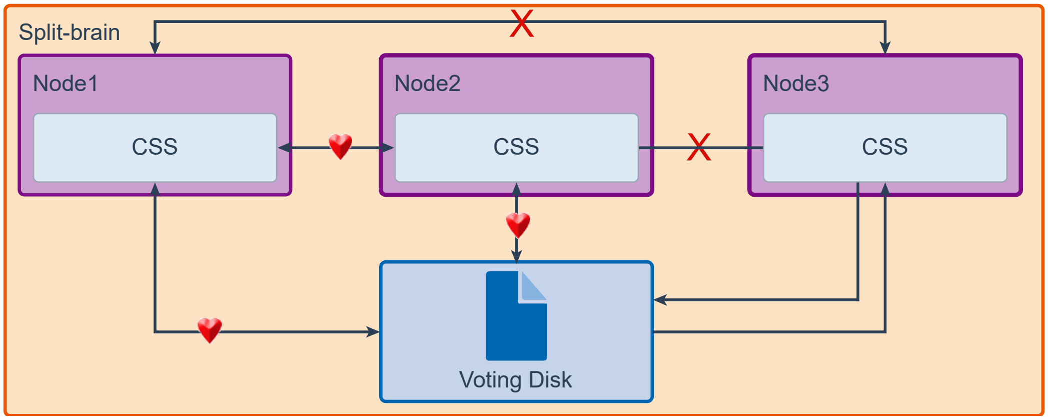
Note: In the diagram in the slide, note that a client process might also exist on node 2 but is not shown for the sake of clarity.

Voting Disk Files



Nodes can see each other by heartbeat sent through the private interconnect and by heartbeat written to the voting disk.

Node 3 can no longer communicate through private interconnect. The other nodes no longer see its heartbeats and evict node 3.



CSS is the service that determines which nodes in the cluster are available, provides cluster group membership and simple locking services to other processes. CSS typically determines node availability via communication through the private interconnect with a voting file used as a secondary communication mechanism. This is done by sending network heartbeat messages through the private network and writing a disk heartbeat to the voting file, as illustrated by the top graphic in the slide. The voting file resides on a clustered file system that is accessible to all nodes in the cluster. Its primary purpose is to help in situations where the private network communication fails. The voting file is then used to communicate the node state information to the other nodes in the cluster. Without the voting file, it can be difficult for isolated nodes to determine whether it is experiencing a network failure or whether the other nodes are no longer available. It would then be possible for the cluster to enter a state where multiple subclusters of nodes would have unsynchronized access to the same database files. The bottom graphic illustrates what happens when Node3 can no longer send network heartbeats to other members of the cluster. When other nodes can no longer see Node3’s heartbeats, they decide to evict that node by using the voting disk. When Node3 reads the removal message or “kill block,” it generally reboots itself to ensure that all outstanding write I/Os are lost.

A "split brain" condition is generally where one of the nodes cannot talk to other nodes via network but is still able to access voting disk. and can happen for any number of reasons. The primary causes are that a cluster node is unable to respond to the heartbeat requests from another node. This can be causes by network failure/interruptions, hardware failures, software failures or resource starvation (probably the most common cause). There are many, many causes, but Oracle Clusterware has a very conservative design, to absolutely guarantee the integrity of the cluster and the data.

Clusterware Tools for Installing, Configuring, and Managing Oracle RAC

Tool or Utility	Purpose or Use
Cluster Verification Utility (CVU)	Validates a range of cluster and Oracle RAC components such as shared storage devices, networking configurations, system requirements, and Oracle Clusterware, also operating system groups and users.
ORAchk	Performs proactive heath checks for the Oracle software stack.
Oracle Enterprise Manager	Oracle Enterprise Manager Cloud Control is a graphical user interface (GUI) that you use to manage single-instance and Oracle RAC databases, Oracle Clusterware, and Oracle ASM.
Server Control (SRVCTL)	Manages the resources defined in the Oracle Cluster Registry (OCR). These resources include the node applications, called nodeapps, that comprise Oracle Clusterware, which includes the Oracle Notification Service (ONS), and the Virtual IP (VIP).
Cluster Ready Services Control (CRSCTL)	Manages Oracle Clusterware daemons. These daemons include Cluster Synchronization Services (CSS), Cluster-Ready Services (CRS), and Event Manager (EVM).
ocrcheck	The OCRCHECK utility displays the version of the OCR's block format, total space available and used space, OCRID, and the OCR locations that you have configured.
ocrdump	The OCRDUMP utility enables you to view OCR and OLR contents by writing the content to a file or stdout in a readable format.
ocrconfig	Use the ocrconfig command to manage OCR. Using this utility you can import, export, add, delete, restore, overwrite, backup, repair, replace, move, upgrade, or downgrade OCR.

The database administration tasks such as managing space usage, security, and performance use the same tools as in a single instance database. The tools shown here are additional tools and utilities to manage the unique aspects of a cluster and a RAC database.

If you have installed Oracle Grid Infrastructure for a cluster for Oracle Database 18c (18.x), then, when using utilities to manage your cluster, databases, database instances, Oracle ASM, and listeners, use the appropriate binary that is in the home directory of the object or component you are managing. Set the ORACLE_HOME environment variable to point to this directory, for example: If you use asmcmd, srvctl, lsnrctl or SQL*Plus to manage Oracle ASM or its listener, then use the executable files located in the grid software home, not the executable files located in the database software home, by setting the ORACLE_HOME environment variable to the location of the Grid home. If you use srvctl, lsnrctl or SQL*Plus to manage a database instance or its listener, then use the binaries located in the database software home where the database instance or its listener is running, and set the ORACLE_HOME environment variable to the location of that database software home.

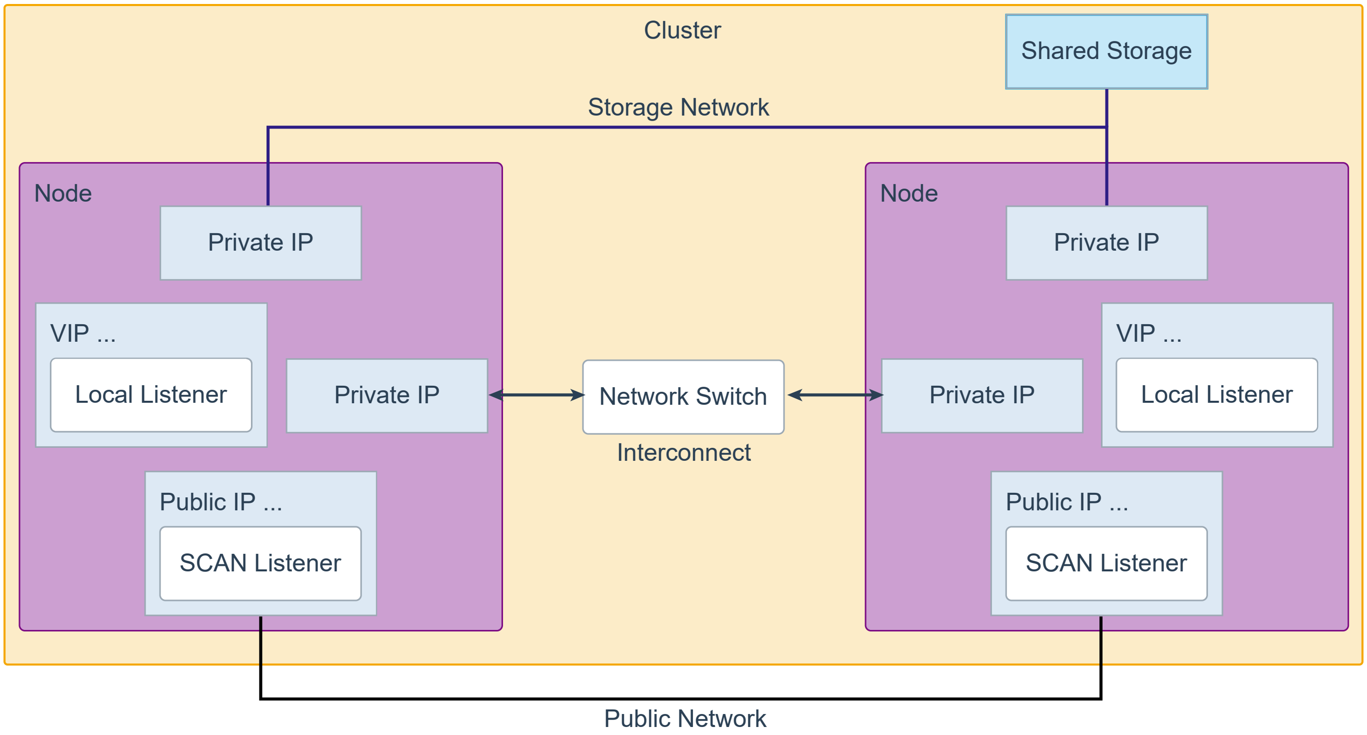
Several of the RAC tools are also listed on the Clusterware Tools page because some tools may be used for either clusterware or RAC database. For example: The Server Control utility has different functions available depending on whether it is started from the database software home, or the Grid Infrastructure software home. CRSCTL and SVRCTL have some overlapping functions.

CRSCTL is the command line interface for the Cluster Administrator to manager Clusterware resources, that are NOT RAC resources. CRSCTL is started from the Grid software home as a user with privileges for Cluster management.

SRVCTL is primarily used for managing RAC database resources. For this purpose, it must be started from the Oracle database software home. SRVCTL also has some Cluster resource functions. For these functions it must be started from the Grid software home

Oracle Enterprise Manager: Oracle Enterprise Manager Cloud Control is a GUI for managing both noncluster database and Oracle RAC database environments. Oracle recommends that you use Oracle Enterprise Manager to perform administrative tasks whenever feasible. You can also manage Oracle RAC One Node databases using Oracle Enterprise Manager Cloud Control.

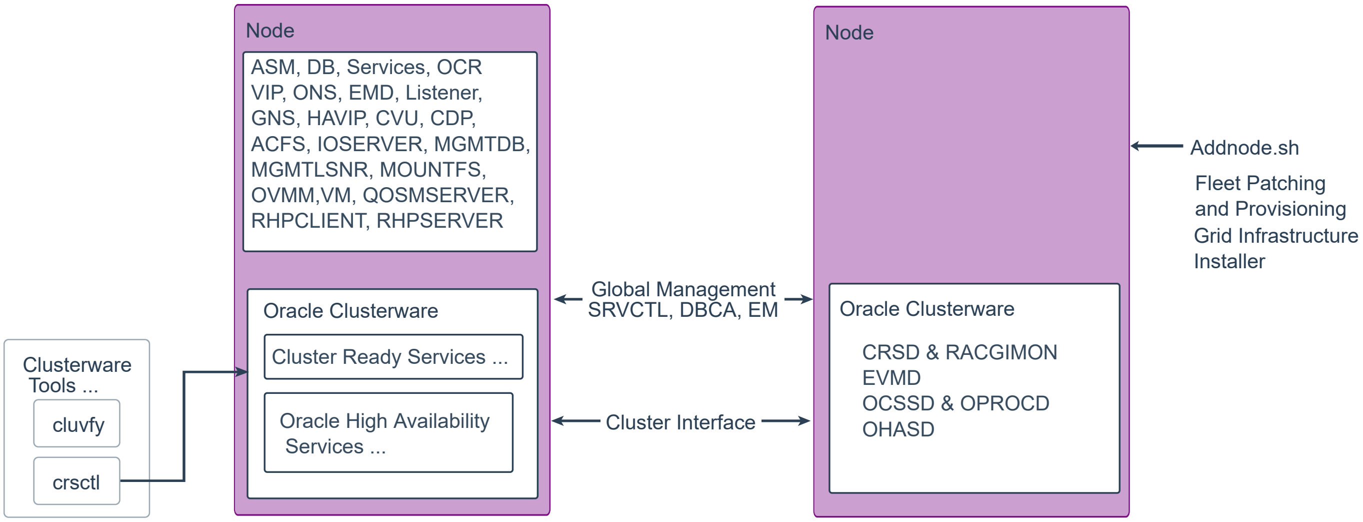
Server Network Configuration



Each Cluster requires certain network resources. It must have a public network access for clients and a private network for the cluster interconnect. For more inforamtion about the Network Interface requirements see Network Interface Hardware Minimum Requirements. A minimum is 2 network interface cards. Oracle also supports the use of link aggregations, bonded, trunked or teamed networks for improved bandwidth and high availability.

The storage network is only needed for network attached storage.

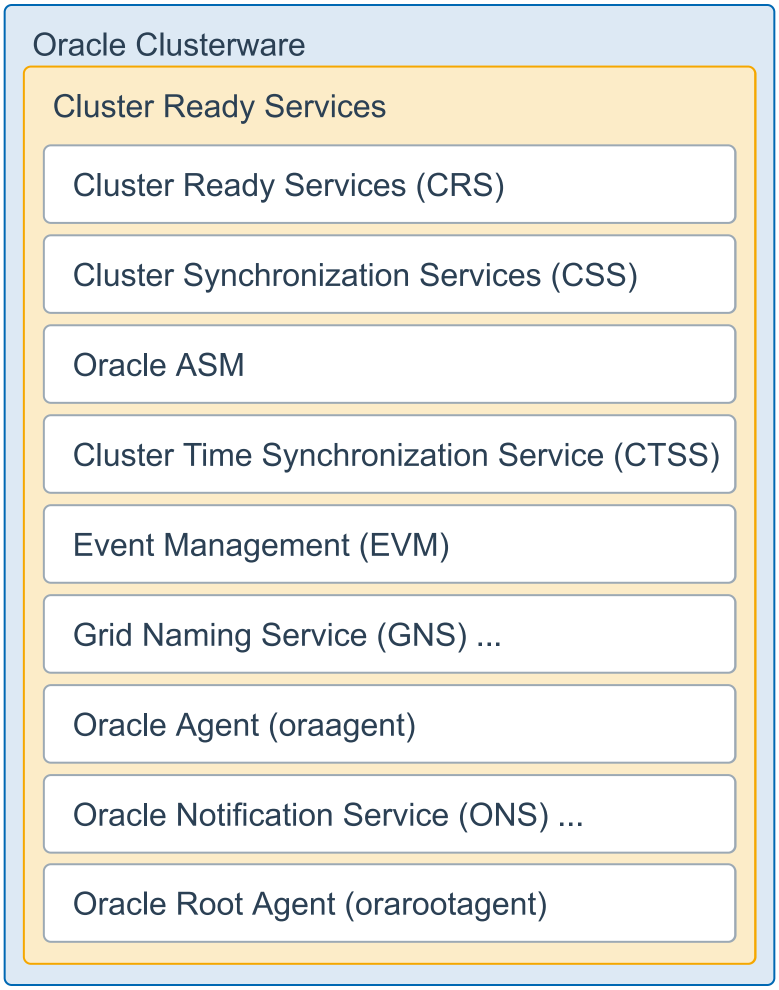
Clusterware Processes and Services



Oracle Clusterware consists of two separate technology stacks: an upper technology stack anchored by the Cluster Ready Services (CRS) daemon (CRSD) and a lower technology stack anchored by the Oracle High Availability Services daemon (OHASD). The Clusterware components vary slightly based on the platform.

You can use the addnode.sh script to add nodes to an existing cluster. In Oracle 19c Grid Infrastructure, you can use the installer, gridSetup.sh or the Fleet Patching and Provisioning utility rhpctl to add a new node to an existing cluster. for more information see Adding a Cluster Node on Linux and UNIX Systems

Cluster Ready Services



Oracle Clusterware

Cluster Ready Services

- Cluster Ready Services (CRS)
- Cluster Synchronization Services (CSS)
- Oracle ASM
- Cluster Time Synchronization Service (CTSS)
- Event Management (EVM)
- Grid Naming Service (GNS) ...
- Oracle Agent (oraagent)
- Oracle Notification Service (ONS) ...
- Oracle Root Agent (orarootagent)

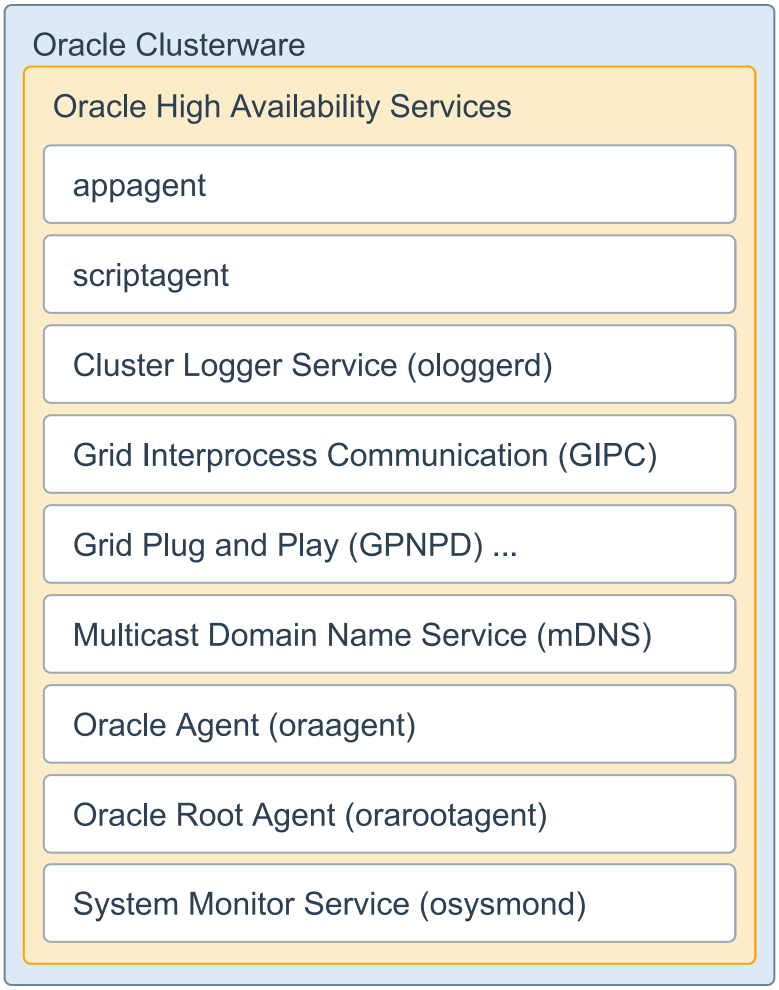
The Cluster Ready Services (CRS) technology stack leverages several processes to manage various services.

Cluster Ready Services (CRS) is the primary program for managing high availability operations in a cluster.

The CRSD manages cluster resources based on the configuration information that is stored in OCR for each resource. This includes start, stop, monitor, and failover operations. The CRSD process generates events when the status of a resource changes. When you have Oracle RAC installed, the CRSD process monitors the Oracle database instance, listener, and so on, and automatically restarts these components when a failure occurs.

Even though CRS and High Availability Services have programs named the same they are distinctly different, such as: oraagent and orarootagent.

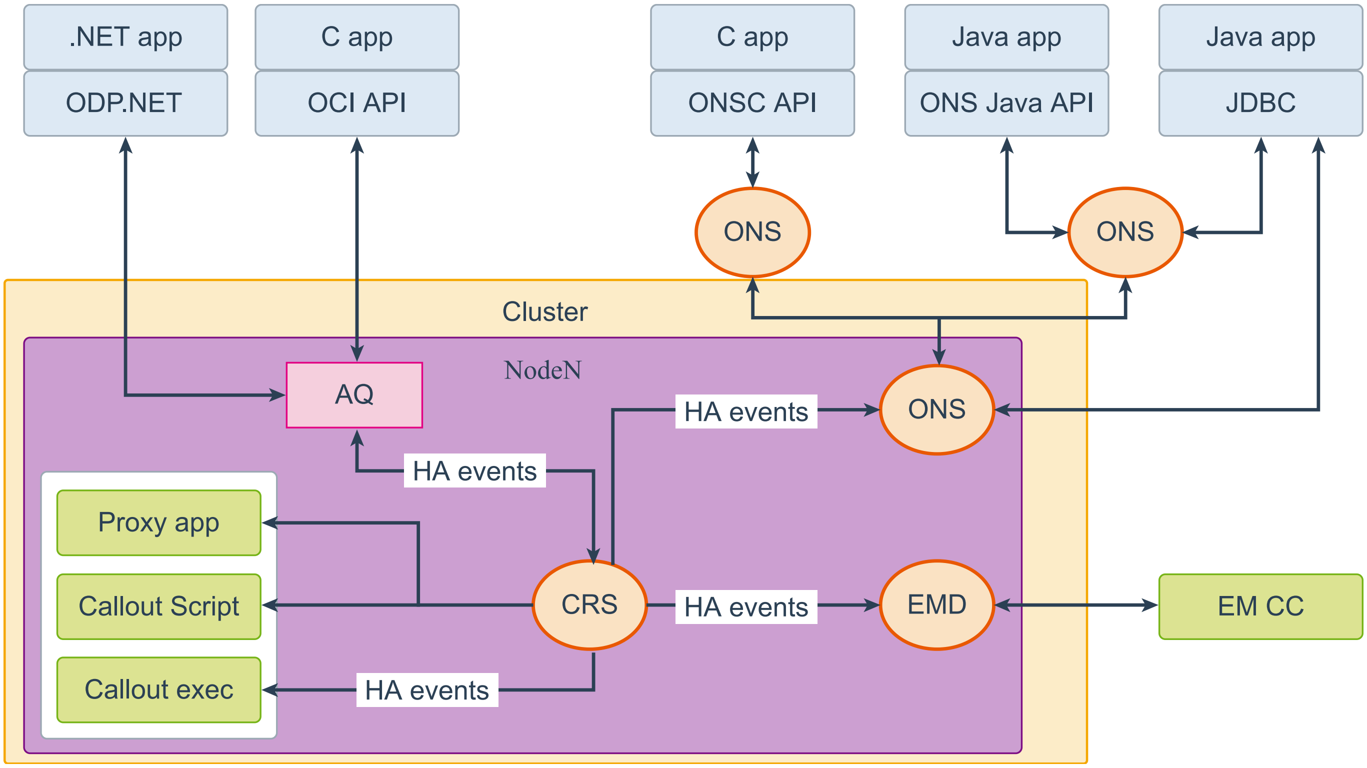
Oracle High Availability Services


Oracle Clusterware
Oracle High Availability Services
appagent
scriptagent
Cluster Logger Service (ologgerd)
Grid Interprocess Communication (GIPC)
Grid Plug and Play (GPNPD) ...
Multicast Domain Name Service (mDNS)
Oracle Agent (oraagent)
Oracle Root Agent (orarootagent)
System Monitor Service (osysmond)

Oracle High Availability Services technology stack stack uses several processes shown in the diagram to provide Oracle Clusterware high availability. For a complete list and descriptions see The Oracle High Availability Services Technology Stack

Even though CRS and High Availability Services have programs named the same they are distinctly different, such as: oraagent and orarootagent.

Oracle Notification Service (ONS)

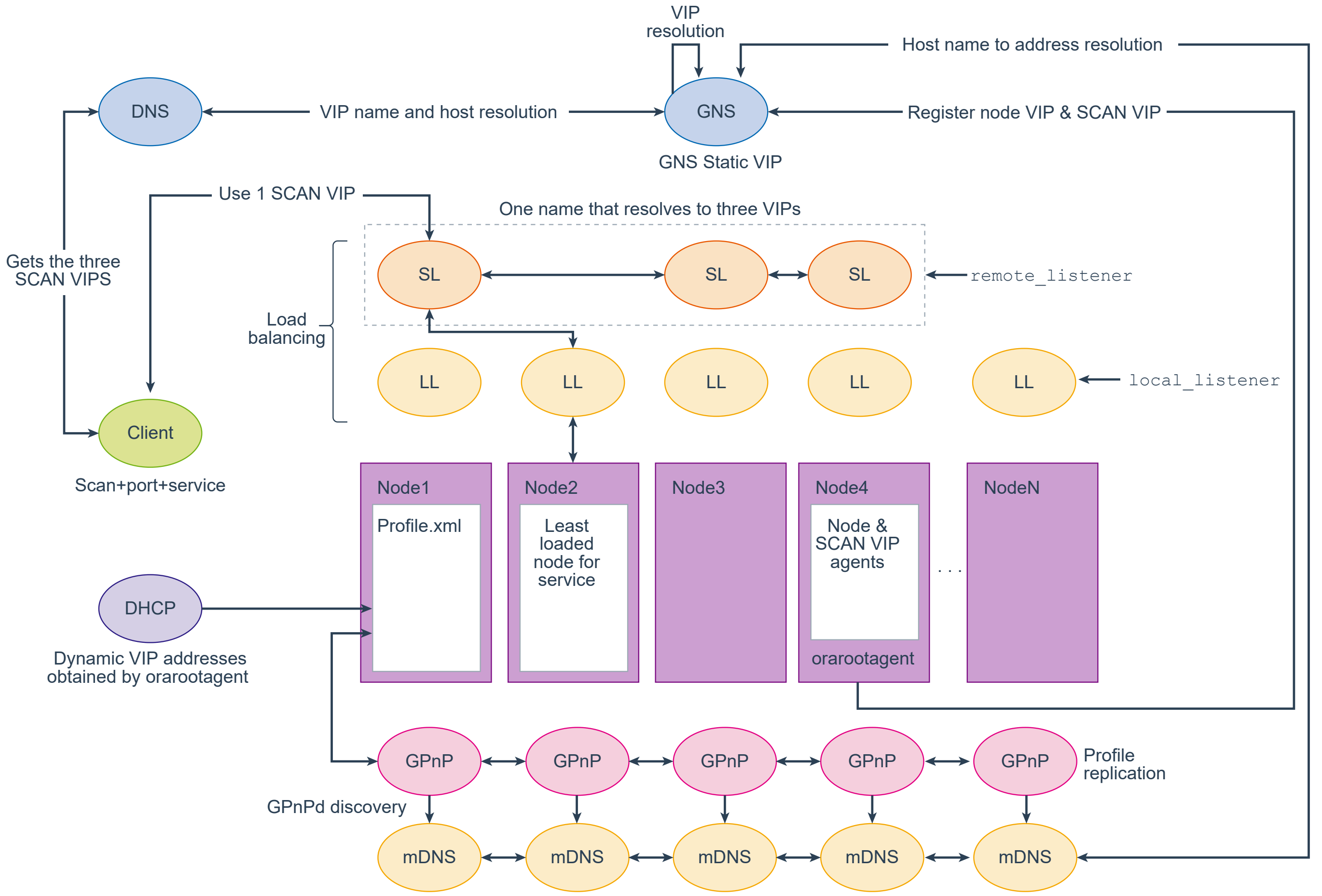


HA events are generated when resources change state within an Oracle Clusterware environment. Oracle Notification Service (ONS) is a facility that creates a bridge with middle-tier servers or applications to transport these events to application logic for handling or reaction. ONS is part of a larger framework known as Fast Application Notification (FAN). With FAN, applications use these events to achieve very fast detection of failures and rebalancing of connection pools, following failures and recovery. When FAN is used together with an Oracle Database, the Advanced Queuing (AQ) feature allows HA events to be received by external applications such as .NET clients. The easiest way to receive all the benefits of FAN, with no effort, is to use a client that is integrated with FAN, such as:

- Java Database Connectivity (JDBC) Implicit Connection Cache
- User-extensible callouts
- Connection Manager (CMAN)
- Listeners
- Oracle Notification Service (ONS) API
- OCI Connection Pool or Session Pool
- Transparent Application Failover (TAF)
- ODP.NET Connection Pool

Note: Not all the preceding applications can receive all types of FAN events.

Global Names Service



GPnP Service The GPnP service is collectively provided by all the GPnP agents. It is a distributed method of replicating profiles. The service is instantiated on each node in the domain as a GPnP agent. The service is peer-to-peer; there is no master process. This allows high availability because any GPnP agent can crash and new nodes will still be serviced. GPnP requires standard IP multicast protocol (provided by mDNS) to locate peer services. Using multicast discovery, GPnP locates peers without configuration. This is how a GPnP agent on a new node locates another agent that may have a profile it should use.

Name Resolution A name defined within a GPnP domain is resolvable in the following cases:

- Hosts inside the GPnP domain use normal DNS to resolve the names of hosts outside of the GPnP domain. They contact the regular DNS service and proceed. They may get the address of the DNS server by global configuration or by having been told by DHCP.
- Within the GPnP domain, host names are resolved using mDNS. This requires an mDNS responder on each node that knows the names and addresses used by this node, and operating system client library support for name resolution using this multicast protocol. Given a name, a client executes gethostbyname, resulting in an mDNS query. If the name exists, the responder on the node that owns the name will respond with the IP address. The client software may cache the resolution for the given time-to-live value.
- Machines outside the GPnP domain cannot resolve names in the GPnP domain by using multicast. To resolve these names, they use their regular DNS. The provisioning authority arranges the global DNS to

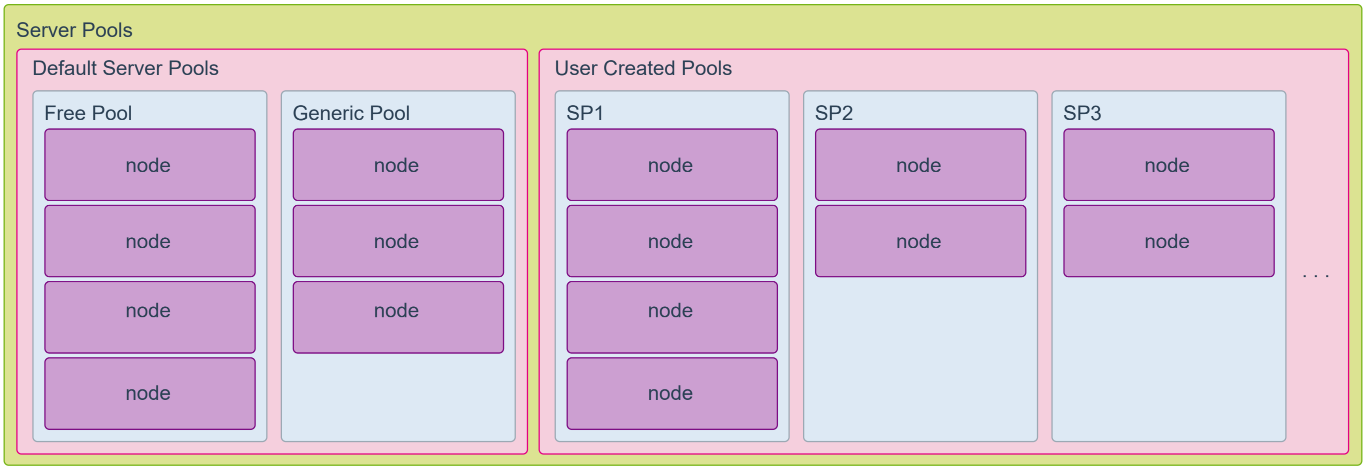
delegate a subdomain (zone) to a known address that is in the GPnP domain. GPnP creates a service called GNS to resolve the GPnP names on that fixed address. The node on which the GNS server is running listens for DNS requests. On receipt, they translate and forward to mDNS, collect responses, translate, and send back to the outside client. GNS is "virtual" because it is stateless. Any node in the multicast domain may host the server. Only the GNS configuration is global:

  - The address on which to listen on standard DNS port 53
  - The name(s) of the domains to be serviced.

There may be as many GNS entities as needed for availability reasons. Oracle-provided GNS may use CRS to ensure availability of a single GNS provider.

SCAN and Local Listeners When a client submits a connection request, the SCAN listener listening on a SCAN IP address and the SCAN port are contacted on the client's behalf. Because all services on the cluster are registered with the SCAN listener, the SCAN listener replies with the address of the local listener on the least-loaded node where the service is currently being offered. Finally, the client establishes a connection to the service through the listener on the node where service is offered. All these actions take place transparently to the client without any explicit configuration required in the client. During installation, listeners are created on nodes for the SCAN IP addresses. Oracle Net Services routes application requests to the least loaded instance providing the service. Because the SCAN addresses resolve to the cluster, rather than to a node address in the cluster, nodes can be added to or removed from the cluster without affecting the SCAN address configuration.

Cluster Server Pools



Server Pools are groups of nodes in a single cluster.

When Oracle Clusterware is installed, two internal server pools are created automatically: Generic and Free. All servers in a new installation are assigned to the Free server pool, initially. Servers move from Free to newly defined server pools automatically.

Server pools provide resource isolation to prevent applications running in one server pool from accessing resources running in another server pool. Oracle Clusterware provides fine-grained role separation between server pools. This capability maintains required management role separation between these groups in organizations that have clustered environments managed by separate groups.

Oracle RAC databases support two types of server pool management styles and deployment models: Policy-based management and administrator-managed.

If you create an Oracle RAC One Node database, Oracle RAC One Node runs only in one server pool. This server pool is treated the same as any other server pool. Online relocation of an Oracle RAC One Node database instance permits planned migrations of an Oracle RAC One Node database from one node to another node. Relocations must always be within a server pool.

Policy-Based Management of Oracle RAC Databases With policy-based management, administrators specify the server pool (excluding the Generic and Free pools) in which the servers run. For example, a database administrator uses srvctl to create a server pool for servers hosting a database or database service. A clusterware administrator uses crsctl to create server pools for non-database use, such as creating a server pool for servers hosting an application. You can configure each server pool with a minimum and maximum size, which determines scalability. Oracle Clusterware manages availability between server pools, and you can further regulate availability by configuring the importance value of individual server pools.

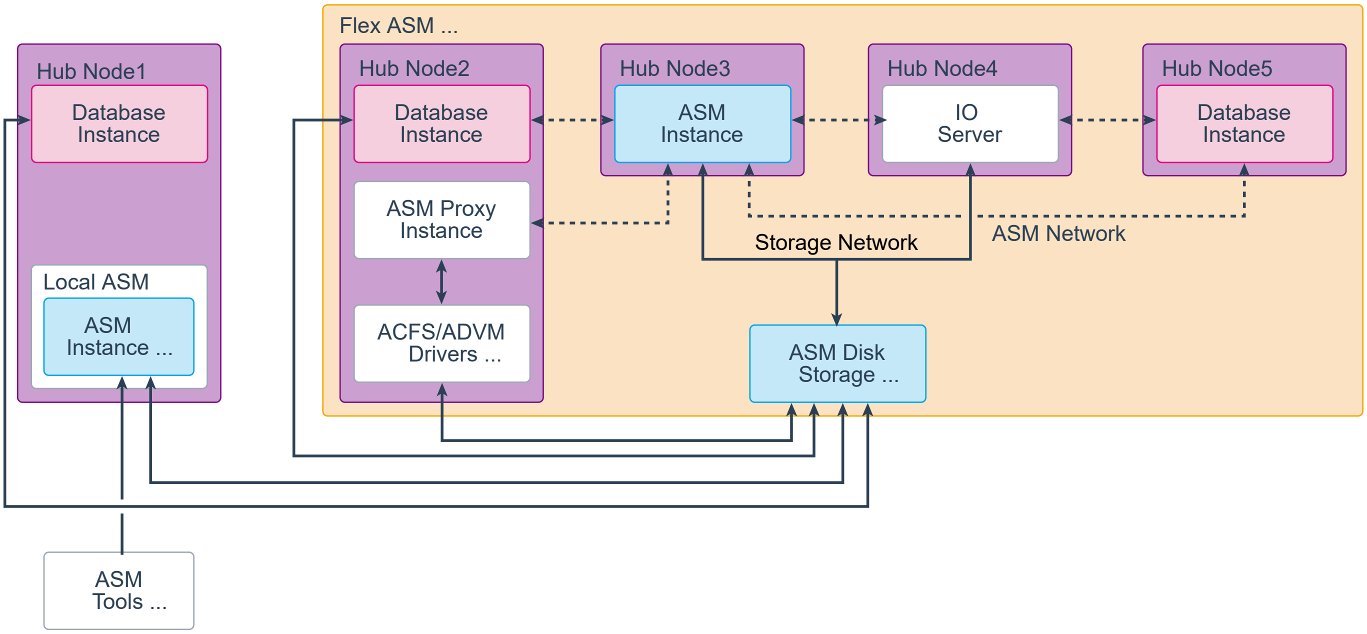
Policy-based management

- Enables online server reallocation based on a defined policy to satisfy workload capacity requirements.
- Guarantees the allocation of required resources for critical work as defined by the policy

- Ensures isolation where necessary, so that you can provide dedicated servers in a cluster for applications and databases
- Enables policies to be configured to change pools in accordance with business needs or application demand, so that pools provide the required capacity at the right time

Administrator-Managed Oracle RAC Databases Deployment is based on the Oracle RAC deployment types that existed before Oracle Database 11g release 2 (11.2) and requires that you statically configure each database instance to run on a specific node in the cluster, and that you configure database services to run on specific instances belonging to a certain database using the preferred and available designation. When you review the database resource for an administrator-managed database, you see a server pool defined with the same name as the Oracle database. This server pool is part of a special Oracle-defined server pool called Generic. Oracle RAC manages the Generic server pool to support administrator-managed databases. When you add or remove an administrator-managed database using either srvctl or DBCA, Oracle RAC creates or removes the server pools that are members of Generic.

Oracle Automatic Storage Management(ASM) Configurations



All ASM installations can be configured to serve Oracle ASM Flex Clients.

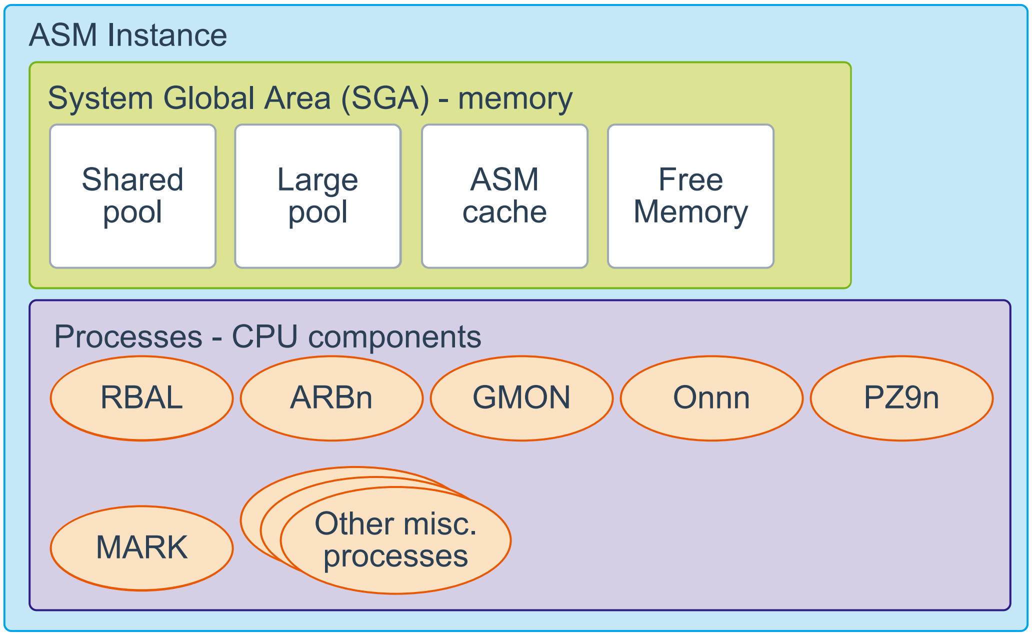
On the left, Hub Node 1 is the traditional Standalone configuration with the ASM instance local to the database instance. In the past this was the only configuration available. Each node had to have an ASM instance, if ASM was used to provide shared storage.

Notice that Hub node 2, is an ASM client node. It uses the ASM service on hub node 3, another node in the cluster, to access the metadata, and then does direct access to ASM disk storage for the data blocks.

To the right, is hub node 5. It has no direct access to the ASM disk storage, but gets all the data through the IO Server (IOS) on hub node 4. The Oracle IO Server was introduced in Oracle Grid Infrastructure version 12.2. IOS enables you to configure client clusters on such nodes. On the storage cluster, clients send their IO requests to network ports opened by an IOServer. The IOServer instance receives data packets from the client and performs the appropriate IO to Oracle ASM disks similar to any other database client. On the client side, databases can use dNFS to communicate with an IOServer instance. However, there is no client side configuration so you are not required to provide a server IP address or any additional configuration information. On nodes and clusters that are configured to access Oracle ASM files through IOServer, the discovery of the Oracle IOS instance occurs automatically.

Each ASM instance in a cluster has access to the ASM disk storage in that cluster. ASM disks are shared disks attached to the nodes of the cluster, in possibly varying ways, as shown in the graphic. ASM manages Disk Groups rather than individual disks. The ASM utilities allow you to add disks, partitions, logical volumes or Network attached files (NFS) to a disk group.

ASM Instance



Every time ASM or a database is started, a shared memory area called the System Global Area (SGA) is allocated and the ASM background processes are started. However, because ASM performs fewer tasks than a database, an ASM SGA is much smaller than a database SGA. The combination of background processes and the SGA is called an Oracle ASM instance. The instance represents the CPU and RAM components of a running ASM environment.

The SGA in an ASM instance is different in memory allocation and usage than the SGA in a database instance. The SGA in the ASM instance is divided into four primary areas as follows:

- Shared Pool: Used for metadata information
- Large Pool: Used for parallel operations
- ASM Cache: Used for reading and writing blocks during rebalance operations
- Free Memory: Unallocated memory available

The minimum recommended amount of memory for an ASM instance is 256 MB. Automatic memory management is enabled by default on an ASM instance and will dynamically tune the sizes of the individual SGA memory components. The amount of memory that is needed for an ASM instance will depend on the amount of disk space that is being managed by ASM.

The second part of the ASM instance is the background processes. An ASM instance can have many background processes; not all are always present. Because the ASM instance shares the same code base as an Oracle database instance, all the required background processes of a database instance will exist in the ASM instance. There are required background processes and optional background processes. Some of these processes may include the following:

- CKPT: The checkpoint process
- DBWn: The database writer processes
- DIAG: The diagnosability process

- LGWR: The log writer process
- PMON: The process monitor process
- PSP0: The process spawner process
- SMON: The system monitor process
- VKTM: The virtual keeper of time process
- MMAN: The memory manager process

The preceding list of processes is not complete. There can be hundreds of database instance background processes running depending on the database options and configuration of the instance. For the ASM instance, these processes will not always perform the same tasks as they would in a database instance. For example, the LGWR process in a database instance copies change vectors from the log buffer section of the SGA to the online redo logs on disk. The ASM instance does not contain a log buffer in its SGA, nor does it use online redo logs. The LGWR process in an ASM instance copies logging information to an ASM disk group.

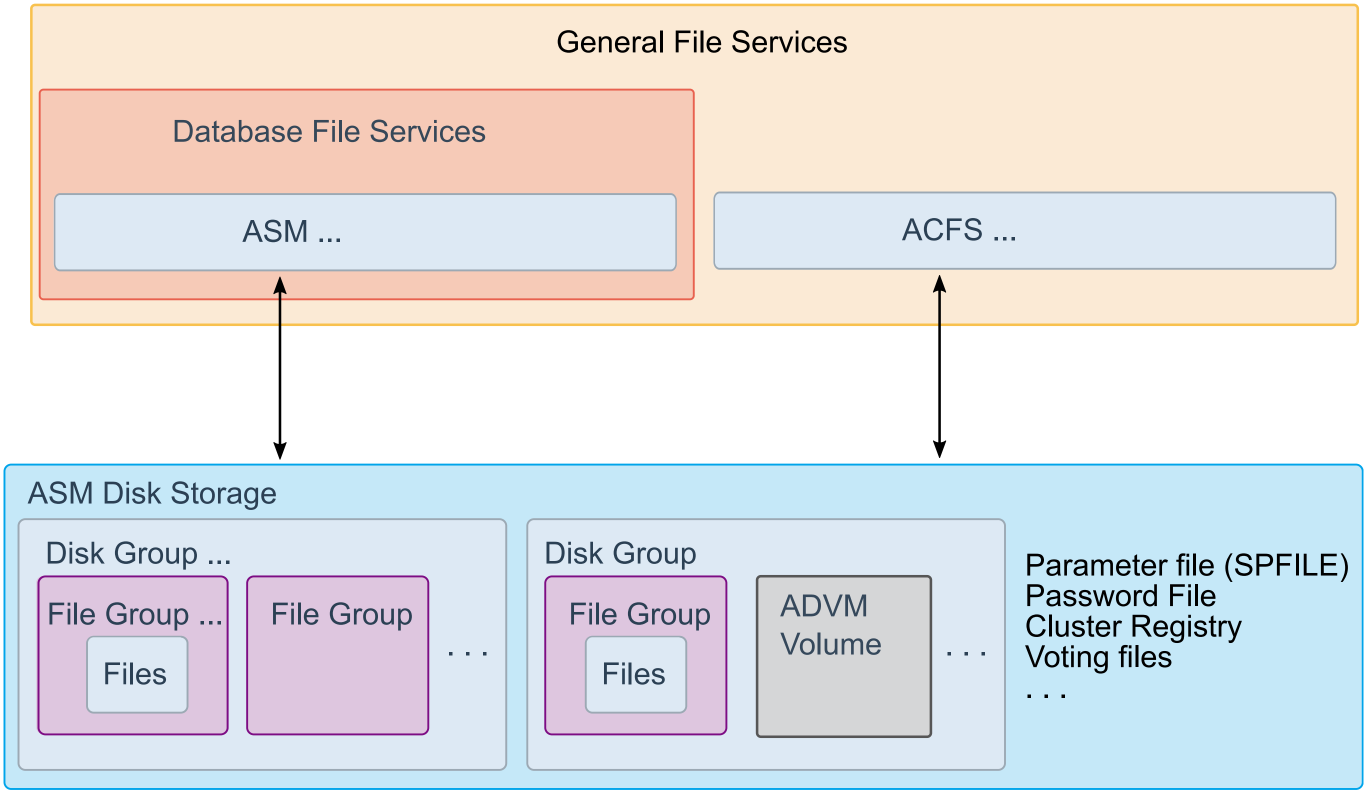
If ASM is clustered, additional processes related to cluster management will be running in the ASM instance. Some of these processes include the following:

- LMON: The global enqueue service monitor process
- LMDn: The global enqueue service daemons
- LMSn: The global cache service processes
- LCKn: The lock processes

Additional processes are started when ADVM volumes are configured.

- VDBG: The Volume Driver Background process forwards ASM requests to lock or unlock an extent for volume operations to the Dynamic Volume Manager driver. The VDBG is a fatal background process; termination of this process brings down the ASM instance.
- VBGn: The Volume Background processes wait for requests from the Dynamic Volume Manager driver, which need to be coordinated with the ASM instance. An example of such a request would be opening or closing an ASM volume file when the Dynamic Volume Manager driver receives an open for a volume (possibly due to a file system mount request) or close for an open volume (possibly due to a file system unmount request), respectively. Unplanned death of any of these processes does not have an effect on the ASM instance.
- VMB: Volume Membership Background coordinates cluster membership with the ASM instance.
- The ASM instance uses dedicated background processes for much of its functionality.

ASM Disk Storage



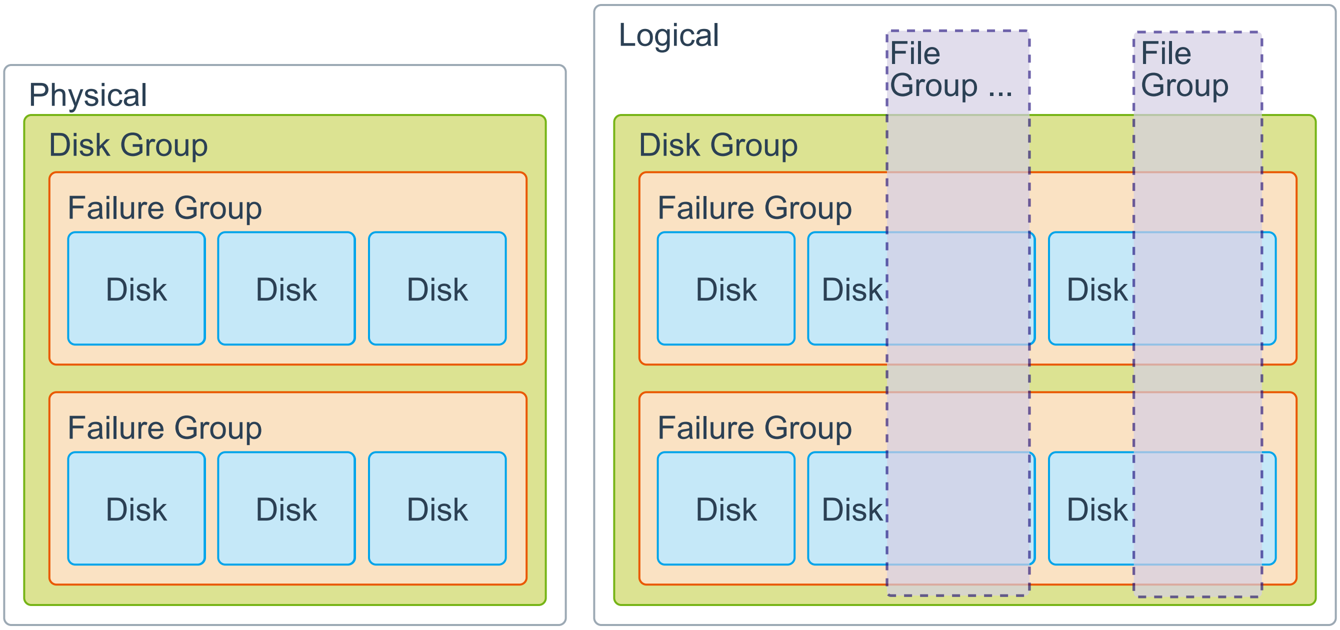
ASM and ACFS both use ASM disk storage. ASM presents disk groups to the database as mounted storage. ACSF presents ADVM volumes to the operating system as mount points.

ASM disk storage has two organizations: physical and logical. The physical is ASM disks. ASM disks are physical disk partitions that are assigned to ASM. These may full disks or disk partitions. Each disk group has two or more failure groups, except when redundancy is set to EXTERNAL.

The unit of organization of ASM disk storage is a disk group. Each disk group has two or more ASM disks assigned to a disk group. Each ASM disk can only be assigned to one disk group. Disk groups have properties that determine their behavior such as: redundancy, and failure groups.

You may set the redundancy property of the disk group to EXTERNAL (no-mirroring), NORMAL (two-way mirroring), HIGH (three-way mirroring), EXTENDED, or FLEX. See Creating a Disk Group for more information about Redundancy properties. When Redundancy is set to FLEX, the mirroring is set for the file group rather than the disk group.

ASM Disk Groups



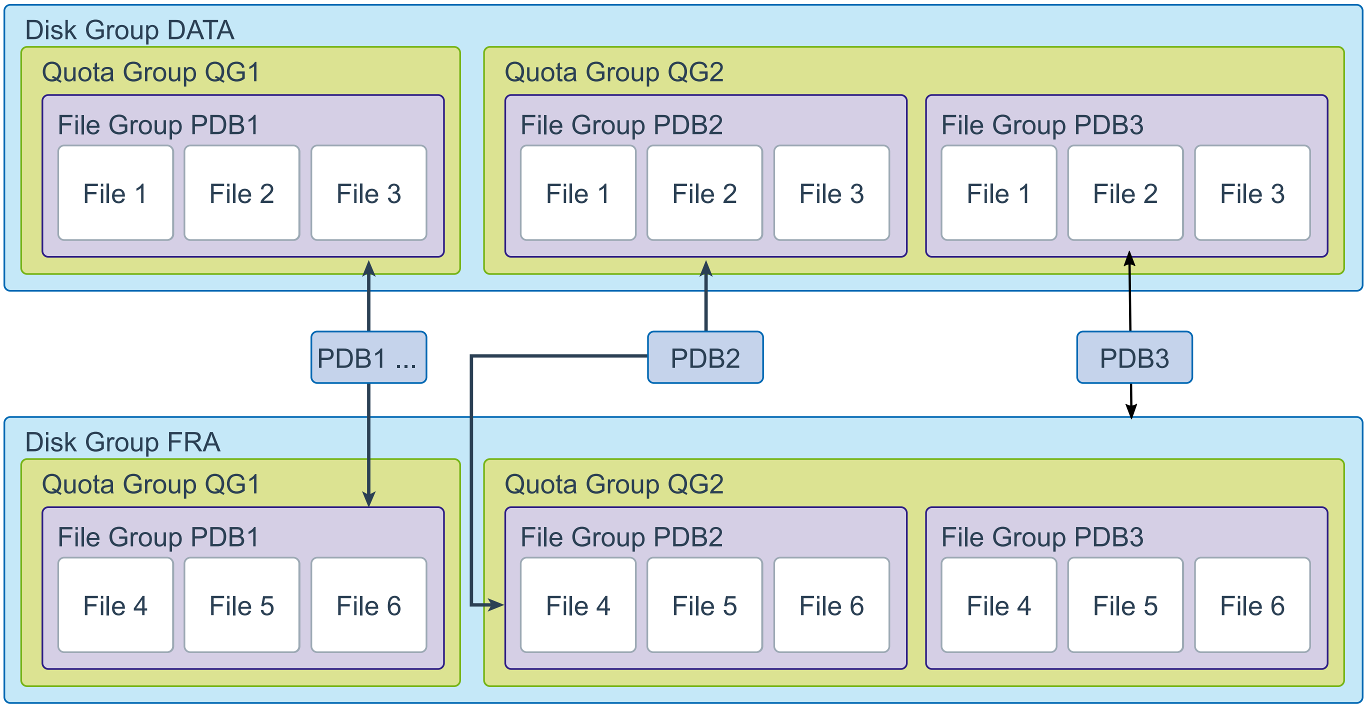
In ASM, the disk group is used to manage several storage properties such as: redundancy, access control, and allowed repair time before drop. Physical disks are assigned to disk groups. Disk groups provide the functionality of RAID with much more flexibility. Disk groups must have a unique name in the ASM instance. All disks in a disk group must be of equal size. A disk can belong to one and only one disk group.

You specify redundancy property for each disk group. The possible settings are EXTERNAL, NORMAL, HIGH, FLEX and EXTENDED. When you specify EXTERNAL redundancy, Oracle ASM does not provide mirroring. If you specify NORMAL redundancy Oracle ASM provides two-way mirroring by default for most files. When HIGH redundancy is specified Oracle ASM provides three-way mirroring. When you specify FLEX or EXTENDED redundancy the disk group redundancy is overridden by the file group redundancy property. EXTENDED redundancy provides two-way mirroring by default within a site, and assumes that you have a vendor supplied disk mirroring solution between sites.

By default each disk is in a different failure group, optionally disks may be grouped into failure groups. Normal redundancy requires a minimum of two disks or two failure groups. High redundancy requires a minimum of three disks or failure groups. A disk group will have a default redundancy that can be overridden at the file or file group level, if there are sufficient failure groups. The example shows only two failure groups so high redundancy is not possible. Depending on the redundancy setting the blocks of a file placed in the disk group or file group will be scattered across the required number of physical disks.

File Groups are optional, but provide more flexible storage management when multiple database, PDBs, or clusters are using the same ASM storage set. To use ASM File Groups, the Disk group redundancy must be set to FLEX or EXTENDED. File Groups allow you to control redundancy, and access by database to specified files within a disk group.

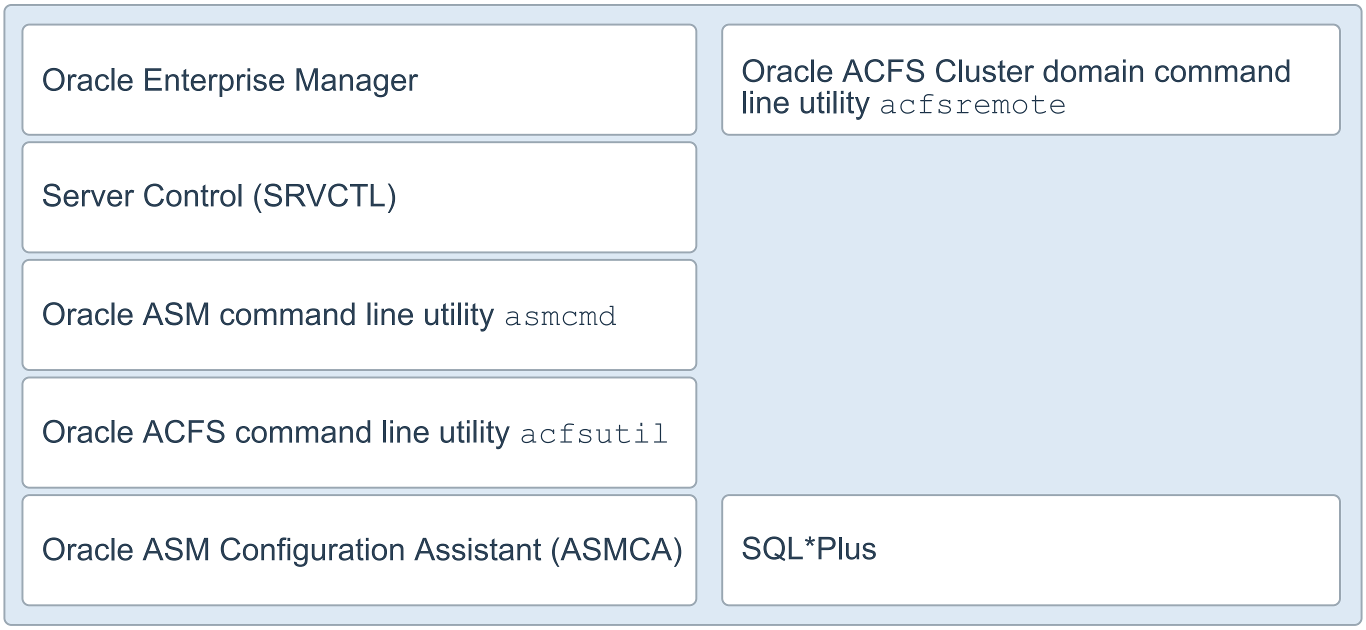
File Groups



The example illustrates file groups in a multitenant environment. Each PDB has a file group, the file group belongs to a quota group, and resides in a disk group. Note that the file group name and the quota group name is unique in a disk group. The same name my be used in different disk groups. You can use this for easy identification. A PDB may have only one file group in a disk group. A quota group may contain multiple file groups. The file groups and quota group may be administered with commands in SQL*Plus or ASMCMD.

- A disk group contains at least one file group, the default file group.
- A disk group can contain multiple file groups.
- A disk group must have FLEX or EXTENDED redundancy to contain a file group.
- A disk group can store files belonging to multiple databases with each database having a separate file group.
- A database can have only one file group in a disk group.
- A database can span multiple disk groups with multiple file groups in different disk groups. The file groups belonging to a database in multiple disk groups should all have the same name for consistency and easy identification.
- A file group can belong to only one disk group.
- A file group can describe only one database, PDB, CDB, volume, or cluster.
- A file group can belong to only one quota group.
- Automatically created file groups are associated with the generic quota group.
- When a database, PDB, or CDB is created, if an existing file group has a client ID or name that matches the ID or name of the database, PDB, or CDB, then that file group is used to describe its files. Otherwise, a new file group is created for the database, PDB, or CDB.

ASM Tools



Oracle Enterprise Manager

Server Control (SRVCTL)

Oracle ASM command line utility `asmcmd`

Oracle ACFS command line utility `acfsutil`

Oracle ASM Configuration Assistant (ASMCA)

Oracle ACFS Cluster domain command line utility `acfsremote`

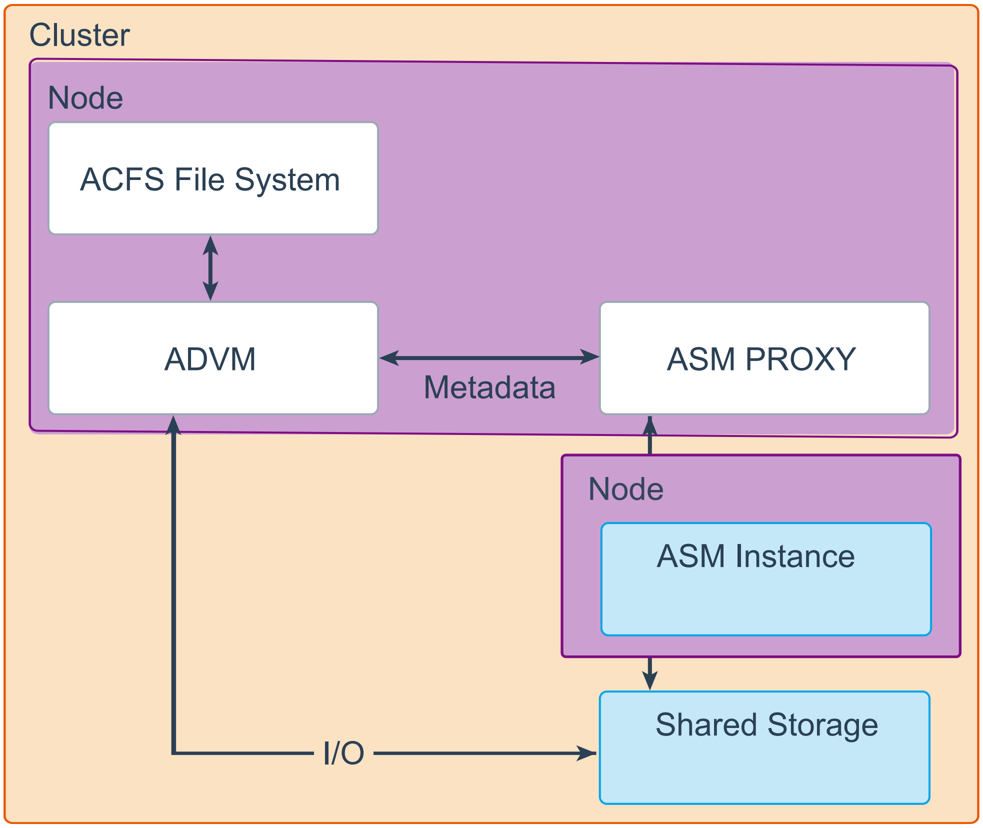
SQL*Plus

Because ASM is a variation of an Oracle Database, many of the same tools used to manage an Oracle database can also be used to manage an ASM instance.

The GUI tool ASMCA is a convenient utility designed specifically for ASM management.

You can also use command line tools: asmcmd, acfsutil, acfsremote that are specific to ASM and the ASM Cluster File System (ACFS).

ASM Cluster File System (ACFS)



Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is POSIX Compliant storage management technology, that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support all customer files.

Oracle ACFS presents a mountable file system to any cluster node that can access ASM. This is a general purpose file system. Oracle ACFS does not support any files associated with the management of Oracle ASM, such as files in the Oracle Grid Infrastructure home and in the Oracle ASM diagnostic directory.

Oracle ACFS supports Oracle Database files and application files, including executables, database data files, database trace files, database alert logs, application reports, BFILEs, and configuration files. Other supported files are video, audio, text, images, engineering drawings, and all other general-purpose application file data. Oracle ACFS conforms to POSIX standards for Linux and UNIX, and to Windows standards for Windows.

An Oracle ACFS file system communicates with Oracle ASM and is configured with Oracle ASM storage.

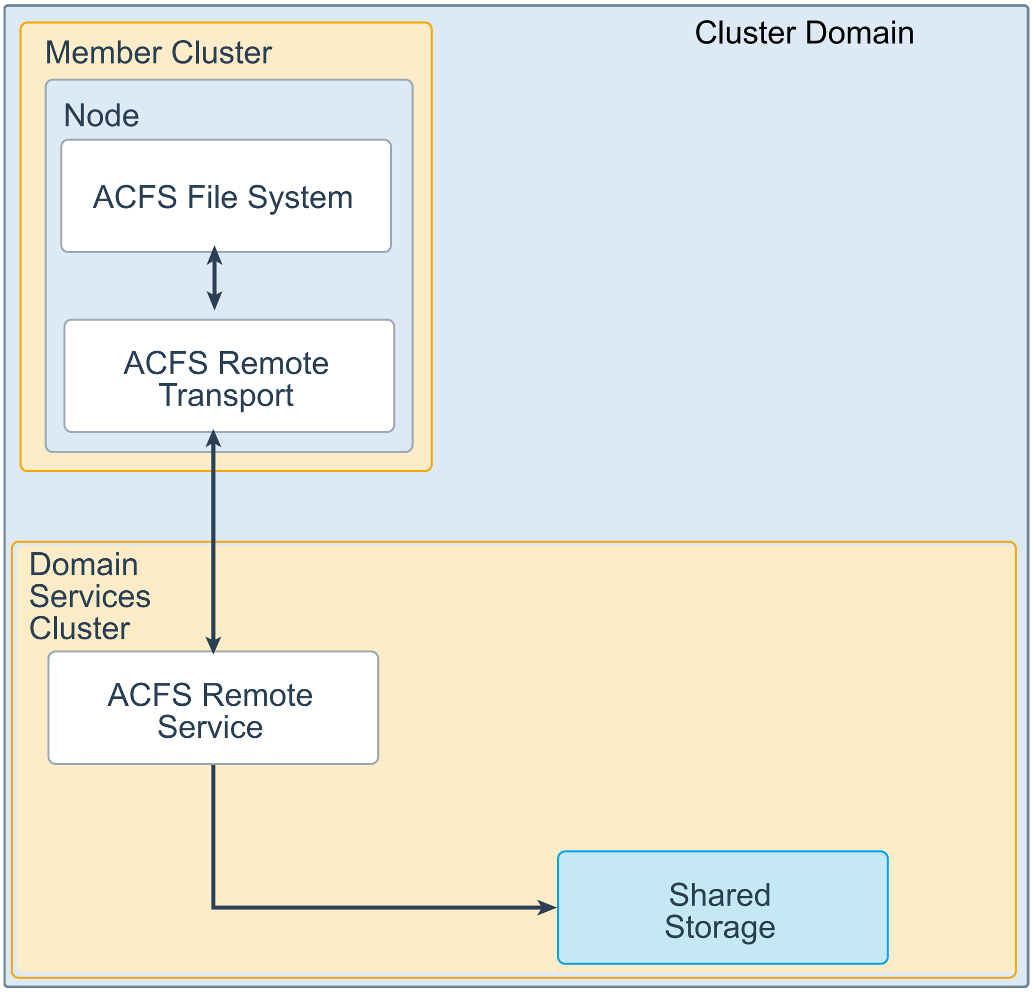
Oracle ADVM extends Oracle ASM by providing a disk driver interface to Oracle ASM storage allocated as Oracle ADVM volume files. You can use Oracle ADVM to create virtual disks that contain file systems. These file systems contained on Oracle ADVM volumes are able to support files beyond Oracle Database files, such as executable files, report files, trace files, alert logs, and other application data files. Because Oracle ADVM volumes are actually Oracle ASM files, they require the same administrative privileges as the Oracle ASM files.

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) communicates with Oracle ASM through the Oracle ADVM interface. With the addition of the Oracle ADVM, Oracle ASM becomes a complete storage solution of user data for both database and non-database file needs. IO requests to ACFS are passed through ADVM and the associated IO Drivers. The Metadata required to access the ASM disk groups is provided through the ASM Proxy, but the ADVM drivers access the ASM blocks directly.

Every ACFS instance requires the ADVM drivers. The ADVM drivers require an ASM Proxy instance on the same node. The ASM instance can be in the same node, another node in the same cluster or provided by the Domain Services Cluster. The ASM Proxy instance is a lightweight ASM instance, that provides the metadata for the AVDM drivers.

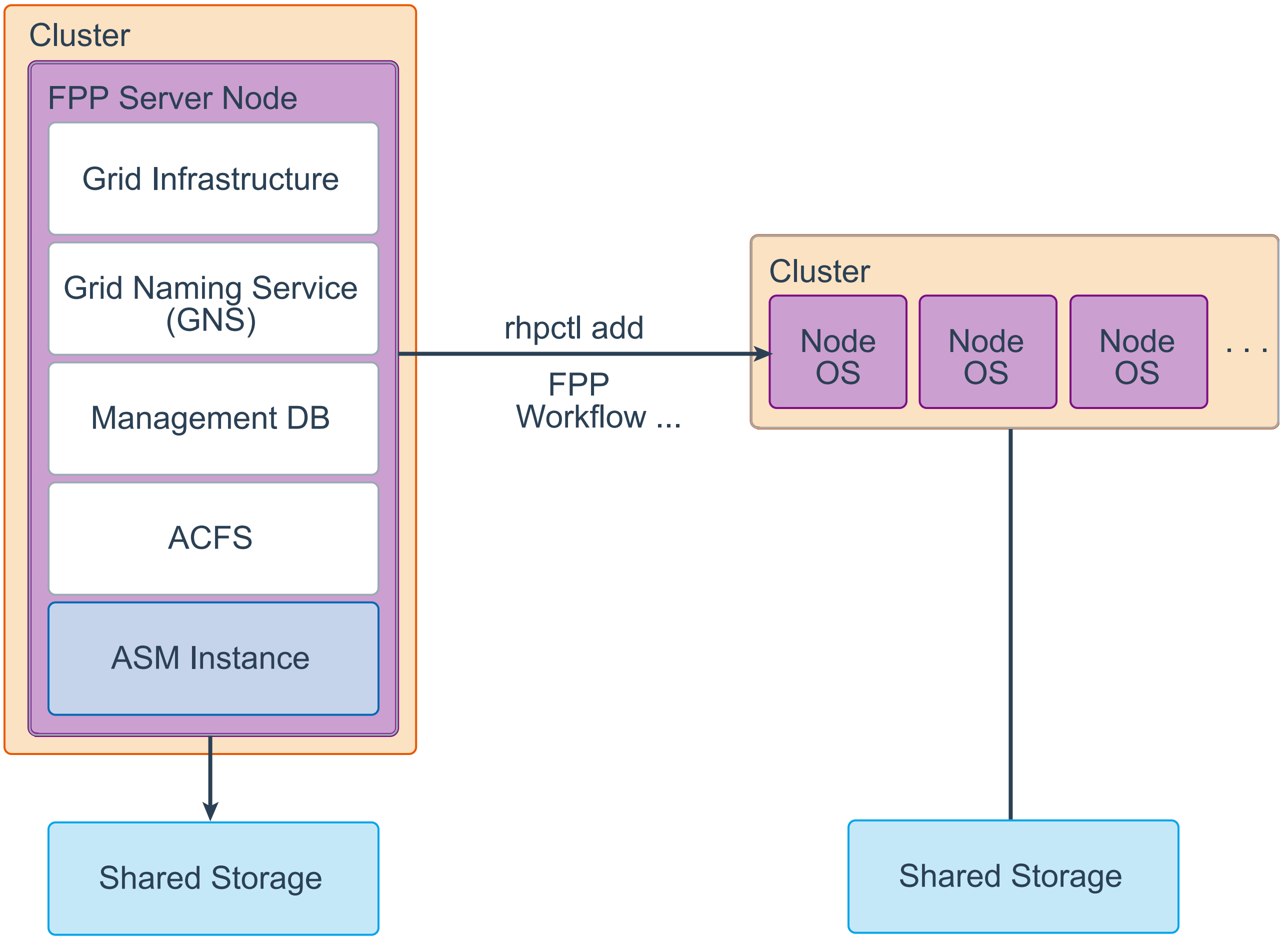
Note: Dynamic volumes supersede traditional device partitioning. Each volume is individually named and may be configured for a single file system. Oracle ADVM volumes may be created on demand from Oracle ASM disk group storage and dynamically resized as required. These attributes make Oracle ADVM volumes far more flexible than physical devices and associated partitioning schemes.

ASM Cluster File System (ACFS) Remote



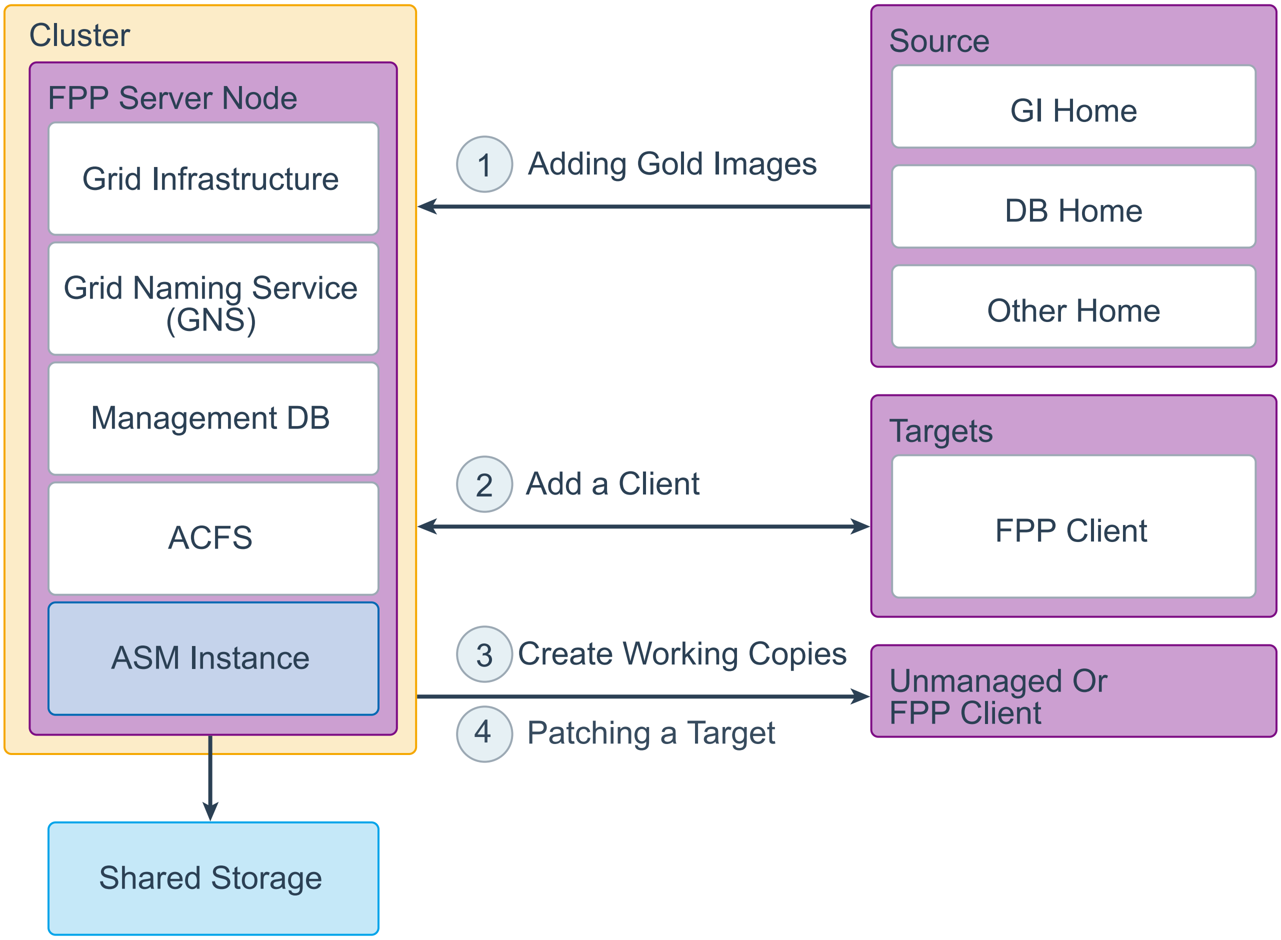
When you are using an Oracle ACFS deployment on the Oracle Domain Services Cluster (DSC), Oracle ACFS provides Oracle ACFS remote service for native Oracle ACFS functionality on Oracle Database member clusters with no attached local storage (InDirect Storage Member Cluster). Oracle ACFS remote service can be used for both Oracle application clusters and Oracle Database member clusters to enable a flexible and file system-based deployment of applications and databases.

Fleet Patching and Provisioning and Maintenance (FPP)



The Fleet Patching and Provisioning, formerly Rapid Home Provisioning (RHP), server has one or more nodes in a cluster configuration. It must have the Grid Infrastructure software installed. It is recommended that ASM configured with ACFS is used for the storage of image files (the RHP repository), but other file systems are supported. It must have Grid Naming Service (GNS), and a Management DB. These are the same components that make up the Domain Service Cluster in a Domain Cluster, but the node does not have to be in a Domain Cluster.

FPP Workflow



An Fleet Patching and Provisioning, formerly Rapid Home Provisioning (RHP), server must have Grid Infrastructure software installed. It is highly recommended that ASM with ASM Cluster File System (ACFS) is also installed to take advantage of the ability to snapshot and store deltas of gold images.

Unmanaged Homes are software homes e.g. Database Home that was not provisioned by RHP. Assume the RHP Server has already been provisioned.

1. Create a gold image from a Software Home that was previously installed with Oracle Universal Installer. Use the rhpctl import … command for the first gold image. You may run this command from the RHP server or RHP Client. You can create gold images from either a source home, or from a working copy using the rhpctl add image … command.
2. Add a client to the RHP server configuration with rhpctl add client … The client is required to have the OS installed, and cluster ready.
3. Create a working copy of the image with rhpctl add workingcopy … on the server, local file system, on the RHP client, or in a NFS mount point. This command creates an software home for the client machine on the client storage as specified in the command. Creating the home on the server requires that the client is using ACFS with access to the same storage cluster as the server. This operation can be on an unmanaged or managed target.
4. Patching a target follows a similar process.