

7 Cloud Operations

Cloud operations is the process of managing and optimizing cloud-based infrastructure, databases, and services. Features in this category enhance the availability, monitoring, maintenance, diagnosability, and security of cloud-based database deployments and reliably guarantees mission-critical high availability and resiliency.

Manageability

Hybrid Read-Only Mode for Pluggable Databases

Administrators can configure pluggable databases (PDBs) to operate in a new mode called hybrid read-only. Hybrid read-only mode enables a PDB to operate as either read-write or read-only, depending on the user who is connected to the PDB. For common users, the PDB will be in read-write mode. For local users, the PDB will be restricted to read-only mode.

Hybrid read-only mode enables you to patch and maintain an application in a safe mode for open PDBs without the risk of local users, including higher privileged ones, interfering with the ongoing maintenance operation of the PDB.

[View Documentation](#)

Real-Time SQL Monitoring Enhancements

Real-time SQL Monitoring works independently and concurrently across multiple PDB containers in an efficient manner. SQL statements, PL/SQL procedures and functions, and DBOPs (Database Operations) are monitored at PDB and CDB levels. You can efficiently query SQL Monitor reports across ad-hoc time ranges, DBIDs (internal database identifiers), and CON_DBIDs (CDB identifiers). This data is also accessible through SQL History Reporting.

Additionally, SQL Monitoring data can be exported along with the Automatic Workload Repository (AWR) and imported into another database or container for longer term storage and analysis.

Real-time SQL Monitoring is now supported per-PDB and CDB levels efficiently by default. As a PDBA persona, you can get a more accurate view of the monitored SQL for your application.

SQL Monitoring data can be transported through the AWR framework to a different container or database for longer term storage and offline analysis.

[View Documentation](#)

Control PDB Open Order

Administrators can define a startup order or priority for each pluggable database (PDB) where the most important PDBs are started first. The priority is applied to PDB opening order and upgrade order as follows:

- Restoring PDB states when opening the CDB
- Setting PDB states when using the `PDB OPEN ALL` statement
- Setting the order for PDB database upgrade operations
- Starting PDBs in an Active Data Guard (ADG) switchover or failover

This feature allows critical PDBs to start and open before less important PDBs, reducing the time for the critical applications to become usable.

[View Documentation](#)

Inter-Instance Resource Management

Inter-Instance Resource Management is a preemptive task scheduling and resource management capability that enables fine-grained control over CPU resources across multiple Container Databases, Pluggable Databases, and non-Database processes residing within servers, hosts, or Virtual Machines, in clustered and non-clustered environments. Inter-Instance Resource Management includes upper limits on CPU resource consumption, as well as lower-bound guarantees to enable bursting for multiple Pluggable Databases within a Container and multiple Container Databases on a server, host, or Virtual Machine.

This feature enables effective use of system resources with high-density database consolidation.

[View Documentation](#)

Optimized Performance for Parallel File System Operations

In environments that contain many PDBs and require multiple `DBMS_FS` requests to be processed in parallel, you can update the number `OFS_THREADS` to increase the number of `DBMS_FS` requests executed in parallel. This increases the number of worker threads

executing the make, mount, unmount, and destroy operations on Oracle file systems in the Oracle database. This will reduce the time needed to execute parallel file system requests in environments with multiple PDBs.

This feature significantly reduces the time required to perform parallel file system requests in consolidation environments containing multiple PDBs.

[View Documentation](#)

Read-Only Users and Sessions

You can control whether a user or session is enabled for read-write operations, irrespective of the privileges of the user that is connected to the database. The `READ_ONLY` session applies to any type of user for any type of container. The `READ_ONLY` user only applies to local users.

Providing the capability to disable and re-enable the read-write capabilities of any user or session without revoking and re-granting privileges provides you with more flexibility to temporarily control the privileges of users or sessions for testing, administration, or application development purposes. It also gives you a simple way to control the read-write behavior within different parts of an application that are used by the same user or session.

[View Documentation](#)

Continuous Availability

Application Continuity Session State Restore with Database Templates

Application Continuity uses database templates to checkpoint the session state, restore the session state at the start of replay, and support session migration during planned maintenance. Database templates restore server-side and client-visible session states at the beginning of the Application Continuity replay, thus increasing Application Continuity protection.

Application Continuity with Database Templates broadens and simplifies the use of Application Continuity to reduce planned maintenance-related downtime. It also enables the migration of more sessions faster during unplanned outages, ensuring higher levels of protection.

[View Documentation](#)

Enhanced Upgrade of Time Zone Data

The process of upgrading timezone data to reflect up-to-date Governmental Daylight Saving Time rules is optimized, taking the actual data content of tables into account. Only tables impacted by a Daylight Saving Time rule change will undergo a data change.

Optimizing the necessary data changes for a Daylight Savings Time rule change significantly improves the overall upgrade of timezone data to the absolute bare minimum to bring a database up to the latest global timezone rules. The implicit analysis and reduction of the data required to change significantly reduces the overall timezone upgrade process and the resources needed.

[View Documentation](#)

Optimized Read-Write Operations for Database Processes

To optimize the read and write operations performed by database processes when you access files managed through OFS or DBFS, specify the new `db_access` mount option for the `dbms_fs.mount_oracle_fs` procedure while mounting the file system.

When you enable `db_access`, both memory consumption and CPU usage reduces. The throughput increases while performing read and write operations by database processes on the files managed by OFS.

[View Documentation](#)

Smart Connection Rebalance

Smart Connection Rebalance transparently reshuffles service-based connections based on real-time performance monitoring across Oracle Real Application Clusters (Oracle RAC) instances.

Smart Connection Rebalance improves database performance by automatically moving sessions across Oracle RAC database instances without needing manual intervention.

[View Documentation](#)

Smooth Reconfiguration of Oracle RAC Instances

The smooth reconfiguration feature reduces the brownout time caused by certain Oracle Real Application Clusters (Oracle RAC) operations, such as nodes joining or leaving an Oracle RAC cluster.

Smooth Reconfiguration of Oracle RAC Instances ensures continuous availability of Oracle RAC services and reduces brownout time for database instances running in an Oracle RAC database.

[View Documentation](#)

Support for the Coexistence of DGPDB and GoldenGate Capture

This project introduces perPDB DataGuard. When DGPDB is configured on a source/Primary database, there are validations that insure there is no GoldenGate Capture pre-existing on the source. GoldenGate capture sessions will be broken if a DGPDB is allowed and executes a role transition.

This project adds support for coexistence of DGPDB and GoldenGate Capture. Changes/support will be required in the LogMiner, redo transport, and Broker layers.

[View Documentation](#)

General

Adaptive Result Cache Object Exclusion

With adaptive result cache object exclusion, the database decides to blocklist certain objects if using the result cache is not beneficial for these objects, based on statistical evidence such as the number of invalidations, the cost savings of using result caching, and others. You have full control over the objects considered for exclusion to ensure you can continue using result cache for all your objects of interest.

Adaptive exclusion of objects that don't benefit or even have a detriment impact on result caching reduces the overall development and management workload for you. It can improve the database performance out of the box.

[View Documentation](#)

Diagnose and Repair SQL Exceptions Automatically at Compile-Time

Automatic error mitigation automatically detects and repairs many severe compile-time SQL exceptions that would otherwise cause SQL statements to fail. If a severe internal error occurs during SQL compilation, the exception is trapped, and the foreground processes will re-parse the statement to try and work around the problem. If re-parse is successful, a SQL patch is used to persist the workaround for future hard parses, and the session parsing the SQL statement will proceed without interruption and without receiving an exception.

This feature improves the robustness of your applications and its service levels.

[View Documentation](#)

Read-Only Tablespace on Object Storage

Read-only tablespaces can be moved to and from Oracle Object Storage transparently, storing portions of a database on lower-cost storage in the Cloud.

Allowing to move tablespaces to Oracle Object Storage enables a data lifecycle management strategy, storing the data on the most cost-effective storage tier based on its business value or access frequency.

[View Documentation](#)

Unified Memory

Unified Memory is a flexible and simple memory configuration for Oracle Databases that uses a single parameter to control database memory allocations, reducing or eliminating the need for system restart to change memory configurations. Unified Memory is especially useful in multiple workload high density database consolidation environments.

Unified Memory simplifies memory management to run multiple workloads in a highly consolidated environment with minimum disruption. It is easier to set the single parameter `MEMORY_SIZE` for configuring the database instance memory instead of using separate parameters like `SGA_TARGET` and `PGA_AGGREGATE_LIMIT`.

[View Documentation](#)