Planned Database Maintenance of Oracle Autonomous Database on Shared Exadata Infrastructure with JD Edwards EnterpriseOne

Performance Analysis of JD Edwards EnterpriseOne with Oracle Autonomous Database on Shared Exadata Infrastructure

October, 2021 | Version 1.0

Purpose Statement

This document outlines the findings of the Oracle JD Edwards performance and benchmarking team with JD Edwards EnterpriseOne on Shared Exadata Infrastructure (ADB-S) configuration.   The purpose of the study is to:

- Identify gaps within EnterpriseOne to support ADB-S and to provide remedies
- Provide a clear picture of the current EnterpriseOne behaviour with ADB-S backed by testing and metrics
- Provide a level of assurance for customers adopting EnterpriseOne with ADB-S

Disclaimer

This document, in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of Contents

List of images

List of tables

Executive Summary

Customers adopting Oracle Autonomous Database (ADB) on Shared Exadata Infrastructure (ADB-S) can benefit from its self-driving, self-patching and self-repairing capabilities. Customers can experience zero downtime of their JD Edwards with ADB-S during maintenance activities, giving them greater business continuity and no loss of revenue.

ADB-S will help in saving several man-hours of effort that otherwise would have been spent on database maintenance activities. It also enables continuous availability of Oracle JD Edwards and protects the data integrity of the EnterpriseOne application.

This document demonstrates that there is no impact to the JD Edwards EnterpriseOne interactive and batch applications in a planned ADB-S maintenance window assuming the processes outlined herein are followed. Furthermore, the adoption of Oracle connection pooling and the configuration of the Oracle database will ensure a nearly zero downtime experience. Implementing all of these brings assurance to customers who wish to adopt the ADB-S architecture as a solution to their business needs.

Overview

The Oracle Autonomous Database is self-driving, self-securing and self-repairing. The complete automation of database maintenance activities including patching provides business continuity and saves significant cost and time for JD Edwards customers.

The Oracle Autonomous Database comes in two deployment options: Dedicated (ADB-D) and Shared (ADB-S). Oracle Autonomous Database on Dedicated Exadata Infrastructure allows customers to provision Autonomous Databases into their own dedicated Exadata cloud infrastructure, instead of into a shared infrastructure with other tenants. With this service, customers can exclusively use the Exadata hardware. This service offers multitenant database architecture, enabling customers to create and manage multiple Autonomous Databases within a single database system. Oracle Autonomous Database dedicated is also available on Exadata Cloud@Customer for customers who want to deploy Autonomous Database in their data center.

Oracle Autonomous Database on Shared Exdata Infrastructure allows customers to provision Autonomous Databases on a shared infrastructure with other tenants. With this service, customers provision and manage only the Autonomous Database, while Oracle deploys and manages the Exadata infrastructure. JD Edwards customers can choose between the two deployment options based on their business requirements and the desired level of control over their database lifecycle management.

JD Edwards customers are wary of taking the latest database patches because they need to have a downtime on their EnterpriseOne systems that directly affects their business continuity. This practice makes their environment more vulnerable and prone to failures. Oracle Autonomous Database on Shared Exadata Infrastructure overcomes this limitation by providing zero downtime due to patching of the database nodes.

The ADB-S patching occurs every week in a rolling RAC fashion to keep the database up to date without any impact to a running application. The fact that the patching happens without any downtime is valuable to JD Edwards customers, as it does not affect their business continuity.

Testing Use Cases

The main objective of the testing to identify gaps in the EnterpriseOne processes during a planned maintenance event of the Oracle Autonomous Database on Shared Exadata Infrastructure that would impact business continuity for customers. The test process included batch and interactive testing as outlined in Table 1.

USE CASE	TESTCASE TYPE	DESCRIPTION
1	Batch	Submission of a set of nine short running UBEs • Submission initiated using the runube utility

		• Approximately 300 batch jobs are executed for each test
2	Interactive	Sales Order Entry (P42101) application • JMeter utility used to generate load • Load consisted of 10 user running the application entering a 10 line item sales order for a total of 100 sales orders for each test

Table 1. Use Case Scenarios for Performance Tuning

Table 1 refers the two test case categories – batch and interactive. The batch test case is the submission of a set of nine short running UBEs. Interactive testing involves ten simultaneous EnterpriseOne users exercising the Sales Order manufacturing application.

Test results performed with these use cases are be covered in a subsequent section of this document.

Extended testing for this performance analysis includes the use of long running batch jobs as well as additional interactive applications for a broader EnterpriseOne application coverage for the comparison. Details listing the extended use cases are specified in Appendix B.

Metrics

Metrics provides an indication whether the application is performing within the range of expected parameters or whether the performance is within acceptable standards. For testing EnterpriseOne in an Oracle ADB-S environment the most important metric for success is the ability of the EnterpriseOne processes to maintain service through the Oracle maintenance window events.

Testing Configuration

Customers desiring more business continuity in their experience with EnterpriseOne during Oracle planned scheduled maintenance events should comply with the following recommended and minimum configurations. These minimum configurations are outlined in more detail in the appendix and they include:

- EnterpriseOne 9.2 application and Tools Release 9.2.6.0
- Oracle Database Server 19c, version 19.11 or greater
- EnterpriseOne Server Oracle Client software 19.11 or greater
- EnterpriseOne architecture configured with the new Oracle connection pooling instead of the legacy JD Edwards connection pooling
- Oracle Server configured with Transparent Application Continuity (TAC). See Appendix A for enabling/disabling the TAC configuration.

This configuration formed the basis of the results in this document. It provides the baseline for customers making the decision to adopt Oracle ADB-S as the solution for their business needs.

Starting with EnterpriseOne Tools Release 9.2.6.0, automatic kernel reconnect feature is introduced to maintain Oracle database and session connection throughout the lifecycle of the the EnterpriseOne processes, even in the event of a Planned Oracle Autonomous Database failover event. Understanding refreshing the kernel is covered extensively in the EnterpriseOne 9.2.6.0 Server Manager Guide, it is also covered at some depth in Appendix A.

EnterpriseOne and Planned Oracle Autonomous Database Failover Events

The EnterpriseOne kernel processes that are affected by the test use cases are described in Table 2. This table provides a high level overview of the specific kernel processes for the testing of ADB-S in an Oracle database planned failover event with EnterpriseOne.

KERNEL DEFINITION	KERNEL NAME	DESCRIPTION
[JDENET_KERNEL_DEF2]	UBE Kernel	Controls the submission of batch process requests from clients and schedules batch jobs in the job queue.
[JDENET_KERNEL_DEF4]	Security Kernel	Provides the credentials and application tokens for security access to the EnterpriseOne application. Additionally it also handles all user profile and security related table updates.
[JDENET_KERNEL_DEF6]	Call Object Kernel (COK)	Responsible for executing the business function logic through requests received from JDENET. If the user has already established a request to the COK process, JDENET will route subsequent requests to the same COK process. The only time a business function will run on a different COK process is when the BSFN is OCM mapped to a different server. Only COK kernels support multi-thread processes. All other kernels handle data and cache one at a time.
[JDENET_KERNEL_DEF30]	Metadata Kernel	The metadata kernel runs in a JVM and processes the XML specification access requests readable by the call object kernel.

Table 2. EnterpriseOne Kernel Process Definitions

There are four kernel processes that are involved with the EnterpriseOne interactive and batch testing performed for this testing. Each kernel process has a specific function for use in the EnterpriseOne application. It is important to understand the function of each kernel process as to have a better understanding of the observed affects of an Oracle failover event and its impact on providing continuous service to customers

Table 3 below provides a summary of the planned failover behaviour design during an Oracle planned maintenance event on ADB-S.

KERNEL DEFINITION	KERNEL NAME	STATIC CONNECTION	FAILOVER BEHAVIOR
[JDENET_KERNEL_DEF2]	UBE Kernel	Yes	Current UBE kernel process will connect to the secondary node upon failover.
[JDENET_KERNEL_DEF4]	Security Kernel	Yes	Current Security kernel process will connect to the secondary node upon failover.
[JDENET_KERNEL_DEF6]	Call Object Kernel (COK)	Yes	A NEW Call Object kernel process will be created on failover with connections to the secondary node. If an Oracle session is present on failover to the primary node, this process will remain and will have to be removed once all sessions are terminated.
[JDENET_KERNEL_DEF30]	Metadata Kernel	Yes	Current Security kernel process will connect to the secondary node upon failover

Table 3. EnterpriseOne Kernel Processes

All kernel processes listed in Table 3 use static pooled connections to the database. A database connection is a physical process connection between the EnterpriseOne server and a database instance. A connection pool is a cache of connections to an Oracle database that are used by the EnterpriseOne application.

For the EnterpriseOne application, all connections to the database will remain static for each kernel process as long as there are EnterpriseOne user processes active in that kernel process. Conversely, a dynamic pooled connection would create and destroy connections in the kernel process. For this discussion, the EnterpriseOne design is static and not dynamic. This means that if a kernel process in EnterpriseOne is connected to a primary node in a 2-node Oracle RAC configuration, it will remain on the primary node for the life of the kernel process.

The EnterpriseOne application kernel processes requires a 'refresh' event to trigger and force the static connections on the primary node to move to a secondary node. In an Oracle planned failover event, this 'refresh' can be triggered manually through the Server Manager Console, see Appendix A for more detail.

It is recommended that customers suspend their interactive and batch processing until the completion of the scheduled Oracle planned maintenance events for JD Edwards EnterpriseOne kernel processes to reconnect to the new Oracle database nodes.

Oracle Planned Maintenance Event Results

Image 1 below is an Oracle session profile of the EnterpriseOne sessions established over a period of time.   The testing involved a 2-node Oracle RAC configuration that is being patched through a scheduled maintenance window. In a two node RAC configuration there are two maintenance events, one for each Oracle RAC node as shown in Image 1.

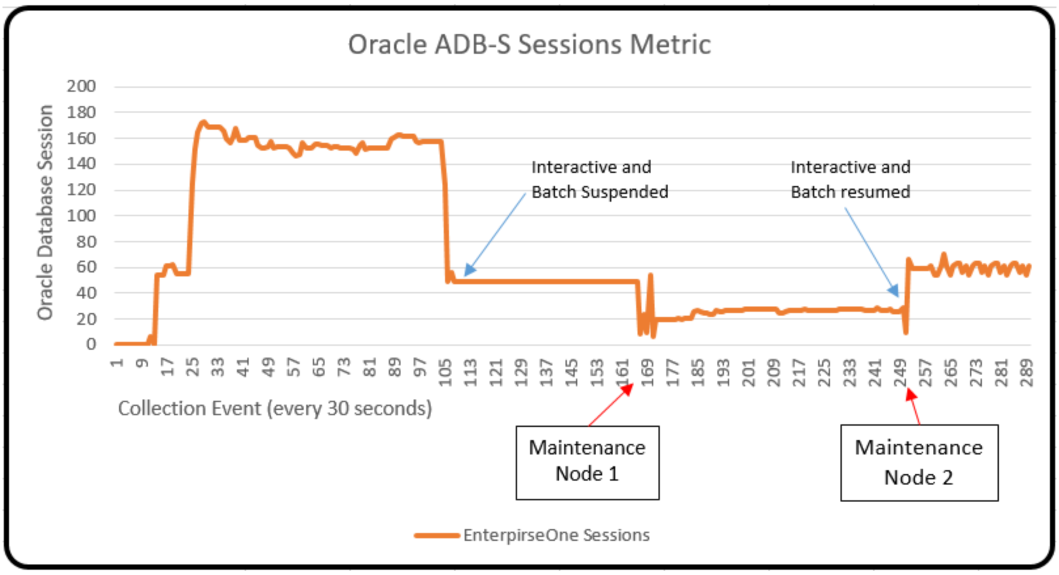


Image 1. Oracle ADB-S Database Connections

A database session is the logical structure in the database instance memory that represents the state of a current user accessing the database. A session lasts from the time the user is authenticated by the security kernel process in EnterpriseOne by the database until the time the user disconnects or exits the EnterpriseOne application. The database connection is also terminated if the EnterpriseOne session timeout expires. Database sessions established by the EnterpriseOne application can be viewed through the Server Manager Console or by querying the active sessions on the database server directly (see appendix).

After the user database session is authenticated by the security kernel, the other kernel processes listed in Table 3 may be involved in the processing of the EnterpriseOne interactive user application and batch requests.

The flow and explanation of the test performed with EnterpriseOne on an Oracle two node RAC configuration is shown in Table 4 below. The flow follows a timeline every 30 seconds where the number of Oracle sessions are measured, thus providing the baseline activity of the database during this test.

TIMELINE	EVENT	OBSERVATIONS
0	EnterpriseOne services started	EnterpriseOne Server and HTML Server services are started. The Oracle 2-node RAC configuration is fully active in an active-active configuration.
25 (12.5 minutes)	Interactive and batch processes started	Interactive user and batch loads are initiated. The number of Oracle session peaked at 170 sessions, but came to a steady state usage around 155 sessions.
105 (52 minutes)	Interactive and batch processes suspended	EnterpriseOne activity is suspended. Batch processes are either complete or held in queue. (Note1) Interactive users are logged out of the application and there are no active sessions registered in the Server Manager Console.

169 and 249 (85 and 125 minutes)	Oracle planned maintenance occurs	During this approximate two hour window (maximum for an ADB-S planned patching window) the Oracle database is patched. After each maintenance event, the EnterpriseOne kernel processes are manually 'refreshed'. (Note 2)
249 (125 minutes)	Interactive and batch processes resume	Interactive users are allowed to continue processing and batch processes are released from their hold.

Table 4. Oracle Session Profile during Planned Maintenance Events

Note1: Holding batch jobs in queue is discussed in the 9.2 JD Edwards EnterpriseOne Tools Administration Guide (E53541-09)

Note2: The manually refreshing of EnterpriseOne kernel processes is covered in Appendix A and the Server Manager Guide for Tools Release 9.2.6.0

Image 1 and Table 4 represent an experience for EnterpriseOne customers during this planned maintenance event. In this two hour span of time, both nodes in the Oracle database are moved to new Oracle patched nodes where they will remain to support the EnterpriseOne application. For example, if the EnterpriseOne application are connecting to nodes 1 and 2 in a 2-node Oracle RAC configuration, because they were moved to facilitate path, then after the planned maintenance they would be connected to nodes 3 and 4, assuming Oracle RAC nodes 3 and 4 are already patched and available. The timeframe of this movement of Oracle connections and sessions to the new nodes takes only a few minutes to complete. Image 1 shows that Oracle sessions are quickly moved from one node to another to be able to provide service to the EnterpriseOne application as seen with a steady state consumption of sessions where approximately 30 sessions are consumed by both Oracle RAC nodes by the processing of EnterpriseOne application kernels.

In EnterpriseOne Tools Release 9.2.6.0, the 'refresh' action is performed on the Server Manager Console (see appendix). The current implementation of EnterpriseOne with ADB-S requires some level of manual intervention during the two hour planned maintenance event to provide business continuity. This process and experience will be improved and automated in subsequent levels of support planned in the future.

Conclusion

Business continuity is having a plan to deal with situations such as with an Oracle planned maintenance event on the Oracle database, so the customer using the EnterpriseOne application can continue to function with minimal disruptions.

With Tools Release 9.2.6.0, the EnterpriseOne application architecture has improved the automation of these processes to limit the requirement for manual intervention to a two hour window for planned Oracle maintenance, which is required to keep customers at the current release of the database in an Oracle ADB-S configuration.

JD Edwards customers no longer need to recycle the EnterpriseOne services for Oracle database maintenance. They only need to 'refresh' the environment from the Server Manager Console.

This initial support for Oracle Autonomous Database on Shared Exadata Infrastructure begins with a few manual steps and achieving planned database maintenance with zero downtime. Oracle JD Edwards plans to improve the adoption process and support additional capabilities of the Oracle Autonomous Database on Shared Exadata Infrastructure in subsequent levels of support planned in the future.

Appendix A – Oracle Autonomous Database Configuration

The Oracle database features for continuous availability and configuration of Transparent Application Continuity (TAC) for the Oracle RAC configuration are described by following the link below:

https://docs.oracle.com/en/cloud/paas/autonomous-database/atpdg/code-continuous-availability.html#GUID-1F36B307-63C4-49CC-BAFE-52077C5D92AB

Oracle FAN and TAC Configuration

The EnterpriseOne software achieves continuous availability when planned maintenance, unplanned outages, and load imbalances of the database can be hidden (shielded) from the application. A combination of application best practice, simple configuration, and Oracle Autonomous Database ensure that your applications are continuously available.

The recommended best approach for hiding planned maintenance activities from your applications is to transparently drain work from each database workload location prior to the maintenance window for that workload location. The Oracle database connection pools and mid-tiers, including the WebLogic Server, Oracle Universal Connection Pool (UCP), OCI Session pool and ODP.NET Unmanaged Provider are Fast Application Notification (FAN) aware and therefore are notified before database services are scheduled to move to allow graceful draining of work before maintenance. FAN notification automatically triggers closing idle connections, opening new connections in the new service location, and allows a configurable time for active work to complete in the soon-to-be-shutdown service location. The major third-party JDBC mid-tiers, such as IBM WebSphere, allow for the same behaviour when configured with UCP. For JDBC-based applications that cannot use UCP, Oracle provides solutions using Oracle Drivers and connection tests.

In order to shield unplanned outages resulting from a component or communication failure Oracle provides:

- **Notification**. FAN is the first step to hiding outages. FAN notifies clients and breaks them out of their current network wait when an outage occurs. This avoids stalling applications for long network waits. Importantly, FAN also invokes rebalancing of sessions when services are available again.
- **Recovery**. After the client is notified, Transparent Application Continuity (TAC) or Application Continuity (AC) re-establish a connection to a new workload location (another database instance in the Real Application Clusters (RAC) configuration running the database) and replays in-flight (uncommitted) work when possible. By replaying in-flight work on the new location, the application can usually continue executing without knowing that any failure happened.

TAC or AC also executes during planned maintenance for those sessions that do not drain (complete their current database operation) during the allocated drain interval.

Enabling/Disabling TAC

Enabling and Disabling of Transparent Application Continuity (TAC) occurs by changing the status of the failover type variable in the dba_services configuration of the Oracle database. This command has to be run with administrator priviledges and on a 2-node Oracle RAC configuration only needs to be executed on one of the nodes to be available on both nodes of the Oracle RAC instance.

The following commands should be initiated as the admin user

```
SQL> set linesize 125;

SQL> column name format a60;

SQL> column failover_type format a20;

SQL> select name, FAILOVER_TYPE, REPLAY_INITIATION_TIMEOUT from dba_services;

NAME                                                         FAILOVER_TYPE        REPLAY_INITIATION_TIMEOUT

------------------------------------------------------------ -------------------- -------------------------

XFXRPZ0TO9LP3WU_IDCATPS_tp.adb.oraclecloud.com               AUTO                                       300
```

Note: A value of AUTO means TAC is enabled, a value of NONE is disabled…for Oracle ADB-S TAC should be enabled by default

To ENABLE TAC

Assume that the Oracle SID is xfxrpz0to9lp3wu_idcatps_tp.adb.oraclecloud.com, you can enable TAC with the following Oracle process:

```
SQL> execute DBMS_APP_CONT_ADMIN.ENABLE_TAC('xfxrpz0to9lp3wu_idcatps_tp.adb.oraclecloud.com', 'AUTO',
1200);
```

To DISABLE TAC

Use the following PL/SQL Oracle process

```
SQL> execute DBMS_APP_CONT_ADMIN.DISABLE_FAILOVER('xfxrpz0to9lp3wu_idcatps_tp.adb.oraclecloud.com');
```

Verify the status of TAC to validate its current state

After any change to the configuration of the database, the configuration of the

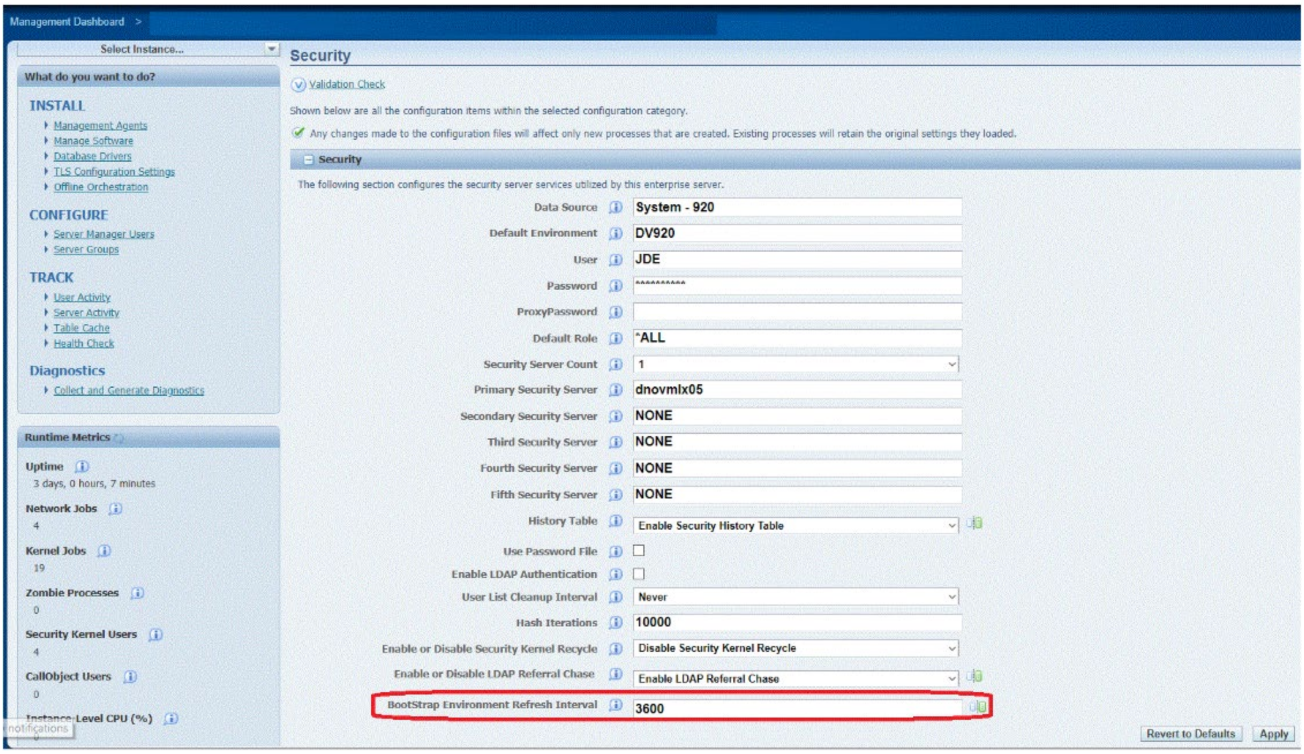
```
SQL> set linesize 125;
SQL> column name format a60;
SQL> column failover_type format a20;
SQL> select name, FAILOVER_TYPE, REPLAY_INITIATION_TIMEOUT from dba_services;
NAME                                                         FAILOVER_TYPE        REPLAY_INITIATION_TIMEOUT
------------------------------------------------------------ -------------------- -------------------------
XFXRPZ0TO9LP3WU_IDCATPS_tp.adb.oraclecloud.com               NONE                                       300
```

Automatic Kernel Refresh

The automatic kernel refresh and on-demand manual kernel refresh features were introduced with Tools Release 9.2.6.0 and the documentation is available in the 9.2 Server Manager Guide. It also will be covered here in a lesser detail.

Automatic Kernel Refresh

In order to avoid planned and unplanned Oracle database connection disruptions, the JD Edwards logic allows an idle kernel process to re-establish connections to the Oracle database. An automatic kernel refresh involves a kernel process logout and reconnection of the bootstrap session with the Oracle database, ensuring connection integrity. The bootstrap security login is controlled on the EnterpriseOne server and is shown in Image 2.



Uptime 3 days, 0 hours, 7 minutes
Network Jobs 4
Kernel Jobs 19
Zombie Processes 0
Security Kernel Users 4
CallObject Users 0
Instance Level CPU (%)

Security

Shown below are all the configuration items within the selected configuration category.

Any changes made to the configuration files will affect only new processes that are created. Existing processes will retain the original settings they loaded.

Security

The following section configures the security server services utilized by this enterprise server.

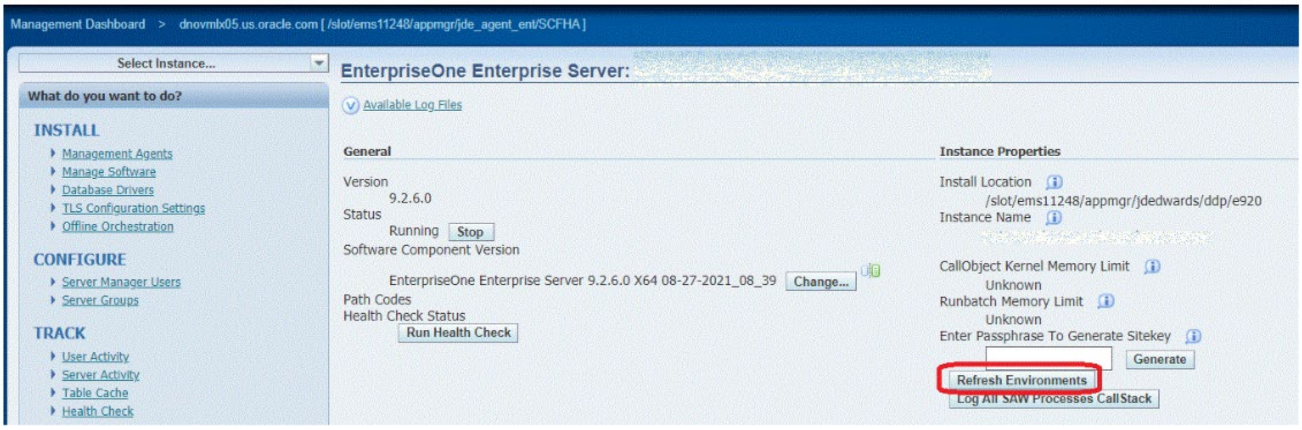
Data Source	System - 920
Default Environment	DV920
User	JDE
Password	**********
ProxyPassword	
Default Role	*ALL
Security Server Count	1
Primary Security Server	dnovmlx05
Secondary Security Server	NONE
Third Security Server	NONE
Fourth Security Server	NONE
Fifth Security Server	NONE
History Table	Enable Security History Table
Use Password File	□
Enable LDAP Authentication	□
User List Cleanup Interval	Never
Hash Iterations	10000
Enable or Disable Security Kernel Recycle	Disable Security Kernel Recycle
Enable or Disable LDAP Referral Chase	Enable LDAP Referral Chase
BootStrap Environment Refresh Interval	3600

Image 2. EnterpriseOne Bootstrap Refresh Interval

Image 2 shows the bootstrap value of 3600 (60 minutes), and is found in the Advanced Configuration setting under Security. The default value for automatic kernel refresh is 0 (disabled).

On-Demand Kernel Refresh

EnterpriseOne Tools Release 9.2.6.0 also provides a mechanism to force the refreshing of the Oracle sessions and connection of the database by the Enterprise kernel processes. The Refresh Environment button on the Enterprise Server dashboard in the Server Manager Console is shown in Image 3.



EnterpriseOne Enterprise Server:

General

Version
9.2.6.0
Status
Running
Software Component Version
EnterpriseOne Enterprise Server 9.2.6.0 X64 08-27-2021_08_39
Path Codes
Health Check Status

Instance Properties

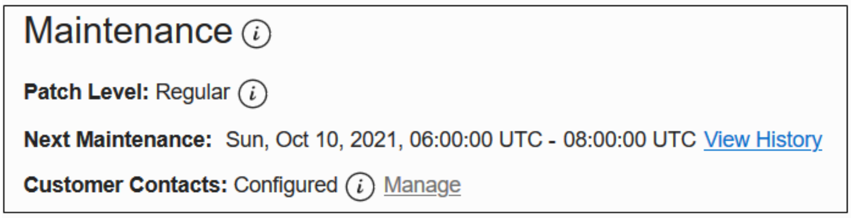
Install Location
/slot/ems11248/appmgr/jdedwards/ddp/e920
Instance Name
CallObject Kernel Memory Limit
Unknown
Runbatch Memory Limit
Unknown
Enter Passphrase To Generate Sitekey

Image 3. EnterpriseOne On-Demand Kernel Refresh

Image 3 shows the Refresh Environment button to be used to force the EnterpriseOne kernels to re-establish connections to the database.

Detecting Planned Maintenance Events

On the Oracle Cloud management dashboard for Autonomous Database is a tab for Autonomous Database Information. Within this tab the planned maintenance events will be listed as shown in Image 4.



Maintenance ⓘ

Patch Level: Regular ⓘ

Next Maintenance: Sun, Oct 10, 2021, 06:00:00 UTC - 08:00:00 UTC View History

Customer Contacts: Configured ⓘ Manage

Image 4. EnterpriseOne Database Connections

Image 4 shows a planned maintenance event is to occur Sun Oct 10, 2021 between 6am and 8am UTC time. The image also shows that the Oracle Cloud dashboard allows customers to view previous history of planned Oracle database events for this Autonomous Database.

This document concentrates on all of the EnterpriseOne kernel processes reconnecting to the Oracle Autonomous Database, an on-demand refresh testing methodology was employed. The challenge to knowing when to click the Refresh Environment button is discussed here.

Autonomous Database Notifications

The Oracle Automated Database provides a notification in the Cloud Platform Autonomous Database Using Oracle Autonomous Databawe on Shared Exadata Infrasturre Guide under the ‘About Events Based Notifications and Automation on Autonomous Database’ section.

The notifications will provide a customer with information on the planned Oracle Autonomous Database events such as:

- A 24 hour notification of maintenance window
- A 1-hour notification of the window
- The time the maintenance has started
- The time the maintenance has ended

A customer can rely on these timings to suspend EnterpriseOne application and batch processes until the planned maintenance window completes.

Detecting Autonomous Database Failover through SQL

A secondary method for detecting an Autonomous Database Failover is by the following SQL script. The advantage to this script is it can detect a planned failover before the notification is received and thus can shorten the actual downtime of the Oracle Autonomous Database event downtime, increasing the business continuity of the EnterpriseOne application.

```
set echo on;
column host_name format a15;
column username format a15;
column service_name format a15;
column machine format a20;
column process format a10;
column program format a35;
-- spool testing.out append;
set linesize 120;
set pagesize 999;
set echo on;
-- select localtimestamp from dual;
select count(s.machine),s.process,s.machine,i.host_name,s.program,s.inst_id from gv$session s join gv$instance i on
(i.inst_id=s.inst_id) where username is not null group by process,machine,host_name,program,s.inst_id order by
host_name,process;
```

OUTPUT:

```
COUNT(S.MACHINE) PROCESS    MACHINE              HOST_NAME       PROGRAM                                INST_ID
---------------- ---------- -------------------- --------------- ----------------------------------- ----------
               1 2080707    idcentrpris                          sqlplus@idcentrpris (TNS V1-V3)              4
               1 352482     idcentrpris                          oracle@e42pod-8qau24 (PPA7)                  6
               1 4049049    idcentrpris                          jdenet_k@idcentrpris (TNS V1-V3)             4
               2 4049049    idcentrpris                          jdenet_k@idcentrpris (TNS V1-V3)             6
               1 4049050    idcentrpris                          jdenet_k@idcentrpris (TNS V1-V3)             4
               2 4049050    idcentrpris                          jdenet_k@idcentrpris (TNS V1-V3)             6
               1 4049060    idcentrpris                          jdenet_k@idcentrpris (TNS V1-V3)             4
```

In this output of the SQL command shown above, a customer can detect more granularly the nodes on which the EnterpriseOne services are running. When a planned maintenance event occurs, a new INST_ID will appear in this output and it is then a customer can press the Refresh Environment button.

Appendix B - Test Cases

The test use cases in this appendix are for information only. The purpose of this document is to show that if a customer suspends the EnterpriseOne activity, then only a ‘refresh’ of the EnterpriseOne kernel processes is necessary for business continuity.

S.NO	TESTCASE TYPE	UNIT TESTING	LOAD TESTING
1	Batch	Submission of long running UBEs - R31410, R3483, R42565 and R43500	Submission of 9 short running UBEs and long running UBEs mentioned in the unit testing
2	Interactive	Execution of below use case script in JMeter for 10 User 10 Iterations • Supplier Ledger Inquiry (P0411) • Apply Receipts (P03B102) • Work Order Completion (P31114) • Sales Order Entry (P42101)	All the Interactive scripts mentioned in unit test executed together as a 40 User test with 10 Iterations each

Table 5. Use-Case Scenarios for Load Testing

Table 5 describes the interactive user and batch processes that were active in the period of time around the planned scheduled maintenance. There were no failures in both of these windows of load testing.

A key observation is that all load test results reflected the conclusions that no other action was necessary for business continuity, except the ‘refresh’ activity on the Server Manager Console for full business continuity, see Appendix A for more detail.

Appendix C – Test configuration

The JD Edwards EnterpriseOne components that were configured in the architecture for the testing process discussed in this document are listed below.

The JD Edwards EnterpriseOne components were implemented on Oracle Cloud Infrastructure as a standard VM.

JD Edwards EnterpriseOne Enterprise Server:

- Oracle Enterprise Linux 7.8
- Oracle Database 19c (19.11.0.0.0) client

Database Server on Shared Exadata Infrastructure:

- Enterprise Edition Release 19.11.0.0.0 - Production

HTML Server

- Oracle Enterprise Linux 7.8
- WebLogic Server 12c (12.2.1.4.0); Java JDK (1.8.0_272)

Deployment Server:

- Windows Server 2016 Standard
- VM Standard 2.4 with 4 OCPUs
- 4 OCPUs x Intel Xeon Platinum 8167M CPU @ 2.00 GHz
- 60 GB RAM

Server Manager Console:

- Oracle Enterprise Linux 7.8

Test Controller:

- Windows Server 2016 Standard
- VM Standard 2.4 with 4 OCPUs
- 4 OCPUs x Intel Xeon Platinum 8167M CPU @ 2.00 GHz
- 60 GB RAM
- Apache JMeter 5.1.1.r1855137

Software: JD Edwards EnterpriseOne Application 9.2 Update 4 with Tools Release 9.2.6.0

Connect with us

Call +**1.800.ORACLENTERPRISEONE** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.