ORACLE

Manual Deployment of Oracle JD Edwards EnterpriseOne with Oracle Autonomous Database

Configurations and Best Practices for Setting Up Oracle JD Edwards Enterprise One with Oracle Autonomous Database

August 2020 | Version 1.01
Copyright © 2020, Oracle and/or its affiliates
Confidential - Public

PURPOSE STATEMENT

This document describes how to migrate an Oracle JD Edwards EnterpriseOne installation that is running on a Compute Database or a database service (DBS) on Oracle Cloud Infrastructure to an Autonomous Database on Dedicated Exadata Infrastructure (ADB-D). Additionally, this document describes the required tasks to modify the configuration of the servers.

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.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

TABLE OF CONTENTS

Purpose Statement 1

Disclaimer 1

Introduction 3

Supported Releases 3

- Oracle JD Edwards EnterpriseOne Releases 3
- Supported Software 3

Prerequisites 4

Understanding the Migration Process 6

Stopping the ORACLE JD Edwards EnterpriseOne MACHINEs 6

Exporting THE ORACLE JD Edwards EnterpriseOne Data from the source Database Server 6

Importing JD Edwards EnterpriseOne Data to ORACLE AUTONOMOUS DATABASE 8

Configuring JD Edwards EnterpriseOne Servers to Connect to ORACLE AUTONOMOUS DATABASE 12

- Configuring Enterprise Servers 12
- Configuring HTML Servers 13
- Configuring AIS Servers 14
- Configuring Deployment Server 14

Verifying the JD Edwards EnterpriseOne Servers with ORACLE AUTONOMOUS DATABASE 15

Cleaning Up Resources 15

Troubleshooting Tips 15

Best practices for using Oracle Autonomous database with JD Edwards EnterpriseOne 16

- Configuring Production and Non-Production Environments 16
  - Enterprise Server 17
  - HTML Server 17
  - AIS Server 17
  - Deployment Server and Development Clients 17
- Criteria for Patching Window Schedule 18
- Setting up the Patching Schedule 18
- Recommended Configuration and Setup 19

conclusion 19

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

INTRODUCTION

Oracle Autonomous Database is the world's first autonomous data management system in the cloud to deliver automated patching, upgrades, and tuning. As part of these processes, all routine database maintenance tasks are also performed without any human intervention while the system is running. Oracle JD Edwards customers can benefit from the self-driving, self-securing, and self-repairing capabilities of Oracle Autonomous Database. The complete automation of database and infrastructure management activities leads to significant reduction in administrative costs and frees up time of IT professionals to focus on innovation, transforming the core areas of business for Oracle JD Edwards customers. Oracle Autonomous Database will make database management more efficient and induce higher levels of innovation among Oracle JD Edwards customers with the saved time and costs.

This document provides steps, guidelines, and best practices for Oracle JD Edwards customers to migrate data into Oracle Autonomous Database on Dedicated Exadata Infrastructure and integrate the autonomous instance with their existing Oracle JD Edwards implementation. Customers who are running Oracle JD Edwards EnterpriseOne on Oracle Cloud Infrastructure using supported Oracle database versions will be able to migrate their data to Oracle Autonomous Database and integrate it with their Oracle JD Edwards EnterpriseOne installation. Customers who are on premise and are looking to leverage Oracle Autonomous Database can move to Oracle Cloud Infrastructure with a non-autonomous Compute Database or a database service and then follow the guidelines in this document to migrate to Oracle Autonomous Database. On-premise customers not wanting to move to public cloud but want to leverage the benefits of a pay-per-use Oracle Autonomous Database can migrate to Oracle Autonomous Database on Exadata Cloud@Customer.

SUPPORTED RELEASES

To leverage Oracle Autonomous Database with Oracle JD Edwards, customers have to be at the following release levels:

Oracle JD Edwards EnterpriseOne Releases

- Oracle JD Edwards EnterpriseOne Tools Release: 9.2.4.3 and above
- Oracle JD Edwards EnterpriseOne Apps Release: 9.2
- Oracle JD Edwards System Bitness: 32 bit and 64 bit

Supported Software

- Platform: Oracle Enterprise Linux 7.x
- Source Database: Oracle Database 19c, Oracle Database 12c
- Target Database: 19c Oracle Autonomous Database (19.7 DBRU and above) on Dedicated Exadata Infrastructure
- Oracle Weblogic Server: 12.2.1.4

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

PREREQUISITES

- Oracle JD Edwards EnterpriseOne servers must be running on Oracle Cloud Infrastructure
    - **Note:** Customers who are on premise can move to Oracle Cloud Infrastructure by following the steps described in the learning path *Upgrading JD Edwards EnterpriseOne on a One-Click Provisioned Target Environment on Linux* and then following the steps mentioned in this document to migrate to Oracle Autonomous Database. On-Premise customers who do not want to move to a public cloud can upgrade to Oracle Database 19c and migrate to Oracle Autonomous Database on Exadata Cloud@Customer.
- Oracle JD Edwards EnterpriseOne Web Server must be running on Weblogic Server 12.2.1.3 or 12.2.1.4
    - **Note:** For customers running Weblogic Server 12.2.1.3, post-migration it is recommended to upgrade to Weblogic Server 12.2.1.4.
- Oracle JD Edwards EnterpriseOne Enterprise Servers must be using Oracle Database Client 19c
    - **Note:** If the source database version is 12c, post-migration it is recommended to upgrade Oracle Database Client version from 12c to 19c on the Enterprise Server.
- Enabling Long Password

  By default, the Oracle JD Edwards environment comes with a short password for database proxy users. As the Oracle Autonomous Database password policy requires setting up a strong password with a minimum of 12 characters, it is required to enable a long password for the JD Edwards EnterpriseOne Database proxy user. Refer to Post-Installation Security Configurations and Configuring SSL/TLS for JDENET in the *Oracle JD Edwards Tools Security Administration Guide* for instructions on how to enable a long password.
    - **Note:** Ensure the new password complies with the Oracle Autonomous Database password policy. Refer to Create Database Users for details on the password policy.

- Configuring Network Access Controls

  To ensure that your cloud users have access to use and create the appropriate kinds of resources required for provisioning Dedicated Exadata Infrastructure for Oracle Autonomous Database, you must:

    - Define the Identity and Access Management (IAM) policies to grant rights of managing the Autonomous Exadata Infrastructure resources to specific user groups.
    - Define the network access controls by creating VCNs and subnets to permit only the appropriate VCN to be used when a dedicated infrastructure is created. Access control also provides network isolation of resources. During this process, you will need to add security rules to ensure that specific network ports are opened to enable connections from your JD Edwards system to your Oracle Autonomous Database instance .

      For more details on the steps to configure the policies and network access controls, refer to: Plan Access Controls and Create Supporting Resources

- Provisioning Autonomous Exadata Infrastructure and the Autonomous Database Instance

  The Autonomous Exadata Infrastructure needs to be provisioned so that it can host your Oracle Autonomous Database. This database needs to be provisioned in the same VCN (in a private subnet) and tenancy as your JD Edwards servers on Oracle Cloud Infrastructure.

  To provision Autonomous Exadata Infrastructure, log in to Oracle Cloud Infrastructure, go to the Autonomous Exadata Infrastructure screen, select the option Create Exadata Infrastructure, and provide necessary details such as the compartment name, display name, availability domain, Exadata system model and configuration, network settings (VCN), maintenance schedule, and license type. Click the Create Autonomous Exadata Infrastructure option to provision the infrastructure for your Oracle Autonomous Database. For more details on the steps to create an Autonomous Exadata Infrastructure, refer to: Create and Manage Autonomous Exadata Infrastructure Resources

  To provision Oracle Autonomous Database, log in to Oracle Cloud Infrastructure, go to the Autonomous Transaction Processing screen, select the option Create Autonomous Database, and provide necessary details such as the database name, instance name, workload type, deployment type, and so on. Ensure that for the creation of Oracle

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

Autonomous Database, the workload type is selected as “Autonomous Transaction Processing” and deployment type as “Dedicated”. It is recommended that you have the same number of OCPUs and the same storage capacity for Oracle Autonomous Database as you have for your current database setup in Oracle Cloud Infrastructure. During the setup, disable automatic scaling as it is not currently supported for Oracle JD Edwards. For more details on Oracle Autonomous Database creation and its workflow see Typical Workflow for Using Autonomous Transaction Processing.

  - **Note:** When Oracle Autonomous Database is provisioned, ensure that you note the database name which is required for the <DBNAME> variable in the document and also for downloading the database wallet which will be used for the connection to the autonomous instance.

- Private Bucket in Oracle Cloud Infrastructure Object Storage

  A private bucket in Oracle Cloud Infrastructure Object Storage is required to store the data exported from the source database and to import the data into Oracle Autonomous Database during the migration process. Refer to Managing Buckets for instructions on how to create the required object storage.

  - **Note:** The values for bucket, region, and tenancy will be used to populate the variables <BUCKET>, <REGION>, and <TENANT>, respectively, in commands mentioned in the following sections of the document.

- Generating an Auth Token

  An Oracle Cloud Infrastructure auth token is required to upload data dump to a private bucket that is created in Oracle Cloud Infrastructure Object Storage. Generate the auth token by referring to Getting an Auth Token.
  Note: In the commands mentioned in the following sections of the document, the generated auth token value will be used to populate the variable <OCI_AUTHTOKEN> and the value for the Oracle Cloud Infrastructure user name will be used to populate the variable <OCI_USERNAME>.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

UNDERSTANDING THE MIGRATION PROCESS

Four major steps need to be followed to move data from the source Oracle database to the target Oracle Autonomous Database instance and configure the autonomous instance to work with the JD Edwards installation. The details of each of the steps are available in the following sections of this document.

1. Stop the Oracle JD Edwards EnterpriseOne Machines. Before starting migration, you must stop the Oracle JD Edwards EnterpriseOne machines using the Server Manager Console.
2. Export Oracle JD Edwards EnterpriseOne data. You must export the existing data from the source Oracle database and upload it to a bucket that that you have already created in Oracle Cloud Infrastructure Object Storage.
3. Import the Oracle JD Edwards EnterpriseOne data to Oracle Autonomous Database. You must create the required roles and grants to run Oracle JD Edwards in Oracle Autonomous Database. Subsequently, you must run the import command to load the exported data into Oracle Autonomous Database.
4. Configure JD Edwards EnterpriseOne servers to connect to Oracle Autonomous Database. After successful migration, you must reconfigure the Oracle JD Edwards servers to enable them to connect to Oracle Autonomous Database. This configuration includes modifying the tnsnames.ora file for each of the Oracle JD Edwards servers so that the servers point to Oracle Autonomous Database. You must also use Server Manager to modify the database host name to match the Oracle Autonomous Database host name.

STOPPING THE ORACLE JD EDWARDS ENTERPRISEONE MACHINES

You must stop the Oracle JD Edwards EnterpriseOne servers before performing the migration tasks.

To stop the JD Edwards EnterpriseOne servers:

1. Access the Server Manager Console and stop all the Oracle JD Edwards EnterpriseOne Web Servers and Enterprise Servers.
2. Stop the Oracle JD Edwards Deployment Server.
3. Log out of all Oracle JD Edwards Development Clients.

EXPORTING THE ORACLE JD EDWARDS ENTERPRISEONE DATA FROM THE SOURCE DATABASE SERVER

Perform the following steps to export the data from the existing Oracle JD Edwards EnterpriseOne database server and upload it on Oracle Cloud Infrastructure Object Storage.

Note: You must perform these steps on all the available source databases. In this example, the value JDEORCL is used as the tns alias. However, if you have multiple source databases, you must specify multiple tns aliases. You can obtain the tns alias value from the tnsnames.ora file residing at <DB_INSTALL_PATH>/network/admin.

1. Log in to the database server machine and enter this command:

   ```
   sudo su - oracle
   ```

2. Change the JD Edwards DB user profile to the default profile using this procedure:

   a. Connect to the database using sqlplus using this command:

      ```
      sqlplus system@JDEORCL
      ```

   b. Alter the database user profile using this command:

      ```
      alter user <USERNAME> PROFILE DEFAULT;
      ```

   Notes:

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

1. Based on the path codes that exist in the source environment, re-run the above ALTER command for each user name specified in the following table.

2. In a multiple-database setup, the ALTER commands for the shared pathcode must be executed in the database that has the system data source.

PATHCODE	USER NAME OR SCHEMA
Shared (Required)*	JDE, SY920, SVM920, OL920, DD920, RTE_WLS, QUARTZUSR
Production	PD920, PRODDTA, PRODCTL
Pristine	PS920, PS920DTA, PS920CTL
Prototype	PY920, CRPDTA, CRPCTL
Development	DV920, DEVSAVE, TESTDTA, TESTCTL

Table 1: Path Code-Specific Details

3. Export the JD Edwards EnterpriseOne data using this procedure:

   a. Create a directory with sufficient free space. For example:

      /u01/dump

   b. Run this command:

      ```
      sqlplus system@JDEORCL
      ```

      Enter the correct password when prompted.

   c. Run the command:

      ```
      Create or replace directory mv_adb_dir as <PATH_TO_DIRECTORY>
      ```

      Where PATH_TO_DIRECTORY is the path of the directory created in Step a.

   d. Exit sqlplus

   e. Run this command to export the dump file:

      ```
      expdp system@jdeorcl DIRECTORY=mv_adb_dir DUMPFILE=jdedmp_01_01.dmp
      SCHEMAS=<LIST_OF_SCHEMAS> exclude=cluster,db_link LOGFILE=mv2adb.log
      ```

      Notes: When prompted, enter the password for the system user. Replace the values for <LIST_OF_SCHEMAS> by referring to Table 1: Path Code-Specific Details.

4. Upload the generated dump files on Oracle Cloud Infrastructure Object Storage by running this command:

   ```
   curl -u '<OCI_USERNAME>:<OCI_AUTHTOKEN>' -X PUT
   https://swiftobjectstorage.<REGION>.oraclecloud.com/v1/<TENANT>/<BUCKET>/ -T jdedmp_01_01.dmp
   ```

   Replace the **highlighted** variable parameters with valid values before running the command.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

IMPORTING JD EDWARDS ENTERPRISEONE DATA TO ORACLE AUTONOMOUS DATABASE

The JD Edwards EnterpriseOne data uploaded on Oracle Cloud Infrastructure Object Storage must be imported into Oracle Autonomous Database by following these steps:

1. Log in to the source database server machine and run this command:

   ```
   sudo su – oracle
   ```

2. Connect to Oracle Autonomous Database using this procedure:

   a. Download the Oracle Autonomous Database wallet from Oracle Cloud Infrastructure Console.

   b. Extract the Oracle Autonomous Database wallet zip file and save it in a directory to which the **oracle** user has access.

3. Export the TNS_ADMIN environment variable using this command:

   ```
   export TNS_ADMIN=<WALLET_LOCATION>;
   ```

   As we are in the source database server machine, the TNS_ADMIN variable points to the tnsnames.ora file of the source database. Hence, it is required to set TNS_ADMIN environment variable to point to tnsnames.ora of Autonomous Database instance.

4. Verify the Oracle Autonomous Database connection by running this command:

   ```
   sqlplus admin@<DBNAME>_high
   ```

5. To create the system-level tablespaces, which are required for JD Edwards EnterpriseOne shared objects, log in to sqlplus as previously described and run the following commands:

   Note: This step is required only for an Autonomous Database instance to which the shared path code data will be migrated.

   ```
   create tablespace sy920t datafile size 250M autoextend on next 25M maxsize 1000M logging
   extent management local segment space management auto;
   create tablespace sy920i datafile size 100M autoextend on next 10M maxsize 1000M logging
   extent management local segment space management auto;
   create tablespace svm920t datafile size 250M autoextend on next 25M maxsize 1000M logging
   extent management local segment space management auto;
   create tablespace svm920i datafile size 100M autoextend on next 10M maxsize 1000M logging
   extent management local segment space management auto;
   create tablespace DD920t datafile size 250M autoextend on next 25M maxsize 1000M logging
   extent management local segment space management auto;
   create tablespace DD920i datafile size 100M autoextend on next 10M maxsize 1000M logging
   extent management local segment space management auto;
   create tablespace OL920t datafile size 250M autoextend on next 25M maxsize 1000M logging
   extent management local segment space management auto;
   create tablespace OL920i datafile size 100M autoextend on next 10M maxsize 1000M logging
   extent management local segment space management auto;
   create tablespace RTE_WLS logging datafile size 1G autoextend on next 2G maxsize unlimited
   extent management local autoallocate segment space management auto  online;
   create tablespace QUARTZ logging datafile size 200M autoextend on next 20M maxsize 1G extent
   management local autoallocate segment space management auto online;
   ```


Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

6. Create path code-specific tablespaces by logging in to Oracle Autonomous Database through sqlplus using the following commands, where each command is a single contiguous line with no line returns:

Note: Replace **TABLESPACE_NAME** variables in the following commands with the relevant values associated with the path codes deployed in the environment:

PATH CODE	TABLESPACE_NAME
Production	PD920
Pristine	PS920
Prototype	PY920
Development	DV920

Table 2: Path Code-andTtablespace Names

```
create tablespace <TABLESPACE_NAME>t datafile size 3072M autoextend on next 300M maxsize 9120M
logging extent management local segment space management auto;

create tablespace <TABLESPACE_NAME>i datafile size 1536M autoextend on next 150M maxsize
3072M logging extent management local segment space management auto;
```

Note: Replace the **TABLESPACE_NAME** variables in the following commands with the relevant values associated with the path codes:

PATH CODE	TABLESPACE_NAME
Production	PROD
Pristine	PS920
Prototype	CRP
Development	TEST

Table 3: Path Code-andTtablespace Names

```
create tablespace <TABLESPACE_NAME>DTAi datafile size 1000M autoextend on next 10M maxsize
2000M logging extent management local segment space management auto;

create tablespace <TABLESPACE_NAME>DTAt datafile size 1000M autoextend on next 20M maxsize
2000M logging extent management local segment space management auto;

create BIGFILE tablespace <TABLESPACE_NAME>CTLi datafile size 1000M autoextend on next 90M
maxsize 2000M logging extent management local segment space management auto;

create BIGFILE tablespace <TABLESPACE_NAME>CTLt datafile size 1000M autoextend on next 125M
maxsize 2000M logging extent management local segment space management auto;
```

Note: For the Development path code, create DEVSAVE tablespace as well using the following commands:

```
create tablespace DEVSAVEt datafile size 100M autoextend on next 300M maxsize 8192M logging
extent management local segment space management auto;

create tablespace DEVSAVEi datafile size 100M autoextend on next 150M;
```


Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

7. Create your Oracle Cloud Infrastructure credential in Oracle Autonomous Database.

    a. Connect to Oracle Autonomous Database using sqlplus and run the following command:

    ```
    execute DBMS_CREDENTIAL.CREATE_CREDENTIAL (credential_name => 'MV2ADB_CRED_NAME',
    username => '<OCI_USERNAME>',password => '<OCI_AUTHTOKEN>');
    ```

    b. If you are using source database version 12c , you need to run the following additional command:

    ```
    alter database property set default_credential ='ADMIN.MV2ADB_CRED_NAME';
    ```

8. Create the JD Edwards-specific roles in the environment. Run the following commands by connecting to sqlplus, creating the roles, and providing grants to those roles.

    ```
    CREATE ROLE "JDE_ROLE";
    CREATE ROLE "JDEADMIN";
    CREATE ROLE "JDEUSER";
    GRANT CREATE SESSION,CREATE TABLE,CREATE VIEW,SELECT ANY DICTIONARY,SELECT_CATALOG_ROLE TO
    "JDE_ROLE";
    ```

9. To find cpu_count:

    a. Connect to sqlplus by running this command:

    sqlplus admin@**<DBNAME>**_high

    b. Run this command :

    ```
    show parameter cpu_count;
    ```

    ```
    SQL> show parameter cpu_count;

    NAME                                 TYPE        VALUE
    ------------------------------------ ----------- ----------
     cpu_count_startup                   integer     46
    cpu_count                            integer     8
    SQL>
    ```

    Figure 1: Retrieving CPU_COUNT Parameter Value

    c. See the value column in the output to find the value of cpu_count.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

10. To import the JD Edwards EnterpriseOne data to Oracle Autonomous Database:

    a. If the source database is running on 19c, use the following command:

    ```
    nohup impdp admin/<password>@<DBNAME>_high CREDENTIAL=MV2ADB_CRED_NAME
    DUMPFILE='https://swiftobjectstorage.<REGION>.oraclecloud.com/v1/<TENANT>/<BUCKET>/jded
    mp_01_01.dmp' REMAP_TABLESPACE=USERS:DATA
    parallel=<CPU_COUNT>
    NOLOGFILE=y &
    ```

    b. If the source database is running on 12c, use the following command:

    ```
    nohup impdp admin/<password>@<DBNAME>_high
    DUMPFILE=default_credential:https://swiftobjectstorage.<REGION>.oraclecloud.com/v1/<TEN
    ANT>/<BUCKET>/jdedmp_01_01.dmp  REMAP_TABLESPACE=USERS:DATA
    parallel=<CPU_COUNT>
    NOLOGFILE=y &
    ```

    Note: Replace the CPU_COUNT variable with the value retrieved in the previous step. The tns service high is used, as it provides better performance for the import operation, as described in the Oracle Autonomous Database documentation Using Oracle Autonomous Transaction Processing on Dedicated Exadata Infrastructure.

11. Verify the nohup.out file and check whether there is any error in import.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

CONFIGURING JD EDWARDS ENTERPRISEONE SERVERS TO CONNECT TO ORACLE AUTONOMOUS DATABASE

To migrate JD Edwards EnterpriseOne data from a Compute Database or a database service to Oracle Autonomous Database, the configuration of EnterpriseOne servers must be changed to connect to Oracle Autonomous Database. The values for the configuration are retrieved from the Oracle Autonomous Database wallet and the JD Edwards connection uses the TP Service during runtime.

Refer to the following screenshot to get the respective values of Connection String, Oracle Autonomous Database host name, and TP Service name. These values will be used later in the configuration.

Figure 2: TP Service Example from Oracle Autonomous Database Wallet

Configuring Enterprise Servers

Perform the following steps on all the Enterprise Servers.

1. Log in to Enterprise Server Machine.
2. Go to **oracle** user using this command:

   ```
   sudo su - oracle
   ```

3. Go to <DB_CLIENT>/network/admin path, and edit tnsnames.ora.
4. Copy the tns connection string of the TP Service from the Oracle Autonomous Database wallet (refer to Figure 2) and paste against the tnsalias. In the following example the alias is JDEORCL, which is derived from the tnsnames.ora file of DB Client.

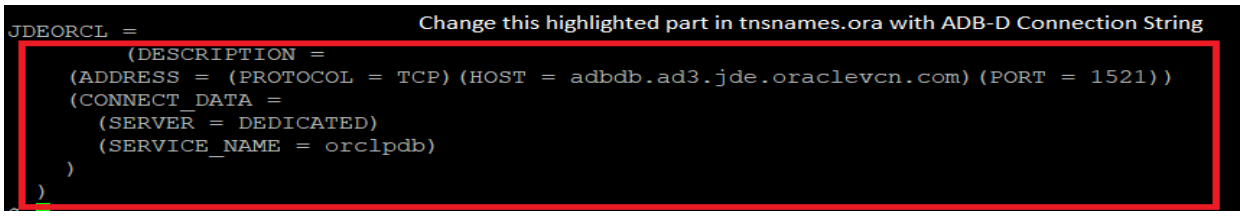


Figure 3: Tnsnames.ora Before Configuration

Repeat this step to map all the available tns aliases to the respective Autonomous Database wallets. The tnsnames.ora file should look like this:

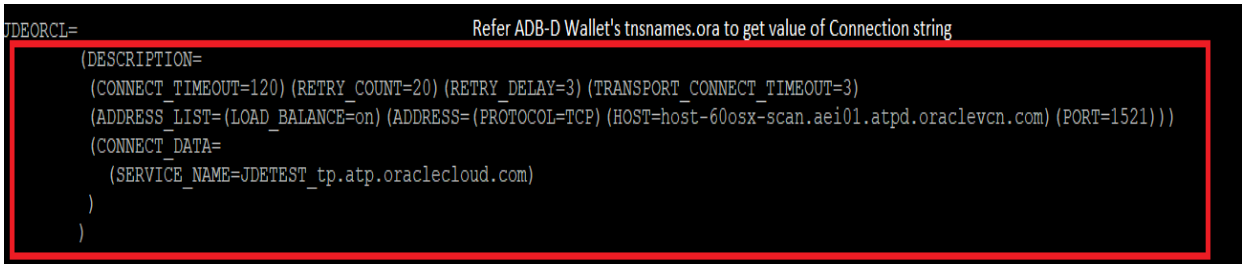


Figure 4; Tnsnames.ora After Configuring the Oracle Autonomous Database Instance

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

5. Go to JDE920 user and run this command:

   sqlplus JDE/<JDE_USER_PWD>@JDEORCL

   If the connection is successfully established, the system will display the Database 19.0.0.0 banner.

6. Log in to the Server Manager console.
7. Go to the Enterprise Server Managed Instance, Configuration, Advanced View, Database, and modify the Database Server Name value with the Oracle Autonomous Database host name (refer to Figure 2).
8. Start the Enterprise Server instance.

Configuring HTML Servers

Perform the following steps on all the HTML Servers.

1. Log in to the Server Manager Console.
2. Go to HTML Managed Instance, Configuration, Advanced View, Database.
3. Inside the JDBj Bootstrap Datasource section, change the Database Server Name value to the Oracle Autonomous Database host name.

Figure 5: Configuration for HTML Servers

4. Copy the tns connection string from the Enterprise Server machine and paste it in the File Contents field in the Oracle Database Settings section.
5. Start the HTML Server instance.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

Configuring AIS Servers

Perform the following steps on all the AIS Servers.

1. Log in to the Server Manager Console.
2. Go to the AIS Managed Instance, Configuration, Advanced View, and Miscellaneous.
3. Inside the Scheduler Configuration section, change the “ojdbc url” value to:

   jdbc:oracle:thin:@<ORACLE AUTONOMOUS DATABASE_HOST>:1521/<TP_Service_URL>

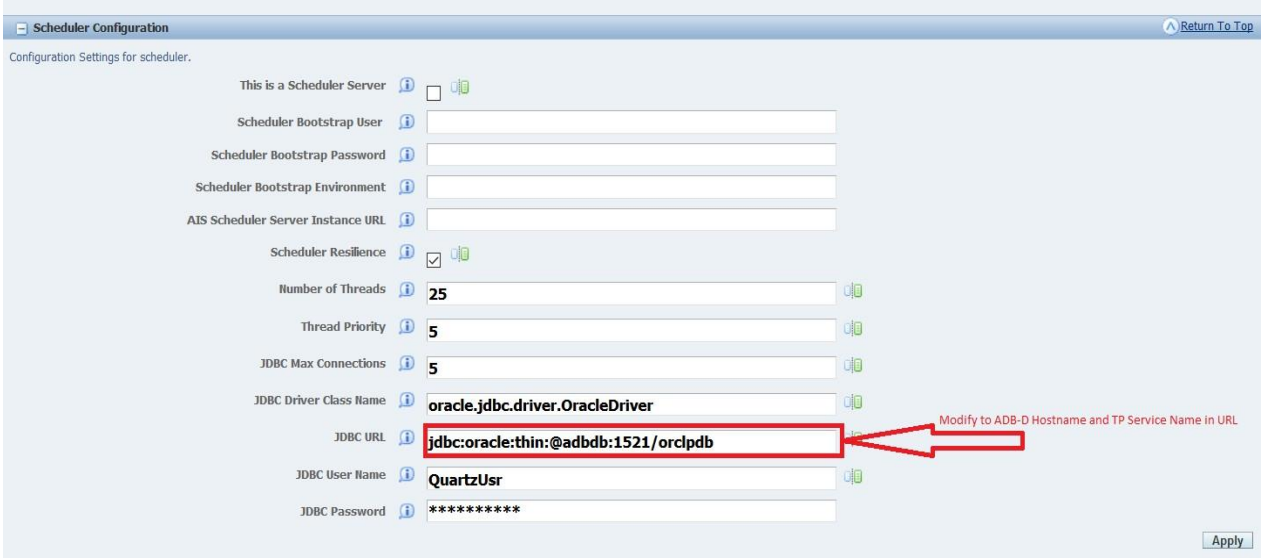


Figure 6: Configuration for AIS Servers

4. Start the AIS Server instance.

Configuring Deployment Server

Perform the following steps on the Deployment Server.

1. Log in to the Deployment Server machine.

2. Copy the tns connection string from the Enterprise Server and paste it at these locations:

   a. E1Local
   <INSTALLATION_DRIVE>:\JDE\DEP\Oracle\E1Local\NETWORK\ADMIN

   b. DB Client
   <INSTALLATION_DRIVE>:\JDE\DEP\oracle1212\product\client_1\network\admin.

   Note: Before you can perform the next step, you must also copy this same tnsnames.ora connection string to all the Development Clients, including the one that exists on the Deployment Server.

3. Log in to the Development Client on the Deployment Server and recreate the Installation Plan with the new data source details as now database hostname changed to Autonomous database hostname. Go to Working with Installation Planner for an Install guide for details of how to work with installation plan

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

VERIFYING THE JD EDWARDS ENTERPRISEONE SERVERS WITH ORACLE AUTONOMOUS DATABASE

After you configure all the JD Edwards EnterpriseOne servers to point to Oracle Autonomous Database, perform these steps to ensure that the servers are functioning as expected.

1. Log in to the Server Manager Console.
2. Go to the Health Check menu.
3. Run separate health checks by setting the Target Type field as ES, JAS, or AIS for the health check of the respective server.
4. Verify that all the health checks are successful.

CLEANING UP RESOURCES

It is recommended to delete the objects that were uploaded during the migration process. To delete the objects:

1. Log in to the Compute Database machine and change the user to the **oracle** user by running this command:

   ```
   sudo su –oracle
   ```

2. Run this command:

   ```
   curl -u '<OCI_USERNAME>:<OCI_AUTHTOKEN>' -X DELETE
   https://swiftobjectstorage.<REGION>.oraclecloud.com/v1/<TENANT>/<BUCKET>/jdedmp_01_01.dmp
   ```

TROUBLESHOOTING TIPS

This section provides a troubleshooting tip to resolve the import error “ODM err:ODM HTTP Bad Gateway” that might be encountered during migration of the JD Edwards installation to Oracle Autonomous Database.

While impdp is running on Oracle Autonomous Database, you might encounter the following error which causes the failure of data import in some tables. This error only happens occasionally and not every time.

```
ORA-31640: unable to open dump file "https://swiftobjectstorage.us-ashburn-
1.oraclecloud.com/v1/ATPDPreview12/test_bucket/expdp_01_01.dmp" for read
ORA-19505: failed to identify file "https://swiftobjectstorage.us-ashburn-
1.oraclecloud.com/v1/ATPDPreview12/test_bucket/expdp_01_01.dmp"
ORA-17503: ksfdopn:11 Failed to open file https://swiftobjectstorage.us-ashburn-
1.oraclecloud.com/v1/ATPDPreview12/test_bucket/expdp_01_01.dmp
ORA-17500: ODM err:ODM HTTP Bad Gateway
```

Follow these steps to resolve the issue:

1. Rerun the import on the tables on which the import failed due to this error.

2. List the tables on which the import failed due to this error by referring to the import logs.

3. Run the following command:

   ```
   impdp admin/<password>@tp CREDENTIAL=MV2ADB_CRED_NAME
   DUMPFILE='https://swiftobjectstorage.<REGION>.oraclecloud.com/v1/<TENANT>/<BUCKET>/jdedmp_01_0
   1.dmp' TABLES=<LIST_OF_TABLE> table_exists_action=replace NOLOGFILE=y
   ```

   Note: In the above command, LIST_OF_TABLE is a comma-separated list of tables on which the import has failed.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

BEST PRACTICES FOR USING ORACLE AUTONOMOUS DATABASE WITH JD EDWARDS ENTERPRISEONE

Oracle Autonomous Database can be configured in different methods based on the variations in your JD Edwards deployment topology, which could be a multi-path code environment or a combination of production and non-production environments. This section provides a few best practices to consider while configuring such environments with Oracle Autonomous Database.

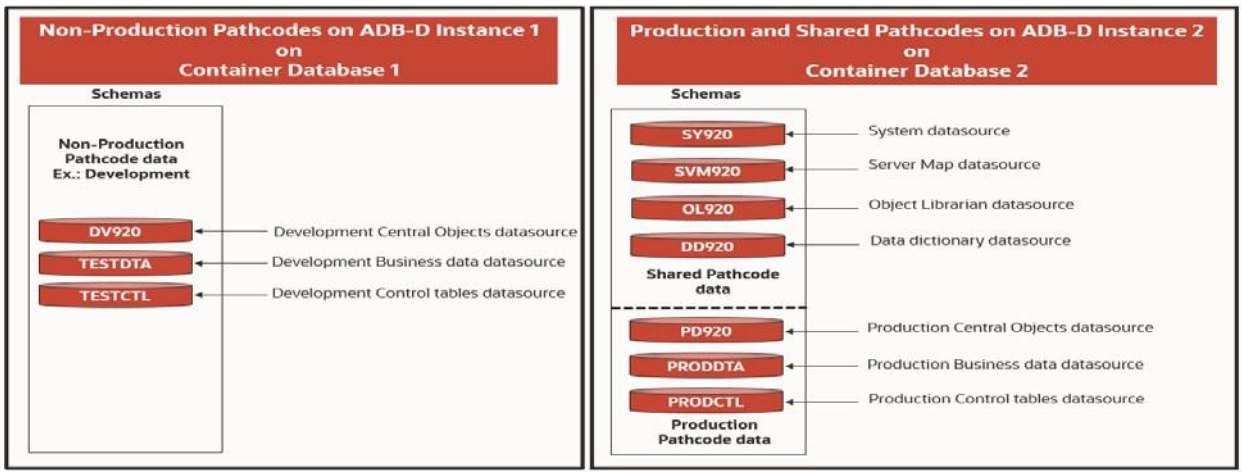
Configuring Production and Non-Production Environments

Many organizations require a non-production JD Edwards environment. It could be used for testing applications for JD Edwards patches and upgrades, or it may be used as a development environment. With Oracle Autonomous Database, the database patches are applied automatically and customers should validate the database patches and assess their impact on existing business processes to mitigate risks before they are applied on the production environment. The non-production environment can be used for this validation.

To serve the purpose of having independent production and non-production environments, Oracle JD Edwards recommends having the production and non-production databases on separate container databases. As automated database patching schedules can be configured at the container database level for Oracle Autonomous Database, this setup provides the flexibility of defining independent patching schedules for the non-production and production environments.

Oracle JD Edwards recommends that the patching schedule on the production environment container database should be at least one month after the patching schedule for the non-production environment of Oracle Autonomous Database. The period between the two patching schedules can be decided based on customer need. The period between the database patching on non-production and production environments enables the customers to perform validation activities on the non-production environment and detect any issues before the patches are applied on the production environment. This gap in patching schedules helps customers to assess the impact of any patching issues, request for fixes, or take decisions on deferring the patching on the production environment by another month if the issues are expected to take longer to detect and fix. If customers have the required processes and tools in place to test the production workload on the patches applied on the non-production environment, they could do so by increasing the capacity of the non-production environment to assess the impact under workload. Validating the database patches on non-production environments provides a a high level of confidence and assurance to the customers about their production database patching. Validation saves significant time and effort by mitigating the impact on the production environment.

JD Edwards recommends segregating non-production and production data into different Autonomous Database instances residing in different autonomous container databases. The following diagram depicts how the JD Edwards production, shared, and non-production data sources should be distributed between different Autonomous Databases residing in different autonomous container databases when data is migrated.

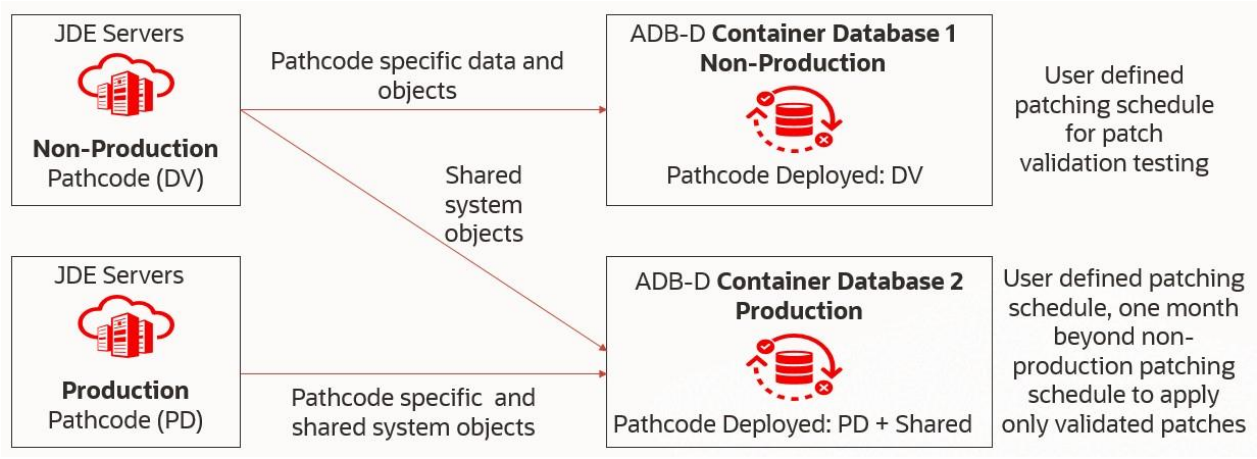


The first autonomous database instance created in autonomous container database 1 is used to store all the non-production path codes. In this example, we have considered the development path code but it could be either pristine, prototype, development, or all three. In this Autonomous Database instance, the data sources of path code-specific central objects, control tables, and business data are stored.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

The second autonomous database instance created in autonomous container database 2 is used to store production and shared path code data. In this database instance, data sources such as a system, a server map, an object librarian, a data dictionary, and pathcode-specific central objects, control tables, and business data are stored.

The integration between the JD Edwards servers and the different data sources that have independent patching schedules is depicted in the below diagram.



As depicted in the above diagram, the non-production pathcode needs to be isolated in instance 1 of Autonomous Database Dedicated created in autonomous container database 1. Additionally, the path code for the production environment and shared objects needs to be located in instance 2 of Autonomous Database Dedicated created in autonomous container database 2.

Non-production servers will require connection to both the Autonomous Database instances; these servers will connect to Autonomous Database instance 2 for the shared system tables and objects and connect to Autonomous Database instance 1 for path code-specific tables and objects. However, the production servers will only interact with Autonomous Database instance 2 for both the path code-specific and shared tables and objects.

The server-wise configurations that are required for each of the JD Edwards components to achieve the recommended topology are detailed below.

Enterprise Server

JD Edwards non-production Enterprise Servers need to be configured with the correct tnsnames.ora, that is tnsnames.ora located in the Database Client needs to have information about both the Autonomous Database instances. The production pathcode Enterprise Server does not require information about Autonomous Database instance 1.

Apart from this, in JDE.ini details need to be provided about the shared system database that is information about Autonomous Database instance 2 in the "[DB SYSTEM SETTINGS]" section for the production server and the non-production server.

HTML Server

For the HTML Servers, you must provide the information of both the Autonomous Database instances. To do so, from the Server Manager Console, update the tnsnames.ora content of that particular HTML instance by navigating to Configuration, Advanced View, Database menu, and enter a valid in the File Content field. Ensure that the configuration files are synchronized for the instance after the configuration is completed.

AIS Server

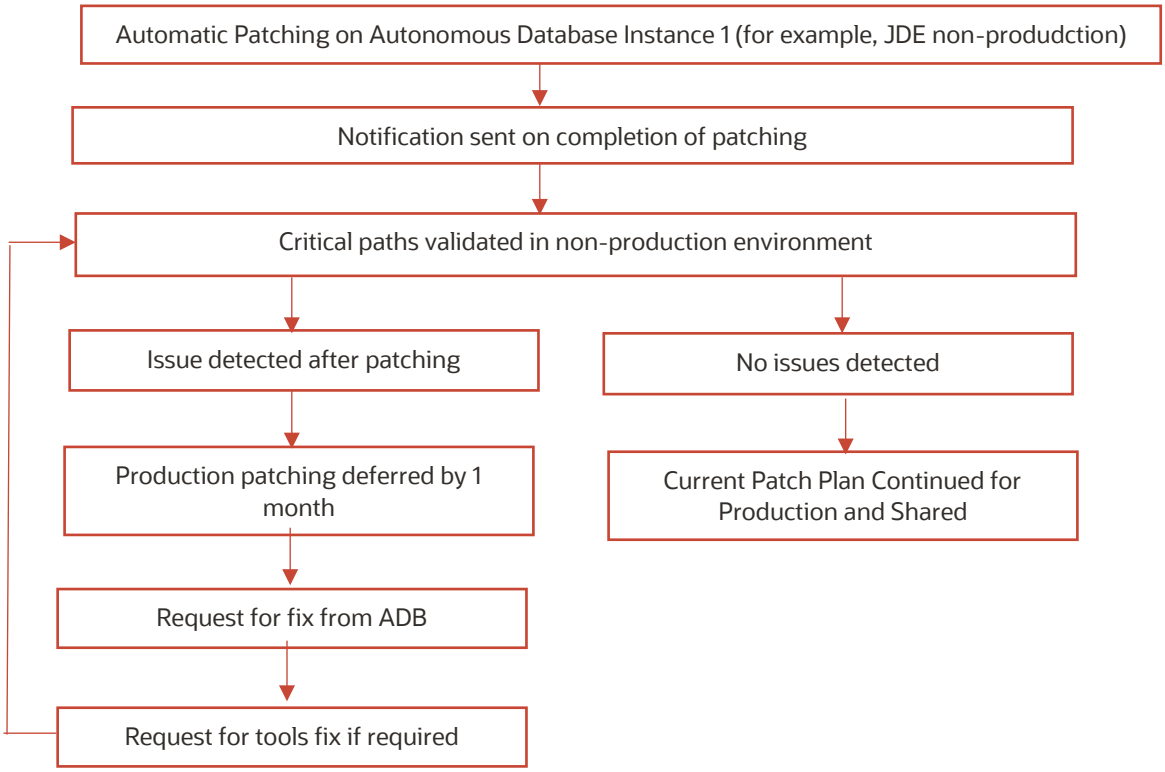
No additional configuration is necessary for the AIS Servers to work with Oracle Autonomous Database.

Deployment Server and Development Clients

Modify the tnsnames.ora for the database client in the Deployment Server and the Development Client machines; ensure that it is configured with the details of both the Autonomous Database instances.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

The following image shows the overall process flow for validating the database patches ahead of applying the patches in the production and shared system Autonomous Database instance:



Criteria for Patching Window Schedule

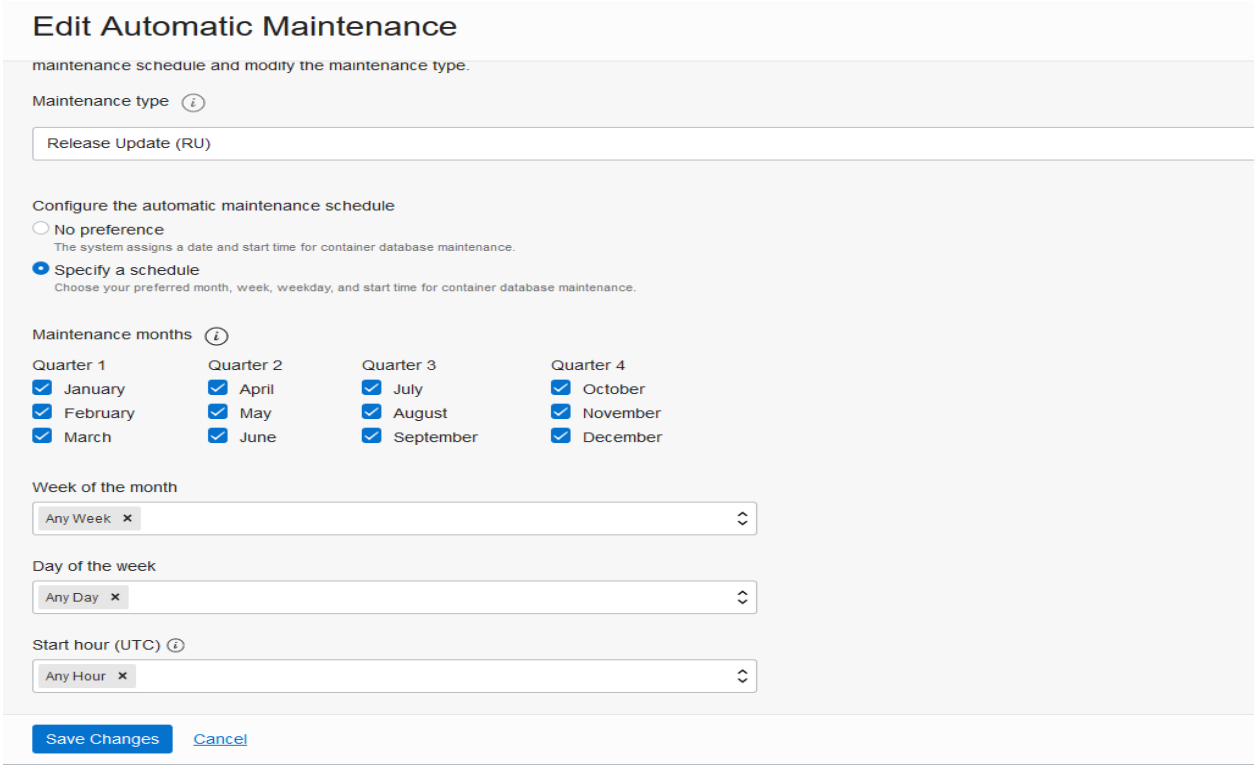
Patching window schedules on both your production and non-production container databases can be configured manually. Oracle JD Edwards recommends you to have the patching window on your non-production container database at least one month prior to the patching window on your production container database to accommodate for the validation of patches before they are applied in the production environment.

Although there is no downtime or impact on processes during the patching window, it is recommended that customers avoid scheduling their patching windows for the production environment during periods of heavy workload on their JD Edwards setup, such as during month-end, quarter-end, or year-end. Patching windows should be scheduled either during non-business hours or on weekends. This timing would help customers mitigate risks that might occur due to failure in patches.

Setting up the Patching Schedule

Patching for the Oracle Autonomous Database occurs at the respective autonomous container database level and hence maintenance and patching schedules are defined at the autonomous container database level. A default preference is not set for the maintenance schedule, therefore patching could happen randomly based on system decided time intervals. Oracle JD Edwards recommends customers to select their preferred patching schedule based on the patching window criteria specified in the previous section. The screenshot below shows the different parameters available for the maintenance schedule. Refer to the management portion of this document: Create and Manage Autonomous Container Database for more details.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public



Recommended Configuration and Setup

Oracle JD Edwards recommends that you have the same number of OCPUs and the same storage capacity for Oracle Autonomous Database, as you have for your current database setup in Oracle Cloud Infrastructure. Oracle Autonomous Database is by default deployed as a Real Application Cluster.

To achieve the best performance of the JD Edwards system with Oracle Autonomous Database, it is recommended to provision Autonomous Database instances for both the production and non-production environments in the same availability domain where the rest of the JD Edwards servers are residing. Provisioning your Oracle Autonomous Database in a different availability domain than your JD Edwards setup might degrade the performance due to network latency.

CONCLUSION

Oracle JD Edwards customers can adopt Oracle Autonomous Database using the manual deployment method described in this document and benefit from better efficiency and higher level of innovation. Using JD Edwards with Oracle Autonomous Database ensures that customers have a high-performaing and highly available system with lower operational costs. By following the best practices described in this document, customers can ensure that they have the right setup, configurations, and topology to get the maximum benefit from Oracle Autonomous Database. By adopting Oracle Autonomous Database with JD Edwards, customers can reimagine IT management and focus more on value-driven innovation.

Copyright © 2020, Oracle and/or its affiliates | Confidential - Public

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

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

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Manual Deployment of Oracle JD Edwards EnterpriseOne with Oracle Autonomous Database
August, 2020