



ORACLE

# Oracle Enterprise Manager: Database Migration Workbench

---

November 2020 | Version 1.00  
Copyright © 2020, Oracle and/or its affiliates

# Table of Contents

<b>Introduction .....</b>	<b>2</b>
<b>Database Migration Challenges .....</b>	<b>2</b>
• Determining which Databases to Migrate .....	3
• Estimating the Size of the Destination Database .....	3
• Choosing the Appropriate Migration Mechanism .....	4
• Measuring the Success of a Migration .....	4
• Comparing Performance after Migration .....	5
• Providing an Integrated Solution.....	6
<b>Introducing Database Migration Workbench.....</b>	<b>7</b>
<b>Unified Guided Wizard .....</b>	<b>8</b>
<b>Sizing Advisor .....</b>	<b>9</b>
<b>Enhanced Schema Advisor.....</b>	<b>10</b>
<b>Integration with Deployment Procedures.....</b>	<b>11</b>
• View the Summary of Existing Migrations .....	11
• Monitor Migrations in Real-Time .....	11
• Schedule Migrations .....	13
• Pause and Resume Migrations.....	13
• Rollback Migrations .....	13
• Customize Migration Workflow .....	13
<b>Analyze Migrations.....</b>	<b>14</b>
<b>Compare Performance after Migration.....</b>	<b>14</b>
• Production Class STS Collections.....	15
• DevOps Class STS Collections.....	15
<b>Integrate with Other Tools .....</b>	<b>16</b>
<b>Summary.....</b>	<b>17</b>

## INTRODUCTION

According to a recent survey conducted by the International Oracle User Group, most DBAs spend too much time dealing with day-to-day maintenance and troubleshooting issues, and not enough time proactively planning ways to enhance the critical business systems under their purview. In response, the IT industry has begun a decisive shift towards cloud services to free up skilled technology resources for more productive pursuits. This transition creates a hybrid estate for most organizations, which requires new management tools and practices. Today's organizations need versatile tools that streamline the process of migrating resources to the cloud and simplify the management of hybrid environments.

Oracle Enterprise Manager includes a Database Migration Workbench that accommodates the new APIs and management structures associated with cloud databases—including Oracle Autonomous Database. The Database Migration Workbench automates the planning cycle, helps DBAs determine which databases to move, selects the optimal migration tools to use for each move, and identifies existing dependencies within the database. For example, the Database Migration Workbench can help you decide which databases will work best with Oracle Autonomous Transaction Processing. It can also help you plan the migration process, migrate the data using the proper migration methodology, and validate the results by comparing before-and-after performance. Once you are satisfied with the migration, you can automate these management tasks going forward. Enterprise Manager launches the appropriate migration tools, moves the workloads, and validates that everything is working properly in the new cloud environment.

## DATABASE MIGRATION CHALLENGES

Many customers use Enterprise Manager as the primary tool to monitor, manage, clone, and diagnose databases within their premises. When an organization decides to move workloads to the cloud, it goes through a common set of challenges in various phases including – Planning, Migration, and Post-Migration Analysis

- **Determining which Databases to Migrate**

Typically the decision to migrate workloads to cloud starts at the application level. A particular application is chosen to migrate to the cloud, typically with its entire stack. In such cases, the challenge for administrators is to find the right set of schema or databases to migrate to the cloud. With years of investment in building and fine-tuning of applications, dependencies within applications and databases increase multi-fold.

Maintaining a relationship among all dependent components using traditional methods becomes extremely difficult.

A migration tool should help administrators identify relationships across various components and provide a list of candidate databases for migration. Often, administrators need to identify a subset of the database fleet for migration. They may be interested in migrating only databases used for DevOps purposes or even only databases used by a particular team in a specific line of business. Without sophisticated tools, it is difficult to maintain such a relationship model and query it when needed.

- **Estimating the Size of the Destination Database**

This is a common question that helps sponsors of migration projects to estimate the cost of migrating workloads to the cloud. Also, application administrators want to ensure that performance of applications is improved after moving to Oracle Cloud. Every organization invests considerable time and effort to optimize the performance of critical applications. So when dev/test workloads are migrated to cloud, administrators want to ensure that they can fine-tune applications with minimal efforts to reach optimal performance.

Also, applications run varied workloads at various times. Sometimes an application runs a peak workload with resource utilization reaching new limits and sometimes it runs low workload resulting in sub-optimal usage of resources. Furthermore, when administrators want to consolidate multiple Databases on Oracle Cloud, they need to assess the performance impact of the aggregated workload.

Additionally, even before the migration process is started, a CTO needs to decide how much capacity should be purchased on the cloud. Even though the cloud credit based model charges, customers, based on the number of cloud credits consumed, it makes sense for any CTO to project the costs of moving to cloud for the entire Financial Year.

Without a sophisticated model, it is difficult to understand the cost and performance impact of consolidation.

- **Choosing the Appropriate Migration Mechanism**

Oracle provides multiple choices for moving data between databases. Users can choose a logical data movement method like Data Pump to move data across versions and platforms. They can choose the Zero Downtime Migration method to avoid downtime for mission-critical applications. In the case where applications are running on a cloud compatible version and platform, they can also choose a simple RMAN Backup and Restore method. Users can also choose variants of these methods to support cross-platform and cross-version migrations.

Another aspect that users need to consider while moving database workloads to the cloud is the type of the destination cloud service. Oracle Autonomous Databases support certain methods for migrating workloads but other Oracle database services like Database Cloud Service on Virtual Machine or Bare Metal support all methods. So often administrators need to analyze which method works best for migrating databases in their data center.

- **Measuring the Success of a Migration**

In the new era of agile organizations, administrators are expected to have applications up and running immediately after migration. But migration projects can take days or months. A migrated database starts accepting requests from an application only after the actual migration work is completed. Multiple issues

cause delays in the migration process. Sometimes, some part of data is not migrated due to compatibility issues, some objects fail to migrate because they are considered duplicate or there can be multiple restrictions on what kind of data is allowed in the destination database.

A database migration process is a long-running process with lots of components like the source database, intermediate storage, destination database, networking, and storage space. If an issue occurs at any of these steps, it is very difficult to identify the component where the issue happens. In such a case, if issues occur during the migration process, administrators need to go through log files for each component to find the root cause.

The success of migration also means that once the database is up and running it performs at better or at least par with the on-premises database. If this is not the case then DBAs should get a comparative report on potential issues so they can fine-tune the database.

Along with these requirements, an automated migration tool is also expected to orchestrate the entire workflow for migration. It is also expected to show the on-going progress granularly so administrators can monitor migrations in real-time. Moreover, typical migration jobs are long-running so in case there is an issue, the migration tool should send alerts or notifications to Administrators immediately. DBAs also expect the ability to pause, resume, or resubmit the migration job in case manual intervention is required while the migration is still in progress.

- **Comparing Performance after Migration**

Sizing allows administrators to estimate the size of a destination database to maintain performance benchmarks. But how do administrators ensure that the performance a critical SQL runs is better or at least remains the same even after migration? Usually, after every migration activity, administrators run multiple tests manually to ensure that migration is completed successfully in all functional and performance aspects. However, this method of ensuring performance parameters can be very error-prone. Often, SQL statements

collected by administrators to form a test case do not accurately represent critical parameters required to measure performance. Also, it is very difficult to isolate and capture a SQL statement that creates a peak workload.

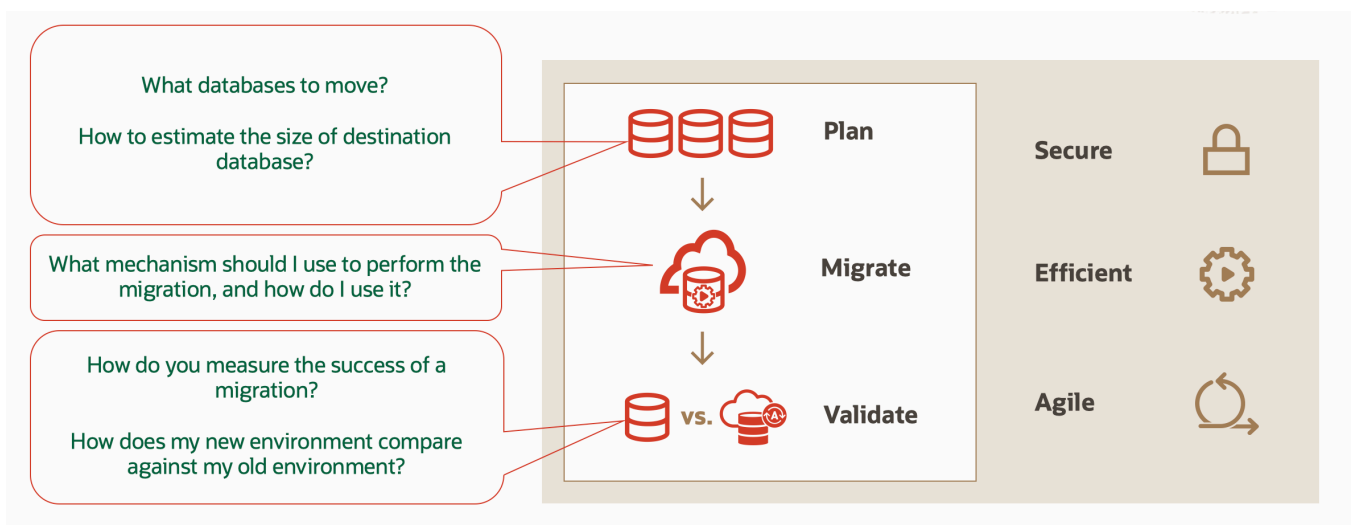
To compare performance correctly it is essential to collect SQL execution data over time along with performance statistics. DBAs expect that such a set of SQL statements should be executed on the source database before migration and destination database after migration to get a realistic comparison of performance metrics.

- **Providing an Integrated Solution**

Based on the issues described above, we can group these challenges into three phases :

1. **Planning Phase** which covers identifying and selecting a source for migrations, analyzing schema for migrations, selecting a destination, estimating capacity sizing, and cost for the destination database
2. **Migration Phase**, which covers all methods which can be used for migration based on various criteria like destination database type., size of the source database, character set and platform of source and destination database, etc.
3. **Analysis Phase (post-migration)**, which covers status of migration and performance comparison before and migration.

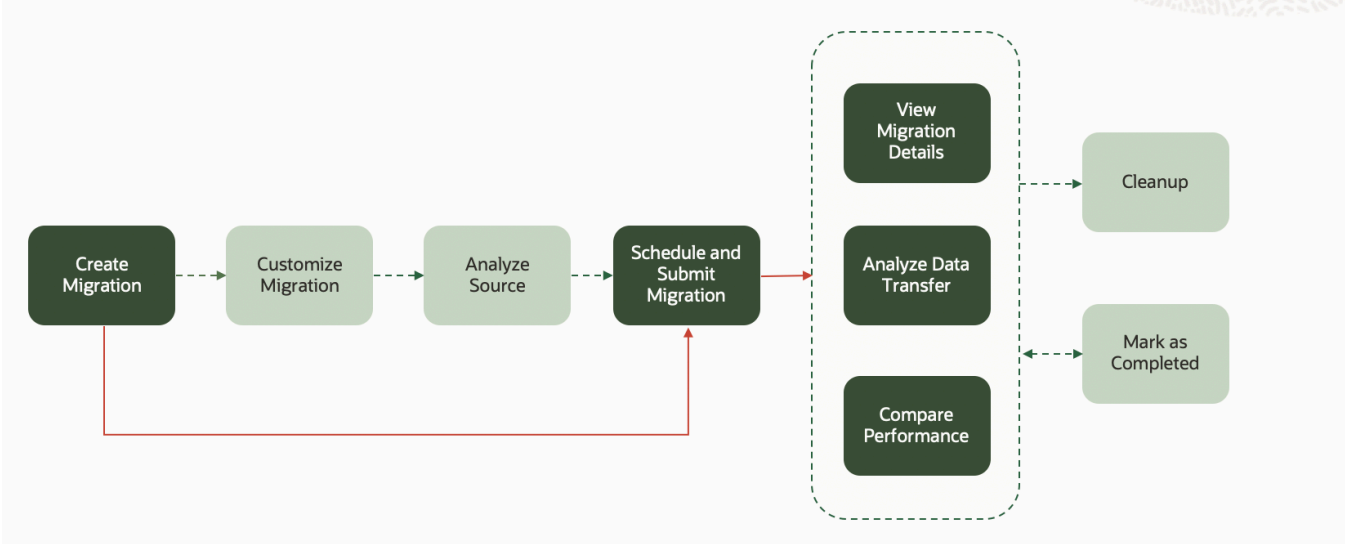
Picture below summarizes challenges in each phase of database migration



To cover various needs for the migration process as explained above, any migration automation tool should cover migrations in these three logical steps. Additionally, any cloud migration tool should be secure, efficient and agile.

## INTRODUCING DATABASE MIGRATION WORKBENCH

Database Migration Workbench is a new feature in Enterprise Manager which provides a one-stop solution for all migration needs. It integrates all the tools needed across the multiple phases of migration - Planning, Migration, and Analysis. It provides a guided workflow to drive migrations where users select the source, destination, and set migration goals. Pictorially, Database Migration Workbench follows the below approach for a typical migration to an Autonomous Database.



To provide a richer experience to existing EM users, the Database Migration Workbench enhances existing functionalities to support Oracle Cloud and Oracle Exadata Cloud at Customer. Database Migration Workbench



also keeps the migration experience consistent for all methods and destination database types. Some key features of Enterprise Manager Database Migration Workbench are:

## Unified Guided Wizard

From the Enterprise Manager console, users can access Database Migration Workbench from the Enterprise menu.

Users can use this single tool to migrate on-premises databases to Autonomous Databases, Database as a Service running on Virtual Machines, Bare Metal, Exadata Cloud at Customer, Exadata Cloud Service, or traditional database deployments on OCI Compute in the same way. It helps users select the most appropriate migration method such as Data Pump, RMAN, Advance Data Guard from the same UI, EMCLI, or REST APIs. Database Migration Workbench also helps users move or clone Databases within on-premises datacenters.

**Database Migration landscape** Migrate

Unified experience for all migration scenarios

**Database Migration**

Create Migration Activity

Name Activity  
Migrate SalesOnPremises Database to Autonomous Database Warehouse

Select Source Database Type

- ORACLE Database 11g
- ORACLE Database 12c
- ORACLE Database 18c
- 19c ORACLE Database

Select Destination Database Cloud Service Type

- ORACLE Database Cloud Service Bare Metal
- ORACLE Database Cloud Service Virtual Machines
- ORACLE Database Cloud Service Exadata Cloud Service
- ORACLE Database Cloud Service Exadata Cloud at Customer
- ORACLE Autonomous Transaction Processing Dedicated Deployments

- Autonomous Databases  
*Datapump*
- Exadata Cloud at Customer  
*Datapump, RMAN, nZDT*
- Exadata Cloud Service  
*Datapump, RMAN, nZDT*
- Databases running on VM and BM  
*Datapump, RMAN, nZDT, continuous refresh*
- Databases running on OCI Compute  
*Datapump, RMAN, nZDT, continuous refresh*
- Compute within own datacenter  
*Datapump, RMAN, nZDT, continuous refresh*

The below picture shows the landing page for Database Migration Workbench. Users can view a summary of existing migration jobs and create new migrations from this landing page.

**Database Migration**

Migration Activities 1 View Data All Auto refresh off

All **9** Completed **3** Succeeded(2) User Marked Co... In Progress **1** Problems **3** Scheduled **0** Warnings **2**

[Create Migration Activity](#)

Activity	Status	Source	Destination	Start	End	Elapsed Time	Owner
Database Migration Thu Jun 18 2020 00:33:51 PDT	<span>▶</span>	sales1.us.oracle.com_CRM	MyATPD690	Jun 18, 2020 12:49:28 AM PDT	—	3 minutes, 9 seconds	SYSMAN
Cleanup Database Migration Fri Jun 12 2020 00:13:33 PDT <span>Ⓞ</span>	<span>✖</span>	sales1.us.oracle.com_CRM	MyATPD690	Jun 12, 2020 12:13:38 AM PDT	Jun 12, 2020 12:14:20 AM PDT	42 seconds	SYSMAN
Database Migration Mon Jun 08 2020 10:43:05 PDT	<span>✖</span>	sales1.us.oracle.com_CRM	MyATPD690	Jun 12, 2020 12:00:02 AM PDT	Jun 12, 2020 12:10:19 AM PDT	10 minutes, 17 seconds	SYSMAN
Database Migration Mon Jun 01 2020 01:14:08 PDT	<span>⚠</span>	sales1.us.oracle.com_CRM	MyATPD690	Jun 1, 2020 1:18:26 AM PDT	Jun 1, 2020 1:46:52 AM PDT	28 minutes, 26 seconds	SYSMAN
Database Migration Sun May 31 2020 21:38:19 PDT	<span>⚠</span>	sales1.us.oracle.com_CRM	MyATPD690	May 31, 2020 9:42:13 PM PDT	May 31, 2020 10:21:11 PM PDT	38 minutes, 58 seconds	SYSMAN
Cleanup Database Migration Sun May 31 2020 18:55:55 PDT <span>Ⓞ</span>	<span>✔</span>	sales1.us.oracle.com_CRM	MyATPD690	May 31, 2020 6:56:02 PM PDT	May 31, 2020 6:56:39 PM PDT	37 seconds	SYSMAN
Database Migration Sun May 31 2020 18:55:55 PDT <span>Ⓞ</span>	<span>✖</span>	sales1.us.oracle.com_CRM	MvATPD690	May 31, 2020 6:05:02 PM PDT	May 31, 2020 6:41:40 PM PDT	36 minutes, 38 seconds	SYSMAN

## Sizing Advisor

The Sizing Advisor helps users estimate the size of the destination database based on the workload of the on-premises source databases. This serves two purposes. First, it helps to create ( or scale-up) the destination database to a size recommended by Sizing Advisor before migration is started. This is more important when the user is consolidating multiple Databases. Second, it also helps CTOs assess the cost to migrate database to cloud.

The destination database size is estimated by calculating historic resource usage for the source database and mapping it to resource capacity on the destination database. For these purposes, EM uses the Migration Planner internally. To estimate the size, Migration Planner

- Collects historic workload and performance metrics from the source database
- Collects capacity details from the source database
- Collects capacity details from the destination database. This is typically a specification for a particular platform like Autonomous Database or Exadata Cloud at Customer.

- Analyzes resource usage on the source database and maps it to resource capacity on the destination database

## Enhanced Schema Advisor

Schema Advisory Report helps users analyze the database schema before migration. Any sort of data type incompatibility or issues in the schema can be rectified even before the migration process is started.

The screenshot shows a web interface for the Database Migration tool. At the top, there is a header 'Database Migration'. Below it, the breadcrumb 'Migration Activities > Schema Advisor Report' is visible. The main content area displays 'Schema Advisor Report : [REDACTED]\_PDB1'. Underneath, the title 'ATPD SCHEMA MIGRATION REPORT FOR EXPUSER' is shown. A list of system parameters follows, including ADB Advisor Version (19.3), Instance Name (cdb), Database Name (CDB), Host Name ([REDACTED]), Database Version (12.1.0.2.0), Pluggable Database (PDB1), Schemas Analyzed (EXPUSER), Analyzing for (Autonomous Transaction Processing (Dedicated)), and Report date (05-DEC-2019 23:20). At the bottom, there are three expandable sections: '▶ ATPD OBJECTS NOT MIGRATED', '▶ ATPD OBJECTS MIGRATED WITH CHANGES', and '▶ ATPD MIGRATION ADDITIONAL INFO'.

As we see above Schema Advisor lists objects that cannot be migrated to a given destination database or objects that can be migrated with some changes. It also shows additional information so DBAs can make suitable changes before migration is initiated (See below).

## Database Migration

### ▲ ATPD MIGRATION ADDITIONAL INFO

1. Single-byte character set is in-use (Count=1):

Note: Character set used by the Autonomous Database is AL32UTF8 (Multi-byte). If you have a single-byte character set (e.g. US7ASCII), it will convert to multi-byte at import time by Data Pump. However, if you are using BYTE semantics for data types, some values may not fit as they now take more bytes. Ensure that you transform to character semantics for data types.  
WEBMSWIN1252

2. Columns are defined using BYTE semantics in a single-byte DB character set (Count=25198):

Note: Character set used by the Autonomous Database is AL32UTF8 (Multi-byte). But if you create a table with a column that uses BYTE semantics, it may not allow you to fit all characters as "Multi-byte" needs more bytes to store one character. Ensure that your BYTE columns are transformed to CHARACTER semantics before the migration as Oracle Data Pump currently does not do this.

```
BIN$12nkQ4+AZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ4+oZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ4/QZNPgU+ks9wrjHA==$0 (COLS=42)
BIN$12nkQ4/uZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ43wZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ45KZNPgU+ks9wrjHA==$0 (COLS=42)
BIN$12nkQ46LZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ46VZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ47HZNPgU+ks9wrjHA==$0 (COLS=42)
BIN$12nkQ48DZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ48SZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ48wZNPgU+ks9wrjHA==$0 (COLS=42)
BIN$12nkQ49OZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ49nZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ5A+ZNPgU+ks9wrjHA==$0 (COLS=42)
BIN$12nkQ5BSZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ5BcZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ5BmZNPgU+ks9wrjHA==$0 (COLS=42)
BIN$12nkQ5D3ZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ5EQZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ5EaZNPgU+ks9wrjHA==$0 (COLS=42)
BIN$12nkQ5F1ZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ5FvZNPgU+ks9wrjHA==$0 (COLS=42)  BIN$12nkQ5EaZNPgU+ks9wrjHA==$0 (COLS=42)
BIN$1rZ4BVHHSZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVHSMZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVHISMZmgU+ks9woW0w==$0 (COLS=42)
BIN$1rZ4BVKASZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVKtSMZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVMCSZmgU+ks9woW0w==$0 (COLS=42)
BIN$1rZ4BVMqSMZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVNDZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVNmSMZmgU+ks9woW0w==$0 (COLS=42)
BIN$1rZ4BVPKSMZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVPoSMZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVQ9SMZmgU+ks9woW0w==$0 (COLS=42)
BIN$1rZ4BVQBSZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVQGSZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVQSMZmgU+ks9woW0w==$0 (COLS=42)
BIN$1rZ4BVQPSZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVSrSMZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVSrSMZmgU+ks9woW0w==$0 (COLS=42)
BIN$1rZ4BVT7SMZmgU+ks9woW0w==$0 (COLS=42)  BIN$1rZ4BVTxSMZmgU+ks9woW0w==$0 (COLS=42)  EXPORT_JOB_13643285771894875 (COLS=42)
```

## Integration with Deployment Procedures

Database Migration Workbench internally uses Deployment Procedures to orchestrate a migration workflow.

Deployment procedures allow users to schedule, pause, or resume the migration process at any time. Database

Migration Workbench lets users:

- **View the Summary of Existing Migrations**

Database Migration Workbench shows the summary of existing migration jobs including completed, in-progress, failed, or scheduled jobs. Users can filter jobs based on various criteria like the status of the job, scheduled time, or completed time. It also allows users to filter based on source database and destination database service type.

- **Monitor Migrations in Real-Time**

Real-Time monitoring allows users to monitor migration jobs in real-time. At any point, the user can view completed, in-progress, and pending job steps. It also lets users predict the time required to complete the migration process.

## Procedure Activity: Database Migration - Wed Dec 04 2019

▶ Elapsed Time: 26 minutes, 26 seconds

### Procedure Steps

View	Show	All Steps
Select	Name	Status
<input type="checkbox"/>	Initialize SQL Performance Analyzer(SPA) data	✓
<input type="checkbox"/>	▶ Prepare SPA at source database	✓
<input type="checkbox"/>	Initialize SPA Procedure	✓
<input type="checkbox"/>	Pack and stage the SQL Tuning Set(STS)	✓
<input type="checkbox"/>	Initialize backup config data	✓
<input type="checkbox"/>	▶ Backup Source database	✓
<input type="checkbox"/>	▶ Backup DB or PDB using DataPump Export	✓
<input type="checkbox"/>	Initialize Export Data Procedure	✓
<input type="checkbox"/>	▶ Export Data	✓
<input type="checkbox"/>	▶ slc09esm.us.oracle.com	✓
<input type="checkbox"/>	Export data from the database	✓
<input type="checkbox"/>	Creates the export component	✓
<input type="checkbox"/>	Initialize cloud upload config data	✓

At any point, the user can view details of step by just selecting the procedure step from the deployment procedure view.

## Procedure Activity: Database Migration - Wed Dec 04 2019 19:03:35 PST

View Data Real Time: Manual Refresh

▶ Elapsed Time: 26 minutes, 26 seconds

### Procedure Steps

View	Show	All Steps
Select	Name	Status
<input type="checkbox"/>	Initialize import config data	✓
<input type="checkbox"/>	▶ Import the data from cloud storage to cloud DB	🚧
<input type="checkbox"/>	Initialize Import Data Procedure	✓
<input type="checkbox"/>	▶ Import	✓
<input type="checkbox"/>	▶ slc09esm.us.oracle.com	✓
<input type="checkbox"/>	Transfer Instant Client	✓
<input checked="" type="checkbox"/>	Import data from dump files	✓
<input checked="" type="checkbox"/>	Finalize Import Data Procedure	✗
<input type="checkbox"/>	Initialize SPA compare performance data	✓
<input type="checkbox"/>	▶ Compare performance using SPA	✓
<input type="checkbox"/>	Initialize SPA Procedure	✓
<input type="checkbox"/>	UnPack the staging table into STS and create SPA task	✓
<input type="checkbox"/>	Run SQL trials and Compare both trials	✓

Import data from dump files x Finalize Import Data Procedure x

Job Summary Download Actions

✓ Import data from dump files

object\_schema=EXPUSER

16 rows selected.

```
[INFO]: reading Object Count...
[INFO]: Parsing import job logs
[INFO]: Parsing for errors..
[INFO]: =====Parsing Completed=====
Use of uninitialized value $parsedOutput in concatenation (.) or string at /tmp/JOB_98ECD58C8BF8276CE0538052F20ADF27/ImportDb.
[INFO]: Output:
admin/*****@igdev_medium logtime=all DIRECTORY=DATA_PUMP_DIR NOLOGFILE=YES TABLE_EXISTS_ACTION=SKIP metrics=y keep_master

[INFO]: reading datapump options...
fetch Job summary
/tmp/import_data_1575515737829/instClient/instantclient_18_3/sqlplus -S admin/Welcome 12345@igdev_medium
Use of uninitialized value $parsedOutput in concatenation (.) or string at /tmp/JOB_98ECD58C8BF8276CE0538052F20ADF27/ImportDb.
[INFO]: Output:
state=COMPLETED total_bytes=1229591824 name=IMPORT_JOB_1084869

[INFO]: reading jobSummary options...
Use of uninitialized value $parsedOutput in concatenation (.) or string at /tmp/JOB_98ECD58C8BF8276CE0538052F20ADF27/ImportDb.
[INFO]: Output:
completed_bytes=
completed_bytes=1246187520 file_name=https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/
file_name=cdb.us.oracle.com_PDB1_01_4410314213307533.dmp

[INFO]: reading Dump Files...
Use of uninitialized value $parsedOutput in concatenation (.) or string at /tmp/JOB_98ECD58C8BF8276CE0538052F20ADF27/ImportDb.
```

- **Schedule Migrations**

The Scheduling feature allows users to schedule migration workflows for a future time. As soon as a user sets a future time to initiate a migration workflow, a new migration activity is scheduled in Enterprise Manager. It is best practice to complete all migrations at a time when there is a least load on the system. The scheduling feature is very useful for users who want to initiate migration at the least peak time and let EM run it automatically.

- **Pause and Resume Migrations**

Users can pause a migration workflow, to proactively diagnose issues during the migration, if any. Once the issue is rectified users can resume the workflow. For example, if users notice that the destination database is down, they can pause the migration workflow, bring up the destination database, and then resume migration workflow.

- **Rollback Migrations**

At any point, users can rollback the migration process. Rollback takes the destination Database to a state just before when migration started. If for some reason the migration process is not completed successfully, it may be easier to start the entire process again from scratch. Moreover, the rollback feature also cleans up intermediate resource usage.

- **Customize Migration Workflow**

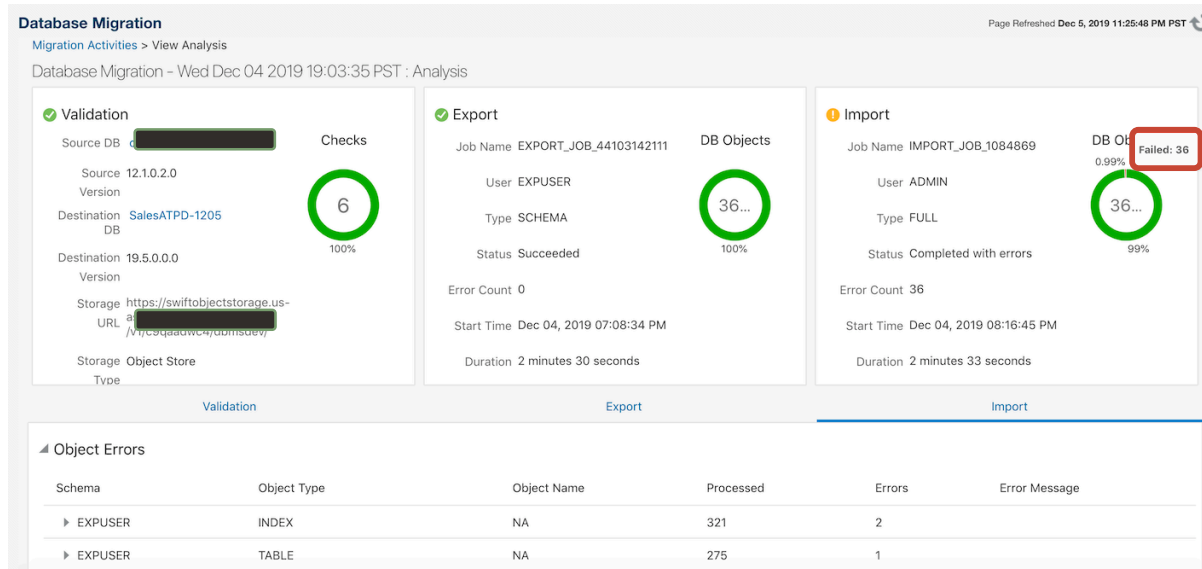
If users want to override certain default values in the out-of-box migration process or add additional checks, they can extend the existing out-of-box deployment procedure using Pre and Post scripts

- **Pre- and Post-Scripts**

Database Migration Workbench allows users to add pre and postscripts. Prescripts are run on the source database before starting migrations and postscripts are run on the destination database after the migration is completed.

## Analyze Migrations

Post Migration Analyzer helps the user assess the success of the migration. Database Migration Workbench also provides an analysis report at the end of the migration. The analysis report provides a deep analysis of what objects are migrated successfully, what objects have failed, and the reason for failure. It also shows the stage at which failure occurred like whether failure occurred at export or import.



DBAs can drill down into a particular error and find out the root cause of the error. Based on the root cause they can decide whether the error can be ignored or to take remedial action.

## Compare Performance after Migration

Post Migration Performance Analysis helps the user compare the performance of the database before migration and understand the impact on the performance after migration. This analysis is based on SQL Performance

Analyzer (SPA). SQL Performance Analyzer automatically collects SQL statements from the Database cache and creates a set of SQL queries called SQL Tuning Set (STS).

- **Production Class STS Collections**

A best practice for Database migration is to enable SQL Performance Analyzer (SPA) to collect SQL statements from the Database cache on an on-going basis. This allows the SPA to record each statement and related performance parameters. If the same SQL statement is run again and again each time only performance metrics for the statement are updated. This method is best suited for production workloads to accurately build a SQL Tuning Set (STS). STS built using this method accurately represents a SQL statement with a very high workload. Database Migration Workbench allows users to select the best STS created by this process. However, it may not be practically possible to enable SQL Performance Analyzer and wait for few weeks before initiating Database migration to build accurate STS so Database Migration Workbench also provides an option of DevOps Class STS Collection.

- **DevOps Class STS Collections**

To overcome the prerequisite of enabling collections before the migration process starts, Database Migration Workbench also provides an alternative that allows users to build STS on-the-fly from AWR reports. Users can choose the number of days for which statements from AWR reports should be considered to build STS. When this option is selected, Database Migration Workbench scans through AWR reports for SQL statements and builds unique STS. This method is simple and quick and can be used for migrations involving databases for dev/test purposes.

In either case, once STS is ready tests are run on the source database and all performance metrics are recorded. During the migration process, STS is migrated to destination Databases. After the migration process is complete, tests are again run on destination Databases and performance metrics are collected. Once performance metrics are collected from source and destination Databases, a comparative report is shown to the user. Based on a



comparison of metrics EM shows overall impact after migrating to Oracle Cloud. An example of such a report generated by Database Migration Workbench is shown below.

**Database Migration**

**Report Summary**

**Projected Workload Change Impact:**

Overall Impact : 24.27%  
 Improvement Impact : 24.27%  
 Regression Impact : 0%

**SQL Statement Count**

SQL Category	SQL Count	Plan Change Count
Overall	129	26
Improved	6	4
Unchanged	27	22
with Errors	96	0

**Top 33 SQL Sorted by Absolute Value of Change Impact on the Workload**

object_id	sql_id	Impact on Workload	Execution Frequency	Metric Before	Metric After	Impact on SQL	Plan Change
202	4d9pwa34kms6	11.76%	253	373086.205533597	153833	58.77%	y
218	8p01x4csmn6	3.97%	20	937891.95	36	100%	y
200	8arwq2femv6	3.07%	7	2067070.14285714	191	99.99%	y
211	7p9y2m9g7c126	2.21%	1	10506906	95783	99.09%	n
154	2gpx0879aumd0	2.11%	1	10068880	98430	99.02%	n
158	22gq01n8ym0	1.16%	233	26298.9957081545	2854	89.15%	y

### Integrate with Other Tools

Many organizations today develop in-house solutions to manage and maintain their multi-vendor environment.

To integrate any new component or solution they rely on REST APIs for easy integration.

All features in Database Migration Workbench can be accessed via all three interfaces – UI, EMCLIs, or REST APIs for easy integration with 3<sup>rd</sup> party interfaces or use within scripts.

Moreover, Database Migration Workbench also helps users create and copy a pre-filled JSON to integrate with other tools.

## SUMMARY

Oracle Enterprise Manager – Database Migration Workbench helps users monitor and manage migrations to the Oracle Cloud easily. It helps users across all three stages of the migration process – Planning, Migration, and Analysis. It provides an end to end solutions to support a wide variety of migration scenarios, including non-container to PDB migrations, cross-platform migrations, character set conversions, etc. Besides, the Database Migration Workbench’s guided migration flows help determine the size of the destination Database, identify the best migration scenario, measure the success of the migration, and ensure the migrated target meets performance expectations. In summary, Database Migration Workbench,

- **Simplifies** the migration of on-premises databases to oracle cloud infrastructure with unified guided workflow,
- **Expedites** the migration process by automating end-to-end workflow,
- **Reduces downtime** by providing the real-time progress of the migration, and
- **Improves the accuracy** of migrations by providing an automatic comparison of performance before and after migration.

## CONNECT WITH US

Call +1.800.ORACLE1 or visit [oracle.com](http://oracle.com).  
Outside North America, find your local office at [oracle.com/contact](http://oracle.com/contact).

 [blogs.oracle.com](http://blogs.oracle.com)

 [facebook.com/oracle](https://facebook.com/oracle)

 [twitter.com/oracle](https://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

White Paper Oracle Enterprise Manager: Database Migration Workbench  
November, 2020  
Authors: Akshay Sangaonkar, Martin Peña

