



EDB

Migration Portal Guide

Release 3.0.0

Migration Portal Guide

Dec 18, 2020

Contents

1	What's New	2
2	Supported Versions	6
3	Using Migration Portal	7
3.1	Overview of the Migration Portal Home Page	10
3.2	Overview of the Migration Portal Projects Page	12
3.3	Overview of the Migration Portal Wiki Page	15
4	Migrating a Database	16
4.1	Performing a Schema Extraction	17
4.1.1	Supported Object Types	20
4.1.2	Unsupported Object Types	20
4.1.3	Oracle System Schemas	21
4.2	Performing a Schema Assessment	22
4.2.1	Generating an Assessment Report	27
4.3	Schema Migration	29
4.3.1	Migrating to an Existing On-Premise EDB Postgres Advanced Server Host	30
	Migrating Schemas on Windows	30
	Migrating Schemas on Linux	34
4.3.2	Migrating to a New On-Premise EDB Postgres Advanced Server Instal-	
	lation	38
	Migrating Schemas on Windows	38
	Migrating Schemas on Linux	44
4.3.3	Migrating to the Cloud	50
4.4	Data Migration	56
5	Advanced Data Migration	57
6	Conclusion	58

Migration Portal is a web-based service for migrating Oracle database schemas to the EDB Postgres™ platform. The Migration Portal assesses and analyzes Oracle database schemas and converts types, tables, sequences, constraints, triggers, views, stored procedures, packages, dblinks, materialized views, and indexes, producing DDLs that are compatible with EDB Postgres Advanced Server.

The user-friendly portal interface simplifies assessment and migration process; log on to the portal and start the process.

The Migration Portal guide provides a high-level description of the steps involved in the migration process. The guide also includes solutions to common migration problems and details of unsupported features and their potential workarounds.

EnterpriseDB has helped companies migrate their existing database systems to Postgres for years. For more information, visit the EnterpriseDB website at:

<https://www.edbpostgres.com/>

CHAPTER 1

What's New

The following enhancements are added to the Migration Portal for this release:

- Assessment for EDB Postgres Advanced Server version 13, which has additional Oracle compatibility improvements and new Postgres features.
- In addition to IBM Cloud, you can now migrate your Oracle schemas to EDB Postgres Advanced Server installed on the following cloud platforms:
 - Amazon AWS
 - Google cloud
 - Microsoft Azure
- We added a contact form so you can reach out to us about migration services if you need an offline assessment of your schemas.
- With this release, the DDL extractor script requires fewer privileges. It now only requires `CONNECT` and `SELECT_CATALOG_ROLE` role and `CREATE TABLE` privilege.

Repair Handler

The following repair handlers convert Oracle syntax to Postgres-compatible syntax so that you don't have to do it manually.

- ERH 1013 - `PACKAGE_BODY_PROC/FUNC_DECLARATION`
Removes `PROCEDURE/FUNCTION` declaration from the `PACKAGE BODY` definition.
For example,

```
CREATE OR REPLACE PACKAGE BODY package_name
IS
    PROCEDURE sample_proc;

    var1 number:=0;

    FUNCTION Func_name RETURN number;

    PROCEDURE proc_name
    IS
    BEGIN
        NULL;
    END;

    PROCEDURE sample_proc
    IS
    BEGIN
        NULL;
    END;

    FUNCTION Func_name RETURN number
    IS
    BEGIN
        return 1;
    END;
END;
```

would become;

```
CREATE OR REPLACE PACKAGE BODY package_name
IS

    var1 number:=0;

    PROCEDURE proc_name
    IS
    BEGIN
        NULL;
    END;

    PROCEDURE sample_proc
    IS
    BEGIN
        NULL;
    END;
```

(continues on next page)

(continued from previous page)

```
FUNCTION Func_name RETURN number
IS
BEGIN
    return 1;
END;
END ``
```

- **ERH 2091- CREATE_TABLE_CREATE_INDEX**

Removes schema-name from the name of the INDEX in CREATE INDEX definition.

For example,

```
CREATE TABLE "HR"."DEPARTMENTS"
(
    "DEPARTMENT_ID" NUMBER(4,0),
    "DEPARTMENT_NAME" VARCHAR2(30) CONSTRAINT "DEPT_NAME_NN" NOT_
↪NULL ENABLE,
    "MANAGER_ID" NUMBER(6,0),
    "LOCATION_ID" NUMBER(4,0)
) ;
CREATE UNIQUE INDEX "HR"."DEPT_ID_PK" ON "HR"."DEPARTMENTS" (
↪"DEPARTMENT_ID");
```

would become;

```
CREATE TABLE "HR"."DEPARTMENTS"
(
    "DEPARTMENT_ID" NUMBER(4,0),
    "DEPARTMENT_NAME" VARCHAR2(30) CONSTRAINT "DEPT_NAME_NN" NOT_
↪NULL ENABLE,
    "MANAGER_ID" NUMBER(6,0),
    "LOCATION_ID" NUMBER(4,0)
) ;
CREATE UNIQUE INDEX "DEPT_ID_PK" ON "HR"."DEPARTMENTS" (
↪"DEPARTMENT_ID");
```

- **ERH 2092- ALTER_TABLE_ADD_PRIMARY_KEY**

Transforms Oracle's ALTER TABLE ADD CONSTRAINT for PRIMARY KEY syntax to Advanced Server compatible syntax.

For example,

```
CREATE TABLE "HR"."DEPARTMENTS"
(
    "DEPARTMENT_ID" NUMBER(4,0),
    "DEPARTMENT_NAME" VARCHAR2(30) CONSTRAINT "DEPT_NAME_NN" NOT_
↪NULL ENABLE,
    "MANAGER_ID" NUMBER(6,0),
```

(continues on next page)

(continued from previous page)

```
        "LOCATION_ID" NUMBER(4,0)
    ) ;
CREATE UNIQUE INDEX "HR"."DEPT_ID_PK" ON "HR"."DEPARTMENTS" (
    ↳"DEPARTMENT_ID");
ALTER TABLE "HR"."DEPARTMENTS" ADD CONSTRAINT "DEPT_ID_PK" PRIMARY_
    ↳KEY ("DEPARTMENT_ID")
    USING INDEX "HR"."DEPT_ID_PK" ENABLE;
```

would become;

```
CREATE TABLE "HR"."DEPARTMENTS"
(
    "DEPARTMENT_ID" NUMBER(4,0),
    "DEPARTMENT_NAME" VARCHAR2(30) CONSTRAINT "DEPT_NAME_NN" NOT_
    ↳NULL ENABLE,
    "MANAGER_ID" NUMBER(6,0),
    "LOCATION_ID" NUMBER(4,0)
) ;
CREATE UNIQUE INDEX "DEPT_ID_PK" ON "HR"."DEPARTMENTS" (
    ↳"DEPARTMENT_ID");
ALTER TABLE "HR"."DEPARTMENTS" ADD PRIMARY KEY
    USING INDEX "DEPT_ID_PK";
```

CHAPTER 2

Supported Versions

The Migration Portal supports assessment and migration from Oracle 11g, 12c, 18c, and 19c to EDB Postgres Advanced Server 10, 11, 12, or 13. Migration Portal is supported on the following browsers and operating systems:

Supported Browsers

For the best user experience, we recommend using the Google Chrome browser. Migration Portal is also supported on the following browsers:

Browser	Supported Version
Apple Safari on Macintosh OS	11 and above
Google Chrome	68 and above
Microsoft Edge	42 and above
Mozilla Firefox	60 and above

Supported Operating Systems

Operating Systems	Supported Version
Macintosh	OS X Sierra
Windows	10
Linux	CentOS 7

CHAPTER 3

Using Migration Portal

Migration Portal allows you to easily migrate your database schema from Oracle to EDB Postgres Advanced Server. You can upload schemas for assessment and get immediate feedback and suggestions. The portal allows you to download assessed DDLs for all objects and create your EDB Postgres™ database on-premises or in the cloud.

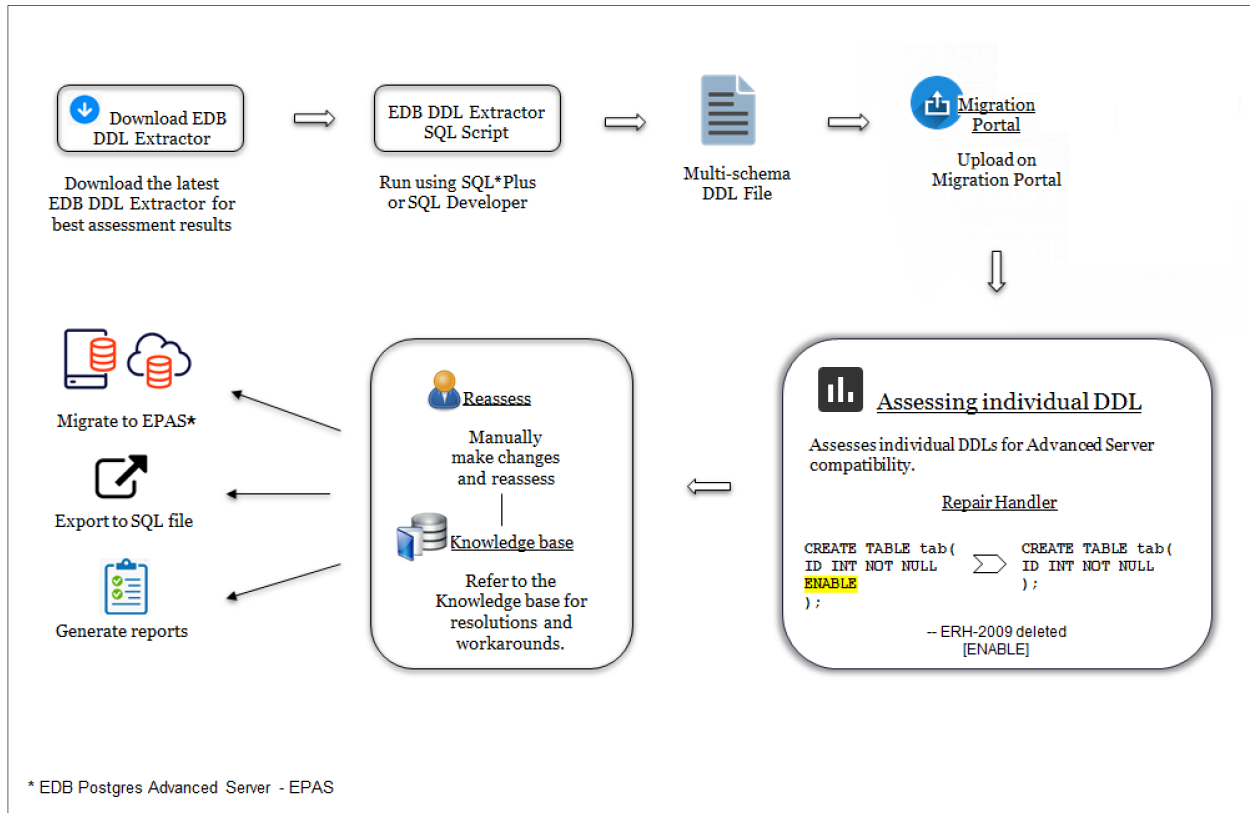


Fig. 1: The assessment and migration process

To access the Migration Portal:

1. Open a browser and navigate to <https://www.edbpostgres.com/>
2. On the EnterpriseDB home page, click Products > Migration Portal.

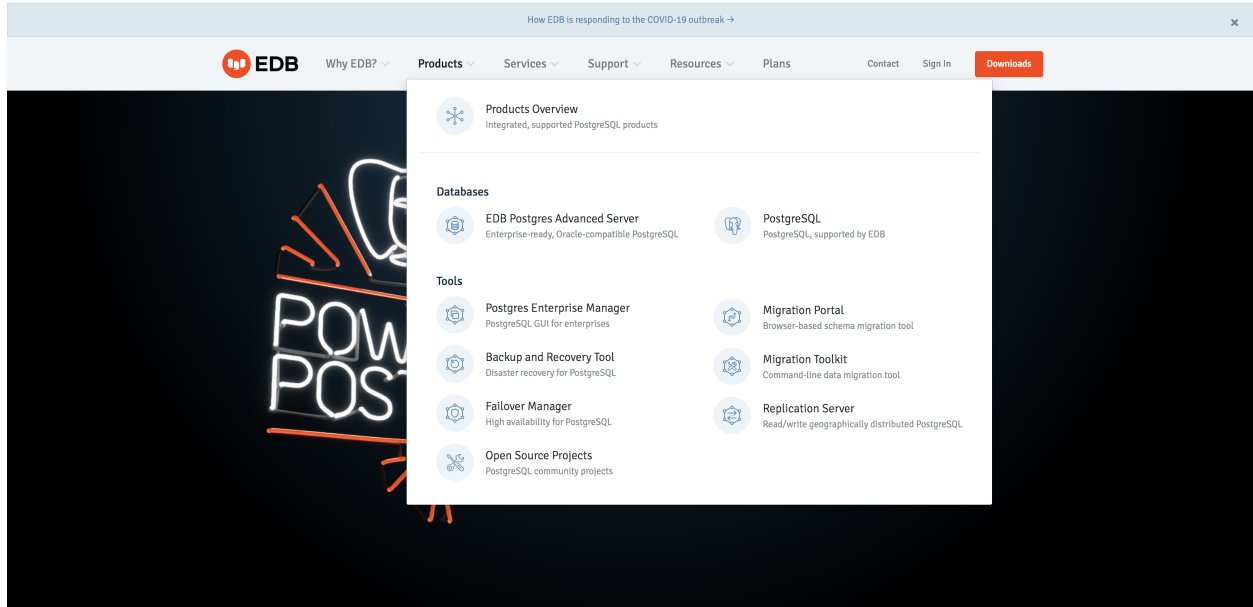


Fig. 2: *Accessing the Migration Portal.*

3. Click Open Migration Portal.
4. Log in using your credentials.

3.1 Overview of the Migration Portal Home Page

The Migration Portal home page provides quick access to migration tools.

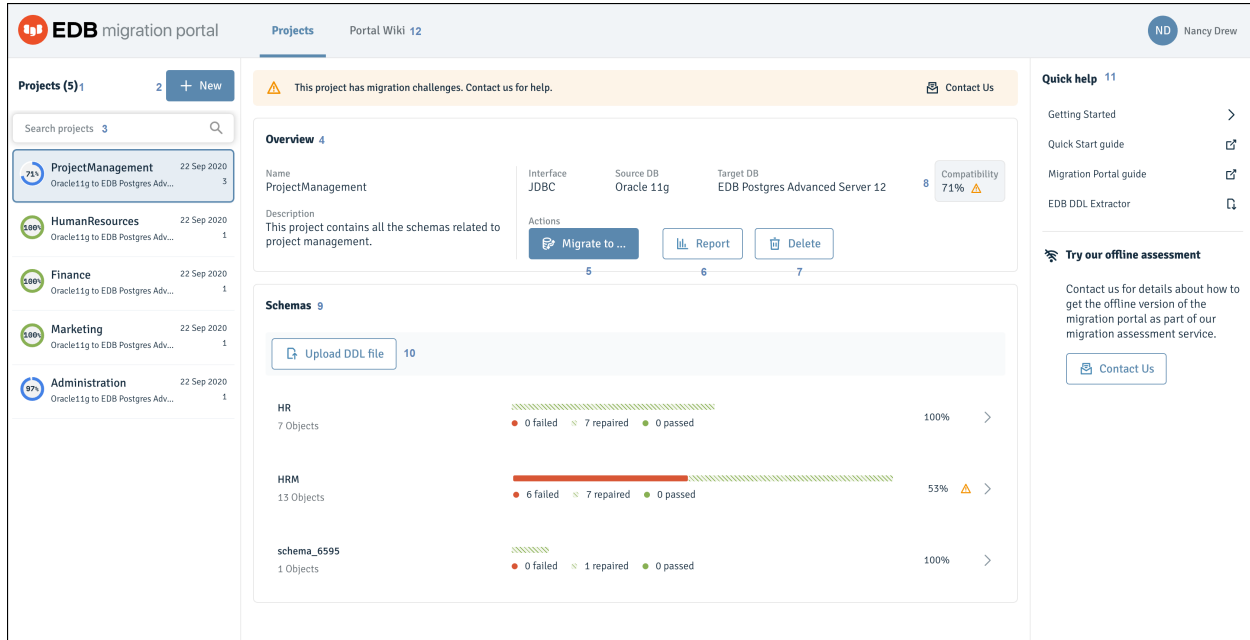


Fig. 3: The Migration Portal home page.

The Migration Portal home page allows access to the following Migration Portal features:

1. **Projects:** The `Projects` panel displays a list of assessed projects.
2. **Create project:** Click `+ New` (the button located to the right of the `Projects` label) to create a new project.
3. **Search Projects:** Use the `Search` box to search for projects.
4. **Overview:** The `Overview` panel provides details about the selected project and displays the compatibility percentage after schema assessment.
4. **Migrate:** Use the `Migrate` button to either download an EDB Postgres Advanced Server compatible `.sql` file or to migrate a schema to an EDB Postgres Advanced Server on-premise or on cloud. For more information, see [Schema Migration](#).
5. **Report:** Use the `Report` button to view and download the schema assessment report.
6. **Delete:** Use the `Delete` button to delete a selected project.
7. **Warning sign:** A warning message is displayed if a project or a schema is less than 70% compatible or any DDL doesn't succeed after multiple attempts.
8. **Upload DDL file:** Use the `Upload new schema` button to upload a new or additional DDL file.

9. **Schemas:** The `Schemas` panel displays the assessment result from an uploaded DDL file.
10. **Quick help:** The `Quick help` panel contains all the help guides.
11. **Portal Wiki:** The `Portal Wiki` has links to product information and different help guides.

3.2 Overview of the Migration Portal Projects Page

The Migration Portal Projects page provides detailed information about your migration.

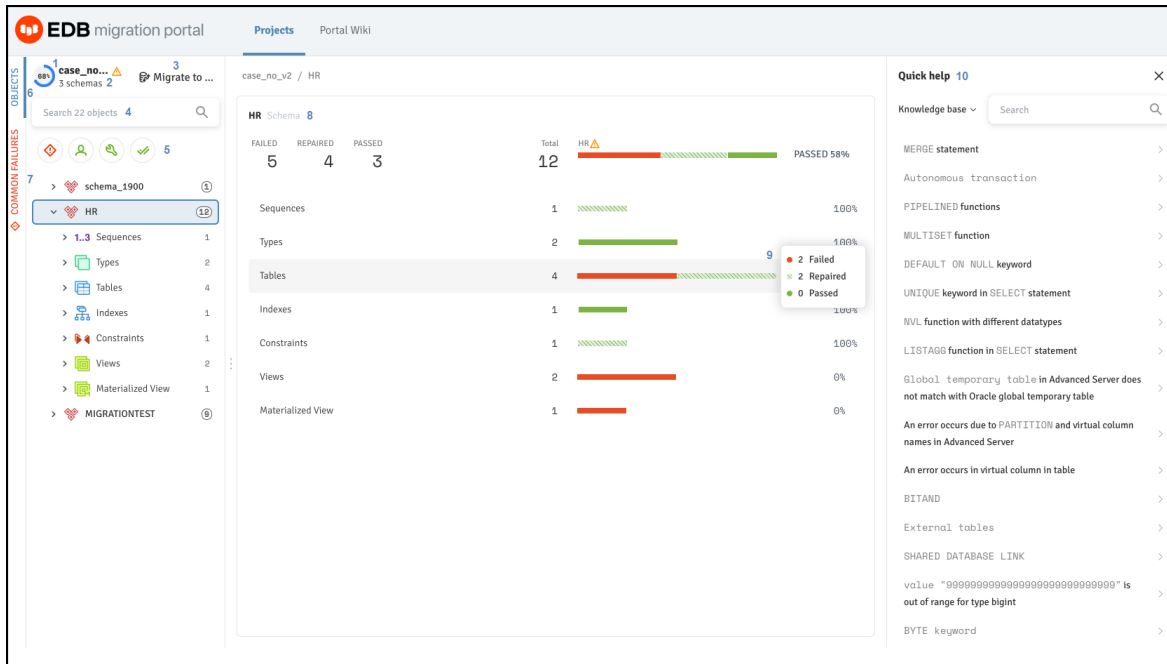


Fig. 4: The Migration Portal Projects Page overview.

Use the following resources to gather information about your migration projects:

1. **Compatible:** The `Compatible` gauge displays the color on the basis of the compatibility percentage of the assessed schema.
2. **Schema Count:** Displays the number of schemas in a project.
3. **Migrate to:** Use the `Migrate to` to migrate the schema to an *EDB Postgres Advanced Server* on-premise or on cloud.
4. **Search objects:** Use the `Search` box to search for objects.
5. **Filters:** You can filter the system repaired and manual repaired objects from the left panel of the Projects page. In addition, you can select one or more filter combinations to refine the information.
6. **Objects:** Displays the objects for the selected schemas.
7. **Common Failures:** Displays the reason for the failed objects for the selected schemas.

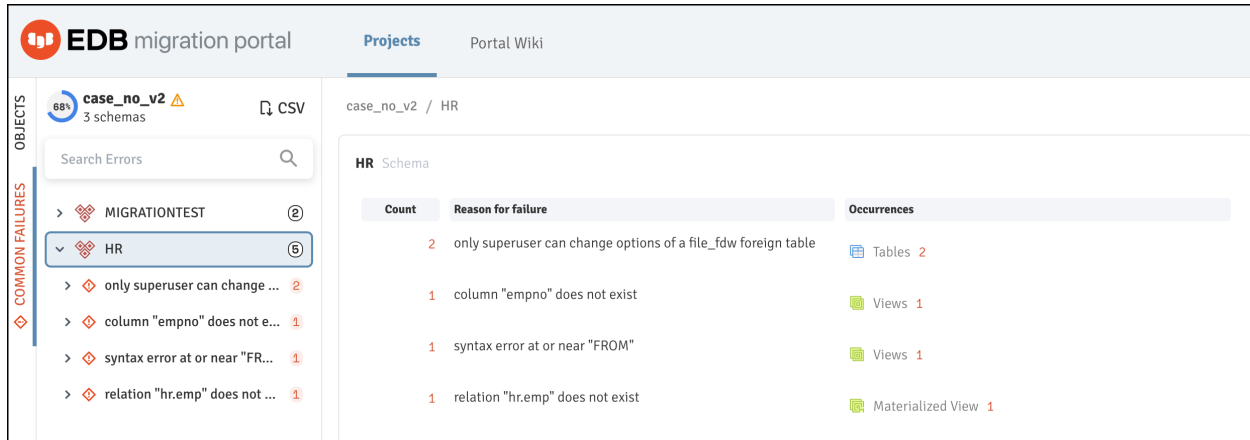


Fig. 5: Common Failures tab

Note: You can download a CSV file for the common failures for the project.

8. **Schema:** The Schema panel displays the result of the assessment.
9. **Tooltip:** Hover over a result set to display a tooltip with the number of passed, failed, and repaired objects.
10. **Quick help:** The Quick help panel displays links to Knowledge base articles and repair handler documentation.
11. **Search:** Use the Search box to search the Knowledge base entries or repair handler documentation for specific information.

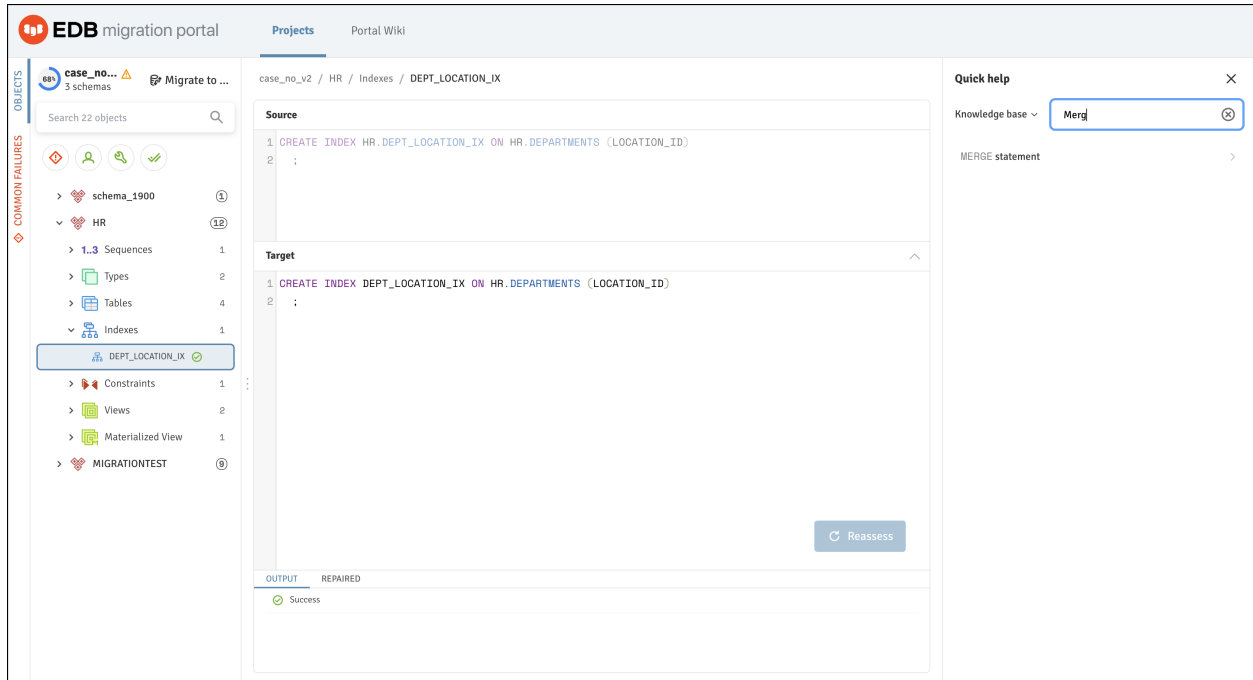


Fig. 6: Searching the Knowledge Base entry.

3.3 Overview of the Migration Portal Wiki Page

The Portal Wiki page provides quick access to information:

- What's New information
- Quick Start guide
- Migrating Schema
- Migrating Data
- DDL Extractor guide
- Knowledge Base
- Repair handlers
- Migration Portal User's Guide
- FAQs

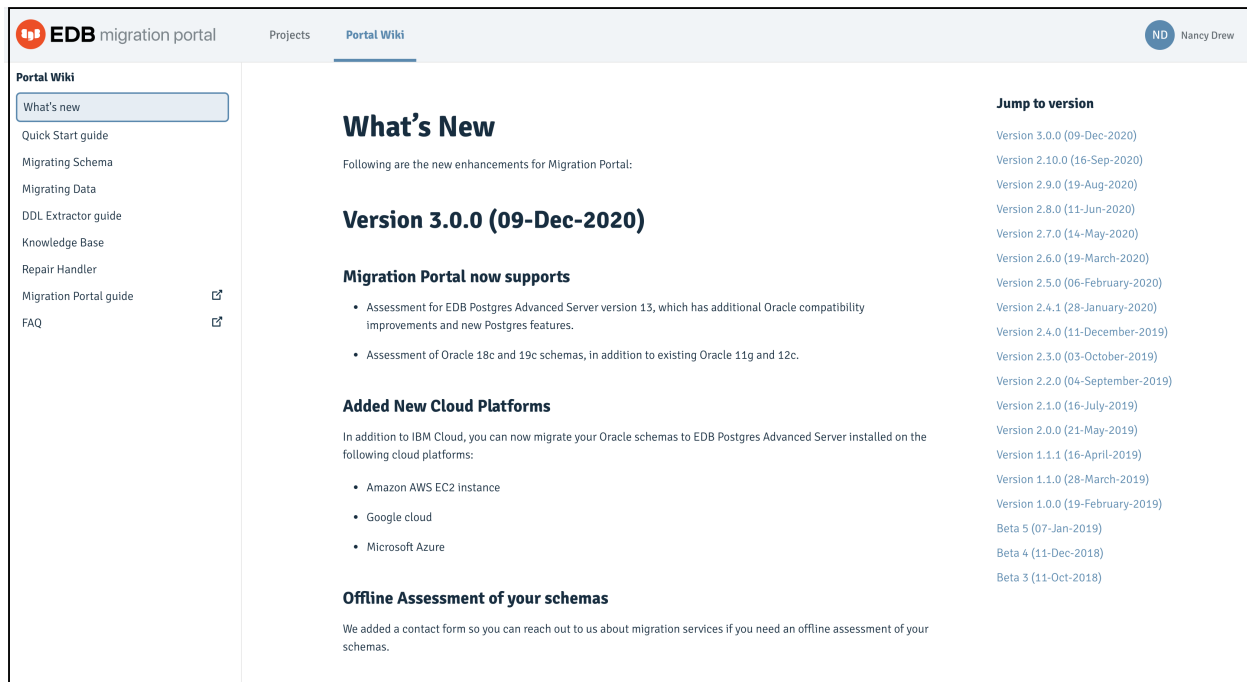


Fig. 7: The Migration Portal Wiki page.

CHAPTER 4

Migrating a Database

To migrate a database, you must complete the following steps:

1. Perform a *Schema Extraction*.
2. Perform a *Schema Assessment*.
3. Perform a *Schema Migration*.
4. Perform a *Data Migration*.

The following sections provide detailed information about each step in the migration process.

4.1 Performing a Schema Extraction

Prerequisites

Before extracting a schema, you must download the latest EDB DDL Extractor script from the Migration Portal `Projects` page or from the link provided in the DDL Extractor guide in the Portal Wiki. The script can be run in SQL Developer or SQL*Plus. It uses Oracle's `DBMS_METADATA` built-in package to extract DDLs for different objects under schemas (specified while running the script). The EDB DDL extractor creates the DDL file that will be uploaded to the portal and analyzed for EDB Postgres compatibility.

Note:

- With this release, the DDL extractor script requires fewer privileges. It now only requires *CONNECT* and *SELECT_CATALOG_ROLE* role and *CREATE TABLE* privilege.

For SQL*Plus

1. Connect to SQL*Plus and run the command:

```
SQL>@edb_ddl_extractor.sql
```

2. Provide the schema name and the path or directory in which the extractor will store the extracted DDL. When extracting multiple schemas, use a comma (',') as a delimiter.

Note: If you want to extract all the user schemas from the current database, do not mention any schema names while extracting. However, it is recommended to mention the schema names that you would like to extract.

3. If you want to extract dependent objects from other schemas, enter yes or no.

For example, on Linux:

```
Enter a comma separated list of schemas to be extracted
(Default all schemas):  HR, SCOTT, FINANCE          Location for
output file (Default current location) :  /home/oracle/
extracted_ddls/
```

WARNING:

Given schema(s) list may contain objects which are dependent on objects from other schema(s), not mentioned in the list. Assessment may fail for such objects. It is suggested to extract all dependent objects together.

```
Extract dependent object from other schemas?(yes/no) (Default
no / Ignored for all schemas option):  yes
```

On Windows:

Enter comma separated list of schemas to be extracted
(Default all schemas): HR, SCOTT, FINANCE Location for
output file (Default current location) : c:\Users\Example\
Desktop\

WARNING:

Given schema(s) list may contain objects which are dependent
on objects from other schema(s), not mentioned in the list.
Assessment may fail for such objects. It is suggested to
extract all dependent objects together.

Extract dependent object from other schemas?(yes/no) (Default
no / Ignored for all schemas option): yes

For SQL Developer

1. Connect to the SQL server and run the following command:

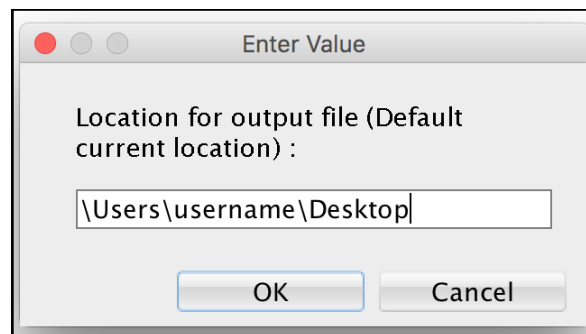


Fig. 1: Enter the path for Linux or Windows.

2. Enter a comma-separated list of schemas:

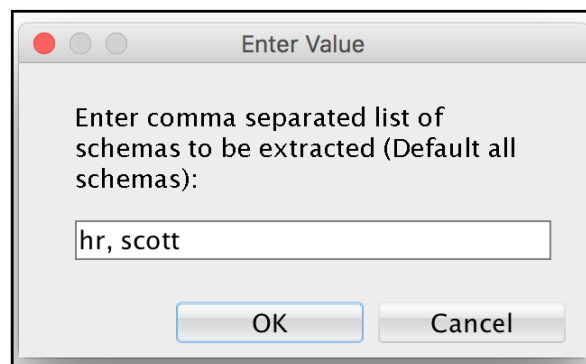


Fig. 2: Provide a list of schemas.

3. Enter the file path for the output file:

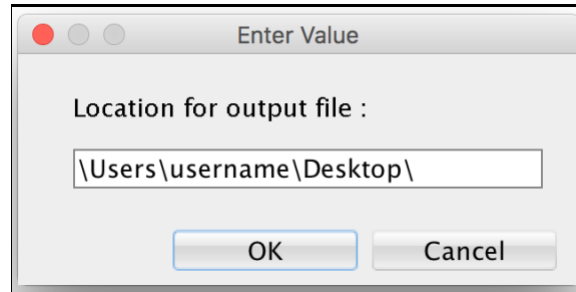


Fig. 3: *Specify the output file path.*

4. Enter (yes/no) to extract dependant objects:

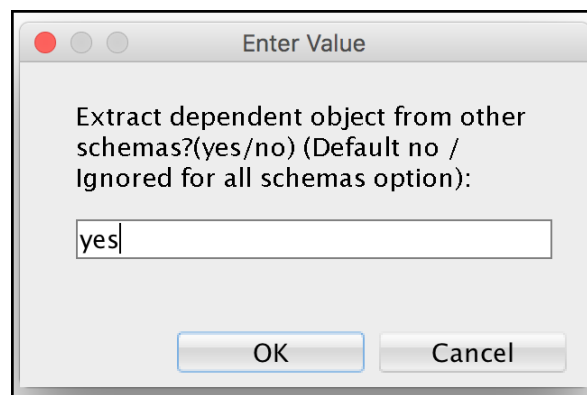


Fig. 4: *Extracting dependent objects.*

Note: You can also enter single schema name in both SQL*Plus and SQL Developer.

The script then iterates through the object types in the source database and once the task is completed, the .SQL output is stored at the entered location, i.e., c:\Users\Example\Desktop\.

Additional Notes

- The EDB DDL Extractor script does not extract objects that were restored using Flashback and still have names like BIN\$b54+4XlEYwPgUAB/AQBWwA==\$0. If you want to extract these objects, you must change the name of the objects and re-run the extraction process.
- DDL Extractor extracts `nologging` tables as normal tables. Once these tables are migrated to EDB Postgres Advanced Server, WAL log files will be created.
- DDL Extractor creates Global Temporary tables to store the schema names and their dependency information. These tables are dropped at the end of successful extraction.
- DDL Extractor script does not extract schemas whose name starts with `PG_` because PostgreSQL does not support it. If you want to extract these schemas, you must change name of schema before extraction.

4.1.1 Supported Object Types

The Migration Portal supports the migration of the following object types:

- Synonyms
- DB Links
- Types and Type Body
- Sequences
- Tables
- Constraints
- Indexes (Except LOB indexes and indexes on materialized views)
- Views
- Materialized Views
- Triggers
- Functions
- Procedures
- Packages

Note: COMMENTS on Columns, Tables, and Materialized Views are also supported.

4.1.2 Unsupported Object Types

- Editions
- Operators
- Schedulers
- LOB indexes and Indexes on Materialized Views
- XML Schemas
- Profiles
- Role and Object Grants
- Tablespaces
- Directories
- Users

- RLS Policy
- Queues

4.1.3 Oracle System Schemas

EDB DDL Extractor script will ignore the following system schemas while extracting from Oracle:

ANONYMOUS	APEX_PUBLIC_USER	APEX_030200
APEX_040000	APEX_040000	APPQOSSYS
AUDSYS	BI	CTXSYS
DMSYS	DBSNMP	DIP
DVF	DVSY	EXFSYS
FLows_FILES	FLows_020100	GSMAD-MIN_INTERNAL
GSMCATUSER	GSMUSER	IX
LBACSYS	MDDATA	MDSYS
MGMT_VIEW	OE	OJVMSYS
OLAPSYS	ORDPLUGINS	ORDSYS
ORDDATA	OUTLN	ORACLE_OCM
OWBSYS	OWBYSS_AUDIT	PM
RMAN	SH	SI_INFORMTN_SCHEMA
SPA-TIAL_CSW_ADMIN_USR	SPA-TIAL_WFS_ADMIN_USR	SYS
SYSBACKUP	SYSDG	SYSKM
SYSTEM SYSMAN	TSMSYS WKPROXY	WKSYS
WK_TEST	WMSYS	XDB
XS\$NULL		

4.2 Performing a Schema Assessment

To assess an Oracle database schema for compatibility with EDB Postgres Advanced Server, you must:

1. Navigate to the [Migration Portal](#).
2. Enter your EDB credentials.
3. Click the `CREATE PROJECT` icon to create a new project.

New project [X]

Project name

Application interface
☒ JDBC ☐ ODBC ☐ .NET ☐ OCI ☐ ProC ☐ Other

Source DB: Version:

Target DB: Version:

DDL file [\(How to get a DDL file?\)](#)
 [Choose file](#)

☐ Add Index Prefix
 Index objects will be prefixed with "idx" as unique names are required for tables and indexes.

Description(Optional)

 200 characters left

[X Cancel](#) [✓ Create & assess](#)

Fig. 5: The Migration Portal New project dialog.

4. On the New project dialog, enter the project name in the Project name field.
5. Specify project details:
 - Select the radio button next to the Application interface.
 - Select a Source DB and Version for the source.
 - Select the Target DB and Version for the target.
6. Click Choose file to upload the .SQL file generated by the latest EDB DDL Extractor for Oracle Database.

Note:

- You should not modify the .SQL file.
- Only the .SQL file generated by the latest EDB Postgres Advanced Server does not support the same name for tables and indexes.

For more information, refer to the *Schema Extraction* section.

7. Check the box next to Add Index Prefix to specify an index prefix (idx) when creating a project to ensure better assessment results, as EDB Postgres Advanced Server does not support the same name for tables and indexes.
8. Click Create & assess to create a new project and to assess the schemas.

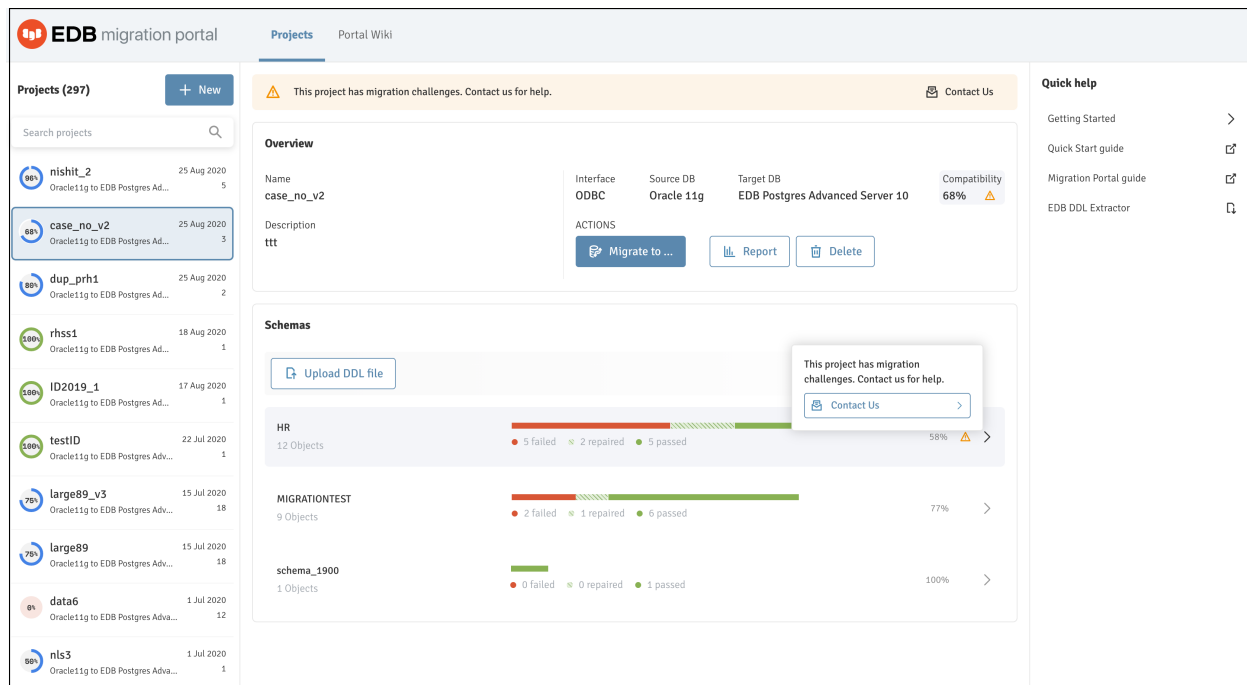


Fig. 6: The Schema analysis result.

The analysis tool will review every construct and execute repair actions to improve compatibility with EDB Postgres Advanced Server, and flag any remaining errors that require manual intervention.

9. Verify the DDL objects (e.g., TABLES) that do not show a 100% success ratio.

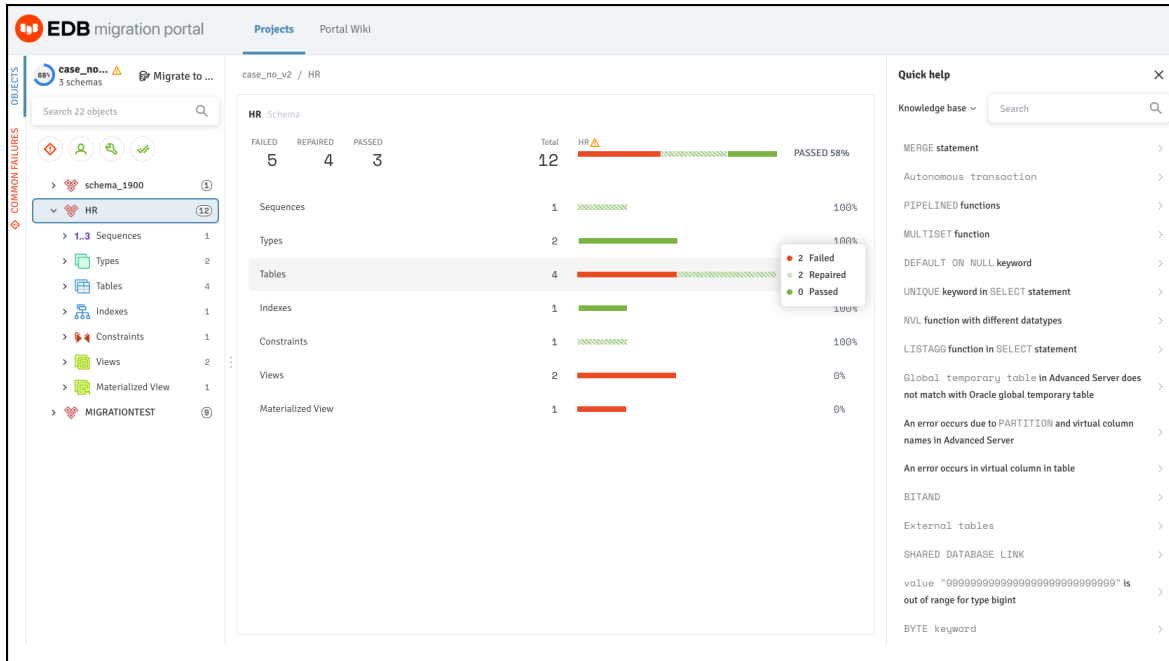


Fig. 7: Verifying the DDL objects.

- Click the objects that are not compatible with EDB Postgres Advanced Server and view the details. At the bottom of the middle panel, you can view repair action details.

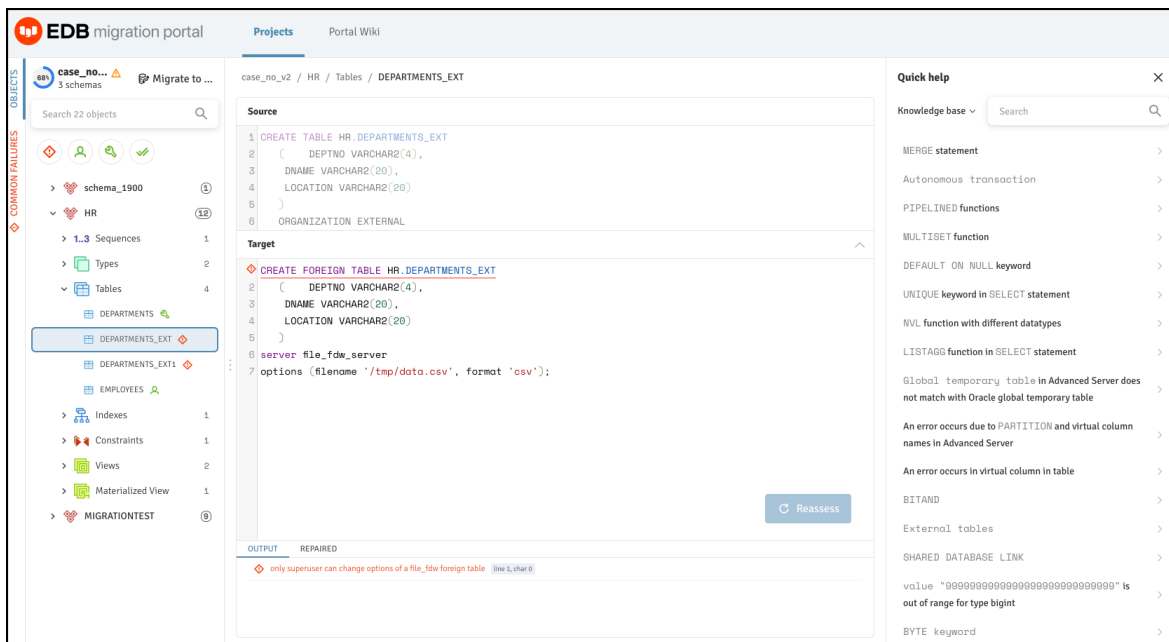


Fig. 8: Incompatible objects are identified.

- Refer to the Knowledge Base information in the right panel to locate the possible workarounds for the objects that are not immediately compatible with EDB Postgres Ad-

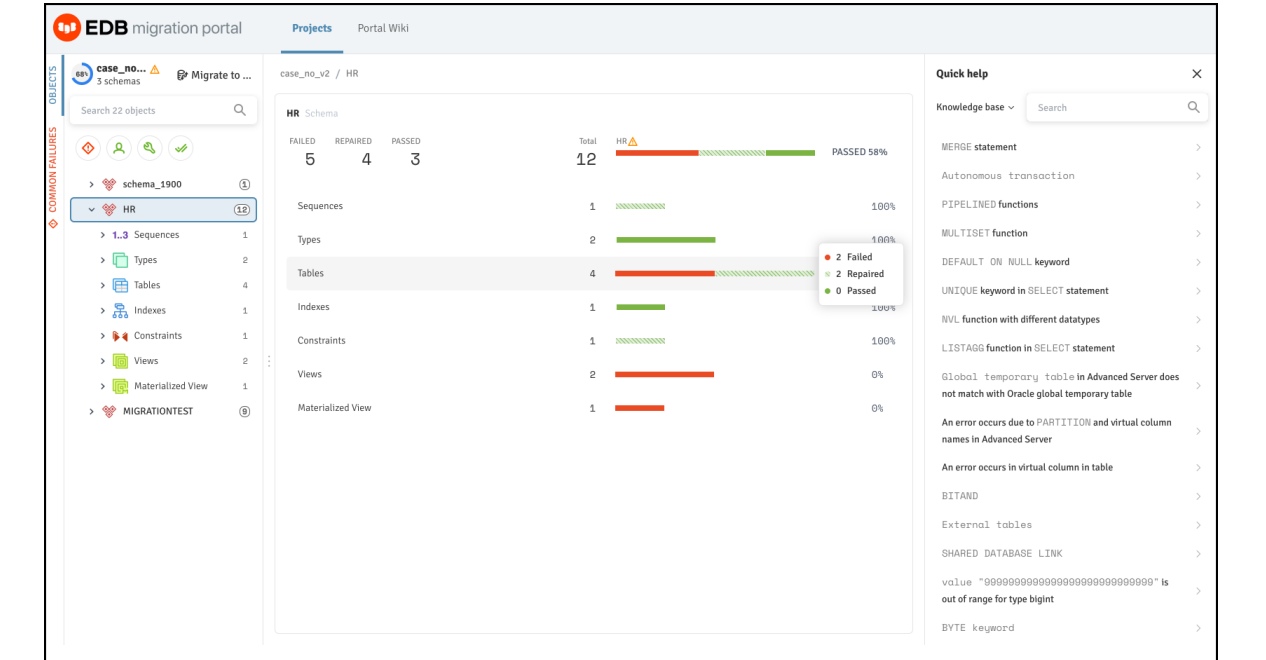


Fig. 9: *Assessment result with errors.*

12. On the **Knowledge Base** tab, you can enter the error message for the incompatible objects with EDB Postgres Advanced Server and click **Search**.

The object detail panel displays the workaround or the resolution for the failed object. You can manually make the changes on the **Assessment** tab for that object, and click **Reassess**.

Note: If any failed object passes while reassessing, the dependent objects for that object are also reassessed.

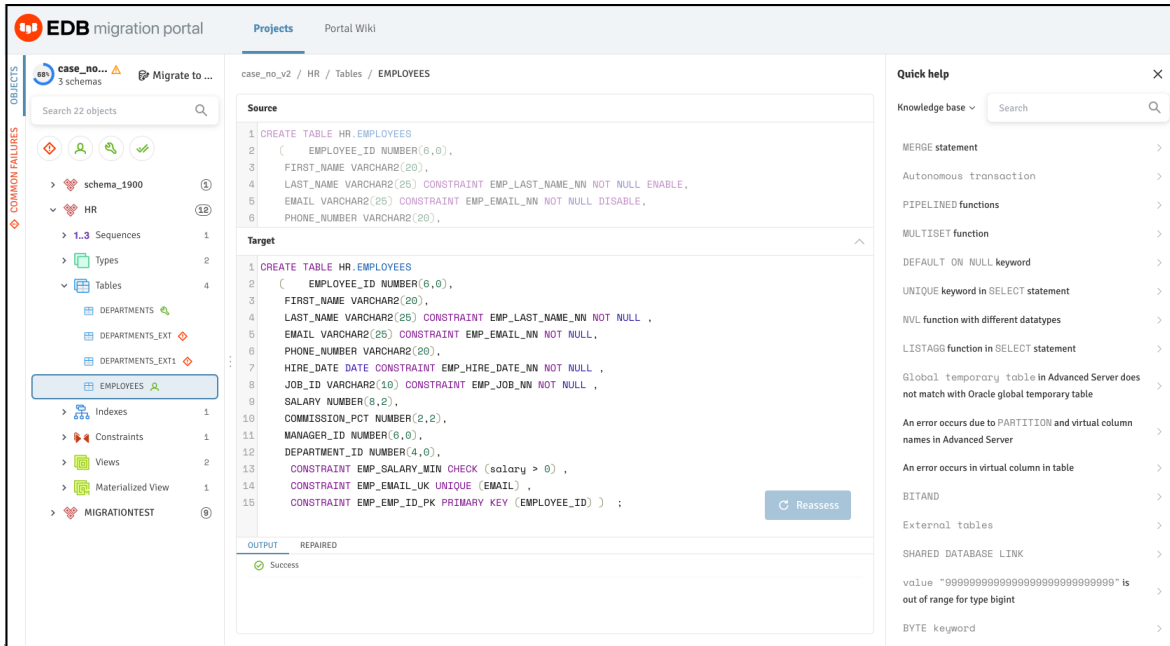


Fig. 10: Workaround or resolution for incompatible objects.

Similarly, you can make all the incompatible objects compatible.

Note: If the information or workaround for incompatible objects is not available in the Knowledge Base, please contact the support team for assistance.

When you have finished working with the DDL, you can either download the modified EDB compatible DDL as a .sql file or migrate the schemas see, [Schema Migration](#).

4.2.1 Generating an Assessment Report

Migration Portal's report functionality provides a high-level assessment summary of the schemas assessed for your project. In addition, the report provides details about the failed objects and the cause of failure.

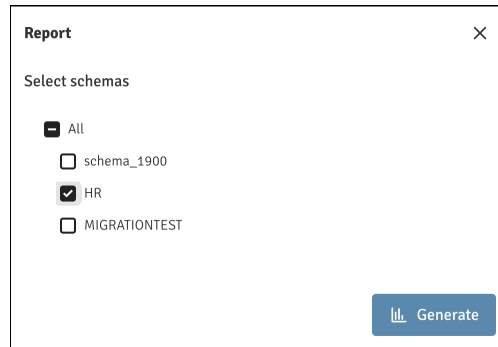


Fig. 11: *Select schemas for reports.*

To generate a report:

1. Click the **Report** button to access the schema selection dialog.
2. Select the schemas that you wish to include in the report.
3. Click **Generate** to generate the onscreen report.

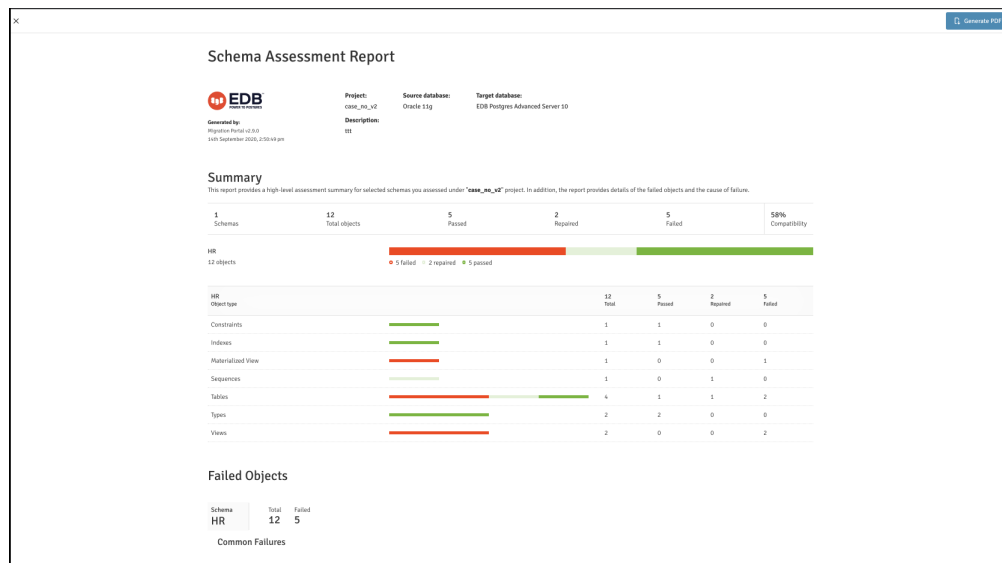


Fig. 12: *The Schema Assessment Report.*

You can optionally select **Generate PDF** to save the report in .pdf format. You can also view the count of distinct repair handlers applied to the DDLs under the selected schemas.

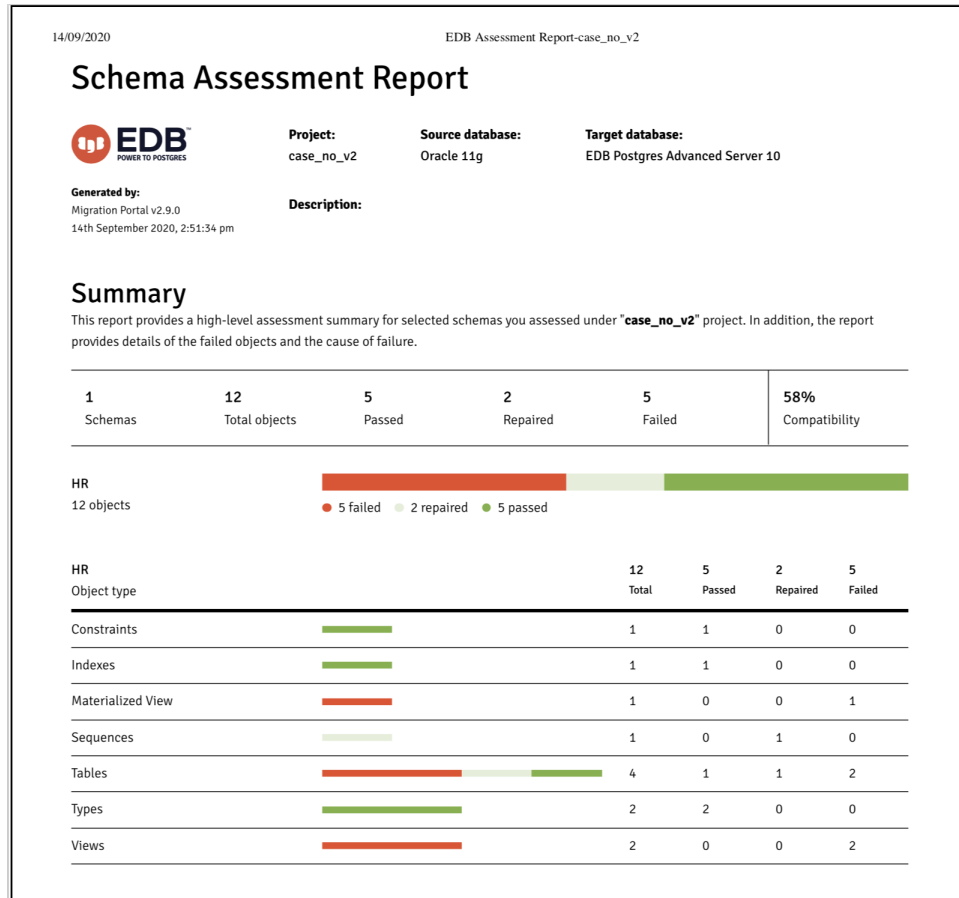


Fig. 13: The saved pdf report.

4.3 Schema Migration

After resolving errors in your schemas, you can use the schemas with a client application such as pgAdmin, ToadEdge, or the PSQL client, or migrate the schema to an EDB Postgres Advanced Server.

Note: For more information about using Toad Edge with EDB Postgres Advanced Server, see Toad Edge® for Postgres.

You can choose one of the following options for migrating schemas; migrate to an:

- Existing on-premise EDB Postgres Advanced Server
- New on-premise EDB Postgres Advanced Server
- EDB Postgres Advanced Server on Cloud

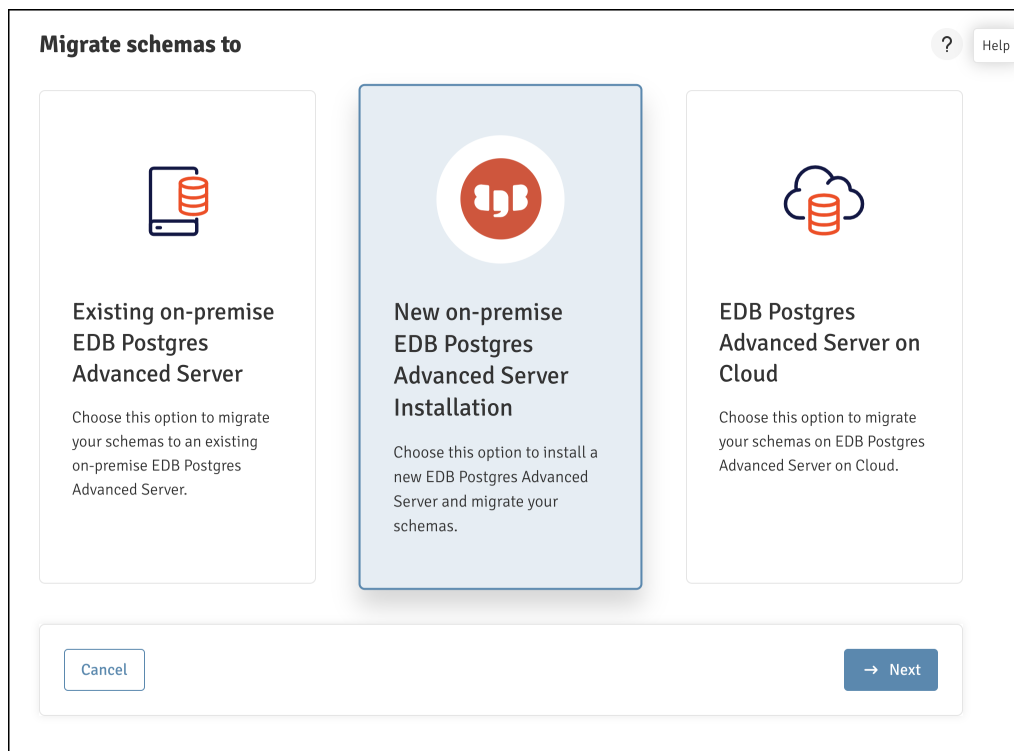


Fig. 14: Schema migration home page

4.3.1 Migrating to an Existing On-Premise EDB Postgres Advanced Server Host

You can migrate schemas to an existing on-premise EDB Postgres Advanced Server on Windows or Linux platforms.

Migrating Schemas on Windows

1. Select the Existing on-premise EDB Postgres Postgres Advanced Server option:

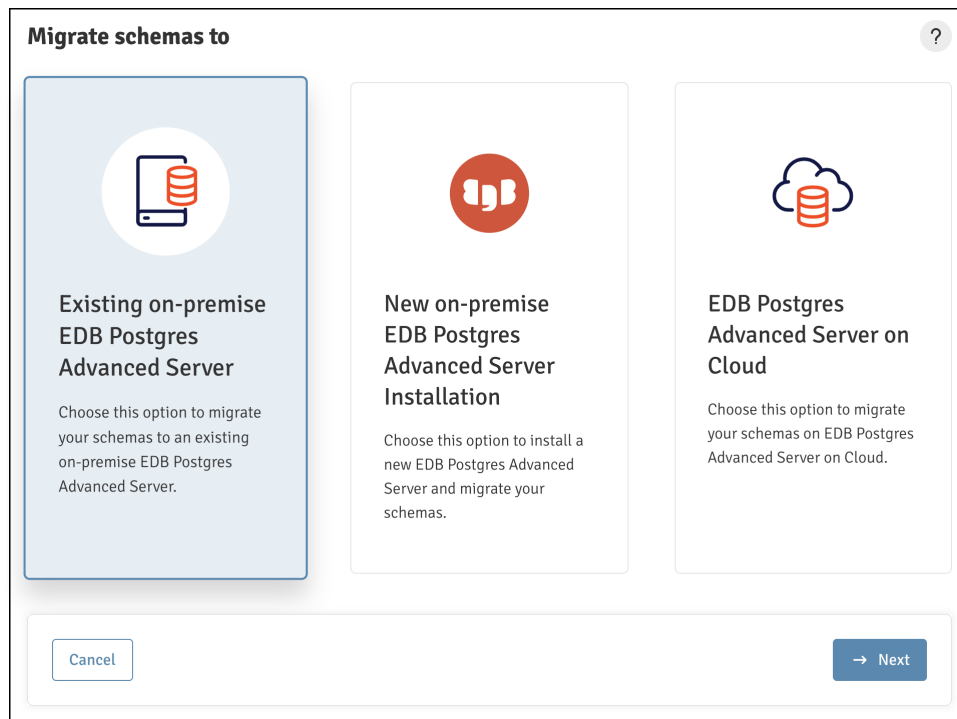


Fig. 15: Existing on-premise EDB Postgres Advanced Server home page

2. Select one or more schemas to migrate to EDB Postgres Advanced Server:

Migrate schemas to existing on-premise EDB Postgres Advanced Server

1 Select Schemas — 2 Download Schema — 3 Import — 4 Finish

⚠ This project has migration challenges. Contact us for help. [Contact Us](#) ×

Project Name	Interface	Source DB	Target DB	Compatibility
case_no_v2	ODBC	Oracle 11g	EDB Postgres Advanced Server 10	68% ⚠

Select one or more schemas

- ☒ All
- ☐ schema_1900
- ☒ HR
- ☐ MIGRATIONTEST

[Cancel](#) [← Previous](#) [→ Next](#)

Fig. 16: *Selecting schemas for migration*

Note: If your schemas are not 100% compatible, a banner will be displayed as shown; complete the [Contact Us](#) form for any assistance required.

Contact Us ×

Your message...

Required project details will be sent along with your message.

☒ By submitting this form, I agree to EDB's Terms & Conditions and Privacy Policy. [Send](#)

Fig. 17: *The Contact Us form*

3. Download the assessed schemas:

Migrate schemas to existing on-premise EDB Postgres Advanced Server

✓ Select Schemas
2 Download Schema
3 Import
4 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Download the assessed schemas

[Download SQL file](#)

Note: Ensure the version number for the EDB Postgres Advanced Server database that you are migrating is 12 (your assessed version).

Cancel
← Previous
→ Next

Fig. 18: Download the assessed schemas

4. Click on Windows:

Migrate schemas to existing on-premise EDB Postgres Advanced Server

✓ Select Schemas
✓ Download Schema
3 Import
4 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Import the selected schemas

Run the following command:

Windows
Linux

On CLI

```
edb=# \i c:\users\...\<project_name>.sql
```

On cmd/shell

```
edb-psql -f <project_name>.sql
```

[PSQL Documentation](#) | [Advanced Server Documentation](#)

Use pgAdmin instead.

Cancel
← Previous
→ Next

Fig. 19: Selecting Windows operating system

5. To import the schemas, run the following command:

- On CLI

```
\i c:\users\...\<project_name>.sql
```

- On cmd/shell

```
edb-psql -f <project_name>.sql
```

Note: You can also use pgAdmin instead.

The converted schemas are migrated to the target server.

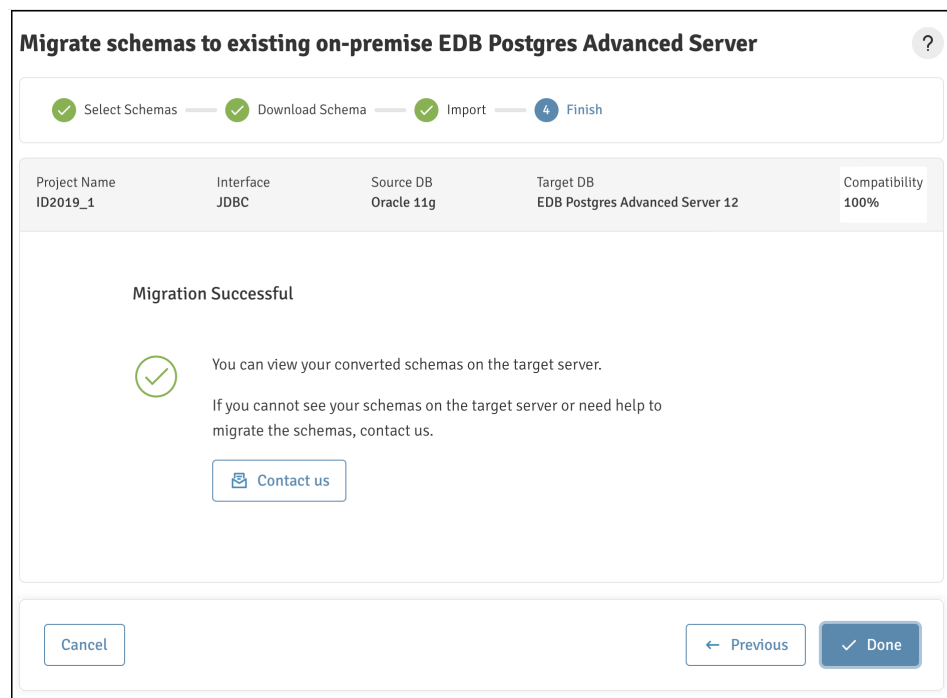


Fig. 20: A successful migration

Migrating Schemas on Linux

To migrate schemas to an existing on-premise EDB Postgres Advanced Server on Linux, complete the following steps:

1. Click the Existing on-premise EDB Postgres Advanced Server option:

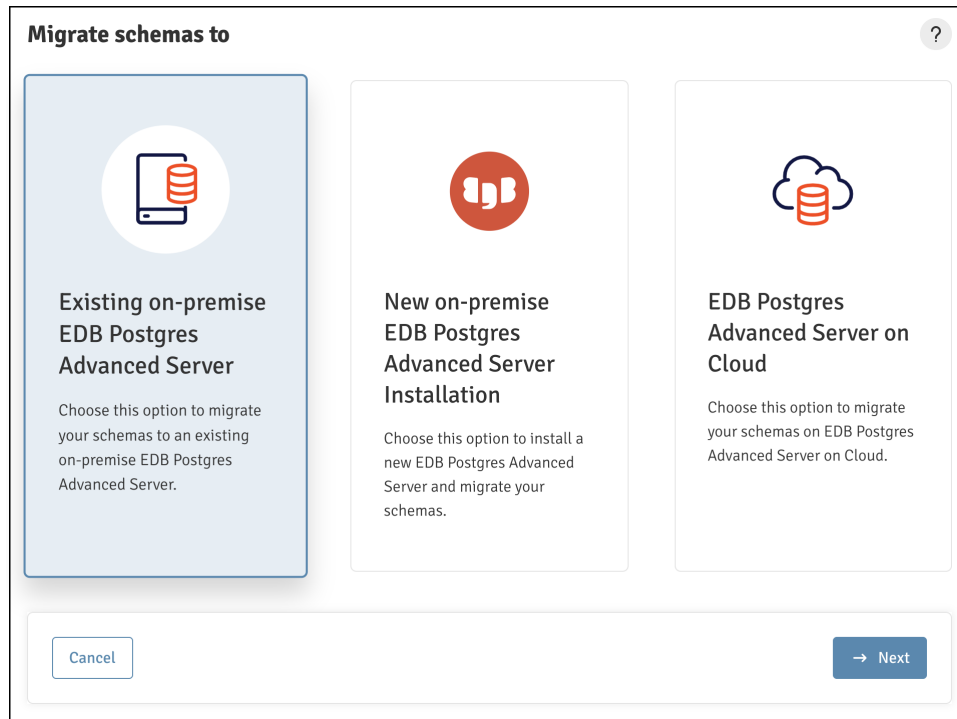


Fig. 21: Existing on-premise EDB Postgres Advanced Server home page

2. Select one or more schemas to migrate to EDB Postgres Advanced Server:

Migrate schemas to existing on-premise EDB Postgres Advanced Server

1 Select Schemas — 2 Download Schema — 3 Import — 4 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Select one or more schemas

☒ All
☒ HR

Cancel Previous Next

Fig. 22: Selecting schemas for migration

3. Download the assessed schemas:

Migrate schemas to existing on-premise EDB Postgres Advanced Server

✓ Select Schemas — 2 Download Schema — 3 Import — 4 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Download the assessed schemas

 Download SQL file

Note: Ensure the version number for the EDB Postgres Advanced Server database that you are migrating is 12 (your assessed version).

Cancel Previous Next

Fig. 23: Download the assessed schemas

4. Click on Linux:

Migrate schemas to existing on-premise EDB Postgres Advanced Server

Progress: ✓ Select Schemas ✓ Download Schema 3 Import 4 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Import the selected schemas

Run the following command:

Windows **Linux**

Run the following command on Terminal

```
shell=$ sudo su - enterprisedb
shell=$ edb-psql edb
edb=# create database <database_name>;
edb=# \connect <database_name>
database_name=# \i <project_name>.sql
```

[PSQL Documentation](#) | [Advanced Server Documentation](#)

Use pgAdmin instead.

Fig. 24: Selecting Linux operating system

5. To import the schemas, invoke the following `edb-psql` client commands:

```
sudo su - enterprisedb
edb-psql edb
create database <database_name>;
\\connect <database_name>
\\i <project_name>.sql
```

Note: You can optionally use the `pgAdmin` client for the import.

The converted schemas are migrated to the target server.

Migrate schemas to existing on-premise EDB Postgres Advanced Server

✓ Select Schemas

✓ Download Schema

✓ Import

4 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Migration Successful

✓

You can view your converted schemas on the target server.

If you cannot see your schemas on the target server or need help to migrate the schemas, contact us.

Contact us

Cancel

← Previous

✓ Done

Fig. 25: A successful schema migration

4.3.2 Migrating to a New On-Premise EDB Postgres Advanced Server Installation

You can install new EDB Postgres Advanced Server on-premise on Windows or Linux platforms and migrate the schemas.

Migrating Schemas on Windows

To migrate schemas to a new on-premise EDB Postgres Advanced Server on Windows, complete the following steps:

1. Click New on-premise EDB Postgres Postgres Advanced Server option.

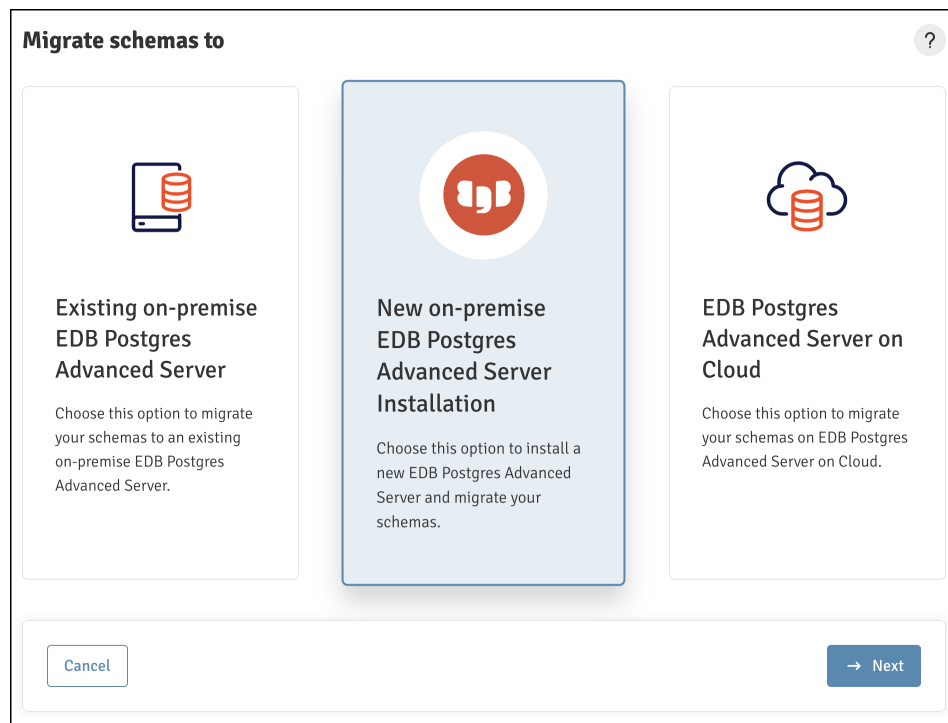


Fig. 26: New on-premise EDB Postgres Advanced Server home page

2. Select one or more schemas to migrate on EDB Postgres Advanced Server.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

1 Select Schemas — 2 Select Platform — 3 Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Ser >

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Select one or more schemas

☒ All

☒ HR

Cancel Previous Next

Fig. 27: Selecting schemas for migration

3. Select the Windows operating system.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

✓ Select Schemas — 2 Select Platform — 3 Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Ser >

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Select the Operating System/Platform

☒ Windows 64-bit

☐ Linux 64-bit

Cancel Previous Next

Fig. 28: Selecting Windows operating system

4. Download Windows Installer.

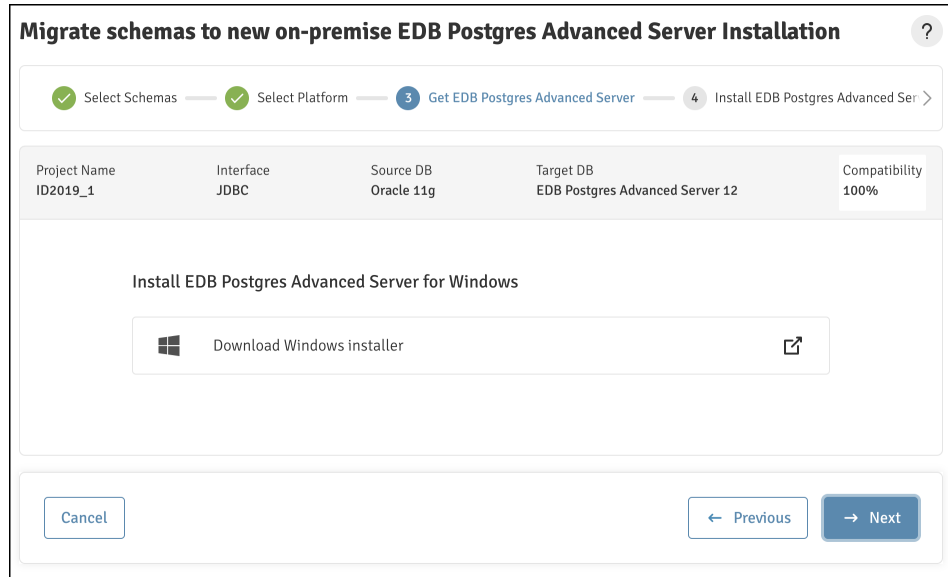


Fig. 29: Downloading Windows installer

- For installation steps, click EDB Postgres™ Advanced Server Installation Guide for Windows.

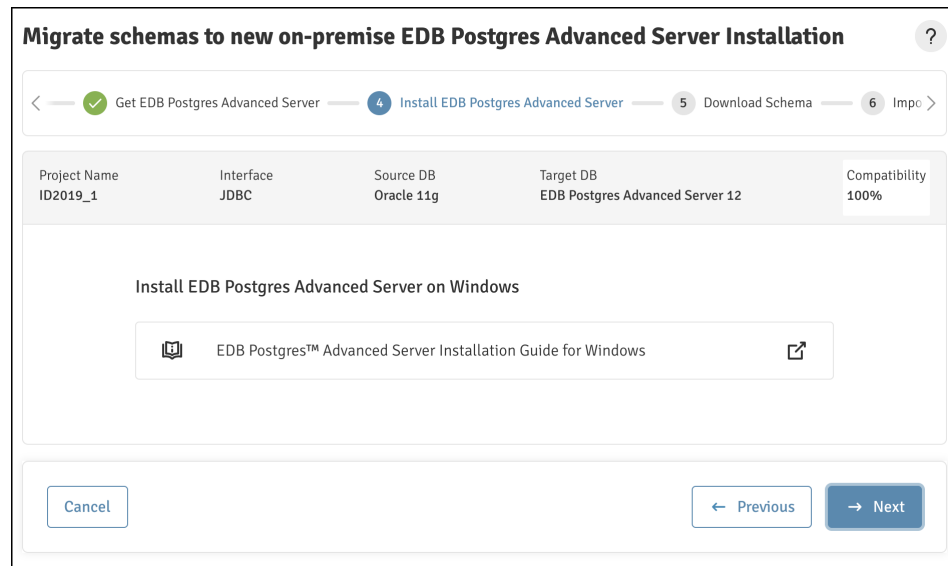


Fig. 30: View Windows installation guide

- Download the assessed schemas.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

< Postgres Advanced Server — ✓ Install EDB Postgres Advanced Server — **5 Download Schema** — 6 Import — 7 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Download the assessed schemas

[Download SQL file](#)

Note: Ensure the version number for the EDB Postgres Advanced Server database that you are migrating is 12 (your assessed version).

[Cancel](#) [← Previous](#) [→ Next](#)

Fig. 31: *Downloading the assessed file*

7. You can import schemas by running the following command:

- On CLI

```
\i c:\users\...\<project_name>.sql
```

- On cmd/shell

```
edb-psql -f <project_name>.sql
```

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

< Postgres Advanced Server

✓ Install EDB Postgres Advanced Server

✓ Download Schema

6 Import

7 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Import schemas into EDB Postgres Advanced Server

On CLI

```
edb=# \i c:\users\...\<project_name>.sql
```

On cmd/shell

```
edb-psql -f <project_name>.sql
```

[PSQL Documentation](#) | [Advanced Server Documentation](#)

Use  pgAdmin instead.

Cancel

← Previous

→ Next

Fig. 32: Importing schemas into EDB Postgres Advanced Server

Note: You can also use pgAdmin instead.

The schemas are migrated to the target server.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

< Postgres Advanced Server

✓ Install EDB Postgres Advanced Server

✓ Download Schema

✓ Import

7 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Migration Successful

✓

You can view your converted schemas on the target server.

If you cannot see your schemas on the target server or need help to migrate the schemas, contact us.

Contact us

Cancel

← Previous

✓ Done

Migrating Schemas on Linux

To migrate schemas to an on-premise EDB Postgres Advanced Server on Linux, complete the following steps:

1. Click New On-premise EDB Postgres Advanced Server option.

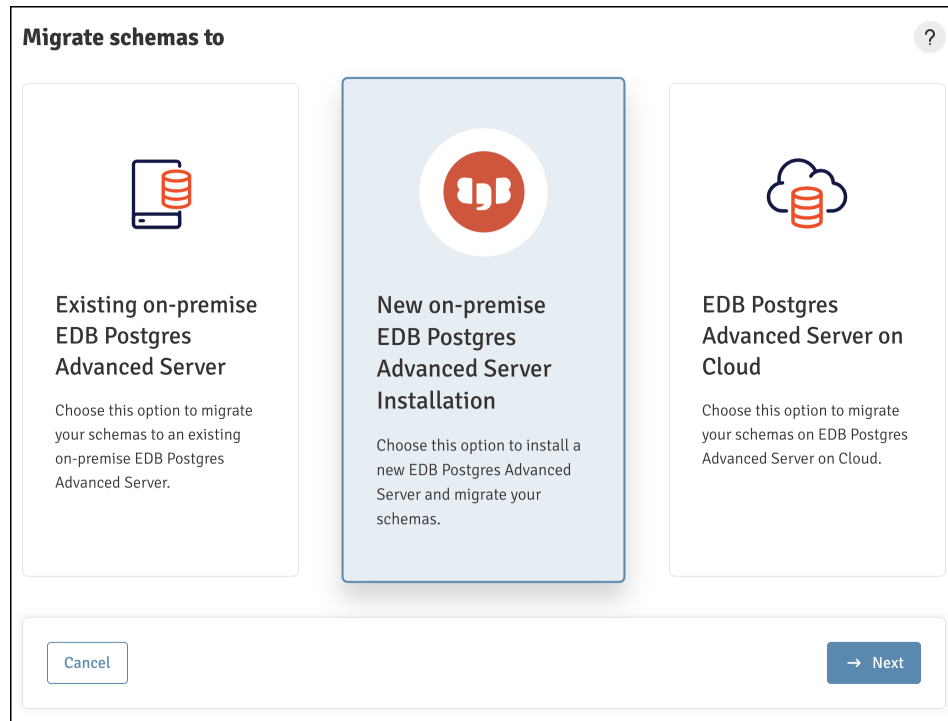


Fig. 33: *Migrating schemas home page*

2. Select one or more schemas to migrate on EDB Postgres Advanced Server.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

1 Select Schemas — 2 Select Platform — 3 Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Server >

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Select one or more schemas

☒ All

☒ HR

Cancel Previous Next

Fig. 34: Selecting schemas for migration

3. Select the Linux operating system.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

✓ Select Schemas — 2 Select Platform — 3 Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Server >

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Select the Operating System/Platform

☐ Windows 64-bit

☒ Linux 64-bit

Cancel Previous Next

Fig. 35: Selecting Linux operating system

4. You can select one of the following options to install the EDB Postgres Advanced Server:

- Repository
- More options

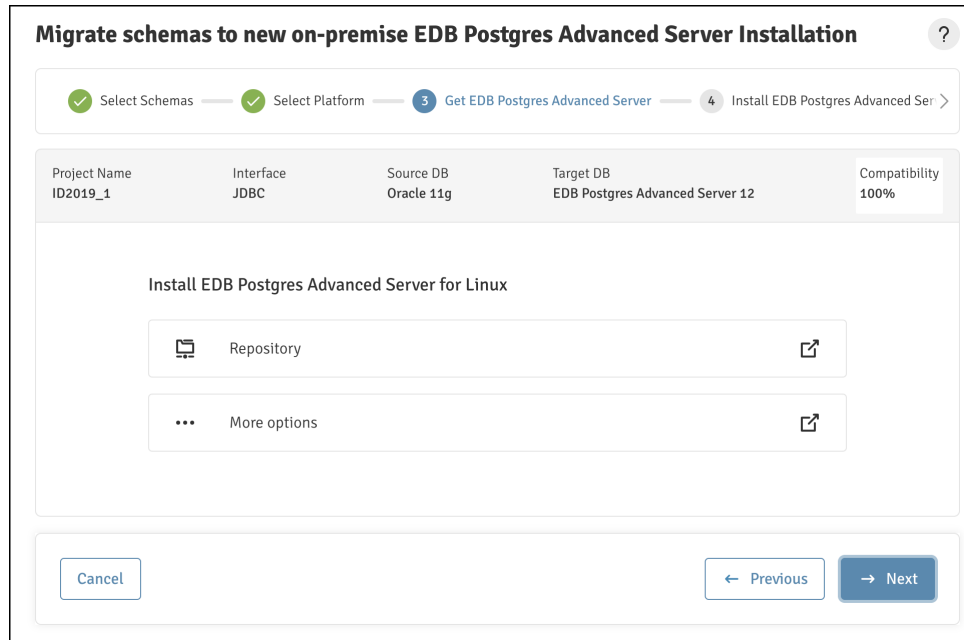


Fig. 36: *Selecting Linux repository*

- For information on the installation procedure, click EDB Postgres™ Advanced Server Installation Guide for Linux:

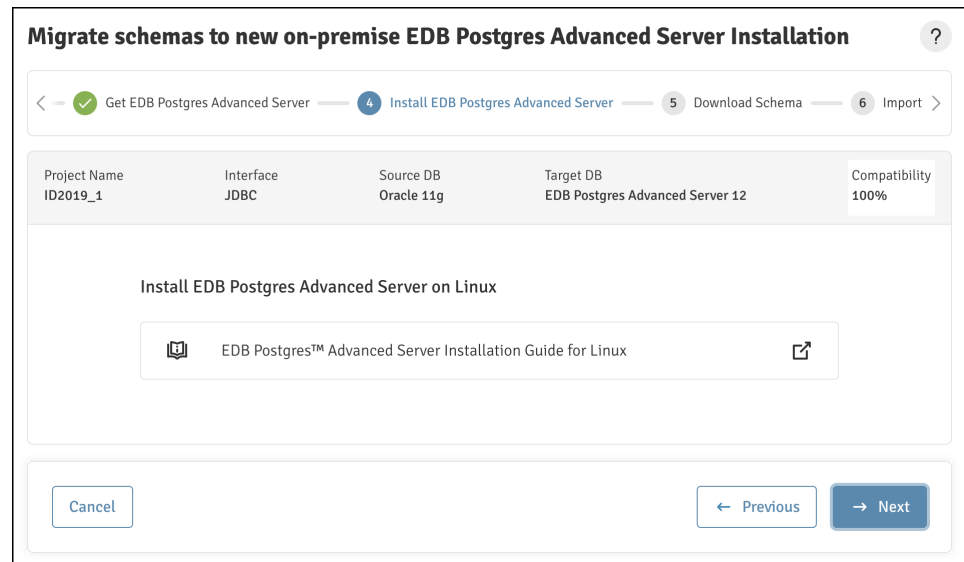


Fig. 37: *Selecting Linux installation guide*

- Download the assessed schemas:

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

< Postgres Advanced Server — ☒ Install EDB Postgres Advanced Server — **5 Download Schema** — 6 Import — 7 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Download the assessed schemas

[Download SQL file](#)

Note: Ensure the version number for the EDB Postgres Advanced Server database that you are migrating is 12 (your assessed version).

[Cancel](#) [← Previous](#) [→ Next](#)

Fig. 38: *Downloading the assessed schemas*

7. To import the schemas, run the following command:

```
sudo su - enterprisedb
edb-psql edb
create database <database_name>;
\connect <database_name>
\i <project_name>.sql
```

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

< Import Schemas to new on-premise EDB Postgres Advanced Server — ✓ Install EDB Postgres Advanced Server — ✓ Download Schema — 6 Import — 7 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Import schemas into EDB Postgres Advanced Server

Run the following command on Terminal

```
shell=$ sudo su - enterprisedb
shell=$ edb-psql edb
edb=# create database <database_name>;
edb=# \connect <database_name>
database_name=# \i <project_name>.sql
```

[PSQL Documentation](#) | [Advanced Server Documentation](#)

Use  [pgAdmin](#) instead.

Cancel< PreviousNext >

Fig. 39: Importing schemas into EDB Postgres Advanced Server

Note: You can also use pgAdmin instead.

The converted schemas are migrated to the target server.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

< Postgres Advanced Server

✓ Install EDB Postgres Advanced Server

✓ Download Schema

✓ Import

7 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Migration Successful

✓

You can view your converted schemas on the target server.

If you cannot see your schemas on the target server or need help to migrate the schemas, contact us.

Contact us

Cancel

← Previous

✓ Done

Fig. 40: A successful schema migration

4.3.3 Migrating to the Cloud

To migrate schemas on EDB Postgres Advanced Server to Cloud, complete the following steps:

1. Click EDB Postgres Advanced Server on Cloud option:

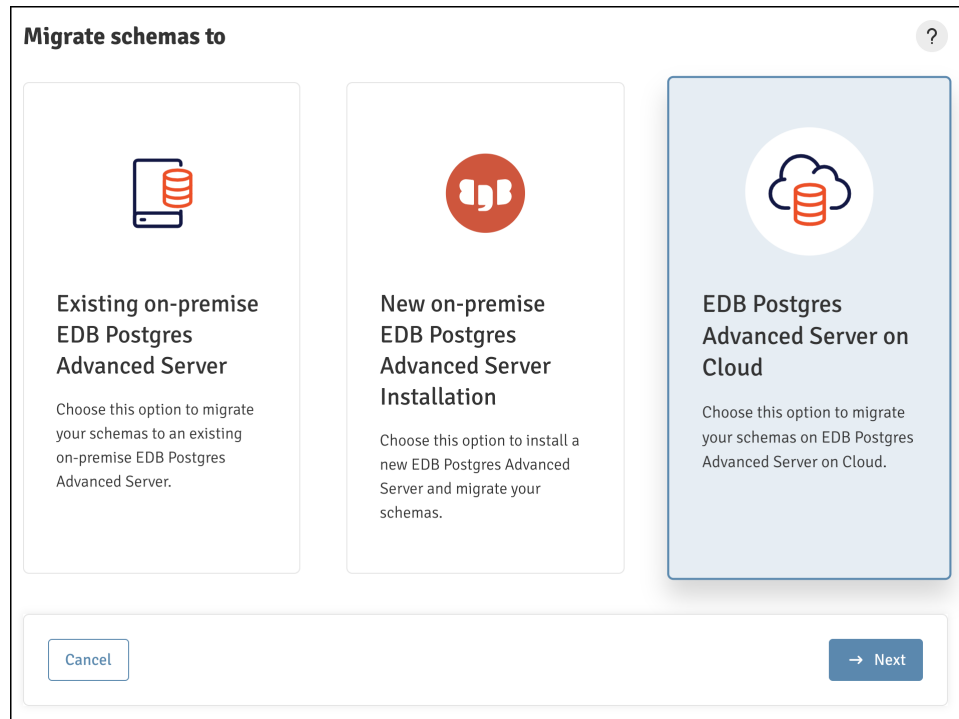


Fig. 41: *EDB Postgres Advanced Server on Cloud*

2. Select one or more schemas to migrate to EDB Postgres Advanced Server:

Migrate schemas to EDB Postgres Advanced Server on Cloud

1 Select Schemas — 2 Select Platform — 3 Launch Cluster — 4 Connect — 5 Deploy — 6 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Select one or more schemas

☒ All
☒ HR

Cancel Previous Next

Fig. 42: Selecting schemas for migration

3. Select the cloud platform. For example, IBM Cloud:

Migrate your schemas to the cloud

1 Select Schemas — 2 Select Platform — 3 Launch Cluster — 4 Connect — 5 Deploy — 6 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
HumanResources	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Select the Cloud Platform

☒ IBM Cloud
☐ AWS
☐ Google Cloud
☐ Microsoft Azure

Cancel Previous Next

Fig. 43: Selecting cloud option for migration

4. To launch a new cluster, click Go to Cloud:

The screenshot shows a wizard titled "Migrate schemas to EDB Postgres Advanced Server on Cloud". At the top, a progress bar indicates six steps: 1. Select Schemas (checked), 2. Select Platform (checked), 3. Launch Cluster (active), 4. Connect, 5. Deploy, and 6. Finish. Below the progress bar, a table displays migration details:

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

The main content area is titled "Launch new IBM™ Cloud cluster" and contains the instruction: "Navigate to IBM™ Cloud to launch a new cluster." Below this is a blue button labeled "→ Go to IBM™ Cloud". A note at the bottom states: "Note: If you have an existing cluster running, click Next." At the bottom of the wizard, there are three buttons: "Cancel", "← Previous", and "→ Next".

Fig. 44: *Launching a cloud cluster*

Or, if you have an existing cluster running, click **Next**.

5. Enter the required connection details on the **Connect** page:

Migrate schemas to EDB Postgres Advanced Server on Cloud

Progress: ✓ Select Schemas ✓ Select Platform ✓ Launch Cluster **4 Connect** 5 Deploy 6 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Connect to IBM™ Cloud cluster

Target Database
ID2019_1

Host Name/Address
Host Name/Address

Port
9999

Note: Ensure that your database server is accessible from Migration Portal.

Maintenance Database
edb

Username
enterprisedb

Password
Password

[Cancel](#) [← Previous](#) [→ Next](#)

Fig. 45: Connecting to the cloud cluster

Note: You can click [Edit Connection](#) to make changes to the connection details and retest the connection details.

7. Click [Test Connection](#) to verify the connection details:

Migrate schemas to EDB Postgres Advanced Server on Cloud

✓ Select Schemas

✓ Select Platform

✓ Launch Cluster

4 Connect

5 Deploy

6 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Target Database

ID2019_1

Host Name/Address	Port
52.5.148.59	9999

Maintenance Database

dmp_exec

Note: Deployment will create a new database **ID2019_1** on target server, if the database does not exist.

Ensure that all schemas have a 100% success ratio before deployment.

Existing schemas with similar names will be dropped during deployment.

Edit Connection

Cancel

← Previous

→ Next

Fig. 46: Verify the connection details

8. Once the connection is successful, click Next:

Migrate schemas to EDB Postgres Advanced Server on Cloud

✓ Select Schemas

✓ Select Platform

✓ Launch Cluster

✓ Connect

✓ Deploy

6 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
ID2019_1	JDBC	Oracle 11g	EDB Postgres Advanced Server 12	100%

Migration Successful

✓

You can view your converted schemas on the target server.

[Download Summary](#)

If you cannot see your schemas on the target server or need help to migrate the schemas, contact us.

[Contact us](#)

[Cancel](#)[← Previous](#)[✓ Done](#)

Fig. 47: A successful migration

The converted schemas are migrated to the target server.

4.4 Data Migration

After performing the schema migration, complete the following steps to migrate data:

1. Use Migration Toolkit to migrate the data. For detailed information about using Migration Toolkit, see the [Migration Toolkit Guide](#).
2. Configure the Migration Toolkit `toolkit.properties` file, ensuring that connection information for the source and target databases is available in the property file:

```
SRC_DB_URL = jdbc:oracle:thin:@localhost:1521:ORCL
SRC_DB_USER = user_name
SRC_DB_PASSWORD = password
TARGET_DB_URL= jdbc:edb://localhost:5444/migration
TARGET_DB_USER = enterprisedb
TARGET_DB_PASSWORD = password
```

For more information, see [Building the toolkit.properties File](#).

3. Invoke Migration Toolkit in `-dataOnly` mode; include the `-truncLoad` keyword to resolve foreign key dependencies across tables.

For example, the following command migrates data to a schema named `hr`:

```
runMTK.sh -dataOnly -targetSchema hr -truncLoad HR
```

The command migrates the specified `source_schema` to the `target_schema`. The data is loaded into the locally installed EDB Postgres instance with a database superuser named `enterprisedb` and the password of `password`.

Note: The tables are truncated before attempting the data load.

CHAPTER 5

Advanced Data Migration

For larger databases that require a parallel data load, you can use one of the following methods:

- Use the EDB Postgres Advanced Server database link feature (for compatibility with Oracle databases).

Or

- Use a Dblink or a database link style migration (if your data contains CLOB data).

For more information, see the [Migration Toolkit Guide](#).

CHAPTER 6

Conclusion

Migration Portal User's Guide

Copyright © 2019 - 2020 EnterpriseDB Corporation. All rights reserved.

EnterpriseDB® Corporation 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E info@enterprisedb.com www.enterprisedb.com

- EnterpriseDB and Postgres Enterprise Manager are registered trademarks of EnterpriseDB Corporation. EDB and EDB Postgres are trademarks of EnterpriseDB Corporation. Oracle is a registered trademark of Oracle, Inc. Other trademarks may be trademarks of their respective owners.
- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres products where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.
- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.
- EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.

A

Advanced Data Migration, [57](#)

C

Conclusion, [58](#)

D

Data Migration, [56](#)

K

knowledge base, [13](#), [15](#)

M

Migrating a Database, [16](#)

Migration Portal, [1](#)

Migration Toolkit, [56](#)

O

Overview of the Migration
Portal Home Page, [10](#)

Overview of the Migration
Portal Projects Page, [12](#)

Overview of the Migration
Portal Wiki Page, [15](#)

P

Performing a Schema Assessment,
[22](#)

Performing a Schema Extraction,
[17](#)

R

repair handlers, [15](#)

resources, [15](#)

S

schema assessment report, [26](#), [27](#)

Schema Migration, [29](#)

supported browsers, [6](#)

supported databases, [6](#)

supported object types, [19](#), [20](#)

supported operating systems, [6](#)

Supported Versions, [6](#)

U

Using Migration Portal, [7](#)

W

What's New, [2](#)