Contents

---

0 Migration Portal

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/

---

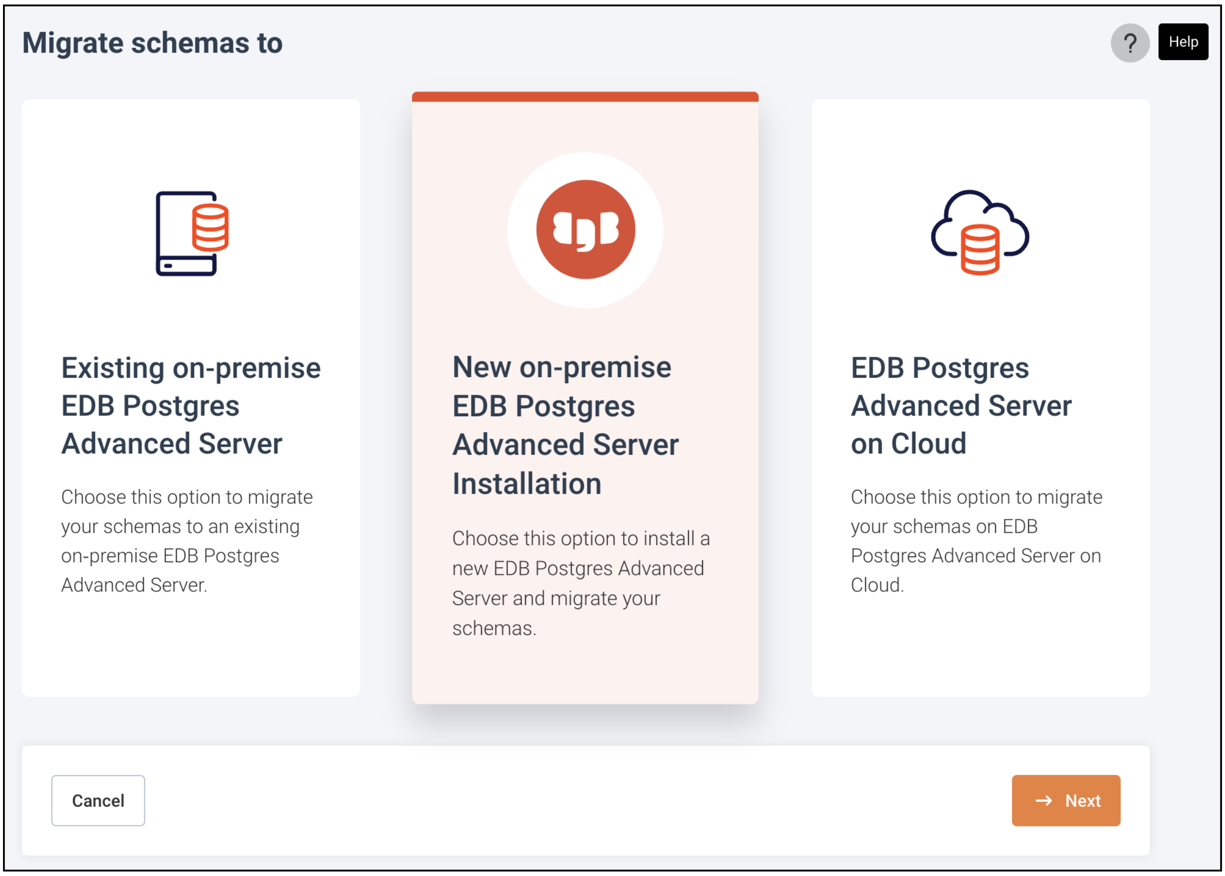
1 What's New

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

- Starting this release, you can install EDB Postgres Advanced Server or use EDB Postgres Advanced Server on the cloud directly from the Migration Portal.

  Following are the options for migrating schemas:

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



Migrate schemas to

Existing on-premise EDB Postgres Advanced Server

Choose this option to migrate your schemas to an existing on-premise EDB Postgres Advanced Server.

New on-premise EDB Postgres Advanced Server Installation

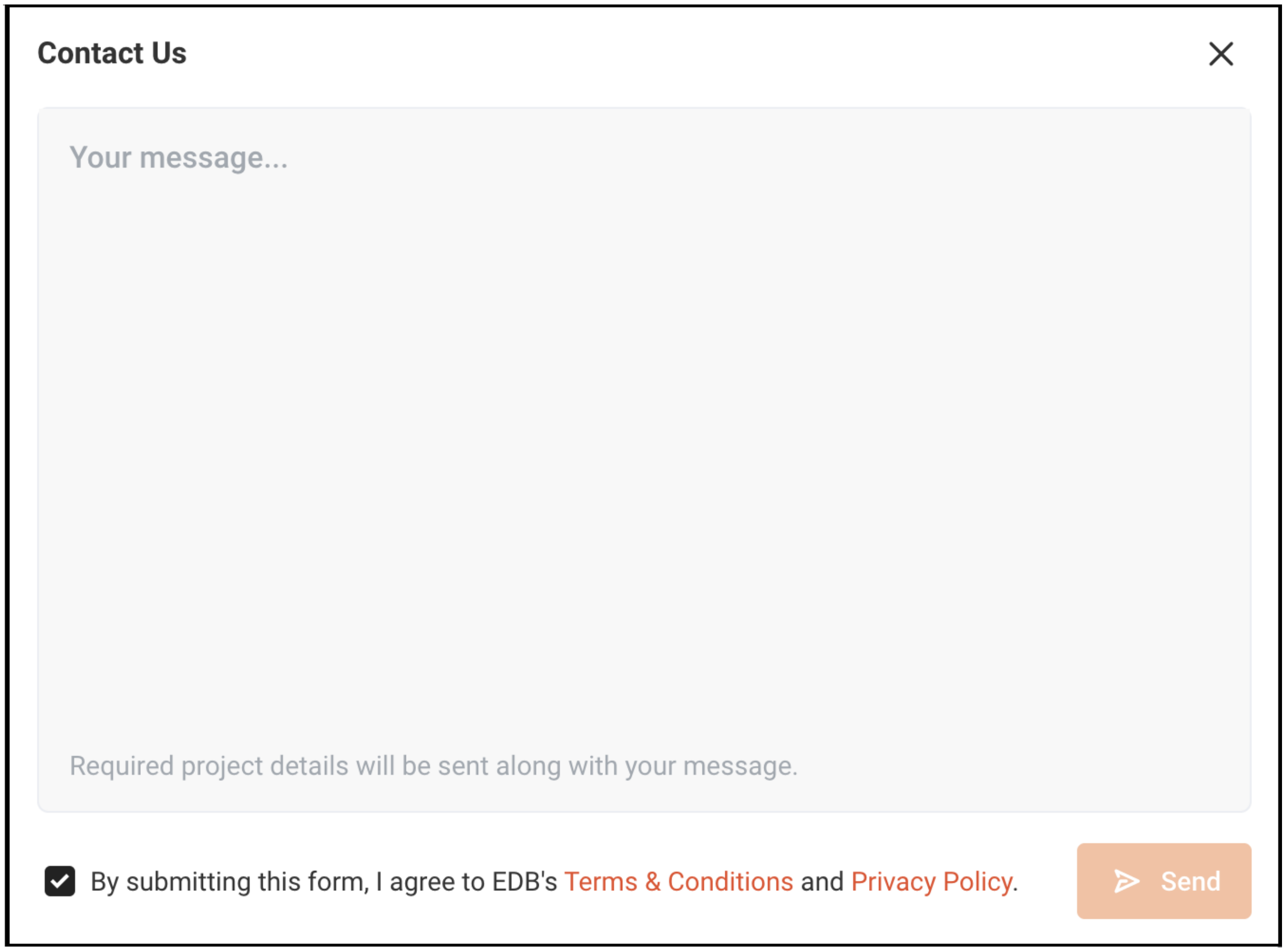
Choose this option to install a new EDB Postgres Advanced Server and migrate your schemas.

EDB Postgres Advanced Server on Cloud

Choose this option to migrate your schemas on EDB Postgres Advanced Server on Cloud.

For more information, see `Schema Migration <mp_schema_migration>` .

- You can now fill up a `Contact Us` form for any help required for migration, installation, or other related queries.



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.

- Many changes are made to the new UI for better user experience.
- Report generating duration for larger projects is optimized drastically. For example, previously, the time required to generate a report was 5 to 7 minutes. However, now it takes less than 5 seconds to generate a report.

New Repair Handler

The following repair handler is added to improve the Advance Server compatibility ratio:

- ERH 2090 - Convert Function:

  Replaces Oracle's `CONVERT` function with EDB Postgres Advanced Server compatible `convert` or `convert_to` function inside any PL/SQL block.

  For example:

  ```
  CREATE VIEW MY_VIEW1 AS SELECT CONVERT('Ä Ê Í Õ Ø A B C D E','AL32UTF8','
  WE8ISO8859P1') as CONVERT_TEST FROM dual;
  CREATE VIEW MY_VIEW2 AS SELECT CONVERT('Ä Ê Í Õ Ø A B C D E','AL32UTF8') as
  CONVERT_TEST FROM dual;
  ```

  would become;

  ```
  CREATE VIEW MY_VIEW1 AS SELECT convert('Ä Ê Í Õ Ø A B C D E','LATIN1','UTF8')
  as CONVERT_TEST FROM dual;
  CREATE VIEW MY_VIEW2 AS SELECT convert_to('Ä Ê Í Õ Ø A B C D E','UTF8')
  as CONVERT_TEST FROM dual;
  ```

2 Supported Versions

The Migration Portal supports assessment and migration from Oracle 11g and 12c to EDB Postgres Advanced Server 10, 11, or 12. 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

3.0 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.

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` .

Accessing the Migration Portal.

1. Click `Open Migration Portal` .
2. 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.

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. **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 <mp_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.

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-prmise 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.

   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.

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

The Migration Portal Wiki page.

4.0 Migrating a Database

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

1. Perform a `Schema Extraction <mp_schema_extraction>` .
2. Perform a `Schema Assessment <mp_schema_assessment>` .
3. Perform a `Schema Migration <mp_schema_migration>` .
4. Perform a `Data Migration <mp_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

The *script user* must have `CONNECT` , `RESOURCE` and `SELECT_CATALOG_ROLE` roles.

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.

1. 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,
   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
   Assessment may fail for such objects. It is suggested to extract all dependent objects to

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

   On Windows:

   ```
   Enter comma separated list of schemas to be extracted (Default all schemas): HR, SCOTT, F
   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
Assessment may fail for such objects. It is suggested to extract all dependent objects to
```

For SQL Developer

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

Enter the path for Linux or Windows.

1. Enter a comma-separated list of schemas:

Provide a list of schemas.

1. Enter the file path for the output file:

Specify the output file path.

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

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.

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.

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

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	DVSYS	EXFSYS
FLOWS_FILES	FLOWS_020100	GSMADMIN_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
SPATIAL_CSW_ADMIN_USR	SPATIAL_WFS_ADMIN_USR	SYS
SYSBACKUP	SYSDG	SYSKM
SYSTEM SYSMAN	TSMSYS WKPROXY	WKSYS
WK_TEST XS$NULL	WMSYS	XDB

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.

The Migration Portal New project dialog.

1. On the `New project` dialog, enter the project name in the `Project name` field.

2. 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.

3. 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 DDL Extractor can be uploaded.

   For more information, refer to the `Schema Extraction <mp_schema_extraction>` section.

4. 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.

5. Click `Create & assess` to create a new project and to assess the schemas.

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.

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

Verifying the DDL objects.

1. 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.

Incompatible objects are identified.

1. 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 Advanced Server. You can also view the Knowledge Base information on the Portal Wiki page.

Assessment result with errors.

1. 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.

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.

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.

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.

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.

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

Schema migration home page

Existing on-Premise EDB Postgres Advanced Server

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:

Existing on-premise EDB Postgres Advanced Server home page

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

Selecting schemas for migration

Note

In case your schemas are not 100% compatible, a banner will be displayed as shown, and you can fill the Contact Us form for any assistance required.

The Contact Us form

1. Download the assessed schemas:

Download the assessed schemas

1. Click on `Windows` :

Selecting Windows operating system

1. 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
  ```

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:

Existing on-premise EDB Postgres Advanced Server home page

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

Selecting schemas for migration

1. Download the assessed schemas:

Download the assessed schemas

1. Click on `Linux` :

Selecting Linux operating system

1. 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
```

New On-Premise EDB Postgres Advanced Server

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.

New on-premise EDB Postgres Advanced Server home page

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

Selecting schemas for migration

1. Select the `Windows` operating system.

Selecting Windows operating system

1. Download Windows Installer.

Downloading Windows installer

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

View Windows installation guide

1. Download the assessed schemas.

Downloading the assessed file

1. 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
     ```

Importing schemas into EDB Postgres Advanced Server

Note

You can also use pgAdmin instead.

The schemas are migrated to the target server.

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.

Migrating schemas home page

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

Selecting schemas for migration

1. Select the `Linux` operating system.

Selecting Linux operating system

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

- More options

  Selecting Linux repository

2. For information on the installation procedure, click `EDB Postgres Advanced Server Installation Guide` for Linux:

Selecting Linux installation guide

1. Download the assessed schemas:

Downloading the assessed schemas

1. 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
   ```

Importing schemas into EDB Postgres Advanced Server

Note

You can also use pgAdmin instead.

The converted schemas are migrated to the target server.

A successful schema migration

EDB Postgres Advanced Server on Cloud

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

1. Click `EDB Postgres Advanced Server on Cloud` option:

EDB Postgres Advanced Server on Cloud

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

Selecting schemas for migration

1. Select the cloud platform. For example, `IBM Cloud` :

Selecting cloud option for migration

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

Launching a cloud cluster

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

1. Enter the required connection details on the Connect page:

Connecting to the cloud cluster

Note

You can click `Edit Connection` to make changes to the connection details and retest the connection details.

1. Click `Test Connection` to verify the connection details:

Verify the connection details

1. Once the connection is successful, click `Next` :

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.

1. 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.

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.