



EDB

Quick start guide

Release 15.2.0

EnterpriseDB Corporation

Mar 03, 2023

CONTENTS

1	Quick start for RHEL/Rocky Linux/AlmaLinux 8	3
1.1	Components of an EDB Postgres Advanced Server deployment	3
1.1.1	Prerequisites	4
1.1.2	Installing and configuring EDB Postgres Advanced Server	4
1.1.3	Starting the cluster	4
1.1.4	Using the psql command line client	5
2	Quick start for CentOS or RHEL 7	7
2.1	Components of an EDB Postgres Advanced Server deployment	7
2.2	Prerequisites	8
2.3	Installing and configuring EDB Postgres Advanced Server	8
2.4	Starting the cluster	8
2.5	Using the psql command line client	9
3	Quick start for Windows	11
3.1	Components of an EDB Postgres Advanced Server deployment	11
3.2	Installation prerequisites	12
3.2.1	User privileges	12
3.2.2	Windows-specific software requirements	12
3.2.3	Migration Toolkit or EDB*Plus installation Prerequisites	12
3.2.4	Installing and configuring EDB Postgres Advanced Server	12
4	EDB pgAdmin4 Quick Start Linux	25
4.1	Installing EDB pgAdmin 4 on a Linux host	25
4.1.1	Create a repository configuration file	25
4.1.2	Install EPEL repository	25
4.1.3	Install EDB pgAdmin 4	26
4.1.4	Starting pgAdmin 4 in desktop mode	26
4.1.5	Registering and connecting to EDB Postgres Advanced Server with pgAdmin 4	27

This book describes following:

- Quick start for RHEL/Rocky Linux/AlmaLinux 8
- Quick start for CentOS or RHEL 7
- Quick start for Windows
- EDB pgAdmin4 Quick Start Linux

QUICK START FOR RHEL/ROCKY LINUX/ALMALINUX 8

EDB Postgres Advanced Server adds extended functionality to the open-source PostgreSQL database that supports database administration, enhanced SQL capabilities, database and application security, performance monitoring and analysis, and application development utilities. EDB Postgres Advanced Server also supports database compatibility features for Oracle users; for detailed information about compatibility features, see the EDB Postgres Advanced Server documentation.

This guide walks you through using `dnf` to install EDB Postgres Advanced Server on a RHEL/Rocky Linux/AlmaLinux 8 system and deploying a database cluster. The database created by this tutorial is well-suited for experimentation and testing. There are likely to be additional security and resource considerations when configuring a production installation that are not covered by this document.

This guide assumes that you are familiar with simple operating system and system administration procedures, and have administrative privileges on the host on which EDB Postgres Advanced Server is installed.

1.1 Components of an EDB Postgres Advanced Server deployment

Among the components that make up an EDB Postgres Advanced Server deployment are:

The database server - The database server (the `postmaster`) is the service that provides the key functionality that allows you to store and manage data. EDB Postgres Advanced Server is built on the PostgreSQL open-source database project; it includes all of the documented features of community PostgreSQL and more.

The database cluster - A cluster is a set of on-disk structures that comprise a collection of databases. A cluster is serviced by a single-instance of the database server. A database cluster is stored in the `data` directory. Don't store the `data` directory of a production database on an NFS file system.

Configuration files - You can use the parameters listed in Postgres configuration files to manage deployment preferences, security preferences, connection behaviors, and logging preferences.

Supporting tools, utilities, and clients - EDB makes available a full suite of tools and utilities that can help you monitor and manage your EDB Postgres Advanced Server deployment. For more information, visit the

[EDB website](#).

Supporting functions, procedures, data types, index types, operators, utilities, and aggregates - EDB Postgres Advanced Server includes a number of features that help you manage your data.

1.1.1 Prerequisites

Before installing EDB Postgres Advanced Server, assume superuser privileges and use `dnf` to install prerequisite packages:

```
dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

If you are installing Migration Toolkit or EDB*Plus, you must first install Java:

```
dnf -y install java
```

1.1.2 Installing and configuring EDB Postgres Advanced Server

Step1: You must register with EDB to receive credentials for the EDB repository. For information about requesting credentials, visit the <https://www.enterprisedb.com/user/login>.

Step2: Assume superuser privileges to install the EDB repository configuration package:

```
dnf -y install https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm
```

Step3: The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`. Ensure the username and password placeholders in the `baseurl` specification matches with the name and password of a registered EDB user.

Step4: Update the cache and install EDB Postgres Advanced Server:

```
dnf makecache
```

```
dnf -y install edb-as14-server
```

Step5: Use `sudo` to assume the identity of the `enterprisedb` database superuser and create an EDB Postgres Advanced Server cluster named `acctg` on listener port 5444 :

```
sudo su - enterprisedb
```

```
/usr/edb/as14/bin/initdb -D /var/lib/edb/as14/acctg
```

1.1.3 Starting the cluster

As the `enterprisedb` user, start the cluster:

```
/usr/edb/as14/bin/pg_ctl start -D /var/lib/edb/as14/acctg
```

You can check the status of the cluster with the command:

```
/usr/edb/as14/bin/pg_ctl status -D /var/lib/edb/as14/acctg
```

1.1.4 Using the psql command line client

After installing the server and initializing a cluster, you can connect to the database with your choice of client. For convenience, the server is deployed with the [pgAdmin 4 graphical client](#) and the *Using the psql command line client*.

As the `enterprisedb` user, open a psql session:

```
/usr/edb/as14/bin/psql -d edb
```

Then, assign a password to the `enterprisedb` user:

```
ALTER ROLE enterprisedb IDENTIFIED BY password;
```

Create a database (named `hr`):

```
CREATE DATABASE hr;
```

Connect to the new database and create a table (named `dept`):

```
\c hr

CREATE TABLE public.dept (deptno numeric(2) NOT NULL CONSTRAINT dept_pk
PRIMARY KEY, dname varchar(14) CONSTRAINT dept_dname_uq UNIQUE, loc
varchar(13));
```

Add data to the table:

```
INSERT INTO dept VALUES (10,ACCOUNTING,NEW YORK);

INSERT into dept VALUES (20,RESEARCH,DALLAS);
```

You can use simple SQL commands to query the database and retrieve information about the data you have added to the table:

```
SELECT * FROM dept;
__OUTPUT__
deptno |  dname   |   loc
-----+-----+-----
    10 | ACCOUNTING | NEW YORK
    20 | RESEARCH  | DALLAS
(2 rows)
```

Or create database users:

```
ALTER ROLE enterprisedb IDENTIFIED BY password;
```

For more detailed product usage information, see the EDB Postgres Advanced Server documentation available at the EDB website.

QUICK START FOR CENTOS OR RHEL 7

EDB Postgres Advanced Server adds extended functionality to the open-source PostgreSQL database that supports database administration, enhanced SQL capabilities, database and application security, performance monitoring and analysis, and application development utilities. EDB Postgres Advanced Server also supports database compatibility features for Oracle users; for detailed information about compatibility features, see the EDB Postgres Advanced Server documentation .

This guide walks you through using `yum` to install EDB Postgres Advanced Server on a RHEL or CentOS 7 system and deploying a database cluster. The database created by this tutorial is well-suited for experimentation and testing. There are additional security and resource considerations when configuring a production installation that are not covered by this document.

This guide assumes that you are familiar with simple operating system and system administration procedures, and have administrative privileges on the host on which EDB Postgres Advanced Server is installed.

2.1 Components of an EDB Postgres Advanced Server deployment

Among the components that make up an EDB Postgres Advanced Server deployment are:

The Database server - The database server (the `postmaster`) is the service that provides the key functionality that allows you to store and manage data. EDB Postgres Advanced Server is built on the PostgreSQL open-source database project; it includes all of the documented features of community PostgreSQL and more.

The database cluster - A cluster is a set of on-disk structures that comprise a collection of databases. A cluster is serviced by a single-instance of the database server. A database cluster is stored in the `data` directory. Don't store the `data` directory of a production database on an NFS file system.

Configuration files - You can use the parameters listed in Postgres configuration files to manage deployment preferences, security preferences, connection behaviors, and logging preferences.

Supporting tools, utilities, and clients - EDB makes available a full suite of tools and utilities that can help you monitor and manage your EDB Postgres Advanced Server deployment. For more information, visit the

[EDB website](#) .

Supporting functions, rocedures, data types, index types, operators, utilities, and aggregates - EDB Postgres Advanced Server includes a number of features that help you manage your data.

2.2 Prerequisites

Before installing EDB Postgres Advanced Server, use yum to install prerequisite packages:

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

If you are installing Migration Toolkit or EDB*Plus, you must first install Java:

```
yum -y install java
```

2.3 Installing and configuring EDB Postgres Advanced Server

Step1: You must register with EDB to receive credentials for the EDB repository. For information about requesting credentials, visit the <https://www.enterprisedb.com/user/login>.

Step2: Assume superuser privileges to install the EDB repository configuration package:

```
yum -y install https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm
```

Step3: The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`. Ensure the username and password placeholders in the `baseurl` specification matches with the name and password of a registered EDB user.

Step4: Update the cache and install EDB Postgres Advanced Server:

```
yum makecache  
  
yum -y install edb-as14-server
```

Step5: Use `sudo` to assume the identity of the `enterprisedb` database superuser and create an EDB Postgres Advanced Server cluster named `acctg` on listener port 5444 :

```
sudo su - enterprisedb  
  
/usr/edb/as14/bin/initdb -D /var/lib/edb/as14/acctg
```

2.4 Starting the cluster

As the `enterprisedb` user, start the cluster:

```
/usr/edb/as14/bin/pg_ctl start -D /var/lib/edb/as14/acctg
```

You can check the status of the cluster with the command:

```
/usr/edb/as14/bin/pg_ctl status -D /var/lib/edb/as14/acctg
```

2.5 Using the psql command line client

After installing the server and initializing a cluster, you can connect to the database with your choice of client. For convenience, the server is deployed with the [pgAdmin 4 graphical client](#) and the *Using the psql command line client*.

As the `enterprisedb` user, open a psql session:

```
/usr/edb/as14/bin/psql -d edb -p 5444
```

Then, assign a password to the `enterprisedb` user:

```
ALTER ROLE enterprisedb IDENTIFIED BY password;
```

Create a database (named `hr`):

```
CREATE DATABASE hr;
```

Connect to the new database and create a table (named `dept`):

```
\c hr

CREATE TABLE public.dept (deptno numeric(2) NOT NULL CONSTRAINT dept_pk
PRIMARY KEY, dname varchar(14) CONSTRAINT dept_dname_uq UNIQUE, loc
varchar(13));
```

Add data to the table:

```
INSERT INTO dept VALUES (10,ACCOUNTING,NEW YORK);

INSERT into dept VALUES (20,RESEARCH,DALLAS);
```

You can use simple SQL commands to query the database and retrieve information about the data you have added to the table:

```
SELECT * FROM dept;
__OUTPUT__
deptno |  dname   |  loc
-----+-----+-----
    10 | ACCOUNTING | NEW YORK
    20 | RESEARCH  | DALLAS
(2 rows)
```

Or, create database users:

```
ALTER ROLE enterprisedb IDENTIFIED BY password;
```

For detailed product usage information, see the EDB Postgres Advanced Server documentation available at the EDB website.

QUICK START FOR WINDOWS

EDB Postgres Advanced Server adds extended functionality to the open-source PostgreSQL database that supports database administration, enhanced SQL capabilities, database and application security, performance monitoring and analysis, and application development utilities. EDB Postgres Advanced Server also supports database compatibility features for Oracle users; for detailed information about compatibility features, see the EDB Postgres Advanced Server documentation.

This guide walks you through using the graphical installer to install EDB Postgres Advanced Server on a Windows system. The database created by this tutorial is well-suited for experimentation and testing. There are likely to be additional security and resource considerations when configuring a production installation that are not covered by this document.

This guide assumes that you are familiar with simple operating system and system administration procedures, and have administrative privileges on the host on which EDB Postgres Advanced Server is installed.

3.1 Components of an EDB Postgres Advanced Server deployment

Among the components that make up an EDB Postgres Advanced Server deployment are:

The database server - The database server (the `postmaster`) is the service that provides the key functionality that allows you to store and manage data. EDB Postgres Advanced Server is built on the PostgreSQL open-source database project; it includes all of the documented features of community PostgreSQL and more.

The database cluster - A cluster is a set of on-disk structures that comprise a collection of databases. A cluster is serviced by a single-instance of the database server. A database cluster is stored in the `data` directory. Don't store the `data` directory of a production database on an NFS file system.

Configuration files - You can use the parameters listed in Postgres configuration files to manage deployment preferences, security preferences, connection behaviors, and logging preferences.

Supporting tools, utilities, and clients - EDB makes available a full suite of tools and utilities that can help you monitor and manage your EDB Postgres Advanced Server deployment. For more information, visit the

[EDB website](#).

Supporting functions, procedures, data types, index types, operators, utilities, and aggregates - EDB Postgres Advanced Server includes a number of features that help you manage your data.

3.2 Installation prerequisites

3.2.1 User privileges

To perform an EDB Postgres Advanced Server installation on a Windows system, you must have administrator privileges. If you are installing EDB Postgres Advanced Server on a Windows system that is configured with `User Account Control` enabled, you can assume sufficient privileges to invoke the graphical installer by right clicking on the name of the installer and selecting `Run as administrator` from the context menu.

3.2.2 Windows-specific software requirements

You should apply Windows operating system updates before invoking the EDB Postgres Advanced Server installer. If (during the installation process) the installer encounters errors, exit the installation, and ensure that your version of Windows is up-to-date before restarting the installer.

3.2.3 Migration Toolkit or EDB*Plus installation Prerequisites

Before using StackBuilder Plus to install Migration Toolkit or EDB*Plus, you must first install Java (version 1.8 or later). If you are using Windows, Java installers and instructions are available online at:

<http://www.java.com/en/download/manual.jsp>

3.2.4 Installing and configuring EDB Postgres Advanced Server

The graphical installer provides a quick and easy way to install EDB Postgres Advanced Server 15 on a Windows system. Use the wizard's dialogs to specify information about your system and system usage; when you have completed the dialogs, the installer performs an installation based on the selections made during the setup process.

To invoke the wizard, you must have administrator privileges. Assume administrator privileges, and double-click the `edb-as15-server-15.x.x-x-windows-x64` executable file.

Note: To install EDB Postgres Advanced Server on some versions of Windows, you may be required to right click on the file icon and select *Run as Administrator* from the context menu to invoke the installer with *Administrator* privileges.

When the `Language Selection` popup opens, select an installation language and click `OK` to continue to the `Setup` window.

Fig. 1: The EDB Postgres Advanced Server installer Welcome window

Click `Next` to continue.

The `EnterpriseDB License Agreement` opens.

Fig. 2: The EnterpriseDB License Agreement

Carefully review the license agreement before highlighting the appropriate radio button; click `Next` to continue.

The `Installation Directory` window opens.

Fig. 3: The Installation Directory window

By default, the EDB Postgres Advanced Server installation directory is:

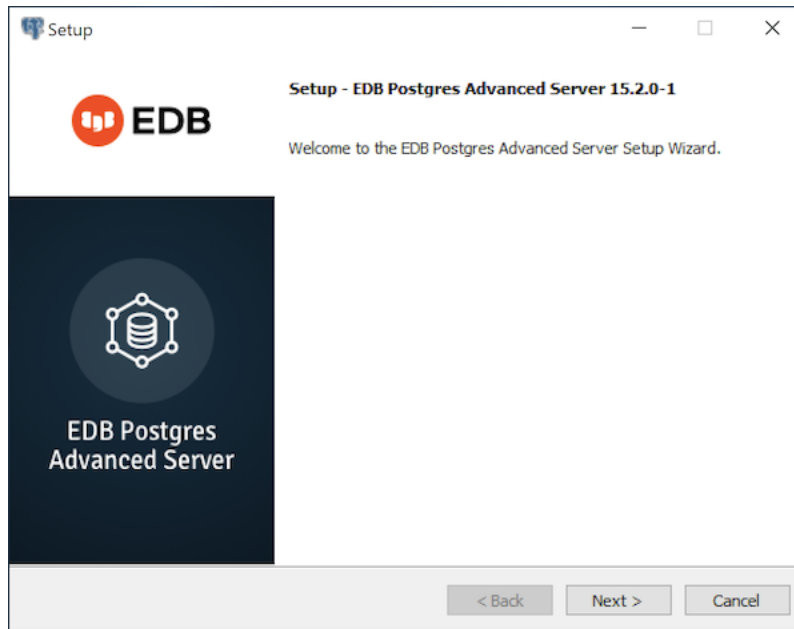


Fig. 1: The EDB Postgres Advanced Server installer Welcome window

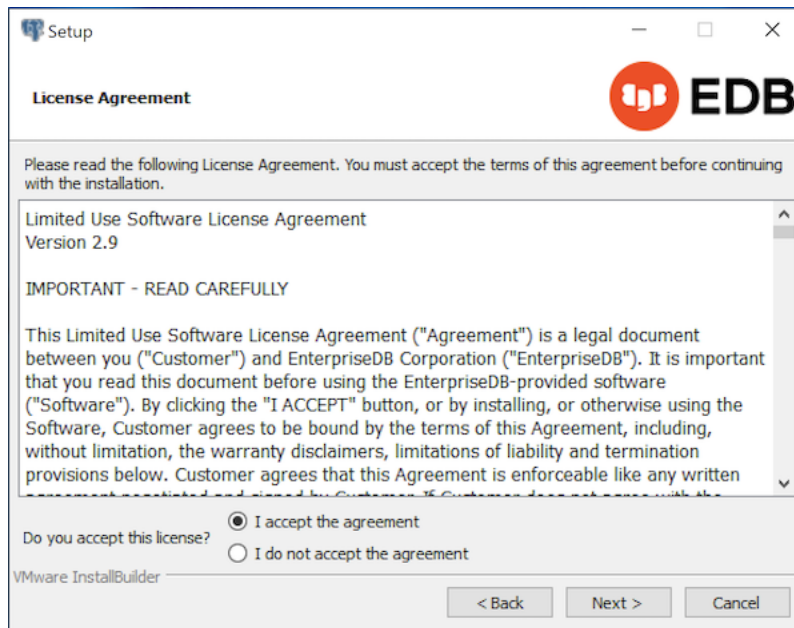


Fig. 2: The EnterpriseDB License Agreement

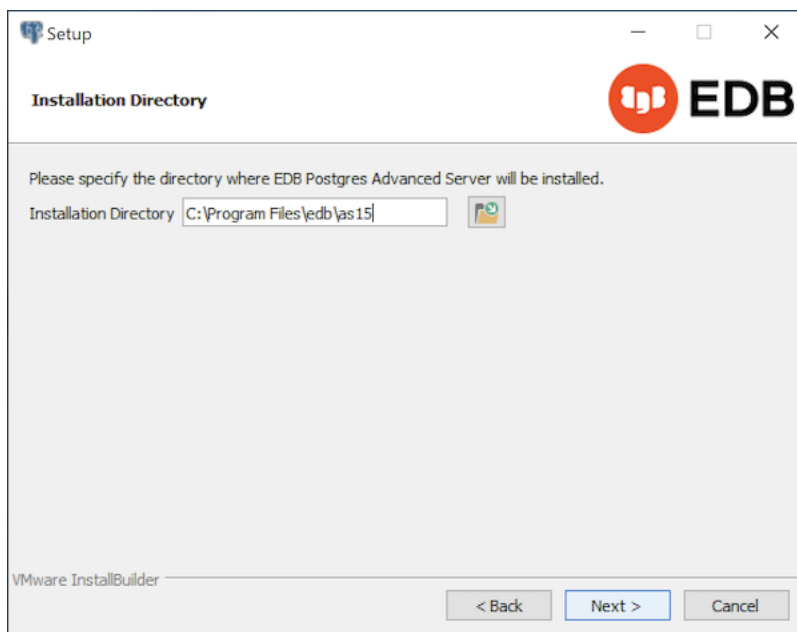


Fig. 3: The Installation Directory window

C:\Program Files\edb\as15

You can accept the default installation location, and click **Next** to continue, or optionally click the **File Browser** icon to open the **Browse For Folder** dialog to choose an alternate installation directory.

Note: The *data* directory of a production database should not be stored on an NFS file system.

Fig. 4: The Select Components window

The **Select Components** window contains a list of optional components that you can install with the EDB PostgreSQL Advanced Server Setup wizard. You can omit a module from the EDB PostgreSQL Advanced Server installation by deselecting the box next to the components name.

The Setup wizard can install the following components while installing EDB PostgreSQL Advanced Server 15:

EDBPostgresAdvancedServer

Select the **EDB PostgreSQL Advanced Server** option to install EDB PostgreSQL Advanced Server 15.

StackBuilderPlus

The **StackBuilder Plus** utility is a graphical tool that can update installed products, or download and add supporting modules (and the resulting dependencies) after your EDB PostgreSQL Advanced Server setup and installation completes.

CommandLineTools

The **Command Line Tools** option installs command line tools and supporting client libraries including:

- libpq
- psql
- EDB*Loader
- ecpgPlus

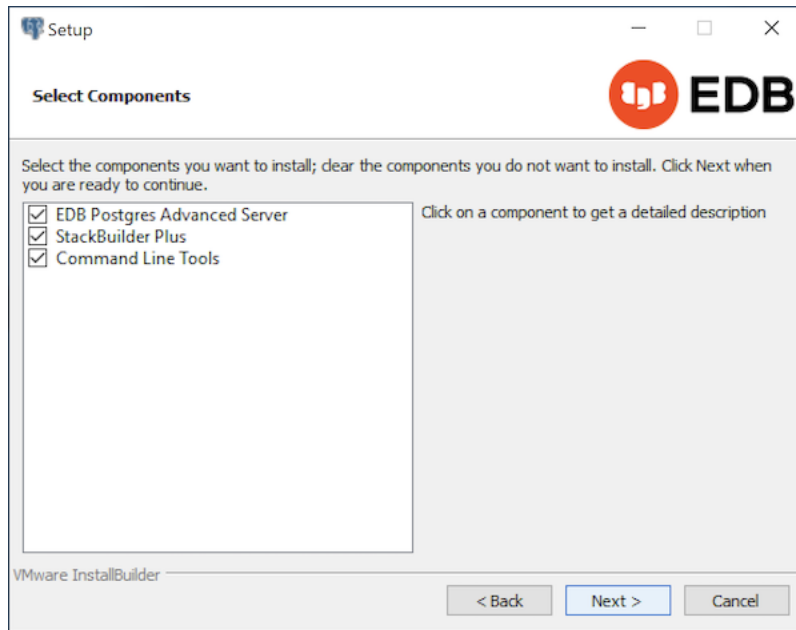


Fig. 4: The Select Components window

- pg_basebackup, pg_dump, and pg_restore
- pg_bench
- and more.

Note: The *Command Line Tools* are required if you are installing EDB Postgres Advanced Server or pgAdmin 4.

After selecting the components you wish to install, click **Next** to open the **Additional Directories** window.

Fig. 5: The Additional Directories window

By default, the EDB Postgres Advanced Server data files are saved to:

C:\Program Files\edb\as15\data

The default location of the EDB Postgres Advanced Server Write-Ahead Log (WAL) Directory is:

C:\Program Files\edb\as15\data\pg_wal

EDB Postgres Advanced Server uses write-ahead logs to promote transaction safety and speed transaction processing; when you make a change to a table, the change is stored in shared memory and a record of the change is written to the write-ahead log. When you perform a COMMIT, EDB Postgres Advanced Server writes contents of the write-ahead log to disk.

Accept the default file locations, or use the **File Browser** icon to select an alternate location; click **Next** to continue to the **EDB Postgres Advanced Server Dialect** window.

Fig. 6: The EDB Postgres Advanced Server Dialect window

Use the drop-down listbox on the **EDB Postgres Advanced Server Dialect** window to choose a server dialect. The server dialect specifies the compatibility features supported by EDB Postgres Advanced Server.

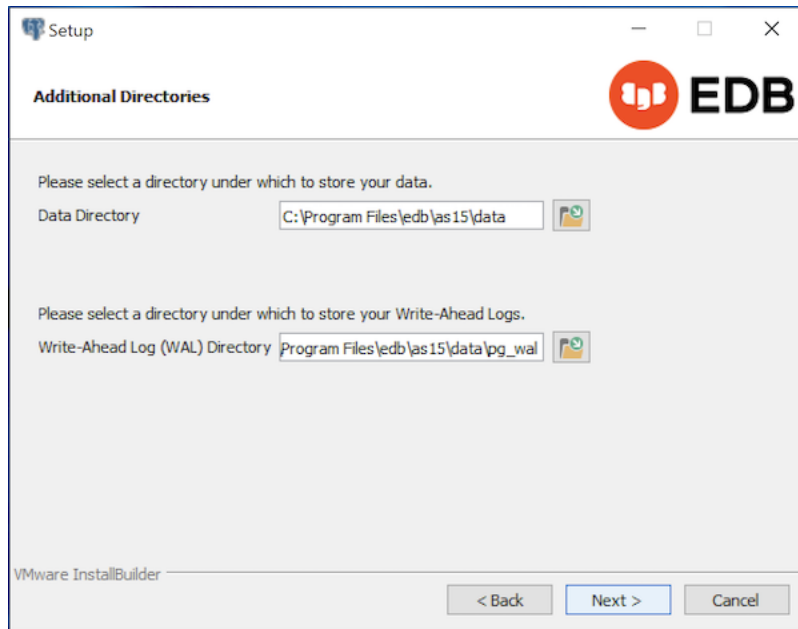


Fig. 5: The Additional Directories window

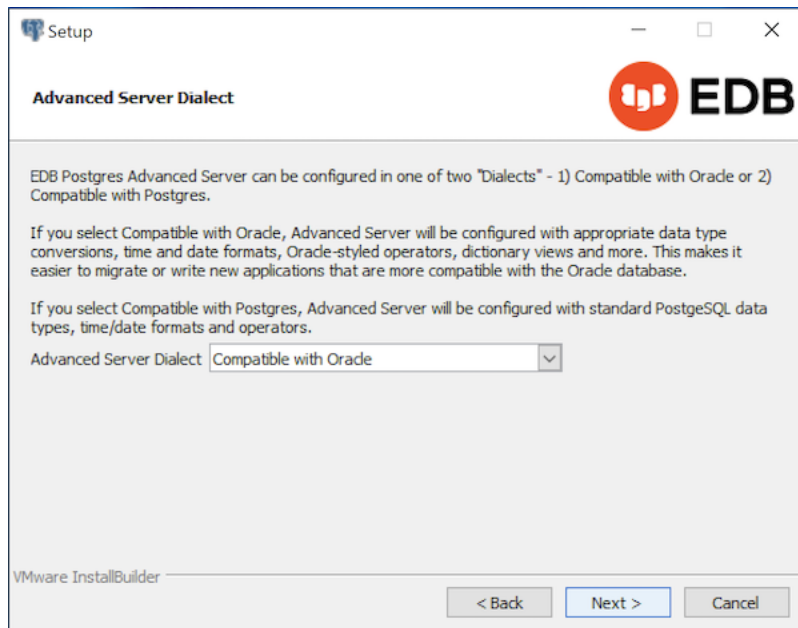


Fig. 6: The EDB Postgres Advanced Server Dialect window

By default, EDB Postgres Advanced Server installs in `Compatible with Oracle` mode; you can choose between `Compatible with Oracle` and `Compatible with PostgreSQL` installation modes.

CompatiblewithOracle

If you select `Compatible with Oracle`, the installation includes the following features:

- Data dictionary views that is compatible with Oracle databases.
- Oracle data type conversions.
- Date values displayed in a format compatible with Oracle syntax.
- Support for Oracle-styled concatenation rules (if you concatenate a string value with a NULL value, the returned value is the value of the string).
- Schemas (`dbo` and `sys`) compatible with Oracle databases added to the `SEARCH_PATH`.
- Support for the following Oracle built-in packages:

Package	Functionality compatible with Oracle databases
<i>dbms_alert</i>	Provides the capability to register for, send, and receive alerts.
<i>dbms_job</i>	Provides the capability for the creation, scheduling, and managing of jobs.
<i>dbms_lob</i>	Provides the capability to manage on large objects.
<i>dbms_output</i>	Provides the capability to send messages to a message buffer, or get messages from the message buffer.
<i>dbms_pipe</i>	Provides the capability to send messages through a pipe within or between sessions connected to the same database cluster.
<i>dbms_rls</i>	Enables the implementation of Virtual Private Database on certain EDB Postgres Advanced Server database objects.
<i>dbms_sql</i>	Provides an application interface to the EDB dynamic SQL functionality.
<i>dbms_utility</i>	Provides various utility programs.
<i>dbms_aqadm</i>	Provides supporting procedures for Advanced Queueing functionality.
<i>dbms_aq</i>	Provides message queueing and processing for EDB Postgres Advanced Server.
<i>dbms_profiler</i>	Collects and stores performance information about the PL/pgSQL and SPL statements that are executed during a performance profiling session.
<i>dbms_random</i>	Provides a number of methods to generate random values.
<i>dbms_redact</i>	Enables the redacting or masking of data that is returned by a query.
<i>dbms_lock</i>	Provides support for the <i>DBMS_LOCK.SLEEP</i> procedure.
<i>dbms_scheduler</i>	Provides a way to create and manage jobs, programs, and job schedules.
<i>dbms_crypto</i>	Provides functions and procedures to encrypt or decrypt RAW, BLOB or CLOB data. You can also use <i>DBMS_CRYPTO</i> functions to generate cryptographically strong random values.
<i>dbms_mview</i>	Provides a way to manage and refresh materialized views and their dependencies.
<i>dbms_session</i>	Provides support for the <i>DBMS_SESSION.SET_ROLE</i> procedure.
<i>utl_encode</i>	Provides a way to encode and decode data.
<i>utl_http</i>	Provides a way to use the HTTP or HTTPS protocol to retrieve information found at an URL.
<i>utl_file</i>	Provides the capability to read from, and write to files on the operating system's file system.
<i>utl_smtp</i>	Provides the capability to send e-mails over the Simple Mail Transfer Protocol (SMTP).
<i>utl_mail</i>	Provides the capability to manage e-mail.
<i>utl_url</i>	Provides a way to escape illegal and reserved characters within an URL.
<i>utl_raw</i>	Provides a way to manipulate or retrieve the length of raw data types.

This is not a comprehensive list of the compatibility features for Oracle included when EDB Postgres Advanced Server is installed in `Compatible with Oracle` mode. For more information, see [Database compatibility for Oracle developers: built-in packages](#).

If you choose to install in `Compatible with Oracle` mode, the EDB Postgres Advanced Server superuser name is `enterprisedb`.

CompatiblewithPostgreSQL

If you select `Compatible with PostgreSQL`, EDB Postgres Advanced Server exhibits compatibility with PostgreSQL version 15. If you choose to install in `Compatible with PostgreSQL` mode, the default EDB Postgres Advanced Server superuser name is `postgres`.

For detailed information about PostgreSQL functionality, visit the official [PostgreSQL website](#).

After specifying a configuration mode, click **Next** to continue to the **Password** window.

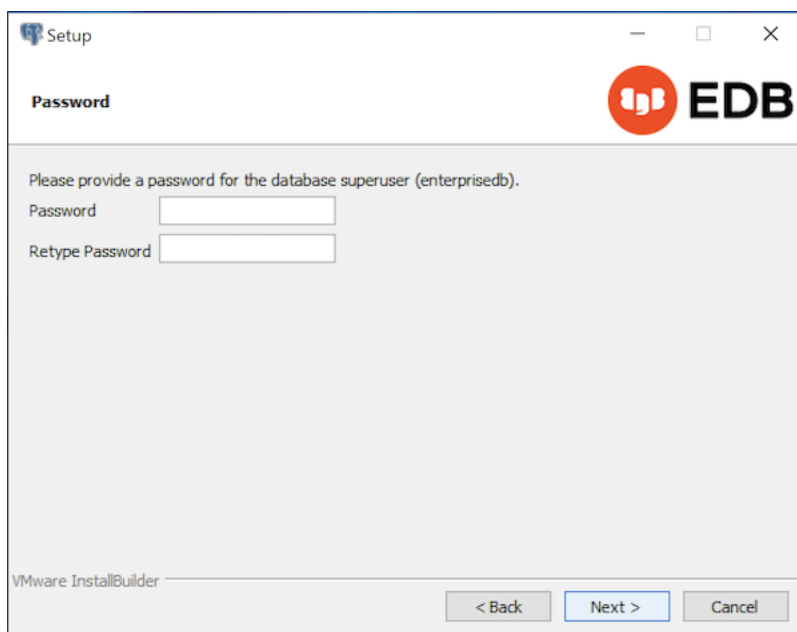


Fig. 7: The Password window

Fig. 7: The Password window

EDB Postgres Advanced Server uses the password specified on the **Password** window for the database superuser. The specified password must conform to any security policies existing on the EDB Postgres Advanced Server host.

After you enter a password in the **Password** field, confirm the password in the **Retype Password** field, and click **Next** to continue.

The **Additional Configuration** window opens.

Fig. 8: The Additional Configuration window

Use the fields on the **Additional Configuration** window to specify installation details:

- Use the **Port** field to specify the port number that EDB Postgres Advanced Server should listen to for connection requests from client applications. The default is 5444.

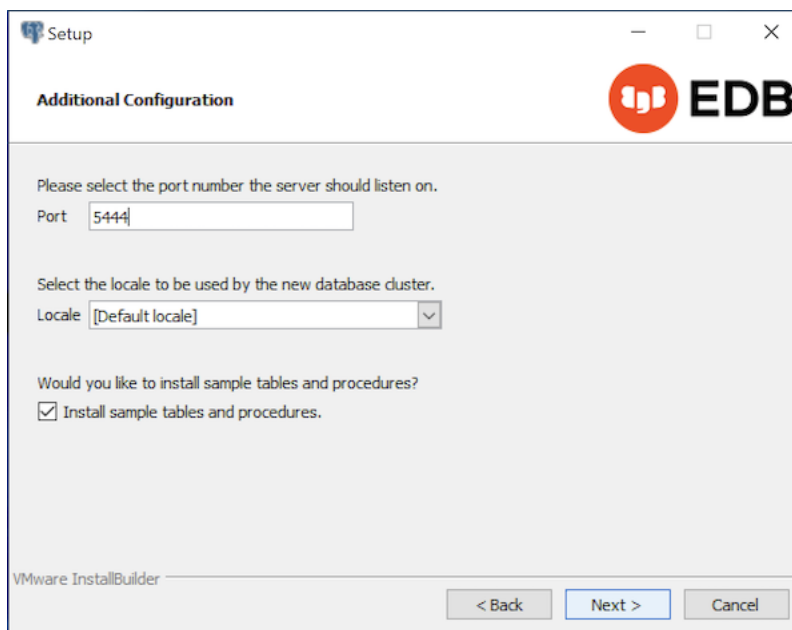


Fig. 8: The Additional Configuration window

- If the **Locale** field is set to `[Default locale]`, EDB Postgres Advanced Server uses the system locale as the working locale. Use the drop-down listbox next to **Locale** to specify an alternate locale for EDB Postgres Advanced Server.
- By default, the Setup wizard installs corresponding sample data for the server dialect specified by the compatibility mode (Oracle or PostgreSQL). Clear the check box next to **Install sample tables and procedures** if you don't wish to have sample data installed.

After verifying the information on the **Additional Configuration** window, click **Next** to open the **Dynatune Dynamic Tuning: Server Utilization** window.

The graphical Setup wizard facilitates performance tuning via the Dynatune Dynamic Tuning feature. Dynatune functionality allows EDB Postgres Advanced Server to make optimal usage of the system resources available on the host machine on which it is installed.

Fig. 9: The Dynatune Dynamic Tuning: Server Utilization window

The `edb_dynatune` configuration parameter determines how EDB Postgres Advanced Server allocates system resources. Use the radio buttons on the **Server Utilization** window to set the initial value of the `edb_dynatune` configuration parameter:

- Select **Development** to set the value of `edb_dynatune` to 33. A low value dedicates the least amount of the host machine's resources to the database server. This is a good choice for a development machine.
- Select **General Purpose** to set the value of `edb_dynatune` to 66. A mid-range value dedicates a moderate amount of system resources to the database server. This would be a good setting for an application server with a fixed number of applications running on the same host as EDB Postgres Advanced Server.
- Select **Dedicated** to set the value of `edb_dynatune` to 100. A high value dedicates most of the system resources to the database server. This is a good choice for a dedicated server host.

After the installation is complete, you can adjust the value of `edb_dynatune` by editing the `postgresql.conf` file, located in the data directory of your EDB Postgres Advanced Server installation. After editing the `postgresql.conf` file, you must restart the server for your changes to take effect.

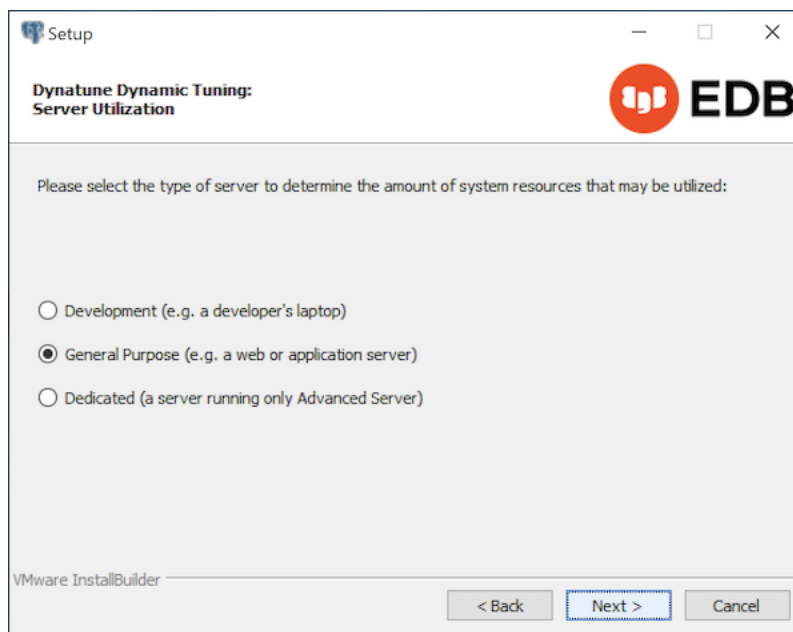


Fig. 9: The Dynatune Dynamic Tuning: Server Utilization window

Select the appropriate setting for your system, and click **Next** to continue to the **Dynatune Dynamic Tuning: Workload Profile** window.

Fig. 10: The Dynatune Dynamic Tuning: Workload Profile window

Use the radio buttons on the **Workload Profile** window to specify the initial value of the `edb_dynatune_profile` configuration parameter. The `edb_dynatune_profile` parameter controls performance-tuning aspects based on the type of work that the server performs.

- Select **Transaction Processing (OLTP systems)** to specify an `edb_dynatune_profile` value of `oltp`. Recommended when EDB Postgres Advanced Server is supporting heavy online transaction processing.
- Select **General Purpose (OLTP and reporting workloads)** to specify an `edb_dynatune_profile` value of `mixed`. Recommended for servers that provide a mix of transaction processing and data reporting.
- Select **Reporting (Complex queries or OLAP workloads)** to specify an `edb_dynatune_profile` value of `reporting`. Recommended for database servers used for heavy data reporting.

After the installation is complete, you can adjust the value of `edb_dynatune_profile` by editing the `postgresql.conf` file, located in the `data` directory of your EDB Postgres Advanced Server installation. After editing the `postgresql.conf` file, you must restart the server for your changes to take effect.

For more information about `edb_dynatune` and other performance-related topics, see the **Dynatune**.

By default, EDB Postgres Advanced Server is configured to start the service when the system boots. Click **Next** to continue.

The **Pre Installation Summary** opens.

Fig. 12: The Pre Installation Summary

The **Pre Installation Summary** provides an overview of the options specified during the **Setup** process. Review the options before clicking **Next**; click **Back** to navigate back through the dialogs and update any options.

The **Ready to Install** window confirms that the installer has the information it needs about your configuration preferences to install EDB Postgres Advanced Server. Click **Next** to continue.

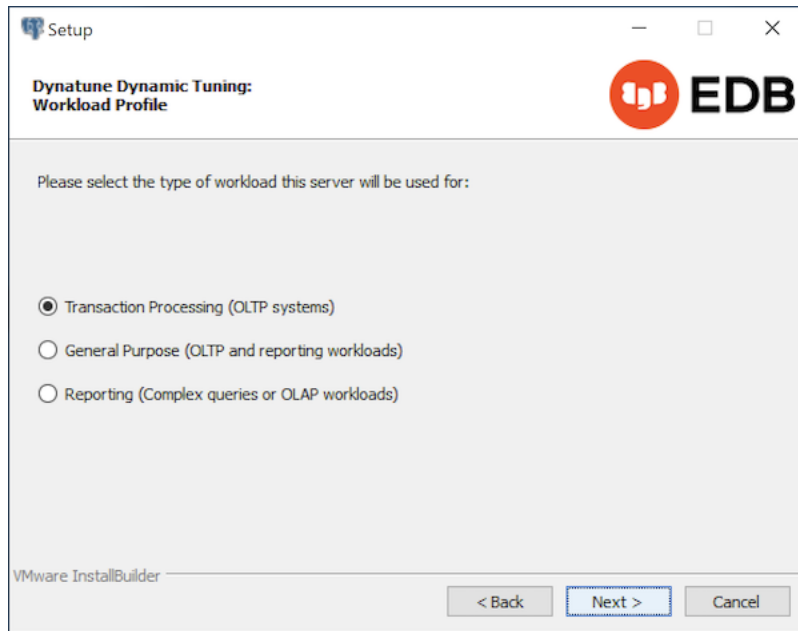


Fig. 10: The Dynatune Dynamic Tuning: Workload Profile window

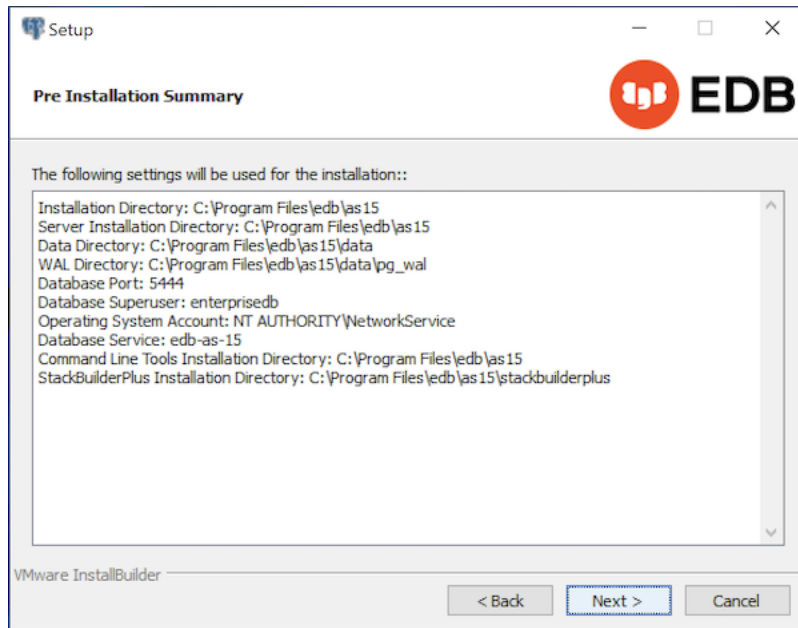


Fig. 11: The Pre Installation Summary

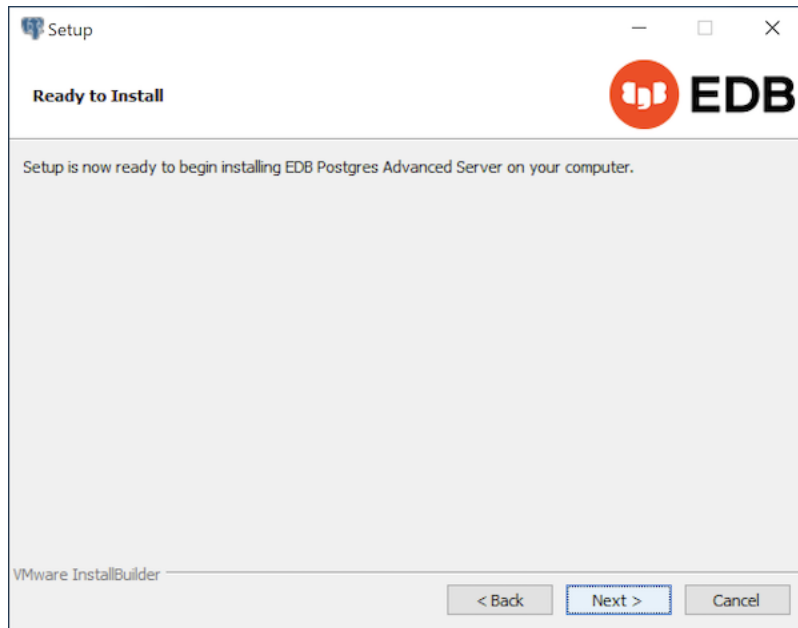


Fig. 12: The Ready to Install window

Fig. 13: The Ready to Install window

Fig. 14: Installing EDB Postgres Advanced Server

As each supporting module is unpacked and installed, the module's installation is confirmed with a progress bar.

Before the Setup wizard completes the EDB Postgres Advanced Server installation, it offers to Launch StackBuilder Plus at exit?

Fig. 15: The Setup wizard offers to Launch StackBuilder Plus at exit

You can clear the StackBuilder Plus check box and click Finish to complete the EDB Postgres Advanced Server installation, or accept the default and proceed to StackBuilder Plus.

EDB Postgres StackBuilder Plus is included with the installation of EDB Postgres Advanced Server and its core supporting components. StackBuilder Plus is a graphical tool that can update installed products, or download and add supporting modules (and the resulting dependencies) after your EDB Postgres Advanced Server setup and installation completes.

For more detailed product usage information, see the EDB Postgres Advanced Server documentation available at the EDB website.

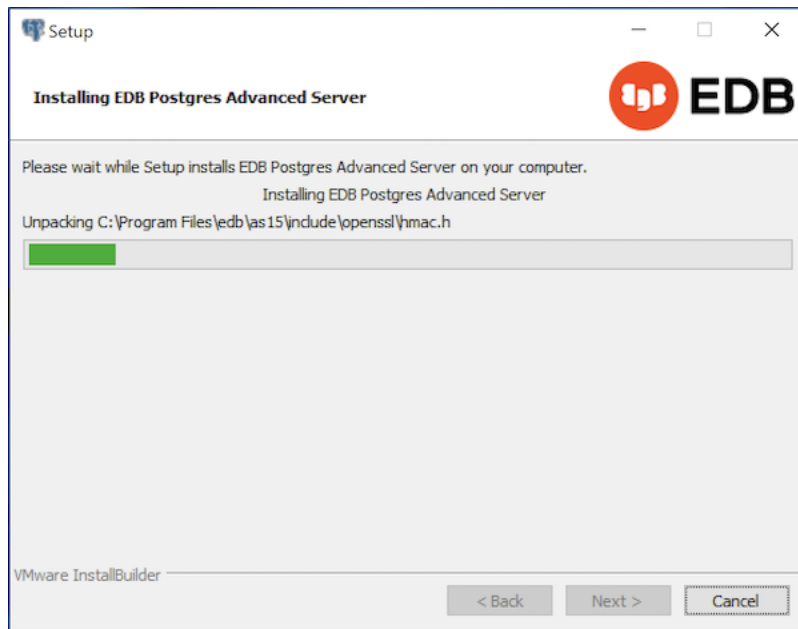


Fig. 13: Installing EDB Postgres Advanced Server

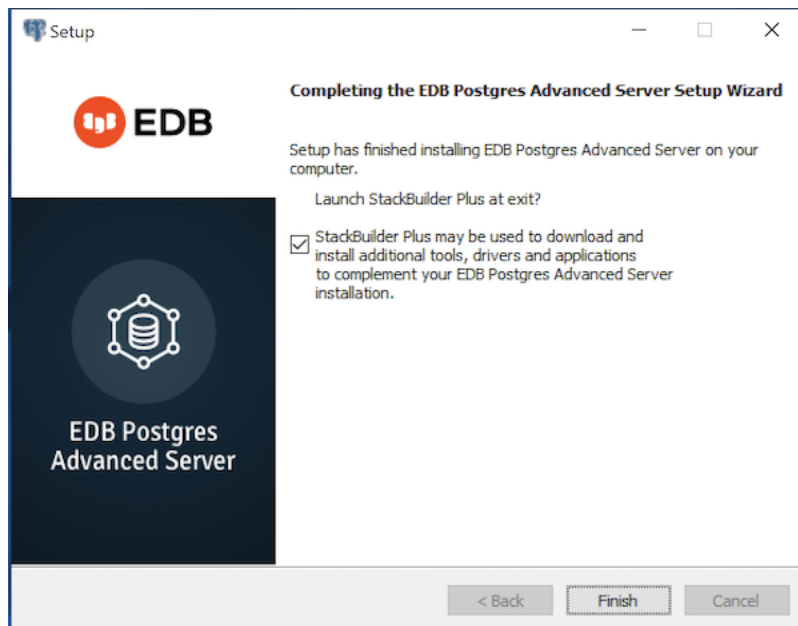


Fig. 14: The Setup wizard offers to Launch StackBuilder Plus at exit

EDB PGADMIN4 QUICK START LINUX

pgAdmin 4 is the leading open source management tool for Postgres databases. EDB pgAdmin 4 is distributed by EDB along with EDB Postgres Advanced Server databases. It's designed to meet the needs of both novice and experienced Postgres users, providing a powerful graphical interface that simplifies creating, maintaining, and using database objects.

You can install EDB pgAdmin 4 for your EDB Postgres Advanced Server databases using yum package manager for RHEL/CentOS 7.x or 8.x platforms.

4.1 Installing EDB pgAdmin 4 on a Linux host

Use the yum package manager to install EDB pgAdmin4.

4.1.1 Create a repository configuration file

Create the repository configuration file if it doesn't already exist.

To create a repository configuration file, you need the credentials that allow access to the EDB repository. For information about requesting credentials, see [EDB Repository Access](#).

To create the repository configuration file, assume superuser privileges and invoke the following command:

```
yum -y install https://yum.enterprisedb.com/edb-repo-rpms/edb-repo-latest.noarch.rpm
```

The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`. After creating the `edb.repo` file, use the following command to replace the `USERNAME` and `PASSWORD` placeholders in the baseurl specification with the username and password of a registered EDB user:

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

4.1.2 Install EPEL repository

For CentOS 7.x, use the following command:

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

Note: To install EPEL repository on CentOS/RHEL 7.x and 8.x see the platform specific steps at [EDB Website](#).

4.1.3 Install EDB pgAdmin 4

After creating the repository configuration file and adding a username and password to the `edb.repo` file, you can install `edb-pgadmin4`. To install `edb-pgadmin4`, assume superuser privileges and invoke the following command:

```
yum install edb-pgadmin4*
```

This command installs following packages:

- `edb-pgadmin4`
- `edb-pgadmin4-desktop-common`
- `edb-pgadmin4-desktop-gnome`
- `edb-pgadmin4-docs`
- `edb-pgadmin4-web`

4.1.4 Starting pgAdmin 4 in desktop mode

Use the following command to start pgAdmin 4 in desktop mode:

```
/usr/edb/pgadmin4/bin/pgAdmin4
```

You can also use the link on the **Applications** menu to start pgAdmin 4 in desktop mode:

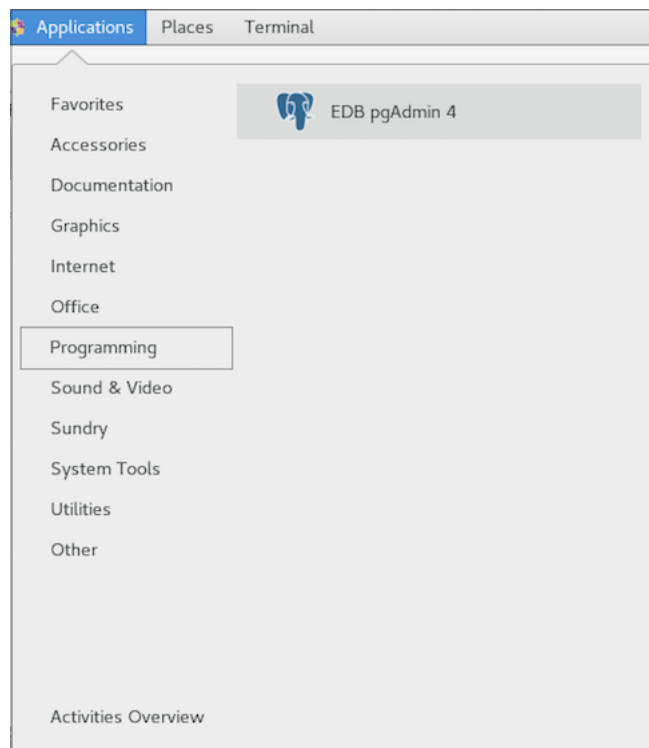


Fig. 1: Accessing pgAdmin 4 from Applications Menu.

4.1.5 Registering and connecting to EDB Postgres Advanced Server with pgAdmin 4

Before managing an EDB Postgres Advanced Server cluster, you must register the server. To register the server, use the fields on the Server dialog box to specify the connection properties. To open the Server dialog box, right-click the Servers node in the tree, and select **Create>Server** .

For detailed information about registering your server, see the
[pgAdmin documentation](#) .

Then, to connect to your EDB Postgres Advanced Server instance, right-click the server name and select **Connect-Server** . Provide your password in the Connect to Server dialog box.

After you connect to the server, you can see the **Dashboard** tab.