



EDB Postgres™ Advanced Server

Release 13

Quick Start Guide for Linux on CentOS or RHEL 8

Nov 29, 2020

Contents

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. Advanced Server also supports database compatibility features for Oracle users; for detailed information about compatibility features, see the Advanced Server documentation.

This guide will walk you through using `dnf` to install EDB Postgres Advanced Server on a RHEL or CentOS 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 Advanced Server will be installed.

Please note that if you are using the pdf version of this document, using cut/paste to copy a command may result in extra spaces or carriage returns in the pasted command. If a command fails, check the command carefully for extra characters.

Components of an EDB Postgres Advanced Server Deployment

Among the components that make up an 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. 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; please note that the `data` directory of a production database should not be stored 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 Advanced Server deployment. For more information, visit the [EDB website](#).

Supporting Functions, Procedures, Data Types, Index Types, Operators, Utilities, and Aggregates - Advanced Server includes a number of features that help you manage your data.

Prerequisites

Before installing 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
```

Installing and Configuring Advanced Server

Step 1: You must register with EDB to receive credentials for the EDB repository. If you have not previously registered with EDB, visit the [EDB site for information about requesting credentials](#).

Step 2: Assume superuser privileges to install the EDB repository configuration package:

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

Step 3: With your choice of editor, edit the `/etc/yum.repos.d/edb.repo` file, replacing the `username` and `password` placeholders in the `baseurl` specification with the name and password of a registered EDB user and ensure that the `gpgcheck` and `enabled` properties are set to 1.

Step 4: Update the cache and install Advanced Server:

```
dnf makecache  
  
dnf -y install edb-as13-server
```

Step 5: Use `sudo` to assume the identity of the `enterprisedb` database superuser and create an Advanced Server cluster named `acctg` on listener port 5444:

```
sudo su - enterprisedb

/usr/edb/as13/bin/initdb -D /var/lib/edb/as13/acctg
```

Starting the Cluster

As the enterprisedb user, start the cluster:

```
/usr/edb/as13/bin/pg_ctl start -D /var/lib/edb/as13/acctg
```

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

```
/usr/edb/as13/bin/pg_ctl status -D /var/lib/edb/as13/acctg
```

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 [psql command line client](#).

As the enterprisedb user, open a psql session:

```
/usr/edb/as13/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;
 deptno |   dname   |   loc
-----+-----+-----
```

(continues on next page)

(continued from previous page)

```
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 [Advanced Server documentation](#), available at the [EDB website](#).