EDB Postgres™ Advanced Server

Release 13

Installation Guide for Linux

Nov 05, 2020

Contents

CHAPTER 1

Introduction

The *EDB Postgres Advanced Server Installation Guide* is a comprehensive guide to installing EDB Postgres Advanced Server (Advanced Server). In this guide you will find detailed information about:

- Software prerequisites for performing an Advanced Server 13 installation on a Linux host.
- Using a package manager to install and update Advanced Server and its supporting components or utilities on a Linux host.
- Managing an Advanced Server installation.
- Configuring an Advanced Server package installation.
- Uninstalling Advanced Server and its components.

CHAPTER 2

Supported Platforms

For information about the platforms and versions supported by Advanced Server, visit the EDB website at:

https://www.enterprisedb.com/product-compatibility

Limitations

The following limitations apply to EDB Postgres Advanced Server:

- The `data` directory of a production database should not be stored on an NFS file system.
- The LLVM JIT package is supported on RHEL or CentOS 7.x or 8.x only. LLVM JIT is not supported on PPC-LE 64 running RHEL or CentOS 7.x.

CHAPTER 3

Using a Package Manager to Install Advanced Server

You can use the `dnf` or `yum` package manager to install Advanced Server or Advanced Server supporting components. `dnf` or `yum` will attempt to satisfy package dependencies as it installs a package, but requires access to the Advanced Server repositories. If your system does not have access to a repository via the Internet, you can use RPM to install an individual package or create a local repository, but you may be required to manually satisfy package dependencies.

You can list the dependencies of a package by running the following command:

- On Fedora | RHEL | CentOS: `repoquery --requires --resolve <package_name>`
- On Debian | Ubuntu: `apt-cache depends <package_name>`

Where, `package_name` is the name of the package that you want to install.

Installing the server package creates a database superuser named `enterprisedb`. The user is assigned a user ID (UID) and a group ID (GID) of `26`. The user has no default password; use the `passwd` command to assign a password for the user. The default shell for the user is `bash`, and the user's home directory is `/var/lib/edb/as13`.

By default, Advanced Server logging is configured to write files to the `log` subdirectory of the `data` directory, rotating the files each day and retaining one week of log entries. You can customize the logging behavior of the server by modifying the `postgresql.conf` file. For more information about `Modifying the postgresql.conf File`, see the *EDB Postgres Advanced Server Guide* available at:

https://www.enterprisedb.com/edb-docs

The RPM installers place Advanced Server components in the directories listed in the table below:

Component	Location
Executables	/usr/edb/as13/bin
Libraries	/usr/edb/as13/lib
Cluster configuration files	/etc/edb/as13
Documentation	/usr/edb/as13/share/doc
Contrib	/usr/edb/as13/share/contrib
Data	/var/lib/edb/as13/data
Logs	/var/log/as13
Lock files	/var/lock/as13
Log rotation file	/etc/logrotate.d/as13
Sudo configuration file	/etc/sudoers.d/as13
Binary to access VIP without sudo	/usr/edb/as13/bin/secure
Backup area	/var/lib/edb/as13/backups
Templates	/usr/edb/as13/share
Procedural Languages	/usr/edb/as13/lib or /usr/edb/as13/lib64
Development Headers	/usr/edb/as13/include
Shared data	/usr/edb/as13/share
Regression tests	/usr/edb/as13/lib/pgxs/src/test/regress
SGML Documentation	/usr/edb/as13/share/doc

3.1 Installation Pre-requisites

Before using an RPM package to install Advanced Server on a Linux host, you must:

Install Linux-specific Software

You must install `xterm, konsole`, or `gnome-terminal` before executing any console-based program installed by EDB installers.

Install Migration Toolkit or EDB*Plus Installation Prerequisites (Optional)

Before using an RPM to install Migration Toolkit or EDB*Plus, you must first install Java version 1.8 or later. On a Linux system, you can use the `dnf` or `yum` package manager to install Java. Open a terminal window, assume superuser privileges, and enter:

- On RHEL or CentOS 7:

```
# yum -y install java
```

- On RHEL or CentOS 8:

```
# dnf -y install java
```

Follow the onscreen instructions to complete the installation.

Request Credentials to the EDB Repository

Before installing the repository configuration file, you must have credentials that allow access to the EDB repository. For information about requesting credentials, visit the EDB website at:

https://www.enterprisedb.com/user/login

After receiving your repository credentials you can:

- Create the repository configuration file.
- Modify the file, providing your user name and password.
- Install the repository keys and additional prerequisite software.
- Install Advanced Server and supporting components.

3.1.1 Installing Advanced Server on a CentOS Host

You can use an RPM package to install Advanced Server on a CentOS host.

- To install the repository configuration file, assume superuser privileges and invoke one of the following platform specific commands:

  On CentOS 7:

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

  On CentOS 8:

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

- Replace the `USERNAME:PASSWORD` variable in the following command with the username and password of a registered EDB user:

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

- Before installing Advanced Server, you must install the `epel-release` package:

  On CentOS 7:

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

# yum makecache
```

  On CentOS 8:

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

# dnf makecache
```

- For CentOS 8, enable the PowerTools repository to satisfy package dependencies:

```
dnf config-manager --set-enabled PowerTools
```

  On CentOS 8, you need to disable the built-in PostgreSQL module:

```
dnf -qy module disable postgresql
```

The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`.

After creating the `edb.repo` file, the `enabled` parameter is set to `1` by default. Replace the `username` and `password` placeholders in the `baseurl` specification with the registered EDB username and password.

```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
↪$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

After saving your changes to the configuration file, you must download and install the repository keys:

Use the following command to download the repository key. Provide the registered `username` and `password` with the `curl` command to download the key.

```
curl -o /etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY https://<username>:<password>
↪@yum.enterprisedb.com/ENTERPRISEDB-GPG-KEY
```

Use the following command to install the key:

```
rpm --import /etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

Then, you can use `yum install` or `dnf install` command to install Advanced Server. For example, to install the server and its core components, use the command:

- On CentOS 7:

  ```
  yum -y install edb-as13-server
  ```

- On CentOS 8:

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

When you install an RPM package that is signed by a source that is not recognized by your system, `yum` may ask for your permission to import the key to your local server. If prompted, and you are satisfied that the packages come from a trustworthy source, enter a `y`, and press `Return` to continue.

After installing Advanced Server, you must configure the installation. For more information, see *Configuring a Package Installation*.

Note: During the installation, `yum` may encounter a dependency that it cannot resolve. If it does, it will provide a list of the required dependencies that you must manually resolve.

3.1.2 Installing Advanced Server on a RHEL Host

You can use an RPM package to install Advanced Server on a RHEL host.

- To install the repository configuration file, assume superuser privileges and invoke one of the following platform specific commands:

  On RHEL 7:

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

  On RHEL 8:

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

- Replace the `USERNAME:PASSWORD` variable in the following command with the username and password of a registered EDB user:

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

- Before installing Advanced Server, you must install the `epel-release` package:

  On RHEL 7:

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

# yum makecache
```

  On RHEL 8:

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

# dnf makecache
```

- Enable the repository:

  On RHEL 7, enable the `optional`, `extras`, and `HA` repositories to satisfy package dependencies:

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable
↪"rhel-*-extras-rpms"  --enable "rhel-ha-for-rhel-*-server-rpms"
```

  On RHEL 8, enable the `codeready-builder-for-rhel-8-*-rpms` repository to satisfy package dependencies:

```
ARCH=$( /bin/arch )

subscription-manager repos --enable "codeready-builder-for-rhel-8-$
↪{ARCH}-rpms"
```

(continues on next page)

(continued from previous page)

On RHEL 8, you need to disable the built-in PostgreSQL module:

```
dnf -qy module disable postgresql
```

The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`.

After creating the `edb.repo` file, the `enabled` parameter is set to `1` by default. Replace the `username` and `password` placeholders in the `baseurl` specification with the registered EDB username and password.

```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
↪$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

After saving your changes to the configuration file, you must download and install the repository keys:

Use the following command to download the repository key. Provide the registered `username` and `password` with the `curl` command to download the key.

```
curl -o /etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY https://<username>:<password>
↪@yum.enterprisedb.com/ENTERPRISEDB-GPG-KEY
```

Use the following command to install the key:

```
rpm --import /etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

Then, you can use `yum install` or `dnf install` command to install Advanced Server. For example, to install the server and its core components, use the command:

- On RHEL 7:

  ```
  yum -y install edb-as13-server
  ```

- On RHEL 8:

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

When you install an RPM package that is signed by a source that is not recognized by your system, `yum` may ask for your permission to import the key to your local server. If prompted, and you are satisfied that the packages come from a trustworthy source, enter a `y`, and press `Return` to continue.

After installing Advanced Server, you must configure the installation. For more information, see *Configuring a Package Installation*.

Note: During the installation, `yum` may encounter a dependency that it cannot resolve. If it does, it will

provide a list of the required dependencies that you must manually resolve.

3.1.3 Installing Advanced Server on a CentOS/RHEL 7 ppc64le Host

You can use an RPM package to install Advanced Server on a CentOS or RHEL 7 ppc64le host.

- To install the Advance Toolchain repository:

  On CentOS or RHEL 7 ppc64le:

```
rpm --import https://public.dhe.ibm.com/software/server/POWER/Linux/
↪toolchain/at/redhat/RHEL7/gpg-pubkey-6976a827-5164221b
```

  The repository configuration file is named `advance-toolchain.repo`. The file resides in `/etc/yum.repos.d`.

- After creating the `advance-toolchain.repo` file, the `enabled` parameter is set to `1` by default.

```
[advance-toolchain]
name=Advance Toolchain IBM FTP
baseurl=https://public.dhe.ibm.com/software/server/POWER/Linux/
↪toolchain/at/redhat/RHEL7
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=ftp://public.dhe.ibm.com/software/server/POWER/Linux/
toolchain/at/redhat/RHELX/gpg-pubkey-6976a827-5164221b
```

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

  On CentOS or RHEL 7 ppc64le:

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

- Replace the `USERNAME:PASSWORD` placeholder in the following command with the username and password of a registered EDB user:

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

- Before installing Advanced Server, you must install the `epel-release` package:

  On CentOS or RHEL 7 ppc64le:

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

# yum makecache
```

- Enable the repository:

  On RHEL 7, enable the `optional`, `extras`, and `HA` repositories to satisfy package dependencies:

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable
↪"rhel-*-extras-rpms"  --enable "rhel-ha-for-rhel-*-server-rpms"
```

The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`.

After creating the `edb.repo` file, the `enabled` parameter is set to `1` by default. Replace the `username` and `password` placeholders in the `baseurl` specification with the registered EDB username and password.

```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
↪$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

After saving your changes to the configuration file, you must download and install the repository keys:

Use the following command to download the repository key. Provide the registered `username` and `password` with the `curl` command to download the key.

```
curl -o /etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY https://<username>:<password>
↪@yum.enterprisedb.com/ENTERPRISEDB-GPG-KEY
```

Use the following command to install the key:

```
rpm --import /etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

Then, you can use `yum install` command to install Advanced Server. For example, to install the server and its core components, use the command:

- On CentOS or RHEL 7 ppc64le:

```
yum -y install edb-as13-server
```

When you install an RPM package that is signed by a source that is not recognized by your system, `yum` may ask for your permission to import the key to your local server. If prompted, and you are satisfied that the packages come from a trustworthy source, enter a `y`, and press `Return` to continue.

After installing Advanced Server, you must configure the installation. For more information, see *Configuring a Package Installation*.

Note: During the installation, `yum` may encounter a dependency that it cannot resolve. If it does, it will provide a list of the required dependencies that you must manually resolve.

3.1.4 Advanced Server RPM Packages

The tables that follow list the RPM packages that are available from EDB. You can also use the `yum search` or `dnf search` command to access a list of the packages that are currently available from your configured repository. Open a command line, assume superuser privileges, and enter:

On RHEL or CentOS 7:

```
yum search package
```

On RHEL or CentOS 8:

```
dnf search package
```

Where `package` is the search term that specifies the name (or partial name) of a package.

Please note: The available package list is subject to change.

Package Name	Package Installs
edb-as13-server	This package installs core components of the Advanced Server database server.
edb-as13-server-client	Client programs and utilities that you can use to access and manage Advanced Server.
edb-as13-server-contrib	Installs contributed tools and utilities that are distributed with Advanced Server. Files for these modules are installed in: Documentation: `/usr/edb/as13/share/doc` Loadable modules: `/usr/edb/as13/lib` Binaries: `/usr/edb/as13/bin`
edb-as13-server-core	Includes the programs needed to create the core functionality behind the Advanced Server database.
edb-as13-server-devel	Installs the header files and libraries needed to compile C or C++ applications that directly interact with an Advanced Server server and the ecpg or ecpgPlus C preprocessor.
edb-as13-server-docs	Installs the readme file.
edb-as13-server-edb-modules	Installs supporting modules for Advanced Server
edb-as13-server-indexadvisor	Installs Advanced Server's Index Advisor feature. The Index Advisor utility helps determine which columns you should index to improve performance in a given workload.
edb-as13-server-libs	Provides the essential shared libraries for any Advanced Server client program or interface.
edb-as13-server-llvmjit	This package contains support for Just in Time (JIT) compiling parts of EDBAS queries.
edb-as13-server-pldebugger	This package implements an API for debugging PL/pgSQL functions on Advanced Server.
edb-as13-server-plperl	Installs the PL/Perl procedural language for Advanced Server. Please note that the `edb-as13-server-plperl` package is dependent on the platform-supplied version of Perl.

continues on next page

Table 1 – continued from previous page

Package Name	Package Installs
edb-as13-server-plpython3	Installs the PL/Python procedural language for Advanced Server. Please note that the PL/Python2 support will no longer be available from Advanced Server version 14 onwards.
edb-as13-server-pltcl	Installs the PL/Tcl procedural language for Advanced Server. Please note that the `edb-as13-pltcl` package is dependent on the platform-supplied version of TCL.
edb-as13-server-sqlprofiler	This package installs Advanced Server's SQL Profiler feature. SQL Profiler helps identify and optimize SQL code.
edb-as13-server-sqlprotect	This package installs Advanced Server's SQL Protect feature. SQL Protect provides protection against SQL injection attacks.
edb-as13-server-sslutils	This package installs functionality that provides SSL support.
edb-as13-server-cloneschema	This package installs the EDB Clone Schema extension. For more information about EDB Clone Schema, see the EDB Postgres Advanced Server Guide.
edb-as13-server-parallel-clone	This package installs functionality that supports the EDB Clone Schema extension.
edb-as13-pgagent	Installs pgAgent; pgAgent is a job scheduler for Advanced Server. Before installing this package, you must install EPEL; for detailed information about installing EPEL, see *Installation Troubleshooting*.
edb-as13-edbplus	The `edb-edbplus` package contains the files required to install the EDB*Plus command line client. EDB*Plus commands are compatible with Oracle's SQL*Plus.
edb-as13-pgsnmpd	SNMP (Simple Network Management Protocol) is a protocol that allows you to supervise an apparatus connected to the network.
edb-as13-pgpool41-extensions	This package creates pgPool extensions required by the server for use with pgpool.
edb-as13-postgis3	Installs POSTGIS meta RPMs.
edb-as13-postgis3-core	This package provides support for geographic objects to the PostgreSQL object-relational database. In effect, PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS), much like ESRI's SDE or Oracle's Spatial extension.
edb-as13-postgis3-docs	This package installs pdf documentation of PostGIS.
edb-as13-postgis-jdbc	This package installs the essential jdbc driver for PostGIS.
edb-as13-postgis3-utils	This package installs the utilities for PostGIS.
edb-as13-postgis3-gui	This package provides a GUI for PostGIS.
edb-as13-slony-replication	Installs the meta RPM for Slony-I.
edb-as13-slony-replication-core	Slony-I builds a primary-standby system that includes all features and capabilities needed to replicate large databases to a reasonably limited number of standby systems.

continues on next page

Table 1 – continued from previous page

Package Name	Package Installs
edb-as13-slony-replication-docs	This package contains the Slony project documentation (in pdf form).
edb-as13-slony-replication-tools	This package contains the Slony altperl tools and utilities that are useful when deploying Slony replication environments. Before installing this package, you must install EPEL; for detailed information about installing EPEL, see *Installation Troubleshooting*.
edb-as13-libicu	These packages contain supporting library files.

The following table lists the packages for Advanced Server 13 supporting components.

Package Name	Package Installs
edb-pgpool41	This package contains the pgPool-II installer. The pgpool-II utility package acts as a middleman between client applications and Server database servers. pgpool-II functionality is transparent to client applications; client applications connect to pgpool-II instead of directly to Advanced Server, and pgpool-II manages the connection. EDB supports the following pgpool-II features: • Load balancing • Connection pooling • High availability • Connection limits pgpool-II runs as a service on Linux systems, and is not supported on Windows systems.
edb-jdbc	The `edb-jdbc` package includes the .jar files needed for Java programs to access an Advanced Server database.
edb-migrationtoolkit	The `edb-migrationtoolkit` package installs Migration Toolkit, facilitating migration to an Advanced Server database from Oracle, PostgreSQL, MySQL, Sybase and SQL Server.
edb-oci	The `edb-oci` package installs the EDB Open Client library, allowing applications that use the Oracle Call Interface API to connect to an Advanced Server database.
edb-oci-devel	This package installs the OCI include files; install this package if you are developing C/C++ applications that require these files.
edb-odbc	This package installs the driver needed for applications to access an Advanced Server system via ODBC.
edb-odbc-devel	This package installs the ODBC include files; install this package if you are developing C/C++ applications that require these files.
edb-pgbouncer114	This package contains PgBouncer (a lightweight connection pooler). This package requires the libevent package.
ppas-xdb	This package contains the xDB installer; xDB provides asynchronous cross-database replication.
ppas-xdb-console	This package provides support for xDB.
ppas-xdb-libs	This package provides support for xDB.

continues on next page

Table 2 – continued from previous page

Package Name	Package Installs
ppas-xdb-publisher	This package provides support for xDB.
ppas-xdb-subscriber	This package provides support for xDB.
edb-pem	The `edb-pem` package installs Management Tool that efficiently manages, monitor, and tune large Postgres deployments from a single remote GUI console.
edb-pem-agent	This package is an agent component of Postgres Enterprise Manager.
edb-pem-docs	This package contains documentation for various languages, which are in HTML format.
edb-pem-server	This package contains server components of Postgres Enterprise Manager.
edb-pgadmin4	This package is a management tool for PostgreSQL capable of hosting the Python application and presenting it to the user as a desktop application.
edb-pgadmin4-desktop-common	This package installs the desktop components of pgAdmin4 for all window managers.
edb-pgadmin4-desktop-gnome	This package installs the gnome desktop components of pgAdmin4
edb-pgadmin4-docs	This package contains documentation of pgAdmin4.
edb-pgadmin4-web	This package contains the required files to run pgAdmin4 as a web application.
edb-efm40	This package installs EDB Failover Manager that adds fault tolerance to database clusters to minimize downtime when a primary database fails by keeping data online in high availability configurations.
edb-rs	This package is a java-based replication framework that provides asynchronous replication across Postgres and EPAS database servers. It supports primary-standby, primary-primary, and hybrid configurations.
edb-rs-client	This package is a java-based command-line tool that is used to configure and operate a replication network via different commands by interacting with the EPRS server.
edb-rs-datavalidator	This package is a java-based command-line tool that provides row and column level data comparison of a source and target database table. The supported RDBMS servers include PostgreSQL, EPAS, Oracle, and MS SQL Server.
edb-rs-libs	This package contains certain libraries that are commonly used by ERPS Server, EPRS Client, and Monitoring modules.
edb-rs-monitor	This package is a java-based application that provides monitoring capabilities to ensure a smooth functioning of the EPRS replication cluster.
edb-rs-server	This package is a java-based replication framework that provides asynchronous replication across Postgres and EPAS database servers. It supports primary-standby, primary-primary, and hybrid configurations.
edb-bart	This package installs the Backup and Recovery Tool (BART) to support online backup and recovery across local and remote PostgreSQL and EDB Advanced Servers.
libevent-edb	This package contains supporting library files.

continues on next page

Table 2 – continued from previous page

Package Name	Package Installs
libiconv-edb	This package contains supporting library files.
libevent-edb-devel	This package contains supporting library files.

3.1.5 Updating an RPM Installation

If you have an existing Advanced Server RPM installation, you can use `yum` or `dnf` to upgrade your repository configuration file and update to a more recent product version. To update the `edb.repo` file, assume superuser privileges and enter:

- On RHEL or CentOS 7:

```
yum upgrade edb-repo
```

- On RHEL or CentOS 8:

```
dnf upgrade edb-repo
```

`yum` or `dnf` will update the `edb.repo` file to enable access to the current EDB repository, configured to connect with the credentials specified in your `edb.repo` file. Then, you can use `yum` or `dnf` to upgrade all packages whose names include the expression `edb`:

- On RHEL or CentOS 7:

```
yum upgrade edb*
```

- On RHEL or CentOS 8:

```
dnf upgrade edb*
```

Note: The `yum upgrade` or `dnf upgrade` command will only perform an update between minor releases; to update between major releases, you must use `pg_upgrade`.

For more information about using yum commands and options, enter `yum --help` on your command line.

For more information about using `dnf` commands and options, visit:

https://docs.fedoraproject.org/en-US/quick-docs/dnf/

3.2 Installing Advanced Server on a Debian or Ubuntu Host

To install Advanced Server on a Debian or Ubuntu host, you must have credentials that allow access to the EDB repository. To request credentials for the repository, visit:

https://www.enterprisedb.com/repository-access-request

The following steps will walk you through using the EDB apt repository to install a debian package. When using the commands, replace the `username` and `password` with the credentials provided by EDB.

- Assume superuser privileges:

```
sudo su -
```

- Configure the EDB repository:

  On Debian 9, Ubuntu 18, and Ubuntu 20:

```
sh -c 'echo "deb https://USERNAME:PASSWORD@apt.enterprisedb.com/
↪$(lsb_release -cs)-edb/ $(lsb_release -cs) main" > /etc/apt/
↪sources.list.d/edb-$(lsb_release -cs).list'
```

  On Debian 10:

  a. Set up the EDB repository:

```
sh -c 'echo "deb [arch=amd64] https://apt.enterprisedb.com/
↪$(lsb_release -cs)-edb/ $(lsb_release -cs) main" > /etc/
↪apt/sources.list.d/edb-$(lsb_release -cs).list'
```

  b. Substitute your EDB credentials for the `username` and `password` placeholders in the following command:

```
sh -c 'echo "machine apt.enterprisedb.com login <USERNAME>␣
↪password <PASSWORD>" > /etc/apt/auth.conf.d/edb.conf'
```

- Add support to your system for secure APT repositories:

```
apt-get -y install apt-transport-https
```

- Add the EBD signing key:

```
wget -q -O - https://apt.enterprisedb.com/edb-deb.gpg.key  | sudo␣
↪apt-key add -
```

- Update the repository metadata:

```
apt-get update
```

- Install Debian package:

```
apt-get -y install edb-as13-server
```

Note: Some Advanced Server supporting components require a Java installation. Before using a native package to add Migration Toolkit or EDB*Plus to your system, please ensure that Java version 8 is installed on your Advanced Server host.

Managing Authentication on a Debian or Ubuntu Host

By default, the server is running with the peer or md5 permission on a Debian or Ubuntu host. You can change the authentication method by modifying the `pg_hba.conf` file, located under `/etc/edb-as/13/main/`.

For more information about modifying the `pg_hba.conf` file, see the PostgreSQL core documentation available at:

https://www.postgresql.org/docs/current/auth-pg-hba-conf.html.

The Debian package manager places Advanced Server and supporting components in the directories listed in the following table:

Component	Location
Server	`/usr/lib/edb-as/13/`
Data and Configuration Directory	`/var/lib/edb-as/13/main` `/etc/edb-as/13/main/`
pgAgent	`/usr/lib/edb-as/13`
Pgpool	`/usr/edb/pgpool4.1/`
Postgis	`/usr/lib/edb-as/13/`
PGSNMPD	`/usr/lib/edb-as/13`
Slony Replication	`/usr/lib/edb-as/13`
pgBouncer	`/usr/edb/pgbouncer1.14/`
pgBouncer Configuration Files	`/etc/edb/pgbouncer1.14/pgbouncer.ini` `/etc/edb/pgbouncer1.14/userlist.txt`
SQL-Profiler	`/usr/lib/edb-as/13/lib`
SQL-Protect	`/usr/lib/edb-as/13/lib`
SSLUTILS	`/usr/lib/edb-as/13/lib`
PL-PERL	`/usr/lib/edb-as/13/lib`
PL-PYTHON	`/usr/lib/edb-as/13/lib`
PLTCL	`/usr/lib/edb-as/13/lib`
EFM	`/usr/edb/efm-4.1/`
JDBC	`/usr/edb/jdbc`
MTK	`/usr/edb/migrationtoolkit/`

3.2.1 Advanced Server Debian Packages

The table that follows lists some of the Debian packages that are available from EDB. You can also use the `apt list` command to access a list of the packages that are currently available from your configured repository. Open a command line, assume superuser privileges, and enter:

```
apt list edb*
```

Please note: The available package list is subject to change.

Package Name	Package Installs
edb-as13-server	Installs core components of the Advanced Server database server.
edb-as13-server-client	Includes client programs and utilities that you can use to access and manage Advanced Server.
edb-as13-server-core	Includes the programs needed to create the core functionality behind the Advanced Server database.
edb-as13-server-dev	The `edb-as13-server-dev` package contains the header files and libraries needed to compile C or C++ applications that directly interact with an Advanced Server server and the ecpg or ecpgPlus C preprocessor.
edb-as13-server-doc	Installs the readme file.
edb-as13-server-edb-modules	Installs supporting modules for Advanced Server.
edb-as13-server-indexadvisor	Installs Advanced Server's Index Advisor feature. The Index Advisor utility helps determine which columns you should index to improve performance in a given workload.
edb-as13-server-pldebugger	This package implements an API for debugging PL/pgSQL functions on Advanced Server.
edb-as13-server-plpython3	Installs the PL/Python procedural language for Advanced Server. Please note that the PL/Python2 support will no longer be available from Advanced Server version 14 onwards.
edb-as13-server-pltcl	Installs the PL/Tcl procedural language for Advanced Server. Please note that the `edb-as13-pltcl` package is dependent on the platform-supplied version of TCL.
edb-as13-server-sqlprofiler	This package installs Advanced Server's SQL Profiler feature. SQL Profiler helps identify and optimize SQL code.
edb-as13-server-sqlprotect	This package installs Advanced Server's SQL Protect feature. SQL Protect provides protection against SQL injection attacks.
edb-as13-server-sslutils	This package installs functionality that provides SSL support.
edb-as13-server-cloneschema	This package installs the EDB Clone Schema extension. For more information about EDB Clone Schema, see the EDB Postgres Advanced Server Guide.
edb-as13-server-parallel-clone	This package installs functionality that supports the EDB Clone Schema extension.

continues on next page

Table 3 – continued from previous page

Package Name	Package Installs
edb-as13-edbplus	The `edb-edbplus` package contains the files required to install the EDB*Plus command line client. EDB*Plus commands are compatible with Oracle's SQL*Plus.
edb-as13-pgsnmpd	SNMP (Simple Network Management Protocol) is a protocol that allows you to supervise an apparatus connected to the network.
edb-as13-pgadmin4	pgAdmin 4 provides a graphical management interface for Advanced Server and PostgreSQL databases.
edb-as13-pgadmin-apache	Apache support module for pgAdmin 4.
edb-as13-pgadmin4-common	pgAdmin 4 supporting files.
edb-as13-pgadmin4-doc	pgAdmin 4 documentation module.
edb-as13-pgpool41-extensions	This package creates pgPool extensions required by the server.
edb-as13-postgis3	This package installs POSTGIS support for geospatial data.
edb-as13-postgis3-scripts	This package installs POSTGIS support for geospatial data.
edb-as13-postgis3-doc	This package provides support for POSTGIS.
edb-as13-postgis3-gui	This package provides a GUI for POSTGIS.
edb-as13-postgis-jdbc	This package provides support for POSTGIS.
edb-as13-postgis-scripts	This package provides support for POSTGIS.
edb-as13-pgagent	This package installs pgAgent; pgAgent is a job scheduler for Advanced Server. Before installing this package, you must install EPEL; for detailed information about installing EPEL, see *Installation Troubleshooting*.
edb-as13-slony-replication	This package installs the meta RPM for Slony-I.
edb-as13-slony-replication-core	This package contains core portions of Slony-I to build a primary-standby system that includes all features and capabilities needed to replicate large databases to a reasonably limited number of standby systems.
edb-as13-slony-replication-docs	This package contains the Slony project documentation (in pdf form).
edb-as13-slony-replication-tools	This package contains the Slony altperl tools and utilities that are useful when deploying Slony replication environments. Before installing this package, you must install EPEL; for detailed information about installing EPEL, see *Installation Troubleshooting*.
edb-as13-hdfs-fdw	The Hadoop Data Adapter allows you to query and join data from Hadoop environments with your Postgres or Advanced Server instances. It is YARN Ready certified with HortonWorks, and provides optimizations for performance with predicate pushdown support.
edb-as13-hdfs-fdw-doc	Documentation for the Hadoop Data Adapter.
edb-as13-mongo-fdw	This EDB Advanced Server extension implements a Foreign Data Wrapper for MongoDB.
edb-as13-mongo-fdw-doc	Documentation for the Foreign Data Wrapper for MongoDB.

continues on next page

Table 3 – continued from previous page

Package Name	Package Installs
edb-as13-mysql-fdw	This EDB Advanced Server extension implements a Foreign Data Wrapper for MySQL.
edb-pgpool41	This package contains the pgPool-II installer. The pgpool-II utility package acts as a middleman between client applications and Server database servers. pgpool-II functionality is transparent to client applications; client applications connect to pgpool-II instead of directly to Advanced Server, and pgpool-II manages the connection. EDB supports the following pgpool-II features: • Load balancing • Connection pooling • High availability • Connection limits pgpool-II runs as a service on Linux systems, and is not supported on Windows systems.
edb-jdbc	The `edb-jdbc` package includes the .jar files needed for Java programs to access an Advanced Server database.
edb-migrationtoolkit	The `edb-migrationtoolkit` package installs Migration Toolkit, facilitating migration to an Advanced Server database from Oracle, PostgreSQL, MySQL, Sybase and SQL Server.
edb-pgbouncer114	This package contains PgBouncer (a lightweight connection pooler). This package requires the libevent package.
edb-efm40	This package installs EDB Failover Manager that adds fault tolerance to database clusters to minimize downtime when a primary database fails by keeping data online in high availability configurations.

3.3 Configuring a Package Installation

The packages that install the database server component create a unit file (on version 7.x or 8.x hosts) and service startup scripts.

3.3.1 Creating a Database Cluster and Starting the Service

The PostgreSQL `initdb` command creates a database cluster; when installing Advanced Server with an RPM package, the `initdb` executable is in `/usr/edb/asx.x/bin`. After installing Advanced Server, you must manually configure the service and invoke `initdb` to create your cluster. When invoking `initdb`, you can:

- Specify environment options on the command line.
- Include the `systemd` service manager on RHEL or CentOS 7.x | 8.x and use a service configuration file to configure the environment.

To review the `initdb` documentation, visit:

https://www.postgresql.org/docs/current/static/app-initdb.html

After specifying any options in the service configuration file, you can create the database cluster and start the service; these steps are platform specific.

On RHEL or CentOS 7.x | 8.x

To invoke `initdb` on a RHEL or CentOS 7.x | 8.x system, with the options specified in the service configuration file, assume the identity of the operating system superuser:

```
su - root
```

To initialize a cluster with the non-default values, you can use the `PGSETUP_INITDB_OPTIONS` environment variable by invoking the `edb-as-13-setup` cluster initialization script that resides under `EPAS_Home/bin`.

To invoke `initdb` export the `PGSETUP_INITDB_OPTIONS` environment variable with the following command:

```
PGSETUP_INITDB_OPTIONS="-E UTF-8" /usr/edb/as13/bin/edb-as-13-setup initdb
```

After creating the cluster, use `systemctl` to start, stop, or restart the service:

```
systemctl { start | stop | restart } edb-as-13
```

On Debian 9.x | 10.x or Ubuntu 18.04 | 20.04

You can initialize multiple clusters using the bundled scripts. To create a new cluster, assume `root` privileges, and invoke the bundled script:

```
/usr/edb/as13/bin/epas_createcluster 13 main2
```

To start a new cluster, use the following command:

```
/usr/edb/as13/bin/epas_ctlcluster 13 main2 start
```

To list all the available clusters, use the following command:

```
/usr/edb/as13/bin/epas_lsclusters
```

Note: The data directory is created under `/var/lib/edb-as/13/main2` and configuration directory is created under `/etc/edb-as/13/main/`.

3.3.1.1 Specifying Cluster Options with INITDBOPTS

You can use the `INITDBOPTS` variable to specify your cluster configuration preferences. By default, the `INITDBOPTS` variable is commented out in the service configuration file; unless modified, when you run the service startup script, the new cluster will be created in a mode compatible with Oracle databases. Clusters created in this mode will contain a database named `edb`, and have a database superuser named `enterprisedb`.

3.3.1.1.1 Initializing the Cluster in Oracle Mode

If you initialize the database using Oracle compatibility mode, the installation includes:

- Data dictionary views 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).
- Support for the following Oracle built-in packages.

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

3.3.1.1.2 Initializing the Cluster in Postgres Mode

Clusters created in PostgreSQL mode do not include compatibility features. To create a new cluster in PostgreSQL mode, remove the pound sign (#) in front of the `INITDBOPTS` variable, enabling the `"--no-redwood-compat"` option. Clusters created in PostgreSQL mode will contain a database named `postgres` and have a database superuser named `postgres`.

You may also specify multiple `initdb` options. For example, the following statement:

```
INITDBOPTS="--no-redwood-compat -U alice --locale=en_US.UTF-8"
```

Creates a database cluster (without compatibility features for Oracle) that contains a database named `postgres` that is owned by a user named `alice`; the cluster uses `UTF-8` encoding.

If you initialize the database using `"--no-redwood-compat"` mode, the installation includes the following package:

Package	Functionality non-compatible with Oracle Databases
dbms_aqadm	Provides supporting procedures for Advanced Queueing functionality.
dbms_aq	Provides message queueing and processing for Advanced Server.
edb_bulkload	Provides direct/conventional data loading capability when loading huge amount of data into a database.
edb_gen	Provides miscellaneous packages to run built-in packages.
edb_objects	Provides Oracle compatible objects such as packages, procedures etc.
waitstates	Provides monitor session blocking.
edb_dblink_libpq	Provides link to foreign databases via libpq.
edb_dblink_oci	Provides link to foreign databases via OCI.
snap_tables	Creates tables to hold wait information. Included with DRITA scripts.
snap_functions	Creates functions to return a list of snap ids and the time the snapshot was taken. Included with DRITA scripts.
sys_stats	Provides OS performance statistics.

In addition to the cluster configuration options documented in the PostgreSQL core documentation, Advanced Server supports the following `initdb` options:

`--no-redwood-compat`

> Include the `--no-redwood-compat` keywords to instruct the server to create the cluster in PostgreSQL mode. When the cluster is created in PostgreSQL mode, the name of the database superuser will be `postgres`, the name of the default database will be `postgres`, and Advanced Server's features compatible with Oracle databases will not be available to the cluster.

`--redwood-like`

> Include the `--redwood-like` keywords to instruct the server to use an escape character (an empty string (‘’)) following the `LIKE` (or PostgreSQL-compatible `ILIKE`) operator in a SQL statement that is compatible with Oracle syntax.

`--icu-short-form`

> Include the `--icu-short-form` keywords to create a cluster that uses a default ICU (International Components for Unicode) collation for all databases in the cluster. For more information about Unicode collations, refer to the *EDB Postgres Advanced Server Guide* available at:
>
> https://www.enterprisedb.com/edb-docs

For more information about using `initdb`, and the available cluster configuration options, see the PostgreSQL Core Documentation available at:

https://www.postgresql.org/docs/current/static/app-initdb.html

You can also view online help for `initdb` by assuming superuser privileges and entering:

```
/path_to_initdb_installation_directory/initdb --help
```

Where `path_to_initdb_installation_directory` specifies the location of the `initdb` binary file.

3.3.2 Modifying the Data Directory Location on CentOS or Redhat 7.x

On a CentOS or RedHat version 7.x host, the unit file is named `edb-as-13.service` and resides in `/usr/lib/systemd/system`. The unit file contains references to the location of the Advanced Server `data` directory. You should avoid making any modifications directly to the unit file because it may be overwritten during package upgrades.

By default, data files reside under `/var/lib/edb/as13/data` directory. To use a data directory that resides in a non-default location, perform the following steps:

- Create a copy of the unit file under the `/etc` directory:

```
cp /usr/lib/systemd/system/edb-as-13.service /etc/systemd/system/
```

- After copying the unit file to the new location, create the service file `/etc/systemd/system/edb-as-13.service`.
- Update the following values with new location of data directory in the `/lib/systemd/system/edb-as-13.service` file:

```
Environment=PGDATA=/var/lib/edb/as13/data
PIDFile=/var/lib/edb/as13/data/postmaster.pid
```

- Delete the entire content of `/etc/systemd/system/edb-as-13.service` file, except the following line:

```
.include /lib/systemd/system/edb-as-13.service
```

- Run the following command to initialize the cluster at the new location:

```
PGSETUP_INITDB_OPTIONS="-E UTF-8" /usr/edb/as13/bin/edb-as-13-setup␣
↪initdb
```

- Use the following command to reload `systemd`, updating the modified service scripts:

```
systemctl daemon-reload
```

- Start the Advanced Server service with the following command:

```
systemctl start edb-as-13
```

3.4 Starting Multiple Postmasters with Different Clusters

You can configure Advanced Server to use multiple postmasters, each with its own database cluster. The steps required are version specific to the Linux host.

On RHEL or CentOS 7.x | 8.x

The `edb-as13-server-core` RPM for version 7.x | 8.x contains a unit file that starts the Advanced Server instance. The file allows you to start multiple services, with unique `data` directories and that monitor different ports. You must have `root` access to invoke or modify the script.

The example that follows creates an Advanced Server installation with two instances; the secondary instance is named `secondary`:

- Make a copy of the default file with the new name. As noted at the top of the file, all modifications must reside under `/etc`. You must pick a name that is not already used in `/etc/systemd/system`.

```
cp /usr/lib/systemd/system/edb-as-13.service /etc/systemd/system/
↪secondary-edb-as-13.service
```

- Edit the file, changing `PGDATA` to point to the new `data` directory that you will create the cluster against.
- Create the target `PGDATA` with user `enterprisedb`.
- Run `initdb`, specifying the setup script:

```
/usr/edb/as13/bin/edb-as-13-setup initdb secondary-edb-as-13
```

- Edit the `postgresql.conf` file for the new instance, specifying the port, the IP address, TCP/IP settings, etc.
- Make sure that new cluster runs after a reboot:

```
systemctl enable secondary-edb-as-13
```

- Start the second cluster with the following command:

```
systemctl start secondary-edb-as-13
```

3.5 Creating an Advanced Server Repository on an Isolated Network

You can create a local repository to act as a host for the Advanced Server RPM packages if the server on which you wish to install Advanced Server (or supporting components) cannot directly access the EDB repository. Please note that this is a high-level listing of the steps requires; you will need to modify the process for your individual network.

To create and use a local repository, you must:

- Use `yum` or `dnf` to install the `epel-release`, `yum-utils`, and `createrepo` packages.

  On RHEL or CentOS 7.x:

  ```
  yum install epel-release
  yum install yum-utils
  yum install createrepo
  ```

  On RHEL or CentOS 8.x:

  ```
  dnf install epel-release
  dnf install yum-utils
  dnf install createrepo
  ```

- Create a directory in which to store the repository:

  ```
  mkdir /srv/repos
  ```

- Copy the RPM installation packages to your local repository. You can download the individual packages or use a tarball to populate the repository. The packages are available from the EDB repository at https://repos.enterprisedb.com/.

- Sync the RPM packages and create the repository.

  ```
  reposync -r edbas13 -p /srv/repos
  createrepo /srv/repos
  ```

- Install your preferred webserver on the host that will act as your local repository, and ensure that the repository directory is accessible to the other servers on your network.

- On each isolated database server, configure `yum` or `dnf` to pull updates from the mirrored repository on your local network. For example, you might create a repository configuration file called `/etc/yum.repos.d/edb repo` with connection information that specifies:

  ```
  [edbas13]
  name=EnterpriseDB Advanced Server 13
  baseurl=https:yum.your_domain.com/edbas13
  enabled=1
  gpgcheck=0
  ```

After specifying the location and connection information for your local repository, you can use `yum` or `dnf` commands to install Advanced Server and its supporting components on the isolated servers. For example:

- On RHEL or CentOS 7.x:

```
yum -y install edb-as13-server
```

- On RHEL or CentOS 8.x:

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

For more information about creating a local `yum` repository, visit:

https://wiki.centos.org/HowTos/CreateLocalRepos

CHAPTER 4

Installation Troubleshooting

Difficulty Displaying Java-based Applications

If you encounter difficulty displaying Java-based server features (controls or text not being displayed correctly, or blank windows), upgrading to the latest `libxcb-xlib` libraries should correct the problem on most distributions. Please visit the following link for other possible work-arounds:

http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6532373

The Installation Fails to Complete Due to Existing data Directory Contents

If an installation fails to complete due to an existing content in the data directory, the server will write an error message to the server logs:

```
A data directory is neither empty, or a recognisable data directory.
```

If you encounter a similar message, you should confirm that the data directory is empty; the presence of files (including the system-generated `lost+found` folder) will prevent the installation from completing. Either remove the files from the data directory, or specify a different location for the data directory before re-invoking the installer to complete the installation.

Difficulty Installing the EPEL Release Package

If you encounter difficulty when installing the `EPEL` release package, you can use the following command to install the `epel-release` package on RHEL or CentOS 7 and 8 platform:

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

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

Please note that you may need to enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum.repos.d`). If `yum` cannot access a repository that contains `epel-release`,

you will get an error message:

```
No package epel available.
Error: Nothing to do
```

If you receive this error, you can download the EPEL rpm package, and install it manually. To manually install EPEL, download the rpm package, assume superuser privileges, navigate into the directory that contains the package, and install EPEL with the command:

```
yum -y install epel-release

dnf -y install epel-release
```

CHAPTER 5

Managing an Advanced Server Installation

Unless otherwise noted, the commands and paths noted in the following section assume that you have performed an installation using the native packages.

5.1 Starting and Stopping Advanced Server and Supporting Components

A service is a program that runs in the background and requires no user interaction (in fact, a service provides no user interface); a service can be configured to start at boot time, or manually on demand. Services are best controlled using the platform-specific operating system service control utility. Many of the Advanced Server supporting components are services.

The following table lists the names of the services that control Advanced Server and services that control Advanced Server supporting components:

Advanced Server Component Name	Linux Service Name	Debian Service Name
Advanced Server	edb-as-13	edb-as@13-main
pgAgent	edb-pgagent-13	edb-as13-pgagent
PgBouncer	edb-pgbouncer-1.14	edb-pgbouncer114
pgPool-II	edb-pgpool-4.1	edb-pgpool41
Slony	edb-slony-replication-13	edb-as13-slony-replication
EFM	edb-efm-4.0	edb-efm-4.0

You can use the Linux command line to control Advanced Server's database server and the services of Advanced Server's supporting components. The commands that control the Advanced Server service on a Linux platform are host specific.

5.1.1 Controlling a Service on CentOS or RHEL 7.x | 8.x

If your installation of Advanced Server resides on version 7.x | 8.x of RHEL and CentOS, you must use the `systemctl` command to control the Advanced Server service and supporting components.

The `systemctl` command must be in your search path and must be invoked with superuser privileges. To use the command, open a command line, and enter:

```
systemctl <action> <service_name>
```

Where:

`service_name` specifies the name of the service.

`action` specifies the action taken by the service command. Specify:

- `start` to start the service.
- `stop` to stop the service.
- `restart` to stop and then start the service.
- `status` to discover the current status of the service.

5.1.2 Controlling a Service on Debian 9.x | 10.x or Ubuntu 18.04 | 20.04

If your installation of Advanced Server resides on version 9x of Debian or 18.04 of Ubuntu, assume superuser privileges and invoke the following commands (using bundled scripts) to manage the service. Use the following commands to:

- Discover the current status of a service:

```
/usr/edb/as13/bin/epas_ctlcluster 13 main status
```

- Stop a service:

```
/usr/edb/as13/bin/epas_ctlcluster 13 main stop
```

- Restart a service:

```
/usr/edb/as13/bin/epas_ctlcluster 13 main restart
```

- Reload a service:

```
/usr/edb/as13/bin/epas_ctlcluster 13 main reload
```

- Control the component services:

```
systemctl restart edb-as@13-main
```

5.1.3 Using pg_ctl to Control Advanced Server

You can use the `pg_ctl` utility to control an Advanced Server service from the command line on any platform. `pg_ctl` allows you to start, stop, or restart the Advanced Server database server, reload the configuration parameters, or display the status of a running server. To invoke the utility, assume the identity of the cluster owner, navigate into the home directory of Advanced Server, and issue the command:

```
./bin/pg_ctl -D <data_directory> <action>
```

`data_directory` is the location of the data controlled by the Advanced Server cluster.

`action` specifies the action taken by the `pg_ctl` utility. Specify:

- `start` to start the service.
- `stop` to stop the service.
- `restart` to stop and then start the service.
- `reload` sends the server a `SIGHUP` signal, reloading configuration parameters
- `status` to discover the current status of the service.

For more information about using the `pg_ctl` utility, or the command line options available, see the official PostgreSQL Core Documentation available at:

https://www.postgresql.org/docs/current/static/app-pg-ctl.html

Choosing Between pg_ctl and the service Command

You can use the `pg_ctl` utility to manage the status of an Advanced Server cluster, but it is important to note that `pg_ctl` does not alert the operating system service controller to changes in the status of a server, so it is beneficial to use the `service` command whenever possible.

5.1.4 Configuring Component Services to AutoStart at System Reboot

After installing, configuring, and starting the services of Advanced Server supporting components on a Linux system, you must manually configure your system to autostart the service when your system reboots. To configure a service to autostart on a Linux system, open a command line, assume superuser privileges, and enter the following command.

On a Redhat-compatible Linux system, enter:

```
/sbin/chkconfig <service_name> on
```

Where `service_name` specifies the name of the service.

5.2 Connecting to Advanced Server with edb-psql

`edb-psql` is a command line client application that allows you to execute SQL commands and view the results. To open the `edb-psql` client, the client must be in your search path. The executable resides in the `bin` directory, under your Advanced Server installation.

Use the following command and options to start the `edb-psql` client:

```
psql -d edb -U enterprisedb
```

Where:

`-d` specifies the database to which `edb-psql` will connect.

`-U` specifies the identity of the database user that will be used for the session.

`edb-psql` is a symbolic link to PostgreSQL community `psql`. For more information about using the command line client, see the PostgreSQL Core Documentation at:

https://www.postgresql.org/docs/current/static/app-psql.html

CHAPTER 6

Installing and Configuring pgAdmin4

pgAdmin 4 is the leading Open Source management tool for Postgres databases. EDB pgAdmin 4 is an enhanced version of open source pgAdmin 4 specifically for Advanced Server databases. It is designed to meet the needs of both novice and experienced Postgres users alike, providing a powerful graphical interface that simplifies the creation, maintenance, and use of database objects.

You can install EDB pgAdmin 4 for your Advanced Server database using the `yum` package manager for RHEL or CentOS 7.x platform and using `dnf` package manager for RHEL or CentOS 8.x platform.

6.1 Installing pgAdmin 4 on a Linux Host

You can use the following steps to install pgAdmin4 using `yum` package manager:

Create a Repository Configuration File

To create a repository configuration file, you must have the credentials that allow to access the EDB repository. For information about requesting credentials, visit:

https://www.enterprisedb.com/user/login

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

```
yum -y install https://yum.enterprisedb.com/edbrepos/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, the `enabled` parameter is set to `1` by default. Replace the `username` and `password` placeholders in the `baseurl` specification with the name and password of a registered EDB user.

```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
↪$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

Note: If you have `edb.repo` already configured then you can skip this step.

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*
```

The following packages will be installed:

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

Start pgAdmin 4 in Desktop Mode

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

You can also start pgAdmin 4 in desktop mode from the `Applications` menu as shown below:

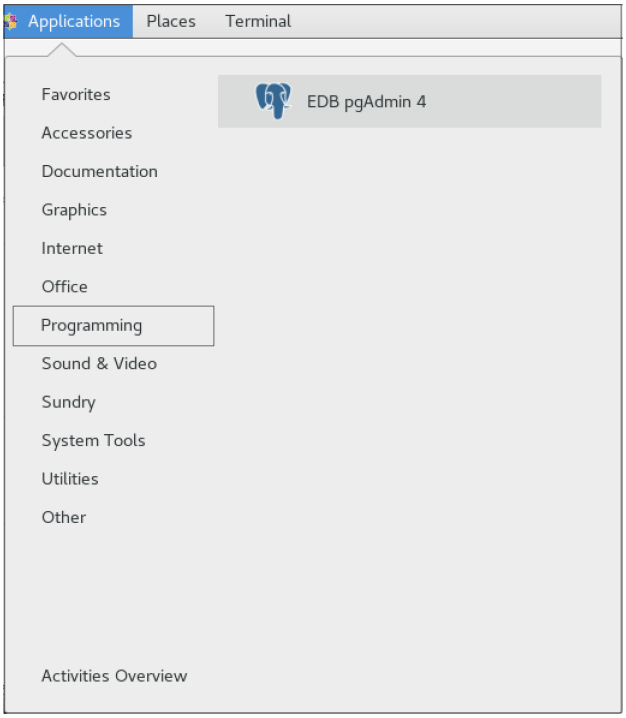

Fig. 1: *Accessing EDB pgAdmin 4 from Applications Menu.*

6.2 Registering and Connecting to Advanced Server with pgAdmin 4

First, you must register Advanced Server on pgAdmin 4. For information about registering your server, visit:

https://www.pgadmin.org/docs/pgadmin4/latest/server_dialog.html.

To connect to your registered Advanced Server instance, right click on your server name, select `Connect Server`, and provide the password:

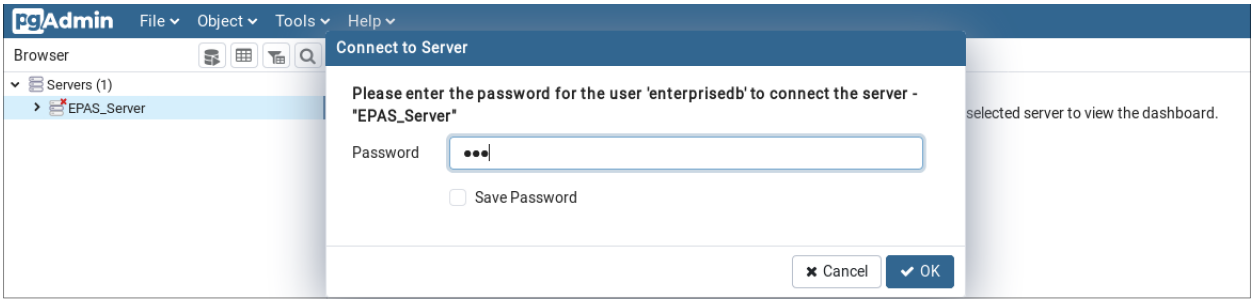

Fig. 2: *Connecting to EPAS Server through EDB pgAdmin 4.*

CHAPTER 7

Uninstalling Advanced Server

Note that after uninstalling Advanced Server, the cluster data files remain intact and the service user persists. You may manually remove the cluster `data` and service user from the system.

7.1 Uninstalling an RPM Package

You can use variations of the `rpm, yum` or `dnf` command to remove installed packages. Note that removing a package does not damage the Advanced Server `data` directory.

Include the `-e` option when invoking the `rpm` command to remove an installed package; the command syntax is:

```
rpm -e <package_name>
```

Where `package_name` is the name of the package that you would like to remove.

You can use the `yum remove` or `dnf remove` command to remove a package installed by `yum` or `dnf`. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

- On RHEL or CentOS 7:

```
yum remove <package_name>
```

- On RHEL or CentOS 8:

```
dnf remove <package_name>
```

Where `package_name` is the name of the package that you would like to remove.

`yum` and RPM will not remove a package that is required by another package. If you attempt to remove a package that satisfies a package dependency, `yum` or RPM will provide a warning.

Note: In RHEL or CentOS 8, removing a package also removes all its dependencies that are not required by other packages. To override this default behavior of RHEL or CentOS 8, you must disable the `clean_requirements_on_remove` parameter in the `/etc/yum.conf` file.

To uninstall Advanced Server and its dependent packages; use the following command:

- On RHEL or CentOS 7:

```
yum remove edb-as13-server*
```

- On RHEL or CentOS 8:

```
dnf remove edb-as13-server*
```

7.2 Uninstalling Advanced Server Components on a Debian or Ubuntu Host

- To uninstall Advanced Server, invoke the following command. The configuration files and data directory remains intact.

```
apt-get remove edb-as13-server*
```

- To uninstall Advanced Server, configuration files, and data directory, invoke the following command:

```
apt-get purge edb-as13-server*
```

CHAPTER 8

Conclusion

EDB Postgres™ Advanced Server Installation Guide for Linux

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

Index