EDB Postgres™ Advanced Server Installation Guide for Linux

EDB Postgres™ Advanced Server 11

March 31, 2021

EDB Postgres™ Advanced Server Installation Guide for Linux
by EnterpriseDB® Corporation
Copyright © 2009 - 2021 EnterpriseDB Corporation. All rights reserved.

2

Table of Contents

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 Advanced Server 11 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.

1.1 Typographical Conventions Used in this Guide

Certain typographical conventions are used in this manual to clarify the meaning and usage of various commands, statements, programs, examples, etc. This section provides a summary of these conventions.

In the following descriptions, a *term* refers to any word or group of words that are language keywords, user-supplied values, literals, etc. A term's exact meaning depends upon the context in which it is used.

- *Italic font* introduces a new term, typically in the sentence that defines it for the first time.

- `Fixed-width (mono-spaced) font` is used for terms that must be given literally such as SQL commands, specific table and column names used in the examples, programming language keywords, etc. For example, `SELECT * FROM emp;`

- *`Italic fixed-width font`* is used for terms for which the user must substitute values in actual usage. For example, `DELETE FROM table_name;`

- A vertical pipe | denotes a choice between the terms on either side of the pipe. A vertical pipe is used to separate two or more alternative terms within square brackets (optional choices) or braces (one mandatory choice).

- Square brackets [ ] denote that one or none of the enclosed terms may be substituted. For example, `[ a | b ]` means choose one of "`a`" or "`b`" or neither of the two.

- Braces {} denote that exactly one of the enclosed alternatives must be specified. For example, `{ a | b }` means exactly one of "`a`" or "`b`" must be specified.

- Ellipses ... denote that the preceding term may be repeated. For example, `[ a | b ] ...` means that you may have the sequence, "`b a a b a`".

2 Requirements Overview

The following sections detail the supported platforms and installation requirements for EDB Postgres Advanced Server 11.

2.1 Supported Platforms

To review a list of supported platforms, visit the EnterpriseDB website at:

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

Note:

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

- Advanced Server is no longer supported on RHEL/CentOS/OEL 6.x platforms. It is strongly recommended that EDB products running on these platforms be migrated to a supported platform.

2.2 Installation Pre-requisites

You can use an RPM package to install Advanced Server and its supporting components on a Linux host. Before installing the Advanced Server, you must:

Install Linux-specific Software

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

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 1.7 or later for Migration Toolkit and Java version 1.8 or later for EDB*Plus. 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 EnterpriseDB Repository

Before installing Advanced Server, you must have credentials that allow access to the EnterpriseDB repository. For information about requesting credentials, visit:

https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%2004-09-2019.pdf

After receiving your repository credentials you can:

- Create the repository configuration file.

- Modify the file, providing your user name and password.

- Install Advanced Server.

3 Limitations

The following limitation applies to the EDB Postgres Advanced Server:

- The LLVM JIT package is supported on RHEL 7.x, 8.x, and SLES. LLVM JIT is not supported on PPC-LE 64 (running RHEL or CentOS 7.x).

4 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 must have credentials for the EnterpriseDB repository to install Advanced Server; to submit a request for credentials, visit:

https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%2004-09-2019.pdf

For a list of the RPM installers and Debian packages available from EnterpriseDB for Advanced Server and supporting components, see Section 4.3.1.

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/as11`.

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, please see Section 6.2.

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

Component	Location
Executables	`/usr/edb/as11/bin`
Libraries	`/usr/edb/as11/lib`
Cluster configuration files	`/etc/edb/as11`
Documentation	`/usr/edb/as11/share/doc`
Contrib	`/usr/edb/as11/share/contrib`
Data	`/var/lib/edb/as11/data`
Logs	`/var/log/as11`
Lock files	`/var/lock/as11`
Log rotation file	`/etc/logrotate.d/as11`
Sudo configuration file	`/etc/sudoers.d/as11`
Binary to access VIP without sudo	`/usr/edb/as11/bin/secure`
Backup area	`/var/lib/edb/as11/backups`

Component	Location
Templates	/usr/edb/as11/share
Procedural Languages	/usr/edb/as11/lib or /usr/edb/as11/lib64
Development Headers	/usr/edb/as11/include
Shared data	/usr/edb/as11/share
Regression tests	/usr/edb/as11/lib/pgxs/src/test/regress
SGML Documentation	/usr/edb/as11/share/doc

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

Debian Package Component	Location
Server	/usr/lib/edb-as/11/
Data and Configuration Directory	/var/lib/edb-as/11/main /etc/edb-as/11/main/
pgAgent	/usr/lib/edb-as/11
Pgpool	/usr/edb/pgpool3.7/
Postgis	/usr/lib/edb-as/11/
PGSNMPD	/usr/lib/edb-as/11
Slony Replication	/usr/lib/edb-as/11
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/11/lib
SQL-Protect	/usr/lib/edb-as/11/lib
SSLUTILS	/usr/lib/edb-as/11/lib
PL-PERL	/usr/lib/edb-as/11/lib
PL-PYTHON	/usr/lib/edb-as/11/lib
PLTCL	/usr/lib/edb-as/11/lib
EFM	/usr/edb/efm-4.1
JDBC	/usr/edb/jdbc
MTK	/usr/edb/migrationtoolkit/

4.1 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:

  a. On CentOS 7:

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

  b. On CentOS 8:

  ```
  dnf -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 EnterpriseDB 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:

  a. On CentOS 7:

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

  b. On CentOS 8:

  ```
  dnf -y install epel-release
  ```

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

  ```
  dnf config-manager --set-enabled powertools
  ```

  On CentOS 8, 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, use your choice of editor to ensure that the value of the `enabled` parameter is `1`, and replace the `username` and `password` placeholders in the `baseurl` specification with a registered EnterpriseDB username and password.

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

After saving your changes to the configuration file, you 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-as11-server
  ```

- On CentOS 8:

  ```
  dnf -y install edb-as11-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; see Section 4.6 , *Configuring a Package Installation*, for details.

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.

4.2 Installing Advanced Server on a RHEL Host

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

Note: Advanced Server version 11 is now supported on RHEL 8 (8.3 and onwards).

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

  a. On RHEL 7:

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

  b. On RHEL 8:

  ```
  dnf -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 EnterpriseDB 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:

  a. On RHEL 7:

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

  b. On RHEL 8:

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

- Enable supporting repositories:

  On RHEL7, use the following command to 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 RHEL8, 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"
```

On RHEL8, 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, use your choice of editor to ensure that the value of the
`enabled` parameter is `1`, and replace the `username` and `password` placeholders in the
`baseurl` specification with a registered EnterpriseDB username and password.

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

After saving your changes to the configuration file, you 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-as11-server
```

- On RHEL 8:

```
dnf -y install edb-as11-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; see Section 4.6 ,
Configuring a Package Installation, for details.

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.

15

4.3 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.

- Use the following command to install the Advance Toolchain repository:

```
rpm --import
https://public.dhe.ibm.com/software/server/POWER/Linux/toolcha
in/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, use your choice of editor to set the value of the `enabled` parameter to `1`, and replace the `username` and `password` placeholders in the `baseurl` specification with a registered EnterpriseDB username and password.

```
[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/to
olchain/at/redhat/RHELX/gpg-pubkey-6976a827-5164221b
```

- To install 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
```

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

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

- On RHEL 7, use the following command to 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, use your choice of editor to ensure that the value of the `enabled` parameter is `1`, and replace the `username` and `password` placeholders in the `baseurl` specification with a registered EnterpriseDB username and password.

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

After saving your changes to the configuration file, you 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-as11-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; see Section <u>4.6</u> , *Configuring a Package Installation*, for details.

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.

4.3.1 Advanced Server RPM Packages

The tables that follow list the RPM packages that are available from EnterpriseDB.

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.

The repository search will return a list of available packages that include the specified search term.

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

Package Name	Package Installs
`edb-as11-server`	This package installs core components of the Advanced Server database server.
`edb-as11-server-client`	The `edb-as11-server-client` package contains client programs and utilities that you can use to access and manage Advanced Server.

`edb-as11-server-contrib`	The `edb-as11-contrib` package installs contributed tools and utilities that are distributed with Advanced Server. Files for these modules are installed in:
	    Documentation: `/usr/edb/as11/share/doc`
	    Loadable modules: `/usr/edb/as11/lib`
	     Binaries: `/usr/edb/as11/bin`

`edb-as11-server-core`	The `edb-as11-server-core` package includes the programs needed to create the core functionality behind the Advanced Server database.
`edb-as11-server-devel`	The `edb-as11-server-devel` 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-as11-server-docs`	The `edb-as11-server-docs` package installs the readme file.
`edb-as11-server-edb-modules`	This package installs supporting modules for Advanced Server
`edb-as11-server-indexadvisor`	This package 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-as11-server-libs`	The `edb-as11-server-libs` package provides the essential

Package Name	Package Installs
	shared libraries for any Advanced Server client program or interface.
`edb-as11-server-llvmjit`	This package contains support for Just in Time (JIT) compiling parts of EDBAS queries.
`edb-as11-server-pldebugger`	This package implements an API for debugging PL/pgSQL functions on Advanced Server.
`edb-as11-server-plperl`	The `edb-as11-server-plperl` package installs the PL/Perl procedural language for Advanced Server. Please note that the `edb-as11-server-plperl` package is dependent on the platform-supplied version of Perl.
`edb-as11-server-plpython` `edb-as11-server-plpython3`	This package installs the PL/Python procedural language for Advanced Server. The `edb-as11-server-plpython` package is dependent on the platform-supplied version of Python. By default, the `edb-as11-server-plpython3` package is not installed; you need to explicitly install the package using `yum`. To install `edb-as11-server-plpython3` package, you must first uninstall the plpython platform-supplied version of Python.
`edb-as11-server-pltcl`	The `edb-as11-pltcl` package installs the PL/Tcl procedural language for Advanced Server. Please note that the `edb-as11-pltcl` package is dependent on the platform-supplied version of TCL.
`edb-as11-server-sqlprofiler`	This package installs Advanced Server's SQL Profiler feature. SQL Profiler helps identify and optimize SQL code.
`edb-as11-server-sqlprotect`	This package installs Advanced Server's SQL Protect feature. SQL Protect provides protection against SQL injection attacks.
`edb-as11-server-sslutils`	This package installs functionality that provides SSL support.
`edb-as11-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-as11-server-parallel-clone`	This package installs functionality that supports the EDB Clone Schema extension.
`edb-as11-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-as11-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 Section 5.
`edb-as11-pgsnmpd`	SNMP (Simple Network Management Protocol) is a protocol that allows you to supervise an apparatus connected to the network.
`edb-as11-pljava`	This package installs PL/Java, providing access to Java stored procedures, triggers and functions via the JDBC interface.
`edb-as11-pgpool37-extensions`	This package creates pgPool extensions required by the server.
`edb-as11-postgis`	This package installs POSTGIS meta RPMs.
`edb-as11-slony-replication`	This package installs the meta RPM for Slony-I.
`edb-as11-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.
`edb-as11-slony-replication-docs`	This package contains the Slony project documentation (in pdf form).
`edb-as11-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

Package Name	Package Installs
	information about installing EPEL, see Section 5.
edb-as11-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.
edb-as11-mongo_fdw	This EnterpriseDB Advanced Server extension implements a Foreign Data Wrapper for MongoDB.
edb-as11-mysql8_fdw edb-as11-mysql5_fdw	This EnterpriseDB Advanced Server extension implements a Foreign Data Wrapper for MySQL.
edb-as11-libicu	These packages contain supporting library files.

The following table lists the packages for Advanced Server 11 supporting components.

Package Name	Package Installs
edb-pgpool37	This package contains the pgPool-II installer. pgPool provides connection pooling for Advanced Server installations.
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 EnterpriseDB 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-pgbouncer19	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. For more information, visit http://www.enterprisedb.com/faq-xdb-multi-master
ppas-xdb-console	This package provides support for xDB.
ppas-xdb-libs	This package provides support for xDB.
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-efm41 edb-efm40 edb-efm39	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-bart	This package installs the Backup and Recovery Tool (BART) to support online backup and recovery across local and remote

Package Name	Package Installs
	PostgreSQL and EDB Advanced Servers.

`libevent-edb`
`libiconv-edb`
`libevent-edb-devel`
	These packages contain supporting library files.

4.3.2 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 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*
  ```

Please note that 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.

4.4 Installing Advanced Server on a SLES Host

You must have credentials that allow access to the EnterpriseDB repository to install Advanced Server on a SLES host. To request credentials for the repository, visit:

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

You can use the zypper package manager to install Advanced Server or Advanced Server supporting components on an SLES host. zypper will attempt to satisfy package dependencies as it installs a package, but some components require access to specific repositories that are not hosted at EnterpriseDB.

To add Advanced Server or supporting components, use the following commands to add EnterpriseDB repository configuration files to your SLES host:

```
zypper addrepo https://zypp.enterprisedb.com/suse/epas11-
sles.repo
zypper addrepo https://zypp.enterprisedb.com/suse/epas-sles-
tools.repo
zypper addrepo https://zypp.enterprisedb.com/suse/epas-sles-
dependencies.repo
```

Each command creates a repository configuration file in the `/etc/zypp/repos.d` directory. The files are named:

- `edbas11suse.repo`
- `edbasdependencies.repo`
- `edbastools.repo`

After creating the repository configuration files, use the `zypper refresh` command to refresh the metadata on your SLES host to include the EnterpriseDB repositories.

```
/etc/zypp/repos.d # zypper refresh
Repository 'Devel project of Java packages for openSUSE:Factory
(SLE_12_SP3)' is up to date.
Repository 'SLE-Module-Legacy12-Pool' is up to date.
Repository 'SLE-Module-Legacy12-Updates' is up to date.
Repository 'SLES12-12-0' is up to date.
Repository 'SLES12-Pool' is up to date.
Repository 'SLES12-Updates' is up to date.
Repository 'SLE-SDK12-Pool' is up to date.
Repository 'SLE-SDK12-Updates' is up to date.
Repository 'SUSE-PackageHub-12' is up to date.
Repository 'SUSE-PackageHub-12-Pool' is up to date.
Repository 'wxWidgets Packages (SLE_12_SP3)' is up to date.
Repository 'Various compilers (SLE_12_SP3)' is up to date.
```

```
Retrieving repository 'EDB Postgres Advanced Server 11 12 –
x86_64' metadata ------------------------[\]

Authentication required for
'https://zypp.enterprisedb.com/11/suse/suse-12-x86_64'
User Name:
Password:

Retrieving repository 'EDB Postgres Advanced Server 11 12 –
x86_64' metadata............................................................................[done]
Building repository 'EDB Postgres Advanced Server 11 12 – x86_64'
cache............................................................................................[done]
Retrieving repository 'EDB Postgres Advanced Server Dependencies
12 – x86_64' metadata.......................................................................[done]
Building repository 'EDB Postgres Advanced Server Dependencies 12
– x86_64' cache..............................................................................[done]
Retrieving repository 'EDB Postgres Advanced Server Tools 12 –
x86_64' metadata............................................................................[done]
Building repository 'EDB Postgres Advanced Server Tools 12 –
x86_64' cache................................................................................[done]
All repositories have been refreshed.
```

When prompted for a `User Name` and `Password`, provide your connection credentials for the EnterpriseDB repository. If you need credentials, contact EnterpriseDB at:

https://www.enterprisedb.com/general-inquiry-form

Before installing EDB Postgres Advanced Server or supporting components, you must also add SUSEConnect and the SUSE Package Hub extension to the SLES host and register the host with SUSE, allowing access to SUSE repositories. Use the commands:

```
zypper install SUSEConnect
SUSEConnect -p PackageHub/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
```

For detailed information about registering a SUSE host, visit:

https://www.suse.com/support/kb/doc/?id=7016626

The following sections provide information about the specific repositories that are not hosted by EnterpriseDB, but that provide software that satisfies dependencies for each listed component. Please note that after adding repository access, you should use the `zypper refresh` command to process the repository content:

```
zypper refresh
```

Java Components:

Please note that only OpenJDK (version 1.8) is supported on SLES hosts of Java components. Before using an RPM package to add Advanced Server or a supporting component that requires Java, use zypper to add supporting repository locations to your system.

Use the commands:

```
zypper addrepo
http://download.opensuse.org/repositories/Java:/Factory/SLE_12_SP
2/Java:Factory.repo
```

```
zypper refresh
```

Then, to install the server and its core components, invoke the command:

```
zypper install edb-as11
```

Slony:

If you are using a package to add Slony to a SLES host, please note that you must add access to the Perl repository:

For SLES 12 SP2:

```
zypper addrepo
http://download.opensuse.org/repositories/devel:languages:perl/SL
E_12_SP2/devel:languages:perl.repo
```

For SLES 12 SP3:

```
zypper addrepo
http://download.opensuse.org/repositories/devel:languages:perl/SL
E_12_SP3/devel:languages:perl.repo
```

Then:

```
zypper refresh
```

PostGIS:

When using an RPM package to add PostGIS to a SLES host, you must add the following repository:

```
zypper addrepo
"http://download.opensuse.org/repositories/server:/Kolab:/3.3/SLE
_12/server:Kolab:3.3.repo"
```

Then:

```
zypper refresh
```

pgAgent:

When adding the pgAgent packages to a SLES host, you must add access to the wxWidgets repository:

For SLES 12 SP2:

```
SUSEConnect -p sle-module-legacy/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
zypper addrepo
http://download.opensuse.org/repositories/X11:/wxWidgets/SLE_12_S
P2/X11:wxWidgets.repo
```

For SLES 12 SP3:

```
SUSEConnect -p sle-module-legacy/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
zypper addrepo
http://download.opensuse.org/repositories/X11:/wxWidgets/SLE_12_S
P3/X11:wxWidgets.repo
```

Then:

```
zypper refresh
```

PEM Server and PEM Agent:

When adding the PEM Server and PEM Agent to a SLES host, you must add access to the wxWidgets repository:

For SLES 12 SP2:

```
SUSEConnect -p sle-module-legacy/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
zypper addrepo
https://download.opensuse.org/repositories/Apache:Modules/SLE_12_
SP2/Apache:Modules.repo
zypper addrepo
http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newto
n:/cisco-apic:/2.3.1/SLE_12_SP2/ pem_opensuse_boost
```

For SLES 12 SP3:

```
SUSEConnect -p sle-module-legacy/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
```

```
zypper addrepo
https://download.opensuse.org/repositories/Apache:Modules/SLE_12_
SP3/Apache:Modules.repo
zypper addrepo
http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newto
n:/cisco-apic:/2.3.1/SLE_12_SP3/ pem_opensuse_boost
```

Then:

```
zypper refresh
zypper install edb-pem-server
```

The PEM server installer includes a script (configure-pem-server.sh) to help automate the configuration process for RPM installations. The script is installed in the /usr/edb/pem/bin directory. To invoke the script, use the command:

```
/usr/edb/pem/bin/configure-pem-server.sh
```

BART:

When installing BART on a SLES host, you must add the SUSE SDK as well as the Archiving repository:

```
SUSEConnect -p sle-sdk/12/x86_64
zypper addrepo
https://download.opensuse.org/repositories/Archiving/SLE_12_SP3/A
rchiving.repo
```

Then:

```
zypper refresh
```

LLVM JIT:

When installing LLVM JIT on a SLES host, you must add the following repository to your system:

```
zypper addrepo
https://download.opensuse.org/repositories/devel:/tools:/compiler
/SLE_12_SP3/devel:tools:compiler.repo
```

Then:

```
zypper refresh
```

Updating Components on a SLES Host:

To update components installed with zypper, use the zypper update command.

4.5 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 EnterpriseDB repository. To request credentials for the repository, visit:

<div align="center">

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

</div>

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

- Assume superuser privileges:

  ```
  sudo su –
  ```

- Configure the EnterpriseDB repository and replace the `username` and `password` placeholders in the following command:

  - On Debian 9:

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

  - On Ubuntu 18:

    - Set the EnterpriseDB repository:

      ```
      sudo su -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'
      ```

    - Substitute your EnterpriseDB credentials for the `username` and `password` placeholders in the following command:

      ```
      sudo su -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-as11-server
```

Note: The OpenJDK version 10 is supported on a Debian or Ubuntu host of Java components. Before using a debian package to add Advanced Server or a supporting component such as MTK or EDB*Plus that requires Java, make sure to install Java version 8 on your system to support EDB Java-based components.

4.5.1 Advanced Server Debian Packages

The table that follows lists some of the Debian packages that are available from EnterpriseDB.

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-as11-server	This package installs core components of the Advanced Server database server.
edb-as11-server-client	The `edb-as11-server-client` package contains client programs and utilities that you can use to access and manage Advanced Server.
edb-as11-server-core	The `edb-as11-server-core` package includes the programs needed to create the core functionality behind the Advanced Server database.
edb-as11-server-dev	The `edb-as11-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-as11-server-doc	The `edb-as11-server-docs` package installs the readme file.
edb-as11-server-edb-modules	This package installs supporting modules for Advanced Server
edb-as11-server-	This package installs Advanced Server's Index Advisor feature.

Package Name	Package Installs
`indexadvisor`	The Index Advisor utility helps determine which columns you should index to improve performance in a given workload.
`edb-as11-server-pldebugger`	This package implements an API for debugging PL/pgSQL functions on Advanced Server.
`edb-as11-server-plpython` `edb-as11-server-plpython3`	This package installs the PL/Python procedural language for Advanced Server. Please note that the `edb-as11-server-plpython` package is dependent on the platform-supplied version of Python. By default, the `edb-as11-server-plpython3` package is not installed; you need to explicitly install the package using `yum`. To install `edb-as11-server-plpython3` package, you must first uninstall the plpython platform-supplied version of Python.
`edb-as11-server-pltcl`	The `edb-as11-pltcl` package installs the PL/Tcl procedural language for Advanced Server. Please note that the `edb-as11-pltcl` package is dependent on the platform-supplied version of TCL.
`edb-as11-server-sqlprofiler`	This package installs Advanced Server's SQL Profiler feature. SQL Profiler helps identify and optimize SQL code.
`edb-as11-server-sqlprotect`	This package installs Advanced Server's SQL Protect feature. SQL Protect provides protection against SQL injection attacks.
`edb-as11-server-sslutils`	This package installs functionality that provides SSL support.
`edb-as11-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-as11-server-parallel-clone`	This package installs functionality that supports the EDB Clone Schema extension.
`edb-as11-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-as11-pgsnmpd`	SNMP (Simple Network Management Protocol) is a protocol that allows you to supervise an apparatus connected to the network.
`edb-as11-pljava`	This package installs PL/Java, providing access to Java stored procedures, triggers and functions via the JDBC interface.
`edb-as11-pgadmin4`	pgAdmin 4 provides a graphical management interface for Advanced Server and PostgreSQL databases.
`edb-as11-pgadmin-apache`	Apache support module for pgAdmin 4.
`edb-as11-pgadmin4-common`	pgAdmin 4 supporting files.
`edb-as11-pgadmin4-doc`	pgAdmin 4 documentation module.
`edb-as11-pgpool37-extensions`	This package creates pgPool extensions required by the server.
`edb-as11-postgis-2.5`	This package installs POSTGIS support for geospatial data.
`edb-as11-postgis-2.5-scripts`	This package installs POSTGIS support for geospatial data.
`edb-as11-postgis-doc-2.5`	This package provides support for POSTGIS.
`edb-as11-postgis-gui-2.5`	This package provides support for POSTGIS.
`edb-as11-postgis-jdbc`	This package provides support for POSTGIS.
`edb-as11-postgis-scripts`	This package provides support for POSTGIS.
`edb-as11-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 Section 5.
`edb-as11-slony-replication`	This package installs the meta RPM for Slony-I.
`edb-as11-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

Package Name	Package Installs
	replicate large databases to a reasonably limited number of standby systems.
`edb-as11-slony-replication-docs`	This package contains the Slony project documentation (in pdf form).
`edb-as11-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 Section 5.
`edb-as11-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-as11-hdfs-fdw`-doc	Documentation for the Hadoop Data Adapter.
`edb-as11-mongo-fdw`	This EnterpriseDB Advanced Server extension implements a Foreign Data Wrapper for MongoDB.
`edb-as11-mongo-fdw-doc`	Documentation for the Foreign Data Wrapper for MongoDB.
`edb-as11-mysql-fdw`	This EnterpriseDB Advanced Server extension implements a Foreign Data Wrapper for MySQL.
`edb-pgpool37`	This package contains the pgPool-II installer. pgPool provides connection pooling for Advanced Server installations.
`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-pgbouncer19`	This package contains PgBouncer (a lightweight connection pooler). This package requires the libevent package.

`edb-efm41`
`edb-efm40`
`edb-efm39`
	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.

4.6 Configuring a Package Installation

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

4.6.1 Creating a Database Cluster and Starting the Service

The PostgreSQL `initdb` command creates a database cluster. 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 use a service configuration file to configure the environment.

To review the `initdb` documentation, visit:

<p align="center">https://www.postgresql.org/docs/11/static/app-initdb.html</p>

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-11-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/as11/bin/edb-as-
11-setup initdb
```

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

```
systemctl { start | stop | restart } edb-as-11
```

On Debian 9.x or Ubuntu 18.04

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

```
/usr/bin/epas_createcluster 11 main2
```

To start a new cluster, use the following command:

```
/usr/bin/epas_ctlcluster 11 main2 start
```

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

```
/usr/bin/epas_lsclusters
```

Note: The data directory is created under `/var/lib/edb-as/11/main2` and configuration directory is created under `/etc/edb-as/11/main/`.

4.6.1.1 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.

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.

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, please refer to the *EDB Postgres Advanced Server Guide* available at:

http://www.enterprisedb.com/products-services-training/products/documentation

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

https://www.postgresql.org/docs/11/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.

4.6.2 Modifying the Data Directory Location on CentOS or Redhat 7.x | 8.x

On a CentOS or RedHat version 7.x | 8.x host, the unit file is named `edb-as-11.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/as11/data` directory. To use a `data` directory that resides in a non-default location, create a copy of the unit file under the `/etc` directory:

```
cp /usr/lib/systemd/system/edb-as-11.service
    /etc/systemd/system/
```

After copying the unit file to the new location, modify the service file (`/etc/systemd/system/edb-as-11.service`) with your editor of choice, correcting any required paths.

Then, use the following command to reload `systemd`, updating the modified service scripts:

```
systemctl daemon-reload
```

Then, start the Advanced Server service with the following command:

```
systemctl start edb-as-11
```

4.7 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-as11-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-11.service
    /etc/systemd/system/secondary-edb-as-11.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/as11/bin/edb-as-11-setup initdb secondary-
    edb-as-11
    ```

- 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-11
    ```

- Start the second cluster with the following command:

    ```
    systemctl start secondary-edb-as-11
    ```

4.8 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 EnterpriseDB 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 to install the `epel-release`, `yum-utils`, and `createrepo` packages:

  On RHEL or CentOS 7:

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

  On RHEL or CentOS 8:

  ```
  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.

- Sync the RPM packages and create the repository.

  ```
  reposync -r edbas11 -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:

```
[edbas11]
name=EnterpriseDB Advanced Server 11
baseurl=https://yum.your_domain.com/edbas11
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:

  ```
  yum -y install edb-as11-server
  ```

- On RHEL or CentOS 8:

  ```
  dnf -y install edb-as11-server
  ```

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

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

5 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 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.x or 8.x 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
```

6 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.

6.1 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-11	edb-as11-main
pgAgent	edb-pgagent-11	edb-as11-pgagent
PgBouncer	edb-pgbouncer-1.14	edb-pgbouncer114
pgPool-II	edb-pgpool-4.1	edb-pgpool41
Slony	edb-slony-replication-11	edb-as11-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.

6.1.1 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:

action

> *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.

service_name

> *service_name* specifies the name of the service.

6.1.2 Controlling a Service on Debian 9.x or Ubuntu 18.04

By default, a server is running with the peer or md5 permission on a Debian or Ubuntu host. You can change the `auth_type` from `pg_hba.conf` file available under:

```
vi /etc/edb-as/11/main/pg_hba.conf
```

To know more about modifying the `pg_hba.conf` file, see Section 6.3.

Please note, all the server configuration files are available under `/etc/edb-as/11/main`.

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

- Discover the current status of a service:

```
/usr/bin/epas_ctlcluster 11 main status
```

- Stop a service:

```
/usr/bin/epas_ctlcluster 11 main stop
```

- Restart a service:

```
/usr/bin/epas_ctlcluster 11 main restart
```

- Reload a service:

```
/usr/bin/epas_ctlcluster 11 main reload
```

- Control the component services:

```
systemctl restart edb-as@11-main
```

6.1.3 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

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

action

>   *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, please see the official PostgreSQL Core Documentation available at:

<div align="center">

https://www.postgresql.org/docs/11/static/app-pg-ctl.html

</div>

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.

6.1.4 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.

6.2 Modifying the postgresql.conf File

Configuration parameters in the `postgresql.conf` file specify server behavior with regards to auditing, authentication, encryption, and other behaviors. The `postgresql.conf` file resides in the `data` directory under your Advanced Server installation.

```
# PostgreSQL Client Authentication Configuration File
# ===================================================
#
# Refer to the "Client Authentication" section in the PostgreSQL
# documentation for a complete description of this file.  A short
# synopsis follows.
#
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which PostgreSQL user names they can use, which
# databases they can access.  Records take one of these forms:
#
# local      DATABASE  USER  METHOD  [OPTIONS]
# host       DATABASE  USER  ADDRESS  METHOD   [OPTIONS]
# hostssl    DATABASE  USER  ADDRESS  METHOD   [OPTIONS]
# hostnossl  DATABASE  USER  ADDRESS  METHOD   [OPTIONS]
#
# (The uppercase items must be replaced by actual values.)
#
# The first field is the connection type: "local" is a Unix-domain
# socket, "host" is either a plain or SSL-encrypted TCP/IP socket,
# "hostssl" is an SSL-encrypted TCP/IP socket, and "hostnossl" is a
# plain TCP/IP socket.
#
# DATABASE can be "all", "sameuser", "samerole", "replication", a
# database name, or a comma-separated list thereof. The "all"
# keyword does not match "replication". Access to replication
# must be enabled in a separate record (see example below).
#
# USER can be "all", a user name, a group name prefixed with "+", or a
# comma-separated list thereof.  In both the DATABASE and USER fields
# you can also write a file name prefixed with "@" to include names
# from a separate file.
#
# ADDRESS specifies the set of hosts the record matches.  It can be a
# host name, or it is made up of an IP address and a CIDR mask that is
# an integer (between 0 and 32 (IPv4) or 128 (IPv6) inclusive) that
# specifies the number of significant bits in the mask.  A host name
# that starts with a dot (.) matches a suffix of the actual host name.
# Alternatively, you can write an IP address and netmask in separate
# columns to specify the set of hosts.  Instead of a CIDR-address, you
# can write "samehost" to match any of the server's own IP addresses,
# or "samenet" to match any address in any subnet that the server is
# directly connected to.
#
# METHOD can be "trust", "reject", "md5", "password", "scram-sha-256",
# "gss", "sspi", "ident", "peer", "pam", "ldap", "radius" or "cert".
# Note that "password" sends passwords in clear text; "md5" or
# "scram-sha-256" are preferred since they send encrypted passwords.
#
# OPTIONS are a set of options for the authentication in the format
```

Figure 6.1 - The postgresql.conf file.

Parameters that are preceded by a pound sign (#) are set to their default value (as shown in the parameter setting). To change a parameter value, remove the pound sign and enter a new value. After setting or changing a parameter, you must either *reload* or *restart* the server for the new parameter value to take effect.

Within the `postgresql.conf` file, some parameters contain comments that indicate `change requires restart` (see Figure 6.1). To view a list of the parameters that require a server restart, execute the following query at the psql command line:

```
SELECT name FROM pg_settings WHERE context = 'postmaster';
```

6.3 Modifying the pg_hba.conf File

Appropriate authentication methods provide protection and security. Entries in the pg_hba.conf file specify the authentication method or methods that the server will use when authenticating connecting clients. Before connecting to the server, you may be required to modify the authentication properties specified in the pg_hba.conf file.

When you invoke the initdb utility to create a cluster, initdb creates a pg_hba.conf file for that cluster that specifies the type of authentication required from connecting clients.

The default authentication configuration specified in the pg_hba.conf file is:

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            ident
# IPv6 local connections:
host    all             all             ::1/128                 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            ident
host    replication     all             ::1/128                 ident
```

To modify the pg_hba.conf file, open the file with your choice of editor. After modifying the authentication settings in the pg_hba.conf file, use the Linux command line to restart the server and apply the changes.

For more information about authentication, and modifying the pg_hba.conf file, see the PostgreSQL Core Documentation at:

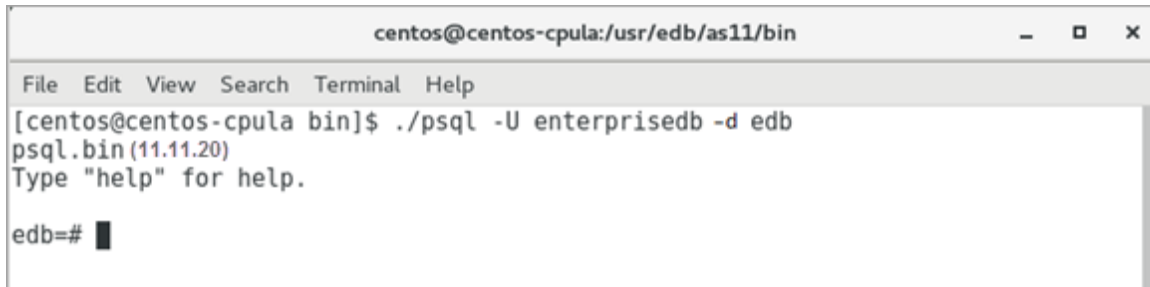https://www.postgresql.org/docs/11/static/auth-pg-hba-conf.html

6.4 Connecting to Advanced Server with psql

`psql` is a command line client application that allows you to execute SQL commands and view the results. To open the 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 psql client (see Figure 6.2):

        psql -d edb -U enterprisedb

Figure 6.2 - Connecting to the server.

Where:

        `-d` specifies the database to which psql will connect;

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

For more information about using the command line client, please refer to the PostgreSQL Core Documentation at:

                https://www.postgresql.org/docs/11/static/app-psql.html

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-as11-server*
```

- On RHEL or CentOS 8:

```
dnf remove edb-as11-server*
```

7.2 Uninstalling Advanced Server Components on a Debian or Ubuntu Host

- To uninstall Advanced Server, invoke the following command:

```
apt-get remove edb-as11-server*
```

Please note: The configuration files and data directory remains intact.

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

```
apt-get purge edb-as11-server*
```