



EDB Postgres™ Advanced Server Installation Guide for Linux

EDB Postgres™ Advanced Server 12

March 17, 2021

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

EnterpriseDB Corporation, 34 Crosby Drive Suite 100, Bedford, MA 01730, USA
T +1 781 357 3390 **F** +1 978 589 5701 **E** info@enterprisedb.com **www**.enterprisedb.com

Table of Contents

1	Introduction.....	4
1.1	Typographical Conventions Used in this Guide.....	5
2	Supported Platforms.....	6
3	Using a Package Manager to Install Advanced Server	7
3.1	Installation Pre-requisites.....	8
3.1.1	Installing Advanced Server on a CentOS Host.....	9
3.1.2	Installing Advanced Server on a RHEL Host.....	11
3.1.3	Installing Advanced Server on a CentOS/RHEL 7 ppc64le Host.....	13
3.1.4	Advanced Server RPM Packages	16
3.1.5	Updating an RPM Installation	19
3.2	Installing Advanced Server on a SLES Host.....	21
3.3	Installing Advanced Server on a Debian or Ubuntu Host	27
3.3.1	Advanced Server Debian Packages	29
3.4	Configuring a Package Installation	32
3.4.1	Creating a Database Cluster and Starting the Service.....	32
3.4.1.1	Specifying Cluster Options with INITDBOPTS	34
3.4.1.1.1	Initializing the Cluster in Oracle Mode	34
3.4.1.1.2	Initializing the Cluster in Postgres Mode.....	35
3.4.2	Modifying the Data Directory Location on CentOS or Redhat 7.x 8.x.....	37
3.5	Starting Multiple Postmasters with Different Clusters.....	38
3.6	Creating an Advanced Server Repository on an Isolated Network	39
4	Installation Troubleshooting	41
5	Managing an Advanced Server Installation	43
5.1	Starting and Stopping Advanced Server and Supporting Components	43
5.1.1	Controlling a Service on CentOS or RHEL 7.x 8.x	44
5.1.2	Controlling a Service on Debian 9.x 10.x or Ubuntu 18.04.....	45
5.1.3	Using pg_ctl to Control Advanced Server	46
5.1.4	Configuring Component Services to AutoStart at System Reboot	47
5.2	Modifying the postgresql.conf File	48
5.3	Modifying the pg_hba.conf File.....	50
5.4	Connecting to Advanced Server with psql	51
6	Uninstalling Advanced Server	52
6.1	Uninstalling an RPM Package.....	52
6.2	Uninstalling Advanced Server Components on a Debian or Ubuntu Host.....	53

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 12 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.

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 7.x and 8.x only. LLVM JIT is not supported on PPC-LE 64 running RHEL or CentOS 7.x or 8.x.

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 Supported Platforms

For information about the platforms and versions supported by Advanced Server 12, visit the EnterpriseDB website at:

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

Note: 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.

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, CentOS, or RHEL: `repoquery --requires --resolve package_name`
- On Debian, Ubuntu: `apt-cache depends package_name`
- On SLES: `zypper search --requires --verbose package_name`

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/as12`.

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 [5.2](#).

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

Component	Location
Executables	<code>/usr/edb/as12/bin</code>
Libraries	<code>/usr/edb/as12/lib</code>
Cluster configuration files	<code>/etc/edb/as12</code>
Documentation	<code>/usr/edb/as12/share/doc</code>
Contrib	<code>/usr/edb/as12/share/contrib</code>
Data	<code>/var/lib/edb/as12/data</code>
Logs	<code>/var/log/as12</code>
Lock files	<code>/var/lock/as12</code>
Log rotation file	<code>/etc/logrotate.d/as12</code>
Sudo configuration file	<code>/etc/sudoers.d/as12</code>
Binary to access VIP without sudo	<code>/usr/edb/as12/bin/secure</code>

Component	Location
Backup area	/var/lib/edb/as12/backups
Templates	/usr/edb/as12/share
Procedural Languages	/usr/edb/as12/lib or /usr/edb/as12/lib64
Development Headers	/usr/edb/as12/include
Shared data	/usr/edb/as12/share
Regression tests	/usr/edb/as12/lib/pgxs/src/test/regress
SGML Documentation	/usr/edb/as12/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 EnterpriseDB 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 1.7 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 the following command:

- 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

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%202004-09-2019.pdf>

After receiving your repository credentials you can:

- Create the repository configuration file.
- Modify the file, providing your user name and password.
- Install 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:

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
```

```
yum makecache
```

b. On CentOS 8:

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

```
dnf makecache
```

- 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 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.

```
[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. Enter 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-as12-server
```

- On CentOS 8:

```
dnf -y install edb-as12-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 [3.4](#), *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.

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:

- 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
```

```
yum makecache
```

b. On RHEL 8:

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

```
dnf makecache
```

- Enable supporting repositories.

On RHEL7, 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"
```

On RHEL 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 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.

```
[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. Enter 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-as12-server
```

- On RHEL 8:

```
dnf -y install edb-as12-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 [3.4](#), *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.

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 Advance Toolchain repository on a CentOS or RHEL 7 ppc64le host:

```
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, 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/toolchain/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
```

```
yum makecache
```

- 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 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.

```
[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. Enter 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-as12-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 [3.4](#), *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.

3.1.4 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 the following command:

- 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-as12-server	This package installs core components of the Advanced Server database server.
edb-as12-server-client	Client programs and utilities that you can use to access and manage Advanced Server.
edb-as12-server-contrib	Installs contributed tools and utilities that are distributed with Advanced Server. Files for these modules are installed in: Documentation: /usr/edb/as12/share/doc Loadable modules: /usr/edb/as12/lib Binaries: /usr/edb/as12/bin
edb-as12-server-core	Includes the programs needed to create the core functionality behind the Advanced Server database.
edb-as12-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 <code>ecpg</code> or <code>ecpgPlus</code> C preprocessor.
edb-as12-server-docs	Installs the readme file.
edb-as12-server-edb-modules	Installs supporting modules for Advanced Server
edb-as12-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-as12-server-libs	Provides the essential shared libraries for any Advanced Server client program or interface.
edb-as12-server-llvmjit	This package contains support for Just in Time (JIT) compiling parts of EDBAS queries.
edb-as12-server-pldebugger	This package implements an API for debugging PL/pgSQL functions on Advanced Server.
edb-as12-server-plperl	Installs the PL/Perl procedural language for Advanced Server. Please note that the <code>edb-as12-server-plperl</code> package is dependent on the platform-supplied version of Perl.
edb-as12-server-plpython edb-as12-server-plpython3	Installs the PL/Python procedural language for Advanced Server. Please note that the <code>edb-as12-server-plpython</code> package is dependent on the platform-supplied version of Python. By default,

EDB Postgres Advanced Server Installation Guide for Linux

Package Name	Package Installs
	the <code>edb-as12-server-plpython3</code> package is not installed; you need to explicitly install the package using <code>yum</code> . To install <code>edb-as12-server-plpython3</code> package, you must first uninstall the <code>plpython</code> platform-supplied version of Python.
<code>edb-as12-server-pltcl</code>	Installs the PL/Tcl procedural language for Advanced Server. Please note that the <code>edb-as12-pltcl</code> package is dependent on the platform-supplied version of TCL.
<code>edb-as12-server-sqlprofiler</code>	This package installs Advanced Server's SQL Profiler feature. SQL Profiler helps identify and optimize SQL code.
<code>edb-as12-server-sqlprotect</code>	This package installs Advanced Server's SQL Protect feature. SQL Protect provides protection against SQL injection attacks.
<code>edb-as12-server-sslutils</code>	This package installs functionality that provides SSL support.
<code>edb-as12-server-cloneschema</code>	This package installs the EDB Clone Schema extension. For more information about EDB Clone Schema, see the EDB Postgres Advanced Server Guide.
<code>edb-as12-server-parallel-clone</code>	This package installs functionality that supports the EDB Clone Schema extension.
<code>edb-as12-pgagent</code>	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 4.
<code>edb-as12-edbplus</code>	The <code>edb-edbplus</code> package contains the files required to install the EDB*Plus command line client. EDB*Plus commands are compatible with Oracle's SQL*Plus.
<code>edb-as12-pgsnmpd</code>	SNMP (Simple Network Management Protocol) is a protocol that allows you to supervise an apparatus connected to the network.
<code>edb-as12-pljava</code>	Installs PL/Java, providing access to Java stored procedures, triggers and functions via the JDBC interface.
<code>edb-as12-pgpool40-extensions</code> <code>edb-as12-pgpool41-extensions</code>	This package creates pgPool extensions required by the server for use with pgpool.
<code>edb-as12-postgis</code>	Installs POSTGIS meta RPMs.
<code>edb-as12-postgis-core</code>	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.
<code>edb-as12-postgis-docs</code>	This package installs pdf documentation of PostGIS.
<code>edb-as12-postgis-jdbc</code>	This package installs the essential jdbc driver for PostGIS.
<code>edb-as12-postgis-utils</code>	This package installs the utilities for PostGIS.
<code>edb-as12-slony-replication</code>	Installs the meta RPM for Slony-I.
<code>edb-as12-slony-replication-core</code>	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.
<code>edb-as12-slony-replication-docs</code>	This package contains the Slony project documentation (in pdf form).
<code>edb-as12-slony-replication-tools</code>	This package contains the Slony <code>altperl</code> 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 4.
<code>edb-as12-libicu</code>	These packages contain supporting library files.

EDB Postgres Advanced Server Installation Guide for Linux

The following table lists the packages for Advanced Server 12 supporting components.

Package Name	Package Installs
edb-pgpool140 edb-pgpool141	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. EnterpriseDB supports the following pgpool-II features: <ul style="list-style-type: none"> • 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 <code>edb-jdbc</code> package includes the <code>.jar</code> files needed for Java programs to access an Advanced Server database.
edb-migrationtoolkit	The <code>edb-migrationtoolkit</code> package installs Migration Toolkit, facilitating migration to an Advanced Server database from Oracle, PostgreSQL, MySQL, Sybase and SQL Server.
edb-oci	The <code>edb-oci</code> 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-pgbouncer114	This package contains PgBouncer (a lightweight connection pooler). This package requires the <code>libevent</code> 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 <code>edb-pem</code> 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

Package Name	Package Installs
	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 libiconv-edb libevent-edb-devel	These packages contain 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*
```

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`.

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 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.

Before installing EDB Postgres Advanced Server or supporting components, you must:

- Install the SUSEConnect to register the host with SUSE, allowing access to SUSE repositories using the following command:

```
zypper install SUSEConnect
```

- Replace `registration code` and `email` with your SUSE registration information. Use the command:

```
SUSEConnect -r 'REGISTRATION_CODE' -e 'EMAIL'
```

- Add SUSE Package Hub extension and sle-sdk extension to the SLES host using the following commands:

```
SUSEConnect -p PackageHub/12.4/x86_64
SUSEConnect -p sle-sdk/12.4/x86_64
```

For detailed information about registering a SUSE host, visit:

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

To add Advanced Server, use the following command to add EnterpriseDB repository configuration files to your SLES host:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

A repository configuration file is created named `edb.repo` in the `/etc/zypp/repos.d` directory.

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

To install Advanced Server:

```
zypper -n install edb-as12-server
```

Then initialize the database cluster:

```
PGSETUP_INITDB_OPTIONS="-E UTF-8" /usr/edb/as12/bin/edb-as-12-  
setup initdb
```

Start the database server:

```
systemctl start edb-as-12
```

Connect to the database server:

```
sudo su - enterprisedb
```

The following sections provide information about the supporting components. You must use the `zypper refresh` command to process the repository content.

Java Components:

The OpenJDK (version 1.8) is supported on SLES host for Java components. Before installing Advanced Server or a supporting component that requires Java, you must install OpenJDK (version 1.8).

```
zypper -n install java-1_8_0-openjdk
```

Then:

```
zypper refresh
```

Install the server:

```
zypper -n install edb-as12-server
```

BART:

When installing BART on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Then:

```
zypper refresh
```

Install BART:

```
zypper -n install edb-bart
```

pgAdmin 4:

When installing pgAdmin 4 on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Then:

```
zypper refresh
```

Install pgAdmin 4:

```
zypper -n install edb-pgadmin4
```

PEM Server:

When installing PEM Server on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Install the following repository for PEM dependencies:

```
zypper addrepo  
https://download.opensuse.org/repositories/Apache:/Modules/SLE_12  
_SP4/Apache:Modules.repo
```

Then:

```
zypper refresh
```

Install PEM server:

```
zypper -n install edb-pem
```

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
```

Failover Manager:

When installing Failover Manager on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Then:

```
zypper refresh
```

Install EFM:

```
zypper -n install edb-efm310
```

JDBC Connector:

When installing JDBC Connector on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Then:

```
zypper refresh
```

Install JDBC:

```
zypper -n install edb-jdbc
```

Migration Toolkit:

When installing Migration Toolkit on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Then:

```
zypper refresh
```

Install Migration Toolkit:

```
zypper -n install edb-migrationtoolkit
```

OCL Connector:

When installing OCL Connector on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Then:

```
zypper refresh
```

Install OCL Connector:

```
zypper -n install edb-oci
```

ODBC Connector:

When installing ODBC Connector on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Then:

```
zypper refresh
```

Install ODBC Connector:

```
zypper -n install edb-odbc
```

pgAgent:

When installing pgAgent on a SLES host, you must add the repository configuration file using the following command:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Then:

```
zypper refresh
```

Install pgAgent:

```
zypper -n install edb-as12-pgagent
```

Updating Components on a SLES Host:

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

Uninstalling Advanced Server on a SLES Host:

To uninstall Advanced Server, assume the identity of a root user and invoke the following command:

```
zypper remove edb-as12-server
```

3.3 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:

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

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 Debian 10 and Ubuntu 18:

- Setup the EnterpriseDB 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'
```

- Substitute your EnterpriseDB 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-as12-server
```

Note: 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.

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

Debian Package Component	Location
Server	/usr/lib/edb-as/12/
Data and Configuration Directory	/var/lib/edb-as/12/main /etc/edb-as/12/main/
pgAgent	/usr/lib/edb-as/12
Pgpool	/usr/edb/pgpool4.0/
Postgis	/usr/lib/edb-as/12/
PGSNMPD	/usr/lib/edb-as/12
Slony Replication	/usr/lib/edb-as/12
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/12/lib
SQL-Protect	/usr/lib/edb-as/12/lib
SSLUTILS	/usr/lib/edb-as/12/lib
PL-PERL	/usr/lib/edb-as/12/lib
PL-PYTHON	/usr/lib/edb-as/12/lib
PLTCL	/usr/lib/edb-as/12/lib
EFM	/usr/edb/efm-4.0
JDBC	/usr/edb/jdbc
MTK	/usr/edb/migrationtoolkit/

3.3.1 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
<code>edb-as12-server</code>	Installs core components of the Advanced Server database server.
<code>edb-as12-server-client</code>	Includes client programs and utilities that you can use to access and manage Advanced Server.
<code>edb-as12-server-core</code>	Includes the programs needed to create the core functionality behind the Advanced Server database.
<code>edb-as12-server-dev</code>	The <code>edb-as12-server-dev</code> package contains the header files and libraries needed to compile C or C++ applications that directly interact with an Advanced Server server and the <code>ecpg</code> or <code>ecpgPlus</code> C preprocessor.
<code>edb-as12-server-doc</code>	Installs the readme file.
<code>edb-as12-server-edb-modules</code>	Installs supporting modules for Advanced Server
<code>edb-as12-server-indexadvisor</code>	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.
<code>edb-as12-server-pldebugger</code>	This package implements an API for debugging PL/pgSQL functions on Advanced Server.
<code>edb-as12-server-plpython</code> <code>edb-as12-server-plpython3</code>	Installs the PL/Python procedural language for Advanced Server. Please note that the <code>edb-as12-server-plpython</code> package is dependent on the platform-supplied version of Python. By default, the <code>edb-as12-server-plpython3</code> package is not installed; you need to explicitly install the package using <code>apt</code> . To install <code>edb-as12-server-plpython3</code> package, you must first uninstall the <code>plpython</code> platform-supplied version of Python.
<code>edb-as12-server-pltcl</code>	Installs the PL/Tcl procedural language for Advanced Server. Please note that the <code>edb-as12-pltcl</code> package is dependent on the platform-supplied version of TCL.
<code>edb-as12-server-sqlprofiler</code>	This package installs Advanced Server's SQL Profiler feature. SQL Profiler helps identify and optimize SQL code.
<code>edb-as12-server-sqlprotect</code>	This package installs Advanced Server's SQL Protect feature. SQL Protect provides protection against SQL injection attacks.
<code>edb-as12-server-sslutils</code>	This package installs functionality that provides SSL support.
<code>edb-as12-server-cloneschema</code>	This package installs the EDB Clone Schema extension. For more information about EDB Clone Schema, see the EDB Postgres Advanced Server Guide.
<code>edb-as12-server-parallel-</code>	This package installs functionality that supports the EDB Clone

EDB Postgres Advanced Server Installation Guide for Linux

Package Name	Package Installs
clone	Schema extension.
edb-as12-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-as12-pgsnmpd	SNMP (Simple Network Management Protocol) is a protocol that allows you to supervise an apparatus connected to the network.
edb-pgadmin4	pgAdmin 4 provides a graphical management interface for Advanced Server and PostgreSQL databases.
edb-pgadmin4-apache	Apache support module for pgAdmin 4.
edb-pgadmin4-common	pgAdmin 4 supporting files.
edb-pgadmin4-doc	pgAdmin 4 documentation module.
edb-as12-pgpool40-extensions edb-as12-pgpool41-extensions	This package creates pgPool extensions required by the server.
edb-as12-postgis-2.5	This package installs POSTGIS support for geospatial data.
edb-as12-postgis-2.5-scripts	This package installs POSTGIS support for geospatial data.
edb-as12-postgis-doc-2.5	This package provides support for POSTGIS.
edb-as12-postgis-gui-2.5	This package provides support for POSTGIS.
edb-as12-postgis-jdbc	This package provides support for POSTGIS.
edb-as12-postgis-scripts	This package provides support for POSTGIS.
edb-as12-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 4.
edb-as12-slony-replication	This package installs the meta RPM for Slony-I.
edb-as12-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-as12-slony-replication-docs	This package contains the Slony project documentation (in pdf form).
edb-as12-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 4.
edb-as12-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-as12-hdfs-fdw-doc	Documentation for the Hadoop Data Adapter.
edb-as12-mongo-fdw	This EnterpriseDB Advanced Server extension implements a Foreign Data Wrapper for MongoDB.
edb-as12-mongo-fdw-doc	Documentation for the Foreign Data Wrapper for MongoDB.
edb-as12-mysql-fdw	This EnterpriseDB Advanced Server extension implements a Foreign Data Wrapper for MySQL.
edb-pgpool40 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. EnterpriseDB supports the following pgpool-II features:

EDB Postgres Advanced Server Installation Guide for Linux

Package Name	Package Installs
	<ul style="list-style-type: none"> • Load balancing • Connection pooling • High availability • Connection limits <p>pgpool-II runs as a service on Linux systems, and is not supported on Windows systems.</p>
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.

3.4 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.

3.4.1 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/12/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-12-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/as12/bin/edb-as-12-setup initdb
```

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


```
systemctl { start | stop | restart } edb-as-12
```

On Debian 9.x / 10.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 12 main2
```

To start a new cluster, use the following command:

```
/usr/bin/epas_ctlcluster 12 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/12/main2` and configuration directory is created under `/etc/edb-as/12/main/`.

3.4.1.1 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`.

3.4.1.1.1 Initializing the Cluster in Oracle Mode

If you initialize the database with 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
<code>dbms_alert</code>	Provides the capability to register for, send, and receive alerts.
<code>dbms_job</code>	Provides the capability for the creation, scheduling, and managing of jobs.
<code>dbms_lob</code>	Provides the capability to manage on large objects.
<code>dbms_output</code>	Provides the capability to send messages to a message buffer, or get messages from the message buffer.
<code>dbms_pipe</code>	Provides the capability to send messages through a pipe within or between sessions connected to the same database cluster.
<code>dbms_rls</code>	Enables the implementation of Virtual Private Database on certain Advanced Server database objects.
<code>dbms_sql</code>	Provides an application interface to the EnterpriseDB dynamic SQL functionality.
<code>dbms_utility</code>	Provides various utility programs.
<code>dbms_aqadm</code>	Provides supporting procedures for Advanced Queueing functionality.
<code>dbms_aq</code>	Provides message queueing and processing for Advanced Server.
<code>dbms_profiler</code>	Collects and stores performance information about the PL/pgSQL and SPL statements that are executed during a performance profiling session.
<code>dbms_random</code>	Provides a number of methods to generate random values.
<code>dbms_redact</code>	Enables the redacting or masking of data that is returned by a query.
<code>dbms_lock</code>	Provides support for the <code>DBMS_LOCK.SLEEP</code> procedure.
<code>dbms_scheduler</code>	Provides a way to create and manage jobs, programs, and job schedules.

Package	Functionality compatible with Oracle Databases
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.

3.4.1.1.2 Initializing the Cluster in Postgres Mode

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.

Package	Functionality non-compatible with Oracle Databases
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, please 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/12/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.

3.4.2 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-12.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/as12/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-12.service
   /etc/systemd/system/
```

After copying the unit file to the new location, create the service file (`/etc/systemd/system/edb-as-12.service`) and include (`/lib/systemd/system/edb-as-12.service`) within the file.

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-12
```

3.5 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-as12-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-12.service
/etc/systemd/system/secondary-edb-as-12.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/as12/bin/edb-as-12-setup initdb secondary-edb-as-12
```

- 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-12
```

- Start the second cluster with the following command:

```
systemctl start secondary-edb-as-12
```

3.6 *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:

- Install the `epel-release`, `yum-utils`, and `createrepo` packages using the following command:

- On RHEL or CentOS 7:

```
yum -y install epel-release
```

```
yum install yum-utils
```

```
yum install createrepo
```

- On RHEL or CentOS 8:

```
dnf -y 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 EnterpriseDB repository at:

<https://repos.enterprisedb.com/>

- Sync the RPM packages and create the repository.

```
reposync -r edbas12 -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:

```
[edbas12]
name=EnterpriseDB Advanced Server 12
baseurl=https://yum.your_domain.com/edbas12
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-as12-server
```

- On RHEL or CentOS 8:

```
dnf -y install edb-as12-server
```

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

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

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.x or 8.x.

```
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
```

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-12	edb-as12-main
pgAgent	edb-pgagent-12	edb-as12-pgagent
PgBouncer	edb-pgbouncer-114	edb-pgbouncer114
pgPool-II	edb-pgpool-4.0	edb-pgpool4.0
Slony	edb-slony-replication-12	edb-as12-slony-replication
EFM	efm-4.0	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:

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.

5.1.2 Controlling a Service on Debian 9.x | 10.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/12/main/pg_hba.conf
```

To know more about Modifying the `pg_hba.conf` file, see Section [5.3](#).

Please note, all the server configuration files are available under `/etc/edb-as/12/main`.

If your installation of Advanced Server resides on version 9.x | 10.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 12 main status
```

- Stop a service:

```
/usr/bin/epas_ctlcluster 12 main stop
```

- Restart a service:

```
/usr/bin/epas_ctlcluster 12 main restart
```

- Reload a service:

```
/usr/bin/epas_ctlcluster 12 main reload
```

- Control the component services:

```
systemctl restart edb-as@12-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

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:

<https://www.postgresql.org/docs/12/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 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. 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';
```

5.3 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/12/static/auth-pg-hba-conf.html>

5.4 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:

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

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/12/static/app-psql.html>

6 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.

6.1 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 following 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-as12-server*
```

- On RHEL or CentOS 8:

```
dnf remove edb-as12-server*
```

6.2 Uninstalling Advanced Server Components on a Debian or Ubuntu Host

- To uninstall Advanced Server, invoke the following command:

```
apt-get remove edb-as12-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-as12-server*
```