



EDB Postgres™ Backup and Recovery Installation and Upgrade Guide

EDB Postgres™ Backup and Recovery 2.4

March 3, 2021

EDB Postgres™ Backup and Recovery Installation Guide
by EnterpriseDB® Corporation
Copyright © 2014 - 2019 EnterpriseDB Corporation. All rights reserved.

EnterpriseDB Corporation, 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA
T +1 781 357 3390 **F** +1 978 467 1307 **E** info@enterprisedb.com **www**.enterprisedb.com

Table of Contents

1	Introduction.....	4
1.1	Typographical Conventions Used in this Guide	5
2	Requirements Overview.....	6
2.1.1	Supported Platforms.....	6
2.1.2	Supported Database Versions	6
2.1.3	Software Requirements.....	7
2.1.3.1	BART Host Components	7
2.1.3.2	Additional Components	7
3	Installing BART using an RPM Package.....	8
3.1	Installing BART using an RPM Package.....	8
3.2	Installing BART on a Debian or Ubuntu Host.....	13
4	Configuration	14
4.1	Establishing the BART user account	14
4.2	Configuring the BART Host.....	15
4.3	Configuring the Database Server	20
4.3.1	Authorizing SSH/SCP Access without a Password	21
4.3.1.1	Enabling Public Key Authentication Usage	21
4.3.1.2	Authorized Public Keys Generation	22
4.3.1.3	BART Connections that Require Authentication without a Password .	24
4.3.2	Setting up a Replication Database User.....	25
4.3.3	Updating the Configuration file (server section)	27
4.3.4	Enabling WAL Archiving.....	31
4.3.4.1	WAL Archiving Configuration.....	31
4.3.4.2	Archive Command Auto Configuration.....	32
4.3.5	Verifying Configuration Settings.....	35
5	Troubleshooting	37
6	Upgrading to BART 2.4.....	38
6.1	Upgrading from BART 2.0 to BART 2.4	39
6.2	Upgrading from BART 2.1, 2.2, or 2.3 to BART 2.4.....	40
7.	Uninstalling BART	42

1 Introduction

The EDB Postgres Backup and Recovery Installation and Upgrade Guide is a comprehensive guide to installing and upgrading EDB Postgres Backup and Recovery. The following sections provide detailed information about installing, upgrading, configuring, and uninstalling BART:

- Section [2](#) provides information about BART installation requirements
- Section [3](#) provides instructions for installing and upgrading BART.
- Section [4](#) provides instructions for configuring BART and the database servers.
- Section [5](#) provides information about troubleshooting.
- Section [6](#) provides information about upgrading to BART 2.4.
- Section [7](#) provides information about uninstalling BART.

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 term(s) 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 requirement for BART installation.

2.1.1 Supported Platforms

BART 2.4 can be installed on the following platforms:

- CentOS 7.x
- Red Hat Enterprise Linux (RHEL) 7.x
- PPC-LE 8 running RHEL or CentOS 7.x
- Ubuntu 18.04 (Bionic)
- Debian 9.x (Stretch)

Note: BART currently runs on 64-bit platforms only. BART 2.4 is no longer supported on CentOS/RHEL/OEL 6.x platforms. It is strongly recommended that EDB products running on these platforms be migrated to a supported platform.

2.1.2 Supported Database Versions

BART 2.4 supports the following database versions.

- Advanced Server versions 9.6, 10, and 11.
- PostgreSQL versions 9.6, 10, 11.

Note: Backups can be taken from database clusters with a WAL segment file size other than the default size of 16 MB. You can use the Advanced Server `initdb --wal-segsize` option to modify the WAL segment file size.

For information about using the Advanced Server `initdb --wal-segsize` option, see Section 3.7 "Customizable WAL Segment File Size" in the *EDB Postgres Advanced Server 11 Guide* available from the EnterpriseDB website located at:

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

2.1.3 Software Requirements

The following sections detail the additional components required for BART installation.

2.1.3.1 BART Host Components

Install the following components on the BART host as well as on any remote database server hosts on which incremental backups are to be restored (these remote hosts also require BART installation):

- EDB Postgres Advanced Server (create database cluster and start the server) or PostgreSQL Postgres libpq library
- Postgres `pg_basebackup` utility program for taking backups from standby servers and for taking full backups from any database server under certain conditions
- For RHEL/CentOS 7, Boost Libraries version 1.53

The BART host components are installed using EnterpriseDB RPM packages. See Section [3.1](#) for installation instructions for these components.

2.1.3.2 Additional Components

In addition to the BART host components, the following components must be enabled and activated:

- The Secure Shell (SSH) server daemon must be enabled and activated on the BART host as well as on any remote database server hosts on which BART will be managing backup and recovery.
- The SSH and Secure Copy (SCP) client programs must be available on the BART host as well as on the remote database server hosts.

See Section [4.3.1](#) for information on SSH and SCP.

BART uses the `pg_basebackup` utility program when taking full backups. Your `pg_basebackup` version must be the same or more recent than the database server that is being backed up. For example, `pg_basebackup` version 10 can back up database server version 10, but it cannot be used to back up database server version 11.

3 Installing BART using an RPM Package

This section outlines the BART installation process if an earlier BART version is not installed on the host.

- Section [3.1](#) describes the BART 2.4 installation process using an RPM package.
- Section [3.2](#) describes the BART 2.4 installation process on a Debian or Ubuntu Host.

3.1 *Installing BART using an RPM Package*

Perform the following steps to install BART 2.4 using an RPM package.

Step 1: Before installing BART, you must install the epel-release package by assuming the superuser privileges.

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.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, see section [5](#) for workaround.

Step 2: You must also 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>

Step 3: After receiving your repository credentials you can:

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

Creating a Repository Configuration File

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

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

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

Modifying the Repository Configuration File

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 the name and password of a registered EnterpriseDB user.

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

Step 4 (Optional): Install the `pg_basebackup` utility program using the server client package. If you do not already have the `pg_basebackup` program installed on the BART host, nor do you later plan to install the PostgreSQL database server or Advanced Server on the BART host, you can install a limited number of files that include the `pg_basebackup` program by installing the server client package with the following command:

```
yum install edb-asxx-server-client
```

In the above command, replace `xx` with the required advanced server version.

An example is shown below:

```
[root@localhost ~]# yum install edb-as10-server-client  
Loaded plugins: fastestmirror, refresh-packagekit, security  
Loading mirror speeds from cached hostfile  
* base: mirror.cs.pitt.edu  
* epel: mirror.cs.princeton.edu  
* extras: mirror.us-midwest-1.nexcess.net  
* updates: mirrors.syringanetworks.net  
edb-repos  
| 2.4 kB    00:00  
edbas10  
| 2.5 kB    00:00  
edbas10/primary db  
| 22 kB    00:00  
enterprisedb-dependencies  
| 2.5 kB    00:00  
enterprisedb-tools  
| 2.5 kB    00:00  
Setting up Install Process
```

EDB Postgres Backup and Recovery Installation and Upgrade Guide

```
Resolving Dependencies
--> Running transaction check
----> Package edb-as10-server-client.x86_64 0:10.1.5-1.rhel6 will be installed
--> Processing Dependency: edb-as10-server-libs(x86-64) = 10.1.5-1.rhel6 for package:
edb-as10-server-client-10.1.5-1.rhel6.x86_64
--> Processing Dependency: libpq.so.5()(64bit) for package: edb-as10-server-client-
10.1.5-1.rhel6.x86_64
--> Running transaction check
----> Package edb-as10-server-libs.x86_64 0:10.1.5-1.rhel6 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

```
=====
Package
Repository              Size              Arch              Version
=====
Installing:
  edb-as10-server-client
  1.rhel6                edbas10           x86_64            10.1.5-
                        1.2 M
Installing for dependencies:
  edb-as10-server-libs
  1.rhel6                edbas10           x86_64            10.1.5-
                        531 k
```

Transaction Summary

```
=====
Install      2 Package(s)
```

Total download size: 1.7 M

Installed size: 7.4 M

Is this ok [y/N]: y

Downloading Packages:

(1/2): edb-as10-server-client-10.1.5-1.rhel6.x86_64.rpm

| 1.2 MB 00:00

(2/2): edb-as10-server-libs-10.1.5-1.rhel6.x86_64.rpm

| 531 kB 00:00

```
-----
Total
```

3.1 MB/s | 1.7 MB 00:00

Running rpm_check_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : edb-as10-server-libs-10.1.5-1.rhel6.x86_64

1/2

Installing : edb-as10-server-client-10.1.5-1.rhel6.x86_64

2/2

Verifying : edb-as10-server-client-10.1.5-1.rhel6.x86_64

1/2

Verifying : edb-as10-server-libs-10.1.5-1.rhel6.x86_64

2/2

Installed:

edb-as10-server-client.x86_64 0:10.1.5-1.rhel6

Dependency Installed:

edb-as10-server-libs.x86_64 0:10.1.5-1.rhel6

Complete!

Step 5: Installing BART

After saving your changes to the configuration file, use the `yum install` command to install BART.

- To install the BART RPM package directly from the EDB Yum Repository website, specify only the package name:

```
yum install edb-bart
```

- To install the local BART RPM package file that has been downloaded to the current working directory, specify the following complete RPM package file name:

```
yum install edb-bart-2.4.x-x.rhel7.x86_64.rpm
```

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.

Step 6 Repeat the installation process described in this section to install BART 2.4 on all remote hosts where incremental backups are to be restored using BART 2.4.

Step 7 Configure BART. For configuration details, refer to Section 4.

To verify the BART installation, invoke the following command:

```
[root@localhost ~]# /usr/bin/bart --version
bart (EnterpriseDB) 2.4.0
```

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.

Note: If the `bart --version` command returns an error stating the `PATH` is not available after switching from root user to another BART user account, adjust the setting of the `PATH` environment variable to include the directory location of the BART 2.4 `bin` subdirectory in the `~/.bashrc` or `~/.bash_profile` files of the following user accounts:

- The BART user account on the BART host. See Section 4 Step 3 for information.
- The remote user account on the remote host to which incremental backups are to be restored. See Section 2.1.5.2 Step 4 of the EDB Postgres Backup and Recovery Guide for details.

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

Upon successful installation, the BART product is installed in the following directory location referred to as `BART_HOME`:

```
/usr/edb/bart
```

3.1.1 Installation Locations

BART components are installed in the following locations:

Table 3-1 - Post-Installation Files

File Name	Location	Description
bart	<i>BART_HOME</i> /bin	BART command line, executable program
bart-scanner	<i>BART_HOME</i> /bin	BART WAL scanner program
bart.cfg.sample	<i>BART_HOME</i> /etc	Sample BART configuration file
xlogreader_ident.so	<i>BART_HOME</i> /lib	Libraries supporting WAL versions
bart_license.txt	<i>BART_HOME</i>	License agreement

3.2 *Installing BART on a Debian or Ubuntu Host*

To install BART on a Debian or Ubuntu host, you must have credentials that allow access to the EnterpriseDB repository. To request credentials for the repository, visit the EnterpriseDB Repository Access Request page at:

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

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.

1. Assume the superuser privileges:

```
sudo su -
```

2. Configure the EnterpriseDB repository:

```
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'
```

3. Add support to your system for secure APT repositories:

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

4. Add the EBD signing key:

```
wget -q -O -
https://username:password@apt.enterprisedb.com/edb-
deb.gpg.key | apt-key add -
```

5. Update the repository metadata:

```
apt-get update
```

6. Install the Debian package:

```
apt-get install edb-bart
```

4 Configuration

To configure BART you must:

1. Establish the BART user account.
2. Configure the BART host.
3. Configure the database server.

4.1 *Establishing the BART user account*

To establish the BART user account, determine the operating system user account that will be used to run the BART command line program. This operating system user is referred to as the *BART user account*.

- The BART user account must:
 - own the BART backup catalog directory.
 - be able to run the `bart` program and the `bart-scanner` program.
 - contain a password-less SSH/SCP connection established to and from each database server managed by BART.
- The `enterprisedb` or `postgres` database user can be selected as the BART user account when the managed database servers are Advanced Server or PostgreSQL (respectively). You can also create a new BART user account.

4.2 Configuring the BART Host

This section describes the initial BART configuration steps that must be performed on the BART host.

Step 1 Use the `bart.cfg.sample` file to create the `bart.cfg` file.

```
cp bart.cfg.sample bart.cfg
```

Step 2 Before configuring parameters, verify the `pg_basebackup` utility program is installed on the BART host. If you have not already done so, see [Section 3.1 Step 7](#) for instructions.

Step 3 Set the environment variable for the BART user account.

Note: If the `libpq` library does not reside in the default installation location, you must add the `libpq` library to the `LD_LIBRARY_PATH` environment variable. The default location of the `libpq` library is `POSTGRES_INSTALL_HOME/lib`. Place the following settings in the BART user account's profile so they take effect upon login:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

export LD_LIBRARY_PATH=/usr/edb/as11/lib:$LD_LIBRARY_PATH
```

Step 4 Create the BART backup catalog.

Note: The BART user account must hold privileges to create subdirectories and files within the location specified in the `backup_path` parameter in the BART configuration file. In the following example, the BART configuration file specifies `/opt/backup` as the parent directory for the BART backup catalog:

```
[BART]
bart_host = bartuser@192.168.2.22
backup_path = /opt/backup
pg_basebackup_path = /usr/edb/as11/bin/pg_basebackup
logfile = /tmp/bart.log
scanner_logfile = /tmp/bart_scanner.log
```

The following example creates and sets the ownership and permissions on the BART backup catalog assuming `bartuser` is the BART user account:

```

su root
mkdir /opt/backup
chown bartuser /opt/backup
chgrp bartuser /opt/backup
chmod 700 /opt/backup

```

If the subdirectory does not exist, BART creates a subdirectory for each database server listed in the configuration file when you invoke the `bart` command line program.

Step 5 Specify some parameters in the server section and others in either the server section or the global section in the bottom of the BART configuration file located in `BART_HOME/etc/bart.cfg` (global section refers to parameters set in the [BART] section and they apply globally to backup and recovery management and to all BART managed database servers).

Note: When a parameter is set in either the [BART] section or the server section of the configuration file, the setting in the server section will override the setting in the [BART] section for that particular database server. If omitted, the default value will be used.

The following table lists the scope of each BART configuration file parameter:

Parameter	Required	Default	Server	Global
<code>bart_host</code>	Yes	N/A		
<code>backup_path</code>	Yes	N/A		
<code>pg_basebackup_path</code>	Yes	N/A		
<code>xlog_method</code>	No	Fetch		
<code>retention_policy</code>	No	0		
<code>wal_compression</code>	No	Disabled		
<code>copy_wals_during_restore</code>	No	Disabled	Yes	yes
<code>logfile</code>	No	Disabled	Yes	yes
<code>scanner_logfile</code>	No	None	Yes	yes
<code>thread_count</code>	No	1	Yes	yes
<code>batch_size</code>	No	49142	Yes	yes
<code>scan_interval</code>	No	0	Yes	yes
<code>mbm_scan_timeout</code>	No	20 seconds	Yes	yes
<code>workers</code>	No	1	Yes	yes

The following is an example of the global section [BART]:

```

[BART]
bart_host = bartuser@192.168.2.22
backup_path = /opt/backup
pg_basebackup_path = /usr/edb/as11/bin/pg_basebackup
retention_policy = 3 MONTHS
logfile = /tmp/bart.log
scanner_logfile = /tmp/bart_scanner.log

```

- **bart_host (required)** - Specify the value for this parameter in the form of `bart_user@bart_host_address`. `bart_host_address` is the IP address of the BART host.

- **backup_path (required)** - Specify the path to the file system parent directory where all BART database server backups and archived WAL files are stored.
- **pg_basebackup_path (required)** - Specify the path to the `pg_basebackup` program that you installed on the BART host. See Section 1.3.1 from the *EDB Postgres Backup and Recovery Guide* for information on `pg_basebackup` restrictions that may apply depending upon the version of chosen. For example:

```
[BART]
bart_host = bartuser@192.168.2.22
backup_path = /opt/backup
pg_basebackup_path = /usr/edb/as11/bin/pg_basebackup
logfile = /tmp/bart.log
scanner_logfile = /tmp/bart_scanner.log
```

- **xlog_method (optional)** - Specify how the transaction log is collected during execution of `pg_basebackup` through the `BACKUP` subcommand.
 - Set `xlog_method` to `fetch` to collect the transaction log files after the backup is completed. This is the default setting.
 - Set to `stream` to stream the transaction log in parallel with the full backup creation. If `stream` is used, the `max_wal_senders` configuration parameter in the `postgresql.conf` file for the affected database servers must account for an additional session for the streaming of the transaction log, (that is, the setting must be a minimum of 2).
- **retention_policy (optional)** - Specify when an active backup should be marked as obsolete. You can specify:
 - `max_number BACKUPS`
 - `max_number DAYS`
 - `max_number WEEKS`
 - `max_number MONTHS`

`max_number` is a positive integer

 - If you do not specify a keyword (`BACKUPS`, `DAYS`, `WEEKS`, or `MONTHS`), the integer is interpreted as `max_number BACKUPS` by default.
 - If the `retention_policy` parameter is not specified, then no backups are marked as obsolete when the `MANAGE` subcommand is used. See Section 3.2 of the *EDB Postgres Backup and Recovery Guide* for information on managing backups using a retention policy.
- **wal_compression (optional)** - Set `wal_compression` to `enabled` to compress the archived WALs files in `gzip` format in the BART backup catalog when the `MANAGE` subcommand is invoked. By default it is set to `disabled`.

Note: The `gzip` compression program must be in the BART user account's `PATH`. See Section 3.4.7 of the *EDB Postgres Backup and Recovery Guide* for information about using the `MANAGE` subcommand for WAL compression.
- **copy_wals_during_restore (optional)** - Use `copy_wals_during_restore` to specify how the archived WAL files are collected when invoking the `RESTORE` subcommand.

Set `copy_wals_during_restore` to:

- enabled to copy the archived WAL files from the BART backup catalog to the `restore_path/archived_wals` directory prior to the database server archive recovery.
- disabled to retrieve the archived WAL files directly from the BART backup catalog during the database server archive recovery. This is the default setting.

During a `RESTORE`, BART will generate a `recovery.conf` file. The `restore_command` in the `recovery.conf` file will be determined by the value specified in the `copy_wals_during_restore` parameter.

If the `RESTORE` subcommand is invoked:

- with the `-c` option, the archived WAL files are copied from the BART backup catalog to the `restore_path/archived_wals` directory, thus overriding any setting of the `copy_wals_during_restore` parameter.
- without the `-c` option, the value specified by the `copy_wals_during_restored` parameter is used.

For more information about the `RESTORE` command, see Section [3.4.8](#) of the *EDB Postgres Backup and Recovery Guide* for additional information.

- **logfile (optional)** - Use `logfile` to specify the path to the location to which output from the `bart` program is written. The default parameter value is `/tmp/bart.log`. The log file will be created the first time you invoke the `bart` command line program using the sample configuration file value. To change the default setting, you must delete the `/tmp/bart.log` file so that a new log file will be created and owned by the new BART user account.
- **scanner_logfile (optional)** - Use `scanner_logfile` to specify the path to the location to which output from the `bart-scanner` program is written. The default parameter value is `/tmp/bart_scanner.log`. The scanner log file will be created the first time you invoke the `bart` command line program using the sample configuration file value. To change the default setting, you must delete the `/tmp/bart.log` file so that a new log file will be created and owned by the new BART user account.
- **thread_count (optional)** - Specify the number of worker threads for copying blocks from the database server to the BART backup catalog when the `BACKUP` subcommand is invoked for incremental backups; the default thread count value is 1. The same parameter applies to full backups where the worker threads copy data files from the database server to the BART backup catalog.
 - If parallel backup is run with N number of worker threads, then it will initiate N+ 1 concurrent connections with the server.
 - The same set of processes are used for the compression operation when taking full backups in order to provide parallel, compressed backups when the `BACKUP` subcommand is specified with the `-z` or `-c` options.

Note: The compression operation does not apply to incremental backups.

- If the `BACKUP` subcommand is invoked with the `--thread-count` option, then the number of worker threads specified by this option overrides any setting of the `thread_count` parameter in the BART configuration file.

- If the `BACKUP` subcommand is invoked without the `--thread-count` option, then the following determines the number of worker threads used:
 - The setting of the `thread_count` parameter in the server section of the BART configuration file overrides the setting of `thread_count` in the global section for that particular database server. If omitted in the server section, the setting of `thread_count` in the global section is used.
 - If the `thread_count` parameter is not specified in either section, the default is 1.

Note: When taking a full backup, if the `thread_count` in effect is only 1, then the `pg_basebackup` utility is used to take the full backup unless the `--no-pg_basebackup` option is specified with the `BACKUP` subcommand.

- **batch_size (optional)** - Specify the number of blocks of memory used for copying modified blocks from the database server to the BART backup catalog when the `BACKUP` subcommand is invoked for incremental backups. Each block is 8192 bytes; the default value is 49142 blocks. The maximum permitted value is 131072 ($131072 * 8192 = 1 \text{ GB}$). The minimum permitted value is 1 ($1 * 8192 = 8192$ bytes). Reduce the setting if the server runs out of memory while executing `pg_read_binary_file()`.
- **scan_interval (optional)** - Specify the number of seconds before forcing a scan of the WAL files in the archive directory of the BART backup catalog. The default value is 0, which means no brute-force scanning will be started.
- **mbm_scan_timeout (optional)** - Specify the number of seconds to wait for MBM files before timing out; the default value is 20 seconds. The `mbm_scan_timeout` parameter value must be greater than 0. If the value is 0 or negative, then an error will be displayed during an incremental backup.

Note: The `mbm_scan_timeout` parameter is applicable only for incremental backup.
- **workers (optional)** - Specify the number of parallel worker processes required to stream the modified blocks of an incremental backup to the restore host. The default value is set to 1.

Step 6 Invoke the `CHECK-CONFIG` subcommand omitting the `-s` option to check the parameter settings in the BART configuration file including `bart_host`, `backup_path`, and `pg_basebackup_path`. The `CHECK-CONFIG` subcommand displays an error message if the required configuration is not properly set.

The following example shows successful checking of the global section of the BART configuration file:

```

• bash-4.1$ bart CHECK-CONFIG
• INFO: Verifying that pg_basebackup is executable
• INFO: success - pg_basebackup (/opt/PostgresPlus/9.6AS/bin/pg_basebackup)
returns version 9.600000

```

4.3 *Configuring the Database Server*

The sections that follow describe the procedure for enabling BART backup and recovery management for a database server:

- Authorizing SSH/SCP access without a password prompt.
- Setting up a replication database user.
- Updating the BART configuration file (server section).
- Enabling WAL archiving. See Section [4.3.4](#).
- Verifying the configuration

Please note that the first three items must be completed before restarting the database server with WAL archiving enabled.

4.3.1 Authorizing SSH/SCP Access without a Password

A fundamental mechanism of BART is the use of the Secure Shell (SSH) and the Secure Copy (SCP) Linux utility programs to copy the backup and WAL files from the BART managed database servers to the BART host as well as to restore backups.

The client/server SSH and SCP connections must not prompt for a password when establishing the connection. A password-less connection uses *authorized public keys*. An authorized public key is the public key of client user account that is to be allowed to connect to the target server. Each client user account generates a public key, which must then be added to the target user account's authorized public keys list on the target server.

The sections that follow describe the following:

- Enable public key authentication usage on the server running the SSH server daemon.
- Provide general instructions on how to set up the authorized public keys file.
- The combination of hosts for BART usage on which a connection must be established without a password prompt.

Specific examples are provided in the *EDB Postgres Backup and Recovery Guide*.

4.3.1.1 Enabling Public Key Authentication Usage

Depending upon the Linux operating system running the SSH server daemon, there may be different steps required to enable the usage of public key authentication. Consult the documentation for your operating system for the platform-specific details.

For example, for CentOS 7, perform the following:

In the SSH server daemon configuration file, `/etc/ssh/sshd_config`, check the following parameter is set to `yes` and is not commented:

```
PubkeyAuthentication yes
```

Reload the configuration file:

```
[root@localhost ssh]# service sshd reload
Reloading sshd: [ OK ]
```

Any of the following commands can be used instead of `service sshd reload`:

```
service sshd stop
service sshd start
service sshd restart
```

Note: For any SSH or SCP errors, examine the following log file:

```
/var/log/secure
```

4.3.1.2 Authorized Public Keys Generation

The target server to which a password-less SSH or SCP connection is to be made must contain an authorized public keys file. The file is named `authorized_keys` and is located under the `USER_HOME/.ssh` directory where `USER_HOME` is the home directory of the user account on the target server that is to be used to establish the remote session.

On each client system that is to make a password-less connection to the target server, the client's public key is generated while logged into the client system with the user account that is to request the SSH or SCP connection. The generated public key must be then copied to the target server and concatenated onto the `USER_HOME/.ssh/authorized_keys` file.

Note: The public key should be appended onto the end of any existing `authorized_keys` file. Any existing `authorized_keys` file should not be replaced in its entirety.

The following are the general instructions for generating a client's public key file and then creating the target server's authorized public keys file.

Step 1 On the client system, log in as the user account that will be initiating the SSH or SCP connection.

Step 2 Change to the user account's home directory and check if there is an existing `.ssh` subdirectory. If not, create one as follows:

```
mkdir .ssh
chown user .ssh
chgrp usergroup .ssh
chmod 700 .ssh
```

Where `user` is the user account name and `usergroup` is the associated group of the user.

Step 3 Generate the public key file with the following command. Accept all prompted defaults and do not specify a passphrase when prompted for one.

```
ssh-keygen -t rsa
```

The public key file named `id_rsa.pub` is created in the `.ssh` subdirectory.

Step 4 Create a copy of file `id_rsa.pub` on the target server.

For example, while logged into the client where you just generated the public key file, use SCP to make a temporary copy of it on the target server:

```
scp ~/.ssh/id_rsa.pub target_user@host_address:tmp.pub
```

Step 5 Log into the target server as *target_user*.

For example, while logged into the client, use SSH to log into the target server:

```
ssh target_user@host_address
```

Step 6 Change to the target user account's home directory and check if there is an existing `.ssh` subdirectory. If not, create one as shown in Step 2.

Step 7 Append the temporary, client's public key file, `tmp.pub`, to the authorized keys file named `authorized_keys`. If an existing authorized keys file does not exist, create a new file, but do not completely replace any existing authorized keys file.

```
cat tmp.pub >> ~/.ssh/authorized_keys
```

Make sure the `authorized_keys` file is only accessible by the file owner and not by groups or other users. If the `authorized_keys` file does not have the required permission setting (600) or it was newly created, change the file permissions as follows:

```
chmod 600 ~/.ssh/authorized_keys
```

Step 8 Delete the temporary public key file, `tmp.pub`.

```
rm tmp.pub
```

Now, when logged into the client system as *user* there should be no prompt for a password when commands such as the following are given:

```
ssh target_user@host_address
```

or

```
scp file_name target_user@host_address:directory_path
```

or

```
scp target_user@host_address:directory_path/file file_name
```

4.3.1.3 BART Connections that Require Authentication without a Password

For BART usage, there are two scenarios that require a password-less SSH/SCP connection:

1. From each BART managed database server (SSH/SCP client) to the BART host (target SSH/SCP server) for supporting WAL archiving in the `postgresql.conf` or `postgresql.auto.conf` `archive_command` parameter.

In this case, the SSH client in which the public key file (`id_rsa.pub`) is generated with the `ssh-keygen -t rsa` command is the database server. The public key file is generated by the user account running the database server.

The target SSH server in which the public key file is to be appended onto the `~/.ssh/authorized_keys` file is the BART host. The `authorized_keys` file is in the BART user account's home directory.

2. From the BART host (SSH/SCP client) to each BART managed database server (target SSH/SCP server) for taking incremental backups and for supporting restoration of the full backup, the archived WAL files, and the modified blocks, which occurs when the `BART RESTORE` subcommand is given.

Note: If backups are to be taken from a given database server host, but restored to a different database server host, the password-less SSH/SCP connections must be configured from the BART host to the database server host from which the backup is to be taken as well as from the BART host to the database server host to which the backup is to be restored.

In this case, the SSH client in which the public key file (`id_rsa.pub`) is generated with the `ssh-keygen -t rsa` command is the BART host. The public key file is generated by the BART user account.

The target SSH server in which the public key file is to be appended onto the `~/.ssh/authorized_keys` file is the database server. The `authorized_keys` file is in the home directory of the user account owning the directory where the database backup is to be restored.

See Section [5.2](#) of the EDB Postgres *Backup and Recovery Guide* for examples of each scenario.

4.3.2 Setting up a Replication Database User

For each Postgres database server that is to be managed by BART, a database user must be chosen to serve as the *replication database user*.

The replication database user serves the following purposes:

- Sets the Postgres `archive_command` configuration parameter when the `INIT` subcommand is invoked.
- Creates backups when the `BACKUP` subcommand is invoked.

The replication database user must be a superuser.

The following example creates a superuser to be the replication database user:

```
CREATE ROLE repuser WITH LOGIN SUPERUSER PASSWORD 'password';
```

The `pg_hba.conf` file must minimally permit the replication database user to have access to the `template1` database as shown for `repuser` in the following example.

The IP address from where the replication database user has access to database `template1` is the location of the BART host.

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host template1 repuser 192.168.2.22/32 md5
host all enterprisedb 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
host replication repuser 192.168.2.22/32 md5
```

For `pg_basebackup` only: The replication database user must also be included in the `pg_hba.conf` file as a replication database connection as shown by the last entry in the example if `pg_basebackup` is to be used for taking any backups such as for standby servers.

The replication database user must be specified with the `user` parameter of the BART configuration file for the database server as shown by the following example:

```
[ACCTG]
host = 192.168.2.24
port = 5444
user = repuser
```

```
cluster_owner = enterprisedb
remote_host = enterprisedb@192.168.2.24
description = "Accounting"
```

There must be no password prompt when connecting to the database server with the replication database user. There are several Postgres standard ways to permit this. A recommended method is to use the `.pgpass` file located in the BART user account's home directory.

For example, if `bartuser` is the BART user account, then the `.pgpass` file located in `/home/bartuser/.pgpass` must contain the following entry.

```
192.168.2.24:5444:*:repuser:password
```

Thus, when `bartuser` runs `BART BACKUP`, the password for the replication database user, `repuser`, is obtained from the `.pgpass` file of `bartuser` to connect to the database server running at `192.168.2.24` on port `5444`.

The `.pgpass` file must contain an entry for each BART managed database server and its corresponding replication database user and password.

4.3.3 Updating the Configuration file (server section)

To manage the backup and recovery of a database server, you must add an entry in the server section of the BART configuration file located in `BART_HOME/etc/bart.cfg`.

Note: The setting in the server section will override the setting in the [BART] section for that particular database server. If omitted, default values will be used.

For each cluster serviced by BART, the following parameters are required:

```
[HR]
host = 192.168.2.24
port = 5432
user = postgres
cluster_owner = postgres
description = "EPAS 11 Server"
allow_incremental_backups = enabled
```

The following table explains the scope of each parameter:

Parameter	Required	Default
[Server Name]	Yes	N/A
backup_name	No	N/A
host	Yes	N/A
port	No	5444
user	Yes	N/A
archive_command	No	N/A
cluster_owner	Yes	Enterprisedb for advanced Server database clusters and installed in the mode compatible with Oracle databases postgres for PostgreSQL database clusters and for Advanced Server database clusters installed in the mode compatible with PostgreSQL databases.
remote_host	Yes	N/A
tablespace_path	No	N/A
retention_policy	No	0
xlog_method	No	Fetch
wal_compression	No	Disabled
copy_wals_during_restore	No	Disabled
allow_incremental_backup	No	Disabled
thread_count	No	1
batch_size	No	49142
scan_interval	No	0
mbm_scan_timeout	No	20
description	Yes	No
workers	No	1

Set the following parameters in the server section of the BART configuration file that applies to the database servers in the server section:

- **[ServerName] (required)** - Specify the server name in the server section of the configuration file. This is the name by which you refer to the database server using BART and the name is case-insensitive when referenced with BART subcommand options. A lowercase conversion of this name is used to create a subdirectory in the BART backup catalog for storing the backups and WAL files for this database server. This parameter is required.
- **backup_name (optional)** – Specify the template for user-defined, friendly names to be assigned to the backups of the database server. The maximum permitted length of backup names is 49 characters.
 - The template is an alphanumeric string that may include the following variables to be substituted by the timestamp values when the backup is taken: 1) `%year` – 4-digit year, 2) `%month` – 2-digit month, 3) `%day` – 2-digit day, 4) `%hour` – 2-digit hour, 5) `%minute` – 2-digit minute, and 6) `%second` – 2-digit second. To include the percent sign (%) as a character in the backup name, specify `%%` in the template.
Note: Do not enclose the template string in quotes even if you want the template to include space characters, otherwise the enclosing quotes are stored as part of the backup name. However, when referenced with the `-i` option by BART subcommands use of space characters in the backup name requires enclosing the backup name in quotes.
 - This parameter can be overridden by the `--backup-name` option of the `BACKUP` subcommand. If this parameter is omitted from the BART configuration file, and the `--backup-name` option with a user-defined name is not specified with the `BACKUP` subcommand, then the backup can only be referenced in BART subcommands by the BART assigned, integer backup identifier.
- **host (required)** - Specify the IP address of the database server to be configured for backup.
- **port (optional)** - Specify the port number identifying the database server instance (that is, the relevant database cluster) to be backed up. The default is port 5444.
- **User (required)** – Specify the replication database user name used by BART to:
 - Establish the connection to the database server for full backups
 - Set the Postgres `archive_command` configuration parameter when running the `INIT` subcommand
 - Take incremental backups. This database user must be a superuser.**Note:** While running as the BART user account, the connection to the database server using this database user must not prompt for a password. Also, the `pg_hba.conf` file must contain a replication connection entry for this database user name. See Section 4.3.2 for more information.
- **archive_command (optional)** - When the `INIT` subcommand is used, the content and variables specified in the BART `archive_command` result in the archive command string to be generated into the Postgres `archive_command` parameter in the `postgresql.auto.conf` file.
Note: This BART `archive_command` parameter in the BART configuration file, and the Postgres `archive_command` parameter in the `postgresql.conf` file and the

postgresql.auto.conf file refer to two different parameters that are to be set in different manners. Ensure to carefully observe the distinction between the BART `archive_command` and the Postgres `archive_command` when configuring these parameters according to the instructions.

The following information applies only to the BART `archive_command` parameter.

- Enclose the command string within single quotes (').
- If the `archive_command` parameter is omitted, it still results in its usage by the `INIT` subcommand as if it were actually specified with a setting of `' scp %p %h:%a/%f'`. Variables are the following:
 - `%p` – path of the file to archive used by the Postgres archiving process
 - `%h` – replaced by the `bart_host` parameter setting
 - `%a` – replaced by the BART `archive_path`
 - `%f` – archived file name used by the Postgres archiving process.

See Section [4.3.4.2](#) for additional information.

- **cluster_owner (required)** – Specify the Linux operating system user account that owns the database cluster. This is typically `enterprisedb` for Advanced Server database clusters installed in the Oracle databases compatible mode, or `postgres` for PostgreSQL database clusters and for Advanced Server database clusters installed in the PostgreSQL databases compatible mode.
- **remote_host (optional)**. Specify the IP address of the remote server to which a backup is to be restored. The value for this parameter must be specified in the form of `remote_user@remote_host_address`.
 - where `remote_user` is the user account on the target database server host that accepts a password-less SSH/SCP login connection and owns the directory where the backup is to be restored.
 - `Remote_host_address` is the IP address of the remote host. For restoring a backup to a remote host or for restoring any backup where `remote_user` and the BART user account are not the same users, either this parameter must be set or it may be specified with the `-r` option with the BART `RESTORE` subcommand.
- **tablespace_path (optional)** – Specify paths to which tablespaces are to be restored specified in the format `OID=tablespace_path;OID=tablespace_path ...`. If the backup is to be restored to a remote host such as specified by the `remote_host` parameter, then the tablespace paths must exist on the remote host. This parameter is optional.
- **allow_incremental_backups (optional)** – Enables usage of the WAL scanner for incremental backups. Permits taking of an incremental backup when the `BACKUP` subcommand is invoked with the `--parent` option.

Set this parameter to:

- enabled to permit incremental backups.
- disabled to disallow incremental backups and thus permit only full backups.
- If the `allow_incremental_backups` parameter is not specified, the default is disabled.

You can refer *EDB Postgres Backup and Recovery Guide* section [3.4.3](#) for information about using the `BACKUP` subcommand for taking incremental backups and section [3.5](#) for information about running the WAL scanner.

- **Description (optional)** – Specify the description of the database server. This parameter is optional.

Set the following parameters if required for that particular database server (see section [4.2 Step 5](#) for details).

- `retention_policy`
- `xlog_method`
- `wal_compression`
- `copy_wals_during_restore.`
- `thread_count.`
- `batch_size.`
- `scan_interval.`
- `mbm_scan_timeout.`
- `workers`

Once you have the BART host and the database servers configured, you can start using BART as described in Section [3](#) of the *EDB Postgres Backup and Recovery Guide*.

The following example shows the configuration settings of three database servers:

```
[ACCTG]
host = 127.0.0.1
port = 5444
user = enterprisedb
cluster_owner = enterprisedb
backup_name = acctg_%year-%month-%dayT%hour:%minute:%second
archive_command = 'cp %p %a/%f'
allow_incremental_backups = enabled
retention_policy = 8 BACKUPS
description = "Accounting"

[MKTG]
host = 192.168.2.24
port = 5444
user = repuser
cluster_owner = enterprisedb
remote_host = enterprisedb@192.168.2.24
allow_incremental_backups = enabled
description = "Marketing"

[HR]
host = 127.0.0.1
port = 5432
user = postgres
cluster_owner = postgres
retention_policy = 4 DAYS
description = "Human Resources"
```

4.3.4 Enabling WAL Archiving

WAL archiving must be enabled for the database server for which BART is to perform backup and recovery management.

For background information about WAL archiving see the *PostgreSQL Core Documentation* available at:

<https://www.postgresql.org/docs/11/static/continuous-archiving.html>

Section [4.3.4.1](#) describes the general, manual configuration process.

Section [4.3.4.2](#) describes a more automated alternative approach.

4.3.4.1 WAL Archiving Configuration

The following configuration parameters must be set in the `postgresql.conf` file to enable WAL archiving:

- Set `wal_level` to `replica` or higher for Postgres versions.
- Set `archive_mode` to `on`.
- Set the `archive_command` to copy the WAL files to the BART backup catalog.
- Set `max_wal_senders` to a value high enough to leave at least one session available for the backup. If the `xlog_method=stream` parameter setting is to be used by this database server as determined in the BART configuration file, the `max_wal_senders` setting must account for an additional session for the transaction log streaming (that is, the setting must be a minimum of 2). See sections [4.2](#) and [4.3](#) for information on the `xlog_method` parameter.

Note: The `archive_command` configuration parameter discussed in this section is located in the `postgresql.conf` file. This *Postgres* `archive_command` parameter is used in a different manner than the *BART* `archive_command` parameter, which may be set in the server sections of the BART configuration file, but only under certain conditions.

The `ARCHIVE_PATH` field displayed by the `BART SHOW-SERVERS` subcommand shows the full directory path where the WAL files should be copied as specified in the Postgres `archive_command` configuration parameter in the `postgresql.conf` file:

```
-bash-4.1$ bart SHOW-SERVERS -s acctg
SERVER NAME      : acctg
HOST NAME       : 192.168.2.24
USER NAME       : repuser
```

```

PORT                : 5444
REMOTE HOST         :
RETENTION POLICY    : none
DISK UTILIZATION    : 0.00 bytes
NUMBER OF ARCHIVES  : 0
ARCHIVE PATH        : /opt/backup/acctg/archived_wals
ARCHIVE COMMAND     : (disabled)
XLOG METHOD          : fetch
WAL COMPRESSION    : disabled
TABLESPACE PATH(s) :
INCREMENTAL BACKUP  : DISABLED
DESCRIPTION         : "Accounting"

```

In the following example, SCP copies the WAL files to directory /opt/backup/acctg/archived_wals at the BART host located at 192.168.2.22 as the bartuser user account.

Using the bartuser account allows the operation to copy to the BART backup catalog owned by bartuser.

```

wal_level = archive          # minimal, archive, or hot_standby
                             # (change requires restart)
.
.
.
archive_mode = on           # allows archiving to be done
                             # (change requires restart)
archive_command = 'scp %p bartuser@192.168.2.22:/opt/backup/acctg/archived_wals/%f'
                             # command to use to archive a logfile segment
                             # placeholders: %p = path of file to archive
                             #                %f = file name only
.
.
.
max_wal_senders = 1        # max number of walsender processes
                             # (change requires restart)

```

The database server must be restarted in order to initiate WAL archiving, but do not do so until you have verified that the full path of the BART backup catalog has been created by some prior BART subcommand, otherwise the archive operation will fail.

Start the WAL scanner by executing the following command:

```
./bart-scanner
```

4.3.4.2 Archive Command Auto Configuration

The Postgres `archive_command` parameter can be automatically configured with the `INIT` subcommand. The `INIT` subcommand invokes the Postgres `ALTER SYSTEM` command to set the Postgres `archive_command` configuration parameter in the `postgresql.auto.conf` file located in the managed database server's `POSTGRES_INSTALL_HOME/data` directory. See Section 3.4.2 of the *EDB Postgres Backup and Recovery Guide* for additional information on the `INIT` subcommand.

The archive command string that the `INIT` subcommand generates into the `postgresql.auto.conf` file is determined by the parameter setting of the BART `archive_command` parameter located in the BART configuration file.

The server section of the BART configuration file can contain a BART `archive_command` parameter to specify the desired format of the archive command string to be generated into the Postgres `archive_command` parameter in the `postgresql.auto.conf` file. If the BART `archive_command` parameter is not set in the server section for a given database server, the command string that is configured uses the following default format:

```
scp %p %h:%a/%f
```

where:

`%p`

Path of the file to archive used by the Postgres archiving process

`%h`

Replaced by the setting of the `bart_host` parameter located in the global section of the BART configuration file

`%a`

Replaced by the *archive path* of where the WAL files are to be stored. The archive path takes the form `backup_path/server_name/archived_wals` where `backup_path` is the BART backup catalog parent directory specified in the global section of the BART configuration file and `server_name` is the lowercase conversion of the database server name specified for this database server in the server section of the BART configuration file.

`%f`

Archived file name used by the Postgres archiving process

The placeholders `%h` and `%a` are replaced by the `INIT` subcommand when creating the archive command string. The placeholders `%p` and `%f` are not replaced by the `INIT` subcommand, but are kept as given to be used by the Postgres archiving process.

For example, to use the default archive command format, the BART configuration file contains the following settings where the BART `archive_command` parameter is omitted from the server section for `ACCTG`:

```
[BART]
bart_host= bartuser@192.168.2.22
```

```

backup_path = /opt/backup
pg_basebackup_path = /usr/edb/as11/bin/pg_basebackup
logfile = /tmp/bart.log
scanner_logfile = /tmp/bart_scanner.log

[ACCTG]
host = 127.0.0.1
port = 5444
user = repuser
cluster_owner = enterprisedb
description = "Accounting"

```

The INIT subcommand is invoked by BART user account `bartuser` as follows:

```

[bartuser@localhost ~]$ bart INIT -s acctg -o
INFO:  setting archive_command for server 'acctg'
WARNING: archive_command is set. server restart is required

```

The BART backup catalog directory will be completed if it has not already been done so.

The resulting Postgres archive command string in the `postgresql.auto.conf` file located in the managed database server's `POSTGRES_INSTALL_HOME/data` directory appears as follows:

```

# Do not edit this file manually!
# It will be overwritten by ALTER SYSTEM command.
archive_command = 'scp %p bartuser@192.168.2.22:/opt/backup/acctg/archived_wals/%f'

```

Note: Run the INIT subcommand with the `-o` option to take advantage of the auto configuration process. This option overrides any existing Postgres `archive_command` setting in the `postgresql.conf` or the `postgresql.auto.conf` file. In addition, the `-o` option must be used to generate the command string if the `archive_mode` configuration parameter is set to `off` even if there are no existing settings of the Postgres `archive_command` in the `postgresql.conf` or `postgresql.auto.conf` files.

In this example, the following BART configuration file is used with an explicit setting of the BART `archive_command` parameter:

```

[BART]
bart_host= enterprisedb@192.168.2.22
backup_path = /opt/backup
pg_basebackup_path = /usr/edb/as11/bin/pg_basebackup
logfile = /tmp/bart.log
scanner_logfile = /tmp/bart_scanner.log

[ACCTG]
host = 127.0.0.1
port = 5444
user = repuser
cluster_owner = enterprisedb
archive_command = 'cp %p %a/%f'
description = "Accounting"

```

The INIT subcommand is invoked by BART user account `enterprisedb` as follows:

```
-bash-4.1$ bart INIT -s acctg -o
INFO:  setting archive_command for server 'acctg'
WARNING: archive_command is set. server restart is required
```

The resulting Postgres `archive_command` parameter in the `postgresql.auto.conf` file appears as follows:

```
# Do not edit this file manually!
# It will be overwritten by ALTER SYSTEM command.
archive_command = 'cp %p /opt/backup/acctg/archived_wals/%f'
```

After generating the desired command string in the `postgresql.auto.conf` file, complete the required WAL archive settings in the `postgresql.conf` file:

- Set `wal_level` to `replica` or higher for Postgres versions
- Set `archive_mode` to `on`.
- Set `max_wal_senders` to a value high enough to leave at least one session available for the backup. If the `xlog_method=stream` parameter setting is to be used by this database server as determined in the BART configuration file, the `max_wal_senders` setting must account for an additional session for the transaction log streaming (that is, the setting must be a minimum of 2). See sections [4.2](#) and [4.3](#) for information on the `xlog_method` parameter.

Restart the database server when you are ready to initiate WAL archiving.

When the database server has been restarted, the `ARCHIVE_COMMAND` field of the `SHOW-SERVERS` subcommand displays the active Postgres archive command as shown by the following example:

```
-bash-4.1$ bart SHOW-SERVERS -s acctg
SERVER NAME      : acctg
HOST NAME       : 127.0.0.1
USER NAME       : repuser
PORT            : 5444
REMOTE HOST     :
RETENTION POLICY : none
DISK UTILIZATION : 48.00 MB
NUMBER OF ARCHIVES : 0
ARCHIVE PATH     : /opt/backup/acctg/archived_wals
ARCHIVE SCOMMAND : cp %p /opt/backup/acctg/archived_wals/%f
XLOG METHOD      : fetch
WAL COMPRESSION : disabled
TABLESPACE PATH(s) :
INCREMENTAL BACKUP : DISABLED
DESCRIPTION     : "Accounting"
```

4.3.5 Verifying Configuration Settings

The `CHECK-CONFIG` subcommand with the `-s` option checks the parameter settings in the database server configuration for which the `-s` option is specified.

```
bart CHECK-CONFIG [ -s server_name ]
```

The `CHECK-CONFIG` subcommand displays an error message if the required configuration is not properly set. The following example shows successful checking of a database server:

```
bash-4.1$ bart CHECK-CONFIG -s mktg
INFO: Checking server mktg
INFO: Verifying cluster_owner and ssh/scp connectivity
INFO: success
INFO: Verifying user, host, and replication connectivity
INFO: success
INFO: Verifying that user is a database superuser
INFO: success
INFO: Verifying that cluster_owner can read cluster data files
INFO: success
INFO: Verifying that you have permission to write to vault
INFO: success
INFO: /opt/backup/mktg
INFO: Verifying database server configuration
INFO: success
INFO: Verifying that WAL archiving is working
INFO: success
INFO: Verifying that bart-scanner is configured and running
INFO: success
```

In addition, certain `postgresql.conf` parameters for the database server must be properly set and the database server must be activated for certain processes. These requirements include the following:

- The `cluster_owner` parameter must be set to the user account owning the database cluster directory.
- A password-less SSH/SCP connection must be set between the BART user and the user account specified by the `cluster_owner` parameter.
- A database superuser must be specified by the BART `user` parameter.
- The `pg_hba.conf` file must contain a replication entry for the database superuser specified by the BART `user` parameter.
- The `archive_mode` parameter in the `postgresql.conf` file must be enabled.
- The `archive_command` parameter in the `postgresql.auto.conf` or the `postgresql.conf` file must be set.
- The `allow_incremental_backups` parameter in the BART configuration file must be enabled for database servers for which incremental backups are to be taken.
- Archiving of WAL files to the BART backup catalog must be in process.
- The WAL scanner program must be running.

5 Troubleshooting

This section provides workaround for installation issues.

Yum cannot access a repository that contains epel-release

If yum cannot access a repository that contains epel-release, you will get an error message:

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

Workaround: In this case, you can download the `EPEL` rpm package and install it manually.

To manually install `EPEL`:

1. Download the rpm package.
2. Assume the superuser privileges and navigate to the directory that contains the package.
3. Install `EPEL` with the command:

```
yum install epel-release
```

6 Upgrading to BART 2.4

This section outlines the process to upgrade to BART 2.4 if an existing BART version is installed on the host.

- Section [6.1](#) describes the upgrade process from BART 2.0 to 2.4.
- Section [6.2](#) describes the upgrade process from BART older versions (except 2.0) to 2.4.

Upgrade Restrictions

Note that the following restrictions apply with regard to older versions of BART (2.0, 2.1, 2.2, and 2.3).

- When using BART 2.4, the BART backup catalog (as specified by the `backup_path` parameter of the BART configuration file) must not be the same directory that was used by the older BART versions for their backup catalogs. In other words, new full backups and incremental backups taken using BART 2.4 must be stored in a new BART backup catalog.
- BART 2.4 cannot be used to take incremental backups for parent full backups or parent incremental backups taken with a BART version older than 2.4. BART 2.4 can only take incremental backups when the parent full backups or parent incremental backup has been taken with BART 2.4.
- BART 2.4 cannot be used to restore incremental backups taken with any BART version older than 2.4. However, BART 2.4 can restore full backups taken with older BART versions.

Note: On any remote host on which incremental backups were restored using the earlier BART version, the same installation process must be performed in a similar manner as on the BART host.

6.1 *Upgrading from BART 2.0 to BART 2.4*

Perform the following steps to upgrade from BART 2.0 to BART 2.4:

Step 1: Install BART 2.4. Refer Section [3.1](#) for BART 2.4 installation details.

Step 2: Save a copy of your BART 2.0 configuration file. The default location of the BART 2.0 configuration file is `/usr/edb/bart2.0/etc/bart.cfg`.

Step 3: Invoke the following command to remove BART 2.0.

```
yum remove edb-bart20
```

Step 4: Place the BART 2.0 configuration file (`bart.cfg`) that you saved in Step 2, in the newly created `/usr/edb/bart/etc` directory. You can use many of the same configuration parameters for BART 2.4, but note that you must use a new directory for the BART backup catalog. A new set of full backups and incremental backups taken using BART 2.4 must be stored in a new BART backup catalog.

To specify an alternative configuration file name or location, use the `-c` option with BART subcommands. For details, see Section [3.3](#) of the *EDB Postgres Backup and Recovery Guide* available at:

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

Note: The `bart.cfg` configuration file is only required on the BART 2.4 host from which you will invoke BART subcommands. BART does not require the `bart.cfg` file on hosts on which an incremental backup will be restored.

Step 5: Adjust the setting of the `PATH` environment variable to include the directory location of the BART 2.4 `bin` subdirectory in the `~/.bashrc` or `~/.bash_profile` files for the following user accounts:

- The BART user account on the BART host. See Section [4.2](#) Step 3 for information.
- The user account on the remote host to which incremental backups are to be restored. For details, see Section [2.1.5.2](#) Step 4 of the *EDB Postgres Backup and Recovery Guide* available at:

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

Step 6: On any remote host on which incremental backups were restored using BART 2.0, the same BART 2.4 installation and BART 2.0 removal process must be performed.

Note: After upgrading to BART 2.4, you must take a new full backup of your system before performing an incremental backup.

6.2 *Upgrading from BART 2.1, 2.2, or 2.3 to BART 2.4*

Perform the following steps to upgrade from BART versions 2.1, 2.2, or 2.3 to BART 2.4:

Step 1: Assume the identity of the BART user account and invoke the following command to stop the BART 2.1/2.2/2.3 WAL scanner program (`bart-scanner`) :

```
bart-scanner STOP
```

Step 2: As the `root` user, upgrade to BART 2.4 with the `yum upgrade` command.

- If you have an existing BART RPM installation, you can use `yum` 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:

```
yum upgrade edb-repo
```

`yum` 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 any installed packages:

```
yum upgrade edb-bart
```

- The following syntax uses the `yum` command to update to the BART RPM package file that has been downloaded to the current working directory. The local package file is installed by using the complete, RPM package file name:

```
yum upgrade edb-bart-2.4.x-x.rhel7.x86_64.rpm
```

Ensure the `backup_path` parameter of the BART 2.4 configuration file is set to a new directory and not to any existing BART 2.1/2.2/2.3 backup catalog. In other words, new full backups and incremental backups taken using BART 2.4 must be stored in a new BART backup catalog.

Note: The `bart.cfg` configuration file is only required on the BART 2.4 host from which you will invoke BART subcommands. BART does not require the `bart.cfg` file on hosts on which an incremental backup will be restored.

Step 3: Repeat the process described in this section to upgrade to BART 2.4 on each remote hosts where an incremental backup will be restored.

See the *EDB Postgres Backup and Recovery Guide* for additional information about restoration of incremental backups on remote hosts; the guide is available at:

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

Note: If the `bart --version` command returns an error stating the `PATH` is not available after switching from root user to another BART user account, adjust the setting of the `PATH` environment variable to include the directory location of the BART 2.4 `bin` subdirectory in the `~/.bashrc` or `~/.bash_profile` files of the following user accounts:

- The BART user account on the BART host. See Section [4.2](#) Step 3 for information.
- The remote user account on the remote host to which incremental backups are to be restored. For details, see Section [2.1.5.2](#) Step 4 of the *EDB Postgres Backup and Recovery Guide* available at:

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

The `PATH` setting should be the same as set for BART 2.1/2.2/2.3 since all versions use `/usr/edb/bart/bin`.

Note: After upgrading to BART 2.4, you must take a new full backup of your system before performing an incremental backup.

7. Uninstalling BART

To uninstall BART, assume the identity of the `root` user and invoke the following command:

```
yum remove edb-bart
```

Uninstalling BART does not delete the backup files and archived WAL files that reside in the BART backup catalog. To permanently delete the backup files and archived WAL files in the BART backup catalog (`/opt/backup`), use the `BART DELETE` subcommand (see Section [3.4.9](#) of the *EDB Postgres Backup and Recovery Guide*), or the Linux command:

```
rm -rf /opt/backup
```