



BART Installation and Upgrade Guide

Release 2.5.1

BART Installation and Upgrade Guide

Apr 29, 2020

Contents

1	Requirements Overview	2
1.1	Supported Platforms	2
1.2	Supported Database Versions	3
1.3	Software Requirements	3
1.3.1	Limitations	3
2	Installing BART	4
2.1	Using an RPM Package to Install BART	4
2.1.1	Installation Locations	6
2.2	Installing BART on a Debian or Ubuntu Host	7
3	Configuring BART	8
3.1	Establishing the BART User Account	8
3.2	Configuring the BART Host	9
3.3	Configuring the Database Server	14
3.3.1	Authorizing SSH/SCP Access	14
3.3.1.1	Enabling Public Key Authentication Usage	14
3.3.1.2	Authorized Public Key Generation	15
3.3.1.3	BART Connections that Require Authentication without a Password	16
3.3.2	Setting up a Replication Database User	17
3.3.3	Updating the Server Configuration Parameters	18
3.3.4	Enabling WAL Archiving	22
3.3.4.1	WAL Archiving Configuration	22
3.3.4.2	Archive Command Auto Configuration	23
3.3.5	Verifying Configuration Settings	26
4	Troubleshooting	27
5	Performing a BART Upgrade	28
5.1	Upgrading from BART 2.1, 2.2, 2.3 or 2.4 to BART 2.5.1	29
5.2	Upgrading from BART 2.0 to BART 2.5.1	30
6	Uninstallation	31
	Index	32

The EDB Postgres Backup and Recovery Installation and Upgrade Guide is a comprehensive guide to install and upgrade EDB Postgres Backup and Recovery. This guide provides detailed information about BART installation:

- The *Requirements Overview* section provides information about BART installation requirements and limitations.
- The *Using an RPM Package to Install BART* section provides instructions for installing and upgrading BART.
- The *Configuring BART* section provides instructions for configuring BART and the database servers.
- The *Troubleshooting* section provides information about installation troubleshooting.
- The *Performing a BART Upgrade* section provides information about upgrading BART.
- The *Uninstallation* section provides information about uninstalling BART.

The following sections detail the requirement for BART installation.

1.1 Supported Platforms

BART 2.5.1 can be installed on the following 64-bit platforms:

- CentOS 6.x or 7.x
- Red Hat Enterprise Linux (RHEL) 6.x or 7.x
- PPC-LE 8 running RHEL or CentOS 7.x
- Ubuntu 18.04 (Bionic)
- Debian 9.x (Stretch)

1.2 Supported Database Versions

BART 2.5.1 supports the following database versions:

- Advanced Server versions 9.5, 9.6, 10, 11, 12
- PostgreSQL versions 9.5, 9.6, 10, 11, 12

1.3 Software Requirements

The following sections detail the components required for BART installation.

‘BART Host Components‘

Use EnterpriseDB packages to add BART host components; see *Installing BART using an RPM Package* for installation instructions.

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 *Authorizing SSH/SCP Access* for information about installing and configuring 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 to be backed up. For example, `pg_basebackup` version 10 can back up database server version 10, but cannot be used to back up database server version 11.

1.3.1 Limitations

- BART can take backups 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 the *Customizable WAL Segment File Size* section in the EDB Postgres Advanced Server User Guide, available at:

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

- BART only supports taking a full backup of standby servers. Incremental backups and parallel backups cannot be taken from standby servers.

Installing BART

The instructions in this section will walk you through installing BART on a host on which an earlier BART version has not been installed:

- *Using an RPM Package to Install BART* describes the BART installation process using an RPM package.
- *Installing BART on a Debian or Ubuntu Host* describes the BART installation process on a Debian or Ubuntu host.

2.1 Using an RPM Package to Install BART

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
```

Note: You must enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum.repos.d`).

The EPEL package contains supporting libraries required by BART; if yum cannot access a repository that contains `epel-release`, see the *Troubleshooting* section for help.

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:

1. Create the repository configuration file.
2. Modify the file, providing your user name and password.
3. 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 file, providing your user name and password

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 This step is 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, 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.

To install the `as11` server client package, execute the command:

```
yum install edb-as11-server-client
```

Step 5 Installing BART

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

- To install the BART RPM package directly from the EDB Yum repository website with the following command:

```
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-2.5.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. 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.

Step 6 Repeat the installation process described in this section to install BART on each remote host on which an incremental backup is to be restored.

Step 7 Configure BART. For configuration details, refer to [Configuration](#) Section.

You can use the following command to verify the BART installation:

```
/usr/bin/bart --version
```

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

- The BART user account on the BART host. See the *Configuration* Section (Step 3) for details.
- The remote user account on the remote host to which incremental backups are to be restored. For details, see the *Restoring an Incremental Backup on a Remote Host* Section of the EDB Postgres Backup and Recovery User Guide available at:

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

Upon successful installation, BART is installed in the `BART_HOME` directory:

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

2.1.1 Installation Locations

BART components are installed in the following locations:

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

2.2 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. For information about requesting credentials, visit:

<https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%202004-09-2019.pdf>

Perform the following steps to install a Debian package using the EnterpriseDB apt repository.

1. Assume the superuser privileges:

```
sudo su -
```

2. Configure the EnterpriseDB repository:

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

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

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

4. Add the EBD signing key; When invoking the command, replace the username and password with the credentials provided by EnterpriseDB:

```
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
```

To configure BART, you must:

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

3.1 Establishing the BART User Account

The *BART user account*, is an operating system user that will run the BART command line program. The BART user account must:

- own the BART backup catalog directory.
- be able to run the `bart` program and the `bart-scanner` program.
- use a password-less SSH/SCP connection established to and from each database server managed by BART.

You can optionally use the `enterprisedb` database user as the BART user account for an Advanced Server database, or the `postgres` database user as the BART user account for a PostgreSQL server. If you do not wish to use an existing database user as the BART user account, you must create an operating system user to assume the role.

3.2 Configuring the BART Host

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

Step 1. Copy the `bart.cfg.sample` file to create the `bart.cfg` file. The BART configuration file is located in `BART_HOME/etc/bart.cfg`. For example, you can use the following command to create the `bart.cfg` file:

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

Step 2. Confirm that the `pg_basebackup` utility program is installed on the BART host. The `pg_basebackup` utility resides in the `bin` directory under your Postgres installation.

Step 3. Ensure that the `LD_LIBRARY_PATH` environment variable includes the location of the `libpq` library. If your `libpq` library does not reside in the default location (`POSTGRES_INSTALL_HOME/lib`), you must add the path to the library to the `LD_LIBRARY_PATH` environment variable. Modify the BART user account's profile so the settings 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. 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
```

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

```
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 values for required BART configuration parameters; some parameters must be set in the server section, while others may be set in either the server section or the global section. The BART configuration file is located in `BART_HOME/etc/bart.cfg`.

In the table that follows, `global` refers to parameters set in the `[BART]` section. Global parameters apply to backup and recovery management on all BART managed database servers. The following table lists the scope of each BART configuration file parameter:

Parameter	Required	Default	Server	Global
<bart_host>	Mandatory	N/A	N/A	yes
<backup_path>	Mandatory	N/A	N/A	yes
<pg_basebackup_path>	Mandatory	N/A	N/A	yes
xlog_method	Optional	Fetch	yes	yes
retention_policy	Optional	<max_number> BACKUPS	yes	yes
wal_compression	Optional	Disabled	Yes	yes
copy_wals_during_restore	Optional	Disabled	Yes	yes
logfile	Optional	/tmp/bart.log	N/A	yes
scanner_logfile	Optional	/tmp/bart_scanner.log	N/A	yes
<bart_socket_directory>	Optional	/tmp	N/A	yes
<thread_count>	Optional	1	Yes	yes
<batch_size>	Optional	49152	Yes	yes
<scan_interval>	Optional	0	Yes	yes
<mbm_scan_timeout>	Optional	20 seconds	Yes	yes
<workers>	Optional	1	Yes	yes

The following parameters are 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)** - `bart_host_address` is the IP address of the BART host. Specify the value for this parameter in the form of `bart_user@bart_host_address`.
- **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. For information about `pg_basebackup` version-specific restrictions, see the *pg basebackup Restrictions* Section of the *EDB Postgres Backup and Recovery User Guide* available at:

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

- **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`. `max_number` is a positive integer; you can specify:

```
max_number BACKUPS
max_number DAYS
max_number WEEKS
```

max_number MONTHS

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. For information about managing backups using a retention policy, see *Managing Backups* in the *EDB Postgres Backup and Recovery User Guide* available at:

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

- **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.

For information about using the MANAGE subcommand for WAL compression, see the *EDB Postgres Backup and Recovery User Guide* available at:

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

- **copy_wals_during_restore (optional)** - Use copy_wals_during_restore to specify how 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, recovery settings will be saved in the postgresql.auto.conf file. The restore_command in the postgresql.auto.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_restore parameter is used.

For more information about the RESTORE command, see the *EDB Postgres Backup and Recovery User Guide* available at:

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

- **logfile (optional)** - Use logfile to specify the path to the location to which output from the bart program is written. The default location 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 location to which output from the bart-scanner program is written. The default location 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_scanner.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.
 - `thread_count` will not be effective if the backup is taken on a standby 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 49152 blocks. The maximum permitted value is 131072 ($131072 * 8192 = 1 \text{ GB}$). The minimum permitted value is 1 ($1 * 8192 = 8192 \text{ bytes}$). 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 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/9.5AS/bin/pg_basebackup) returns
version 9.500000
```

3.3 Configuring the Database Server

This section describes the procedure for enabling BART backup and recovery management for a database server:

- Authorizing SSH/SCP access without a password prompt.
- Creating and configuring a replication database user.
- Updating the BART configuration file (server section).
- Enabling WAL archiving of the server.
- Verifying the server configuration settings.

Note: You must authorize SSH/SCP access and set up a replication database user before restarting the database server with WAL archiving enabled.

3.3.1 Authorizing SSH/SCP Access

BART uses the Secure Shell (`ssh`) and 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` commands must not prompt for a password when establishing a connection with the server. A password-less connection uses *authorized public keys* to authenticate with the server. An *authorized public key* is the public key of a client user account that is authorized to connect to the target server. You must add the public key of each client user account to the target user account's authorized public keys list on the target server.

The sections that follow describe how to:

- Enable public key authentication on the server running the SSH server daemon.
- Configure 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 Reference Guide*, available at:

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

3.3.1.1 Enabling Public Key Authentication Usage

The following example enables SSH/SCP access on a CentOS 6.x host; similar (platform-specific) steps will apply to other platforms/versions.

1. First, enable public key authentication; in the SSH server daemon configuration file (`/etc/ssh/sshd_config`) ensure that the following parameter is set to `yes` and is not commented:

```
PubkeyAuthentication yes
```

2. 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
```

If you get any SSH or SCP errors, examine the following log file:

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

3.3.1.2 Authorized Public Key Generation

The target server (the server to which a password-less connection is being made) must contain an `authorized_keys` file located under the `USER_HOME/.ssh` directory. `USER_HOME` is the home directory of the user account on the target server that will be used to establish the remote session.

The generated public key of each client that will connect to the target server must be copied to the target server and concatenated onto the `USER_HOME/.ssh/authorized_keys` file. 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 general instructions will walk you through generating a client's public key file and 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. Navigate to the user account's home directory and check for an existing `.ssh` subdirectory. If the `.ssh` directory does not exist, use the following commands to create it:

```
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. Navigate into the target user account's home directory and check for 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
```

3.3.1.3 BART Connections that Require Authentication without a Password

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

- When connecting from each BART managed database server (SSH/SCP client) to the BART host (target SSH/SCP server) to support WAL archiving as implemented by the `archive_command` parameter. In this case, the public key file (`id_rsa.pub`) is generated with the `ssh-keygen -t rsa` command on the host of the database server.

The public key file should be generated by the user account running the database server. The public key file name should be appended to the `~/.ssh/authorized_keys` file on the BART host. The `authorized_keys` file is in the BART user account's home directory.

- When connecting 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.

In this case, the public key file (`id_rsa.pub`) is generated with the `ssh-keygen -t rsa` command on the BART host. The public key file is generated by the BART user account. The public key file name should be appended to the `~/.ssh/authorized_keys` file on the host of the database server. The `authorized_keys` file is in the home directory of the user account that owns the directory where the database backup is to be restored.

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.

For examples of each scenario, see the *EDB Postgres Backup and Recovery Reference Guide* available at:

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

3.3.2 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:

- 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`.

When executed with the `PSQL` client, the following PostgreSQL command 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 which 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
```

When `bartuser` invokes a 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.

3.3.3 Updating the Server Configuration Parameters

To manage the backup and recovery of a database server, you must add entries to the server section of the BART configuration file (`BART_HOME/etc/bart.cfg`). Settings in the server section will override the settings 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 lists the server-specific parameters and their default values (where applicable):

Parameter	Required	Default
[Server Name]	Mandatory	N/A
<backup_name>	Optional	N/A
host	Mandatory	N/A
port	Optional	5444
user	Mandatory	N/A
<archive_command>	Optional	N/A
<cluster_owner>	Mandatory	enterprisedb for Advanced Server database clusters installed in the mode compatible with Oracle databases. postgres for PostgreSQL database clusters and for Advanced Server database clusters installed in PostgreSQL mode.
<remote_host>	Optional	N/A
<tablespace_path>	Optional	N/A
retention_policy	Optional	<max_number> BACKUPS
xlog_method	Optional	Fetch
wal_compression	Optional	Disabled
copy_wals_during_restore	Optional	Disabled
allow_incremental_backup	Optional	Disabled
thread_count	Optional	1
batch_size	Optional	49152
scan_interval	Optional	0
mbm_scan_timeout	Optional	20
description	Optional	N/A
workers	Optional	1

Set the following parameters in the server section of the BART configuration file. The parameter setting in the server section overrides the setting in the global [BART] section for that particular database server. If omitted, the default value will be used.

- **[ServerName] (required)** - Specify the server name; this parameter is required. This is the name by which you refer to the database server when using BART; 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.

- **backup_name (optional)** – Specify the template for user-defined, friendly names that will 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 that will be replaced with the timestamp values when the backup is taken:

`%year` – 4-digit year

`%month` – 2-digit month

`%day` – 2-digit day

`%hour` – 2-digit hour

`%minute` – 2-digit minute

`%second` – 2-digit second

To include a percent sign (%) as a character in the backup name, specify %% in the template.

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.

While running as the BART user, the connection to the database server must not prompt for a password. The `pg_hba.conf` file must contain a replication connection entry for this database user name. See [Setting up a Replication Database User](#) 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.

The BART `archive_command` parameter in the BART configuration file, and the Postgres `archive_command` parameter in the `postgresql.conf` file (or the `postgresql.auto.conf` file) refer to two different parameters that are to be set in different manners. You should carefully observe the distinction between the BART `archive_command` and the PostgreSQL `archive_command` when configuring these parameters.

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`. The variables represent:
 - * `%p` – The path of the file to archive used by the Postgres archiving process.

- * %h – Will be replaced by the `bart_host` parameter setting.
- * %a – Will be replaced by the BART `archive path`.
- * %f – The archived file name used by the Postgres archiving process.

See *Archive Command Auto Configuration* 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 compatible mode, or `postgres` for Advanced Server database clusters installed in the PostgreSQL compatible mode and PostgreSQL database clusters.
- **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 in the format `OID = tablespace_path`; If the backup is to be restored to a remote host specified by the `remote_host` parameter, then the tablespace paths must exist on the remote host.
- **allow_incremental_backups (optional)** – Enables use of the WAL scanner for incremental backups. Permits taking incremental backups 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`.

For information about using the `BACKUP` subcommand and running the WAL scanner, please see the EDB Postgres Backup and Recovery User Guide available at:

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

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

Refer to the *Configuring the BART host* section for information about configuring the following optional parameters.

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

After configuring the BART host and the database server(s), you can start using BART. For information about using BART, see the *EDB Postgres Backup and Recovery User Guide* available at:

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

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"
```

3.3.4 Enabling WAL Archiving

WAL archiving must be enabled for the database server for which BART is to perform backup and recovery management. For detailed information about WAL archiving, see the [PostgreSQL Core Documentation](#). The following sections provide information about configuring WAL Archiving for BART:

- The *WAL Archiving Configuration* section describes the manual WAL Archiving Configuration process.
- The *Archive Command Auto Configuration* section describes an automated WAL Archiving process.

3.3.4.1 WAL Archiving Configuration

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

- Set `wal_level` to `archive` for Postgres 9.5 or to `replica` for Postgres 9.6 or later.
- Set `archive_mode` to `on`.
- Set the PostgreSQL `archive_command` parameter to copy the WAL files to the BART backup catalog. The `archive_command` configuration parameter mentioned here is located in the `postgresql.conf` file; the PostgreSQL `archive_command` parameter is used in a different manner than the BART `archive_command` parameter previously mentioned in this guide.
- 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 (the setting must be a minimum of 2). See *Configuring the BART host* for information about the `xlog_method` parameter.

The `ARCHIVE_PATH` field displayed by the `BART SHOW-SERVERS` subcommand displays the full directory path where the WAL files should be copied as specified in the PostgreSQL `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"
```

The parameter settings in the following example will copy the WAL files to a directory named `/opt/backup/acctg/archived_wals` on the BART host located at `192.168.2.22` as the `bartuser` user account. Using the `bartuser` account ensures that the operation will have sufficient permissions 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 or the archive operation will fail.

Start the WAL scanner by executing the following command:

```
./bart-scanner
```

3.3.4.2 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. For additional information about the `INIT` subcommand, see the EDB Postgres Backup and Recovery User Guide available at:

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

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* 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 `archive` for Postgres 9.5 or to `replica` for Postgres 9.6 or later.
- 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 *Configuring the BART host* section 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"

```

3.3.5 Verifying Configuration Settings

The CHECK-CONFIG subcommand with the `-s` option checks the parameter settings of the database server specified after the `-s` option:

```
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 the results from a successful configuration:

```
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
```

The command confirms that:

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

This section provides workaround for the following installation issue:

```
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:

To fix this issue, you must 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
```

Performing a BART Upgrade

This section outlines the process of upgrading to BART 2.5.1 if an existing BART version is installed on the host.

- *Upgrading from BART 2.0 to BART 2.5.1* describes the upgrade process from BART 2.0 to 2.5.1.
- *Upgrading from BART 2.1, 2.2, 2.3 or 2.4 to BART 2.5.1* describes the upgrade process from previous BART versions (except 2.0) to 2.5.1.

Upgrade Restrictions

The following restrictions apply with regard to previous BART versions (2.0, 2.1, 2.2, 2.3, and 2.4).

- When using the BART 2.5.1, 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 previous BART versions for their backup catalogs. New full backups and incremental backups taken using BART 2.5.1 must be stored in a new BART backup catalog.
- BART 2.5.1 cannot be used to take incremental backups for parent full backups or parent incremental backups taken with a BART version older than 2.5.1. BART 2.5.1 can only take incremental backups when the parent full backups or parent incremental backup has been taken with BART 2.5.1.
- BART 2.5.1 cannot be used to restore *incremental* backups taken with any BART version older than 2.5.1. However, BART 2.5.1 can restore *full* backups taken with older BART versions.

5.1 Upgrading from BART 2.1, 2.2, 2.3 or 2.4 to BART 2.5.1

Perform the following steps to upgrade from BART versions 2.1, 2.2, 2.3, or 2.4 to BART 2.5.1:

Step 1: Assume the identity of the BART user account and invoke the following command to stop the BART 2.x WAL scanner program (`bart-scanner`):

```
bart-scanner STOP
```

Step 2: As the `root` user, upgrade to BART 2.5.1 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
```

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

```
yum upgrade edb-bart-2.5.x-x.rhel7.x86_64.rpm
```

Ensure the `backup_path` parameter of the BART 2.5.1 configuration file is set to a new directory and not to any existing BART 2.x backup catalog. New full backups and incremental backups taken using BART 2.5.1 must be stored in a new BART backup catalog.

Note: The `bart.cfg` configuration file is only required on the BART 2.5.1 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 2.5.1 on each remote hosts where an incremental backup will be restored.

For additional information about restoration of incremental backups on remote hosts, see the *EDB Postgres Backup and Recovery User Guide* available at:

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

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 location of the BART 2.5 executable (the `bin` subdirectory) in the `~/.bashrc` or `~/.bash_profile` files of the following user accounts:

- The BART user account on the BART host. See *Configuring the BART host* for more information about BART user account.
- The remote user account on the remote host to which incremental backups are to be restored. For details, see the *EDB Postgres Backup and Recovery User Guide* available at:

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

The `PATH` setting should be the same as set for BART 2.x since all versions use `/usr/edb/bart/bin`.

Note: After upgrading to BART 2.5.1, you must take a new full backup of your system before performing an incremental backup.

5.2 Upgrading from BART 2.0 to BART 2.5.1

Perform the following steps to upgrade BART 2.0 to BART 2.5.1:

Step 1: Install BART 2.5.1; for details, see *Using an RPM Package to Install BART*.

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 2.5.1, 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 2.5.1 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 the EDB Postgres Backup and Recovery User Guide available at:

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

Note: The `bart.cfg` configuration file is only required on the BART 2.5.1 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 location of the BART 2.5.1 executable (the `bin` subdirectory) in the `~/.bashrc` or `~/.bash_profile` files for the following user accounts:

- The BART user account on the BART host. See *Configuring the BART host*, Step 3 for more information.
- The user account on the remote host to which incremental backups will be restored. For details, see the *EDB Postgres Backup and Recovery User Guide* available at:

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

Step 6: Perform the BART 2.5.1 installation and BART 2.0 removal process on each remote host on which an incremental backup was restored using BART 2.0.

Note: After upgrading to BART 2.5.1, you must take a new full backup of your system before performing an incremental backup.

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 (for information about `BART DELETE` subcommand refer the EDB Postgres Backup and Recovery User Guide available at <https://www.enterprisedb.com/edb-docs/>) or the following Linux command:

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

A

Archive Command Auto Configuration, 23
Authorized Public Key Generation, 15
Authorizing SSH/SCP Access, 14

B

BART Connections that Require Authentication without a Password, 16

C

Configuring BART, 8
Configuring the BART Host, 9
Configuring the Database Server, 14

E

Enabling Public Key Authentication Usage, 14
Enabling WAL Archiving, 22
Establishing the BART User Account, 8

H

host components, 3

I

Installing BART on a Debian or Ubuntu Host, 7

P

Performing a BART Upgrade, 28
prerequisites, 3

R

Requirements Overview, 2

S

Setting up a Replication Database User, 17
Software Requirements, 3
Supported Database Versions, 3
Supported Platforms, 2

T

Troubleshooting, 27

U

Uninstallation, 31
Updating the Server Configuration Parameters, 18
Upgrading from BART 2.0 to BART 2.5.1, 30
Upgrading from BART 2.1, 2.2, 2.3 or 2.4 to BART 2.5.1, 29
Using an RPM Package to Install BART, 4

V

Verifying Configuration Settings, 26

W

WAL Archiving Configuration, 22