EDB Postgres™ Backup and Recovery Quickstart Guide

EDB Postgres™ Backup and Recovery 2.4

March 3, 2021

EDB Postgres™ Backup and Recovery Quickstart Guide
by EnterpriseDB® Corporation

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

1 Quickstart

This document provides shortcuts that allow you to quickly install and configure the EDB Postgres Backup and Recovery Tool (BART), and take a full and incremental backup of a database server.

Please note that your system may have requirements that are not addressed in this document. For detailed information about BART installation and configuration, see the *EDB Postgres Backup and Recovery Installation and Upgrade Guide.*

BART supports the following platforms (64 bit only):

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

BART supports the following database versions:

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

1.1 *Installation*

Before installing BART, ensure that your repository configuration allows access to the EDB repository. For information about requesting credentials to the EnterpriseDB repository, visit:

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

1. Use `yum` to create the repository configuration file.

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

2. Modify the repository configuration file (named `edb.repo` located in `/etc/yum.repos.d`) ensuring 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.

Software Requirements

1. Before installing other software prerequisite, use `yum` to install the EPEL 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`).

2. Then, use `yum` to install the following software on the BART host and any remote server on which an incremental backup will be restored:

   - EDB Postgres Advanced Server or PostgreSQL
   - Postgres libpq library
   - Postgres pg_basebackup utility program
   - Boost Libraries version 1.53 for RHEL/CentOS 7.

3. Enable and activate Secure Shell (SSH) and Secure Copy (SCP) client programs on the BART host as well as on the remote database server hosts. The BART host and target database server host must accept a password-less SSH/SCP login connection.

4. After meeting the prerequisites, you can install the BART RPM package directly from the EDB Yum Repository website by specifying the package name only:

   ```
   yum install edb-bart
   ```

5. 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.

The BART product is installed in the `/usr/edb/bart` directory location (referred to as `BART_HOME`).

1.2 *Configuring BART*

To configure the BART host and each database server that is to be managed by BART:

1. Establish the BART user account and ensure it runs the `bart` and the `bart-scanner` program.

2. Use the `bart.cfg.sample` file to create the `bart.cfg` file.

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

3. Set the environment variable for the BART user account. 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 and 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
```

4. Set the following parameters in the [BART] section of the BART configuration file (located in the BART_HOME/etc/bart.cfg directory).

   Note: The parameter setting in the server section overrides the setting in the global [BART] section for that particular database server.

   - **bart_host (required) -** Specify this parameter value in the form of *bart_user@bart_host_address*.
   - **backup_path (required) -** Create the BART backup catalog and specify the path to the file system parent directory where BART database server backups and archived WAL files are stored. Ensure the BART user account owns the location specified in the backup_path parameter**.**
   - **pg_basebackup_path (required) -** Specify the path to the pg_basebackup program on the BART host.
   - **xlog_method (optional) –** Set to fetch (default) to collect the transaction log files after the backup is completed. Set to stream to stream the transaction log in parallel with the full backup creation.
   - **retention_policy (optional) -** Specify when an active backup should be marked as obsolete when the MANAGE subcommand is used. Specify *max_number* BACKUPS (default setting), *max_number DAYS*, *max_number WEEKS*, *or* max_number *MONTHS* where *max_number* is a positive integer.
   - **wal_compression (optional) -** Set to enabled to compress the archived WAL files in gzip format when the MANAGE subcommand is invoked. **Note:** The gzip compression program must be in the BART user account's PATH.
   - **copy_wals_during_restore (optional) -** Set 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.
   - **logfile (optional) -** Use logfile to specify the path to the location to which output from the bart program is written. The log file will be created the first time you invoke the bart command using the sample configuration file value.
   - **scanner_logfile (optional) -** Use scanner_logfile to specify the path to the location to which output from the bart-scanner program is written. The

scanner log file will be created the first time you invoke the `bart` command using the sample configuration file value.

- **thread_count (optional) -** Specify the number of worker threads to copy blocks from the database server to the BART backup catalog when the `BACKUP` subcommand is invoked for incremental backups.
  Note: When taking a full backup, if the thread count is `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. The maximum permitted value is 131072 (131072 * 8192 = 1 GB). The minimum permitted value is 1 (1 * 8192 = 8192 bytes).
- **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. 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.

5. 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`.

6. Set the following parameters for each database server in the server section of the BART configuration file.

   Note: 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 a database server name.
   - **backup_name (optional)** – Specify user-friendly name for the backups of the database server.
   - **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 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.

- **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 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 with a setting of `'scp %p %h:%a/%f'` where `%p` is the path of the file to archive used by the Postgres archiving process, `%h` is replaced by the `bart_host` parameter setting, %a is replaced by the BART archive path, and `%f` is the archived file name used by the Postgres archiving process.
- **cluster_owner (required) -** Specify the Linux operating system user account that owns the database cluster. This is typically `enterprisedb` for Advanced Server clusters installed in compatible mode, or `postgres` for PostgreSQL or Advanced Server clusters installed in the PostgreSQL compatible mode.
- **remote_host (optional) -** Specify this parameter value in the form of *`remote_user@remote_host_address`* where *`remote_user`* is the user account on the target database server host and *`remote_host_address`* is the IP address of the remote host.
- **tablespace_path (optional) -** Specify the path to which tablespaces are to be restored in the *`OID=tablespace_path;OID=tablespace_path …`* format. If the backup is to be restored to a remote host (specified by the `remote_host` parameter), then the tablespace must exist on the remote host.
- **allow_incremental_backups (optional) -** Set to `enabled` to permit incremental backups.
- **description (optional)** –Specify the database server description.

Refer to the BART installation guide for information about configuring the optional parameters: `Retention_policy, xlog_method, wal_compression, copy_wals_during_restore, thread_count, batch_size, scan_interval, mbm_scan_timeout, and workers.`

1.3 *Configuring the Database Server*

To configure the database server, you must:

1. authorize SSH/SCP access
2. set up a replication database user
3. enable WAL archiving
4. verify configuration settings

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

1.3.1 Authorizing SSH/SCP Access

To authorize the SSH/SCP access:

1. Enable the usage of public key authentication for CentOS 7:

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

   ```
   PubkeyAuthentication yes
   ```

   b. Reload the configuration file using the `service sshd reload`, `service sshd stop`, `service sshd start`, or `service sshd restart`.

Note: For any SSH or SCP errors, examine the log file `/var/log/secure`

2. Execute the following command to create a password-less connection.

   ```
   ssh-copy-id target_user@host_address
   ```

For more information about how to generate the authorized public key, see section 4.3.1.2 of the *EDB Postgres Backup and Recovery Installation and Upgrade Guide*.

1.3.2 Setting up a Replication Database User

To set up a replication database user:

1. Choose a database user to serve as the *replication database user (superuser)* for each Postgres database server to be managed by BART.

2. Modify the `pg_hba.conf` file to allow the replication database user to access the `template1` database. Include the replication database user in the `pg_hba.conf` file as a `replication` database connection if `pg_basebackup` is to be used for taking any backups.

3. Specify the replication database user for the database server in the BART configuration file in the `user` parameter.

1.3.3 Enabling WAL Archiving

To enable WAL archiving, set the following parameters in the `postgresql.conf` for the database server for which BART is to perform backup:

- wal_level to replica for Postgres versions.
- archive_mode to on.
- archive_command to copy the WAL files to the BART backup catalog.
- 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 server, the max_wal_senders setting must account for an additional session for transaction log streaming.

Note: Run the INIT subcommand with the -o option to override any existing archive_command setting in the postgresql.conf or the postgresql.auto.conf file.

1. After verifying that the full path of the BART backup catalog has been created, restart the database server to initiate WAL archiving.

2. Start the WAL scanner by executing the following command:

   ```
   ./bart-scanner
   ```

1.3.4 Verifying Configuration Setting

Use the CHECK-CONFIG subcommand with the –s option to verify the parameter settings in the database server configuration for which the -s option is specified.

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

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

- 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.

1.4 *Taking a Backup*

This section provides information about creating a full or incremental backup of a database server. For detailed information about taking a full backup, incremental backup, point-in-time recovery and the restore process, see *EDB Postgres Backup and Recovery Guide.*

The syntax of the `BACKUP` subcommand is:

```
bart BACKUP -s { server_name | all } [ -F { p | t } ]
  [ -z ] [ -c compression_level ]
  [ --parent { backup_id | backup_name } ]
  [ --backup-name backup_name ]
  [ --thread-count number_of_threads ]
  [ { --with-pg_basebackup | --no-pg_basebackup } ]
  [ --check ]
```

Note: While a `BACKUP` subcommand is in progress, no other processes must run in parallel.

In the above syntax along with the `BACKUP` subcommand:

- Specify the `-s` option, and replace the *`server_name`* with the name of the server that will be backed up (it must be configured in the BART configuration file). Specify `all` to take a backup of all servers. The `-s` option is mandatory.

  The backup is saved in the following directory:

  ```
  backup_path/server_name/backup_id.
  ```

  Along with the `bart BACKUP -s server_name` or `bart BACKUP -s all` subcommand, specify the following options only if required. If you do not specify any of the following options, backup is created with the default setting:

  - Specify the `-F p` option to create backup in the plain text format and `-F t` to create backup in the tar format (default). If the transaction log streaming method is used, then the `-F p` option must be specified.
  - Specify the `-z` option to use gzip compression on the tar file output using the default compression level. This option is applicable only for the tar format.

- Specify the `-c` option to apply the gzip compression level on the tar file output, and replace *`compression_level`* with the digit 1 through 9, with 9 being the best compression (applicable only for the tar format).
- Specify the `--parent` option and replace *`backup_id`* with the backup identifier of a parent backup or replace *`backup_name`* with the parent backup name to take an incremental backup. Incremental backup can only be taken in the plain text format (`-F p`). Specify the option `--check` before taking an incremental backup to verify if the required MBM files are present in the BART backup catalog. The `--parent` option must be specified when the `--check` option is used.
- Specify the `--backup-name` option and replace *`backup_name`* with the user-friendly name assigned to the backup.
- Specify the `-thread count` option and replace *`number_of_threads`* with the number of worker threads to run in parallel to copy blocks for incremental backups.
- Specify `--with-pg_basebackup` option to use `pg_basebackup` to take a full backup. The number of thread counts in effect is ignored as given by the `thread_count` parameter in the BART configuration file.
  Note: If the thread count in effect is greater than `1`, then the `pg_basebackup` utility is not used to take the full backup unless the `--with-pg_basebackup` option is specified with the `BACKUP` subcommand.
- Specify the `--no pg_basebackup` option to not use `pg_basebackup` to take a full backup.

The following example creates a full backup for the server name `mktg` in the default tar format with gzip compression.

```
-bash-4.2$ bart BACKUP -s mktg -z
DEBUG: Server: acctg, No. Backups 8
DEBUG: Server: hr, Now: 2016-10-27 10:41:07 EDT, RetentionWindow: 345600 (secs)
==> 96 hour(s)
DEBUG: Exec Command: /opt/PostgresPlus/9.6AS/bin/pg basebackup --version
INFO:  creating backup for server 'mktg'
INFO:  backup identifier: '1477579267918'
DEBUG: internal backup Command to be execute:
'/opt/PostgresPlus/9.6AS/bin/pg_basebackup -D /opt/backup/mktg/1477579267918 -X
fetch -P  -Ft -z -d "host=192.168.2.24 port=5443 user=repuser" '
55006/55006 kB (100%), 3/3 tablespaces

INFO:  backup completed successfully
DEBUG: Exec Command: tar  -C /opt/backup/mktg/1477579267918 -xzf
/opt/backup/mktg/1477579267918/base.tar.gz backup_label
WARNING: log_timezone is not set in the server, using the local timezone
information
DEBUG: calculate checksum for backup '/opt/backup/mktg/1477579267918'
DEBUG: calculating checksum of file
'/opt/backup/mktg/1477579267918/17283.tar.gz'
INFO:  backup checksum: 4f69a5f2ed7092aede490de040e685fb of 17283.tar.gz
DEBUG: calculating checksum of file
'/opt/backup/mktg/1477579267918/17284.tar.gz'
INFO:  backup checksum: 103e1e39003e0eb6acad11d4f791be45 of 17284.tar.gz
DEBUG: calculating checksum of file
'/opt/backup/mktg/1477579267918/base.tar.gz'
INFO:  backup checksum: 6b5efb3e701ac30372db74e3ad8eac21 of base.tar.gz
WARNING: cannot get the tablespace(s) information for backup '1477579267918'
```

```
DEBUG: start time: 1477582868, stop time: 1477582870, duration: 2
DEBUG: Backup Info file created at '/opt/backup/mktg/1477579267918/backupinfo'
INFO:
BACKUP DETAILS:
BACKUP STATUS: active
BACKUP IDENTIFIER: 1477579267918
BACKUP NAME: none
BACKUP PARENT: none
BACKUP LOCATION: /opt/backup/mktg/1477579267918
BACKUP SIZE: 5.45 MB
BACKUP FORMAT: tar.gz
XLOG METHOD: fetch
BACKUP CHECKSUM(s): 3
 ChkSum                           File
 4f69a5f2ed7092aede490de040e685fb   17283.tar.gz
 103e1e39003e0eb6acad11d4f791be45   17284.tar.gz
 6b5efb3e701ac30372db74e3ad8eac21   base.tar.gz

TABLESPACE(s): 4294967295

START WAL LOCATION: 000000010000000200000051
BACKUP METHOD: streamed
BACKUP FROM: master
START TIME: 2016-10-27 10:41:08 EDT
STOP TIME: 2016-10-27 10:41:10 EDT
TOTAL DURATION: 2 sec(s)
```

The following example shows an incremental backup taken by specifying the `--parent` option. The `-F p` option must be specified as well for plain text format.

```
-bash-4.1$ bart BACKUP -s hr -F p --parent hr full 1 --backup-name hr incr 1
INFO:  creating incremental backup for server 'hr'
INFO:  checking mbm files /opt/backup/hr/archived_wals
INFO:  new backup identifier generated 1490819642608
INFO:  reading directory /opt/backup/hr/archived_wals
INFO:  all files processed
NOTICE:  pg stop backup complete, all required WAL segments have been archived
INFO:  incremental backup completed successfully
INFO:
BACKUP DETAILS:
BACKUP STATUS: active
BACKUP IDENTIFIER: 1490819642608
BACKUP NAME: hr incr 1
BACKUP PARENT: 1490819418664
BACKUP LOCATION: /opt/backup/hr/1490819642608
BACKUP SIZE: 16.53 MB
BACKUP FORMAT: plain
BACKUP TIMEZONE: US/Eastern
XLOG METHOD: fetch
BACKUP CHECKSUM(s): 0
TABLESPACE(s): 0
START WAL LOCATION: 000000010000000000000007
STOP WAL LOCATION: 000000010000000000000007
BACKUP METHOD: pg start backup
BACKUP FROM: master
START TIME: 2017-03-29 16:34:04 EDT
STOP TIME: 2017-03-29 16:34:05 EDT
TOTAL DURATION: 1 sec(s)
```