Backup and Recovery QuickStart Guide

Release 2.6

QuickStart Guide

Oct 06, 2020

Contents

This tutorial demonstrates using the `dnf` command to install and configure the EDB Backup and Recovery Tool (BART) 2.6 on a CentOS 8 host with minimal configuration settings. The tutorial assumes that the user has some knowledge of installation and system administration procedures, and has administrative privileges on the host.

For detailed information about BART installation and configuration, see the *BART Installation and Upgrade Guide* available at the EDB website.

- BART is tested with the following database versions:

    - Advanced Server - 9.6, 10, 11, 12, and 13.

    - PostgreSQL - 9.6, 10, 11, 12, and 13.

Installing BART

The following steps describe installing BART on CentOS 8.x OS.

1. Assume superuser privileges and use `dnf` to create the repository configuration file:

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

2. Create an EDB user account to request credentials to the EDB repository; for a user account visit the EnterpriseDB website.

3. Use your choice of editor to open the repository configuration file (named `edb.repo`, located in `/etc/yum.repos.d`) and set the value of the `enabled` parameter to `1`, and replace the `username` and `password` placeholders in the `baseurl` specification with the username and password of a registered EnterpriseDB user.

4. Update the cache:

```
dnf makecache
```

5. Install an Advanced Server or PostgreSQL database server.

   To install Advanced Server, execute the following command:

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

   Use sudo to assume the identity of the `enterprisedb` database superuser:

```
sudo su - enterprisedb
```

   Create an Advanced Server cluster named `acctg` on listener port `5444`:

```
/usr/edb/as12/bin/initdb -D /var/lib/edb/as12/acctg
```

   As the `enterprisedb` user, start the cluster:

```
/usr/edb/as12/bin/pg_ctl start -D /var/lib/edb/as12/acctg
```

   You can check the status of the cluster with the following command:

```
/usr/edb/as12/bin/pg_ctl status -D /var/lib/edb/as12/acctg
```

   Note: The BART host server is not required to have an Advanced Server or PostgreSQL installation, but must include a copy of the Postgres `libpq` library, the `pg_basebackup` utility program, and Boost Libraries 1.66 version for CentOS 8.

---

6. If you do not already have the `pg_basebackup` program installed on the BART host, you can use the following command to install a limited number of files that include the `pg_basebackup` program:

```
dnf install edb-asxx-server-client
```

7. As a root user, use the following command to install the BART RPM package:

```
dnf install edb-bart
```

BART (the `bart` program and `bart-scanner`) is installed in the `/usr/edb/bart/bin` directory, referred to as `<BART_HOME>`. Repeat the installation process described in this section to install BART on all remote hosts where incremental backups are to be restored.

Configuring BART

Before configuring BART, establish the BART user account (the operating system user) that will run the BART command line program. Then, to configure the BART host and each database server that is to be managed by BART, perform the following steps:

1. Assume superuser privileges, create the directory that will hold the BART backup catalog, and assign its ownership (with restrictive privileges) to the BART user account:

   In this example, `bartuser` is the BART user account and `/opt/backup` is the BART backup catalog.

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

2. Navigate to the `/usr/edb/bart/etc` directory and copy the `bart.cfg.sample` file to create the BART configuration file (`bart.cfg`):

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

3. Open the BART configuration file (`bart.cfg`) using an editor of your choice and scroll through the BART configuration file to edit the file as required; sample settings are included for your reference. You must add the mandatory parameters to the `[BART]` and `[ServerName]` sections. Default values may be used for optional parameters. For detailed information about parameter settings, see the *BART Installation and Upgrade Guide*, available at the EDB website.

   Parameters set in the `[BART]` section are applicable to all BART managed database servers, while parameters set in the `[ServerName]` section are applicable only to the specific server; `[ServerName]` settings override `[BART]` section settings.

   In the following example, only mandatory parameters are set:

   ```
   [BART]
   bart_host= bartuser@192.168.169.199
   backup_path = /opt/backup
   pg_basebackup_path = /usr/edb/as12/bin/pg_basebackup

   [EPAS12]
   host = 127.0.0.1
   user = repuser
   cluster_owner = enterprisedb
   ```

The following table describes only mandatory parameters:

Parameters/Placeholder	Section	Description
bart_host	[BART]	Use this field to specify the BART user and the IP address of the host on which the BART utility is installed. Specify the value in the form of `<bart_user>@<bart_host_address>`.

Table 1 – continued from previous page

Parameters/Placeholder	Section	Description
`backup_path`	`[BART]`	Use this field to specify the path where all BART backups and archived WAL files will be stored. Ensure the BART user account holds privileges to create subdirectories and files within the location specified in the `backup_path` parameter. The default `backup_path` is BART backup catalog (`/opt/backup`).
`pg_basebackup_path`	`[BART]`	Use this field to specify the path to the pg_basebackup utility (`/usr/edb/as<xx>/bin/pg_basebackup`).
`[ServerName]`	`[ServerName]`	Specify the name of the database server to be backed up (for example, [EPAS12]).
`host`	`[ServerName]`	Specify the IP address of the database server to be configured for backup.
`user`	`[ServerName]`	Specify the replication database user name used by BART to establish the connection to the database server for full backups.
`cluster_owner`	`[ServerName]`	Specify the Linux operating system user account that owns the database cluster.

4. As a BART user, navigate to `/usr/edb/bart/bin` and invoke the following subcommand (omitting the `-s` option) to verify the [BART] section parameter settings:

```
bart CHECK-CONFIG
```

5. Authorize *SSH/SCP access* between the server and the BART host without a password prompt.

6. Create a *replication database user* for each database server that BART manages.

7. To enable continuous WAL archiving for any database server for which BART is to perform a backup, modify the `postgresql.conf` file, setting:

   - `wal_level` to `replica` (for Postgres 9.6 or later)

   - `archive_mode` to `on`

   - `archive_command` (if it is not set in the `bart.cfg` file)

   - `max_wal_senders` to a value high enough to leave at least one session available for the backup.

   After setting the parameters, restart the database server.

8. To start the WAL scanner, navigate to `/usr/edb/bart/bin` as a BART user and execute the following command:

```
./bart-scanner
```

9. If you are using the default `archive_command`, then navigate to `/usr/edb/bart/bin` as a BART user, run the `INIT` subcommand without the `-o` option, and restart the database server:

```
bart INIT [ -s { <server_name> | all } ]
```

Where `<server_name>` is the name of the database server to be backed up.

If you have customized the `archive_command` setting in the `bart.cfg` file, run the `INIT` subcommand with the `-o` option to override any existing Postgresql `archive_command` setting in the `postgresql.conf` or the `postgresql.auto.conf` file, and restart the database server.

4

```
bart INIT [ -s { <server_name> | all } ] [ -o ]
```

10. To verify the database server parameter settings, as a BART user navigate to `/usr/edb/bart/bin` and invoke the `CHECK-CONFIG` subcommand with the -s option:

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

BART is now configured successfully. For detailed information about using BART, see the *EDB Backup and Recovery Tool User Guide* available at the EDB website.

Creating a Passwordless Connection

The following example enables SSH/SCP access on a CentOS 7.x host; similar (platform-specific) steps will apply to other platforms/versions. You must create a passwordless connection between the BART host (SSH/SCP client) and the database server (target SSH/SCP server), as well as a passwordless connection between the database server (SSH/SCP client) and the BART host (target SSH/SCP server).

1. Log in as the user account on the BART host that will be initiating the SSH or SCP connection and navigate to the user account's home directory and check for an existing `.ssh` subdirectory. If the `.ssh` directory does not exist, create one with the required privileges.

2. As a root user navigate to `/usr/edb/bart`, open the `/etc/ssh/sshd_config` file and set the `PubkeyAuthentication` parameter to `yes`.

3. Reload the configuration file:

```
service sshd reload
```

If you get any SSH or SCP errors, examine the log file (`/var/log/secure`).

4. As a BART user, use the following command to generate the public key file; you can accept the default responses:

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

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

5. Use `SCP` to make a temporary copy of the public key file on the target server:

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

6. As a `target_user`, log into the target server using `ssh target_user@host_address` command and navigate to the user account's home directory to check if there is an existing `.ssh` subdirectory. If it does not exist, create one with the required privileges.

7. Append the temporary client's public key file, `tmp.pub`, to the `authorized_keys` file:

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

If an `authorized_keys` file does not exist, create a new file, but be careful not to completely replace any existing `authorized_keys` file.

8. Ensure only the file owner (and not other groups or users) has access to `authorized_keys` file:

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

9. Delete the temporary public key file:

```
rm tmp.pub
```

5

Now, when logged into the BART host as a user, there should be no prompt for a password when you are connecting to the target database server:

```
ssh target_user@database_server_address
```

Creating a Passwordless Connection Between the Database Server and the BART Host

If backups are to be taken from a given database server host, but restored to a different database server host, the passwordless 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.

An example of how to create a passwordless connection is documented in the *EDB Postgres Backup and Recovery Reference Guide*, available at the EDB website.

Even when the Advanced Server database is on the same host as BART, and the Advanced Server database cluster owner is also the BART user account, a passwordless SSH/SCP connection must be established from the same user account to itself.

1. On the database server, navigate into the target user account's home directory to check for an existing `.ssh` subdirectory. If it does does not exist, create one in the user account's home directory with the required privileges.

2. As a database server user, generate the public key file:

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

3. Create a temporary copy of the public key file:

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

4. As a target user, log into the BART host and navigate to the user account's home directory to check if there is an existing `.ssh` subdirectory. If it does not exist, create one with the required privileges:

```
ssh target_user@host_address
```

5. Append the client's temporary public key file to the `authorized_keys` file:

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

If the `authorized_keys file` does not exist, create a new file, but do not completely replace any existing `authorized_keys` file.

6. Ensure that only the file owner (and not other groups or users) has access to `authorized_keys` file:

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

7. Delete the temporary public key file:

```
rm tmp.pub
```

Now, when logged into the database server as a user, there should be no prompt for a password when you are connecting to the BART host:

```
ssh bart_user@bartip_address
```

Creating a Replication Database User

1. To create a replication database user (a superuser), connect to the database server with the psql client, and invoke the following PostgreSQL command:

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

2. Specify this replication database user in the `user` parameter of the `bart.cfg` file.

3. The pg_hba.conf file must minimally permit the replication database user to have access to the database. The IP address from which the replication database user has access to the database is the BART host location. The replication database user must also be included in the `pg_hba.conf` file as a replication database connection if `pg_basebackup` is to be used for taking any backups.

4. To ensure there is no password prompt when connecting to the database server with the replication database user, a recommended method is to use the `.pgpass` file located in the BART user account's home directory (if it does not exist, you need to create the `.pgpass` file with the required privileges). The `.pgpass` file must contain an entry for each BART managed database server, and its corresponding replication database user and password.