How to Use pg_dump & pg_restore with Postgres Plus(R) in Linux(R)

A Postgres Evaluation Quick Tutorial From EnterpriseDB

November 30, 2009

EnterpriseDB Corporation, 235 Littleton Road, Westford, MA 01866, USA
T +1 978 589 5700 **F** +1 978 589 5701 **E** info@enterprisedb.com **www**.enterprisedb.com

Introduction

Learn how to use pg_dump and pg_restore to safeguard Postgres Plus databases. You will then be able to build a database and an application for a Technical Evaluation, knowing you can easily create intermittent database backups of your work and restore them if needed.

This EnterpriseDB Quick Tutorial helps you get started with the or Postgres Plus Advanced Server database products in a Linux environment. It is assumed that you have already downloaded and installed Postgres Plus Standard Server or Postgres Plus Advanced Server on your desktop or laptop computer.

This Quick Tutorial is designed to help you expedite your Technical Evaluation of Postgres Plus Standard Server or Postgres Plus Advanced Server. For more informational assets on conducting your evaluation of Postgres Plus, visit the self-service web site, Postgres Plus Open Source Adoption.

In this Quick Tutorial you will learn how to do the following:

- Distinguish between backup formats
- Choose among various backup and restore options
- Create a plain text backup and restore it
- Create a custom archive backup and restore it

Feature Description

There are various methods and options available to back up and restore a Postgres Plus database. This Quick Tutorial will show you how to use the Postgres Plus utility programs `pg_dump` and `pg_restore`. These programs are executed on the command line and can therefore be incorporated into scripts if desired.

For complete information on how to create a backup file using pg_dump, see pg_dump in Chapter "PostgreSQL Client Applications" under VI. "Reference" of the PostgreSQL Core Documentation found on the Postgres Plus documentation web page.

For complete information on how to restore a backup file using pg_restore, see pg_restore in Chapter "PostgreSQL Client Applications" under VI. "Reference" of the PostgreSQL Core Documentation found on the Postgres Plus documentation web page.

For a complete discussion of all the different backup and restore strategies available in Postgres Plus, see Chapter 24, "Backup and Restore" of the PostgreSQL Core Documentation found on the Postgres Plus documentation web page.

Backup File Formats

Three different backup file formats can be created by `pg_dump`:

- **Plain-Text Format.** A plain-text script file containing SQL statements and commands that can be executed by the `psql` command line terminal program to recreate the database objects and load the table data. Use the `psql` program to restore from a plain-text backup file.
- **Custom Archive Format.** A binary file that allows for restoration of all or only selected database objects from the backup file. Use the `pg_restore` program to restore from a custom archive backup file.
- **Tar Archive Format.** A tar archive file that allows for restoration of all or only selected database objects from the backup file. Use the `pg_restore` program to restore from a tar archive backup file.

A plain-text backup file can be edited in a text editor if desired before restoring its database objects with the `psql` program. Plain-text format is normally recommended for smaller databases.

A custom archive backup file cannot be edited. However, you can use the `pg_restore` program to select which database objects to restore from the backup file. Custom archive format is recommended for medium to large databases for which you may want to select the database objects to restore from the backup file.

A tar archive backup file can be manipulated by standard Linux tools such as `tar`. Like custom archive format, the `pg_restore` program can be used to select which database objects to restore from the backup file.

File compression can be applied by the `pg_dump` program to plain-text or custom archive backup files to reduce the backup file size. The default action is no compression when producing a plain-text backup file. A moderate level of compression is applied by default when producing a custom archive backup file. The `pg_dump` program cannot apply compression to tar archive backup files.

Backup and Restore Options

Using the various options available with the `pg_dump` and `pg_restore` programs, you can control which database objects are saved in a backup file, which database objects are restored from a backup file, and how they are restored.

The following are examples of some of the options available:

- Dump or restore the schema only (table, view, and sequence definitions, constraints, triggers, and functions), not the table data. (If you are using Postgres Plus Advanced Server, SPL functions, procedures, triggers, and packages can also be backed up and restored.)

- Dump or restore the table data only, not the schema.
- Dump or restore database objects belonging to selected schemas.
- Exclude selected schemas when creating a backup.
- Dump or restore selected tables.
- Exclude selected tables when creating a backup.
- Allow the restore operation to create a new database with the same name as the database from which the backup was created, and restore the database objects into this newly created database.
- Restore database objects into any existing database.
- Retain ownership of restored database objects using the same role names that owned the objects when the backup was created.
- Assign the role name of the user running the restore operation as the owner of all restored database objects.

For complete, detailed instructions on how to create a backup file, see pg_dump in Chapter "PostgreSQL Client Applications" under VI. "Reference" of the PostgreSQL Core Documentation found on the Postgres Plus documentation web page.

For complete, detailed instructions on how to restore a custom archive backup file, see pg_restore in Chapter "PostgreSQL Client Applications" under VI. "Reference" of the PostgreSQL Core Documentation found on the Postgres Plus documentation web page.

The instructions that follow illustrate a common scenario where you want to back up the entire contents of a database, and then at a later point in time, you want to recreate the entire database from the backup file. This scenario will be demonstrated with a plain-text backup file and with a custom archive backup file.

Tutorial Steps

The following assumptions are made about your database environment:

- The database cluster into which you are restoring your database contains the role names (user names and group names) that were the owners of the database objects when the backup was created. That is, you are either restoring into the same database cluster from which you created the backup (and you have not deleted any roles that owned any database objects at the time the backup was created), or you are restoring into a new database cluster in which you have added the same set of role names that existed in the database cluster from which you created the backup.
- The database cluster into which you are restoring your database does not already contain a database with the same name as the database from which the backup file was created. (If you are restoring into the same database cluster from which you created the backup file, you have either deleted or renamed your database after you created the backup.)

Note: A *database cluster* is a set of databases run by the same Postgres Plus instance. A database cluster is uniquely identified by its IP address and port number.

Creating a Database Backup in a Plain-Text Backup File

The following steps describe how to use the `pg_dump` program to create a plain-text backup file of a database.

Step 1: Log onto the computer on which the Postgres Plus database server is running. Any valid account on the computer can be used.

Step 2: Use the `cd` command to make the Postgres Plus `bin` directory your current working directory.

```
cd /opt/PostgresPlus/8.4SS/bin
```

Note: If you are using Postgres Plus Advanced Server, use the `cd` command to make `dbserver/bin` your current working directory.

Step 3: Run the `pg_dump` program with a role name that has the superuser privilege (`-U` option), the `-C` option to include a `CREATE DATABASE` statement in the backup file, the name to be given to the backup file (`-f` option), and the name of the database for which a backup file is to be created (last parameter of the command line).

```
./pg_dump -U postgres -C -f /home/user/sample_backup sample
```

You have just created a backup of the `sample` database to a backup file named `sample_backup`. A portion of file `sample_backup` is shown as follows:

```
--
-- PostgreSQL database dump
--

SET statement_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

--
-- Name: sample; Type: DATABASE; Schema: -; Owner: postgres
--

CREATE DATABASE sample WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE
= 'en_US.UTF-8' LC_CTYPE = 'en_US.UTF-8';


ALTER DATABASE sample OWNER TO postgres;

\connect sample

SET statement_timeout = 0;
```



```
SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

--
-- Name: plpgsql; Type: PROCEDURAL LANGUAGE; Schema: -; Owner: postgres
--

CREATE PROCEDURAL LANGUAGE plpgsql;


ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;

SET search_path = public, pg_catalog;
            .
            .
            .
```

Restoring a Database From a Plain-Text Backup File

The following steps describe how to restore a database from a plain-text backup file using the `psql` program.

Note: If you are using Postgres Plus Advanced Server, the `edb-psql` program can be used as well.

The plain-text backup file, `sample_backup`, created from the `sample` database in the preceding example will be used to restore the `sample` database.

Step 1: Log onto the computer on which the Postgres Plus database server is running. Any valid account on the computer can be used.

Step 2: If you are restoring into a different database cluster than the one from which the backup file was created, or if you have deleted roles from your database cluster, be sure that all role names that owned database objects when the backup file was created exist in the database cluster into which you want to restore the backup file.

Note: If you do not know what roles owned database objects when the backup file was created, you can scan the backup file using a text editor for `ALTER` *`object`* `OWNER TO` *`role`* statements, some examples of which are shown by the following:

```
ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;
ALTER TYPE public.emp_query_type OWNER TO postgres;
ALTER FUNCTION public.emp_comp(p_sal numeric, p_comm numeric) OWNER TO postgres;
```

You can list the roles that currently exist in a database cluster by connecting to the database cluster with the `psql` program and running the `\dg` command as shown by the following:

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./psql -d postgres -U postgres
```



```
Password for user postgres:
psql (8.4.1)
Type "help" for help.

postgres=# \dg
            List of roles
 Role name | Attributes  | Member of
-----------+-------------+-----------
 postgres  | Superuser   | {}
           : Create role
           : Create DB
```

If you need to create roles, use the CREATE ROLE statement.

Note: If the original owner's role name of a database object does not exist in the database cluster into which you are restoring, an error message will be displayed when the `ALTER` statement cannot assign the ownership. The database object will end up being assigned to the role given by the `-U` option of the `psql` program when you perform the restore operation.

Step 3: Be sure there is no existing database in the database cluster to which you are restoring that has the same name as the database from which the backup file was created.

Note: If you do not know the name of the database from which the backup file was created, you can scan the backup file using a text editor for the `CREATE DATABASE` statement, an example of which is shown by the following:

```
CREATE DATABASE sample WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE
= 'en_US.UTF-8' LC_CTYPE = 'en_US.UTF-8';
```

You can list the databases that currently exist in a database cluster by connecting to the database cluster with the `psql` program and running the `\l` command.

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./psql -d postgres -U postgres
Password for user postgres:
psql (8.4.1)
Type "help" for help.

postgres=# \l
                                 List of databases
   Name    |  Owner   | Encoding |  Collation  |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
                                                             : postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
                                                             : postgres=CTc/postgres
(3 rows)
```

If there is a database in the database cluster with the same name as the database from which the backup file was made, you can rename the existing database using the `RENAME TO` option of the ALTER DATABASE statement, or delete it using the DROP DATABASE statement.

Note: During the restore operation, if the database cluster contains a database with the same name as the database from which the backup file was created, an error message will be displayed when the `CREATE DATABASE` statement cannot create a new database with the duplicate name. The `psql` program will then proceed to recreate the database objects from the backup file within the existing database. The result will most likely be a database that contains unwanted database objects and incorrect database settings.

Step 4: Use the `cd` command to make the Postgres Plus `bin` directory your current working directory.

```
cd /opt/PostgresPlus/8.4SS/bin
```

Note: If you are using Postgres Plus Advanced Server, use the `cd` command to make `dbserver/bin` your current working directory.

Step 5: Run the `psql` program with the name of a database to which a connection is to be established (`-d` option), a role name that has the superuser privilege (`-U` option), and the directory path to the backup file (`-f` option).

Note: The database you specify with the `-d` option is not affected by the restore operation. The `psql` program requires a session to be established over a database connection before it can process the SQL statements and `psql` commands from the backup file.

```
./psql -d postgres -U postgres -f /home/user/sample_backup
```

You have just recreated the `sample` database from the backup file named `sample_backup`. The following shows a portion of the messages displayed by the `psql` program as it processes the SQL statements and `psql` commands in the backup file.

```
$ ./psql -d postgres -U postgres -f /home/user/sample_backup
Password for user postgres:
SET
SET
SET
SET
SET
SET
CREATE DATABASE
ALTER DATABASE
psql (8.4.1)
You are now connected to database "sample".
            .
            .
            .
```

Creating a Database Backup in a Custom Archive Backup File

The following steps describe how to use the `pg_dump` program to create a custom archive backup file of a database.

Step 1: Log onto the computer on which the Postgres Plus database server is running. Any valid account on the computer can be used.

Step 2: Use the `cd` command to make the Postgres Plus `bin` directory your current working directory.

```
cd /opt/PostgresPlus/8.4SS/bin
```

Note: If you are using Postgres Plus Advanced Server, use the `cd` command to make `dbserver/bin` your current working directory.

Step 3: Run the `pg_dump` program with a role name that has the superuser privilege (`-U` option), the `-Fc` option to specify custom archive format, the name to be given to the backup file (`-f` option), and the name of the database for which a backup file is to be created (last parameter of the command line).

```
./pg_dump -U postgres -Fc -f /home/user/sample_backup sample
```

You have just created a backup of the `sample` database to a backup file named `sample_backup`. Though a custom archive backup file cannot be viewed directly, a table of contents of the backup file can be generated using the `-l` option of the `pg_restore` program as follows:

```
$ ./pg_restore -l /home/user/sample_backup
;
; Archive created at Thu Nov 12 15:12:12 2009
;     dbname: sample
;     TOC Entries: 44
;     Compression: -1
;     Dump Version: 1.11-0
;     Format: CUSTOM
;     Integer: 4 bytes
;     Offset: 8 bytes
;     Dumped from database version: 8.4.1
;     Dumped by pg_dump version: 8.4.1
;
;
; Selected TOC Entries:
;
6; 2615 2200 SCHEMA - public postgres
1818; 0 0 COMMENT - SCHEMA public postgres
1819; 0 0 ACL - public postgres
323; 2612 17798 PROCEDURAL LANGUAGE - plpgsql postgres
312; 1247 17801 TYPE public emp_query_type postgres
19; 1255 17802 FUNCTION public emp_comp(numeric, numeric) postgres
20; 1255 17803 FUNCTION public emp_query(numeric, numeric, character varying) postgres
21; 1255 17804 FUNCTION public emp_query_caller() postgres
22; 1255 17805 FUNCTION public emp_sal_trig() postgres
            .
            .
            .
```

Restoring a Database From a Custom Archive Backup File

The following steps describe how to restore a database from a custom archive backup file using the `pg_restore` program.

The custom archive backup file, `sample_backup`, created from the `sample` database in the preceding example will be used to restore the `sample` database.

Step 1: Log onto the computer on which the Postgres Plus database server is running. Any valid account on the computer can be used.

Step 2: If you are restoring into a different database cluster than the one from which the backup file was created, or if you have deleted roles from your database cluster, be sure that all role names that owned database objects when the backup file was created exist in the database cluster into which you want to restore the backup file.

Note: If you do not know what roles owned database objects when the backup file was created, you can generate a SQL text version of the backup from the custom archive backup file. To accomplish this, run the `pg_restore` program giving the backup file as the only parameter. You can then scan the text for `ALTER` *`object`* `OWNER TO` *`role`* statements.

This method is shown in the following example:

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./pg_restore /home/user/sample_backup | grep 'OWNER TO'
ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;
ALTER TYPE public.emp_query_type OWNER TO postgres;
ALTER FUNCTION public.emp_comp(p_sal numeric, p_comm numeric) OWNER TO
postgres;
           .
           .
           .
```

You can list the roles that currently exist in a database cluster by connecting to the database cluster with the `psql` program and running the `\dg` command as shown by the following:

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./psql -d postgres -U postgres
Password for user postgres:
psql (8.4.1)
Type "help" for help.

postgres=# \dg
            List of roles
 Role name | Attributes  | Member of
-----------+-------------+-----------
 postgres  | Superuser   | {}
           : Create role
           : Create DB
```

If you need to create roles, use the CREATE ROLE statement.

Note: If the original owner's role name of a database object does not exist in the database cluster into which you are restoring, an error message will be displayed when the `ALTER` statement cannot assign the ownership. The ownership of the database object will end up

being assigned to the role given by the `-U` option of the `pg_restore` program when you perform the restore operation.

Step 3: Be sure there is no existing database in the database cluster to which you are restoring that has the same name as the database from which the backup file was created.

Note: If you do not know the name of the database from which the backup file was created, you can list a table of contents of the custom archive backup file using the `pg_restore` program with the `-l` option and the backup file as the only parameters. The database name is given in the dbname field at the top of the table of contents.

This method is shown in the following example:

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./pg_restore -l /home/user/sample_backup
;
; Archive created at Wed Nov 18 12:34:10 2009
;     dbname: sample
           .
           .
           .
```

You can list the databases that currently exist in a database cluster by connecting to the database cluster with the `psql` program and running the `\l` command.

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./psql -d postgres -U postgres
Password for user postgres:
psql (8.4.1)
Type "help" for help.

postgres=# \l
                                 List of databases
   Name    |  Owner   | Encoding |  Collation  |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
                                                             : postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
                                                             : postgres=CTc/postgres
(3 rows)
```

If there is a database in the database cluster with the same name as the database from which the backup file was made, you can rename the existing database using the `RENAME TO` option of the ALTER DATABASE statement, or delete it using the DROP DATABASE statement.

Note: During the restore operation, if the database cluster contains a database with the same name as the database from which the backup file was created, an error message will be displayed when the `CREATE DATABASE` statement cannot create a new database with the duplicate name. The `pg_restore` program will then proceed to recreate the database objects from the backup file within the existing database. The result will most likely be a database that contains unwanted database objects and incorrect database settings.

Step 4: Use the `cd` command to make the Postgres Plus `bin` directory your current working directory.

```
cd /opt/PostgresPlus/8.4SS/bin
```

Note: If you are using Postgres Plus Advanced Server, use the `cd` command to make `dbserver/bin` your current working directory.

Step 5: Run the `pg_restore` program with the name of a database to which a connection is to be established (`-d` option), a role name that has the superuser privilege (`-U` option), the `-C` option to specify that the restore operation is to create a new database using the same name as the database from which the backup file was created, and the directory path to the backup file (last parameter of the command line).

Note: The database you specify with the `-d` option is not affected by the restore operation if used along with the `-C` option. The `pg_restore` program requires a session to be established over a database connection before it can create the new database and restore the database objects from the backup file.

```
./pg_restore -d postgres -U postgres -C /home/user/sample_backup
```

You have just recreated the `sample` database from the backup file named `sample_backup`.

Conclusion

In this Quick Tutorial you learned how to perform the basic operations of backing up and restoring a Postgres Plus database on a Linux system.

You should now be able to proceed confidently with a Technical Evaluation of Postgres Plus. Using the backup and restore features will allow you to make backups of the different stages of your work and restore them as needed.

The following resources should help you move on with this step:

- Postgres Plus Technical Evaluation Guide
- Postgres Plus Getting Started resources
- Postgres Plus Getting Started resources
- Postgres Plus Quick Tutorials
- Postgres Plus User Forums
- Postgres Plus Documentation
- Postgres Plus Webinars

