Backup Databases Using PostgreSQL Backup Tools: pg_dump & pg_dumpall

http://www.postgresqltutorial.com/postgresql-backup-database/

Summary: in this tutorial, we will show you how to backup databases using **PostgreSQL backup** tools including pg_dump and pg_dumpall.

Backing up databases is one of the most critical tasks in database administration. Before backing up the databases, you should consider the following points:

- Full / partial databases
- Both data and structures, or only structures
- Point In Time recovery
- Restore performance

PostgreSQL provides pg_dump and pg_dumpalltools to help you backup databases easily and effectively.

For ones who want to see the command to backup databases quickly, here it is:

>pg_dump -U username -W -F t database_name > c:\backup_file.tar

In this following section, we will show you practical ways to backup one database, all databases, and only database objects.

How to backup one database

To backup one database, you can use the pg_dump tool. The pg_dump dumps out the content of all database objects into a single file.

First, navigate to PostgreSQL BIN folder:

```
C:\>cd C:\Program Files\PostgreSQL\9.2\bin
```

Second, execute the pg_dump program and use the following options to backup the dvdrentaldatabase to the dvdrental.tar file in the c:\pgbackup\ folder.

>pg_dump -U postgres -W -F t dvdrental > c:\pgbackup\dvdrental.tar

Let's examine the options in more detail.

-U postgres: specifies the user to connect to PostgreSQL database server. We used postgres in this example.

-W: forces pg_dump to prompt for the password before connecting to the PostgreSQL database server. After you hit enter, pg_dump will prompt for the password of postgres user.

- F : specifies the output file format that can be one of the following:

- c: custom-format archive file format
- d: directory-format archive
- t:tar
- p: plain text SQL script file).

Because we want the output file to be a tar-format archive file, we use **-F t** in this example.

dvdrental: is the name of the database that we want to back

> c:\pgbackup\dvdrental.tar is the output backup file path.

How to backup all databases

To backup all databases, you can run the individual pg_dump command above sequentially, or parallel if you want to speed up the backup process.

- First, from the psql, use the command \list to list all available databases in your cluster
- Second, back up each individual database using the pg_dump program as described in the above section.

Besides the pg_dump program, PostgreSQL also provides you with the pg_dumpall tool that allows you to backup all databases at once. However, it is not recommended to use this tool because of the following reasons:

- The pg_dumpall program exports all databases, one after another, into a single script file, which prevents you from performing the parallel restore. If you backup all databases this way, the restore process will take more time.
- The processing of dumping all databases takes longer than each individual one so you do not know which dump of each database relates to a specific point in time.

•

If you have a good reason to use the pg_dumpallto backup all databases, the following is the command:

>pg_dumpall -U postgres > c:\pgbackup\all.sql

The options of the pg_dumpall program are similar to the options of the pg_dump program. We omit the -W option to avoid typing the password for each individual database, which is quite tedious.

How to backup database object definitions

Sometimes, you want to backup only database object definitions so that you can restore the schema only. This is helpful in the test phase, which you do not want keep the old test data populated during the testing period.

To backup all objects in all databases, including roles, tablespaces, databases, schemas, tables, indexes, triggers, functions, constraints, views, ownerships and privileges, you use the following command:

>pg_dumpall --schema-only > c:\pgdump\definitiononly.sql

If you want to backup role definition only, use the following command:

>pg_dumpall --roles-only > c:\pgdump\allroles.sql

If you want to backup *tablespaces* definition, use the following command:

```
>pg_dumpall --tablespaces-only > c:\pgdump\allroles.sql
```

How to backup using pgAdmin

The pgAdmin provides an intuitive user interface that allows you to backup a database using pg_dump tool.

For example to backup the dvdrental database to a dvdrental.tar in the c:\pgbackup\dvdrental.tar file, you can follow the following steps:

First, right mouse click on the dvdrental database, and choose the Backup... menu item.



Second, enter the output file name and choose the file format.

G	X
Filename	C: \pgbackup \dvdrental.tar
Format	Tar
Compress Ratio	
Encoding	▼
Rolename	postgres 💌
_	
File Options D	ump Options #1 Dump Options #2 Objects Messages
Help	Backup Cancel

pgAdmin backup tool provides various dump options as follows:

	S
Sections Pre-data Data Post-data	
Type Of Objects Only data Only schema Blobs	
Don't save Owner Privilege Tablespace Unlogged table data	
File Options Dump Options #1 Dump Options #2 Objects Messages Help Backup Cancel	

	23
Queries Include CREATE DATABASE statement Include DROP DATABASE statement Use Column Inserts Use Insert commands	
Disable Trigger \$ quoting Miscellanous	
 Use SET SESSION AUTHORIZATION With OIDs Verbose messages Force double quotes on identifiers 	
File Options Dump Options #1 Dump Options #2 Objects Messages Help Backup Ca	ncel

In the objects tab, you can select which objects to backup:

	X	J
Databa	ase dvdrental	1
📄 🖃 🔽 pul	blic	
	account	
	account_role	
	actor	
	address	
	category	
	city	
	country	
	customer	
	film	
	film_actor	
	film_category	
	inventory	
	language	
	page	
	payment	1
	rental	1
	role	1
···· ·	sales2007q1	
	sales2007q2 🔹	
File Options	Dump Options #1 Dump Options #2 Objects Messages	
Help	Backup Cancel]

Third, click OK button to start performing a backup. The messages tag provides you with detailed messages of the backup process.

C:/Program Files/PostgreSQL/9.2/bin\pg_dump.exehost localhostport 5432t pg_dump: reading schemas pg_dump: reading user-defined tables pg_dump: reading user-defined functions pg_dump: reading user-defined functions pg_dump: reading user-defined functions pg_dump: reading user-defined aggregate functions pg_dump: reading user-defined operators pg_dump: reading user-defined operator classes pg_dump: reading user-defined operator families pg_dump: reading user-defined text search parsers pg_dump: reading user-defined text search templates pg_dump: reading user-defined text search configurations pg_dump: reading user-defined foreign-data wrappers pg_dump: reading user-defined foreign-data wrappers pg_dump: reading user-defined conversions pg_dump: reading type casts pg_dump: reading type casts pg_dump: reading table inheritance information pg_dump: finding extension members
File Options Dump Options #1 Dump Options #2 Objects Messages
Help Done Cancel

In this tutorial, we have shown you some practical way to backup PostgreSQL databases by using pg_dump and pg_dumpall tools.