

# Greenplum 使用 pg\_dump 最备份数据库

Greenplum 使用 pg_dump 最备份数据库.....	1
1 说明.....	1
2 备份数据库数据.....	3
2.1 创建需要备份的数据库.....	3
2.2 执行备份数据库的命令.....	3
2.3 查看备份出来的数据格式.....	3
3 恢复数据库备份数据.....	4
3.1 创建需要恢复的数据库.....	4
3.2 导入备份数据.....	4
4 导出具体表备份.....	5
4.1 执行需要备份的表.....	5
4.2 查看备份的数据.....	6
5 导入备份的具体表.....	7
5.1 执行需要导入备份的表.....	7
5.2 查看导入的结果.....	7
6 schema 级别备份数据.....	8
6.1 备份 schema 数据.....	8
6.2 查看备份后的数据.....	8
7 导入 schema 备份的数据.....	9
7.1 导入备份的数据.....	9
7.2 查看导入后的数据.....	9

## 1 说明

### 1.1 备份说明

Greenplum 支持逻辑备份。我们使用 Greenplum 自带的 `pg_dump` 命令实现逻辑备份功能，导出备份文件，再通过 `psql` 导入到 Greenplum 中，达到备份的效果。

### 1.2 参数说明

用法:

```
pg_dump [选项]... [数据库名字]
```

一般选项:

```
-f, --file=文件名      输出文件名
```

-F, --format=c|t|p      输出文件格式 (定制,tar, 明文)  
 -v, --verbose          详细模式  
 -Z, --compress=0-9      被压缩格式的压缩级别  
 --lock-wait-timeout=TIMEOUT 在等待表锁超时后操作失败  
 --help                  显示此帮助信息,然后退出  
 --version                输出版本信息,然后退出

控制输出内容选项:

-a, --data-only          只转储数据,不包括模式  
 -b, --blobs              在转储中包括大对象  
 -c, --clean              在重新创建之前,先清除(删除)数据库对象  
 -C, --create              在转储中包括命令,以便创建数据库  
 -E, --encoding=ENCODING 转储以 ENCODING 形式编码的数据  
 -n, --schema=SCHEMA      只转储指定名称的模式  
 -N, --exclude-schema=SCHEMA 不转储已命名的模式  
 -o, --oids                在转储中包括 OID  
 -O, --no-owner            在明文格式中,忽略恢复对象所有者  
  
 -s, --schema-only        只转储模式,不包括数据  
 -S, --superuser=NAME      在转储中,指定的超级用户名  
 -t, --table=TABLE        只转储指定名称的表  
 -T, --exclude-table=TABLE 只转储指定名称的表  
 -x, --no-privileges      不要转储权限 (grant/revoke)  
 --binary-upgrade          只能由升级工具使用  
 --inserts                以 INSERT 命令,而不是 COPY 命令的形式转储数据  
 --column-inserts          以带有列名的 INSERT 命令形式转储数据  
 --disable-dollar-quoting 取消美元 (符号)引号,使用 SQL 标准引号  
 --disable-triggers        在只恢复数据的过程中禁用触发器  
 --no-tablespaces         不转储表空间分配信息  
 --role=ROLENAME          在转储前运行 SETROLE  
 --use-set-session-authorization  
                             使用 SESSION AUTHORIZATION 命令代替  
                             ALTER OWNER 命令来设置所有权

联接选项:

-h, --host=主机名          数据库服务器的主机名或套接字目录  
 -p, --port=端口号          数据库服务器的端口号  
 -U, --username=名字        以指定的数据库用户联接  
 -w, --no-password          永远不提示输入口令  
  
 -W, --password            强制口令提示 (自动)

## 2 备份数据库数据

### 2.1 创建需要备份的数据库

创建 dump1 数据库，并在数据库中创建 test1 的 schema,创建几张表并插入数据

### 2.2 执行备份数据库的命令

```
pg_dump -U username -h hostname -p port databasename -f filename
```

参数说明如下：

username: 本地数据库用户名

hostname: 本地数据库主机名，如果是在本地数据库主机登录，可以使用 localhost

port: 本地数据库端口号

databasename: 要备份的本地数据库名

filename: 要生成的备份文件名称

实例：

```
$ pg_dump -h 192.168.100.55 -U gpmon -p 2345 dump1 -f dump1.sql
```

Password:

### 2.3 查看备份出来的数据格式

```
$ head -n 40 dump1.sql
```

```
--
```

```
-- Greenplum Database database dump
```

```
--
```

```
SET statement_timeout = 0;
```

```
SET client_encoding = 'UTF8';
```

```
SET standard_conforming_strings = on;
```

```
SET check_function_bodies = false;
```

```
SET client_min_messages = warning;
```

```
SET default_with_oids = false;
```

```
--
```

```
-- Name: test1; Type: SCHEMA; Schema: -; Owner: gpmon
```

```
--
```

```
CREATE SCHEMA test1;

ALTER SCHEMA test1 OWNER TO gpmon;

SET search_path = public, pg_catalog;

SET default_tablespace = '';

--
-- Name: test1; Type: TABLE; Schema: public; Owner: gpmon; Tablespace:
--

CREATE TABLE test1 (
    id integer,
    name character varying(255)
) DISTRIBUTED RANDOMLY;

ALTER TABLE public.test1 OWNER TO gpmon;

--
-- Data for Name: test1; Type: TABLE DATA; Schema: public; Owner: gpmon

*****
```

## 3 恢复数据库备份数据

### 3.1 创建需要恢复的数据库

```
CREATE DATABASE dump2;
```

### 3.2 导入备份数据

```
psql -U username -h hostname -d destinationdb -p port -f dumpfilename.sql
```

参数说明如下：

username: RDS 上的 PostgreSQL 数据库用户名

hostname: RDS 上的 PostgreSQL 数据库地址

port: RDS 上的 PostgreSQL 数据库端口号

dbname: RDS 上的 PostgreSQL 数据库名

filename: 本地备份数据文件名

实例:

```
$ psql -U gpmon -h 192.168.100.55 -d dump2 -p 2345 -f dump1.sql
```

Password for user gpmon:

SET

SET

SET

SET

SET

SET

CREATE SCHEMA

ALTER SCHEMA

SET

SET

CREATE TABLE

ALTER TABLE

CREATE TABLE

ALTER TABLE

SET

CREATE TABLE

ALTER TABLE

REVOKE

REVOKE

GRANT

GRANT

## 4 导出具体表备份

### 4.1 执行需要备份的表

```
$ pg_dump -h 192.**.11 -t xiaoxu.test_dump -U gpadmin staging -f test_yml_dump.sql
```

192.\*\*.11:IP 地址

xiaoxu.test\_dump : xiaoxu 是 schema 的名字, test\_dump 是表的名字

gpadmin : 使用户的名字

staging : 是数据库的名字

test\_yml\_dump.sql : 备份文件的名称

## 4.2 查看备份的数据

```
$ head -n 40 test_yml_dump.sql
--
-- Greenplum Database database dump
--

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

SET search_path = xiaoxu, pg_catalog;

SET default_tablespace = '';

SET default_with_oids = false;

--
-- Name: test_dump; Type: TABLE; Schema: xiaoxu; Owner: gpadmin; Tablespace:
--

CREATE TABLE test_dump (
    filed1 text,
    filed2 character varying
) DISTRIBUTED RANDOMLY;

ALTER TABLE xiaoxu.test_dump OWNER TO gpadmin;

--
-- Data for Name: test_dump; Type: TABLE DATA; Schema: xiaoxu; Owner: gpadmin
--

COPY test_dump (filed1, filed2) FROM stdin;
A    64
A    71
A    47
A    32
A    68
A    54
A    51
```

\*\*\*\*\*

## 5 导入备份的具体表

### 5.1 执行需要导入备份的表

```
$ psql -h 192.***.11 -Ugpadmin staging -f test_yml_dump.sql
SET
SET
SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
```

192.\*\*\*.11: IP 的地址

gpadmin: 备份的用户

staging: 数据库信息

test\_yml\_dump.sql: 备份出来的 SQL 文件

### 5.2 查看导入的结果

```
staging=# select count(*) from xiaoxu.test_dump;
count
-----
      100
(1 row)
```

```
staging=# select * from xiaoxu.test_dump limit 10;
 filed1 | filed2
-----+-----
 A      | 52
 A      | 44
 A      | 38
 A      | 76
 A      | 57
```

```
A      | 85
A      | 42
A      | 6
A      | 95
A      | 34
```

(10 rows)

## 6 schema 级别备份数据

### 6.1 备份 schema 数据

```
$ pg_dump -h 192.***.11 -n test_schema -U gpadmin -f test_schema.sql
```

192.\*\*\*.11 : IP 地址

test\_schema : schema 名字

gpadmin : 用户名字

stagging : 数据库名字

test\_schema.sql:备份的文件名字

### 6.2 查看备份后的数据

```
$ vim test_schema.sql
```

```
--
```

```
-- Greenplum Database database dump
```

```
--
```

```
SET statement_timeout = 0;
```

```
SET client_encoding = 'UTF8';
```

```
SET standard_conforming_strings = on;
```

```
SET check_function_bodies = false;
```

```
SET client_min_messages = warning;
```

```
SET default_with_oids = false;
```

```
--
```

```
-- Name: test_schema; Type: SCHEMA; Schema: -; Owner: gpadmin
```

```
--
```

```
CREATE SCHEMA test_schema;
```



```
ALTER SCHEMA test_schema OWNER TO gpadmin;
```

```
SET search_path = test_schema, pg_catalog;
```

```
SET default_tablespace = '';
```

```
*****
```

## 7 导入 schema 备份的数据

### 7.1 导入备份的数据

```
$ psql -h 192.168.209.11 -Ugpadmin staging -n test_schema -f test_schema.sql
```

```
psql: warning: extra command-line argument "test_schema" ignored
```

```
SET
```

```
SET
```

```
SET
```

```
SET
```

```
SET
```

```
SET
```

```
CREATE SCHEMA
```

```
ALTER SCHEMA
```

```
SET
```

```
SET
```

```
CREATE TABLE
```

```
ALTER TABLE
```

```
CREATE TABLE
```

```
ALTER TABLE
```

### 7.2 查看导入后的数据

查看 schema 信息

```
staging=# \dn
```

```
      List of schemas
```

Name	Owner
test_schema	gpadmin

```
(1 rows)
```

查看表信息

staging=# \dt test\_schema.

List of relations

Schema	Name	Type	Owner	Storage
test_schema	test1	table	gpadmin	heap
test_schema	test2	table	gpadmin	heap

(2 rows)

select \* from information\_schema.tables where table\_schema='test\_schema';

查询创建工具 显示/隐藏列

1 select \* from information\_schema.tables where table\_schema='test\_schema';

信息 结果1

table_catalog	table_schema	table_name	table_type	self_referencing_column_r	reference_generation	user_define
▶ staging	test_schema	test1	BASE TABLE	(Null)	(Null)	(Null)
staging	test_schema	test2	BASE TABLE	(Null)	(Null)	(Null)