

pg_staging

Dimitri Fontaine

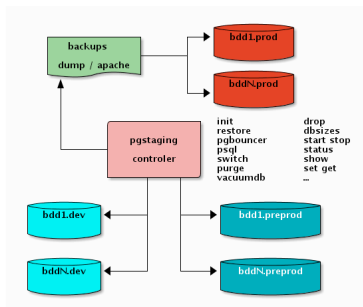
May, 20 2010

Table of contents

- 1 Bird Eye View
- 2 Current dependencies
- 3 Quick Setup
- 4 Services

pg_staging in one image

This image seems not to help people at all, from my blog's readers, so I figured I had to show it up somehow.



What do you need for `pg_staging` to run

So for `pg_staging` to work you need to install:

- `apache` for publishing your backups, `database_YYYY-MM-DD`
- `pgbouncer` to switch database without messing with the applications setup
- some non interactive `ssh` facility
- a `sudo` account, typically user `pgstaging`
- the `pg_staging client` on each machine where you restore

What do you need for pg_staging to run

So for `pg_staging` to work you need to install:

- `apache` for publishing your backups, `database_YYYY-MM-DD`
- `pgbouncer` to switch database without messing with the applications setup
- some non interactive `ssh` facility
- a `sudo` account, typically user `pgstaging`
- the *`pg_staging client`* on each machine where you restore

What do you need for `pg_staging` to run

So for `pg_staging` to work you need to install:

- `apache` for publishing your backups, `database_YYYY-MM-DD`
- `pgbouncer` to switch database without messing with the applications setup
- some non interactive `ssh` facility
- a `sudo` account, typically user `pgstaging`
- the *`pg_staging client`* on each machine where you restore

What do you need for pg_staging to run

So for `pg_staging` to work you need to install:

- `apache` for publishing your backups, `database_YYYY-MM-DD`
- `pgbouncer` to switch database without messing with the applications setup
- some non interactive `ssh` facility
- a `sudo` account, typically user `pgstaging`
- the *pg_staging client* on each machine where you restore

What do you need for `pg_staging` to run

So for `pg_staging` to work you need to install:

- `apache` for publishing your backups, `database_YYYY-MM-DD`
- `pgbouncer` to switch database without messing with the applications setup
- some non interactive `ssh` facility
- a `sudo` account, typically user `pgstaging`
- the *`pg_staging client`* on each machine where you restore

Setting up can get engaging

So for `pg_staging` to work you need to install:

- Setup apache to serve the backup files
- Setup pgbouncer to expose the postgres database
- Setup ssh and sudo and install the *client*
- Use the console!

Setting up can get engaging

So for `pg_staging` to work you need to install:

- Setup apache to serve the backup files
- Setup pgbouncer to expose the postgres database
- Setup ssh and sudo and install the *client*
- Use the console!

Setting up can get engaging

So for `pg_staging` to work you need to install:

- Setup `apache` to serve the backup files
- Setup `pgbouncer` to expose the postgres database
- Setup `ssh` and `sudo` and install the *client*
- Use the console!

Setting up can get engaging

So for `pg_staging` to work you need to install:

- Setup `apache` to serve the backup files
- Setup `pgbouncer` to expose the postgres database
- Setup `ssh` and `sudo` and install the *client*
- Use the console!

Now what

- `init`, `dump`, `redump`, `restore`, `drop`, `switch`
- `purge`, `vacuumdb`, `load`, `createdb`, `fetch`, `presql`, `postsql`
- `databases` `backups` `dbsize` `dbsizes` `psql` `show` `search_path`
- `pgbouncer` `pause` `resume`
- `londiste` `stop` `start` `status` `restart`
- `get` `set`
- `nodata` `catalog` `triggers`

What's the watch saying?

If we have some time left and I still can breeze, let's talk some more about:

- catalog and triggers command
- TODO: fetching backups method, bypassing apache for local files
- pg_staging 0.12 should be simplifying the setup needs

Conclusion

NEXT!