Title: Dockerized PostgreSQL: psql: FATAL: the database system is starting up

Post Body:

I am trying to build and run two Docker containers hosting PostgreSQL and Citus extension using `ansible-container`. I am aware that Citus provides containers, but I want to build my own.

My `container.yaml` looks as follows:

```
version: '2'  services:    database_master:     image: hackermd/ubuntu-trusty-python     user: postgres     expose:       - 50
```

During the build process I can start and stop the cluster via `pg_ctlcluster` and it finishes successfully. However, when I subsequently run the containers, I get the following error:

```
$ docker logs ansible_database_master_1 Removed stale pid file. Warning: connection to the database failed, disabling startup
```

When I build containers with `command: []` and run `ps aux` inside the container, I see the following process:

```
postgres    14  1.6  0.1 307504  3480 ?        Ds   16:46   0:00 postgres: 9.6/master: startup process
```

I've also tried without the `dumb-init` entrypoint. What am I missing?

Accepted Answer:

The problem is related to the default shutdown method of the `pg_ctl stop` mode (`pg_ctl` gets called by `pg_ctlcluster`). Stopping the cluster via `pg_ctlcluster` with the `pg_ctl` option `-m smart` during the build process solves this problem:

```
pg_ctlcluster 9.6 master stop -- -m smart
```

The 'smart' method waits for active clients to disconnect and online backups to finish before shutting down in contrast to the default 'fast' method. This is explained in the documentation of pg_ctl.

In addition, the container would exit once the `pg_ctlcontrol` process successfully started the database cluster via `postgres` (`pg_ctlcontrol` -> `pg_ctl` -> `postgres`). To prevent this, `postgres` can be called directly. The `container.yml` file would then look as follows:

```
version: '2'  services:    database_master:     image: hackermd/ubuntu-trusty-python     user: postgres     expose:       - 50
```

Highest Rated Answer:

My problem was with starting postgres using pg_ctl and right after running tests in my docker container. What fixed it was adding 'smart mode' to my command, i.e:

```
su - postgres -c 'pg_ctl start -D /var/lib/postgresql/data -l /var/lib/postgresql/log.log -m smart'
```