PITR made easy

Joshua Drake

What is PITR

Log shipping/Warm Standby/Replication

Why is it hard?

Like all good things Open Source, pain to configure, stable as can be.

What makes it easy?

PITR tools
BSD Licensed
Written in Python

How does it work?

cmd_archiver.py
Archives files using rsync and ssh
Assumes ssh keys are configured

Configure the archiver

```
Options:
  -h, --help            show this help message and exit
  -A, --archive         Whether or not to archive
  -F FILE, --file=FILE  Archive file
  -C FILE, --config=FILE
                        the name of the archiver config file
  -f, --flush           Flush all remaining archives to slave
  -P, --push    Push archives to a remote host
```

cmd_standby.py

Works in conjunction with pg_standby
Supports failover
Supports recover to point in time

Configure the standby

```
Usage: cmd_standby [options] arg1 arg2

Options:
  -h, --help            show this help message and exit
  -A start|stop, --action=start|stop
                        Start or Stop PostgreSQL
  -B, --basebackup      Start/Stop a base backup
  -C FILE, --config=FILE
                        the name of the archiver config file
  -F VALUE, --failover=VALUE
                        If you are serious, set -F999
  -I, --dbinit          Use before -B
  -P, --ping            Is my master alive?
  -R TIMESTAMP, --recovertotime=TIMESTAMP
                        If you need to restore to a specific point
in time
  -S, --standby         Enter standby mode
```

Features?

Warm standby
Cold Storage
Fail over (actionable)
Arbitrary alerts (monitoring)

Done.