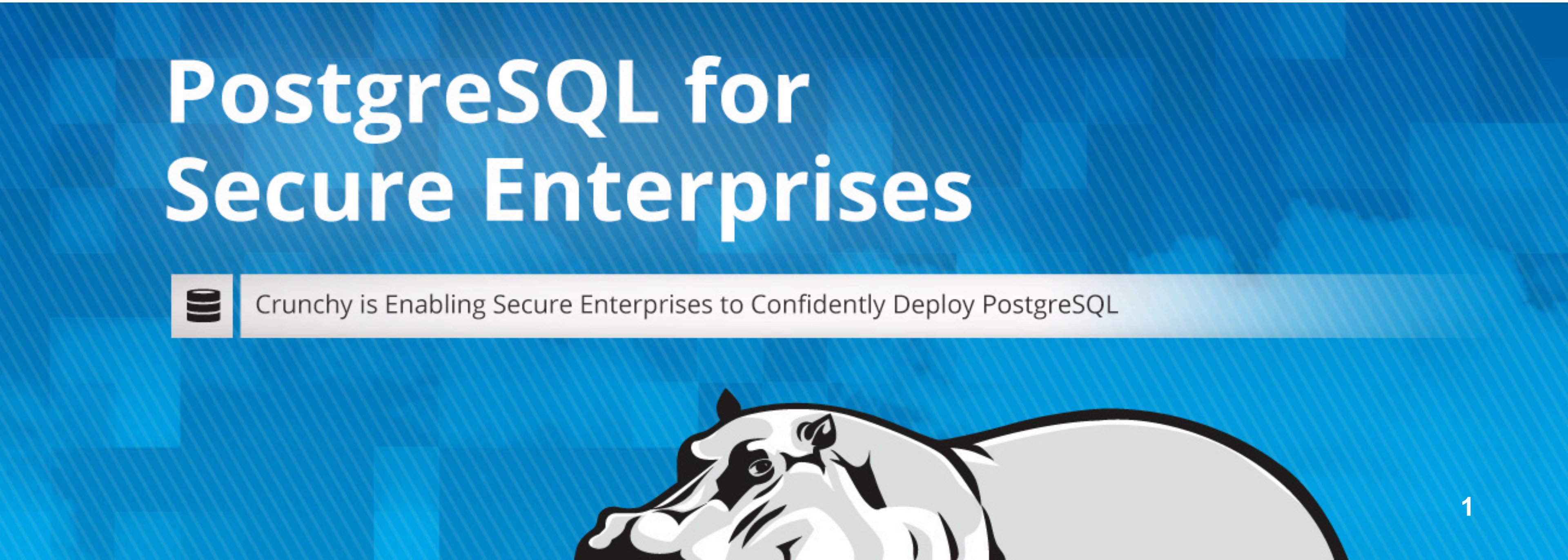
CRUNCHY
DATA SOLUTIONS, INC.

Flexible PostgreSQL Server Configuration

Greg Smith


PostgreSQL for Secure Enterprises
Crunchy is Enabling Secure Enterprises to Confidently Deploy PostgreSQL

PostgreSQL Configuration

- postgresql.conf is the largest file
  - You can include other files into it
- pg_hba.conf controls authentication
  - pg_ident.conf may also be relevant
- New to 9.4: postgresql.auto.conf
  - Supports config changes from SQL!
  - Builds on existing include

Generated Locale Settings

```
datestyle = 'iso, mdy'
#intervalstyle = 'postgres'
timezone = 'US/Eastern'
...
# These settings are initialized by initdb, but they can be changed.
lc_messages = 'en_US.UTF-8'        # locale for system error message
                                   # strings
lc_monetary = 'en_US.UTF-8'        # locale for monetary formatting
lc_numeric = 'en_US.UTF-8'         # locale for number formatting
lc_time = 'en_US.UTF-8'            # locale for time formatting

# default configuration for text search
default_text_search_config = 'pg_catalog.english'
```

Memory oriented settings

```
#listen_addresses = 'localhost'
                # what IP address(es) to listen on;
                # comma-separated list of addresses;
                # defaults to 'localhost
                # use '*' for all
                # (change requires restart)
#port = 5432            # (change requires restart)
max_connections = 100   # (change requires restart)
# - Memory -
shared_buffers = 128MB  # min 128kB
                        # (change requires restart)
```

In-Place Editing

```
# - Memory -
#shared_buffers = 128MB  # min 128kB
           # (change requires restart)
#shared_buffers = 16384MB
           # 2013-08-10 Increased to
           # 16GB based on pgtune
shared_buffers = 8192MB
           # Decreased based on app
           # benchmarking
```

Connection Settings

```
# - Connection Settings -

#listen_addresses = 'localhost’
      # what IP address(es) to listen on;
      # comma-separated list of addresses;
      # defaults to 'localhost';
      # use '*' for all
      # (change requires restart)
#port = 5432         # (change requires restart)
max_connections =100 # (change requires restart)
```

Config file size

```
$ wc -l postgresql.conf
     613 postgresql.conf
```

- Settings are easily lost in there
- New version migrations are a mess
- Small differences between roles
  - Development, Production, Standby, Reports…

PostgreSQL up to 9.3

```
# These options allow settings
# to be loaded from files other
# than the default postgresql.conf.
#include_if_exists = 'exists.conf’
#include = 'special.conf’
```

New postgresql.conf ending

```
#---
# CUSTOMIZED OPTIONS
#---
include_if_exists = 'memory.conf’
include_if_exists = ’connection.conf’
```

memory.conf

```
shared_buffers = 8192MB
wal_buffers = 32MB
maintenance_work_mem = 2048MB
work_mem = 64MB
checkpoint_segments = 256
```

network.conf

```
listen_addresses = '192.169.1.53'
port = 5432
# (change requires restart)
max_connections = 300
# (change requires restart)
```

PostgreSQL 9.4 include_dir

```
# These options allow settings to be loaded from
files other than the default postgresql.conf.

#include_dir = 'conf.d'
      # include files ending in '.conf' from
      # directory 'conf.d'
#include_if_exists = 'exists.conf'
      # include file only if it exists
#include = 'special.conf'         # include file
```

Include Directory

```
#port = 5432            # (change requires restart)
#max_connections = 100  # (change requires restart)
# - Memory -
#shared_buffers = 128MB # min 128kB
                        # (change requires restart)
...
include_dir = 'conf.d’
```

- Put server, app, and time of day settings into include files
- Comment out ***everything*** except locale settings in the postgresql.conf

Major types of config blocks

- Locale
- Memory
- Replication
  - Master and standby are different
- Logging
- Vacuum
- Time of day variations

conf.d/memory-64GB.conf

```
shared_buffers = 8GB
wal_buffers = 32MB
maintenance_work_mem = 2GB
work_mem = 64MB
effective_cache_size = 32GB
checkpoint_segments = 256
```

conf.d/network-internal.conf

```
listen_addresses = '192.169.1.53'
port = 5432
# (change requires restart)
max_connections = 300
# (change requires restart)
```

conf.d/master.conf

```
wal_level = hot_standby
max_wal_senders = 5
archive_mode = on
archive_command = ' /var/lib/
postgresql/scripts/archive_wal -s -
c -H db2 -d /var/lib/postgresql/
9.3/archive %p %f’
statement_timeout = 86400000 # in
milliseconds = 24 hours
```

conf.d/template/log-normal.conf

```
log_destination = 'syslog'
log_line_prefix = '%t [%p]: [%l-1] user=%u,db=%d '
log_min_duration_statement = 1000
             # -1 is disabled, 0 logs all statements,
             # >0 logs longer running ones
log_min_messages = notice
log_checkpoints = on
log_connections = off
log_temp_files = 1024  # temporary files equal or larger
log_lock_waits = on    # log lock waits when >=
deadlock_timeout
deadlock_timeout = 1000ms
log_autovacuum_min_duration = 1000 # -1 disables, 0 logs
all actions, >0 logs longer running ones
```

conf.d/template/log-detail.conf

```
log_destination = 'syslog’
log_line_prefix = '%t [%p]: [%l-1] user=%u,db=%d '
log_min_duration_statement = 100
             # -1 is disabled, 0 logs all statements,
             # >0 logs longer running ones
log_min_messages = info
log_checkpoints = on
log_connections = off
log_temp_files = 1024  # temporary files equal or larger
log_lock_waits = on    # log lock waits when >=
deadlock_timeout
deadlock_timeout = 100ms
log_autovacuum_min_duration = 1000 # -1 disables, 0 logs
all actions, >0 logs longer running ones
```

Switch between log detail level

Normal operation:

```
$ cp $PGDATA/template/conf/log-normal.conf $PGDATA/conf/logging.conf
$ pg_ctl reload
```

Server has performance issues:

```
$ cp $PGDATA/template/conf/log-normal.conf $PGDATA/conf/logging.conf
$ pg_ctl reload
```

Autovacuum Scheduling

- Break the day into peak and off-peak time periods
- Allow small amount of vacuum work to be during peak times
  - Bad idea to have none, catalog tables need cleanup at all times
- Use a scheduler like cron to copy the appropriate template file and signal the server:

```
cp $PGDATA/template/conf/vac-normal.conf $PGDATA/conf/vacuum.conf
pg_ctl reload
```

- Resist the urge to do this with symbolic links instead
  - Eventually someone will edit one of the templates

conf.d/template/vac-normal.conf

```
#
# Off-peak setting that allows more work.
# 200 is the database default
#
vacuum_cost_limit = 200  # 1-10000 credits
```

conf.d/template/vac-peak.conf

```
# Peak time period setting
vacuum_cost_limit = 5    # 1-10000 credits
```

conf.d/querylog.conf

```
shared_preload_libraries = 'pg_stat_statements'
pg_stat_statements.max = 10000
pg_stat_statements.track = all
```

postgresql.conf with config blocks

- Most modified settings (shared_buffers, max_connections, etc.) are commented out. Most systems can leave the locale settings there.
- 9.3 and earlier:

```
include = 'conf/memory-64GB.conf’
include = 'conf/network-internal.conf’
include = 'conf/master.conf’
include = 'conf/logging.conf’
include = 'conf/vacuum.conf’
include = 'conf/querylog.conf’
```

- 9.4:

```
include_dir = 'conf.d’
```

ALTER SYSTEM SET

```
$ psql
psql (9.4devel)
Type "help" for help.

postgres=# ALTER SYSTEM SET
work_mem='128MB';
ALTER SYSTEM
```

Default auto include file

```
postgresql.auto.conf:

# Do not edit this file manually!
# It will be overwritten by the
ALTER SYSTEM command.
```

Successful include change

```
$ cat postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by ALTER SYSTEM command.
work_mem = '128MB'
```

Bonuses!

- List of settings you have modified, including what file those changes came from:

```
SELECT * FROM pg_settings;
```

- See PostgreSQL Wiki for more ideas.

Control Source

- Treat the database configuration files as source code
  - Put them into version control near the application itself
- Consider deployment automation tools
  - puppet, chef, etc.
- Study how to use diff to see what is changing
  - Much easier to do this with block configurations

Prefer syslog

- Using syslog gives you log rotation, compression, and related tools.
- Built-in PostgreSQL logging makes those your problem.
- syslog does durable writes by default, syncing them to disk
- When your kernel crashes, you *really* want to see the last log messages before it went down.
- Disable line sync on any file that’s storing PostgreSQL logs.

Sample syslog durability change

Debian Squeeze: create a new

`/etc/syslog.d/postgresql.conf`

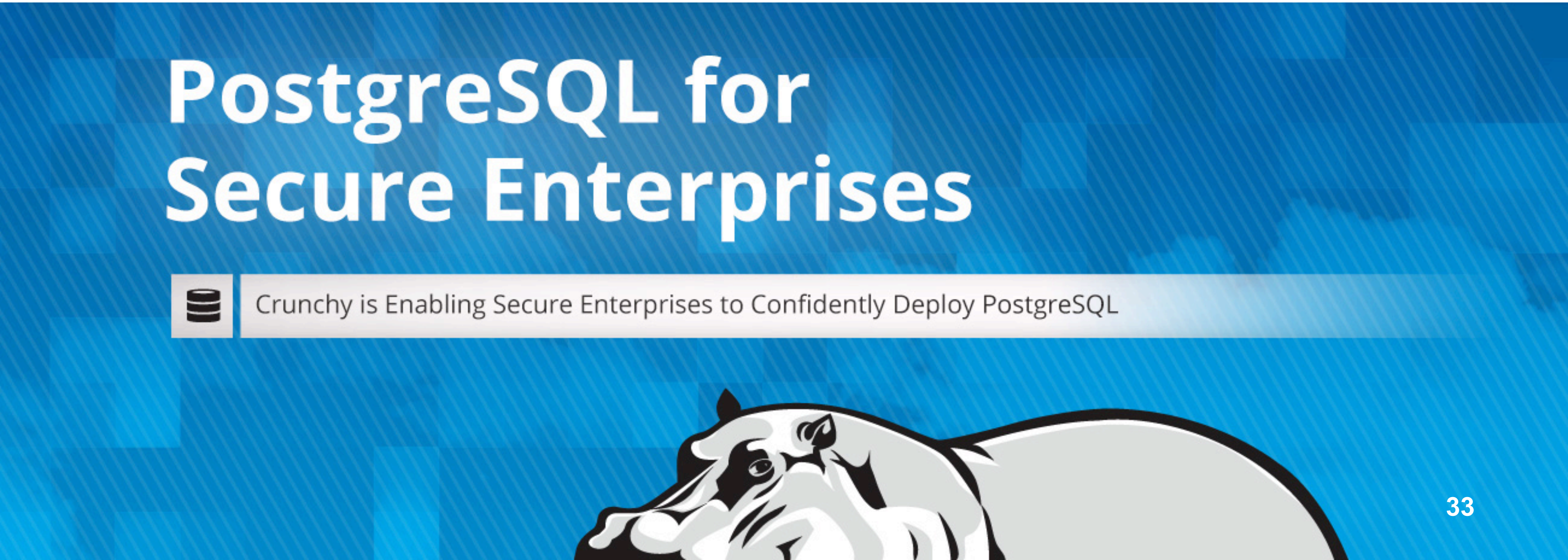
file with the following contents:

```
LOCAL0.*          -/var/log/postgresql
```

```
/etc/init.d/rsyslog restart
```

CRUNCHY
DATA SOLUTIONS, INC.

http://www.crunchydatasolutions.com/



PostgreSQL for Secure Enterprises

Crunchy is Enabling Secure Enterprises to Confidently Deploy PostgreSQL