

data egret



Meine
"default" postgresql.conf Datei,
Schritt für Schritt



- 269 Parametern in der Version 10
- Die Parametern im *postgresql.conf* sind um manuell zu verändern
- *postgresql.auto.conf* kann nur durch **ALTER SYSTEM** geändert werden
- **pg_stat_settings** view fasst alles zusammen

```

postgres=# \x
Expanded display is on.
postgres=# select * from pg_settings where name ~ 'checkpoint_timeout';
-[ RECORD 1 ]-----+-----
name           | checkpoint_timeout
setting        | 3600
unit           | s
category       | Write-Ahead Log / Checkpoints
short_desc     | Sets the maximum time between automatic WAL checkpoints.
extra_desc     |
context        | sighup
vartype        | integer
source         | configuration file
min_val        | 30
max_val        | 86400
enumvals       |
boot_val       | 300
reset_val      | 3600
sourcefile     | /etc/postgresql/10/main/postgresql.conf
sourceline     | 208
pending_restart | f

```

```
postgres=# select distinct(context) from pg_settings ;
           context
-----
postmaster
superuser-backend
user
internal
backend
sighup
superuser
(7 rows)
```

- Bei der manuellen Konfiguration die Reihenfolge lieber nicht ändern

- Bei der manuellen Konfiguration die Reihenfolge lieber nicht ändern
- *postgresql.conf* unterstützt include-Dateien

- Bei der manuellen Konfiguration die Reihenfolge lieber nicht ändern
- *postgresql.conf* unterstützt include-Dateien
- **pg_stat_settings** ist die letzte Instanz

- Bei der manuellen Konfiguration die Reihenfolge lieber nicht ändern
- *postgresql.conf* unterstützt include-Dateien
- **pg_stat_settings** ist die letzte Instanz
- Und jetzt los...

- * oder 127.0.0.1

- * oder 127.0.0.1
- 127.0.0.1 ist OK, wenn man pgbouncer nutzt

- * oder 127.0.0.1
- 127.0.0.1 ist OK, wenn man pgbouncer nutzt
- Die Datenbank **muss** vor dem Internet durch ein Firewall geschützt werden

- Connection ruft ein schweres Unix-Prozess auf

- Connection ruft ein schweres Unix-Prozess auf
- Deshalb werden Werte wie *max_connections = 1000* niemals funktionieren

- Connection ruft ein schweres Unix-Prozess auf
- Deshalb werden Werte wie *max_connections = 1000* niemals funktionieren
- Eine viel bessere Idee: *max_connections = 100 oder 200* und kleinere Pool-Größen in pgbouncer oder einem anderen Applikationsserver

- Wenn max_connections erschöpft sind, braucht man eine superuser-Verbindung, um etwas zu reparieren

- Wenn max_connections erschöpft sind, braucht man eine superuser-Verbindung, um etwas zu reparieren
- Mindestens 5, besser 10

- Wenn Netzwerk nicht stabil ist, können 5 Sekunden von keepalive alles retten

- Ein Faustregel: 25% von RAM

- Ein Faustregel: 25% von RAM
- Man braucht auch schnelle Festplatten, um 16/32/64Gb von **shared_buffers** nutzen zu können

- Ein Faustregel: 25% von RAM
- Man braucht auch schnelle Festplatten, um 16/32/64Gb von **shared_buffers** nutzen zu können
- Wenn Datenbank definitiv kleiner als RAM ist, kann man auch 75% von RAM für *shared_buffers* einsetzen

- Ein Faustregel: wenn man mehr als 8-16Gb *shared_buffers* hat, braucht man auch Huge Pages

- Ein Faustregel: wenn man mehr als 8-16Gb *shared_buffers* hat, braucht man auch Huge Pages
- Huge Pages müssen zuerst im Kernel eingeschaltet worden sein

- RAM per Prozess, Postgres Worker nutzt dieses RAM für Sortierung, HashJoin etc.

- RAM per Prozess, Postgres Worker nutzt dieses RAM für Sortierung, HashJoin etc.
- 128Mb ist ein guter Wert zum Start

- RAM per Prozess, Postgres Worker nutzt dieses RAM für Sortierung, HashJoin etc.
- 128Mb ist ein guter Wert zum Start
- Zu hohe Werte können locker ein OOM verursachen

- RAM per Prozess, Postgres Worker nutzt dieses RAM für Sortierung, HashJoin etc.
- 128Mb ist ein guter Wert zum Start
- Zu hohe Werte können locker ein OOM verursachen
- Kann individuell, für jede Sitzung konfiguriert werden

- Wie *work_mem* aber für Superuser Verbindungen

- Wie *work_mem* aber für Superuser Verbindungen
- 256-512Mb, wenn genug RAM zur Verfügung steht

- Wie *work_mem* aber für Superuser Verbindungen
- 256-512Mb, wenn genug RAM zur Verfügung steht
- Sehr nützlich z. B. für **CREATE INDEX CONCURRENTLY**

- Wie *work_mem* aber für Superuser Verbindungen
- 256-512Mb, wenn genug RAM zur Verfügung steht
- Sehr nützlich z. B. für **CREATE INDEX CONCURRENTLY**
- *autovacuum_work_mem* ist ein Teil des *maintenance_work_mem*, darf etwas kleiner sein

- *wal_level = replica*

- *wal_level = replica*
- *checkpoint_timeout = 60min*, wenn solche Recovery Target erlaubt ist, kann für Performance besser sein
- *max_wal_size = 16GB*

- *wal_level = replica*
- *checkpoint_timeout = 60min*, wenn solche Recovery Target erlaubt ist, kann für Performance besser sein
- *max_wal_size = 16GB*
- *checkpoint_completion_target = 0.9*

- Background Writer hilft Checkpointer Dirty Pages aufzuräumen

- Background Writer hilft Checkpointer Dirty Pages aufzuräumen
- Leider ein sehr alter und nicht bester Teil von Postgres

- Background Writer hilft Checkpointer Dirty Pages aufzuräumen
- Leider ein sehr alter und nicht bester Teil von Postgres
- Alle Werte zu Maximum:
 - ▶ *bgwriter_delay = 10ms*
 - ▶ *bgwriter_lru_maxpages = 1000*
 - ▶ *bgwriter_lru_multiplier = 10.0*

- *autovacuum_vacuum_threshold* = 50
- *autovacuum_vacuum_scale_factor* = 0.05

- *autovacuum_vacuum_threshold = 50*
- *autovacuum_vacuum_scale_factor = 0.05*
- *autovacuum_naptime = 1s*

- *autovacuum_vacuum_threshold = 50*
- *autovacuum_vacuum_scale_factor = 0.05*
- *autovacuum_naptime = 1s*
- *autovacuum_max_workers = 10*

- *autovacuum_vacuum_threshold = 50*
- *autovacuum_vacuum_scale_factor = 0.05*
- *autovacuum_naptime = 1s*
- *autovacuum_max_workers = 10*
- *autovacuum_analyze_threshold = 50*
- *autovacuum_analyze_scale_factor = 0.05*

- *autovacuum_vacuum_threshold = 50*
- *autovacuum_vacuum_scale_factor = 0.05*
- *autovacuum_naptime = 1s*
- *autovacuum_max_workers = 10*
- *autovacuum_analyze_threshold = 50*
- *autovacuum_analyze_scale_factor = 0.05*

- *autovacuum_vacuum_threshold = 50*
- *autovacuum_vacuum_scale_factor = 0.05*
- *autovacuum_naptime = 1s*
- *autovacuum_max_workers = 10*
- *autovacuum_analyze_threshold = 50*
- *autovacuum_analyze_scale_factor = 0.05*

- *autovacuum_vacuum_threshold = 50*
- *autovacuum_vacuum_scale_factor = 0.05*
- *autovacuum_naptime = 1s*
- *autovacuum_max_workers = 10*
- *autovacuum_analyze_threshold = 50*
- *autovacuum_analyze_scale_factor = 0.05*
- *autovacuum_freeze_min_age = 5000000*
- *autovacuum_freeze_table_age = 15000000*

- *shared_preload_libraries = 'pg_stat_statements'*
- *pg_stat_statements.max = 10000*
- *pg_stat_statements.track = top*

Fragen?

ik@dataegret.com

