Riding Elephants Safely

Working with PostgreSQL
When Your DBA is Not Around

SCaLE 21x
2024.03.14
Richard Yen

EDB

About Me

- Software Developer/Support Engineer @ EDB since 2015
- Using Postgres to back Golang and Python apps
- Perl web developer before getting thrust into a DBA role
- Been working with PostgreSQL since 2002

Is this you?

- Software developer, QA Engineer, Data scientist, etc.
- Seeking independence from someone else who manages your development or UAT environments
- DBA is on vacation
  - ... or quit
  - ... or you never had a DBA 😱

There's a Lot to Cover!

- Installation
- Performance tuning
- Query tuning
- Memory management
- Indexes
- Views
- Tablespaces
- Backups
- Replication
- Pooling
- Foreign Data Wrappers
- Statistics collection
- Autovacuum tuning
- Monitoring
- Triggers
- Logical Decoding
- Encodings
- Timestamps
- Query planning
- PL/SQL
- Access control
- WAL
- Disaster recovery
- PITR
- Logging
- Constraints
- Data types
- ... and more!

What We Hope to Achieve Today

- Log into the database, start and stop it
- Take a backup before catastrophic damage occurs
- Diagnose performance/stability issues by reading the logs
- Identify any schema changes that could improve performance
- Understand PostgreSQL's file/directory structure
- 📣 Keep the database up and running!

Starting and Stopping the Database

- SSH access allows start/stop the database as needed
- Managed databases will need to be stopped/started from the console (i.e. RDS)
- Perform sanity checks before starting up
  - Disk space - know if the database is capable of starting up
  - Database logs - know why the database shut down earlier
- `sudo systemctl start postgresql`
  - May need to use `postgresql-${version}` in some cases
- `pg_ctl`
  - 📣 Need to know `${PGDATA}` to perform start/stop
  - `pg_ctl -d ${PGDATA} start`
  - `pg_ctl -d ${PGDATA} stop`
  - `pg_ctl -d ${PGDATA} -m f stop`
  - `pg_ctl -d ${PGDATA} -m i stop`
    - This technically crashes the database, starts in recovery mode

Connecting to the Database

- What you need:
  - Hostname
  - Port (5432 by default)
  - Username
  - Password
  - Check application config (your app) and database config (`postgresql.conf`) if defaults don't work
- `psql` -- Postgres' default command line interface
- GUI database applications
  - PgAdmin
  - DBeaver

Connecting to the Database

```
psql -h "database.example.com" -U edb_admin edb_admin
psql (15.3 (Homebrew), server 15.3 (Debian 15.3-1.pgdg100+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

edb_admin=> \l
                                                 List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    | ICU Locale | Locale Provider | Access privileges
-----------+----------+----------+-------------+-------------+------------+-----------------+------------------
 edb_admin | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 template0 | rdsadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =c/postgres     +
           |          |          |             |             |            |                 | postgres=CTc/
postgres
(4 rows)
```

Connecting to the Database

```
psql -h "database.example.com" -U edb_admin edb_admin
psql (15.3 (Homebrew), server 15.3 (Debian 15.3-1.pgdg100+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

edb_admin=> \d
                 List of relations
 Schema |          Name           | Type |  Owner
--------+-------------------------+------+----------
 public | pg_stat_statements      | view | postgres
 public | pg_stat_statements_info | view | postgres
(2 rows)

edb_admin=> \dn
          List of schemas
     Name      |       Owner
---------------+-------------------
 my_schema     | edb_admin
 public        | pg_database_owner
 results       | edb_admin
(3 rows)
```

What's Going On in the Database?

- `SELECT * FROM pg_stat_activity;`
  - Shows what is going on at this very instant
  - Limited visibility in managed databases (i.e. RDS, Google Cloud, Azure)
- Postgres logs are also very informative (`SHOW log_directory;`)
- RDS logs may be helpful if available
- Superusers can cancel a query or terminate a session
  - `SELECT pg_cancel_backend(${pid})`
  - `SELECT pg_terminate_backend(${pid})`

What's Going On in the Database?

```
 datid | datname  |  pid   | leader_pid | usesysid | usename  | application_name | client_addr | client_hostname | client_port |
backend_start          |          xact_start           |          query_start          |         state_change          | wait_event_type |
wait_event      | state  | backend_xid | backend_xmin | query_id |               query               |         backend_type
-------+----------+--------+------------+----------+----------+------------------+-------------+-----------------+-------------
+-------------------------------+-------------------------------+-------------------------------+-------------------------------
+-----------------+---------------------+--------+-------------+--------------+----------+----------------------------------
+------------------------------
       |          |   7212 |            |       10 | postgres |                  |             |                 |             |
2023-08-16 21:56:45.037403-07 |                               |                               |                               | Activity
| LogicalLauncherMain |        |             |              |          |                                   | logical replication launcher
       |          |   7211 |            |          |          |                  |             |                 |             |
2023-08-16 21:56:45.033525-07 |                               |                               |                               | Activity
| AutoVacuumMain      |        |             |              |          |                                   | autovacuum launcher
     5 | postgres | 212820 |            |       10 | postgres | psql             |             |                 |          -1 |
2023-08-31 00:23:18.95734-07  | 2023-08-31 00:23:23.702319-07 | 2023-08-31 00:23:23.702319-07 | 2023-08-31 00:23:23.702323-07 |
|                     | active |             |         6796 |          | select * from pg_stat_activity ; | client backend
       |          |   7208 |            |          |          |                  |             |                 |             |
2023-08-16 21:56:44.813366-07 |                               |                               |                               | Activity
| BgWriterMain        |        |             |              |          |                                   | background writer
       |          |   7207 |            |          |          |                  |             |                 |             |
2023-08-16 21:56:44.807436-07 |                               |                               |                               | Activity
| CheckpointerMain    |        |             |              |          |                                   | checkpointer
       |          |   7210 |            |          |          |                  |             |                 |             |
2023-08-16 21:56:45.041297-07 |                               |                               |                               | Activity
| WalWriterMain       |        |             |              |          |                                   | walwriter
(6 rows)
```

What's Going On in the Database?

```
postgres=# \x
Expanded display is on.
postgres=# select * from pg_stat_activity limit 2 offset 1;
-[ RECORD 1 ]----+------------------------------------------
datid            |
datname          |
pid              | 7211
leader_pid       |
usesysid         |
usename          |
application_name |
client_addr      |
client_hostname  |
client_port      |
backend_start    | 2023-08-16 21:56:45.033525-07
xact_start       |
query_start      |
state_change     |
wait_event_type  | Activity
wait_event       | AutoVacuumMain
state            |
backend_xid      |
backend_xmin     |
query_id         |
query            |
backend_type     | autovacuum launcher
```

```
-[ RECORD 2 ]----+------------------------------------------
datid            | 5
datname          | postgres
pid              | 212820
leader_pid       |
usesysid         | 10
usename          | postgres
application_name | psql
client_addr      |
client_hostname  |
client_port      | -1
backend_start    | 2023-08-31 00:23:18.95734-07
xact_start       | 2023-08-31 00:24:40.08706-07
query_start      | 2023-08-31 00:24:40.08706-07
state_change     | 2023-08-31 00:24:40.087064-07
wait_event_type  |
wait_event       |
state            | active
backend_xid      |
backend_xmin     | 6796
query_id         |
query            | select * from pg_stat_activity limit 2 offset 1;
backend_type     | client backend
```

What's Going On in the Database?

```
edb_admin=> select * from pg_stat_activity where state =
'active' or state = 'idle in transaction';
-[ RECORD 1 ]----+----------------------------------------
datid            | 16445
datname          | edb_admin
pid              | 31432
leader_pid       |
usesysid         | 16397
usename          | postgres
application_name | psql
client_addr      | 23.125.120.241
client_hostname  |
client_port      | 62749
backend_start    | 2024-03-14 05:41:07.537071+00
xact_start       | 2024-03-14 05:41:25.528078+00
query_start      | 2024-03-14 05:41:25.528078+00
state_change     | 2024-03-14 05:41:25.52808+00
wait_event_type  | Lock
wait_event       | relation
state            | active
backend_xid      | 621470
backend_xmin     | 621470
query_id         | 1407004101491583163
query            | vacuum full verbose analyze upload ;
backend_type     | client backend
```

```
-[ RECORD 2 ]----+-----------------------------
datid            | 16445
datname          | edb_admin
pid              | 30958
leader_pid       |
usesysid         | 16397
usename          | postgres
application_name | psql
client_addr      | 23.125.120.241
client_hostname  |
client_port      | 62643
backend_start    | 2024-03-14 05:32:15.356503+00
xact_start       | 2024-03-14 05:40:42.320476+00
query_start      | 2024-03-14 05:40:55.032098+00
state_change     | 2024-03-14 05:40:55.032642+00
wait_event_type  | Client
wait_event       | ClientRead
state            | idle in transaction
backend_xid      |
backend_xmin     |
query_id         | -3849865870178790307
query            | select  * from upload limit 1;
backend_type     | client backend
```

What's Going On in the Database?

```
edb_admin=> select * from pg_locks where not granted;
-[ RECORD 1 ]------+------------------------------
locktype           | relation
database           | 16445
relation           | 16533
page               |
tuple              |
virtualxid         |
transactionid      |
classid            |
objid              |
objsubid           |
virtualtransaction | 6/237739
pid                | 31432
mode               | AccessExclusiveLock
granted            | f
fastpath           | f
waitstart          | 2024-03-14 05:41:25.528223+00
```

```
edb_admin=> begin;
BEGIN
edb_admin=*> lock table upload in exclusive
mode;
LOCK TABLE
edb_admin=*> select  * from upload limit 1;
-[ RECORD 1 ]----+---------------------------
runid            | 32323232-32-322
actor            | TEST
ref              | testref
ref_type         | branch
commit_hash      |
cloudsmith_tags  |
github_event     |
created_at       | 2023-01-16 10:20:18.437358
maturity         | snapshot
product_id       | 168
id               | 366
product_version  |
fingerprint      |
rejected_by_user | f
rejected_by      |
```

Configuration

- All contained in `postgresql.conf`
  - `postgresql.auto.conf` controlled by `ALTER SYSTEM` commands
- Can be viewed in `psql` with `SHOW ALL;`
- Some can be changed without a system restart
  - `SELECT name, setting FROM pg_settings WHERE context IN ('sighup','user');`
  - `SET <param> TO <value>;`
  - `ALTER SYSTEM SET <param> TO <value>;`
- Commit changes in `psql`: `SELECT pg_reload_conf();`
- From the OS: `systemctl reload` or `kill -HUP ${pid}`

EDB™

Some Useful Config Params

- `shared_buffers`, `max_connections` -- important, but may not be useful for you
- `search_path` -- namespaces to look for tables
- `work_mem` -- memory to allocated for sorting and hashing (be careful)
- `maintenance_work_mem` -- VACUUM performance
- `log_*` params -- controls what gets logged
  - `log_line_prefix`
  - `log_checkpoints`
  - `log_connections/log_disconnections`
  - `log_autovacuum_min_duration`
  - `log_hostname`
- ⚠️ Database logs != WAL logs

EDB™

WAL Logs

- WAL stands for **W**rite **A**head **L**og
- WAL files live in `${PGDATA}/pg_wal/`
  - If you see `pg_xlog`, be very careful, as you're working with an ancient relic🏺
- Basically a journal of all the write activity on the database
- Provides a means of disaster recovery
- Eventually synced/merged with actual database files in `${PGDATA}/base/`
- ⛔ DO NOT DELETE THESE FILES ⛔

EDB™

WAL Logs

```
postgres@davinci:~/15/main$ pwd
/var/lib/postgresql/15/main
postgres@davinci:~/15/main$ ls -al
total 92
drwx------ 19 postgres postgres 4096 Aug 16 21:56 .
drwxr-xr-x  3 postgres postgres 4096 Aug 16 21:52 ..
drwx------  6 postgres postgres 4096 Aug 17 01:34 base
drwx------  2 postgres postgres 4096 Aug 18 09:25 global
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_commit_ts
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_dynshmem
drwx------  4 postgres postgres 4096 Aug 28 09:33 pg_logical
drwx------  4 postgres postgres 4096 Aug 16 21:52 pg_multixact
drwx------  2 postgres postgres 4096 Aug 25 01:42 pg_notify
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_replslot
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_serial
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_snapshots
drwx------  2 postgres postgres 4096 Aug 16 21:56 pg_stat
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_stat_tmp
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_subtrans
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_tblspc
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_twophase
-rw-------  1 postgres postgres    3 Aug 16 21:52 PG_VERSION
drwx------  3 postgres postgres 4096 Aug 25 02:22 pg_wal
drwx------  2 postgres postgres 4096 Aug 16 21:52 pg_xact
-rw-------  1 postgres postgres   88 Aug 16 21:52 postgresql.auto.conf
-rw-------  1 postgres postgres  130 Aug 16 21:56 postmaster.opts
-rw-------  1 postgres postgres  100 Aug 16 21:56 postmaster.pid
```

```
postgres@davinci:~/15/main$ ls -al pg_wal/
total 65548
drwx------  3 postgres postgres     4096 Aug 25 02:22 .
drwx------ 19 postgres postgres     4096 Aug 16 21:56 ..
-rw-------  1 postgres postgres 16777216 Aug 28 09:33 00000001000000000000000B
-rw-------  1 postgres postgres 16777216 Aug 17 01:24 00000001000000000000000C
-rw-------  1 postgres postgres 16777216 Aug 17 01:30 00000001000000000000000D
-rw-------  1 postgres postgres 16777216 Aug 25 02:18 00000001000000000000000E
drwx------  2 postgres postgres     4096 Aug 16 21:52 archive_status
```

Authentication

- Contained in `pg_hba.conf`
- Allows connections to specific databases by specific users and IP addresses
- Changes are committed with HUP or `pg_reload_conf()`

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            password
# IPv6 local connections:
host    all             all             ::1/128                 password
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     password
host    replication     all             127.0.0.1/32            password
host    replication     all             ::1/128                 password
```

Vacuuming

- Upholds performance by preventing bloat
- `UPDATE` or `DELETE` simply flag rows as deleted
- Vacuum flags deleted rows as re-usable for future `INSERT` or `UPDATE`
- Autovacuum will vacuum certain tables after some time
- Usually best to wait for any heavy vacuuming to finish (beware of `autovacuum_vacuum_cost_delay`)
  - If absolutely necessary, use `pg_terminate_backend()`
  - Manually vacuum the table immediately.
  - Run with `SET vacuum_cost_delay TO 0;`

Backups

- `pg_dump`
  - plaintext dump of the database
  - Can filter based on namespace, table
  - Can dump a compressed/binary version as well, to save space
  - Less likely to copy corruption
- `pg_basebackup`
  - Takes a snapshot of the entire `${PGDATA}` directory, includes indexes, FK constraints, etc.
  - Requires `max_wal_senders`, and a user with `REPLICATION` privilege
  - Faster, but if the database is corrupt, the corruption will be copied

Monitoring

- Important logging parameters
  - `log_line_prefix`
    - `%m [%p]: [%l] [txid=%x] user=%u,db=%d,app=%a,client=%r`
  - `log_min_duration_statement`
- Other logging parameters
  - `log_statement` - Logs statement before executing
  - `log_min_error_statement` - Logs specific types of messages
    - `WARNING`, `ERROR`, `FATAL`, `PANIC`
  - `log_duration` - Logs a duration only (consider extension `pg_stat_statements`)
  - `log_connections` - Logs when as session begins

Performance

- `EXPLAIN` v. `EXPLAIN ANALYZE`
  - Query performance can be evaluated in the logs
  - As a developer, `auto_explain` is a very helpful tool, especially if you're using an ORM

Explain v. Explain Analyze

```
postgres=# EXPLAIN SELECT * FROM pgbench_accounts a JOIN pgbench_branches b ON (a.bid=b.bid) WHERE a.aid < 100000;
                                   QUERY PLAN
--------------------------------------------------------------------------------
 Nested Loop  (cost=0.00..4141.00 rows=99999 width=461)
   Join Filter: (a.bid = b.bid)
   ->  Seq Scan on pgbench_branches b  (cost=0.00..1.01 rows=1 width=364)
   ->  Seq Scan on pgbench_accounts a  (cost=0.00..2890.00 rows=99999 width=97)
         Filter: (aid < 100000)
(5 rows)
```

```
postgres=# EXPLAIN ANALYZE SELECT * FROM pgbench_accounts a JOIN pgbench_branches b ON (a.bid=b.bid) WHERE a.aid < 100000;
                                                QUERY PLAN
----------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=0.00..4141.00 rows=99999 width=461) (actual time=0.039..56.582 rows=99999 loops=1)
   Join Filter: (a.bid = b.bid)
   ->  Seq Scan on pgbench_branches b  (cost=0.00..1.01 rows=1 width=364) (actual time=0.025..0.026 rows=1 loops=1)
   ->  Seq Scan on pgbench_accounts a  (cost=0.00..2890.00 rows=99999 width=97) (actual time=0.008..25.752 rows=99999 loops=1)
         Filter: (aid < 100000)
         Rows Removed by Filter: 1
 Planning Time: 0.306 ms
 Execution Time: 61.031 ms
(8 rows)
```

Improving Performance

- Data types
  - Be sure to use the right one
  - Don't use all text
  - JSONB when working with JSON
- Indexing
  - Very important to have proper indexes
  - Identify any needed indexes with `EXPLAIN ANALYZE`

Improving Performance

```
postgres=# UPDATE pgbench_accounts SET bid = aid;
UPDATE 100000
postgres=# EXPLAIN ANALYZE SELECT * FROM pgbench_accounts WHERE bid = 1;
                                                     QUERY PLAN
-----------------------------------------------------------------------------------------------------------------
 Seq Scan on pgbench_accounts  (cost=0.00..5778.24 rows=199939 width=97) (actual time=19.322..45.161 rows=1 loops=1)
   Filter: (bid = 1)
   Rows Removed by Filter: 99999
 Planning Time: 0.101 ms
 Execution Time: 45.191 ms
(5 rows)

postgres=# CREATE INDEX pgba_bid_idx ON pgbench_accounts (bid);
CREATE INDEX
postgres=# EXPLAIN ANALYZE SELECT * FROM pgbench_accounts WHERE bid = 1;
                                                     QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Index Scan using pgba_bid_idx on pgbench_accounts  (cost=0.29..8.31 rows=1 width=97) (actual time=0.076..0.077 rows=1
loops=1)
   Index Cond: (bid = 1)
 Planning Time: 0.312 ms
 Execution Time: 0.119 ms
(4 rows)
```

What NOT to do

- `kill -9` on any Postgres process
  - Causes Postgres to crash and enter into recovery mode
- Idle Transactions
  - Always commit/rollback any transactions
  - Otherwise other users will be held up
  - Look in `pg_stat_activity` for `idle in transaction` sessions (different from simply `idle`)
  - Cross reference with `pg_locks`
- Don't drop anything (columns, schemas, indexes, etc.)
  - Rename them (or wait until the DBA comes back)
- Do not delete any files from `$PGDATA` (especially files in `pg_wal` or `pg_xlog`)
- `\h` and `\?` can be very useful in `psql`

Where to Find Help

- **Postgres Slack** (postgresteam.slack.com)
- **Postgres Community Mailing Lists** (postgresql.org/list)
- **IRC** (postgresql.org/community/irc)
- **Wiki** (wiki.postgresql.org)
- **Docs** (postgresql.org/docs/current)
- **EDB Support** (enterprisedb.com/support-center)

- 🌴 linktr.ee/postgres_help 🌴

THANK YOU

Happy Pi Day! 🥧