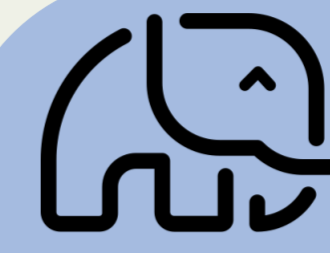


Happiness Hints



Keep close the people who know where your towel is

Data Types

- Use bigint & numeric (not int4 & float)
- Use jsonb (not json)
- Use bytea/text and external storage (not LOBs)

Manage DB schema with code (not pgAdmin, etc)

- Run DDL from app tier (not directly from pipelines/actions)

Use Connection Pooling

- Centralized (eg. pgbouncer) and/or decentralized (eg. JDBC)
 - Use transaction pooling (if at all possible)
- Recycle server connections (eg. server_lifetime)
- Limit or avoid dynamic growth when practical; queue at a tier above the DB

Set Default Limits: Temp Usage, Lock, Statement, Transaction, and Idle Transaction Timeouts

- Lock timeout in seconds. SQL timeouts in minutes. Increase at session level if needed.

Single DB per Instance (when practical)

Enable huge pages (when practical)

Enable checksums (always)

Track IO timing (always)

Use builtin collation pg_c_utf8

- Linguistic sort at expression or column level

Scaling

- Measure:
 - conn count in hundreds (not thousands)
 - conn rate in double digits per minute (not hundreds)
 - table count in thousands (not hundreds of thousands)
 - relation size in GB (not TB)
 - indexes per table in single digits (not double digits)
- Higher ranges work, but require budget for experienced & expensive Postgres staff to manage
- To scale workloads, shard across instances or CAREFULLY partition tables

Move about a bit

Multiple Physical Data Centers (= Multi-AZ on AWS)

Physical Backups

- Minimum 1 month retention
- Regular restore testing

Logical Backups (at least one)

- Scheduled exports/dumps and redrive/replay
- Logical replication

What's missing or poorly worded?
What are **YOUR** happiness hints?

Leave a note! Tell me what to add!

Always have something absurdly pleasant ahead of you

Updates and Upgrades

- Postgres quarterly stable "minors" = security/critical fix only
 - Some commercial forks: minors have new development work
- Before major version upgrade, compare plans and latencies of important SQL on upgraded test copy
- Remember to upgrade extensions; it's not automatic
- Stats/analyze after major version upgrade
 - v18+ automatically transfers most optimizer stats

Make life easier for your future self

Logging

- Minimum 1 month retention
 - on AWS: max retention and publish to cloudwatch
- Log autovacuum min duration = 10 seconds or lower
- Log lock waits
- Log temp usage when close to the default limit
- On AWS: autovacuum force logging level = warning

Pay attention to things while they are happening

Active Session Monitoring (= Database Insights on AWS)

- Save snapshots of pg_stat_activity incl. wait events
- Keep historical data, minimum 1 month (hopefully more)

SQL and Catalog and other DB Stats Monitoring

- Preload pg_stat_statements
 - Purge entries if total query text size exceeds threshold
- Save snapshots of pg_stat_statements & key stats
 - Exec plans (eg. auto_explain or others)
 - Large relation sizes (bytes & row counts, incl catalogs)
 - Unused indexes
 - Rates:
 - tuple fetch & return, DDL statements
 - WAL records & bytes & FPIs
 - XID, subtransaction, multixact
- Keep historical data, minimum 1 month (hopefully more)

OS Monitoring

- Rate of logging output
- OS stats w granularity of 10 seconds or lower (1 second if possible)
- Keep historical data, minimum 1 month (hopefully more)

Don't Panic

Alarms

- Average Active Sessions (= DBLoad on AWS)
- Memory / swap
- Disk: %space & %inodes (+ local storage on Aurora)
- Hot standby & logical repl lag / WAL size on primary
- Unexpected errors in logs (both DB & app tier)
- Max used transaction IDs (aka time to wraparound)
- Checkpoint: time since latest & warnings in logs
- Key expected alarms/metrics/logs are absent or paused