



Monitoring PostgreSQL made simple
pgconf.be 2023

Senior Consultant/Developer

Pavlo Golub

MAIL

pavlo.golub@cybertec.at

Twitter

@PavloGolub

WEB

www.cybertec-postgresql.com



ABOUT CYBERTEC



Highly specialized,
fast growing
IT company



International Team
(10 countries),
six locations worldwide



PostgreSQL Services &
Support



Owner managed
since 2000

AUSTRIA (HQ)

CYBERTEC POSTGRESQL
INTERNATIONAL (HQ)

SWITZERLAND

CYBERTEC POSTGRESQL
SWITZERLAND

URUGUAY

CYBERTEC POSTGRESQL
SOUTH AMERICA

ESTONIA

CYBERTEC POSTGRESQL
NORDIC

POLAND

CYBERTEC POSTGRESQL
POLAND

SOUTH AFRICA

CYBERTEC POSTGRESQL
SOUTH AFRICA

CUSTOMER SECTORS

- ICT
- Finance
- Regulatory
- Automotive
- Production
- Trade
- Universities
- and many more.



DATABASE - PRODUCTS



WHY PostgreSQL?



www.cybertec-postgresql.com



AGENDA

- Different levels of database monitoring
- PostgreSQL monitoring approaches
- PostgreSQL monitoring tools available
- pgwatch2
- pgwatch3? 🤔

Monitoring PostgreSQL made simple



Different levels of database
monitoring

Why to monitor?

- Failure / Downtime detection
- Slowness / Performance analysis
- Proactive predictions
- Maybe wasting money?



Different levels of database monitoring

- High level service availability
- System monitoring
- PostgreSQL land





High level service availability

- Try to periodically connect/query from an outside system
- DIY - e.g. a simple Cron script
- SaaS - lots of service providers

Who will guard the guards?

- You'll probably want two services for more critical stuff...

System monitoring

Too many tools, no real standards. Just make sure to understand what you're measuring!

- Do you know what does the CPU load number actually mean?
 - Is it a good metric?
- What's the difference between VIRT, RES, SHR memory values for a process?



PostgreSQL land

- Log analysis
- Statistics Collector
- Extensions



Log analysis

- “Just in case” storing of logs for possible ad hoc needs
 - Moving logs to a central place makes sense
 - rsync + Cron
- Active parsing
 - grep + Cron
 - DIY (file_fdw, Graylog, ELK, ...)
 - pgBadger (JSON format)
 - Some cloud service (Loggly, Splunk, ...)

Logging configuration

- Some settings to note
 - log_destination (CSV format recommended)
 - log_statement = 'none' (default)
 - log_min_duration_statement / log_duration
 - log_min_messages / log_min_error_statement

```
postgres=# SELECT count(*) FROM pg_settings  
WHERE category LIKE 'Reporting and Logging%';
```

count

Statistics Collector

- Not all `track_*` parameters enabled by default
- Dynamic views
 - `pg_stat_activity`, `pg_stat_(replication|wal_receiver)`
 - `pg_locks`, `pg_stat_ssl`, `pg_stat_progress_*`
- Cumulative views
 - Most `pg_stat_*` views
 - Long uptimes cause “lag” for problem detection
- Selective stats reset possible

Extensions

- Most notably **pg_stat_statements** (“top statements”)
- **pgstattuple** (bloat)
- **pg_buffercache** (what’s in the shared buffers)
- **auto_explain** (e.g. to analyze “jumping runtimes”)
- **pg_qualstats** (WHERE predicate stats)
- ...

Real life

Typically a mixed approach for bigger “shops”

- DYI
 - Log collection / parsing
 - Continuous storing of `pg_stat*` snapshots via some tool
 - Alerting and trends predictions (it's hard!)
- Application performance monitoring (APM)
 - A more high level concept, requires some trust / lock-in
 - AppDynamics, New Relic, DataDog, ...

Monitoring PostgreSQL made simple



PostgreSQL Monitoring Tools

PostgreSQL Monitoring Tools

FIND MORE



<https://wiki.postgresql.org/wiki/Monitoring>

No shortage of tools!

● Ad hoc troubleshooting

Open Source “ad-hoc” tools

- pg_activity
- pgcenter
- pg_top
- pghero
- pgAdmin4
- DBeaver
-

pg_activity

```
PostgreSQL 14.5 - dbserver - postgres@var/run/postgresql:5432/postgres - Ref.: 2s - Duration mode: query
* Global: 10 minutes uptime · 1.75G dbs size · 113.85K/s growth · 99.98% cache hit ratio
  Sessions: 31/100 total · 31 active · 0 idle · 0 idle in txn · 0 idle in txn abrt · 28 waiting
  Activity: 1421 tps · 1416 insert/s · 4249 update/s · 0 delete/s · 10603 tuples returned/s · 0 temp files · 0B temp size
* Worker processes: 0/8 total · 0/4 logical workers · 0/8 parallel workers
  Other processes & info: 0/3 autovacuum workers · 0/10 wal senders · 0 wal receivers · 0/10 repl. slots
* Mem.: 5.68G total · 2.61G (45.94%) free · 1.67G (29.37%) used · 1.40G (24.69%) buff+cached
  Swap: 5.88G total · 5.88G (100.00%) free · 0B (0.00%) used
  I/O: 3023/s max iops · 0B/s - 0/s read · 11.81M/s - 3023/s write
  Load average: 1.12 0.71 0.41
```

RUNNING QUERIES

PID	DATABASE	CPU%	MEM%	READ/s	WRITE/s	TIME+	Waiting	IOW	state	Query
10939	bench	6.2	0.7	0B	369.70K	0.114618	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10912	bench	5.7	0.6	0B	373.50K	0.113902	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10918	bench	7.1	0.6	0B	428.48K	0.069780	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10930	bench	6.6	0.7	0B	420.89K	0.048208	tuple	N	active	UPDATE pgbench_tellers SET tbalanc
10917	bench	6.7	0.6	0B	394.35K	0.042520	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10937	bench	6.7	0.7	0B	401.94K	0.041018	tuple	N	active	UPDATE pgbench_tellers SET tbalanc
10920	bench	6.2	0.7	0B	396.25K	0.032763	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10940	bench	6.7	0.6	0B	420.89K	0.031979	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10913	bench	7.1	0.7	0B	401.94K	0.023534	tuple	N	active	UPDATE pgbench_tellers SET tbalanc
10933	bench	5.2	0.6	0B	337.47K	0.018544	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10936	bench	6.7	0.7	0B	424.69K	0.015634	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10926	bench	6.2	0.6	0B	386.77K	0.014867	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10921	bench	6.6	0.6	0B	400.04K	0.012748	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10924	bench	6.2	0.6	0B	386.77K	0.012305		N	active	UPDATE pgbench_branches SET bbalan
10911	bench	7.1	0.6	0B	413.31K	0.010530	tuple	N	active	UPDATE pgbench_tellers SET tbalanc
10931	bench	6.6	0.7	0B	405.73K	0.009143	tuple	N	active	UPDATE pgbench_tellers SET tbalanc
10923	bench	5.7	0.6	0B	337.47K	0.008341	tuple	N	active	UPDATE pgbench_tellers SET tbalanc
10914	bench	6.6	0.7	0B	413.31K	0.007549	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10935	bench	7.6	0.6	0B	475.88K	0.004729	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10915	bench	5.7	0.6	0B	288.18K	0.004509	tuple	N	active	UPDATE pgbench_branches SET bbalan
10934	bench	7.1	0.7	0B	401.94K	0.004141	transactionid	N	active	UPDATE pgbench_tellers SET tbalanc
10932	bench	7.1	0.6	0B	430.37K	0.003338	tuple	N	active	UPDATE pgbench_tellers SET tbalanc

F1/1 Running queries F2/2 Waiting queries F3/3 Blocking queries Space Pause/unpause q Quit h Help

pgAdmin 4

Dashboard Properties SQL Statistics Dependencies Dependents

Database sessions

Transactions per second

Tuples in

Tuples out

Block I/O

Server activity

Sessions Locks Prepared Transactions

	PID	User	Application	Client	Backend start	State	Wait Event	Blocking PIDs
✖️	19172	enterprisedb	pgAdmin 4 - DB:edb	172.19.13.220	2019-02-06 12:31:11 IST	idle	Client: ClientRead	
✖️	27463	enterprisedb	pgAdmin 4 - DB:edb	172.19.13.220	2019-02-06 13:23:27 IST	active	:	
✖️	31875	enterprisedb	Postgres Enterprise Manager - Agent Mo...	127.0.0.1	2019-02-06 13:51:20 IST	idle	Client: ClientRead	
✖️	31877	enterprisedb	Postgres Enterprise Manager - Agent Mo...	127.0.0.1	2019-02-06 13:51:20 IST	idle	Client: ClientRead	

Search

Continuous monitoring frameworks

Commercial (mostly “agent” based)

- AppDynamics
- New Relic
- Datadog
- Vividcortex
- EDB Enterprise Manager
- pganalyze
- ...

Continuous monitoring frameworks

Generic Open Source

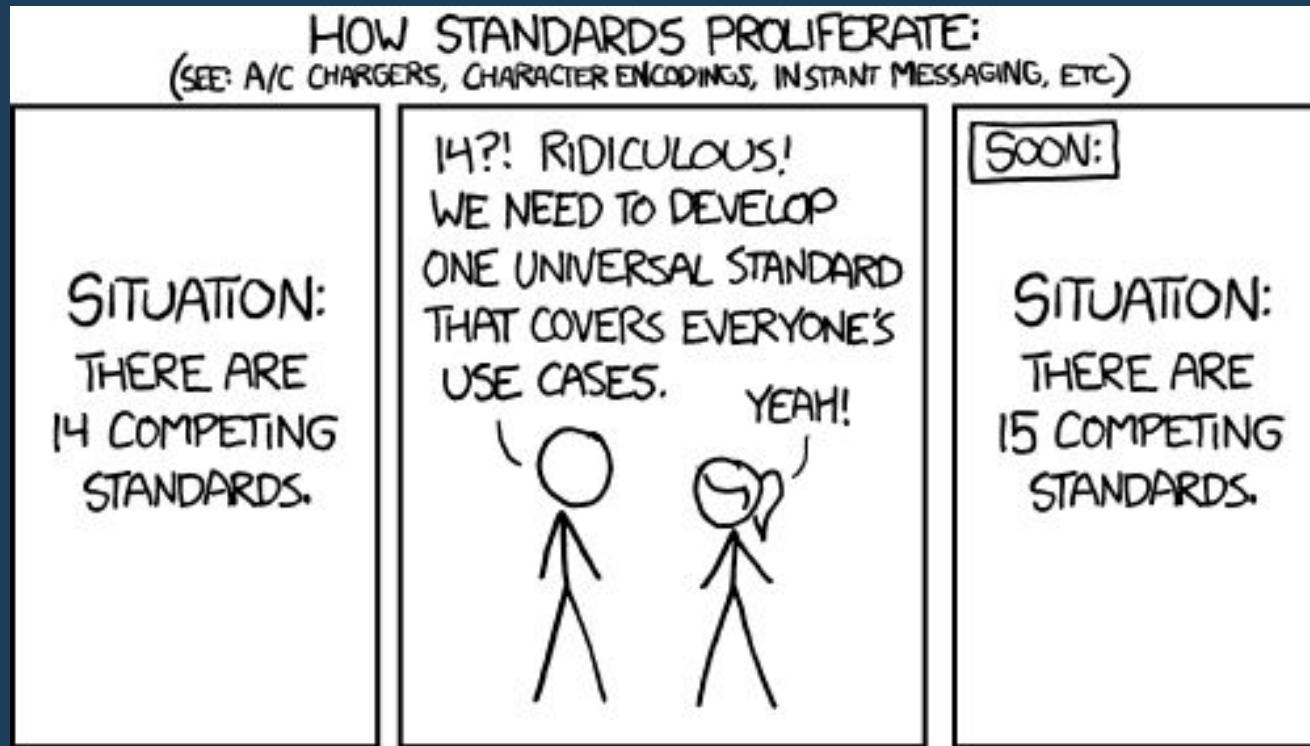
- Nagios
- Icinga
- Munin
- Zabbix

Base mostly on “check_postgres” script or derivatives

Postgres specific

- pgHero
- PoWA (server side, quite advanced - pg_qualstats, pg_stat_kcache, pg_wait_sampling)
- PgCluu (server side, with “sar” system info)
- pgwatch2 (client or server side)
- ...

● pgwatch2?



Monitoring PostgreSQL made simple



pgwatch2



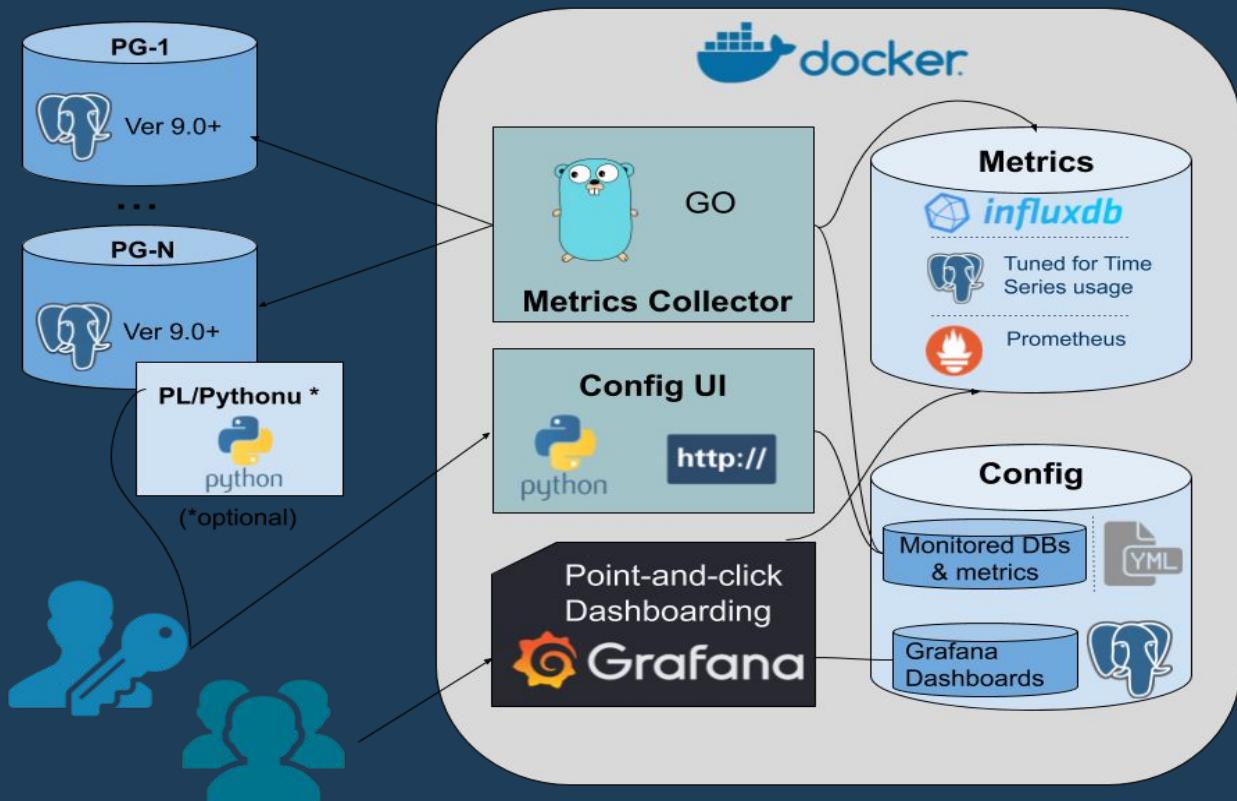
Main principles - why another tool?

- 1-minute setup
 - Docker (custom setup also possible)
- User changeable visuals / dashboarding
- Non-invasive
 - No extensions or superuser needed for base functionality
- Easy extensibility, do minimal work needed
 - SQL metrics
- Simple alerting via Grafana possible

Architecture components

- Metrics gathering daemon (Golang)
- Configuration database / YAML files / ENV vars
- Metrics storage layer
 - PostgreSQL (with TimescaleDB optionally)
 - InfluxDB (deprecated)
 - Graphite
 - Prometheus
- Optional simple Web UI for administration
- Grafana for intuitive point-and-click dashboarding

Architecture components





Features

- “Ready to go”
 - Default metrics cover almost all pg_stat* views
 - Pre-configured dashboards for almost all metrics
- Supports Postgres 9.0+ out of the box
- Configurable security - passwords, SSL
- Reuse of existing Postgres, Grafana, InfluxDB installations possible
- Kubernetes/OpenStack ready

Features

- Per DB setup with optional auto-discovery of all DBs of a cluster
- Change detection for table/index/sproc/configuration events
- AWS RDS CloudWatch metrics support
- PgBouncer & Pgpool2 metrics support
- Patroni support
- Very low resource requirements
- A Ping mode to test connectivity to monitored databases
- Extensible
 - Custom metrics via SQL, i.e. usable also for business layer!
 - Built-in log parsing and OS level metrics
 - Grafana has plugins for visual gimmicks also

Alerting

- Quite easy with Grafana, “point-and-click”
 - Most important alerting services covered
 - Email
 - Slack
 - PagerDuty
 - Web hooks
 - ...
- Based on graph panels only currently
- Use **pg_timetable** scheduler to analyze and send alerts



Getting started

1. docker run -d --restart=unless-stopped \
-p 3000:3000 -p 8080:8080 \
--name pw2 cybertec/pgwatch2-postgres
2. Wait some seconds and open browser at localhost:8080
3. Insert your DB connection strings
4. Start viewing/editing dashboards in 5 min...

or check online demo at **<https://demo.pgwatch.com>**

Admin web UI

PgWatch2 CYBERTEC

DBs | Metrics | Logs | Log out

Databases under monitoring

ID	Unique name	DB host	DB port	DB dbname <small>?</small>	DB user	DB password	Is superuser?	SSL Mode	Preset config	Custom config	Statement timeout [seconds]	Last modified	Enabled?	
1	test	localhost	5432	pgwatch2	pgwatch2	...	<input type="checkbox"/>	disable	exhaustive	<input type="text"/> 5	2017-09-19 19:54:05+00:00	<input checked="" type="checkbox"/>	<button>Save</button> <button>Delete</button>	
		sales	db.host1.com	5432	app1	monitor	<input type="checkbox"/>	disable	exhaustive	<input type="text"/> 5	2017-09-19 19:54:05+00:00	<input checked="" type="checkbox"/>	<button>New</button>

Active metrics listing

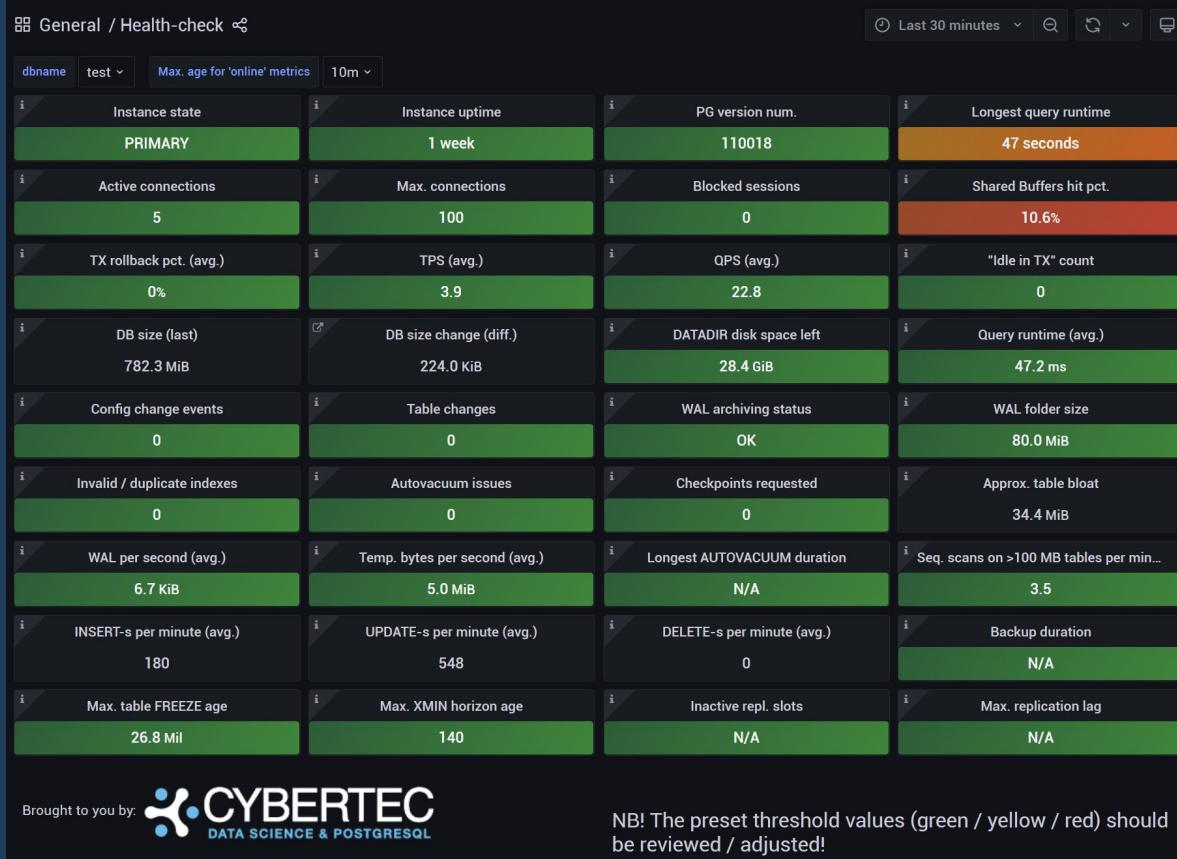
```
backends [ver: 9.9.6] bgwriter [ver: 9] blocking_locks [ver: 9.2] buffercache_by_db [ver: 9.2] buffercache_by_type [ver: 9.2] change_events [ver: 9] configuration_hashes [ver: 9] cpu_load [ver: 9] db_stats [ver: 9] get_load_average [ver: 9] get_stat_statements [ver: 9] get_table_bloat_approx [ver: 9.5] index_hashes [ver: 9] index_stats [ver: 9] kpi [ver: 9.9.6,10] locks [ver: 9] locks_mode [ver: 9] pg_stat_database_conflicts [ver: 9.2] pg_stat_ssl [ver: 9.5] replication [ver: 9.1,10] sproc_hashes [ver: 9] sproc_stats [ver: 9] stat_statements [ver: 9.2] stat_statements_calls [ver: 9.2] table_bloat_approx_stattuple [ver: 9.5] table_bloat_approx_summary [ver: 9.5] table_hashes [ver: 9] table_io_stats [ver: 9] table_stats [ver: 9] wal [ver: 9.2,10]
```

InfluxDB metrics data cleanup

DB "Unique name" (NB! It could take up to 3min for background gatherers to stop so no point to click directly after removing a host from monitoring):

Delete single DB metrics Delete all metrics for all non-active DBs

Health Check Dashboard



Database Overview Dashboard



Stat Statements Dashboard



Monitoring PostgreSQL made simple



pgwatch3?

CLASSIFIED

DON'T BE A STRANGER



PERSONAL GITHUB

www.github.com/pashagolub



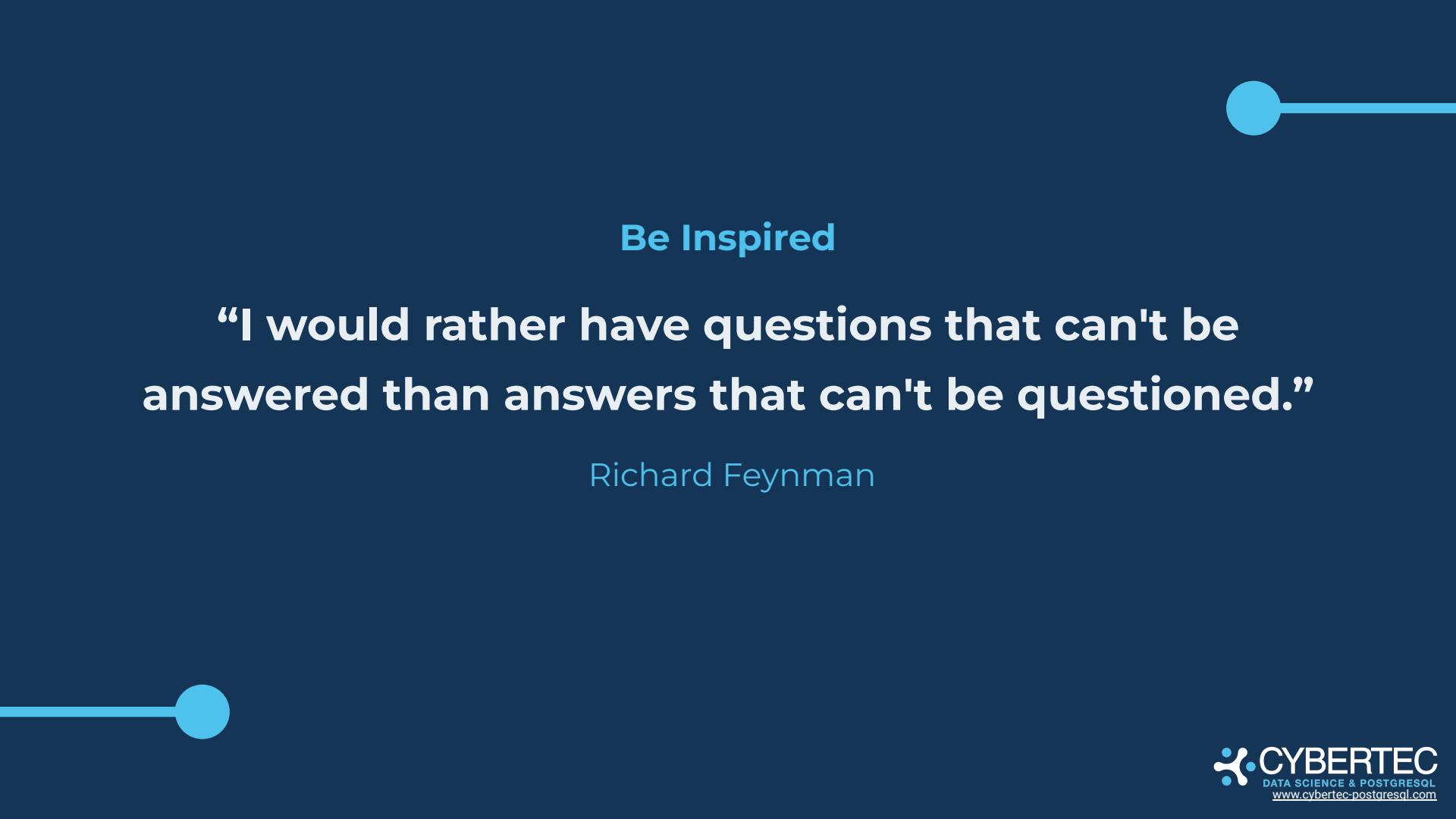
CYBERTEC BLOG

www.cybertec-postgresql.com/en/blog/



CYBERTEC GITHUB

www.github.com/cybertec-postgresql



Be Inspired

**“I would rather have questions that can't be
answered than answers that can't be questioned.”**

Richard Feynman

#StandWithUkraine

