

POSTGRESQL- AS-A-SERVICE

Vergleich von Cloud Providern

Michael Banck <michael.banck@netapp.com>
NetApp Deutschland GmbH
PGConf.DE 2024



Agenda

- Funktionalität, Features und Erweiterungen
- Major/Minor Versions-Aktualität
- Community Engagement
- Widerstandskraft gegen Ressourcen-Knappheit
- Server-Veränderungen

Prior Art

- Christophe Pettus, FOSDEM PGDay 2020: “Hosted Environments for PostgreSQL: An Objective Look”
 - <https://www.postgresql.eu/events/fosdem2020/sessions/session/2920/slides/282/hosted-postgresql-fosdem-2020.pdf>

- Michael Banck, Percona Live Online 2021: “PostgreSQL-As-A-Service: Comparison of Cloud Providers”
 - https://share.credativ.com/~mba/talks/postgresql-as-a-service_percona_live_online_2021.pdf

Überblick

Betrachtet wurden:

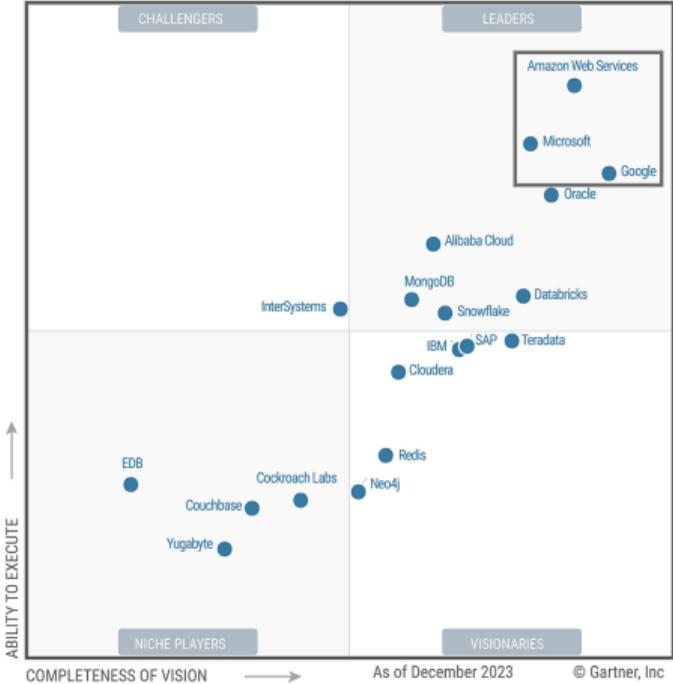
- Microsoft Azure Database for PostgreSQL (Azure)
- Google Cloud SQL for PostgreSQL (GCSQL)
- Amazon RDS for PostgreSQL (RDS)

Nicht betrachtet wurden:

- Amazon Aurora
- Azure Database for PostgreSQL - Hyperscale (Citus)
- Kleinere Managed Postgres Provider: Heroku Postgres, Supabase, Aiven, Tembo Cloud, Neon, Crunchy Bridge, BigAnimal, Instaclustr

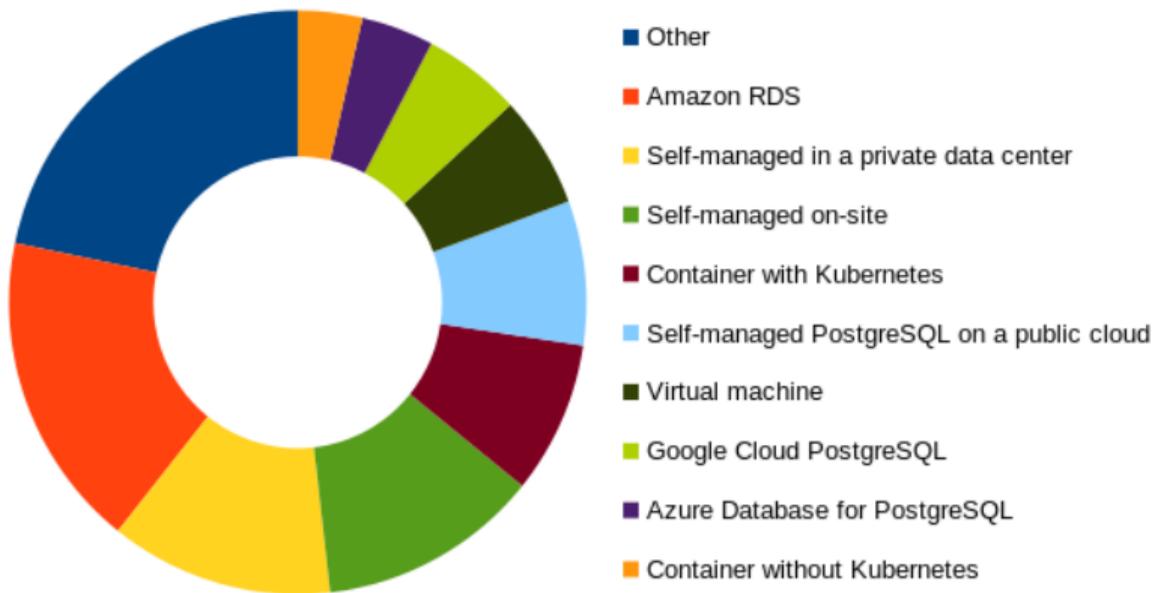
Gartner 2023 Magic Quadrant

Cloud Database Management Systems



Timescale State of Postgres 2023

How do you deploy PostgreSQL?



FUNKTIONALITÄT FEATURES ERWEITERUNGEN

Gemeinsame Funktionalität

Alle Postgres-As-A-Service Provider bieten:

- SQL-Zugang, keinen Shell-Zugriff
- Self-Service, normalerweise wird keine Interaktion mit Support benötigt
- Kein SUPERUSER, aber benötigte Funktionalität ist vorhanden
 - Instanzbesitzer mit CREATEROLE, CREATEDB Rechten
 - Zusätzliche Rechte über vordefinierte Rollen / Server-Veränderungen
 - Viele Erweiterungen verfügbar
- Web-Interface/Portal/Console, SDK/CLI und Terraform Integration
 - Instanz Erstellung/Löschung/Neustart
 - Änderung von Konfigurations-Parametern
 - Protokolldateien in irgendeiner Form
 - Whitelisting von IP Adressen/Bereichen

Gemeinsame Funktionalität

- Storage-Verschlüsselung mit Vendor- oder Kunden-Schlüssel
- Storage-Vergrößerung ohne Downtime
- Instanzgrößen Hoch/Runterskalierung
- IOPS skalieren mit Größe des provisionierten Storage, mehrere Storage-Typen verfügbar
- Hoch-Verfügbarkeit
- Lese-Replikas
- Logische Replikation
- Automatische Instanz-weite Backups
- (Point-in-Time) Restore von Backups durch die Erstellung einer neuen Instanz
- Automatische Installation von Patch-Releases während Wartungsfenstern
- Kunden-initiierte Major-Version Upgrades

Feature-Vergleich

	RDS	GCSQL	Azure
Verfügbar seit	2013	2018	2020
Neueste Major Version	16	15	16
Früheste Major Version	11	9.6	11
Aktuelles Maintenance Release	24Q1	23Q4	24Q1
Konfigurations-Parameter	386	250	243
Erweiterungen	92	74	60
Pooler-Integration	X	-	X
AD-Integration	X	-	X
Speicher GUC Änderung bei Resize	X	X	X
CPU GUC Änderung bei Resize	X	-	-

Erweiterungen

- Core Erweiterungen, die bei allen vorhanden sind:
 - amcheck, bloom, btree_gin, btree_gist, citext, cube, dblink, dict_int, dict_xsyn, earthdistance, fuzzystmatch, hstore, intagg, intarray, isn, lo, ltree, pg_buffercache, pg_freespacemap, pg_prewarm, pg_trgm, pg_visibility, pageinspect, pgcrypto, pgrowlocks, pgstattuple, plpgsql, postgres_fdw, sslinfo, tablefunc, tsm_system_rows, tsm_system_time, unaccent, uuid-oss
- Externe Erweiterungen, die bei allen vorhanden sind:
 - address_standardizer, address_standardizer_data_us, orafce, pg_repack, pg_partman, pglogical, postgis, postgis_raster, postgis_tiger_geocoder, postgis_sfcgal, postgis_topology, vector
- Gemeinsame shared_preload_libraries Erweiterungen:
 - auto_explain, pgbaudit, pglogical, pg_cron, pg_hint_plan, pg_stat_statements

Erweiterungen

Bedeutende Externe Erweiterungen (1/2)

Erweiterung	RDS	GCSQL	Azure
anon	-	1.0.0	1.2.0
hll	2.18	2.18	-
hypopg	1.4.0	-	1.4.0
ip4r	2.4	2.4	-
orafce	4.9	4.7	4.4
pg_bigm	1.2	1.2	-
pg_failover_slots	-	-	1.0.1
pg_partman	5.0.0	4.7.4	4.7.1
pg_proctab	0.0.10	0.0.10	-
pg_repack	1.5.0	1.5.0	1.4.7
pg_similarity	1.0	1.0	-

Erweiterungen

Bedeutende Externe Erweiterungen (2/2)

Erweiterung	RDS	GCSQL	Azure
pg_squeeze	-	1.5	1.6
pg_tle	1.1.1	-	-
pg_wait_sampling	-	1.1	-
pgactive	2.1.2	-	-
plprofiler	4.2	-	-
pgrouting	3.5.0	3.3.0	-
plproxy	-	2.11.0	-
postgis	3.4.1	3.2.5	3.3.3
prefix	1.2.0	1.2.0	-
rdkit	4.4.0	4.3.0	-
vector	0.6.0	0.5.1	0.6.1

Erweiterungen

Prozedurale Sprachen und Foreign Data Wrapper

Erweiterung	RDS	GCSQL	Azure
bool_plperl	X	-	-
hstore_plperl	X	-	-
jsonb_plperl	X	-	-
plcoffee	X	-	-
plperl	X	-	-
plrust	X	-	-
pltcl	X	-	-
plv8	X	X	X
mysql_fdw	X	-	-
oracle_fdw	X	X	-
tds_fdw	X	-	X

Server Logdateien

- Amazon RDS
 - Stündliche Logdateien - Ansicht, Folgen oder Download
 - CloudWatch Alarmierungen
- Google Cloud SQL
 - Logs Explorer
 - Suche, Ansicht, Folgen oder Download von Logdateien
- Microsoft Azure Database
 - Stündliche Logdateien - Download
 - Microsoft Log Analytics

Auditierung

- PGAudit Erweiterung ist bei allen verfügbar
- Objekt- (`pgaudit.role = auditor`) oder Session- (Datenbank/Rollen-basiert) Auditing möglich
 - `GRANT SELECT ON table(column) TO auditor;`
 - `ALTER DATABASE database SET pgaudit.log = 'write'`
 - `ALTER USER user SET pgaudit.log = 'read'`
- Amazon RDS und Microsoft Azure Database speichern AUDIT: Logzeilen in den Server-Logs
 - Können durch Suche nach AUDIT: gefunden werden
 - Können von Alarmierungen oder Watchern verwendet werden
- Google Cloud SQL speichert AUDIT: Logzeilen separat von den Server-Logs

Benchmarks

- Timing von einigen üblichen Operationen
- Zeitnahme zwischen Initiierung der Aktion und Notifikation der Beendigung

Operation	RDS	GCSQL	Azure
Instanz Erstellung	3:35	2:45	5:25
Downtime beim Node-Neustart	0:20	0:35	0:20
Downtime beim Ändern der Cluster-Grösse	5:00	2:50	5:50
Laufzeit der Cluster-Änderung	5:15	4:30	7:15
Downtime während Major Upgrade	4:00	5:30	12:00
Laufzeit des Major Upgrades	7:00	10:30	13:15
Verzögerung von Logzeilen	0:07	0:20	5:00

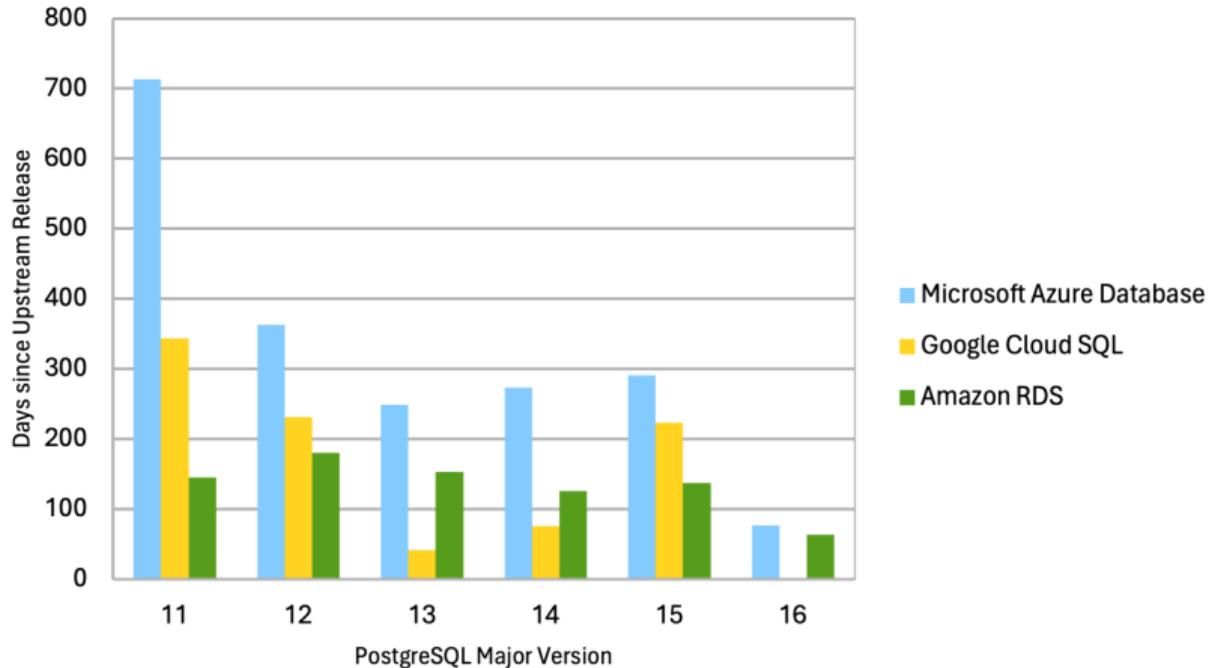
Funktionalität, Features und Erweiterungen

Zusammenfassung

- Hauptsächliche Funktionalität wird von allen drei Providern zur Verfügung gestellt
- Alle drei Provider bieten die beliebtesten und wichtigsten Erweiterungen
- Amazon RDS hat beste Abdeckung bei FDWs und prozeduralen Sprachen
- Amazon RDS zeigt Logzeilen am schnellsten an, Microsoft Azure Database ist hier viel langsamer

MAJOR/MINOR VERSIONS-AKTUALITÄT

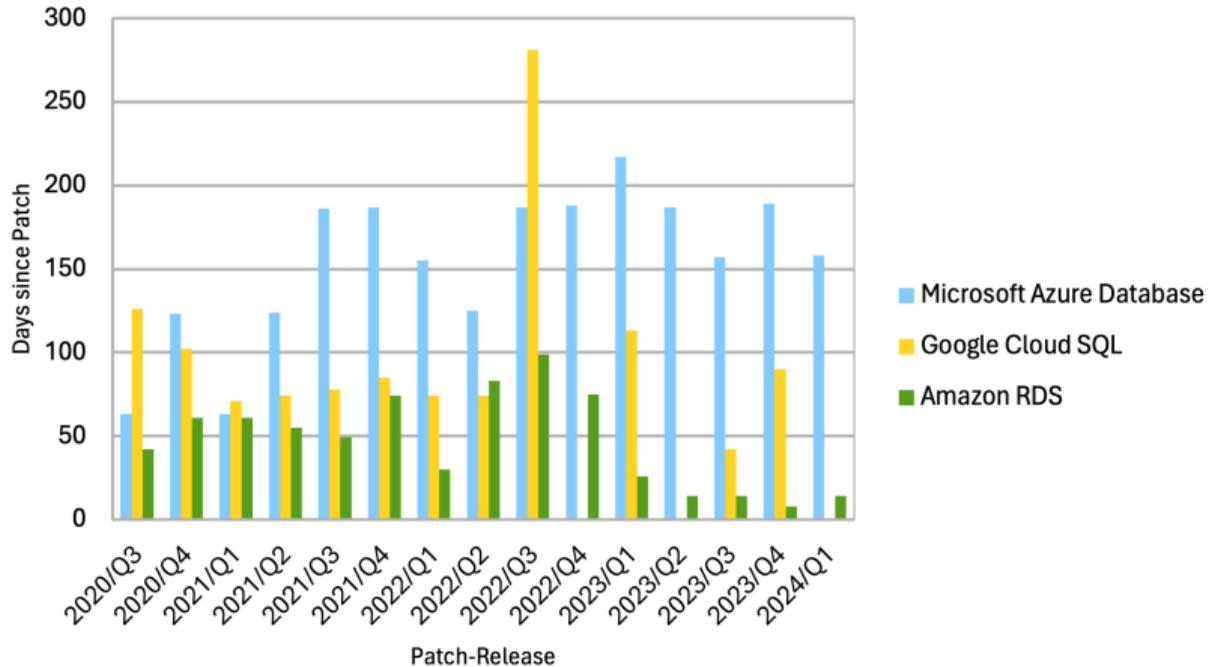
Verzögerung bei der Freigabe von Major Versionen



Verzögerung bei Patch-Releases

- Schwierig zu messen ohne andauernd Server zu provisionieren/monitoren
- Release Note Daten sind nicht immer exakt
- Roll-Out von neuen Maintenance Releases kann graduell erfolgen, vor allem bei Google Cloud SQL
- Microsoft Azure Database hat nur Monatsangaben in Release Notes, keine exakten Daten
 - Monatsmitte angenommen

Verzögerung bei Patch-Releases



Major/Minor Versions-Aktualität

Zusammenfassung

- Amazon RDS hat den Besten Track Record, sowohl für Major also auch Minor Versionen
- Microsoft Azure Database ist langsamer, aber darin konsistent
- Google Cloud SQL war besser als Azure in 2021, aber seitdem eher erratisch

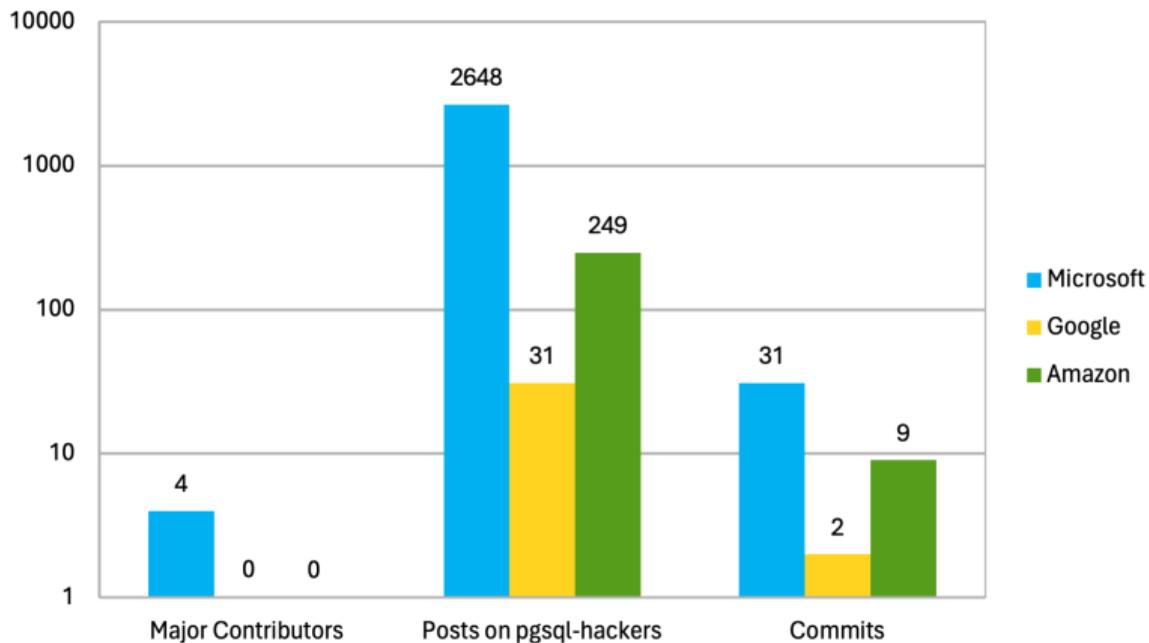
COMMUNITY ENGAGEMENT

Community Mitglieder

- Amazon
 - Hat OpenSCG in 2018 übernommen
 - Core-Team: Jonathan Katz (kein Committer)
 - Major Contributoren: Nathan Bossart, Joe Conway, Jeff Davis, Peter Geoghegan, Michael Paquier, Masahiko Sawada, Dave Cramer (kein Committer)
- Google
 - Major Contributoren: Noah Misch
- Microsoft
 - Hat Citus in 2019 übernommen
 - Core-Team: Andres Freund
 - Major Contributoren: Daniel Gustafsson, Amit Langote, Thomas Munro, David Rowley, Melanie Plageman (kein Committer)

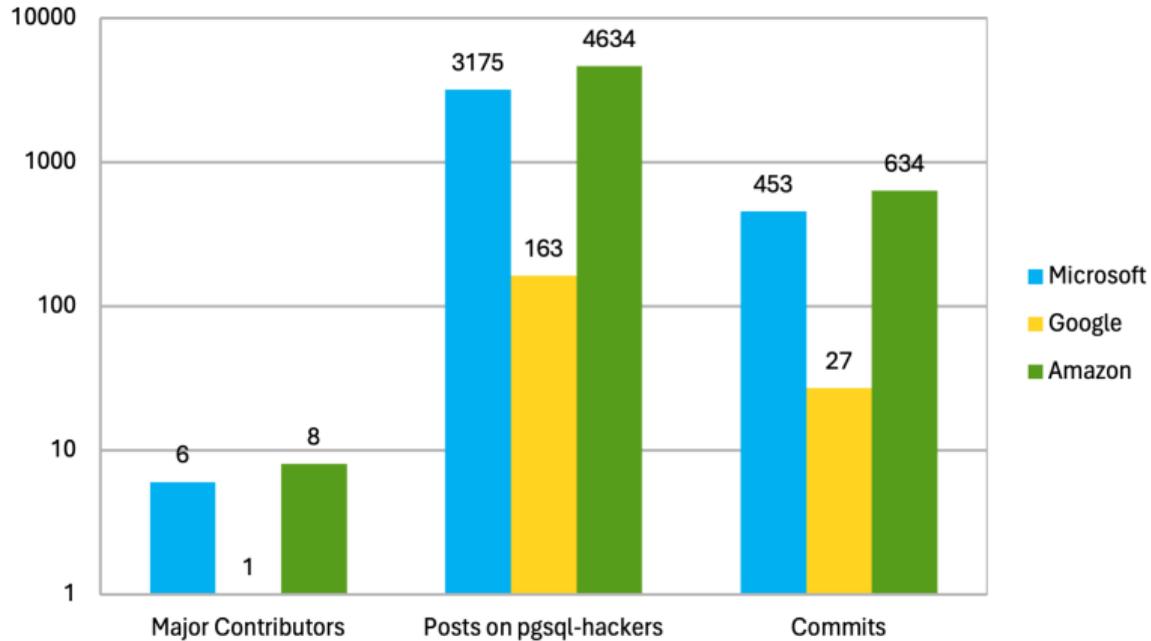
Community Engagement

v14 Entwicklungs-Zyklus



Community Engagement

v17 Entwicklungs-Zyklus



Community Engagement

Konferenz-Sponsoring

	RDS	GCSQL	Azure
PGConf.DE 2022	-	-	-
PGConf.EU 2022	-	S	G
PGConf.DE 2023	-	-	B
PGConf.EU 2023	B	-	P
PGConf.DE 2024	-	G	P

- Sponsoren Level:

- P: Platin
- G: Gold
- S: Silber
- B: Bronze

Community Engagement

Zusammenfassung

- Im Vergleich zu vor drei Jahren sind alle Zahlen verbessert
- Amazon RDS ist an Microsoft Azure vorbeigezogen in Bezug auf Community Engagement
- Google läuft weiterhin deutlich den anderen beiden hinterher
- Alle drei großen Managed Postgres Provider zusammen haben inzwischen einen signifikanten Anteil an der Postgres Entwicklung:
 - 28% von allen Core-Team Mitgliedern / Major Contributoren
 - 31% von `pgsql-hackers` Mails im Verlauf des letzten Jahres
 - 48% von v17 Commits
- Microsoft sponsort europäische Postgres-Konferenzen deutlich mehr als Amazon und Google

WIDERSTANDSKRAFT GEGEN RESSOURCEN-KNAPPHEIT

Out-Of-Memory Verhalten

- Das korrekte Verhalten wurde 2021 bei Microsoft Azure Database beobachtet:

```
postgres=> SELECT * FROM pg_class AS a, pg_class AS b, pg_class AS c
postgres-> ORDER BY random();
```

```
ERROR:  out of memory
```

```
DETAIL:  Failed on request of size 512 in memory context "Caller tuples".
```

```
ERROR:  out of memory
```

```
DETAIL:  Failed on request of size 512.
```

```
STATEMENT:  ANALYZE SELECT * FROM pg_class AS a, pg_class AS b,
            pg_class AS c ORDER BY random();
```

Out-Of-Memory Verhalten

- Heutzutage crashen alle Managed Service Provider

```
postgres=> SELECT * FROM pg_class AS a, pg_class AS b, pg_class AS c  
postgres-> ORDER BY random();
```

server closed the connection unexpectedly

This probably means the server terminated abnormally
before or while processing the request.

The connection to the server was lost. Attempting reset: Failed.

LOG: server process was terminated by signal 9: Killed

LOG: terminating any other active server processes

LOG: all server processes terminated; reinitializing

FATAL: the database system is in recovery mode

Verbindungs-Knappheit

FATAL: remaining connection slots are reserved for non-replication superuser connections

- Pseudo-Superuser Rolle darf bestehende nicht-Superuser Verbindungen abbrechen/beenden
- Wenn `max_connections - superuser_reserved_connections` erreicht sind können keine weiteren Verbindungen mehr aufgebaut werden
- Nur (nicht-verfügbare) Superuser können sich dann via `reserved connections` verbinden
 - Nur Amazon RDS erlaubt seinem `rds_superuser` sich weiterhin einzuloggen
 - `rds.rds_superuser_reserved_connections`
- Alle drei erlauben dem Kunden den Parameter `max_connections` zu ändern
- Ab v16 kann für die vordefinierte Rolle `pg_use_reserved_connections` eine Anzahl an `reserved_connections` vorgehalten werden
 - Wird bisher nur von Amazon RDS (als Ersatz für `rds.rds_superuser_reserved_connections`) verwendet

SERVER-UMGEBUNG UND VERÄNDERUNGEN

Instanz-Umgebung und Veränderungen

- SUPERUSER Nutzer werden vom Cloud-Provider für Wartungsaufgaben verwendet
- Alle Provider bieten einen initialen (Login-)Nutzer, der weitere Nutzer und Datenbanken anlegen kann
- Alle Provider bieten eine pseudo-SUPERUSER Gruppe, die den initialen (Login-)Nutzer als Mitglied hat
 - Amazon RDS: `rds_superuser`
 - Google Cloud SQL: `cloudsqlsuperuser`
 - Microsoft Azure Database: `azure_pg_admin`
- Alle Provider bieten Erweiterungs-Management
- Zugriff auf vordefinierte Rollen mit Betriebssystem-Zugriff (`pg_read_server_files` etc.) bei allen nicht möglich
- Zugriff auf vordefinierte Rolle `pg_monitor` von allen erlaubt

Server-Veränderungen

```
postgres=> ALTER ROLE test SET log_min_duration_statement = '5s';  
ERROR:  permission denied to set parameter "log_min_duration_statement"
```

- Server-Veränderungen nicht mehr so wichtig wie früher auf Grund von Weiterentwicklungen in Postgres
- Amazon RDS
 - Per-Rolle/Datenbank Audit-Einstellungen
 - Per-Rolle/Datenbank Log-Einstellungen
 - Pseudo-SUPERUSER reserved connection pool
 - Tablespace-Management
- Google Cloud SQL
 - Per-Rolle/Datenbank Audit-Einstellungen
- Microsoft Azure Database
 - Per-Rolle/Datenbank Audit-Einstellungen
- Sicherheits-Probleme
 - <https://www.wiz.io/blog/the-cloud-has-an-isolation-problem-postgresql-vulnerabilities>
 - <https://www.wiz.io/blog/wiz-research-discovers-extrareplica-cross-account-database-vulnerability-in-azure-postgresql>

Server-Veränderungen

Zusammenfassung

	RDS	GCSQL	Azure
Service-spezifische GUCs	10	17	15
Pseudo-SU-Kontext GUCs	37	9	0
Erweiterungs-Management	X	X	X
<code>pg_read_server_files</code> blockiert	X	X	X
<code>pg_monitor</code> erlaubt	X	X	X
Per Rollee/DB Audit/Log Einstellung	X	X	X
Pseudo-SU reservierte Verbindungen	X	-	-
Tablespace-Management	X	-	-

ZUSAMMENFASSUNG

Zusammenfassung

- Alle drei Managed Postgres Provider sind ausgereift und bieten umfassende Funktionalität
- Aktuelle High-End Hardware zusammen mit verfügbaren Datenbank-Admins vermutlich besser und billiger
- Am Besten verwendet man den Managed Service von der Plattform, die bereits in Verwendung ist
- Amazon RDS ist das populärste und ausgereifteste Produkt der drei
- Amazon RDS Update-Zyklus ist der sinnvollste und am besten vorhersehbare
- Microsoft und Amazon sind deutlich mehr in die PostgreSQL Entwicklung involviert als Google

THANK YOU