



PERCONA
LIVEONLINE
MAY 12 - 13th
2021

PostgreSQL-As-A-Service

Comparison of Cloud Providers

Michael Banck <michael.banck@credativ.de>
Percona Live Online 2021

- Functionality
- Community
- Major/Minor Versions Up-To-Dateness
- Out-Of-Memory Behaviour

- 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>

Being looked at:

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

Gartner 2020 Magic Quadrant of Cloud DBMS



Functionality

All Postgres-As-A-Service vendors provide:

- SQL access, no shell access
- No SUPERUSER, but required functionality present
- Web-interface/portal, SDK/CLI and Terraform integration
 - Create/Delete/Restart instances
 - Changing server parameters
 - Server logs in some form
 - Whitelisting of IP addresses/ranges
- Replication/high-availability of some sort
- Automatic instance-wide backups
- Restore of backups by spinning up of a new instance
- Automatic installation of minor releases during maintenance windows

- Single Server (Azure SS)
 - Windows-based
 - Supports 9.6, 10 and 11
 - Read-Only Replicas, manual failover
 - Azure Active Directory integration
- Flexible Server (Azure FS)
 - Preview [1]
 - Linux-based
 - Supports 11 and 12
 - Synchronous (cold) Standby, automatic failover
 - Azure Active Directory integration
 - Transparent pgBouncer integration

[1] <https://docs.microsoft.com/en-us/azure/postgresql/flexible-server/release-notes#release-april-26-2021>

- Supports 9.6, 10, 11, 12 and 13
- Synchronous (cold) Standby, automatic failover
- Read replicas

- Supports 9.6, 10, 11, 12 and 13
- Amazon proprietary Multi-AZ with cold Standby, automatic failover
- Read replicas

- Azure, Single Server
 - Hourly log files - download only
 - Microsoft Log Analytics
- Azure, Flexible Server
 - No direct server log access
 - Microsoft Log Analytics
- GCP
 - No direct server log access
 - Logs Explorer - stream or save query results
- RDS
 - Hourly log files - view, watch or download
 - CloudWatch alarms

- Extensions provided by all:

address_standardizer, address_standardizer_data_us, btree_gin, btree_gist, citext, dblink, dict_int, earthdistance, fuzzystmatch, hstore, intarray, isn, ltree, pg_buffercache, pg_prewarm, pg_stat_statements, pg_trgm, pgaudit, pgcrypto, pgrowlocks, pgstattuple, plpgsql, postgis, postgis_tiger_geocoder, postgis_topology, postgres_fdw, tablefunc, unaccent

- Total number of available Extensions:
 - Azure Single Server: 39
 - Azure Flexible Server: 45
 - GCP: 47
 - RDS: 68

Extension	Azure SS	Azure FS	GCP	RDS
amcheck	-	X	-	X
bloom	-	X	-	X
hll	-	-	X	X
pg_freespacemap	-	X	X	X
pg_partman	X	-	X	X
pg_repack	-	-	X	X
pg_wait_sampling	X	-	-	-
pgfincore	-	-	X	-
pgrouting	X	-	-	X
plproxy	-	-	X	-
plv8	X	-	-	X

Extension	Azure SS	Azure FS	GCP	RDS
auto_explain	X	-	-	X
orafce	-	-	-	X
hypopg	-	X	-	-
pgaudit	X	X	X	X
pglogical	-	X	-	X
pg_bigm	-	-	-	X
pg_cron	-	X	-	X
pg_hint_plan	-	-	-	X
pg_prewarm	-	X	-	X
pg_stat_statements	X	X	X	X
plprofiler	-	-	-	X
timescaledb	X	-	-	-

- pgaudit extension is available from all vendors
 - Activation on Google Cloud SQL not possible via web UI
 - `gcloud sql instances patch ${INSTANCE} --database-flags cloudsql.enable_pgaudit=on`
- AUDIT: log lines are emitted to the Postgres log, can be consumed by log watchers, alerts etc.
- Session-based auditing (`pgaudit.role = auditor`) or role-based auditing
- Role-based auditing only possible on Google Cloud SQL and Amazon RDS
 - Errors on Azure, making targetted auditing difficult/impossible

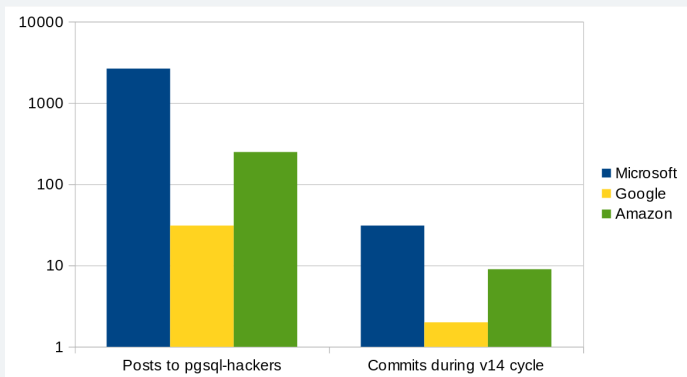
```
postgres=> ALTER ROLE mba SET pgaudit.log = 'WRITE';  
ERROR:  permission denied to set parameter "pgaudit.log"
```


Community

- Microsoft
 - Acquired Citus in 2019
 - Andres Freund: Core Team, Committer
 - Jeff Davis, Thomas Munro, David Rowley: Major Contributor, Committer
 - Dimitri Fontaine: Contributor
- Google
 - Hannu Krosing: Contributor
 - Paul Martinez: Patch Author
- Amazon
 - Acquired OpenSCG in 2018
 - Jan Wieck: Core Team Alumni [1]
 - Nathan Bossart, Jim Nasby: Contributor, Patch Author
 - Bertran Drouvot, James Hunter: Patch Author

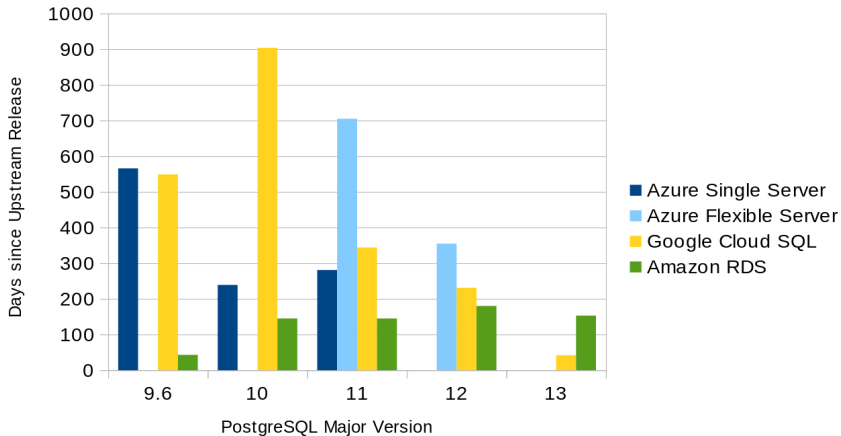
[1] <https://blog.crunchydata.com/news/jan-wieck-former-postgres-core-team-member-joins-crunchy-data>

- Posts to the central postgres-hackers development list during v14 development cycle (03/2020-03/2021)
- Commits authored during v14 development cycle

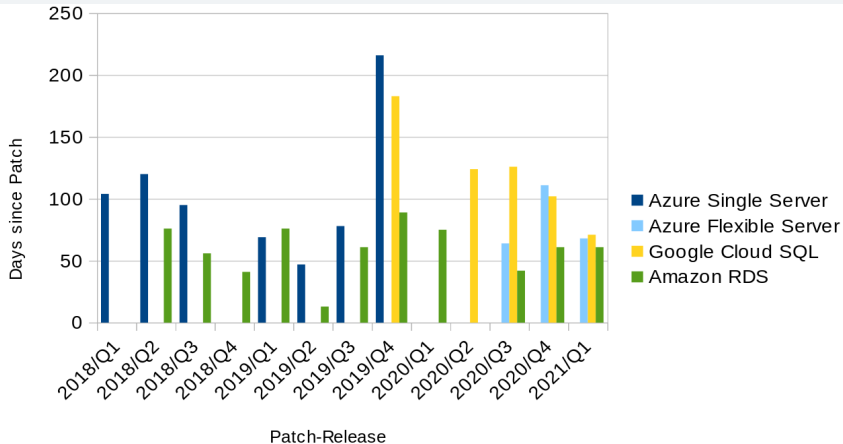


Major/Minor Versions Up-To-Dateness

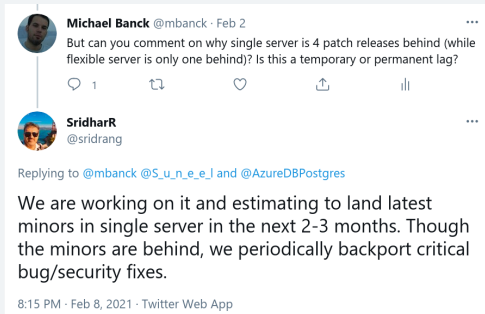
Major Version Introduction Delay



Patch-Release Delay



- Amazon RDS has the most consistent track record
- Information about Google Cloud SQL patch dates from before 2020 unavailable
- Azure Single Server seems to be abandonware, Flexible Server taking over
- Azure team claims Single Server security issues are being patched:



Out-Of-Memory Behaviour

- The environment should be configured so that PostgreSQL does not get killed by the Out-Of-Memory (OOM) Killer
- The proper response to memory pressure is an “out of memory” (ENOMEM) error and aborting the query demanding the memory allocation
- For an on-metal server, that means disabling memory overcommit
- If cgroups are used by the cloud provider, proper out-of-memory behaviour is difficult to achieve

- Same for Single Server and Flexible Server

```
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".
```

```
2021-03-19 10:10:04 UTC ERROR:  out of memory
```

```
2021-03-19 10:10:04 UTC DETAIL:  Failed on request of size 536.
```

```
2021-03-19 10:10:04 UTC STATEMENT:  EXPLAIN ANALYZE SELECT * FROM pg_class
AS a, pg_class AS b, pg_class AS c ORDER BY random();
```

```
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.
!?!>
```

```
2021-03-19 09:32:55 UTC LOG: server process (PID 542) was terminated by
signal 9: Killed
```

```
2021-03-19 09:32:55 UTC LOG: terminating any other active server processes
```

```
2021-03-19 09:32:55 UTC LOG: all server processes terminated; reinitializing
```

```
2021-03-19 09:32:55 UTC FATAL: the database system is in recovery mode
```

```
postgres=> SELECT * FROM pg_class AS a, pg_class AS b, pg_class AS c
postgres-> ORDER BY random();
SSL SYSCALL error: EOF detected
The connection to the server was lost. Attempting reset: Succeeded.
psql (13.2 (Debian 13.2-1.pgdg100+1), server 12.5)
postgres=>
```

```
2021-03-19 09:52:47 UTC LOG: server process (PID 11509) was terminated by
        signal 9: Killed
2021-03-19 09:52:47 UTC LOG: terminating any other active server processes
2021-03-19 09:52:47 UTC FATAL: the database system is in recovery mode
```

- Questions?
- Michael Banck <michael.banck@credativ.de>
- <https://www.credativ.com>
- <https://www.credativ.com/portfolio/support/postgresql-competence-center>
- <https://www.credativ.com/career>
- <https://www.credativ.com/blog>
- <https://github.com/credativ>