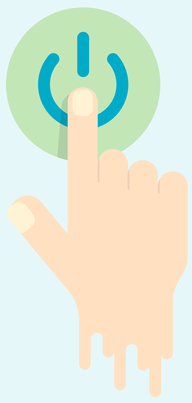
Top 10 metrics in PostgreSQL monitoring with Prometheus

Learn the top 10 metrics in PostgreSQL monitoring, with alert examples, both for PostgreSQL instances in Kubernetes and AWS RDS! Read the full guide in https://sysdig.com/blog/postgresql-monitoring/

1



AVAILABILITY

Server is up

Checking that your instance is up and running should be the first step in PostgreSQL monitoring. The exporter will monitor the connection and availability of the PostgreSQL instance.

```
pg_up == 0
```

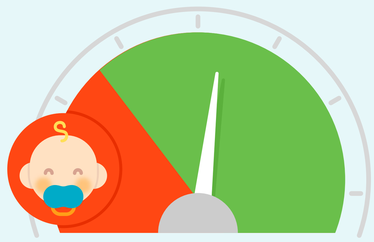
2

AVAILABILITY

Postmaster Service Uptime

The minimum postmaster service uptime should reflect the last known controlled server restart. Otherwise, a server may have been restarted for unknown reasons.

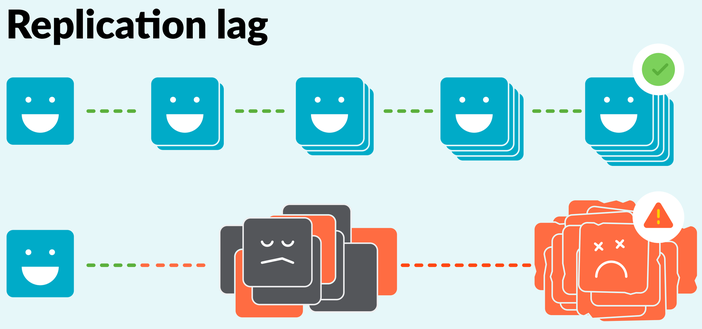
```
time() - pg_postmaster_start_time_seconds < 3600
```



3

REPLICATION

Replication lag



A high replication lag rate can lead to coherence problems if the master goes down.

```
pg_replication_lag > 10
```

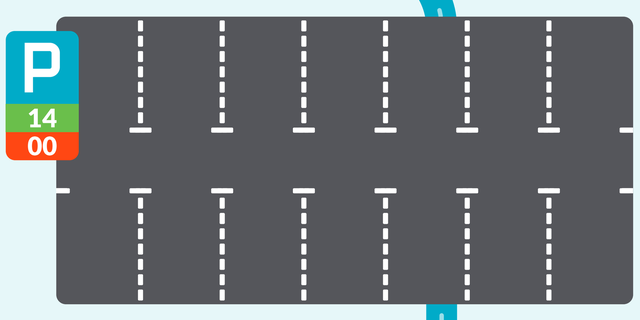
4

STORAGE

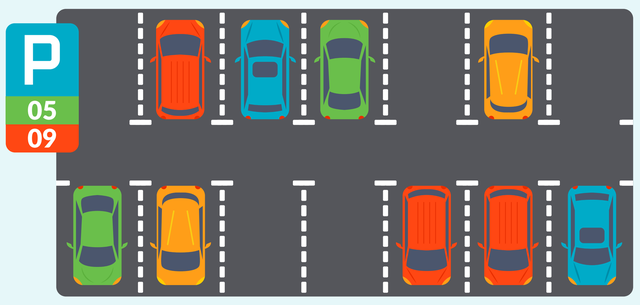
Database size

Running out of disk is a common problem in all databases. Let's figure out what is the storage usage of each of the PostgreSQL databases in our instance.

```
pg_database_size_bytes
```



5



STORAGE

Available storage

Check also the available disk in your instance.

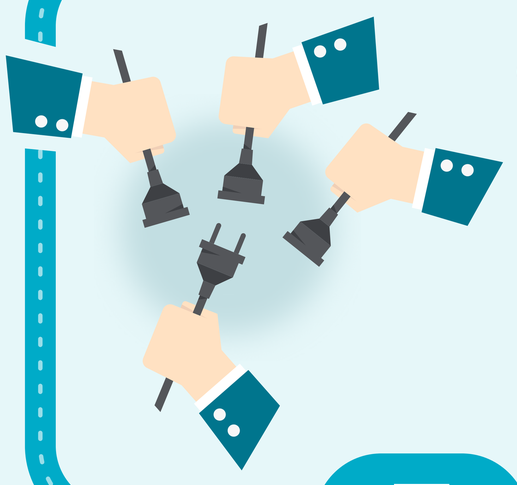
KUBERNETES

```
predict_linear(node_filesystem_free_bytes[1w],
3600 * 24) / (1024 * 1024 * 1024) < 1
```

AWS RDS POSTGRESQL

```
predict_linear(aws_rds_free_storage_space_
average[48h], 48 * 3600) < 0
```

6



NETWORKING

Number of available connections

A common problem in databases is running out of network connections.

```
((sum(pg_settings_max_connections) by (server) -
sum(pg_settings_superuser_reserved_connections)
by (server)) - sum(pg_stat_activity_count) by
(server)) / sum(pg_settings_max_connections) by
(server)) * 100 < 10
```

7

PERFORMANCE

Latency

Latency is a key indicator of your database performance. You can measure how long it takes to get the result of the slowest active transaction.

```
pg_stat_activity_max_tx_duration{state="active"} > 2
```

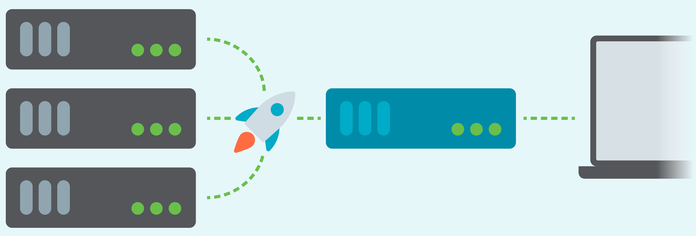
8

PERFORMANCE

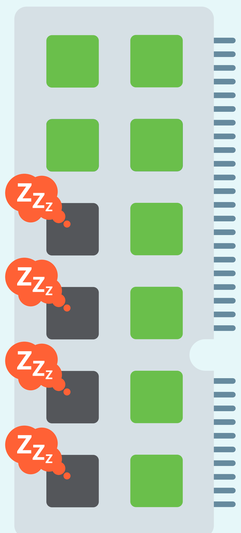
Cache hit rate

High latency can be a consequence of problems with cache in memory, which increases disk usage, so everything is slower.

```
100 * (rate(pg_stat_database_blks_hit[$__interval]) /
((rate(pg_stat_database_blks_hit[$__interval]) +
rate(pg_stat_database_blks_read[$__interval]))>0)) < 80
```



9



PERFORMANCE

Memory available

The solution for a low hit rate is increasing the memory usage of your instance. Check if there's enough memory available.

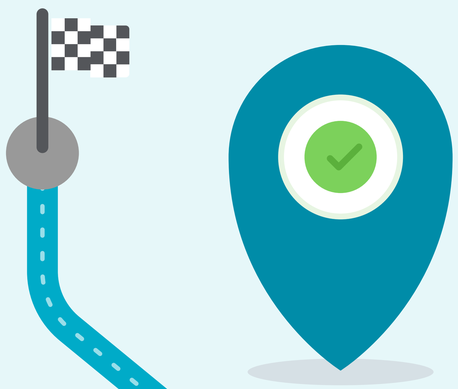
KUBERNETES

```
sum
by(namespace,pod,container)(kube_pod_containe
r_resource_limits{resource="memory"}) - sum
by(namespace,pod,container)(container_memory_
usage_bytes{container!="POD",container!=""})
```

AWS RDS POSTGRESQL

```
aws_rds_freeable_memory_average
```

10



PERFORMANCE

Requested buffer checkpoints

PostgreSQL uses the buffer checkpoints to write the dirty buffers on the disk. A high number of requested checkpoints compared to the number of scheduled checkpoints can directly impact performance.

```
rate(pg_stat_bgwriter_checkpoints_req[5m]) /
(rate(pg_stat_bgwriter_checkpoints_req[5m]) +
rate(pg_stat_bgwriter_checkpoints_timed[5m])) * 100
```

sysdig