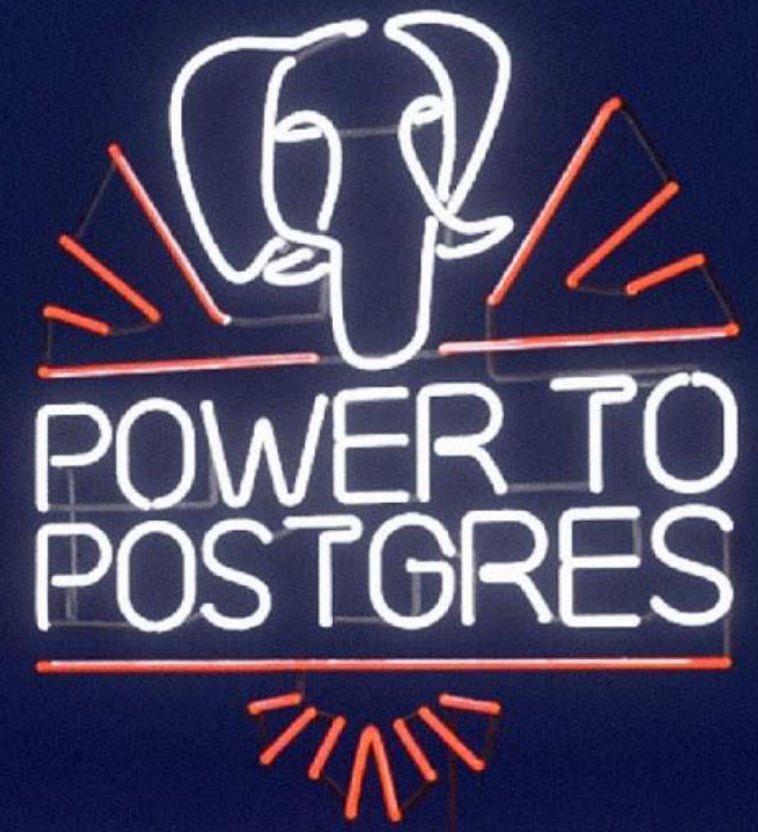
# Monitoring Best Practices




Kanchan Mohitey & Ajay Patel

February 2022

EDB

# Introductions

Know your Speakers

## Kanchan Mohitey

Sr Director, Managed Support Services  EDB

Accomplished and Result-driven Senior professional with more than 15 years of experience in Database domain.

Diverse experience across Product Engineering, Project Management and Quality assurance and Support services and Remote DBA teams

## Ajay Patel

Sr Manager, Technical Support Services  EDB

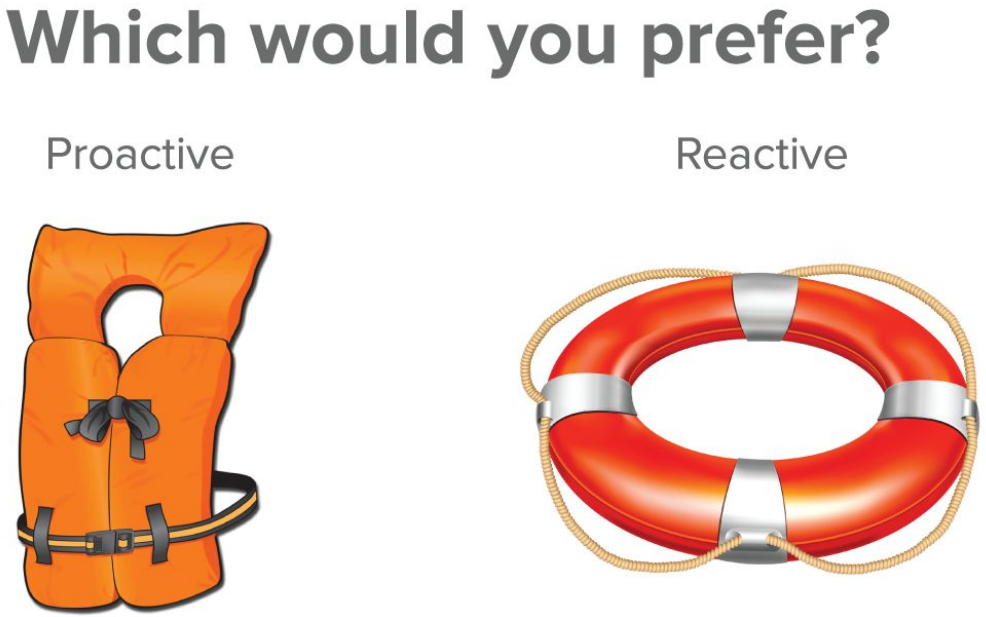
PostgreSQL evangelist and leader of North America support.

Helping customers to be successful through technology and by solving their challenging problems.

# Agenda

- Why to monitor Postgres?
- Monitoring approaches
- Top monitoring points
- When not to monitor?
- Comparison matrix(Tools)
- RemoteDBA Services

# Monitoring Approach






EDB POSTGRES

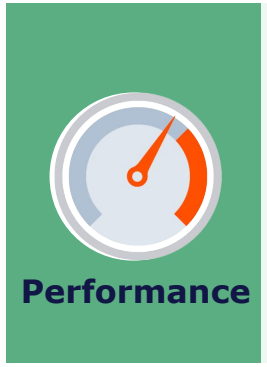
# Why monitor my Postgres Database ?

## Improve Business Process


Availability

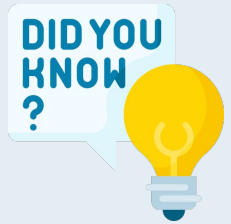

Capacity Planning


System optimization



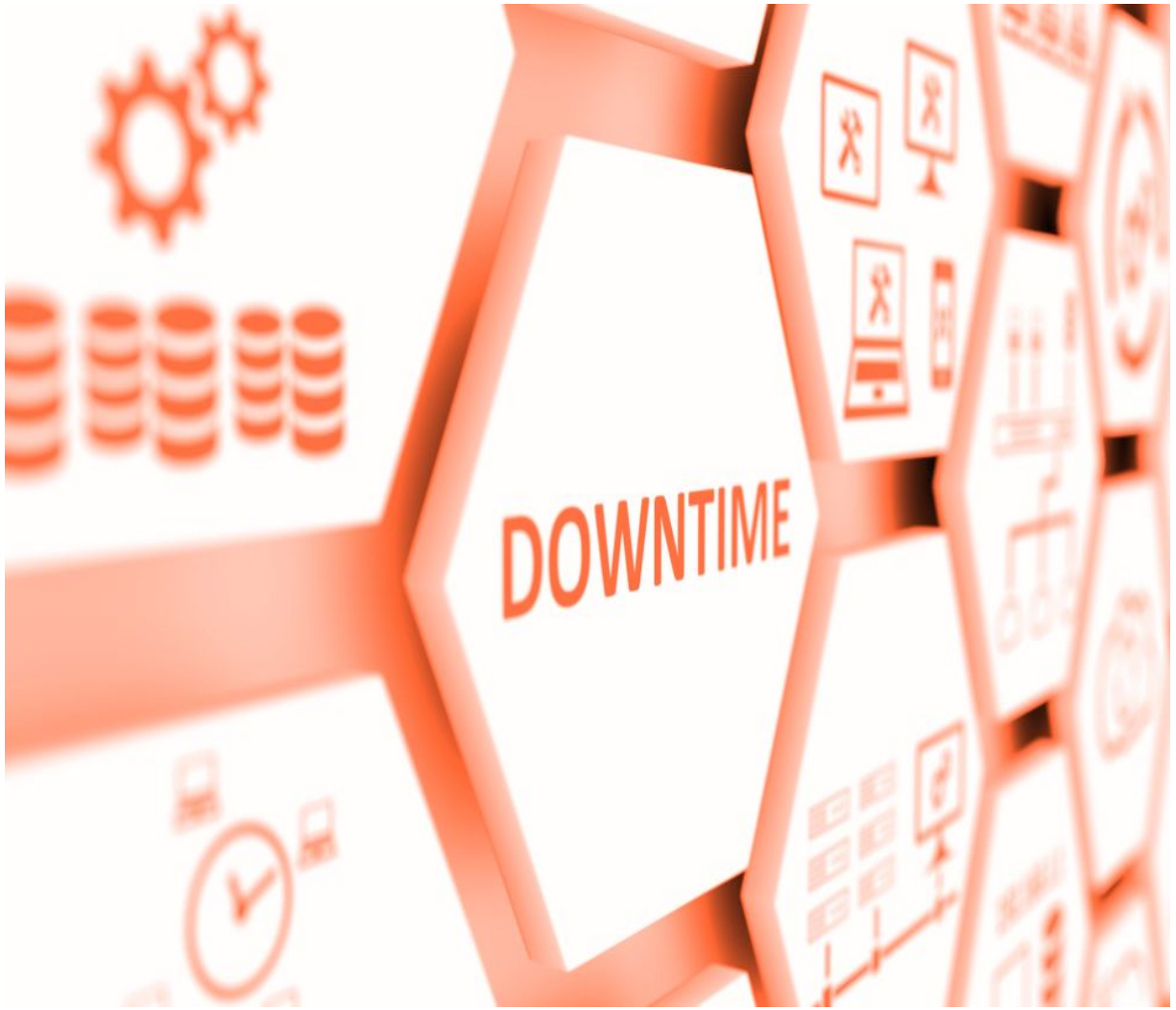

Performance Problem

# Right monitoring strategy



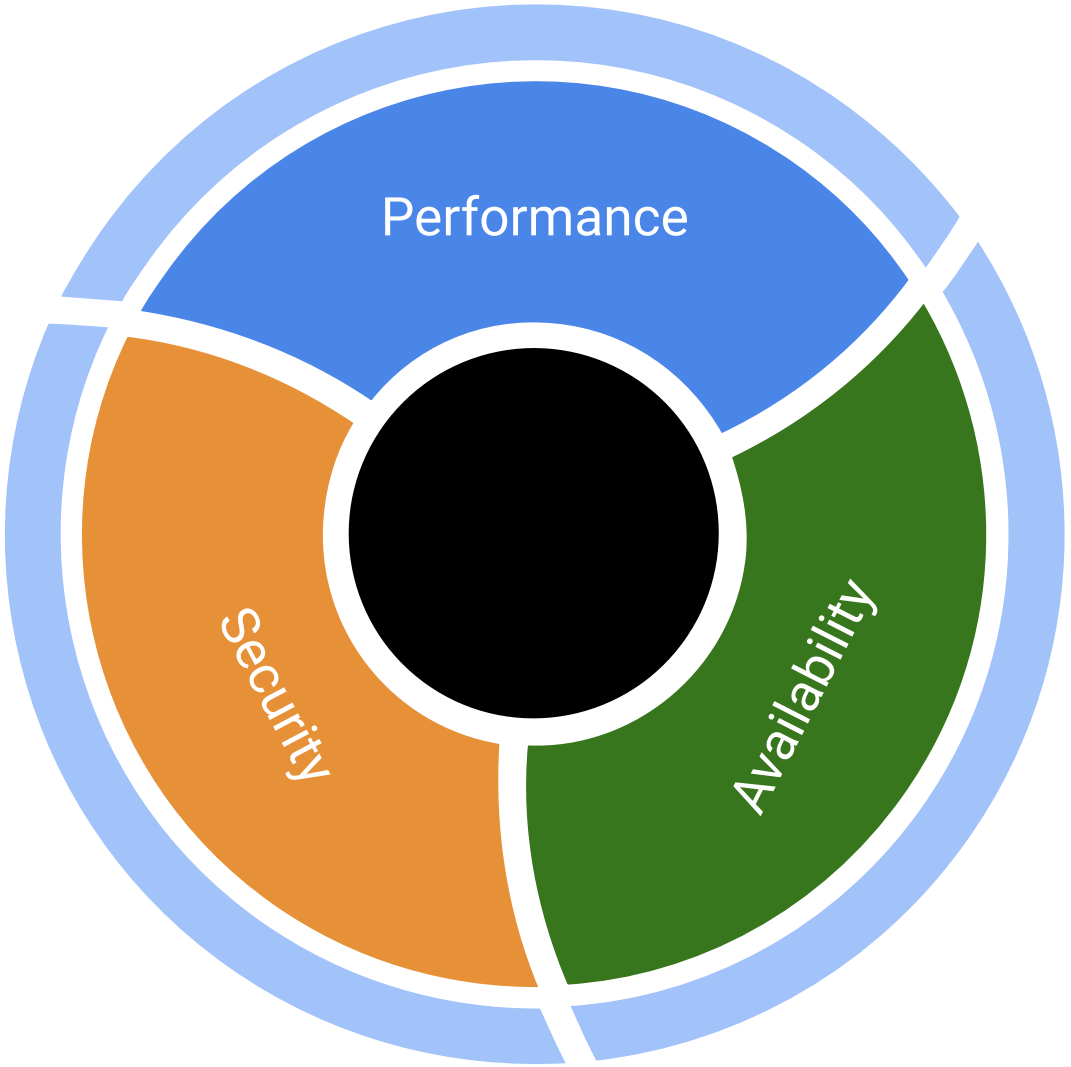

AWS Q4 Outage - ~12 hrs
Affected - Multiple services
Impact - Global

## Avoid unplanned downtime




- Reduce troubleshooting efforts
- Improve Mean Time to Resolve (MTTR)
- Identify trends and Intelligence
- Significant Improvement on System availability
- Maximise Database Uptime
- Improve end user experience

# Aspects of Monitoring




What?

Why?

How?



EDB POSTGRES

# Database availability

Why?

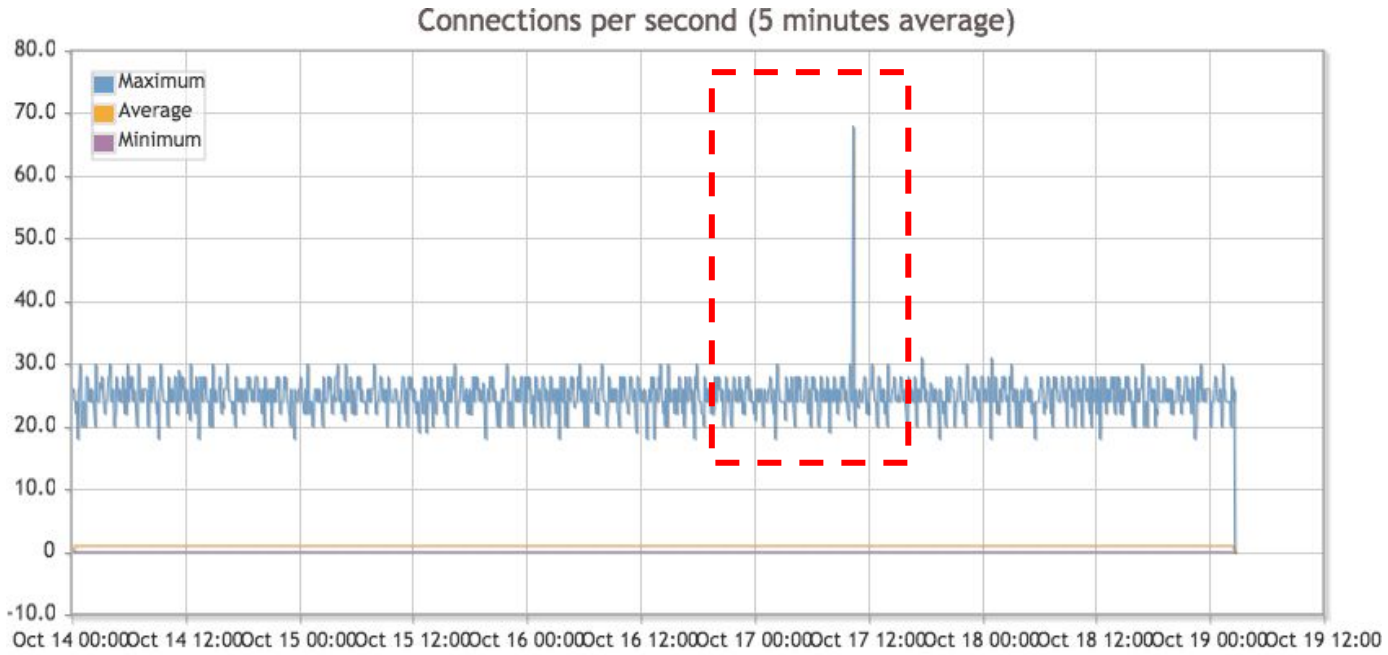
- Database status
- Database is up but not accepting connections

How?

- Monitoring postgres services
- Do a ‘select(1) from pg_stat_activity’  from non-super users

Availability
&
Performance

# Database connection (active/Idle)




Why?

- Identify application usage pattern
- Capacity planning
- Idle connections

How?

- Query Pg_stat_activity for idle/active connections
- **PEM** performance diagnostic(real time)
- PgBadger(after the fact)



EDB POSTGRES

# Transaction wraparound

Why?

- XID limit reaches : Database stop accepting the connection
- 4 Billion is the limit

How?

- Query pg_database and compare it with ‘autovacuum_freeze_max_age’

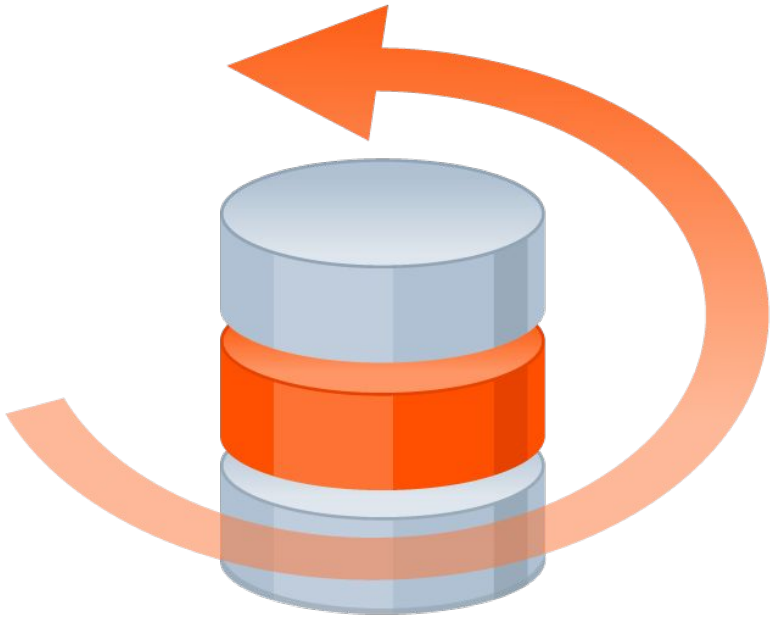
# Database restart/reload

Why?

- Ad-hoc changes on database

How?

- Database  logs
- pg_ctl from system logs.



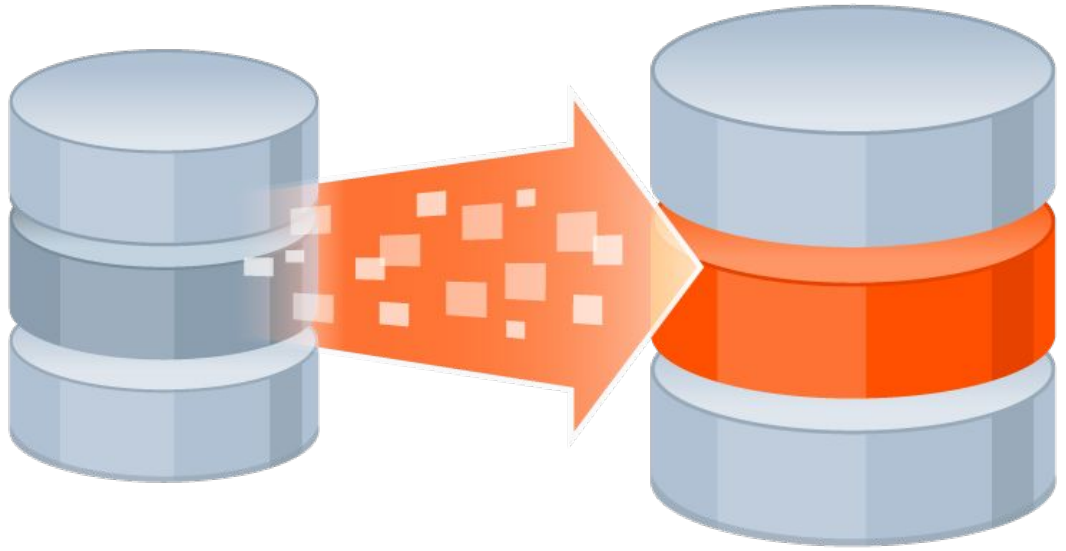
# WAL's ready to be archived

Why?

- Identify reason for WAL generation.
- Avoid Database crash

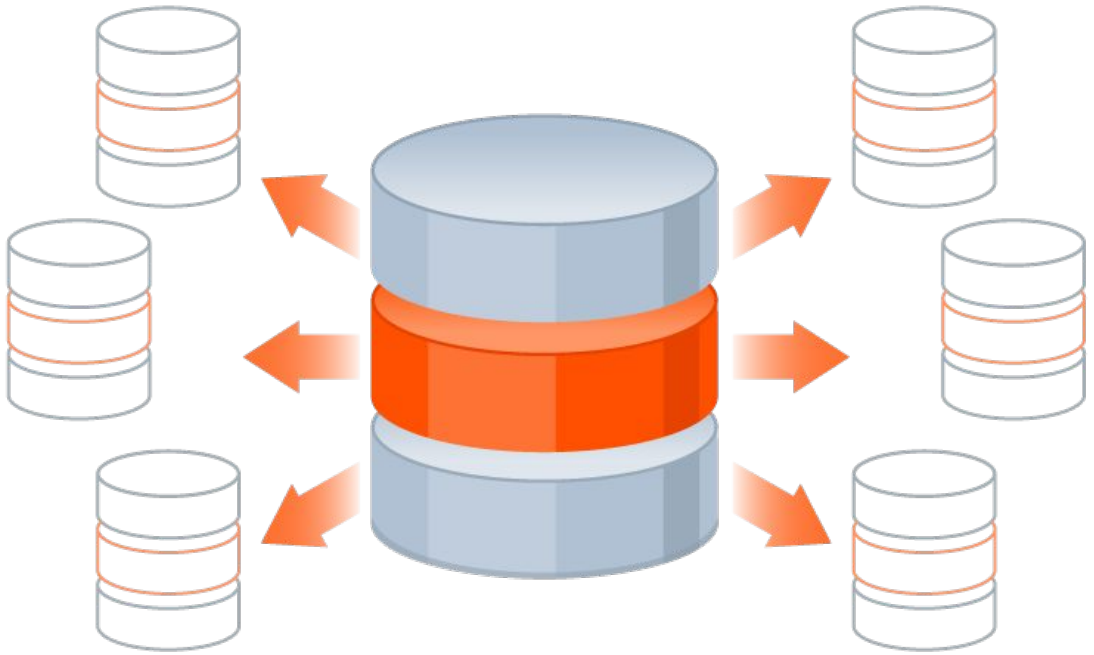
How?

- Count the files under WAL location.



# Database backup

Over **50%** DBA's do not have a backup strategy



## Why?

- Backup is successful
- Not missing any backups
- Logical backup of tables(Very large databases)

## How?

- Make sure script/tool which uses Pg_Basebackup/pg_dump has notification mechanism

# Streaming replication

Why?

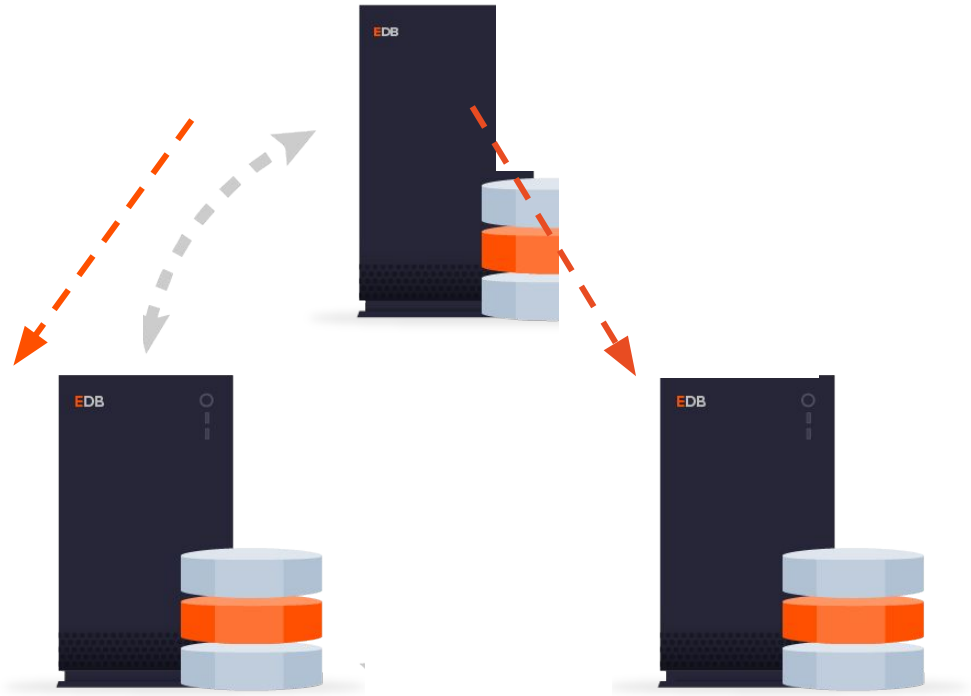
- To meet RPO (Failover)
- Network issue

How?

- Pg_last_wal_receive_lsn &

  pg_last_wal_replay_lsn

- pg_stat_replication(Postgres10 onwards)






EDB POSTGRES

# Database Error

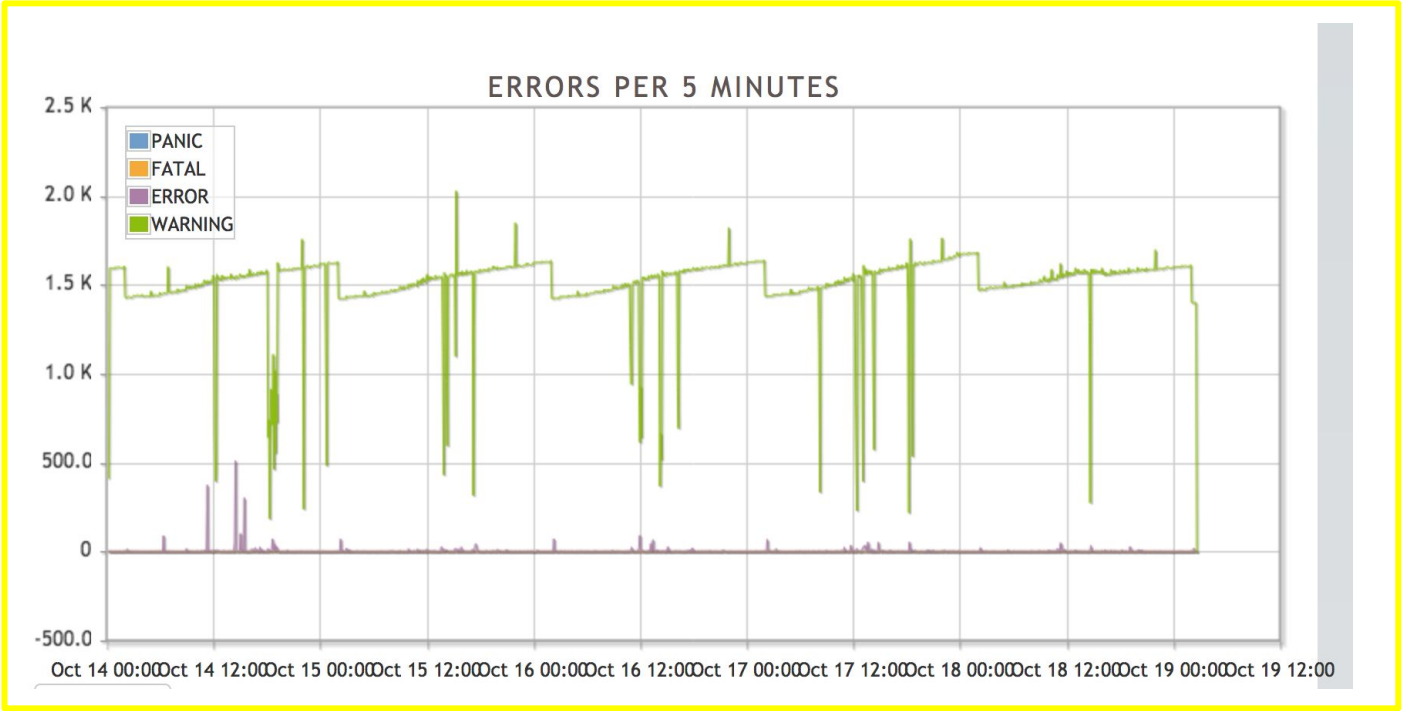
```
2019-10-14 11:40:27 EDT [78568]: [21-1] user=postgres,db=prod1,app=prod_web,client=127.0.0.1 ERROR:  out of memory
2019-10-14 11:40:45 EDT [203624]: [3-1] user=postgres,db=prod1,app=prod_web,client=127.0.0.1 ERROR:  out of memory
2019-10-14 11:40:45 EDT [123464]: [3-1] user=postgres,db=prod1,app=prod_web,client=127.0.0.1 ERROR:  out of memory
2019-10-14 11:41:25 EDT [105036]: [3-1] user=postgres,db=prod1,app=prod_web,client=127.0.0.1 ERROR:  out of memory
2019-10-14 11:41:35 EDT [189712]: [3-1] user=postgres,db=prod1,app=prod_web,client=127.0.0.1 ERROR:  out of memory
2019-10-14 11:41:36 EDT [190400]: [3-1] user=postgres,db=prod1,app=prod_web,client=127.0.0.1 ERROR:  out of memory
```

## Why?

- Capture ERROR, FATAL, PANIC situations
- Data corruption
- Database not able to start up

## How?

- Postgres logs
- PgBadger






EDB POSTGRES

# Long running queries

Time consuming queries

| Rank | Total duration | Times executed | Min duration | Max duration | Avg duration | Query |
|---|---|---|---|---|---|---|
| 1 | 1h1m8s | 156 Details | 10s66ms | 57s559ms | 23s514ms | SELECT lock_name FROM semaphore_locks WHERE lock_name = ? FOR UPDATE; Examples User(s) involved |

## Why?

- Application queries are not running within defined limit
- Change into data/plan or something else

## How?

- Periodically  query Pg_stat_activity
- Postgres logs with PgBadger report
- Use **PEM**

# Locking (conflicts/deadlocks)

Why?

- Application conflicts
- Deadlocks

How?

- Periodically query pg_locks
- Postgres logs with PgBadger
- Use PEM to monitor real time monitoring

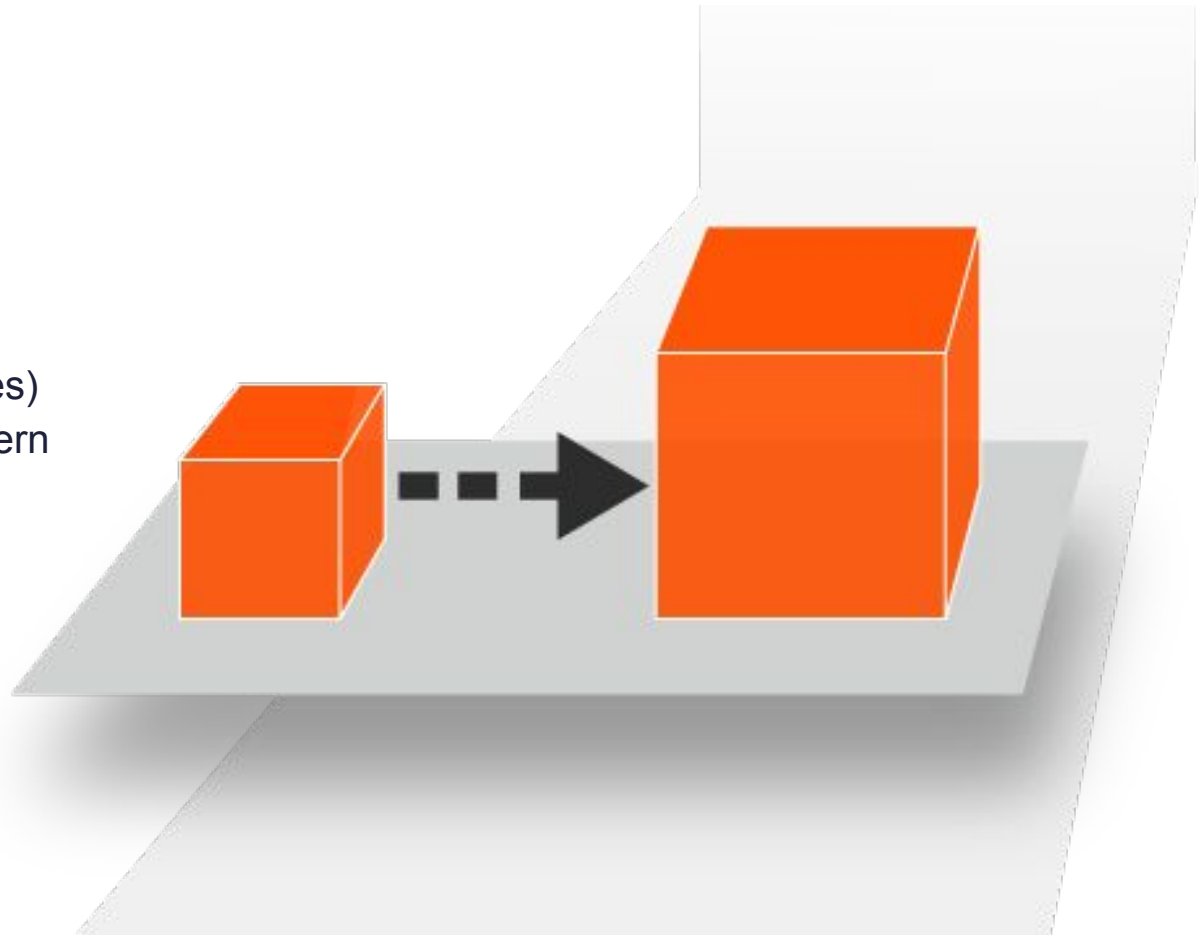
# Object size

Why?

- Identify largest object(table/indexes)
- Identify change in application pattern
- Database redesign

How?

- pg_total_stat_relation(relid)
- pg_relation_size(relid)



# Bloating

Why?

- Performance issue(slower reads and writes)
- To control wasted space
- Need for manual vacuum

How?

- Check_postgres.pl
- Use pg_stats,pg_index,pg_class
- PEM

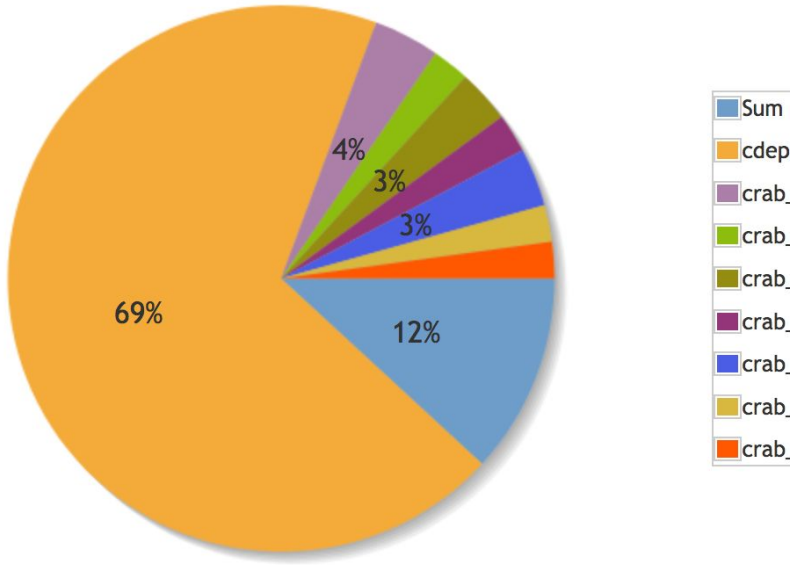
# Table/index stats

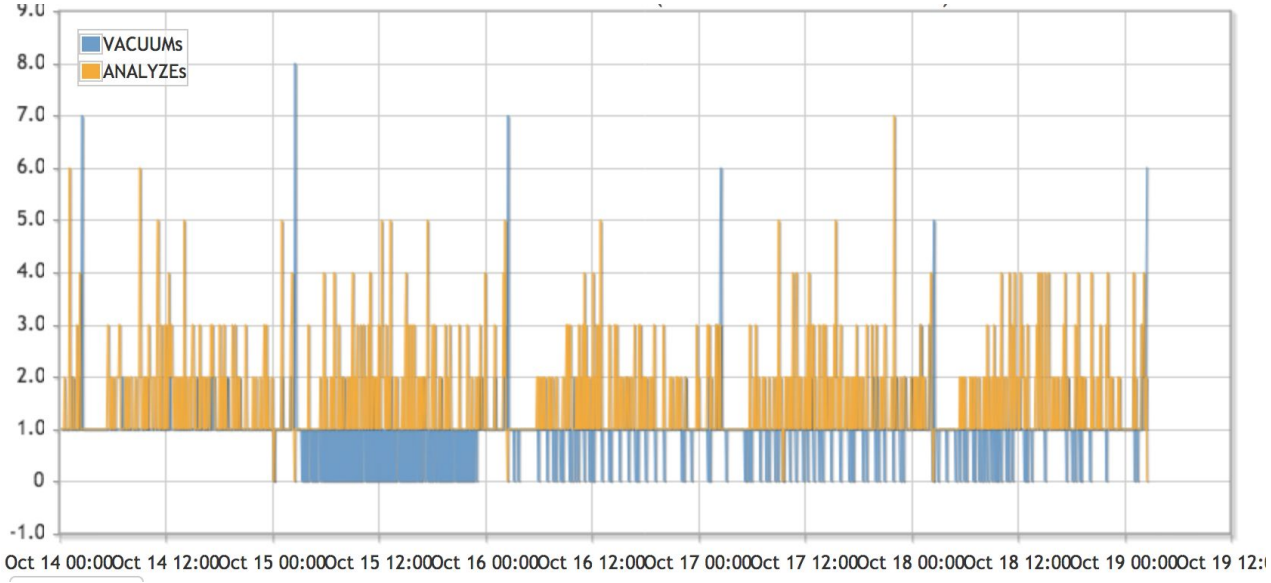
## Why?

- New index requirement
- Autovacuum aggressiveness
- Unused index
- Index usage changed

## How?

- Use pgbadger report
- Pg_stat_user_tables for table
- Pg_stat_user_indexes for index






EDB POSTGRES

# Monitoring points (host)

- Load average
- Disk IO
- CPU
- Memory
- Disk space
- **Inode**

# When not to monitor?

- Don’t monitor dev machines like production
- Don’t monitor DR/read-replica machines like production
- Planned maintenance
- Define frequency (Key element)
  - Don’t need bloat information daily
  - Don’t need vacuum information daily

# Takeaway

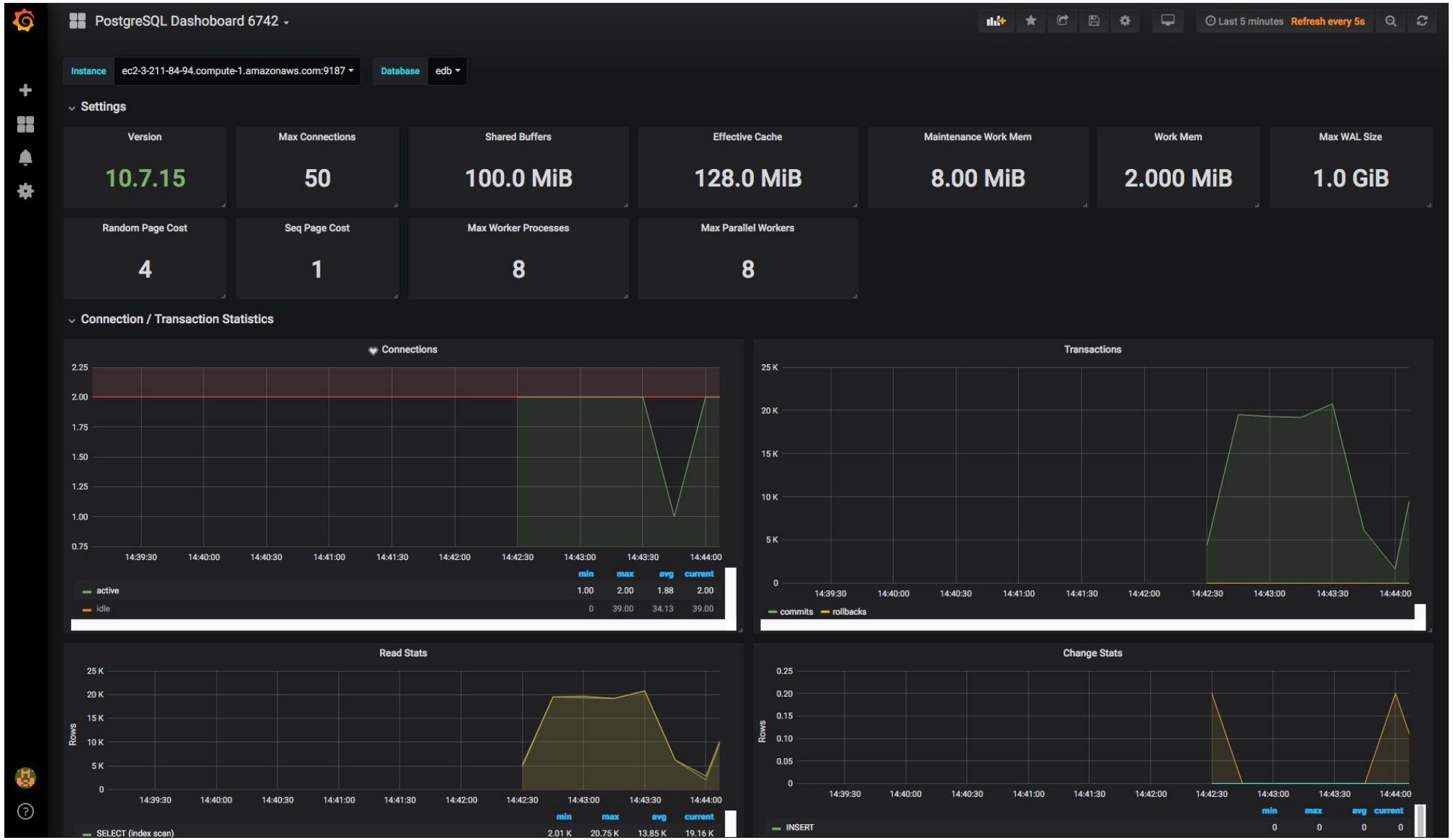
| Aspects | Database | Host |
|---|---|---|
| Availability | Availability | Disk space |
| | Connections | Inode |
| | Transaction wraparound | |
| | Restart/reload | |
| | WAL ready to archive | |
| | Backup | |
| | Streaming replication | |
| | Database error | |
| Performance | Long running query | Load average |
| | Locking | Disk IO |
| | Object size | CPU |
| | Bloating | Memory |
| | Table/Index stats | |
| Security | NA | NA |

EDB POSTGRES

# Tools for monitoring

- Prometheus + Grafana
- Nagios Core + check_postgres.pl
- Zabbix
- Scripts(shell/python)
- PgBadger
- Postgres Enterprise Manager(Enterprise)

# Prometheus + Grafana



# Nagios-core

Nagios®

**General**

- Home
- Documentation

**Current Status**

- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

**Reports**

- Availability
- Trends (Legacy)
- Alerts

**Current Network Status**
Last Updated: Fri Nov 22 18:58:31 UTC 2019
Updated every 90 seconds
Nagios® Core™ 4.4.2 - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

**Host Status Totals**

| Up | Down | Unreachable | Pending |
|---|---|---|---|
| 2 | 1 | 0 | 0 |

| All Problems | All Types |
|---|---|
| 1 | 3 |

**Service Status Totals**

| Ok | Warning | Unknown | Critical | Pending |
|---|---|---|---|---|
| 22 | 1 | 6 | 2 | 0 |

| All Problems | All Types |
|---|---|
| 9 | 31 |

**Service Status Details For All Hosts**

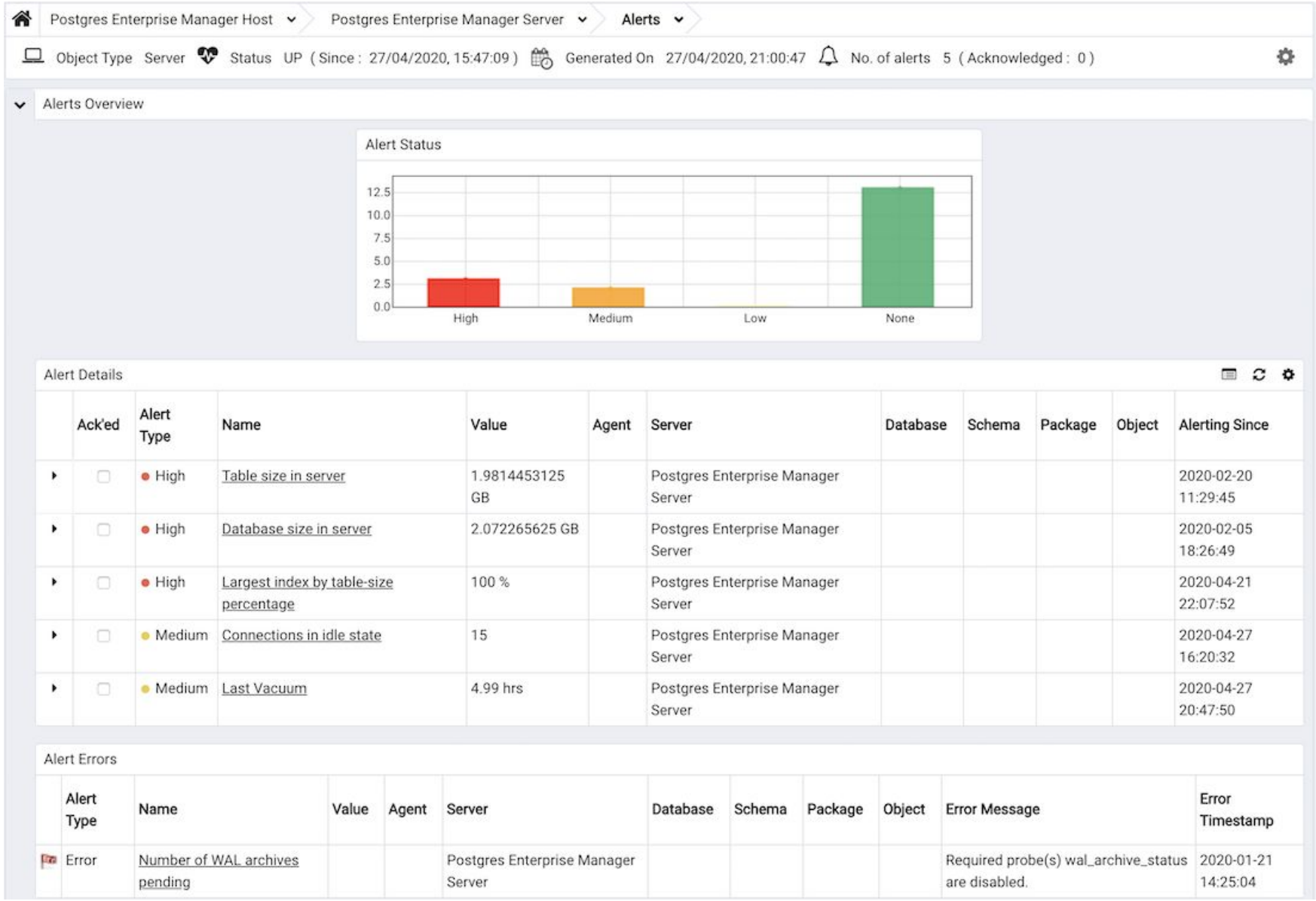
Limit Results: 100

| Host | Service | Status | Last Check | Duration | Attempt | Status Information |
|---|---|---|---|---|---|---|
| localhost | Current Load | OK | 11-22-2019 18:55:03 | 7d 22h 14m 32s | 1/4 | OK - load average: 0.00, 0.01, 0.05 |
| | Current Users | OK | 11-22-2019 18:56:42 | 7d 22h 13m 54s | 1/4 | USERS OK - 0 users currently logged in |
| | HTTP | WARNING | 11-22-2019 18:58:14 | 4d 0h 45m 17s | 4/4 | HTTP WARNING: HTTP/1.1 403 Forbidden - 5179 bytes in 0.001 second response time |
| | PING | OK | 11-22-2019 18:56:52 | 7d 22h 12m 39s | 1/4 | PING OK - Packet loss = 0%, RTA = 0.04 ms |
| | Root Partition | OK | 11-22-2019 18:56:30 | 7d 22h 12m 2s | 1/4 | DISK OK - free space: / 18299 MB (89.40% inode=99%): |
| | SSH | OK | 11-22-2019 18:57:08 | 7d 22h 11m 24s | 1/4 | SSH OK - OpenSSH_7.4 (protocol 2.0) |
| | Swap Usage | CRITICAL | 11-22-2019 18:56:42 | 7d 22h 10m 47s | 4/4 | SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size. |
| | Total Processes | OK | 11-22-2019 18:58:24 | 7d 22h 10m 9s | 1/4 | PROCS OK: 43 processes with STATE = RSZDT |
| postgres1 | Current Load | UNKNOWN | 11-22-2019 18:52:17 | 3d 13h 46m 14s | 3/3 | check_load: Could not parse arguments |
| | Current Users | UNKNOWN | 11-22-2019 18:53:16 | 3d 13h 45m 15s | 3/3 | (No output returned from plugin) |
| | PING | OK | 11-22-2019 18:58:20 | 1d 17h 40m 11s | 1/3 | PING OK - Packet loss = 0%, RTA = 0.57 ms |
| | Root Partition | UNKNOWN | 11-22-2019 18:51:13 | 3d 13h 47m 18s | 3/3 | check_disk: Could not parse arguments |
| | SSH | OK | 11-22-2019 18:58:13 | 7d 0h 0m 19s | 1/3 | SSH OK - OpenSSH_7.4 (protocol 2.0) |
| | Total Processes | UNKNOWN | 11-22-2019 18:57:13 | 3d 13h 41m 19s | 3/3 | NRPE: Command 'check_total_procs' not defined |
| | Total Processes zombie | UNKNOWN | 11-22-2019 18:58:09 | 3d 13h 40m 22s | 3/3 | NRPE: Command 'check_zombie_procs' not defined |

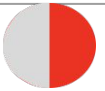


EDB POSTGRES

# Postgres Enterprise Manager

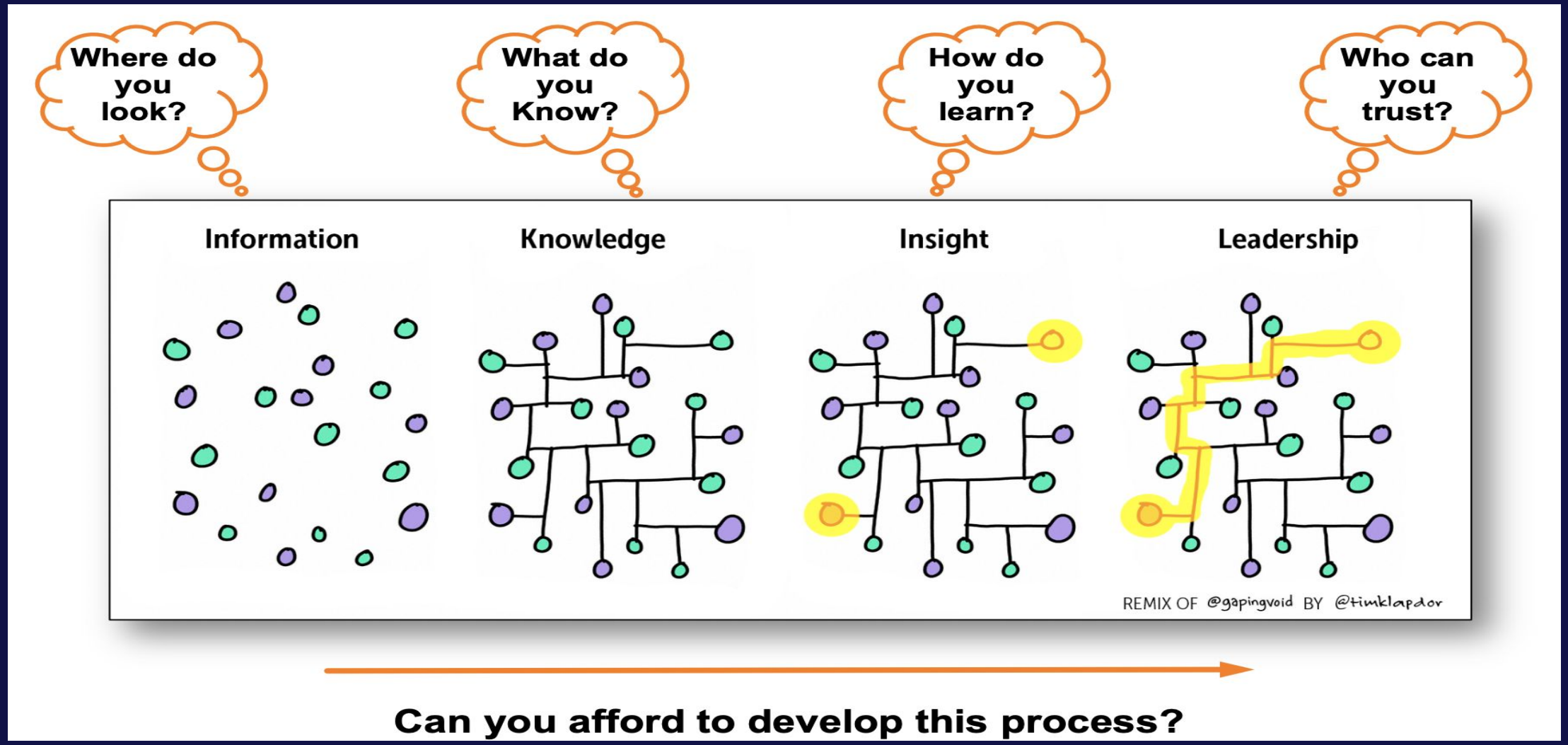


# Comparison matrix

| Type | Prometheus / Grafana | Nagios Core | Zabbix | Postgres Enterprise Manager(PEM) |
|---|---|---|---|---|
| Database(Connections, Queries, Stats, replication) |  |  |  |  |
| Backup |  |  |  |  |
| Advisory/Tuning |  |  |  |  |
| Capacity Planning |  |  |  |  |
| Error Reporting |  |  |  |  |
| Server(CPU,Disk & I,Memory,Network) |  |  |  |  |
| Notification |  |  |  |  |
| Custom Probes |  |  |  |  |



EDB POSTGRES


Where do you look?
What do you Know?
How do you learn?
Who can you trust?
Information
Knowledge
Insight
Leadership
REMIX OF @gapingvoid BY @timklapdor
Can you afford to develop this process?

# Remote DBA Service


**Around-the-clock Assurance**
On-premises and in the cloud

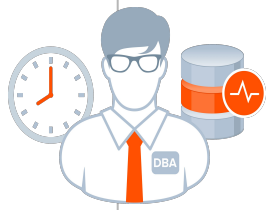
**Premium Management**
More than Monitoring Alone

**Timely, Affordable, and Reliable**
Always at Your Service

# Remote DBA Ongoing Operational tasks

# Remote DBA Offering

| | RemoteDBA | Cloud DBA |
|---|---|---|
| 24x7 Monitoring | ✅ | ✅ |
| 24x7 DBA Operations | ✅ | ✅ |
| Database & Tools Upgrades | ✅ | ✅ |
| Query Tuning Advice | ✅ | ✅ |
| Proactive Health Scan | ✅ | ✅ |
| Monthly Trend Review | ✅ | ✅ |
| Quarterly Review | ✅ | ✅ |
| Account Technical Lead | ✅ | ✅ |
| Supported Platforms | OpenShift, RHEL, CentOS, Debian, Microsoft Windows | • AWS EC2<br>• Azure VM<br>• GCP |



# Thank you