Glossary

Colored Contents	Represents
BLACK	Normal DB Activities (Managed Services)
RED	New Implementations (Professional Services)

Support Type	Related Tasks
Full Support	Covers all **BLACK** + **RED** Colored tasks mentioned in this document. **NOTE**: All **RED** colored contents will be executed in Serial manner (one task at a time). If any multiple **RED** colored tasks needed to be executed in parallel will be carried over with additional costs based on the workload proposed.
Normal Support	Covers ONLY BLACK colored tasks mentioned in this document

---

Equivalent Work Experience

Tasks Level	
Level-0	Freshers
Level-1	1-2 years exp
Level-2	3-5 years exp
Level-3	5+ years exp
Level-4	8+ Technology & Leadership exp
ITIL	Process Expertise

PostgreSQL Database Project & Support Services

Database Health Check (level-1)

a) DB Services availability status check
b) DB error check: Postgres error log/Daily log analysis using pgbadger tool
c) Backup status Check (pg_basebackup/pg_dump/Barman/Pg_backrest)
d) DR synchronization check
e) Database Space check
f) OS Space check
g) Performance check
- Invalid objects
- Table locks
- Invalid sessions
- Postmaster health check status

Database Design, Installation & Configuration. (Level-3)

a. Design OS requirements: Choosing & configuring the appropriate CPU, Memory, storage, and ancillary software
b. Design Database requirements: Creating scalable database architectures that allow for expansion
c. Conducting design reviews
d. Designing a database for high-speed, high-volume transactions
e. Installation and configuration of PostgreSQL database from source or Packages on all types of Operating Systems.
f. Installation and configuration of PostgreSQL-related tools like pgBouncer, pgpool, etc.
g. Preparing PostgreSQL documentation
h. Installation & configuration of all types Database client side & server tools for both windows & Linux.

Backup & Disaster Recovery Implementation & Management (Level-2)

a) Configure new backups

b) Maintain and manage the full and incremental backup of databases using:

- pg_backrest (for large size of databases)
- SQL Dump
- File system backup

c) Re-Run failed backups

d) Ensure all backups (data & WAL logs) are stored in a safer remote location for restore consistency.

e) Table refresh, schema refresh, full db refresh using pg_dump,pg_restore.

f) Continuous Archiving and Point-in-Time Recovery (PITR)

g) Restoring user objects in case of any human error on end-user data management.

	Open-Source	Incremental	Differential	Backup from Slave	Compression	Cloud Support	Stream_to_Cloud	Parallel	Retention	Encryption
pg_basebackup	☑	☐	☐	☑	☑	☐	☐	☐	☐	☐
pgBackrest	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
BARMAN	☑	☑	☐	☑	☑	☐	☐	☑	☑	☐
WAL-g	☑	☐	☐	☑	☑	☑	☑	☑	☑	☑

Standby Database Management (Database Replication) (Level-2 & 3)

a) Architecting & Building a new standby database.

b) Warm Standby: A standby that cannot be connected to until it is promoted to a master server.

c) Hot Standby: A standby that can accept connections and serve read-only queries.

d) Failover: Switching to standby after an abnormal termination of the master server.

e) Logical replication: Table-level replication to the same version as well as a different version of the Postgresql database.

f) Replication and Failover management for Postgres using Repmgr tool

g) Re-sync Primary & Standby, if any gaps

- File or disk-based
- Log shipping-based (WAL)
- SQL based

Database User and Security Management (Level-1 & 2)

a) User creation with proper authentication(credentials) & authorization (Privleges).

b) Restricted privileges/authentication using pg_hba.conf (mini firewall for your DB)

c) RLS – Row Level Security Policies

d) IP TABLES

e) postgressql.conf

f) Data Encryption
   - Whole-disk encryption
   - Per Column encryption
   - Pgcrypto: query level (unencrypted data and key can appear in the log)
   - SSL Configuration

g) Restrict Access
   - Database to group
   - SCHEMA to group
   - TABLE to group
   - COLUMN to group

Database Capacity Planning & Space Management (Level-3)

a. Designing database storage requirements by considering immediate future growth.

b. Performing periodic capacity reviews (daily, Weekly, Monthly, Quarterly) based on customer requirements & send database growth reports regularly.

c. Add additional space by taking approval from management.

d. Pro-active housekeeping activities to clear old WAL logs, Backup files, Backup logs & database logs from the OS drive, as per customer standard to avoid last-minute space hiccups.

e. Archiving the wal log files to new destinations and maintaining the space constraints.

Database Performance Tuning (level 2 & 3)

a) Frequent monitoring of running queries

b) Max connections monitoring and clean up idle connections.

c) Finding slow, long-running & blocking queries, and reporting to the application team.

d) Index validity check

e) Finding blocking session/wait events

f) Checking Bloat Tables, and Dead Tuples and analyzing based on performance.

g) Maintenance & removal of bloat with pg_repack and vacuum.

h) Log analysis using the PGbadger tool.

i) EXPLAIN PLAN – Const analysis

j) Extended Statistics

k) checkpoint_segments temporarily during bulk data loads

l) Query re-write

m) Implementing Table level partitioning for better performance.

n) Tune parameters like:
- wal segments,
- checkpoints
- shared_buffer
- wal_buffer
- work_mem(session level)
- maintenance_work_mem
- synchronous_commit
- pg_hint_plan

o) Postgres OS level tuning:
- hugepages
- kernel parameters
- vm.swappiness
- vm.overcommit_memory
- vm.overcommit_ratio
- vm.dirty_backgroud_ratio
- vm.dirty_background_bytes
- vm.dirty_ratio
- vm.dirty_bytes

Database Patch & Upgrade (Level-2)

a) Upgrade Postgres to any higher version.

b) dbpatch - PostgreSQL database patch change management extension. This extension supports conducting database changes and deploying them in a robust and automated way.

Upgrade method	Pro	Cons
Dump/restore	• Simple • Safe • Somewhat flexible	• Slowest method • Per database approach has some pitfalls lurking
Binary in-place	• Fast / very fast (depending on chosen mode) • Old instances not affected in default mode	• More complex than Dump / Restore • Somewhat risky in “link” mode • Possibly loses standby servers • Double the disk space required in default mode
Logical Replication	• Shortest possible downtime • Safe, with the possibility of thorough “live tests” • Very flexible	• Most complex method • Possibly some schema changes needed • Not everything is transferred (sequence state, large objects) • Possibly “slowish” • Always per database

Database High-Availability Configuration & Management (Level-2 & 3)

- Design & Configuration of new HA setup and Modification in the existing setup.

a) EFM cluster management for primary and multiple standbys
b) PgCluster
c) Pgpool-II
d) RubyRep
e) Bucardo
f) Postgres-XC
g) Citus
h) Postgres-XL

- Maintainance of existing HA setup.

Database Refresh (Level-2)

a) Perform DB refresh using pg backrest restore Techniques.

b) Refresh data using the pgdump method

c) Refresh data from TAPE (storage) using pg backrest

d) Refresh Tables

Database Migration: (Level-3)

a) Postgresql database migration from one host to another host.

b) Cloud Migration: Migrate the Postgres database from on-premises to any cloud. Eg: AWS EC2 Postgres.

c) Cross Technologies Migration: Migration from other database Technologies to PostgreSQL database. Eg: Oracle to Postgresql, MariaDB to Postgresql, and vice-versa

d) Cross-Platform Migration: Migrate Postgres DB from one OS to another OS.
Eg: centos to Ubuntu to reduce the cost of a license for OS.

e) DBlink creation (FDW) from Oracle to Postgresql and PostgreSQL to MySQL, MySQL to Postgres.

f) Database Cloning

Vendor & Licensing Management (Level-2)

a) Utilizing our vendor relationships, we increase turn-around speed and get licensing information quickly.

b) We navigate the licensing maze, providing you with all the licensing options available.

c) We regularly work with software vendors to get the “best pricing” for our clients.

Database Continuous Monitoring (Level-1)

a) Implementation of Customised Monitoring using Shell & PowerShell.

b) Database Availability, Lister, Storage and Process, DB Performance.

c) Monitor the logs and raise on-call alerts to DBAs at the time of an incident.

d) Proactive checks based on the monitoring alerts help DBAs to keep the system healthy.

Database Process Implementation, Documentation & Orientation (ITIL)

- Introducing best-fit processes for database administration using ITIL global standards.
- Deriving & implementing new processes & periodically reviewing existing processes, in Incident Management, Problem Management, Change Management & Release management.
- Can Manage ITIL portfolio by supplying, Incident Manager, Problem Manager, Change Manager, Release Manager.
- Documenting the SOP (Standard Operating Procedures) for end-to-end database administration portfolio for your various environments Prod, QA & Development.
- Also we educate your IT / Non-IT staffs for the Process & SOP structured.

Database Status Reporting to IT Management (Level 1)

g) **Backup Status Report:** (Daily/Weekly/Monthly/Quarterly) backup status reports for end-to-end database environment.

h) **Daily Status Check Report:** Services Availability Status, Storage Status & Standby Sync status, HA Availability Status.

i) **Performance Status Report:** Periodic database Locking, Blocking, Inactive sessions status.

j) **Capacity Growth Report:** Periodic database storage growth (Daily/Weekly/Monthly/Quarterly) for Production databases.

k) Others: Other Database related report as the customers customized requirements.

Cloud ‘Database as Service’ support (Level-2 & 3)

End to End Database Consultation, Administration & Management in all Cloud Technologies (as well as in Hybrid Model)

We are expertise in supporting below cloud environments

Cloud We Support

Azure aws ORACLE CLOUD Google Cloud

SpeedCloud

Also we are in Technical Partnership with **SpeedCloud** cloud company and provide special discounted rates for our customers on our Techno-partner **SpeedCloud** cloud platform.

Our Database Support in cloud includes: -

- **Rehost patterns** (from on-perm database to self managed* in Amazon EC2, Azure VM, Compute Engine in GCC etc.)
- **Re-platform patterns** (from on-perm database to cloud DBaaS*)
- **Re-architect patterns** (from on-perm one database Technology to other open-source or cloud-native database Technologies)
- Setup monitoring of DBs using cloud native tools. (Eg: Cloud watch)
- Setup and manage DB in muti, hybrid cloud environment like AWS, AZURE & GCC along with On-Perm
- **Database Consulting in Cloud Infra**
  - Start with a comprehensive plan and a governance framework
  - Run the right database in the right cloud
  - Use data services that support multi-cloud environments
  - Exploit managed database services, or DBaaS
  - Consider database portability across multiple clouds
  - Optimize data access for applications and end users
  - Connect cloud networks to reduce data latency

- **Cloud Security**

    - Define standards, security, and compliance policies. Cloud database vendors rarely enforce more than the most obvious weaknesses in the out-of-the-box installations of their platforms
    - Run vulnerability assessments. Since databases are often an organization's largest repository of sensitive information, they should be evaluated to not only search for potential vulnerabilities but also to ensure they fulfill any relevant regulatory compliance requirements
    - Understand user privilege and access. As people change roles or leave an organization, user privileges are often not kept up-to-date, and, as a result, organizations lack a full understanding of who has access to sensitive data.
    - Use data analytics to mitigate risks. Remediating high-risk vulnerabilities and misconfigurations within your databases not only reduces your risk of compromise, but it also narrows the scope of any required compensating controls you might need, such as exploit monitoring.
    - Respond to policy violations in real time. For vulnerabilities that cannot be remediated or patched in a timely manner, real-time database activity monitoring (DAM) can be an appropriate compensating control.
    - Database encryption
    - Data Masking
    - Multi-Factor Authentication (MFA)

- **Cloud Database Migration**

    - Migrate your existing database platform from On-premises datacenter to any cloud Technologies includes (AWS, Azure, Google Cloud, Oracle Cloud & other 3rd party cloud services) & Vice versa, Cloud to On-premises datacenter
    - New database design and implementation in Cloud platform.
    - Cloud database integration with other cloud services as per your business functional requirements.

- **Cloud Database Performance Tuning**

DBaaS* (Database as a service), Managed database service that are fully managed by the vendor, which could be a cloud platform provider or another database vendor that runs its cloud DBMS on a platform provider's infrastructure

Self-managed database*: This is an infrastructure as a service (IaaS) environment, in which the database runs in a virtual machine on a system operated by a cloud provider

PostgreSQL Extensions (Level-3)

Modify/Extend the way that Postgres works

- File_fdw
- Dblink
- Postgres_fdw
- Pg_stat_statement
- Pg_trgm
- Hstore
- Postgis
- Postgis_topology
- TimescaleDB
- Pg_cron
- Pg_metrics
- Pg_repack
- pgBadger
- pgAudit
- pg_Promethus

Support using various PostgreSQL Tools (Level-3)

Monitoring

- **PoWA** - PoWA (PostgreSQL Workload Analyzer) is a PostgreSQL workload analysis tool
- **PgCluu** - pgCluu is a PostgreSQL performance monitoring and auditing tool.
- **Pgwatch2** - It is based on Grafana and provides monitoring functions for the PostgreSQL DB.
- **PgAudit** – provides detailed session and/or object audit logging via standard Postgresql logging facility

Logic and trigger-based replication tools

- **pgLogical -** It is a logical replication tool implemented in the form of PostgreSQL extension plug-ins.

Multi-master replication tool

- BDR - Bi-Directional Replication for PostgreSQL

High availability and failover tools

- Repmgr - Repmgr is an open-source tool for PostgreSQL server cluster replication and failover
- PAF - PostgreSQL Automatic Failover-Automatic Failover Tool
- Patroni - Template that uses Python for highly available solutions for maximum usability
- Stolon - Stolon is a cloud-native PostgreSQL high-availability management tool

Connection Pooling Tools

- PgBouncer - PgBouncer allows client access to PostgreSQL database operations greater than the maximum number of connections it can provide
- PgPool-II - The functions it can provide include query-based replication, connection pool function, load balancing, parallel query, etc

Table partitioning tool

- Pg_Partman – It is an extension of PostgreSQL, used to create and manage time-based or sequence-based table partitions
- pg_Pathman - Optimized partition solutions for large distributed databases

Migration tool

- Ora2pg - Tool for migrating Oracle or MySQL databases to PostgreSQL
- pgloader - loads data into PostgreSQL and enables Continuous Migration from your existing database to PostgreSQL. It can load data from files like CSV or Fixed-File Format or convert an entire database to PostgreSQL

Scheduling Tool

- Pg_cron using for cronjob internally in Postgres.