

FM-KED-002 — PostgreSQL Table and Index Bloat (Missing or Ineffective VACUUM)

Severity: S2 — High

Recovery Class: B — Standard Recovery

Covered by Monthly Support: Yes

Description

PostgreSQL database performance degrades over time due to table and index bloat caused by insufficient or ineffective `VACUUM` operations.

This issue manifests gradually and is commonly observed on systems with high write activity, long-running transactions, or misconfigured autovacuum settings.

Typical Symptoms

- Slow queries without obvious query plan changes
 - Increased disk usage on database volumes
 - Tables or indexes significantly larger than expected
 - Elevated I/O usage
 - Application timeouts under normal load
-

Diagnostic Checklist

Identify Database Size and Largest Tables

```
SELECT
  relname AS table_name,
  pg_size_pretty(pg_total_relation_size(relid)) AS total_size
FROM pg_catalog.pg_statio_user_tables
ORDER BY pg_total_relation_size(relid) DESC
```

```
LIMIT 10;
```

Check Autovacuum Activity

```
SELECT
  relname,
  last_vacuum,
  last_autovacuum,
  n_dead_tup
FROM pg_stat_user_tables
ORDER BY n_dead_tup DESC;
```

Recovery Procedure

Follow steps **carefully**. Some operations are I/O intensive.

1. Run Manual VACUUM (Non-blocking)

```
VACUUM (VERBOSE, ANALYZE);
```

Recommended for moderate bloat and active systems.

2. Vacuum Specific Tables

```
VACUUM (VERBOSE, ANALYZE) table_name;
```

Use when bloat is localized.

3. Reclaim Disk Space (Blocking)

```
VACUUM FULL table_name;
```

⚠ Locks the table for the duration of the operation

⚠ Use during maintenance windows only

4. Reindex Bloated Indexes

```
REINDEX TABLE table_name;
```

Or concurrently, when supported:

```
REINDEX INDEX CONCURRENTLY index_name;
```

Preventive Notes

- Ensure autovacuum is enabled and properly tuned
 - Monitor `n_dead_tup` growth over time
 - Avoid long-running transactions
 - Schedule periodic maintenance for write-heavy tables
-

Operational Notes

- Disk space reclaimed by `VACUUM` is reusable by PostgreSQL, not always returned to the OS
 - `VACUUM FULL` physically rewrites tables and should be used sparingly
-
-

Revision #1

Created 2026-01-07 14:25:11 UTC by Sergei Zhitenev

Updated 2026-01-07 14:25:48 UTC by Sergei Zhitenev