

# Automated PostgreSQL Backup & Monitoring Proposal (NDA-Compliant)

## Overview

This document outlines a generic, NDA-compliant example of a database automation and monitoring workflow suitable for PostgreSQL environments. It does not reference any company data, internal systems, or proprietary information. It demonstrates technical capability only.

### 1. Objective

Implement an automated daily backup and health-monitoring routine for a PostgreSQL instance to:

- Ensure reliable point-in-time recovery
- Track key performance metrics
- Alert on abnormal behavior
- Reduce manual DBA workload

### 2. Architecture Summary

#### A. Scheduled Backup Automation

Runs nightly using a cron job or scheduled cloud function.

- Performs full logical backup via pg\_dump or base backup via pg\_basebackup.
- Compresses and encrypts backup files.
- Uploads backups to secure storage.

#### B. Lightweight Monitoring Script

Executes every 5 minutes and collects:

- Active connections
- Slow queries
- Replication lag
- Disk usage

#### C. Alerting Logic

Triggers alert when thresholds are exceeded (CPU, connections, replication lag, slow queries).

### 3. Sample Backup Script

```
#!/bin/bash

DATE=$(date +"%Y-%m-%d_%H-%M")

BACKUP_DIR="/var/backups/postgres"

FILE="pg_backup_$DATE.sql.gz"

mkdir -p $BACKUP_DIR

pg_dump -U $PGUSER -h $PGHOST -p $PGPORT $PGDATABASE | gzip > $BACKUP_DIR/$FILE

find $BACKUP_DIR -type f -mtime +7 -delete

echo "Backup completed: $FILE"
```

### 4. Monitoring Snippet

```
SELECT count(*) AS active_connections FROM pg_stat_activity;

SELECT * FROM pg_stat_statements WHERE mean_time > 500;

SELECT now() - pg_last_xact_replay_timestamp() AS replication_lag;
```

### 5. Benefits

- Reduces manual work
- Ensures backup consistency
- Enables proactive alerting
- Supports scaling

### 6. NDA Compliance

Fully anonymized. No schema, credentials, or proprietary data.