Debugging Log Files and PgBadger

Oz · Following

3 min read · Mar 20, 2024

Debugging PostgreSQL performance issues often involves sifting through extensive log files to identify bottlenecks and errors. PgBadger, a powerful log analyzer for PostgreSQL, simplifies this process by providing insightful reports and metrics. In this guide, we'll explore how to use PgBadger to analyze PostgreSQL log files effectively.

logging_collector

- Description: When enabled, a background process captures messages sent to stderr and redirects them to log files.
- Default: Off.
- Importance: Enabling this parameter ensures comprehensive logging without loss of messages, especially critical under high workloads.

log_destination

- Description: Determines where log output is directed.
- Default: stderr (text format).
- Other Options: csvlog (comma-separated format).
- Importance: Choosing the appropriate destination facilitates compatibility with external analyzers or applications.

log_directory

- Description: Specifies the directory for storing log files.
- Dependent: Requires logging_collector to be enabled.
- Recommendation: Use separate storage to prevent disk congestion, particularly under heavy workloads.

log_filename

- Description: Defines the naming convention for log files.
- Dependent: Requires logging_collector to be enabled.
- Example: postgresql-%Y-%m-%d_%H%M%S.log.
- Importance: Establishes a consistent naming scheme for organized log management.

log_rotation_age

- Description: Controls time-based log rotation.
- Default: 24 hours (1440 minutes).
- Flexibility: Can be set to 0 to deactivate time-based rotation.
- Recommendation: Adjust based on workload; daily rotation is typical but weekly rotation may suffice for lighter usage.

log_rotation_size

- Description: Manages size-based log rotation.
- Default: 10MB.
- Zero Option: Setting to 0 disables size-based rotation.
- Importance: Prevents log files from growing excessively, ensuring efficient disk usage.

log_truncate_on_rotation

- Description: Determines whether existing log files are appended or overwritten upon rotation.
- Dependent: Requires logging_collector to be enabled.
- Functionality: Enables control over the number of log files stored.

- Example: If log_filename is configured with hourly rotation, PostgreSQL retains a limited number of files based on this parameter.

log_line_prefix

- Description: Prefix added to each log line.
- Example: '%t [%p]: [%l-1] user=%u,db=%d,host=%h,app=%a,cilent=%h '
- Importance: Formats log lines for better readability and understanding of information.

log_connections / log_disconnections

- Description: Determines whether connection establishment and disconnection operations are logged.
- Importance: Useful for tracking database connection and disconnection events.

log_min_duration_statement

- Description: Controls logging of SQL statements that take longer than a specified duration.
- Importance: Helps identify long-running queries to diagnose performance issues.

log_lock_waits

- Description: Determines whether lock wait situations are logged.
- Importance: Lock wait situations provide important clues for identifying causes of performance issues.

log_autovacuum_min_duration

- Description: Controls logging of autovacuum operations that take longer than a specified duration.
- Importance: Useful for monitoring and adjusting autovacuum operation performance.

These parameters make PostgreSQL log content more meaningful, facilitating easier monitoring of database operations and debugging.

```
log_autovacuum_min_duration: 5000
log_checkpoints: true
log_connections: true
log_directory: /pg_log/log
log_disconnections: true
log_filename: postgresql-%Y-%m-%d_%H%M%S.log
log_line_prefix: '%t [%p]: [%l-1] user=%u,db=%d,host=%h,app=%a,cilent=%h '
log_lock_waits: true
log_min_duration_statement: 5000
log_statement: ddl
log_temp_files: 0
```

Installing and Using PgBadger

Once PostgreSQL logging is configured, install PgBadger to streamline log analysis:

```
dnf install pgbadger
```

PgBadger can parse PostgreSQL logs and generate HTML reports with valuable insights. Let's create a report using PgBadger:

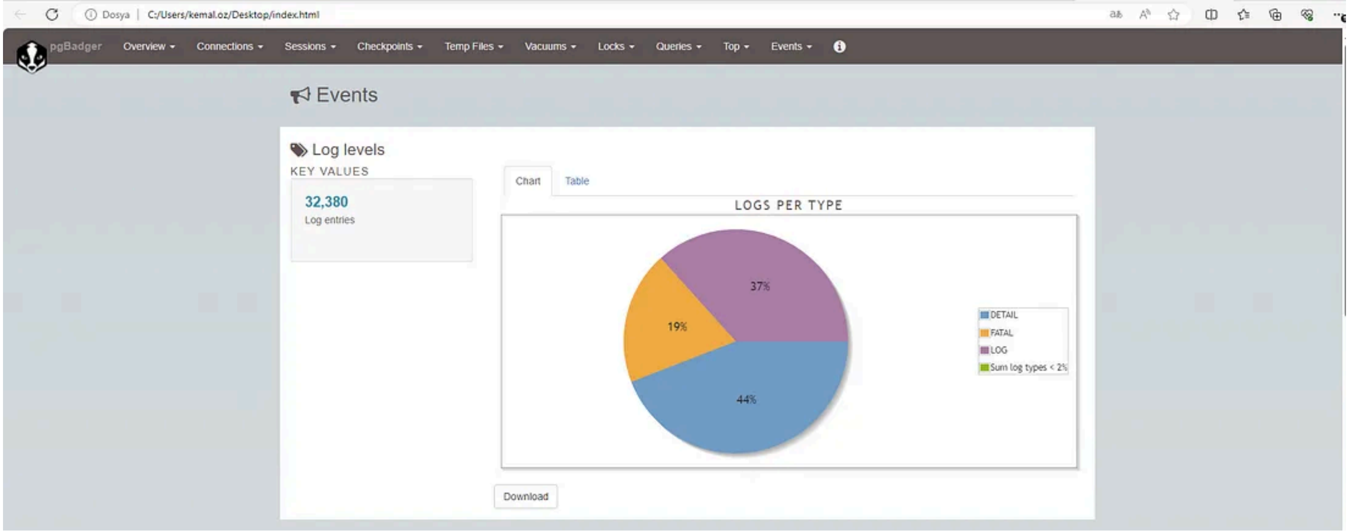
```
pgbadger /pg_log/log/postgresql-2024-03-* -o /tmp/index.html
```

Analyzing PgBadger Reports

PgBadger reports provide a wealth of information, including:

- Overview: Summarizes key metrics like total requests, errors, and slow queries.
- Timeline: Visualizes database activity over time, highlighting peaks and troughs.
- Top Queries: Identifies the most time-consuming queries and their execution frequencies.
- Error Analysis: Pinpoints errors and exceptions encountered during database operations.

- Performance Optimization Suggestions: Offers recommendations for improving database performance based on observed patterns.



PgBadger Report Example

For more detailed and technical articles like this, keep following our blog on Medium. If you have any questions or need further assistance, feel free to reach out in the comments below and directly.

Following

Written by Oz

149 Followers · 13 Following

Database Administrator 🐘

No responses yet

Gvadakte

What are your thoughts?

More from Oz

Queries

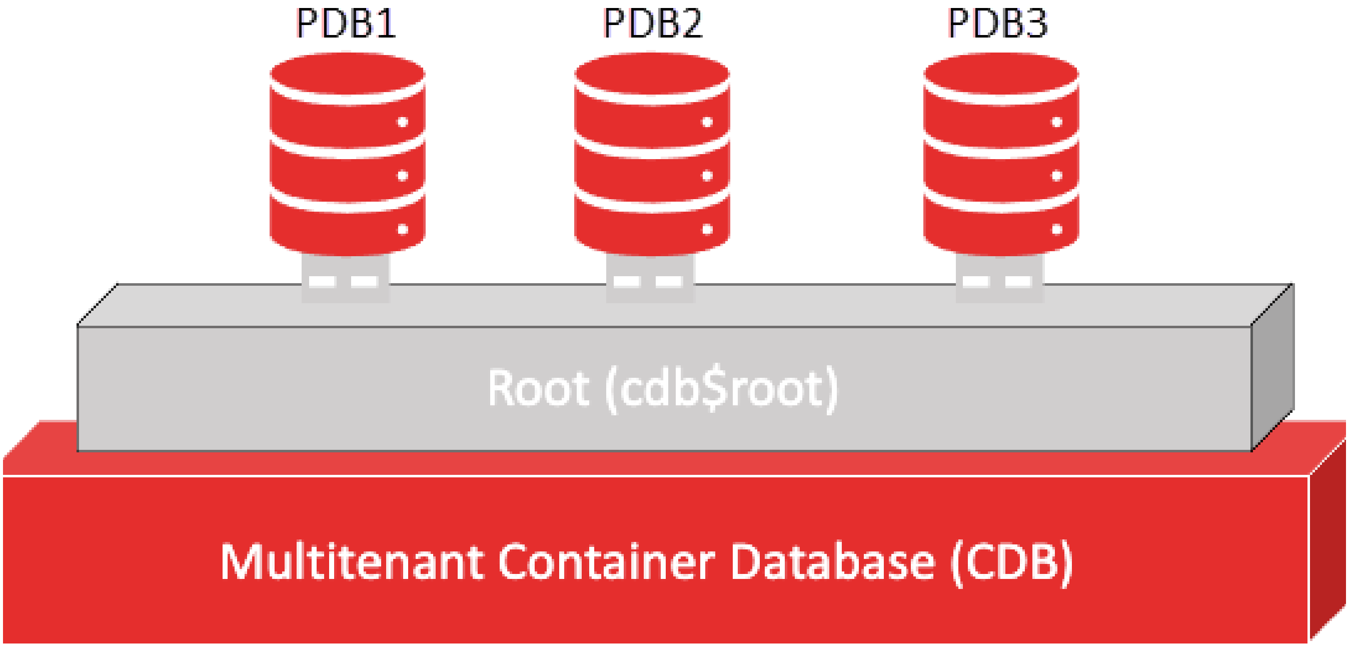
time	Service	username	Query	Execution_time
2024-09-26 13:15:00	pmm-server-postgresql	pmm-ma…	SELECT /* agent='pgstatstatements' */ "pg_stat_statements"."userid", "pg_stat_statements"."dbid", "pg_stat_statements"."queryi…	0.0242
2024-09-26 13:19:00	pmm-server-postgresql	pmm-ma…	SELECT current_database() datname, schemaname, relname, heap_blks_read, heap_blks_hit, idx_blks_read, idx_blks_hit, toast_b…	0.0165
2024-09-26 13:10:48	pmm-server-postgresql	pmm-ma…	SELECT * FROM pg_stat_database_conflicts	0.0159
2024-09-26 13:18:00	pmm-server-postgresql	pmm-ma…	SELECT /* agent='pgstatstatements' */ "pg_stat_statements"."userid", "pg_stat_statements"."dbid", "pg_stat_statements"."queryi…	0.0131
2024-09-26 13:19:00	pmm-server-postgresql	pmm-ma…	SELECT /* agent='pgstatstatements' */ "pg_stat_statements"."userid", "pg_stat_statements"."dbid", "pg_stat_statements"."queryi…	0.0131
2024-09-26 13:12:00	pmm-server-postgresql	pmm-ma…	SELECT /* agent='pgstatstatements' */ "pg_stat_statements"."userid", "pg_stat_statements"."dbid", "pg_stat_statements"."queryi…	0.0126
2024-09-26 13:15:00	pmm-server-postgresql	pmm-ma…	SELECT current_database() datname, schemaname, relname, heap_blks_read, heap_blks_hit, idx_blks_read, idx_blks_hit, toast_b…	0.0113

Oz

Installing Percona Monitoring & Management (PMM) with Postgres

Introduction:

Sep 26, 2024 54 1

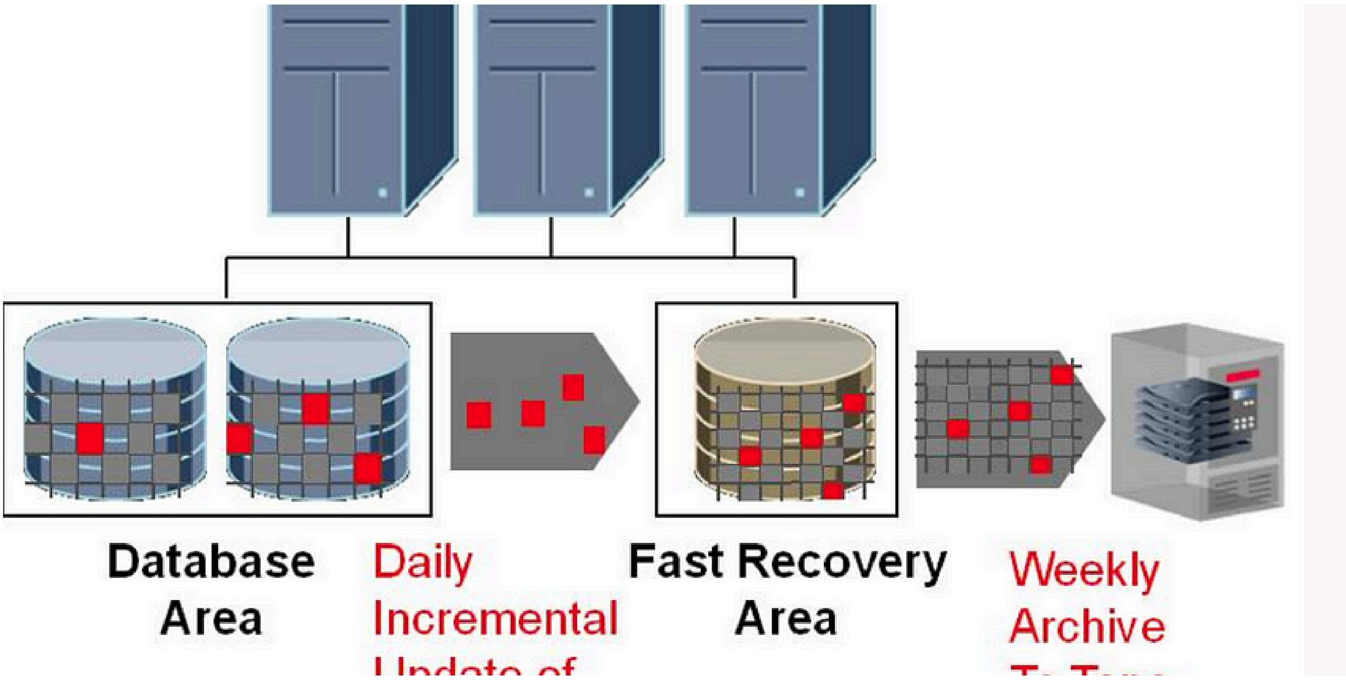


Oz

Pluggable Database Command

—————————— -————————————— - create pluggable database pdb1 admin user root identified by test123; alter pluggable database…

May 12, 2023



Oz

RMAN Backup Basic Commands

rman target / rman target sys/password@YDKTST; backup database; backup database format ‘/backup/path/%d_%t_%s.rman’; backup tablespace…

✦ May 11, 2023 👏 1

Oz

delete jobs

✦ May 8, 2023

See all from Oz

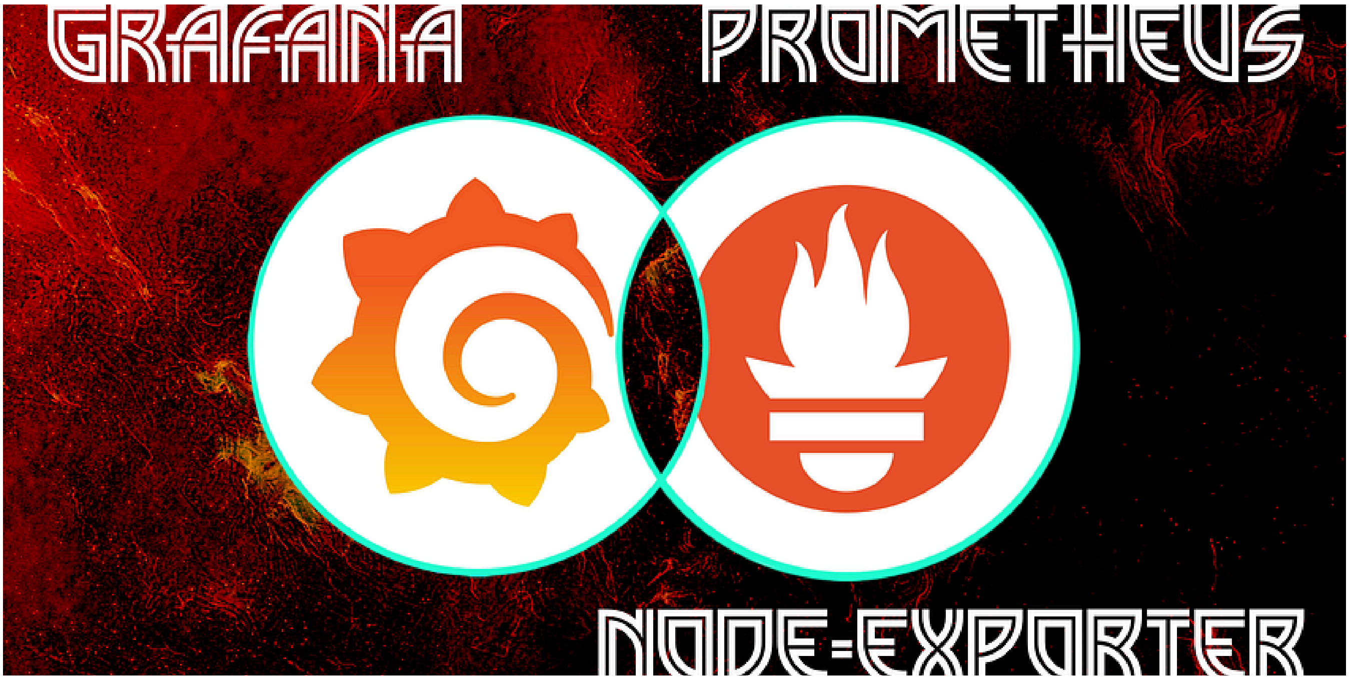
Recommended from Medium

Shashank Khasare

Sending Logs to a Database with Log4j2: A Step-by-Step Guide

Logging is an essential aspect of monitoring and troubleshooting in modern applications. Effective logging helps in understanding...

Oct 24, 2024



crptcpchk

Grafana, Prometheus & Node-Exporter | Setup Guide

Grafana open source software enables you to query, visualize, alert on, and explore your metrics, logs, and traces wherever they are stored...

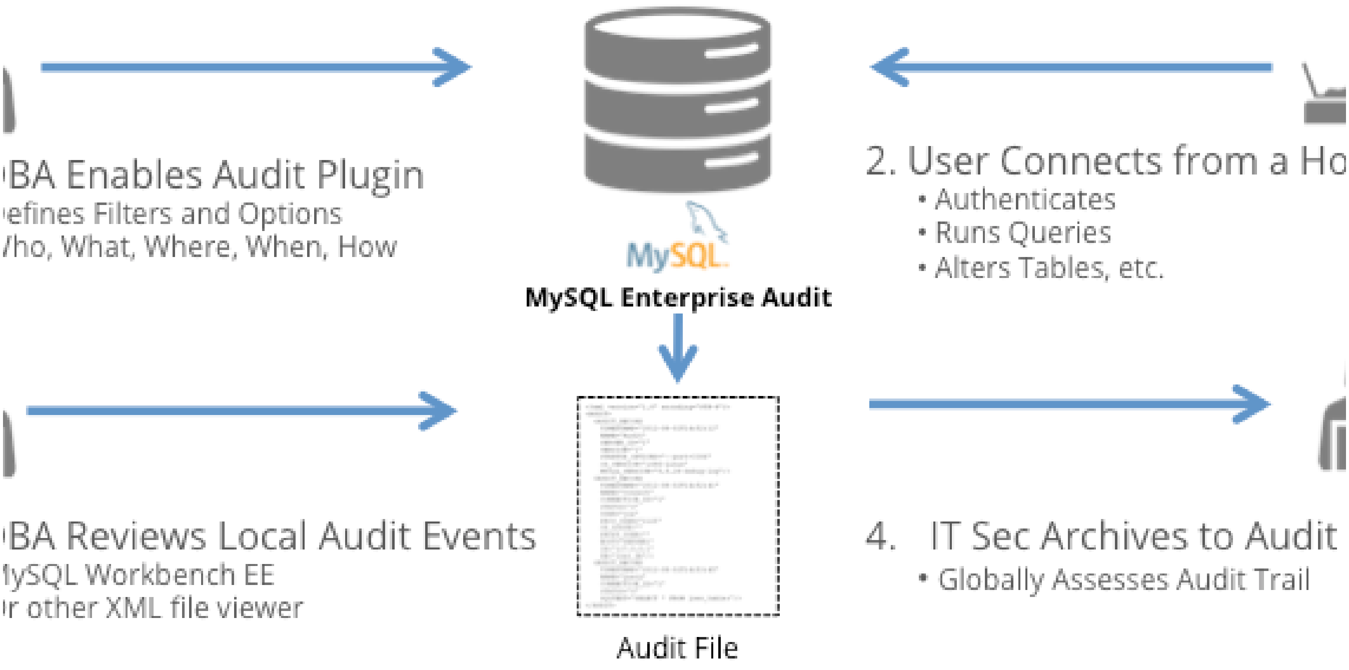
Oct 30, 2024 8 1

mehmetozanguven

Running PostgreSQL with Podman

Instead of running PostgresSQL locally, we can easily run with Podman. Here are the basic steps you should follow.

Mar 28 2


BA Enables Audit Plugin
efines Filters and Options
/ho, What, Where, When, How
2. User Connects from a Ho
• Authenticates
• Runs Queries
• Alters Tables, etc.
MySQL
MySQL Enterprise Audit
BA Reviews Local Audit Events
lySQL Workbench EE
r other XML file viewer
Audit File
4. IT Sec Archives to Audit
• Globally Assesses Audit Trail

Hari

How to Enable MYSQL DB audit logs

Step 1: Install the MySQL Enterprise Audit Plugin

✦ Nov 12, 2024 53

In Towards Dev by Nakul Mitra

PostgreSQL Performance Optimization—Cleaning Dead Tuples & Reindexing

Performance optimization is crucial in PostgreSQL to ensure efficient query execution and minimal resource consumption.

Mar 28 1

In DevOps Playbook by Shlpa S Behani

🚀 Ultimate Guide to Installing and Configuring the ELK Stack (Elasticsearch, Logstash, Kibana) on…

🔹 Introduction

✦ Mar 10 👏 3

See more recommendations