Postgres Security 101: PostgreSQL Settings (6/8)

Oz · Following

11 min read · Oct 4, 2024

PostgreSQL is known for its robust security features, offering a wide range of settings that can be fine-tuned to protect your data and ensure the integrity of your database. In this part of the **Postgres Security 101** series, we dive into some essential PostgreSQL settings that every database administrator should be familiar with. From managing authentication methods to configuring connection encryption, these settings are key to fortifying your database environment against potential threats. Whether you're just starting out with PostgreSQL or looking to enhance your existing setup, this guide will give you the foundational knowledge needed to safeguard your system.

6.1 Understanding Attack Vectors and Runtime Parameters

- Gain a thorough understanding of potential attack vectors and how to configure runtime parameters.

```
Examples;
--------------
1. Weak Authentication
2. SQL injection
3. Privilege abuse
4. Excessive privileges
5. Phishing
6. Inadequate logging and weak auditing
7. Denial of service
8. Exploiting unpatched services
9. Insecure system architecture
10. Inadequate Backup
```

6.2 Ensure ‘Backend’ Runtime Parameters Are Configured Correctly

- Configure backend parameters to enhance security and performance. A denial of service is possible by denying the use of indexes and by slowing down client

access to an unreasonable level. Unsanctioned behavior can be introduced by introducing rogue libraries which can then be called in a database session. Logging can be altered and obfuscated inhibiting root cause analysis. All changes made on this level will affect the overall behavior of the server. These changes can only be affected by a server restart after the parameters have been altered in the configuration files.

```
SELECT name, setting FROM pg_settings WHERE context IN
('backend','superuser-backend') ORDER BY 1;

name          | setting
----------------------+---------
 ignore_system_indexes | off
 jit_debugging_support | off
 jit_profiling_support | off
 log_connections       | on
 log_disconnections    | on
 post_auth_delay       | 0
ps -few | grep -E -- '[p]ost.*-[D]'
""" Output"""
postgres 2744612       1  0 Jun10 ?
00:00:04 /usr/pgsql-14/bin/postgres -D /var/data/
--config-file=/var/data/postgresql.conf
--listen_addresses=10.5.56.67 --port=5432
--cluster_name=denemek --wal_level=replica
--hot_standby=on
--max_connections=100
--max_wal_senders=10
--max_prepared_transactions=0
--max_locks_per_transaction=64
--track_commit_timestamp=off
--max_replication_slots=10
--max_worker_processes=8
--wal_log_hints=on
```

6.3 Ensure ‘Postmaster’ Runtime Parameters Are Configured Correctly (Manual)

- Manually review and configure postmaster parameters. The postmaster process is the supervisory process that assigns a backend process to an incoming client connection. The postmaster manages key runtime parameters that are either shared by all backend connections or needed by the postmaster process itself to run. The following parameters can only be set at server start by the owner of the PostgreSQL server process and cluster, typically the UNIX user account postgres.

Therefore, all exploits require the successful compromise of either that UNIX account or the postgres superuser account itself.

```
SELECT name, setting FROM pg_settings WHERE context = 'postmaster'
ORDER BY 1;
                 name                  |            setting
---------------------------------------+-------------------------------
 archive_mode                          | on
 autovacuum_freeze_max_age             | 200000000
 autovacuum_max_workers                | 3
 autovacuum_multixact_freeze_max_age   | 400000000
 bonjour                               | off
 bonjour_name                          |
 cluster_name                          | denemek
 config_file                           | /var/data/postgresql.conf
 data_directory                        | /var/data
 data_sync_retry                       | off
 dynamic_shared_memory_type            | posix
 event_source                          | PostgreSQL
 external_pid_file                     |
 hba_file                              | /var/data/pg_hba.conf
 hot_standby                           | on
 huge_pages                            | try
 huge_page_size                        | 0
 ident_file                            | /var/data/pg_ident.conf
 ignore_invalid_pages                  | off
 jit_provider                          | llvmjit
 listen_addresses                      | 10.20.23.12
 logging_collector                     | on
 max_connections                       | 100
 max_files_per_process                 | 1000
 max_locks_per_transaction             | 64
 max_logical_replication_workers       | 4
 max_pred_locks_per_transaction        | 64
 max_prepared_transactions             | 0
 max_replication_slots                 | 10
 max_wal_senders                       | 10
 max_worker_processes                  | 8
 min_dynamic_shared_memory             | 0
 old_snapshot_threshold                | -1
 port                                  | 5432
 recovery_target                       |
 recovery_target_action                | pause
 recovery_target_inclusive             | on
 recovery_target_lsn                   |
 recovery_target_name                  |
 recovery_target_time                  |
 recovery_target_timeline              | latest
 recovery_target_xid                   |
 shared_buffers                        | 16384
```

```
 shared_memory_type                  | mmap
 shared_preload_libraries            | set_user,$libdir/passwordcheck
 superuser_reserved_connections      | 3
 track_activity_query_size           | 1024
 track_commit_timestamp              | off
 unix_socket_directories             | /var/run/postgresql, /tmp
 unix_socket_group                   |
 unix_socket_permissions             | 0777
 wal_buffers                         | 512
 wal_level                           | replica
 wal_log_hints                       | on
(54 rows)


ps -few | grep -E -- '[p]ost.*-[D]'
""" Output"""
postgres 2744612       1  0 Jun10 ?
00:00:04 /usr/pgsql-14/bin/postgres -D /var/data/
--config-file=/var/data/postgresql.conf
--listen_addresses=10.5.56.67 --port=5432
--cluster_name=denemek --wal_level=replica
--hot_standby=on
--max_connections=100
--max_wal_senders=10
--max_prepared_transactions=0
--max_locks_per_transaction=64
--track_commit_timestamp=off
--max_replication_slots=10
--max_worker_processes=8
--wal_log_hints=on
```

6.4 Ensure ‘SIGHUP’ Runtime Parameters Are Configured Correctly

- Manually configure SIGHUP parameters for signal handling. In order to define server behavior and optimize server performance, the server’s superuser has the privilege of setting these parameters which are found in the configuration files postgresql.conf and pg_hba.conf. Alternatively, those parameters found in postgresql.conf can also be changed using a server login session and executing the SQL command ALTER SYSTEM which writes its changes in the configuration file postgresql.auto.conf. All changes made on this level will affect the overall behavior of the server. These changes can be effected by editing the PostgreSQL configuration files and by either executing a server SIGHUP from the command line or, as superuser postgres, executing the SQL command select pg_reload_conf(). A denial of service is possible by the over-allocating of limited resources, such as RAM. Data can be corrupted by allowing damaged pages to load or by changing parameters to reinterpret values in an unexpected fashion,

e.g. changing the time zone. Client messages can be altered in such a way as to interfere with the application logic. Logging can be altered and obfuscated inhibiting root cause analysis.

```
SELECT name, setting FROM pg_settings WHERE context = 'sighup'
ORDER BY 1;
                  name                  |
----------------------------------------+--------------------------------------
 archive_cleanup_command                |
 archive_command                        | pgbackrest --stanza=cbs_backup archiv
 archive_timeout                        | 0
 authentication_timeout                 | 60
 autovacuum                             | on
 autovacuum_analyze_scale_factor        | 0.1
 autovacuum_analyze_threshold           | 50
 autovacuum_naptime                     | 60
 autovacuum_vacuum_cost_delay           | 2
 autovacuum_vacuum_cost_limit           | -1
 autovacuum_vacuum_insert_scale_factor  | 0.2
 autovacuum_vacuum_insert_threshold     | 1000
 autovacuum_vacuum_scale_factor         | 0.2
 autovacuum_vacuum_threshold            | 50
 autovacuum_work_mem                    | -1
 bgwriter_delay                         | 200
 bgwriter_flush_after                   | 64
 bgwriter_lru_maxpages                  | 100
 bgwriter_lru_multiplier                | 2
 checkpoint_completion_target           | 0.9
 checkpoint_flush_after                 | 32
 checkpoint_timeout                     | 300
 checkpoint_warning                     | 30
 db_user_namespace                      | off
 fsync                                  | on
 full_page_writes                       | on
 hot_standby_feedback                   | off
 krb_caseins_users                      | off
 krb_server_keyfile                     | FILE:/etc/sysconfig/pgsql/krb5.keytab
 log_autovacuum_min_duration            | -1
 log_checkpoints                        | off
 log_destination                        | stderr
 log_directory                          | log
 log_file_mode                          | 0600
 log_filename                           | postgresql-%a.log
 log_hostname                           | off
 log_line_prefix                        | %m [%p]
 log_recovery_conflict_waits            | off
 log_rotation_age                       | 1440
 log_rotation_size                      | 0
 log_timezone                           | US/Eastern
```

```
log_truncate_on_rotation               | on
max_pred_locks_per_page                | 2
max_pred_locks_per_relation            | -2
max_slot_wal_keep_size                 | -1
max_standby_archive_delay              | 30000
max_standby_streaming_delay            | 30000
max_sync_workers_per_subscription      | 2
max_wal_size                           | 1024
min_wal_size                           | 80
pre_auth_delay                         | 0
primary_conninfo                       | user=repuser passfile=/tmp/pgpass hos
primary_slot_name                      | pg_node2
promote_trigger_file                   |
recovery_end_command                   |
recovery_init_sync_method              | fsync
recovery_min_apply_delay               | 0
remove_temp_files_after_crash          | on
restart_after_crash                    | on
restore_command                        | pgbackrest --stanza=cbs_backup archiv
set_user.block_alter_system            | on
set_user.block_copy_program            | on
set_user.block_log_statement           | on
set_user.exit_on_error                 | on
set_user.nosuperuser_target_allowlist  | *
set_user.superuser_allowlist           | *
set_user.superuser_audit_tag           | AUDIT
ssl                                    | on
ssl_ca_file                            | /var/data/root.crt
ssl_cert_file                          | /var/data/server.crt
ssl_ciphers                            | HIGH:MEDIUM:+3DES:!aNULL
ssl_crl_dir                            |
ssl_crl_file                           |
ssl_dh_params_file                     |
ssl_ecdh_curve                         | prime256v1
ssl_key_file                           | server.key
ssl_max_protocol_version               |
ssl_min_protocol_version               | TLSv1.2
ssl_passphrase_command                 |
ssl_passphrase_command_supports_reload | off
ssl_prefer_server_ciphers              | on
stats_temp_directory                   | pg_stat_tmp
synchronous_standby_names              |
syslog_facility                        | local0
syslog_ident                           | postgres
syslog_sequence_numbers                | on
syslog_split_messages                  | on
trace_recovery_messages                | log
vacuum_defer_cleanup_age               | 0
wal_keep_size                          | 128
wal_receiver_create_temp_slot          | off
wal_receiver_status_interval           | 10
wal_receiver_timeout                   | 60000
wal_retrieve_retry_interval            | 5000
```

```
 wal_sync_method                 | fdatasync
 wal_writer_delay                | 200
 wal_writer_flush_after          | 128
(97 rows)
```

6.5 Ensure 'Superuser' Runtime Parameters Are Configured Correctly (Manual)

- Manually configure parameters that apply to superusers.

```
SELECT name, setting FROM pg_settings WHERE context = 'superuser' ORDER BY 1;
            name             |   setting
-----------------------------+-------------
 allow_in_place_tablespaces  | off
 allow_system_table_mods     | off
 backtrace_functions         |
 commit_delay                | 0
 compute_query_id            | auto
 deadlock_timeout            | 1000
 debug_discard_caches        | 0
 dynamic_library_path        | $libdir
 ignore_checksum_failure     | off
 jit_dump_bitcode            | off
 lc_messages                 | en_US.UTF-8
 lo_compat_privileges        | off
 log_duration                | off
 log_error_verbosity         | default
 log_executor_stats          | off
 log_lock_waits              | off
 log_min_duration_sample     | -1
 log_min_duration_statement  | -1
 log_min_error_statement     | error
 log_min_messages            | warning
 log_parameter_max_length    | -1
 log_parser_stats            | off
 log_planner_stats           | off
 log_replication_commands    | off
 log_statement               | none
 log_statement_sample_rate   | 1
 log_statement_stats         | off
 log_temp_files              | -1
 log_transaction_sample_rate | 0
 max_stack_depth             | 2048
 session_preload_libraries   |
 session_replication_role    | origin
 temp_file_limit             | -1
 track_activities            | on
 track_counts                | on
```

```
 track_functions              | none
 track_io_timing              | off
 track_wal_io_timing          | off
 update_process_title         | on
 wal_compression              | off
 wal_consistency_checking     |
 wal_init_zero                | on
 wal_recycle                  | on
 zero_damaged_pages           | off
(44 rows)
```

6.6 Ensure 'User' Runtime Parameters Are Configured Correctly (Manual)

- Manually configure parameters that apply to regular users. These PostgreSQL runtime parameters are managed at the user account (ROLE) level. In order to improve performance and optimize features, a ROLE has the privilege of setting numerous parameters in a transaction, session, or entity attribute. Any ROLE can alter any of these parameters. A denial of service is possible by the over-allocating of limited resources, such as RAM. Changing VACUUM parameters can force a server shutdown which is standard procedure preventing data corruption from transaction ID wraparound. Data can be corrupted by changing parameters to reinterpret values in an unexpected fashion, e.g. changing the time zone. Logging can be altered and obfuscated to inhibit root cause analysis.

```
SELECT name, setting FROM pg_settings WHERE context = 'user' ORDER BY 1;
                 name                  |      setting
---------------------------------------+--------------------
 application_name                      | psql
 array_nulls                           | on
 backend_flush_after                   | 0
 backslash_quote                       | safe_encoding
 bytea_output                          | hex
 check_function_bodies                 | on
 client_connection_check_interval      | 0
 client_encoding                       | UTF8
 client_min_messages                   | notice
 commit_siblings                       | 5
 constraint_exclusion                  | partition
 cpu_index_tuple_cost                  | 0.005
 cpu_operator_cost                     | 0.0025
 cpu_tuple_cost                        | 0.01
 cursor_tuple_fraction                 | 0.1
 DateStyle                             | ISO, MDY
 debug_pretty_print                    | on
```

```
debug_print_parse                    | off
debug_print_plan                     | off
debug_print_rewritten                | off
default_statistics_target            | 100
default_table_access_method          | heap
default_tablespace                   |
default_text_search_config           | pg_catalog.english
default_toast_compression            | pglz
default_transaction_deferrable       | off
default_transaction_isolation        | read committed
default_transaction_read_only        | off
effective_cache_size                 | 524288
effective_io_concurrency             | 1
enable_async_append                  | on
enable_bitmapscan                    | on
enable_gathermerge                   | on
enable_hashagg                       | on
enable_hashjoin                      | on
enable_incremental_sort              | on
enable_indexonlyscan                 | on
enable_indexscan                     | on
enable_material                      | on
enable_memoize                       | on
enable_mergejoin                     | on
enable_nestloop                      | on
enable_parallel_append               | on
enable_parallel_hash                 | on
enable_partition_pruning             | on
enable_partitionwise_aggregate       | off
enable_partitionwise_join            | off
enable_seqscan                       | on
enable_sort                          | on
enable_tidscan                       | on
escape_string_warning                | on
exit_on_error                        | off
extra_float_digits                   | 1
force_parallel_mode                  | off
from_collapse_limit                  | 8
geqo                                 | on
geqo_effort                          | 5
geqo_generations                     | 0
geqo_pool_size                       | 0
geqo_seed                            | 0
geqo_selection_bias                  | 2
geqo_threshold                       | 12
gin_fuzzy_search_limit               | 0
gin_pending_list_limit               | 4096
hash_mem_multiplier                  | 1
idle_in_transaction_session_timeout  | 0
idle_session_timeout                 | 0
IntervalStyle                        | postgres
jit                                  | on
jit_above_cost                       | 100000
```

```
jit_expressions                 | on
jit_inline_above_cost           | 500000
jit_optimize_above_cost         | 500000
jit_tuple_deforming             | on
join_collapse_limit             | 8
lc_monetary                     | en_US.UTF-8
lc_numeric                      | en_US.UTF-8
lc_time                         | en_US.UTF-8
local_preload_libraries         |
lock_timeout                    | 0
logical_decoding_work_mem       | 65536
log_parameter_max_length_on_error | 0
maintenance_io_concurrency      | 10
maintenance_work_mem            | 65536
max_parallel_maintenance_workers | 2
max_parallel_workers            | 8
max_parallel_workers_per_gather | 2
min_parallel_index_scan_size    | 64
min_parallel_table_scan_size    | 1024
parallel_leader_participation   | on
parallel_setup_cost             | 1000
parallel_tuple_cost             | 0.1
password_encryption             | scram-sha-256
plan_cache_mode                 | auto
quote_all_identifiers           | off
random_page_cost                | 4
row_security                    | on
search_path                     | "$user", public
seq_page_cost                   | 1
standard_conforming_strings     | on
statement_timeout               | 0
synchronize_seqscans            | on
synchronous_commit              | on
tcp_keepalives_count            | 0
tcp_keepalives_idle             | 0
tcp_keepalives_interval         | 0
tcp_user_timeout                | 0
temp_buffers                    | 1024
temp_tablespaces                |
TimeZone                        | Europe/Istanbul
timezone_abbreviations          | Default
trace_notify                    | off
trace_sort                      | off
transaction_deferrable          | off
transaction_isolation           | read committed
transaction_read_only           | on
transform_null_equals           | off
vacuum_cost_delay               | 0
vacuum_cost_limit               | 200
vacuum_cost_page_dirty          | 20
vacuum_cost_page_hit            | 1
vacuum_cost_page_miss           | 2
vacuum_failsafe_age             | 1600000000
```

```
 vacuum_freeze_min_age                | 50000000
 vacuum_freeze_table_age              | 150000000
 vacuum_multixact_failsafe_age        | 1600000000
 vacuum_multixact_freeze_min_age      | 5000000
 vacuum_multixact_freeze_table_age    | 150000000
 wal_sender_timeout                   | 60000
 wal_skip_threshold                   | 2048
 work_mem                             | 4096
 xmlbinary                            | base64
 xmloption                            | content
(133 rows)
```

6.7 Ensure FIPS 140–2 OpenSSL Cryptography Is Used

- Use FIPS 140–2 compliant cryptography for enhanced security. Install, configure, and use OpenSSL on a platform that has a NIST certified FIPS 140–2 installation of OpenSSL. This provides PostgreSQL instances the ability to generate and validate cryptographic hashes to protect unclassified information requiring confidentiality and cryptographic protection, in accordance with the data owner's requirements. Configure OpenSSL to be FIPS compliant as PostgreSQL uses OpenSSL for cryptographic modules. To configure OpenSSL to be FIPS 140–2 compliant, see the ***official RHEL Documentation.***

```
fips-mode-setup --check

#Output
Installation of FIPS modules is not completed.
FIPS mode is disabled.
fips-mode-setup --enable
#Output
Kernel initramdisks are being regenerated. This might take some time.
Setting system policy to FIPS
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
FIPS mode will be enabled.
Please reboot the system for the setting to take effect.
fips-mode-setup --check
#Output
FIPS mode is enabled.
openssl version
#Output
OpenSSL 3.0.7 1 Nov 2022 (Library: OpenSSL 3.0.7 1 Nov 2022)
```

6.8 Ensure TLS Is Enabled and Configured Correctly

- Enable and properly configure TLS for secure communication. If TLS is not enabled and configured correctly, this increases the risk of data being compromised in transit.

```
SHOW ssl;
ssl - - - off (1 row)

SELECT name, setting, source FROM pg_settings WHERE name = 'ssl';
 name | setting | source
- - -+ - - - - -+ - - - - - - - - - - -
ssl | off | default
(1 row)
```

If your output like this above, please read the ***Securing PostgreSQL with SSL Encryption***. However, do not forget A self-signed certificate can be used for **testing**. On the other hand, a certificate signed by a certificate authority (CA) (either one of the global CAs or a local one) should be used in **production** so that clients can verify the server's identity. **If all the database clients are local to the organization, using a local CA is recommended**

6.9 Ensure a Cryptographic Extension Is Installed

- Install and use cryptographic extensions for data encryption. When considering or undertaking any form of encryption, it is critical to understand the state of the encrypted data at all stages of the data lifecycle. The use of pgcrypto ensures that the data at rest in the tables (and therefore on disk) is encrypted, but for the data to be accessed by any users or applications, said users/applications will, by necessity, have access to the encrypt and decrypt keys and the data in question will be encrypted/decrypted in memory and then transferred to/from the user/application in that form

```
SELECT * FROM pg_available_extensions WHERE name='pgcrypto';
   name   | default_version | installed_version |         comment
----------+-----------------+-------------------+-------------------------
 pgcrypto | 1.3             |                   | cryptographic functions

CREATE EXTENSION pgcrypto;
--The pgcrypto extension is included with the PostgreSQL 'contrib' package. Alt
```

```
SELECT * FROM pg_available_extensions WHERE name='pgcrypto';
   name   | default_version | installed_version |         comment
----------+-----------------+-------------------+-------------------------
 pgcrypto | 1.3             | 1.3               | cryptographic functions
```

6.10 Ensure a Data Anonymization Extension Is Installed

Use data anonymization extensions to protect sensitive information. Also, you can read this *Enhancing Data Security in PostgreSQL: Using pgcrypto and Anonymizer Extensions*

```
dependencies:
-------------
ddlx_14
python3-faker
rpm -iv ddlx_14-0.27-1PGDG.rhel9.noarch.rpm
rpm -iv python3-faker-13.3.3-1.el9.noarch.rpm
rpm -iv postgresql_anonymizer_14-1.1.0-1.rhel9.x86_64.rpm
create database test;
ALTER DATABASE test SET session_preload_libraries = 'anon';  -- if this is not
\c test;
CREATE EXTENSION anon CASCADE;
SELECT anon.init();

create database employee;
\c employee;
CREATE TABLE people (
    id INT,
    firstname VARCHAR(10),
    lastname VARCHAR(10),
    phone VARCHAR(15)
);
INSERT INTO people (id, firstname, lastname, phone) VALUES (1, 'Kemal', 'Oz', '
create role hr LOGIN;
GRANT SELECT ON people TO hr;
select * from people;
 id | firstname | lastname |   phone
----+-----------+----------+------------
  1 | Kemal     | Oz       | 9012345678

CREATE EXTENSION IF NOT EXISTS anon CASCADE;
SELECT anon.start_dynamic_masking();
CREATE ROLE hr LOGIN;
GRANT SELECT ON people TO hr;
SECURITY LABEL FOR anon ON ROLE hr IS 'MASKED';
SECURITY LABEL FOR anon ON COLUMN people.lastname IS 'MASKED WITH FUNCTION anor
SECURITY LABEL FOR anon ON COLUMN people.phone IS 'MASKED WITH FUNCTION anon.pa
```

```
employee=> select * from people;
 id | firstname |  lastname  |   phone
----+-----------+------------+------------
  1 | Kemal     | Wintheiser | 90******78
select * from people;
 id | firstname | lastname |   phone
----+-----------+----------+------------
  1 | Kemal     | Kshlerin | 90******78
```

Understanding and correctly configuring PostgreSQL security settings is crucial to maintaining a secure and resilient database. By leveraging these configurations, you can protect your data, control access, and ensure that your system is safeguarded against unauthorized access and attacks. As security threats evolve, so must your approach to database security, making it essential to stay informed and proactive. Make sure to stay tuned for the next article in this series, ***Postgres Security 101: Replication (7/8)***, where we'll explore best practices for managing users and roles, ensuring that your PostgreSQL environment remains both secure and efficient. 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.

Database Security | Postgres Security | Security | Cybersecurity | Technology

Following

Written by Oz

149 Followers · 13 Following

Database Administrator

No responses yet

Gvadakte

What are your thoughts?

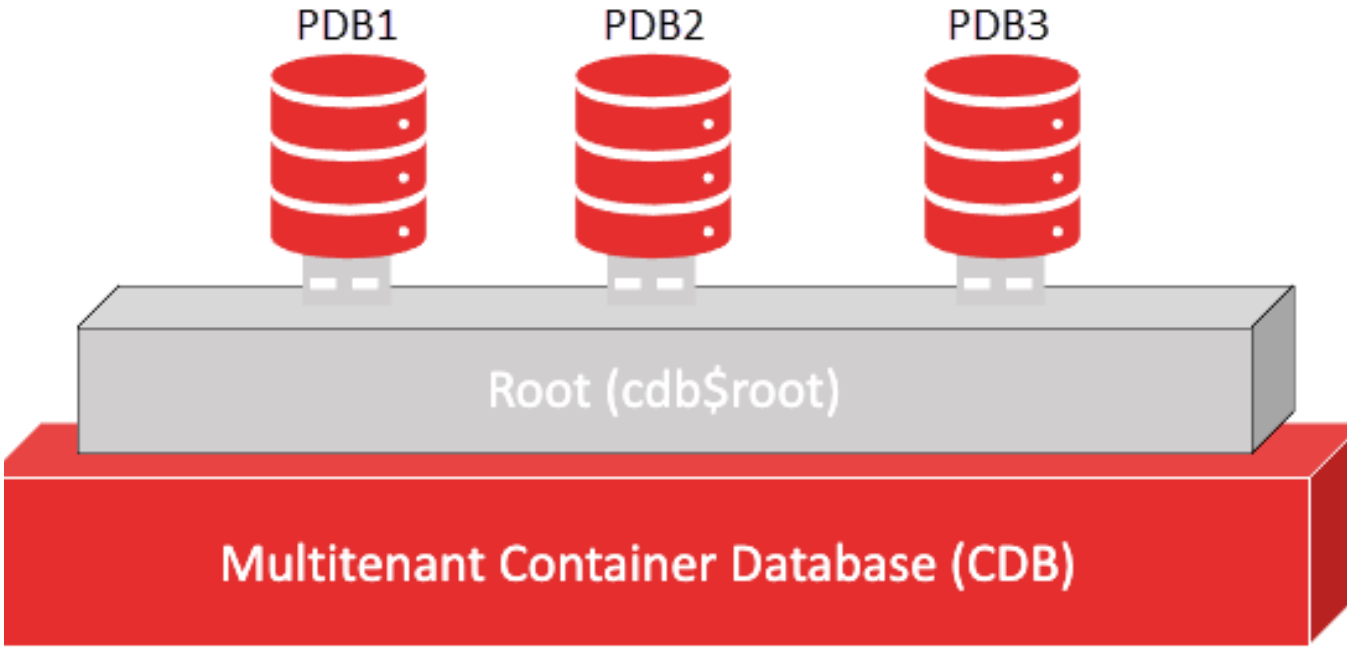
More from Oz

Oz

Installing Percona Monitoring & Management (PMM) with Postgres

Introduction:

Sep 26, 2024 54 1

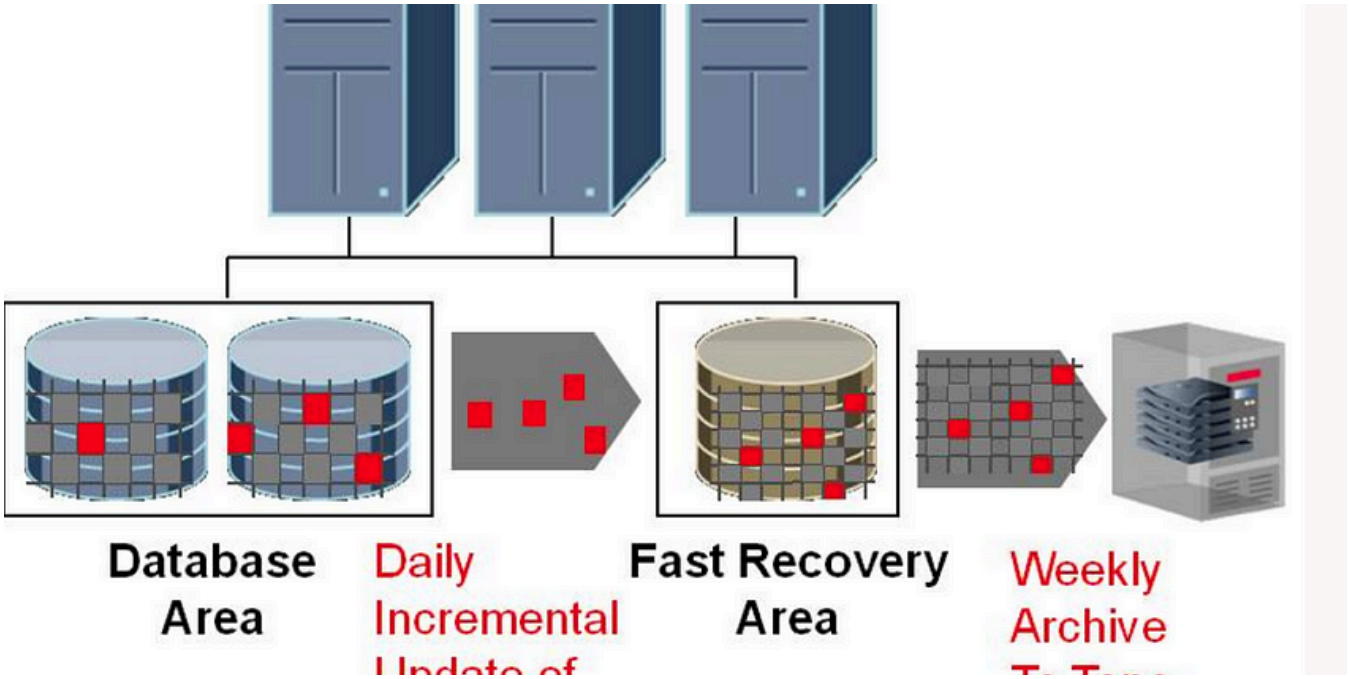


Oz

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

Tihomir Manushev

Vector Search with pgvector in PostgreSQL

Simple AI-powered similarity search

Mar 9

In Databases by Sergey Egorenkov

Making SQL query 40x faster for 10 million rows table

Make your SQL query really fast using this approach

Mar 17 8

at it Means	Best Used For
e directly in the row	Simple data like INT
e in the row (unless large)	Larger types, but tri
npress + store out-of-row	Long texts, large ob
e out-of-row, no compression	When compression

Udbhav Singh

Storages in PostgreSQL

What Are Storage Types in PostgreSQL — And Why Should You Care?

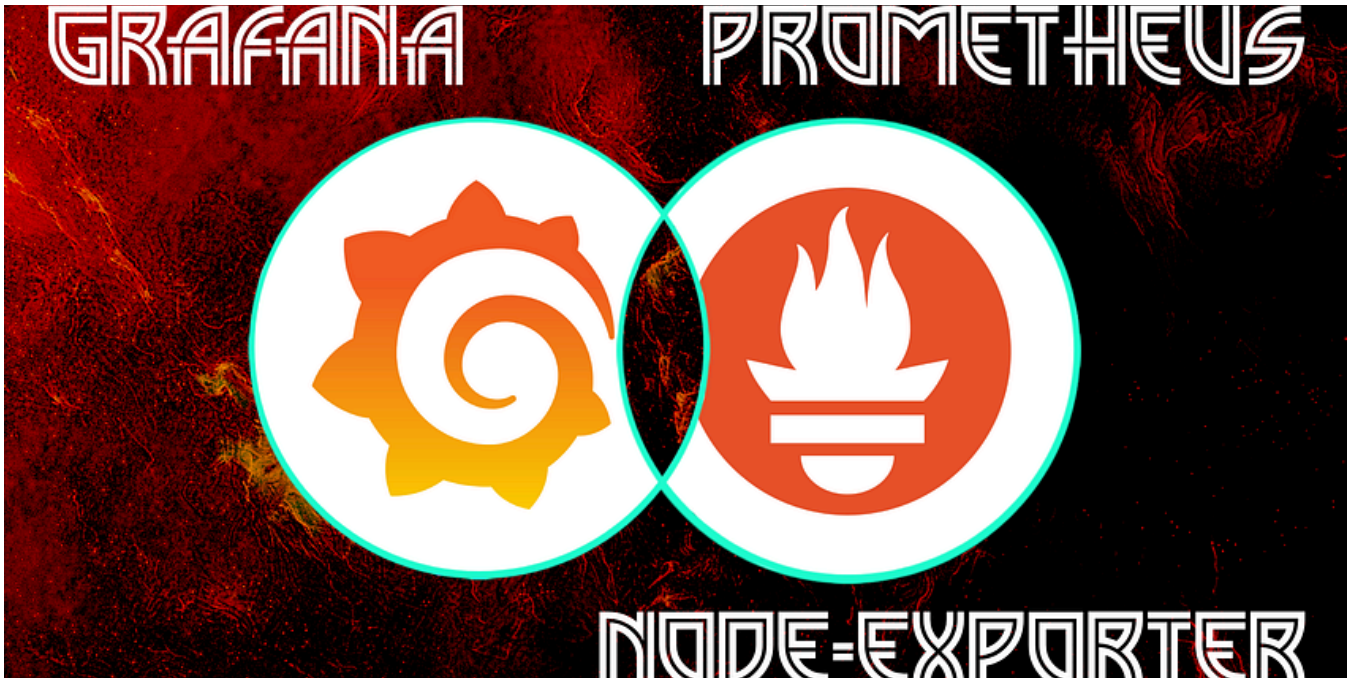
6d ago 18

In Dev Genius by Doran Gao

Efficient Large Table Cleanup in PostgreSQL

“We can only see a short distance ahead, but we can see plenty there that needs to be done.” — Alan Turing

Oct 31, 2024 1



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

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

See more recommendations