

ENGLISH



TUNING OF POSTGRESQL FOR ZABBIX

ARTURS LONTONS
TECHNICAL SUPPORT ENGINEER



01

CONFIGURING ZABBIX FOR POSTGRESQL

ENABLING ZABBIX FOR POSTGRESQL
DATABASE BACKEND



INITIAL ZABBIX CONFIGURATION

- ⊗ Open the zabbix_server.conf file:

```
### Option: DBHost
# Database host name.
# If set to localhost, socket is used for MySQL.
# If set to empty string, socket is used for PostgreSQL.
#
# Mandatory: no
# Default:
DBHost=
```

- ⊗ Be careful. By default preconfigured with localhost!

ENGLISH

ZABBIX 
MEETUP ONLINE '20

 **PostgresPro**



INITIAL ZABBIX CONFIGURATION

- ☑ Next, specify your DB name:

```
### Option: DBName
# Database name.
DBName=zabbix
```

- ☑ And specify your DBSchema:

```
### Option: DBSchema
# Schema name. Used for PostgreSQL.
DBSchema=public
```

- ☑ Empty or commented out value will also reference the public schema!

ENGLISH

ZABBIX 
MEETUP ONLINE '20

 **PostgresPro**



INITIAL ZABBIX CONFIGURATION

- ☑ List all of the available schemas:

```
zabbix-> \dn
List of schemas
Name | Owner
-----+-----
public | postgres
```

- ☑ Show Zabbix table schema:

```
zabbix-> \dt
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | acknowledges | table | zabbix
public | actions | table | zabbix
...
```

ENGLISH

ZABBIX 
MEETUP ONLINE '20

PostgresPro



INITIAL ZABBIX CONFIGURATION

- ⊗ Specify DBUser and DBPassword

```
### Option: DBUser
#       Database user.
DBUser=zabbix

### Option: DBPassword
#       Database password.
DBPassword=zabbix
```

ENGLISH



MEETUP ONLINE '20



INITIAL ZABBIX CONFIGURATION

- ☑ Specify database port:

```
### Option: DBPort
# Database port when not using local socket.DBUser=zabbix
DBPort=5432
```

- ☑ Database port can be either empty or commented out. Default port – 5432 will be used.
- ☑ Custom port for DB backend can be specified in the DB configuration file:

```
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
port = 5433
```

ENGLISH


ZABBIX 
MEETUP ONLINE '20

PostgresPro



INITIAL ZABBIX CONFIGURATION - FRONTEND

- Specify DB credentials for frontend connections:



Details ▲ Cannot connect to the database.

Error connecting to database.

Database type ▼

Database host

Database port 0 - use default port

Database name

Database schema

Store credentials in Plain text HashiCorp Vault

User

Password

Database TLS encryption



INITIAL ZABBIX CONFIGURATION - FRONTEND

- ⊘ The cause for this issue can be found in pg_hba.conf:

```
# "local" is for Unix domain socket connections only
local    all                all                                peer
# IPv4 local connections:
host     all                all                127.0.0.1/32          ident
# IPv6 local connections:
host     all                all                ::1/128              ident
```

ENGLISH

ZABBIX 
MEETUP ONLINE '20


PostgresPro



INITIAL ZABBIX CONFIGURATION - FRONTEND

- ☑ We need to change the authentication method from ident, to md5:

```
# "local" is for Unix domain socket connections only
local  all          all          peer
# IPv4 local connections:
host   zabbix       zabbix      127.0.0.1/32    md5
# IPv6 local connections:
host   all          all          ::1/128        ident
```

- ☑ Note that we also restricted the connection permissions to our specific database and user. It is also possible to restrict the address.

ENGLISH

ZABBIX 
MEETUP ONLINE '20

PostgresPro



02



TUNING POSTGRESQL FOR ZABBIX WORKLOADS

MODIFYING THE POSTGRESQL DB
CONFIGURATION FOR ZABBIX
SPECIFIC SQL WORKLOADS



TUNING ZABBIX GATHERING PROCESSES

- ⊗ Depending on the size of our instance, we need change the number of different internal processes:
- ⊗ StartPollers
- ⊗ StartPingers
- ⊗ StartTrappers
- ⊗ StartDBSyncers
- ⊗ And many others!

ENGLISH

ZABBIX 

MEETUP ONLINE '20

PosgresPro



TUNING ZABBIX GATHERING PROCESSES

- ⊘ Seems simple enough, right? Let's raise our pollers to 200! (Imagine this is a large instance)

```
### Option: StartPollers  
StarPollers=200
```

```
[Z3001] connection to database 'zabbix' failed: [0] FATAL: remaining  
connection slots are reserved for non-replication superuser connections  
  
database is down: reconnecting in 10 seconds
```

ENGLISH

ZABBIX 
MEETUP ONLINE '20

PostgresPro



ADJUST MAX_CONNECTIONS

- 🔍 Looks like we have exhausted our max_connections. But how come?

```
### Option: StartDBSyncers  
StartDBSyncers=4
```

- 🔍 Only 4 History syncer processes connecting to the DB
- 🔍 Looks like pollers connect to the DB too!

```
### Option: StartPollers  
StarPollers=200
```

- 🔍 Pollers do connect to the DB! (Fixed in 5.4 - ZBXNEXT-782)

ENGLISH

ZABBIX 
MEETUP ONLINE '20


PostgresPro



ADJUST MAX_CONNECTIONS (2)

- ⊗ Don't forget about the Web Backend!
- ⊗ For php-fpm:

```
pm.max_children = 50
```

- ⊗ That is extra 50 potential connections to the DB!
- ⊗ Estimate the max connections accordingly:

```
max_connections = 300
```

ENGLISH

ZABBIX 

MEETUP ONLINE '20

PosgresPro



KEY PERFORMANCE CONSIDERATIONS

- ⊗ What about other parameters – Buffers, caches, workers etc?
- ⊗ It can be very complex to estimate these!
- ⊗ Database size and hardware performance has a very large impact on the optimal configuration parameters!
- ⊗ Most precise results come from trial and error approaches – but this is also very time consuming!
- ⊗ We can also go by general best practice
- ⊗ Third party tools are available for generating the potential configuration parameters

ENGLISH

ZABBIX 
MEETUP ONLINE '20

 **PostgresPro**



KEY PERFORMANCE CONSIDERATIONS

- ☑ We can estimate these database configuration parameters by using best practice approaches
- ☑ Or we can also use third party tools. For example, PG Tune:

Home [How it works](#) [light](#)

PG Tune

Parameters of your system

DB version what is this?
12

OS Type what is this?
Linux

DB Type what is this?
Mixed type of applications

Total Memory (RAM) what is this?
32 GB

Number of CPUs what is this?
8

Number of Connections what is this?
200

Data Storage what is this?
SSD storage

[Generate](#)

postgresql.conf ALTER SYSTEM

Add/modify this settings in **postgresql.conf** and restart database

```
# DB Version: 12
# OS Type: linux
# DB Type: mixed
# Total Memory (RAM): 32 GB
# CPUs num: 8
# Connections num: 200
# Data Storage: ssd

max_connections = 200
shared_buffers = 8GB
effective_cache_size = 24GB
maintenance_work_mem = 2GB
checkpoint_completion_target = 0.9
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 1.1
effective_io_concurrency = 200
work_mem = 5242kB
min_wal_size = 1GB
max_wal_size = 4GB
max_worker_processes = 8
max_parallel_workers_per_gather = 4
max_parallel_workers = 8
max_parallel_maintenance_workers = 4
```



TUNING POSTGRESQL PARAMETERS

- ☑ Let's take a look at how we should tune PostgreSQL buffers and caches:

shared_buffers - 25% of your RAM. PostgreSQL data buffer.

effective_cache_size - How much memory you expected to be available on the OS. Used for query planning! Usually around 50% of your RAM.

huge_pages – force, try or disable using huge pages.

- ☑ Using huge pages reduces the overhead of page management
- ☑ This is highly recommended in very many PostgreSQL use cases!

ENGLISH

ZABBIX 
MEETUP ONLINE '20

 **PostgresPro**



TUNING POSTGRESQL PARAMETERS

🕒 Next, let's configure memory related parameters

maintenance_work_mem - memory used by maintenance procedures such as vacuums, index creation, alter table, etc.

Maintenance operations like vacuums should be performed on a scheduled basis and also before performing an upgrade.

work_mem - used for in memory sorts. This is defined per a single sort.

max_connections*work_mem – potential memory ceiling for sorts.

ENGLISH

ZABBIX 

MEETUP ONLINE '20

PosgresPro



TUNING POSTGRESQL PARAMETERS

- ☑ When it comes to concurrency, worker processes can help us out

`max_worker_processes` – workers for maintenance and parallel queries

`max_parallel_workers` – max workers for parallel operations

`max_parallel_workers_per_gather` – max parallel workers per query

`max_parallel_maintenance_workers` – max parallel workers for maintenance tasks

- ☑ Max worker processes/parallel workers = count of CPU cores

- ☑ Workers per gather/Maintenance workers = Depending on count of cores. Usually ½ or ¼

ENGLISH



MEETUP ONLINE '20



ENABLING AND CONFIGURING VACUUM

- ☑ To clean up dead tuples and free up disk space, vacuum must be executed
- ☑ We can automate this by configuring autovacuum

```
autovacuum=on #turn on autovacuum
vacuum_cost_page_hit = 1 #cost of work if page in buffer
vacuum_cost_page_miss = 10 #cost of work if page not in buffer
vacuum_cost_page_dirty = 20 #cost of work if page is cleaned up
autovacuum_vacuum_threshold = 50 #threshold to trigger vacuum
autovacuum_vacuum_scale_factor = 0.01
autovacuum_vacuum_cost_delay = 20ms #delay after reaching limit
autovacuum_vacuum_cost_limit = 3000 #limit of vacuum cost
autovacuum_max_workers = 6 #number of vacuum processes
```



ENABLING AND CONFIGURING VACUUM

- ✓ The cost limit is global – applies to all workers!
- ✓ Each worker gets only $1/\text{autovacuum_max_workers}$ of the total cost limit
- ✓ Increasing the cost limit is a simple way to improve vacuum performance (if the hardware limitations permit it)

ENGLISH

ZABBIX 
MEETUP ONLINE '20

 **PostgresPro**



ENGLISH



THANK YOU!

