

Documentation

Zabbix

ZABBIX

Warning

Cette documentation a été rédigée pour un serveur Zabbix 7.x avec la configuration suivante : | Ubuntu 24.04LTS | PostgreSQL | Nginx

Backup

DB



Avant d'initier une backup sur la DB postgres, il est préférable de stopper le service du serveur Zabbix afin d'éviter une surcharge des transactions sur la DB.

```
systemctl stop zabbix-server
```

Par la suite, il est possible de faire une backup complète de la base de données (Option 1), ou bien de faire une backup excluant les plus grosses tables (Option 2). Ce choix dépend de l'objectif et de votre RTO.

Option 1 :

```
sudo -u zabbix pg_dump zabbix -h localhost -F c -f zabbix_backup.sql zabbix
```

Option 2 :

```
sudo -u zabbix pg_dump --exclude-table-data=history* --exclude-table-data=trends* zabbix
```

Configuration web et Zabbix


```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/ #pas nécessaire si pas d'installation Apache2
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

Restore

DB

Note

Cette section est partiellement basée sur [cet article](#) du blog Zabbix officiel.

Afin de restaurer une DB postgresql zabbix, il est nécessaire d'avoir postgresql installé (v15, v16...) sur le serveur (ici Ubuntu 24.04) et d'avoir créé une DB "zabbix" vide.

```
sudo -u postgres createuser --pwprompt zabbix #si l'utilisateur n'est pas déjà existant (après un DR par exemple)
sudo -u postgres createdb -O zabbix zabbix
```

A partir de cela, il est possible de restaurer le dump SQL avec la commande suivante :abbr:

```
psql zabbix < zabbix_db_2025-11-30_03-00-01.sql
```

Update

Upgrade between minor versions

Il est possible de mettre Zabbix à jour entre des versions 7.4.x mineures (for example, from 7.4.1 to 7.4.3).

Il est d'abord nécessaire de mettre à jour les infos du repository :

```
sudo apt update
```

Ensuite, il suffit seulement de mettre le serveur à jour aussi avec apt :

```
sudo apt install --only-upgrade 'zabbix*'
```

Installation and Configuration of TimescaleDB

Note

PostgreSQL version 16.11

Sources

- [Tiger Data Docs](#)
- [Tillnet.se](#)
- [Zabbix Docs](#)

TimescaleDB agit comme un plug-in s'ajoutant à une base de données Zabbix déjà existante.

Ce plug-in permet à la fois d'améliorer les performances de housekeeping du serveur en question tout en limitant l'espace de stockage utilisé (grâce notamment à la compression).

Installation

Afin d'installer l'extension TimescaleDB sur la base de données postgres existante, il est nécessaire de se connecter à l'interface shell du serveur.

Ajout du repository et de la GPG

```
echo "deb https://packagecloud.io/timescale/timescaledb/ubuntu/ $(lsb_release -c -s) main" | sudo tee /etc/apt/sources.list.d/timescaledb.list
wget --quiet -O - https://packagecloud.io/timescale/timescaledb/gpgkey | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/timescaledb.gpg
sudo apt update
```

Installation du paquet

Il est possible d'installer le paquet TimescaleDB pour Zabbix de 2 manières différentes :

- Option 1 : Installer la dernière version de TimescaleDB mais en changeant la configuration de Zabbix
- Option 2 : Installer la dernière version de TimescaleDB supportée par Zabbix

Option 1

```
sudo apt install timescaledb-2-postgresql-16 timescaledb-2-loader-postgresql-16 postgresql-client-16 -y
```

Option 2

```
sudo apt install timescaledb-2-postgresql-16='2.18.0*' timescaledb-2-loader-postgresql-16='2.18.0*' postgresql-client-16
```

Configuration

```
timescaledb-tune --max-conns=125
```

L'option "--max-conns=125" permet d'augmenter les connexions simultanées à la base de données. (Par défaut à 100)

Lors du setup, il est recommandé de répondre "Yes" à toutes les questions. Le tuning permet l'optimisation de postgres pour l'utilisation de l'extension TimescaleDB.

```
Is this correct? [(y)es/(n)o]: y
Writing backup to:
/tmp/timescaledb_tune.backup202512100841

shared_preload_libraries needs to be updated
Current:
#shared_preload_libraries = ''
Recommended:
shared_preload_libraries = 'timescaledb'
Is this okay? [(y)es/(n)o]: y
success: shared_preload_libraries will be updated

Tune memory/parallelism/WAL and other settings? [(y)es/(n)o]: y
Recommendations based on 3.82 GB of available memory and 2 CPUs for PostgreSQL 16

Memory settings recommendations
Current:
shared_buffers = 128MB
#effective_cache_size = 4GB
#maintenance_work_mem = 64MB
#work_mem = 4MB
Recommended:
shared_buffers = 1002346kB
effective_cache_size = 2936MB
maintenance_work_mem = 501173kB
work_mem = 4009kB
Is this okay? [(y)es/(s)kip/(q)uit]: y
success: memory settings will be updated

Parallelism settings recommendations
Current:
missing: timescaledb.max_background_workers
#max_worker_processes = 8
#max_parallel_workers_per_gather = 2
#max_parallel_workers = 8
Recommended:
timescaledb.max_background_workers = 16
max_worker_processes = 21
max_parallel_workers_per_gather = 1
max_parallel_workers = 2
Is this okay? [(y)es/(s)kip/(q)uit]: y
success: parallelism settings will be updated

WAL settings recommendations
Current:
#wal_buffers = -1
min_wal_size = 80MB
Recommended:
wal_buffers = 16MB
min_wal_size = 512MB
Is this okay? [(y)es/(s)kip/(q)uit]: y
success: WAL settings will be updated

Background writer settings recommendations
success: background writer settings are already tuned

Miscellaneous settings recommendations
Current:
#default_statistics_target = 100
#random_page_cost = 4.0
checkpoint_completion_target = 0.9
max_connections = 100
#max_locks_per_transaction = 64
#autovacuum_max_workers = 3
#autovacuum_naptime = 1min
#default_toast_compression = 'pglz'
#jit = on
#effective_io_concurrency = 1
Recommended:
default_statistics_target = 100
random_page_cost = 1.1
checkpoint_completion_target = 0.9
max_connections = 125
max_locks_per_transaction = 128
autovacuum_max_workers = 10
autovacuum_naptime = 10
default_toast_compression = lz4
jit = off
effective_io_concurrency = 256
Is this okay? [(y)es/(s)kip/(q)uit]: y
```

Après l'exécution du setup avec succès, le service postgres nécessite d'être redémarré.

```
sudo systemctl restart postgresql
```

Activation de l'extension

Activer TimescaleDB pour la base de données "zabbix".

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql --dbname=zabbix
```

Migration des tables

Warning

Selon la taille de la base de données, la migration des tables peut prendre beaucoup de temps !

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

Le script "schema.sql" paramètre le housekeeping comme suit :

- Override item history period
- Override item trend period

Ces paramètres doivent impérativement être activés pour utiliser le partitionnement TimescaleDB des tables "history" et "trends". Il est cependant possible de l'activer seulement pour l'un ou pour l'autre individuellement.

2 autres paramètres sont aussi configurés grâce au script :

- Enable compression
- Compress records older than 7 days

Ces paramètres sont trouvables dans la web GUI via Administration > Housekeeping

ZABBIX <> ☰

SRV-MON01

- Dashboards
- Monitoring
- Services
- Inventory
- Reports
- Data collection
- Alerts
- Users
- Administration**
 - General
 - Audit log
 - Housekeeping**
 - Proxy groups
 - Proxies
 - Macros
 - Queue
- Support
- Integrations
- Help
- User settings
- Sign out

Housekeeping

Events and alerts

Enable internal housekeeping

* Trigger data storage period

* Service data storage period

* Internal data storage period

* Network discovery data storage period

* Autoregistration data storage period

Services

Enable internal housekeeping

* Data storage period

User sessions

Enable internal housekeeping

* Data storage period

History

Enable internal housekeeping

Override item history period

* Data storage period

Trends

Enable internal housekeeping

Override item trend period

* Data storage period

History, trends and audit log compression

Enable compression

* Compress records older than

Audit log

Audit settings

Update **Reset defaults**