ZABBIX

RHEIN-MAIN SOLUTIONS

Zabbix Database

Configuration Backup with Change Data Capturing

Zabbix Conference Germany 2023

Thomas Oftring – Zabbix Team Lead/Zabbix Certified Trainer

Speaker Information

- Thomas Oftring
- Zabbix Team Lead
- Zabbix Certified Trainer
- Database Architect

https://www.rhein-main-solutions.com/zabbix/

Thomas.Oftring@Rhein-Main-Solutions.com

Anforderung

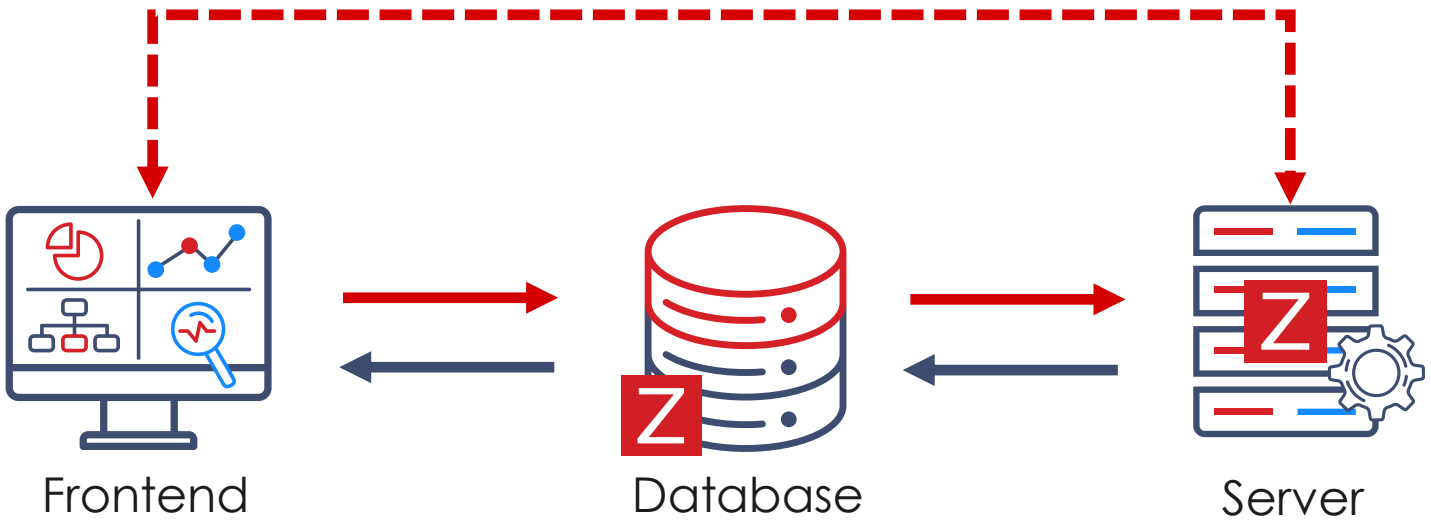
- Backup der Zabbix Konfiguration
- Konsistenter Zustand
- Schnelle Wiederherstellung für Testumgebungen
- Trennung der Konfiguration von den gesammelten Daten
- Flexibilität welche Konfigurationen gesichert werden sollen

Welche Konfigurationen hat Zabbix

Zabbix Config

- Dashboards
- Network Maps
- Services und SLAs
- Scheduled Reports
- Configurations Data Collection
- Configurations Alerts
- Configurations Administration

Die Zabbix Datenbank



Backup mit Hilfe von Export/Import

Vorteile

- Direkt aus der Weboberfläche von Zabbix verfügbar
- Textdateien (YAML, XML, JSON)
- Versionsverwaltung möglich (z. B. mit Git)
- Einfacher Import durch Benutzer möglich

Nachteile

- Nicht alle Konfigurationen können exportiert werden (z. B. Dashboards)
- Muss immer manuell durch Benutzer erfolgen
- Momentaufnahme
- keine Konsistenz der Konfigurationen
- Wiederherstellung einer Datenbank für Testumgebung oder Neustart nicht möglich

Backup mit Hilfe Zabbix API

Vorteile	Nachteile
• Zabbix API ist sehr gut dokumentiert (Configuration export/import)	• Nicht alle Konfigurationen können exportiert werden (z. B. Dashboards)
• Programmatischer Export möglich	• Benötigt Programmierwissen
• Textdateien (YAML, XML, JSON)	• Momentaufnahme
• Versionsverwaltung möglich (z. B. mit Git)	• keine Konsistenz der Konfigurationen
• Einfacher Import durch Benutzer möglich	• Wiederherstellung einer Datenbank für Testumgebung oder Neustart nicht möglich

Backup mit Hilfe von pgcopydb

Vorteile

- Konsistente Momentaufnahmen möglich
- Konsistente Übernahme von Änderungen durch Change Data Capturing
- Schnelles Backup und Restore für Testumgebungen möglich

Nachteile

- Versionsverwaltung nicht möglich (z. B. mit Git)
- Logische Replikation muss auf Zabbix DB System aktiviert werden
- Benötigt Datenbankwissen im Bereich PostgreSQL
- Benötigt ein zusätzliches System für die PostgreSQL Instanz für das Backup

10 Schritte für Clone mit pgcopydb

1. Schema transferieren mit Hilfe von pg_dump
2. Informationen aus dem pg_catalog auslesen
3. Restore pre-data (Filtermöglichkeiten)
4. Paralleler Import der Daten mit dem COPY Befehl
5. Extra Schritt für BLOB Daten
6. Parallele Erstellung der Indexe auf den Tabellen
7. Erstellen der Primary Keys
8. Ausführen von VACUUM ANALYZE
9. Sequences abgleichen
10. Restore post-data des Schemas (z. B. Constraints)

7 Schritte für Follow mit pgcopydb

1. Erstellen eines Datenbank Snapshots
2. Erstellen eines Logischen Replikations-Slots
3. Ausführen von pgcopydb Clone (Vorgänger Folie)
4. Änderungen fortlaufend übernehmen (Logical Replication)
5. Sequences erneut abgleichen und synchronisieren
6. Aufräumen der Change Data Caption Resourcen aus Punkt 2
7. Snapshot freigeben

Informationen zu pgcopydb

1. PostgreSQL Open Source License
2. Maintainer Dimitri Fontaine (PostgreSQL Major Contributor)
3. Github pgcopydb
4. Dokumentation von pgcopydb
5. Verfügbar über die PostgreSQL Standard Repositories für die gängigen Linux Distributionen
6. Baut auf bewährte Standard-Tools pg_dump/pg_restore auf und erweitert den Funktionsumfang

Vergleich pgcopydb vs. pg_dump/pg_restore

Feature	pgcopydb	pg_dump ; pg_restore
Single-command operation	✓	✗
Snapshot consistency	✓	✓
Ability to resume partial run	✓	✗
Advanced filtering	✓	✓
Tables concurrency	✓	✓
Same-table concurrency	✓	✗
Index concurrency	✓	✓
Constraint index concurrency	✓	✗
Schema	✓	✓
Large Objects	✓	✓
Vacuum Analyze	✓	✗
Copy Freeze	✓	✗
Roles	✓	✗ (needs pg_dumpall)
Tablespaces	✗	✗ (needs pg_dumpall)
Follow changes	✓	✗

Feature Matrix pgcopydb

Installation von pgcopydb

1. PostgreSQL Standard Repositories für die gängigen Linux Distributionen
   a) Debian/Ubuntu
      sudo apt-get install pgcopydb
   b) RPM based systems(Versionsauswahl notwendig)
      dnf install pgcopydb_15

2. Docker Image

   $ docker pull ghcr.io/dimitri/pgcopydb:latest

   $ docker run --rm -it ghcr.io/dimitri/pgcopydb:latest pgcopydb --version

   $ docker run --rm -it ghcr.io/dimitri/pgcopydb:latest pgcopydb --help

Demo Install pgcopydb

Installation von pgcopydb

```
zabbix@confbak:~$ sudo apt info pgcopydb
[sudo] password for zabbix:
Package: pgcopydb
Version: 0.13-1.pgdg22.04+1
Priority: optional
Section: database
Maintainer: Dimitri Fontaine <dim@tapoueh.org>
Installed-Size: 714 kB
Depends: postgresql-client, libc6 (>= 2.34), libpq5 (>= 9.1~)
Homepage: https://github.com/dimitri/pgcopydb
Download-Size: 250 kB
APT-Manual-Installed: yes
APT-Sources: https://apt.postgresql.org/pub/repos/apt jammy-pgdg/main arm64 Packages
Description: Copy an entire PostgreSQL database from source to target
 This tool copies an entire PostgreSQL database from source to target. It
 implements `pg_dump | pg_restore` on steroids, including advanced concurrency
 tricks to make the operation faster.

zabbix@confbak:~$ sudo apt install -y pgcopydb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
pgcopydb is already the newest version (0.13-1.pgdg22.04+1).
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
zabbix@confbak:~$ sudo mkdir -p /opt/pgcopydb_wd && sudo chmod 777 /opt/pgcopydb_wd
zabbix@confbak:~$ pgcopydb version
07:29:37 26322 INFO   Running pgcopydb version 0.13-1.pgdg22.04+1 from "/usr/bin/pgcopydb"
pgcopydb version 0.13-1.pgdg22.04+1
compiled with PostgreSQL 15.3 (Ubuntu 15.3-1.pgdg22.04+1) on aarch64-unknown-linux-gnu, compiled by gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0, 64-bit
compatible with Postgres 10, 11, 12, 13, 14, and 15
zabbix@confbak:~$
```

Installation von pgcopydb

```
zabbix@confbak:~$ pgcopydb follow --help
07:29:48 26326 INFO   Running pgcopydb version 0.13-1.pgdg22.04+1 from "/usr/bin/pgcopydb"
pgcopydb follow: Replay changes from the source database to the target database
usage: pgcopydb follow  --source ... --target ...

  --source              Postgres URI to the source database
  --target              Postgres URI to the target database
  --dir                 Work directory to use
  --filters <filename>  Use the filters defined in <filename>
  --restart             Allow restarting when temp files exist already
  --resume              Allow resuming operations after a failure
  --not-consistent      Allow taking a new snapshot on the source database
  --snapshot            Use snapshot obtained with pg_export_snapshot
  --plugin              Output plugin to use (test_decoding, wal2json)
  --slot-name           Use this Postgres replication slot name
  --create-slot         Create the replication slot
  --origin              Use this Postgres replication origin node name
  --endpos              Stop replaying changes when reaching this LSN

zabbix@confbak:~$
```

Filter in pgcopydb

1. Ausgeprägte Filtermöglichkeiten
   a) Schema
      - include-only-schema
      - exclude-schema
   b) Table
      - include-only-table
      - exclude-table
   c) Table Data
      - exclude-table-data
   d) Index
      - exclude-index

```
 1 [exclude-table-data]
 2 public.acknowledges
 3 public.alerts
 4 public.escalations
 5 public.event_recovery
 6 public.event_suppress
 7 public.event_symptom
 8 public.event_tag
 9 public.events
10 public.history
11 public.history_log
12 public.history_str
13 public.history_text
14 public.history_uint
15 public.problem
16 public.problem_tag
17 public.trends
18 public.trends_uint
```

Demo filter pgcopydb

Clone with Follow pgcopydb

```
export PGCOPYDB_SOURCE_PGURI="port=5432 host=confprod dbname=zabbix\ user=postgres password=Zabbix11!"

export PGCOPYDB_TARGET_PGURI="port=5432 host=confbak dbname=zabbix\ user=postgres password=Zabbix11!“

pgcopydb ping


pgcopydb clone --follow  --table-jobs 8 \

--index-jobs 8 –filter pgcopydb.zabbix.filter &

pgcopydb stream sentinel set endpos --current

pgcopydb stream cleanup
```

Change Data Capture

Demo Clone with Follow

Clone with Follow pgcopydb

```
postgres=# \l
                                                List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+-----------------------
 postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
(3 rows)

postgres=# CREATE DATABASE zabbix OWNER zabbix;
CREATE DATABASE
postgres=# \l
                                                List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+-----------------------
 postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 zabbix    | zabbix   | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
(4 rows)

postgres=#
```

```
[0] 0:confprod (local)- 1:confbak (remote)*                                                    "confprod" 11:56 17-Nov-23
```

Clone with Follow pgcopydb

```
zabbix@confprod:~/demo_pgcopydb$ export PGCOPYDB_SOURCE_PGURI="port=5432 host=confprod dbname=zabbix user=postgres password=Zabbix11!"
zabbix@confprod:~/demo_pgcopydb$ export PGCOPYDB_TARGET_PGURI="port=5432 host=confbak dbname=zabbix user=postgres password=Zabbix11!"
zabbix@confprod:~/demo_pgcopydb$ printenv |egrep -i PGCOPYDB_
PGCOPYDB_SOURCE_PGURI=port=5432 host=confprod dbname=zabbix user=postgres password=Zabbix11!
PGCOPYDB_TARGET_PGURI=port=5432 host=confbak dbname=zabbix user=postgres password=Zabbix11!
zabbix@confprod:~/demo_pgcopydb$
[0] 0:confprod (local)* 1:confbak (remote)-                                    "confprod" 11:58 17-Nov-23
```

Clone with Follow pgcopydb

```
zabbix@confprod:~/demo_pgcopydb$ pgcopydb clone --follow --table-jobs 8 --index-jobs 8 --filter pgcopydb.zabbix.filter &
[1] 47025
zabbix@confprod:~/demo_pgcopydb$ 11:58:27 47025 INFO   Running pgcopydb version 0.13-1.pgdg22.04+1 from "/usr/bin/pgcopydb"
11:58:27 47025 INFO   [SOURCE] Copying database from "postgres://postgres@confprod:5432/zabbix"
11:58:27 47025 INFO   [TARGET] Copying database into "postgres://postgres@confbak:5432/zabbix"
11:58:27 47025 INFO   Created logical replication slot "pgcopydb" with plugin "test_decoding" at 0/F531130 and exported snapshot 00000007-0001FA6A-1
11:58:27 47025 INFO   create schema if not exists pgcopydb
11:58:27 47025 INFO   drop table if exists pgcopydb.sentinel
11:58:27 47025 WARN   NOTICE:  table "sentinel" does not exist, skipping
11:58:27 47025 INFO   create table pgcopydb.sentinel(startpos pg_lsn, endpos pg_lsn, apply bool,  write_lsn pg_lsn, flush_lsn pg_lsn, replay_lsn pg_lsn)
11:58:27 47025 INFO   Created logical replication origin "pgcopydb" at LSN 0/F531130
11:58:27 47029 INFO   STEP 1: fetch source database tables, indexes, and sequences
11:58:27 47029 INFO   Fetched information for 1 extensions
11:58:27 47033 INFO   Resuming streaming at LSN 0/F531130 from replication slot "pgcopydb"
11:58:27 47035 INFO   Waiting until the pgcopydb sentinel apply is enabled
11:58:27 47029 INFO   Fetched information for 169 tables, with an estimated total of 138 456 tuples and 28 MB
11:58:27 47029 INFO   Fetched information for 421 indexes
11:58:27 47029 INFO   Fetching information for 4 sequences
11:58:27 47033 INFO   Reported write_lsn 0/F531130, flush_lsn 0/F531130, replay_lsn 0/0
11:58:27 47029 INFO   STEP 2: dump the source database schema (pre/post data)
11:58:27 47029 INFO    /usr/bin/pg_dump -Fc --snapshot 00000007-0001FA6A-1 --section pre-data --file /tmp/pgcopydb/schema/pre.dump postgres://postgres@confprod:54
32/zabbix
11:58:27 47029 INFO    /usr/bin/pg_dump -Fc --snapshot 00000007-0001FA6A-1 --section post-data --file /tmp/pgcopydb/schema/post.dump postgres://postgres@confprod:
5432/zabbix
11:58:27 47029 INFO   STEP 3: restore the pre-data section to the target database
11:58:27 47029 INFO    /usr/bin/pg_restore --dbname postgres://postgres@confbak:5432/zabbix --single-transaction --use-list /tmp/pgcopydb/schema/pre.list /tmp/pgc
opydb/schema/pre.dump
11:58:27 47029 INFO   STEP 6: starting 8 CREATE INDEX processes
11:58:27 47029 INFO   STEP 7: constraints are built by the CREATE INDEX processes
11:58:27 47029 INFO   STEP 8: starting 8 VACUUM processes
11:58:27 47029 INFO   STEP 5: copy Large Objects (BLOBs) in 1 sub-process
11:58:27 47029 INFO   STEP 9: reset sequences values
11:58:27 47066 INFO   Reset sequences values on the target database
11:58:27 47048 INFO   STEP 4: starting 8 table data COPY processes
11:58:27 47065 INFO   Copying large objects
[0] 0:confprod (local)* 1:confbak (remote)-                                                                    "confprod" 11:58 17-Nov-23
```

Clone with Follow pgcopydb

```
22639 | public |               report_user |         73ms |      4 |               192ms
22648 | public |             report_usrgrp |         69ms |      4 |               128ms
21320 | public |                     rights |         84ms |      3 |               118ms
22857 | public |                scim_group |         78ms |      2 |                96ms
21740 | public |            service_alarms |         61ms |      3 |               108ms
22667 | public |           service_problem |         70ms |      3 |               128ms
22685 | public |       service_status_rule |         59ms |      2 |                71ms
21345 | public |            services_links |         94ms |      3 |               137ms
22733 | public |     sla_excluded_downtime |        120ms |      2 |                74ms
22725 | public |             sla_schedule |         93ms |      2 |                74ms
22247 | public |    sysmap_element_trigger |         93ms |      3 |               104ms
21474 | public |              sysmap_user |         88ms |      3 |               141ms
21482 | public |             sysmap_usrgrp |         81ms |      3 |               119ms
21446 | public |     sysmaps_link_triggers |         97ms |      3 |               121ms
22146 | public |                      task |         95ms |      3 |               103ms
22218 | public |          task_acknowledge |         97ms |      1 |                55ms
22349 | public |            task_check_now |         90ms |      1 |                37ms
22156 | public |       task_close_problem |         89ms |      1 |                48ms
21504 | public |             timeperiods |        103ms |      1 |                45ms
22527 | public |            trigger_queue |         71ms |      1 |                32ms
22864 | public |          user_scim_group |         51ms |      3 |               132ms
22838 | public |    userdirectory_idpgroup |         77ms |      3 |               124ms
22821 | public |       userdirectory_media |         60ms |      3 |               120ms
22830 | public |      userdirectory_usrgrp |         62ms |      4 |               200ms


                                        Step   Connection    Duration   Concurrency
-------------------------------------------   ----------   ----------   ------------
                                 Dump Schema       source        98ms             1
  Catalog Queries (table ordering, filtering, etc)       source       237ms             1
                               Prepare Schema       target       244ms             1
    COPY, INDEX, CONSTRAINTS, VACUUM (wall clock)         both       4s515         8 + 16
                              COPY (cumulative)         both         14s             8
                    Large Objects (cumulative)         both        52ms             1
           CREATE INDEX, CONSTRAINTS (cumulative)       target        24s             8
                              Finalize Schema       target       264ms             1
-------------------------------------------   ----------   ----------   ------------
                       Total Wall Clock Duration         both       5s373         8 + 16
-------------------------------------------   ----------   ----------   ------------

[0] 0:confprod (local)* 1:confbak (remote)-                                  "confprod" 11:58 17-Nov-23
```

Clone with Follow pgcopydb

```
zabbix@confprod:~/demo_pgcopydb$ pgcopydb list progress
11:59:44 47106 INFO   Running pgcopydb version 0.13-1.pgdg22.04+1 from "/usr/bin/pgcopydb"
11:59:44 47106 INFO   A previous run has run through completion
             |  Total Count |  In Progress |         Done
-------------+--------------+--------------+-------------
      Tables |          169 |            0 |          169
     Indexes |          421 |            0 |          421
zabbix@confprod:~/demo_pgcopydb$
```

```
[0] 0:confprod (local)* 1:confbak (remote)-                                    "confprod" 12:00 17-Nov-23
```

Clone with Follow pgcopydb

```
 hostid |                    host                    | status
--------+--------------------------------------------+--------
  10602 | demotest02                                 |      0
  10084 | Zabbix server                              |      0
  10603 | demoserver01                               |      0
  10333 | {#HV.UUID}                                 |      0
  10334 | {#VM.UUID}                                 |      0
  10367 | {#HV.UUID}                                 |      0
  10368 | {#VM.DNS}                                  |      0
  10388 | {#HOSTNAME}                                |      0
  10389 | {#HOSTNAME}                                |      0
  10595 | Acronis CPC MSP {#SCOPE}                   |      0
  10533 | Azure virtual machine {#NAME}              |      0
  10511 | API {#NAME}                                |      0
  10512 | Controller manager {#NAME}                 |      0
  10513 | Scheduler {#NAME}                          |      0
  10514 | Kubelet {#NAME}                            |      0
  10523 | Consul {#NODE_NAME}                        |      0
  10536 | {#AWS.EC2.INSTANCE.ID}                     |      0
  10537 | {#AWS.RDS.INSTANCE.ID}                     |      0
  10538 | {#AWS.S3.NAME}                             |      0
  10541 | Azure MySQL server {#NAME}                 |      0
  10545 | Azure PostgreSQL server {#NAME}            |      0
  10549 | {#SERIAL}                                  |      0
  10550 | {#NAME}                                    |      0
  10554 | {#SERVER.NAME}                             |      0
  10559 | Azure Microsoft SQL database {#NAME}       |      0
  10567 | {#NAME}                                    |      1
  10569 | Azure Cosmos DB {#NAME}                    |      0
  10578 | {#GCP.PROJECT.ID}-{#CLOUD_SQL.INSTANCE.NAME} |      0
  10579 | {#GCP.PROJECT.ID}-{#CLOUD_SQL.INSTANCE.NAME} |      0
  10580 | {#GCP.PROJECT.ID}-{#CLOUD_SQL.INSTANCE.NAME} |      0
  10581 | {#GCE.INSTANCE.ID}                         |      0
  10585 | {#AWS.ECS.CLUSTER.NAME}                    |      0
  10588 | OpenStack {#SERVICE_NAME}                  |      0
  10592 | {#UUID}                                    |      0
  10600 | {#CLIENT.ID}                               |      0
  10601 | {#SERVER.ID}                               |      0
(36 rows)

zabbix=#
[0] 0:confprod (local)- 1:confbak (remote)*                                  "confprod" 12:00 17-Nov-23
```

ZABBIX

RHEIN-MAIN SOLUTIONS

Vielen Dank
Für Ihre Aufmerksamkeit!

Zabbix Conference Germany 2023

Thomas Oftring – Zabbix Team Lead/Zabbix Certified Trainer