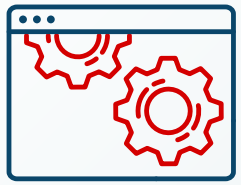
ZABBIX '25
CONFERENCE
GERMANY

Deep Dive into TimescaleDB Architecture, Tuning and Monitoring in Zabbix

Jochen Weschta

Senior Consultant Infrastructure Architect
Zabbix Certified Trainer

SCADIX GmbH

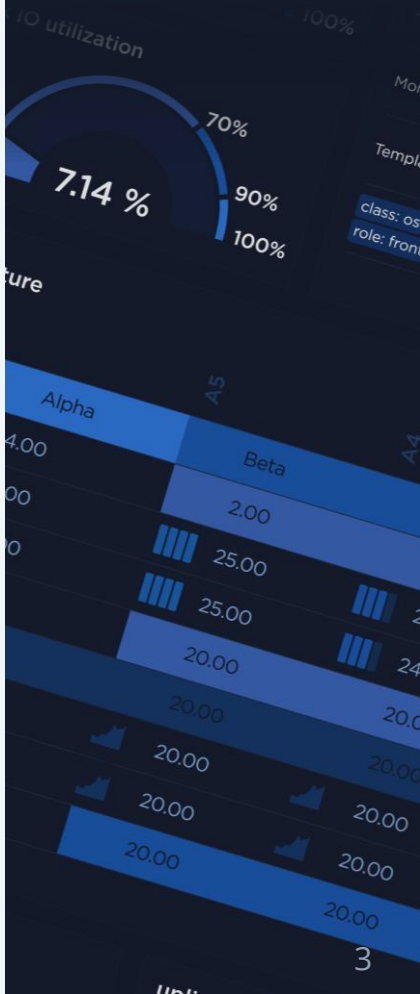


Einleitung

Einstieg – Das Problem

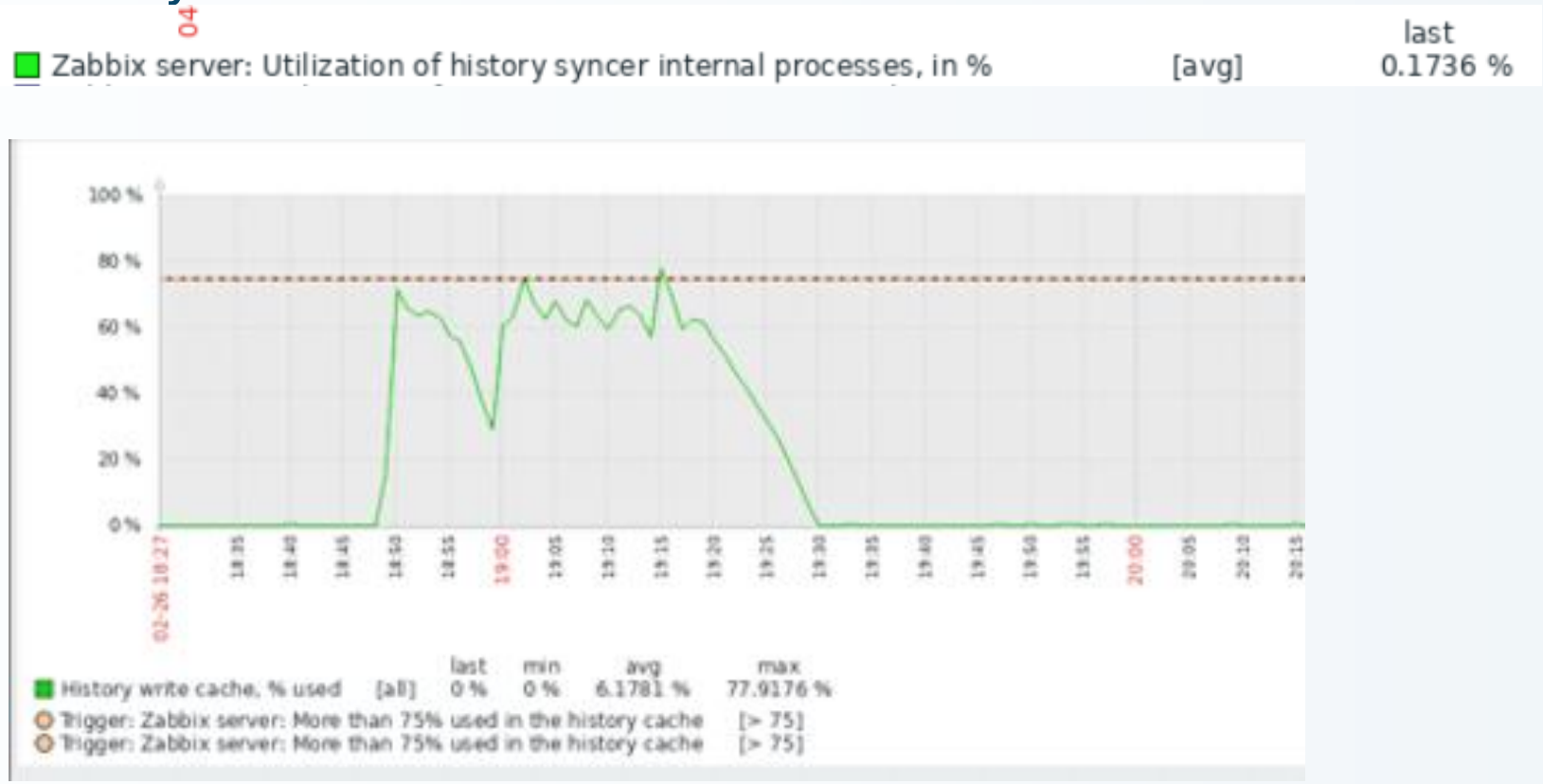
- Dashboards *laden extrem langsam*.

```
********************* Script profiler *********************
Total time: 63.063377
Total SQL time: 62.045856
SQL count: 116 (selects: 58 | executes: 72)
Peak memory usage: 4M
Memory limit: 128M

1.   role.get [CRoleHelper.php:183]
...
```



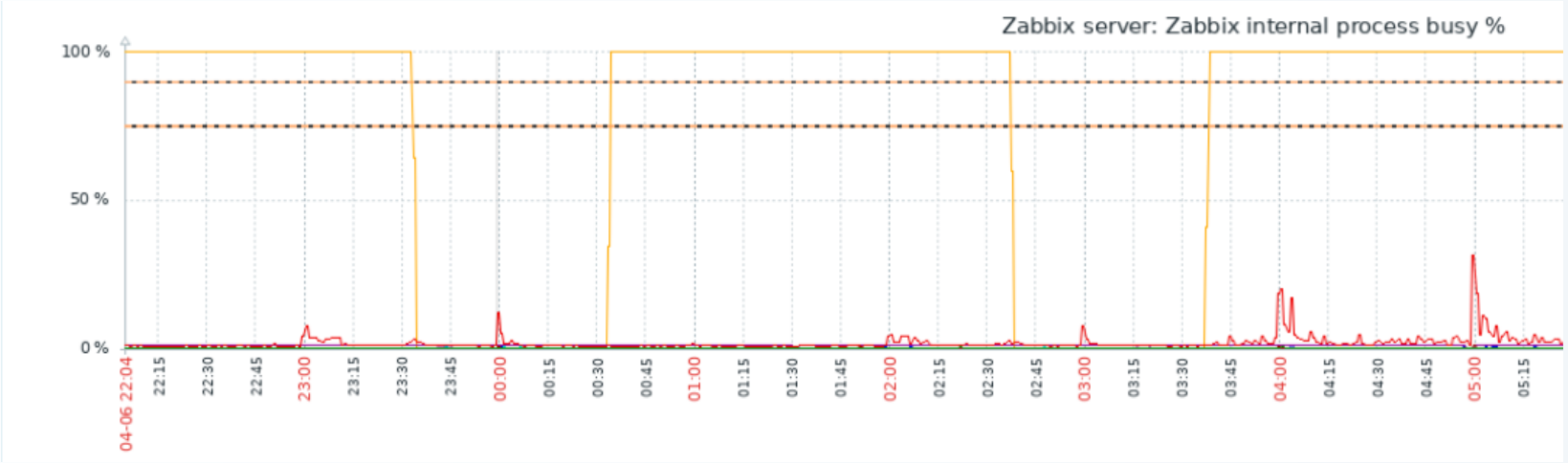
Einstieg – Das Problem

- History-Write Cache läuft voll.



Einstieg – Das Problem

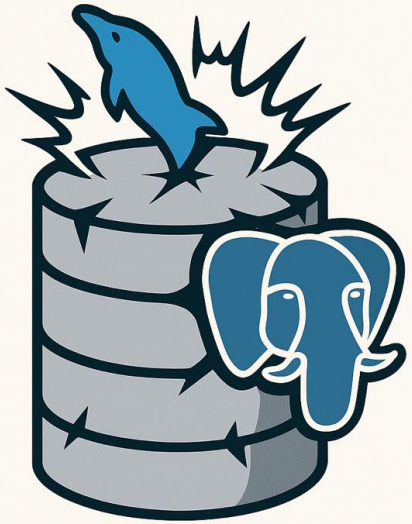
- Housekeeper Laufzeit zu lang > 1h und wird länger.



Einstieg – Das Problem

- Datenbank wird zu groß
  - Backuplaufzeiten zu lang
  - Speicherplatzbedarf steigt an
  - Restore Mechanismus dauert zu lange

=> Unhandlich

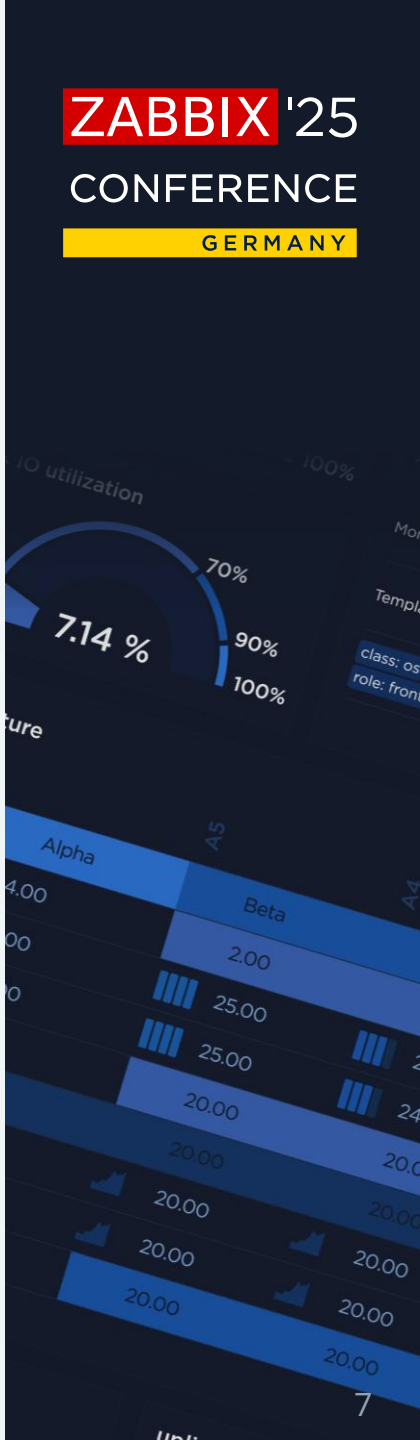


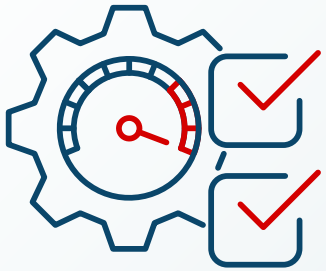
Warum dieser Talk?

- Große Zabbix Installationen haben oft DB-Performance Problem
- Tuning Möglichkeiten sind oft schon ausgereizt.

=> Zeit, das DB-Backend architektonisch neu zu denken.

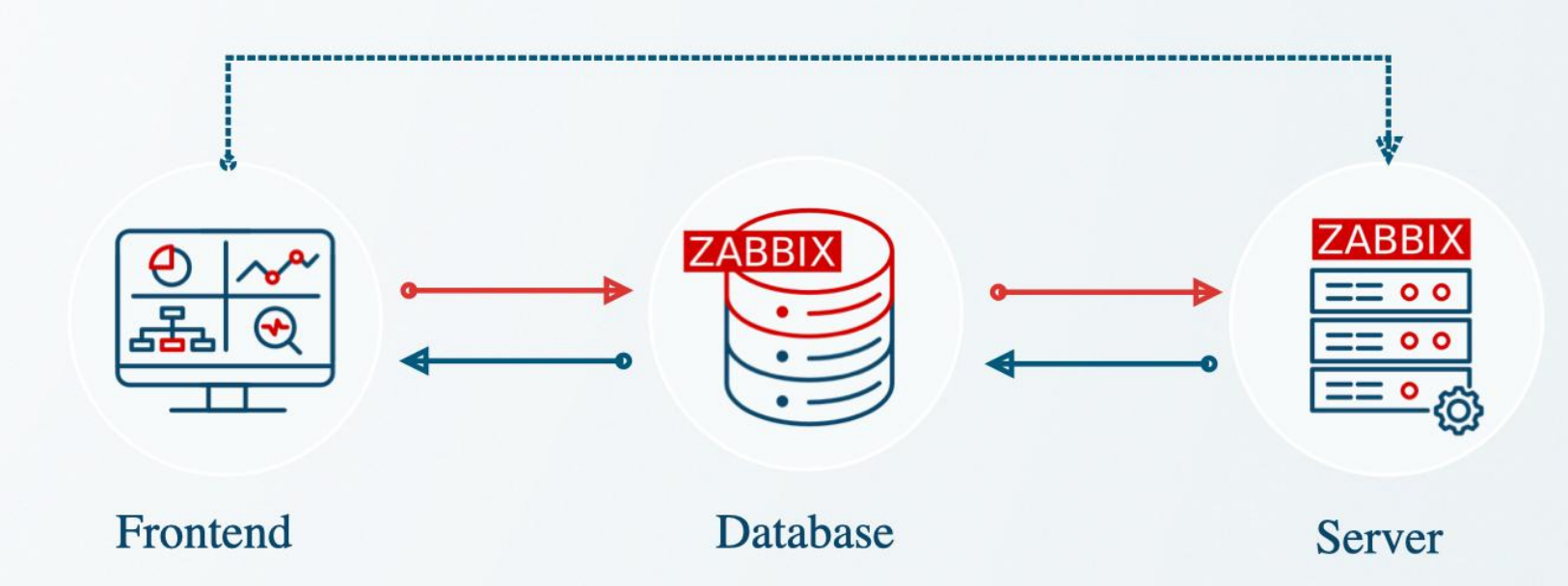
- Ziel:
  - Mehr Performance
  - Bessere Skalierbarkeit
  - Effizienteres Housekeeping
  - ...





Grundlagen

Zabbix-Architektur



PostgreSQL

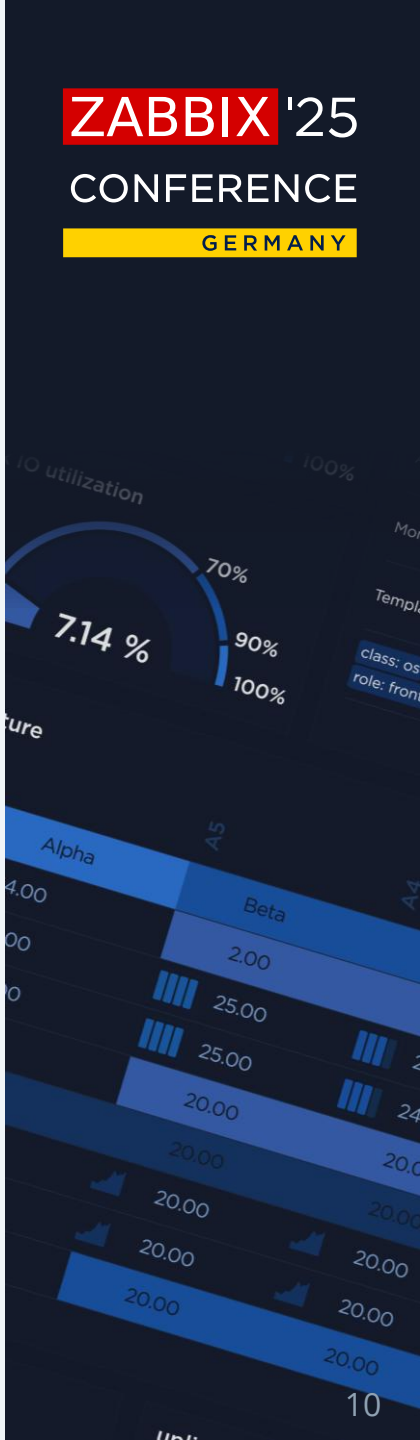
Timescale

Datenbank Architektur

monolithische Tabellen

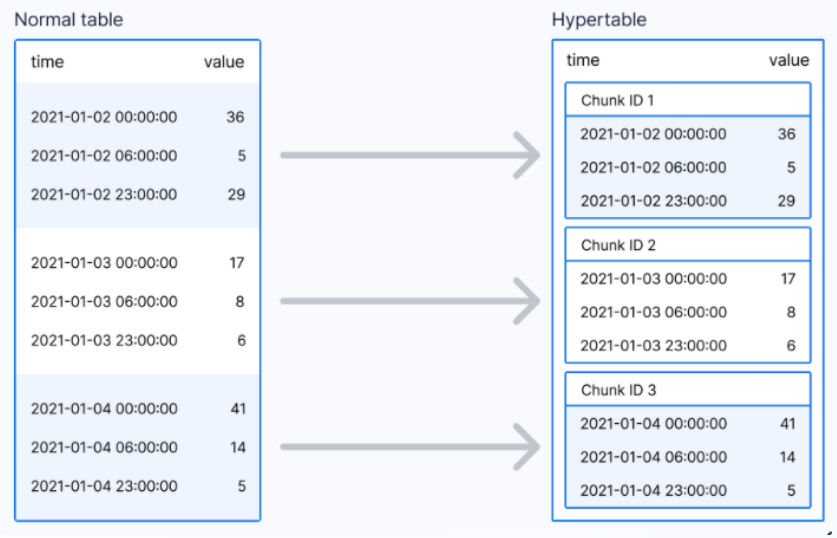
time	value
2021-01-02 00:00:00	36
2021-01-02 06:00:00	5
2021-01-02 23:00:00	29
2021-01-03 00:00:00	17
2021-01-03 06:00:00	8
2021-01-03 23:00:00	6
2021-01-04 00:00:00	41
2021-01-04 06:00:00	14
2021-01-04 23:00:00	5

- Schlechte Querry Performance durch fehlendes Pruning
- Langsames und Performance intensives Housekeeping
- Datenbanktabellen fragmentieren
- Unflexible Datenspeicherung/Wiederherstellung

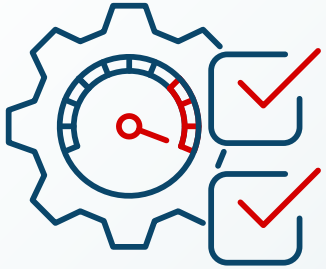


Datenbank Architektur

Partitionierte Tabellen



- Hohe Querry Performance durch Pruning
- Schnelles Housekeeping durch DROP/TRUNCATE an statt DELETE Statements
- Verbessere Datenbanktabellen Fragmentierung
- Effizientere Datenbankwartung (Vacuum, Analyse, Reindexing)



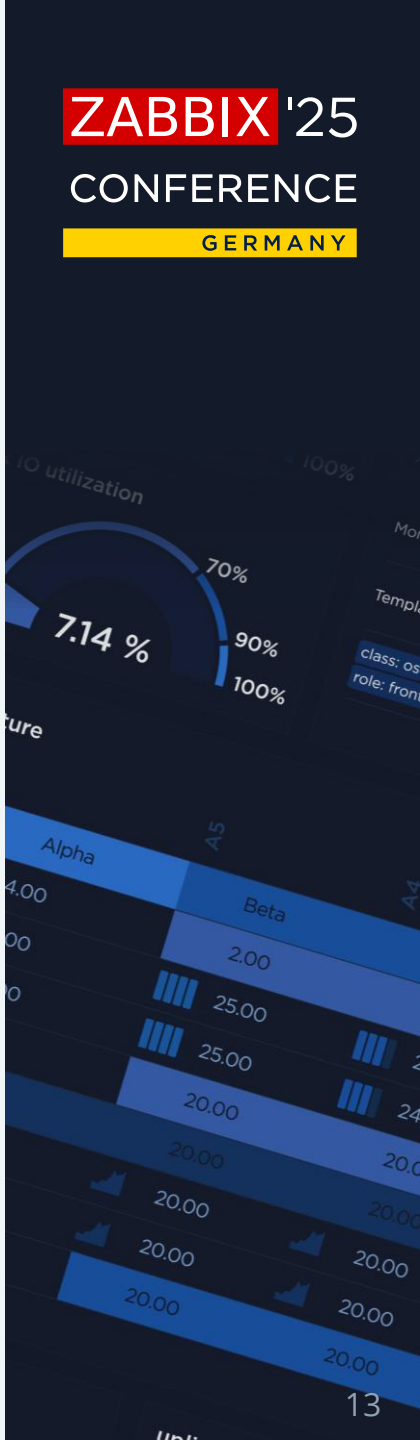
Was ist die TimescaleDB?

Was ist die TimescaleDB?

- Open-Source Erweiterung für Postgresql DB optimiert für Zeitreihendaten

- Ist eine erweitere Form von klassischem Table-Partitioning

  - Klassiche Tabellen werden in sog. “Hypertables” umgebaut
  - Automatisierte Verwaltung von sog. “Chunks” Partitionen
  - Komprimierung von Chunks für speicheroptimierte Ablage
  - Natives Policy basiertes Housekeeping



Installation der Timescale

https://docs.timescale.com/self-hosted/latest/install/installation-linux/

- **WICHTIG: Prüfung der unterstützten TimescaleDB Version**

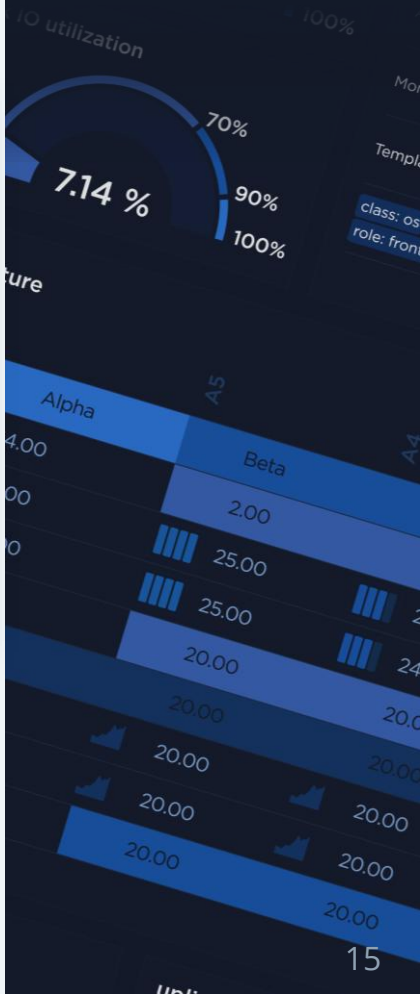
ZABBIX 20 YEARS Documentation 7.2 (current) English

PostgreSQL		13.0-17.X	Required if PostgreSQL is used as Zabbix backend database. Depending on the installation size, it might be required to increase PostgreSQL *work_mem* configuration property (4MB being the default value), so that the amount of memory used by the database for particular operation is sufficient and query execution does not take too much time.
TimescaleDB for PostgreSQL		2.13.0-2.18.X	Required if TimescaleDB is used as a PostgreSQL database extension. Make sure to install TimescaleDB Community Edition, which supports compression. Note that PostgreSQL 15 is supported since TimescaleDB 2.10. You may also refer to Timescale documentation for details regarding PostgreSQL and TimescaleDB version compatibility.

Installation der Timescaledb Extention

- Wenn TSDB nicht latest => Packages auf mark hold setzen.

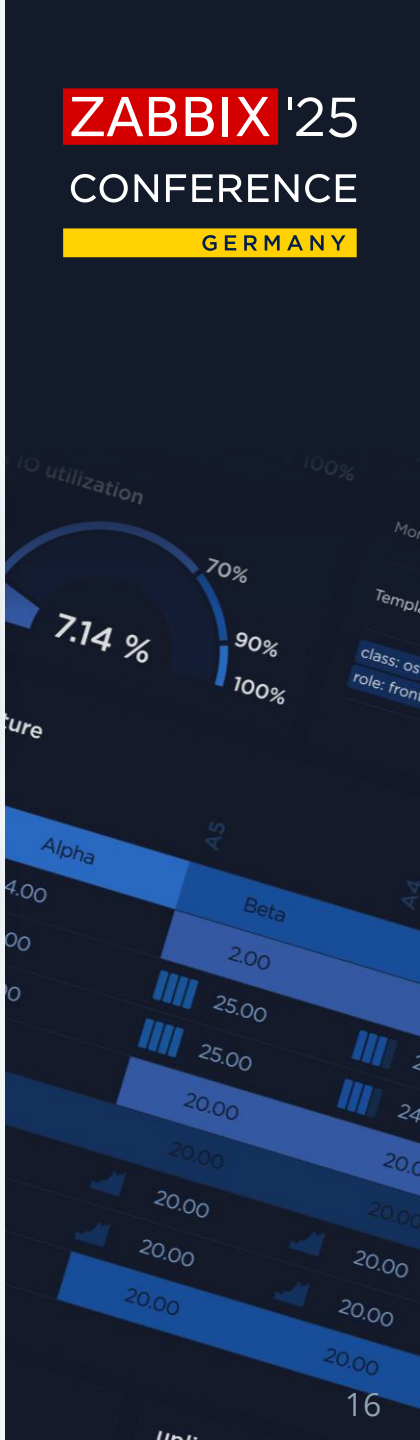
```
root@zabbix:/home/jochen.weschta# dpkg -l |grep timescale
ii  timescaledb-2-loader-postgresql-16     2.18.2~ubuntu22.04
ii  timescaledb-2-postgresql-16            2.18.2~ubuntu22.04
ii  timescaledb-toolkit-postgresql-16      1:1.18.0~ubuntu22.04
ii  timescaledb-tools                      0.18.0~ubuntu22.04

root@zabbix:/home/jochen.weschta# apt-mark hold timescaledb-2-postgresql-16
timescaledb-2-postgresql-16 set on hold.

root@zabbix:/home/jochen.weschta# apt-mark hold timescaledb-2-loader-postgresql-16
timescaledb-2-loader-postgresql-16 set on hold.

root@zabbix:/home/jochen.weschta# apt-mark showhold
timescaledb-2-loader-postgresql-16
timescaledb-2-postgresql-16
```



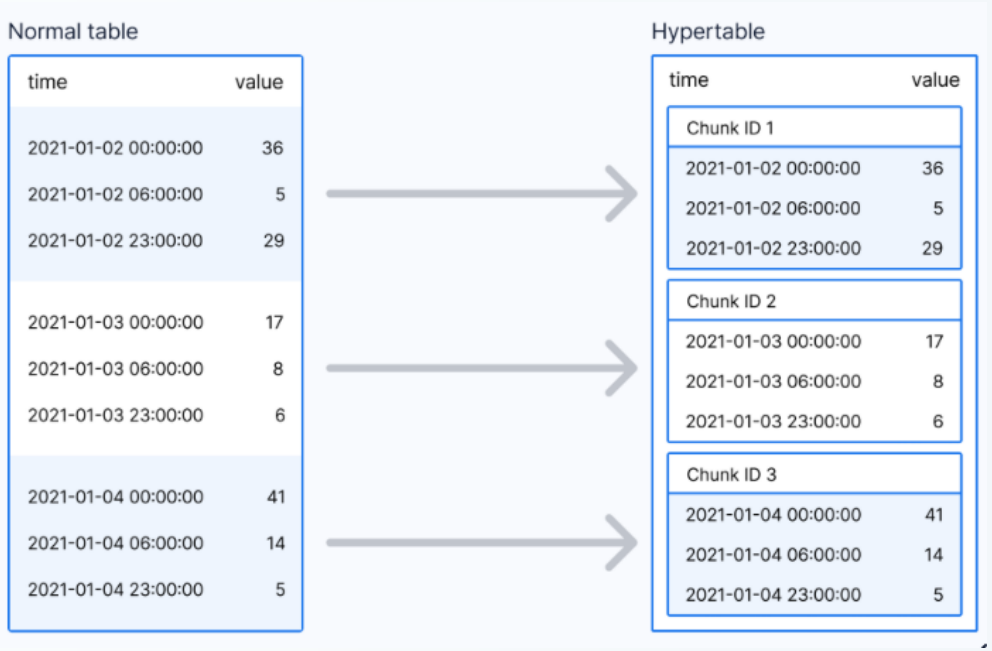
Wie aktiviert man die Partitioning (Hypertables)

```
root@zabbix:/home/jochen.weschta# dpkg -l zabbix-sql-scripts
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name               Version               Architecture Description
+++-==================-=====================-============-=================================================
ii  zabbix-sql-scripts 1:7.2.5-1+ubuntu22.04 all          Zabbix network monitoring solution - sql-scripts
```

```
root@zabbix:/home/jochen.weschta# dpkg -L zabbix-sql-scripts |grep schema.sql
/usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql
```

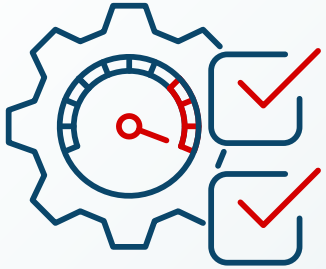


Wie aktiviert man die Hypertables



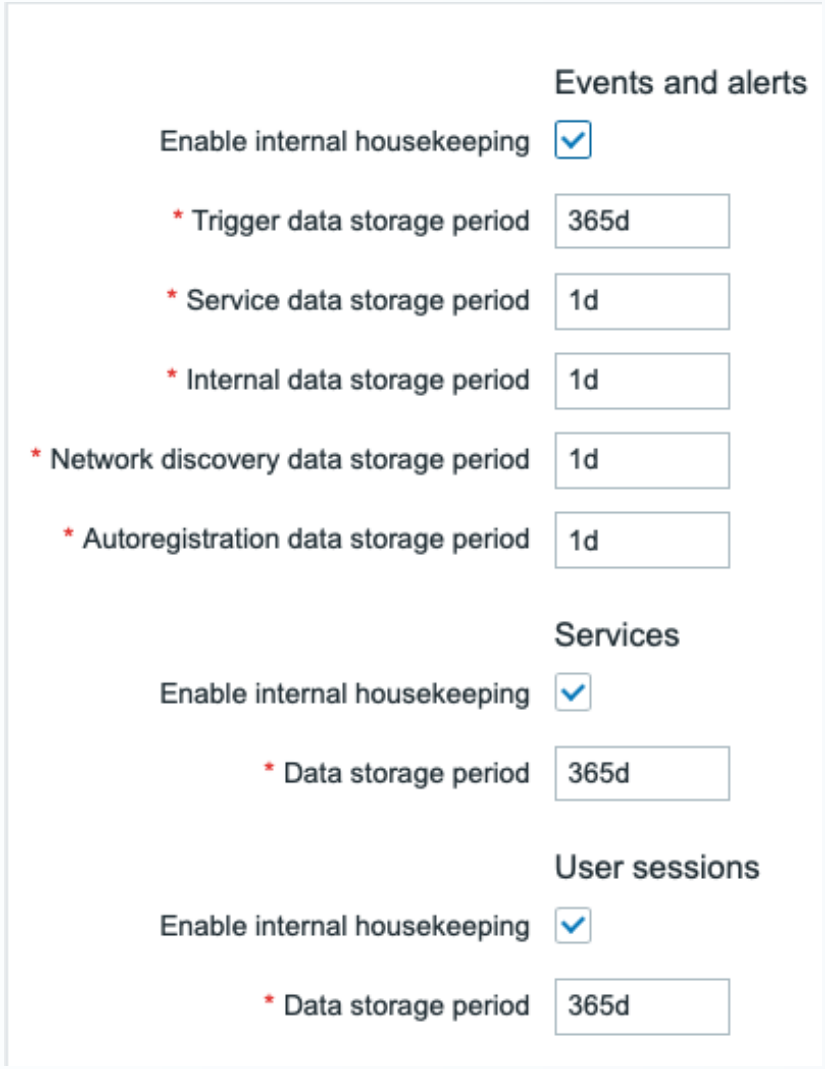

```
root@zabbix:/home/jochen.weschta# cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql |grep "PERFORM create_hypertable"
        PERFORM create_hypertable('history', 'clock', chunk_time_interval => 86400, migrate_data => true, if_not_exists => true);
        PERFORM create_hypertable('history_uint', 'clock', chunk_time_interval => 86400, migrate_data => true, if_not_exists => true);
        PERFORM create_hypertable('history_log', 'clock', chunk_time_interval => 86400, migrate_data => true, if_not_exists => true);
        PERFORM create_hypertable('history_text', 'clock', chunk_time_interval => 86400, migrate_data => true, if_not_exists => true);
        PERFORM create_hypertable('history_str', 'clock', chunk_time_interval => 86400, migrate_data => true, if_not_exists => true);
        PERFORM create_hypertable('history_bin', 'clock', chunk_time_interval => 86400, migrate_data => true, if_not_exists => true);
        PERFORM create_hypertable('auditlog', 'auditid', chunk_time_interval => 604800,
        PERFORM create_hypertable('trends', 'clock', chunk_time_interval => 2592000, migrate_data => true, if_not_exists => true);
        PERFORM create_hypertable('trends_uint', 'clock', chunk_time_interval => 2592000, migrate_data => true, if_not_exists => true);
```

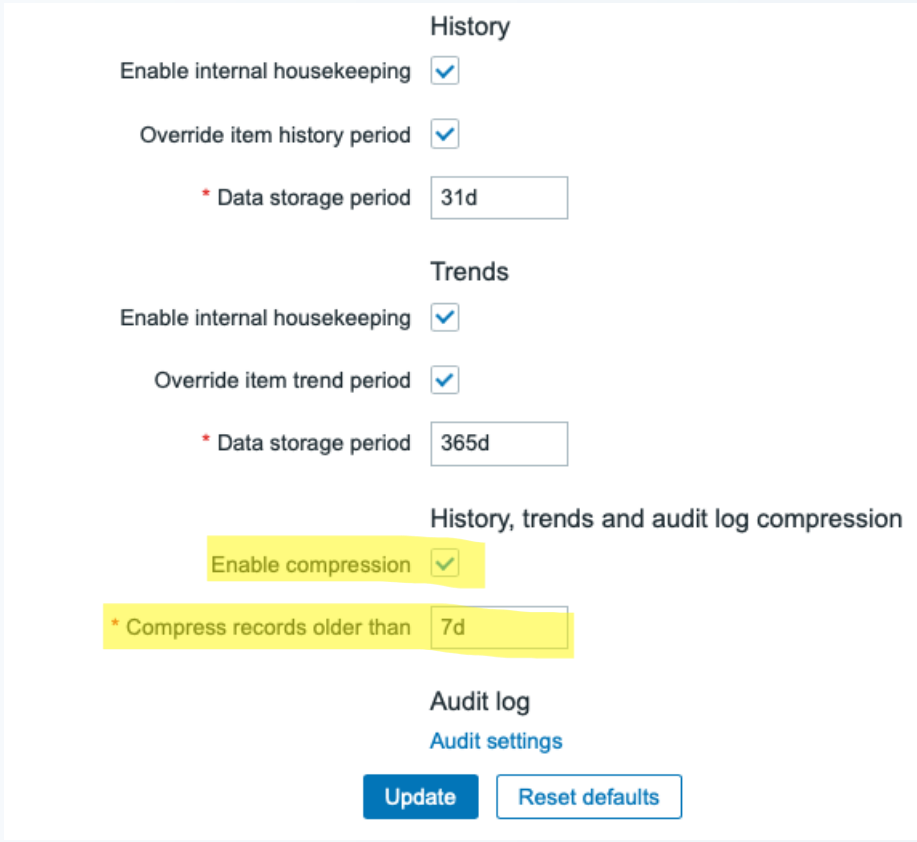
- **Kann je nach Datenmenge viel Zeit und Speicherplatz in Anspruch nemen.**

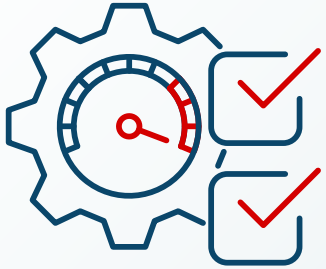


Integration in Zabbix

Zabbix Housekeeping Einstellungen







TimescaleDB internals

TimescaleDB Internals - Chunks

```
zabbix=# SELECT * FROM timescaledb_information.hypertables;

 hypertable_schema | hypertable_name | owner  | num_dimensions | num_chunks | compression_enabled | tablespaces
-------------------+-----------------+--------+----------------+------------+---------------------+-------------
 public            | history         | zabbix |              1 |         32 | t                   |
 public            | history_uint    | zabbix |              1 |         32 | t                   |
 public            | history_log     | zabbix |              1 |          0 | t                   |
 public            | history_text    | zabbix |              1 |         32 | t                   |
 public            | history_str     | zabbix |              1 |         32 | t                   |
 public            | history_bin     | zabbix |              1 |          0 | t                   |
 public            | auditlog        | zabbix |              1 |          2 | t                   |
 public            | trends          | zabbix |              1 |         14 | t                   |
 public            | trends_uint     | zabbix |              1 |         14 | t                   |
(9 rows)
```

```
zabbix=# select hypertable_name, chunk_name, is_compressed,range_start_integer,range_end_integer,chunk_creation_time FROM
timescaledb_information.chunks where hypertable_name = 'history';

 hypertable_name |      chunk_name      | is_compressed | range_start_integer | range_end_integer |      chunk_creation_time
-----------------+----------------------+---------------+---------------------+-------------------+-------------------------------
 history         | _hyper_1_1418_chunk | t             |          1743811200 |        1743897600 | 2025-04-05 00:00:00.838528+00
 history         | _hyper_1_1426_chunk | t             |          1743897600 |        1743984000 | 2025-04-06 00:00:00.813261+00
 history         | _hyper_1_1434_chunk | f             |          1743984000 |        1744070400 | 2025-04-07 00:00:00.799798+00
 history         | _hyper_1_1442_chunk | f             |          1744070400 |        1744156800 | 2025-04-08 00:00:00.800917+00
 history         | _hyper_1_1450_chunk | f             |          1744156800 |        1744243200 | 2025-04-09 00:00:00.805203+00
 history         | _hyper_1_1458_chunk | f             |          1744243200 |        1744329600 | 2025-04-10 00:00:00.802479+00
 history         | _hyper_1_1467_chunk | f             |          1744329600 |        1744416000 | 2025-04-11 00:00:00.802337+00
 history         | _hyper_1_1475_chunk | f             |          1744416000 |        1744502400 | 2025-04-12 00:00:00.80168+00
 history         | _hyper_1_1485_chunk | f             |          1744502400 |        1744588800 | 2025-04-13 00:00:00.80331+00
 history         | _hyper_1_1493_chunk | f             |          1744588800 |        1744675200 | 2025-04-14 00:00:00.799156+00
```

TimescaleDB Internals - Compression

```
zabbix=# select
    chunk_name, hypertable_name
    pg_size_pretty(before_compression_total_bytes) as before_compression,
    pg_size_pretty(after_compression_total_bytes) as after_compression,
    round(
        100.0 * (before_compression_total_bytes - after_compression_total_bytes) / before_compression_total_bytes,
        2
    ) as percent_saved
    from
    chunk_compression_stats('history')
    where
    before_compression_total_bytes > 0;
```

chunk_name	before_compression	after_compression	percent_saved
_hyper_1_1393_chunk	42 MB	3032 kB	92.88
_hyper_1_1402_chunk	42 MB	3040 kB	92.88
_hyper_1_1410_chunk	42 MB	3008 kB	92.95
_hyper_1_1418_chunk	42 MB	3032 kB	92.89
_hyper_1_1329_chunk	42 MB	3032 kB	92.94
_hyper_1_1337_chunk	42 MB	3008 kB	92.94
_hyper_1_1345_chunk	42 MB	3048 kB	92.86
_hyper_1_1426_chunk	42 MB	3032 kB	92.93
_hyper_1_1353_chunk	42 MB	3040 kB	92.86
_hyper_1_1279_chunk	42 MB	3040 kB	92.88

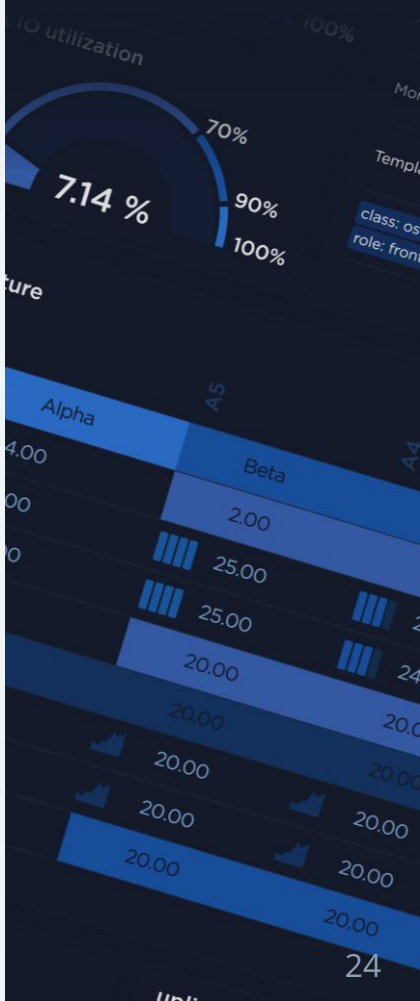
TimescaleDB Internals - Compression

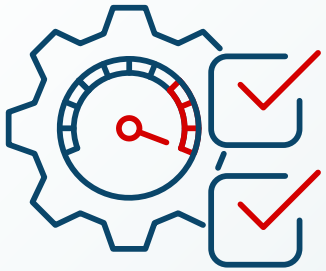
```
zabbix=# zabbix=# SELECT
    total_chunks,
    number_compressed_chunks,
    pg_size_pretty(before_compression_table_bytes) AS before_compression,
    pg_size_pretty(after_compression_total_bytes) AS after_compression,
    round(
        100.0 * (before_compression_table_bytes - after_compression_total_bytes) / before_compression_table_bytes,
        2
    ) AS percent_saved
FROM
    hypertable_compression_stats('history');

 total_chunks | number_compressed_chunks | before_compression | after_compression | percent_saved
--------------+--------------------------+--------------------+-------------------+---------------
           32 |                       24 | 480 MB             | 71 MB             |         85.19
```

TimescaleDB Internals – Compression Policies

```
zabbix=# zabbix=# SELECT    hypertable_name,    schedule_interval,    config ->> 'compress_after' AS compress_after FROM
timescaledb_information.jobs WHERE    proc_name = 'policy_compression';

 hypertable_name | schedule_interval | compress_after
-----------------+-------------------+----------------
 history         | 1 day             | 612000
 history_uint    | 1 day             | 612000
 history_log     | 1 day             | 612000
 history_text    | 1 day             | 612000
 history_str     | 1 day             | 612000
 history_bin     | 1 day             | 612000
 auditlog        | 1 day             |
 trends          | 1 day             | 612000
 trends_uint     | 1 day             | 612000
(9 rows)
```

```
zabbix=# SELECT    job_id,    hypertable_name,    proc_name,    schedule_interval,    config FROM    timescaledb_information.jobs;

 job_id | hypertable_name |             proc_name              | schedule_interval |                              config
--------+-----------------+------------------------------------+-------------------+-------------------------------------------------------------
-------
      2 |                 | policy_job_error_retention         | 1 mon             | {"drop_after": "1 month"}
      3 |                 | policy_job_stat_history_retention  | 1 mon             | {"drop_after": "1 month"}
      1 |                 | policy_telemetry                   | 24:00:00          |
   1000 | history         | policy_compression                 | 1 day             | {"hypertable_id": 1, "compress_after": 612000}
   1001 | history_uint    | policy_compression                 | 1 day             | {"hypertable_id": 2, "compress_after": 612000}
   1002 | history_str     | policy_compression                 | 1 day             | {"hypertable_id": 5, "compress_after": 612000}
   1003 | history_text    | policy_compression                 | 1 day             | {"hypertable_id": 4, "compress_after": 612000}
   1004 | history_log     | policy_compression                 | 1 day             | {"hypertable_id": 3, "compress_after": 612000}
   1005 | history_bin     | policy_compression                 | 1 day             | {"hypertable_id": 6, "compress_after": 612000}
   1006 | trends          | policy_compression                 | 1 day             | {"hypertable_id": 8, "compress_after": 612000}
   1007 | trends_uint     | policy_compression                 | 1 day             | {"hypertable_id": 9, "compress_after": 612000}
   1008 | auditlog        | policy_compression                 | 1 day             | {"hypertable_id": 7, "compress_created_before":
"170:00:00"}
(12 rows)
```





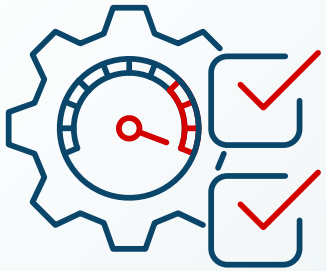
Limits

TimescaleDB - Limits

- Alle Meriken werden gleich behandelt. Separate retention Times für Items werden nicht mehr unterstützt.
- Komprimierte Chunks sind readonly

```
zabbix=# SELECT chunk_schema, chunk_name, compression_status FROM chunk_compression_stats('history');

chunk_schema      |     chunk_name     | compression_status |
----------------------+--------------------+--------------------+
_timescaledb_internal | _hyper_1_1_chunk   | Uncompressed       |
_timescaledb_internal | _hyper_1_4_chunk   | Compressed         |
```

```
#Compress
zabbix=# SELECT compress_chunk('_timescaledb_internal._hyper_1_1143_chunk',false);

#Uncompress
zabbix=# SELECT compress_chunk('_timescaledb_internal._hyper_1_1143_chunk',true);
```



Monitoring

TimescaleDB - Monitoring

- Zukunft Chunks unterliegen nicht dem Housekeeping
- Zukunft Chunks unterliegen nicht der Kompression
- Entstehen bei der Verwendung von Zabbix-Agent active Items mit “Zeit” in der Zukunft

```
zabbix=# select hypertable_name, chunk_name, is_compressed,range_start_integer,range_end_integer,chunk_creation_time FROM
timescaledb_information.chunks where hypertable_name = 'history';

 hypertable_name |     chunk_name      | is_compressed | range_start_integer | range_end_integer |      chunk_creation_time
-----------------+---------------------+---------------+---------------------+-------------------+-------------------------------
 history         | _hyper_1_1475_chunk | f             |          1744416000 |        1744502400 | 2025-04-12 00:00:00.80168+00
 history         | _hyper_1_1485_chunk | f             |          1744502400 |        1744588800 | 2025-04-13 00:00:00.80331+00
 history         | _hyper_1_1493_chunk | f             |          1744588800 |        1744675200 | 2025-04-14 00:00:00.799156+00
 history         | _hyper_1_1502_chunk | f             |          1744675200 |        1744761600 | 2025-04-15 00:00:00.803004+00
 history         | _hyper_1_1509_chunk | f             |          1744761600 |        1744848000 | 2025-04-16 00:00:00.987261+00
 history         | _hyper_1_1513_chunk | f             |          1776297600 |        1776384000 | 2025-04-16 06:24:21.448281+00
 history         | _hyper_1_1517_chunk | f             |          1807833600 |        1807920000 | 2025-04-16 06:27:51.791707+00
 history         | _hyper_1_1522_chunk | f             |          1839456000 |        1839542400 | 2025-04-16 06:28:51.077505+00
```

SCADIX

Connect

Persönlich auf Veranstaltungen

https://scadix.de

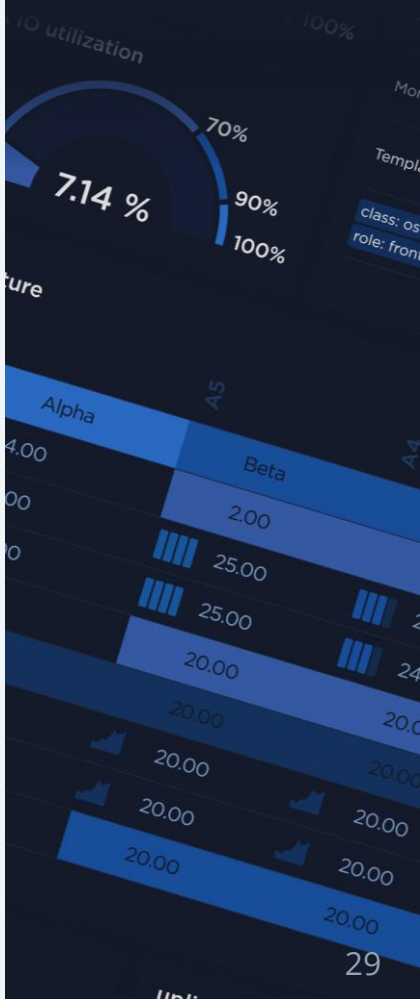
jochen.weschta@scadix.de

[m] jochen.weschta:matrix.scadix.de

https://github.com/scadix-gmbh

https://www.linkedin.com/company/scadix

ZABBIX '25 CONFERENCE GERMANY



ZABBIX '25
CONFERENCE
GERMANY

Thank you!

Jochen Weschta

Senior Consultant Infrastructure Architect
Zabbix Certified Trainer

SCADIX GmbH