

Stream metrics storage in Timescale DB

- [Postgresql and Timescale DB installation](#)
- [Timescale DB setup](#)
- [Data retention setup using cron](#)
- [DB structure](#)

[Timescale](#) is Postgresql based open source time series DB. Unlike Influx, Timescale does not store all the indexes in RAM, therefore it can be used to store large amount of metric values. In this case, the same Postgresql DB can be used to store metric values, stream history and data acquisition settings.

Postgresql and Timescale DB installation

A specific Postgresql version (9.6, 10 or 11) must be installed during setup process. If earlier Postgresql version is installed (for example, Centos 7 default repository contains Postgresql 9.2), it should be fully removed including `/usr/bin` executables.

To setup Postgresql 9.6+Timescale DB on CentOS 7, do the following:

1. Install Postgresql repository

```
yum install -y https://download.postgresql.org/pub/repos/yum/11/redhat/rhel-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

2. Create `/etc/yum.repos.d/timescale_timescaledb.repo` file

```
[timescale_timescaledb]
name=timescale_timescaledb
baseurl=https://packagecloud.io/timescale/timescaledb/el/7/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/timescale/timescaledb/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

3. Install Postgresql

```
yum install -y postgresql96-server
```

4. Install TimescaleDB

```
yum install -y timescaledb-postgresql-9.6
```

5. Initialize Postgresql DB

```
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

6. Configure Postgresql to work with Timescale DB

```
timescaledb-tune --yes --pg-config=/usr/pgsql-9.6/bin/pg_config
```

7. Configure access to Postgresql tables in `/var/lib/pgsql/9.6/data/pg_hba.conf` file

```
# IPv4 local connections:
host    all             all             127.0.0.1/32      md5
# IPv6 local connections:
host    all             all             ::1/128          md5
```

8. Launch Postgresql

```
su - postgres
/usr/pgsql-9.6/bin/postgres &
```

9. Enter Postgresql console

```
/usr/pgsql-9.6/bin/psql
```

10. Create user and DB

```
CREATE USER w csoam WITH PASSWORD 'w csoam';
CREATE DATABASE w csstat;
```

11. Grant DB privileges to user

```
\c w csstat
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO "w csoam";
```

12. Initialize Timescale extension

```
CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;
```

13. Create DB to [store stream history and data acquisition settings](#)

```
CREATE DATABASE w csoam;
\c w csoam
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO "w csoam";
```

14. Exit Postgresql console

```
\q
exit
```

Timescale DB setup

To configure Timescale DB for metric storage, do the following:

1. Set the following parameter in w csoam.properties file

```
metric_store=timescale
```

2. Set metrics storage chunk interval in init_t sdb.properties file

```
timescale_chunk_interval=2 days
```

By default, metrics are stored in 2 days chunks.

3. Launch DB setup script

```
./init_t sdb.sh
```

Data retention setup using cron

To save disk space, data retention can be set using cron utility. For example, set crontab as follows to check conditions table every 5 minutes and delete metrics data older than 5 days:

```
*/5 * * * * PGPASSWORD="wcsoam" /usr/pgsql-9.6/bin/psql -h localhost -p 5432 -U wcsoam -d wcsstat -w -c "SELECT drop_chunks(interval '5 days', 'conditions');"
```

DB structure

The following table should be in Postgresql to store metrics data

Field	Type	Note
CONDITIONS		
time	timestampz	NOT NULL
node_id	bigint	NOT NULL
media_id	character varying(255)	
video_height	bigint	
video_width	bigint	
video_rate	bigint	
video_sync	bigint	
video_fps	bigint	
video_nack	bigint	
video_pli	bigint	
video_codec	bigint	
audio_sync	bigint	
audio_rate	bigint	
audio_lost	bigint	
audio_codec	bigint	

Metrics data are fragmented to chunks and are stored to subtables, each subtable volume is defined by chunk duration. To limit subtables amount, it is recommended to setup [data retention using cron](#) with maximum data storage interval.