Database requirements

Unimus can be deployed using an embedded file-based database (HSQL - only for small deployments), or a full dedicated DB.

You can find the full list of supported DB types and versions here.

Supported databases and versions

Database	Oldest supported version	Newest supported version
PostgreSQL	9.2	18.0
MariadDB	10.2	12.2
MYSQL	5.7	9.6
MSSQL	14.0 (from 2017)	17.0 (from 2025)
HSQL	N/A	N/A

Recommended DB settings:

For optimal performance and compatibility of your database with **Unimus** you can find recommend DB settings it he following table.
In case your database use any other collation, please make sure to choose one that is always case-insensitive.

Database	Characters set	Collation
PostgreSQL	utf8	`C` (equivalent to `POSIX`)
MariadDB	utf8mb4	utf8mb4_unicode_ci
MYSQL	utf8mb4	utf8mb4_unicode_ci
MSSQL	UCS-2 / UTF-16	Latin1_General_CI SQL_Latin1_General_CP1_CI
HSQL	NOT RELEVANT	NOT RELEVANT

Using this combination ensures better global language support and consistent case-insensitive behavior in text comparisons.

HSQL (file-based) database considerations

Please note we do not recommend using the HSQL (file-based) database for deploys over 100 devices, and **do not support** HSQL deploys with 1000+ devices.

Since HSQL is a file based database, that keeps all data in a few flat text-based files, as the DB size grows, performance is severely impacted. If you want to use HSQL long-term, you should:

- setup data retention policies for both History Jobs and Backups (in "Other Settings")

- monitor database directory size (it should not reach multi-gigabyte sizes)
- monitor disk usage of the disk where HSQL stores its data

HSQL databases can get corrupted if the disk on which data is stored runs out of space.

PostgreSQL details

Until Unimus 2.6, only PostgreSQL 11 or older were supported. Starting with Unimus 2.6, we now support all newer version of Postgre.

If you wish to migrate from an older Postgre version to newer, please see the instructions below.

Step by step for PostgreSQL migration:

Before you start with migration and for more information related to PostgreSQL we highly recommend to visit visit the official page: https://www.postgresql.org/docs/9.0/migration.html

Upgrading PostgreSQL from Version 11 to 12+

- **Prepare the new database**:
  - Set up your new PostgreSQL 12+ instance where the data will be migrated.
- **Stop both databases**:
  - Make sure to stop both the old (PostgreSQL 11) and the new (PostgreSQL 12+) databases to avoid any conflicts during migration.
- **Access PostgreSQL**:
  - Log in to your old PostgreSQL 11 database and run the following commands:
    - ALTER TABLE backup SET WITHOUT OIDS;
    - ALTER TABLE push_output_group SET WITHOUT OIDS;
  - These commands ensure that tables containing OIDs are altered to be compatible with newer versions of PostgreSQL (tables no longer support OIDs in version 12+). No errors should occur when executing them.
- **Pre-check the upgrade**:
  - Run the following command to check if your system is ready for the upgrade:
    - pg_upgrade --check
  - This will confirm whether the upgrade can proceed without issues.
- **Backup the old database**:
  - Make sure to create a full dump of your old database for safekeeping, using the `pg_dump` command.
- **Migrate the data**:
  - After ensuring everything is in order, migrate your data from PostgreSQL 11 to 12+ using:
    - pg_upgrade

Upgrading PostgreSQL from Version 12 to 18

- **Backup Your Old Database**:
  - Before starting the migration, create a full dump of your old PostgreSQL 12 database. This step ensures you have a backup in case anything goes wrong during the upgrade process.
- **Check Compatibility**:
  - Ensure that both your old (PostgreSQL 12) and new (PostgreSQL 18) databases are compatible for migration. You can do this by running the following command:
    - pg_upgrade --check
  - This command will verify that the upgrade can proceed without any issues.
- **Migrate the Data**:
  - Once you have confirmed compatibility, you can proceed with the migration using:
    - pg_upgrade