



How to upgrade and update a blank Aero-Webb database dump

How to upgrade and update a blank Aero-Webb database dump

Context

We already have one or several blank database dumps for standard Aero-Webb and Optimal Aero-Webb, done in a previous Aero-Webb version and now we need to upgrade and update those dumps to the last Aero-Webb version, for specific PostgreSQL versions.

In this document, we will illustrate the whole procedure with the following existing dumps that we'll upgrade and update to Aero-Webb 7.8.9-hf3:

PostgreSQL 14/AW 7.8.x Standard

File	aerowebbV7_STANDARD_VIERGE_7.8.0_20240925.backup
Aero-Webb	7.8.0 Standard
PostgreSQL	14.13 with user 'postgres'

Database creation script:

```
SET client_encoding = 'UTF8';
SET standard_conforming_strings = 'on';
SELECT pg_catalog.set_config('search_path', '', false);

DROP DATABASE standard_vierge;
CREATE DATABASE standard_vierge WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE =
'French_France.1252';
```

PostgreSQL 14/AW 7.8.x OPTIMAL

File	aerowebbV7_OPTIMAL_VIERGE_7.8.0_20240925.backup
Aero-Webb	7.8.0 OPTIMAL
PostgreSQL	14.13 with user 'postgres'

Database creation script:

```

SET client_encoding = 'UTF8';
SET standard_conforming_strings = 'on';
SELECT pg_catalog.set_config('search_path', '', false);

DROP DATABASE optimal_vierge;
CREATE DATABASE optimal_vierge WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE =
'French_France.1252';

```

PostgreSQL 12/AW 7.6.x Standard

File	aerowebbV7_BDDVIERGE_20210111.backup
Aero-Webb	7.6.0 Standard
PostgreSQL	12.4 with user 'postgres'

Database creation script:

```

SET client_encoding = 'UTF8';
SET standard_conforming_strings = 'on';
SELECT pg_catalog.set_config('search_path', '', false);

DROP DATABASE "aerowebbV7_BDDVIERGE_20210111";
CREATE DATABASE "aerowebbV7_BDDVIERGE_20210111" WITH TEMPLATE = template0 ENCODING = 'UTF8'
LC_COLLATE = 'English_United_States.1252' LC_CTYPE = 'English_United_States.1252';

```

Prerequisites

- Install a WSL Ubuntu VM using the following procedure: [How to install Ubuntu WSL](#).
- Have the desired version(s) of PostgreSQL in this new WSL VM. In this doc, we will illustrate operations with PostgreSQL versions 12 and 14. Consequently, you will need to have installed in the WSL VM:
 - PostgreSQL 12 server and clients packages.
 - PostgreSQL 14 server and clients packages.
- Download the following artifacts and copy them in a folder in your WSL home directory. In this document, we will name this folder `<PATH_TO_ARCHIVE_FOLDER>` :
 - [7.8.9-hf3_OPTIMAL.zip](#)
 - [7.8.9-hf3_STANDARD.zip](#)
- Get the previously mentioned blank Aero-Webb database dumps, **or similar ones**. You should find them in the [GitLab database models repository](#). Next, copy them in a folder in the WSL VM. In this document, we will name this folder `<PATH_TO_DUMP_FOLDER>` .
- Having cloned the Aero-Webb Services project from [here](#) into your `$HOME/Workspaces/Aero-Webb/Backend` folder in your WSL VM and having `git checkout` the needed target branch (in this document, the target branch is the "version\7.8.9-hf3").

Install PostgreSQL 12 & 14

Prerequisites

1. Check the existence of the file `/etc/apt/sources.list.d/pgdg.list`

```
cat /etc/apt/sources.list.d/pgdg.list
```

with the following content:

```
deb http://apt.postgresql.org/pub/repos/apt/ noble-pgdg main
```

2. Check the existence of the desired PostgreSQL packages:

```
apt search postgresql-12 2> /dev/null | wc -l  
apt search postgresql-14 2> /dev/null | wc -l
```

In both cases, the returned value must be greater than 0.

If the `pgdg.list` file does not exist and the previous commands don't return a value greater than zero, then you must install the PostgreSQL repository, as follows:

```
sudo apt install -y postgresql-common  
sudo /usr/share/postgresql-common/pgdg/apt.postgresql.org.sh
```

Then, the procedure can be continued.

Note

You can install PostgreSQL 12 & 14 in whatever order you want. But be aware that the port number assigned may be different.

Install PostgreSQL 14

```
sudo apt install postgresql-14 postgresql-client-14
```

Result :

- Executables are installed under: `/usr/lib/postgresql/14` .
- Configuration files are installed under: `/etc/postgresql/14` .
- A port is automatically assigned: 5433 (because PostgreSQL 16 was already installed on 5432).
- A Unix socket is created: `/var/run/postgresql/.s.PGSQL.5433` .
- A system service is created, enabled at system startup and launched: `postgresql@14-main.service` .

Install PostgreSQL 12

```
sudo apt install postgresql-12 postgresql-client-12
```

Result :

- Executables are installed under: `/usr/lib/postgresql/12`.
- Configuration files are installed under: `/etc/postgresql/12`
- A port is automatically assigned: 5434 (because PostgreSQL 16 was already installed on 5432 and PostgreSQL 14 was already installed on 5433).
- A Unix socket is created: `/var/run/postgresql/.s.PGSQL.5434`.
- A system service is created, enabled at system startup and launched: `postgresql@12-main.service`.

Assign a password to the database user "postgres"

Assign a password to the "postgres" user in order to be able to connect to it via TCP/IP.

Procedure

First, connect via Unix socket:

```
sudo -u postgres /usr/lib/postgresql/<PG_VERSION>/bin/psql -h /var/run/postgresql -p <PG_PORT>
```

If the connection is OK:

```
psql (<PG_VERSION> (Ubuntu <PG_VERSION>-1.pgdg24.04+1))
Saisissez « help » pour l'aide.
```

Then, change the password:

```
postgres=# alter user postgres with password '<NEW_PASSWORD>';
ALTER ROLE
postgres=# \q
```

Finally, test with a connection via TCP/IP:

```
/usr/lib/postgresql/<PG_VERSION>/bin/psql -h localhost -p <PG_PORT> -U postgres
```

Type the password previously assigned:

```
Mot de passe pour l'utilisateur postgres :
```

If the connection is OK:

```
psql (<PG_VERSION> (Ubuntu <PG_VERSION>-1.pgdg24.04+1))
Connexion SSL (protocole : TLSv1.3, chiffrement : TLS_AES_256_GCM_SHA384, bits : 256, compression
: désactivé)
Saisissez « help » pour l'aide.
```

Then quit the `psql` console:

```
postgres=# \q
```

Example with PostgreSQL 12

```
sudo -u postgres /usr/lib/postgresql/12/bin/psql -h /var/run/postgresql -p 5434
```

```
psql (12.20 (Ubuntu 12.20-1.pgdg24.04+1))
Saisissez « help » pour l'aide.
```

Here the chosen password is "postgres":

```
postgres=# alter user postgres with password 'postgres';
ALTER ROLE
postgres=# \q
```

```
/usr/lib/postgresql/12/bin/psql -h localhost -p 5434 -U postgres
```

Mot de passe pour l'utilisateur postgres :

```
psql (12.20 (Ubuntu 12.20-1.pgdg24.04+1))
Connexion SSL (protocole : TLSv1.3, chiffrement : TLS_AES_256_GCM_SHA384, bits : 256, compression
: désactivé)
Saisissez « help » pour l'aide.

postgres=# \q
```

Upgrade and update blank database dumps

For the PostgreSQL 14/AW 7.8.x Standard dump

1. Create a folder to centralize the logs:

```
mkdir -p ~/tmp/logs/pg14/std
```

2. Create the database manually, to ensure you have a correctly defined locale:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -W -U postgres
```

```
postgres=# CREATE DATABASE standard_vierge WITH TEMPLATE = template0 ENCODING = 'UTF8';
CREATE DATABASE
postgres=# \q
```

3. Restore the existing dump into PostgreSQL 14:

```
/usr/lib/postgresql/14/bin/pg_restore --verbose -h localhost -p 5433 -W -U postgres --no-owner -d standard_vierge
<PATH_TO_DUMP_FOLDER>/aerowebbv7_STANDARD_VIERGE_7.8.0_20240925.backup >
~/tmp/logs/pg14/std/restore_aw.log 2> ~/tmp/logs/pg14/std/restore_aw_error.log
```

4. Upgrade and update the database to 7.8.9-hf3:

Extract the `7.8.9-hf3_STANDARD.zip` file in your `tmp` folder:

```
cd ~/tmp
unzip <PATH_TO_ARCHIVE_FOLDER>/7.8.9-hf3_STANDARD.zip
```

In the `Aero-Webb_scripts` folder:

```
cd ~/tmp/7.8.9-hf3_STANDARD/Aero-Webb_scripts
```

Run each of the scripts present, in the lexicographic order of their name, adapting the name of the log files:

- `90_01_update_upgrade_STANDARD.sql`

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d standard_vierge -a -f
90_01_update_upgrade_STANDARD.sql -v mandant="'STA'" >
~/tmp/logs/pg14/std/90_01_update_upgrade.log 2>
~/tmp/logs/pg14/std/90_01_update_upgrade_error.log
```

5. Run the "sanity check" scripts on the upgraded database:

In the Domain sub-project `monitoring` folder, from branch `version/7.8.9-hf3`:

```
cd $HOME/Workspaces/Aero-Webb/Backend/services/DOMAIN/src/main/resources/com/twomoro/aerowebb/domain/request/monitoring
```

Run each of the scripts present, in no particular order, adapting the name of the log files:

- `sanity_v7_pg_aggregated_measure_check.sql`:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d standard_vierge -f
sanity_v7_pg_aggregated_measure_check.sql >
~/tmp/logs/pg14/std/sanity_v7_pg_aggregated_measure_check.log 2>
~/tmp/logs/pg14/std/sanity_v7_pg_aggregated_measure_check_error.log
```

- `sanity_v7_pg_get_inconsistent_data.sql`:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d standard_vierge -f
sanity_v7_pg_get_inconsistent_data.sql >
~/tmp/logs/pg14/std/sanity_v7_pg_get_inconsistent_data.log 2>
~/tmp/logs/pg14/std/sanity_v7_pg_get_inconsistent_data_error.log
```

- `sanity_v7-8_pg_FK_PK_check.sql` (needs a function present from AW 7.8.9):

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d standard_vierge -f
sanity_v7-8_pg_FK_PK_check.sql > ~/tmp/logs/pg14/std/sanity_v7-8_pg_FK_PK_check.log 2>
~/tmp/logs/pg14/std/sanity_v7-8_pg_FK_PK_check_error.log
```

6. Save the upgraded and updated database (= recreation of the dump):

```
/usr/lib/postgresql/14/bin/pg_dump --verbose -h localhost -p 5433 -d standard_vierge -U
postgres -W --no-owner --clean --create --format=c -f
<PATH_TO_DUMP_FOLDER>/aerowebbv7_STANDARD_VIERGE_PG14_7.8.9-hf3_20240927.backup >
~/tmp/logs/pg14/std/dump_aw.log 2> ~/tmp/logs/pg14/std/dump_aw_error.log
```

For the PostgreSQL 14/AW 7.8.x OPTIMAL dump

1. Create a folder to centralize the logs:

```
mkdir ~/tmp/logs/pg14/optimal
```

2. Create the database manually, to ensure you have a correctly defined locale:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -W -U postgres
```

```
postgres=# CREATE DATABASE optimal_vierge WITH TEMPLATE = template0 ENCODING = 'UTF8';
CREATE DATABASE
postgres=# \q
```

3. Restore the existing dump into PostgreSQL 14:

```
/usr/lib/postgresql/14/bin/pg_restore --verbose -h localhost -p 5433 -W -U postgres --no-
owner -d optimal_vierge <PATH_TO_DUMP_FOLDER>/aerowebbv7_OPTIMAL_VIERGE_7.8.0_20240925.backup
> ~/tmp/logs/pg14/optimal/restore_aw.log 2> ~/tmp/logs/pg14/optimal/restore_aw_error.log
```

4. Upgrade and update the database to 7.8.9-hf3:

Extract the 7.8.9-hf3_OPTIMAL.zip file in your tmp folder:

```
cd ~/tmp
unzip <PATH_TO_ARCHIVE_FOLDER>/7.8.9-hf3_OPTIMAL.zip
```

In the Aero-Webb_scripts folder:

```
cd ~/tmp/7.8.9-hf3_OPTIMAL/Aero-Webb_scripts
```

Run each of the scripts present, in the lexicographic order of their name, adapting the name of the log files:

- 90_01_update_upgrade_OPTIMAL.sql:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d optimal_vierge -a -f
90_01_update_upgrade_OPTIMAL.sql -v mandant="'STA'" >
~/tmp/logs/pg14/optimal/90_01_update_upgrade.log 2>
~/tmp/logs/pg14/optimal/90_01_update_upgrade_error.log
```

- 99_00_aw_optimal_data.sql:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d optimal_vierge -a -f
99_00_aw_optimal_data.sql -v mandant="'STA'" >
~/tmp/logs/pg14/optimal/99_00_aw_optimal_data.log 2>
~/tmp/logs/pg14/optimal/99_00_aw_optimal_data_error.log
```

- 99_02_activate_aw_custom_optimal.sql:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d optimal_vierge -a -f
99_02_activate_aw_custom_optimal.sql -v mandant="'STA'" >
~/tmp/logs/pg14/optimal/99_02_activate_aw_custom_optimal.log 2>
~/tmp/logs/pg14/optimal/99_02_activate_aw_custom_optimal_error.log
```

- 99_03_update_data_to_french.sql:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d optimal_vierge -a -f
99_03_update_data_to_french.sql -v mandant="'STA'" >
~/tmp/logs/pg14/optimal/99_03_update_data_to_french.log 2>
~/tmp/logs/pg14/optimal/99_03_update_data_to_french_error.log
```

- 99_04_upgrade_indexes_postgres_20240819.sql:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d optimal_vierge -a -f
99_04_upgrade_indexes_postgres_20240819.sql -v mandant="'STA'" >
~/tmp/logs/pg14/optimal/99_04_upgrade_indexes_postgres.log 2>
~/tmp/logs/pg14/optimal/99_04_upgrade_indexes_postgres_error.log
```

5. Run the “sanity check” scripts on the upgraded database:

In the Domain sub-project `monitoring` folder, from branch `version/7.8.9-hf3`:

```
cd $HOME/Workspaces/Aero-
Webb/Backend/services/DOMAIN/src/main/resources/com/twomoro/aerowebb/domain/request/monitoring
```

Run each of the scripts present, in no particular order, adapting the name of the log files:

- sanity_v7_pg_aggregated_measure_check.sql:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d optimal_vierge -f
sanity_v7_pg_aggregated_measure_check.sql >
~/tmp/logs/pg14/optimal/sanity_v7_pg_aggregated_measure_check.log 2>
~/tmp/logs/pg14/optimal/sanity_v7_pg_aggregated_measure_check_error.log
```

- sanity_v7_pg_get_inconsistent_data.sql:

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d optimal_vierge -f
sanity_v7_pg_get_inconsistent_data.sql >
~/tmp/logs/pg14/optimal/sanity_v7_pg_get_inconsistent_data.log 2>
~/tmp/logs/pg14/optimal/sanity_v7_pg_get_inconsistent_data_error.log
```

- sanity_v7-8_pg_FK_PK_check.sql (needs a function present from AW 7.8.9):

```
/usr/lib/postgresql/14/bin/psql -h localhost -p 5433 -U postgres -d optimal_vierge -f
sanity_v7-8_pg_FK_PK_check.sql > ~/tmp/logs/pg14/optimal/sanity_v7-8_pg_FK_PK_check.log 2>
~/tmp/logs/pg14/optimal/sanity_v7-8_pg_FK_PK_check_error.log
```

6. Save the upgraded and updated database (= recreation of the dump):

```
/usr/lib/postgresql/14/bin/pg_dump --verbose -h localhost -p 5433 -d optimal_vierge -U
postgres -W --no-owner --clean --create --format=c -f
<PATH_TO_DUMP_FOLDER>/aerowebbV7_OPTIMAL_VIERGE_PG14_7.8.9-hf3_20240927.backup >
~/tmp/logs/pg14/optimal/dump_aw.log 2> ~/tmp/logs/pg14/optimal/dump_aw_error.log
```

For the PostgreSQL 12/AW 7.6.x Standard dump

1. Create a folder to centralize the logs:

```
mkdir -p ~/tmp/logs/pg12/std
```

2. Create the database manually, to ensure you have a correctly defined locale:

```
/usr/lib/postgresql/12/bin/psql -h localhost -p 5434 -W -U postgres
```

```
postgres=# CREATE DATABASE standard_vierge WITH TEMPLATE = template0 ENCODING = 'UTF8';
CREATE DATABASE
postgres=# \q
```

3. Restore the existing dump into PostgreSQL 12:

```
/usr/lib/postgresql/12/bin/pg_restore --verbose -h localhost -p 5434 -W -U postgres --no-
owner -d standard_vierge <PATH_TO_DUMP_FOLDER>/aerowebbV7_BDDVIERGE_20210111.backup >
~/tmp/logs/pg12/std/restore_aw.log 2> ~/tmp/logs/pg12/std/restore_aw_error.log
```

4. Upgrade and update the database to 7.8.9-hf3:

If it hasn't already been done, extract the 7.8.9-hf3_STANDARD.zip file in your tmp folder:

```
cd ~/tmp
unzip <PATH_TO_ARCHIVE_FOLDER>/7.8.9-hf3_STANDARD.zip
```

In the Aero-Webb_scripts folder:

```
cd ~/tmp/7.8.9-hf3_STANDARD/Aero-Webb_scripts
```

Run each of the scripts present, in the lexicographic order of their name, adapting the name of the log files:

- 90_01_update_upgrade_STANDARD.sql

```
/usr/lib/postgresql/12/bin/psql -h localhost -p 5434 -U postgres -d standard_vierge -a -f
90_01_update_upgrade_STANDARD.sql -v mandant="'STA'" >
~/tmp/logs/pg12/std/log_90_01_update_upgrade_STANDARD.log 2>
~/tmp/logs/pg12/std/error_90_01_update_upgrade_STANDARD.log
```

5. Run the “sanity check” scripts on the upgraded database:

In the Domain sub-project monitoring folder, from branch version/7.8.9-hf3:

```
cd $HOME/Workspaces/Aero-Webb/Backend/services/DOMAIN/src/main/resources/com/twomoro/aerowebb/domain/request/monitoring
```

Run each of the scripts present, in no particular order, adapting the name of the log files:

- `sanity_v7_pg_aggregated_measure_check.sql`:

```
/usr/lib/postgresql/12/bin/psql -h localhost -p 5434 -U postgres -d standard_vierge -f
sanity_v7_pg_aggregated_measure_check.sql >
~/tmp/logs/pg12/std/sanity_v7_pg_aggregated_measure_check.log 2>
~/tmp/logs/pg12/std/sanity_v7_pg_aggregated_measure_check_error.log
```

- `sanity_v7_pg_get_inconsistent_data.sql`:

```
/usr/lib/postgresql/12/bin/psql -h localhost -p 5434 -U postgres -d standard_vierge -f
sanity_v7_pg_get_inconsistent_data.sql >
~/tmp/logs/pg12/std/sanity_v7_pg_get_inconsistent_data.log 2>
~/tmp/logs/pg12/std/sanity_v7_pg_get_inconsistent_data_error.log
```

- `sanity_v7-8_pg_FK_PK_check.sql` (needs a function present from AW 7.8.9):

```
/usr/lib/postgresql/12/bin/psql -h localhost -p 5434 -U postgres -d standard_vierge -f
sanity_v7-8_pg_FK_PK_check.sql > ~/tmp/logs/pg12/std/sanity_v7-8_pg_FK_PK_check.log 2>
~/tmp/logs/pg12/std/sanity_v7-8_pg_FK_PK_check_error.log
```

6. Save the upgraded and updated database (= recreation of the dump):

```
/usr/lib/postgresql/12/bin/pg_dump --verbose -h localhost -p 5434 -d standard_vierge -U
postgres -W --no-owner --clean --create --format=c -f
<PATH_TO_DUMP_FOLDER>/aerowebbV7_STANDARD_VIERGE_PG12_7.8.9-hf3_20240930.backup >
~/tmp/logs/pg12/std/dump_aw.log 2> ~/tmp/logs/pg12/std/dump_aw_error.log
```