Installing or Updating an Extension

PostgreSQL supports a wide range of extensions that add extra functionality to the core database system. Extensions like uuid-ossp, pg_trgm, and postgis are often used to provide features for text search, spatial data, UUID generation, and more. If you are running PostgreSQL on Elestio, you can enable many of these extensions directly within your database. This document explains how to enable, manage, and troubleshoot PostgreSQL extensions in an Elestio-hosted environment. It also includes guidance on checking extension compatibility with different PostgreSQL versions.

Installing and Enabling Extensions

PostgreSQL extensions can be installed in each database individually. Most common extensions are included in the PostgreSQL installation on Elestio. To enable an extension, you need to connect to your database using a tool like psql.

Start by connecting to your PostgreSQL database. You can follow the detailed documentation as provided here.

Once connected, you can enable an extension using the CREATE EXTENSION command. For example, to enable the uuid-ossp extension:

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
```

To check which extensions are already installed in your current database, use the `\dx` command within psql. If you want to see all available extensions on the server, use:

```
SELECT * FROM pg_available_extensions ORDER BY name;
```

If the extension you need is not listed in the available extensions, it may not be installed on the server.

Checking Extension Compatibility

Each PostgreSQL extension is built for a specific PostgreSQL version. Not all extensions are compatible across major versions. Before upgrading PostgreSQL or deploying an extension, it is important to check whether the extension is compatible with the version you are using.

To check the installed version of an extension and the default version provided by the system, run:

```
SELECT name, default_version, installed_version
FROM pg_available_extensions
WHERE name = 'pg_trgm';
```

If you are planning to upgrade your PostgreSQL version, it is recommended to deploy a new instance with the target version and run the above query to see if the extension is available and compatible. Some extensions may require specific builds for each version of PostgreSQL. After upgrading your database, you may also need to update your extensions using:

```
ALTER EXTENSION <extension_name> UPDATE;
```

This ensures the extension objects in the database match the new database version.

Troubleshooting Common Extension Issues

There are some common issues users may encounter when working with extensions. These usually relate to missing files, permission problems, or version mismatches.

If you see an error like could not open extension control file, it means the extension is not installed on the server. This usually happens when the extension is not included in the PostgreSQL installation. If the error message says that the extension already exists, it means it has already been installed in the database. You can confirm this with the \dx command or the query:

```
SELECT * FROM pg_extension;
```

If you need to reinstall it, you can drop and recreate it. Be careful, as dropping an extension with CASCADE may remove objects that depend on it:

```
DROP EXTENSION IF EXISTS <extension_name> CASCADE;
CREATE EXTENSION <extension_name>;
```

Another common issue appears after upgrading PostgreSQL, where some functions related to the extension stop working. This is often due to the extension not being updated. Running the following command will usually fix this.

```
ALTER EXTENSION <name> UPDATE;
```

In some cases, you may get a permission denied error when trying to create an extension. This means your database role does not have the required privileges. You will need to connect using a superuser account like postgres, or request that Elestio enable the extension for you.

Revision #1
Created 2 April 2025 12:06:24 by kaiwalya
Updated 2 April 2025 15:36:48 by kaiwalya