

Roteiro de Instalação do PLDEBUGGER_14 no PostgreSQL 14 hospedado em Linux CentOS 7

MODIFICAÇÕES NESTE DOCUMENTO

Item: Criação do Documento

Alteração/Inclusão: todo o documento

Responsável: Gilberto A. Oliveira

Data: 27/12/2022

Índice

Objetivo.....	4
Pré-requisitos.....	4
Instalação do PLDEBUGGER_14.....	5
Validação da Instalação.....	6

Objetivo

O objetivo desse roteiro é servir como guia (howto) para:

- Instalação da extensão de debug para depurar código no banco de dados PostgreSQL 14 em um sistema operacional Linux CentOS 7.

O leitor desse documento deve saber que, neste guia não temos a preocupação de explicar os fundamentos dos comandos apresentados e nem as razões que orientaram as escolhas aqui destacadas.

Alerta: Este documento não tem a pretensão de substituir as documentações oficiais dos sistemas abordados. Recomenda-se as documentações oficiais do PostgreSQL 14 (<https://www.postgresql.org/docs/14/index.html>) e Linux CentOS 7 (<https://docs.centos.org/en-US/docs/>) para exaurir dúvidas e/ou fundamentos que não estejam cobertos por esse roteiro.

Pré-requisitos

Este material considera que o leitor já tenha um ambiente montado e configurado com o banco de dados PostgreSQL 14 em uma distribuição Linux CentOS 7, com todos os pré-requisitos de softwares instalados e configurados.

- PGSQLDEV-14
- gcc
- make

O leitor pode recorrer o roteiro Postgresql com Login_hook no CentOS 7, presente neste repositório, para atender os pré-requisitos desse documento.

Instalação do PLDEBUGGER_14

A extensão PLDEBUGGER_14 permite ao desenvolvedor a depuração do código de suas aplicações que interagem com o banco de dados PostgreSQL. Faz parte desse contexto a necessidade de instalar e criação a referida extensão. A documentação e download dos arquivos necessários estão disponíveis em <https://qtibia.com/how-to-install-the-pl-pgsql-debugger-and-use-it-in-pgadmin/>

Outras documentações de referência:

<https://www.techsupportpk.com/2020/12/how-to-install-pldebugger-centos.html>

<https://github.com/EnterpriseDB/pldebugger>

ATENÇÃO: Antes de continuar se certifique que os cuidados de pré-instalação pgsqlddev-14, make, gcc e path relatados anteriormente tenham sido tomados.

1 – instalar o pacote pldebugger_14 no CENTOS7

```
# sudo yum install pldebugger_14
#
```

2 – conectado no banco de dados criar a extensão

ATENÇÃO: O comando abaixo deve ser executado para cada novo banco!!

```
CREATE EXTENSION pg_cron;
```

```
# su - postgres
Último login: Seg Dez 26 14:10:17 -03 2022 em pts/0

postgres=# CREATE EXTENSION pldbgapi;
CREATE EXTENSION
postgres=#
```

3 – Acrescentar a extensão no arquivo de configuração POSTGRESQL.CONF

3.1 - Para ver o local do arquivo você deve conectar no banco de dados e executar o comando

```
SHOW config_file;
```

```
# su - postgres
Último login: Seg Dez 26 14:12:31 -03 2022 em pts/0

postgres=# SHOW config_file;

               config_file
-----
/postgresql/data/data/postgresql.conf
(1 row)

postgres=#
```

3.2 – Editar o arquivo postgresql.conf e acrescentar a nova extensão na linha

`shared_preload_libraries = 'plugin_debugger,pg_cron'`

```
#local_preload_libraries = ''
#session_preload_libraries = ''
session_preload_libraries = 'login_hook'

#shared_preload_libraries = '' # (change requires restart)
shared_preload_libraries = 'plugin_debugger,pg_cron' ←
#cron.database_name = 'postgres'
#jit_provider = 'llvmjit'           # JIT library to use

# - Other Defaults -
```

Você deve observar que nesse exemplo, o parâmetro `shared_preload_libraries` já continha a extensão `pg_cron` vinculada. Desse modo, a nova extensão foi adicionada com a vírgula como separador de parâmetro e delimitado pelas mesmas aspas.

3.3 – restart o serviço do PostgreSQL

`systemctl restart postgresql-14`

```
# systemctl restart postgresql-14
#
```

Validação da Instalação

1 – Alterne para o usuário postgres

`su - postgres`

```
# su - postgres
Último login: Sex Abr 22 12:14:39 -03 2022 em pts/0
-bash-4.2$
```

2 – Faça uma conexão com o banco de dados

`psql`

```
-bash-4.2$ psql
```

```
postgres=#
```

3 – Verifique se a biblioteca foi carregada.

`SHOW shared_preload_libraries;`

```
postgres=# SHOW shared_preload_libraries;
 shared_preload_libraries
-----
 plugin_debugger,pg_cron
(1 row)
```

```
postgres=#
```

O resultado esperado é o retorno dos nomes das bibliotecas, caso não apareça será necessário revisar os passos anteriores.