

# 2ndQuadrant

## BDR3 plugin for OmniDB

Version 1.0  
August 2019

2ndQuadrant

# Contents

<b>Introduction</b>	<b>3</b>
<b>Installation</b>	<b>3</b>

## Introduction

The goal of this document is to explain how to use OmniDB's BDR3 plugin do manage and monitor BDR3 clusters.

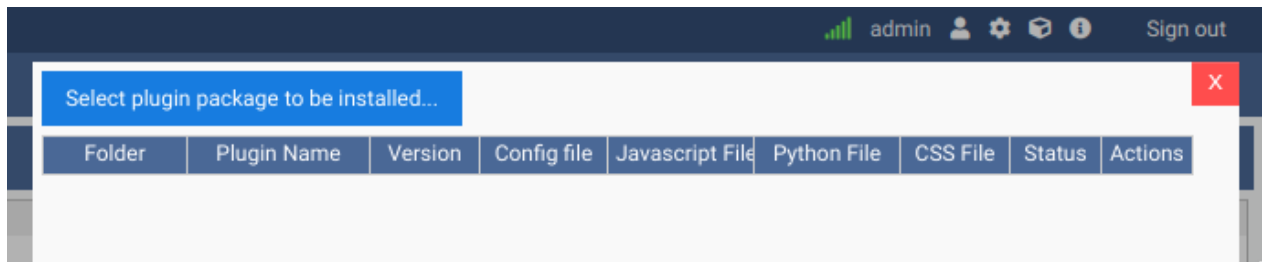
OmniDB is a desktop/web application that offers out-of-the-box features to manage community PostgreSQL.

By installing BDR3 plugin, OmniDB offers additional features specific to PostgreSQL instances that are part of a BDR3 cluster.

## Installation

After installing and opening OmniDB, to install the plugin you just have to download its package, which is a small zip package, and load it using OmniDB's interface.

Click on the **View Plugins** (box icon) at the top left part of the screen to view the list of installed plugins.



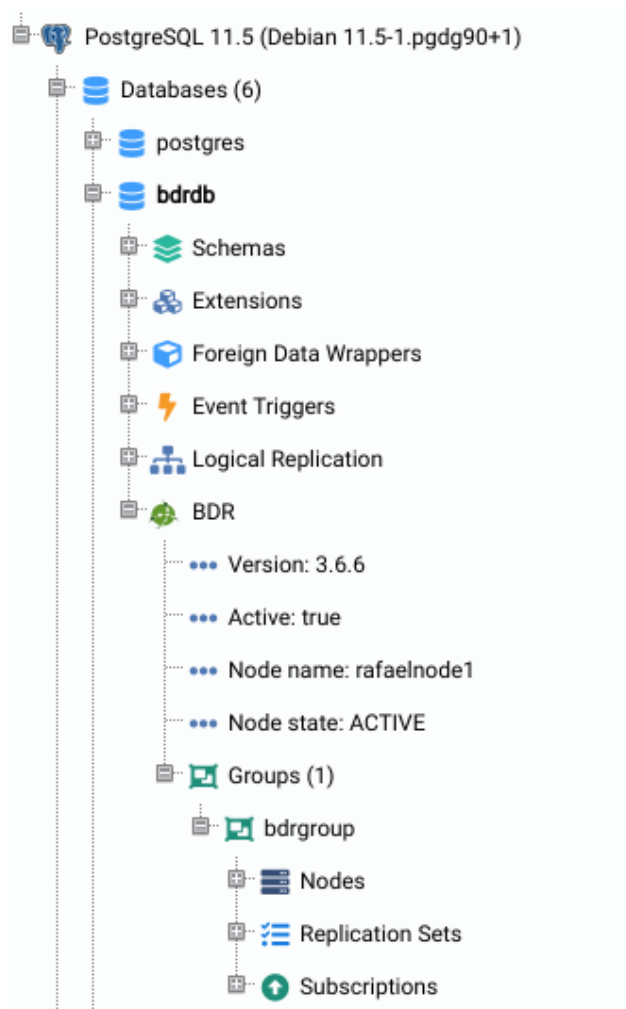
Click on **Select plugin package to be installed...** and select the downloaded package.

After installing, OmniDB will ask you to restart the application so the plugin's library is properly loaded. Looking at the plugins list after restarting should look like:

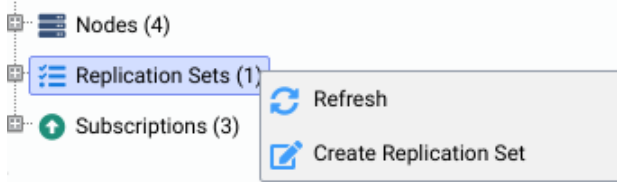
Folder	Plugin Name	Version	Config file	Javascript File	Python File	CSS File	Status	Actions
bdr3	bdr3	1.6.0	✓	✓	✓	✓	✓	✗

### # Usage

With the plugin properly installed, OmniDB will be able to detect if the database currently connected to has the BDR extension enabled and, if it does, include the BDR icon in the tree of structures, which is the starting point to perform BDR specific actions and manage the cluster:



Right-clicking tree nodes allow you to access specific features, for instance, adding replication sets:



## ## Monitoring

OmniDB comes with a monitoring dashboard containing default units to monitor a community PostgreSQL instance. The plugin extends this feature by adding units specifically to monitor a BDR cluster. Currently, users can monitor the following aspects of their BDR clusters:

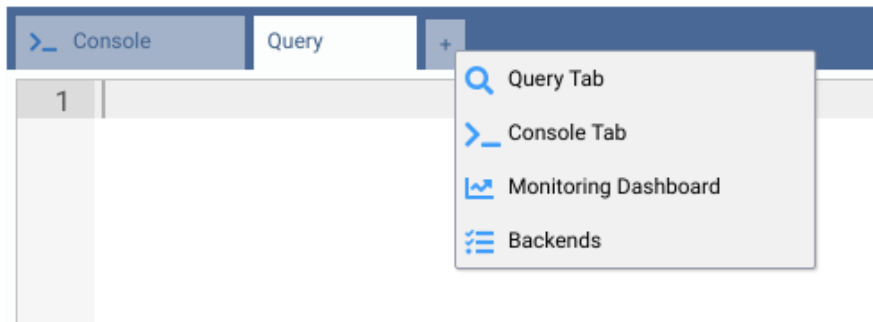
- BDR write lag (seconds): Time elapsed between flushing recent WAL locally and receiving notification that the peer node has written it
- BDR flush lag (seconds): Time elapsed between flushing recent WAL locally and receiving

notification that the peer node has flushed it

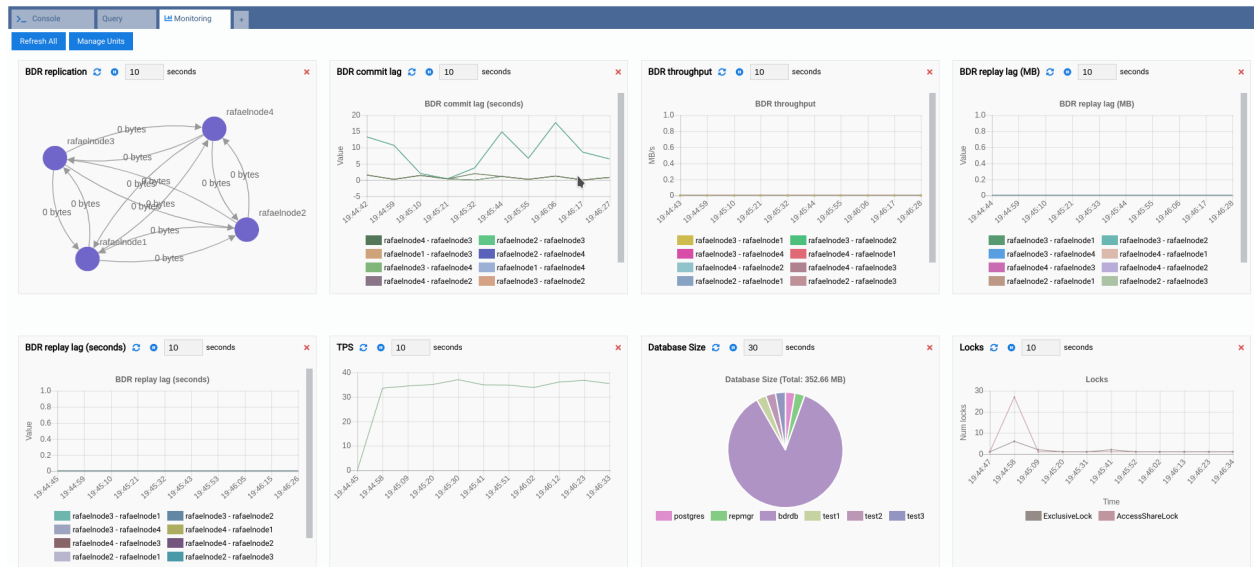
- BDR replay lag (seconds): Time elapsed between flushing recent WAL locally and receiving notification that the peer node has replayed it
- BDR write lag (MB): Difference in WAL positions between the local server's current WAL write position and the last position confirmed written by the peer node
- BDR flush lag (MB): Difference in WAL positions between the local server's current WAL write position and the last position confirmed flushed by the peer node
- BDR replay lag (MB): Difference in WAL positions between the local server's current WAL write position and the last position confirmed replayed by the peer node
- BDR throughput: Replication rate (MB/s) between local and peer node
- BDR commit lag: Time elapsed since the last commit applied locally, created by the peer node
- BDR replication: Graph containing nodes, representing the BDR3 nodes, and edges, representing replication between nodes. Edges contain information about lag in MB between its nodes

BDR specific charts will present information between all nodes, not only between the local and peer nodes.

To access the monitoring dashboard just click in the + in the internal tab list:



If the dashboard was never used for that particular connection, OmniDB will open default units, which includes several BDR only charts:



Users can customize their dashboards by selecting the units they find more useful. Future accesses to the dashboard will re-open its saved state.