



EDB Postgres Distributed Command Line Interface

リリース *1.0.0*

EDB

2022 年 06 月 04 日

目次

第 1 章	PGD CLI Release notes	3
1.1	Release notes for PGD CLI version 1	3
第 2 章	Requirements	5
2.1	特権	5
2.2	サポートされているバージョン	5
第 3 章	Installing the EDB Postgres Distributed Command Line Interface	7
3.1	TPAexec でインストール	7
3.2	手動インストール	7
3.2.1	データベースコネクション文字列の指定	7
3.2.2	構成ファイルの使用	8
第 4 章	Using the EDB Postgres Distributed Command Line Interface	9
4.1	構成ファイルの指定	9
4.2	データベースコネクション文字列を渡す	9
第 5 章	pgd	11
5.1	概要	11
5.2	オプション	11
第 6 章	check-health	13
6.1	概要	13
6.2	例	13
6.3	オプション	14
6.4	親コマンドから継承されたオプション	14
第 7 章	show-camo	15
7.1	概要	15
7.2	例	15
7.3	オプション	16
7.4	親コマンドから継承されたオプション	16
第 8 章	show-clockskew	17
8.1	概要	17
8.2	例	17
8.3	オプション	18
8.4	親コマンドから継承されたオプション	18
第 9 章	show-events	19
9.1	概要	19
9.2	例	19

9.3	オプション	20
9.4	親コマンドから継承されたオプション	20
第 10 章	show-nodes	21
10.1	概要	21
10.2	例	22
10.3	オプション	23
10.4	親コマンドから継承されたオプション	23
第 11 章	show-raft	25
11.1	概要	25
11.2	例	25
11.3	オプション	26
11.4	親コマンドから継承されたオプション	26
第 12 章	show-replslots	27
12.1	概要	27
12.2	例	27
12.3	オプション	30
12.4	親コマンドから継承されたオプション	30
第 13 章	show-subscriptions	31
13.1	概要	31
13.2	例	31
13.3	オプション	33
13.4	親コマンドから継承されたオプション	33
第 14 章	show-version	35
14.1	概要	35
14.2	例	35
14.3	オプション	36
14.4	親コマンドから継承されたオプション	36

EDB Postgres 分散コマンドラインインターフェイス (PGD CLI) は、BDR クラスターを管理するためのツール。BDR クラスターに対してコマンドを実行できます。

これを使用して、クラスターリソースを検査および管理し、次に関する情報を取得できます。

- ノードの概要
- 各ノードの BDR/ Postgres バージョン
- レプリケーションとサブスクリプションの詳細
- メンバーシップとワーカーイベント
- ラフトコンセンサスの詳細
- CAMO (Commit at Most once) の詳細
- クロックスキュー
- クラスター状態の概要

第1章 PGD CLI Release notes

PGD CLI は、BDR クラスターを管理するためのツール。BDR クラスターに対してコマンドを実行できます。

PGD CLI の文書には、マイナーリリースとパッチを含む PGD CLI の最新バージョンが説明されています。このセクションのリリースノートには、各リリースの新機能に関する情報が記載されています。

Version	Release Date
<i>Release notes for PGD CLI version 1</i>	2022 May 17

1.1 Release notes for PGD CLI version 1

これは、EDB Postgres Distributed (PGD) コマンドラインインターフェイス (CLI) の初期リリースです。PGD CLI バージョン 1 には以下が含まれます。

Type	Description
Feature	PGD CLI の初期リリース。次のコマンドが含まれます。
	<i>pgd check-health</i> - BDR クラスターの状態を確認します。
	<i>pgd show-camo</i> - BDR CAMO (1 回だけコミット) の詳細を表示します。
	ペア各 BDR ノード間のクロックスキューのステータスを表示します。
	<i>pgd show-events</i> - バックグラウンドワーカエラーやノードメンバーシップの変更などのイベントを表示します。
	<i>pgd show-nodes</i> - BDR クラスター内のすべてのノードとその概要を表示します。
	<i>pgd show-raft</i> - BDR Raft (コンセンサスプロトコル) の詳細を表示します。
	<i>pgd show-replslots</i> - BDR レプリケーションスロットのステータスを表示します。
	<i>pgd show-subscriptions</i> - BDR サブスクリプション (着信レプリケーション) の詳細を表示します。
	<i>pgd show-version</i> - 各ノードにインストールされている BDR と Postgres のバージョンを表示します。

第2章 Requirements

2.1 特権

PGD CLI を実行するには、postgres のスーパーユーザ権限が必要です。

2.2 サポートされているバージョン

PGD CLI は BDR 4.1 以降をサポートしています。

第3章 Installing the EDB Postgres Distributed Command Line Interface

3.1 TPAexec でインストール

TPAexec (v22.14 以降) を使用して EDB Postgres Distributed を展開している場合、デフォルトで各 BDR ノードに PGD CLI パッケージ (`edb-pgd-cli`) がインストールされます。また、クラスター内の各ノードのエンドポイントが事前に入力された構成ファイル (`/etc/edb/pgd-config.yml`) を作成するため、追加の構成は必要ありません。

クラスター内の非 BDR インスタンスに PGD CLI をインストールする場合は、展開する前に TPAexec の構成ファイルでそのインスタンスに `pgdcli` ロールをアタッチするだけです。詳細については、[TPAexec でインストール](#) を参照してください。

3.2 手動インストール

BDR repository から入手可能な `.deb` および `.rpm`

パッケージソフトを使用して、Linux マシンに PGD CLI を手動でインストールできます。パッケージ名前は `edb-pgd-cli` です。例：

```
# for Debian
sudo apt-get install edb-pgd-cli
```

PGD CLI が TPAexec で構成されている場合、自動的に接続されますが、マニュアルインストールでは BDR クラスターへの接続文字列を指定する必要があります。

3.2.1 データベースコネクション文字列の指定

構成ファイルを使用して、クラスターのデータベースコネクション文字列を指定するか（次のセクションを参照）、接続文字列 sample use case を参照）。

3.2.2 構成ファイルの使用

pgd-config.yml 構成ファイルを使用して、クラスターのデータベースコネクション文字列を指定します。構成ファイルには、クラスタ内の少なくとも 1 つの BDR ノードのデータベースコネクション文字列が含まれている必要があります。クラスター名前はオプションであり、検証されません。

例：

```
cluster:
  name: cluster-name
  endpoints:
    - "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
    - "host=bdr-b1 port=5432 dbname=bdrdb user=postgres "
    - "host=bdr-c1 port=5432 dbname=bdrdb user=postgres "
```

pgd-config.yml は、デフォルトで/etc/edb ディレクトリにあります。PGD CLI は、次の場所で pgd-config.yml を検索します（優先オーダー- 高いものから低いもの）。

1. /etc/edb（デフォルト）
2. \$HOME/.edb
3. .（作業ディレクトリ）

ファイルの名前を変更するか、別の場所に移動する場合は、コマンドを入力するときにオプションの -f または --config-file フラグを使用して、新しい名前と場所を指定します。

sample use case を参照してください。

第4章 Using the EDB Postgres Distributed Command Line Interface

pgd は、PGD コマンドラインインタフェースのコマンド名前です。

コマンドの構文は次のとおりです。

```
pgd -f, --config-file string    configuration file (default
                                "/etc/edb/pgd-config.yml")
    --dsn string                database connection string
    -h, --help                  help for pgd
    -L, --log-level string      logging level: debug, info,
                                warn, error (default "error")
    -o, --output string         output format: json, yaml
```

次のフラグのサンプル ユースケースを参照してください。

- -f, --config-file
- -dsn
- -o, --output
- -h, --help

4.1 構成ファイルの指定

ファイルの名前を変更するか、別の場所に移動する場合は、オプションの -f または --config-file フラグを使用して新しい名前と場所を指定します。例：

```
pgd show-nodes -f /opt/my-config.yml
```

4.2 データベースコネクション文字列を渡す

--dsn フラグを使用して、データベースコネクション文字列をコマンドに直接渡します。このフラグを使用して接続文字列を渡す場合、構成ファイルは必要ありません。構成ファイルが存在する場合、フラグが優先されます。例：

```
pgd show-nodes --dsn "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
``` ## 出力フォーマットの指定
```

PGD CLI は、次の出力形式をサポートしています。

```
.. csv-table::
:header: Format,Considerations
:widths: 10,30
:class: longtable

tabular,Default format. Presents the data in tabular form.
json,"Presents the raw data with no formatting. For some commands, the json output
→may show more data than shown in the tabular output such as extra fields and more
→detailed messages."
yaml,Same as json except field order is alphabetical. Experimental and may not be
→fully supported in future versions.
```

`-o` または `--output` フラグを使用して、デフォルトの出力フォーマットを json または yaml に変更します。例：

```
```sh
pgd show-nodes -o json
``` ##コマンドラインヘルプへのアクセス
```

サポートされているコマンドをリストするには、次のように入力します。

```
```sh
pgd help
```

特定のコマンドとそのパラメーターのヘルプを表示するには、`pgd help <command_name>` と入力します。例：

```
pgd help show-nodes
```

第5章 pgd

pgd は、PGD コマンドラインインタフェースのコマンド名前です。

5.1 概要

EDB Postgres 分散コマンドラインインターフェイス (PGD CLI) は、BDR クラスターを管理するためのツール。BDR クラスターに対してコマンドを実行できます。これを使用して、クラスターリソースを検査および管理できます。

5.2 オプション

<code>-f, --config-file string</code>	config file; ignored if --dsn flag is present (default "/etc/edb/pgd-config.yml")
<code>--dsn string</code>	database connection string e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
<code>-h, --help</code>	help for pgd
<code>-L, --log-level string</code>	logging level: debug, info, warn, error (default "error")
<code>-o, --output string</code>	output format: json, yaml

第6章 check-health

BDR クラスターのヘルスを確認します。

6.1 概要

すべてのノードにアクセスできるか、すべてのレプリケーションスロットが動作しているか、CAMO ペアが接続されているかなど、さまざまなチェックを実行します。

```
pgd check-health [flags]
```

6.2 例

Example 1 (3 node cluster, bdr-a1 and bdr-c1 are up, bdr-b1 is down; bdr-a1 and bdr-b1 are CAMO partners)

```
$ pgd check-health
```

Check	Status	Message
-----	-----	-----
CAMO	Critical	At least 1 CAMO partner is not connected
ClockSkew	Critical	Clockskew cannot be determined for at least 1 BDR node pair
Connection	Critical	The node bdr-b1 is not accessible
Raft	Warning	There is no RAFT_LEADER, an election might be in progress
Replslots	Critical	There is at least 1 BDR replication slot which is inactive
Version	Warning	There is at least 1 node that is not accessible

Example 2 (3 node cluster, all nodes are up but system clocks are not in sync)

```
$ pgd check-health
```

Check	Status	Message
-----	-----	-----
CAMO	Ok	All CAMO pairs are connected
ClockSkew	Warning	At least 1 BDR node pair has clockskew greater than 2 seconds
Connection	Ok	All BDR nodes are accessible

(次のページに続く)

(前のページからの続き)

```

Raft      Ok      Raft Consensus is working correctly
Replslots Ok      All BDR replication slots are working correctly
Version   Ok      All nodes are running same BDR versions

```

Example 3 (3 node cluster, all nodes are up and all checks are Ok)

```
$ pgd check-health
```

```

Check      Status Message
-----
CAMO       Ok      All CAMO pairs are connected
ClockSkew  Ok      All BDR node pairs have clockskew within permissible limit
Connection Ok      All BDR nodes are accessible
Raft       Ok      Raft Consensus is working correctly
Replslots  Ok      All BDR replication slots are working correctly
Version    Ok      All nodes are running same BDR versions

```

6.3 オプション

```
-h, --help  help for check-health
```

6.4 親コマンドから継承されたオプション

```

-f, --config-file string  config file; ignored if
                           --dsn flag is present (default "/etc/edb/pgd-config.yml")
--dsn string              database connection string
                           e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
-L, --log-level string    logging level: debug, info, warn, error (default "error")
-o, --output string       output format: json, yaml

```

第7章 show-camo

BDR CAMO (Commit at Most once) の詳細を表示します。

7.1 概要

CAMO パートナーの名前、接続と準備状況、保留中および未解決の CAMO トランザクション、`apply_lsn` と `receive_lsn` の違いなどの BDR CAMO (コミット アット ワンス) の詳細を表示します。このコマンドは、EDB Postgres Extended および EDB Postgres Advanced Server (v14 以降) でのみ使用できます。

```
pgd show-camo [flags]
```

7.2 例

Example 1 (3 node cluster, bdr-a1 and bdr-b1 are CAMO partner but bdr-b1 is down)

```
$ pgd show-camo
```

Node	CAMO Partner	Connected	Ready	Transactions Resolved	Apply LSN	Receive LSN	Apply Queue Size
bdr-a1	bdr-b1	false	false	true	0/E42C99B0	0/E42C99B0	0

Example 2 (3 node cluster, bdr-b1 was down and it has just been restarted)

```
$ pgd show-camo
```

Node	CAMO Partner	Connected	Ready	Transactions Resolved	Apply LSN	Receive LSN	Apply Queue Size
bdr-a1	bdr-b1	true	true	true	0/E533DAB8	0/E533DAB8	0
bdr-b1	bdr-a1	true	false	true	3/7AE81A28	3/7AE81A28	0

Example 3 (3 node cluster, all nodes are up and in streaming state)

(次のページに続く)

(前のページからの続き)

```
$ pgd show-camo
```

```

Node   CAMO Partner Connected Ready Transactions Resolved Apply LSN  Receive LSN
→Apply Queue Size
-----
→-----
bdr-a1 bdr-b1      true      true  true                0/E56AE520 0/E56AE520 0
bdr-b1 bdr-a1      true      true  true                3/7B180BA8 3/7B180BA8 0

```

7.3 オプション

```
-h, --help    help for show-camo
```

7.4 親コマンドから継承されたオプション

```

-f, --config-file string  config file; ignored if
                           --dsn flag is present (default "/etc/edb/pgd-config.yml")
    --dsn string          database connection string
                           e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
-L, --log-level string    logging level: debug, info, warn, error (default "error")
-o, --output string       output format: json, yaml

```

第8章 show-clockskew

ペア BDR ノードのクロックスキューのステータスを表示します。

8.1 概要

クラスタ内のペア BDR ノードのクロックスキューのステータスを表示します。

記号 意味 --- - --- * ok ~ ワーニング (skew > 2 秒) !クリティカル (スキュー> 5 秒) ×ダウン/到達不能? 不明 - 該当なし

```
pgd show-clockskew [flags]
```

8.2 例

Example 1 (3 node cluster, bdr-a1 and bdr-c1 are up, bdr-b1 is down)

```
$ pgd show-clockskew
```

Node	bdr-a1	bdr-b1	bdr-c1	Current Time
bdr-a1	*	?	*	2022-03-30 07:02:21.334472
bdr-b1	x	*	x	x
bdr-c1	*	?	*	2022-03-30 07:02:21.186809

Example 2 (3 node cluster, all nodes are up)

```
$ pgd show-clockskew
```

Node	bdr-a1	bdr-b1	bdr-c1	Current Time
bdr-a1	*	*	*	2022-03-30 07:04:54.147017
bdr-b1	*	*	*	2022-03-30 07:04:54.340543
bdr-c1	*	*	*	2022-03-30 07:04:53.90451

8.3 オプション

<code>-h, --help</code> help for show-clockskew

8.4 親コマンドから継承されたオプション

<code>-f, --config-file string</code>	config file; ignored if --dsn flag is present (default "/etc/edb/pgd-config.yml")
<code>--dsn string</code>	database connection string e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
<code>-L, --log-level string</code>	logging level: debug, info, warn, error (default "error")
<code>-o, --output string</code>	output format: json, yaml

第9章 show-events

バックグラウンドワーカエラーやノードメンバーシップの変更などのイベントを表示します。

9.1 概要

バックグラウンドワーカエラーやノードメンバーシップの変更などのイベントを表示します。出力は Time 列のオーダーでソートされます。メッセージ列は数行後に切り捨てられます。完全なメッセージをビューするには、json 出力フォーマット（「-o json」）を使用します。

各ノードの状態の詳細については、show-nodes コマンドのヘルプ（「pgd show-nodes -h」）を参照してください。

```
pgd show-events [flags]
```

9.2 例

Example 1 (3 node cluster)

```
$ pgd show-events --lines 10
```

Time	Observer Node	Subject Node	Type	Message
2022-04-19 19:45:43.077712+00	bdr-b1	bdr-c1	receiver worker error	pglogical worker received fast finish request, exiting
2022-04-19 19:45:43.066804+00	bdr-c1	bdr-a1	receiver worker error	pglogical worker received fast finish request, exiting
2022-04-19 19:45:43.057598+00	bdr-b1	bdr-a1	receiver worker error	pglogical worker received fast finish request, exiting
2022-04-19 19:45:43.046515+00	bdr-c1	bdr-b1	receiver worker error	pglogical worker received fast finish request, exiting
2022-04-19 19:45:43.033369+00	bdr-a1	bdr-c1	receiver worker error	pglogical worker received fast finish request, exiting
2022-04-19 19:45:43.013203+00	bdr-a1	bdr-b1	receiver worker error	pglogical worker received fast finish request, exiting
2022-04-19 19:45:40.024662+00	bdr-c1	bdr-c1	node state change	ACTIVE
2022-04-19 19:45:40.024575+00	bdr-b1	bdr-c1	node state change	ACTIVE

(次のページに続く)

(前のページからの続き)

2022-04-19 19:45:40.022788+00	bdr-a1	bdr-c1	node state change	ACTIVE
2022-04-19 19:45:38.961424+00	bdr-c1	bdr-c1	node state change	
 PROMOTING				

9.3 オプション

-h, --help	help for show-events
-n, --lines int	show top n lines

9.4 親コマンドから継承されたオプション

-f, --config-file string	config file; ignored if --dsn flag is present (default "/etc/edb/pgd-config.yml")
--dsn string	database connection string e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
-L, --log-level string	logging level: debug, info, warn, error (default "error")
-o, --output string	output format: json, yaml

第10章 show-nodes

BDR クラスター内のすべてのノードとその概要を表示します。

10.1 概要

BDR クラスター内のすべてのノードと、名前、ノード ID、グループ、現在/ターゲットの状態などの概要を表示します。

ノードの状態

- **NONE**：ワーカーの起動時にノードの状態は設定解除され、すぐに現在の既知の状態に設定されることが予想されます。
- **CREATED**：`bdr.create_node()` は実行されましたが、ノードはまだ BDR クラスターのメンバではありません。
- **JOIN_START**：`bdr.join_node_group()` は、ローカルノードを既存の BDR クラスターに結合させ始めます。
- **JOINING**：ノード結合が開始され、現在初期同期フェーズにあり、ノードにスキーマとデータを作成しています。
- **CATCHUP**：初期同期フェーズが完了しました。これで、結合は、結合が開始されてから上流の同等なノードとされたトランザクションを取得して適用する最後のステップになりました。
- **STANDBY**：ノード結合は完了しましたが、変更のブロードキャストはまだ開始されていません。すべての参加はこの状態でしばらくかかりますが、ロジカルスタンバイとして定義されている場合、ノードはこの状態で続行されます。
- **PROMOTE**：ノードはロジカルスタンバイであり、`bdr.promote_node` を呼び出してノードの状態を **ACTIVE** に移動しました。これら 2 つの **PROMOTE** 状態は、1 つのノードのみが **STANDBY** よりも高く、**ACTIVE** よりも低い状態にできるという事実と一貫している必要があります。
- **PROMOTING**：ロジカルスタンバイからフル BDR ノードへの昇格が進行中です。
- **ACTIVE**：ノードはフル BDR ノードであり、現在 **ACTIVE** です。これは、最も一般的なノードのステータスです。
- **PART_START**：ノードは **ACTIVE** または **STANDBY** で、`bdr.part_node` を呼び出して BDR クラスターからノードを削除しました。
- **PARTING**：ノードは他のノードから切断し、コンセンサスまたはレプリケーションでそれ以上のパートを果たしません。

- PART_CATCHUP : 非分割ノードは、最近分割されたノードから欠落しているデータを同期します。
- PARTED : ノードの分割オペレーションがすべてのノードで完了しました。

一度に1つのノードのみが、状態 PROMOTE または PROMOTING のいずれかになります。

[現在の状態] または [ターゲットの状態] 列に示されているように、ノードの読み取り専用状態は STANDBY と示されていることに注意してください。

```
pgd show-nodes [flags]
```

10.2 例

Example 1 (3 node cluster, bdr-a1 and bdr-c1 are up, bdr-b1 is down)

```
$ pgd show-nodes
```

Node	Node ID	Node Group	Current State	Target State	Status	Seq ID
bdr-a1	3136956818	bdrgroup	ACTIVE	ACTIVE	Up	1
bdr-b1	2380210996	bdrgroup	ACTIVE	ACTIVE	Unreachable	2
bdr-c1	1804769977	bdrgroup	ACTIVE	ACTIVE	Up	3

Example 2 (3 node cluster, all nodes are up)

```
$ pgd show-nodes
```

Node	Node ID	Node Group	Current State	Target State	Status	Seq ID
bdr-a1	3136956818	bdrgroup	ACTIVE	ACTIVE	Up	1
bdr-b1	2380210996	bdrgroup	ACTIVE	ACTIVE	Up	2
bdr-c1	1804769977	bdrgroup	ACTIVE	ACTIVE	Up	3

Example 3 (cluster with witness, logical standby and subscriber-only nodes)

Note: In contrast to logical standby, the subscriber-only nodes are fully joined node to the cluster

```
$ pgd show-nodes
```

Node	Node ID	Node Group	Current State	Target State	Status	Seq ID
bdr-a1	3136956818	bdrgroup	ACTIVE	ACTIVE	Up	1
bdr-b1	2380210996	bdrgroup	ACTIVE	ACTIVE	Up	2
witness-c1	1450365472	bdrgroup	ACTIVE	ACTIVE	Up	3
logical-standby-a1	1140256918	bdrgroup	STANDBY	STANDBY	Up	4

(次のページに続く)

(前のページからの続き)

logical-standby-b1	3541792022	bdrgroup	STANDBY	STANDBY	Up	5
subscriber-only-c1	2448841809	subscriber-only	ACTIVE	ACTIVE	Up	6

10.3 オプション

`-h, --help` help for show-nodes

10.4 親コマンドから継承されたオプション

`-f, --config-file string` config file; ignored if
`--dsn string` database connection string
 e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
`-L, --log-level string` logging level: debug, info, warn, error (default "error")
`-o, --output string` output format: json, yaml

第11章 show-raft

BDR Raft（コンセンサスプロトコル）の詳細を表示します。

11.1 概要

Raft 状態（リーダー、フォロワー）、Raft 選挙 ID、投票ノード数などの BDR Raft（コンセンサスプロトコル）の詳細を表示します。

注：ネットワークパーティションなどの場合、出力は CLI が接続されているノードによって異なる場合があります。

```
pgd show-raft [flags]
```

11.2 例

Example 1 (3 node cluster, bdr-a1 and bdr-c1 are up, bdr-b1 is down)

```
$ pgd show-raft
```

Node	Raft State	Raft Term	Commit Index	Nodes	Voting Nodes	Protocol Version
bdr-c1	RAFT_FOLLOWER	29	6081272	3	3	4002
bdr-a1	RAFT_LEADER	29	6081272	3	3	4002
bdr-b1						

Example 2 (3 node cluster, all nodes are up)

```
$ pgd show-raft
```

Node	Raft State	Raft Term	Commit Index	Nodes	Voting Nodes	Protocol Version
bdr-c1	RAFT_FOLLOWER	38	6132327	3	3	4002
bdr-a1	RAFT_LEADER	38	6132331	3	3	4002
bdr-b1	RAFT_FOLLOWER	38	6132336	3	3	4002

Example 3 (3 node cluster, all nodes are up but bdr-a1 is not able to connect)

(次のページに続く)

(前のページからの続き)

```
to other nodes; following is the output when cli is connected to bdr-a1)

$ pgd show-raft

Node      Raft State      Raft Term Commit Index Nodes Voting Nodes Protocol Version
-----
bdr-c1
bdr-a1 RAFT_FOLLOWER 40          6176769      3      3          4002
bdr-b1

Example 4 (cluster with witness, logical standby and subscriber-only nodes)
Note: Unlike full-bdr (or witness node), logical standby and subscriber-only
nodes dont have raft voting rights.

$ pgd show-raft

Node      Raft State      Raft Term Commit Index Nodes Voting Nodes Protocol Version
-----
bdr-a1      RAFT_LEADER    0           10268        6      3          4003
bdr-b1      RAFT_FOLLOWER  0           10279        6      3          4003
witness-c1  RAFT_FOLLOWER  0           10281        6      3          4003
logical-standby-a1 RAFT_FOLLOWER  0           10281        6      3          4003
logical-standby-b1 RAFT_FOLLOWER  0           10281        6      3          4003
subscriber-only-c1 RAFT_FOLLOWER  0           10281        6      3          4003
```

11.3 オプション

```
-h, --help    help for show-raft
```

11.4 親コマンドから継承されたオプション

```
-f, --config-file string  config file; ignored if
                           --dsn flag is present (default "/etc/edb/pgd-config.yml")
                           --dsn string      database connection string
                                           e.g."host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
-L, --log-level string    logging level: debug, info, warn, error (default "error")
-o, --output string       output format: json, yaml
```

第12章 show-replslots

BDR レプリケーションスロットのステータスを表示します。

12.1 概要

BDR レプリケーションスロットのステータスを表示します。verbose フラグを指定した出力は、スロットがアクティブかどうか、レプリケーション状態（切断、ストリーミング、キャッチアップ）、おおよそのラグなどの詳細を提供します。

記号 意味 --- - --- * ok ~ ワーニング (lag > 10M) !クリティカル (ラグ> 100M OR スロットが「非アクティブ」または「切断」) x ダウン/到達不能-n/a

マトリクスビューでは、バイトがカッコに表示される場合があります。maxOf(WriteLag, FlushLag, ReplayLag, SentLag) です。

```
pgd show-replslots [flags]
```

12.2 例

Example 1 (3 node cluster, bdr-a1 and bdr-c1 are up, bdr-b1 is down)

```
$ pgd show-replslots
```

```
Node    bdr-a1 bdr-b1 bdr-c1
----    -
bdr-a1  *      ! (6.6G) *
bdr-b1  x      *      x
bdr-c1  *      ! (6.9G) *
```

```
$ pgd show-replslots --verbose
```

```
Origin Node Target Node Status (active/state) Write Lag (bytes/duration) Flush Lag
↳(bytes/duration)      Replay Lag (bytes/duration) Sent Lag (bytes)
-----
↳-----
bdr-a1      bdr-b1      f / disconnected      6.6G / 8 days 02:58:36.243723 6.6G / 8
↳days 02:58:36.243723 6.6G / 8 days 02:58:36.243723 6.6G
```

(次のページに続く)

(前のページからの続き)

```

bdr-a1      bdr-c1      t / streaming      0B / 00:00:00      0B / 00:00:00
→00:00:00      0B / 00:00:00      0B
bdr-c1      bdr-a1      t / streaming      0B / 00:00:00.000812      0B / 00:00:00.000812
→00:00:00.000812      0B / 00:00:00.000812      0B
bdr-c1      bdr-b1      f / disconnected      6.9G / 8 days 02:58:36.004415 6.9G / 8 days 02:58:36.004415 6.9G / 8 days 02:58:36.004415 6.9G
→days 02:58:36.004415 6.9G / 8 days 02:58:36.004415 6.9G

```

Example 2 (3 node cluster, bdr-b1 was down and it has just been restarted)

```
$ pgd show-replslots
```

```

Node      bdr-a1 bdr-b1 bdr-c1
-----
bdr-a1 *      ! (6.9G) *
bdr-b1 *      *      *
bdr-c1 *      ! (5.8G) *

```

```
$ pgd show-replslots --verbose
```

```

Origin Node Target Node Status (active/state) Write Lag (bytes/duration) Flush Lag
→(bytes/duration) Replay Lag (bytes/duration) Sent Lag (bytes)

```

```

-----
→-----
bdr-a1      bdr-b1      t / catchup      6.9G / 00:00:00.000778      6.9G / 00:00:00.000778
→00:00:00.000778      6.9G / 00:00:00.000778      6.9G
bdr-a1      bdr-c1      t / streaming      0B / 00:00:00.104121      0B / 00:00:00.104133
→00:00:00.104133      0B / 00:00:00.104133      0B
bdr-b1      bdr-a1      t / streaming      0B / 00:00:00      0B / 00:00:00
→00:00:00      0B / 00:00:00      0B
bdr-b1      bdr-c1      t / streaming      0B / 00:00:00      0B / 00:00:00
→00:00:00      0B / 00:00:00      0B
bdr-c1      bdr-a1      t / streaming      6.8K / 00:00:00      6.8K / 00:00:00
→00:00:00      6.8K / 00:00:00      6.8K
bdr-c1      bdr-b1      t / catchup      5.5G / 00:00:00.008257      5.5G / 00:00:00.008257
→00:00:00.008257      5.5G / 00:00:00.008257      5.5G

```

Example 3 (3 node cluster, all nodes are up and in streaming state)

```
$ pgd show-replslots
```

```

Node      bdr-a1 bdr-b1 bdr-c1
-----
bdr-a1 *      *      *
bdr-b1 *      *      *
bdr-c1 *      *      *

```

(次のページに続く)

(前のページからの続き)

```
$ pgd show-replslots --verbose
```

```
Origin Node Target Node Status (active/state) Write Lag (bytes/duration) Flush Lag
↳(bytes/duration) Replay Lag (bytes/duration) Sent Lag (bytes)
```

```
-----
↳-----
bdr-a1      bdr-b1      t / streaming      0B / 00:00:00      0B / ↳
↳00:00:00      0B / 00:00:00      0B
bdr-a1      bdr-c1      t / streaming      0B / 00:00:00      0B / ↳
↳00:00:00      0B / 00:00:00      0B
bdr-b1      bdr-a1      t / streaming      0B / 00:00:00      0B / ↳
↳00:00:00      0B / 00:00:00      0B
bdr-b1      bdr-c1      t / streaming      0B / 00:00:00      0B / ↳
↳00:00:00      0B / 00:00:00      0B
bdr-c1      bdr-a1      t / streaming      0B / 00:00:00      528B / ↳
↳00:00:00      528B / 00:00:00      0B
bdr-c1      bdr-b1      t / streaming      528B / 00:00:00      528B / ↳
↳00:00:00      528B / 00:00:00      0B
```

Example 4 (cluster with witness, logical standby and subscriber-only nodes;
upstream for logical-standby-a1 is bdr-a1 and for logical-standby-b1 it is bdr-b1)

Note:

1. A logical standby is sent data only by one source node, but no other nodes receive replication changes from it
2. Subscriber-only node subscribes to replication changes from other nodes in the cluster, but no other nodes receive replication changes from it

```
$ pgd show-replslots
```

```
Node          bdr-a1 bdr-b1 logical-standby-a1 logical-standby-b1 subscriber-
↳only-c1 witness-c1
```

```
-----
↳-----
bdr-a1          *      *      *          -          *
      *
bdr-b1          *      *      -          *          *
      *
logical-standby-a1 -      -      *          -          -
      -
logical-standby-b1 -      -      -          *          -
      -
subscriber-only-c1 -      -      -          -          *
      -
witness-c1      *      *      -          -          *
```

*

(次のページに続く)

12.3 オプション

<code>-h, --help</code>	help for show-replslots
<code>-v, --verbose</code>	verbose output (default true)

12.4 親コマンドから継承されたオプション

<code>-f, --config-file string</code>	config file; ignored if --dsn flag is present (default "/etc/edb/pgd-config.yml")
<code>--dsn string</code>	database connection string e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
<code>-L, --log-level string</code>	logging level: debug, info, warn, error (default "error")
<code>-o, --output string</code>	output format: json, yaml

第13章 show-subscriptions

BDR サブスクリプション（受信レプリケーション）の詳細を表示します。

13.1 概要

オリジン/ターゲットノード、最後にリプレイされたトランザクションのトランザクションタイムスタンプまでのラグなど、BDR サブスクリプション（受信レプリケーション）の詳細を表示します。

```
pgd show-subscriptions [flags]
```

13.2 例

Example 1 (3 node cluster, bdr-a1 and bdr-c1 are up, bdr-b1 is down)

```
$ pgd show-subscriptions
```

Origin Node	Target Node	Last Transaction Replayed At	Lag Duration (seconds)
bdr-a1	bdr-c1	2022-04-23 13:13:40.854433+00	0.514275
bdr-b1	bdr-a1		
bdr-b1	bdr-c1		
bdr-c1	bdr-a1	2022-04-23 13:13:40.852233+00	0.335464

Example 2 (3 node cluster, bdr-b1 was down and it has just been restarted)

```
$ pgd show-subscriptions
```

Origin Node	Target Node	Last Transaction Replayed At	Lag Duration (seconds)
bdr-a1	bdr-b1	2022-04-23 13:14:45.669254+00	0.001686
bdr-a1	bdr-c1	2022-04-23 13:14:46.157913+00	-0.002009
bdr-b1	bdr-a1		
bdr-b1	bdr-c1		
bdr-c1	bdr-a1	2022-04-23 13:14:45.698472+00	0.259521
bdr-c1	bdr-b1	2022-04-23 13:14:45.667979+00	0.002961

(次のページに続く)

Example 3 (3 node cluster, all nodes are up and in streaming state)

```
$ pgd show-subscriptions
```

Origin Node	Target Node	Last Transaction Replayed At	Lag Duration (seconds)
bdr-a1	bdr-b1	2022-04-23 13:15:39.732375+00	0.034462
bdr-a1	bdr-c1	2022-04-23 13:15:40.179618+00	0.002647
bdr-b1	bdr-a1	2022-04-23 13:15:39.719994+00	0.305814
bdr-b1	bdr-c1	2022-04-23 13:15:40.180886+00	0.001379
bdr-c1	bdr-a1	2022-04-23 13:15:39.714397+00	0.311411
bdr-c1	bdr-b1	2022-04-23 13:15:39.714397+00	0.052440

Example 4 (cluster with witness, logical standby and subscriber-only nodes;
upstream for logical-standby-a1 is bdr-a1 and for logical-standby-b1 it is bdr-b1)

Note: Logical standby and subscriber-only nodes receive changes but do not
send changes made locally to other nodes

```
$ pgd show-subscriptions
```

Origin Node	Target Node	Last Transaction Replayed At	Lag Duration (seconds)
bdr-a1	bdr-b1	2022-04-23 13:40:49.106411+00	0.853665
bdr-a1	logical-standby-a1	2022-04-23 13:40:50.72036+00	0.138430
bdr-a1	logical-standby-b1		
bdr-a1	subscriber-only-c1	2022-04-23 13:40:50.72036+00	0.016226
bdr-a1	witness-c1	2022-04-23 13:40:50.470142+00	0.001514
bdr-b1	bdr-a1	2022-04-23 13:40:49.10174+00	1.095422
bdr-b1	logical-standby-a1		
bdr-b1	logical-standby-b1	2022-04-23 13:40:50.713666+00	0.271213
bdr-b1	subscriber-only-c1	2022-04-23 13:40:50.713666+00	0.022920
bdr-b1	witness-c1	2022-04-23 13:40:50.471789+00	-0.000133
witness-c1	bdr-a1	2022-04-23 13:40:49.107706+00	1.089456
witness-c1	bdr-b1	2022-04-23 13:40:49.107706+00	0.852370
witness-c1	logical-standby-a1		
witness-c1	logical-standby-b1		
witness-c1	subscriber-only-c1	2022-04-23 13:40:50.719844+00	0.016742

13.3 オプション

<code>-h, --help</code> help for show-subscriptions

13.4 親コマンドから継承されたオプション

<code>-f, --config-file string</code>	config file; ignored if --dsn flag is present (default "/etc/edb/pgd-config.yml")
<code>--dsn string</code>	database connection string e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
<code>-L, --log-level string</code>	logging level: debug, info, warn, error (default "error")
<code>-o, --output string</code>	output format: json, yaml

第14章 show-version

各ノードにインストールされている BDR と Postgres のバージョンを表示します。

14.1 概要

クラスター内の各ノードにインストールされている BDR と Postgres のバージョンを表示します。

```
pgd show-version [flags]
```

14.2 例

Example 1 (3 node cluster, bdr-a1 and bdr-c1 are up, bdr-b1 is down)

```
$ pgd show-version
```

Node	BDR Version	Postgres Version
bdr-c1	4.1.0	14.2 (EDB Postgres Extended Server 14.2.0) (Debian 2:14.2.0edbpge- →1.buster+1)
bdr-a1	4.1.0	14.2 (EDB Postgres Extended Server 14.2.0) (Debian 2:14.2.0edbpge- →1.buster+1)
bdr-b1		

Example 2 (3 node cluster, all nodes are up)

```
$ pgd show-version
```

Node	BDR Version	Postgres Version
bdr-c1	4.1.0	14.2 (EDB Postgres Extended Server 14.2.0) (Debian 2:14.2.0edbpge- →1.buster+1)
bdr-a1	4.1.0	14.2 (EDB Postgres Extended Server 14.2.0) (Debian 2:14.2.0edbpge- →1.buster+1)
bdr-b1	4.1.0	14.2 (EDB Postgres Extended Server 14.2.0) (Debian 2:14.2.0edbpge- →1.buster+1)

14.3 オプション

<code>-h, --help</code> help for show-version

14.4 親コマンドから継承されたオプション

<code>-f, --config-file string</code>	config file; ignored if --dsn flag is present (default "/etc/edb/pgd-config.yml")
<code>--dsn string</code>	database connection string e.g. "host=bdr-a1 port=5432 dbname=bdrdb user=postgres "
<code>-L, --log-level string</code>	logging level: debug, info, warn, error (default "error")
<code>-o, --output string</code>	output format: json, yaml