

# PostgreSQL のバックアップとリカバリー

## 論理バックアップ

### 技術を知る

<input type="checkbox"/> 導入／環境設定	<input type="checkbox"/> 移行	<input type="checkbox"/> 性能	<input type="checkbox"/> チューニング	<input checked="" type="checkbox"/> バックアップ／リカバリー
<input type="checkbox"/> 冗長化／負荷分散	<input type="checkbox"/> 監視	<input type="checkbox"/> データ連携	<input checked="" type="checkbox"/> 災害対策	<input type="checkbox"/> 豆知識

ここでは、PostgreSQL データベースの論理バックアップとリカバリーについて解説します。

#### 1. 論理バックアップには何があるのか？

論理バックアップは、PostgreSQL を起動した状態でデータベースに格納されているデータをファイルに書き出すバックアップ方法です。論理バックアップは、バックアップを実行した時点へのリカバリーのみが可能です。PostgreSQL データベースは停止する必要ありません。論理バックアップを実施する方法として以下の 3 つがあります。

- **pg\_dumpall コマンド**

データベースクラスタ全体の内容を SQL の形でバックアップします。取得したバックアップデータは、psql コマンドを使ってバックアップを取得した時点へリカバリーします。

- **pg\_dump コマンド**

指定したデータベースの内容を SQL の形でバックアップします。スクリプト形式のバックアップデータは、psql コマンドを使ってバックアップを取得した時点へリカバリーします。アーカイブファイル形式のバックアップデータは、pg\_restore コマンドを使ってバックアップを取得した時点へリカバリーします。

- **SQL コマンドの COPY**

指定したテーブルのデータを独自のテキスト形式、バイナリー形式、または CSV 形式でバックアップします。取得したバックアップデータは、COPY を使ってバックアップを取得した時点へリカバリーします。

論理バックアップの種類について、以下の表にまとめます。なお、表中記号の意味は以下のとおりです。

レ：機能有り、N/A：機能無し

概要	バックアップ方法	バックアップ対象	リカバリー方法	リカバリーの復旧時点 (リストアポイント)		
				バックアップ	最新	任意
データベース起動状態で、SQL の形で取り出して保存。定義・データ単位のバックアップも可能。	pg_dumpall コマンド	データベースクラスタ	psql コマンド	レ	N/A	N/A
	pg_dump コマンド	データベース	psql コマンド (スクリプトファイル形式)	レ	N/A	N/A
			pg_restore コマンド (アーカイブファイル形式)			
	SQL コマンドの COPY	テーブル	SQL コマンドの COPY	レ	N/A	N/A

バックアップの対象をデータベースクラスタ全体とするのか、データベース単位またはテーブル単位とするのかにより、バックアップ方法を選択できます。さらには、テーブルの定義だけ、テーブル内のデータだけ、テーブル定義とデータの両方、などバックアップの内容をオプションで指定することもできます。

このバックアップ方法は、PostgreSQL データベースを停止する必要が無いため、いつでもバックアップを取得できます。

物理バックアップに比べて、バックアップのデータサイズは小さいですが、処理時間が長くなります。また、バックアップを取得したデータを使うだけなので、バックアップ時点への復旧しか選択できません。

## 2. 論理バックアップを使ってみる

PostgreSQL データベースで構築した人事情報システムを想定しました。

動作 OS	Red Hat® Enterprise Linux® 7 (for Intel64)
PostgreSQL バージョンレベル	PostgreSQL 10.3
インスタンス名	inst1
データベース名	mydb
テーブル名	人事テーブル 1 (jtbl1)

この条件に基づいて、PostgreSQL データベースにおけるバックアップとリカバリーの操作例について実機を用いた動画で説明します。

## 2.1 データベースクラスタ全体をバックアップ・リカバリーする

`pg_dumpall` コマンドを使ってデータベースクラスタ全体をバックアップし、`psql` コマンドを使ってリカバリーする例を説明します。

### 想定する業務要件

サーバーA の現インスタンス (inst1) を、同一サーバー内の新インスタンス (inst2) に移行することになりました。  
以下の手順で作業を進めます。

1. 現インスタンス (inst1) において、`pg_dumpall` コマンドを使ってデータベースクラスタをバックアップします。
2. 新インスタンス (inst2) を作成し、`psql` コマンドを使って現インスタンス (inst1) のバックアップデータを移行します。  
この際、同一サーバーに複数のインスタンスが存在することになるため、新インスタンス (inst2) のポート番号は現インスタンス (inst1) と違う設定にする必要があります。

以下に、業務要件のイメージを示します。

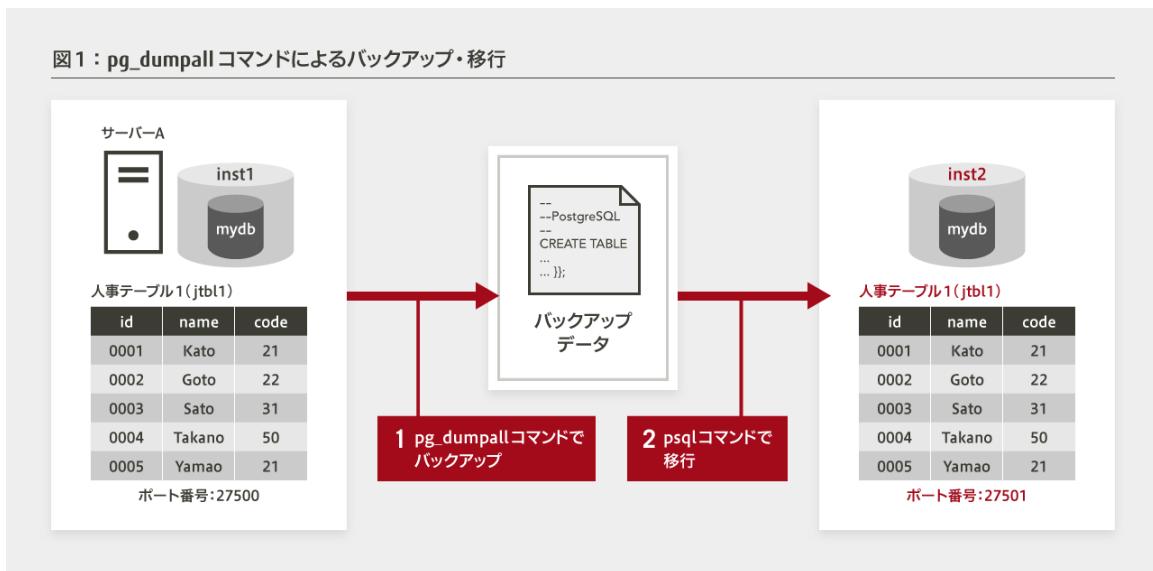


図1：`pg_dumpall` コマンドによるバックアップ・移行

### 動画による操作解説

「想定する業務要件」を以下の手順で実行する様子を動画にて解説します。また、動画中の白文字はコマンドなどの実行操作およびそれに対するサーバーからの応答メッセージなど、黄色文字は解説です。

動画：[https://youtu.be/anPE4\\_Texmk](https://youtu.be/anPE4_Texmk)

1. 現インスタンス (inst1) のデータベース一覧を表示し、テーブルの中身を表示します。
2. `pg_dumpall` コマンドを使って現インスタンス (inst1) のデータベースクラスタをバックアップします。
3. 生成したバックアップファイルの中身を確認します。
4. 新インスタンス (inst2) を作成します。
  - a. データ格納先用のディレクトリを作成

- b. バックアップデータ格納先用のディレクトリを作成
- c. トランザクションログ格納先用のディレクトリを作成
- d. initdb コマンドで新インスタンス (inst2) のデータベースクラスタを作成

5. 設定ファイルのポート番号を編集します。
6. 新インスタンス (inst2) を起動します。
7. psql コマンドを使って新インスタンス (inst2) にバックアップファイルを移行します。
8. 新インスタンス (inst2) のデータベースの内容が正しいかを確認します。

## 2.2 特定のテーブルをバックアップ・リカバリーする

pg\_dump コマンドを使って特定のテーブルをバックアップし、psql コマンドを使ってリカバリーする例を説明します。

### 想定する業務要件

組織変更により、人事テーブル1 (jtbl1) の部署コード (code) の値を変更することになりました。以下の手順で作業を進めます。

1. 現状の人事テーブル1 (jtbl1) に対し、pg\_dump コマンドを使ってバックアップします。
2. 人事テーブル1 (jtbl1) の code の値を変更しますが、更新実行ファイルを間違え、誤った値に更新してしまいました。
3. psql コマンドを使って、バックアップ時点の値に戻します。
4. 人事テーブル1 (jtbl1) の code を正しく変更します。

以下に、業務要件のイメージを示します。

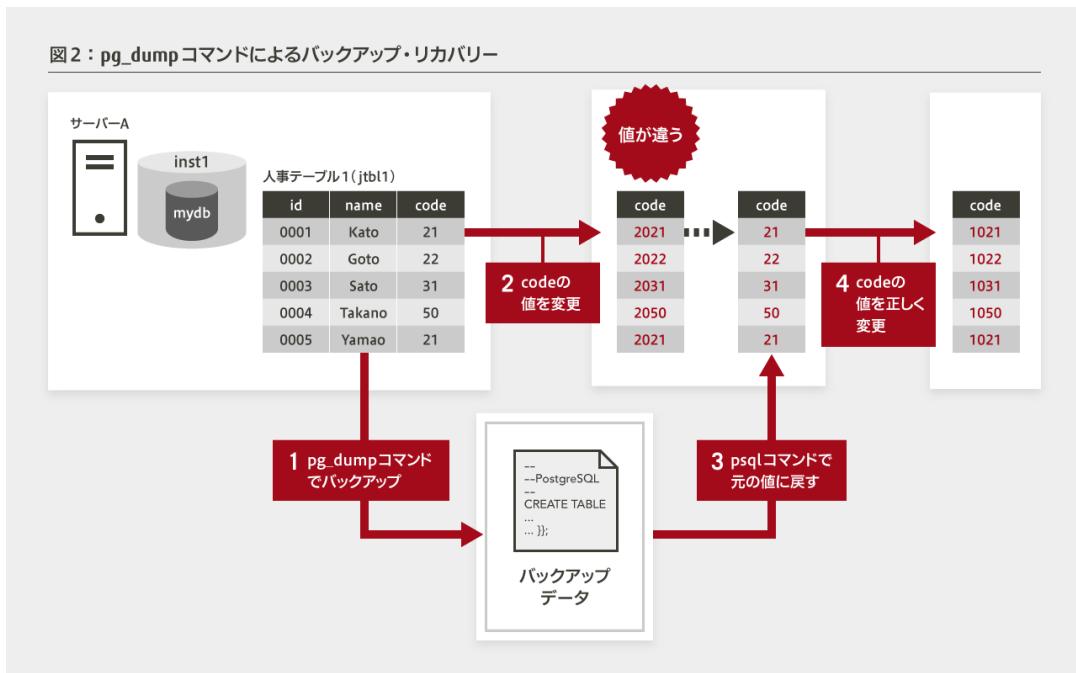


図2 : pg\_dump コマンドによるバックアップ・リカバリー

## 動画による操作解説

「想定する業務要件」を以下の手順で実行する様子を動画にて解説します。また、動画中の白文字はコマンドなどの実行操作およびそれに対するサーバーからの応答メッセージなど、黄色文字は解説です。

動画：<https://youtu.be/K4mTl7NpRHI>

1. 人事テーブル1 (jtbl1) を表示します。
2. pg\_dump コマンドを使って、人事テーブル1 (jtbl1) をバックアップします。
3. バックアップファイルの中身を確認します。
4. 人事テーブル1 (jtbl1) の code の値を変更したところ数値に誤りが発覚しました。
5. psql コマンドを使って、バックアップ取得時点に戻します。
6. 人事テーブル1 (jtbl1) のデータがバックアップ時点に戻ったかを確認します。
7. 改めて、人事テーブル1 (jtbl1) の code の値を変更します。

## 注意事項

pg\_dump コマンド実行時に、-c オプションまたは--clean オプションを指定すると、データベースオブジェクトを作成する SQL コマンドの前に、データベースオブジェクトを削除する SQL コマンドも出力されます。指定要否を十分に検討してください。詳細は「事前にデータベースオブジェクトを削除する手間を省いて、リストアを実行したい」をご覧ください。

## 2.3 COPY を使ってバックアップ・リカバリーする

SQL コマンドの COPY を使用して、バックアップデータを取得します。COPY を使用したバックアップでは、テーブルに格納されているデータをファイルに出力します。

### 想定する業務要件

業務拡張により、人事テーブル1 (jtbl1) と同じ構成の人事テーブル2 (jtbl2) を作ることになりました。以下の手順で作業を進めます。

1. 現状の人事テーブル1 (jtbl1) に対し、SQL コマンドの COPY を使ってバックアップします。
2. 人事テーブル2 (jtbl2) を作成し、SQL コマンドの COPY を使ってバックアップデータを格納します。

以下に、業務要件のイメージを示します。

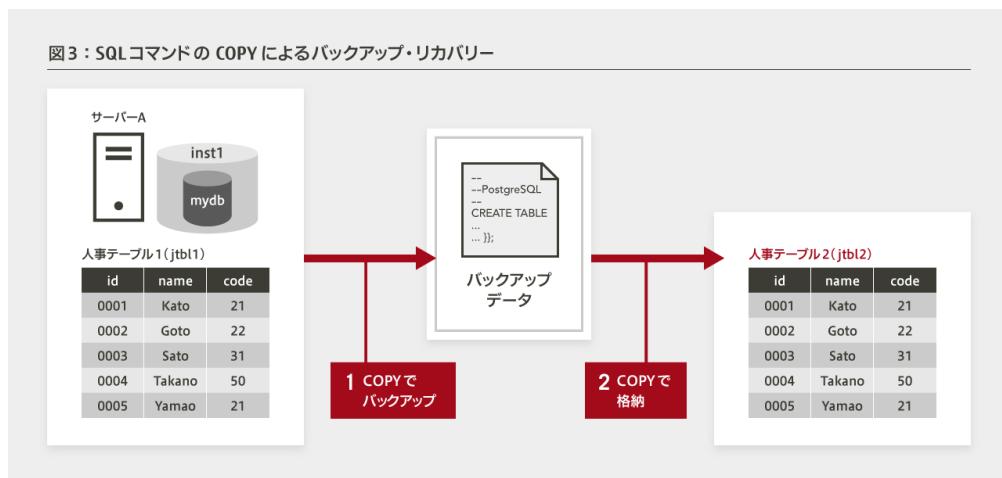


図3：SQLコマンドのCOPYによるバックアップ・リカバリー

## 動画による操作解説

「想定する業務要件」を以下の手順で実行する様子を動画にて解説します。また、動画中の白文字はコマンドなどの実行操作およびそれに対するサーバーからの応答メッセージなど、黄色文字は解説です。

動画：<https://youtu.be/imr18hb8oPY>

1. 人事テーブル1 (jtbl1) を表示します。
2. COPY を使って、人事テーブル1 (jtbl1) のデータをバックアップします。
3. バックアップファイルの中身を確認します。
4. 人事テーブル2 (jtbl2) を作成します。
5. COPY を使って、人事テーブル1 (jtbl1) のバックアップデータを人事テーブル2 (jtbl2) へ格納します。
6. 人事テーブル2 (jtbl2) のデータを確認します。

## よくあるトラブル事例と対処

トラブル	COPY で、テーブルからファイルへのデータコピーを実行すると、期待した文字コードでデータがコピーされない。
原因	コピーデータは ENCODING オプションまたは現在のクライアント符号化方式で符号化される。COPY の ENCODING パラメーターを省略したため、クライアントの符号化方式が採用され、期待した文字コードでデータがコピーされなかった。
対処	COPY の ENCODING パラメーターに、符号化方式を指定してください。

お使いのシステムにおける運用管理要件に応じた論理バックアップを実行することで、“あんしん”できる業務データの保守運用を実現します。

2018 年 11 月 30 日