



Postgres Workload Report

リリース 1.0.0

EnterpriseDB Corporation.

2024 年 02 月 20 日

目次

第 1 章	Release notes	3
1.1	Postgres Workload Report 1.0.0 release notes	3
第 2 章	Installing Postgres Workload Report	5
第 3 章	Configuring Postgres Workload Report	7
3.1	pwr 構成ファイル	7
3.1.1	input_dir	7
3.1.2	Output_dir	7
3.1.3	Report_name	7
3.1.4	log_file	8
3.1.5	log_level	8
3.1.6	log_format	8
3.1.7	attempts_dir	8
第 4 章	Using Postgres Workload Report	9
4.1	前提条件	9
4.2	レポートのソース情報	9
4.3	execute オプションの例	9
4.4	report オプションの例	10

Postgres Workload ReportPWR は、HTML、Markdown、DOCX、PDF モードで PostgreSQL ワークロードレポートを構築するために使用される Python ベースのツールで、Oracle の自動ワークロードリポジトリ AWR レポートツールが提供するレポートを模倣しています。

Postgres 接続を使用すると、レポートが必要なサーバーにアクセスできる任意のマシンから Postgres ワークロードレポートを実行できます。Postgres ワークロードレポートはデータの収集に Lasso を使用するため、Lasso が前提条件です。edb-pwr パッケージは、インストールされる edb-lasso に明示的な依存関係があります。

Postgres サーバーでは、edb_wait_states バージョン 1.2 以降をインストールおよびロードしておく必要があります。拡張機能も、できればメインデータベースで作成する必要があるため、Postgres ワークロードレポートで edb_wait_states ファンクションを使用して、保存されたスナップショットデータを収集できます。

PWR の詳細については、以下を参照してください。

- *[Installing Postgres Workload Report](#)*
- *[Configuring Postgres Workload Report](#)*
- *[Using Postgres Workload Report](#)*

第1章 Release notes

Postgres Workload ReportPWR ドキュメントでは、マイナーリリースとパッチを含む PWR 1 の最新バージョンについて説明しています。このセクションのリリースノートは、各リリースの新機能に関する情報を提供します。マイナーまたはパッチリリースで導入された新機能の場合、コンテンツ内にどのリリースで機能が導入されたかを示すインジケーターもあります。

Version	Release Date
<i>Postgres Workload Report 1.0.0 release notes</i>	15 Feb 2024

1.1 Postgres Workload Report 1.0.0 release notes

リリース日 2024 年 2 月 15 日

Postgres Workload Report 1.0.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Feature	<i>pwr execute</i> CLI コマンドを追加します。このコマンドは、ワークロードレポートを生成するために使用されます。 <code>`pwr report`</code> コマンドでは、ユーザーが Lasso を実行し、そのデータを入力として提供する必要がありますが、 <code>`pwr execute`</code> はエンドツーエンドの実行、つまり Lasso と <code>`pwr report`</code> を暗黙的に実行します。 <code>`pwr execute`</code> コマンドには、 <code>edb-lasso >=4.13.0</code> 、 <code>edb_wait_states >=1.2.0</code> が必要です。Postgres サーバー要件およびワークロードレポートの出力形式の詳細については、PWR ドキュメントを参照してください。
Feature	<i>pwr report</i> CLI コマンドの追加 <code>`pwr report`</code> コマンドは、ワークロードレポートを生成するために使用されます。PWR ワークロードレポートは、HTML、Markdown、PDF、DOCX の中の 1 つ以上の形式で書くことができます。 <code>`pwr report`</code> コマンドの詳細については、PWR ドキュメントを参照してください。

第2章 Installing Postgres Workload Report

Postgres ワークロードレポートは、サポートされているすべてのオペレーティングシステムの Python ソース配布パッケージとして提供されます。これは、**enterprise** および **standard** リポジトリにあります。

次のコマンドは、Postgres Workload Report をインストールし、必要に応じて、リポジトリから **edb-lasso** パッケージを取得してインストールします。

Postgres Workload Report をインストールするには、次のコマンドを入力します。

```
sudo <package-manager> -y install edb-pwr
```

<package-manager> は、オペレーティングシステムで使用するパッケージマネージャーです。

Package manager	Operating system
apt	Debian / Ubuntu
dnf	RHEL 8/9 and derivatives
yum	RHEL 7 and derivatives, CentOS 7
zypper	SLES

第3章 Configuring Postgres Workload Report

3.1 pwr 構成ファイル

`pwr` の実行時に必要なコマンドライン引数の数を減らすには、構成ファイルを使用して、常に同じ値を持つオプションを指定できますが、その値はデフォルトとは異なります。

実行中に、`pwr` は、最初に見つかったものを使用して、`~/.pwr.conf` および `/etc/pwr.conf` の順序で既存の構成ファイルを検索します。ただし、`--config` オプションで構成ファイルを指定している場合、それはデフォルトの場所をオーバーライドします。

インストールパッケージは、`/etc/pwr.conf.template` に構成ファイルのテンプレートを作成します。このファイルを `pwr` が構成ファイルを検索する2つの場所のいずれかにコピーし、必要に応じてテンプレートのオプションを編集することをお勧めします。

以下のセクションでは、構成可能なオプションについて説明します。

3.1.1 input_dir

Lasso レポートの `edb_wait_states` コンテンツが存在する既存のディレクトリを識別します。このオプションは、主に `pwr report` 実行に使用されます [Using Postgres Workload Report](#) を参照してください。

3.1.2 Output_dir

レポートを出力する場所を指定します。実行中に、Postgres Workload Report は、ディレクトリが存在しない場合は作成しようとします。

3.1.3 Report_name

生成されたレポートファイルの名前を提供します。通常、レポートごとに名前が異なるため、このオプションはコマンドラインから指定します。

3.1.4 log_file

pwr が stdout および stderr ログを書き込むファイルへのフルパスを提供します。

3.1.5 log_level

Postgres ワークロードレポートを実行するときに使用するログレベルを指定します。有効な値は、より冗長なものからより低いものです。

DEBUG INFO 指定しない場合のデフォルト WARNING ERROR CRITICAL

ログレベルの詳細については、

[the Python logging](#) ドキュメントを参照してください。

3.1.6 log_format

ログファイルに書き込まれるログメッセージの形式を提供します。ログのフォーマットの詳細については、

[the Python logging](#) ドキュメントを参照してください。

3.1.7 attempts_dir

HTML 出力のフォーマットに使用される jinja テンプレートと PDF 出力に使用される CSS を見つけられるディレクトリを示します。通常、デフォルトで使用するディレクトリは `/usr/share/pwr/assets` で、これには、`edb-pwr` パッケージで提供されるアセットが含まれています。

第4章 Using Postgres Workload Report

4.1 前提条件

Postgres ワークロードレポートは、`edb_wait_states` 拡張機能バージョン 1.2 以降がロードされている Postgres サーバーのレポートのみを提供できます。さらに、PWR は、`edb_wait_states` 拡張機能がサーバーにロードされた時間間隔のクエリ待機レポートのみを提供できます。

詳細については、`edb_wait_states` を参照してください。

4.2 レポートのソース情報

`edb-pwr` および `edb-lasso` パッケージがマシンにインストールされ、`edb_wait_states` 拡張機能が十分な期間ロードされた状態でサーバーが実行された後、Postgres Workload Report を使用して、実行中に実行されていたクエリの待機状態のレポートを抽出できます。指定された時間間隔。

あるいは、Postgres ワークロードレポートは、`edb_wait_states` がロードされたサーバーでレポートが実行されたと仮定して、既存の Lasso レポートからレポートを生成できます。このため、Postgres ワークロードレポートには必須の第 1 引数があり、次の 2 つのいずれかです。

- `execute`
- `report`

`execute` 引数はエンドツーエンドの実行を実行し、適切なオプションで `lasso` を呼び出し、生成された tarball レポートを入力として使用して、HTML、Markdown、DOCX、または PDF レポートを生成します。

`report` 引数は、待機状態レポートを処理および生成するための入力として、`lasso` レポートコンテンツを保存するディレクトリを使用します。`report` 引数を使用するには、解凍および解凍された `lasso` レポートが既に存在する必要があります。

4.3 `execute` オプションの例

この例では、1 月 10 日 9:00 から 1 月 10 日 13:00 までの間隔での Postgres サーバー `myserver` のクエリ待機に関するレポートを生成します。その頃、インシデントが発生しました。根本的な原因を提供するために調査する必要があります。`edb_wait_states` がインストールされるサーバーのメインデータベースは `my-oltp` です。

このようなレポートを HTML 形式で取得するには、次のコマンドを使用します。

```
pwr execute --host-name myserver --sampling-start 2024-01-10 09:00+00 \  
--sampling-end 2024-01-10 13:00+00 --html \  
--report-name Jan10_incident my-oltp postgres
```

`pwr execute -h` を実行して、使用可能なオプションの完全な出力を取得できます。

4.4 report オプションの例

場合によっては、既に Lasso レポートがあり、PWR に Lasso レポートをソースとして使用し、それに基づいてレポートを構築したい場合があります。これらの場合、`pwr report` オプションは役立ちます。

この例では、既に説明したのと同じシナリオを使用できますが、上記で共有した時間境界を使用して既に実行されたなげな Lasso レポートを使用します。Lasso レポートの名前が `edb-lasso-Jan10-incident.tar.bz2` で、`pwr report` が実行されるマシンのホームディレクトリにあると仮定します。

次のコマンドを実行すると、`~/pwr_output/Jan10_incident.html` に保存された HTML レポートが生成されます。

```
cd ~/\  
mkdir -p pwr_tmp/  
tar jxf ../edb-lasso-Jan10-incident.tar.bz2 -C ~/pwr_tmp/ --strip-components=1  
pwr report \  
--input-dir ~/pwr_tmp/postgresql/dbs/my-oltp/edb_wait_states/ \  
--html --output-dir ~/pwr_output/ --report-name Jan10_incident
```

`pwr report -h` を実行して、使用可能なオプションの完全な出力を取得できます。