

Alibaba Cloud 分析型データ#PostgreSQL版

クイックスタート

Document Version20200213

目次

1 概要	1
2 インスタンスの作成	2
3 ホワइटリストの設定	4
4 ネットワークタイプの設定	6
5 アカウントの設定	8
6 AnalyticDB for PostgreSQL データベースへの接続	10

1 概要

AnalyticDB for PostgreSQL は、分散型クラウドデータベースです。複数のグループで構成され、MPP (超並列処理) データウェアハウスサービスを提供します。**AnalyticDB for PostgreSQL** はオープンソースデータベースプログラムの **Greenplum** をベースに開発されており、**Alibaba Cloud** による拡張機能で強化されています。

AnalyticDB for PostgreSQL は **Greenplum** 環境と互換性があり、OSS ストレージ、JSON データ型、**HyperLogLog** 近似解析などの機能をサポートしています。**AnalyticDB for PostgreSQL** の機能と制限については、「[機能と制限](#)」をご参照ください。

AnalyticDB for PostgreSQL を使用するには、次のタスクを完了する必要があります。

1. [インスタンスの作成](#)
2. [ホワイトリストの設定](#)、[アカウントの設定](#)、[ネットワークタイプの設定](#) などのインスタンスの設定
3. [データベースへの接続](#)
4. データのインポート データのインポートは、[OSS 外部テーブルを用いた並列インポート](#)、[MySQLからのインポート](#)、[PostgreSQLからのインポート](#)、[COPY コマンドを使用したインポート](#)のいずれかの方法で行えます。

2 インスタンスの作成

次のいずれかの方法で、AnalyticDB for PostgreSQL インスタンスを作成または購入できます。

- ・ AnalyticDB for PostgreSQL コンソールからインスタンスを作成します。
- ・ AnalyticDB for PostgreSQL 購入ページからインスタンスを購入します。

ここでは、AnalyticDB for PostgreSQL インスタンスをコンソールから作成する詳細な手順を説明します。

課金方法

AnalyticDB for PostgreSQL は、従量課金のみ対応しています。

前提条件

アカウントを登録し、サインアップしていること。

手順

1. [AnalyticDB for PostgreSQL コンソール](#)にログインします。
2. [インスタンスの作成] をクリックします。
3. [インスタンス設定] を選択します。選択肢は次のとおりです。
 - ・ リージョンとゾーン: 選択方法についての案内は、「[リージョンとゾーン](#)」をご参照ください。
 - ・ エンジン: データベースタイプです。[含まれるストレージ]のみサポートしています。
 - ・ インスタンスクラス: インスタンスタイプです。計算リソースの単位を表します。クラスによってストレージ容量と計算能力が異なります。詳細は、「[インスタンスタイプ](#)」をご参照ください。
 - ・ [インスタンスグループ]: 購入したインスタンスの数です。最小は2です。グループが多いと、線形パフォーマンスも高くなります。
4. 注文情報を確認して、[今すぐ購入]をクリックします。
5. [有効化] をクリックしてインスタンスを有効化します。
6. [AnalyticDB for PostgreSQL コンソール](#)から [インスタンスリスト] ページを開くと、新しく作成したインスタンスが表示されます。



注:

インスタンスの初期化には時間がかかることがあります。インスタンスのステータスが「実行中」になってからしか次の操作は行えません。

3 ホワイトリストの設定

インスタンスを起動する前に、ホワイトリストを設定する必要があります。データベースへのアクセスを許可する IP アドレス または IP セグメントを設定し、安全性と安定性を確保します。

背景

AnalyticDB for PostgreSQL データベースへのアクセスには、3 種類のシナリオがあります。

- ・ インターネットからのアクセス
- ・ イン트라ネットからのアクセス。 **AnalyticDB for PostgreSQL** と ECS インスタンスのネットワークタイプは同一である必要があります。
- ・ イン트라ネットとインターネットからの同時アクセス。 **AnalyticDB for PostgreSQL** と ECS インスタンス のネットワークタイプ が同一である必要があります。



注：

ネットワークタイプを設定するには、「[ネットワークタイプの設定](#)」をご参照ください。

手順

1. **AnalyticDB for PostgreSQL** コンソールにログインします。
2. 対象となるインスタンスのリージョンを選択します。
3. インスタンスの ID をクリックして、インスタンスの [基本 情報] ページを表示させます。
4. 左側のナビゲーションウィンドウで、[セキュリティ 管理] をクリックします。
5. [ホワイトリスト設定] ページで、デフォルトのホワイトリストグループの下にある [変更] をクリックし、[グループの変更] ページを開きます。



注：

また、デフォルトのホワイトリストグループの下にある [クリア] をクリックして設定されている IP アドレスをクリアしてから、[ホワイトリストグループを 追加] をクリックしてカスタムグループを作成することもできます。

6. ホワイトリストからデフォルトアドレス **"127.0.0.1"** を削除し、カスタムホワイトリストを入力します。設定項目の説明は、以下のとおりです。

- ・ **グループ名** : 2 文字以上 32 文字以内で、小文字アルファベット、数字、アンダースコア (_) が使用できます。グループ名は、小文字アルファベットで始まり、文字または数字で終わらなければなりません。デフォルトのグループ名は変更または削除できません。
- ・ **ホワイトリスト** : データベースへのアクセスを許可する **IP** アドレスまたはセグメントを入力します。**IP** アドレスやセグメントは、コンマ (,) で区切ります。
 - ホワイトリストには、**IP** アドレス ("**10.10.10.1**" など) またはセグメント ("**10.10.10.X**" という形式に当てはまるすべての **IP** アドレスがデータベースにアクセスできることを示す "**10.10.10.0/24**" など) を設定できます。
 - **"%"** または "**0.0.0.0/0**" は、全ての **IP** アドレスがデータベースにアクセスできることを示します。



注:

データベースのセキュリティを大いに低下させる可能性があるため、必要がなければ、この設定は使わないことを推奨します。

- インスタンスが作成されると、ローカルのループバック **IP** アドレス **"127.0.0.1"** がデフォルトのホワイトリストに追加され、外部 **IP** アドレスからインスタンスへのアクセスを全て阻止します。
- ・ **既存 ECS の IP アドレスを選択** : クリックすると、アカウントが所有する全ての **ECS** インスタンスが表示されます。**ECS** の **IP** アドレスを選択すると、その **ECS** インスタンスをホワイトリストに追加できます。

7. **[OK]** をクリックしてホワイトリストを追加します。

次に

ホワイトリストは、**AnalyticDB for PostgreSQL** の高度なアクセス保護を提供します。そのため、ホワイトリストを定期的にメンテナンスすることを推奨します。

以降のメンテナンスでは、グループ名の下 **[変更]** をクリックして既存のグループに変更を加えたり、**[削除]** をクリックして既存のカスタムグループを削除することができます。

4 ネットワークタイプの設定

Alibaba Cloud ApsaraDB は、クラシックネットワークと Virtual Private Cloud (VPC) の 2 種類のネットワークタイプをサポートしています。Cloud (VPC). デフォルトでは、AnalyticDB for PostgreSQL はクラシックネットワークを使用します。VPC を使用する場合は、AnalyticDB for PostgreSQL インスタンスと VPC が同じリージョンにあることを確認してください。region.

このドキュメントは主に、2 つのネットワークタイプの違いやその設定方法について説明します。

背景

クラシックネットワークと VPC には、次のような違いがあります。

- ・ **クラシックネットワーク**: クラシックネットワーク内のクラウドサービスは隔離されておらず、不正アクセスはクラウドサービスのホワイトリストポリシーによってのみブロックされます。
- ・ **VPC (Virtual Private Cloud)**: VPC を使用すると、隔離されたネットワーク環境を Alibaba Cloud 上に構築できます。VPC のルートテーブル、IP アドレス範囲、ゲートウェイはカスタマイズ可能です。専用回線や VPN を使ってお客様の IDC (Integrated Data Center) と Alibaba Cloud VPC 内のクラウドリソースを仮想 IDC に統合し、シームレスにアプリケーションをクラウドに移行することもできます。

手順

1. 対象となる HybridDB for PostgreSQL インスタンスと同じリージョンに VPC を作成します。
2. 『[HybridDB for PostgreSQL コンソール](#)』にログインします。
3. 対象インスタンスのリージョンを選択します。
4. インスタンスの ID をクリックして、[基本情報] ページに移動します。
5. [データベース接続] をクリックします。
6. [VPC への切り替え] をクリックします。
7. VPC と仮想スイッチを選択して、[OK] をクリックします。



注:

ネットワークが VPC に切り替わると、元のイントラネットアドレスはクラシックネットワークのものから VPC アドレスに変わります。クラシックネットワーク上の ECS は、

AnalyticDB for PostgreSQL インスタンスにアクセスできなくなります。元のインターネットアドレスは変更されません。

5 アカウントの設定

ここでは、アカウントの作成方法、および **AnalyticDB for PostgreSQL** インスタンスのパスワードをリセットする方法について説明します。

アカウントの作成

前提条件

AnalyticDB for PostgreSQL インスタンスを使用する前に、データベース用のアカウントを作成する必要があります。



注：

- ・ 初期アカウントは、一度作成すると削除できません。
- ・ コンソールからは他のアカウントを作成することはできませんが、データベースにログイン後、**SQL** 文を実行することで作成できます。

手順

1. **AnalyticDB for PostgreSQL コンソール**にログインします。
2. 対象となるインスタンスのリージョンを選択します。
3. インスタンスの **ID** をクリックして、インスタンスの **[基本情報]** ページに移動します。
4. 左側のナビゲーションウィンドウで、**[アカウント管理]** をクリックします。
5. **[アカウントの作成]** をクリックします。
6. データベースアカウントとパスワードを入力して、**[OK]** をクリックします。
 - ・ **[データベースアカウント]**：2 文字以上 16 文字以内で、小文字アルファベット、数字、アンダースコア (_) が使用できます。小文字アルファベットで始まり、文字または数字で終わらなければなりません。例: " user4example "
 - ・ **[パスワード]**：8 文字以上 32 文字以内です。次のうち、少なくとも 3 種類の文字を使用する必要があります。大文字アルファベット、小文字アルファベット、数字、記号
 - ・ **[パスワードの確認]**：パスワードをもう一度入力します。

パスワードのリセット

AnalyticDB for PostgreSQL を使用するにあたって、データベースアカウントのパスワードを忘れてしまった場合は、**AnalyticDB for PostgreSQL コンソール** からパスワードをリセットできます。



注:

データセキュリティを考慮し、定期的にパスワードを変更することを推奨します。

手順

1. [AnalyticDB for PostgreSQL コンソール](#)にログインします。
2. 対象となるインスタンスのリージョンを選択します。
3. 対象インスタンスの[操作]列の下の[管理]をクリックして、インスタンスの[基本情報]ページを開きます。
4. 左側のナビゲーションウィンドウで[アカウント管理]をクリックします。
5. 管理したいアカウントの[パスワードのリセット]をクリックします。
6. 新しいパスワードを入力して確認し、[OK]をクリックします。



注:

パスワードは8文字以上32文字以内で、大文字アルファベット、小文字アルファベット、数字、記号のうち、少なくとも3種類の文字を使用する必要があります。以前と同じパスワードは使用できません。

6 AnalyticDB for PostgreSQL データベースへの接続

クラウドデータベース AnalyticDB for PostgreSQL は PostgreSQL 8.2 のメッセージプロトコルと完全な互換性があり、**libpq**、**JDBC**、**ODBC**、**psycopg2**、**pgadmin III** などの PostgreSQL 8.2 のメッセージプロトコルに対応しているツールに直接アクセスできます。

Greenplum は JDBC、ODBC、**libpq** を含むインストールパッケージを提供しており、それを利用すると簡単にインストールできます。詳細は、[Greenplum の公式文書](#)をご参照ください。

GUI ツール

AnalyticDB for PostgreSQL のユーザーは、[SQL Workbench](#)、[Navicat Premium](#)、[Navicat For PostgreSQL](#)、[pgAdmin III \(1.6.3\)](#) といった Greenplum 対応のグラフィカルなクライアントツールをそのまま利用できます。

以下では、**pgAdmin III** を例にグラフィカルクライアントツールの使用方法を説明します。

pgAdmin III

pgAdmin III は PostgreSQL の GUI クライアントで、**HybridDB for PostgreSQL** へ直接接続できます。**HybridDB for PostgreSQL**. 詳細については、『[pgAdmin 公式サイト](#)』をご参照ください。

pgAdmin III 1.6.3 は『[PostgreSQL 公式サイト](#)』からダウンロードできます。**pgAdmin III 1.6.3** は Windows、MacOS、Linux などさまざまプラットフォームをサポートしています。



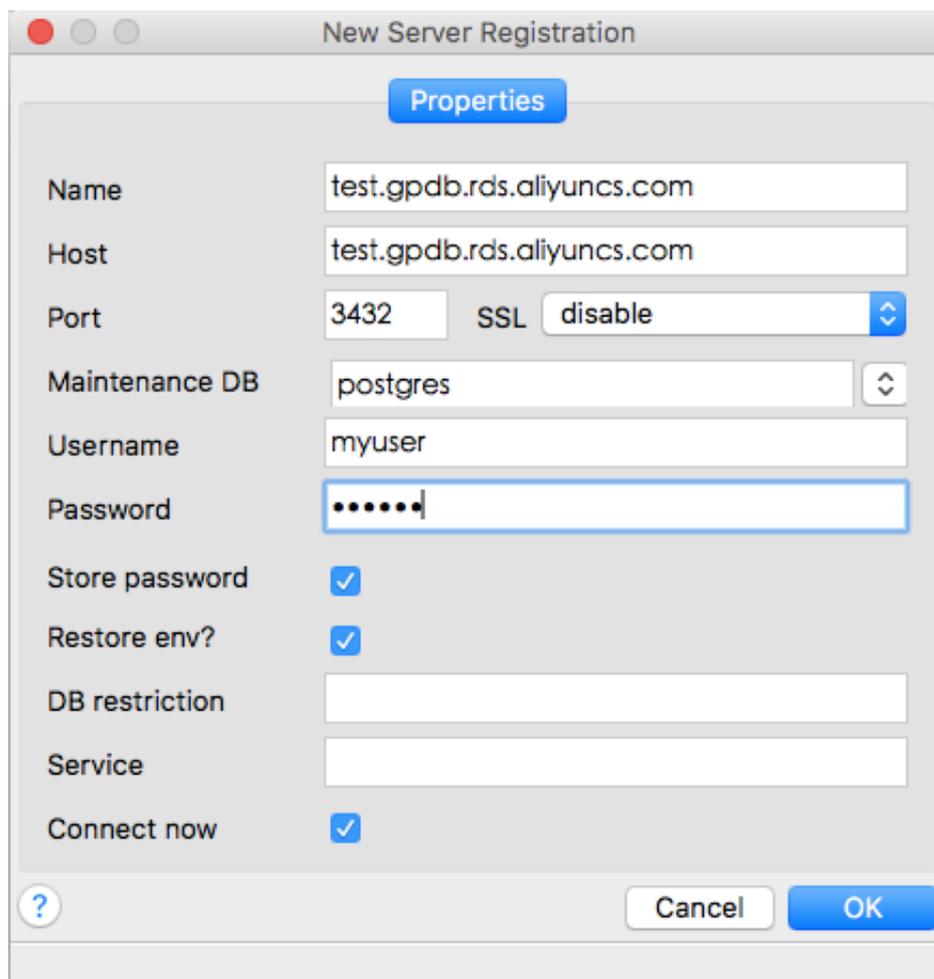
注:

AnalyticDB for PostgreSQL は PostgreSQL 8.2 と互換性があるため、**pgAdmin** もそれに対応するバージョンのものを使用する必要があります。**pgAdmin III 1.6.3** またはそれ以前のバージョンのものをご使用ください。

手順

1. **pgAdmin III 1.6.3** またはそれ以前のバージョンをダウンロードしてインストールします。
2. [ファイル]、[サーバーの追加] を選択し、**File > Add Server**[新しいサーバーの登録] ページを開きます。

3. 次の図のとおりに [プロパティ] を入力します。



4. [OK] をクリックして AnalyticDB for PostgreSQL データベースに接続します。

コマンドラインツール

下記のようなコマンドラインツールを使用して AnalyticDB for PostgreSQL インスタンスのデータベースに接続することもできます。

psql

psql は、Red Hat Enterprise Linux (RHEL) 6、RHEL 7、CentOS 6、CentOS 7 の AnalyticDB for PostgreSQL 向けの一般的なコマンドラインクライアントツールです。

Alibaba Cloud は、解凍後すぐに使えるソフトウェアパッケージの圧縮ファイルを提供しています。

- ・ RHEL 6 または CentOS 6 プラットフォーム用は、[ダウンロード](#) をクリックして下さい。
- ・ RHEL 7 または CentOS 7 プラットフォーム用は、[ダウンロード](#) をクリックして下さい。

他の Linux プラットフォームについては、ソースコードをダウンロードし、コンパイルとインストールを行って下さい。コンパイル方法は次のとおりです。

1. ソースコードを入手するには、次の 2 通りの方法があります。

- **git** ディレクトリを入手する (**git** がインストールされていることをご確認ください)。**git tool**).

```
git clone https://github.com/greenplum-db/gpdb.git
cd gpdb
git checkout 5d870156
```

- コードを直接ダウンロードする

```
wget https://github.com/greenplum-db/gpdb/archive/5d87015609
abd330c68a5402c1267fc86cbc9e1f.zip
unzip 5d87015609abd330c68a5402c1267fc86cbc9e1f.zip
cd gpdb-5d87015609abd330c68a5402c1267fc86cbc9e1f
```

2. コードのコンパイルやソフトウェアのインストールには、**GCC** などのコンパイラが必要です **software**.

```
./configure
make -j32
make install
```

インストール完了後、**psql** のパスは次のようになります。

```
psql: `/usr/local/pgsql/bin/psql`
```

上記ディレクトリに移動して **psql** を実行すると、**AnalyticDB for PostgreSQL** インスタンスのデータベースに接続できます。手順は次のとおりです。

1. 以下のいずれかの方法でデータベースに接続します。

- ・ 接続文字列

```
psql "host=yourgpdbaddress.gpdb.rds.aliyuncs.com port=3432 dbname=postgres user=gpdbaccount password=gpdbpassword"
```

- ・ パラメータの指定

```
psql -h yourgpdbaddress.gpdb.rds.aliyuncs.com -p 3568 -d postgres -U gpdbaccount
```

パラメータの説明

- **-h:** ホストアドレスを指定します。
- **-p:** ポート番号を指定します。
- **-d:** データベースを指定します (デフォルトのデータベースは **postgres** です)。
- **-U:** 接続ユーザーを指定します。

その他のパラメータについては、`psql -- help` を実行すると参照できます。また、**psql** プロンプトで `\?` を実行すると、サポートされている **psql** のコマンドを表示できます。

2. パスワードを入力して **psql** シェルインターフェイスに入ります。 **psql** シェル は次のように表示されます。

```
postgres=>
```

リファレンス

- ・ **Greenplum psql** の詳細な使用方法については、[psql](#) をご参照ください。
- ・ **PostgreSQL** の **psql** コマンドも使用できますが、細かい使い方が異なることにご注意ください。詳細については、[『PostgreSQL 8.3.23 Documentation — psql』](#) をご参照ください。

JDBC

JDBC を使用して **AnalyticDB for PostgreSQL** インスタンスのデータベースに接続することができます。ツールの入手方法は 2 通りあります。

- ・ [『PostgreSQL JDBC Driver』](#) をクリックして **PostgreSQL** 公式ウェブサイトから **JDBC** をダウンロードします。使用前に、ダウンロードした **JDBC** をクラス変数 `CLASSPATH` に追加します。
- ・ **Greenplum** 公式サイトのツールパッケージを使用します。詳細については、[『Greenplum Database 4.3 Connectivity Tools for UNIX』](#) をご参照ください。

以下にコード例を参考として示します。適宜変更の上、ご使用ください。

コード例

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class gp_conn {
    public static void main(String[] args) {
        try {
            Class.forName("org.postgresql.Driver");
            Connection db = DriverManager.getConnection("jdbc:
postgresql://mygpdbpub.gpdb.rds.aliyuncs.com:3432/postgres","mygpdb",
mygpdb");
            Statement st = db.createStatement();
            ResultSet rs = st.executeQuery("select * from gp_segment
_configuration");
            while (rs.next()) {
                System.out.print(rs.getString(1));
                System.out.print(" | ");
                System.out.print(rs.getString(2));
                System.out.print(" | ");
                System.out.print(rs.getString(3));
                System.out.print(" | ");
                System.out.print(rs.getString(4));
                System.out.print(" | ");
                System.out.print(rs.getString(5));
                System.out.print(" | ");
                System.out.print(rs.getString(6));
                System.out.print(" | ");
                System.out.print(rs.getString(7));
                System.out.print(" | ");
                System.out.print(rs.getString(8));
                System.out.print(" | ");
                System.out.print(rs.getString(9));
                System.out.print(" | ");
                System.out.print(rs.getString(10));
                System.out.print(" | ");
                System.out.println(rs.getString(11));
            }
            rs.close();
            st.close();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

詳細は、[『PostgreSQL JDBC Interface』](#)をご参照ください。

Python

Python を使って **AnalyticDB for PostgreSQL** インスタンスのデータベースに接続することもできます。**Python** は **psycopg2** ライブラリを使って **Greenplum** と **PostgreSQL** に接続します。ツールの使用手順は次のとおりです。

1. psycopg2 をインストールします。CentOS では、3 種類の方法があります。

- `yum -y install python-psycopg2` を実行します。
- `pip install psycopg2` を実行します。
- ソースコードからインストールします。

```
yum install -y postgresql-devel*
wget http://initd.org/psycopg/tarballs/PSYCOPG-2-6/psycopg2-2.6.
tar.gz
tar xf psycopg2-2.6.tar.gz
cd psycopg2-2.6
python setup.py build
sudo python setup.py install
```

2. インストール完了後、環境変数 PYTHONPATH を設定してから使用します。例：

```
import psycopg2
sql = 'select * from gp_segment_configuration;'
conn = psycopg2.connect(database='gpdb', user='mygpdb', password='
mygpdb', host='mygpdbpub.gpdb.rds.aliyuncs.com', port=3432)
conn.autocommit = True
cursor = conn.cursor()
cursor.execute(sql)
rows = cursor.fetchall()
for row in rows:
    print row
conn.commit()
conn.close()
```

次のような結果が返されます。

```
(1, -1, 'p', 'p', 's', 'u', 3022, '192.168.2.158', '192.168.2.158',
None, None)
(6, -1, 'm', 'm', 's', 'u', 3019, '192.168.2.47', '192.168.2.47',
None, None)
(2, 0, 'p', 'p', 's', 'u', 3025, '192.168.2.148', '192.168.2.148',
3525, None)
(4, 0, 'm', 'm', 's', 'u', 3024, '192.168.2.158', '192.168.2.158',
3524, None)
(3, 1, 'p', 'p', 's', 'u', 3023, '192.168.2.158', '192.168.2.158',
3523, None)
(5, 1, 'm', 'm', 's', 'u', 3026, '192.168.2.148', '192.168.2.148',
3526, None)
```

libpq

libpq は PostgreSQL データベースの C 言語インターフェイスです。C 言語のプログラムから **libpq** を使って PostgreSQL データベースにアクセスし、操作することができます。**Greenplum or** または **PostgreSQL** をインストールすると、*lib* ディレクトリに静的ライブラリと動的ライブラリが格納されています。

- 関連する事例については、『[libpq Example Programs](#)』をご参照ください。

- ・ **libpq** の詳細については、『[PostgreSQL 9.4 Documentation - Chapter 31. libpq - C Library](#)』をご参照ください。

ODBC

PostgreSQL ODBC は、**LGPL (GNU Lesser General Public License)** プロトコルに基づくオープンソース版です。『[PostgreSQL 公式ウェブサイト](#)』からダウンロードできます。

手順

1. ドライバをインストールします。

```
yum install -y unixODBC.x86_64
yum install -y postgresql-odbc.x86_64
```

2. ドライバの設定を確認します。

```
cat /etc/odbcinst.ini
# Example driver definitions
# Driver from the postgresql-odbc package
# Setup from the unixODBC package
[PostgreSQL]
Description      = ODBC for PostgreSQL
Driver           = /usr/lib/psqlodbcw.so
Setup            = /usr/lib/libodbcpsqlS.so
Driver64         = /usr/lib64/psqlodbcw.so
Setup64          = /usr/lib64/libodbcpsqlS.so
FileUsage        = 1
# Driver from the mysql-connector-odbc package
# Setup from the unixODBC package
[MySQL]
Description      = ODBC for MySQL
Driver           = /usr/lib/libmyodbc5.so
Setup            = /usr/lib/libodbcmyS.so
Driver64         = /usr/lib64/libmyodbc5.so
Setup64          = /usr/lib64/libodbcmyS.so
FileUsage        = 1
```

3. 次のコードのように **DSN** を設定します。コード中の ******** の部分を、実際の接続情報に変更してください。

```
[mygpdb]
Description = Test to gp
Driver = PostgreSQL
Database = ****
Servername = ****.gpdb.rds.aliyuncs.com
Username = ****
Password = ****
Port = ****
ReadOnly = 0
```

4. 接続のテスト

```
echo "select count(*) from pg_class" | isql mygpdb
+-----+
| Connected! |
+-----+
```

```
sql-statement
help [tablename]
quit
-----+
SQL> select count(*) from pg_class
-----+
| count |
-----+
| 388   |
-----+
SQLRowCount returns 1
1 rows fetched
```

5. ODBC がインスタンスに接続されたら、アプリケーションを ODBC に接続します。詳細は、[『PostgreSQL ODBC driver』](#)と[『psqlODBC HOWTO - C#』](#)をご参照ください。

Windows およびその他のプラットフォーム

[『Pivotal Greenplum Client』](#) から、**Windows** やその他のプラットフォーム用のクライアントツールをダウンロードできます。

リファレンス

- [Pivotal Greenplum 公式文書](#)
- [PostgreSQL psql ODBC](#)
- [PostgreSQL ODBC のコンパイル](#)
- [Greenplum ODBC のダウンロード](#)
- [Greenplum JDBC のダウンロード](#)