Bablefish for Aurora Postgres ワークショップ

2022/11/03

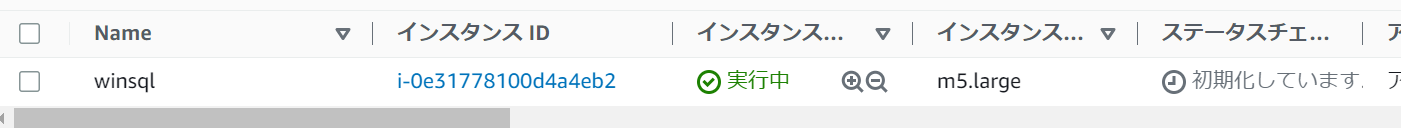
シニアエバンジェリスト 亀田

[SQL Studio用EC2の起動]

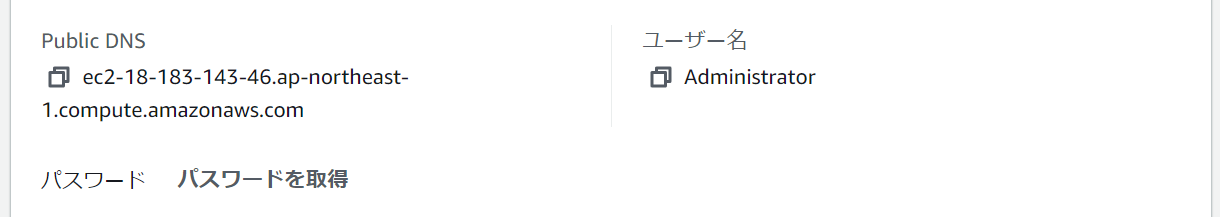
1. EC2のマネージメントコンソールに移動します
2. 左ペインから[インスタンス]をおします
3. 画面右上の[インスタンスを起動]をおします
4. 適当な名前を入力します
5. [sql]と入力して検索し以下のイメージを選択します（SQL Serverが入っていればなんでもよいです）



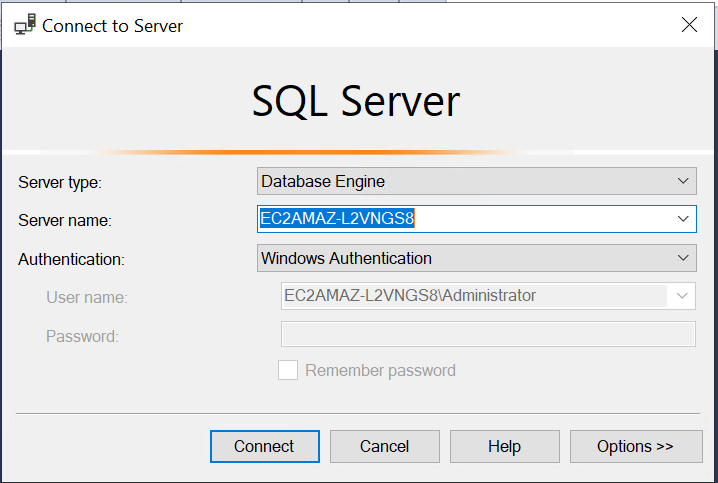
1. インスタンスタイプは,m5.largeを選びます
2. [新しいキーペアの作成]をおして適当な名前でキーペアを作成します。鍵がDLされてきますので、どこかに保存しておきます。
3. ネットワーク設定ではデフォルトVPCとなっていることを確認し、全てデフォルトのまま[インスタンスを起動]をおします
4. 起動が完了するまでしばらくまちます。（ステータスチェックがそのうち緑の表示になります）



1. 起動が完了したらインスタンスにチェックを入れて、[接続]をおします
2. [RDPクライアント]のタブを選びます
3. [リモートデスクトップファイルのダウンロード]をおします
4. [パスワードの取得]をクリックします

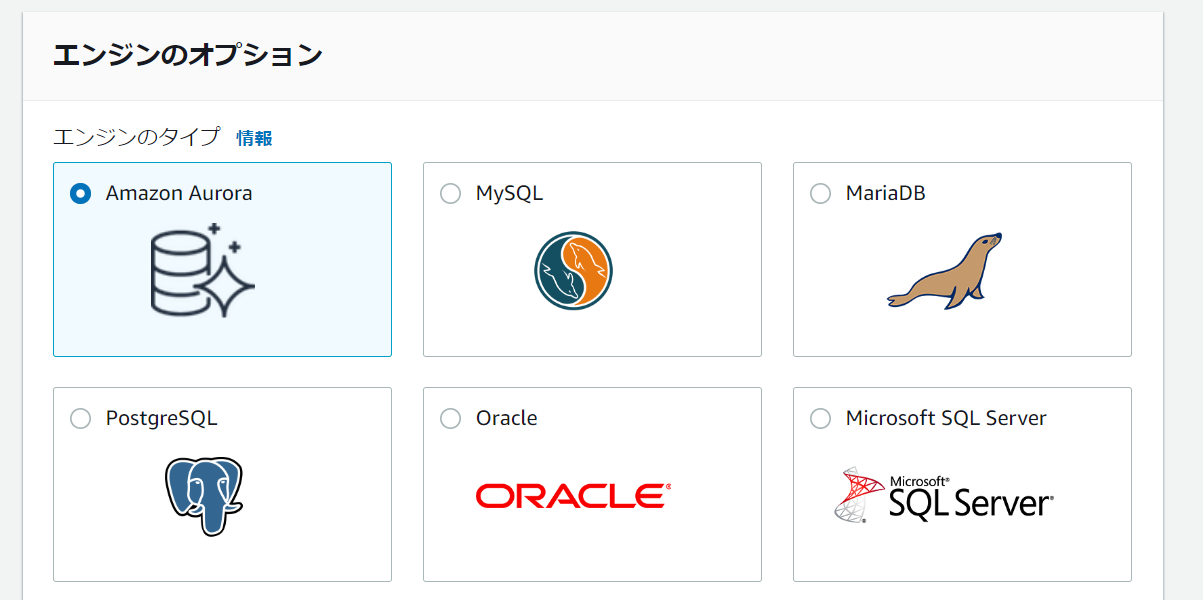


1. 先程ダウンロードした鍵をコピペし[パスワードを復号化]をおします
2. 表示されたパスワードを用いてリモートデスクトップ接続を行います
3. 画面左下の虫眼鏡アイコンからSQL Server Management Studioを探し起動します
4. 以下の画面が表示されれば起動完了です

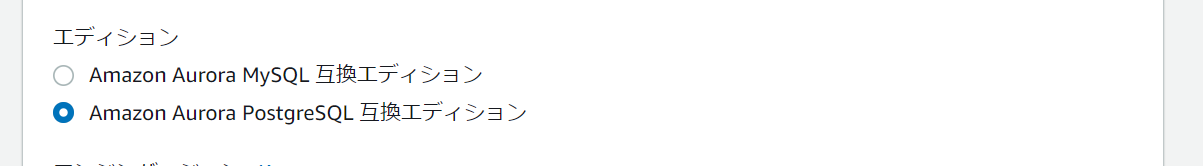


[Babelfish for Aurora Postgresの起動]

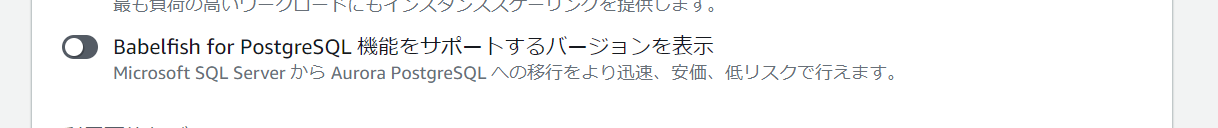
1. RDSのマネージメントコンソールに移動します
2. [データベースの作成]をおします
3. エンジンはAuroraを選びます



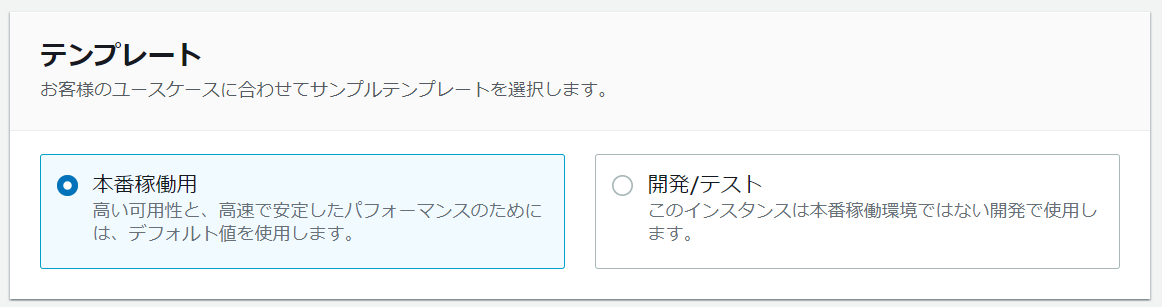
1. エディションはPostgresを選びます



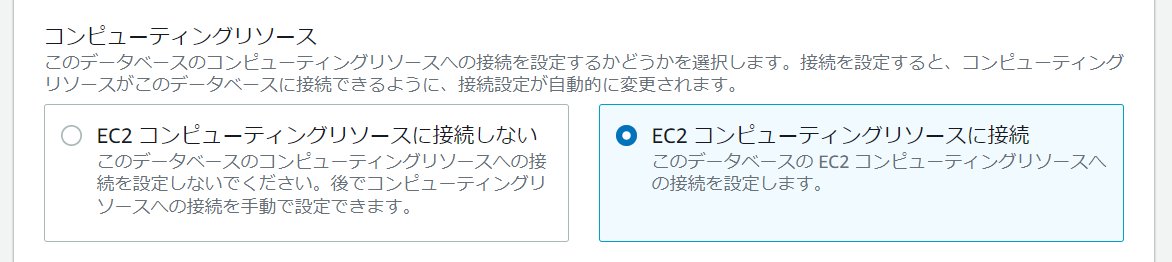
1. 以下のトグルをオンにします



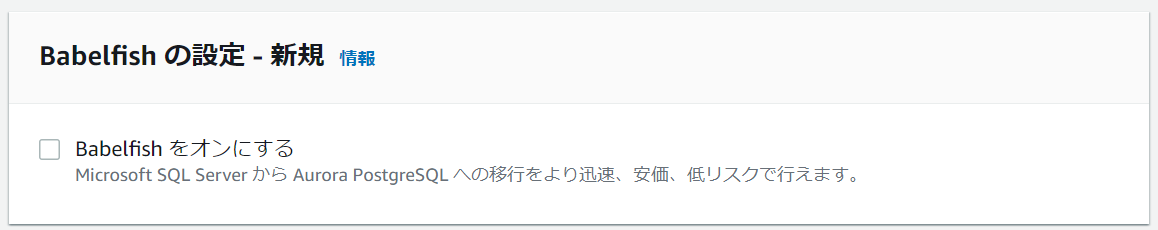
1. [開発/テスト]を選らびます



1. パスワードは適当なものを入力しますが、忘れないようにしてください
2. [接続]の個所で[EC2コンピューティングリソースに接続]を選びます



1. 先程作成したEC2を選びます
2. [Babelfishをオンにする]を選びます



1. [Performance Insights]を外します
2. [データベースの作成]をおします
3. 起動するまでしばらくまちます。ステータスが作成中から利用可能にしばらく待つとなります
4. 作成が完了したらクラスターをクリックして画面一番下を見るとEC2へのネットワーク設定が表示されています



このセキュリティグループはpostgresポート(5432)のみが設定されています。Babelfishは1433ポートを利用するため二つのセキュリティグループにそれぞれ追加する必要があります。

[rds-ec2] : インバウンドの1433

[ec2-rds] : アウトバウンドの1433

それぞれ以下を参考に設定します。ソースはあらかじめ入っているものと同じセキュリティグループを設定します

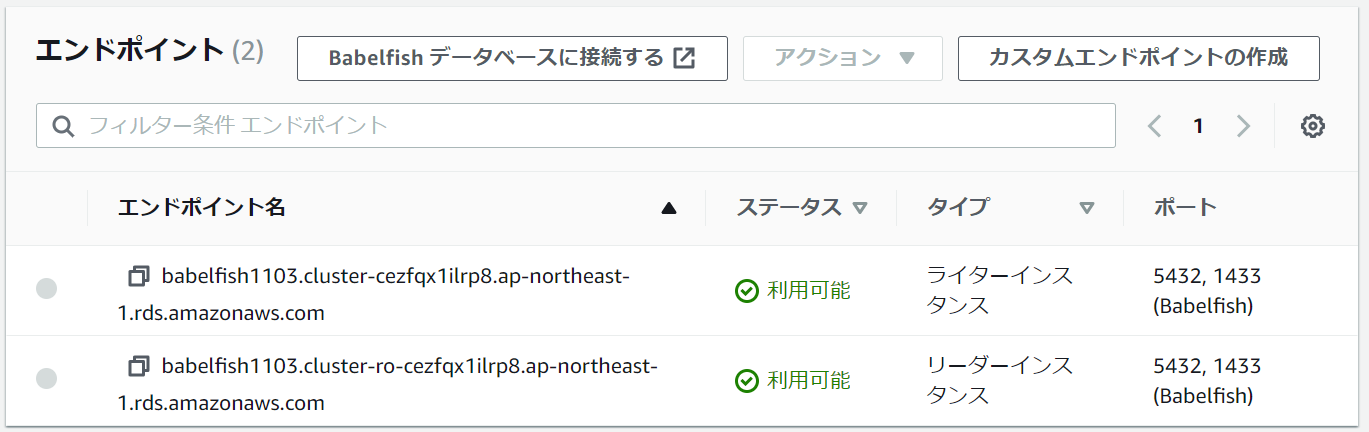
[rds-ec2]



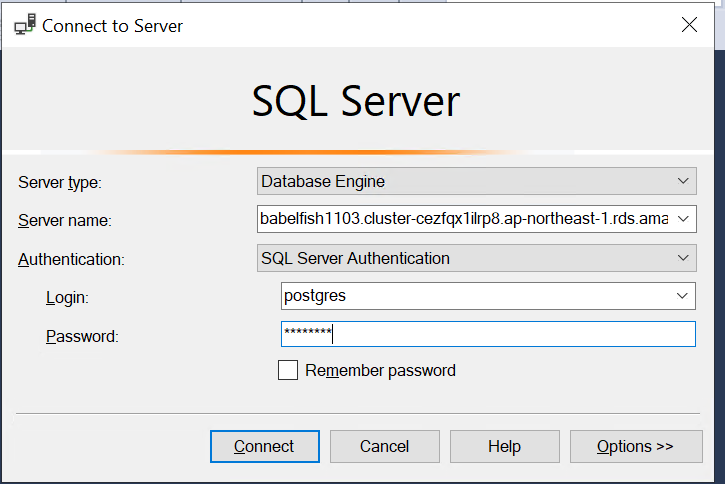
[ec2-rds]



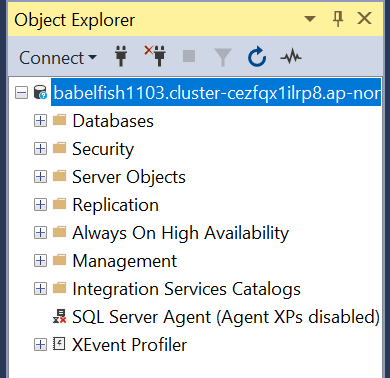
1. ライターインスタンスオのエンドポイントをコピーします



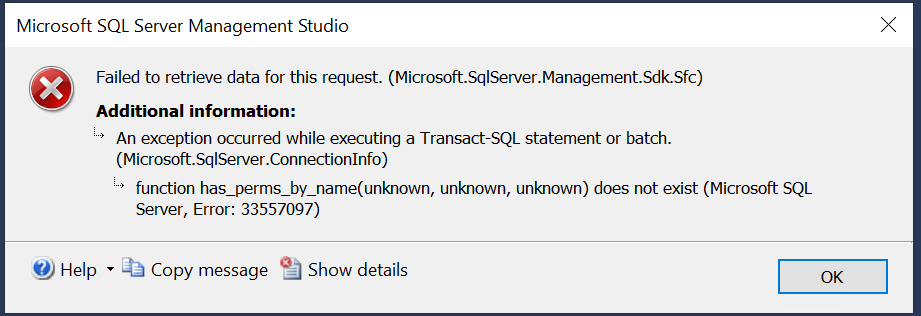
1. リモートデスクトップに戻りインスタンスに対して接続します。AuthenticationはSQL Serverに切り替えます。Login: は起動時に何も指定していない限り[postgres]です



1. 以下のように表示されたら起動が完了です



1. 色々操作するとエラーが以下のように出ます。BabelfishはTSQLを模しているだけでありSQL Serverをエミュレートしているわけではないためです



1. 画面上部[New Query]をおします



1. 以下のSQLを実行します

USE [master]

DROP DATABASE IF EXISTS MySSMSTest --Delete the database so that it won't conflict with CREATE DATABASE

CREATE DATABASE MySSMSTest --Create a new database named MySSMSTest

コピペした後Executeをおします

1. 続いて以下のSQLを実行します

USE [MYSSMSTest] --For the rest of the query, use the database MySSMSTest

CREATE TABLE [dbo].[Student] (

[StudentId] [int] NOT NULL,

[Name] [nvarchar](50) NOT NULL,

[Age] [tinyint] NOT NULL,

CONSTRAINT [PK\_Student] PRIMARY KEY

(

[StudentId] ASC

)

) ON [PRIMARY]

GO

-- Insert some sample data

INSERT INTO [dbo].[Student]([StudentId],[Name],[Age]) VALUES (1,'John',15)

INSERT INTO [dbo].[Student]([StudentId],[Name],[Age]) VALUES (2,'Kate',20)

INSERT INTO [dbo].[Student]([StudentId],[Name],[Age]) VALUES (3,'Sara',65)

INSERT INTO [dbo].[Student]([StudentId],[Name],[Age]) VALUES (4,'Mike',19)

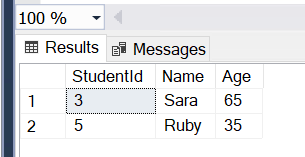
INSERT INTO [dbo].[Student]([StudentId],[Name],[Age]) VALUES (5,'Ruby',35)

1. 先程Insertしたデータをクエリしてみます

-- Select the rows

USE [MYSSMSTest] --For the rest of the query use the database MySSMSTest

SELECT \* FROM [dbo].[Student] WHERE Age > 30



1. TSQLベースでAurora Postgresが操作できていることがわかります

日本語対応について：Babelfish 2.1 で日本語照合順序がサポートされており日本語も対応しています。ただしRDS SQLと異なり、Japanese\_CS\_AS、Japanese\_CI\_ASには対応していますがJapanese\_BINは対応していません。

また、既知の問題としてPostgreSQL側から文字をinsert/updateした場合、SQL Server側の設定以上のバイト数が入ってしまう可能性があり、そのデータをSQL Server側から検索した場合、タイムアウトまで待ってしまう、という問題があります。

SQL Serevr のchar/varcharの設定はバイトで行いますPostgreSQLのchar/varcahrの設定は文字数で行うため、SQL Serverとして想定以上のバイト数が格納されてしまうケースで発生します。

[Postgresエンドポイントでの確認]

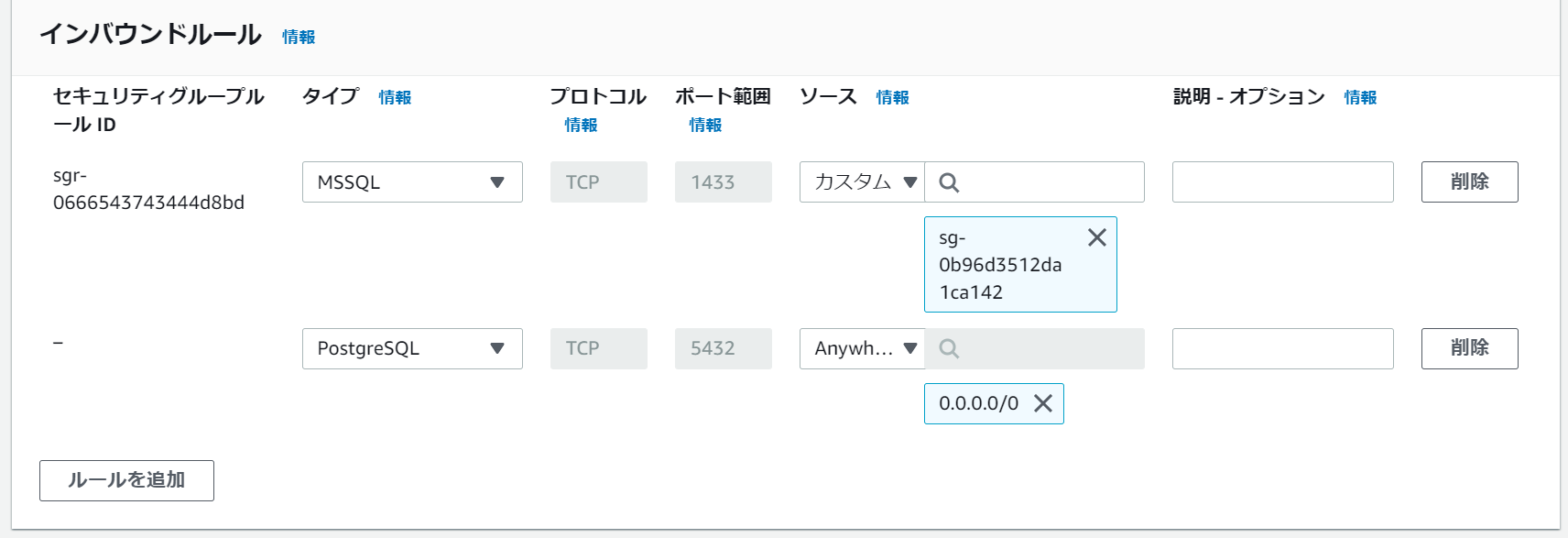
先程行った操作（テーブル作成等）を今度はPostgresから確認します

1. Cloud9のマネージメントコンソールへ移動します（ブラウザ別タブで）
2. [Create environment]をおします
3. 適当な名前を付け[Next Step]をおします
4. すべてデフォルトのまま[Next Step]をおします。次の画面で[Create environment]をおします
5. 以下のコマンドでpsqlをインストールします

sudo yum install postgresql

途中入力が求められますので[y]をおします

1. 先程修正を行ったセキュリティグループ[rds-ec2]を以下に修正します



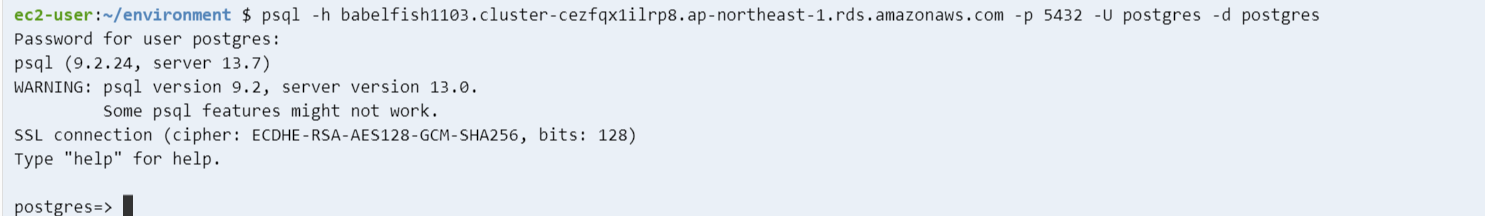
タイプPostgresを一度削除したのち再度追加して、ソースをAnywhereに変更

1. インストールが完了したら以下のコマンドを実行し接続します

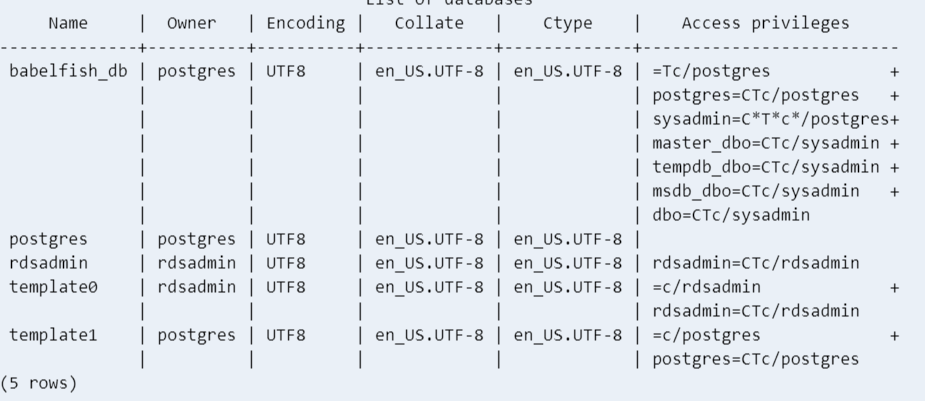
psql -h ホスト名 -p ポート番号 -U ロール名 -d データベース名

（日本語名は全て環境に置き換えてください。以下は例です）

1. パスワードを入力し、以下の様になれば接続成功です



1. [\l]と実行してください。[babelfish\_db]とはいっているのがわかります。Babelfish経由で作成されたオブジェクトは全てここに含まれています



1. [\c babelfish\_db]を実行して接続先データベースを切り替えます
2. [select \* from dbo.student;]を実行すると先程のテーブルが確認できます

[INSERT INTO dbo.student (studentid,name,age) VALUES (6,'kame',49);]で実行したものをSQL Studioから確認することもできます

おつかれさまでした！

削除は以下を行ってください：

EC2

Cloud9

RDS