

[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)

 Microsoft

Windows

検証済みプロバイダー

Microsoft Windows Server 2019 with SQL Server 2017 Standard
ami-0052ef2f89f1bfb4f (64 ビット (x86))
Microsoft Windows 2019 Datacenter edition, Microsoft SQL Server 2017 Standard. [English]
プラットフォーム: windows ルートデバイスタイプ: ebs 仮想化: hvm ENA 有効: はい

選択

64 ビット (x86)

6. インスタンスタイプは,m5.large を選びます
7. [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。
8. 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
9. 起動が完了するまでしばらく待ちます。(ステータスチェックがそのうち緑の表示になります)

<input type="checkbox"/>	Name	インスタンス ID	インスタンス...	インスタンス...	ステータスチェ...
<input type="checkbox"/>	winsql	i-Oe31778100d4a4eb2	実行中	m5.large	初期化しています。ア

10. 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
11. [RDP クライアント]のタブを選びます
12. [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
13. [パスワードの取得]をクリックします

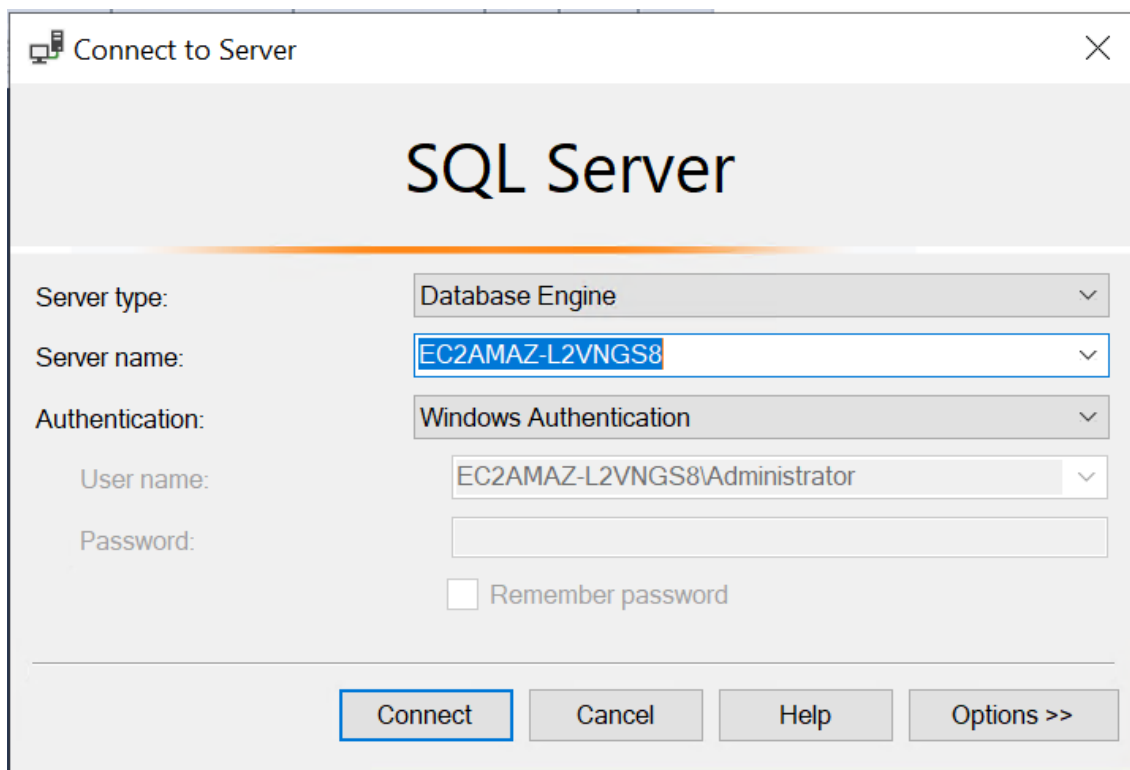
Public DNS
ec2-18-183-143-46.ap-northeast-1.compute.amazonaws.com

パスワード パスワードを取得

ユーザー名
Administrator

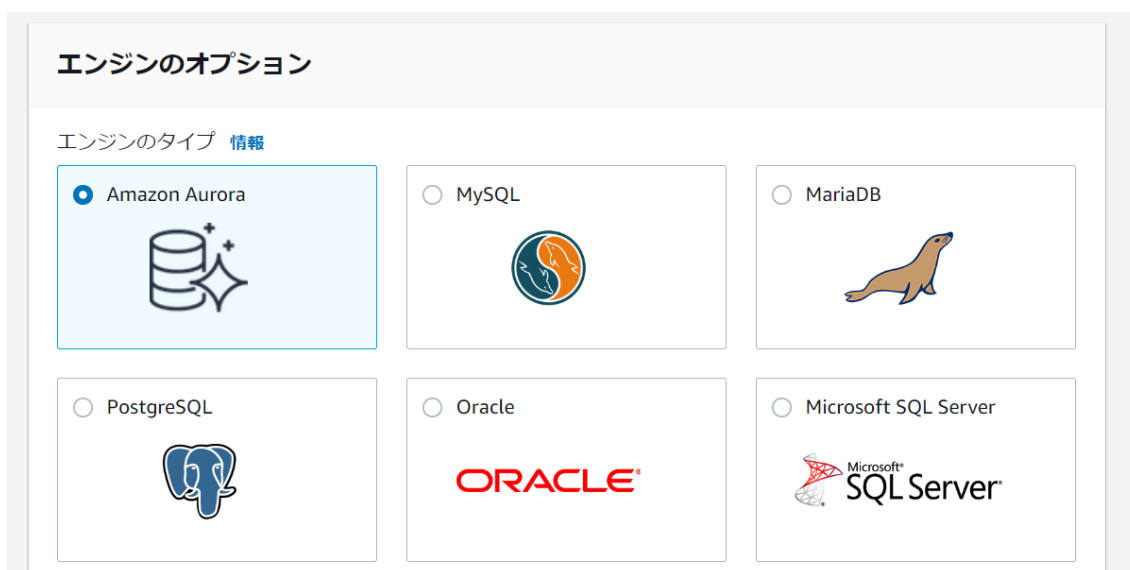
14. 先程ダウンロードした鍵をコピーし[パスワードを復号化]をおします
15. 表示されたパスワードを用いてリモートデスクトップ接続を行います

16. 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
17. 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]

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



21. エディションは Postgres を選びます

エディション

- ☐ Amazon Aurora MySQL 互換エディション
- ☒ Amazon Aurora PostgreSQL 互換エディション

22. 以下のトグルをオンにします

最も高いパフォーマンスとスケーラビリティを提供します。

- ☒ Babelfish for PostgreSQL 機能をサポートするバージョンを表示
Microsoft SQL Server から Aurora PostgreSQL への移行をより迅速、安価、低リスクで行えます。

23. [開発/テスト] を選びます

テンプレート

お客様のユースケースに合わせてサンプルテンプレートを選択します。

- ☒ 本番稼働用
高い可用性と、高速で安定したパフォーマンスのために、デフォルト値を使用します。
- ☐ 開発/テスト
このインスタンスは本番稼働環境ではない開発で使われます。

24. 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

25. [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

コンピューティングリソース

このデータベースのコンピューティングリソースへの接続を設定するかどうかを選択します。接続を設定すると、コンピューティングリソースがこのデータベースに接続できるように、接続設定が自動的に変更されます。

- ☐ EC2 コンピューティングリソースに接続しない
このデータベースのコンピューティングリソースへの接続を設定しないでください。後でコンピューティングリソースへの接続を手動で設定できます。
- ☒ EC2 コンピューティングリソースに接続
このデータベースの EC2 コンピューティングリソースへの接続を設定します。

26. 先程作成した EC2 を選びます

27. [Babelfish をオンにする] を選びます

Babelfish の設定 - 新規 情報

- ☐ Babelfish をオンにする
Microsoft SQL Server から Aurora PostgreSQL への移行をより迅速、安価、低リスクで行えます。

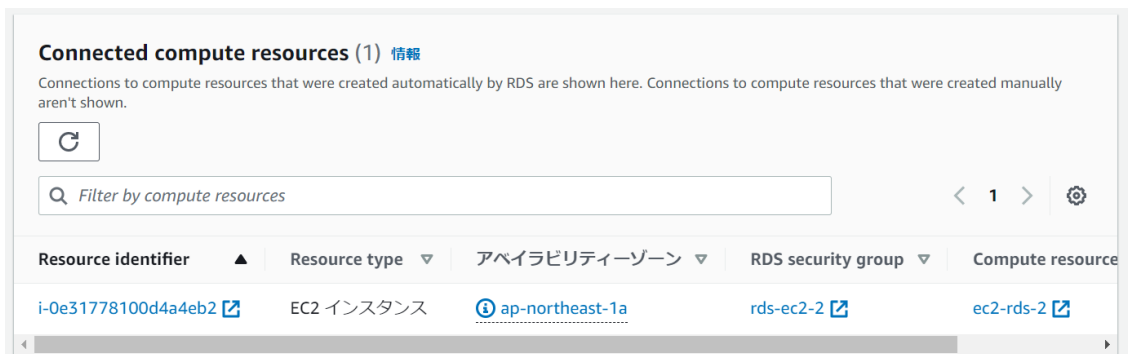
28. [Performance Insights] を外します

29. [データベースの作成] をおします

30. 起動するまでしばらく待ちます。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

31. 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]



32. 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

エンドポイント (2)

Babelfish データベースに接続する

アクション

カスタムエンドポイントの作成

Q フィルター条件 エンドポイント

< 1 >

エンドポイント名	ステータス	タイプ	ポート
 babelfish1103.cluster-cezfqx1ilrp8.ap-northeast-1.rds.amazonaws.com	 利用可能	ライターインスタンс	5432, 1433 (Babelfish)
 babelfish1103.cluster-ro-cezfqx1ilrp8.ap-northeast-1.rds.amazonaws.com	 利用可能	リーダーインスタンс	5432, 1433 (Babelfish)

33. 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

Connect to Server

SQL Server

Server type: Database Engine ▼

Server name: babelfish1103.cluster-cezfqx1ilrp8.ap-northeast-1.rds.ama ▼

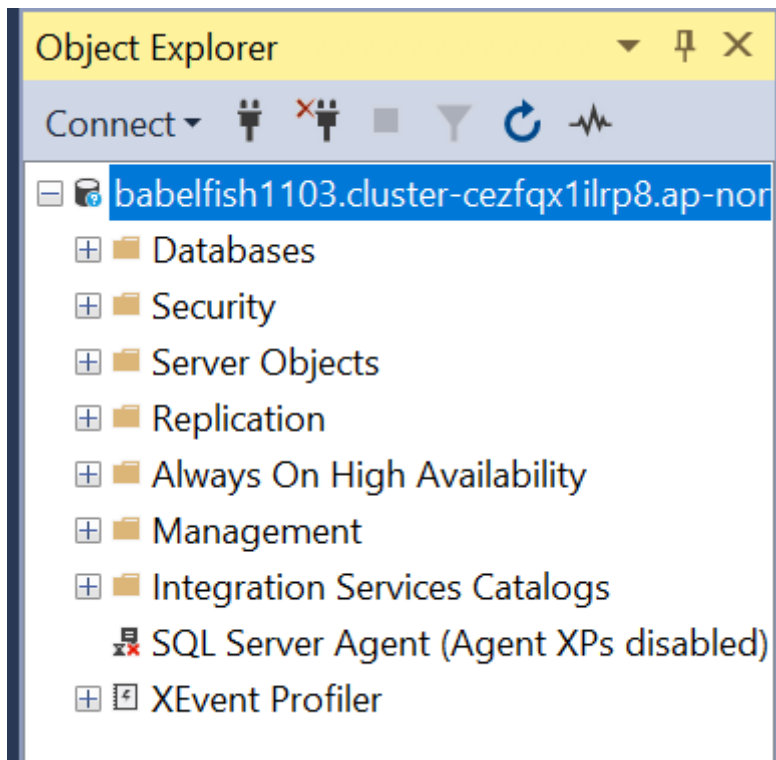
Authentication: SQL Server Authentication ▼

Login: postgres ▼

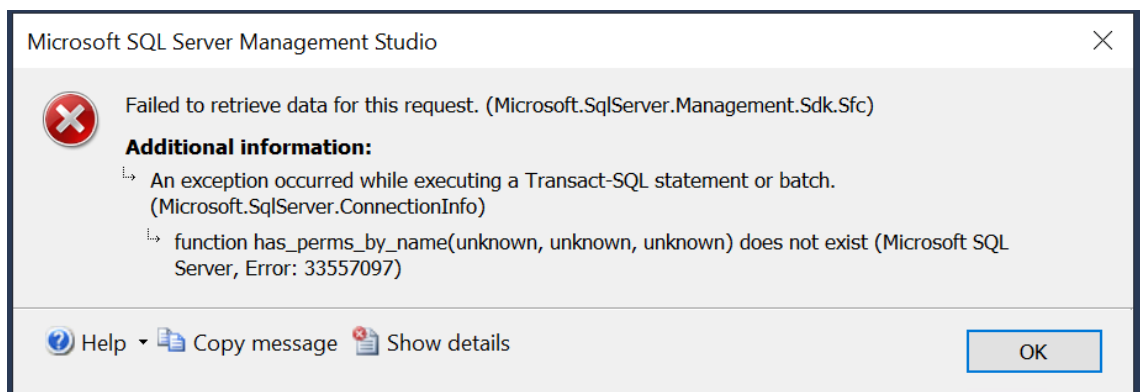
Password:

☐ Remember password

34. 以下のように表示されたら起動が完了です



35. 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



36. 画面上部[New Query]をおします



37. 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
```

コピーした後 Execute をおします

38. 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)
```

39. 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
```

100 %

Results		Messages	
	StudentId	Name	Age
1	3	Sara	65
2	5	Ruby	35

40. 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つという問題があります。

SQL Server の char/varchar の設定はバイトで行います PostgreSQL の char/varchar 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が発生します。

[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

41. 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）
42. [Create environment]をおします
43. 適当な名前を付け[Next Step]をおします
44. 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
45. 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
46. 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

47. 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)

48. パスワードを入力し、以下の様になれば接続成功です

```
ec2-user:~/environment $ psql -h babelfish1103.cluster-cezfqx1ilrp8.ap-northeast-1.rds.amazonaws.com -p 5432 -U postgres -d postgres
Password for user postgres:
psql (9.2.24, server 13.7)
WARNING: psql version 9.2, server version 13.0.
         Some psql features might not work.
SSL connection (cipher: ECDHE-RSA-AES128-GCM-SHA256, bits: 128)
Type "help" for help.

postgres=>
```

49. [¥]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。Bablefish 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

Name	Owner	Encoding	Collate	Ctype	Access privileges
babelfish_db	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=Tc/postgres +
					postgres=Ctc/postgres +
					sysadmin=C*t*c*/postgres+
					master_dbo=Ctc/sysadmin +
					tempdb_dbo=Ctc/sysadmin +
					msdb_dbo=Ctc/sysadmin +
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	dbo=Ctc/sysadmin
rdsadmin	rdsadmin	UTF8	en_US.UTF-8	en_US.UTF-8	rdsadmin=Ctc/rdsadmin
template0	rdsadmin	UTF8	en_US.UTF-8	en_US.UTF-8	=c/rdsadmin +
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	rdsadmin=Ctc/rdsadmin
					=c/postgres +
					postgres=Ctc/postgres

(5 rows)

50. [¥c babelfish_db]を実行して接続先データベースを切り替えます

51. [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