RDS Proxy / Lambda ワークショップ

2021/09/29

シニアエバンジェリスト
亀田治伸

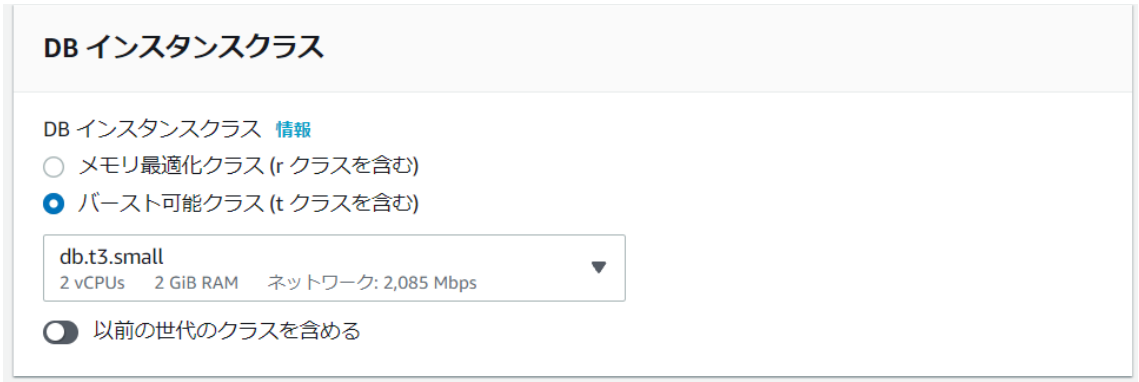
[Aurora の起動]
まずは、Lambda から RDS Proxy を経由して呼び出される RDS インスタンスの Aurora を起動します。2021/09 現在 RDS Proxy は Aurora に加えて、MySQL, Postgres RDS をサポートしています。

1. マネージメントコンソールの RDS 画面に移動します

2. [データベースの作成]をおします
   デフォルトで Aurora の MySQL 互換が選択されていますのでそのままにします。
3. [テンプレート]で[開発/テスト]を選択し 1 台構成を選びます

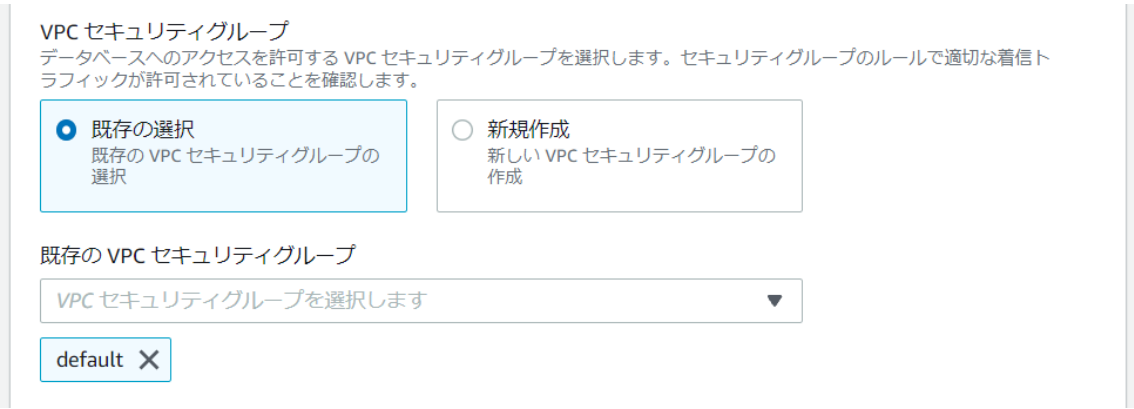
4. [設定]で任意のパスワードを指定します。このサンプルでは[password]を使います

5. [DB インスタンスクラス]で[バースト可能クラス]を選び一番小さい t3.small を選びます

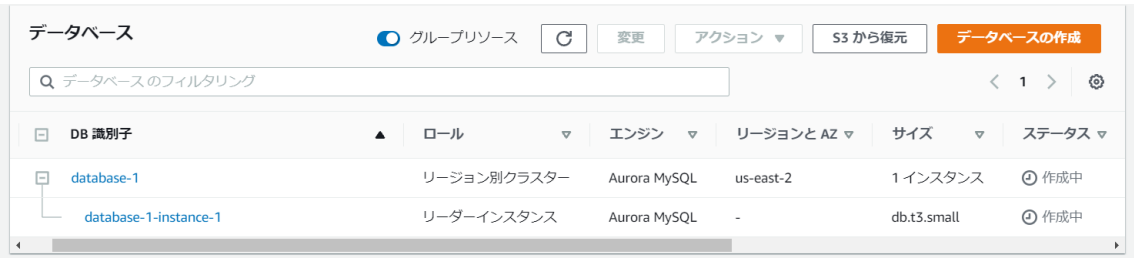


6. [可用性と耐久性]はデフォルトのままシングル構成にします
7. [接続]はデフォルト VPC を選びます

8. [VPC セキュリティグループ]は[default]を選びます



9. [データベースの作成]をおします
10. 10 分から 15 分程度まちます

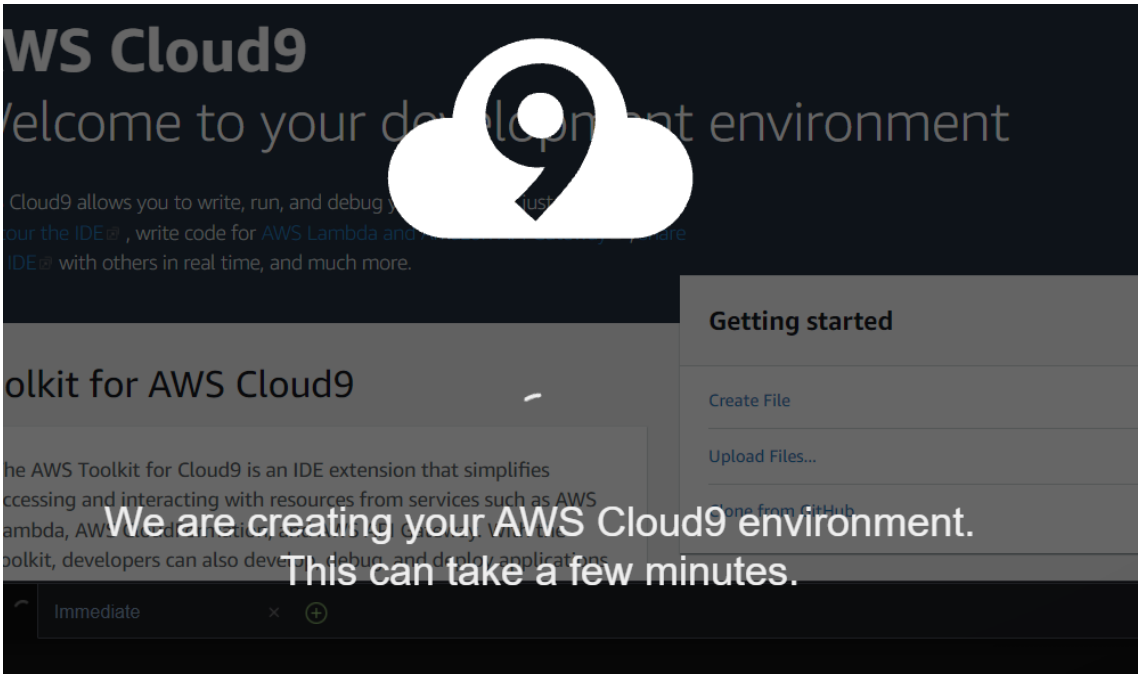


[Cloud9 の構築]

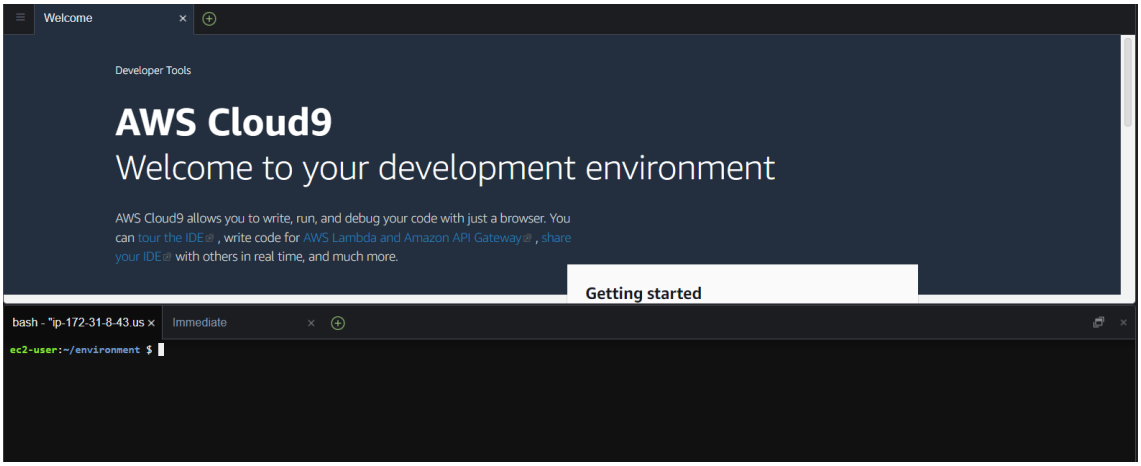
構築された MySQL に接続を行いテスト用データを書き込む Cloud9 を構築します

11. マネーコメントコンソールで Cloud9 の画面にアクセスします（ブラウザの別タブを開いた方が便利です）
12. [Create environment]ボタンをおします

13. [Name]に適当な名前をつけ、[Next Step]をおします
14. 全てデフォルトのまま[Next Step]をおします。この際、[Network settings(advanced)]で、RDS と同じデフォルト VPC が選ばれていることを確認してください。
15. [Create environment]をおします
16. 作成中が表示されますのでしばらく待ちます



17. 以下のようにコンソールで操作可能となれば構築成功です



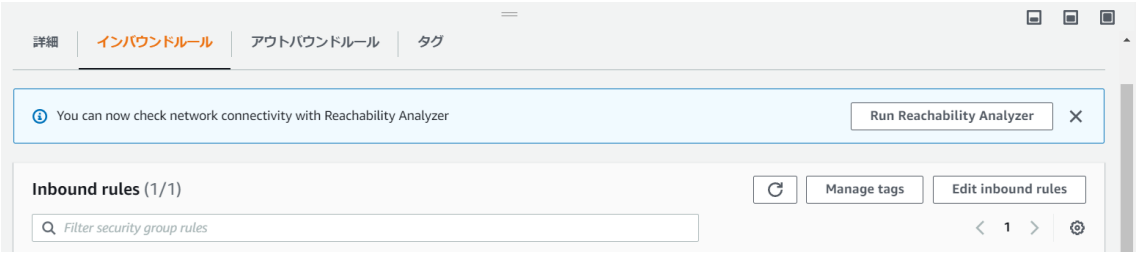
[RDS Aurora のネットワーク解放]

構築した Cloud9 や後ほど構築する Lambda 関数からの接続を可能とするため、セキュリティグループを書き換えます

18. RDS のマネージメントコンソールに戻り Aurora が起動済であることを確認します

19. ライターをクリックしてセキュリティグループを特定します。(注意、クラスターはライターに対する CNAME の存在ですので、設定変更は実体であるライターに対して行います)

20. セキュリティグループを右クリックして新しいタブでセキュリティグループの設定画面を開き、[インバウンドルール]のタブをクリックします



21. [Edit inbound rules]をおして[削除]ボタンで今あるルールを全て削除します

22. [ルールの追加]をおして、以下のように VPC 内のすべての通信を通すようなルールを設定し、[ルールを保存]をおします

[Cloud9 からの接続テスト]

Cloud9 からまずは Aurora に対して接続テストを行ってみます。

23. Aurora のマネージメントコンソールで、リージョン別クラスターをクリックして[ライターインスタンス]のエンドポイントをコピーします

24. Cloud9 コンソールから以下のコマンドを実行します。<cluster endpoint>は上記のものに置換してください

    mysql -u admin -h <cluster endpoint> -p

25. パスワードを入力して以下が表示されたらログインが成功しています

```
ec2-user:~/environment $ mysql -u admin -h database-1.cluster-cagbxmtcmyq2.us-east-2.rds.amazonaws.com -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.7.12 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
```

26. 試しに[show databases;]を実行してみてください

[Secrets Manager]の設定

RDS Proxy は RDS への接続に用いるパスワードは独自ではなく Secrets Manager と連携して管理します。暗号化やローテーションも Secrets Manager で管理されます。

27. マネージメントコンソールで Secrets Manager にアクセスします（ブラウザの別タブで開くことをお勧めします）

28. [新しいシークレットを保存する]をおします

29. [シークレットの種類を選択]から[RDS データベースの認証情報]を選びます

30. ユーザー名、パスワードに Aurora へのログイン情報を入力します
31. 先程作成した Aurora が表示されていますので、そちらを選択し、[次へ]をおします

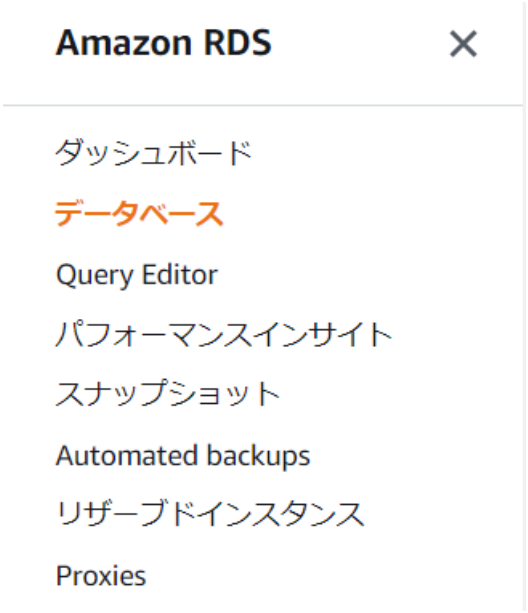
32. [シークレットの名前]に適当な値を入力し、[次へ]をおします
33. 次のシークレットのローテーション設定画面ではそのままデフォルトの状態で[次へ]をおします
34. 最後の画面で[保存]をおします。以下のようにシークレットが保存されます

[RDS Proxy の起動]

上記で作成した Secrets Manager に保存されている ID/パスワードを用いて、作成済

Aurora へ接続するための RDS Proxy を起動します

35. RDS のマネージメントコンソール、左ペインから[Proxies]をおします



36. [プロキシを作成]をおします
37. [プロキシ識別子]に任意の名前をいれます

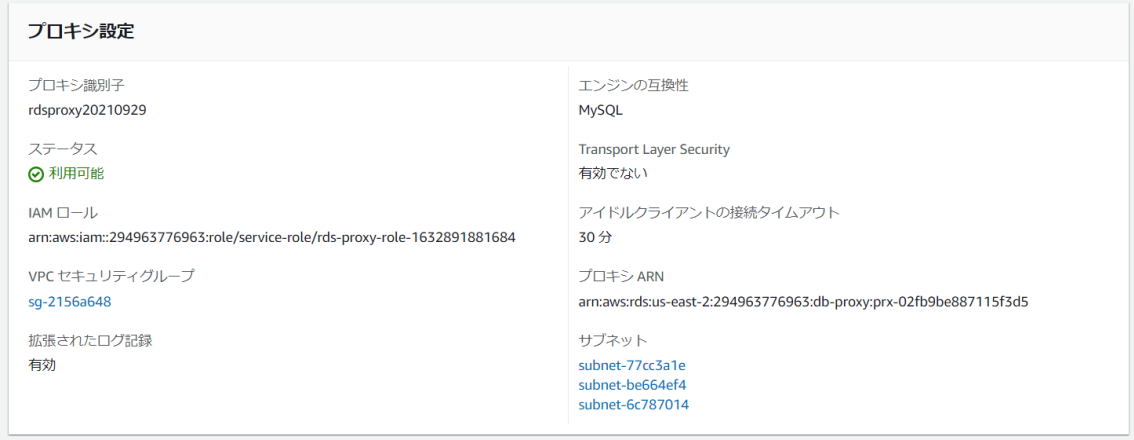
38. [エンジンの互換性]は[MySQL]を選びます

39. [データベース]に Aurora クラスターをドロップダウンから選びます

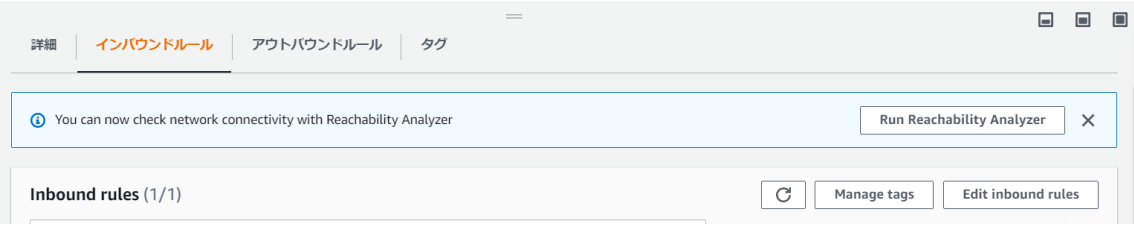
40. [接続]の項目で先程作成した Secrets Manager の値をドロップダウンから選びます

41. [拡張されたログ記録を有効にする]にチェックをつけ、[プロキシを作成]をおします
42. 作成中となりますので、10 分程度待ちます

43. 作成されたら、プロキシ識別子で作成済プロキシのリンクをクリックして、[VPC セキュリティグループ]を右クリックで開きます

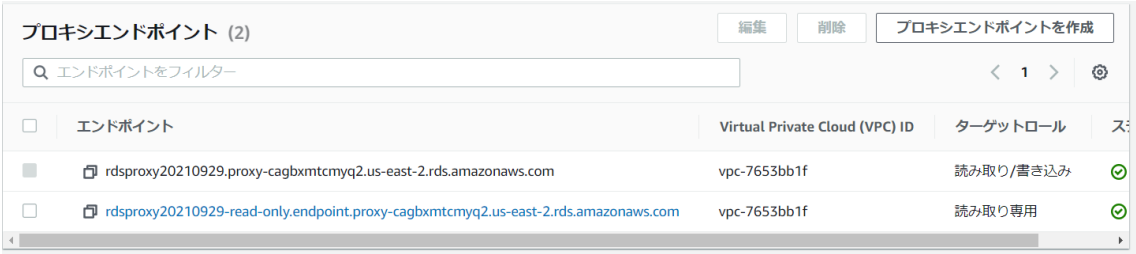


44. [インバウンドルール]のタブから[Edit inbound rules]をおします



45. [削除]をおして今あるルールを全て削除します
46. [ルールの追加]をおして以下のように VPC 内部の通信を全てとおして、[ルールを保存]をおします

47. 再度 RDS Proxy のマネージメントコンソールから、[エンドポイント]をコピーします（読み取り/書き込みの方です）



48. Cloud9 から以下のコマンドを実行します

mysql -u admin -h <proxyendpoint> -p

49. 以下が表示されれば完了です

```
ec2-user:~/environment $ mysql -u admin -h rdsproxy20210929.proxy-cagbxmtcmyq2.us-east-2.rds.amazonaws.com -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 3642024877
Server version: 5.7.12 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

50. 以下のコマンドを 1 行づつ実行します

```
show databases;
create database test;
create table test.sample (id int, value varchar(10));
INSERT INTO test.sample VALUES (1,"test");
select * from test.sample;
```

[Lambda 関数の作成]

今までの手順で、VPC 内部から RDS Proxy 経由で RDS へアクセスできるようになりました。これから VPC 内部へアクセス可能な Lambda 関数を作成していきます

51. Lambda のマネージメントコンソールを開きます
52. [関数の作成]をおします
53. [一から作成]を選んで、[関数名]に適当な名前を入れます。ランタイムは[Node.js14]を選びます

54. [詳細設定]からデフォルト VPC を選んで、すべてのサブネットを登録します

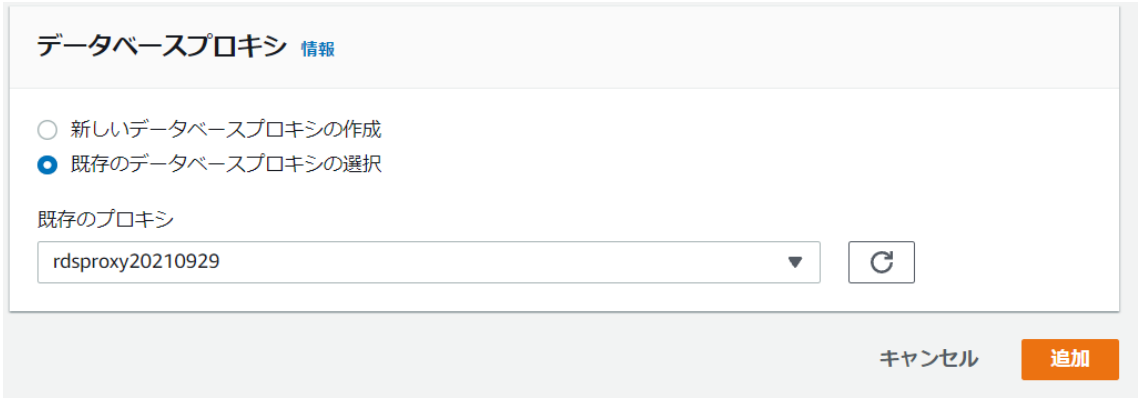
55. セキュリティグループはデフォルトを選んで、以下の通り VPC 内部のすべての通信が通るようになっていることを確認してください

56. [関数の作成]をおします
57. VPC 内部へのアクセスインターフェースを作成するため、いつもより Lambda 関数の作成に時間がかかりますが待ちます。以下のように緑のバーが表示されれば作成完了です

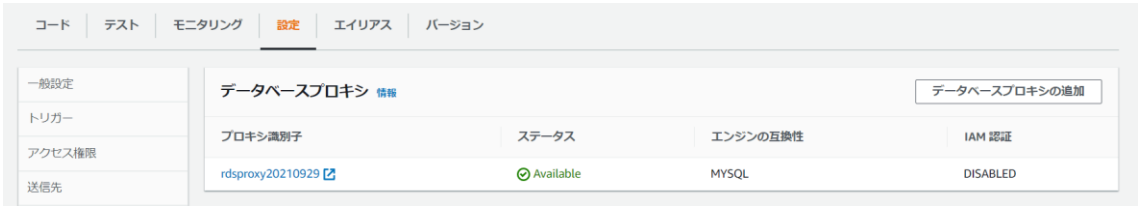
58. 画面中ほどにあるタブで[設定]をクリックし、左ペインから[VPC]をクリックして下さい

59. 以下の通り VPC 内部の通信が通るようになっていることを確認します

60. 左ペインから[データベースプロキシ]をクリックします。
61. [データベースプロキシの追加]をおします
62. [既存のデータベースプロキシの選択]を選び、作成済の RDS Proxy を選び、[追加]をおします

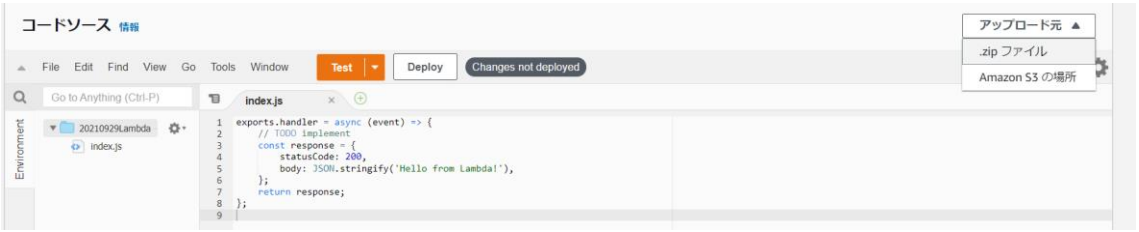


63. 以下のように表示されれば設定が完了です



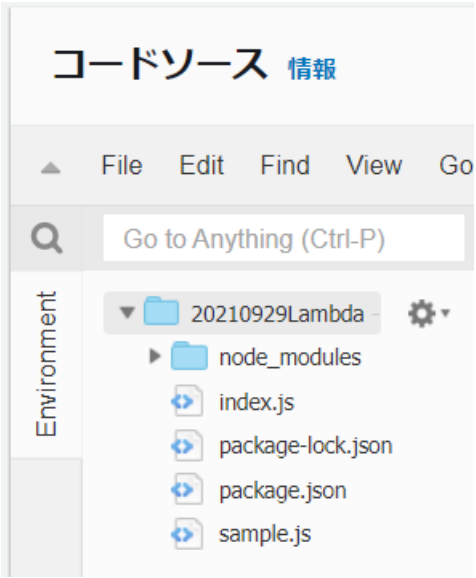
64. タブで[コード]を選びます

65. [アップロード元]ボタンから[.zip ファイル]を選びます

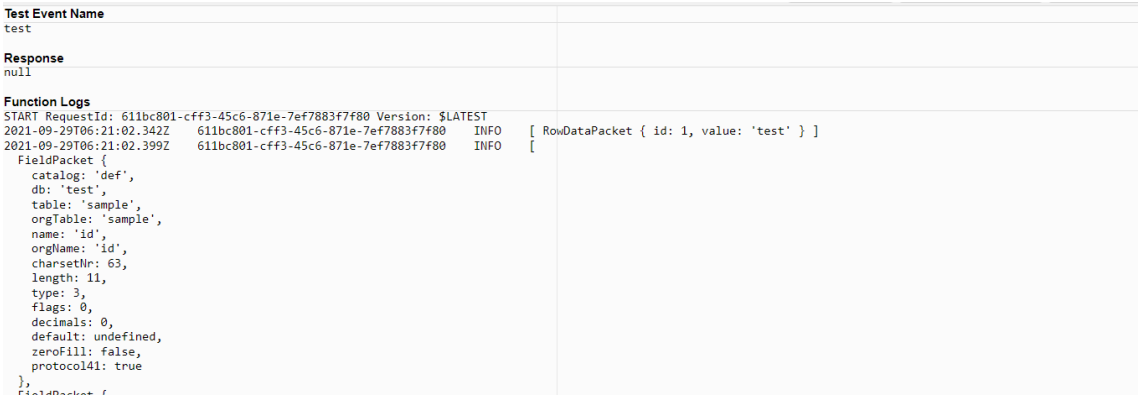


66. Github にある、[lambdamysql.zip]を適当な個所に保存しアップロードします

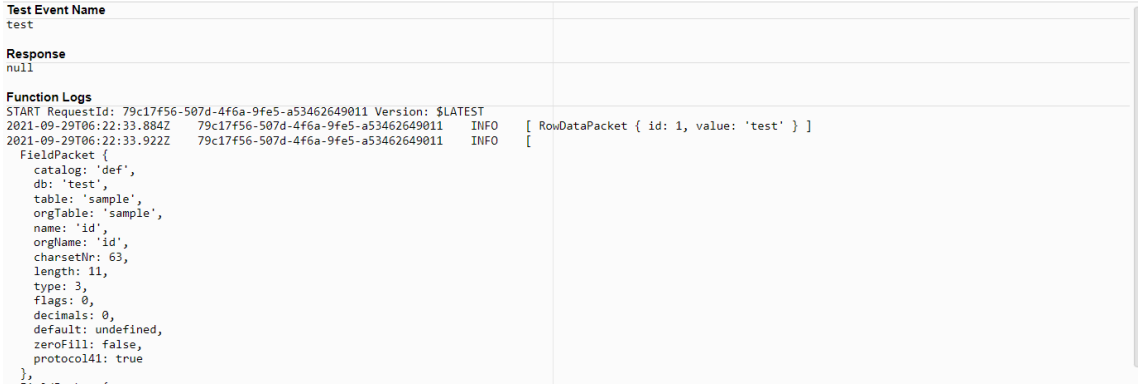
67. 以下のようにパッケージがインポートされます



68. Index.js を開いて、[host]部分をまずは RDS のエンドポイントに設定します
69. [Deploy]をおして、[Test]をおします
70. イベント名に適当な名前を付けて[作成]をおします
71. 再度[Test]をおします
72. 以下の通り SQL の実行結果が出力されたら成功です



73. 再度、index.js の[host]を RDS Proxy のエンドポイントに置き換えて、[Deploy],[Test]の順番におします
74. 以下の通り Select クエリの実行結果が表示されれば成功です



おつかれさまでした！

以下を削除してください

- RDS Proxy
- RDS
- Lambda 関数
- IAM ロール（Lambda 用、RDS Proxy 用）