RDS Proxy / Lambda ワークショップ

2021/09/29

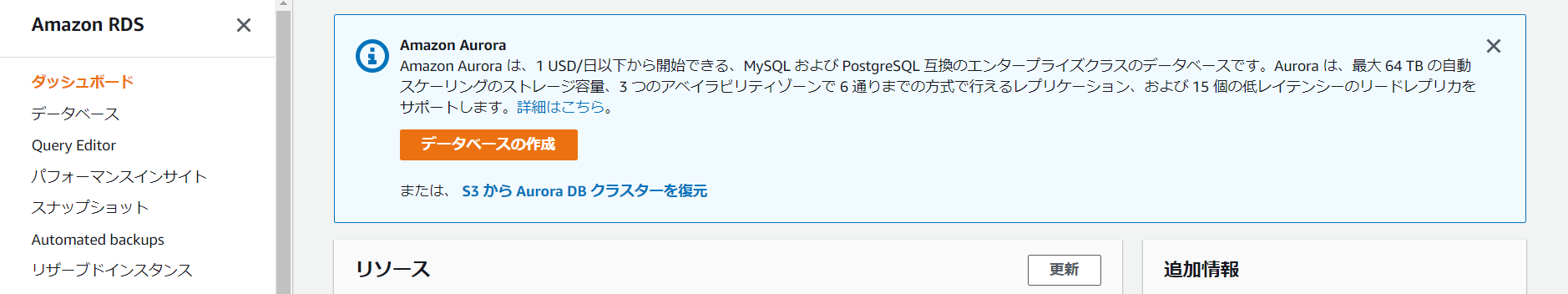
シニアエバンジェリスト

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。

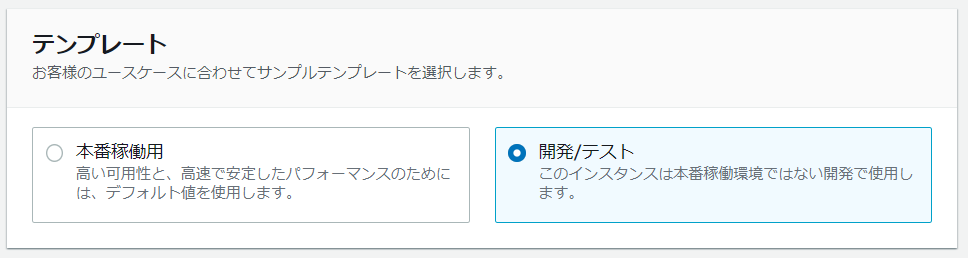
1. マネージメントコンソールのRDS画面に移動します



1. [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。

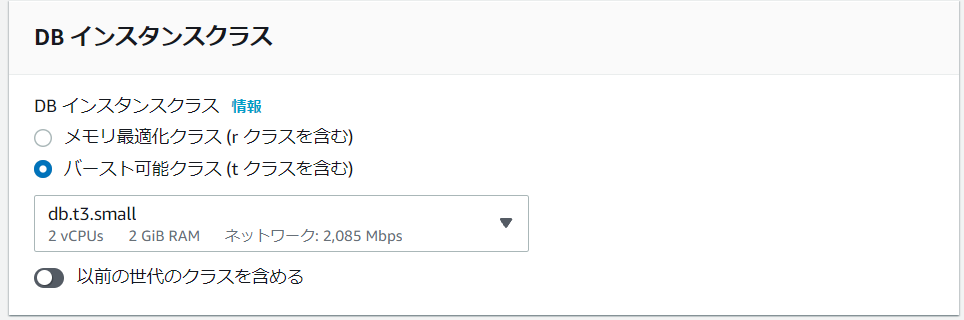
1. [テンプレート]で[開発/テスト]を選択し1台構成を選びます



1. [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



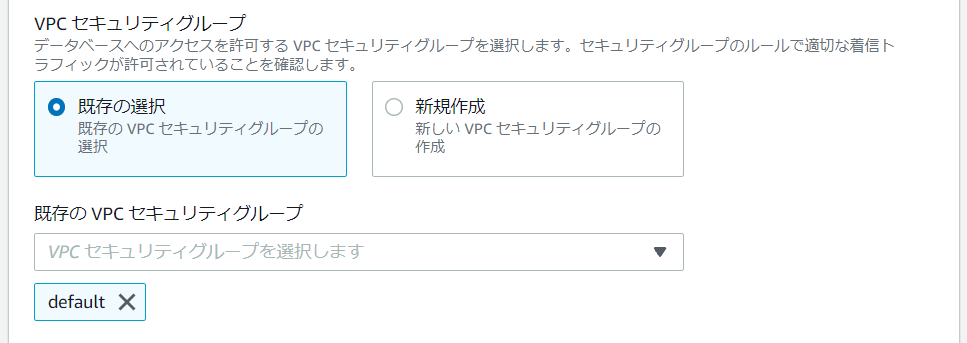
1. [DBインスタンスクラス]で[バースト可能クラス]を選び一番小さいt3.smallを選びます



1. [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
2. [接続]はデフォルトVPCを選びます



1. [VPCセキュリティグループ]は[default]を選びます



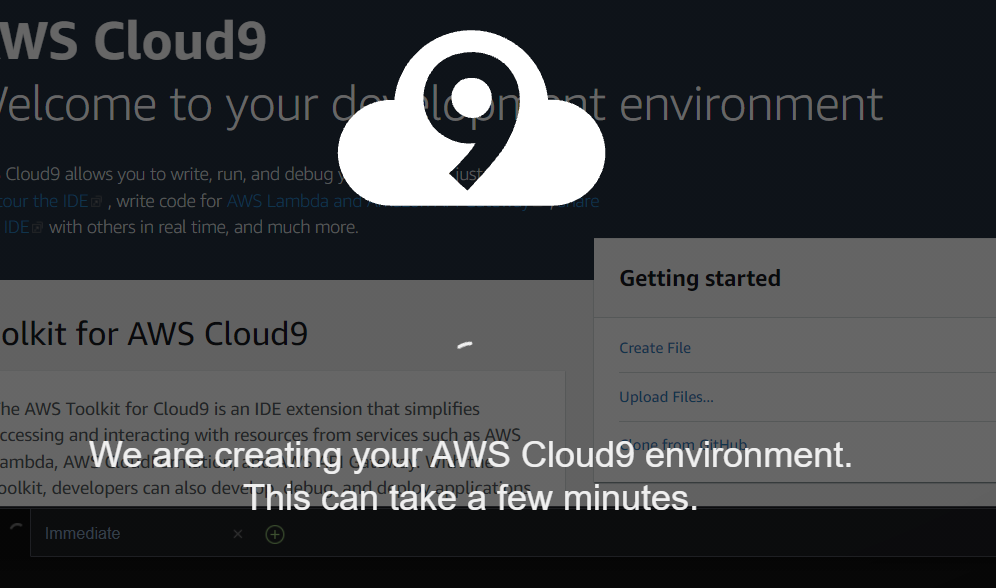
1. [データベースの作成]をおします
2. 10分から15分程度まちます



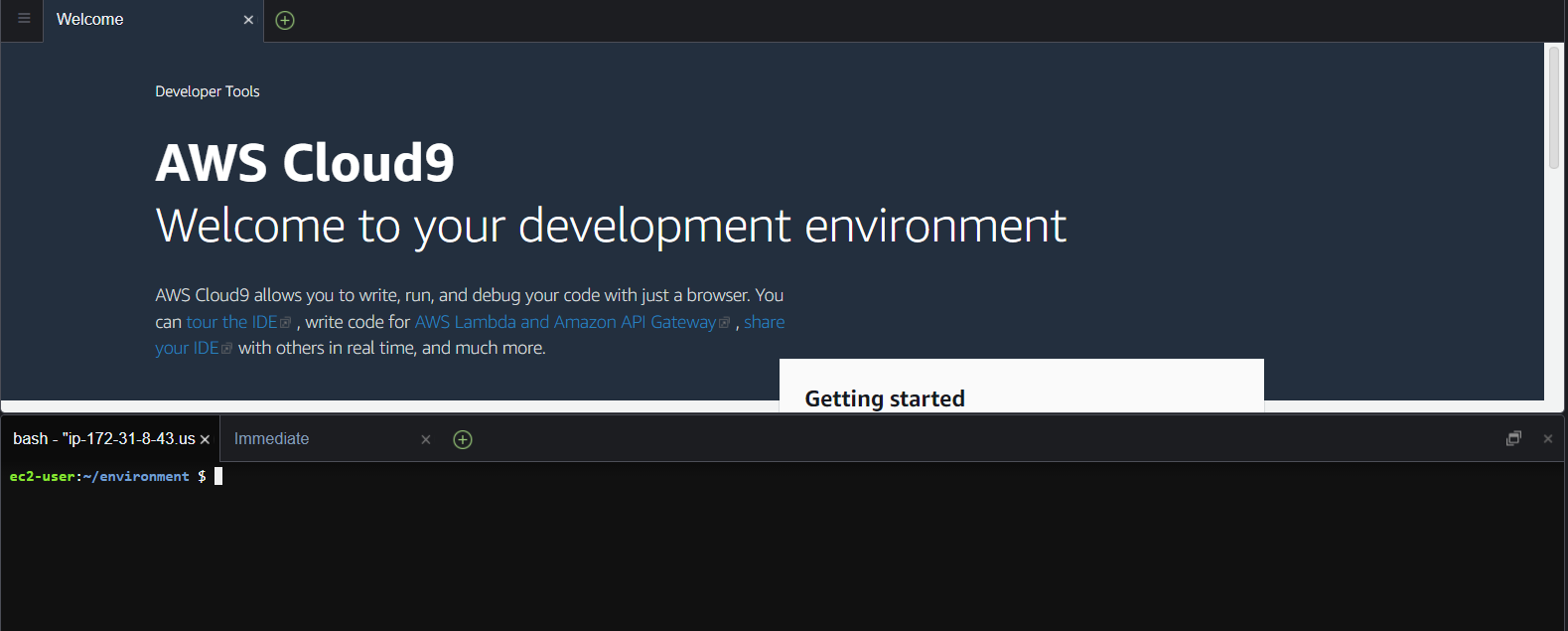
[Cloud9の構築]

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

1. 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）
2. [Create environment]ボタンをおします
3. [Name]に適当な名前をつけ、[Next Step]をおします
4. 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。
5. [Create environment]をおします
6. 作成中が表示されますのでしばらく待ちます



1. 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



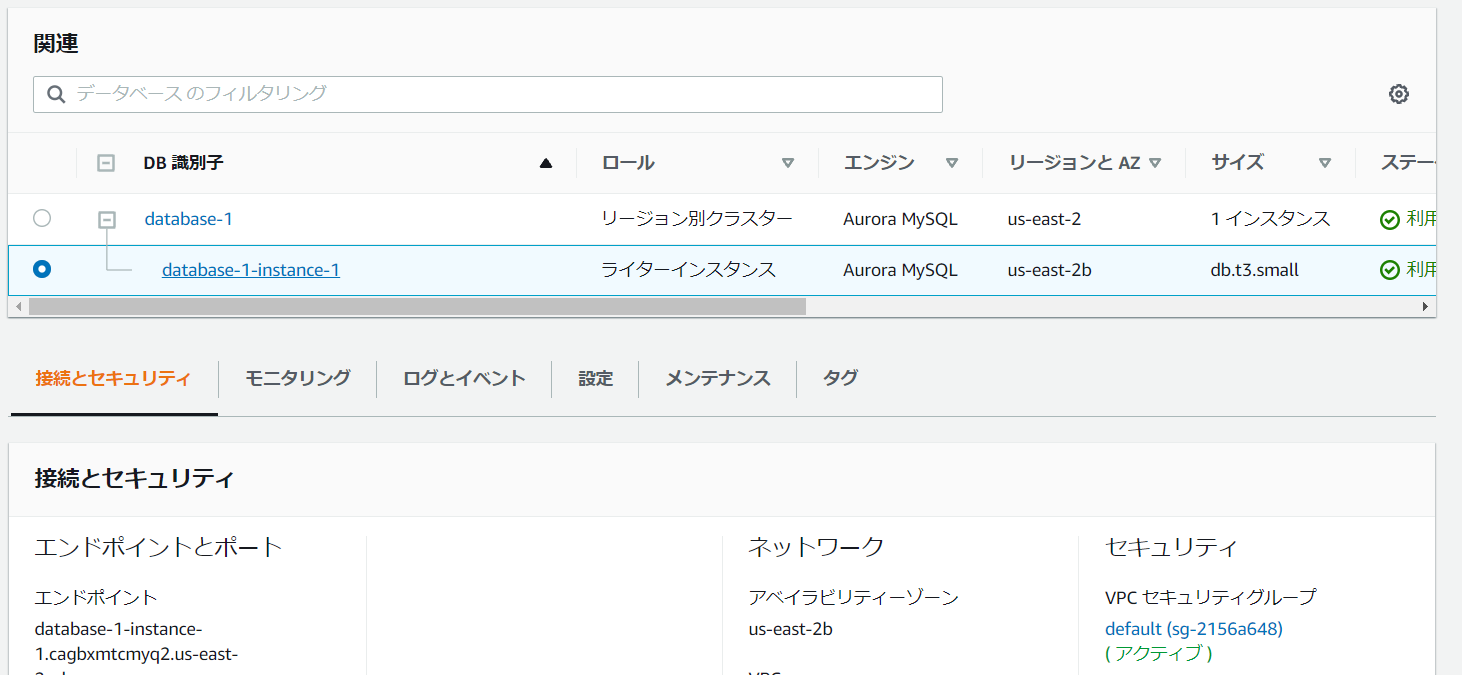
[RDS Auroraのネットワーク解放]

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

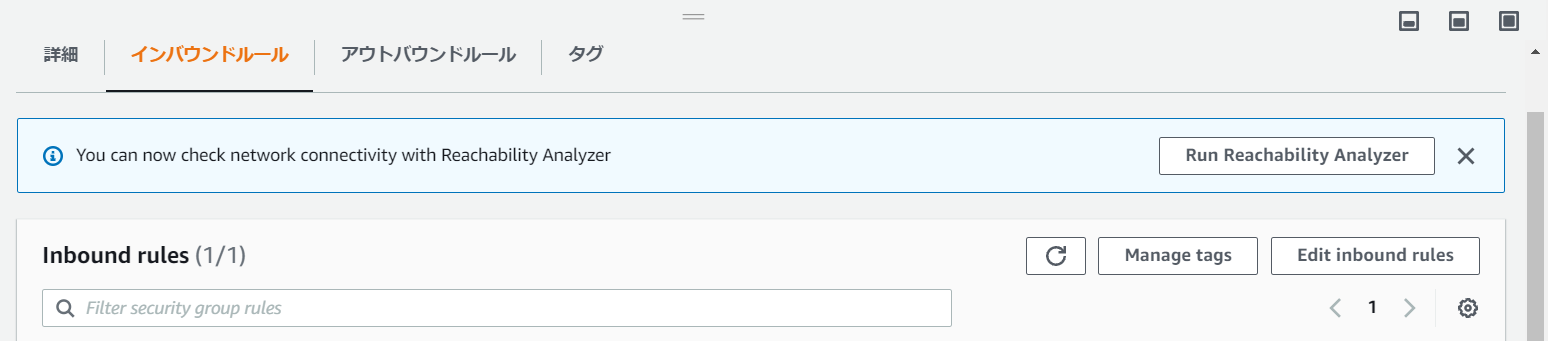
1. RDSのマネージメントコンソールに戻りAuroraが起動済であることを確認します



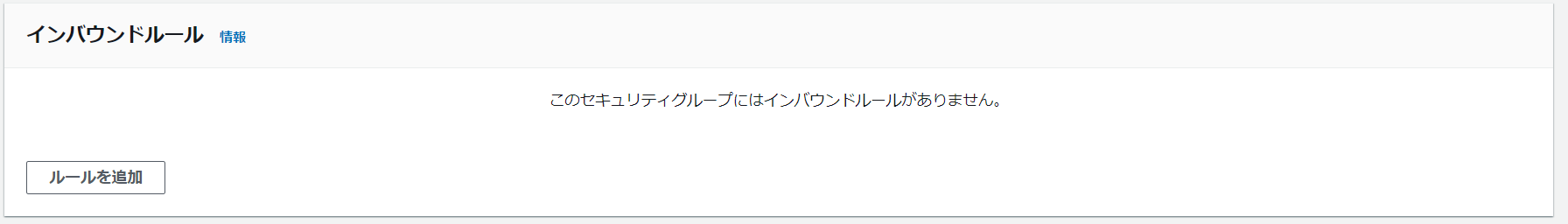
1. ライターをクリックしてセキュリティグループを特定します。（注意、クラスターはライターに対するCNAMEの存在ですので、設定変更は実体であるライターに対して行います）



1. 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



1. [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



1. [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。

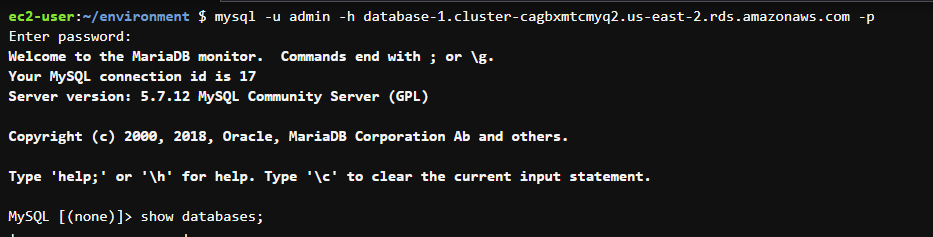
1. Auroraのマネージメントコンソールで、リージョン別クラスターをクリックして[ライターインスタンス]のエンドポイントをコピーします



1. 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

1. 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



1. 試しに[show databases;]を実行してみてください

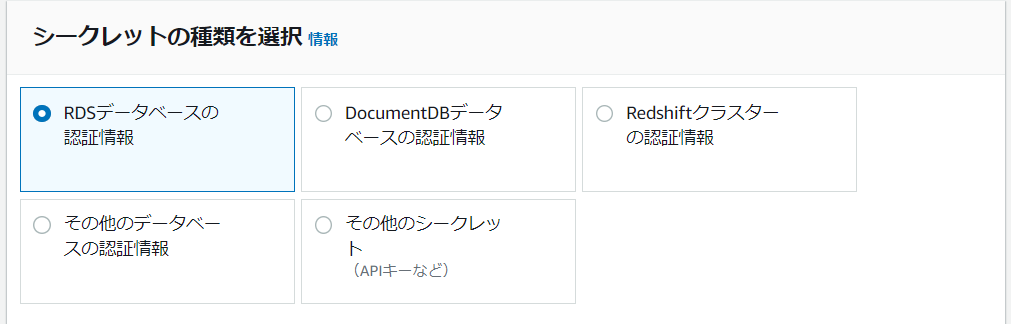
[Secrets Manager]の設定

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。

1. 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）
2. [新しいシークレットを保存する]をおします



1. [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



1. 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
2. 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



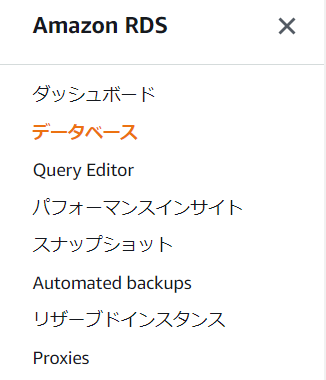
1. [シークレットの名前]に適当な値を入力し、[次へ]をおします
2. 次のシークレットのローテーション設定画面ではそのままデフォルトの状態で[次へ]をおします
3. 最後の画面で[保存]をおします。以下のようにシークレットが保存されます



[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

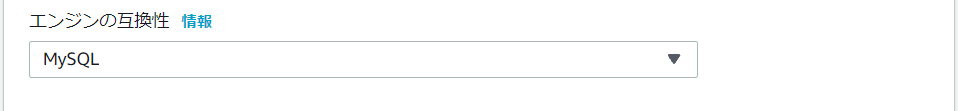
1. RDSのマネージメントコンソール、左ペインから[Proxies]をおします



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



1. [エンジンの互換性]は[MySQL]を選びます



1. [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



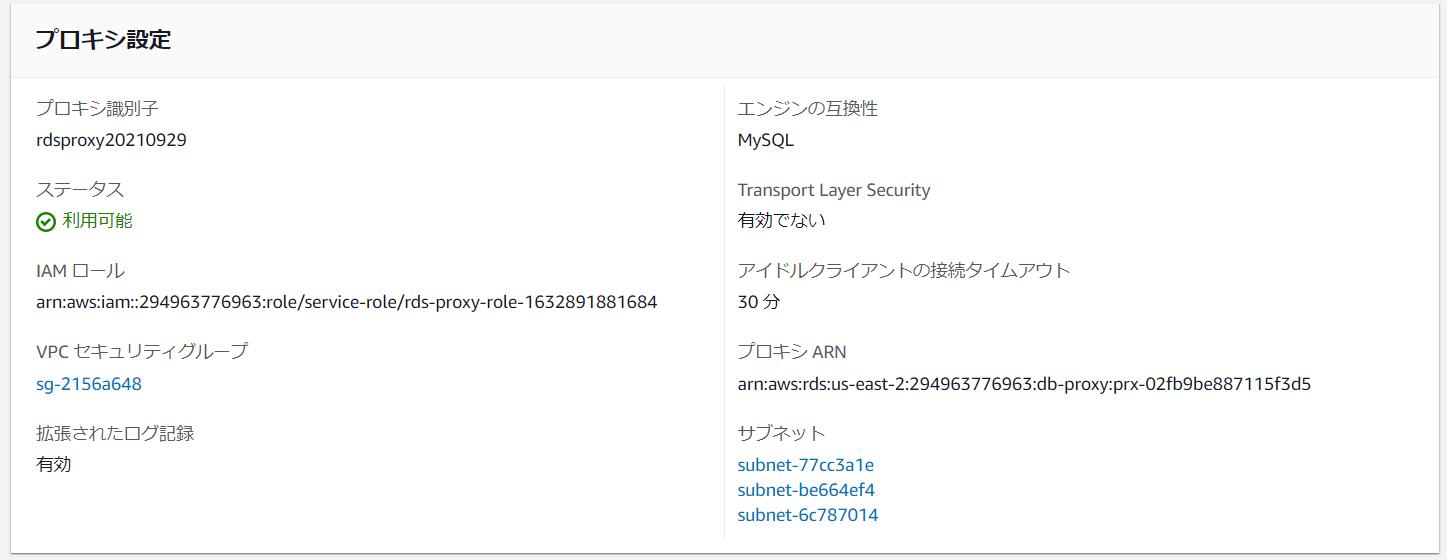
1. [接続]の項目で先程作成したSecrets Managerの値をドロップダウンから選びます



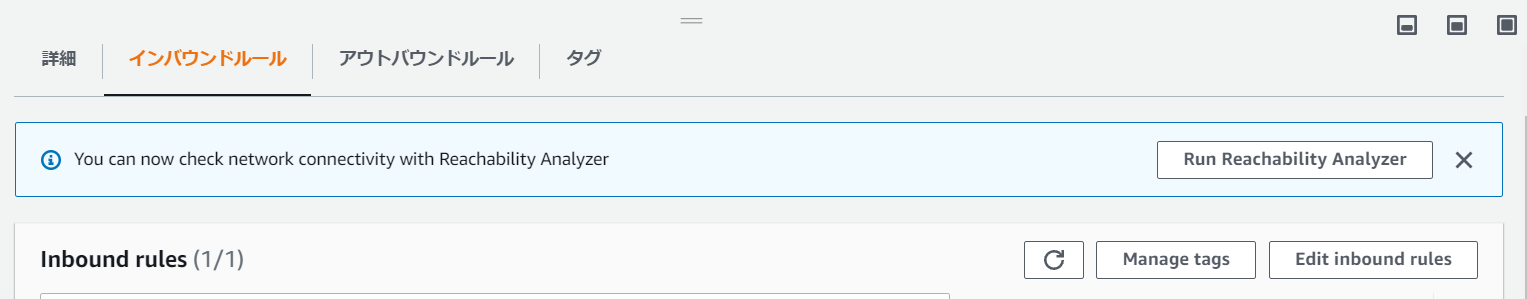
1. [拡張されたログ記録を有効にする]にチェックをつけ、[プロキシを作成]をおします
2. 作成中となりますので、10分程度待ちます



1. 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



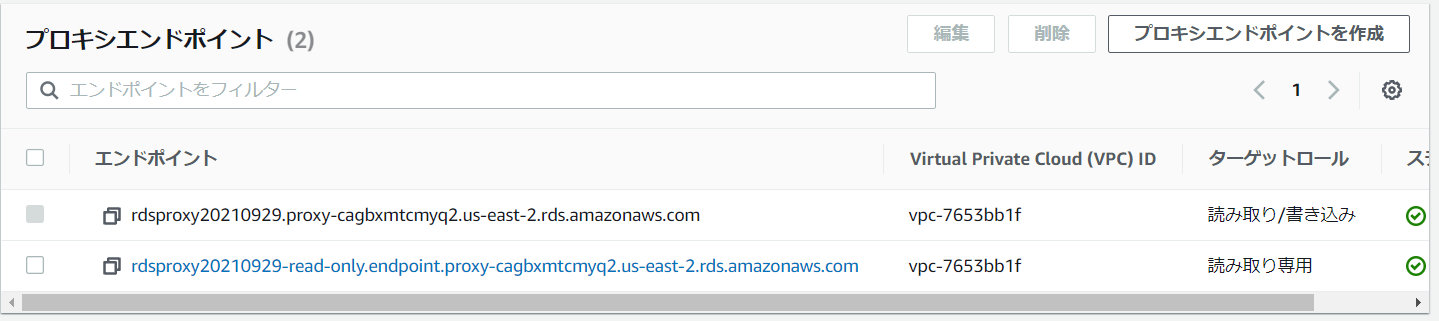
1. [インバウンドルール]のタブから[Edit inbound rules]をおします



1. [削除]をおして今あるルールを全て削除します
2. [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



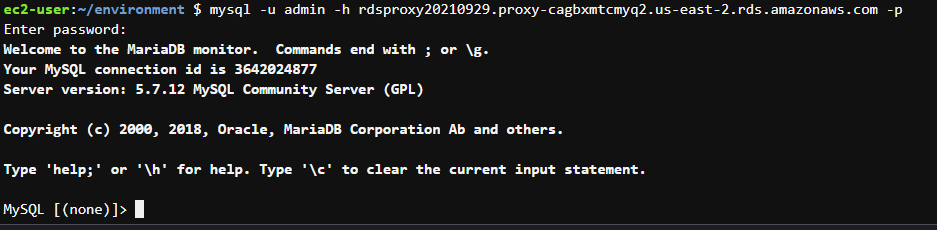
1. 再度RDS Proxyのマネージメントコンソールから、[エンドポイント]をコピーします（読み取り/書き込みの方です）



1. Cloud9から以下のコマンドを実行します

mysql -u admin -h <proxyendpoint> -p

1. 以下が表示されれば完了です



1. 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

1. Lambdaのマネージメントコンソールを開きます
2. [関数の作成]をおします
3. [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



1. [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

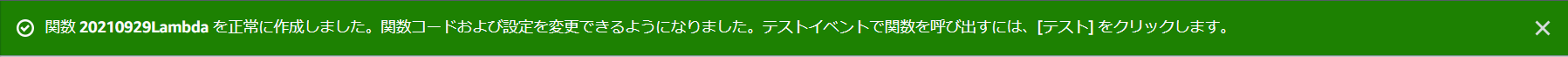


1. セキュリティグループはデフォルトを選んで、以下の通りVPC内部のすべての通信が通るようになっていることを確認してください



1. [関数の作成]をおします
2. 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





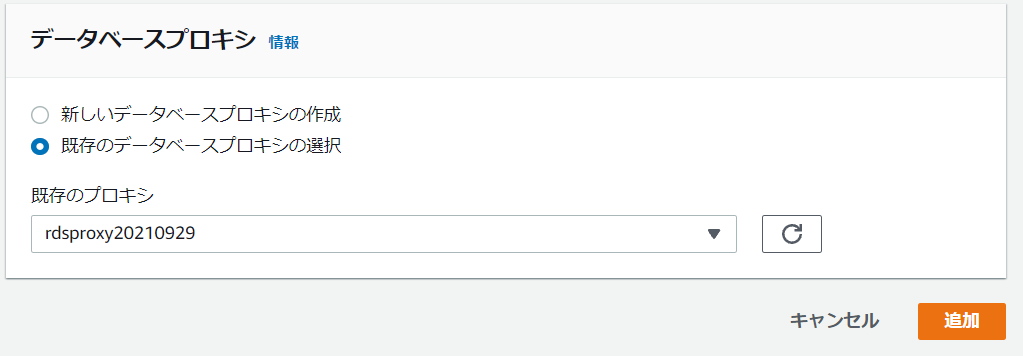
1. 画面中ほどにあるタブで[設定]をクリックし、左ペインから[VPC]をクリックして下さい



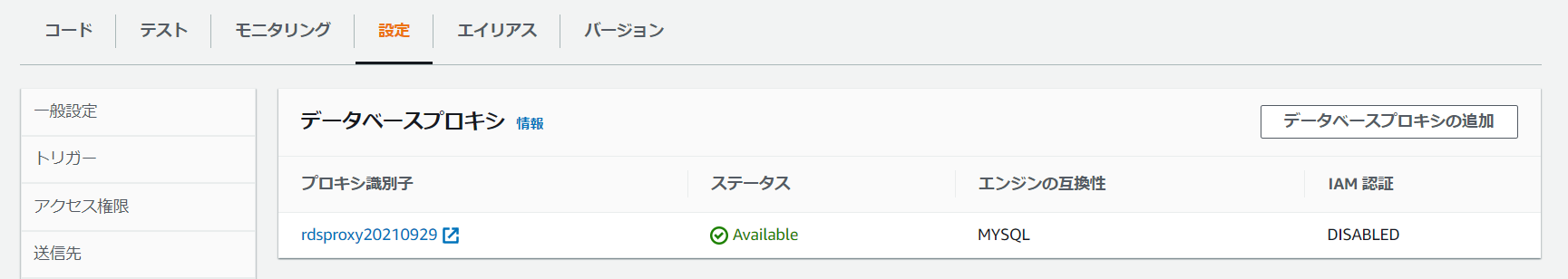
1. 以下の通りVPC内部の通信が通るようになっていることを確認します



1. 左ペインから[データベースプロキシ]をクリックします。
2. [データベースプロキシの追加]をおします
3. [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



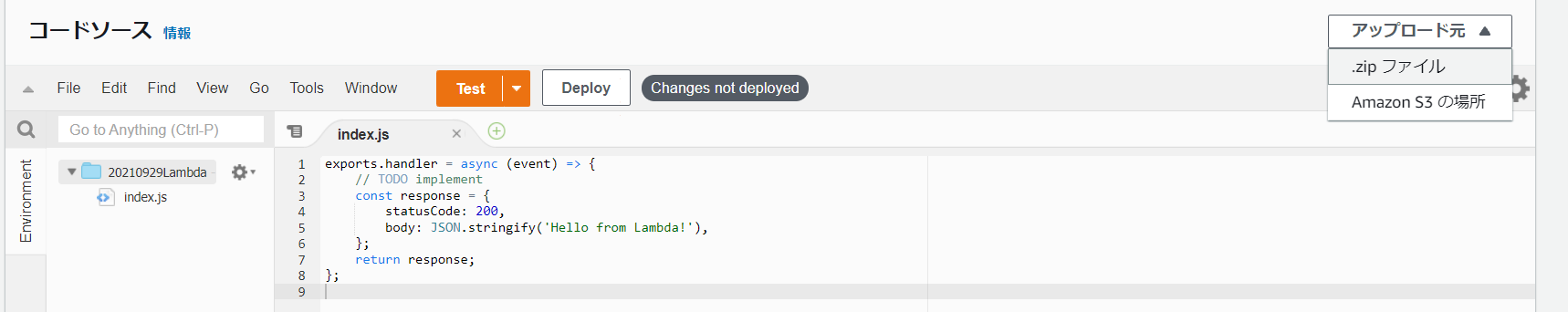
1. 以下のように表示されれば設定が完了です



1. タブで[コード]を選びます



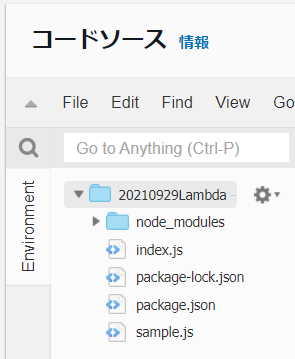
1. [アップロード元]ボタンから[.zipファイル]を選びます



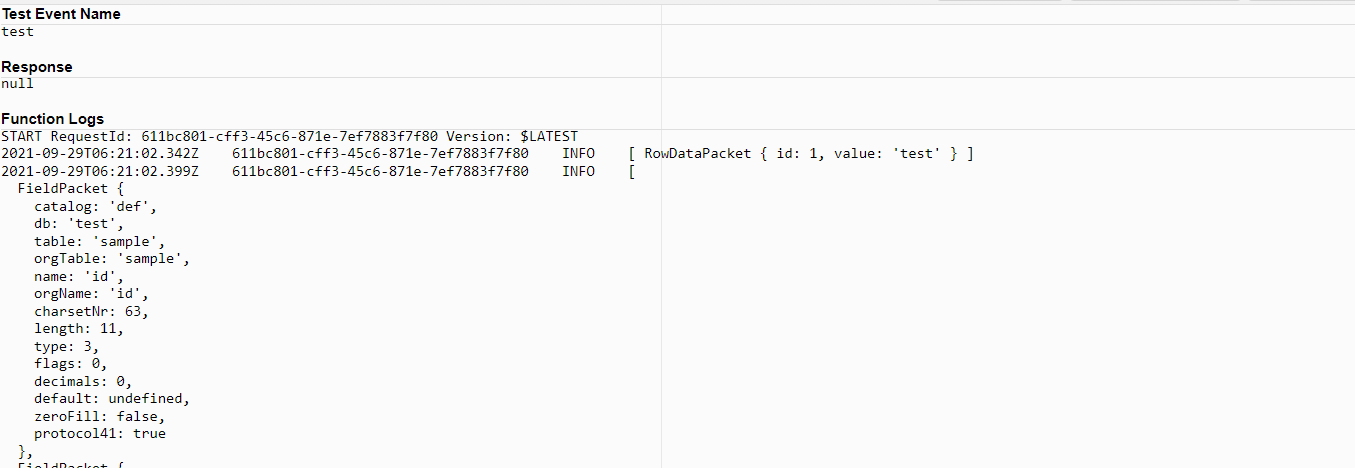
1. Githubにある、[lambdamysql.zip]を適当な個所に保存しアップロードします



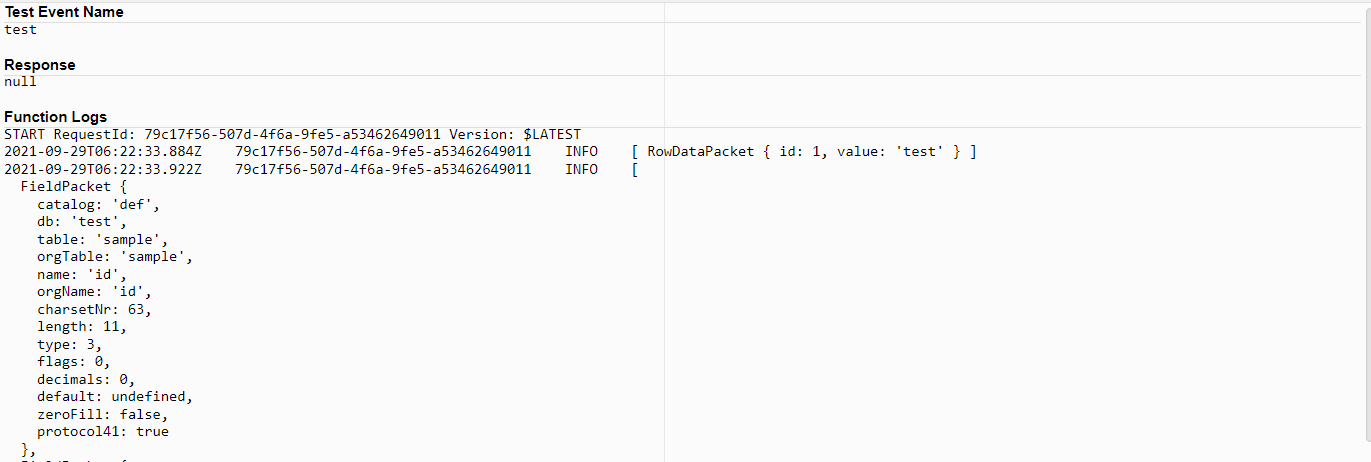
1. 以下のようにパッケージがインポートされます



1. 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
2. [Deploy]をおして、[Test]をおします
3. イベント名に適当な名前を付けて[作成]をおします
4. 再度[Test]をおします
5. 以下の通りSQLの実行結果が出力されたら成功です



1. 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
2. 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）