Amazon Aurora Serverless v2 ワークショップ

2022/06/11

シニアエバンジェリスト 亀田 治伸

はじめに：Aurora Serverless はアプリケーションのニーズに基づいてデータベースの容量を拡大または縮小することのできる Amazon Aurora 用のオンデマンドの自動スケーリング設定です。Aurora PostgreSQL と MySQLそれぞれ以下の環境に対応しています。

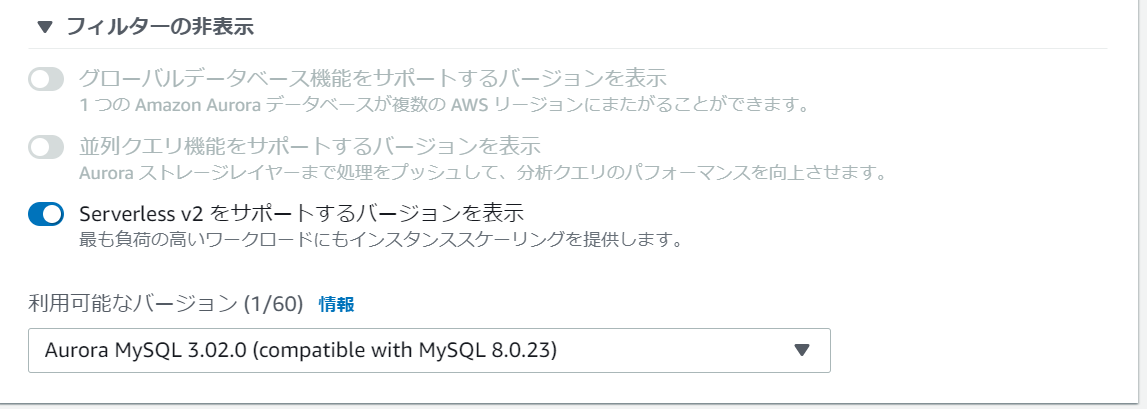
Aurora MySQL 3.02.0 or higher. (MySQL 8.0.)

Aurora PostgreSQL 13.6 or higher.

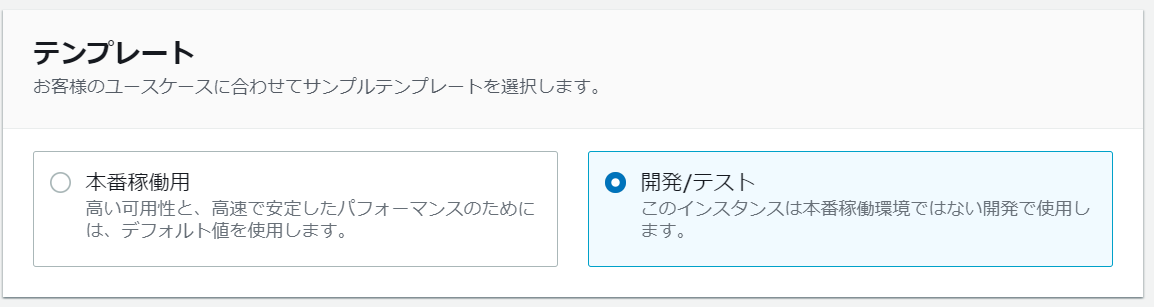
1つのAuroraクラスターの中にインスタンス型とサーバレス型を混在させることができるのが大きな特徴です。

まず1台構成のMySQLクラスターをAurora Serverlessを用いずに起動させます。

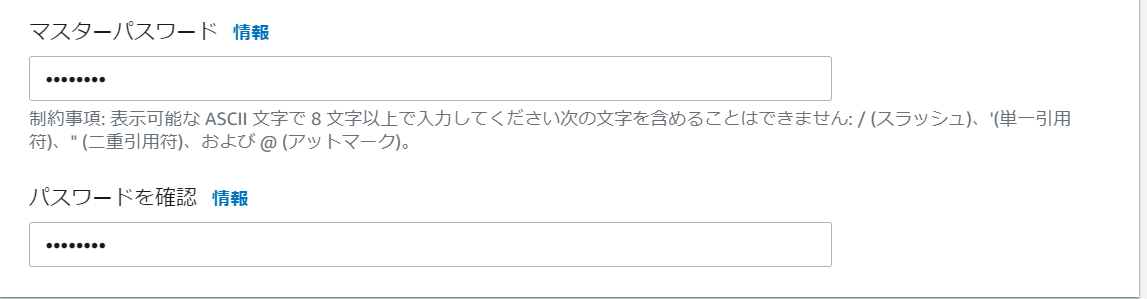
1. RDSのマネージメントコンソールに移動します
2. [データベースの作成]をおします
3. [フィルターの表示]をクリックし、[Serverless v2 をサポートするバージョンを表示]をオンにします



1. 適当なバージョンを選びます(2022/06/01現在、バージョンは1つのみになります)
2. [テンプレート]で[開発/テスト]を選びます



1. 適当なパスワードを入力します

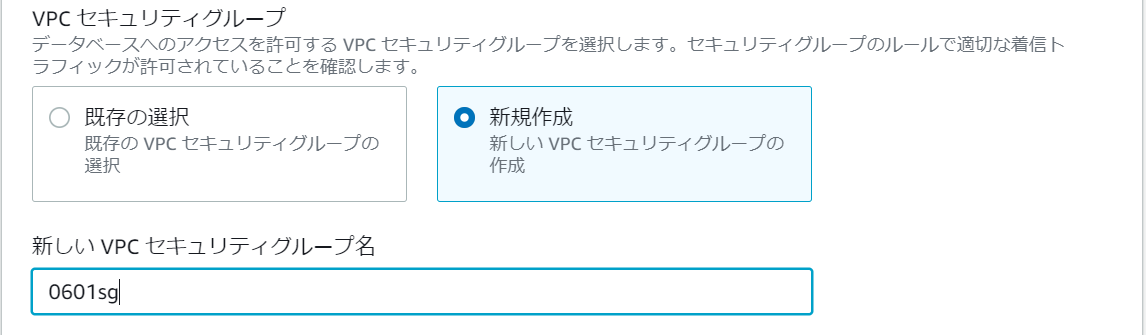


1. [インスタンスの設定]で[バースト可能クラス]を選びます

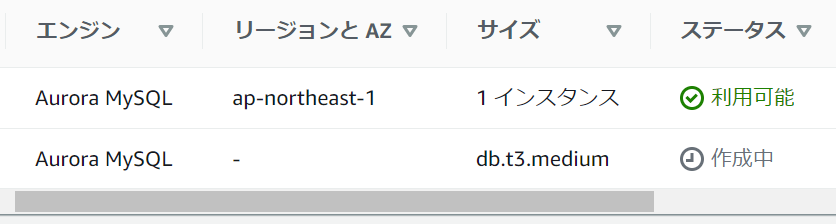


（インスタンスはデフォルトで問題ありません）

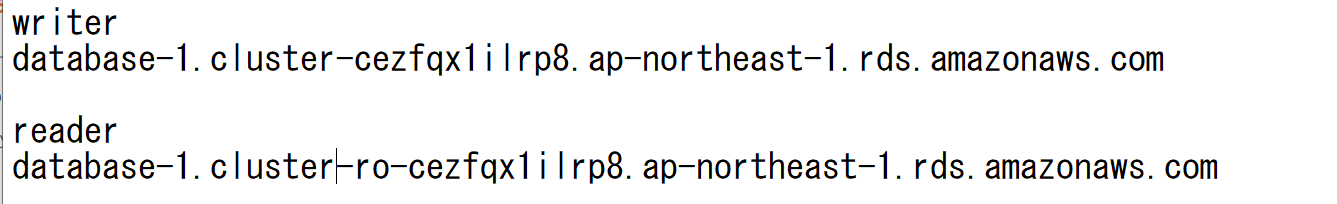
1. VPCはデフォルトVPCをこのハンズオンでは用いますが、別のVPCを選択しても動作します
2. セキュリティグループに適当な名前を付けます。あとで使いますので名前はメモを取っておいてください。



1. [データベースの作成]をおします
2. 起動中の画面に切り替わりますので、数分間まちます。起動されました！と表示してもクラスター全体が起動していないケースがありますので以下のステータスを確認します

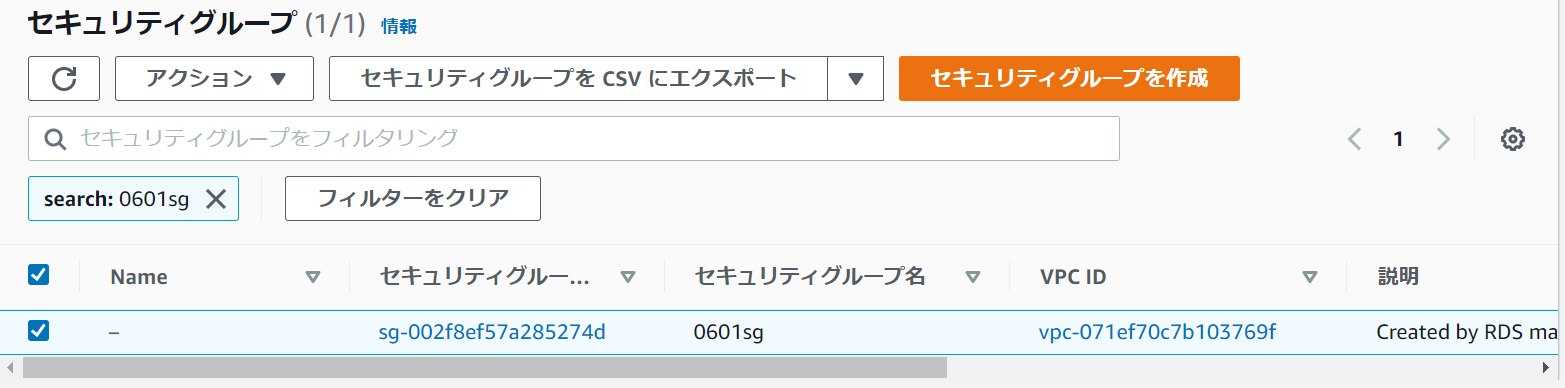


1. 2つとも利用可能になったら[database-1]をクリックし、2つのエンドポイントをメモしておきます

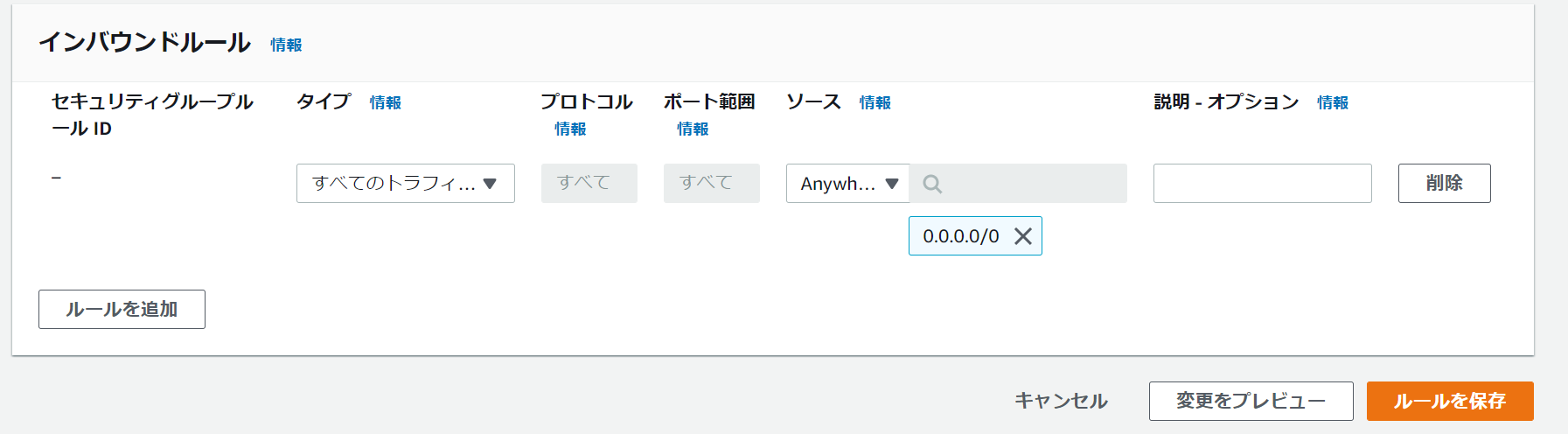


Cloud9を起動し、MySQLへログインし新しいデータベースを作ります

1. Cloud9のマネージメントコンソールを新しいタブで開きます
2. [create environment]をおします
3. 適当な名前を入力し[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。（上記手順でRDSをデフォルトVPC以外に起動した方は、同じVPCを選んでください）
5. [Create environment]をおします
6. しばらく待つとコンソールへのアクセスが可能となります。
7. さらにブラウザの別タブを開き、VPCのマネージメントコンソールへ移動します
8. 左ペインから[セキュリティグループ]をクリックします
9. 先程メモしておいたセキュリティグループ名で検索をかけ、特定しクリックします



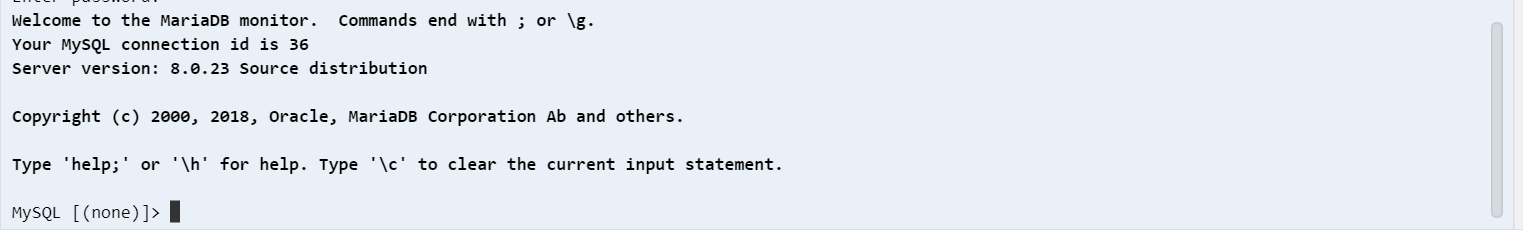
1. 画面下部、インバウンドルールのタブをクリックし、[インバウンドのルールを編集]をおします
2. 今あるルールを削除し、[ルールを追加]を押し以下のように設定します



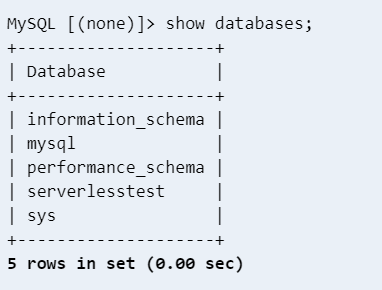
1. [ルールの保存]をおします。これによりVPC内部のすべての通信がRDSへ届くようになったため、Cloud9からの通信が届くようになります。
2. Cloud9のコンソールに戻り以下のコマンドを実行します。<writer endpoint>は先程コピーしたエンドポイントに置換します

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

1. パスワードを入力します。以下の様になれば成功です



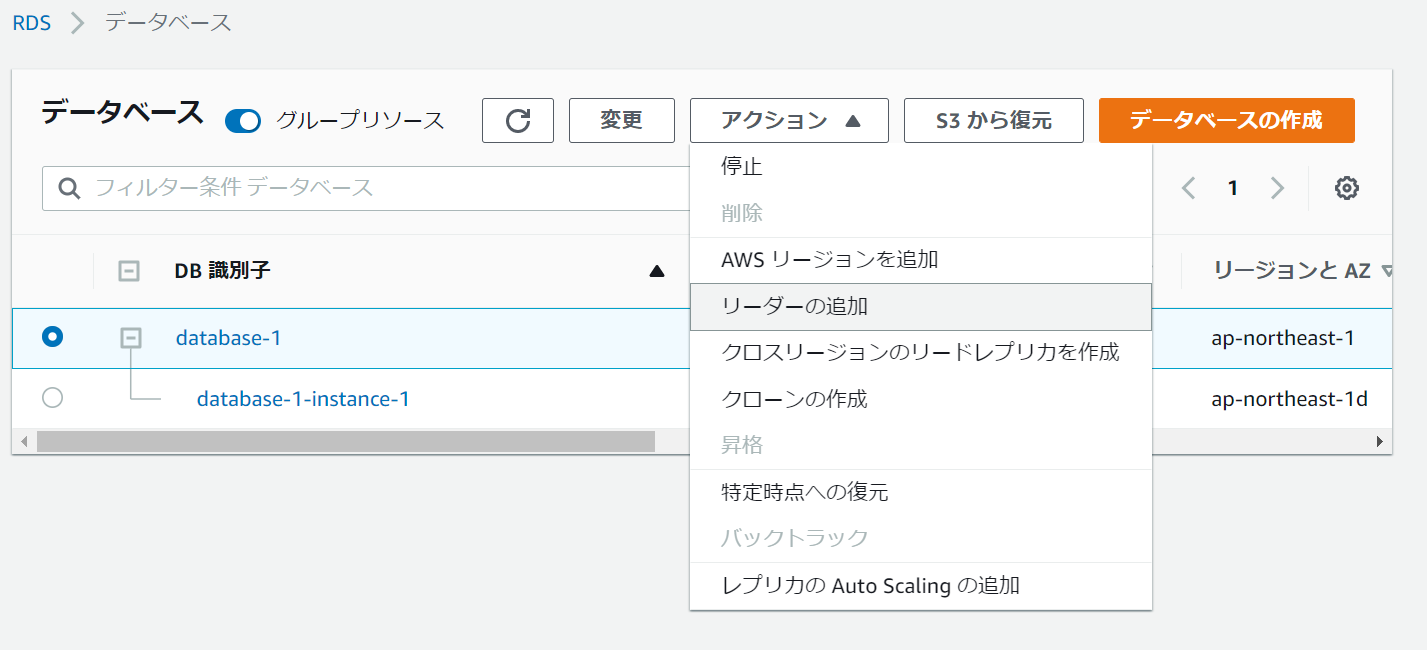
1. [create database serverlesstest;]、[show databases;]を続けて実行します。以下のように新しいデータベースが作成されます



1. 先程と異なるリーダーエンドポイントに同様のコマンドでログインして[show databases;]を実行すると新しいデータベースが見えるようになっています

Aurora Serverless v2のリードレプリカ追加

1. RDSのマネージメントコンソールに戻ります
2. クラスターを選び、[アクション]から[リーダーの追加]を選びます



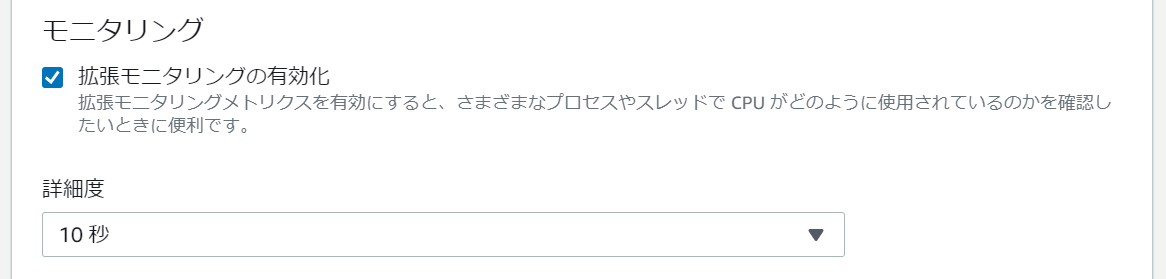
1. [DBインスタンス識別子]に[serverless]と入力します



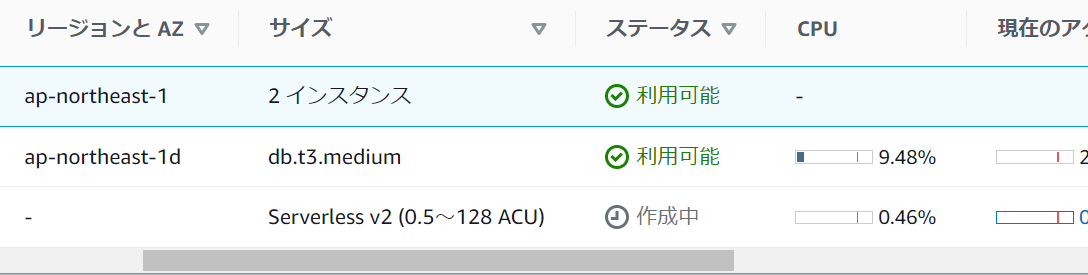
1. [インスタンスの設定]で[Serverless v2]を選びます。ACU (Aurora Capacity Unit)はそのままで大丈夫です



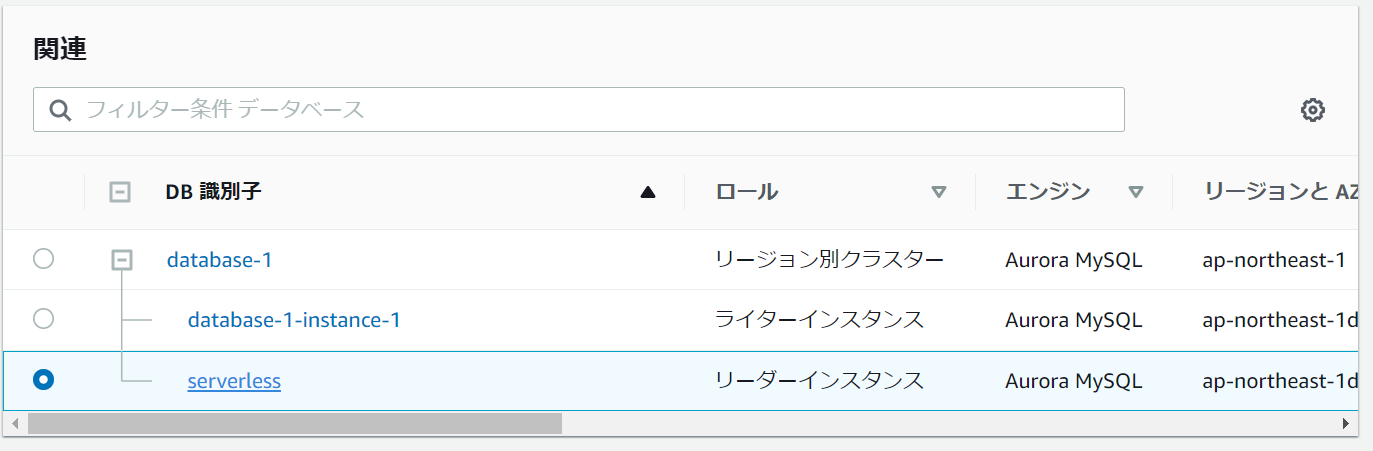
1. 拡張モニタリングを有効化し[詳細度]を10秒に設定します。（Aurora Serverless v2 はv1と異なりインスタンス単位でメトリックスを収集可能です。また拡張モニタリングは微小ながらモニタリングでACU自体を消費することに注意して下さい）



1. [Add reader]を押し、しばらく待ちます。以下のように作成中になります



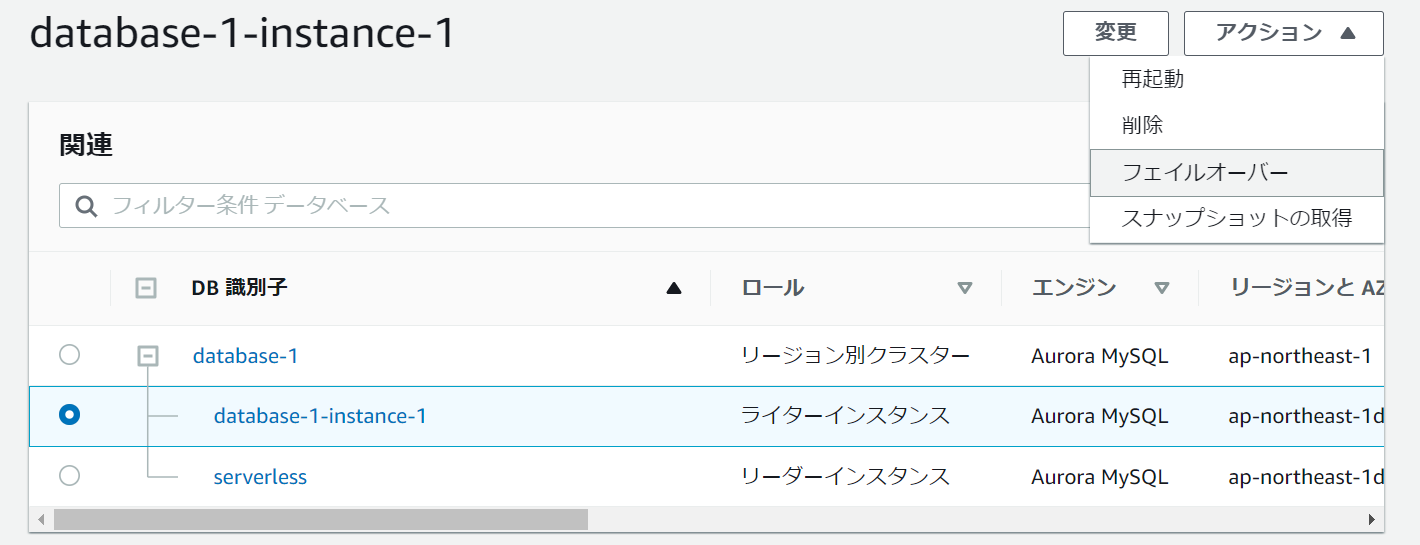
1. 利用可能となった後、再度リーダーにCloud9コンソールからログインし、[show databases;]を実行すると、先程作成したデータベースが表示され、正しくクラスターに参加できていることがわかります。
2. Auroraマネージメントコンソール上でserverlessをクリックします



1. もう一つ別のエンドポイントが存在していることがわかります。これは先程リーダーとして追加したserverless v2 インスタンスの実エンドポイントです。上の手順でログインしたリーダーエンドポイントは、この実エンドポイントに対する仮想エンドポイントです。(CNAME)　この仕組みにより複数のリーダーが存在していたとしても、アプリケーションは常に一つのエンドポイント名でアクセスできるようになっています。先程と同様に実エンドポイントでもログインして[show databases;]を実行してみてください。ログインしいている先は同じなので、同じ結果がでます。

フェイルオーバーによるライターの切り替え

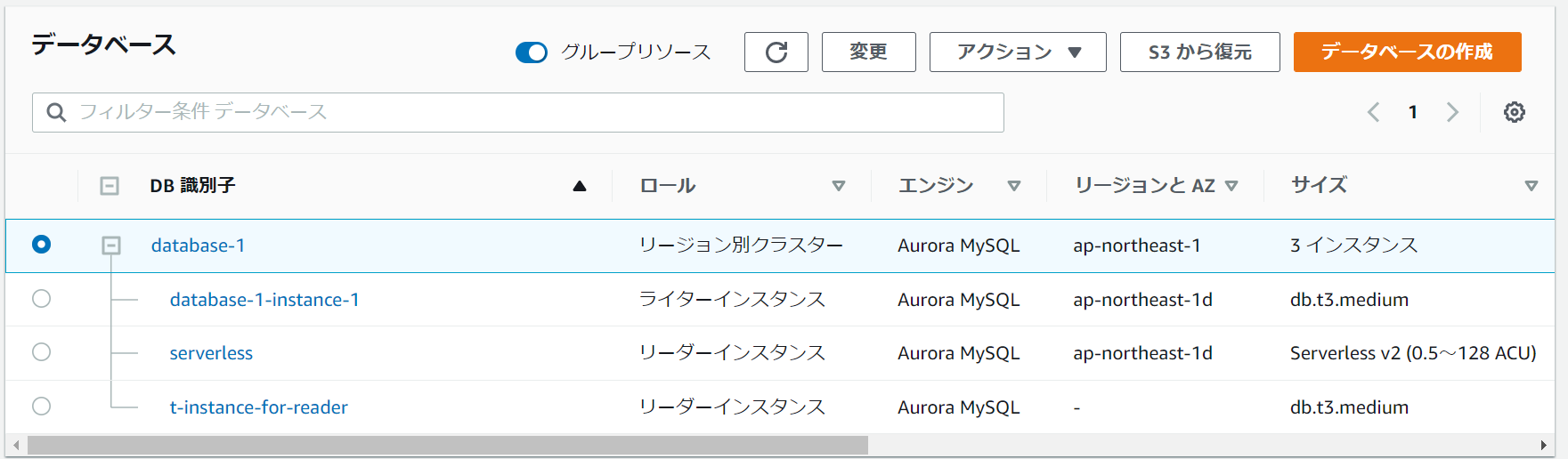
1. RDSのマネージメントコンソールからライターインスタンスを選び、[アクション]から[フェイルオーバー]を選びます



1. 次の画面で[フェールオーバー]のボタンをおします
2. しばらく画面をリロードしながら待つと、以下の通りライターとリーダーが入れ替わっていることがわかります。このようにAurora Serverless v2 は既存インスタンスとの混在で利用することが可能です。



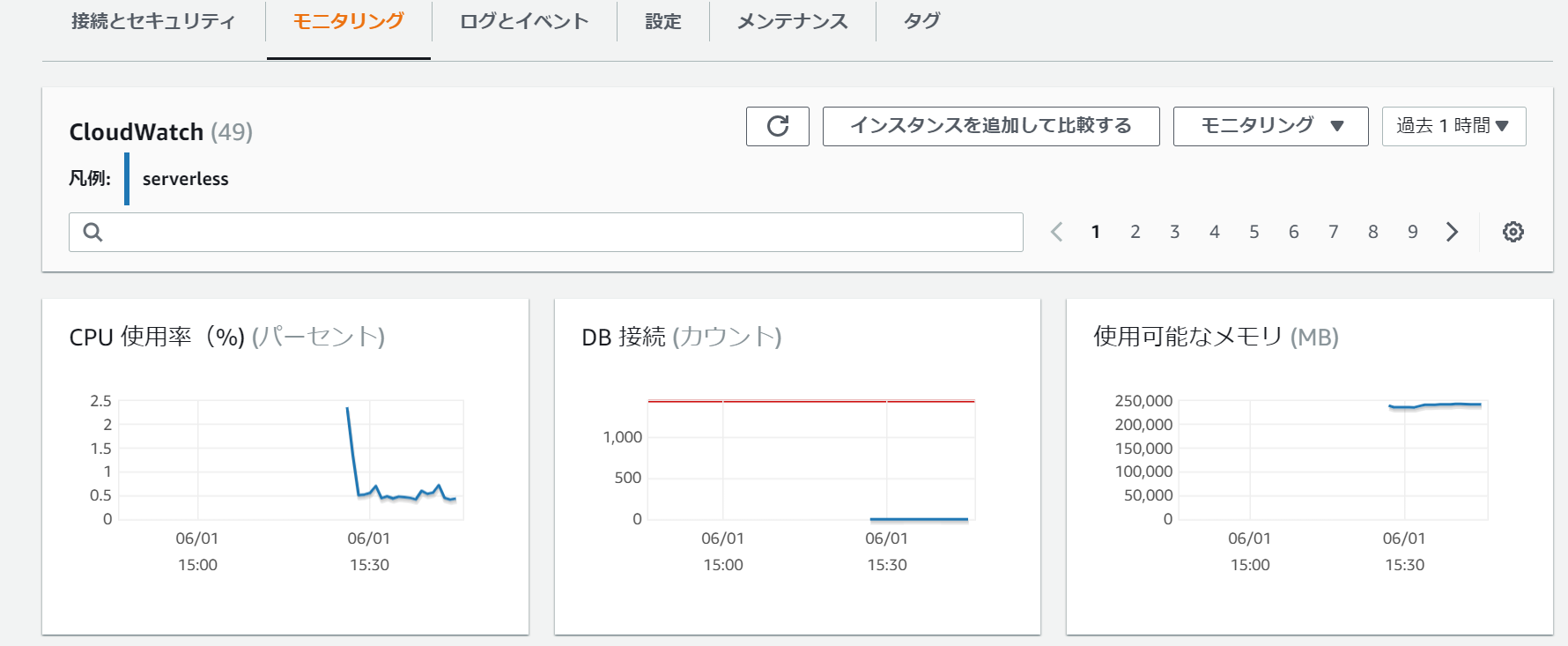
ライターとリーダーで区別するだけではなく以下のようにリーダーのなかで混在させることも可能です。



モニタリング

mysqlslapなどに慣れている方は、再度serverlessをライターに昇格させ、コマンドを実行をして、少しだけDBに負荷をかけてみてください。。

1. Serverlessをクリックして[モニタリング]のタブをクリックすると以下のように様々なグラフが取得可能です

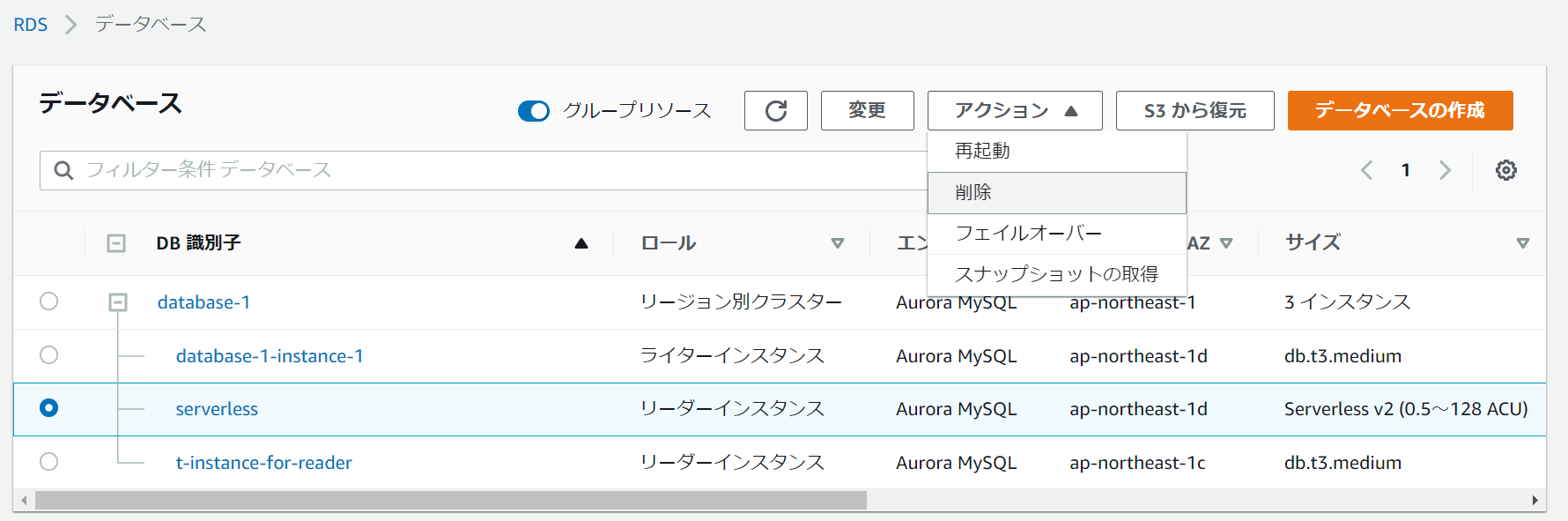


1. 画面右上[モニタリング]をクリックし[拡張モニタリング]を選ぶとさらに様々なメトリックスが取得可能になります。



おつかれさまでした！削除は以下を行ってください

・リーダーの削除



・ライターの削除

・Cloud9の削除

・サブネットグループ（RDS）

・CloudWatchログ