Amazon MSK ワークショップ

2022/03/16

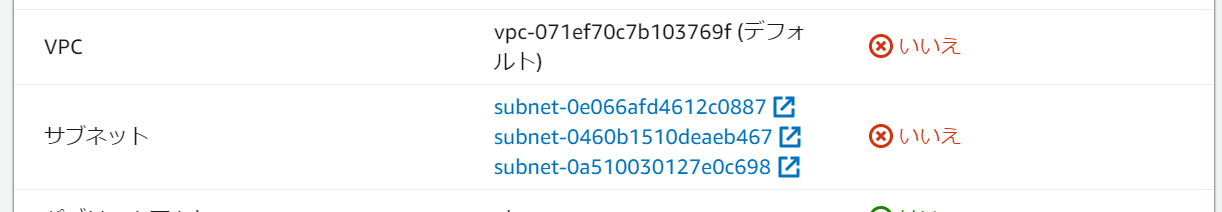
シニアエバンジェリスト 亀田

[MSK クラスターの作成]

MSKはKinesisと異なり、各AZごとにクラスターを作成する必要があります。クラスターのAZ配置は自動化されますが、起動時に対応するAZを指定する必要があります。

東京やノースバージニアなど古いAZを保有するAWSアカウントではエラーになる可能性がありますので、別のRegionを使ってさい。

1. MSKのマネージメントコンソールにアクセスします
2. [クラスターの作成]をおします
3. クラスター名に[MSKTutorialCluster]と入力します
4. クイック作成では以下の通りデフォルトVPCと3つのSubnetが設定されています。サブネットとセキュリティグループの値をコピーしメモしておきます。その後[クラスターの作成]をおします

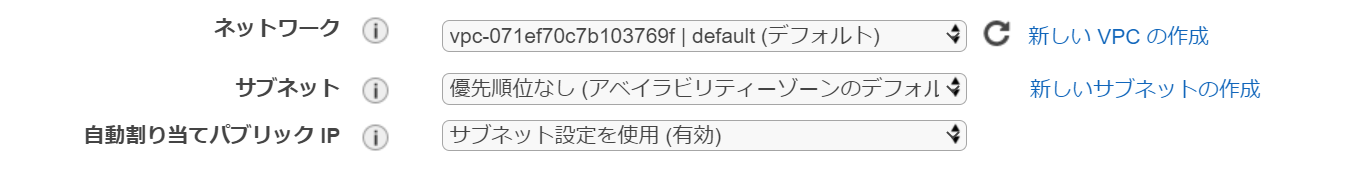


デフォルトVPC以外を指定する場合は（デフォルトVPCが存在しない場合は）、カスタム作成を行うか、リージョンを変更し作業を行ってください。

[クライアントの作成]

クラスターの作成は30分程度かかりますので、その間にクライアントを作成します。MSKクライアントはKinesisと同様にデータを送出するProducer、データを受け取るConsumerというコンセプトがあります。このハンズオンでは、1台のEC2で両方を行いますが、商用環境では勿論、ProducerとConsumerは別々の存在になります。また、LambdaはイベントソースとしてMSKに対応していますので、Producerがメッセージを送出したイベントをもとに関数を起動することが可能です。

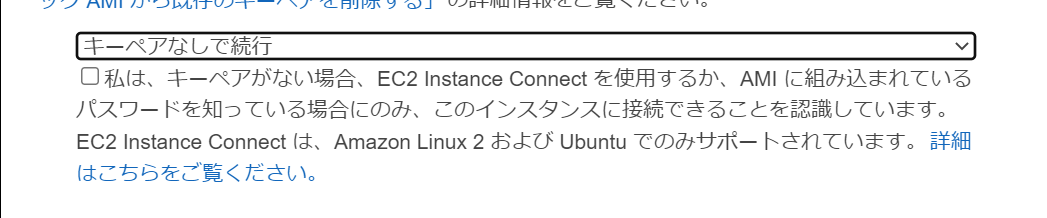
1. EC2のマネージメントコンソールにアクセスします。（ブラウザの別タブを開きます）
2. [インスタンスを起動]をおします
3. AMIは” Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type”を選択します
4. インスタンスタイプは”t2.xlarge”を選び、[次のステップ]をおします
5. ネットワークから先程MSKクラスターを作成したVPCを選択し、パブリックＩＰ割り当てが有効になっていることを確認します。



1. [次のステップ]をおします
2. ストレージのサイズに30を指定します



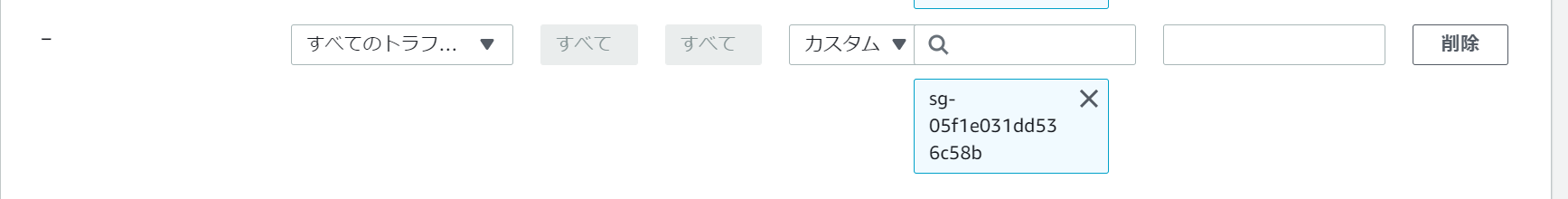
1. [確認と作成]をおし、次の画面で[起動]をおします
2. キーペアは”なし”を選んで、[インスタンスの作成]をおします



1. インスタンスが起動した後、インスタンス名をクリックし詳細画面へ移動し、”セキュリティ”タブでセキュリティグループをコピーしメモしておきます。



1. 左ペインから“セキュリティグループ”をクリックし、先程**MSKクラスター起動時にコピーした**セキュリティグループを特定しクリックします。(EC2用ではないので注意してください)
2. [インバウンドルールの編集]をおします。MSKクラスターに対して入ってくる通信（ProducerであるEC2が出す通信=インバウンド通信）を受け入れる設定を行います。
3. [ルールを追加]をおします
4. EC2のクライアントに設定されているセキュリティグループから出る全てのトラフィックを受け付けるように設定します。以下の画面を参考にしてください。



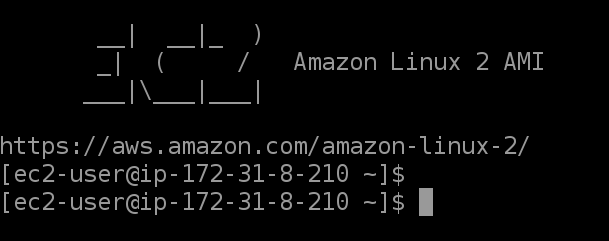
1. [ルールを保存]をおします
2. EC2マネージメントコンソールから、先程作成したEC2にチェックを付けます。（インスタンス名はクリックしません）



1. [接続]ボタンをおします
2. [EC2 Instance Connect]のタブを選んで[接続]をおします



1. 以下のようにコンソールが起動します。

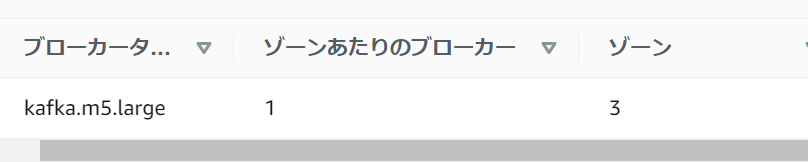


1. (Chromeの場合)ブラウザのタブを右クリックし、グループをProducerとして作成しておきます。



1. [sudo yum install java-1.8.0] を実行します。途中確認が求められますので”y”を入力してください
2. [wget https://archive.apache.org/dist/kafka/2.6.2/kafka\_2.12-2.6.2.tgz]を実行し、Apache Kafkaクライアントライブラリを入手します
3. [tar -xzf kafka\_2.12-2.6.2.tgz]を実行し、ダウンロードしたライブラリを解凍します
4. 事前準備はこれで完了です。MSKクラスター作成が完了するまでまちます。

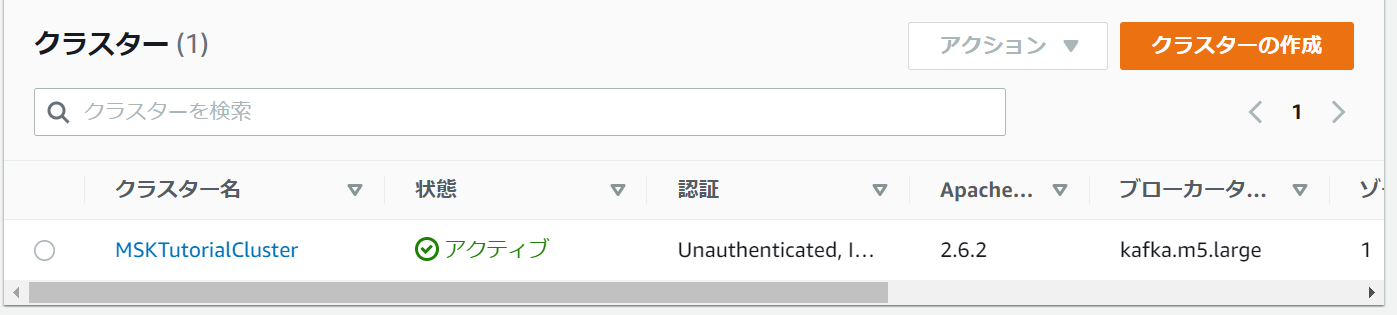
[MSKの構成]



起動時に指定した3つのAZに対してそれぞれ1個づつブローカーと言われるメッセージを取り扱うノードが作成されています。アプリケーションの構成に応じてこれらをスケールさせていく必要があります。本ハンズオンでは割愛しますが興味のある方は以下に挑戦してみてください。

<https://catalog.us-east-1.prod.workshops.aws/workshops/c2b72b6f-666b-4596-b8bc-bafa5dcca741/en-US>

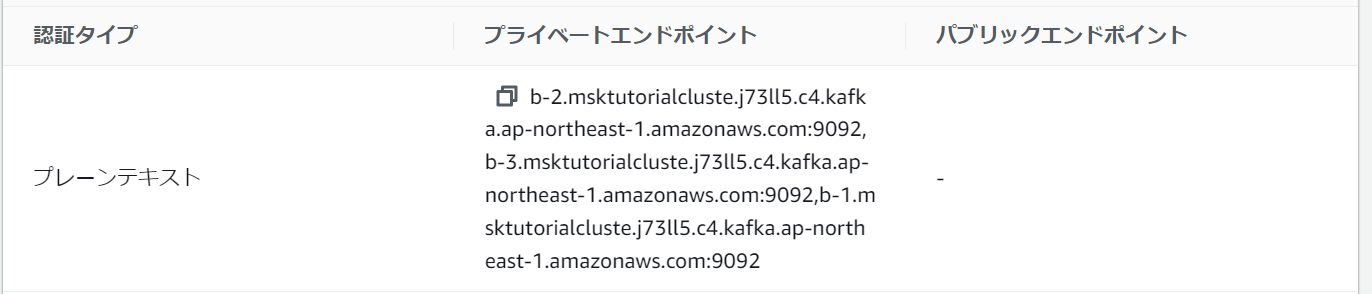
以下の状態になればMSKクラスターの作成が完了です

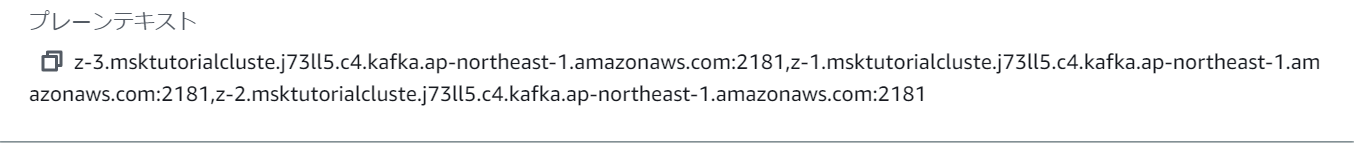


[メッセージのテスト]

ここから、先程設定したEC2のProducerでメッセージを投入してみます

1. MSKクラスターのクリックし、[クライアント情報の表示]をおし、以下2つの文字列をコピーしメモしておきます。非常に長い文字列ですが、改行が入らないことに注意してください





1. EC2上で、kafka\_2.12-2.6.2 ディレクトリに移動します
2. [bin/kafka-topics.sh --create --zookeeper **ZookeeperConnectString** --replication-factor 3 --partitions 1 --topic MSKTutorialTopic]

をコピーして実行します。その際”ZookeeperConnectString”を先程コピーしたものに置換してください

1. “Created topic MSKTutorialTopic.”と表示されれば成功です。このトピックとはSNSと同じ概念です。複数のメッセージ種別を取り扱う際などのメッセージパイプライン識別子になります。トピックの詳細に興味がある方は”aws pub sub”などで検索してみてください。疎結合モデルの非常に重要なアーキテクチャ設計指針の1つです。
2. [cd bin]を実行し移動します
3. [vi client.properties]を実行します
4. [security.protocol=PLAINTEXT]を1行名にペーストします。
5. 以下の順番にキーを入力します(ファイルを保存します。viが使える方は気にせず保存してください)

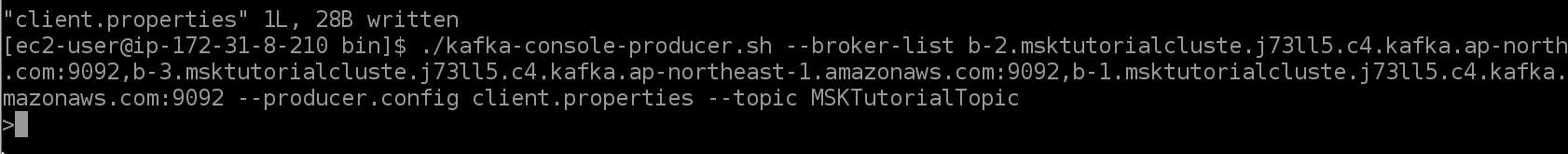
[ESC][:] [w] [q]

1. [./kafka-console-producer.sh --broker-list **BootstrapBrokerString** --producer.config client.properties --topic MSKTutorialTopic]

を実行します。

BootstrapBrokerString は先ほどコピーした“認証タイプ：プレーンテキスト”の文字列置換します。

1. 以下のように、 “>” が表示されたら接続ができていますので、メッセージを適当に3行ほど入力します。



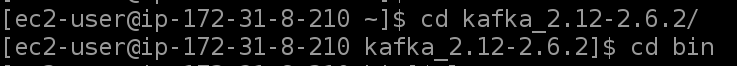
[Consumerの起動]

今までの手順でブローカーにはメッセージが挿入されています。今度はこれをConsumerとして呼び出します。

1. 同じEC2インスタンスtestに対してもう1回Instance Connectを実行し、ブラウザのタブ機能でわかりやすい名前を付けておきます。



1. 以下のディレクトリに移動します

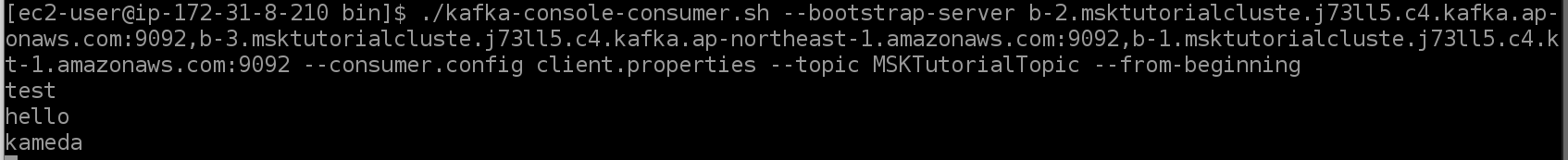


1. [./kafka-console-consumer.sh --bootstrap-server **BootstrapBrokerStringTls** --consumer.config client.properties --topic MSKTutorialTopic --from-beginning]

を実行します。

BootstrapBrokerStringTlsの値は先程と同じく認証：プレインテキストとしてコピーした文字列に置換します（tlsと書いてあるが気にしなくて大丈夫です）

1. 以下のようにメッセージが出てきたら成功です！



1. 更なるシナリオは以下にありますので興味がある方は挑戦してみて下さい

<https://catalog.us-east-1.prod.workshops.aws/v2/workshops/c2b72b6f-666b-4596-b8bc-bafa5dcca741/>

おつかれさまでした！

以下を削除してください

・MSKクラスター

・EC2