記事
Toshihiko Minamoto · 2021年1月26日 7m read

InterSystems IRIS における AWS Glue の使用について

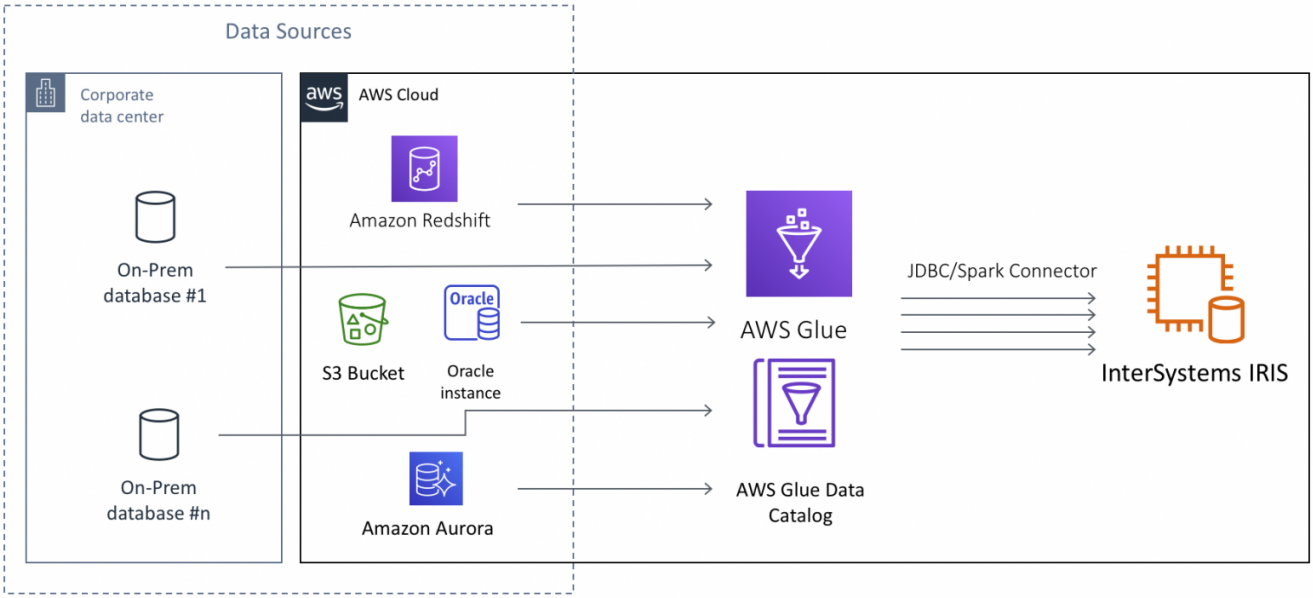
2019年 10月 17日

Anton Umnikov

InterSystems シニアクラウドソリューションアーキテクト
AWS CSAA、GCP CACE

AWS Glue は、完全に管理された ETL (抽出、変換、読み込み) サービスです。データの分類、クリーンアップ、強化、そして様々なデータストア間でデータを確実に移動させるという作業を簡単にかつコスト効率の良いかたちで行えるようにするものです。

InterSystems IRIS の場合、AWS Glue
を使用すれば、大規模なデータをクラウドとオンプレミスのデータソースの両方から IRIS
に移動させることができます。
ここで考えられるデータソースは、オンプレミスのデータベース、CSV、JSON、S3 バケットに保管されている Parquet ファイルならびに Avro ファイル、AWS Redshift や Aurora
といったクラウドネイティブのデータベースを含みますが、これらに限定されません。



本記事では、読者の皆さんが、AWS Glue について少なくとも AWS Glue の入門チュートリアル
を完了している程度の基本的な知識をお持ちであるという前提で話を進めていきます。 InterSystems IRIS
をデータターゲット、すなわち「データシンク」として使用するかたちで AWS Glue Jobs
を設定する際の技術的な側面に着目します。

画像ソース https://docs.aws.amazon.com/glue/latest/dg/components-key-concepts.html

AWS Glue Jobs は「サーバーレス」で実行されます。
ジョブの実行に必要なリソースはすべて、ジョブが実際に実行されている間だけ AWS により動的にプロビジョニングされ、ジョブが完了すると同時に破棄されます。つまり、必要なインフラストラクチャにかかる継続的なコストをプロビジョニングしたり、管理したり、負担したりすることが不要であり、ジョブが実際に実行されている間

だけ請求が発生するため、ジョブのコードを書くことだけに労力を費やすことができます。
「アイドル」時間中は、S3
バケット、ジョブコードの保管、および設定に使用される以外のリソースが消費されることはありません。

通常、Glue Job は動的にプロビジョニングされる Apache Spark で実行され、また PySpark
コードで記述されますが、選択肢は他にも複数存在します。 Glue Job は、データがデータソースから抽出される
the "Extract" 、Glue API を使ってビルドされる一連の *"Transformations"*
、そして最後に最終変換データがターゲットシステムに書き込まれる *"Load"* または *「シンク」*
の各部分で構成されます。

AWS Glue が IRIS と対話できるようにするには、以下を確認する必要があります。

- Glue から関連する IRIS インスタンスへのネットワークアクセスが確立されている
- Glue Job は IRIS JDBC ドライバーの JAR ファイルにアクセスできる
- Glue Job は InterSystems IRIS JDBC と互換性のある API を使用している

それでは、必要な各ステップを詳しく見ていきましょう。

IRIS との接続を確立する

AWS コンソールで、AWS Glue->Connections->Add Connection、と順に選択します。

接続名を入力し、Connection Type に「JDBC」を選択します。

JDBC の URL 欄に、IRIS インスタンスの JDBC 接続文字列、ユーザー名、パスワードを入力します。

次のステップは非常に大切で、**_Glue がそのエンドポイントを IRIS インスタンスと同じ VPC に配置することを確認する必要があります。**IRIS インスタンスの VPC と Subnet を選択します。 すべての TCP
ポートにおいて自己参照する受信規則が設定されたセキュリティグループであれば、どれでも使用できます。
(IRIS インスタンスのセキュリティグループなど)

JDBC ドライバーへのアクセス権を持つ IAM ロール

まだ行っていない場合は、S3 バケットに IRIS JDBC ドライバーの JAR ファイル「intersystems-jdbc-3.0.0.jar」をアップロードしてください。 この例では、s3://irisdistr バケットを使用しています。
アカウントによって異なると思います。

そのファイルにアクセスできる (Glue Job の) IAM ロールを Glue
がスクリプトやログなどを保管するのに使用する別の S3 バケットと一緒に作成する必要があります。

それが JDBC ドライバーのバケットに対して読み取りアクセスを持っていることを確認してください。
今回は、このロール (GlueJobRole) と事前定義されている AWSGlueServiceRole
にすべてのバケットへの読み取り専用アクセスを与えています。
このロールのアクセス権限はさらに制限することができます。

Glue Job の作成と設定

新規 Glue Job を作成します。 先ほどのステップで作成した IAM ロールを選択します。
それ以外はすべてデフォルトのままにします。

「Security configuration, script libraries, and job parameters (optional)」で、「Dependent jars path」を S3
バケット内にある intersystems-jdbc-3.0.0.jar の場所に設定します。

ソースには、既存のデータソースの 1 つを使用します。
先ほど触れたチュートリアルの内容を実行された方は、少なくとも 1 つはお持ちのはずです。

「Create tables in your data target」オプションを使い、前のステップで作成した IRIS への接続を選択します。
それ以外はすべてデフォルトのままにします。

ここまでの手順を正しく行っていれば、下のような画面が表示されるはずです。

もう少しです！ IRIS にデータを読み込むには、スクリプトに小さな変更を 1 つだけ加える必要があります。

スクリプトの調整

Glue が生成したスクリプトには、Spark を対象とした AWS の専用拡張機能 AWS Glue Dynamic Frame が使用されています。 ETL ジョブにいくつかのメリットがあるほか、AWS がマネージドサービスオファリングを提供していないデータベースにはデータが書き込まれないようにしてくれます。

ここで良いお知らせです。データベースにデータを書き込む時点では、「ダ–ディ」なデータに対してスキーマを強制しないなど、Dynamic Dataframe が提供する利点はすべて不要となりました
(データを書き込む段階でデータは「クリーン」と判断されるため)。また、Dynamic Dataframe を AWS に管理されるターゲットだけでなく IRIS にも対応できる Spark のネイティブ Dataframe に変換するという作業も簡単に行えます。

変更が必要な行は、上の画像で示す40行目です。 最終行の 1 つ前の行です。

以下のように変更する必要があります。

```
#datasink4 = glueContext.write_dynamic_frame.from_jdbc_conf(frame = dropnullfields3,
catalog_connection = "IRIS1", connection_options = {"dbtable": "elb_logs", "database"
: "USER"}, transformation_ctx = "datasink4")
dropnullfields3.toDF().write \
    .format("jdbc") \
    .option("url", "jdbc:IRIS://172.30.0.196:51773/USER/") \
    .option("dbtable", "orders") \
    .option("user", irisUsername) \
    .option("password", irisPassword) \
    .option("isolationlevel","NONE") \
    .save()
```

irisUsername と irisPassword には、IRIS JDBC への接続に使うユーザー名とパスワードが入ります。

注意: *パスワードは絶対にソースコードに保管してはいけません！！ AWS Secrets Manager などのツールを使用することをおすすめしますが、本記事でセキュリティの詳細をそこまで深く掘り下げるのは、割愛させていただきます。 AWS Glue における AWS Secrets Manager の使用法については、こちらの記事をおすすめします。*

「Run Job」ボタンをクリックした後は、AWS Glue が ETL を実行してくれるのでリラックスしてお待ちください。

まあ最初はエラーがいくつか出ると思いますが... 珍しいことではないと思います。
入力ミスがあったり、セキュリティグループのポートが間違っていたり... AWS Glue は CloudWatch を使って、すべての実行データやエラーログを保存します。 問題の原因については、ロググループ */aws-glue/jobs/error* と */aws-glue/jobs/output* を参照してください。

クラウドで ETL をお楽しみください！

-Anton

#AWS #Python #SQL #クラウド #データベース #ビッグデータ #InterSystems IRIS

ソースURL:https://jp.community.intersystems.com/post/intersystems-iris-%E3%81%AB%E3%81%8A%E3%81%91%E3%82%8B-aws-glue-%E3%81%AE%E4%BD%BF%E7%94%A8%E3%81%AB%E3%81%A4%E3%81%84%E3%81%A6