Jupyter Notebook to query Database (Readonly)

Overview

Create an AWS SageMaker notebook instance that has read only access to the RDS Aurora Postgres cluster.

Data scientists can create notebooks that connect to the database via IAM user with read only access to the database.

Content

SageMaker notebook instance component:

repository: https://gitlab.com/mc-components/terraform-aws-sage-notebook

sage.tf : creates a SageMaker notebook instance in the same VPC subnet as the RDS and with the same security group as the RDS

iam.tf : defines the IAM policy on the SageMaker notebook instance. Defines SageMaker policies, CloudWatch access, logs access, and access to RDS.

variables.tf: defines input variables for sage

Blueprint that calls SageMaker components:

repository: https://gitlab.com/mc-estate-infra/min-au/paspo/paspo-sage-notebook

Accessing SageMaker notebook instance:

To allow develops access to the SageMaker notebook instance, developers need the following level of IAM policies:

```
statement {
  sid    = "AllowAccessToSageMaker"
  effect = "Allow"

  actions = [
    "sagemaker:CreatePresignedNotebookInstanceUrl",
    "sagemaker:DescribeNotebookInstance",
    "sagemaker:DescribeNotebookInstanceLifecycleConfig",
    "sagemaker:ListNotebookInstances",
    "sagemaker:ListNotebookInstanceLifecycleConfigs",
    "sagemaker:StartNotebookInstance",
    "sagemaker:StopNotebookInstance",
    "sagemaker:UpdateNotebookInstance"
  ]

  resources = ["*"]
}
```

Then the developers navigate to the AWS account -> SageMaker -> SageMaker notebook instance -> open in JupyterHub.

Developers create their own notebook files with the naming convention: [NAME]_Notebook.ipynb

Connect to the database with a DB user and generated authorization token.

```
import psycopg2
import boto3

client = boto3.client('rds')

token = client.generate_db_auth_token(DBHostname='[hostname].[clustername].us-east-1.rds.amazonaws.com',
                                      Port=5432,DBUsername='[username]',Region='us-east-1')

try:

    # CONNECTION INFO


    connection = psycopg2.connect(user = "[username]",
                                  password = token,
                                  host = "[hostname].[clustername].us-east-1.rds.amazonaws.com",
                                  port = "5432",
                                  database = "[dbname]")

    cursor = connection.cursor()
    # Print PostgreSQL Connection properties
    print ("DB Connection Info: ", connection.get_dsn_parameters(),"\n")

    # Print PostgreSQL version
    cursor.execute("SELECT version();")



    # *************** BEGIN EXECUTING READ ONLY COMMANDS HERE ***************************

    cursor.execute("SELECT * FROM [DB_NAME].[DB_TABLE_NAME] LIMIT 1")
```

References

Connect to DB with DB user: https://aws.amazon.com/premiumsupport/knowledge-center/users-connect-rds-iam/

Sync GH with Sage: https://docs.aws.amazon.com/sagemaker/latest/dg/nbi-git-repo.html#nbi-git-resource