AWS POC 4 - Load Government Debt Total % GDP via a custom Python Job from a local host

Overview

Process Flow

1. Push the formatted CSV file to the staging S3 bucket
2. Run the custom AWS Glue Job (Python)
3. Save the result to the World Bank Data S3 bucket
4. Create an AWS Glue Table over the S3 Bucket and query in Athena

Glue Job Python Code

```python
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ['JOB_NAME'])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
print ("The args: ", args)


# Script generated for node AWS Glue Data Catalog
AWSGlueDataCatalog_node1708237359808 = glueContext.create_dynamic_frame.from_catalog(database="aws-poc2-db", tab

# Script generated for node Remove Prior to 1990
RemovePriorto1990_node1708238527332 = ApplyMapping.apply(frame=AWSGlueDataCatalog_node1708237359808, mappings=[(

# Script generated for node Amazon S3
AmazonS3_node1708238624148 = glueContext.write_dynamic_frame.from_options(frame=RemovePriorto1990_node1708238527

job.commit()
```

marthinusswart/aws_glue_jobs_poc

Created by marthinusswart • Updated 3 minutes ago

local running glue jobs

Github

AWS Glue Table

SQL DDL

```
CREATE EXTERNAL TABLE `world_data_indicators_parquet_custom_python`(
  `country_name` string COMMENT '',
  `country_code` string COMMENT '',
  `indicator_name` string COMMENT '',
  `indicator_code` string COMMENT '',
  `1991` decimal COMMENT '',
  `1992` decimal COMMENT '',
  `1993` decimal COMMENT '',
  `1994` decimal COMMENT '',
  `1995` decimal COMMENT '',
  `1996` decimal COMMENT '',
  `1997` decimal COMMENT '',
  `1998` decimal COMMENT '',
  `1999` decimal COMMENT '',
  `2000` decimal COMMENT '',
  `2001` decimal COMMENT '',
  `2002` decimal COMMENT '',
  `2003` decimal COMMENT '',
  `2004` decimal COMMENT '',
  `2005` decimal COMMENT '',
  `2006` decimal COMMENT '',
  `2007` decimal COMMENT '',
  `2008` decimal COMMENT '',
  `2009` decimal COMMENT '',
  `2010` decimal COMMENT '',
  `2011` decimal COMMENT '',
  `2012` decimal COMMENT '',
  `2013` decimal COMMENT '',
  `2014` decimal COMMENT '',
  `2015` decimal COMMENT '',
  `2016` decimal COMMENT '',
  `2017` decimal COMMENT '',
  `2018` decimal COMMENT '',
  `2019` decimal COMMENT '',
  `2020` decimal COMMENT '',
  `2021` decimal COMMENT '',
  `2022` decimal COMMENT '')
ROW FORMAT SERDE
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
  's3://poc-world-bank-data/GC.DOD.TOTL.GD.ZS_python_poc/'
TBLPROPERTIES (
  'classification'='parquet')
```

JSON Schema

```
[
  {
    "Name": "country_name",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "country_code",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "indicator_name",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "indicator_code",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "1991",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "1992",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "1993",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "1994",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "1995",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "1996",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "1997",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "1998",
```

```
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "1999",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2000",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2001",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2002",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2003",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2004",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2005",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2006",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2007",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2008",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2009",
    "Type": "decimal",
    "Comment": ""
  },
```

```
  {
    "Name": "2010",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2011",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2012",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2013",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2014",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2015",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2016",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2017",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2018",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2019",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2020",
    "Type": "decimal",
    "Comment": ""
  },
  {
    "Name": "2021",
    "Type": "decimal",
```

```
    "Comment": ""
  },
  {
    "Name": "2022",
    "Type": "decimal",
    "Comment": ""
  }
]
```

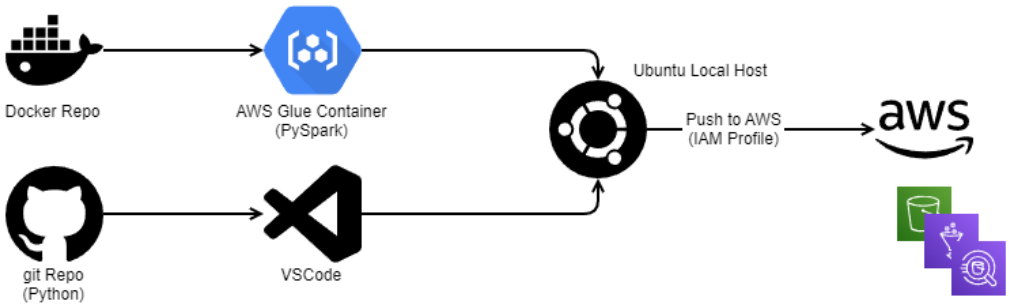
Query Table

```
select * from world_data_indicators_parquet_custom_python;
```

Result

#	country_name	country_code	indicator_name	indicator_code	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001
1	Aruba	ABW	Central government debt, total (% of GDP)	GC.DOD.TOTL.GD.ZS											
2	Africa Eastern and Southern	AFE	Central government debt, total (% of GDP)	GC.DOD.TOTL.GD.ZS											
3	Afghanistan	AFG	Central government debt, total (% of GDP)	GC.DOD.TOTL.GD.ZS											
4	Africa Western and Central	AFW	Central government debt, total (% of GDP)	GC.DOD.TOTL.GD.ZS											
5	Angola	AGO	Central government debt, total (% of GDP)	GC.DOD.TOTL.GD.ZS											
6	Albania	ALB	Central government debt, total (% of GDP)	GC.DOD.TOTL.GD.ZS					36	37	53	56			

AWS Glue Docker Container



Process Flow

1. Fetch AWS Glue Container from the public Docker Repo
2. Run the AWS Glue Container on a local Ubuntu host
3. Fetch the Python code from Git
4. Run the Python job using VSCode on Ubuntu
5. The Python job has full access to the AWS environment via the IAM profile setup to read and write to the S3 buckets