

# AWS Glue

Glue Hail VCF to Parquet job runs on EMR .

Glue job runs on EMR with Hail jar .

?? ??

1. `hail-all-spark.jar` jar file .
2. Amazon S3 bucket contains `hail-all-spark.jar` jar file .
3. Glue job `hail-all-spark.jar` jar file . Copy S3 URL .

Glue job runs on EMR with Hail jar . AWS Glue job runs on EMR .

## AWS IAM

IAM Role `GenomicsAnalysis-Genomics-JobRole-*` . Role `GenomicsAnalysis-Genomics-JobRole-*` .

## GetRole, PassRole

1. Create inline policy .
2. Policy JSON . `{account-id}` . AWS `{GenomicsAnalysis-Genomics-JobRole-*}` .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
```

```

    "iam:GetRole",
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam::**{account-id}**:role/**{GenomicsAnalysis-Genomics-JobRole-*}**"
  ]
}

```

3. Click **Create policy** .
4. Click **Role** **MyGluePolicy** . (Click **MyGluePolicy**)

## S3 Read

1. Click **Add permissions** > **Attach policies**
2. **AmazonS3ReadOnlyAccess** .
3. Click **2 Policy** **2** . (Click **MyGluePolicy** , **AmazonS3ReadOnlyAccess**)

## AWS Glue

1. Click **AWS Glue** .
2. **ETL jobs > Notebook** .

Click **IAM role** **GenomicsAnalysis-Genomics-JobRole-\*** .

3. Glue notebook **hail-all-spark.jar** **S3 URI** **+** .

```

%idle_timeout 2880
%glue_version 4.0
%worker_type G.1X
%number_of_workers 5
%additional_python_modules hail
%extra_jars "**{S3 URI}hail-all-spark.jar S3 URI**"

```

```
%%configure
{
  "--conf": "spark.serializer=org.apache.spark.serializer.KryoSerializer --conf
spark.kryo.registrator=is.hail.kryo.HailKryoRegistrator"
}
```

```
import sys

from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job


import hail as hl


sc = SparkContext.getOrCreate()
hl.init(sc=sc)
```

```
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)

job.init("JobNameEx")















vds = hl.import_vcf("s3://**{[redacted] [redacted] }**/genomics-tertiary-analysis-and-data-lakes-using-aws-glue-and-
amazon-athena/latest/variants/vcf/variants.vcf.gz", force_bgz=True, reference_genome='GRCh38')

vds.make_table().to_spark().write.mode('overwrite').parquet("s3://**{[redacted] [redacted] }**/genomics-tertiary-
analysis-and-data-lakes-using-aws-glue-and-amazon-athena/latest/variants/vcf_to_parquet")

job.commit()
```

4. S3 Parquet .

## Optional ??

-   S3  Query with S3 Select             

---

Revision #3

Created 24 April 2024 12:52:18 by Hyunmin Kim

Updated 26 April 2024 09:07:35 by Hyunmin Kim