HOW TO DEFINE AND RUN A JOB IN AWS GLUE

Here we show how to run a simple job in Amazon Glue.

The basic procedure, which we'll walk you through, is to:

- Create a Python script file (or PySpark)
- Copy it to Amazon S3
- Give the Amazon Glue user access to that S3 bucket
- Run the job in AWS Glue
- Inspect the logs in Amazon CloudWatch

Create Python script

First we create a simple Python script:

```
arr=

for i in range(len(arr)):
    print(arr)
```

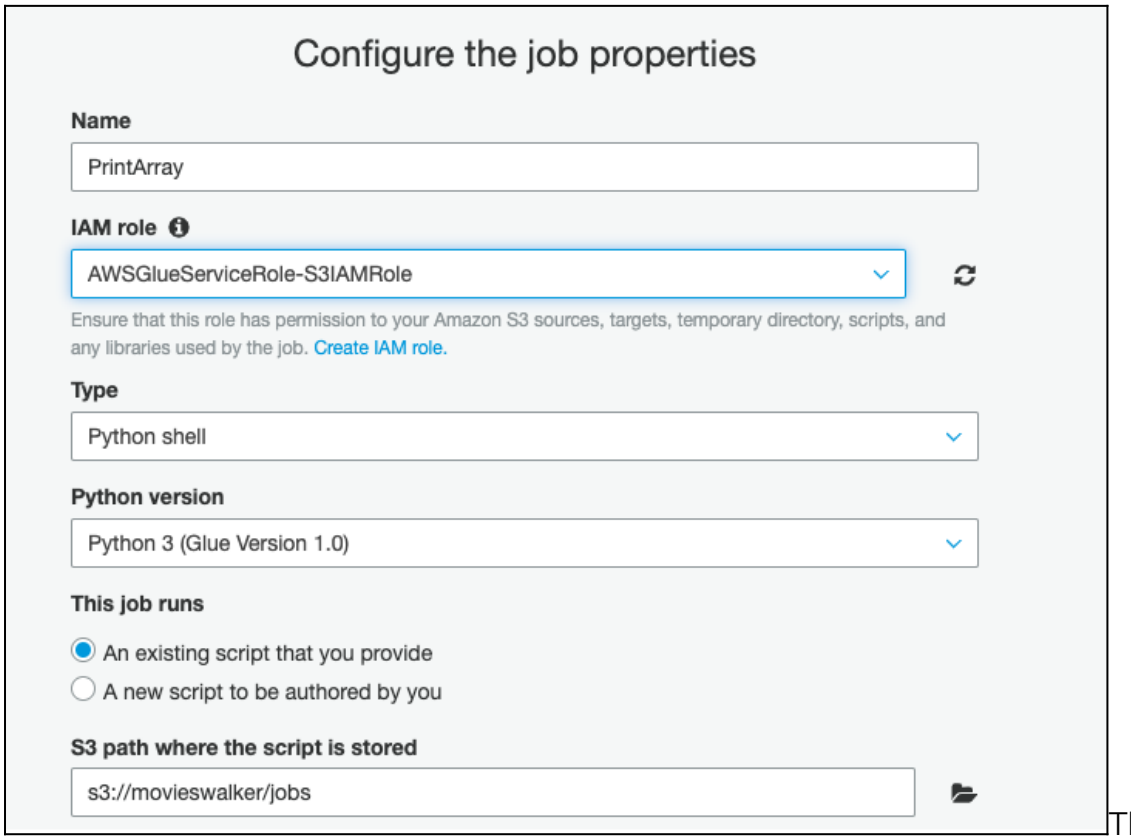
Copy to S3

Then use the Amazon CLI to create an S3 bucket and copy the script to that folder.

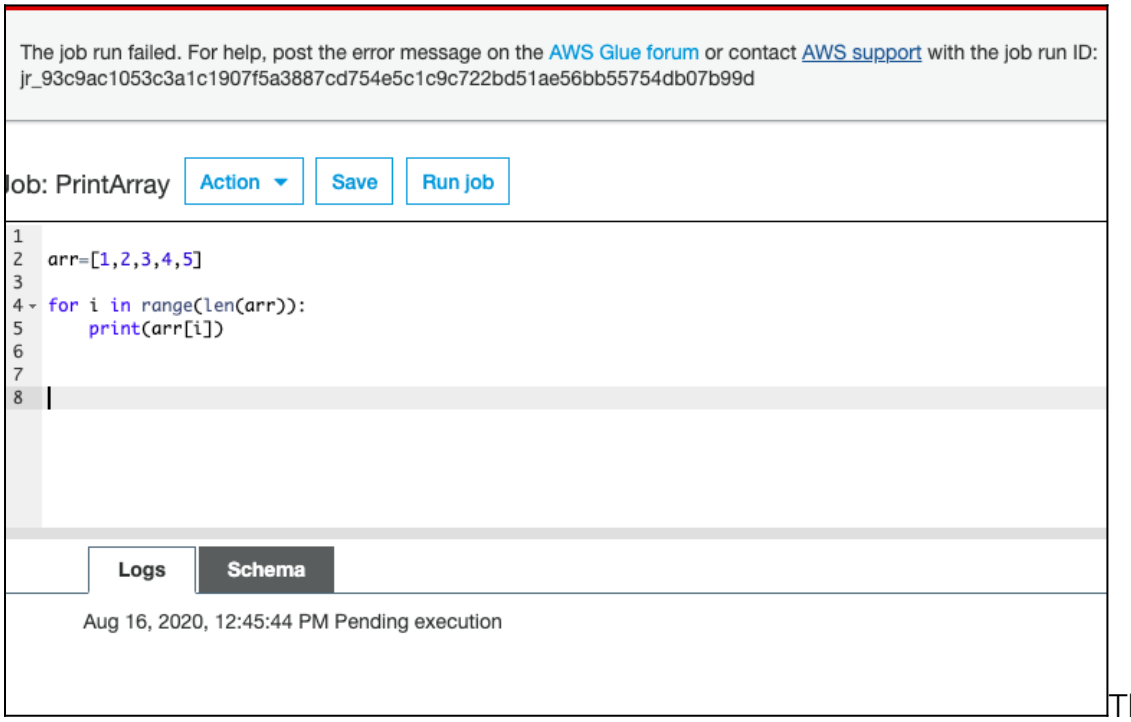
```
aws s3 mb s3://movieswalker/jobs
aws s3 cp counter.py s3://movieswalker/jobs
```

Configure and run job in AWS Glue

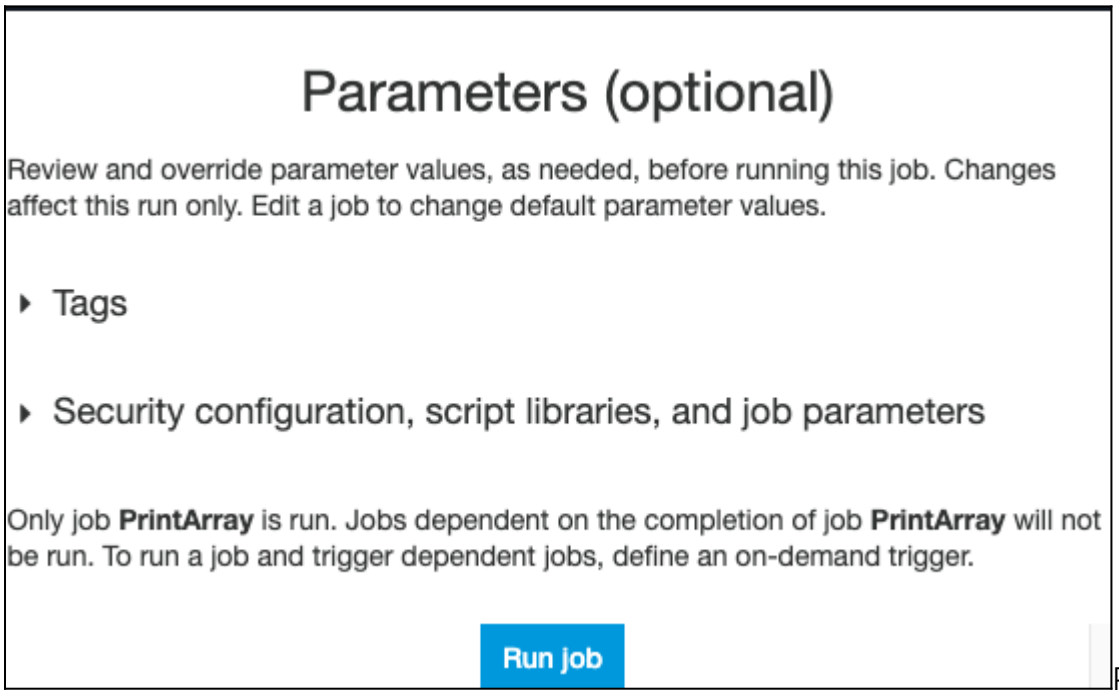
Log into the Amazon Glue console. Go to the Jobs tab and add a job. Give it a name and then pick an Amazon Glue role. The role **AWSGlueServiceRole-S3IAMRole** should already be there. If it is not, add it in IAM and attach it to the user ID you have logged in with. See instructions at the end of this article with regards to the role.



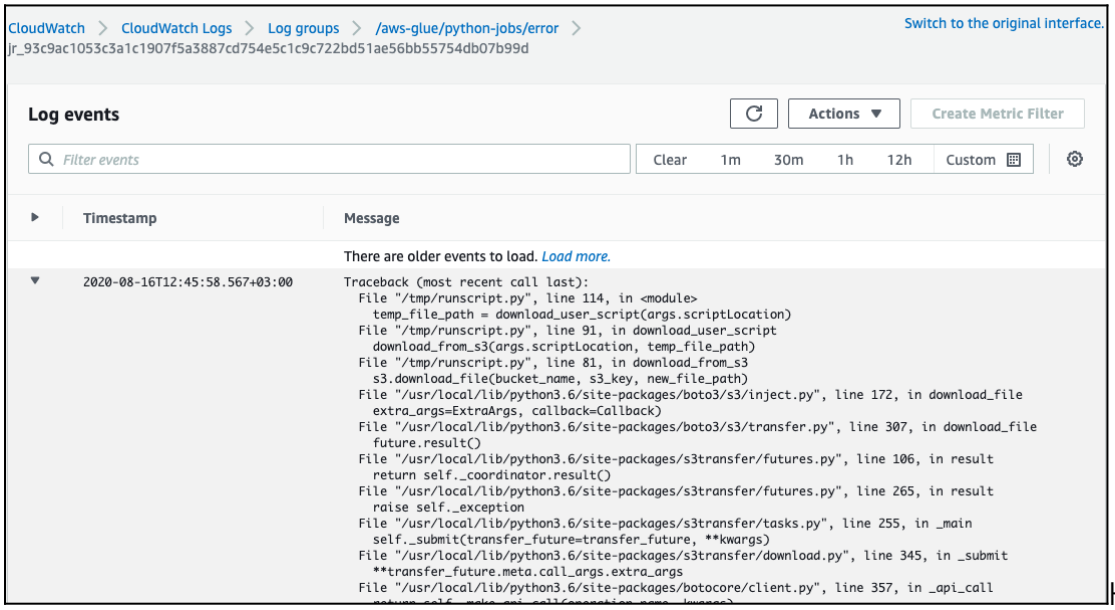
The script editor in Amazon Glue lets you change the Python code.



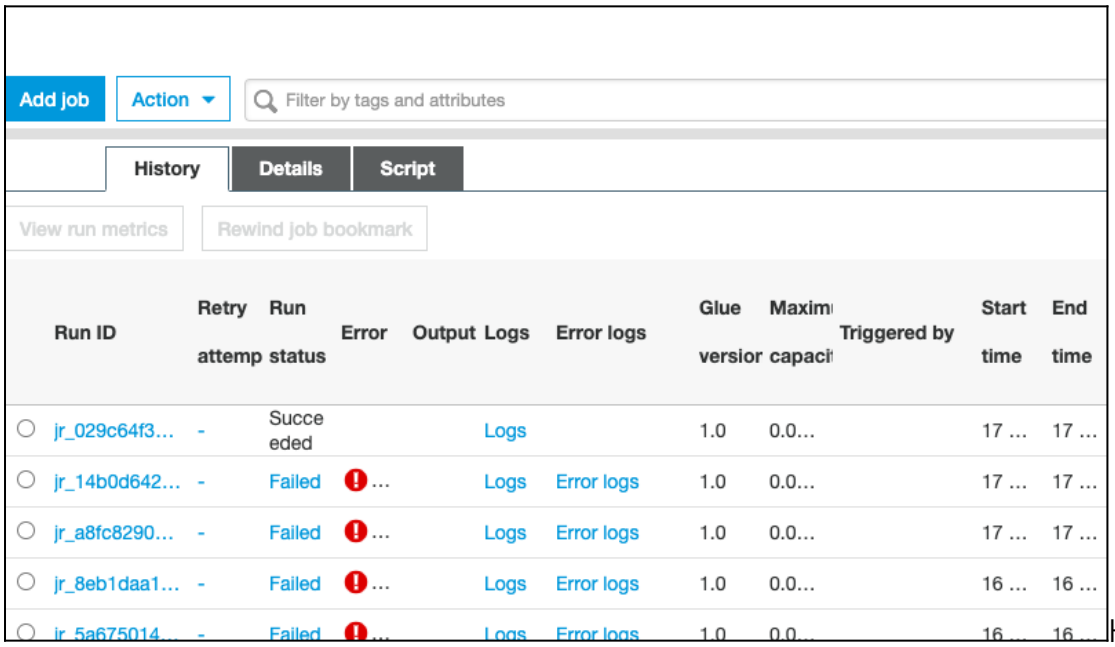
This screen shows that you can pass run-time parameters to the job:



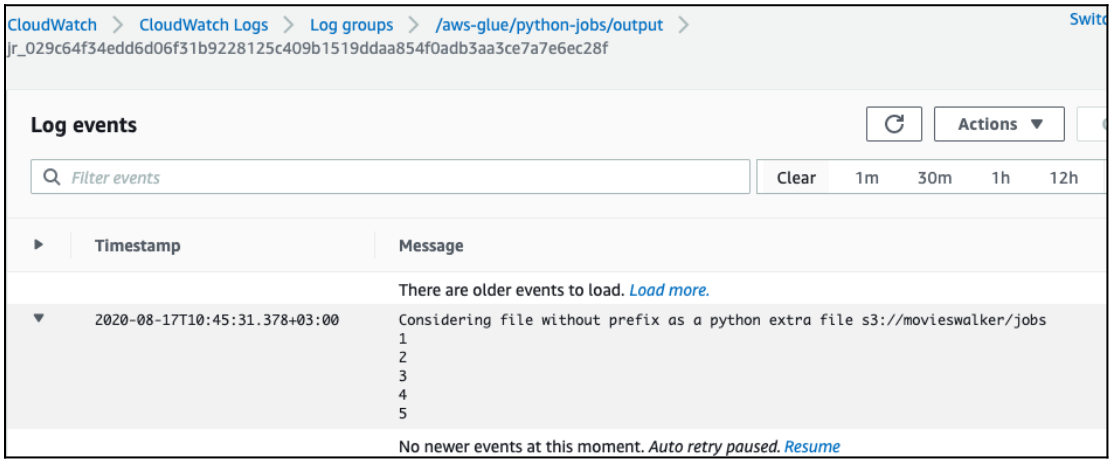
Run the job. When you run it, if there is any error you are directed to CloudWatch where you can see that. The error below is an S3 permissions error:



Here is the job run history.



Here is the log showing that the Python code ran successfully. In this simple example it just printed out the numbers 1,2,3,4,5. Click the **Logs** link to see this log.



Give Glue user access to S3 bucket

If you have run any of our other tutorials, like running a crawler or joining tables, then you might already have the **AWSGlueServiceRole-S3IAMRole**. What's important for running a Glue job is that the role has access to the S3 bucket where the Python script is stored.

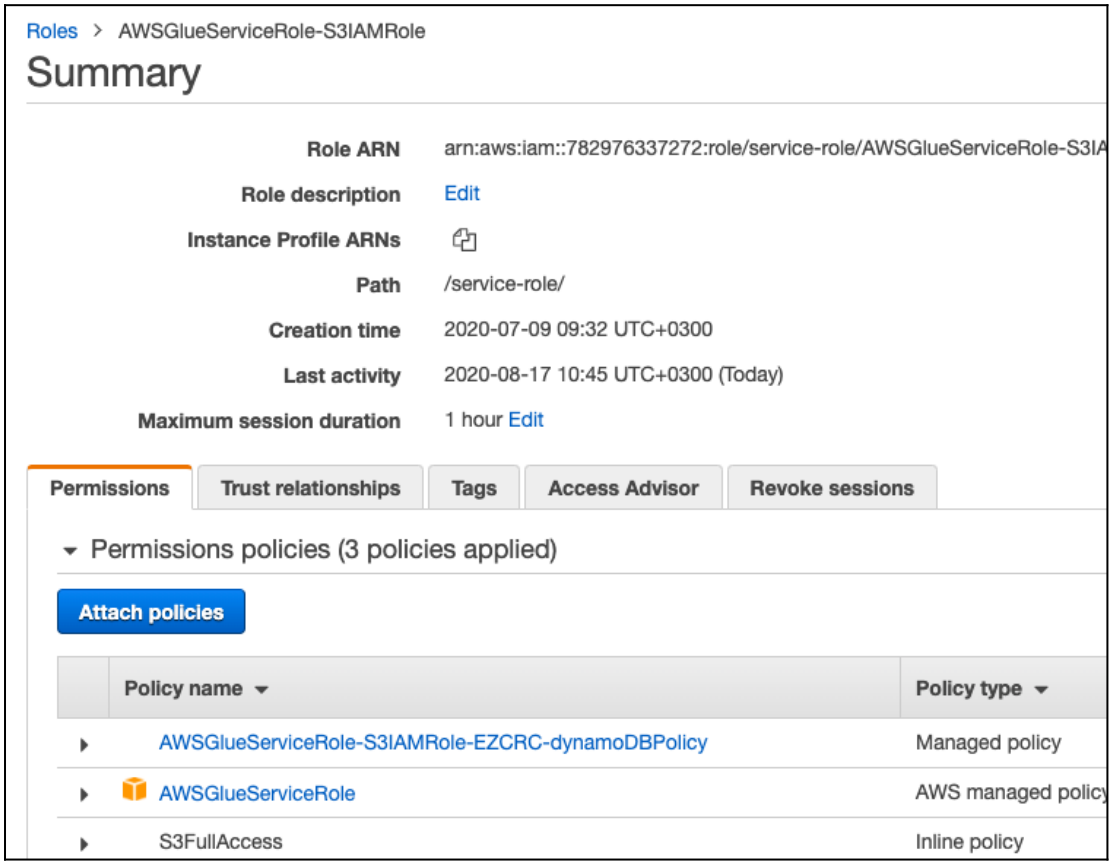
In this example, I added that manually using the JSON Editor in the IAM roles screen and pasted in this policy:

```
{
    "Version": "2012-10-17",
    "Statement":
}
```

If you don't do this, or do it incorrectly, you will get this error:

```
File "/usr/local/lib/python3.6/site-packages/botocore/client.py", line 661,
in _make_api_call
    raise error_class(parsed_response, operation_name)
botocore.exceptions.ClientError: An error occurred (403) when calling the
HeadObject operation: Forbidden
```

Here we show that the user has the **AWSGlueserviceRole** policy and the S3 policy we just added in the **AWSGlueServiceRole-S3IAMRole** role**.** That, of course, must be attached to your IAM userid.



Additional resources

Explore these resources:

- AWS Glue ETL Transformations
- BMC Machine Learning & Big Data Blog
- Apache Spark Guide
- AWS Guide