EMR Serveless

I am Roles

1. Create role for EMR notebook. [emr_notebook_role.json]
    - Create AmazonElasticMapReducePolicy. [AmazonElasticMapReduceEditorsRolepolicy.json]
    - Create AmazonS3FullAccessPolicy. [s3_full_access_policy.json]

Attach the above 2 policies to the EMR notebook Role.

2. Create role for EMR serverless Execution. [emr_serverless_role.json]
    - Create emr serverless policy.[emr_serverless_policy.json]

Attach the above policy to EMR serverless Execution Role.

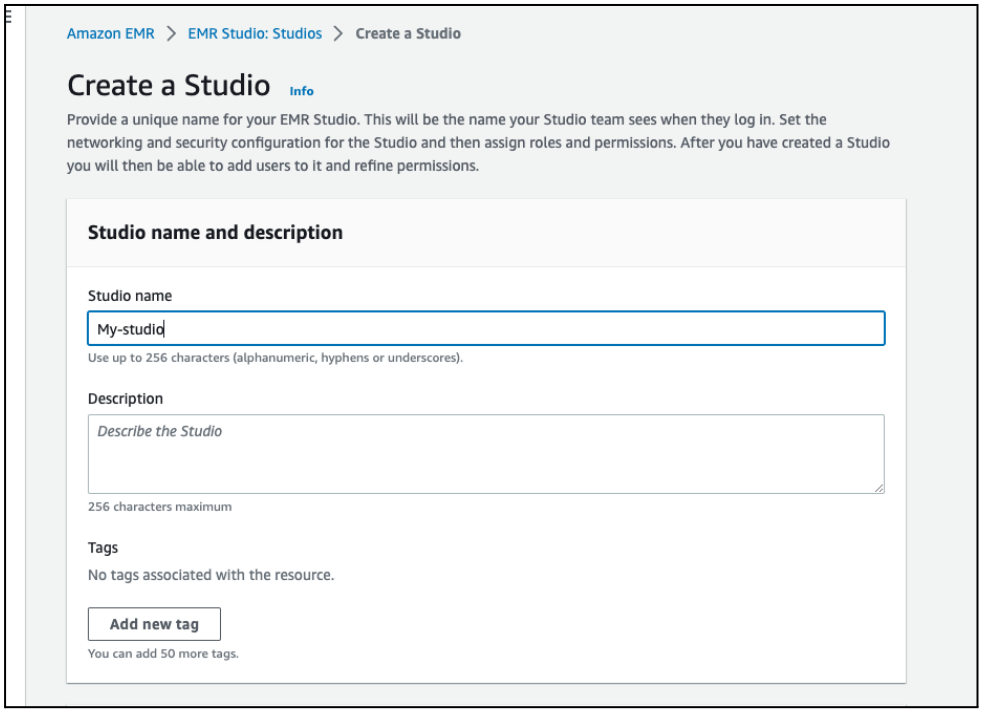
S3 Bucket Name : new-s3-bucket-cli-test

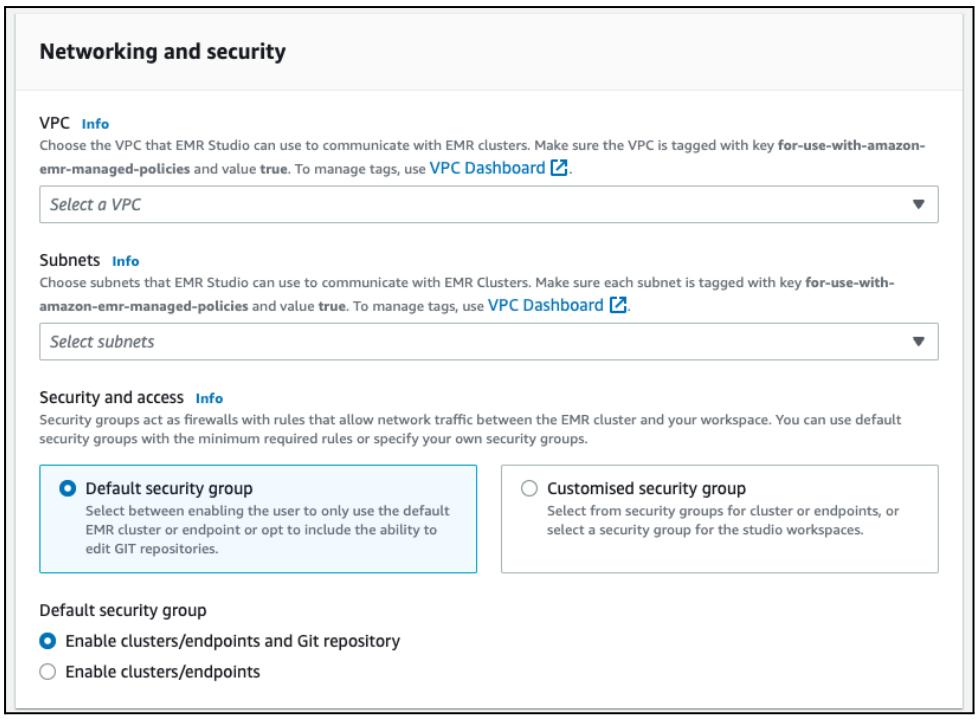
- Under s3 bucket create a folder called scripts to keep the Wordcount .py script or hive script [hive_statements.sql]
- Create another folder called query_results to store the outputs.
- Create another folder to upload a csv file required for running the script.

For submitting Spark job using UI Directly:

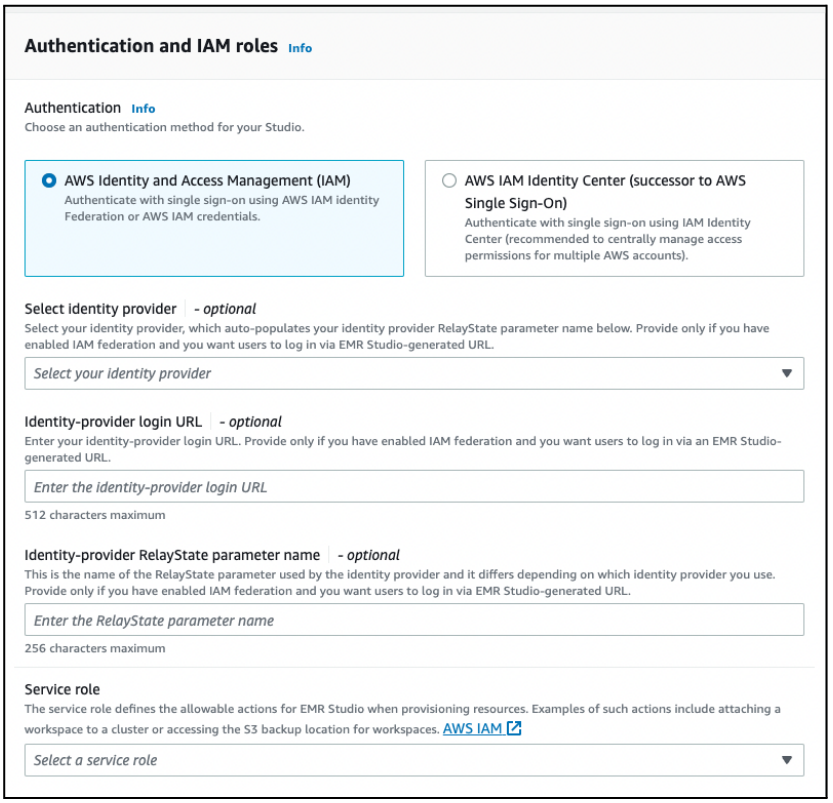
Go to the AWS console -> type EMR -> Click -> choose EMR Studio -> create studio to manage your applications.

Provide a name for the studio name.

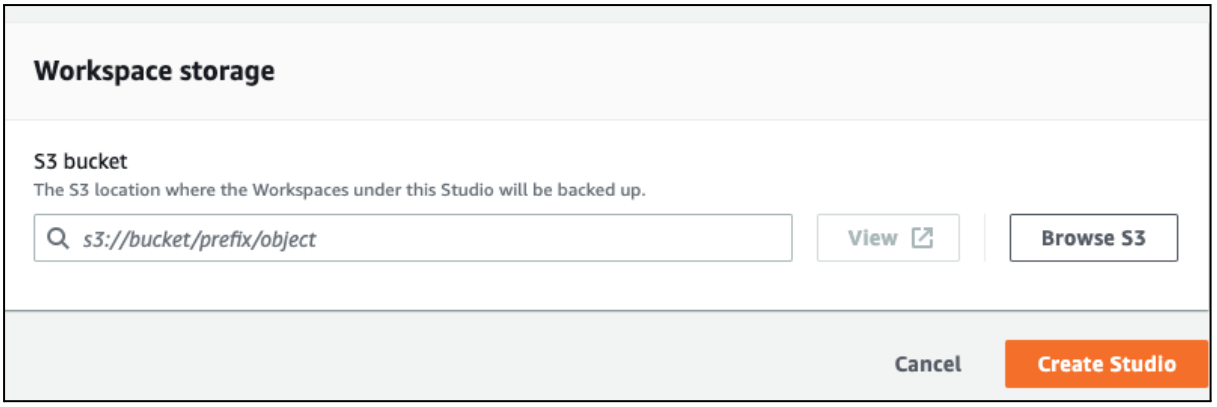




Under network and security. ->Choose the default VPC or your VPC.-> choose 2 or 3 subnets in the subnets field-> rest can be the left with default options.



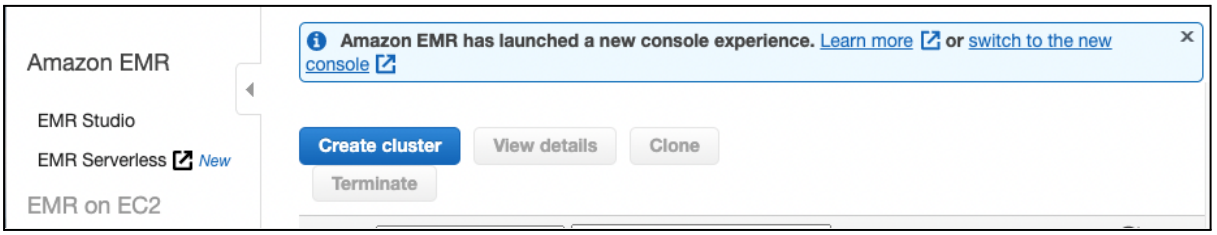
Under the authentication and i am roles -> in service field -> choose the created EMR notebook Role.



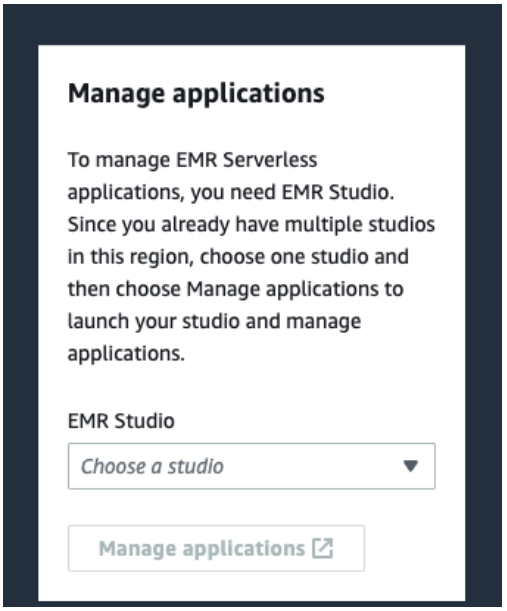
Under workspace and storage -> In s3 Bucket field provide the bucket name directly or browse the bucket name-> click on create Studio.

Successful studio is created !

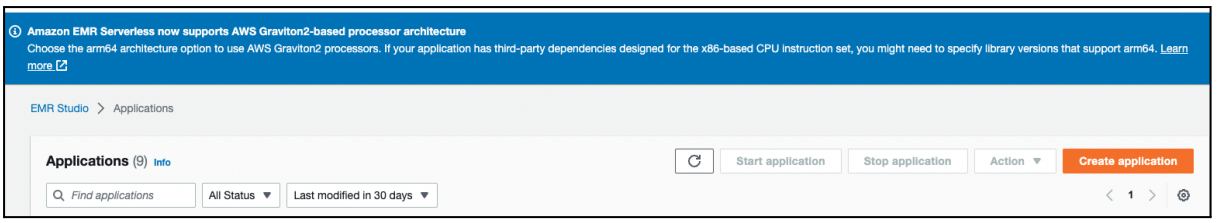
In the AWS EMR console -> click on EMR serveless.



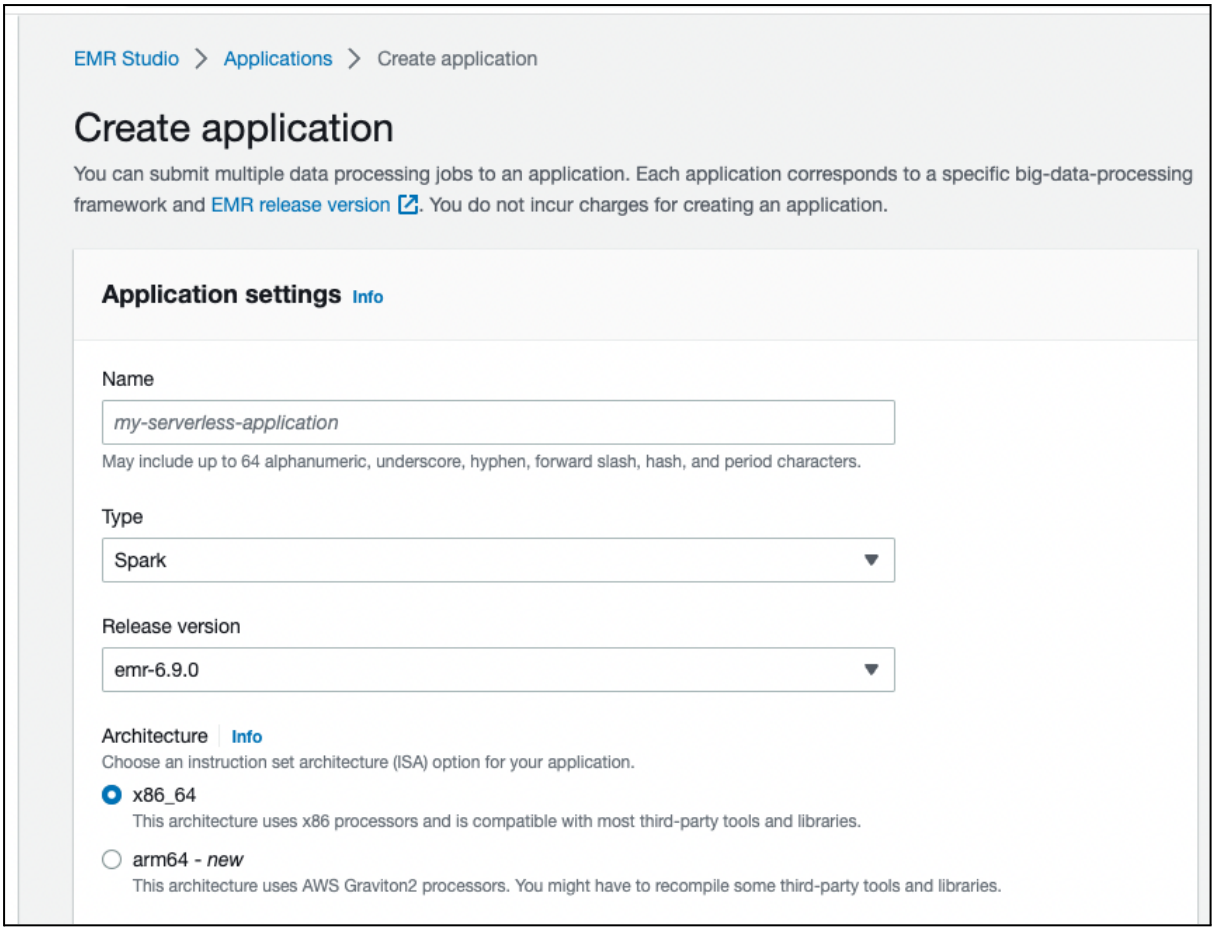
Choose the Studio name you have created and click manage applications -> a tab will open take to the applications page.



Create Application to submit the Spark job or hive job.

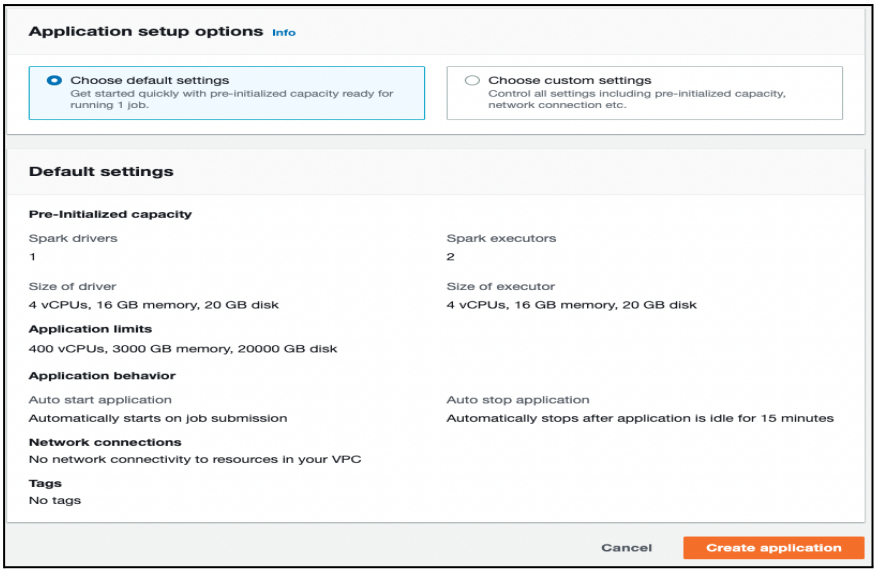


Create an application.

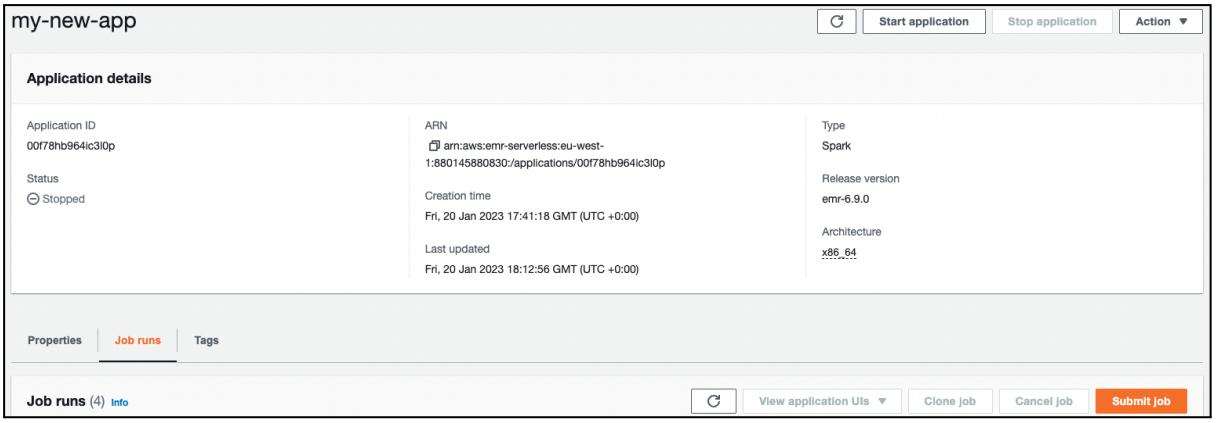


1. Under the application settings in the name provide a name for the application.
2. Under the type field choose spark or Hive Job.
3. Under the Release version choose the version from emr-6.6.0 to emr-69.0. Emr-6.9.0 is the recent version.
4. Leave the Architecture field with the default option.

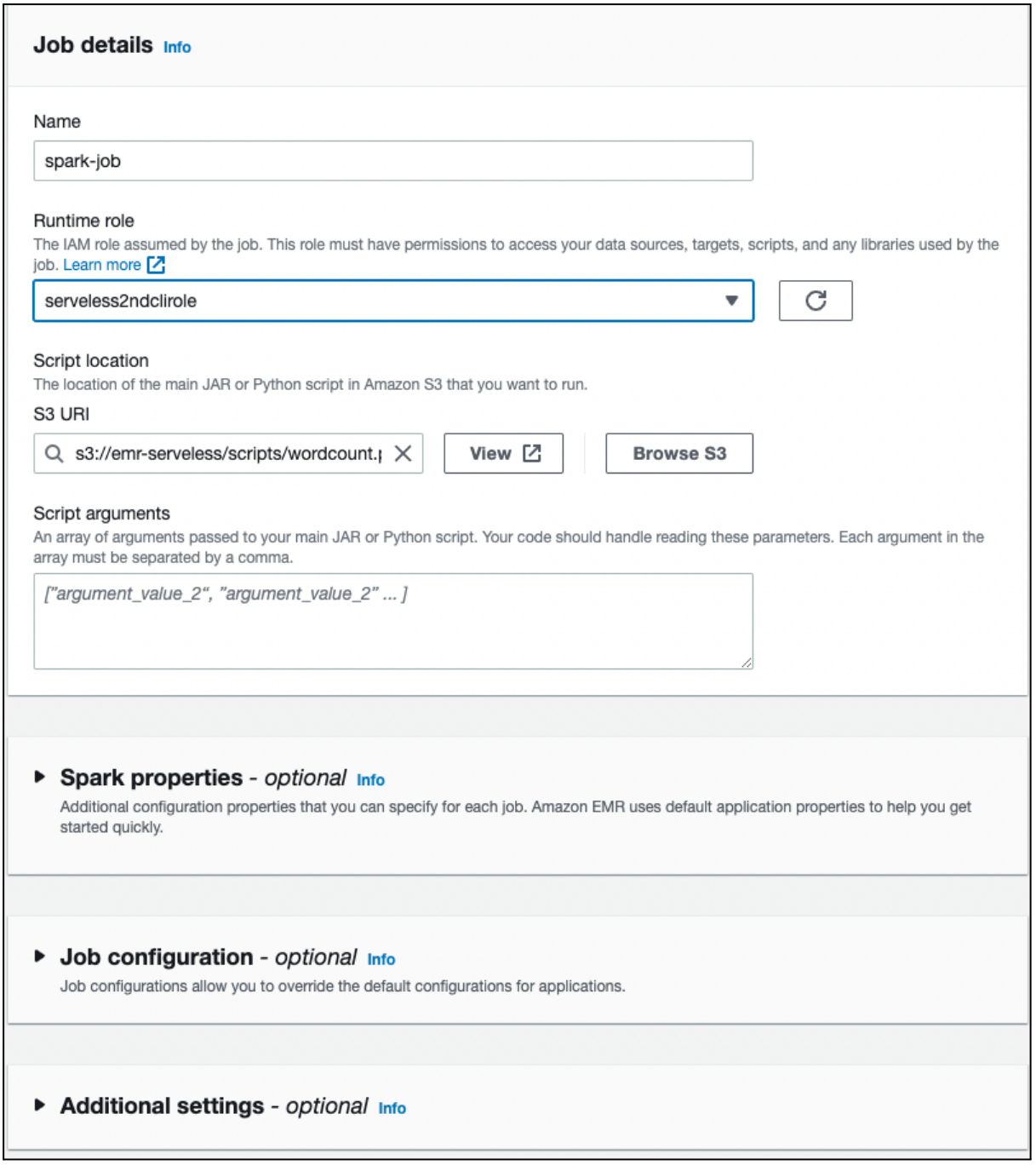
Under the Application setup option leave with the default settings or choose custom settings to customise the application according to the job requirements. -> click on create application.



Once the application created Successfully -> click on the app ->

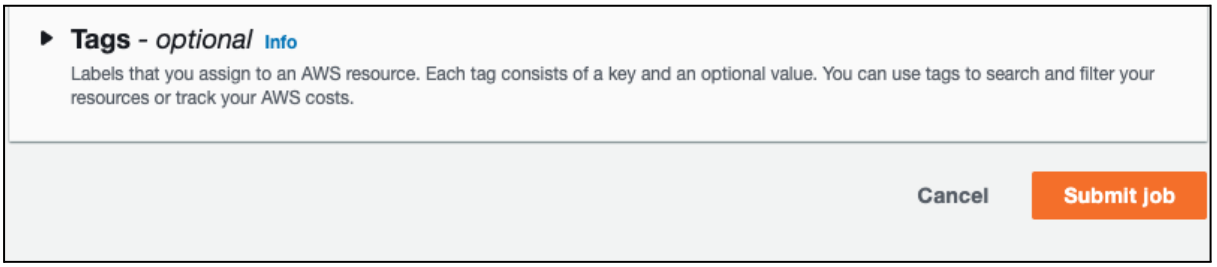


Click on the submit job .

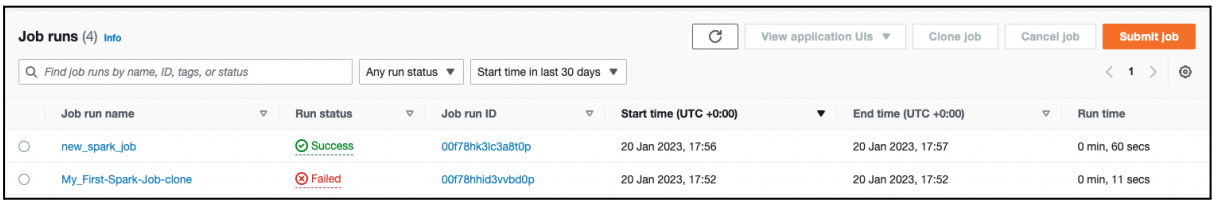


Under job details application.

1. Provide a name for the spark job.
2. Choose the created EMR serverless Execution Role.
3. Under the script location filed provide the path for the wordcount .py.



Click on submit job.



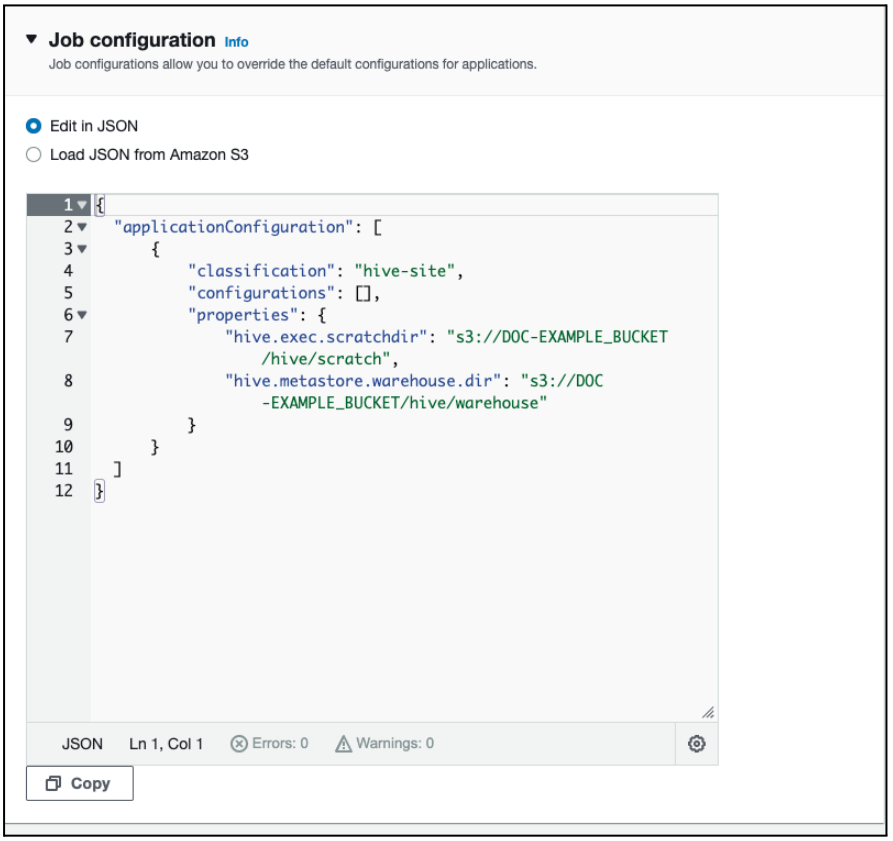
Under jobs will be able to see the job status like -> starting, Scheduling, running, failed or successful.

In the s3 bucket will be able to see the logs of the submitted spark job.

Submitting Hive Job

For submitting hive job create application -> choose -> Type ->hive and version -> create a hive application.

For submitting hive job -> job details -> give a name -> provide sql script uri.



Provide the s3 bucket name in the json and submit the job.

Submitting a spark job using AWS CLI.

1. To create s3 bucket

```
aws s3api create-bucket \
   --bucket  kavitha-s3-test  \
   --region eu-west-1 \
   --create-bucket-configuration LocationConstraint=eu-west-1
```

2. Create folder in s3 bucket and upload the word count.py file

```
aws s3api put-object --bucket kavitha-s3-test --key SparkScript/wordcount.py --body
/Users/kavitharajendran/Documents/wordcount.py
```

3. To create IAM policy.

```
aws iam create-policy --policy-name glue-policy --policy-document
file:///Users/kavitharajendran/Documents/emr-serverless/Glue.yml or json
```

Note: Provide the policy name and the path of the file yml or json file can be used.

4. To create an IAM Role.

```
aws iam create-role --role-name AwsEmrServerlessRole--assume-role-policy-document
file:///Users/kavitharajendran/Documents/emr-serverless/emr_notebook_role.yml or json
```

Note: provide the name and the path of the file.

5. To attach attach the policy with IAm role.

```
aws iam attach-role-policy --policy-arn
arn:aws:iam::880145880830:policy/S3fullaccesstocli-bucket --role-name
AwsEmrServerlessRoledemo
```

Note: Provide the arn of the policy and the role name to attach the policy to role.

6. To Create an application.

```
aws emr-serverless create-application \
   --release-label emr-6.6.0 \
   --type "SPARK" \
   --name was-cli-spark-application-test
```

Provide the name for the spark application, in type provide spark or hive job and provide the version.

7 .For submitting the spark job.

```
aws emr-serverless start-job-run \
   --application-id 00f76oth3gmvog0p --execution-role-arn
arn:aws:iam::880145880830:role/serveless2ndclirole --name  my-application\
   --job-driver '{
      "sparkSubmit": {
```

```
        "entryPoint": "s3://kavitha-s3-test/scripts/wordcount.py",
        "entryPointArguments": ["s3://kavitha-s3-test/SparkScripts/output"],
        "sparkSubmitParameters": "--conf spark.executor.cores=1 --conf
spark.executor.memory=4g --conf spark.driver.cores=1 --conf spark.driver.memory=4g --conf
spark.executor.instances=1"
      }
   }'
```

Note : Provide the name of the application, provide the arn of the EMR serverless Execution Role. and application name.

8. To get the details of the created application

```
aws emr-serverless get-application \
   --application-id 00f76naiqncul30p
```

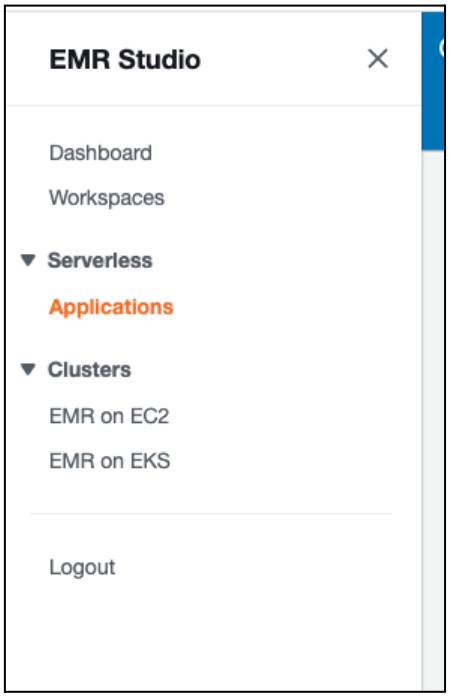
Note: Provide application ID : ”00f76naiqncul30p”

9. To get the status of the running job

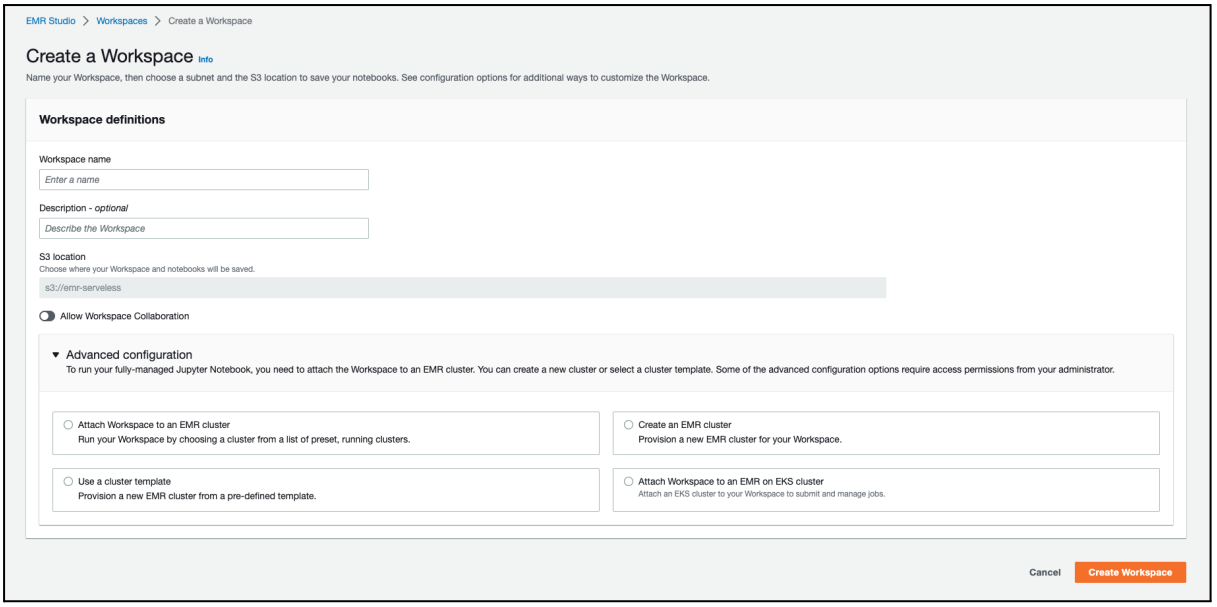
```
aws emr-serverless get-job-run \
   --application-id 00f76oth3gmvog0p\
   --job-run-id 00f76phhr1vv9h0p
```

Note: Provide application ID : ”00f76oth3gmvog0p” and job run id: “00f76phhr1vv9h0p”

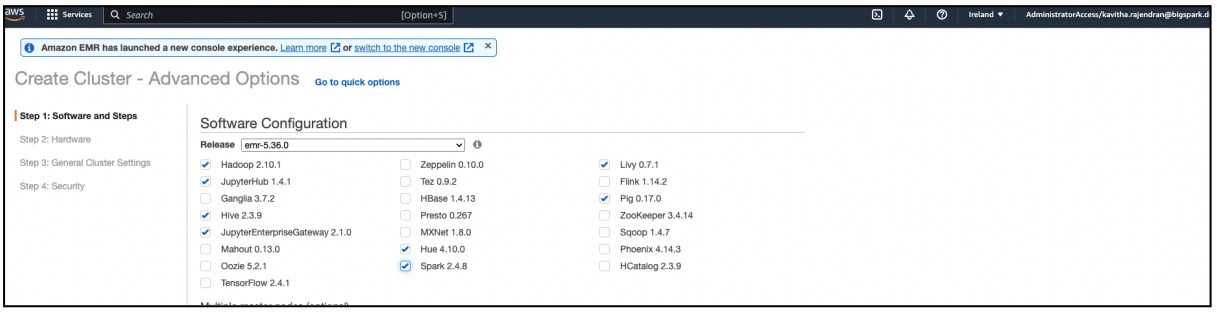
To create a Jupyter notebook :



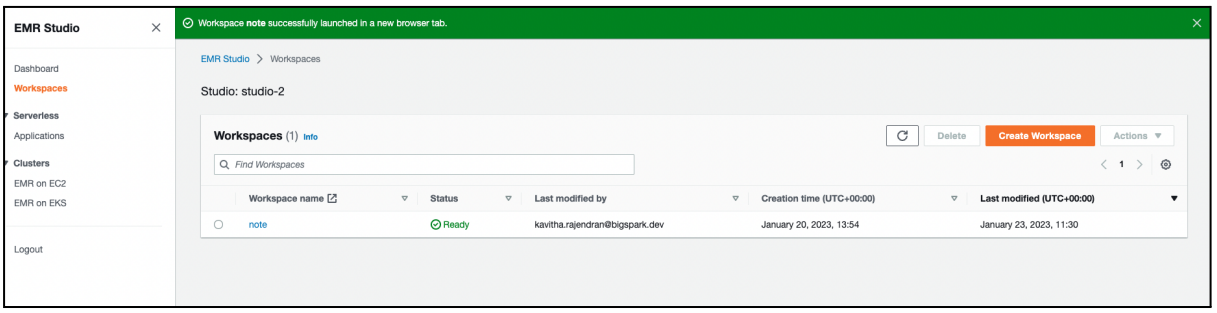
Choose workspace - > create workspace.

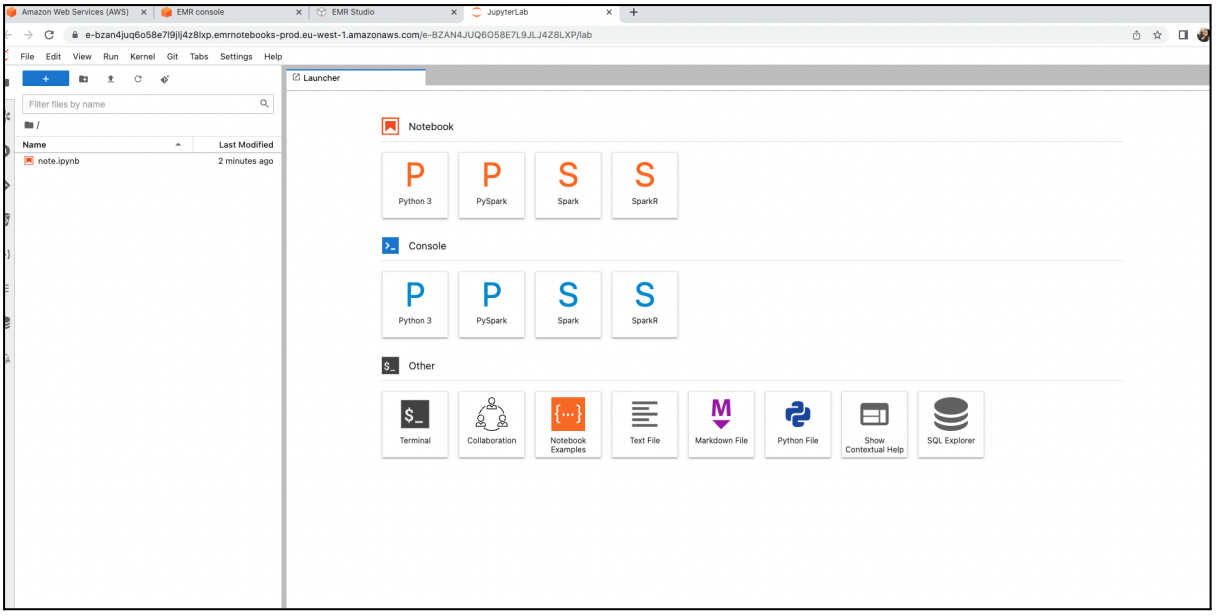


1. Provide a name for the workspace.
2. Create an EMR cluster or attach workspace to an EMR cluster which is created already.
3. Note: if you are attaching to the cluster which is created separately . make sure you choose (jupyterHub,hadoop, JupyterenterpriseGateway,spark,livy,hue,pig)
4. Click create workspace.



Click on the created workspace to launch the JupyterLab




Amazon Web Services (AWS)
EMR console
EMR Studio
JupyterLab
e-bzan4juq6o58e7l9jlj4z8lxp.emrnotebooks-prod.eu-west-1.amazonaws.com/e-BZAN4JUQ6O58E7L9JLJ4Z8LXP/lab
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/
Name
Last Modified
note.ipynb
2 minutes ago
Launcher
Notebook
Python 3
PySpark
Spark
SparkR
Console
Python 3
PySpark
Spark
SparkR
Other
Terminal
Collaboration
Notebook Examples
Text File
Markdown File
Python File
Show Contextual Help
SQL Explorer