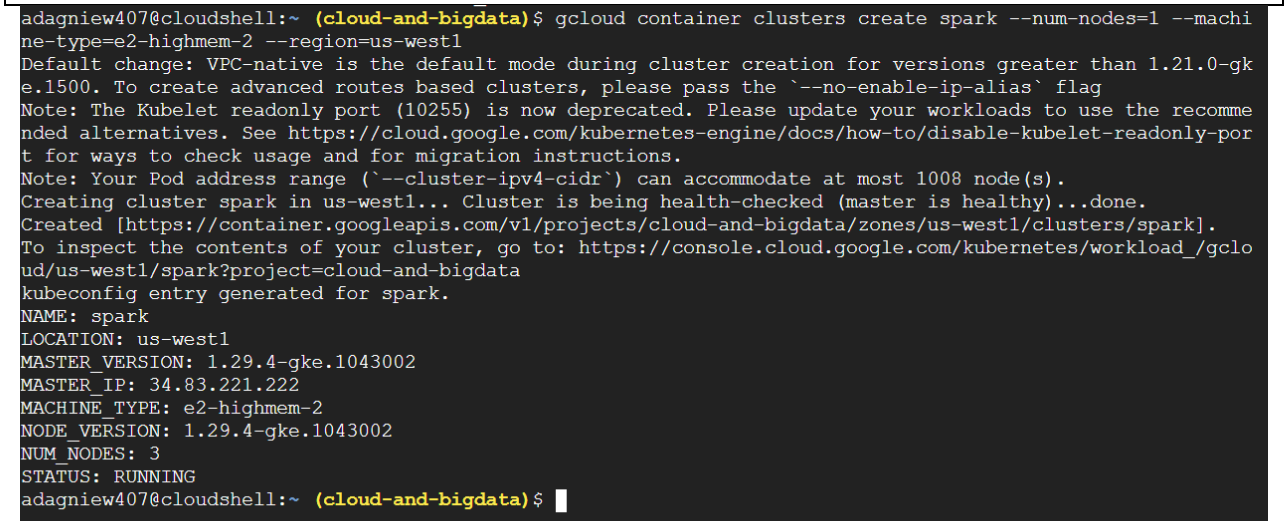
Wordcount, PageRank, running on Spark, deploying to Kubernetes on GKE

Name: Aron Sbhatu Dagniew
ID: 20073

Cluster Setup

1. **Create a Cluster on GKE:**

```
gcloud container clusters create spark --num-nodes=1 --machine-type=e2-highmem-2 --
region=us-west1
```



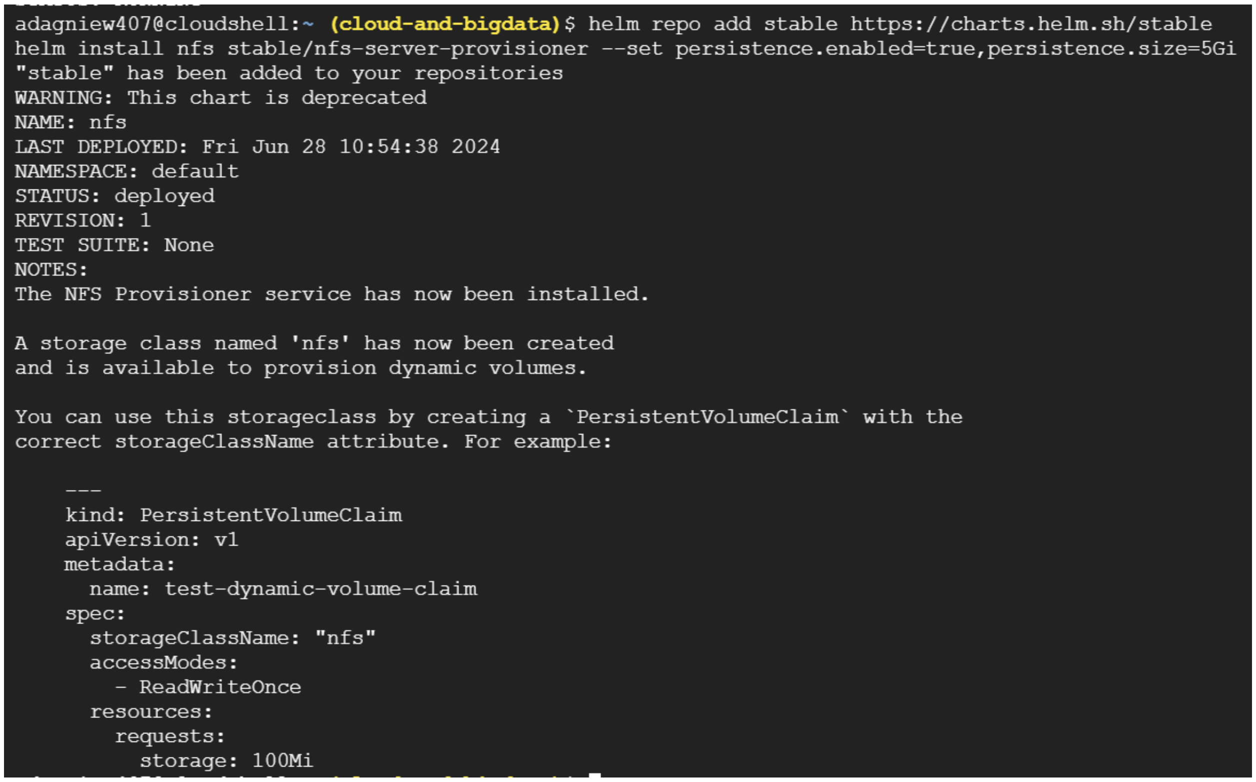
```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gcloud container clusters create spark --num-nodes=1 --machi
ne-type=e2-highmem-2 --region=us-west1
Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gk
e.1500. To create advanced routes based clusters, please pass the `--no-enable-ip-alias` flag
Note: The Kubelet readonly port (10255) is now deprecated. Please update your workloads to use the recomme
nded alternatives. See https://cloud.google.com/kubernetes-engine/docs/how-to/disable-kubelet-readonly-por
t for ways to check usage and for migration instructions.
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
Creating cluster spark in us-west1... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/cloud-and-bigdata/zones/us-west1/clusters/spark].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gclo
ud/us-west1/spark?project=cloud-and-bigdata
kubeconfig entry generated for spark.
NAME: spark
LOCATION: us-west1
MASTER_VERSION: 1.29.4-gke.1043002
MASTER_IP: 34.83.221.222
MACHINE_TYPE: e2-highmem-2
NODE_VERSION: 1.29.4-gke.1043002
NUM_NODES: 3
STATUS: RUNNING
adagniew407@cloudshell:~ (cloud-and-bigdata)$
```

This command sets up a Google Kubernetes Engine (GKE) cluster named `spark` with a single node. The `e2-highmem-2` machine type is selected for its high memory capacity, suitable for running Spark.

NFS Server Provisioner

2. **Install the NFS Server Provisioner:**

```
helm repo add stable https://charts.helm.sh/stable
helm install nfs stable/nfs-server-provisioner --set persistence.enabled=true,persistence.size=5Gi
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ helm repo add stable https://charts.helm.sh/stable
helm install nfs stable/nfs-server-provisioner --set persistence.enabled=true,persistence.size=5Gi
"stable" has been added to your repositories
WARNING: This chart is deprecated
NAME: nfs
LAST DEPLOYED: Fri Jun 28 10:54:38 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The NFS Provisioner service has now been installed.

A storage class named 'nfs' has now been created
and is available to provision dynamic volumes.

You can use this storageclass by creating a `PersistentVolumeClaim` with the
correct storageClassName attribute. For example:

    ---
    kind: PersistentVolumeClaim
    apiVersion: v1
    metadata:
      name: test-dynamic-volume-claim
    spec:
      storageClassName: "nfs"
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 100Mi
```

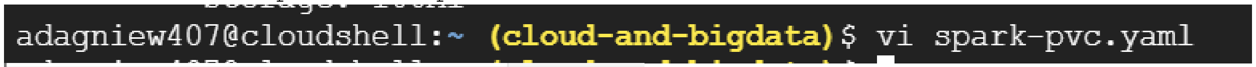
The NFS Server Provisioner is installed using Helm, which facilitates creating persistent storage in the cluster. This setup provides a 5GiB NFS volume for sharing data across the cluster nodes.

Persistent Volume and Pod

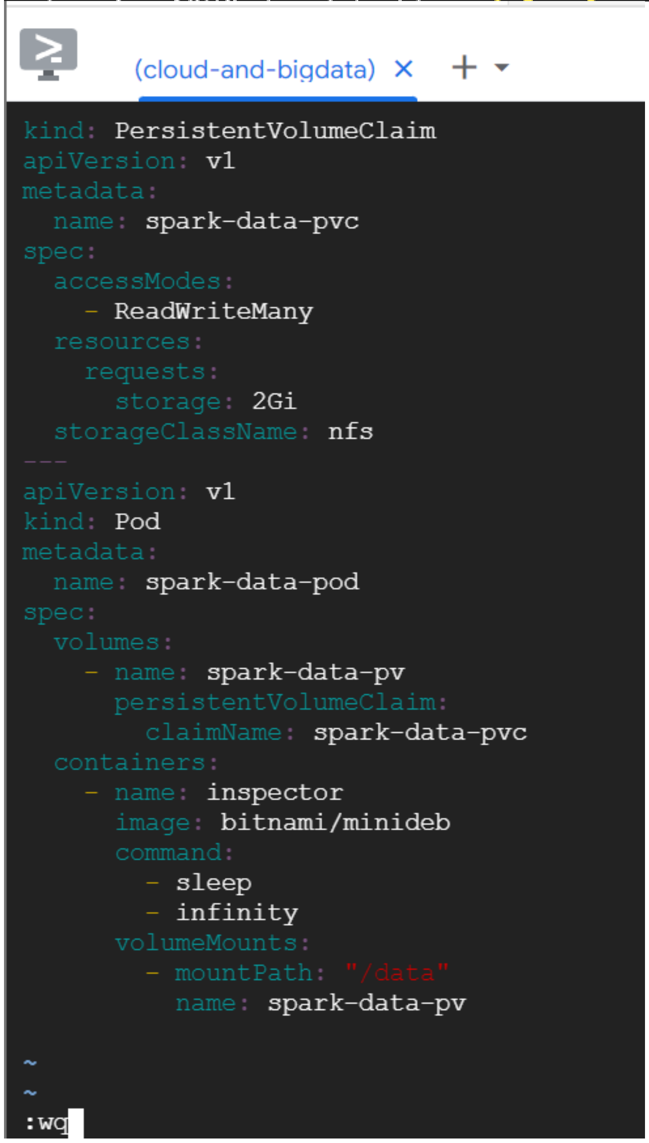
3. **Create a Persistent Disk Volume and a Pod to Use NFS:** Create a YAML file `spark-pvc.yaml` with the following content:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: spark-data-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 2Gi
  storageClassName: nfs
---
apiVersion: v1
kind: Pod
metadata:
  name: spark-data-pod
spec:
  volumes:
    - name: spark-data-pv
```

```
      persistentVolumeClaim:
        claimName: spark-data-pvc
  containers:
    - name: inspector
      image: bitnami/minideb
      command:
        - sleep
        - infinity
      volumeMounts:
        - mountPath: "/data"
          name: spark-data-pv
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ vi spark-pvc.yaml
```

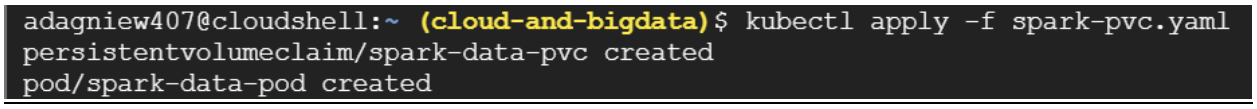


```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: spark-data-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 2Gi
  storageClassName: nfs
---
apiVersion: v1
kind: Pod
metadata:
  name: spark-data-pod
spec:
  volumes:
    - name: spark-data-pv
      persistentVolumeClaim:
        claimName: spark-data-pvc
  containers:
    - name: inspector
      image: bitnami/minideb
      command:
        - sleep
        - infinity
      volumeMounts:
        - mountPath: "/data"
          name: spark-data-pv

~
~
:wq
```

This YAML configuration defines a PersistentVolumeClaim (PVC) named `spark-data-pvc` and a pod named `spark-data-pod` that mounts the PVC. The PVC allows multiple pods to read and write data concurrently.

4. **Apply the YAML Descriptor:**

```
kubectl apply -f spark-pvc.yaml
```



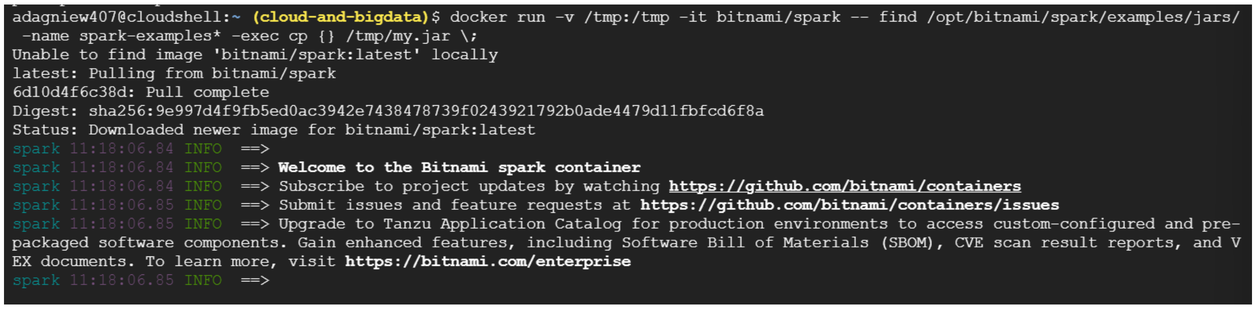
```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl apply -f spark-pvc.yaml
persistentvolumeclaim/spark-data-pvc created
pod/spark-data-pod created
```

This command applies the `spark-pvc.yaml` configuration to the Kubernetes cluster, creating the persistent volume and the pod.

Application Preparation

5. **Create and Prepare Your Application JAR File:**

```
docker run -v /tmp:/tmp -it bitnami/spark -- find /opt/bitnami/spark/examples/jars/ -name spark-examples* -exec cp {} /tmp/my.jar \;
```

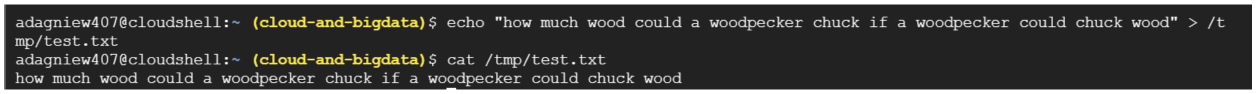


```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ docker run -v /tmp:/tmp -it bitnami/spark -- find /opt/bitnami/spark/examples/jars/
 -name spark-examples* -exec cp {} /tmp/my.jar \;
Unable to find image 'bitnami/spark:latest' locally
latest: Pulling from bitnami/spark
6d10d4f6c38d: Pull complete
Digest: sha256:9e997d4f9fb5ed0ac3942e7438478739f0243921792b0ade4479d11fbfcd6f8a
Status: Downloaded newer image for bitnami/spark:latest
spark 11:18:06.84 INFO  ==>
spark 11:18:06.84 INFO  ==> Welcome to the Bitnami spark container
spark 11:18:06.84 INFO  ==> Subscribe to project updates by watching https://github.com/bitnami/containers
spark 11:18:06.85 INFO  ==> Submit issues and feature requests at https://github.com/bitnami/containers/issues
spark 11:18:06.85 INFO  ==> Upgrade to Tanzu Application Catalog for production environments to access custom-configured and pre-
packaged software components. Gain enhanced features, including Software Bill of Materials (SBOM), CVE scan result reports, and V
EX documents. To learn more, visit https://bitnami.com/enterprise
spark 11:18:06.85 INFO  ==>
```

This command runs a Docker container with Spark installed and copies the example JAR file to the local `/tmp` directory, renaming it to `my.jar`. This JAR file contains the word count application.

6. **Add a Test File:**

```
echo "how much wood could a woodpecker chuck if a woodpecker could chuck wood" > /tmp/test.txt
```

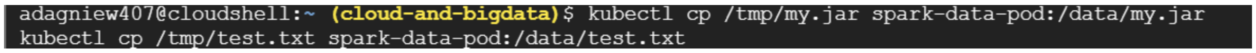


```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ echo "how much wood could a woodpecker chuck if a woodpecker could chuck wood" > /t
mp/test.txt
adagniew407@cloudshell:~ (cloud-and-bigdata)$ cat /tmp/test.txt
how much wood could a woodpecker chuck if a woodpecker could chuck wood
```

A test file named `test.txt` is created with a sample sentence. This file will be used later for the word count task.

7. **Copy the JAR File and Test File to the PVC:**

```
kubectl cp /tmp/my.jar spark-data-pod:/data/my.jar
kubectl cp /tmp/test.txt spark-data-pod:/data/test.txt
```

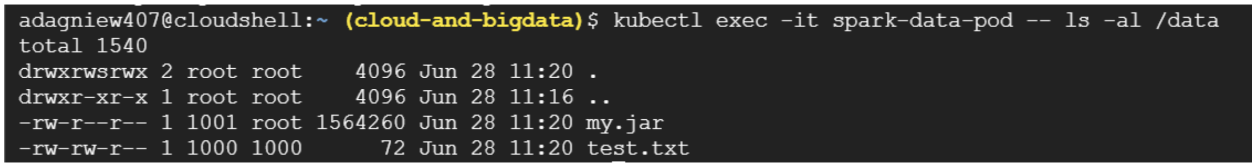


```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl cp /tmp/my.jar spark-data-pod:/data/my.jar
kubectl cp /tmp/test.txt spark-data-pod:/data/test.txt
```

These commands copy the JAR file and the test file from the local system to the PVC mounted at `/data` in the `spark-data-pod`.

8. **Ensure Files are Inside the Persistent Volume:**

```
kubectl exec -it spark-data-pod -- ls -al /data
```



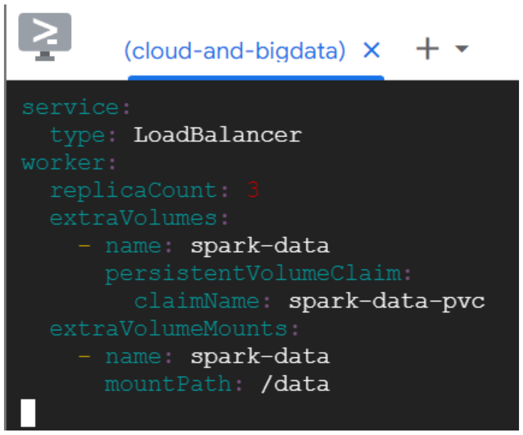
```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl exec -it spark-data-pod -- ls -al /data
total 1540
drwxrwsrwx 2 root root    4096 Jun 28 11:20 .
drwxr-xr-x 1 root root    4096 Jun 28 11:16 ..
-rw-r--r-- 1 1001 root 1564260 Jun 28 11:20 my.jar
-rw-rw-r-- 1 1000 1000      72 Jun 28 11:20 test.txt
```

This command lists the contents of the `/data` directory inside the `spark-data-pod` to verify that the files have been copied correctly.

Spark Deployment

9. **Deploy Apache Spark on Kubernetes Using the Shared Volume:** Create a YAML file `spark-chart.yaml` with the following content:

```
service:
  type: LoadBalancer
worker:
  replicaCount: 3
  extraVolumes:
    - name: spark-data
      persistentVolumeClaim:
        claimName: spark-data-pvc
  extraVolumeMounts:
    - name: spark-data
      mountPath: /data
```

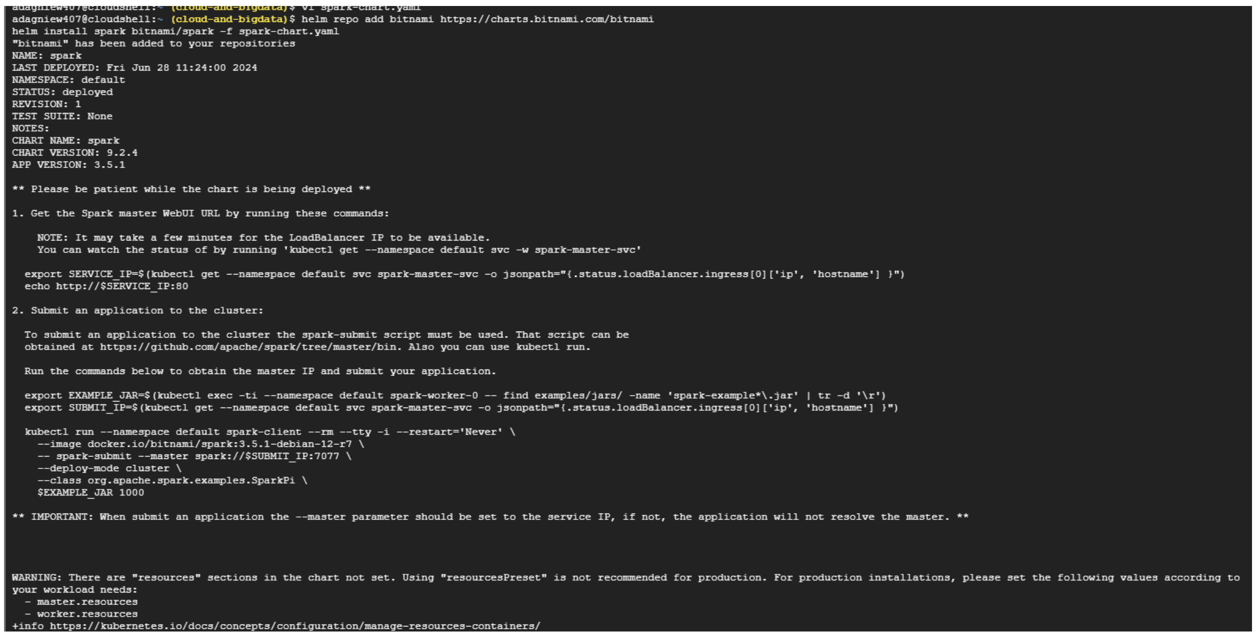


```
service:
  type: LoadBalancer
worker:
  replicaCount: 3
  extraVolumes:
    - name: spark-data
      persistentVolumeClaim:
        claimName: spark-data-pvc
  extraVolumeMounts:
    - name: spark-data
      mountPath: /data
```

This configuration sets up a Spark deployment on Kubernetes with three worker nodes. It uses a shared volume (`spark-data`) mounted at `/data`.

10. **Deploy Apache Spark Using the Bitnami Helm Chart:**

```
helm repo add bitnami https://charts.bitnami.com/bitnami
helm install spark bitnami/spark -f spark-chart.yaml
```

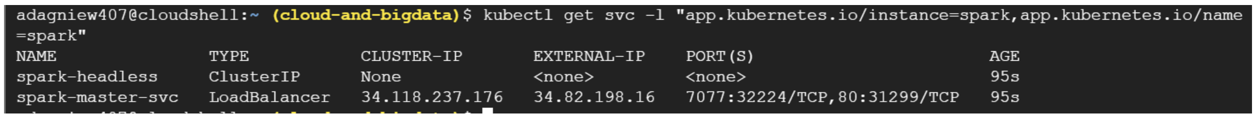


```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ vi spark-chart.yaml
adagniew407@cloudshell:~ (cloud-and-bigdata)$ helm repo add bitnami https://charts.bitnami.com/bitnami
helm install spark bitnami/spark -f spark-chart.yaml
"bitnami" has been added to your repositories
NAME: spark
LAST DEPLOYED: Fri Jun 28 11:24:00 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: spark
CHART VERSION: 9.2.4
APP VERSION: 3.5.1

** Please be patient while the chart is being deployed **

1. Get the Spark master WebUI URL by running these commands:

    NOTE: It may take a few minutes for the LoadBalancer IP to be available.
    You can watch the status of by running 'kubectl get --namespace default svc -w spark-master-svc'

  export SERVICE_IP=$(kubectl get --namespace default svc spark-master-svc -o jsonpath="{.status.loadBalancer.ingress[0]['ip', 'hostname'] }")
  echo http://$SERVICE_IP:80

2. Submit an application to the cluster:

  To submit an application to the cluster the spark-submit script must be used. That script can be
  obtained at https://github.com/apache/spark/tree/master/bin. Also you can use kubectl run.

  Run the commands below to obtain the master IP and submit your application.

  export EXAMPLE_JAR=$(kubectl exec -ti --namespace default spark-worker-0 -- find examples/jars/ -name 'spark-example*\.jar' | tr -d '\r')
  export SUBMIT_IP=$(kubectl get --namespace default svc spark-master-svc -o jsonpath="{.status.loadBalancer.ingress[0]['ip', 'hostname'] }")

  kubectl run --namespace default spark-client --rm --tty -i --restart='Never' \
    --image docker.io/bitnami/spark:3.5.1-debian-12-r7 \
    -- spark-submit --master spark://$SUBMIT_IP:7077 \
    --deploy-mode cluster \
    --class org.apache.spark.examples.SparkPi \
    $EXAMPLE_JAR 1000

** IMPORTANT: When submit an application the --master parameter should be set to the service IP, if not, the application will not resolve the master. **



WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is not recommended for production. For production installations, please set the following values according to
your workload needs:
  - master.resources
  - worker.resources
+info https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
```

The Bitnami Helm chart for Spark is used to deploy Spark on the Kubernetes cluster. The `spark-chart.yaml` configuration is applied to set up the cluster.

11. **Get the External IP of the Running Pod:**

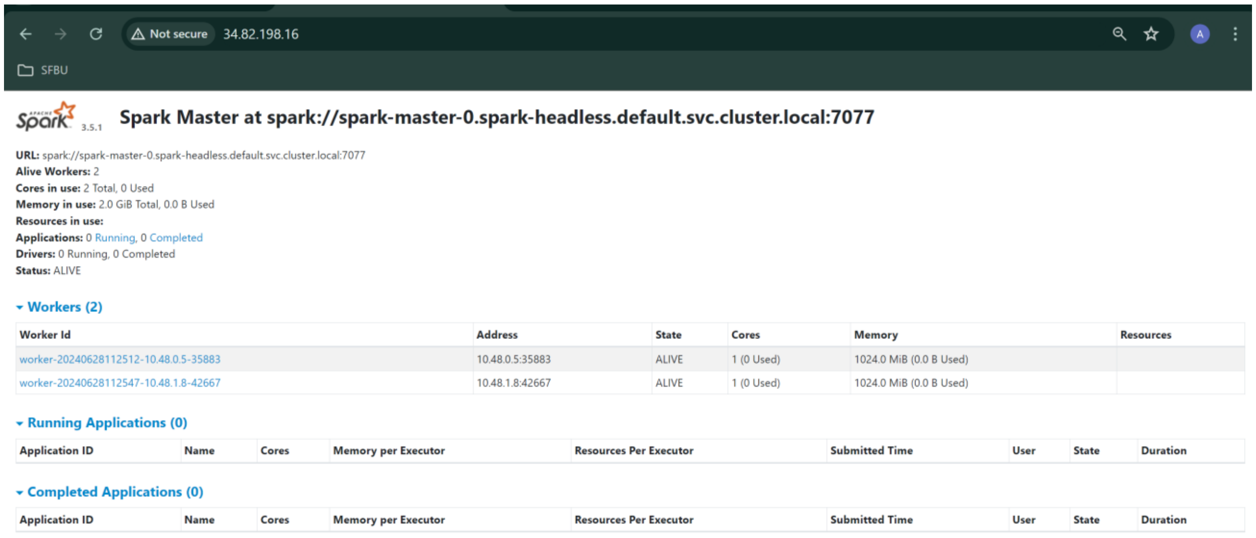
```
kubectl get svc -l "app.kubernetes.io/instance=spark,app.kubernetes.io/name=spark"
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl get svc -l "app.kubernetes.io/instance=spark,app.kubernetes.io/name
=spark"
NAME               TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)                       AGE
spark-headless     ClusterIP      None             <none>         <none>                        95s
spark-master-svc   LoadBalancer   34.118.237.176   34.82.198.16   7077:32224/TCP,80:31299/TCP   95s
```

This command retrieves the external IP address of the running Spark pod, which is necessary for accessing the Spark cluster from a web browser.

12. Open the external ip on your browser



Spark Master at spark://spark-master-0.spark-headless.default.svc.cluster.local:7077

URL: spark://spark-master-0.spark-headless.default.svc.cluster.local:7077
Alive Workers: 2
Cores in use: 2 Total, 0 Used
Memory in use: 2.0 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20240628112512-10.48.0.5-35883	10.48.0.5:35883	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20240628112547-10.48.1.8-42667	10.48.1.8:42667	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

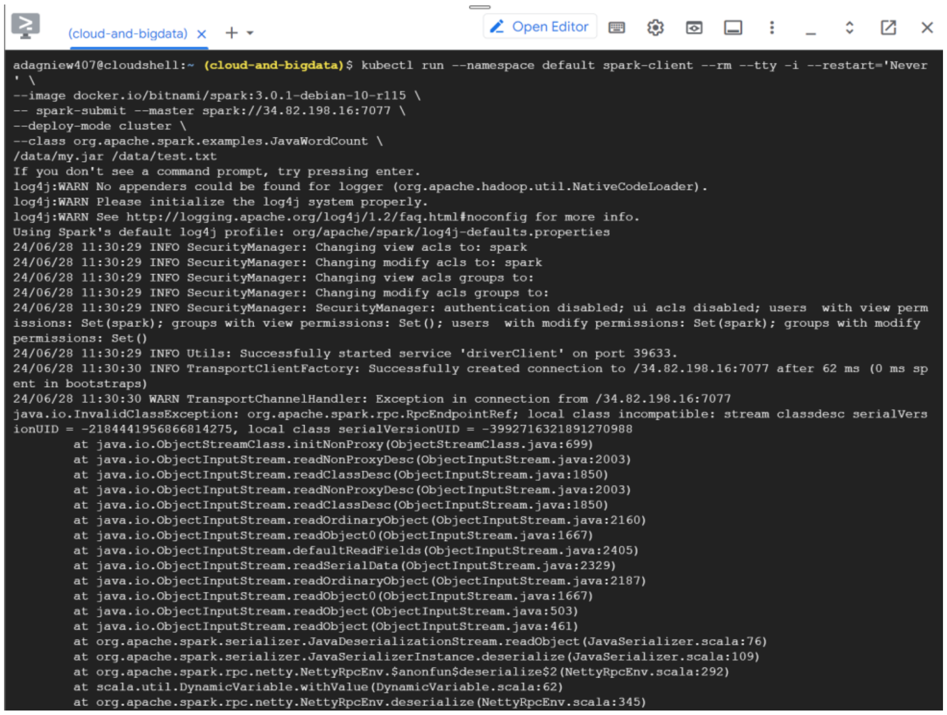
Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Word Count on Spark

12. **Submit a Word Count Task:**

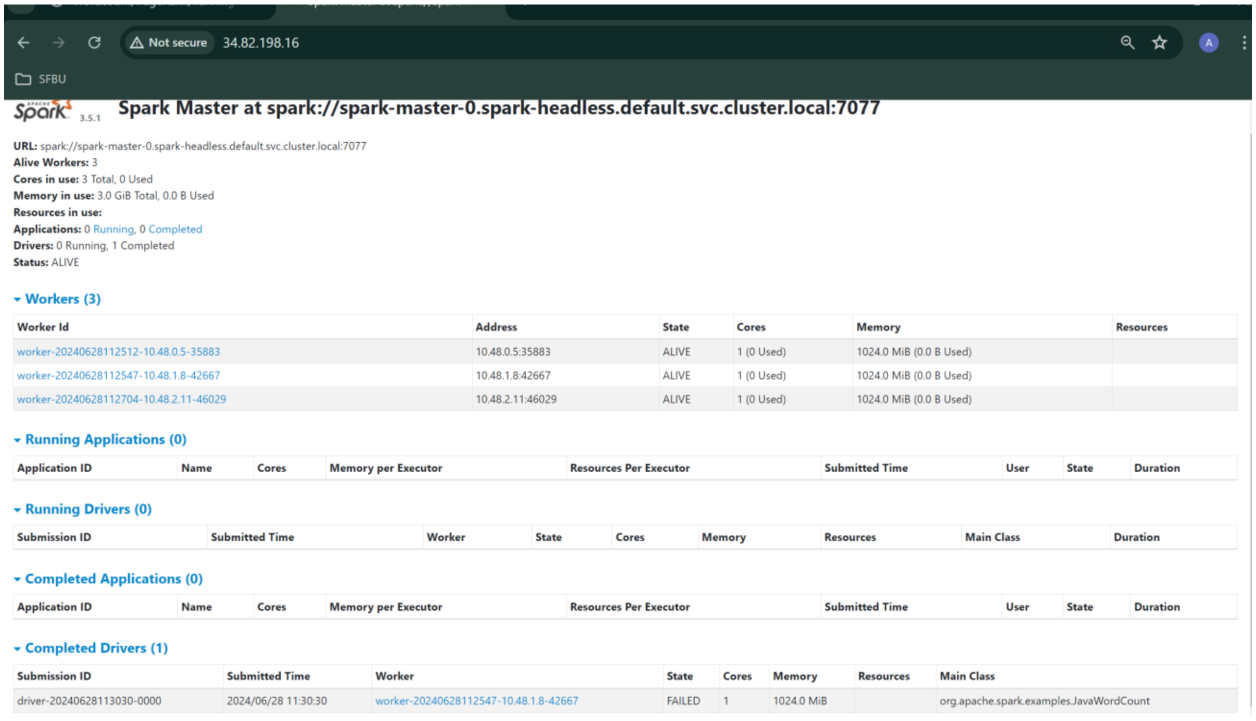
```
kubectl run --namespace default spark-client --rm --tty -i --restart='Never' --image
docker.io/bitnami/spark:3.0.1-debian-10-r115 -- spark-submit --master spark://LOAD-BALANCER-
External-ip-ADDRESS:7077 --deploy-mode cluster --class
org.apache.spark.examples.JavaWordCount /data/my.jar /data/test.txt
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl run --namespace default spark-client --rm --tty -i --restart='Never
' \
--image docker.io/bitnami/spark:3.0.1-debian-10-r115 \
-- spark-submit --master spark://34.82.198.16:7077 \
--deploy-mode cluster \
--class org.apache.spark.examples.JavaWordCount \
/data/my.jar /data/test.txt
If you don't see a command prompt, try pressing enter.
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.NativeCodeLoader).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
24/06/28 11:30:29 INFO SecurityManager: Changing view acls to: spark
24/06/28 11:30:29 INFO SecurityManager: Changing modify acls to: spark
24/06/28 11:30:29 INFO SecurityManager: Changing view acls groups to:
24/06/28 11:30:29 INFO SecurityManager: Changing modify acls groups to:
24/06/28 11:30:29 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view perm
issions: Set(spark); groups with view permissions: Set(); users  with modify permissions: Set(spark); groups with modify
permissions: Set()
24/06/28 11:30:29 INFO Utils: Successfully started service 'driverClient' on port 39633.
24/06/28 11:30:30 INFO TransportClientFactory: Successfully created connection to /34.82.198.16:7077 after 62 ms (0 ms sp
ent in bootstraps)
24/06/28 11:30:30 WARN TransportChannelHandler: Exception in connection from /34.82.198.16:7077
java.io.InvalidClassException: org.apache.spark.rpc.RpcEndpointRef; local class incompatible: stream classdesc serialVers
ionUID = -2184441956866814275, local class serialVersionUID = -3992716321891270988
        at java.io.ObjectStreamClass.initNonProxy(ObjectStreamClass.java:699)
        at java.io.ObjectInputStream.readNonProxyDesc(ObjectInputStream.java:2003)
        at java.io.ObjectInputStream.readClassDesc(ObjectInputStream.java:1850)
        at java.io.ObjectInputStream.readNonProxyDesc(ObjectInputStream.java:2003)
        at java.io.ObjectInputStream.readClassDesc(ObjectInputStream.java:1850)
        at java.io.ObjectInputStream.readOrdinaryObject(ObjectInputStream.java:2160)
        at java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1667)
        at java.io.ObjectInputStream.defaultReadFields(ObjectInputStream.java:2405)
        at java.io.ObjectInputStream.readSerialData(ObjectInputStream.java:2329)
        at java.io.ObjectInputStream.readOrdinaryObject(ObjectInputStream.java:2187)
        at java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1667)
        at java.io.ObjectInputStream.readObject(ObjectInputStream.java:503)
        at java.io.ObjectInputStream.readObject(ObjectInputStream.java:461)
        at org.apache.spark.serializer.JavaDeserializationStream.readObject(JavaSerializer.scala:76)
        at org.apache.spark.serializer.JavaSerializerInstance.deserialize(JavaSerializer.scala:109)
        at org.apache.spark.rpc.netty.NettyRpcEnv.$anonfun$deserialize$2(NettyRpcEnv.scala:292)
        at scala.util.DynamicVariable.withValue(DynamicVariable.scala:62)
        at org.apache.spark.rpc.netty.NettyRpcEnv.deserialize(NettyRpcEnv.scala:345)
        at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:365)
        at io.netty.channel.DefaultChannelPipeline.fireChannelRead(DefaultChannelPipeline.java:919)
        at io.netty.channel.nio.AbstractNioByteChannel$NioByteUnsafe.read(AbstractNioByteChannel.java:163)
        at io.netty.channel.nio.NioEventLoop.processSelectedKey(NioEventLoop.java:714)
        at io.netty.channel.nio.NioEventLoop.processSelectedKeysOptimized(NioEventLoop.java:650)
        at io.netty.channel.nio.NioEventLoop.processSelectedKeys(NioEventLoop.java:576)
        at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:493)
        at io.netty.util.concurrent.SingleThreadEventExecutor$4.run(SingleThreadEventExecutor.java:989)
        at io.netty.util.internal.ThreadExecutorMap$2.run(ThreadExecutorMap.java:74)
        at io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
        at java.lang.Thread.run(Thread.java:748)
24/06/28 11:30:30 ERROR ClientEndpoint: Error connecting to master (34.82.198.16:7077).
24/06/28 11:30:30 ERROR ClientEndpoint: Cause was: java.io.InvalidClassException: org.apache.spark.rpc.RpcEndpointRef; lo
cal class incompatible: stream classdesc serialVersionUID = -2184441956866814275, local class serialVersionUID = -3992716
321891270988
24/06/28 11:30:30 ERROR ClientEndpoint: No master is available, exiting.
24/06/28 11:30:30 INFO ShutdownHookManager: Shutdown hook called
24/06/28 11:30:30 INFO ShutdownHookManager: Deleting directory /tmp/spark-364b52bf-9d32-4a81-85f4-6d8ab5ba832b
pod "spark-client" deleted
```

This command submits a word count task to the Spark cluster, using the JAR file and test file stored in the PVC. The external IP address of the load balancer is used to connect to the Spark master.

Error: Task Failed



Spark Master at spark://spark-master-0.spark-headless.default.svc.cluster.local:7077

URL: spark://spark-master-0.spark-headless.default.svc.cluster.local:7077
Alive Workers: 3
Cores in use: 3 Total, 0 Used
Memory in use: 3.0 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 1 Completed
Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory	Resources
worker-20240628112512-10.48.0.5-35883	10.48.0.5:35883	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20240628112547-10.48.1.8-42667	10.48.1.8:42667	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20240628112704-10.48.2.11-46029	10.48.2.11:46029	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Running Drivers (0)

Submission ID	Submitted Time	Worker	State	Cores	Memory	Resources	Main Class	Duration

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

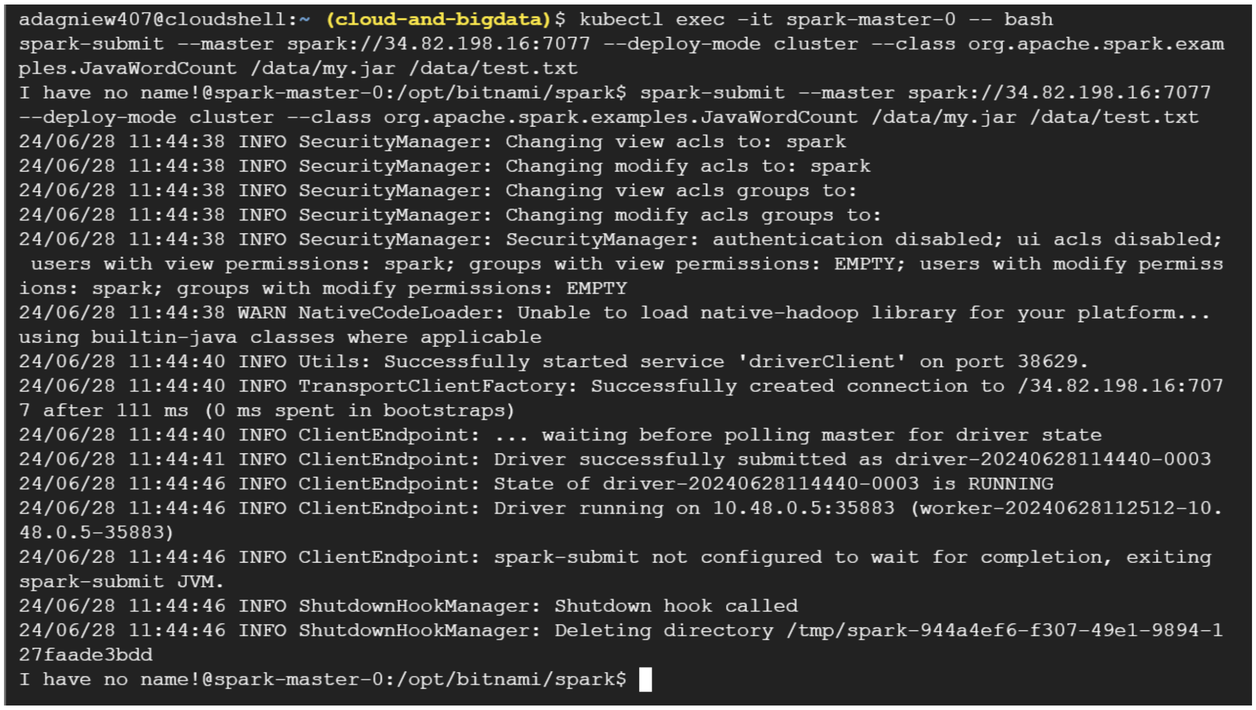
Completed Drivers (1)

Submission ID	Submitted Time	Worker	State	Cores	Memory	Resources	Main Class
driver-20240628113030-0000	2024/06/28 11:30:30	worker-20240628112547-10.48.1.8-42667	FAILED	1	1024.0 MiB		org.apache.spark.examples.JavaWordCount

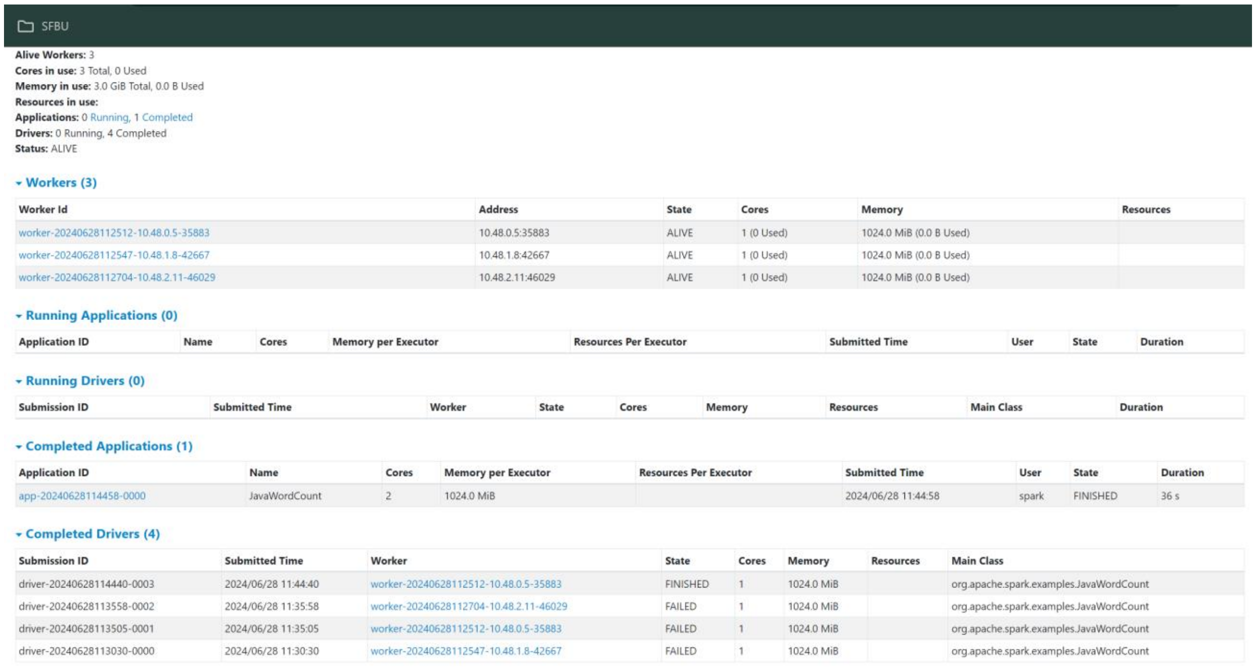
After encountering issues with submitting Spark jobs using `gcloud`, I switched to using `kubectl exec` from the Spark master node to successfully run the job.

Try to run submit the job from the master:

```
kubectl exec -it spark-master-0 -- spark-submit --master spark://34.82.198.16:7077 --deploy-mode cluster --class org.apache.spark.examples.JavaWordCount /data/my.jar /data/test.txt
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl exec -it spark-master-0 -- bash
spark-submit --master spark://34.82.198.16:7077 --deploy-mode cluster --class org.apache.spark.exam
ples.JavaWordCount /data/my.jar /data/test.txt
I have no name!@spark-master-0:/opt/bitnami/spark$ spark-submit --master spark://34.82.198.16:7077
--deploy-mode cluster --class org.apache.spark.examples.JavaWordCount /data/my.jar /data/test.txt
24/06/28 11:44:38 INFO SecurityManager: Changing view acls to: spark
24/06/28 11:44:38 INFO SecurityManager: Changing modify acls to: spark
24/06/28 11:44:38 INFO SecurityManager: Changing view acls groups to:
24/06/28 11:44:38 INFO SecurityManager: Changing modify acls groups to:
24/06/28 11:44:38 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled;
 users with view permissions: spark; groups with view permissions: EMPTY; users with modify permiss
ions: spark; groups with modify permissions: EMPTY
24/06/28 11:44:38 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform...
using builtin-java classes where applicable
24/06/28 11:44:40 INFO Utils: Successfully started service 'driverClient' on port 38629.
24/06/28 11:44:40 INFO TransportClientFactory: Successfully created connection to /34.82.198.16:707
7 after 111 ms (0 ms spent in bootstraps)
24/06/28 11:44:40 INFO ClientEndpoint: ... waiting before polling master for driver state
24/06/28 11:44:41 INFO ClientEndpoint: Driver successfully submitted as driver-20240628114440-0003
24/06/28 11:44:46 INFO ClientEndpoint: State of driver-20240628114440-0003 is RUNNING
24/06/28 11:44:46 INFO ClientEndpoint: Driver running on 10.48.0.5:35883 (worker-20240628112512-10.
48.0.5-35883)
24/06/28 11:44:46 INFO ClientEndpoint: spark-submit not configured to wait for completion, exiting
spark-submit JVM.
24/06/28 11:44:46 INFO ShutdownHookManager: Shutdown hook called
24/06/28 11:44:46 INFO ShutdownHookManager: Deleting directory /tmp/spark-944a4ef6-f307-49e1-9894-1
27faade3bdd
I have no name!@spark-master-0:/opt/bitnami/spark$
```



Alive Workers: 3
Cores in use: 3 Total, 0 Used
Memory in use: 3.0 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 1 Completed
Drivers: 0 Running, 4 Completed
Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory	Resources
worker-20240628112512-10.48.0.5-35883	10.48.0.5:35883	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20240628112547-10.48.1.8-42667	10.48.1.8:42667	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20240628112704-10.48.2.11-46029	10.48.2.11:46029	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Running Drivers (0)

Submission ID	Submitted Time	Worker	State	Cores	Memory	Resources	Main Class	Duration

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20240628114458-0000	JavaWordCount	2	1024.0 MiB		2024/06/28 11:44:58	spark	FINISHED	36 s

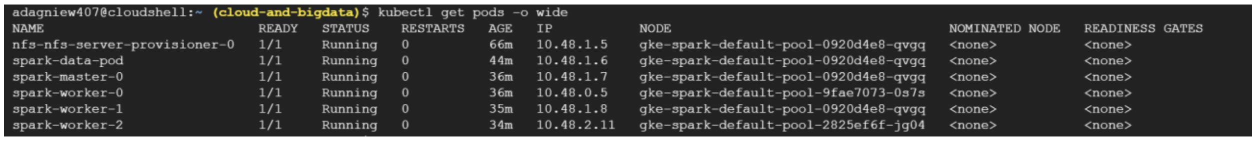
Completed Drivers (4)

Submission ID	Submitted Time	Worker	State	Cores	Memory	Resources	Main Class
driver-20240628114440-0003	2024/06/28 11:44:40	worker-20240628112512-10.48.0.5-35883	FINISHED	1	1024.0 MiB		org.apache.spark.examples.JavaWordCount
driver-20240628113558-0002	2024/06/28 11:35:58	worker-20240628112704-10.48.2.11-46029	FAILED	1	1024.0 MiB		org.apache.spark.examples.JavaWordCount
driver-20240628113505-0001	2024/06/28 11:35:05	worker-20240628112512-10.48.0.5-35883	FAILED	1	1024.0 MiB		org.apache.spark.examples.JavaWordCount
driver-20240628113030-0000	2024/06/28 11:30:30	worker-20240628112547-10.48.1.8-42667	FAILED	1	1024.0 MiB		org.apache.spark.examples.JavaWordCount

As you can see on the image the job is finished now lets check our output file.

13. **View the Output of the Completed Jobs:**
    - **Find Worker Node IP Address:**

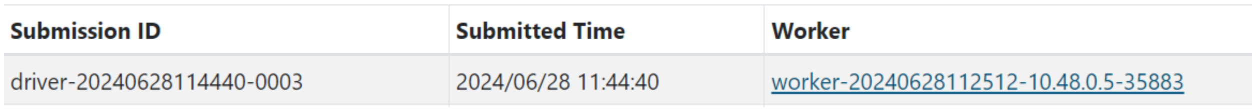
```
kubectl get pods -o wide | grep
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl get pods -o wide
NAME                          READY   STATUS    RESTARTS   AGE   IP           NODE                                     NOMINATED NODE   READINESS GATES
nfs-nfs-server-provisioner-0  1/1     Running   0          66m   10.48.1.5    gke-spark-default-pool-0920d4e8-qvgq     <none>           <none>
spark-data-pod                1/1     Running   0          44m   10.48.1.6    gke-spark-default-pool-0920d4e8-qvgq     <none>           <none>
spark-master-0                1/1     Running   0          36m   10.48.1.7    gke-spark-default-pool-0920d4e8-qvgq     <none>           <none>
spark-worker-0                1/1     Running   0          36m   10.48.0.5    gke-spark-default-pool-9fae7073-0s7s     <none>           <none>
spark-worker-1                1/1     Running   0          35m   10.48.1.8    gke-spark-default-pool-0920d4e8-qvgq     <none>           <none>
spark-worker-2                1/1     Running   0          34m   10.48.2.11   gke-spark-default-pool-2825ef6f-jg04     <none>           <none>
```

This command retrieves the IP address of the worker node that processed the word count task which is 10.48.0.5 in my case which we can see it in the website as well. The name is spark-worker-0

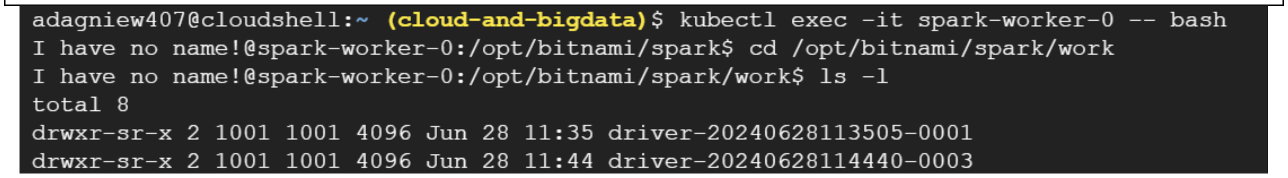
```
kubectl exec -it spark-worker-0 – bash
```



Submission ID	Submitted Time	Worker
driver-20240628114440-0003	2024/06/28 11:44:40	worker-20240628112512-10.48.0.5-35883

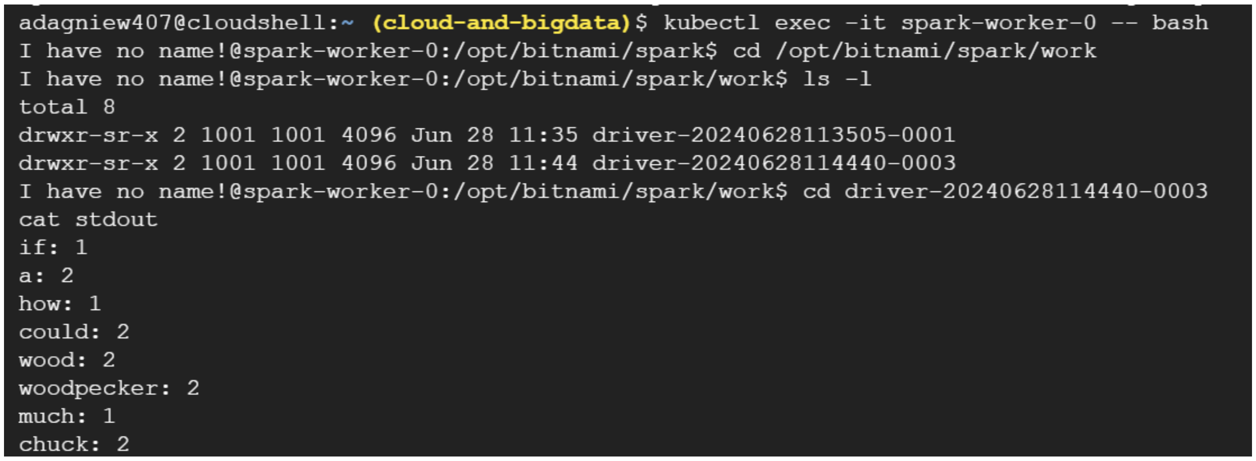
Execute the Pod and See the Result:

```
kubectl exec -it spark-worker-0 -- bash
cd /opt/bitnami/spark/work
ls -l
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl exec -it spark-worker-0 -- bash
I have no name!@spark-worker-0:/opt/bitnami/spark$ cd /opt/bitnami/spark/work
I have no name!@spark-worker-0:/opt/bitnami/spark/work$ ls -l
total 8
drwxr-sr-x 2 1001 1001 4096 Jun 28 11:35 driver-20240628113505-0001
drwxr-sr-x 2 1001 1001 4096 Jun 28 11:44 driver-20240628114440-0003
```

```
cd driver-20240628114440-0003
cat stdout
```



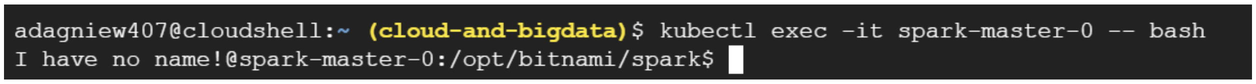
```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl exec -it spark-worker-0 -- bash
I have no name!@spark-worker-0:/opt/bitnami/spark$ cd /opt/bitnami/spark/work
I have no name!@spark-worker-0:/opt/bitnami/spark/work$ ls -l
total 8
drwxr-sr-x 2 1001 1001 4096 Jun 28 11:35 driver-20240628113505-0001
drwxr-sr-x 2 1001 1001 4096 Jun 28 11:44 driver-20240628114440-0003
I have no name!@spark-worker-0:/opt/bitnami/spark/work$ cd driver-20240628114440-0003
cat stdout
if: 1
a: 2
how: 1
could: 2
wood: 2
woodpecker: 2
much: 1
chuck: 2
```

These commands allow you to access the worker node pod and view the result of the word count task by reading the `stdout` file.

Running PageRank on PySpark

14. **Execute the Spark Master Pods:**

```
kubectl exec -it spark-master-0 – bash
```

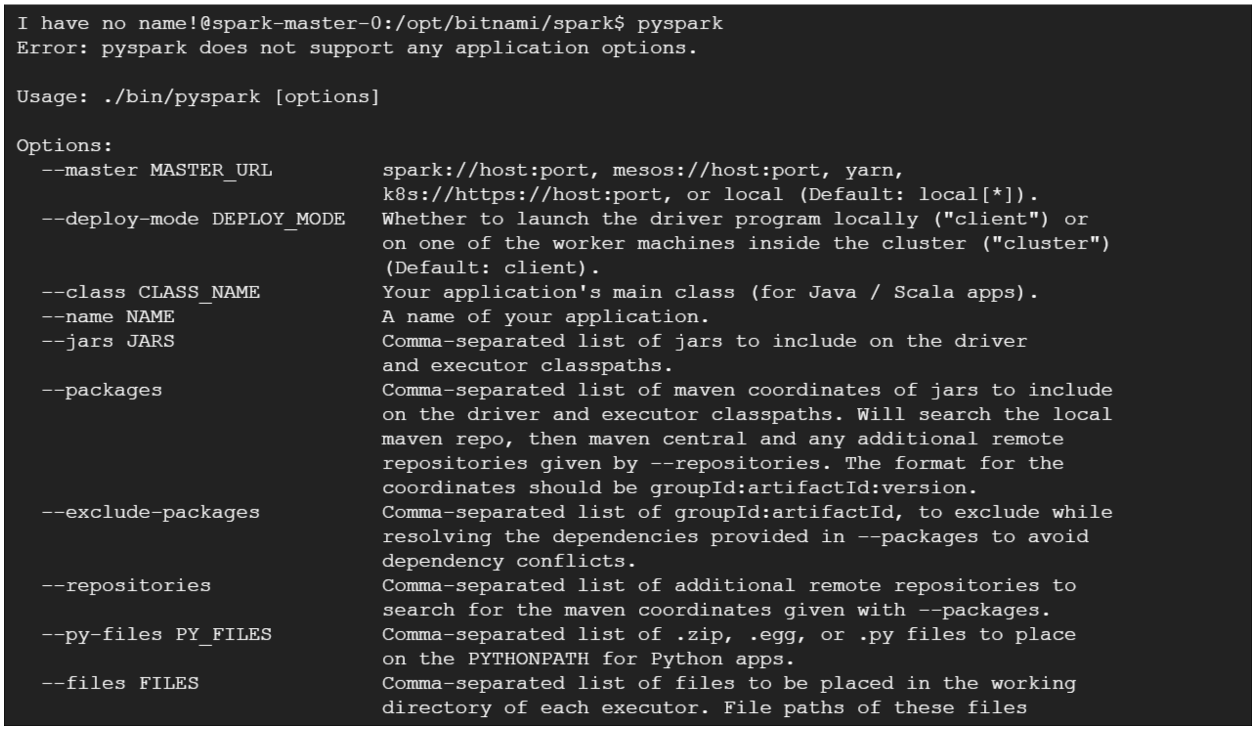


```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl exec -it spark-master-0 -- bash
I have no name!@spark-master-0:/opt/bitnami/spark$
```

This command provides access to the Spark master pod's shell, allowing you to run further commands.

15. **Start PySpark:**

```
pyspark
```



```
I have no name!@spark-master-0:/opt/bitnami/spark$ pyspark
Error: pyspark does not support any application options.

Usage: ./bin/pyspark [options]

Options:
  --master MASTER_URL         spark://host:port, mesos://host:port, yarn,
                              k8s://https://host:port, or local (Default: local[*]).
  --deploy-mode DEPLOY_MODE   Whether to launch the driver program locally ("client") or
                              on one of the worker machines inside the cluster ("cluster")
                              (Default: client).
  --class CLASS_NAME          Your application's main class (for Java / Scala apps).
  --name NAME                 A name of your application.
  --jars JARS                 Comma-separated list of jars to include on the driver
                              and executor classpaths.
  --packages                  Comma-separated list of maven coordinates of jars to include
                              on the driver and executor classpaths. Will search the local
                              maven repo, then maven central and any additional remote
                              repositories given by --repositories. The format for the
                              coordinates should be groupId:artifactId:version.
  --exclude-packages          Comma-separated list of groupId:artifactId, to exclude while
                              resolving the dependencies provided in --packages to avoid
                              dependency conflicts.
  --repositories              Comma-separated list of additional remote repositories to
                              search for the maven coordinates given with --packages.
  --py-files PY_FILES         Comma-separated list of .zip, .egg, or .py files to place
                              on the PYTHONPATH for Python apps.
  --files FILES               Comma-separated list of files to be placed in the working
                              directory of each executor. File paths of these files
```

If you face the above issue solution is available in the below github link:

https://github.com/bitnami/containers/issues/38139#issuecomment-1600923429

It seems to be the `--name` argument that is causing the issue in script: `/opt/bitnami/spark/bin/pyspark` - line 68:

```
exec "${SPARK_HOME}"/bin/spark-submit pyspark-shell-main --name "PySparkShell" "$@"
```

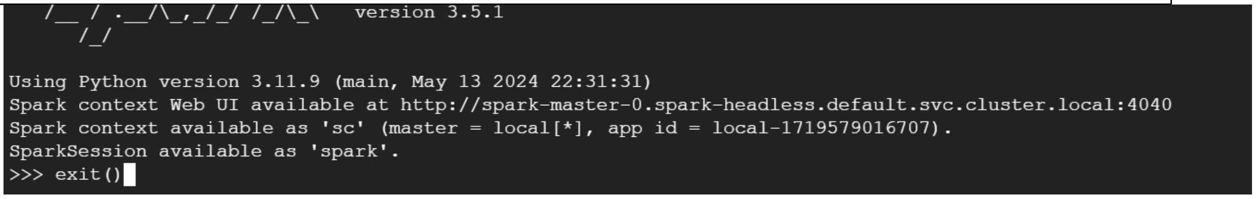
When I run the steps of that script manually without the `--name` arg, I can get an interactive PySpark shell:

```
export PYTHONPATH=/opt/bitnami/spark/python/lib/py4j-0.10.9.7-
src.zip:/opt/bitnami/spark/python/:/opt/bitnami/spark/python/:
export PYTHONSTARTUP=/opt/bitnami/spark/python/pyspark/shell.py
exec "${SPARK_HOME}"/bin/spark-submit pyspark-shell-main
```

This command starts the PySpark shell within the Spark master pod.

16. **Exit PySpark:**

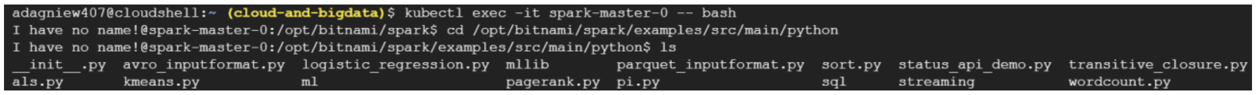
```
exit()
```



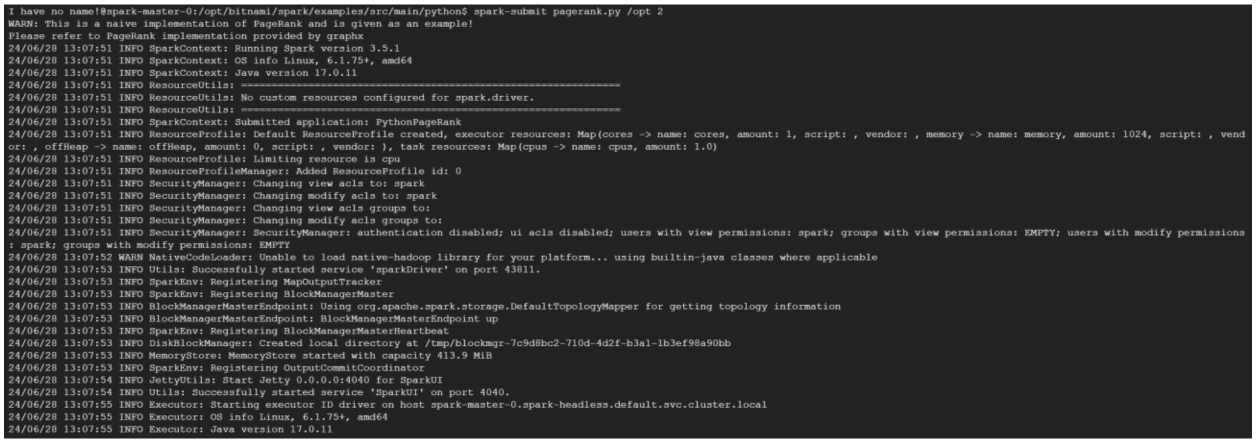
```
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.1
      /_/

Using Python version 3.11.9 (main, May 13 2024 22:31:31)
Spark context Web UI available at http://spark-master-0.spark-headless.default.svc.cluster.local:4040
Spark context available as 'sc' (master = local[*], app id = local-1719579016707).
SparkSession available as 'spark'.
>>> exit()
```

This command exits the PySpark shell.

17. **Go to the Directory Containing `pagerank.py`:**



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ kubectl exec -it spark-master-0 -- bash
I have no name!@spark-master-0:/opt/bitnami/spark$ cd /opt/bitnami/spark/examples/src/main/python
I have no name!@spark-master-0:/opt/bitnami/spark/examples/src/main/python$ ls
__init__.py  avro_inputformat.py  logistic_regression.py  mllib        parquet_inputformat.py  sort.py  status_api_demo.py  transitive_closure.py
als.py       kmeans.py            ml                      pagerank.py  pi.py                   sql      streaming           wordcount.py
```



```
I have no name!@spark-master-0:/opt/bitnami/spark/examples/src/main/python$ spark-submit pagerank.py /opt 2
WARN: This is a naive implementation of PageRank and is given as an example!
Please refer to PageRank implementation provided by graphx
24/06/28 13:07:51 INFO SparkContext: Running Spark version 3.5.1
24/06/28 13:07:51 INFO SparkContext: OS info Linux, 6.1.75+, amd64
24/06/28 13:07:51 INFO SparkContext: Java version 17.0.11
24/06/28 13:07:51 INFO ResourceUtils: ==============================================================
24/06/28 13:07:51 INFO ResourceUtils: No custom resources configured for spark.driver.
24/06/28 13:07:51 INFO ResourceUtils: ==============================================================
24/06/28 13:07:51 INFO SparkContext: Submitted application: PythonPageRank
24/06/28 13:07:51 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vend
or: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/06/28 13:07:51 INFO ResourceProfile: Limiting resource is cpu
24/06/28 13:07:51 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/06/28 13:07:51 INFO SecurityManager: Changing view acls to: spark
24/06/28 13:07:51 INFO SecurityManager: Changing modify acls to: spark
24/06/28 13:07:51 INFO SecurityManager: Changing view acls groups to:
24/06/28 13:07:51 INFO SecurityManager: Changing modify acls groups to:
24/06/28 13:07:51 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: spark; groups with view permissions: EMPTY; users with modify permissions
: spark; groups with modify permissions: EMPTY
24/06/28 13:07:52 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/06/28 13:07:53 INFO Utils: Successfully started service 'sparkDriver' on port 43811.
24/06/28 13:07:53 INFO SparkEnv: Registering MapOutputTracker
24/06/28 13:07:53 INFO SparkEnv: Registering BlockManagerMaster
24/06/28 13:07:53 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/06/28 13:07:53 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/06/28 13:07:53 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/06/28 13:07:53 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-7c9d8bc2-710d-4d2f-b3a1-1b3ef98a90bb
24/06/28 13:07:53 INFO MemoryStore: MemoryStore started with capacity 413.9 MiB
24/06/28 13:07:53 INFO SparkEnv: Registering OutputCommitCoordinator
24/06/28 13:07:54 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/06/28 13:07:54 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/06/28 13:07:55 INFO Executor: Starting executor ID driver on host spark-master-0.spark-headless.default.svc.cluster.local
24/06/28 13:07:55 INFO Executor: OS info Linux, 6.1.75+, amd64
24/06/28 13:07:55 INFO Executor: Java version 17.0.11
```

```
cd /opt/bitnami/spark/examples/src/main/python
```

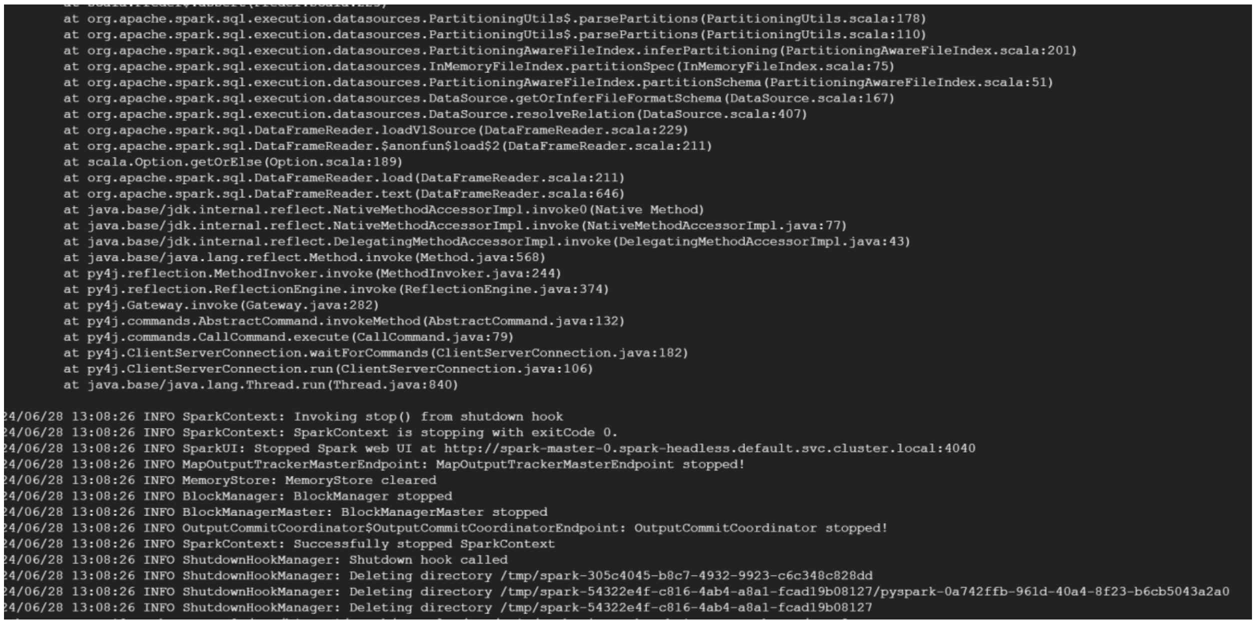
This command navigates to the directory where the `pagerank.py` script is located.

18. **Run the PageRank Using PySpark:**

Execute the Spark Master Pods:

```
kubectl exec -it spark-master-0 – bash

spark-submit pagerank.py /opt 2
```

This command runs the PageRank algorithm using the PySpark script, with `/opt` as the input directory and `2` as the number of iterations. You can modify these parameters as needed.



```
        at org.apache.spark.sql.execution.datasources.PartitioningUtils$.parsePartitions(PartitioningUtils.scala:178)
        at org.apache.spark.sql.execution.datasources.PartitioningUtils$.parsePartitions(PartitioningUtils.scala:110)
        at org.apache.spark.sql.execution.datasources.PartitioningAwareFileIndex.inferPartitioning(PartitioningAwareFileIndex.scala:201)
        at org.apache.spark.sql.execution.datasources.InMemoryFileIndex.partitionSpec(InMemoryFileIndex.scala:75)
        at org.apache.spark.sql.execution.datasources.PartitioningAwareFileIndex.partitionSchema(PartitioningAwareFileIndex.scala:51)
        at org.apache.spark.sql.execution.datasources.DataSource.getOrInferFileFormatSchema(DataSource.scala:167)
        at org.apache.spark.sql.execution.datasources.DataSource.resolveRelation(DataSource.scala:407)
        at org.apache.spark.sql.DataFrameReader.loadV1Source(DataFrameReader.scala:229)
        at org.apache.spark.sql.DataFrameReader.$anonfun$load$2(DataFrameReader.scala:211)
        at scala.Option.getOrElse(Option.scala:189)
        at org.apache.spark.sql.DataFrameReader.load(DataFrameReader.scala:211)
        at org.apache.spark.sql.DataFrameReader.text(DataFrameReader.scala:646)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
        at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.base/java.lang.reflect.Method.invoke(Method.java:568)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:374)
        at py4j.Gateway.invoke(Gateway.java:282)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:132)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.ClientServerConnection.waitForCommands(ClientServerConnection.java:182)
        at py4j.ClientServerConnection.run(ClientServerConnection.java:106)
        at java.base/java.lang.Thread.run(Thread.java:840)

24/06/28 13:08:26 INFO SparkContext: Invoking stop() from shutdown hook
24/06/28 13:08:26 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/06/28 13:08:26 INFO SparkUI: Stopped Spark web UI at http://spark-master-0.spark-headless.default.svc.cluster.local:4040
24/06/28 13:08:26 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/06/28 13:08:26 INFO MemoryStore: MemoryStore cleared
24/06/28 13:08:26 INFO BlockManager: BlockManager stopped
24/06/28 13:08:26 INFO BlockManagerMaster: BlockManagerMaster stopped
24/06/28 13:08:26 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/06/28 13:08:26 INFO SparkContext: Successfully stopped SparkContext
24/06/28 13:08:26 INFO ShutdownHookManager: Shutdown hook called
24/06/28 13:08:26 INFO ShutdownHookManager: Deleting directory /tmp/spark-305c4045-b8c7-4932-9923-c6c348c828dd
24/06/28 13:08:26 INFO ShutdownHookManager: Deleting directory /tmp/spark-54322e4f-c816-4ab4-a8a1-fcad19b08127/pyspark-0a742ffb-961d-40a4-8f23-b6cb5043a2a0
24/06/28 13:08:26 INFO ShutdownHookManager: Deleting directory /tmp/spark-54322e4f-c816-4ab4-a8a1-fcad19b08127
```