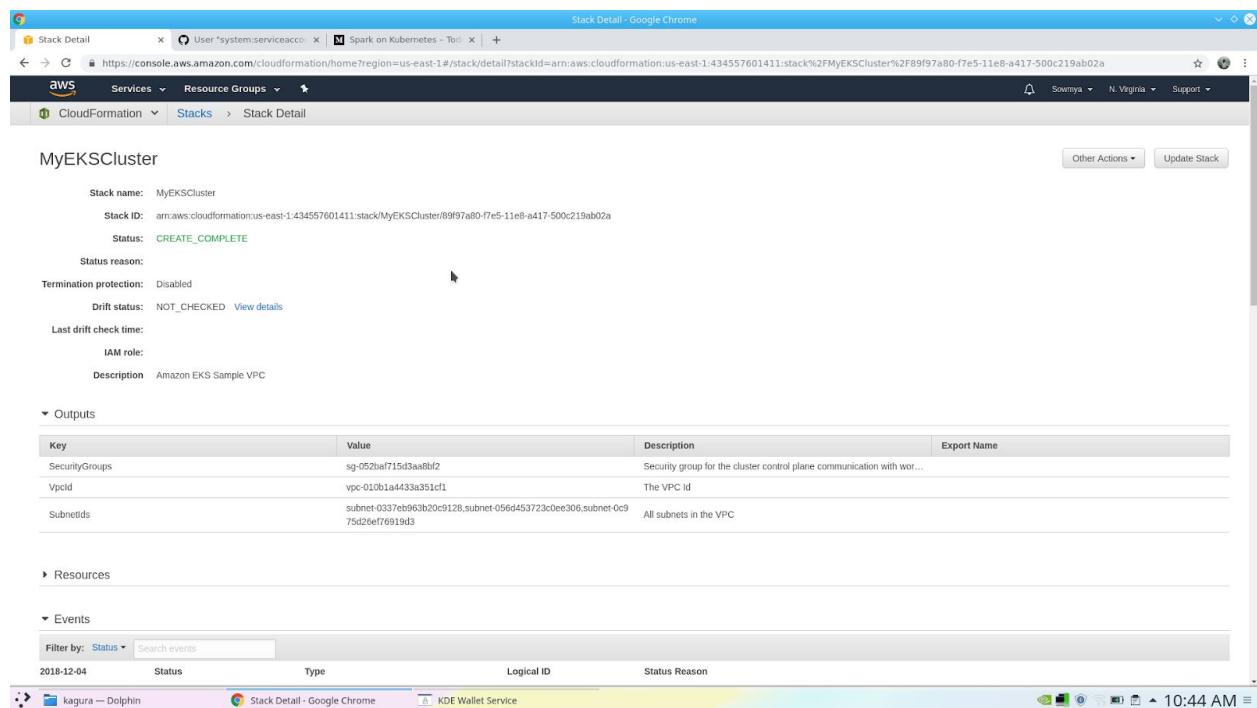


SPARK ON KUBERNETES - USING EKS AND MINIKUBE

EKS (Amazon Web Services' Managed Kubernetes Service)

Cluster VPC (Created using CloudFormation and an S3 Template URL)

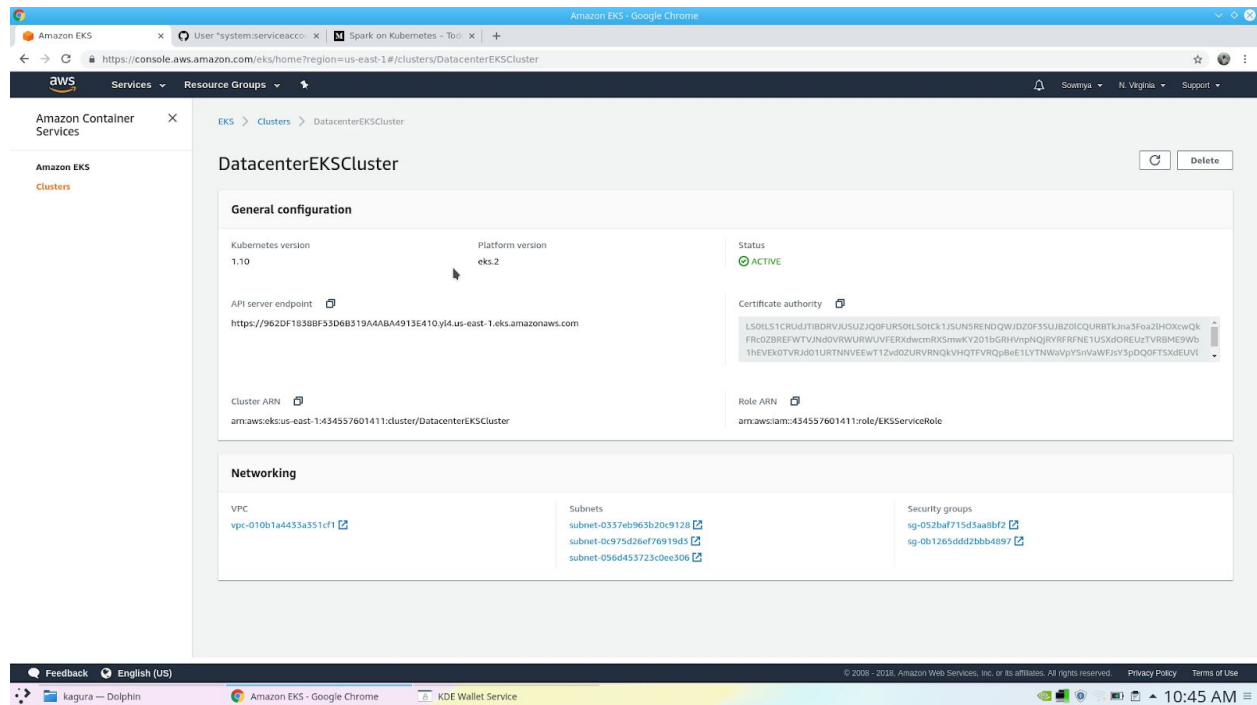


The screenshot displays the AWS CloudFormation console for a stack named 'MyEKSCluster'. The stack is in the 'CREATE_COMPLETE' status. The console shows various details including the stack name, ID, status, and reason. It also lists the outputs of the stack, which include the VPC ID, subnets, and security groups. The 'Outputs' section is expanded, showing a table with the following data:

Key	Value	Description	Export Name
SecurityGroups	sg-052ba715d3aa8bf2	Security group for the cluster control plane communication with wor...	
VpcId	vpc-010b1a4433a351ef1	The VPC id	
SubnetIds	subnet-0337eb963b20c9128,subnet-056d453723c0ee306,subnet-0c975d26ef76919d3	All subnets in the VPC	

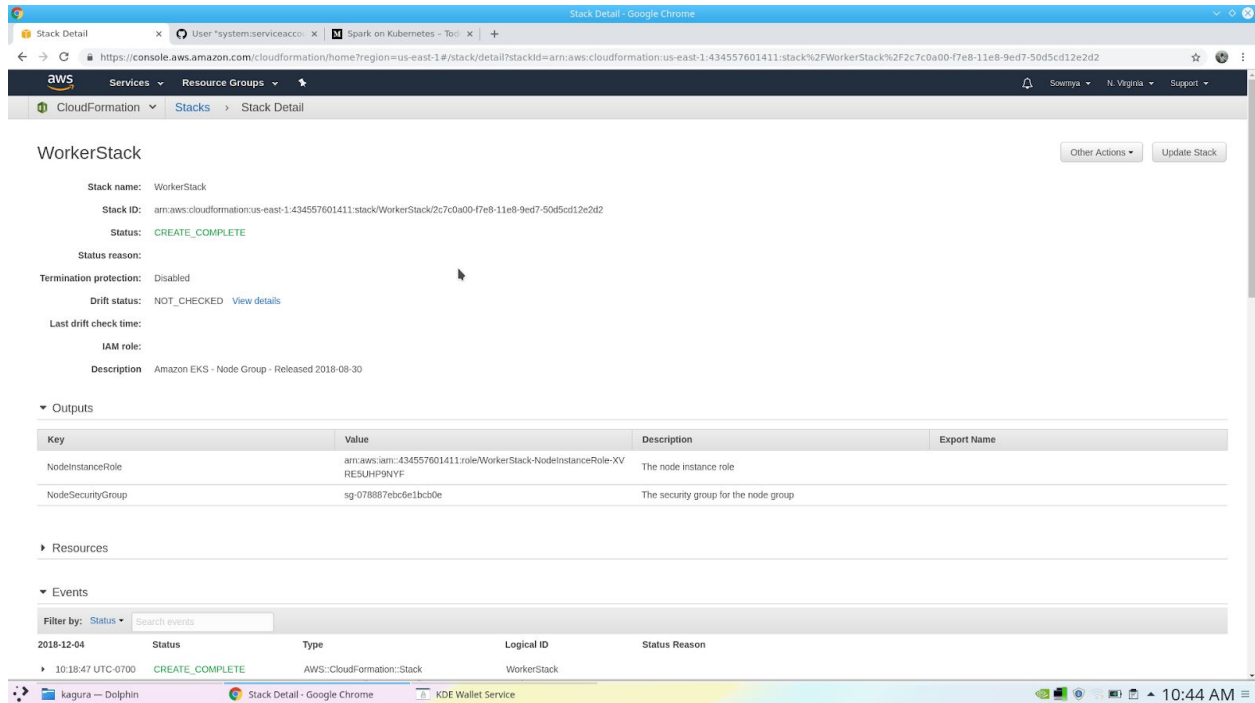
The 'Events' section is also visible, showing a list of events with columns for Date, Status, Type, Logical ID, and Status Reason. The first event is dated 2018-12-04 and has a status of 'Status'.

AWS EKS Cluster Plane (Created using EKS)



- Here, care should be taken to select appropriate VPC (created earlier through CloudFormation template) and thus, appropriate security groups.
- The console automatically shows the default VPC and its associated security groups, and creating the cluster with these will not serve up a running Kube-DNS pod and hence show TCP Connection Issues when the spark job is run. This is because, the default security groups do not give the cluster access to worker nodes. This is uniquely what the ControlPlaneSecurityGroup does, which is part of the Cluster VPC created.

Worker Nodes



The screenshot shows the AWS CloudFormation console for a stack named 'WorkerStack'. The stack is in the 'CREATE_COMPLETE' state. The console displays various details including the stack ID, status, termination protection, drift status, last drift check time, IAM role, and description. Below these details, there are sections for Outputs, Resources, and Events. The Outputs section shows two outputs: 'NodeInstanceRole' and 'NodeSecurityGroup'. The Resources section shows the stack itself. The Events section shows a list of events, including the 'CREATE_COMPLETE' event.

WorkerStack

Stack name: WorkerStack

Stack ID: arn:aws:cloudformation:us-east-1:434557601411:stack/WorkerStack/2c7c0a00-f7e8-11e8-9ed7-50d5cd12e2d2

Status: **CREATE_COMPLETE**

Status reason:

Termination protection: Disabled

Drift status: NOT_CHECKED [View details](#)

Last drift check time:

IAM role:

Description: Amazon EKS - Node Group - Released 2018-08-30

▼ Outputs

Key	Value	Description	Export Name
NodeInstanceRole	arn:aws:iam::434557601411:role/WorkerStack-NodeInstanceRole-XVRESUHP9NYF	The node instance role	
NodeSecurityGroup	sg-078887ebc8e1bc30e	The security group for the node group	

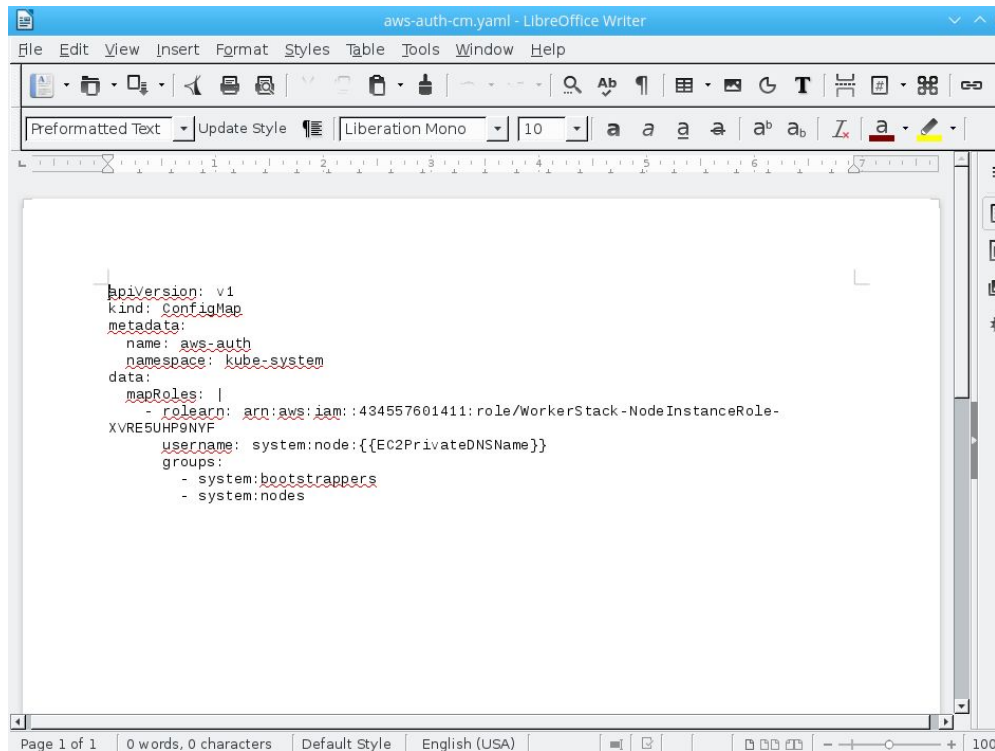
► Resources

▼ Events

Filter by: **Status**

2018-12-04	Status	Type	Logical ID	Status Reason
10:18:47 UTC-0700	CREATE_COMPLETE	AWS::CloudFormation::Stack	WorkerStack	

The Instance Role ARN is copied into the aws-auth-cm.yaml file.

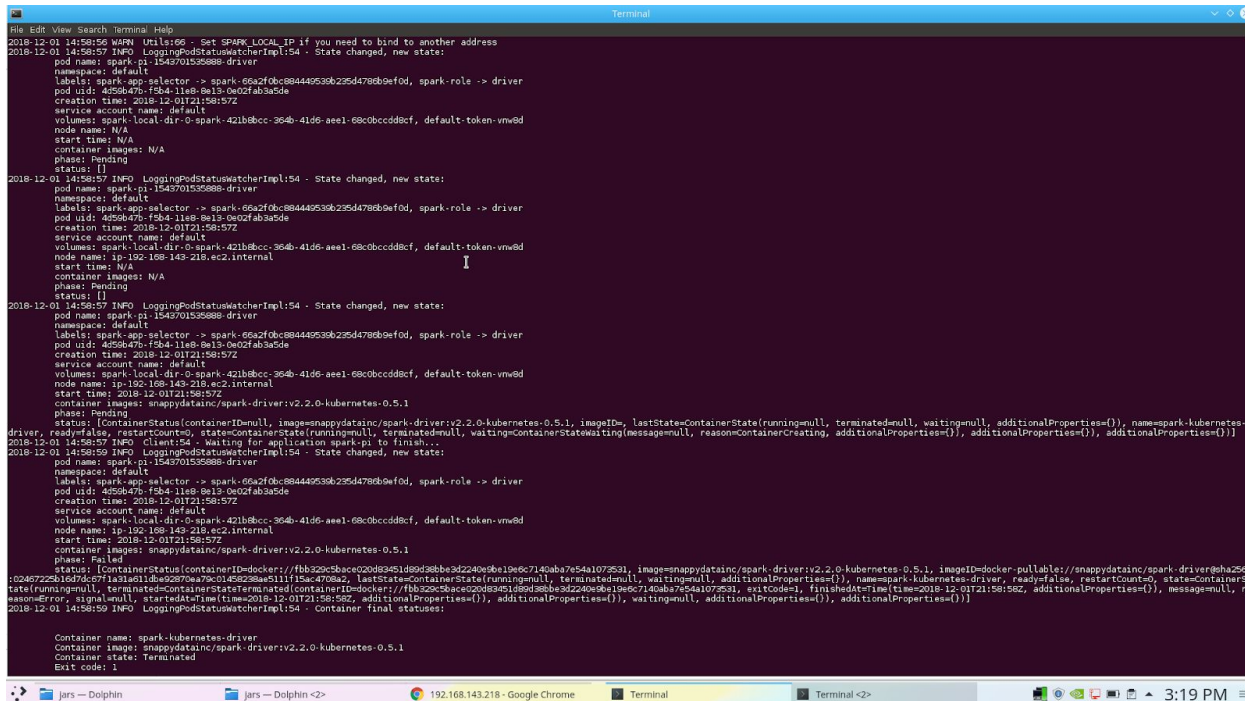


The screenshot shows a LibreOffice Writer document titled 'aws-auth-cm.yaml'. The document contains the following YAML content:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
data:
  mapRoles: |
    roleARN: arn:aws:iam::434557601411:role/WorkerStack-NodeInstanceRole-XVRESUHP9NYF
    username: system:node:{{EC2PrivateDNSName}}
    groups:
      - system:bootstrappers
      - system:nodes
```

The document is displayed in the LibreOffice Writer application, showing the menu bar, toolbar, and status bar.

Failure of EKS Spark Job



```
File Edit View Search Terminal Help
2018-12-01 14:58:56 WARN Utils:66 - Set SPARK_LOCAL_IP if you need to bind to another address
2018-12-01 14:58:57 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
  pod name: spark-pi-1543701535888-driver
  namespace: default
  labels: spark-app-selector -> spark-65a2f0bc804449535b225d478dbefed, spark-role -> driver
  pod uid: 4259b476-f3b4-11e8-9a13-0e02fab3a5de
  creation time: 2018-12-01T21:58:57Z
  service account name: default
  volumes: spark-local-dir-0-spark-421bb8cc-364b-41d6-aeel-68c0bccddcf, default-token-vmw6d
  node name: N/A
  start time: N/A
  container images: N/A
  phase: Pending
  status: {}
2018-12-01 14:58:57 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
  pod name: spark-pi-1543701535888-driver
  namespace: default
  labels: spark-app-selector -> spark-65a2f0bc804449535b225d478dbefed, spark-role -> driver
  pod uid: 4259b476-f3b4-11e8-9a13-0e02fab3a5de
  creation time: 2018-12-01T21:58:57Z
  service account name: default
  volumes: spark-local-dir-0-spark-421bb8cc-364b-41d6-aeel-68c0bccddcf, default-token-vmw6d
  node name: ip-192-168-143-218.ec2.internal
  start time: N/A
  container images: N/A
  phase: Pending
  status: {}
2018-12-01 14:58:57 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
  pod name: spark-pi-1543701535888-driver
  namespace: default
  labels: spark-app-selector -> spark-65a2f0bc804449535b225d478dbefed, spark-role -> driver
  pod uid: 4259b476-f3b4-11e8-9a13-0e02fab3a5de
  creation time: 2018-12-01T21:58:57Z
  service account name: default
  volumes: spark-local-dir-0-spark-421bb8cc-364b-41d6-aeel-68c0bccddcf, default-token-vmw6d
  node name: ip-192-168-143-218.ec2.internal
  start time: 2018-12-01T21:58:57Z
  container images: snappydatainc/spark-driver:v2.2.0-kubernetes-0.5.1
  phase: Pending
  status: {ContainerStatus{containerId=null, image=snappydatainc/spark-driver:v2.2.0-kubernetes-0.5.1, imageId=, lastState=ContainerState{running=null, terminated=null, waiting=null, additionalProperties={}}, name=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState{running=null, terminated=null, waiting=ContainerStateWaiting{message=null, reason=ContainerCreating, additionalProperties={}}, additionalProperties={}}}
2018-12-01 14:58:57 INFO client:54 - Waiting for application spark-pi to finish..
2018-12-01 14:58:59 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
  pod name: spark-pi-1543701535888-driver
  namespace: default
  labels: spark-app-selector -> spark-65a2f0bc804449535b225d478dbefed, spark-role -> driver
  pod uid: 4259b476-f3b4-11e8-9a13-0e02fab3a5de
  creation time: 2018-12-01T21:58:57Z
  service account name: default
  volumes: spark-local-dir-0-spark-421bb8cc-364b-41d6-aeel-68c0bccddcf, default-token-vmw6d
  node name: ip-192-168-143-218.ec2.internal
  start time: 2018-12-01T21:58:57Z
  container images: snappydatainc/spark-driver:v2.2.0-kubernetes-0.5.1
  phase: Failed
  status: {ContainerStatus{containerId=docker://fb322c5bace20d83451d9d38bbe3d2240e9b1e6c7140ba7e54a1073531, image=snappydatainc/spark-driver:v2.2.0-kubernetes-0.5.1, imageId=docker-pullable://snappydatainc/spark-driver@sha256:82672226b9f0c71a31a018b0c20e70e79c0149e22e8e11115c470842, lastState=ContainerState{running=null, terminated=null, waiting=null, additionalProperties={}}, name=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState{running=null, terminated=ContainerStateTerminated{containerId=docker://fb322c5bace20d83451d9d38bbe3d2240e9b1e6c7140ba7e54a1073531, exitCode=1, finishedAtTime=2018-12-01T21:58:58Z, additionalProperties={}}, message=null, reason=Error, signal=null, startedAtTime=2018-12-01T21:58:58Z, additionalProperties={}}, waiting=null, additionalProperties={}}, additionalProperties={}}}
2018-12-01 14:58:59 INFO LoggingPodStatusWatcherImpl:54 - Container final statuses:
  Container name: spark-kubernetes-driver
  Container image: snappydatainc/spark-driver:v2.2.0-kubernetes-0.5.1
  Container state: Terminated
  Exit code: 1
```

On observing the logs in the Kubernetes Dashboard, the following issues were observed and debugged:

- **Error: Could not find or load main class SparkPi** - This was resolved by uploading the jar file to a public S3 bucket and using the link in the spark-submit command.
- **MountVolume.SetUp failed for volume "spark-init-properties" : configmaps "spark-pi-f35e896061393afebbb78dc7f3e40a7e-init-config" not found** - This was resolved by creating cluster role bindings for default namespace.

Also, we tried the steps given in this github repo :

<https://github.com/SnappyDataInc/spark-on-k8s>

We used EKS, and in the Setup Helm Charts step, faced a major issue, which, till date, we haven't been able to rectify: After installing appropriate Helm version, and doing helm init (which installs Tiller), and after confirming that tiller is installed, when we tried helm ls or helm version, the client (helm) successfully showed up but the server (Tiller pod) stayed in ContainerCreating state and it hence said 'Could not find a ready tiller pod', which was weird. Anyhow, this approach was then abandoned and we were able to successfully create cluster and run the spark application the easier way!

Screenshots showing successful run:

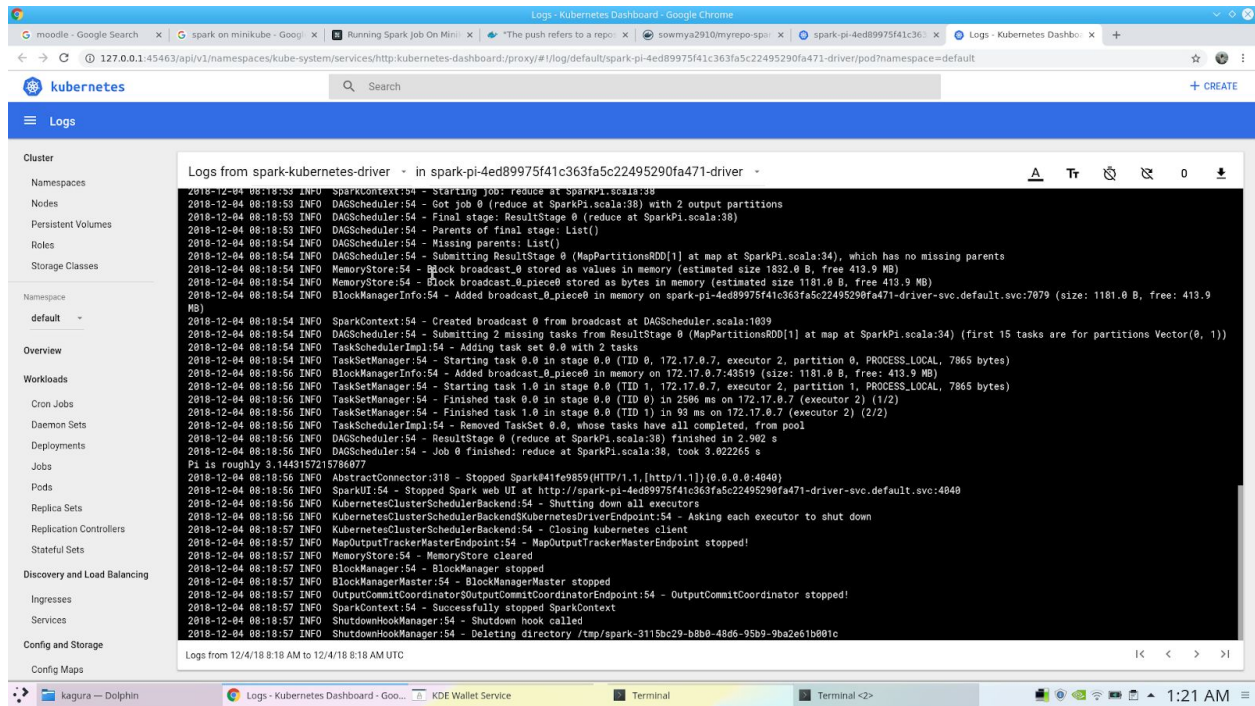
```

Hls Edit View Search Terminal Help
creation time: 2018-12-04T17:31:03Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-cz9zf
node name: ip-102-106-111-150.ec2.internal
start time: 2018-12-04T17:31:03Z
container images: gcr.io/cloud-solutions-images/spark:v2.3.0-gcs
phase: Pending
status: {ContainerStatus{containerId=null, image=gcr.io/cloud-solutions-images/spark:v2.3.0-gcs, imageP0m, lastState=ContainerState{running=null, terminated=null, waiting=null, additionalProperties={}}, names=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState{running=null, terminated=null, waiting=ContainerState{waiting(message=null, reasons=PodInitializing, additionalProperties={}), additionalProperties={}}, additionalProperties={}}}
2018-12-04 10:31:22 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-driver
namespace: default
labels: spark-app-selector -> spark-c80716456ddc41748d8745459e0bf41, spark-role -> driver
pod uid: 5ff493c7-77ea-11e8-a749-12b4823df2e
creation time: 2018-12-04T17:31:03Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-cz9zf
node name: ip-102-106-111-150.ec2.internal
start time: 2018-12-04T17:31:03Z
container images: gcr.io/cloud-solutions-images/spark:v2.3.0-gcs
phase: Pending
status: {ContainerStatus{containerId=null, image=gcr.io/cloud-solutions-images/spark:v2.3.0-gcs, imageP0m, lastState=ContainerState{running=null, terminated=null, waiting=null, additionalProperties={}}, names=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState{running=null, terminated=null, waiting=ContainerState{waiting(message=null, reasons=PodInitializing, additionalProperties={}), additionalProperties={}}, additionalProperties={}}}
2018-12-04 10:31:23 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-driver
namespace: default
labels: spark-app-selector -> spark-c80716456ddc41748d8745459e0bf41, spark-role -> driver
pod uid: 5ff493c7-77ea-11e8-a749-12b4823df2e
creation time: 2018-12-04T17:31:03Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-cz9zf
node name: ip-102-106-111-150.ec2.internal
start time: 2018-12-04T17:31:03Z
container images: gcr.io/cloud-solutions-images/spark:v2.3.0-gcs
phase: Running
status: {ContainerStatus{containerId=docker://06256ee241c3409e0ff3679a16c15607390c8433e22a21178f50ff774c, image=gcr.io/cloud-solutions-images/spark:v2.3.0-gcs, imageP0m=docker-pullable://gcr.io/cloud-solutions-images/spark:v2.3.0-gcs, ready=true, restartCount=0, state=ContainerState{running=true, terminated=false, waiting=false, additionalProperties={}}, names=spark-kubernetes-driver, ready=true, restartCount=0, state=ContainerState{running=ContainerState{running(startedAt=Time{time=2018-12-04T17:31:22Z, additionalProperties={}}, terminated=null, waiting=null, additionalProperties={}}, additionalProperties={}}}
2018-12-04 10:31:57 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-driver
namespace: default
labels: spark-app-selector -> spark-c80716456ddc41748d8745459e0bf41, spark-role -> driver
pod uid: 5ff493c7-77ea-11e8-a749-12b4823df2e
creation time: 2018-12-04T17:31:03Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-cz9zf
node name: ip-102-106-111-150.ec2.internal
start time: 2018-12-04T17:31:03Z
container images: gcr.io/cloud-solutions-images/spark:v2.3.0-gcs
phase: Succeeded
status: {ContainerStatus{containerId=docker://06256ee241c3409e0ff3679a16c15607390c8433e22a21178f50ff774c, image=gcr.io/cloud-solutions-images/spark:v2.3.0-gcs, imageP0m=docker-pullable://gcr.io/cloud-solutions-images/spark:v2.3.0-gcs, ready=false, restartCount=0, state=ContainerState{running=null, terminated=ContainerState{terminated(containerId=docker://06256ee241c3409e0ff3679a16c15607390c8433e22a21178f50ff774c, exitCode=0, finishedAt=Time{time=2018-12-04T17:31:56Z, additionalProperties={}), message=null, additionalProperties={}}, additionalProperties={}}}
2018-12-04 10:31:57 INFO LoggingPodStatusWatcherImpl:54 - Container final status:
Container name: spark-kubernetes-driver
Container image: gcr.io/cloud-solutions-images/spark:v2.3.0-gcs
Container state: Terminated
Exit code: 0
2018-12-04 10:31:57 INFO Client:54 - Application spark-pi finished.
2018-12-04 10:31:57 INFO ShutdownHookManager:54 - Shutdown hook called
2018-12-04 10:31:57 INFO ShutdownHookManager:54 - Deleting directory /tmp/spark-38b199e5-bb44-473b-9137-3060501bf108
You're up!

```

The screenshot displays the Kubernetes Dashboard interface. The top navigation bar includes the Kubernetes logo and a search bar. The left sidebar lists various cluster components, with 'Pods' currently selected. The main panel shows a table of pods. The table columns are Name, Node, Status, Restarts, and Age. The pods listed are all in a 'Terminated: Error' state, except for the first one which is 'Completed'. The error messages for the terminated pods indicate a failure to mount a volume due to a missing configmap. The bottom of the dashboard shows a pagination indicator '1 - 10 of 11' and navigation arrows.

Name	Node	Status	Restarts	Age
spark-pi-4ec89975f41c363fa5c22495290fa471-driver	minikube	Terminated: Completed	0	3 minutes
spark-pi-20a589bce636356ac15eba591c8ce67-driver	minikube	Terminated: Error	0	8 minutes
spark-pi-967aafde8f793b0ab0298ea52aca5f97-driver	minikube	Terminated: Error	0	9 minutes
spark-pi-00fda5a7f40332c29a3c1e0634581b7-driver	minikube	Terminated: Error	0	10 minutes
spark-pi-d362110b3f7b3f8999bdf895263705e-driver	minikube	Terminated: Error	0	13 minutes
spark-pi-ffda42ed88e431149cd61473520a734b-driver	minikube	Terminated: Error	0	15 minutes
spark-pi-983cc179ee630fb8666ca458d65d048-driver	minikube	Terminated: Error	0	15 minutes
spark-pi-e5ed988e62b733548f3bc53480a3a96-driver	minikube	Terminated: Error	0	21 minutes
spark-pi404ddab18a8540a02c3e29d82e2a979-driver	minikube	Terminated: Error	0	22 minutes
spark-pi-0a26539911b23778a1ae519db67d619d-driver	minikube	Terminated: Error	0	32 minutes

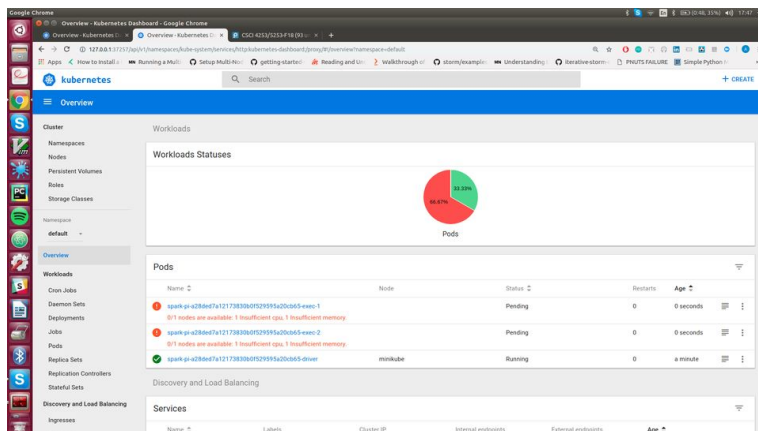


MINIKUBE

The issues we faced were similar to those observed during EKS implementation - Main class and configmaps, and were resolved the same way - use of S3 and creating cluster role bindings.

One other significant issue we faced was that of Insufficient CPU, Memory.

- **Insufficient CPU, Insufficient Memory:**



This was resolved by restarting kubectl pods (stopping and deleting minikube and starting it back up) - minikube stop ; minikube delete ; minikube start --memory 8192 --cpus 4

Minikube start

```
yorozuya% cd /usr/local/bin
yorozuya% minikube start --memory 8192 --cpus 4
Starting local Kubernetes v1.10.0 cluster...
Starting VM...
Getting VM IP address...
Moving files into cluster...
Setting up certs...
Connecting to cluster...
Setting up kubeconfig...
Starting cluster components...
Kubectl is now configured to use the cluster.
Loading cached images from config file.
yorozuya%
```

Creation of Spark Docker Image

```
File Edit View Search Terminal Help
yorozuya% sudo ./bin/docker-image-tool.sh -m -t spark-docker build
E1204 16:03:45.011959 4159 env.go:390] Error setting machine env variable(s): Error getting ip from host: machine does not exist
Sending build context to Docker daemon 258.1MB
Step 1/14 : FROM openjdk:8-alpine
--> 97bc1352afda
Step 2/14 : ARG spark_jars jars
--> Using cache
--> 4170cc39add5
Step 3/14 : ARG img_path=kubernetes/dockerfiles
--> Using cache
--> 44d0765c7723
Step 4/14 : RUN set -ex && apk upgrade --no-cache && apk add --no-cache bash tini libc6-compat && mkdir -p /opt/spark && mkdir -p /opt/spark/work-dir && touch /opt/spark/RELEASE && rm /bin/sh && ln -sv /bin/bash /bin/sh && chgrp root /etc/passwd && chmod u+rw /etc/passwd
--> Using cache
--> 0bf2192e4885
Step 5/14 : COPY $(spark_jars) /opt/spark/jars
--> Using cache
--> 4e45501ef0a0
Step 6/14 : COPY bin /opt/spark/bin
--> Using cache
--> 6d63c11f7a52
Step 7/14 : COPY sbin /opt/spark/sbin
--> Using cache
--> 35e003b20d36
Step 8/14 : COPY conf /opt/spark/conf
--> Using cache
--> 90w90w05430
Step 9/14 : COPY $(img_path)/spark/entrypoint.sh /opt/
--> Using cache
--> f05ac00c90w6
Step 10/14 : COPY examples /opt/spark/examples
--> Using cache
--> 0bwa28824250
Step 11/14 : COPY data /opt/spark/data
--> Using cache
--> cf06ba05f2a0
Step 12/14 : ENV SPARK_HOME /opt/spark
--> Using cache
--> c50754eb07a
Step 13/14 : WORKDIR /opt/spark/work-dir
--> Using cache
--> 79e56b957837
Step 14/14 : ENTRYPOINT /opt/entrypoint.sh
--> Using cache
--> 6191f7025982
Successfully built 6191f7025982
Successfully tagged spark:spark-docker
yorozuya%
```

```
File Edit View Search Terminal Help
yorozuya% sudo docker image ls
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
spark               spark-docker    6191f7025982   16 hours ago   346MB
sowmya2910/myrepo-spark v2.3.0         6191f7025982   16 hours ago   346MB
sowmya2910/myrepo/spark v2.3.0         6191f7025982   16 hours ago   346MB
<none>              <none>         2f066f1d4256   18 hours ago   346MB
openjdk             8-alpine       97bc1352afda   5 weeks ago    103MB
hello-world         latest         4ab4c602aa5e   2 months ago   1.84kB
yorozuya%
```

Building the image for push to docker hub repository

```
yorozuya% sudo ./bin/docker-image-tool.sh -t sowmya2910/myrepo2 -t v2.3.0 build
Sending build context to Docker daemon 258.1MB
Step 1/14 : FROM openjdk:8-alpine
--> 97bc1352afde
Step 2/14 : ARG spark_jars=jars
--> Using cache
--> 4176bc39add5
Step 3/14 : ARG img_path=kubernetes/dockerfiles
--> Using cache
--> ad4078545773
Step 4/14 : RUN set -ex && apk upgrade --no-cache && apk add --no-cache bash tini libc6-compat && mkdir -p /opt/spark && mkdir -p /opt/spark/work-dir && touch /opt/spark/RELEASE && rm /bin/sh && ln -sv /bin/bash /bin/sh && chgrp root /etc/passwd && chmod u+rw /etc/passwd
--> Using cache
--> bbf2192c4836
Step 5/14 : COPY ${spark_jars} /opt/spark/jars
--> Using cache
--> e045681ef0e0
Step 6/14 : COPY bin /opt/spark/bin
--> Using cache
--> d6dcb117382
Step 7/14 : COPY sbin /opt/spark/sbin
--> Using cache
--> 35e83b28d36
Step 8/14 : COPY conf /opt/spark/conf
--> Using cache
--> 90e960e05430
Step 9/14 : COPY ${img_path}/spark/entrypoint.sh /opt/
--> Using cache
--> f65d40c8060
Step 10/14 : COPY examples /opt/spark/examples
--> Using cache
--> 0bee28832436
Step 11/14 : COPY data /opt/spark/data
--> Using cache
--> cf6d0ca05f38
Step 12/14 : ENV SPARK_HOME /opt/spark
--> Using cache
--> c92754ebd57e
Step 13/14 : WORKDIR /opt/spark/work-dir
--> Using cache
--> 79e54b967837
Step 14/14 : ENTRYPOINT /opt/entrypoint.sh
--> Using cache
--> 6191f7025982
Successfully built 6191f7025982
Successfully tagged sowmya2910/myrepo2/spark:v2.3.0
yorozuya%
```

Docker hub login (for access to repository)

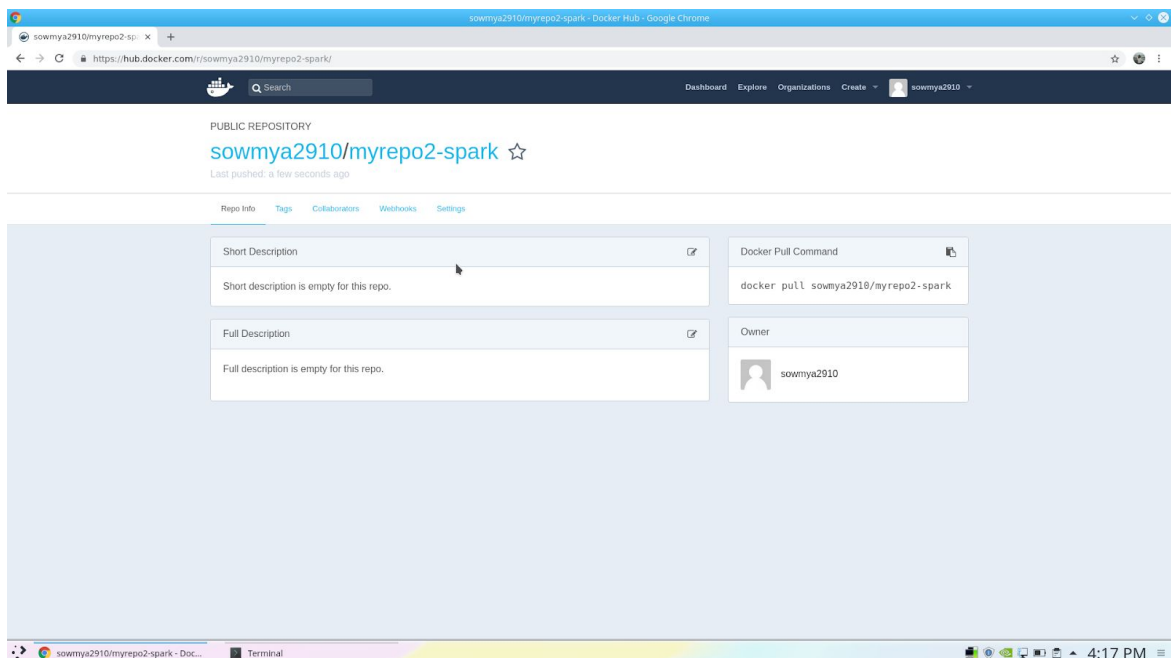
```
File Edit View Search Terminal Help
yorozuya% sudo docker login --username sowmya2910
Password:
Login Succeeded
yorozuya% █
```


Renaming repo and pushing image

```
yorozuya% sudo docker tag sowmya2910/myrepo2/spark sowmya2910/myrepo2-spark
Error response from daemon: No such image: sowmya2910/myrepo2/spark:latest
yorozuya% sudo docker tag sowmya2910/myrepo2/spark:v2.3.0 sowmya2910/myrepo2-spark:v2.3.0
yorozuya% sudo docker push sowmya2910/myrepo2-spark:v2.3.0
The push refers to a repository [docker.io/sowmya2910/myrepo2-spark]
fd99da662d66: Pushed
52f485c165ba: Pushed
781dedd9877c: Pushed
7fa21420da76: Pushed
f22a61e86d49: Pushed
fd4e89e918a4: Pushed
6e97b177306c: Pushed
4933ed0cc7e7: Pushed
ed6f0bd39121: Pushed
0c3170905795: Pushed
df64d3292fd6: Pushed
v2.3.0: digest: sha256:2fd2b4fbca902d9f87f120f6db857e967c28daf19a67eec7b62532762ebbcd7d size: 2624
yorozuya%
```

The need to rename was found as a solution to an issue, during which the push didn't happen since the repo could not be accessed (even after login). A workaround suggested that we rename the repo, (it apparently doesn't work if it has more than two parameters! (sowmya2910/myrepo2/spark) along with the image tag) and hence it was renamed as sowmya2910/myrepo2-spark, and it pushed.!

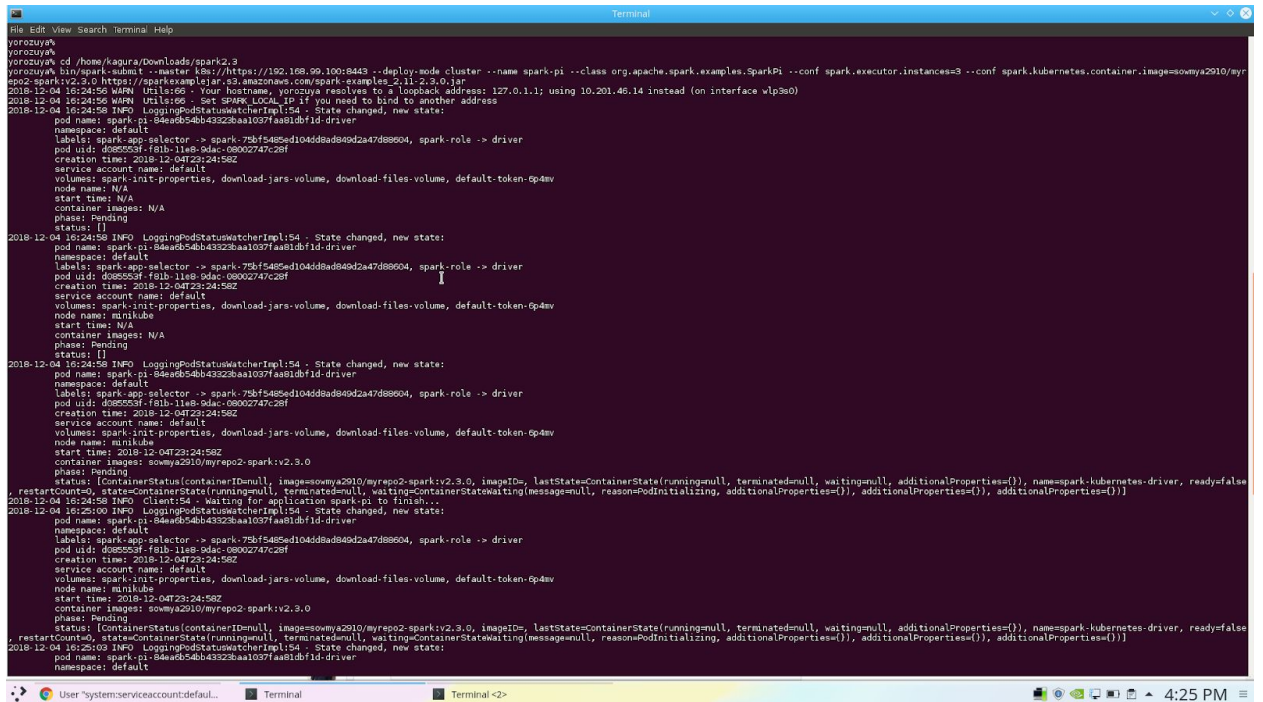
Screenshot showing repo creation and last push notification:



Creation of Cluster role bindings

```
yorozuya% kubectl create clusterrolebinding cluster-system-anonymous --clusterrole=cluster-admin --user=system:anonymous
clusterrolebinding.rbac.authorization.k8s.io/cluster-system-anonymous created
yorozuya% kubectl create clusterrolebinding default-admin --clusterrole=cluster-admin --serviceaccount=default:default
clusterrolebinding.rbac.authorization.k8s.io/default-admin created
yorozuya%
yorozuya%
yorozuya%
```

Screenshots showing successful run:



```
yorozuya% cd /home/kagura/Downloads/spark2.3
yorozuya% bin/spark-submit --master k8s://https://192.168.99.100:8443 --deploy-mode cluster --name spark-pi --class org.apache.spark.examples.SparkPi --conf spark.executor.instances=3 --conf spark.kubernetes.container.image=sowmya2010/myr
2018-12-04 16:24:56 WARN Utils:66 - Your hostname, yorozuya resolves to a loopback address: 127.0.0.1; using 10.201.46.14 instead (on interface wlp3s0)
2018-12-04 16:24:56 WARN Utils:66 - Set SPARK_LOCAL_IP if you need to bind to another address
2018-12-04 16:24:56 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-84eab54b433238aa1037aa81dbfd-driver
namespace: default
labels: spark-app-selector -> spark-75bf5485ed104dd8ad849d2a47d88604, spark-role -> driver
pod uid: d865553f-f81b-11e8-9dac-08002747c26f
creation times: 2018-12-04T23:24:58Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-6p4wv
node name: minikube
start time: N/A
container images: N/A
phase: Pending
status: {}
2018-12-04 16:24:58 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-84eab54b433238aa1037aa81dbfd-driver
namespace: default
labels: spark-app-selector -> spark-75bf5485ed104dd8ad849d2a47d88604, spark-role -> driver
pod uid: d865553f-f81b-11e8-9dac-08002747c26f
creation times: 2018-12-04T23:24:58Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-6p4wv
node name: minikube
start time: N/A
container images: N/A
phase: Pending
status: {}
2018-12-04 16:24:58 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-84eab54b433238aa1037aa81dbfd-driver
namespace: default
labels: spark-app-selector -> spark-75bf5485ed104dd8ad849d2a47d88604, spark-role -> driver
pod uid: d865553f-f81b-11e8-9dac-08002747c26f
creation times: 2018-12-04T23:24:58Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-6p4wv
node name: minikube
start time: 2018-12-04T23:24:58Z
container images: sowmya2010/myrepo2-spark:v2.3.0
phase: Pending
status: ContainerStatus(containerId=null, image=sowmya2010/myrepo2-spark:v2.3.0, imageID=, lastState=ContainerState(running=null, terminated=null, waiting=null, additionalProperties={}), name=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState(running=null, terminated=null, waiting=ContainerStateWaiting(message=null, reason=PodInitializing, additionalProperties={}), additionalProperties={}))
2018-12-04 16:24:58 INFO Client:54 - Waiting for application spark-pi to finish...
2018-12-04 16:25:00 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-84eab54b433238aa1037aa81dbfd-driver
namespace: default
labels: spark-app-selector -> spark-75bf5485ed104dd8ad849d2a47d88604, spark-role -> driver
pod uid: d865553f-f81b-11e8-9dac-08002747c26f
creation times: 2018-12-04T23:24:58Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-6p4wv
node name: minikube
start time: 2018-12-04T23:24:58Z
container images: sowmya2010/myrepo2-spark:v2.3.0
phase: Pending
status: ContainerStatus(containerId=null, image=sowmya2010/myrepo2-spark:v2.3.0, imageID=, lastState=ContainerState(running=null, terminated=null, waiting=null, additionalProperties={}), name=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState(running=null, terminated=null, waiting=ContainerStateWaiting(message=null, reason=PodInitializing, additionalProperties={}), additionalProperties={}))
2018-12-04 16:25:03 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-84eab54b433238aa1037aa81dbfd-driver
namespace: default
```

```
File Edit View Search Terminal Help
creation time: 2018-12-04T23:24:58Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-qp4wv
node name: minikube
start time: 2018-12-04T23:24:58Z
container images: sowmya2910/myrepo2-spark:v2.3.0
phase: Pending
status: ContainerStatus(containerIDnull, image=sowmya2910/myrepo2-spark:v2.3.0, imageID, lastState=ContainerState(runningnull, terminatednull, waitingnull, additionalProperties={}), name=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState(runningnull, terminatednull, waiting=ContainerStateWaiting(message=null, reason=PodInitializing, additionalProperties={}), additionalProperties={}))
2018-12-04 16:25:03 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver
namespace: default
labels: spark-app-selector -> spark-75bf5485ed104dd8a8d9d2a47d88604, spark-role -> driver
pod uid: d685953f-f81b-11e8-9dac-08002747c28f
creation time: 2018-12-04T23:24:58Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-qp4wv
node name: minikube
start time: 2018-12-04T23:24:58Z
container images: sowmya2910/myrepo2-spark:v2.3.0
phase: Pending
status: ContainerStatus(containerIDnull, image=sowmya2910/myrepo2-spark:v2.3.0, imageID, lastState=ContainerState(runningnull, terminatednull, waitingnull, additionalProperties={}), name=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState(runningnull, terminatednull, waiting=ContainerStateWaiting(message=null, reason=PodInitializing, additionalProperties={}), additionalProperties={}))
2018-12-04 16:25:04 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver
namespace: default
labels: spark-app-selector -> spark-75bf5485ed104dd8a8d9d2a47d88604, spark-role -> driver
pod uid: d685953f-f81b-11e8-9dac-08002747c28f
creation time: 2018-12-04T23:24:58Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-qp4wv
node name: minikube
start time: 2018-12-04T23:24:58Z
container images: sowmya2910/myrepo2-spark:v2.3.0
phase: Running
status: ContainerStatus(containerID=docker://b24d937ecc9272c19594d338a1951398d981d11d2f8ca5e9c30a35f966263, image=sowmya2910/myrepo2-spark:v2.3.0, imageID=docker-pullable://sowmya2910/myrepo2-spark@sha256:2f2db4fbc9a02d9f67f120f6d857e967c28d8f1a8e7ee7b6252726abb0d7d, lastState=ContainerState(runningnull, terminatednull, waitingnull, additionalProperties={}), name=spark-kubernetes-driver, ready=true, restartCount=0, state=ContainerState(running=ContainerStateRunning(startedAt=2018-12-04T23:25:03Z, additionalProperties={}), terminatednull, waitingnull, additionalProperties={}))
2018-12-04 16:25:43 INFO LoggingPodStatusWatcherImpl:54 - State changed, new state:
pod name: spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver
namespace: default
labels: spark-app-selector -> spark-75bf5485ed104dd8a8d9d2a47d88604, spark-role -> driver
pod uid: d685953f-f81b-11e8-9dac-08002747c28f
creation time: 2018-12-04T23:24:58Z
service account name: default
volumes: spark-init-properties, download-jars-volume, download-files-volume, default-token-qp4wv
node name: minikube
start time: 2018-12-04T23:24:58Z
container images: sowmya2910/myrepo2-spark:v2.3.0
phase: Succeeded
status: ContainerStatus(containerID=docker://b24d937ecc9272c19594d338a1951398d981d11d2f8ca5e9c30a35f966263, image=sowmya2910/myrepo2-spark:v2.3.0, imageID=docker-pullable://sowmya2910/myrepo2-spark@sha256:2f2db4fbc9a02d9f67f120f6d857e967c28d8f1a8e7ee7b6252726abb0d7d, lastState=ContainerState(runningnull, terminatednull, waitingnull, additionalProperties={}), name=spark-kubernetes-driver, ready=false, restartCount=0, state=ContainerState(running=ContainerStateRunning(startedAt=2018-12-04T23:25:03Z, additionalProperties={}), terminated=ContainerStateTerminated(containerID=docker://b24d937ecc9272c19594d338a1951398d981d11d2f8ca5e9c30a35f966263, exitCode=0, finishedAt=2018-12-04T23:25:42Z, additionalProperties={}), message=null, reason=Completed, signal=null, startedAt=2018-12-04T23:25:03Z, additionalProperties={}), additionalProperties={}))
2018-12-04 16:25:43 INFO LoggingPodStatusWatcherImpl:54 - Container final statuses:
Container name: spark-kubernetes-driver
Container image: sowmya2910/myrepo2-spark:v2.3.0
Container state: Terminated
Exit code: 0
2018-12-04 16:25:43 INFO Client:54 - Application spark-pi finished.
2018-12-04 16:25:43 INFO ShutdownHookManager:54 - Shutdown hook called
2018-12-04 16:25:43 INFO ShutdownHookManager:54 - Deleting directory /tmp/spark-186a6847-bcd5-4254-8859-ec440facbcb
jyorozyah
```

Overview - Kubernetes Dashboard - Google Chrome

127.0.0.1:32933/api/v1/namespaces/kube-system/services/http:kubernetes-dashboard/proxy/#/overview?namespace=default

kubernetes

Overview

Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Workloads Statuses

Pods

0.00% 100.00%

Pods

Name	Node	Status	Restarts	Age
spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver	minikube	Terminated: Completed	0	a minute
spark-pi-e251f75cb57d3215816b1b83c622e094-driver	minikube	Terminated: Error	0	5 minutes

MountVolume.SetUp failed for volume "spark-init-properties": configmaps "spark-pi-e251f75cb57d3215816b1b83c622e094-init-config" not found

Discovery and Load Balancing

Services

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
spark-pi-84ea6b54bb43323baa1037faa		None	spark-pi-84ea6b54bb43323baa1037faa	-	a minute
spark-pi-e251f75cb57d3215816b1b83c		None	spark-pi-e251f75cb57d3215816b1b83c	-	5 minutes
kubernetes	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP	-	25 minutes

The screenshot shows the Kubernetes Dashboard for a pod named `spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver`. The pod is in the `Succeeded` state. The 'Containers' section shows the `spark-kubernetes-driver` container with the following details:

- Image:** `sowmya2910/myrepo2-sparkv2.3.0`
- Environment variables:**
 - `SPARK_DRIVER_MEMORY`: `1g`
 - `SPARK_DRIVER_CLASS`: `org.apache.spark.examples.SparkPi`
 - `SPARK_DRIVER_ARGS`: `SPARK_DRIVER_BIND_ADDRESS: (v1:status.podIP)`
 - `SPARK_MOUNTED_CLASSPATH`: `/var/spark-data/spark-jars:/var/spark-examples_2.11-2.3.0.jar:/var/spark-data/spark-jars/spark-examples_2.11-2.3.0.jar`
 - `SPARK_MOUNTED_FILES_DIR`: `/var/spark-data/spark-files`
 - `SPARK_JAVA_OPTS_0`: `-Dspark.kubernetes.container.image=sowmya2910/myrepo2-sparkv2.3.0`
 - `SPARK_JAVA_OPTS_1`: `-Dspark.app.name=spark-pi`
 - `SPARK_JAVA_OPTS_2`: `-Dspark.kubernetes.driver.pod.name=spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver`
 - `SPARK_JAVA_OPTS_3`: `-Dspark.executor.instances`
 - `SPARK_JAVA_OPTS_4`: `-Dspark.submit.deployMode=cluster`
 - `SPARK_JAVA_OPTS_5`: `-Dspark.driver.blockManager.port=7079`
 - `SPARK_JAVA_OPTS_6`: `-Dspark.driver.port=7078`
 - `SPARK_JAVA_OPTS_7`: `-Dspark.master=k8s://https://192.168.99.100:8443`
 - `SPARK_JAVA_OPTS_8`: `-Dspark.kubernetes.initContainer.configMapKey=spark-init.properties`
 - `SPARK_JAVA_OPTS_9`: `-Dspark.kubernetes.initContainer.configMapName=spark-pi-84ea6b54bb43323baa1037faa81dbf1d-init-config`
 - `SPARK_JAVA_OPTS_10`: `-Dspark.kubernetes.executor.podNamePrefix=spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver`
 - `SPARK_JAVA_OPTS_11`: `-Dspark.driver.host=spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver-svc.default.svc`
 - `SPARK_JAVA_OPTS_12`: `-Dspark.app.id=spark-75bf5485ed104dd8a849d2a47d88604`
 - `SPARK_JAVA_OPTS_13`: `-Dspark.jars=https://sparkxamplejar.s3.amazonaws.com/spark-examples_2.11-2.3.0.jar`

The screenshot shows the logs of the `spark-kubernetes-driver` pod. The logs show the execution of the SparkPi job, including the DAGScheduler, TaskSetManager, and BlockManagerInfo. The logs are as follows:

```
2018-12-04 23:25:37 INFO DAGScheduler:54 - Got job 0 (reduce at SparkPi.scala:38) with 2 output partitions
2018-12-04 23:25:37 INFO DAGScheduler:54 - Final stage: ResultStage 0 (reduce at SparkPi.scala:38)
2018-12-04 23:25:37 INFO DAGScheduler:54 - Parents of final stage: List()
2018-12-04 23:25:37 INFO DAGScheduler:54 - Missing parents: List()
2018-12-04 23:25:37 INFO DAGScheduler:54 - Submitting ResultStage 0 (MapPartitionsRDD[1] at map at SparkPi.scala:34), which has no missing parents
2018-12-04 23:25:37 INFO MemoryStore:54 - Block broadcast_0 stored as values in memory (estimated size 1832.0 B, free 413.9 MB)
2018-12-04 23:25:37 INFO MemoryStore:54 - Block broadcast_0_piece0 stored as bytes in memory (estimated size 1181.0 B, free 413.9 MB)
2018-12-04 23:25:37 INFO BlockManagerInfo:54 - Added broadcast_0_piece0 in memory on spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver-svc.default.svc:7079 (size: 1181.0 B, free: 413.9 MB)
2018-12-04 23:25:37 INFO SparkContext:54 - Created broadcast 0 from broadcast at DAGScheduler.scala:1039
2018-12-04 23:25:37 INFO DAGScheduler:54 - Submitting 2 missing tasks from ResultStage 0 (MapPartitionsRDD[1] at map at SparkPi.scala:34) (first 15 tasks are for partitions Vector(0, 1))
2018-12-04 23:25:37 INFO TaskSchedulerImpl:54 - Adding task set 0.0 with 2 tasks
2018-12-04 23:25:37 INFO TaskSetManager:54 - Starting task 0.0 in stage 0.0 (TID 0, 172.17.0.6, executor 2, partition 0, PROCESS_LOCAL, 7865 bytes)
2018-12-04 23:25:37 INFO TaskSetManager:54 - Starting task 1.0 in stage 0.0 (TID 1, 172.17.0.7, executor 1, partition 1, PROCESS_LOCAL, 7865 bytes)
2018-12-04 23:25:41 INFO BlockManagerInfo:54 - Added broadcast_0_piece0 in memory on 172.17.0.7:43845 (size: 1181.0 B, free: 413.9 MB)
2018-12-04 23:25:41 INFO TaskSetManager:54 - Finished task 1.0 in stage 0.0 (TID 1) in 3967 ms on 172.17.0.7 (executor 1) (1/2)
2018-12-04 23:25:41 INFO BlockManagerInfo:54 - Added broadcast_0_piece0 in memory on 172.17.0.6:46173 (size: 1181.0 B, free: 413.9 MB)
2018-12-04 23:25:42 INFO TaskSetManager:54 - Finished task 0.0 in stage 0.0 (TID 0) in 4735 ms on 172.17.0.6 (executor 2) (2/2)
2018-12-04 23:25:42 INFO DAGScheduler:54 - ResultStage 0 (reduce at SparkPi.scala:38) finished in 4.947 s
2018-12-04 23:25:42 INFO TaskSchedulerImpl:54 - Removed TaskSet 0.0, whose tasks have all completed, from pool
2018-12-04 23:25:42 INFO DAGScheduler:54 - Job 0 finished: reduce at SparkPi.scala:38, took 5.470111 s
Pi is roughly 3.142323716178581
2018-12-04 23:25:42 INFO AbstractConnector:318 - Stopped Spark84e76dc(HTTP/1.1, [http://172.17.0.0:8040])
2018-12-04 23:25:42 INFO SparkUI:54 - Stopped Spark web UI at http://spark-pi-84ea6b54bb43323baa1037faa81dbf1d-driver-svc.default.svc:4040
2018-12-04 23:25:42 INFO KubernetesClusterSchedulerBackend:54 - Shutting down all executors
2018-12-04 23:25:42 INFO KubernetesClusterSchedulerBackend:54 - Closing Kubernetes client
2018-12-04 23:25:42 INFO MapOutputTrackerMasterEndpoint:54 - MapOutputTrackerMasterEndpoint stopped!
2018-12-04 23:25:42 INFO MemoryStore:54 - MemoryStore cleared
2018-12-04 23:25:42 INFO BlockManager:54 - BlockManager stopped
2018-12-04 23:25:42 INFO BlockManagerMaster:54 - BlockManagerMaster stopped
2018-12-04 23:25:42 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint:54 - OutputCommitCoordinator stopped!
2018-12-04 23:25:42 INFO SparkContext:54 - Successfully stopped SparkContext
2018-12-04 23:25:42 INFO ShutdownHookManager:54 - Shutdown hook called
2018-12-04 23:25:42 INFO ShutdownHookManager:54 - Deleting directory /tmp/spark-4345dd8a-a1f9-4ba2-918c-48c7d3a9a55e
```

Thus, we were successfully able to run spark jobs on clusters created by both EKS as well as Minikube. The process was a little herculean, with it taking more time than we expected and (too) many errors encountered along the way, but lots of support made sure we crossed the line! Lots of learning too. Next steps are to integrate MongoDB

database to Kubernetes and to find a way to transfer spark query results to the Django App!