

Hands-On Exercise 4-2 Building an Apache Spark cluster using Kubernetes

1 CScLuster at UIS

11-node Hadoop nodes

- CPU: Intel Xeon **4C/8T** per node
- Memory: **266 GB** in total
 - Master (head) node: 74GB RAM
 - 2nd Master node: 48GB RAM
 - 9 worker nodes: 16GB/node
- Storage: **10 TB SSD** in total
 - Master node: 400GB,
 - 10 worker nodes: 1TB SSD/node
- Hadoop: **Cludera CDH 6.3**
 - HDFS, MapReduce
 - Spark 2, HBase, Hive
 - Pig, HUE, and etc.

VMs for Docker/K8S (You will use these VMs)

- CPU: Intel Xeon **8 core/node**
- Memory: **16GB RAM/node**
- Storage: **100GB**

3-node Cassandra cluster (NoSQL-Column family DB)

- CPU: Intel Xeon **4C/8T** per node
- Memory: **48GB RAM** (16GB/node)
- Storage: **3TB SSD** (1TB/node)

PostgreSQL node (Relational DB)

- CPU: Intel Xeon **4C/8T** per node
- Memory: **18 GB RAM**
- Storage: **1TB SSD**

MongoDB node (NoSQL-Document DB)

- CPU: Intel Xeon **4C/8T** per node
- Memory: **16 GB RAM**
- Storage: **1TB SSD**



2 Accessing your VM for Docker and Kubernetes

2.1 Accessing campus network

If you are in UIS campus, you are fine. If you are not in UIS campus, you should install a Cisco VPN client software. The VPN client software gives remote users a secure and encrypted VPN (Virtual Private Network) connection to the UIS campus network. Please see below website, <https://www.uis.edu/informationtechnologyservices/connect/vpn/>

2.2 Accessing your VM using SSH

After you install the VPN client software and make a VPN connection to UIS network, you can access your VM using a terminal (Mac), PowerShell (Windows), or Putty (Windows).

- Check the IP address for your VM in the ‘Course Information’ under ‘Modules’ in Canvas

Type below command in your preferred SSH shell client:

```
ssh your-login@10.92.128.36
```

your-login: your UIS NetID (for example, **slee675** from slee675@uis.edu)

Initial password & IPs: See the Virtual Machine IPs page in the ‘Course Information’ under ‘Modules’ in Canvas.

After you logged in to your VM, you will see below prompt. The ‘us2004lts’ is a name of your VM and stands for ‘Ubuntu Sever 20.04 LTS’ that we are using as an OS.

```
your-Login@us2004lts:~$
```

Example1: using terminal in Mac

```
LeeMBP15:~ sslee777$ ssh sslee777@10.92.128.36
The authenticity of host '10.92.128.36 (10.92.128.36)' can't be established.
ECDSA key fingerprint is SHA256:dXhKfHsYIXe/53hvU+HOK2V6fVrTbz/QxmUhpnpXpZA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.92.128.36' (ECDSA) to the list of known hosts.
sslee777@10.92.128.36's password: <== Use initial password until you change it
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-33-generic x86_64)
...
sslee777@us2004lts:~$
```

3 Building an Apache Spark cluster using Kubernetes

3.1 Overview

Apache Spark is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming. See <http://spark.apache.org/docs/2.4.6/index.html>

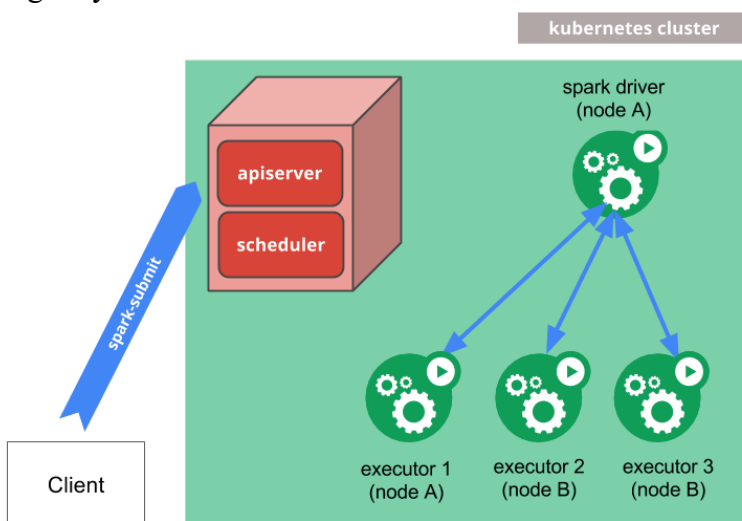
Why Spark on Kubernetes? Kubernetes can work with wide variety of application cluster, your entire application stack it can host. Easy to deploy and manage different types of applications and its different stages. If you have a Kubernetes cluster in your infrastructure, this is the best option available to run a spark.

We want to submit a job to a kubernetes master with a pre-defined docker image to derive the driver container from and run it inside of a pod.

The submission mechanism works as follows: Spark creates a Spark driver running within a Kubernetes pod. The driver creates executors which are also running within Kubernetes pods and connects to them and executes application code. When the application completes, the executor pods terminate and are cleaned up, but the driver pod persists logs and remains in “completed” state in the Kubernetes API until it’s eventually garbage collected or manually cleaned up. Note that in the completed state, the driver pod does not use any computational or memory resources.

The driver and executor pod scheduling are handled by Kubernetes. Communication to the Kubernetes API is done via fabric8. It is possible to schedule the driver and executor pods on a subset of available nodes through a node selector using the configuration property for it.

If you want to learn more about Apache Spark, take Big data analytics or Data Mining course taught by me.



Source: <https://spark.apache.org/docs/2.4.6/running-on-kubernetes.html#how-it-works>

Before we continue, let's delete current Kubernetes on minikube.

```
sslee777@us20041ts:~/mySparkCluster$ minikube stop
👉 Stopping "minikube" in docker ...
🔴 Powering off "minikube" via SSH ...
🔴 Node "minikube" stopped.
sslee777@us20041ts:~/mySparkCluster$ minikube delete
🔥 Deleting "minikube" in docker ...
🔥 Deleting container "minikube" ...
🔥 Removing /home/sslee777/.minikube/machines/minikube ...
👤 Removed all traces of the "minikube" cluster.
```

3.2 Configuring Minikube

We will create a working directory, mySparkCluster

```
sslee777@us20041ts:~$ mkdir mySparkCluster
sslee777@us20041ts:~$ cd mySparkCluster/
sslee777@us20041ts:~/mySparkCluster$
```

Due to default minikube configuration is not enough for running Spark applications, Spark recommend 3 CPUs and 4g of memory to be able to start a simple Spark application with a single executor. See prerequisites (<http://spark.apache.org/docs/latest/running-on-kubernetes.html>)

We will use

- CPU: 6 cpus (recommended 3 cpu)
- Memory: 14 GiB (recommended at least 4 GiB)

Pass the memory and CPU options to the minikube start command

```
sslee777@us20041ts:~/mySparkCluster$ minikube start --cpus 6 --memory 14336
😄 minikube v1.11.0 on Ubuntu 20.04
🌟 Automatically selected the docker driver
👍 Starting control plane node minikube in cluster minikube
🔥 Creating docker container (CPUs=6, Memory=14336MB) ...
🐳 Preparing Kubernetes v1.18.3 on Docker 19.03.2 ...
   ▪ kubeadm.pod-network-cidr=10.244.0.0/16
🔍 Verifying Kubernetes components...
🌟 Enabled addons: default-storageclass, storage-provisioner
🚀 Done! kubectl is now configured to use "minikube"
sslee777@us20041ts:~/mySparkCluster$
```

Or you may use minikube config command like below.

```
sslee777@us20041ts:~/mySparkCluster$ minikube config set memory 414336
! These changes will take effect upon a minikube delete and then a minikube start
sslee777@us20041ts:~/mySparkCluster$ minikube config set cpus 6
! These changes will take effect upon a minikube delete and then a minikube start
sslee777@us20041ts:~/mySparkCluster$
```

Minikube addon list is shown below.

```
sslee777@us20041ts:~/mySparkCluster$ minikube addons list
```

ADDON NAME	PROFILE	STATUS
ambassador	minikube	disabled
dashboard	minikube	disabled
default-storageclass	minikube	enabled ✓
efk	minikube	disabled
freshpod	minikube	disabled
gvisor	minikube	disabled
helm-tiller	minikube	disabled
ingress	minikube	disabled
ingress-dns	minikube	disabled
istio	minikube	disabled
istio-provisioner	minikube	disabled
logviewer	minikube	disabled
metallb	minikube	disabled
metrics-server	minikube	disabled
nvidia-driver-installer	minikube	disabled
nvidia-gpu-device-plugin	minikube	disabled
olm	minikube	disabled
registry	minikube	disabled
registry-aliases	minikube	disabled
registry-creds	minikube	disabled
storage-provisioner	minikube	enabled ✓
storage-provisioner-gluster	minikube	disabled

```
sslee777@us20041ts:~/mySparkCluster$
```

3.3 Creating Spark Docker images

3.3.1 Downloading Apache Spark

Go to <https://spark.apache.org/downloads.html>

Choose a Spark release: 2.4.6 (Jun 05 2020)

Choose a package type: Prebuilt for Apache Hadoop 2.7

Download Spark: **Click** the link

(See screenshots in next page)



[Download](#) [Libraries](#) [Documentation](#) [Examples](#) [Community](#) [Developers](#)

Download Apache Spark™

1. Choose a Spark release: [2.4.6 \(Jun 05 2020\)](#)
2. Choose a package type: [Pre-built for Apache Hadoop 2.7](#)
3. Download Spark: [spark-2.4.6-bin-hadoop2.7.tgz](#)
4. Verify this release using the 2.4.6 [signatures](#), [checksums](#) and [project release KEYS](#).

Note that, Spark 2.x is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12. Spark 3.0+ is pre-built with Scala 2.12.



COMMUNITY-LED DEVELOPMENT

[Projects](#) [People](#) [Community](#)

We suggest the following mirror site for your download:

<http://mirrors.advancedhosters.com/apache/spark/spark-2.4.6/spark-2.4.6-bin-hadoop2.7.tgz>

Other mirror sites are suggested below.

It is essential that you [verify the integrity](#) of the downloaded file using the PGP signature ([.asc](#) file)

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or

HTTP

<http://apache.mirrors.hoobly.com/spark/spark-2.4.6/spark-2.4.6-bin-hadoop2.7.tgz>

<http://apache.mirrors.pair.com/spark/spark-2.4.6/spark-2.4.6-bin-hadoop2.7.tgz>

Copy one of the link addresses then download it using 'wget' command.

```

sslee777@us2004lts:~/mySparkCluster$ wget
http://mirror.cc.columbia.edu/pub/software/apache/spark/spark-2.4.6/spark-2.4.6-bin-
hadoop2.7.tgz

--2020-07-10 19:43:58--
http://mirror.cc.columbia.edu/pub/software/apache/spark/spark-2.4.6/spark-2.4.6-bin-
hadoop2.7.tgz
Resolving mirror.cc.columbia.edu (mirror.cc.columbia.edu)... 128.59.59.71
Connecting to mirror.cc.columbia.edu (mirror.cc.columbia.edu)|128.59.59.71|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 233215067 (222M) [application/x-gzip]
Saving to: 'spark-2.4.6-bin-hadoop2.7.tgz'

spark-2.4.6-bin-had 100%[=====>] 222.41M  68.4MB/s   in 4.4s

2020-07-10 19:44:02 (50.2 MB/s) - 'spark-2.4.6-bin-hadoop2.7.tgz' saved
[233215067/233215067]

sslee777@us2004lts:~/mySparkCluster$

```

Unzip the spark distribution.

```

sslee777@us2004lts:~/mySparkCluster$ tar xzvf spark-2.4.6-bin-hadoop2.7.tgz
spark-2.4.6-bin-hadoop2.7/
spark-2.4.6-bin-hadoop2.7/bin/
...
sslee777@us2004lts:~/mySparkCluster$

```

Go inside your extracted spark folder and check the shell script for building spark-docker image:
docker-image-tool.sh

```

sslee777@us2004lts:~/mySparkCluster$ cd spark-2.4.6-bin-hadoop2.7/

sslee777@us2004lts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ ls
LICENSE  R          RELEASE  conf  examples  kubernetes  python  yarn
NOTICE  README.md bin       data  jars      licenses    sbin

sslee777@us2004lts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ ls bin
beeline                pyspark                spark-class2.cmd      spark-submit
beeline.cmd            pyspark.cmd            spark-shell            spark-submit.cmd
docker-image-tool.sh   pyspark2.cmd           spark-shell.cmd        spark-submit2.cmd
find-spark-home        run-example            spark-shell2.cmd       sparkR
find-spark-home.cmd    run-example.cmd        spark-sql              sparkR.cmd
load-spark-env.cmd     spark-class            spark-sql.cmd          sparkR2.cmd
load-spark-env.sh      spark-class.cmd        spark-sql2.cmd

sslee777@us2004lts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$

```


3.3.2 Compatibility issue and modifying the Docker file

Currently, Apache Spark Kubernetes has compatibility issue. Bugs in okhttp of fabric8io need to be fixed. See

<https://github.com/fabric8io/kubernetes-client/issues/2145>,
<https://github.com/fabric8io/kubernetes-client/issues/2168>,
<https://github.com/fabric8io/kubernetes-client/issues/2175>, and etc.

Until the bugs in okhttp are fixed, we need to be careful on choosing Java version. So, we will modify the Docker file like below. Open the Dockerfile and update it to use openjdk:8u242-slim image which is based on OpenJDK 1.8.0_242. Original image is based on OpenJDK 1.8.0_252.

```
sslee777@us2004lts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ vi
kubernetes/dockerfiles/spark/Dockerfile

#FROM openjdk:8-jdk-slim # Java 1.8.0_252 Docker image      # Comment this line
FROM openjdk:8u242-slim # Java 1.8.0_242 Docker image      # Add this line!

ARG spark_jars=jars
ARG img_path=kubernetes/dockerfiles
ARG k8s_tests=kubernetes/tests

RUN set -ex && \
    apt-get update && \
    ln -s /lib /lib64 && \
    apt install -y bash tini libc6 libpam-modules libnss3 && \
    mkdir -p /opt/spark && \
    mkdir -p /opt/spark/work-dir && \
    touch /opt/spark/RELEASE && \
    rm /bin/sh && \
    ln -sv /bin/bash /bin/sh && \
    echo "auth required pam_wheel.so use_uid" >> /etc/pam.d/su && \
    chgrp root /etc/passwd && chmod ug+rw /etc/passwd && \
    rm -rf /var/cache/apt/*

COPY ${spark_jars} /opt/spark/jars
COPY bin /opt/spark/bin
COPY sbin /opt/spark/sbin
COPY ${img_path}/spark/entrypoint.sh /opt/
COPY examples /opt/spark/examples
COPY ${k8s_tests} /opt/spark/tests
COPY data /opt/spark/data

ENV SPARK_HOME /opt/spark

WORKDIR /opt/spark/work-dir

ENTRYPOINT [ "/opt/entrypoint.sh" ]
```

3.3.3 Creating a Spark Docker image

Kubernetes requires users to supply images that can be deployed into containers within pods. The images are built to be run in a container runtime environment that Kubernetes supports. Spark (starting with version 2.3) ships with a Dockerfile that can be used for this purpose, or

customized to match an individual application's needs. It can be found in the `kubernetes/dockerfiles/` directory. Spark also ships with a `bin/docker-image-tool.sh` script that can be used to build and publish the Docker images to use with the Kubernetes backend.

To build a spark-docker image, run the command below. It takes time.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ ./bin/docker-image-  
tool.sh -m -t v1.0.uis build  
  
Sending build context to Docker daemon 264.7MB  
Step 1/15 : FROM openjdk:8u242-slim  
---> ceab49f001b1  
...  
(Very Long Logs and takes time)  
...  
Successfully tagged spark:v1.0.uis  
...  
Successfully tagged spark-py:v1.0.uis  
...  
Step 9/9 : ENTRYPOINT [ "/opt/entrypoint.sh" ]  
---> Running in 89cfb38ad6bc  
Removing intermediate container 89cfb38ad6bc  
---> 27e7b6ad220e  
Successfully built 27e7b6ad220e  
Successfully tagged spark-r:v1.0.uis  
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$
```

Note: The `-m` parameter here indicates a minikube build. Using minikube when building images will do so directly into minikube's Docker daemon. There is no need to push the images into minikube in that case, they'll be automatically available when running applications inside the minikube cluster. See <https://github.com/apache/spark/blob/master/bin/docker-image-tool.sh>

Note2: you should see `openjdk:8u242-slim` in the output. You should also see three different spark images:

- For Scala: `spark:v1.0.uis`
- For Python: `spark-py:v1.0.uis`
- For R: `spark-r:v1.0.uis`

Check the created images using ‘docker image ls’ command.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
cassandra	3.11	b4271bf432c4	3 weeks ago	379MB
cassandra	latest	b4271bf432c4	3 weeks ago	379MB
mypostgresql	latest	6a1f499741b3	4 weeks ago	394MB
portainer/portainer	latest	cd645f5a4769	5 weeks ago	79.1MB
test	latest	a1ea900d3495	6 weeks ago	82.6MB
alpine	latest	a24bb4013296	6 weeks ago	5.57MB
dpage/pgadmin4	latest	06b3bc2e116f	7 weeks ago	252MB
postgres	latest	adf2b126dda8	8 weeks ago	313MB
gcr.io/k8s-minikube/kicbase	v0.0.10	e6bc41c39dc4	2 months ago	974MB
ubuntu	16.04	005d2078bdfa	2 months ago	125MB
ubuntu	latest	1d622ef86b13	2 months ago	73.9MB
postgres	9.4	ed5a45034282	5 months ago	251MB
hello-world	latest	bf756fb1ae65	6 months ago	13.3kB

```
sslee777@us20041ts: ~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$
```

You see all the images you have. However, you cannot find three spark images you just created!

To find your spark images, you need to connect to minikube’s docker daemon.

<https://kubernetes.io/docs/setup/learning-environment/minikube/#use-local-images-by-re-using-the-docker-daemon>

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ eval $(minikube docker-env)
```

NOTE: You need to run above command again if you logged off and on.

Now, you should see three new images you created: spark-r, spark-py, and spark.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
spark-r	v1.0.uis	4279fc62420e	39 seconds ago	1.11GB
spark-py	v1.0.uis	3be401edb55f	2 minutes ago	1.06GB
spark	v1.0.uis	c2b0416ad575	4 minutes ago	555MB
k8s.gcr.io/kube-proxy	v1.18.3	3439b7546f29	7 weeks ago	117MB
k8s.gcr.io/kube-apiserver	v1.18.3	7e28efa976bd	7 weeks ago	173MB
k8s.gcr.io/kube-controller-manager	v1.18.3	da26705ccb4b	7 weeks ago	162MB
k8s.gcr.io/kube-scheduler	v1.18.3	76216c34ed0c	7 weeks ago	95.3MB
kubernetesui/dashboard	v2.0.0	8b32422733b3	2 months ago	222MB
openjdk	8u242-slim	24c99a4e95d3	3 months ago	284MB
k8s.gcr.io/pause	3.2	80d28bedfe5d	5 months ago	683kB
k8s.gcr.io/coredns	1.6.7	67da37a9a360	5 months ago	43.8MB
k8s.gcr.io/etcd	3.4.3-0	303ce5db0e90	8 months ago	288MB
kubernetesui/metrics-scraper	v1.0.2	3b08661dc379	8 months ago	40.1MB
gcr.io/k8s-minikube/storage-provisioner	v1.8.1	4689081edb10	2 years ago	80.8MB

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$
```

Later, when you no longer wish to use the Minikube host, you can undo this change by running

```
$ eval $(minikube docker-env -u)
```

The command ‘minikube docker-env’ gives some environment variables (export). You, then, are adding those variables in your shell environment.

https://www.tutorialspoint.com/unix_commands/export.htm,

https://www.tutorialspoint.com/unix_commands/eval.htm,
<https://stackoverflow.com/questions/52310599/what-does-minikube-docker-env-mean>

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ minikube docker-env
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://172.17.0.3:2376"
export DOCKER_CERT_PATH="/home/sslee777/.minikube/certs"
export MINIKUBE_ACTIVE_DOCKERD="minikube"

# To point your shell to minikube's docker-daemon, run:
# eval $(minikube -p minikube docker-env)
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$
```

3.4 Running Spark app (analytical queries) using spark-submit

There are two ways of submitting jobs: client or cluster mode. There is a subtle difference between the two. If you use client mode, you can tell the driver to run on dedicated infrastructure (separate than executors) whereas if you choose cluster mode, both drivers and executors run in the same cluster. You can use Spark configurations as well as Kubernetes specific options within your command. See, <https://aws.amazon.com/blogs/containers/optimizing-spark-performance-on-kubernetes/>

In this exercise, we will use cluster mode.

- Cluster Mode: <https://spark.apache.org/docs/2.4.6/running-on-kubernetes.html#cluster-mode>
- Client Mode: <https://spark.apache.org/docs/2.4.6/running-on-kubernetes.html#client-mode>

More detail, see <https://www.lightbend.com/blog/how-to-manage-monitor-spark-on-kubernetes-introduction-spark-submit-kubernetes-operator#:~:text=How%20Does%20Spark%2DSubmit%20Work,pods%20in%20your%20Kubernetes%20cluster.>

The spark-submit script in Spark's bin directory is used to launch applications on a cluster. It can use all of Spark's supported cluster managers through a uniform interface, so you don't have to configure your application especially for each one.

We will run an example Spark app that will calculate Pi. If you are interested in how to calculate Pi, See <https://www.mathscareers.org.uk/article/calculating-pi/>

3.4.1 Installing Java JRE and compatibility

To run 'spark-submit', you need a Java. Due to Java compatibility issue mentioned earlier, we will use default-jre based on openjdk-11 image.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ java

Command 'java' not found, but can be installed with:

sudo apt install default-jre # version 2:1.11-72, or
```

```

sudo apt install openjdk-11-jre-headless # version 11.0.7+10-3ubuntu1
sudo apt install openjdk-13-jre-headless # version 13.0.3+3-1ubuntu2
sudo apt install openjdk-14-jre-headless # version 14.0.1+7-1ubuntu1
sudo apt install openjdk-8-jre-headless # version 8u252-b09-1ubuntu1

```

```

sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ sudo apt install
default-jre

```

```

[sudo] password for sslee777:
Reading package lists... Done

```

```

...
done.
done.

```

```

sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$

```

The default-jre and openjdk-14 are tested and work well. However, openjdk-8-jre doesn't work.

Check your Java's version

```

sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ java -version
openjdk version "11.0.7" 2020-04-14
OpenJDK Runtime Environment (build 11.0.7+10-post-Ubuntu-3ubuntu1)
OpenJDK 64-Bit Server VM (build 11.0.7+10-post-Ubuntu-3ubuntu1, mixed mode, sharing)

```

Run the following command to see the included kube-system pods:

```

sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ kubectl get pods --
all-namespaces

```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
kube-system	coredns-66bff467f8-jsj5q	1/1	Running	0	3h10m
kube-system	coredns-66bff467f8-zgh77	1/1	Running	0	3h10m
kube-system	etcd-minikube	1/1	Running	0	3h11m
kube-system	kube-apiserver-minikube	1/1	Running	0	3h11m
kube-system	kube-controller-manager-minikube	1/1	Running	0	3h11m
kube-system	kube-proxy-hp2tk	1/1	Running	0	3h10m
kube-system	kube-scheduler-minikube	1/1	Running	0	3h11m
kube-system	storage-provisioner	1/1	Running	0	3h11m
kubernetes-dashboard	dashboard-metrics-scraper-84bdf55ff-8m9nt	1/1	Running	0	53m
kubernetes-dashboard	kubernetes-dashboard-696dbcc666-b5k2v	1/1	Running	0	53m

```

sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$

```

3.4.2 Role-based access control (RBAC)

Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within your organization. See

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/>

In Kubernetes clusters with RBAC enabled, users can configure Kubernetes RBAC roles and service accounts used by the various Spark on Kubernetes components to access the Kubernetes API server. See, <https://spark.apache.org/docs/2.4.6/running-on-kubernetes.html#rbac>

In this exercise, you should create one service account for running the spark job with the required access as below if you do not have one to use. Allow service account default:spark access namespace default

```
sslee777@us20041ts:~$ kubectl create serviceaccount spark
serviceaccount/spark created

sslee777@us20041ts:~$ kubectl create clusterrolebinding spark-role --clusterrole=edit
--serviceaccount=default:spark --namespace=default
clusterrolebinding.rbac.authorization.k8s.io/spark-role created

sslee777@us20041ts:~$
```

You also need to add a parameter in spark-submit script, see 3.4.4 Submitting Spark job using spark-submit.

3.4.3 Preparing for running a spark job

First, check your Kubernetes master's IP address in Minikube: e.g., 172.17.0.3

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ kubectl cluster-info
Kubernetes master is running at https://172.17.0.3:8443
KubeDNS is running at https://172.17.0.3:8443/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$
```

Or you may use a command 'minikube ip'

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ minikube ip
172.17.0.3
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$
```

3.4.4 Submitting Spark job using spark-submit

```
./bin/spark-submit --master k8s://https://172.17.0.3:8443 \
--verbose \
--deploy-mode cluster \
--name spark-pi \
--class org.apache.spark.examples.SparkPi \
--conf spark.executor.instances=3 \
--conf spark.kubernetes.container.image=spark:v1.0.uis \
--conf spark.kubernetes.driver.pod.name=spark-pi-driver \
--conf spark.kubernetes.authenticate.driver.serviceAccountName=spark \
local:///opt/spark/examples/jars/spark-examples_2.11-2.4.6.jar
```

We will use cluster mode (deploy-mode cluster) and name it spark-pi. We will use SparkPi class to calculate Pi with 3 instances/executors/nodes. We will use the spark image we created before, spark:v1.0.uis and named the driver pod as spark-pi-driver. We need to give serviceAccountName=spark for RBAC, see Section 3.4.2. The spark-examples jar will be used.

The location of the jar file is for Spark Docker image/container/Pod we created, not for downloaded Apache Spark in Section 3.3.1.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ ./bin/spark-submit --
master k8s://https://172.17.0.3:8443 \
> --verbose \
> --deploy-mode cluster \
> --name spark-pi \
> --class org.apache.spark.examples.SparkPi \
> --conf spark.executor.instances=3 \
> --conf spark.kubernetes.container.image=spark:v1.0.uis \
> --conf spark.kubernetes.driver.pod.name=spark-pi-driver \
> --conf spark.kubernetes.authenticate.driver.serviceAccountName=spark \
> local:///opt/spark/examples/jars/spark-examples_2.11-2.4.6.jar
Using properties file: null
...
Spark properties used, including those specified through
--conf and those from the properties file null:
(spark.kubernetes.driver.pod.name,spark-pi-driver)
(spark.executor.instances,3)
(spark.kubernetes.container.image,spark:v1.0.uis)
...
    container images: spark:v1.0.uis
    phase: Succeeded
...
    Container name: spark-kubernetes-driver
    Container image: spark:v1.0.uis
    Container state: Terminated
    Exit code: 0
...
```

Also look at more options for submitting applications:

<https://spark.apache.org/docs/2.4.6/submitting-applications.html>

After we submit the app (analytic queries), they will form a Spark cluster (One Spark driver and three executors) to calculate Pi and then return the results (answers).

If everything went fine, we should be able to check logs and see the result.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
spark-pi-driver      0/1     Completed   0          8m38s
```

Then we can cat or grep a log from that pod to see the results.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.5-bin-hadoop2.6$ kubectl logs spark-pi-
driver | grep "Pi is roughly"
```

This should give us a result. This result is shown in the log

```
Pi is roughly 3.144075720378602
```

3.4.5 Error?

If you have any error? You may want to check the pod's status.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
spark-pi-driver	0/1	Error	0	8m38s

You may try to debug it using kubectl logs.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.5-bin-hadoop2.6$ kubectl logs spark-pi-driver
```

If you see an error message below,

```
Exception in thread "main" io.fabric8.kubernetes.client.KubernetesClientException: Failure executing: POST at: https://172.17.0.3:8443/api/v1/namespaces/default/pods. Message: Forbidden! User minikube doesn't have permission. pods "spark-pi-driver" is forbidden: error looking up service account default/spark: serviceaccount "spark" not found.
```

Check RBAC.

For debugging, See <https://spark.apache.org/docs/2.4.6/running-on-kubernetes.html#debugging>

You may want to delete the pod. If you delete the pod, related service will also be deleted.

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
spark-pi-driver	0/1	Error	0	8m38s

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	124m
spark-pi-driver-svc	ClusterIP	None	<none>	7078/TCP,7079/TCP	8m45s

```
sslee777@us20041ts:~/mySparkCluster/spark-2.4.6-bin-hadoop2.7$ kubectl delete pods spark-pi-driver
pod "spark-pi-driver" deleted

sslee777@us20041ts:~/mySparkCluster/spark-2.4.5-bin-hadoop2.7$ kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16m

If you want to start from the scratch? Stop and delete minikube cluster and start from the scratch.


```

sslee777@us2004lts:~/mySparkCluster$ minikube stop
👉 Stopping "minikube" in docker ...
🔴 Powering off "minikube" via SSH ...
🔴 Node "minikube" stopped.
sslee777@us2004lts:~/mySparkCluster$ minikube delete
🔥 Deleting "minikube" in docker ...
🔥 Deleting container "minikube" ...
🔥 Removing /home/sslee777/.minikube/machines/minikube ...
💀 Removed all traces of the "minikube" cluster.
sslee777@us2004lts:~/mySparkCluster$

```

4 Submit

Submit a word document (docx, doc, or PDF) to Canvas.

Assignments

- Run your code/scripts, take screenshots, and explain about it.
 - a. Section 3.2 Configuring Minikube
 - b. Section 3.3 Creating Spark Docker images
 - c. Section 3.4 Running Spark app (analytical queries) using spark-submit
- Run the minikube Dashboard again.
 - a. Take screenshots of pages below and [explain what information they show](#)
 - i. Cluster > Nodes > minikube
 - ii. Workloads > Pods > spark-pi-driver
 - iii. Discovery and Load Balancing > Services > spark-pi-xxxxx-driver-svc

If you have any problems or questions regarding this exercise, post messages in the 'Discussions' in Canvas. **Posting exact answer codes for your assignment or links to an article that have exact answer codes are not allowed.**

5 Reference (Note: Links may not work in PDF, try to copy&paste it)

1. Running Spark on Kubernetes, <https://spark.apache.org/docs/2.4.6/running-on-kubernetes.html>
2. Spark Overview, <http://spark.apache.org/docs/2.4.6/index.html>
3. Deploying Spark on Kubernetes, <https://testdriven.io/blog/deploying-spark-on-kubernetes/>
4. Quickstart: Apache Spark on Kubernetes, <https://dzone.com/articles/quickstart-apache-spark-on-kubernetes>
5. cheatsheet: How to run Spark 2.4 on OSX/MiniKube, <https://medium.com/@andreyonistchuk/how-to-run-spark-2-4-on-osx-minikube-f0e5fdeb27be>
6. Bigdata Session 4-Apache Spark on Kubernetes, <https://haridas.in/bigdata-session-4-apache-spark-on-kubernetes.html>
7. Apache Spark Download, <https://spark.apache.org/downloads.html>
8. Java docker image, 8u242-jre-slim, https://hub.docker.com/_/openjdk?tab=tags&page=1&name=242

9. Use local images by re-using the Docker daemon,
<https://kubernetes.io/docs/setup/learning-environment/minikube/#use-local-images-by-re-using-the-docker-daemon>
10. What does minikube docker-env mean?,
<https://stackoverflow.com/questions/52310599/what-does-minikube-docker-env-mean>
11. How to Delete Pods from a Kubernetes Node, <https://www.bluematador.com/blog/safely-removing-pods-from-a-kubernetes-node>
12. KUBERNETES I - RUNNING KUBERNETES LOCALLY VIA MINIKUBE,
<https://www.bogotobogo.com/DevOps/DevOps-Kubernetes-1-Running-Kubernetes-Locally-via-Minikube.php>
13. Running Spark on Kubernetes: a fully functional example and why it makes sense for OLX, <https://tech.olx.com/running-spark-on-kubernetes-a-fully-functional-example-and-why-it-makes-sense-for-olx-d56b6a61fcbe>
14. spark/bin/docker-image-tool.sh, <https://github.com/apache/spark/blob/master/bin/docker-image-tool.sh>
15. Quick Start Guide—Submit Spark (2.4) Jobs on Minikube/AWS,
<https://medium.com/@aelbuni/apache-spark-2-4-3-running-jobs-in-kubernetes-cluster-ebd7a28b99cd>
16. ===== Error related =====
17. After updating to Azul OpenJDK 8.46 (~=8u252) the client fails with
java.net.SocketException: Software caused connection abort: socket write error #2145,
<https://github.com/fabric8io/kubernetes-client/issues/2145>
18. run spark-submit failed on Kubernetes cluster #2168,
<https://github.com/fabric8io/kubernetes-client/issues/2168>
19. Using newer Java versions fails RBAC auth on k8s v1.17 and later #4039,
<https://github.com/spring-cloud/spring-cloud-dataflow/issues/4039>
20. kubernetes-client could not work with java 8u252 #2212,
<https://github.com/fabric8io/kubernetes-client/issues/2212>
21. Spark with K8S : Forbidden!Configured service account doesn't have access,
<https://stackoverflow.com/questions/55498702/spark-with-k8s-forbiddenconfigured-service-account-doesnt-have-access/56339507>
22. ImagePullBackOff error in Kubernetes while pulling docker images from private
dockerhub registry, <https://stackoverflow.com/questions/55163450/imagepullbackoff-error-in-kubernetes-while-pulling-docker-images-from-private-do>
23. ImagePullBackOff error Google Kubernetes Engine,
<https://stackoverflow.com/questions/49298520/imagepullbackoff-error-google-kubernetes-engine>
24. Kubernetes Troubleshooting Walkthrough – imagepullbackoff,
<https://managedkube.com/kubernetes/k8sbot/troubleshooting/imagepullbackoff/2019/02/23/imagepullbackoff.html>
25. Kubernetes 1.9 can't initialize SparkContext,
<https://stackoverflow.com/questions/49152771/kubernetes-1-9-cant-initialize-sparkcontext>
26. How to fix error 'Operation [get] for kind [pod] with name [spark-wordcount-driver] in
namespace [non-default-namespace] failed,

<https://stackoverflow.com/questions/53908840/how-to-fix-error-operation-get-for-kind-pod-with-name-spark-wordcount-driv>

27. Java 8 vs Java 11 – What are the Key Changes?,
<https://blog.idrsolutions.com/2019/05/java-8-vs-java-11-what-are-the-key-changes/>
28. How To Uninstall JDK From Linux ?,
<https://novicestuffs.wordpress.com/2017/04/25/how-to-uninstall-java-from-linux/>