

M1-Spark Stadnalone Cluster

[미션](#)

[Spark Standalone Cluster](#)

[Dockerfile](#)

[docker-compose.yaml](#)

[실행](#)

[결과](#)

[트러블슈팅](#)

[문제 1.](#)

[해결방법](#)

[참고설정](#)

▼ 미션

학습목표

이 프로젝트의 목적은 Docker를 사용하여 Apache Spark 독립형 클러스터를 빌드하고 Spark 작업을 실행하고 결과를 검증하는 것입니다.

사전지식

기능요구사항

클러스터 설정:

클러스터는 하나의 Spark 마스터와 두 개의 Spark 워커 노드로 구성되어야 합니다. 각 노드는 다른 노드와 통신할 수 있어야 하며, 마스터는 작업자에 대한 작업을 예약할 수 있어야 합니다. Spark 웹 UI는 작업 진행 상황과 클러스터 상태를 모니터링할 수 있어야 합니다.

Spark 작업 실행:

Spark 작업은 마운트된 볼륨이나 지정된 입력 경로에서 데이터 세트를 읽을 수 있어야 합니다. 이 작업은 데이터 변환(예: π 추정)을 수행하고 결과를 지정된 출력 경로에 기록해야 합니다. 출력은 올바르게 분할되어 CSV나 Parquet 등의 형식으로 저장되어야 합니다.

오류 처리:

설정은 Docker 컨테이너 간 네트워크 문제 및 구성 오류를 포함한 모든 오류를 원활하게 처리해야 합니다. Spark 작업이나 클러스터 설정과 관련된 문제를 디버깅하려면 로그에 접근할 수 있어야 합니다.

재현성:

제공된 Dockerfile과 docker-compose.yml을 사용하면 전체 설정을 재현할 수 있습니다. Docker 이미지를 빌드하고, 클러스터를 시작하고, 작업을 제출하고, 결과를 확인하는 방법에 대한 명확한 지침을 제공해야 합니다.

프로그래밍관점

Dockerfile:

Apache Spark로 Docker 이미지를 빌드하기 위해 Dockerfile을 만듭니다. Docker 이미지에는 Java, Python, Spark가 포함되어야 합니다. 이미지가 독립 실행형 모드에서 실행되도록 구성되어 있는지 확인하세요.

Docker Compose 파일:

- 하나의 마스터 노드와 두 개의 워커 노드로 구성된 Spark 독립형 클러스터를 설정하려면 docker-compose.yml 파일을 만듭니다.
- Spark 웹 UI와 마스터에 필요한 포트를 노출하도록 Docker Compose 파일을 구성합니다.

스파크 잡:

- Spark 배포 파일에 포함된 샘플 작업 스크립트 중 하나를 사용하세요(예: examples/src/main/python/pi.py).
- 스크립트가 데이터 세트를 읽고 변환을 수행하고(예: 몬테카를로 방법을 사용하여 π 추정) 결과를 지정된 출력 위치에 기록하는지 확인합니다.

제출 스크립트:

- spark-submit을 사용하여 Spark 작업을 독립형 클러스터에 제출하는 셸 스크립트를 만듭니다.

시청결과 및 동작예시

- <http://localhost:8080> 에서 Spark 웹 UI를 확인하여 작업을 모니터링합니다.
- 출력(π 의 추정 값)은 Spark 작업 로그에 인쇄됩니다.

Spark Standalone Cluster

▼ Dockerfile

```
FROM ubuntu:22.04

ENV PDSH_RCMD_TYPE=ssh
ENV JAVA_HOME=/usr/lib/jvm/java-11-openjdk-arm64
ENV SPARK_VERSION=3.5.1
ENV SPARK_HOME=/opt/spark
ENV SPARK_CONFIG_DIR=$SPARK_HOME/conf
ENV PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin:$JAVA_HOME/
ENV MASTER=spark://spark-master:7077

RUN apt-get update && \
    apt-get install -y openjdk-11-jdk python3 python3-pip s
    apt-get clean

# Download and extract Hadoop to /opt
RUN wget http://apache.mirror.cdnetworks.com/spark/spark-$S
    tar -xzf /opt/spark-$SPARK_VERSION-bin-hadoop3.tgz -C
    mv /opt/spark-$SPARK_VERSION-bin-hadoop3 $SPARK_HOME &&
    rm /opt/spark-$SPARK_VERSION-bin-hadoop3.tgz

# 사용자 및 그룹 생성 및 권한 설정
RUN groupadd -g 1000 hadoop && \
    useradd -m -u 1001 -g 1000 spark && \
    echo "spark ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers

RUN chown -R spark:hadoop $SPARK_HOME

ENV SPARK_NO_DAEMONIZE=true
USER spark
```

```
CMD ["you must override this"]
```

▼ docker-compose.yml

```
version: "3.8"
services:
  spark-master:
    image: jnamu/spark-node:latest
    container_name: spark-master
    hostname: spark-master
    ports:
      - "8080:8080"
      - "4040:4040"
      - "7077:7077"
      - "7777:7777"
    command: ["/opt/spark/sbin/start-master.sh"]

  spark-worker-1:
    image: jnamu/spark-node:latest
    container_name: spark-worker-1
    hostname: spark-worker-1
    depends_on:
      - spark-master
    ports:
      - "8081:8081"
    command: ["/opt/spark/sbin/start-worker.sh", "spark://s

  spark-worker-2:
    image: jnamu/spark-node:latest
    container_name: spark-worker-2
    hostname: spark-worker-2
    depends_on:
      - spark-master
    ports:
      - "8082:8081"
```

```
command: ["/opt/spark/sbin/start-worker.sh", "spark://s
```

실행

이미지 빌드

```
docker build -t jnamu/spark-node:latest .
```

클러스터 실행

```
docker-compose up -d
```

스파크 잡 실행(pi.py)

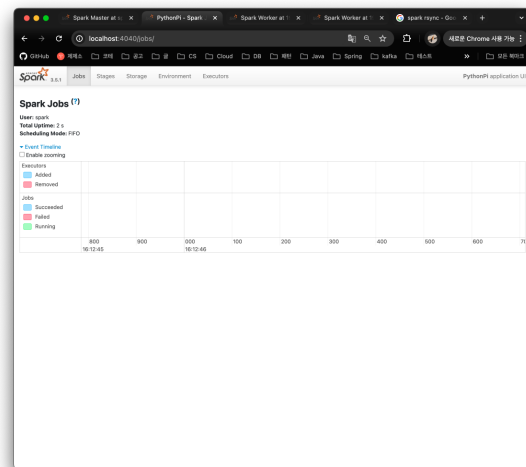
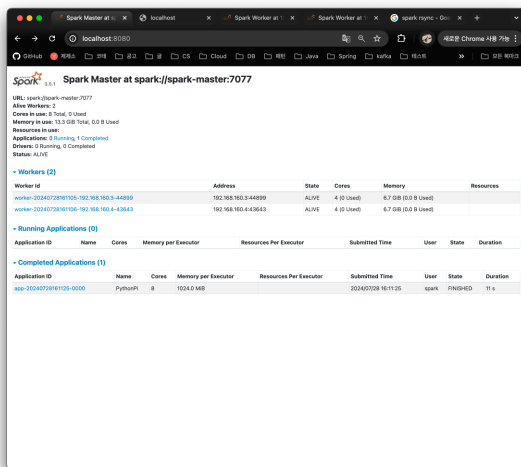
```
docker exec -it spark-master /bin/bash
# 컨테이너 셸 안에서
spark-submit --master spark://spark-master:7077 $SPARK_HOME/e
```

결과

example job - pi.py

```
24/07/28 15:53:03 INFO DAGScheduler: Job 0 is finished: cancelling potential speculative or zombie tasks for this job
24/07/28 15:53:03 INFO TaskSchedulerImpl: Killing all running tasks in stage 0: Stage finished
24/07/28 15:53:03 INFO DAGScheduler: Job 0 finished: reduce at /opt/spark/examples/src/main/python/pi.py:42, took 2.358899 s
Pi is roughly 3.143400
24/07/28 15:53:03 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/07/28 15:53:03 INFO SparkUI: Stopped Spark web UI at http://spark-master:4040
24/07/28 15:53:03 INFO StandaloneSchedulerBackend: Shutting down all executors
24/07/28 15:53:03 INFO StandaloneSchedulerBackend$StandaloneDriverEndpoint: Asking each executor to shut down
24/07/28 15:53:03 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/07/28 15:53:03 INFO MemoryStore: MemoryStore cleared
24/07/28 15:53:03 INFO BlockManager: BlockManager stopped
24/07/28 15:53:03 INFO BlockManagerMaster: BlockManagerMaster stopped
24/07/28 15:53:03 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/07/28 15:53:03 INFO SparkContext: Successfully stopped SparkContext
24/07/28 15:53:04 INFO ShutdownHookManager: Shutdown hook called
24/07/28 15:53:04 INFO ShutdownHookManager: Deleting directory /tmp/spark-7ba65a6f-87b9-4730-9377-97e42e36b986
24/07/28 15:53:04 INFO ShutdownHookManager: Deleting directory /tmp/spark-350e5619-b812-4378-93a5-b61b0d030920
24/07/28 15:53:04 INFO ShutdownHookManager: Deleting directory /tmp/spark-7ba65a6f-87b9-4730-9377-97e42e36b986/pyspark-216c947a-cd8e-499e-b13c-71e57202145f
```

잡이 너무 일찍 끝나서 제대로 표시가 안되는것.



트러블슈팅

문제 1.

스파크 잡 실행 시 아래와 같이 IOException 발생

- 결과는 정상적으로 출력하지만, 작업을 종료할 때 예외가 발생한다.
- `pi.py` 에는 결과를 파일로 작성하는 작업이 없다.

```
spark-submit --master spark://spark-master:7077 pi.py
```

```

Pi is roughly 3.137800
24/07/26 01:19:39 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/07/26 01:19:39 WARN AbstractConnector:
java.io.IOException: No such file or directory
    at sun.nio.ch.NativeThread.signal(Native Method)
    at sun.nio.ch.ServerSocketChannelImpl.implCloseSelectableChannel(ServerSocketChannelImpl.java:291)
    at java.nio.channels.spi.AbstractSelectableChannel.implCloseChannel(AbstractSelectableChannel.java:241)
    at java.nio.channels.spi.AbstractInterruptibleChannel.close(AbstractInterruptibleChannel.java:115)
    at org.sparkproject.jetty.server.ServerConnector.close(ServerConnector.java:371)
    at org.sparkproject.jetty.server.AbstractNetworkConnector.shutdown(AbstractNetworkConnector.java:104)
    at org.sparkproject.jetty.server.Server.doStop(Server.java:444)
    at org.sparkproject.jetty.util.component.AbstractLifecycle.stop(AbstractLifecycle.java:94)
    at org.apache.spark.ui.ServerInfo.stop(JettyUtils.scala:532)
    at org.apache.spark.ui.WebUI.$anonfun$stop$2(WebUI.scala:182)
    at org.apache.spark.ui.WebUI.$anonfun$stop$2$adapted(WebUI.scala:182)
    at scala.Option.foreach(Option.scala:407)
    at org.apache.spark.ui.WebUI.stop(WebUI.scala:182)
    at org.apache.spark.ui.SparkUI.stop(SparkUI.scala:157)
    at org.apache.spark.SparkContext.$anonfun$stop$7(SparkContext.scala:2253)
    at org.apache.spark.SparkContext.$anonfun$stop$7$adapted(SparkContext.scala:2253)
    at scala.Option.foreach(Option.scala:407)
    at org.apache.spark.SparkContext.$anonfun$stop$6(SparkContext.scala:2253)
    at org.apache.spark.util.Utils$.tryLogNonFatalError(Utils.scala:1375)
    at org.apache.spark.SparkContext.stop(SparkContext.scala:2253)

```

워커 노드에서 해당 스파크 잡에 로그를 보면 다음과 같은 내용이 찍혀있다.



해결방법

그동안 amd64 이미지를 사용해왔다. (설치되는 Java도)

⇒ arm으로 사용하자 위 문제가 해결됐다.

중간 계층에서 amd→arm 간 instruction 변환을 수행하긴 하지만 완벽하게 지원하지는 않는 모양이다.

위에 문제는 IOException(File or Directory Not Found)이었지만 실제로 pi.py파일에는 쓰기 작업이 존재하지 않는다. + 위 로그에 찍힌 디렉토리 경로에 권한을 부여해도 문제는 해결되지 않는다.

단순히 arm 아키텍처에 적합한 이미지를 선택하자.

+) arm으로 변경 후 모든 instruction 속도가 획기적으로 빨라졌다

```
1 FROM --platform=linux/amd64 ubuntu:22.04
2
3 ENV PDSH_RCMD_TYPE=ssh
4 ENV JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
5 ENV SPARK_VERSION=3.5.1
6 ENV SPARK_HOME=/opt/spark
7 ENV SPARK_CONFIG_DIR=$SPARK_HOME/conf
8 ENV PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin:$JAVA_HOME/bin
9 ENV MASTER=spark://spark-master:7077
10
```

```
1 FROM ubuntu:22.04
2
3 ENV PDSH_RCMD_TYPE=ssh
4 ENV JAVA_HOME=/usr/lib/jvm/java-11-openjdk-arm64
5 ENV SPARK_VERSION=3.5.1
6 ENV SPARK_HOME=/opt/spark
7 ENV SPARK_CONFIG_DIR=$SPARK_HOME/conf
8 ENV PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin:$JAVA_HOME/bin
9 ENV MASTER=spark://spark-master:7077
```

```
24/07/28 15:53:03 INFO DAGScheduler: Job 0 is finished: Cancelling potential speculative or zombie tasks for this job
24/07/28 15:53:03 INFO TaskSchedulerImpl: Killing all running tasks in stage 0: Stage finished
24/07/28 15:53:03 INFO DAGScheduler: Job 0 finished: reduce at /opt/spark/examples/src/main/python/pi.py:42, took 2.358899 s
Pi is roughly 3.143400
24/07/28 15:53:03 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/07/28 15:53:03 INFO SparkUI: Stopped Spark web UI at http://spark-master:4040
24/07/28 15:53:03 INFO StandaloneSchedulerBackend: Shutting down all executors
24/07/28 15:53:03 INFO StandaloneSchedulerBackend$StandaloneDriverEndpoint: Asking each executor to shut down
24/07/28 15:53:03 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/07/28 15:53:03 INFO MemoryStore: MemoryStore cleared
24/07/28 15:53:03 INFO BlockManager: BlockManager stopped
24/07/28 15:53:03 INFO BlockManagerMaster: BlockManagerMaster stopped
24/07/28 15:53:03 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/07/28 15:53:03 INFO SparkContext: Successfully stopped SparkContext
24/07/28 15:53:04 INFO ShutdownHookManager: Shutdown hook called
24/07/28 15:53:04 INFO ShutdownHookManager: Deleting directory /tmp/spark-7ba65a6f-87b9-4730-9377-97e42e36b986
24/07/28 15:53:04 INFO ShutdownHookManager: Deleting directory /tmp/spark-350e5619-b812-4378-93a5-b61b0d030920
24/07/28 15:53:04 INFO ShutdownHookManager: Deleting directory /tmp/spark-7ba65a6f-87b9-4730-9377-97e42e36b986/pyspark-216c947a-cd8e-499e-b13c-71e57202145f
```

참고설정

```
ENV PDSH_RCMD_TYPE=ssh
ENV JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
ENV SPARK_VERSION=3.5.1
ENV SPARK_HOME=/usr/local/spark
ENV SPARK_CONFIG_DIR=$SPARK_HOME/conf
ENV PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
```

```
RUN wget http://apache.mirror.cdnetworks.com/spark/spark-$SPARK_VERSION-bin-hadoop3.tgz -C /opt
tar -xzf /opt/spark-$SPARK_VERSION-bin-hadoop3.tgz -C /opt
mv /opt/spark-$SPARK_VERSION-bin-hadoop3 $SPARK_HOME && \
```



```
rm /opt/spark-$SPARK_VERSION-bin-hadoop3.tgz
```

```
RUN cp $SPARK_CONF_DIR/spark-env.sh.template $SPARK_CONF_DIR/  
cp $SPARK_CONF_DIR/spark-defaults.conf.template $SPARK_CO
```