

[분산] Docker로 Hadoop 클러스터 구축하기 - Hadoop 3.2.3

태그

작성 완료

목표

- Docker 활용 하둡 클러스터 구성하기
- 클러스터 구성 - Master1, Worker1, Worker2
 - 여기서 Worker는 작성자에 따라 Slave로 쓰기도 한다.

개요

Hadoop을 뿔하고 설치하는 건 불가능하다. hadoop이 돌아갈 OS(Ubuntu)를 돌리고 그 위에서 환경설정을 해주어야한다.

SSAFY 실습은 Hadoop-3.2.2을 사용하지만 현재 미러사이트에서는 해당 버전을 찾을 수 없다. latest, stable 버전 모두 3.5.5이고 Hadoop 2도 2.10.2 정도 배포하는 듯하다.

따라서 이번 클러스터 구축은 사전실습과 가장 가까운 3.2.3 버전을 설치해보기로한다.

클러스터 구축

서버 환경

OS	Ubuntu 22.04.2 LTS
Hadoop	Hadoop-3.2.3
Docker	Docker version 23.0.1

1. 우분투 컨테이너 설치

```
docker run -i -t --name hadoop-base ubuntu
```

Unable to find image 'ubuntu:latest' locally 가 떠도 된다. 알아서 docker hub에서 해당 이미지를 pull해서 가져온다.
실행된 bash에서는 Ctrl + P + Q 로 컨테이너 종료없이 빠져나올 수 있다. 기억하자.

command 설치

```
apt-get update
apt-get upgrade -y
apt-get install -y curl openssh-server rsync wget vim iputils-ping http openjdk-8-jdk ssh
```

Hadoop 다운로드 및 압축해제

```
# 현재 위치에 hadoop 압축파일을 받는다.
wget <http://mirror.naver.com/apache/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz>
# 압축해제
tar zxvf hadoop-3.2.3.tar.gz
# 남아있는 압축 파일 삭제
rm hadoop-3.2.3.tar.gz
# 압축 해제한 hadoop 폴더 이동
mv ./hadoop-3.2.3 /usr/local/hadoop/
```

미러사이트에서 원하는 하둡 버전을 받는다. 미러사이트는 다음과 같은 곳이 있다. 공식문서에서 가이드하는 사이트는 apache의 미러사이트다.

```
http://mirror.navercorp.com/apache/hadoop/common/
http://apache.mirror.cdnetworks.com/hadoop/common/
```

환경변수 셋팅

```
vi ~/.bashrc
```

위의 `~`은 현재 사용자의 홈 디렉토리를 나타낸다. root는 `/root` 디렉토리가, 다른 사용자는 보통 `/home/{username}`의 경로를 가진다. 현재 별도의 유저 생성 및 이동을 거치지 않았기 때문에 `/root/.bashrc`의 파일이 열렸을 것이다. 열린 파일의 하단에 다음을 덧붙인다.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
```

`:wq`으로 파일을 저장하고 다음 명령어를 실행하여 작성한 **환경변수를 적용**시킨다.

```
source ~/.bashrc
```

👉 notice 1

환경에 따라 이후 기술되는WARN util.NativeCodeLoader에 대해 HADOOP_OPTS 환경변수를 수정할 수 있다. 일단 알아두자.

👉 notice 2

이후 복사/붙여넣기 할 일이 많은데, vim 편집기를 열고나면 docker-vim 편집기 클립보드가 따로 인식되는지 `ctrl+v`도 `shift+insert`도 안먹는 경우가 있다. 본인은 `docker cp {filepath} {container}:{filepath}` 명령어와 vim의 `nyy`(n은 숫자) 단축키를 활용하여 해결하였다. 참고 바람~

core-site.xml 세팅

```
vi $HADOOP_HOME/etc/hadoop/core-site.xml
```

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master1:9000</value>
  </property>
</configuration>
```

yarn-site.xml 세팅

```
vi $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

```
</property>
</configuration>
```

mapred-site.xml 세팅

```
vi $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
  </property>
</configuration>
```

masters 세팅

masters 파일이 없을 수도 있다. 그냥 작성한다.

```
vi $HADOOP_HOME/etc/hadoop/masters
```

```
# 다음 작성 후 :wq로 나오기
master1
```

workers 세팅

master와 worker로 사용한 컨테이너를 적는다.

나는 `vi $HADOOP_HOME/etc/hadoop/workers` 를 입력하면 localhost가 한 줄 적혀있었는데, 지우고 다음을 적었다.

```
vi $HADOOP_HOME/etc/hadoop/workers
```

```
# 다음 작성 후 :wq로 나오기
master1
worker1
worker2
```

master, worker 컨테이너 생성

공통 환경 설정이 끝났다. exit로 호스트로 돌아간다.

이미지 커밋

대상 컨테이너를 바탕으로 이미지를 만든다.

```
# 이미지를 만들 hadoop-base의 컨테이너 아이디를 확인한다.
docker ps -a
# 해당 컨테이너 ID를 넣고 이미지를 만든다. 대략 5~10초 소요된다.
docker commit -m "hadoop install in ubuntu" <container id> ubuntu:hadoop
```

컨테이너 생성

root 권한 필요. 이후 각 컨테이너마다 해야할 일 이 있기때문에, 실행 후 매번 `Ctrl+P+Q` 로 빠져나온다.
포트번호 설정은 왜 이렇게 한 건지 필요성이 있는지 잘 모르겠다..

```
docker run -i -t -h master1 --name master1 -p 29870:9870 -p 28088:8088 -p 29888:19888 ubuntu:hadoop
docker run -i -t -h worker1 --name worker1 --link master1:master1 ubuntu:hadoop
docker run -i -t -h worker2 --name worker2 --link master1:master1 ubuntu:hadoop
```

컨테이너 ip 확인 및 host 설정

출력되는 텍스트에서 IPAddress를 확인한다.

```
docker inspect master1 # (172.17.0.4)
docker inspect worker1 # (172.17.0.5)
docker inspect worker2 # (172.17.0.6)
```

각각 컨테이너 bash로 접속해서 다음을 추가한다.

나의 경우 일부 작성된 부분이 있어서 없는 부분만 작성해주었다.

```
docker attach {container name}
vi /etc/hosts

# 다음을 작성하고 나온다.
<IP주소>    master1
<IP주소>    worker1
<IP주소>    worker2
```

master, worker ssh 키 생성

worker끼리 서로 ssh로 통신이 가능하도록 하기 위해, public key를 생성하여 잘 보관해주기로 한다.

다음 작업은 세 컨테이너 모두에서 이뤄져야한다.

Key 생성

```
# 키 생성. 키 파일 위치와 passphrase를 묻는다. 전부 엔터로 넘어감.
ssh-keygen -t rsa
```

```
root@master1:/# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/7HGAXQjrGIF10y6XSN15CvrTQvnxm0BWAc1rnAvxvEY root@master1
The key's randomart image is:
+----[RSA 3072]-----+
|      .+E+*o.      |
|      =+==+      |
|      o.*==+      |
|      oBo=oo      |
|      ...S oo      |
|      = o          |
|      o.+o.o       |
|      . *oo=o      |
|      ..+oo        |
+----[SHA256]-----+
```

Key 복사

각 키를 출력해보면 끝에 `root@master1`와 같이 어떤 사용자로 어떤 컨테이너에 접속 가능한지 나타나있다.

```
cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
cat ~/.ssh/authorized_keys
```

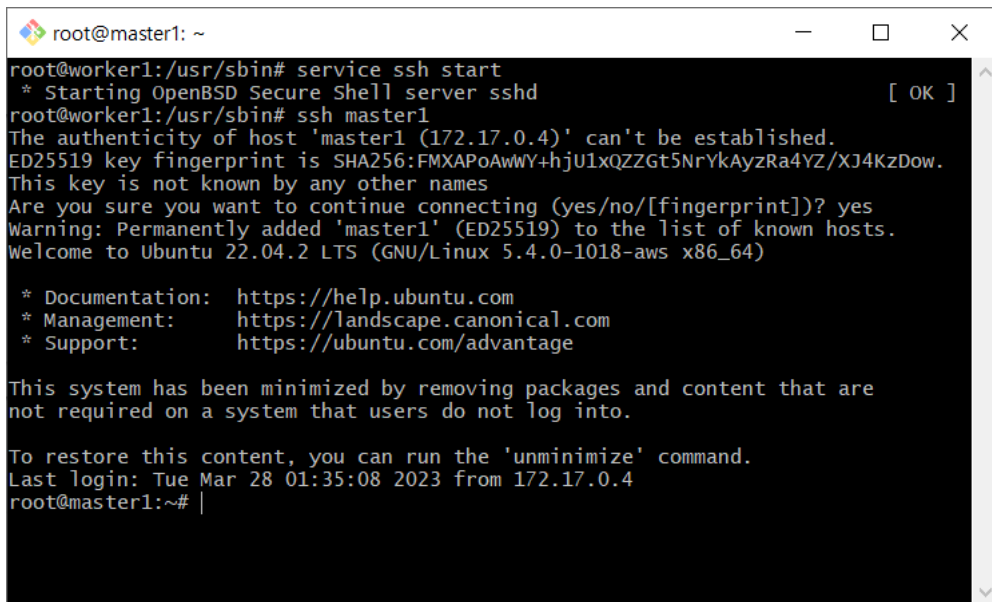
Key 등록

각 컨테이너에서 복사한 public key를 authorized_keys 파일에 저장한다.

```
vi ~/.ssh/authorized_keys
```

접속 확인

```
# 각 컨테이너에서 다음 모두 확인
ssh master1
ssh worker1
ssh worker2
```



```
root@master1: ~
root@worker1:/usr/sbin# service ssh start
* Starting OpenBSD Secure Shell server sshd [ OK ]
root@worker1:/usr/sbin# ssh master1
The authenticity of host 'master1 (172.17.0.4)' can't be established.
ED25519 key fingerprint is SHA256:FMXAPoAwWY+hjU1xQZZGt5NrYkAyzRa4YZ/XJ4KzDow.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'master1' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.4.0-1018-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Tue Mar 28 01:35:08 2023 from 172.17.0.4
root@master1:~#
```

worker1에서 master1로 접속이 된 모습이다!
첫 접속 시에는

```
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

라고 물어보는데, yes라고 하면 무사히 접속된다. (아직 fingerprint는 개념을 잘 모르겠다)

주의

ssh: Could not resolve hostname worker1: Name or service not known

vi ~/etc/hosts에서 각 IP와 서버이름을 제대로 적었는지 확인.

ssh: connect to host {container_name} port 22: Connection refused

접속하려는 컨테이너에서 ssh 서비스가 돌아가고 있는지 확인.

```
service ssh status
```

sshd is not running 이라면 다음으로 ssh 실행

```
service ssh start
```

Master 셋팅

디렉토리 생성

```
cd /usr/local/hadoop
mkdir -p hadoop_tmp/hdfs/namenode
mkdir -p hadoop_tmp/hdfs/datanode
chmod 777 hadoop_tmp/
```

hadoop-env.sh 설정

```
vi $HADOOP_HOME/etc/hadoop/hadoop-env.sh

# 다음 작성하고 나오기
export JAVA_HOME='/usr/lib/jvm/java-8-openjdk-amd64'

export HDFS_NAMENODE_USER=root
export HDFS_DATANODE_USER=root
export HDFS_SECONDARYNAMENODE_USER=root
export YARN_NODEMANAGER_USER=root
export YARN_RESOURCEMANAGER_USER=root
```

hdfs-site.xml 설정

참고한 블로그에서는 namenode와 datanode의 임시 디렉토리 경로가 잘못 지정되어있었다. 나는 위에서 만든 디렉토리 경로로 작성하였다.

```
vi $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>>false</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/usr/local/hadoop/hadoop_tmp/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/usr/local/hadoop/hadoop_tmp/hdfs/datanode</value>
  </property>
</configuration>
```

Worker 세팅

디렉토리 생성

```
cd /usr/local/hadoop
mkdir -p hadoop_tmp/hdfs/datanode
chmod 777 hadoop_tmp/
```

hadoop-env.sh 설정

```
vi $HADOOP_HOME/etc/hadoop/hadoop-env.sh

export JAVA_HOME='/usr/lib/jvm/java-8-openjdk-amd64'
```

hdfs-site.xml 설정

이것도 같다. 참고 블로그는 잘못되어 있다. 위에서 생성한 디렉토리인 `/usr/local/hadoop/hadoop_tmp/hdfs/datanode` 으로 지정해주었다.

```
vi $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>>false</value>
  </property>
```

```
<property>
  <name>dfs.datanode.data.dir</name>
  <value>/usr/local/hadoop/hadoop_tmp/hdfs/datanode</value>
</property>
</configuration>
```

작동 확인

hadoop 시작

master1에서 다음을 실행한다.

```
# 나의 경우 다음을 실행하면 SHUTDOWN_MSG: Shutting down NameNode at master1/172.17.0.4가 출력된다. 다음 진행에 무리는 없다.
hdfs namenode -format
start-all.sh
```

jps 명령어

각 노드에서 jps를 입력하여 결과를 확인한다. 각각 다음과 같이 출력되면 성공이다.

master1

```
root@master1:/usr/local/hadoop# jps
353 DataNode
242 NameNode
754 ResourceManager
1191 Jps
873 NodeManager
525 SecondaryNameNode
```

workers

```
root@worker1:/usr/local/hadoop# jps
243 NodeManager
341 Jps
137 DataNode
```

🔔 notice 3

WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

64비트 운영체제에서 32비트 하둡을 돌려서 생기는 문제.

이 상태에서 hdfs dfs 명령을 입력해도 듣지 않는다. master1에서 다음을 설정한다.

```
vi ~/.bashrc

# export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"에서 다음으로 수정한다.
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

source ~/.bashrc
# 환경변수 적용을 위한 하둡깎다키기
stop-all.sh
start-all.sh
```

다음과 같이 테스트 해보면 경고메시지 없이 문제 없이 잘 출력되는 것을 볼 수 있다.

```
root@master1:/usr/local/hadoop# hdfs dfs -mkdir /test
root@master1:/usr/local/hadoop# hdfs dfs -rm -r /test
Deleted /test
root@master1:/usr/local/hadoop#
```

🔔 notice 4 Datanode가 나타나지 않을 때

각 컨테이너에 들어가서 로그를 확인한다.

```
# ex) cat /usr/local/hadoop/logs/hadoop-root-datanode-master1.log
cat /usr/local/hadoop/logs/hadoop-root-datanode-{container_name}.log
```

좀 더 복잡한 원인이 있을 수 있겠지만, 나는 임시 데이터 디렉토리 문제였다.

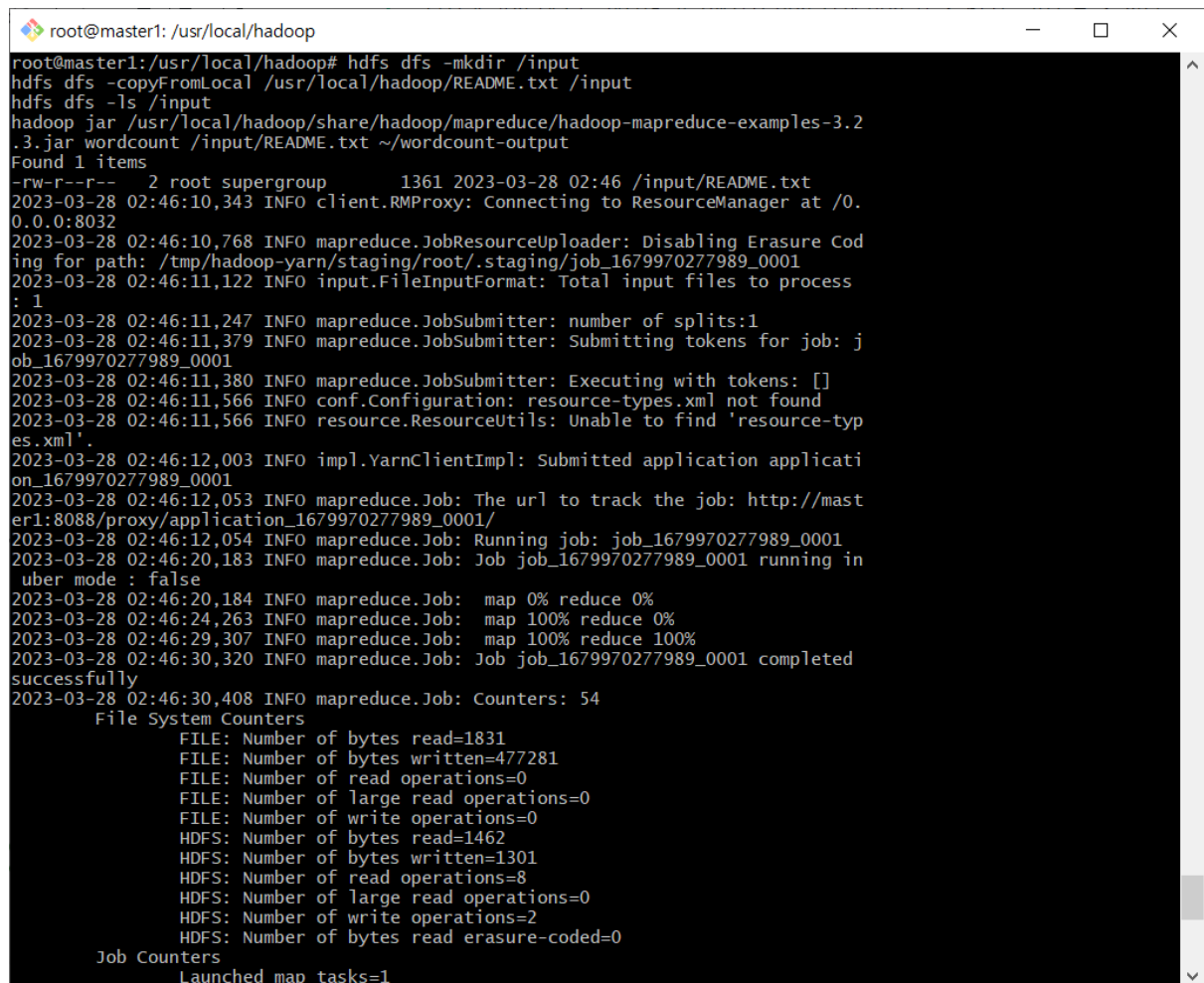
하둡을 여러번 재실행하는 과정에서 임시 데이터가 생성되었고, 해당 데이터 때문에 실행시 문제가 생겼던 것. 초기 설치가 아닌 하둡 사용 중 namenode를 포맷해야한다면 hdfs의 data 폴더를 지우고 다시 생성한 후, namenode를 포맷해야한다고 한다.

```
stop-all.sh
rm -rf /usr/local/hadoop/hadoop_tmp/hdfs/datanode
mkdir -p /usr/local/hadoop/hadoop_tmp/hdfs/datanode
hadoop namenode -format
start-all.sh
```

wordCount 실행

하둡에서 기본으로 제공하는 예제파일을 실행해본다. 실행 내용이 출력될 것이다.

```
hdfs dfs -mkdir /input
hdfs dfs -copyFromLocal /usr/local/hadoop/README.txt /input
hdfs dfs -ls /input
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.3.jar wordcount /input/README.txt ~/wordcount-output
```



```
root@master1: /usr/local/hadoop
root@master1:/usr/local/hadoop# hdfs dfs -mkdir /input
hdfs dfs -copyFromLocal /usr/local/hadoop/README.txt /input
hdfs dfs -ls /input
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.3.jar wordcount /input/README.txt ~/wordcount-output
Found 1 items
-rw-r--r-- 2 root supergroup 1361 2023-03-28 02:46 /input/README.txt
2023-03-28 02:46:10,343 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2023-03-28 02:46:10,768 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1679970277989_0001
2023-03-28 02:46:11,122 INFO input.FileInputFormat: Total input files to process : 1
2023-03-28 02:46:11,247 INFO mapreduce.JobSubmitter: number of splits:1
2023-03-28 02:46:11,379 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1679970277989_0001
2023-03-28 02:46:11,380 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-03-28 02:46:11,566 INFO conf.Configuration: resource-types.xml not found
2023-03-28 02:46:11,566 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-03-28 02:46:12,003 INFO impl.YarnClientImpl: Submitted application application_1679970277989_0001
2023-03-28 02:46:12,053 INFO mapreduce.Job: The url to track the job: http://master1:8088/proxy/application_1679970277989_0001/
2023-03-28 02:46:12,054 INFO mapreduce.Job: Running job: job_1679970277989_0001
2023-03-28 02:46:20,183 INFO mapreduce.Job: Job job_1679970277989_0001 running in uber mode : false
2023-03-28 02:46:20,184 INFO mapreduce.Job: map 0% reduce 0%
2023-03-28 02:46:24,263 INFO mapreduce.Job: map 100% reduce 0%
2023-03-28 02:46:29,307 INFO mapreduce.Job: map 100% reduce 100%
2023-03-28 02:46:30,320 INFO mapreduce.Job: Job job_1679970277989_0001 completed successfully
2023-03-28 02:46:30,408 INFO mapreduce.Job: Counters: 54
File System Counters
  FILE: Number of bytes read=1831
  FILE: Number of bytes written=477281
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=1462
  HDFS: Number of bytes written=1301
  HDFS: Number of read operations=8
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
  HDFS: Number of bytes read erasure-coded=0
Job Counters
  Launched map tasks=1
```

마지막으로 결과로 생성된 파일을 확인해본다.

```
hdfs dfs -ls ~/wordcount-output
hdfs dfs -cat /root/wordcount-output/part-r-000000 | more
```



```

root@master1: /usr/local/hadoop
root@master1:/usr/local/hadoop# hdfs dfs -ls ~/wordcount-output
Found 2 items
-rw-r--r-- 2 root supergroup 0 2023-03-28 02:46 /root/wordcount-output/_SUCCESS
-rw-r--r-- 2 root supergroup 1301 2023-03-28 02:46 /root/wordcount-output/part-r-00000
root@master1:/usr/local/hadoop# hdfs dfs -cat /root/wordcount-output/part-r-00000 | more
(BIS), 1
(ECCN) 1
(TSU) 1
(see 1
5D002.C.1, 1
740.13) 1
<http://www.wassenaar.org/> 1
Administration 1
Apache 1
BEFORE 1
BIS 1
Bureau 1
Commerce, 1
Commodity 1
Control 1
Core 1
Department 1
ENC 1
Exception 1
Export 2
For 1
Foundation 1
Government 1
Hadoop 1

```

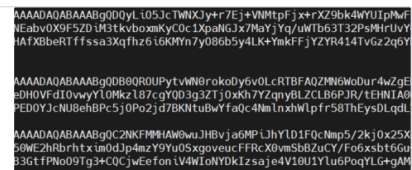
참고

hadoop 3.2.3

하둡 클러스터 구성

단일 ec2 서버에 빅데이터 분석처리를 위한 hadoop cluster 서버를 구축할 일이 생겼다. 원래는 다른 서버에 데이터 노드와 네임노드를 만들어야 하지만 나에게 서버는 하나 밖에 없어 우분투 이미지를 바탕으로 각각 하둡을 설치해 master 1개와 worker 2개

<https://velog.io/@kws04254/하둡-클러스터-구성>



hadoop 2.7.7 - 현재는 다운 받을 수 없다

docker로 hadoop 클러스터 (우분투 기반) 구축하기

1. 우분투 컨테이너 설치 docker run -i -t --name hadoop-base ubuntu ctrl + P, Q 로 컨테이너 중지하지 않고 쉘 빠져나오기 가능 단, docker run -it 옵션인 경우에만 가능 2. openjdk 설치 (컨테이너) add-apt-repository ppa:openjdk-r/ppa apt-get update apt-get install openjdk-8-jdk java -version 3. 하둡 설치 (컨테이너) apt-get

<https://it-sunny-333.tistory.com/79>

All Applications													
App Ring	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes				
1	0	0	0 B	24 GB	0 B	0	24	0	2				
Scheduling Resource Type						Minimum Allocation							
(MEMORY)						<memory1024 vCores1>							
User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus						
cat	word count	MAPREDUCE	default	Tue Apr 21 12:04:14 +0900 2020	Tue Apr 21 12:04:24 +0900 2020	FINISHED	SUCCEEDED						

Hadoop 공식문서

Apache Hadoop

<https://hadoop.apache.org/>

Hadoop Datanode가 실행 안되는 현상

OKKY - hadoop datanode가 실행 안 되는 현상


네임노드 포맷할 때 y가 아닌 y를 눌러서 그랬는지 datanode 실행이 되지 않아서구글링을 통해 다음과 같은 정보를 얻을 수 있었습니다. 하둡을 실행하였는데 DataNode 가 실행되지 않았다. #jps 4825 NameNode 5391 TaskTracker 5242 JobTracker 5477 Jps 5140 SecondaryNameNode 이렇게

<https://okky.kr/questions/426584>

WARN util.NativeCodeLoader 문제 해결

[HADOOP] WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable 이 예러는 64비트 리눅스에서 32비트 하둡을 돌려서 생긴다고 합니다.hadoop-env.sh 나 .bashrc나, 어디에든 다음을 추가해주면 해결됩니다.원래는 \$HADOOP_HOME/lib 으로 되어 있는 부분을 \$HADOOP_HOME/lib/native 로 바꾸면 됩니다. 원래는 export HADOOP_OPTS="\$HADOOP_OPTS -Djava.library.path=\$HADOOP_PREFIX/lib" 이었던 것을

 <https://crmn.tistory.com/7>