Hadoop 포팅 매뉴얼

Tags

1. 개발 도구

- visual studio code
- mobaXterm

2. 개발 환경

- Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1051-aws x86_64)
- python 3.8.10
- fastapi 0.110.0
- pyspark 3.5.1
- hadoop 3.3.6
- numpy 1.24.4

- pandas 2.0.3
- uvicorn 0.29.0
- java 11.0.22

3. ubuntu 환경변수 설정

- ssh 설정 및 인스턴스 접속

```
# ssh key 복사
mkdir -p ~/identity
mv ~/Downloads/hadoop_eco_system.pem ~/identity/hadoop_eco_syste

# pem 키 권한 변경 -> 600이 아닐 경우 보안 취약으로 판단
chmod 600 ~/identity/hadoop_eco_system.pem

# ssh key 만들기
ssh-keygen -t rsa
# enter 세 번 탁! 탁! 탁!

# config 파일 설정
vim ~/.ssh/config

# 아래 내용 추가 후 저장
Host HostName
        HostName ???.???.???.??? # 인스턴스의 Public IP
        User ubuntu
        IdentityFile ~/identity/hadoop_eco_system.pem

# config 파일 권한 수정
chmod 440 ~/.ssh/config
```

- 인스턴스 원격 접속

```
# ssh 접속 테스트
ssh HostName
# 진짜 연결하시겠습니까? yes 입력 후 Enter
Are you sure you want to continue connecting (yes/no/[fingerpri
```

- apt-get 라이브러리 설치

```
# 업데이트 목록 갱신
sudo apt-get -y update
# 현재 패키지 업그레이드
sudo apt-get -y upgrade
# 신규 업데이트 설치
sudo apt-get -y dist-upgrade
# 필요 라이브러리 설치
sudo apt-get install -y vim wget unzip ssh openssh-* net-tools
```

- java 설치

```
# EC2 Ubuntu terminal

# Java 11 설치
sudo apt-get install -y openjdk-11-jdk
# Java 버전 확인
java -version
# Java 경로 확인
sudo find / -name java-11-openjdk-amd64 2>/dev/null
# /usr/lib/jvm/java-11-openjdk-amd64
```

4. 하둡 설치 및 환경 설정 (싱글 노드)

- 하둡 설치

```
# 설치파일 관리용 디렉토리 생성
sudo mkdir /install_dir && cd /install_dir
# Hadoop 3.3.6 설치
sudo wget https://dlcdn.apache.org/hadoop/common/hadoop-3.2.3/ha
# Hadoop 3.3.6 압축 해제
sudo tar -zxvf hadoop-3.3.6.tar.gz -C /usr/local
# Hadoop 디렉토리 이름 변경
sudo mv /usr/local/hadoop-3.3.6 /usr/local/hadoop
```

- 하둡 환경설정

```
# Hadoop 시스템 환경변수 설정
sudo vim /etc/environment

# 아래 내용 추가 후 저장
PATH 뒤에 ":/usr/local/hadoop/bin" 추가
PATH 뒤에 ":/usr/local/hadoop/sbin" 추가
HADOOP_HOME="/usr/local/hadoop"

# 시스템 환경변수 활성화
source /etc/environment

# Hadoop환 사용자 환경변수 설정
sudo echo 'export HADOOP_HOME=/usr/local/hadoop' >> ~/.bashrc
sudo echo 'export HADOOP_COMMON_HOME=$HADOOP_HOME' >> ~/.bashrc
sudo echo 'export HADOOP_HDFS_HOME=$HADOOP_HOME' >> ~/.bashrc
sudo echo 'export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop' >> ~/
sudo echo 'export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop' >> ~/.b
sudo echo 'export HADOOP_YARN_HOME=$HADOOP_HOME' >> ~/.bashrc
sudo echo 'export HADOOP_MAPRED_HOME=$HADOOP_HOME' >> ~/.bashrc
```

```
# 사용자 환경변수 활성화
source ~/.bashrc
```

- hdfs-site.xml 파일 편집

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

```
<configuration>
        <property>
                <name>dfs.replication</name>
                <value>1</value>
        </property>
</configuration>
```

- `dfs.replication` : HDFS 파일 블록 복제 개수 지정한다.

- core-site.xml 파일 편집

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

```
<configuration>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://localhost:9000</value>
        </property>
        <property>
                <name>hadoop.tmp.dir</name>
                <value>/home/ubuntu/media/hadoop</value>
```

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

- fs.default : HDFS의 기본 파일 시스템 디렉토리를 지정한다.
- hadoop.tmp.dir: 임시 파일이 저장될 경로

- yarn-site.xml 파일 편집

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

```
<configuration>

<!-- Site specific YARN configuration properties -->
        <property>
                <name>yarn.nodemanager.aux-services</name>
                <value>mapreduce_shuffle</value>
        </property>
        <property>
                <name>yarn.nodemanager.env-whitelist</name>
                <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_
        </property>
        <property>
                <name>yarn.resourcemanager.hostname</name>
                <value>{your ip}</value>
        </property>
</configuration>
```

- mapred-site.xml 파일 편집

```
sudo vim $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

```
<configuration>
        <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <name>mapreduce.application.classpath</name>
        <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HAD
    </property>
</configuration>
```

- hadoop-env.sh 파일 편집

```
sudo vim $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

- hadoop ver 2 인 경우만

```
# Java
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

# Hadoop
export HADOOP_HOME=/usr/local/hadoop
```

5. spark 설치 및 환경설정

- spark 설치

```
# 설치 관리용 디렉토리 이동
cd /install_dir
```

```
# Spark 3.5.1 설치
sudo wget https://dlcdn.apache.org/spark/spark-3.5.1/spark-3.5.1
# Spark 3.5.1 압축 해제
sudo tar -xzvf spark-3.5.1-bin-hadoop3.tgz -C /usr/local
# Spark 디렉토리 이름 변경
sudo mv /usr/local/spark-3.5.1-bin-hadoop3.tgz /usr/local/spark
```

- python 설치 및 pyspark 설치

```
# Python 설치
sudo apt-get install -y python3-pip
# Python 버전 확인
python3 -V
# PySpark 설치
sudo pip3 install pyspark findspark
```

- Spark 환경설정

```
sudo vim /etc/environment

PATH 뒤에 ":/usr/local/spark/bin" 추가
PATH 뒤에 ":/usr/local/spark/sbin" 추가
SPARK_HOME="/usr/local/spark"

# 시스템 환경변수 활성화
source /etc/environment

#  Spark 사용자 환경변수 설정
echo 'export SPARK_HOME=/usr/local/spark' >> ~/.bashrc

# 사용자 환경변수 활성화
source ~/.bashrc
```

- spark-env.sh 파일 편집

```
cd $SPARK_HOME/conf
sudo cp spark-env.sh.template spark-env.sh

sudo vim spark-env.sh
```

```
export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PYSPARK_PYTHON="/usr/bin/python3"
export HADOOP_HOME="/usr/local/hadoop"
```

- spark-defaults.conf

```
# Spark spark-defaults.conf.template 파일 복사
sudo cp /usr/local/spark/conf/spark-defaults.conf.template /usr/

# Spark spark-defaults.conf 파일 설정
sudo vim /usr/local/spark/conf/spark-defaults.conf
```

```
spark.history.fs.logDirectory   hdfs:///sparklog
spark.eventLog.dir              hdfs:///sparklog
spark.eventLog.enabled          true
spark.history.provider          org.apache.spark.deploy.history
```

- python 환경 설정

```
# 시스템 환경변수 편집
sudo vim /etc/environment
```

```
# 아래 내용 추가 후 저장
PATH 뒤에 ":/usr/bin/python3" 추가

# 시스템 환경변수 활성화
source /etc/environment

#  Python & PySpark 사용자 환경변수 설정
sudo echo 'export PYTHONPATH=/usr/bin/python3' >> ~/.bashrc
sudo echo 'export PYSPARK_PYTHON=/usr/bin/python3' >> ~/.bashrc

# 사용자 환경변수 활성화
source ~/.bashrc
```

6. 하둡 yarn 클러스터 실행

- Namenode 초기화 (최초 한번만 실행)

```
hdfs namenode -format
```

- 실행

```
start-all.sh
```

- jps로 실행 확인

```
jps

NameNode
DataNode
```

```
SecondaryNameNode
ResourceManager
Jps
```