

KOREN SDI기반 오픈플랫폼 실증

■ 문서의 연혁

버전	날짜	작성자	비고
초안 - 0.1	2018. 12. 11	김주민, 김예림, 장병철, 임병돈, 유다금	
0.2	2018. 12. 12	김주민, 김예림, 장병철, 임병돈, 유다금	

본 연구는 한국정보화진흥원(NIA)의 미래네트워크선도시험망 (KOREN) 사업 지원과제의 연구결과로 수행되었음 (17-951-00-001).
 This research was one of KOREN projects supported by National Information Society Agency (17-951-00-001).

Contents

#2-3. Big Data App Container Service Intelligence OpenAPI

1. 개요	4
2 Cloudera manager 설치 및 설정	0
2.1 Cloudera manager 설치	0
2.2 Cloudera manager 설정	0
3 git 설치 및 clone	0
4 mongodb 설치 및 설정	0
4.1 Docker 설치/mongodb 이미지 설치 및 실행	0
4.2 mongodb 설정	0
5 HDFS 폴더 생성 및 권한 설정	0
5.1 HDFS 폴더 생성	0
5.2 HDFS 권한 설정	0
6 스파크 실행 권한 부여	0
7 nodejs 설치	0
8 Swagger 동작을 위한 JAVA 설치	0
9 Spark Page 표시	0
10 Spark Application 추가법	0

#2-3. Big Data App Container Service Intelligence OpenAPI

1. 개요

해당 문서는 “KOREN SDI 기반 오픈플랫폼 실증”의 “[Task 2-3] KOREN SmartX 오픈플랫폼을 위한 Intelligence OpenAPI 설계/구현 및 시험 적용”의 개발자들을 위한 설치 및 사용법을 기술한 가이드 문서다. OS는 과제 구현 환경인 Ubuntu 16.04 버전을 기반으로 서술하고 있으며 그 외의 OS에서의 구현을 위한 설치법은 다소 차이점이 있을 수 있다. 또한 Spark Cluster를 구성한 하드웨어의 성능 부족 여부에 따라 Spark Application 동작에 어려움이 있을 수 있다.

2 Cloudera manager 설치 및 설정

2.1 Cloudera manager 설치

a. `wget https://archive.cloudera.com/cm5/installer/<version>/cloudera-manager-installer.bin`
<version>에 설치하고자 하는 버전을 입력한다. Ubuntu 16.04 환경은 6.0.0 버전부터 지원한다. Cloudera manager는 <https://www.cloudera.com> Cloudera 홈페이지에서 OS에 따른 버전 호환성을 알아본 뒤 설치하는 것이 좋다.

b. 설치한 `cloudera-manager-installer.bin`은 권한을 바꾸고 실행하여 설치한다.

```
chmod u+x cloudera-manager-installer.bin
```

```
sudo ./cloudera-manager-installer.bin
```

설치 과정 중 본 과제에서는 사용하지 않는 Hive, Hue 등은 설치하지 않아도 된다.

2.2 Cloudera manager 설정

a. 브라우저에서 Cloudera manager에 접속하여 설정을 변경한다. 기본 접속 포트는 7180이며, Cloudera manager 버전에 따라 바뀔 수도 있다.

b. 설치 후 기본 ID 와 Password는 admin/admin이다. 로그인 후 우측 상단의 admin 탭의 암호 변경메뉴에서 비밀번호를 바꿔준다.

c. 상단의 검색에서 session_timeout을 검색하여 로그인 session 시간을 원하는 만큼 늘려준다. (기본 30분)

d. 관리 탭의 설정에서 자신의 spark server에 맞도록 설정을 해준다. 해당 설정은 사용하는 하드웨어의 성능이 부족할 경우, 수 많은 경우의 수가 생기므로 해당 문서에서는 기술하지

않는다.

3. git 설치 및 clone

- a. git이 설치되어 있지 않을 경우 `sudo apt-get install git` 명령어로 git을 설치
- b. 그 후 `git clone` 명령어로 본 과제의 git을 clone한다.

4. mongodb 설치 및 설정

mongodb의 설치 방법에는 제한 사항은 없다. 본 과제에서는 Docker를 사용하여 설치하였다.

4.1 Docker 설치/mongodb 이미지 설치 및 실행

- a. `curl -fsSL https://get.docker.com/ | sudo sh`
- b. `docker pull mongodb`
- c. `docker start -it mongodb bash`

4.2 mongodb 설정

위 과정까지 끝났다면 mongodb 이미지에 `bash` `shall`로 접속했을 것이다.

- a. `mongo` 명령어로 `mongo` `shall`로 접속
- b. `use meta --db` 생성
- c. `db.createCollection('apps')` --apps collection 생성
`db.createCollection('datas')` --datas collection 생성
`db.createCollection('results')` --results collection 생성
`db.createCollection('users')` --users collection 생성

각각 업로드 된 application과 data의 metadata/동작이 끝난 application의 결과값/ 로그인 정보를 가진다.

5. HDFS 폴더 생성 및 권한 설정

우선 cloudera의 버전에 따라 hdfs의 port 번호가 바뀌기 때문에 port 번호부터 확인한다.

6.0.0버전 이후는 9870을 이전 버전은 50070을 쓴다.

확인한 버전에 맞도록 config 폴더의 config파일의 설정을 바꿔준다.

5.1 HDFS에 폴더 생성

a. `hdfs dfs -mkdir /result`

b. `hdfs dfs -mkdir /data`

각각 Application 실행 후 결과값과 업로드한 분석 데이터를 저장한다.

5.2 HDFS 권한 설정

`sudo -u hdfs hadoop fs -mkdir /result --HDFS에 폴더 생성`

`sudo -u hdfs hadoop fs -chown (username) /result --폴더의 권한을(username)에게 준다.`

`sudo -u hdfs hadoop fs -mkdir /data --HDFS에 폴더 생성`

`sudo -u hdfs hadoop fs -chown (username) /data --폴더의 권한을(username)에게 준다.`

username은 server를 실행 할 계정의 이름으로 한다.

6. 스파크 실행 권한 부여

Spark Application을 실행하기 위해서 HDFS의 폴더 권한을 설정해야한다.

/tmp 폴더의 권한을 풀지 않을 경우 spark의 mapreduce 작업이 정상작동하지 않는 경우가 있기 때문에 tmp의 권한을 풀어줘야 한다.

`sudo -u hdfs hadoop fs -chmod -R 777 /tmp`

그후 /user/spark 폴더의 권한을 준다.

`sudo -u hdfs hdfs dfs -chown -R (username) /user`

`sudo -u hdfs hdfs dfs -chown -R (username) /user/spark`

username은 server를 실행할 계정의 이름을 입력한다.

7. nodejs 설치

본 과제는 nodejs 기반이기 때문에 nodejs의 설치가 필요하다.

- a. `sudo apt-get install node.js`
- b. clone 폴더에서 `npm install` 명령어로 서버 실행에 필요한 node module을 설치해준다.

8. Swagger 동작을 위한 JAVA 설치

Swagger를 동작하기 위한 JAVA를 설치한다.

- a. `sudo apt-get install openjdk-8-jre`
- b. `sudo apt-get install openjdk-8-jdk`

9. Spark Page 표시

xframe을 사용하고 있기 때문에 Chrome 기준으로 ignore-xframe headers 확장프로그램을 추가할 필요가 있다.

- a.
<https://chrome.google.com/webstore/detail/ignore-x-frame-headers/gleekbfjekiniecknbkamfmkohkpodhe>에 접속
- b. Chrome에 추가하기 버튼 클릭

10. Spark Application 추가법

본 과제에서 사용하는 Spark Application은 일반적인 Spark Application에 동적으로 Parameter를 입력하여 작동할 수 있도록 특별 제작되어야 한다. 또한 Application 업로드 할 때 Application의 정보를 담은 Meta Data 파일을 JSON 형식으로 함께 업로드 할 필요가 있다.

- a. `wordcount_search.py`의 예제


```
from pyspark import SparkContext
import argparse

#spark context
sc = SparkContext()

#make & received outer argument
parser = argparse.ArgumentParser()

parser.add_argument("--file", help=": file name")
parser.add_argument("--word", help=": search word name")
parser.add_argument("--optArg", help=": (optional arg)")

filename = parser.parse_args().file
search = parser.parse_args().word
optionalArg = parser.parse_args()

#read file route
text_file = sc.textFile("hdfs:///data/" + filename)

#check optionalArg
if optionalArg.optArg:
    ..... print("select optional arg")

#word search and count
counts = text_file.flatMap(lambda line: line.split(" "))\
    .filter(lambda i : i == search)\
    .map(lambda word: (word, 1)) \
    .reduceByKey(lambda a, b: a + b)

#print wordcount
print counts.collect()
```

예제에서는 parser를 이용하여 shall에서 spark-submit 명령어를 사용할 때 Application 내부

로 인자 값을 받을 수 있도록 만들었다.

ex) spark-submit wordcount_search.py --file=AtoZ.txt --word=A

filename은 --file=(filename) 식으로 HDFS에 저장된 분석 데이터의 이름을 지정하여 동적으로 분석 데이터를 선택하기 위해서 만들어진 인자 값이다. filename 인자 값은 해당 시스템에 업로드하는 모든 Spark Application이 필수적으로 가져야하는 인자 값이다.

search는 --word=(search word) 식으로 사용자가 검색하기를 원하는 단어를 알기 위한 인자 값이다. search는 해당 Application을 사용하기 위하여 필요한 필수적인 인자 값으로 해당하는 인자 값이 없으면 Application이 작동할 수 없을 경우 반드시 받아야 할 경우 사용할 수 있는 방식이다.

optionalArg는 선택적인 인자값으로 해당 Application을 동작하는데 선택적으로 필요한 인자 값을 받을 때 사용할 수 있는 방식이다. 해당 예제에서는 선언만 했고 사용하지는 않았다.

b. wordcount_search.py의 meta data 파일, wordcount_search.json 예제

```
{
  "appName" : "wordcount_search.py",
  "description" : "Apps that count the number of specific characters",
  "author" : {
    "name" : "choi",
    "email" : "ghwchlaks@naver.com"
  },
  "parameters" : [
    {
      "name" : "--word",
      "description" : "Option to count the number of specific words",
      "inputType" : {
        "boxType" : "text",
        "min" : 1,
        "max" : 10
      },
      "default" : "A"
    }
  ],
  "version" : "0.0.1"
}
```

appName : Application의 이름

description : Application의 간단한 동작 설명

author : Application 제작자의 이름과 이메일 정보

parameters : 해당 Application에서 사용하는 parameter들과 각각의 정보, Application에 기술한 인자 값과 꼭 맞춰줘야한다.

version : Application의 버전 정보