|  |
| --- |
| SeokRae Kim  2017-7-13 |

|  |
| --- |
| FICS |
| Hadoop |
| 설치 가이드 |

목 차

[**I.** **하둡 살펴보기** 1](#_Toc489297539)

[**1.** **하둡(Hadoop)이란?** 1](#_Toc489297540)

[**2.** **하둡 프로젝트 설치 프로그램** 1](#_Toc489297541)

[**II.** **하둡 프로젝트 개발 준비** 2](#_Toc489297542)

[**2.** **VMware 설정** 2](#_Toc489297543)

[**1)** **CentOS(Linux) 설치** 2](#_Toc489297544)

[**2)** **CentOS(Linux) 설정** 11](#_Toc489297545)

[**3.** **하둡 실행 계정 생성** 19](#_Toc489297546)

[**4.** **IP 설정** 20](#_Toc489297547)

[**5.** **호스트 설정** 23](#_Toc489297548)

[**6.** **인코딩(Encoding)설정** 26](#_Toc489297549)

[**7.** **자바 설치** 29](#_Toc489297550)

[**1)** **자바 다운로드** 29](#_Toc489297551)

[**2)** **자바 프로그램 확인** 29](#_Toc489297552)

[**3)** **기존 자바 삭제** 30](#_Toc489297553)

[**4)** **자바 경로 이동** 31](#_Toc489297554)

[**5)** **자바 설치** 32](#_Toc489297555)

[**6)** **자바 Path 설정** 34](#_Toc489297556)

[**8.** **VMware PC 4대 만들기** 35](#_Toc489297557)

[**1)** **VMware Clone 만들기 (wikibooks02 ~ 04 PC)** 35](#_Toc489297558)

[**2)** **Mac Address 설정 (wikibooks02 ~ 04 PC)** 38](#_Toc489297559)

[**3)** **Ip 수정 (wikibooks02 ~ 04 PC)** 40](#_Toc489297560)

[**4)** **호스트 수정 (wikibooks02 ~ 04 PC)** 41](#_Toc489297561)

[**9.** **SSH 설정** 42](#_Toc489297562)

[**10.** **하둡(Hadoop) 설치 및 설정** 46](#_Toc489297563)

[**1)** **하둡 다운 (NameNode에서 실행)** 46](#_Toc489297564)

[**2)** **하둡 압축 해제 및 심볼릭 링크 설정** 47](#_Toc489297565)

[**3)** **하둡 설정파일 수정** 49](#_Toc489297566)

[**4)** **하둡 Path 설정 (wikibooks01 ~ 04 PC)** 55](#_Toc489297567)

[**5)** **하둡 실행** 56](#_Toc489297568)

[**6)** **WordCount 예제 실행** 58](#_Toc489297569)

[**III.** **이클립스 설치** 59](#_Toc489297570)

[**1.** **이클립스 다운** 59](#_Toc489297571)

[**2.** **이클립스 압축풀기** 60](#_Toc489297572)

[**3.** **이클립스 실행** 61](#_Toc489297573)

[**IV.** **메이븐 설치** 63](#_Toc489297574)

[**1.** **메이븐 다운** 63](#_Toc489297575)

[**2.** **메이븐 압축 풀기** 63](#_Toc489297576)

[**3.** **메이븐 Path 설정** 64](#_Toc489297577)

[**4.** **메이븐 설치 확인** 65](#_Toc489297578)

[**5.** **이클립스와 연동하기** 65](#_Toc489297579)

[**6.** **메이븐을 이용한 HDFS 예제 실행** 72](#_Toc489297580)

[**7.** **메이븐을 이용한 MapReduce – WordCount 예제 실행** 74](#_Toc489297581)

[**V.** **GIT(GitHub) 연동 – 인터넷에서 원하는 방식 찾아 볼 것** 84](#_Toc489297582)

[**1.** **GitHub 설치** 84](#_Toc489297583)

[**2.** **GitHub - Repositoy 생성** 84](#_Toc489297584)

[**VI.** **MapReduce 애플리케이션 개발** 85](#_Toc489297585)

[**1.** **항공 데이터 분석 시작하기** 85](#_Toc489297586)

[**2.** **공통 클래스 구현** 88](#_Toc489297587)

[**3.** **항공 출발 지연 데이터 분석** 88](#_Toc489297588)

[**VII.** **소스코드** 89](#_Toc489297589)

1. **하둡 살펴보기**
   1. **하둡(Hadoop)이란?**

* 하둡은 대용량 데이터를 분산 처리할 수 있는 자바 기반의 오픈 소스 프레임워크이다.
* 하둡은 분산 파일 시스템인 HDFS(Hadoop Distributed File System)에 데이터를 저장하고, 분산 처리 시스템인 맵리듀스(MapReduce)를 이용해 데이터를 처리 한다.
* 하둡은 여러 대의 서버에 데이터를 저장하고, 데이터가 저장된 각 서버에서 동시에 데이터를 처리하는 방식이다.
  1. **하둡 프로젝트 설치 프로그램**

|  |  |  |  |
| --- | --- | --- | --- |
| **프로그램** | **종류** | **버전** | **사이트** |
| **Virtual Machine** |  | 12.0 | <https://www.vmware.com/products/workstation/workstation-evaluation.html> |
| **Linux** |  | 7.3(1611) | <http://isoredirect.centos.org/centos/7/isos/x86_64/CentOS-7-x86_64-DVD-1611.iso> |
| **JAVA** | java에 대한 이미지 검색결과 | 1.8.0\_131 | <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html> |
| **Hadoop** |  | 1.2.1 | <http://hadoop.apache.org/releases.html> |
| **Eclipse** |  | Oxygen | <https://www.eclipse.org/downloads/?> |
| **Maven** |  | 3.5.0 | <https://maven.apache.org/download.cgi> |

1. **하둡 프로젝트 개발 준비**
   1. **실행 모드 설정  
      - 가상 분산(Pseudo-distributed) 모드**하나의 장비에 모든 하둡 환경설정을 하고, 하둡 서비스도 이 장비에서만 제공하는 방식, HDFS와 맵리듀스와 관련된 데몬을 하나의 장비에서만 실행하게 된다. 주로 하둡을 처음 공부하는 분들이 이와 같은 방식으로 테스트 환경을 구성한다.  
      **- 완전 분산(Fully distributed) 모드**  
      여러 대의 장비에 하둡이 설치된 경우, 하둡으로 라이브 서비스를 하게 될 경우 이와 같은 방식으로 구성한다.
   2. **VMware 설정**
      1. **CentOS(Linux) 설치**

|  |
| --- |
| **#OS 설치하기 – “New Virtual Machine”** |
| C:\Users\kslbs\Desktop\vm 설정\1.png |

|  |
| --- |
| **#설치 방식 선택 – “Typical”** |
| C:\Users\kslbs\Desktop\vm 설정\2.png |

|  |
| --- |
| **#OS Install 방식 선택 – “I will install the operating system later”** |
| C:\Users\kslbs\Desktop\vm 설정\3.png |

|  |
| --- |
| **#OS 종류 선택 – “CentOS 64-bit”** |
| C:\Users\kslbs\Desktop\vm 설정\4.png |

|  |
| --- |
| **#경로 선택 – “C\hadoop” (폴더 생성)** |
| C:\Users\kslbs\Desktop\vm 설정\5.png |
| **#경로 선택 –** “**C:\hadoop\**wibibooks01” |
| C:\Users\kslbs\Desktop\vm 설정\6.png |

|  |
| --- |
| **#Disk Capacity 설정 – “100GB”** |
| C:\Users\kslbs\Desktop\vm 설정\7.png |

|  |
| --- |
| **#Setting확인 – “Finish”** |
| C:\Users\kslbs\Desktop\vm 설정\8.png |

|  |
| --- |
| **#가상 OS사양 설정 - ”Edit virtual Machine Settings”** |
| C:\Users\kslbs\Desktop\vm 설정\9.png |
| **#Memory – “2GB(2048)”** |
| C:\Users\kslbs\Desktop\vm 설정\10.png |

|  |
| --- |
| **#Processors 변경 – “2 Core”** |
| C:\Users\kslbs\Desktop\vm 설정\11.png |

|  |
| --- |
| **#ISO 이미지 찾기 – “Browse…”** |
| C:\Users\kslbs\Desktop\vm 설정\12.png |

|  |
| --- |
| **#ISO 이미지 선택 – “CentOS-7-x86\_64-DVD-1611.iso”** |
| C:\Users\kslbs\Desktop\vm 설정\13.png |

|  |
| --- |
| **#ISO 이미지 넣기 끝** |
| C:\Users\kslbs\Desktop\vm 설정\14.png |

|  |
| --- |
| **#VMware에서 OS 화면 크기조절 – “1024 x 768”** |
| C:\Users\kslbs\Desktop\하둡 설치\vm 설정\15.png |

|  |
| --- |
| **#VMware 설정 끝 – “Power on this virtual machine”** |
| C:\Users\kslbs\Desktop\vm 설정\16.png |

|  |
| --- |
| **#CentOS 설치 – “Install CentOS Linux 7”** |
| C:\Users\kslbs\Desktop\vm 설정\17.png |

* + 1. **CentOS(Linux) 설정**

|  |
| --- |
| **#CentOS 설치 시작 – “한국어”** |
| C:\Users\kslbs\Desktop\vm 설정\18.png |

|  |
| --- |
| **#설치 요약 메뉴 시작 – “네트워크 및 호스트명”** |
| C:\Users\kslbs\Desktop\vm 설정\25.png |

|  |
| --- |
| **#이더넷 연결 – “끔 -> 켬”** |
| C:\Users\kslbs\Desktop\vm 설정\21.pngC:\Users\kslbs\Desktop\vm 설정\22.png |

|  |
| --- |
| **#설치 요약 – “소프트웨어 선택”** |
| C:\Users\kslbs\Desktop\vm 설정\20.png |

|  |
| --- |
| **#OS 소프트웨어 선택 – “서버 – GUI 사용(DNS, SMTP, FTP)”** |
| C:\Users\kslbs\Desktop\vm 설정\23.png |

|  |
| --- |
| **#서버 – GUI사용(“개발용 도구”)** |
| C:\Users\kslbs\Desktop\vm 설정\24.png |

|  |
| --- |
| **#CentOS 설정 끝 – “설치 시작”** |
| C:\Users\kslbs\Desktop\vm 설정\25.png |

|  |
| --- |
| **#설치 중 – “사용자 설정 메뉴 설정하기”** |
| C:\Users\kslbs\Desktop\vm 설정\28.png |

|  |
| --- |
| **#사용자 설정 메뉴 – “Root 암호 설정”** |
| C:\Users\kslbs\Desktop\vm 설정\26.png |

|  |
| --- |
| **#사용자 설정 메뉴 – “Hadoop 로컬 계정 생성”** |
| C:\Users\kslbs\Desktop\vm 설정\27.png |

|  |
| --- |
| **#CentOS 설정 끝 – “재부팅”** |
| C:\Users\kslbs\Desktop\하둡 설치\vm 설정\29.png |

|  |
| --- |
| **#초기 설정 메뉴** |
| C:\Users\kslbs\AppData\Local\Microsoft\Windows\INetCache\Content.Word\30.png |

|  |
| --- |
| **#라이센스 정보 - ”약관에 동의합니다”** |
| C:\Users\kslbs\Desktop\하둡 설치\vm 설정\31.png |

|  |
| --- |
| **#설정 완료 - ”설치 완료”** |
| C:\Users\kslbs\Desktop\하둡 설치\vm 설정\32.png |

|  |
| --- |
| **#CentOS 7 설치 끝** |
| C:\Users\kslbs\Desktop\하둡 설치\vm 설정\33.png |

* 1. **하둡 실행 계정 생성**

|  |
| --- |
| # |
| 기본적으로는 하둡 프로그램 전용 계정을 새로 만들어 사용해야 하지만  현재 테스트로 하기 때문에 root 계정으로 진행한다. |

* 1. **IP 설정**

|  |
| --- |
| **#최종 목표** |
| C:\Users\kslbs\Desktop\하둡 설치\설치 목표\1.png  마스터(네임노드) 하나에 데이터 노드 3개로 완전 분산 모드 테스트 서비스를 구현한다. |

|  |
| --- |
| **#우측 상단 – “Wired Settings”** |
| **C:\Users\kslbs\Desktop\IP 설정\1.png** |

|  |
| --- |
| **#현 네트워크 상태 IPv4를 확인 – “우측 하단 설정 버튼”** |
| **C:\Users\kslbs\Desktop\IP 설정\2.png** |

|  |
| --- |
| **#현 네트워크 설정 상태 – “IPv4 탭”** |
| **C:\Users\kslbs\Desktop\IP 설정\3.png** |

|  |
| --- |
| **#Addresses – “Manual”** |
| **C:\Users\kslbs\Desktop\IP 설정\4.png** |

|  |
| --- |
| **#Addresses – “IP, S/M, G/W, DNS” 수정 -> Apply** |
| **C:\Users\kslbs\Desktop\IP 설정\5.png** |

|  |
| --- |
| **#네트워크 – “OFF -> ON”** |
| **C:\Users\kslbs\Desktop\IP 설정\6.png** |

|  |
| --- |
| **#IP 확인 – “ifconfig”** |
| **C:\Users\kslbs\Desktop\IP 설정\7.png** |

* 1. **호스트 설정**

|  |
| --- |
| **#하둡 테스트 환경 구성** |
| |  |  | | --- | --- | | **호스트명** | **하둡 설치 내용** | | wikibooks01 | 네임노드(NameNode) | | wikibooks02 | 보조네임노드(SecondaryNameNode), 데이터노드(DataNode) | | wikibooks03 | 데이터노드 | | wikibooks04 | 데이터노드 | |

|  |
| --- |
| **#호스트 설정하기** |
| |  | | --- | | **#호스트명 변경** | | [hadoop@localhost ~]$ hostnamectl set-hostname wikibooks01  새 콘솔 열어 변경확인 |   C:\Users\kslbs\Desktop\IP 설정\8.png   |  | | --- | | **#호스트명 정보 파일 변경** | | [hadoop@localhost ~]$ vi /etc/hosts |  |  | | --- | | **#호스트명 정보 파일 변경** | | 192.168.42.120 wikibooks01  192.168.42.121 wikibooks02  192.168.42.122 wikibooks03  192.168.42.123 wikibooks04 | | :wq |   C:\Users\kslbs\Desktop\IP 설정\9.png |

|  |
| --- |
| **#호스트 적용 확인 – “새 콘솔창 켜서 확인”** |
|  |

* 1. **인코딩(Encoding)설정**

|  |
| --- |
| **#현 인코딩 상태 확인** |
| [hadoop@wikibooks01 ~]$ echo $LANG  en\_US.utf8 |

|  |
| --- |
| **#인코딩 방식을 변경** |
| [hadoop@wikibooks01 ~]$ su  Password:  [root@wikibooks01 hadoop]# vi /etc/sysconfig/i18n  LANG="ko\_KR.UTF-8"  SUPPORTED="en\_US.UTF-8:en\_US:ko\_KR.eucKR:ko\_KR:ko"  SYSFONT="lat0-sun16"  SYSFONTACM="8859-15" |

|  |
| --- |
| **#인코딩 적용 후 확인** |
| [root@wikibooks01 hadoop]# source /etc/sysconfig/i18n  [root@wikibooks01 hadoop]# echo $LANG  ko\_KR.UTF-8 |

|  |
| --- |
| **#언어관련 파라미터 UTF-8인지 확인 명령어 – “locale”** |
| [root@wikibooks01 hadoop]# locale  LANG=ko\_KR.UTF-8  LC\_CTYPE="ko\_KR.UTF-8"  LC\_NUMERIC=ko\_KR.UTF-8  LC\_TIME=ko\_KR.UTF-8  LC\_COLLATE="ko\_KR.UTF-8"  LC\_MONETARY=ko\_KR.UTF-8  LC\_MESSAGES="ko\_KR.UTF-8"  LC\_PAPER=ko\_KR.UTF-8  LC\_NAME="ko\_KR.UTF-8"  LC\_ADDRESS="ko\_KR.UTF-8"  LC\_TELEPHONE="ko\_KR.UTF-8"  LC\_MEASUREMENT=ko\_KR.UTF-8  LC\_IDENTIFICATION="ko\_KR.UTF-8"  LC\_ALL= |

* 1. **자바 설치**

|  |
| --- |
| **#자바 설치 목표경로 – “/usr/local/”** |
|  |

* + 1. **자바 다운로드**

|  |
| --- |
| **#자바 다운로드 – “jdk-8u131-linux-x64.tar.gz”** |
| Linux x64 176.95 MB jdk-8u131-linux-x64.tar.gz  <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> |

* + 1. **자바 프로그램 확인**

|  |
| --- |
| **#자바 설치 확인** |
| [root@wikibooks01 hadoop]# rpm -qa|grep jdk  copy-jdk-configs-1.2-1.el7.noarch  java-1.8.0-openjdk-headless-1.8.0.131-3.b12.el7\_3.x86\_64  java-1.8.0-openjdk-1.8.0.131-3.b12.el7\_3.x86\_64 |

* + 1. **기존 자바 삭제**

|  |
| --- |
| **#기존 자바 프로그램 삭제** |
| [root@wikibooks01 hadoop]# yum –y remove copy-jdk-configs-1.2-1.el7.noarch |

* + 1. **자바 경로 이동**

|  |
| --- |
| **#자바 경로 확인** |
| [root@wikibooks01 hadoop]# ls  Desktop Documents Downloads Music Pictures Public Templates Videos  [root@wikibooks01 hadoop]# cd Downloads  [root@wikibooks01 Downloads]# ls  jdk-8u131-linux-x64.tar.gz |

|  |
| --- |
| **#자바 설치 경로로 tar파일 이동** |
| [root@wikibooks01 Downloads]# mv jdk-8u131-linux-x64.tar.gz /usr/local/  [root@wikibooks01 Downloads]# ls  [root@wikibooks01 Downloads]# cd /usr/local  [root@wikibooks01 local]# ls  bin games jdk-8u131-linux-x64.tar.gz lib64 sbin src  etc include lib libexec share |

* + 1. **자바 설치**

|  |
| --- |
| **#자바 tar파일 권한 설정** |
| [root@wikibooks01 local]# ls  bin games jdk-8u131-linux-x64.tar.gz lib64 sbin src  etc include lib libexec share  [root@wikibooks01 local]# chmod 755 jdk-8u131-linux-x64.tar.gz  [root@wikibooks01 local]# ls  bin games jdk-8u131-linux-x64.tar.gz lib64 sbin src  etc include lib libexec share    권한 설정 이후 파일이 빨간색에서 초록색으로 바뀌면서 접근 가능해 진다. |

|  |
| --- |
| **#자바 설치 및 심볼릭 링크** |
| [root@wikibooks01 local]# tar xvfz jdk-8u131-linux-x64.tar.gz  [root@wikibooks01 local]# ln -s jdk1.8.0\_131/ java  [root@wikibooks01 local]# ls -al  합계 181192  drwxr-xr-x. 13 root root 197 7월 17 22:46 .  drwxr-xr-x. 13 root root 155 7월 17 13:44 ..  drwxr-xr-x. 2 root root 6 11월 6 2016 bin  drwxr-xr-x. 2 root root 6 11월 6 2016 etc  drwxr-xr-x. 2 root root 6 11월 6 2016 games  drwxr-xr-x. 2 root root 6 11월 6 2016 include  lrwxrwxrwx. 1 root root 13 7월 17 22:46 java -> jdk1.8.0\_131/  -rwxr-xr-x. 1 hadoop hadoop 185540433 7월 17 22:00 jdk-8u131-linux-x64.tar.gz  drwxr-xr-x. 8 10 143 255 3월 15 17:35 jdk1.8.0\_131  drwxr-xr-x. 2 root root 6 11월 6 2016 lib  drwxr-xr-x. 2 root root 6 11월 6 2016 lib64  drwxr-xr-x. 2 root root 6 11월 6 2016 libexec  drwxr-xr-x. 2 root root 6 11월 6 2016 sbin  drwxr-xr-x. 5 root root 49 7월 17 13:44 share  drwxr-xr-x. 2 root root 6 11월 6 2016 src |

* + 1. **자바 Path 설정**

|  |
| --- |
| **#profile 파일 수정** |
| [root@wikibooks01 local]# vi /etc/profile  #Java Path  export JAVA\_HOME=/usr/local/java  export PATH=$PATH:$JAVA\_HOME/bin  export CLASS\_PATH="."    [root@wikibooks01 local]# source /etc/profile |

|  |
| --- |
| **#자바 버전 확인** |
| [root@wikibooks01 ~]#java –version |

* 1. **VMware PC 4대 만들기**
     1. **VMware Clone 만들기 (wikibooks02 ~ 04 PC)**

|  |
| --- |
| **#가상머신 복사하기(Clone)** |
|  |

|  |
| --- |
| # |
|  |

|  |
| --- |
| # |
|  |

|  |
| --- |
| **#가상 이미지 경로 설정하기** |
|  |

|  |
| --- |
| **#클론 만드는 중** |
|  |

|  |
| --- |
| **#클론 만들기 완료** |
|  |

|  |
| --- |
| **#가상 이미지 4대 설정 완료** |
|  |

* + 1. **Mac Address 설정 (wikibooks02 ~ 04 PC)**

|  |
| --- |
| **#Edit** |
| * 여기는 첫 번째 wikibooks01 PC외에 모든 PC의 설정을 동일하게 합니다. |

|  |
| --- |
| **#Network Adapter – “Advanced”** |
|  |

|  |
| --- |
| **#Mac Address 바꾸기 – “Generate”** |
|  |

* + 1. **Ip 수정 (wikibooks02 ~ 04 PC)**

|  |
| --- |
| **#IPv4 Address 수정** |
| IP 마지막주소 121 ~ 123로 알맞게 수정  **Wikibooks02**    **Wikibooks03**    **Wikibooks04** |

* + 1. **호스트 수정 (wikibooks02 ~ 04 PC)**

|  |
| --- |
| **#복사한 가상 이미지 Hostname 수정** |
| Wikibooks02 ~ 04 모든 복사된 PC에 알맞은 설정을 적용한다. |

|  |
| --- |
| **#인터넷 OFF -> ON 및 가상머신 재부팅으로 HostName 적용되었는지 확인** |
| **Wikibooks02**    **Wikibooks03** |

* 1. **SSH 설정**

|  |
| --- |
| **#SSH 공개키 만들기 (wikibooks01 ~ 04 PC)** |
| [root@wikibooks01 ~]# ssh-keygen -t rsa -f ~/.ssh/id\_rsa  [root@wikibooks02 ~]# ssh-keygen -t rsa -f ~/.ssh/id\_rsa  [root@wikibooks03 ~]# ssh-keygen -t rsa -f ~/.ssh/id\_rsa  [root@wikibooks04 ~]# ssh-keygen -t rsa -f ~/.ssh/id\_rsa |

|  |
| --- |
| **#.ssh 폴더 확인 (wikibooks01 ~ 04 PC)** |
| [root@wikibooks01 ~]# ls –al |

|  |
| --- |
| **#NameNode PC 공개키 복사** |
| [root@wikibooks01 .ssh]# cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys  [root@wikibooks01 .ssh]# ls -al |

|  |
| --- |
| **#NameNode에서 SSH를 통해 나머지 PC의 id\_rsa.pub 내용을 복사** |
| [root@wikibooks01 .ssh]# ssh root@wikibooks02 cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys  [root@wikibooks01 .ssh]# ssh root@wikibooks03 cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys  [root@wikibooks01 .ssh]# ssh root@wikibooks04 cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys  세 명령어에 대해 계속 접근 할 건지에 대답 yes와 Password를 입력한다. |

|  |
| --- |
| **#NameNode의 authorized\_keys에 나머지 공개키 내용이 복사되었는지 확인** |
| [root@wikibooks01 .ssh]# cat authorized\_keys |

|  |
| --- |
| **#4개 PC의 공개키의 내용이 있는 authorized\_keys를 배포** |
| [root@wikibooks01.ssh]# scp authorized\_keys root@wikibooks02:~/.ssh/authorized\_keys  root@wikibooks02's password:  authorized\_keys 100% 1990 1.9KB/s 00:00  [root@wikibooks01 .ssh]# scp authorized\_keys root@wikibooks03:~/.ssh/authorized\_keys  root@wikibooks03's password:  authorized\_keys 100% 1990 1.9KB/s 00:00  [root@wikibooks01 .ssh]# scp authorized\_keys root@wikibooks04:~/.ssh/authorized\_keys  root@wikibooks04's password:  authorized\_keys 100% 1990 1.9KB/s 00:00 |

|  |
| --- |
| **#SSH 연결 확인** |
| [root@wikibooks01 ~]# ssh wikibooks02  [root@wikibooks01 ~]# ssh wikibooks03  [root@wikibooks01 ~]# ssh wikibooks04  비밀번호를 치지 않고 접속 될 경우 성공 |

* 1. **하둡(Hadoop) 설치 및 설정**

**하둡을 설치 할 때 주의 할 점 \*\***

**/etc/profile**파일 Path설정, 하둡을 관리할 계정의 **.bash\_profile**파일 Path설정

**Xml** 파일을 수정 할 때 하둡 경로를 자신이 설치한 경로로 수정해서 입력 할 것

* + 1. **하둡 다운 (NameNode에서 실행)**

|  |
| --- |
| **#하둡 다운로드 – 명령어** |
| Wget <http://ftp.daum.net/apache/hadoop/common/hadoop-1.2.1/hadoop-1.2.1.tar.gz> |

|  |
| --- |
| **#하둡 다운로드 – 직접 다운로드** |
| 버전 1.2.1 다운 받을 수 있는 URL - hadoop-1.2.1.tar.gz  <http://www-eu.apache.org/dist/hadoop/core/hadoop-1.2.1/> |

* + 1. **하둡 압축 해제 및 심볼릭 링크 설정**

|  |
| --- |
| **#다운받은 하둡을 설치할 경로로 이동** |
| [root@wikibooks01 Downloads ]# mv hadoop-1.2.1.tar.gz /usr/local/  그림과는 다른 경로일 수 있지만 목표는 /usr/local/ 경로에 hadoop-1.2.1.tar.gz를 옮기는 것 |

|  |
| --- |
| **#하둡 파일 권한 수정** |
| [root@wikibooks01 local]# chmod 755 hadoop-1.2.1.tar.gz  [root@wikibooks01 local]# ls –al |

|  |
| --- |
| **#하둡 tar 파일 압축 해제** |
| [root@wikibooks01 local]# tar xvfz hadoop-1.2.1.tar.gz |

|  |
| --- |
| **#하둡 경로 심볼릭 링크 생성** |
| [root@wikibooks02 local]# ln -s hadoop-1.2.1 hadoop  [root@wikibooks02 local]# ls -al |

* + 1. **하둡 설정파일 수정**
       1. **hadoop-env.sh 파일 수정**

|  |
| --- |
| **#하둡을 실행하는 셸 스크립트 파일에서 필요한 환경변수를 설정** |
| JDK 경로, 클래스 패스, 데몬 실행 옵션 등의 환경변수 설정  [root@wikibooks01 conf]# vi hadoop-env.sh  export JAVA\_HOME=/usr/local/java    export HADOOP\_PID\_DIR=/usr/local/hadoop/pids |

* + - 1. **.bash\_profile 파일 수정 (wikibooks01 ~ 04 PC)**

|  |
| --- |
| **#하둡 운영 시 하둡 홈 디렉터리에 손쉽게 접근하기 위한 설정** |
| [root@wikibooks01 ~]# vi .bash\_profile  HADOOP\_HOME=/usr/local/hadoop  :wq    #Path관련 파일은 수정 이후 적용  [root@wikibooks01 ~]# source .bash\_profile |

* + - 1. **masters 파일 수정**

|  |
| --- |
| **#보조네임 노드를 실행할 서버를 설정** |
| [root@wikibooks01 conf]# vi masters  wikibooks02 |

* + - 1. **slaves 파일 수정**

|  |
| --- |
| **#데이터 노드를 실행할 서버를 설정** |
| [root@wikibooks01 conf]# vi slaves  wikibooks02  wikibooks03  wikibooks04 |

* + - 1. **core-site.xml 파일 수정**

|  |
| --- |
| **#HDFS의 기본이름 설정, 임시 데이터 저장하기 위한 경로 설정** |
| [root@wikibooks01 conf]# vi core-site.xml  <configuration>  <property>  <name>fs.default.name</name>  <value>hdfs://wikibooks01:9000</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>/usr/local/hadoop-data</value>  </property>  </configuration> |

* + - 1. **hdfs-site.xml 파일 수정**

|  |
| --- |
| **#HDFS관련 설정** |
| [root@wikibooks01 conf]# vi hdfs-site.xml   * HDFS의 저장될 데이터 복제본의 개수 * 네임 노드용 웹서버의 주소 * 보조네임노드용 웹서버의 주소 값   <configuration>  <property>  <name>dfs.replication</name>  <value>3</value>  </property>  <property>  <name>dfs.http.address</name>  <value>wikibooks01:50070</value>  </property>  <property>  <name>dfs.secondary.http.address</name>  <value>wikibooks02:50090</value>  </property>  </configuration> |

* + - 1. **mapred-site.xml 파일 수정**

|  |
| --- |
| #잡트래커(JobTracker) 데몬의 주소 설정 |
| [root@wikibooks01 conf]# vi mapred-site.xml  <configuration>  <property>  <name>mapred.job.tracker</name>  <value>wikibooks01:9001</value>  </property>  </configuration> |

* + - 1. **하둡 환경설정 파일 배포**

|  |
| --- |
| **#하둡 설정파일 폴더 압축** |
| [root@wikibooks01 local]# tar -zcvf hadoop.tar.gz hadoop-1.2.1/  [root@wikibooks01 local]# ls –al |

|  |
| --- |
| **#하둡 설정파일 파일 배포** |
| [root@wikibooks01 local]# scp –rp hadoop.tar.gz root@wikibooks02:/usr/local/  hadoop.tar.gz 100% 61MB 60.9MB/s 00:01  [root@wikibooks01 local]# scp –rp hadoop.tar.gz root@wikibooks03:/usr/local/  hadoop.tar.gz 100% 61MB 60.9MB/s 00:00  [root@wikibooks01 local]# scp –rp hadoop.tar.gz root@wikibooks04:/usr/local/  hadoop.tar.gz 100% 61MB 60.9MB/s 00:01 |

* + - 1. **하둡 설치**

|  |
| --- |
| **#SSH 명령어를 이용해 전체 서버에 하둡 설치** |
| [root@wikibooks01 local]# ssh root@wikibooks02 "cd /usr/local; tar xfz hadoop.tar.gz; rm hadoop.tar.gz"  [root@wikibooks01 local]# ssh root@wikibooks03 "cd /usr/local; tar xfz hadoop.tar.gz; rm hadoop.tar.gz"  [root@wikibooks01 local]# ssh root@wikibooks04 "cd /usr/local; tar xfz hadoop.tar.gz; rm hadoop.tar.gz" |

|  |
| --- |
| **#NameNode로부터 배포 받은 각 데이터 노드에서 하둡 심볼릭 링크 걸기** |
| **wikibooks02 ~ 04 PC**  [root@wikibooks02 local]# ln –s hadoop-1.2.1 hadoop  [root@wikibooks03 local]# ln –s hadoop-1.2.1 hadoop  [root@wikibooks04 local]# ln –s hadoop-1.2.1 hadoop |

* + 1. **하둡 Path 설정 (wikibooks01 ~ 04 PC)**

|  |
| --- |
| **#하둡 경로 설정** |
| [root@wikibooks01 ~]# vi /etc/profile  export HADOOP\_HOME=/usr/local/hadoop  export PATH=$PATH:$HADOOP\_HOME/bin    [root@wikibooks01 ~]# source /etc/profile |

* + 1. **하둡 실행**
       1. **하둡 네임노드 포맷**

|  |
| --- |
| [root@wikibooks01 hadoop]# ./bin/hadoop namenode -format |

* + - 1. **하둡 데몬 실행 및 실행 여부 확인**

|  |
| --- |
| [root@wikibooks01 hadoop]# ./bin/start-all.sh  [root@wikibooks01 hadoop]# jps    [root@wikibooks02 hadoop]# jps  Jps(Java Virtual Machine Process Status Tool)은 시스템에서 실행 중인 자바 프로세스를 출력   * 네임노드용 서버에서는 **NameNode, JobTracker**가 출력 되어야 한다. * 데이터노드용 서버에서는 **TaskTracker, DataNode**가 출력 되어야 한다. * wikibooks02에는 보조네임노드를 설치 했으므로 **SecondaryNameNode**가 추가로 출력되어야 한다. |

|  |
| --- |
| **#웹 인터페이스로 하둡 서버 확인** |
| 웹에서 접근하려면 편의상 방화벽을 내려준다.  [root@wikibooks01 hadoop]# systemctl stop firewalld  [root@wikibooks01 hadoop]# systemctl disable firewalld |

* + 1. **WordCount 예제 실행**

|  |
| --- |
| # |
| [root@wikibooks01 hadoop]# hadoop fs -put /usr/local/hadoop/conf/hadoop-env.sh conf/hadoop-env.sh  [root@wikibooks01 hadoop]# hadoop jar hadoop-examples-\*.jar wordcount conf/hadoop-env.sh wordcount\_output |

1. **이클립스 설치**
   1. **이클립스 다운**

|  |
| --- |
| **#이클립스 다운로드** |
| [**https://www.eclipse.org/downloads/**](https://www.eclipse.org/downloads/)  **eclipse로 검색해서 이클립스 다운로드 사이트에 들어가면 운영체제에 맞는 버전을 띄워준다.** |

* 1. **이클립스 압축풀기**

|  |
| --- |
| **#이클립스 tar 파일 옮기기** |
| [root@wikibooks01 ]# cd /home/hadoop/다운로드  [root@wikibooks01 다운로드]# mv eclipse-inst-linux64.tar.gz /usr/local  [root@wikibooks01 다운로드]# cd /usr/local |

|  |
| --- |
| **#이클립스 압축 풀기 및 설치** |
| [root@wikibooks01 local]# tar xfz eclipse-inst-linux64.tar.gz  [root@wikibooks01 local]# cd eclipse-installer  [root@wikibooks01 eclipse-installer]# ls  artifacts.xml eclipse-inst features p2 readme  configuration eclipse-inst.ini icon.xpm plugins  [root@wikibooks01 eclipse-installer]# ./eclipse-inst  편의상 하둡 계정 경로(/home/hadoop/)에 eclipse를 설치 하도록 한다. |

* 1. **이클립스 실행**

|  |
| --- |
| **#이클립스 경로로 들어가기** |
| [root@wikibooks01 local]# cd /home/hadoop/  [root@wikibooks01 hadoop]# ls  eclipse eclipse-workspace 공개 다운로드 문서 바탕화면 비디오 사진 서식 음악  [root@wikibooks01 hadoop]# cd eclipse/jee-oxygen/eclipse  [root@wikibooks01 eclipse]# ./eclipse |

1. **메이븐 설치**
   1. **메이븐 다운**

|  |
| --- |
| **#메이븐 다운로드** |
| [**https://maven.apache.org/download.cgi**](https://maven.apache.org/download.cgi) |

* 1. **메이븐 압축 풀기**

|  |
| --- |
| **#메이븐 압축 풀기** |
| [root@wikibooks01 eclipse]# cd /home/hadoop  [root@wikibooks01 hadoop]# ls  eclipse eclipse-workspace 공개 다운로드 문서 바탕화면 비디오 사진 서식 음악  [root@wikibooks01 hadoop]# cd 다운로드  [root@wikibooks01 다운로드]# ls  dapache-maven-3.5.0-bin.tar.gz  [root@wikibooks01 다운로드]# mv apache-maven-3.5.0-bin.tar.gz /usr/local  [root@wikibooks01 다운로드]# ls  [root@wikibooks01 다운로드]# cd /usr/local  [root@wikibooks01 local]# ls  apache-maven-3.5.0-bin.tar.gz etc hadoop-1.2.1 java lib64 share  bin games hadoop-data jdk1.8.0\_141 libexec src  eclipse-installer hadoop include lib sbin  [root@wikibooks01 local]# tar xfz apache-maven-3.5.0-bin.tar.gz |

* 1. **메이븐 Path 설정**

|  |
| --- |
| **#메이븐 Path 설정** |
| [root@wikibooks01 local]# vi /etc/profile  #Maven Path  export PATH=$PATH:/usr/local/apache-maven-3.5.0/bin    [root@wikibooks01 local]# source /etc/profile |

* 1. **메이븐 설치 확인**

|  |
| --- |
| **#메이븐 실행** |
| [root@wikibooks01 local]# mvn |

* 1. **이클립스와 연동하기**

|  |
| --- |
| **#새 프로젝트 만들기** |
| “New” -> “New Maven Project” 선택 |
| 첫 번째와 두 번째 체크박스를 체크  “Use default workspace location”에 있는 “Browse” 버튼 클릭 프로젝트에서 사용할 경로 설정 |

|  |
| --- |
| **#패키지, 폴더 이름 설정** |
| **Group Id: com.wikibooks.hadoop**  **Artifact Id: hadoop-beginner-examples**  **Version:** |

|  |
| --- |
| **#프로젝트 생성 후 pom.xml 설정** |
| 바로 전 설정 내용과 일치 하는지 계정 비밀번호와 hadoop 경로가 맞는지 확인 \*\*  **pom.xml 내용 수정**  **<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">**  **<modelVersion>4.0.0</modelVersion>**  **<groupId>com.wikibooks.hadoop</groupId>**  **<artifactId>hadoop-beginner-examples</artifactId>**  **<version>0.0.1-SNAPSHOT</version>**    **<properties>**  **<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>**  **<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>**  **<hadoop.version>1.2.1</hadoop.version>**  **</properties>**    **<repositories>**  **<repository>**  **<id>repository.jboss.org</id>**  **<url>https://repository.jboss.org/nexus/content/repositories/releases/</url>**  **<snapshots>**  **<enabled>false</enabled>**  **</snapshots>**  **</repository>**  **</repositories>**    **<dependencies>**  **<dependency>**  **<groupId>org.apache.hadoop</groupId>**  **<artifactId>hadoop-core</artifactId>**  **<version>${hadoop.version}</version>**  **</dependency>**  **<dependency>**  **<groupId>org.apache.hadoop</groupId>**  **<artifactId>hadoop-client</artifactId>**  **<version>${hadoop.version}</version>**  **</dependency>**  **</dependencies>**    **<build>**  **<plugins>**  **<plugin>**  **<groupId>org.apache.maven.plugins</groupId>**  **<artifactId>maven-antrun-plugin</artifactId>**  **<version>1.8</version>**  **<dependencies>**  **<dependency>**  **<groupId>org.apache.ant</groupId>**  **<artifactId>ant-jsch</artifactId>**  **<version>1.9.1</version>**  **</dependency>**  **<dependency>**  **<groupId>com.jcraft</groupId>**  **<artifactId>jsch</artifactId>**  **<version>0.1.50</version>**  **</dependency>**  **</dependencies>**  **<executions>**  **<execution>**  **<phase>install</phase>**  **<goals>**  **<goal>run</goal>**  **</goals>**  **<configuration>**  **<tasks>**  **<taskdef name="SCP" classname="org.apache.tools.ant.taskdefs.optional.ssh.Scp" classpathref="maven.dependency.classpath" />**  **<scp file="target/${project.artifactId}-${project.version}.jar" todir="root:1234@wikibooks01:/usr/local/hadoop" trust="true"/>**  **</tasks>**  **</configuration>**  **</execution>**  **</executions>**  **</plugin>**  **</plugins>**  **</build>**    **</project>** |

|  |
| --- |
| **# clean install** |
| **프로젝트 우측 클릭 해서 “maven build”**  **Base directory:** pom.xml이 존재하는 곳    **Clean install 완료 화면** |

|  |
| --- |
| **# Maven Install** |
|  |

* 1. **메이븐을 이용한 HDFS 예제 실행**

|  |
| --- |
| **#SingleFileWriteRead.java** |
| package wikibooks.hadoop.chapter03;  import org.apache.hadoop.conf.Configuration;  import org.apache.hadoop.fs.FSDataInputStream;  import org.apache.hadoop.fs.FSDataOutputStream;  import org.apache.hadoop.fs.FileSystem;  import org.apache.hadoop.fs.Path;  public class SingleFileWriteRead {  public static void main(String[] args) {  if(args.length !=2) {  System.err.println("Usage: SingleFileWriteRead <filename> <contents>");  System.exit(2);  }    try {  Configuration conf = new Configuration();  FileSystem hdfs = FileSystem.get(conf);    Path path = new Path(args[0]);  if(hdfs.exists(path)) {  hdfs.delete(path, true);  }    FSDataOutputStream outStream = hdfs.create(path);  outStream.writeUTF(args[1]);  outStream.close();    FSDataInputStream inputStream = hdfs.open(path);  String inputString = inputStream.readUTF();  inputStream.close();    System.out.println("## inputString:" + inputString);  } catch(Exception e) {  e.printStackTrace();  }  }  }  **Java 파일 생성** |

|  |
| --- |
| **#하둡에서 클래스 실행** |
| [root@wikibooks01 hadoop]# ./bin/hadoop jar hadoop-beginner-examples-0.0.1-SNAPSHOT.jar wikibooks.hadoop.chapter03.SingleFileWriteRead input.txt Hello,HDFS  ## inputString:Hello,HDFS  [root@wikibooks01 hadoop]# ./bin/hadoop fs -ls input\*  Found 1 items  -rw-r--r-- 3 root supergroup 12 2017-07-24 21:00 /user/root/input.txt  [root@wikibooks01 hadoop]# ./bin/hadoop fs -cat input.txt  Hello,HDFS |

* 1. **메이븐을 이용한 MapReduce – WordCount 예제 실행**
* **Mapper 구현**
* **Reducer 구현**
* **Driver(Mapper, Reducer를 실행하는 클래스) 구현**
* **WordCount 빌드**
* **WordCount 실행**

|  |
| --- |
| **#WordCount 예제를 위한 프로젝트 생성** |
| **“New” -> “Maven Project” 생성**    **New Maven Project 생성**    **Group Id: com.wikibooks.hadoop**  **Artifact Id: hadoop-beginner-wordcount**    **Pom.xml 설정**  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com.wikibooks.hadoop</groupId>  <artifactId>hadoop-beginner-wordcount</artifactId>  <version>0.0.1-SNAPSHOT</version>      <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <hadoop.version>1.2.1</hadoop.version>  </properties>    <repositories>  <repository>  <id>repository.jboss.org</id>  <url>https://repository.jboss.org/nexus/content/repositories/releases/</url>  <snapshots>  <enabled>false</enabled>  </snapshots>  </repository>  </repositories>    <dependencies>  <dependency>  <groupId>org.apache.hadoop</groupId>  <artifactId>hadoop-core</artifactId>  <version>${hadoop.version}</version>  </dependency>  <dependency>  <groupId>org.apache.hadoop</groupId>  <artifactId>hadoop-client</artifactId>  <version>${hadoop.version}</version>  </dependency>  </dependencies>    <build>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-antrun-plugin</artifactId>  <version>1.8</version>  <dependencies>  <dependency>  <groupId>org.apache.ant</groupId>  <artifactId>ant-jsch</artifactId>  <version>1.9.1</version>  </dependency>  <dependency>  <groupId>com.jcraft</groupId>  <artifactId>jsch</artifactId>  <version>0.1.50</version>  </dependency>  </dependencies>  <executions>  <execution>  <phase>install</phase>  <goals>  <goal>run</goal>  </goals>  <configuration>  <tasks>  <taskdef name="SCP" classname="org.apache.tools.ant.taskdefs.optional.ssh.Scp" classpathref="maven.dependency.classpath" />  <scp file="target/${project.artifactId}-${project.version}.jar" todir="root:1234@wikibooks01:/usr/local/hadoop" trust="true"/>  </tasks>  </configuration>  </execution>  </executions>  </plugin>  </plugins>  </build>  </project> |

|  |
| --- |
| **#MapReduce 프로그램 만들기** |
| * **Driver**   Mapper, Reducer 클래스 실행을 위한 클래스   * **Mapper**   **문장을 입력 받아 글자 수를 계산하는 클래스**   * **Reducer**   **글자와 글자수로 구성된 입력 파라미터를 받아 글자 수를 합산 해 출력하는 클래스** |

|  |
| --- |
| **#매퍼(Mapper) 구현** |
| package wikibooks.hadoop.chapter04;  import java.io.IOException;  import java.util.StringTokenizer;  import org.apache.hadoop.io.IntWritable;  import org.apache.hadoop.io.LongWritable;  import org.apache.hadoop.io.Text;  import org.apache.hadoop.mapreduce.Mapper;  public class WordCountMapper extends Mapper<LongWritable, Text, Text, IntWritable> {  private final static IntWritable one = new IntWritable(1);  private Text word = new Text();    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {  StringTokenizer itr = new StringTokenizer(value.toString());    while(itr.hasMoreTokens()) {  word.set(itr.nextToken());  context.write(word,one);  }  }  } |

|  |
| --- |
| **#리듀서(Reducer) 구현** |
| package wikibooks.hadoop.chapter04;  import java.io.IOException;  import org.apache.hadoop.io.IntWritable;  import org.apache.hadoop.io.Text;  import org.apache.hadoop.mapreduce.Reducer;  public class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable>{  private IntWritable result = new IntWritable();    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {  int sum = 0;  for (IntWritable val : values) {  sum += val.get();  }  result.set(sum);  context.write(key, result);  }  } |

|  |
| --- |
| **#드라이버(Driver) 클래스 구현** |
| package wikibooks.hadoop.chapter04;  import java.io.IOException;  import org.apache.hadoop.conf.Configuration;  import org.apache.hadoop.fs.Path;  import org.apache.hadoop.io.IntWritable;  import org.apache.hadoop.io.Text;  import org.apache.hadoop.mapreduce.Job;  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;  public class WordCount {  public static void main(String[] args) throws IOException, ClassNotFoundException, InterruptedException {  Configuration conf = new Configuration();  if(args.length != 2) {  System.err.println("Usage : WordCount <input> <output>");  System.exit(2);  }  Job job = new Job(conf, "WordCount");    job.setJarByClass(WordCount.class);  job.setMapperClass(WordCountMapper.class);  job.setReducerClass(WordCountReducer.class);    job.setInputFormatClass(TextInputFormat.class);  job.setOutputFormatClass(TextOutputFormat.class);    job.setOutputKeyClass(Text.class);  job.setOutputValueClass(IntWritable.class);    FileInputFormat.addInputPath(job, new Path(args[0]));  FileOutputFormat.setOutputPath(job, new Path(args[1]));    job.waitForCompletion(true);  }  } |

|  |
| --- |
| **#WordCount 빌드** |
| **Goals: “clean install”** |

|  |
| --- |
| **#WordCount 실행** |
| **# 파일 만들기**  [root@wikibooks01 hadoop]# touch test.txt  [root@wikibooks01 hadoop]# vi test.txt  read a book  write a book    **#파일 HDFS에 업로드**  [root@wikibooks01 hadoop]# ./bin/hadoop fs –put test.txt test.txt  **#HDFS에 업로드된 파일 확인**  [root@wikibooks01 hadoop]# ./bin/hadoop fs –ls    **#WordCount 실행**  [root@wikibooks01 hadoop]# ./bin/hadoop jar hadoop-beginner-wordcount-0.0.1-SNAPSHOT.jar wikibooks.hadoop.chapter04.WordCount test.txt wordcount\_output    **#출력 페이지의 핵심**  17/07/25 17:50:48 INFO mapred.JobClient: Map input records=2  맵의 입력으로 2건의 데이터가 입력  17/07/25 17:50:48 INFO mapred.JobClient: Map output records=6  맵의 출력으로 6건이 출력  17/07/25 17:50:48 INFO mapred.JobClient: Reduce input groups=4  17/07/25 17:50:48 INFO mapred.JobClient: Reduce output records=4  리듀스의 출력으로 4건이 생성 |

|  |
| --- |
| **#WordCount 실행 결과 확인** |
| **#HDFS 폴더 확인**  [root@wikibooks01 hadoop]# ./bin/hadoop fs –ls  **#리듀서 클래스의 출력 파일 조회**  [root@wikibooks01 hadoop]# ./bin/hadoop fs –cat wordcount\_output/part-r-00000  (맵리듀스 잡의 출력 경로에는 리듀서의 출력 데이터 외에도 각 단계별 로그 파일 및 설정 정보가 저장 되어있기 때문에 part-r-00000파일에 접근 하여 출력)  [root@wikibooks01 hadoop]# ./bin/hadoop fs –ls wordcount\_output |

|  |
| --- |
| **#웹에서 확인** |
| * Job이 성공했는지(**Status**) 확인 * 작업 시간이 얼마나 걸렸는지(**Finished in**)확인 * HDFS에서 읽고 쓰인 파일과 맵리듀스에서 처리된 정보확인 |

|  |
| --- |
| **맵리듀스 개발 과정 정리**   1. 맵리듀스 단계별로 사용할 파라미터를 키와 값의 형태로 설계 2. 매퍼 클래스를 상속받아 WordCountMapper클래스를 구현 3. 리듀서 클래스를 상속받아 WordCountReducer클래스를 구현 4. 맵리듀스 잡을 실행할 잡 객체를 생성하고 실행 5. 완성된 맵리듀스 클래스를 하둡에서 실행 6. 맵리듀스 출력 결과물이 원하는 대로 나왔는지 확인 |

**Reference**

GitHub - <https://github.com/SeokRae/hadoop_beginner>