

빅데이터시스템

하둡(Hadoop)

목 차

01 빅데이터의 이해와 활용

빅데이터의 이해, 빅데이터의 활용

02 하둡(Hadoop)

하둡의개념, 환경설정, 하둡설치

03 Spark설치

04 전달 사항

로드맵, 교재

1. 빅데이터의 이해와 활용

⌚ 빅데이터의 이해

■ 빅데이터의 분류

구분	설명	수집 및 처리 난이도
정형 데이터	<ul style="list-style-type: none"> 고정된 필드에 저장 관계형 데이터베이스처럼 스키마 형식에 맞게 저장 예: RDB, 스프레드시트 	<ul style="list-style-type: none"> 내부 시스템에 의한 데이터라 수집하기 쉬움 파일 형태의 스프레드시트는 형식을 가지고 있어 처리하기 쉬움 처리 난이도: 하
반정형 데이터	<ul style="list-style-type: none"> 고정된 필드에 저장되어 있지는 않지만 메타 데이터나 스키마 등을 포함 예: XML, HTML, JSON, 웹 문서, 웹 로그 	<ul style="list-style-type: none"> API 형태로 제공되므로 데이터 처리 기술이 필요함 처리 난이도: 중
비정형 데이터	<ul style="list-style-type: none"> 데이터 구조가 일정하지 않음 규격화된 데이터 필드에 저장되지 않음 예: 소셜 데이터, 텍스트 문서, 이미지/동영상/음성 데이터, 문서 파일(PDF) 	<ul style="list-style-type: none"> 파일을 데이터 형태로 파싱해야 하므로 처리하기 어려움 처리 난이도: 상

1. 빅데이터의 이해와 활용

⌚ 빅데이터의 이해

■ 빅데이터의 분석환경

요소	과거의 데이터 분석 환경	현재의 빅데이터 분석 환경
데이터	<ul style="list-style-type: none"> 정형화된 수치 중심의 자료 	<ul style="list-style-type: none"> 비정형의 다양한 데이터 예: 문자 데이터(SMS, 검색어), 영상 데이터(CCTV, 동영상), 위치 데이터 등
하드웨어	<ul style="list-style-type: none"> 고가의 저장 장치 데이터베이스 대규모 데이터웨어하우스 	<ul style="list-style-type: none"> 클라우드 컴퓨팅: 비용 대비 효율성 증대
소프트웨어 분석 방법	<ul style="list-style-type: none"> 관계형 데이터베이스: RDBMS 통계 패키지: SAS, SPSS 데이터 마이닝 머신러닝 지식 발견 	<ul style="list-style-type: none"> 오픈 소스 형태의 무료 소프트웨어 오픈 소스 통계 솔루션: R 텍스트 마이닝 오피니언 마이닝 감성 분석

1. 빅데이터의 이해와 활용

⌚ 빅데이터의 이해

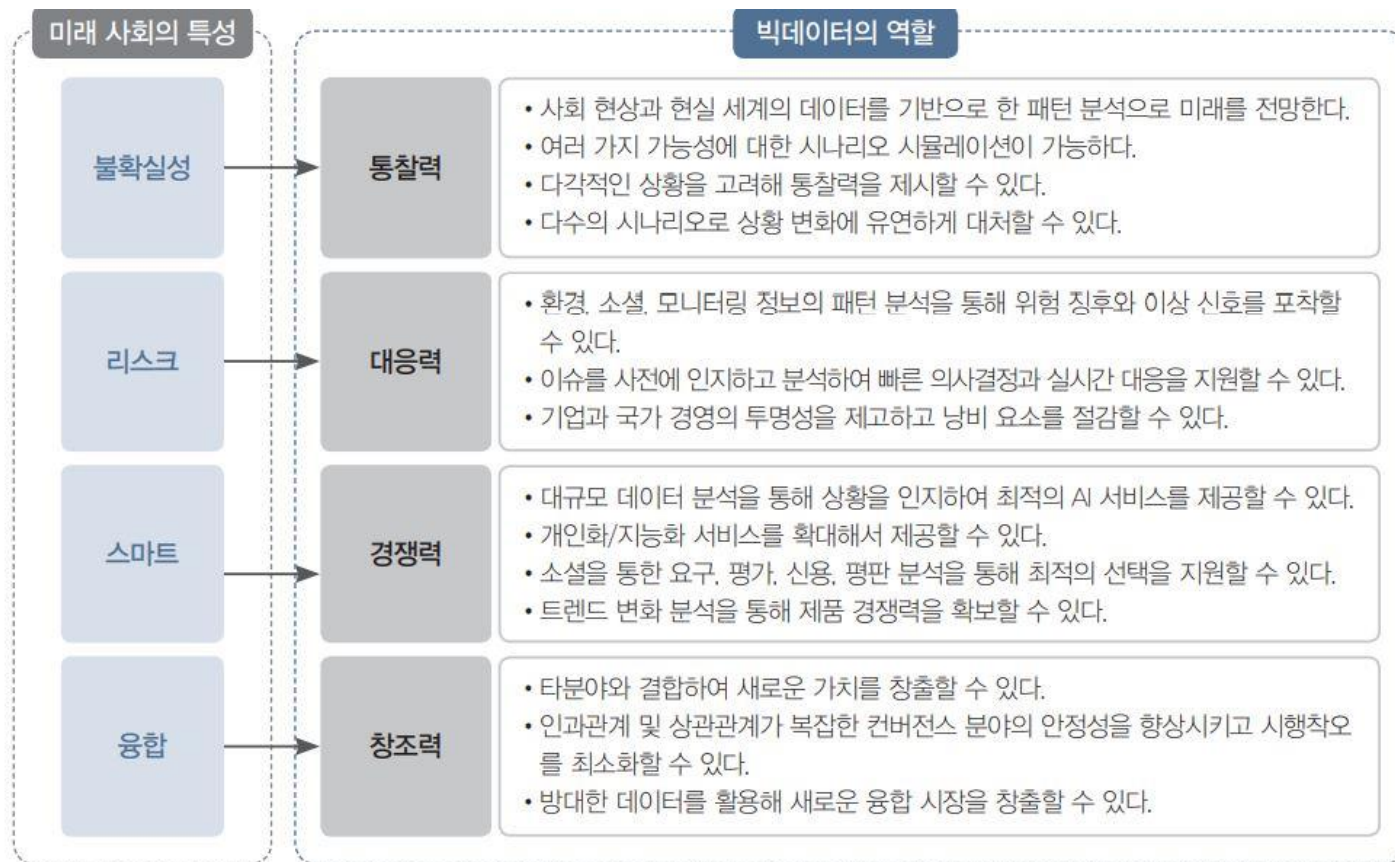
■ 빅데이터의 처리방식

구분	이전의 데이터 처리 방식	빅데이터 처리 방식
데이터 트래픽	<ul style="list-style-type: none"> • 테라바이트 수준 	<ul style="list-style-type: none"> • 페타바이트 수준: 최소 100테라바이트 이상 • 정보의 장기간 수집 및 분석 • 방대한 처리량
데이터 유형	<ul style="list-style-type: none"> • 정형 데이터 중심 	<ul style="list-style-type: none"> • 비정형 데이터 비중이 높음: SNS 데이터, 로그 파일, 클릭스트림 데이터, 콜센터 로그 통신, CDR 로그 등 • 처리 복잡성 증대
프로세스 및 기술	<ul style="list-style-type: none"> • 단순한 프로세스 및 기술 • 정형화된 처리 및 분석 결과 • 원인 및 결과 규명 중심 	<ul style="list-style-type: none"> • 다양한 데이터 소스와 복잡한 로직 처리 • 처리 복잡도가 높아 분산 처리 기술 필요 • 새롭고 다양한 처리 방법 필요: 정의된 데이터 모델/상관관계/절차 등이 없음 • 상관관계 규명 중심 • 하둡, NoSQL 등 개방형 소프트웨어 사용

1. 빅데이터의 이해와 활용

⌚ 빅데이터의 활용

■ 빅데이터의 역할



1. 빅데이터의 이해와 활용

⌚ 빅데이터의 활용

■ 기업의 성공적인 활용

조건	내용
리더십	목표 설정을 위해 빅데이터를 활용한 성공이 무엇인지를 명확히 정의하고 이를 강력하게 추진할 수 있는 리더십이 필요하다.
역량 관리	데이터 과학자, 시스템 개발자 등과 같은 전문 인력의 역량을 관리해야 한다.
기술 도입	빅데이터 관련 시스템에 최적화된 기술을 도입하고 조직 내·외부의 데이터를 통합 및 가시화하는 기술을 도입해야 한다.
의사결정	빅데이터 분석에 기반한 의사결정으로 조직의 유연성을 보장해야 한다.
기업 문화	빅데이터를 활용할 수 있는 조직 문화가 필요하다.

1. 빅데이터의 이해와 활용

⌚ 빅데이터의 활용

■ 활용 가능한 빅데이터 발견

단계	내용과 과제	방법
저장	<ul style="list-style-type: none"> • 조직의 독자적인 데이터를 생성 및 저장하는 단계 • 인터넷을 통해 외부 데이터 수집(검색) 가능 • 데이터의 신뢰성과 품질 제고를 위한 노력이 필요 	생성, 저장, 수집(검색)
공유	<ul style="list-style-type: none"> • 기업 데이터를 외부 기관과 상호 교환하는 단계 • 1:1 또는 1:n의 공유 및 연계 가능 	연계, 공유
통합	<ul style="list-style-type: none"> • 특정 활동이나 목적을 위하여 연합, 그룹, 클럽이 상호 협력하는 공동의 장(집단)을 형성하는 단계 • 표준 데이터 풀과 연계하여 국경을 초월한 정보 교환과 상호 이용이 가능 	참여, 협력
공동 창출	<ul style="list-style-type: none"> • 오픈 플랫폼으로 데이터를 공유하는 단계 • 상호 협력과 참여로 공동의 자원을 창조 	오픈, 창조

1. 빅데이터의 이해와 활용

⌚ 빅데이터의 활용

■ 처리 단계별 기술영역

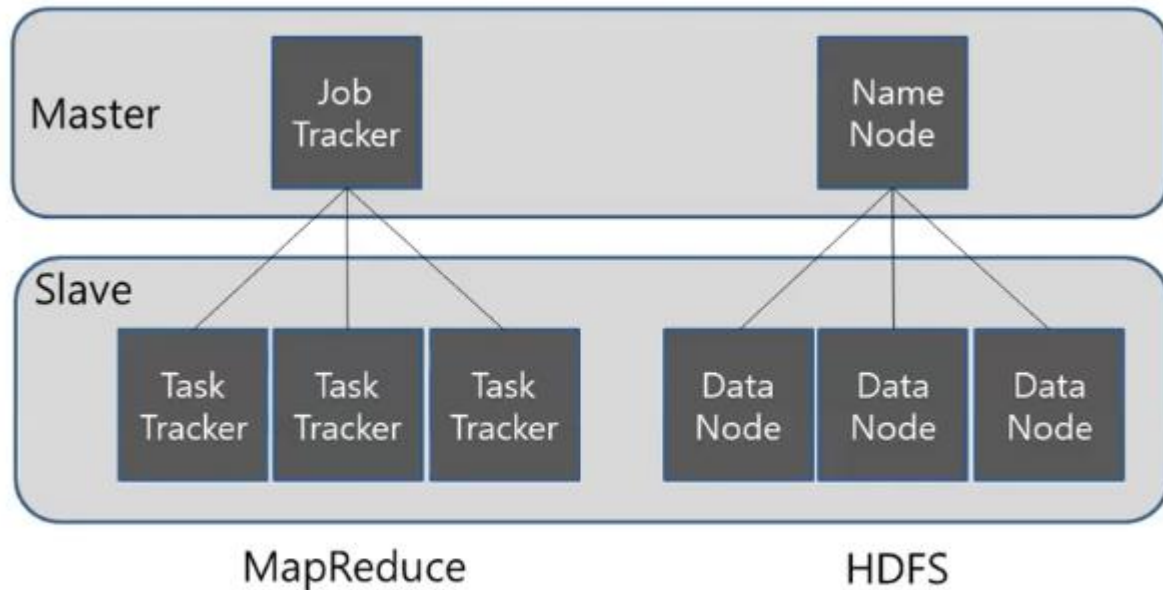
단계	기술 영역	내용
데이터 소스	내부 데이터	데이터베이스, 파일 관리 시스템
	외부 데이터	파일, 멀티미디어, 스트리밍
수집	크롤링 <small>crawling</small>	검색 엔진 로봇을 이용한 데이터 수집
	ETL: 추출 <small>Extraction</small> , 변환 <small>Transformation</small> , 적재 <small>Loading</small>	소스 데이터의 추출, 전송, 변환, 적재
저장	데이터 관리: NoSQL	비정형 데이터 관리
	저장소	빅데이터 저장
	서버	초경량 서버
처리	맵리듀스 <small>mapReduce</small>	데이터 추출
	작업 처리	다중 작업 처리
분석	신경 언어 프로그래밍 <small>NLP, Neuro Linguistic Programming</small>	자연어 처리
	머신러닝	데이터 패턴 발견
	직렬화 <small>serialization</small>	데이터 간 순서화
표현	시각화 <small>visualization</small>	데이터를 도표나 그래픽으로 표현
	획득 <small>acquisition</small>	데이터의 획득 및 재해석

2. Hadoop

⌚ 하둡(Hadoop) 개념

※ Hadoop(High-Availability Distributed Object-Oriented Platform) : Java 로 개발되었으며, 클러스터에서 사용할 수 있는 분산파일시스템과 분산처리시스템을 제공하는 아파치 소프트웨어 재단의 오픈 소스 프레임워크

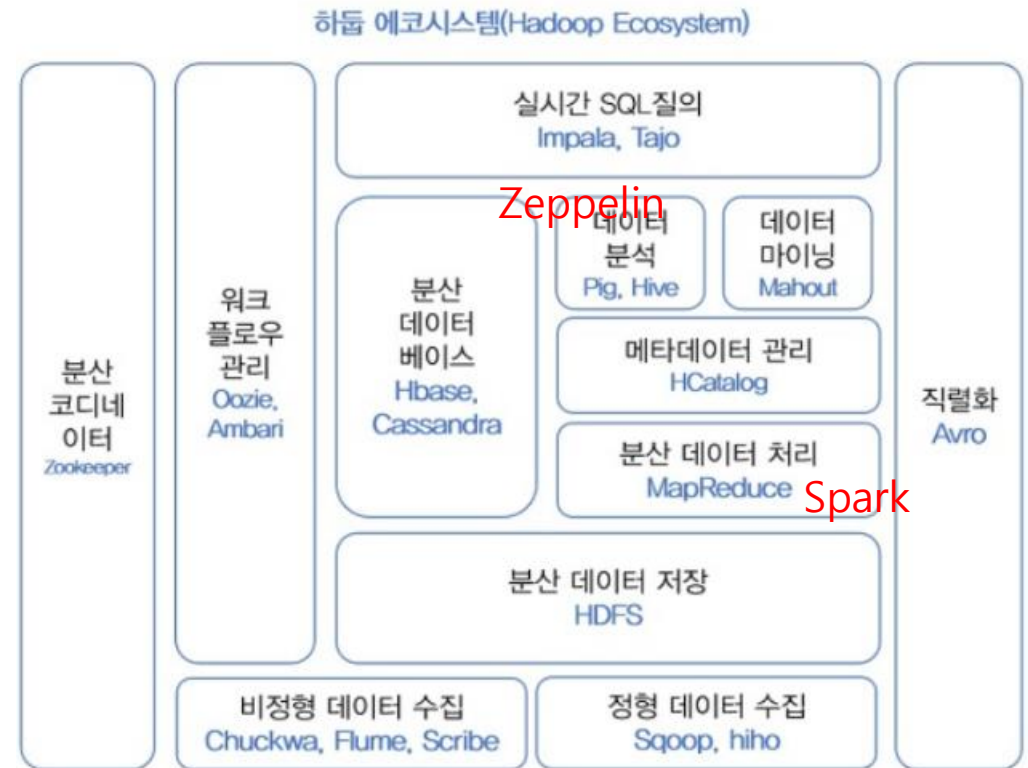
기본골격



한기철, K-ICT 빅데이터 교육교재 중 발췌

- 맵리듀스는 일을 어떻게 분배할 것인지 결정
- HD파일시스템은 데이터를 어떻게 분산저장할지를 결정

기능확장



하둡 프로그래밍(위키북스)

2-1. 환경설정

⌚ 가상머신 (wsl : windows subsystem for linux)

Powershell 관리자권한으로 실행
디렉토리생성 : _seok

wsl -l -o

```
PS C:\>
PS C:\> cd \
PS C:\> mkdir _seok_

디렉터리 : C:\

Mode                LastWriteTime
----                -
d-----          2023-09-08    오전 4:42

PS C:\> cd _seok_
PS C:\_seok>
PS C:\_seok>
```

```
PS C:\_seok> wsl -l -o
다음은 설치할 수 있는 유효한 배포 목록입니다.
기본 배포는 '*' 로 표시됩니다.
'wsl --install -d <Distro>'을(를) 사용하여 설치하세요.

NAME                                FRIENDLY NAME
* Ubuntu                            Ubuntu
Debian                              Debian GNU/Linux
kali-linux                          Kali Linux Rolling
Ubuntu-18.04                        Ubuntu 18.04 LTS
Ubuntu-20.04                        Ubuntu 20.04 LTS
Ubuntu-22.04                        Ubuntu 22.04 LTS
OracleLinux_7_9                     Oracle Linux 7.9
OracleLinux_8_7                     Oracle Linux 8.7
OracleLinux_9_1                     Oracle Linux 9.1
openSUSE-Leap-15.5                  openSUSE Leap 15.5
SUSE-Linux-Enterprise-Server-15-SP4 SUSE Linux Enterprise Server 15 SP4
SUSE-Linux-Enterprise-15-SP5        SUSE Linux Enterprise 15 SP5
openSUSE-Tumbleweed                 openSUSE Tumbleweed
PS C:\_seok>
```


2-1. 환경설정

 wsl

wsl --install -d Ubuntu-22.04

```
PS C:\_seok>
PS C:\_seok> wsl --install -d Ubuntu-22.04
```

```
설치 중: 가상 머신 플랫폼
가상 머신 플랫폼이(가) 설치되었습니다.
설치 중: Linux용 Windows 하위 시스템
Linux용 Windows 하위 시스템이(가) 설치되었습니다.
설치 중: Linux용 Windows 하위 시스템
Linux용 Windows 하위 시스템이(가) 설치되었습니다.
설치 중: Ubuntu 22.04 LTS
Ubuntu 22.04 LTS이(가) 설치되었습니다.
요청한 작업이 잘 실행되었습니다. 시스템을 다시 시작하면 변경 사항이 적용됩니다.
PS C:\_seok>
```

2-1. 환경설정

wsl

dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

wsl --set-default-version 2

```
PS C:\_seok>
PS C:\_seok> dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /nores

배포 이미지 서비스 및 관리 도구
버전: 10.0.19041.844

이미지 버전: 10.0.19045.3324

기능을 사용하도록 설정하는 중
[=====100.0%=====]
작업을 완료했습니다.
PS C:\_seok>
```

2-1. 환경설정

⌚ wsl

```
PS C:\_seok>
PS C:\_seok> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

배포 이미지 서비스 및 관리 도구
버전: 10.0.19041.844

이미지 버전: 10.0.19045.3324

기능을 사용하도록 설정하는 중
[=====100.0%=====]
작업을 완료했습니다.
PS C:\_seok> _
```

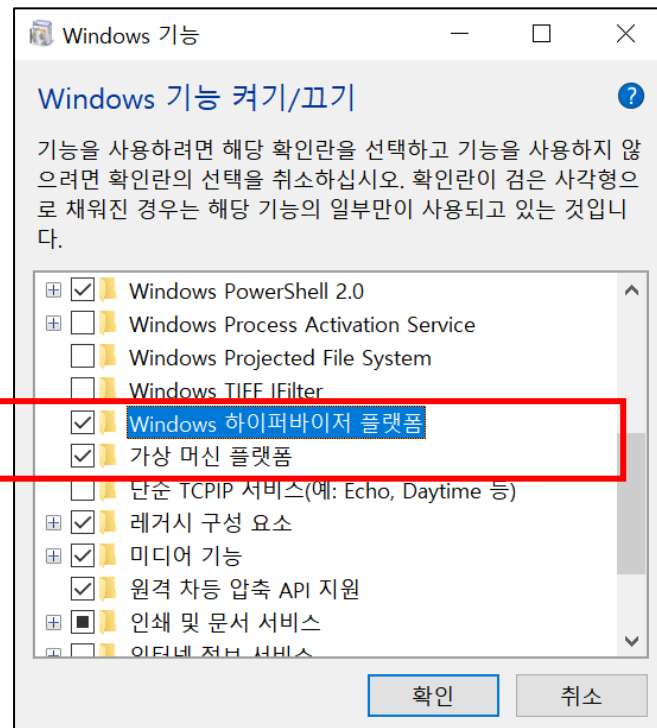
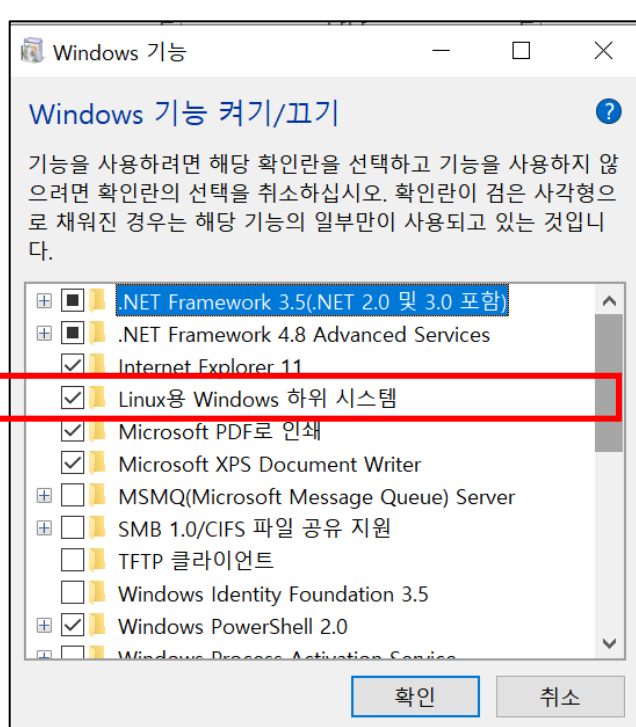
```
PS C:\_seok>
PS C:\_seok> wsl --set-default-version 2
WSL 2와의 주요 차이점에 대한 자세한 내용은 https://aka.ms/wsl2를 참조하세요

작업을 완료했습니다.
PS C:\_seok>
PS C:\_seok>
PS C:\_seok>
```

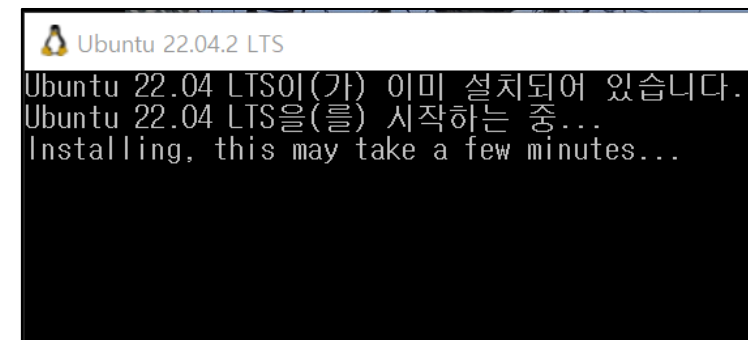

2-1. 환경설정

⌚ wsl

제어판-프로그램및기능-Windows 기능 켜기/끄기



PC 리부트



왼쪽화면이 수정이 되면, PC를 리스타트 한다는 화면이 뜬다

세 곳이 체크되어야 하며, 만약 모두 되어 있다면, 일부러 체크를 한번 끄고, <확인>한 다음, 다시 체크를 켜고 <확인>하면, PC를 리스타트 한다는 화면이 나올 것임

2-1. 환경설정



WSL

Username : seokill

Password : 10041004

```
seokill@DESKTOP-66SF44L: ~
Ubuntu 22.04 LTS이(가) 이미 설치되어 있습니다.
Ubuntu 22.04 LTS을(를) 시작하는 중...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: seokill
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

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

This message is shown once a day. To disable it please create the
/home/seokill/.hushlogin file.
seokill@DESKTOP-66SF44L:~$
```

PC가 자동으로 리스타트 되면서, 위와 같이 Username 과 Password 를 새롭게 등록하도록 하는 화면이 뜸 (비밀번호는 눈에 보이지 않음으로 천천히 10041004를 입력해야 함)

만약, PC가 리부팅이 되어도 왼쪽과 같이 자동으로 Username을 묻는 화면이 뜨지 않으면,

Powershell 관리자권한으로 실행하고, 아래 명령을 차례대로 수행

wsl --set-default Ubuntu-22.04

wsl --distribution Ubuntu-22.04

```
관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Windows\system32> wsl -l -v
NAME                STATE              VERSION
* Ubuntu-22.04      Stopped            2
Ubuntu-20.04        Stopped            2
Ubuntu-20.04        Stopped            2
PS C:\Windows\system32> wsl --set-default Ubuntu-22.04
작업을 완료했습니다.
PS C:\Windows\system32> wsl --distribution Ubuntu-22.04
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

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

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/joy/.hushlogin file.
joy@mnt/c/Windows/system32 $
```

위와 같이 화면이 뜨지 않고, 화면 멈춤 상태면, 72 페이지 참조 후, 11페이지부터 다시 시작

2-1. 환경설정

⌚ wsl

sudo passwd root
10041004

```
seokill@DESKTOP-66SF44L:~$ sudo passwd root
[sudo] password for seokill:
New password:
Retype new password:
passwd: password updated successfully
seokill@DESKTOP-66SF44L:~$
seokill@DESKTOP-66SF44L:~$
```

su -

```
seokill@DESKTOP-66SF44L:~$
seokill@DESKTOP-66SF44L:~$ su -
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advan

This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@DESKTOP-66SF44L:~#
root@DESKTOP-66SF44L:~#
```

또는

앞 화면에서 Username 을 생성하지 않고, 우분투로
들어오면, 아래와 같이 root 로 직접 들어 오게 됨

모든 작업은 root 로 하기 때문에 Username 은 생성하지
않아도 됨.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

* Strictly confined Kubernetes makes edge and IoT secure. Learn how M
just raised the bar for easy, resilient and secure K8s cluster depl

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/root/.hushlogin file
[root@linux system32]#
```

root 로 직접 들어 온 상태라면,

root 의 비밀번호를 바꾸어 줌

passwd root
10041004

2-1. 환경설정

⌚ wsl

cd ~

vi .bashrc

vi 가 어려우면 77페이지참조

```
root@DESKTOP-66SF44L:~#
root@DESKTOP-66SF44L:~#
root@DESKTOP-66SF44L:~# vi .bashrc
```

PS1='[\u@linux \W]# '

```
PS1='[\u@linux \W]# '
```

source .bashrc

```
root@DESKTOP-66SF44L:~#
root@DESKTOP-66SF44L:~# source .bashrc
[root@linux ~]#
[root@linux ~]#
```

mkdir /util

cd /util

exit

```
[root@linux util]#
[root@linux util]#
[root@linux util]# exit
```

exit

```
[seokill@linux ~]#
[seokill@linux ~]# exit
```

cd _seok

관리자: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/powershell

PS C:\WINDOWS\system32> cd \_seok
PS C:\_seok>
PS C:\_seok>
```

2-1. 환경설정

wsl

wsl -l -v

wsl --export Ubuntu-22.04 c:_seok\u-22.04.tar

dir

wsl --unregister Ubuntu-22.04

wsl -l -v

```
PS C:\_seok>
PS C:\_seok> wsl -l -v
    NAME                STATE              VERSION
* Ubuntu-22.04         Stopped            2
PS C:\_seok> wsl --export Ubuntu-22.04 c:\_seok\u-22.04.tar
내보내기가 진행 중입니다. 이 작업은 몇 분 정도 걸릴 수 있습니다.
작업을 완료했습니다.
PS C:\_seok> dir

디렉터리 : C:\_seok

Mode                LastWriteTime         Length Name
----                -
-a-----          2023-09-08 오전 6:11      1099366400 u-22.04.tar

PS C:\_seok> wsl --unregister Ubuntu-22.04
등록 취소 중입니다.
작업을 완료했습니다.
PS C:\_seok>
PS C:\_seok> wsl -l -v
Linux용 Windows 하위 시스템에 설치된 배포판이 없습니다.

'wsl.exe --list --online'를 사용하여 사용 가능한 배포판을 나열하고
'wsl.exe --install <Distro>'를 사용하여 설치하세요.

배포판은 Microsoft Store
(https://aka.ms/wslstore)를
방문하여 설치할 수도 있습니다.
Error code: Wsl/WSL_E_DEFAULT_DISTRO_NOT_FOUND
PS C:\_seok>
```

2-1. 환경설정



wsl

wsl -update

wsl --import Ubuntu-22.04 c:_seok\Ubuntu-22.04\ c:_seok\u-22.04.tar

메모장 열고 아래 두줄 저장

wsl --set-default Ubuntu-22.04

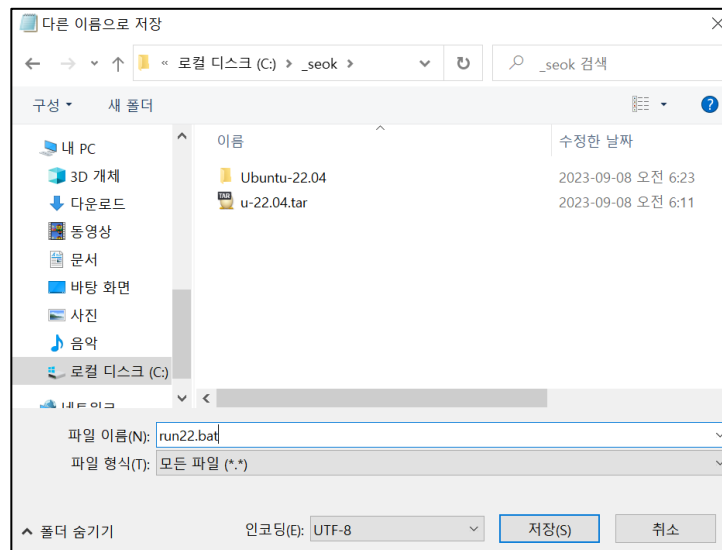
wsl --distribution Ubuntu-22.04

run22.bat - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

wsl --set-default Ubuntu-22.04

wsl --distribution Ubuntu-22.04



c:_seok\ 밑에 run22.bat 으로 저장

여기까지, 정상적으로 완료가 된 경우에는,

앞으로 PC를 켜고, wsl 을 실행하고 싶을 때는, 아래의 방법으로 접근함

- 1) 방법-1 (최적의 방법)
관리자모드로 파워셸을 열고,
cd _seok
.\run22
- 2) 방법-2 (단순 방법)
관리자모드로 파워셸을 열고,
cd _seok
wsl
- 3) 방법-3 (이것은 43페이지 학습 후)

2-1. 환경설정

jvm

cd /util

Java -version

```
[root@linux util]#
[root@linux util]# java -version
Command 'java' not found, but can be installed with:
apt install openjdk-11-jre-headless # version 11.0.20.1+1-0ubuntu1~22.04, or
apt install default-jre # version 2:1.11-72build2
apt install openjdk-17-jre-headless # version 17.0.8.1+1~us1-0ubuntu1~22.04
apt install openjdk-18-jre-headless # version 18.0.2+9-2~22.04
apt install openjdk-19-jre-headless # version 19.0.2+7-0ubuntu3~22.04
apt install openjdk-8-jre-headless # version 8u382-ga-1~22.04.1
[root@linux util]#
```

```
[root@linux Downloads]# java -version
openjdk version "11.0.19" 2023-04-18
OpenJDK Runtime Environment (build 11.0.19+7-post-Ubuntu-0ubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 11.0.19+7-post-Ubuntu-0ubuntu122.04.1, mixed mode, sharing)
[root@linux Downloads]#
```

2-1. 환경설정

⌚ jvm

<https://www.oracle.com/kr/java/technologies/downloads/#jdk19-linux>

※ 어제 수업 기준으로,
jvm이 jdk-21 로 업그레이드 됨

이 페이지 이후로는
jdk-21 로 이름을 바꾸어서
환경설정을 해야 함

Java downloads Tools and resources Java archive

JDK 20 JDK 17 GraalVM for JDK 20 GraalVM for JDK 17

JDK Development Kit 20.0.2 downloads

JDK 20 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#).

JDK 20 will receive updates under these terms, until September 2023 when it will be superseded by JDK 21.

Linux macOS Windows

Product/file description	File size	Download
ARM64 Compressed Archive	181.55 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-aarch64_bin.tar.gz
ARM64 RPM Package	181.27 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-aarch64_bin.rpm
x64 Compressed Archive	183.11 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.tar.gz
x64 Debian Package	155.91 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.deb

오른쪽 마우스 버튼
링크 주소 복사

2-1. 환경설정

⌚ jvm

cd /util

wget https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.tar.gz

tar -zxvf jdk-20_linux-x64_bin.tar.gz

```
[root@linux ~]#
[root@linux ~]# cd /util
[root@linux util]#
[root@linux util]# wget https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.tar.gz
--2023-09-08 13:07:18-- https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.tar.gz
Resolving download.oracle.com (download.oracle.com)... 23.78.216.31
Connecting to download.oracle.com (download.oracle.com)|23.78.216.31|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 192003505 (183M) [application/x-gzip]
Saving to: 'jdk-20_linux-x64_bin.tar.gz'

jdk-20_linux-x64_bin.tar.gz  100%[=====>] 183.11M  10.1MB/s

2023-09-08 13:07:38 (9.32 MB/s) - 'jdk-20_linux-x64_bin.tar.gz' saved [192003505/192003505]

[root@linux util]# ls
jdk-20_linux-x64_bin.tar.gz
[root@linux util]# tar -zxvf jdk-20_linux-x64_bin.tar.gz
[root@linux util]#
[root@linux util]#
```

만약 tar 가 설치되어 있지
않다면,

➔ yum install tar

만약 yum 이 설치되어 있지
않다면,

➔ sudo apt install yum4

➔ yum4 install tar

2-1. 환경설정

⌚ jvm

```
ls
mv jdk-20.0.2 /usr/jdk-20.0.2
ls -al /usr
```

```
[root@linux util]# ls
jdk-20_linux-x64_bin.tar.gz
[root@linux util]# tar -zxf jdk-20_linux-x64_bin.tar.gz
[root@linux util]#
[root@linux util]# ls
jdk-20.0.2 jdk-20_linux-x64_bin.tar.gz
[root@linux util]# mv jdk-20.0.2 /usr/jdk-20.0.2
[root@linux util]# ls -al /usr
total 84
drwxr-xr-x 15 root root 4096 Sep  8 13:16 .
drwxr-xr-x 20 root root 4096 Sep  8 09:56 ..
drwxr-xr-x  2 root root 20480 Sep  8 05:26 bin
drwxr-xr-x  2 root root 4096 Apr 18  2022 games
drwxr-xr-x  3 root root 4096 May  2 06:35 include
drwxr-xr-x  9 root root 4096 Sep  8 13:10 jdk-20.0.2
drwxr-xr-x 59 root root 4096 Sep  8 05:26 lib
drwxr-xr-x  2 root root 4096 May  2 06:34 lib32
drwxr-xr-x  2 root root 4096 May  2 06:35 lib64
drwxr-xr-x  7 root root 4096 May  2 06:35 libexec
drwxr-xr-x  2 root root 4096 May  2 06:34 libx32
drwxr-xr-x 10 root root 4096 May  2 06:34 local
drwxr-xr-x  2 root root 12288 Sep  8 05:26 sbin
drwxr-xr-x 90 root root 4096 May  2 06:35 share
drwxr-xr-x  2 root root 4096 Apr 18  2022 src
[root@linux util]#
```

2-1. 환경설정

⌚ jvm

```
[root@linux util]#  
[root@linux util]# cd ~  
[root@linux ~]#  
[root@linux ~]# vi .bashrc
```

cd ~

vi .bashrc

또는

cd ~

gnome-text-editor .bashrc

```
# enable programmable completion features  
# this, if it's already enabled in /etc/b  
# sources /etc/bash.bashrc).  
#if [ -f /etc/bash_completion ] && ! shopt  
#    . /etc/bash_completion  
#fi  
  
PS1='[\u@linux \w]# '  
  
export JAVA_HOME=/usr/jdk-20.0.2  
export PATH=$JAVA_HOME/bin:$PATH
```

<vi 명령어 사용>

j 로 아래로 이동 → fi 문자 까지

o 누르고, EnterKey 입력 → 두줄 아래부터 키 입력 가능

#-----JVM -----

export JAVA_HOME=/usr/jdk-20.0.2

export PATH=\$JAVA_HOME/bin:\$PATH

ESC 콜론(:) wq EnterKey 입력

2-1. 환경설정

⌚ jvm

```
[root@linux ~]#  
[root@linux ~]# source .bashrc  
[root@linux ~]#  
[root@linux ~]# echo $JAVA_HOME  
/usr/jdk-20.0.2  
[root@linux ~]#  
[root@linux ~]# java -version  
java version "20.0.2" 2023-07-18  
Java(TM) SE Runtime Environment (build 20.0.2+9-78)  
Java HotSpot(TM) 64-Bit Server VM (build 20.0.2+9-78, mixed mode, sharing)  
[root@linux ~]#  
[root@linux ~]#
```

source .bashrc

echo \$JAVA_HOME

java -version

2-1. 환경설정

⌚ python3

```
[root@linux ~]# python -V
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
[root@linux ~]# python3 -V
Python 3.10.12
[root@linux ~]# apt-get install -y python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2-0)
0 upgraded, 0 newly installed, 0 to remove and 52 not installed.
```

```
python -V
python3 -V
```

```
apt-get install -y python3-pip
```

만약, python 이 없다면,

```
#-- python 3.10
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
##### python 3.10 확인
apt list | grep python3.10
##### 설치
sudo apt-get install python3.10
##### alternatives 에 등록
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.8 1
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.10 2
##### 기본 호출을 3.10 으로 변경
sudo update-alternatives --config python3

##### 오류 발생 시
sudo apt-get remove python3-apt
sudo apt-get install python3-apt
sudo apt-get install --reinstall python3-apt
```

2-1. 환경설정

⌚ python3

```
[root@linux ~]#
[root@linux ~]# pip3 install pyspark findspark
Collecting pyspark
  Downloading pyspark-3.4.1.tar.gz (310.8 MB)
    310.8/310.8 MB 798.6 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Collecting py4j==0.10.9.7
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)
    200.5/200.5 KB 5.3 MB/s eta 0:00:00
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.4.1-py2.py3-none-any.whl size=311285409 sha256=3735226b05d79812eceda475b72058ef56409f6a2d7482
  Stored in directory: /root/.cache/pip/wheels/0d/77/a3/ff2f74cc9ab41f8f594dabf0579c2a7c6de920d584206e
Successfully built pyspark
Installing collected packages: py4j, findspark, pyspark
Successfully installed findspark-2.0.1 py4j-0.10.9.7 pyspark-3.4.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with
the package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[root@linux ~]#
```

pip3 install pyspark findspark

2-1. 환경설정

⌚ python3

```
[root@linux util]#  
[root@linux util]# cd ~  
[root@linux ~]#  
[root@linux ~]# which python3  
/usr/bin/python3  
[root@linux ~]#  
[root@linux ~]# vi .bashrc
```

```
PS1='[#u@linux #W] # '  
export JAVA_HOME=/usr/jdk-20.0.2  
export PATH=$JAVA_HOME/bin:$PATH  
#----- PYTHON -----  
export PYTHONPATH=/usr/bin/python3  
export PYSPARK_PYTHON=/usr/bin/python3  
export PATH=$PATH:/usr/bin/python3  
".bashrc" 110L, 3353B
```

cd ~

which python3

vi .bashrc



source .bashrc

<vi 명령어 사용>

j 로 아래로 이동 → export 문자 까지
o 누르고, EnterKey 입력 → 두줄 아래부터 키 입력 가능

#----- PYTHON -----

```
export PYTHONPATH=/usr/bin/python3  
export PYSPARK_PYTHON=/usr/bin/python3  
export PATH=$PATH:/usr/bin/python3
```

입력

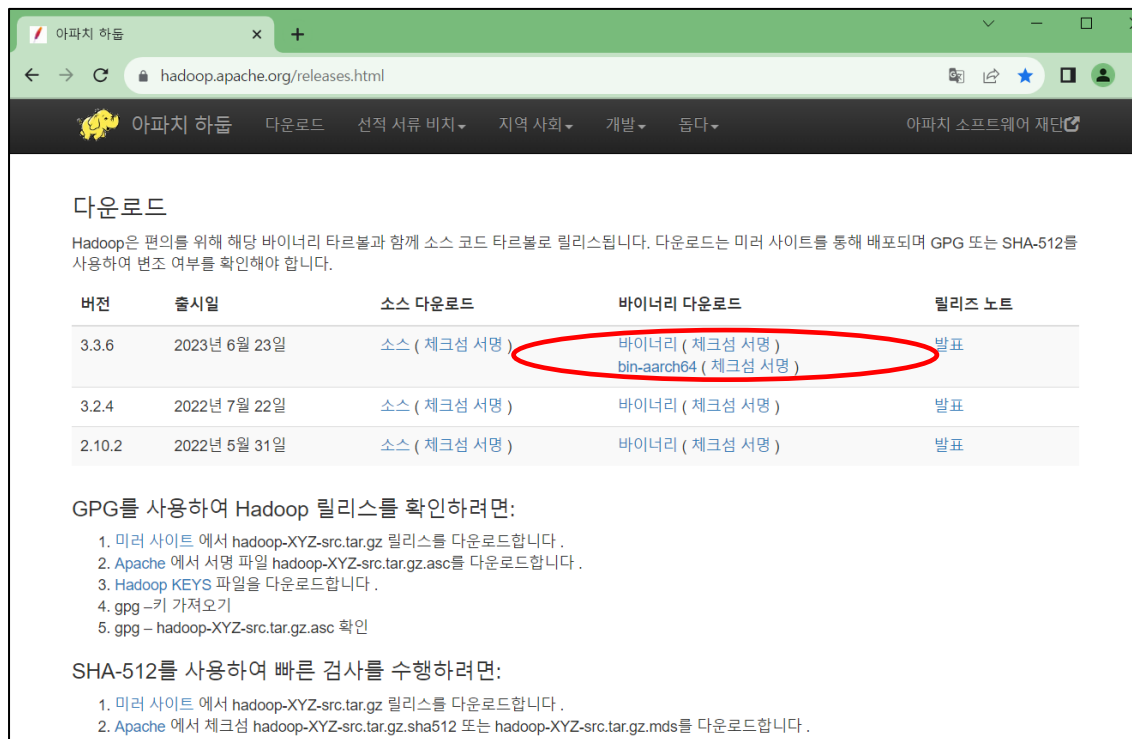
ESC 콜론(:) wq EnterKey 입력

2-2. Hadoop설치

1

다운로드/설치

- <https://hadoop.apache.org/releases.html>



다운로드

Hadoop은 편의를 위해 해당 바이너리 타르볼과 함께 소스 코드 타르볼로 릴리스됩니다. 다운로드를 미리 사이트를 통해 배포되며 GPG 또는 SHA-512를 사용하여 변조 여부를 확인해야 합니다.

버전	출시일	소스 다운로드	바이너리 다운로드	릴리스 노트
3.3.6	2023년 6월 23일	소스 (체크섬 서명)	바이너리 (체크섬 서명) bin-aarch64 (체크섬 서명)	발표
3.2.4	2022년 7월 22일	소스 (체크섬 서명)	바이너리 (체크섬 서명)	발표
2.10.2	2022년 5월 31일	소스 (체크섬 서명)	바이너리 (체크섬 서명)	발표

GPG를 사용하여 Hadoop 릴리스를 확인하려면:

1. [미러 사이트](#)에서 `hadoop-XYZ-src.tar.gz` 릴리스를 다운로드합니다.
2. [Apache](#)에서 서명 파일 `hadoop-XYZ-src.tar.gz.asc`를 다운로드합니다.
3. [Hadoop KEYS](#) 파일을 다운로드합니다.
4. `gpg -키` 가져오기
5. `gpg -hadoop-XYZ-src.tar.gz.asc` 확인

SHA-512를 사용하여 빠른 검사를 수행하려면:

1. [미러 사이트](#)에서 `hadoop-XYZ-src.tar.gz` 릴리스를 다운로드합니다.
2. [Apache](#)에서 체크섬 `hadoop-XYZ-src.tar.gz.sha512` 또는 `hadoop-XYZ-src.tar.gz.mds`를 다운로드합니다.



커뮤니티 주도 개발 "THE APACHE WAY"

다음 사이트에서 다운로드하는 것이 좋습니다.

<https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz>

대체 다운로드 위치는 아래에 제안되어 있습니다.

PGP 서명(파일) 또는 해시(또는 파일)를 사용하여 다운로드.

이 필수적입니다. .as

새 탭에서 링크 열기
새 창에서 링크 열기
시크릿 창에서 링크 열기
다른 이름으로 링크 저장...
링크 주소 복사
검사

HTTP

<https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz>

- 최신버전의 <바이너리(체크섬 서명)> 클릭

- 링크에서 오른쪽 마우스 <링크 주소 복사> 클릭

2-2. Hadoop설치

```
[root@linux ~]#  
[root@linux ~]# cd /util  
[root@linux util]#  
[root@linux util]# wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz  
--2023-09-10 07:26:59-- https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz  
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644  
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 730107476 (696M) [application/x-gzip]  
Saving to: 'hadoop-3.3.6.tar.gz'
```

cd /util

wget <https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz>

```
drwxr-xr-x  2 root root    4096 Sep 10 07:26 .  
drwxr-xr-x 21 root root    4096 Sep 10 05:22 ..  
-rw-r--r--  1 root root 730107476 Jun 26 08:35 hadoop-3.3.6.tar.gz  
-rw-r--r--  1 root root 192003505 Jun 16 03:35 jdk-20_linux-x64_bin.tar.gz  
[root@linux util]#  
[root@linux util]# tar -zxvf hadoop-3.3.6.tar.gz
```

tar -zxvf hadoop-3.3.6.tar.gz

2-2. Hadoop설치

2

환경 설정

⌚ 환경설정파일

파일 구분	내 용	비 고
hdfs-site.xml	<ul style="list-style-type: none"> 하둡 파일시스템 환경설정 	
core-site.xml	<ul style="list-style-type: none"> HDFS, MapReduce 환경설정 	
yarn-site.xml	<ul style="list-style-type: none"> Resource Manager 및 Node Manager 환경설정 	
mapred-site.xml	<ul style="list-style-type: none"> MapReduce 어플리케이션 환경설정 	
hadoop-env.sh	<ul style="list-style-type: none"> 하둡이 구동되는 데 필요한 환경 설정 	
workers	<ul style="list-style-type: none"> 하둡의 worker 로 동작할 서버 호스트 이름 설정 	<ul style="list-style-type: none"> slaves
masters	<ul style="list-style-type: none"> 하둡의 master 로 동작할 서버 호스트 이름 설정 	

2-2. Hadoop설치

```
[root@linux util]#
[root@linux util]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.18.55.150 netmask 255.255.240.0 broadcast 172.18.63.255
    inet6 fe80::215:5dff:fe12:9d47 prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:12:9d:47 txqueuelen 1000 (Ethernet)
```

ifconfig

```
[root@linux util]#
[root@linux util]#
[root@linux util]# vi /etc/hosts
```

vi /etc/hosts

본인의 우분투 IP주소로
변경해야 함

```
root@DESKTOP-28CEK7O: ~
# This file was automatically generated by WSL
# [network]
# generateHosts = false
127.0.0.1 localhost
127.0.1.1 DESKTOP-28CEK7O. DESKTO
172.18.55.150 linux.home
172.18.55.150 nn1
172.18.55.150 nn2
172.18.55.150 dn1
172.18.55.150 dn2
172.18.55.150 dn3
```

<vi 명령어 사용>

j 로 맨 아래로 이동
o 누르면, 한줄 아래부터 키 입력 가능

```
172.18.55.150 linux.home
172.18.55.150 nn1
172.18.55.150 nn2
172.18.55.150 dn1
172.18.55.150 dn2
172.18.55.150 dn3
```

ESC 콜론(:) wq EnterKey 입력

2-2. Hadoop설치

```
[root@linux util]#  
[root@linux util]#  
[root@linux util]# mv hadoop-3.3.6 hadoop  
[root@linux util]#  
[root@linux util]#  
[root@linux util]# cd ~  
[root@linux ~]#  
[root@linux ~]# vi .bashrc
```

mv hadoop-3.3.4 hadoop

cd ~

vi .bashrc



source .bashrc

```
#----- PYTHON -----  
export PYTHONPATH=/usr/bin/python3  
export PYSPARK_PYTHON=/usr/bin/python3  
  
#----- HADOOP -----  
export HADOOP_HOME=/util/hadoop  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export HADOOP_YARN_HOME=$HADOOP_HOME  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

<vi 명령어 사용>

j 로 맨 아래로 이동
o 누르면, 한줄 아래부터 키 입력 가능

```
#----- HADOOP -----  
export HADOOP_HOME=/util/hadoop  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export HADOOP_YARN_HOME=$HADOOP_HOME  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

ESC 콜론(:) wq EnterKey 입력

2-2. Hadoop설치

```
[root@linux ~]#  
[root@linux ~]# env | grep HADOOP  
HADOOP_HOME=/util/hadoop  
HADOOP_HDFS_HOME=/util/hadoop  
HADOOP_COMMON_HOME=/util/hadoop  
HADOOP_YARN_HOME=/util/hadoop  
HADOOP_CONF_DIR=/util/hadoop/etc/hadoop  
HADOOP_MAPRED_HOME=/util/hadoop  
[root@linux ~]#  
[root@linux ~]#
```

env | grep HADOOP

```
[root@linux ~]#  
[root@linux ~]# cd /util/hadoop/etc/hadoop  
[root@linux hadoop]#  
[root@linux hadoop]# ls  
capacity-scheduler.xml      hadoop-user-functions.sh.example  
configuration.xml           hdfs-rbf-site.xml  
container-executor.cfg      hdfs-site.xml  
core-site.xml               httpfs-env.sh  
hadoop-env.cmd              httpfs-log4j.properties  
hadoop-env.sh               httpfs-site.xml  
hadoop-metrics2.properties  kms-acls.xml  
hadoop-policy.xml           kms-env.sh
```

cd /util/hadoop/etc/hadoop

ls

2-2. Hadoop설치

- 하둡 파일시스템 환경설정

```
[root@linux hadoop]#  
[root@linux hadoop]#  
[root@linux hadoop]# vi hdfs-site.xml
```

<configuration>

<!-- configuration hadoop -->

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.http.address</name>

<value>nn1:50070</value>

</property>

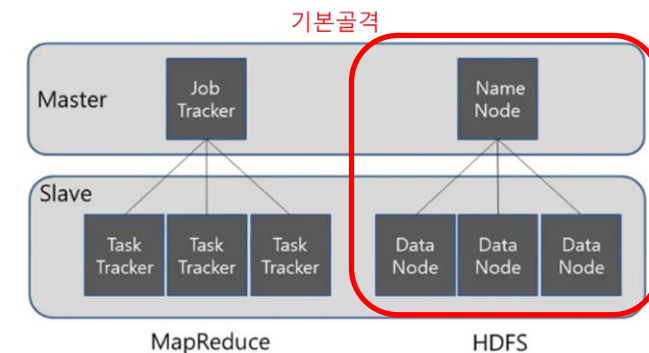
<property>

<name>dfs.secondary.http.address</name>

<value>nn2:50090</value>

</property>

</configuration>



한기철, K-ICT 빅데이터 교육교재 중 발췌

vi hdfs-site.xml

2-2. Hadoop설치

- HDFS, MapReduce 환경설정

```
[root@linux hadoop]#
```

```
[root@linux hadoop]#
```

```
[root@linux hadoop]# vi core-site.xml
```

vi core-site.xml

```
<configuration>
```

```
  <property>
```

```
    <name>fs.default.name</name>
```

```
    <value>hdfs://nn1:9000</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>hadoop.tmp.dir</name>
```

```
    <value>/util/hadoop/logs/</value>
```

```
  </property>
```

```
</configuration>
```

2-2. Hadoop설치

- Resource Manager 및 Node Manager 환경설정

```
[root@linux hadoop]#
```

```
[root@linux hadoop]#
```

```
[root@linux hadoop]# vi yarn-site.xml
```

vi 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.aux-services.mapreduce_shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>nn1</value>

</property>

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>>false</value>

</property>

</configuration>

2-2. Hadoop설치

- MapReduce 어플리케이션 환경설정

```
[root@linux hadoop]#
```

```
[root@linux hadoop]#
```

```
[root@linux hadoop]# vi mapred-site.xml
```

vi mapred-site.xml

```
<configuration>
```

```
<property>
```

```
<name>mapred.job.tracker</name>
```

```
<value>nn1:9001</value>
```

```
</property>
```

```
</configuration>
```

2-2. Hadoop설치

- 하둡이 구동되는 데 필요한 환경 설정

```
[root@linux hadoop]#  
[root@linux hadoop]#  
[root@linux hadoop]# vi hadoop-env.sh
```

vi hadoop-env.sh

```
# Supplemental options for privileged registry DNS  
# By default, Hadoop uses jsvc which needs to know to launch  
# server jvm.  
# export HADOOP_REGISTRYDNS_SECURE_EXTRA_OPTS="-jvm server  
export JAVA_HOME=/usr/jdk-20.0.2  
export HADOOP_HOME=/util/hadoop
```

```
export JAVA_HOME=/usr/jdk-20.0.2  
export HADOOP_HOME=/util/hadoop
```

<vi 명령어 사용>

\$G 로 맨 아래로 이동 → # export 문자 까지
o 누르고, EnterKey 입력 → 두줄 아래부터 키 입력 가능

```
export JAVA_HOME=/usr/jdk-20.0.2  
export HADOOP_HOME=/util/hadoop
```

ESC 콜론(:) wq EnterKey 입력

2-2. Hadoop설치

- 하둡의 worker 로 동작할 서버 호스트 이름 설정

```
[root@linux hadoop]#  
[root@linux hadoop]#  
[root@linux hadoop]# vi workers_
```

vi workers



cp workers slaves

```
> root@linux:/util/hadoop/etc/hadoop
```

```
#localhost  
dn1  
dn2  
dn3  
~  
~  
~
```

기존에 있던, localhost 는 # 으로 막음

```
dn1  
dn2  
dn3
```

<vi 명령어 사용>

dd 로 첫 줄 삭제 또는 i 누르고 # 입력 후, ESC
o 를 누르고 아래 줄 삽입

```
dn1  
dn2  
dn3
```

ESC 콜론(:) wq EnterKey 입력

2-2. Hadoop설치

- 하둡의 master 로 동작할 서버 호스트 이름 설정

```
[root@linux hadoop]#  
[root@linux hadoop]#  
[root@linux hadoop]# vi masters
```

vi masters

```
> root@linux:/util/hadoop/etc/hadoop
```

```
#localhost  
nn1  
nn2
```

<vi 명령어 사용>

i 누르고,

nn1
nn2

ESC 콜론(:) wq EnterKey 입력

파일 신규 생성

nn1
nn2

2-2. Hadoop설치

3

실행

방화벽 중지

```
[root@linux hadoop]# service iptables stop
Failed to stop iptables.service: Unit iptables.service not loaded.
[root@linux hadoop]#
[root@linux hadoop]#
```

service iptables stop

오류 발생 시, 그대로 둠

ssh 로 로그인 ※ cmd 창을 새롭게 띄우고 접속

```
D:\>ssh joy@172.18.55.150
joy@172.18.55.150's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

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

joy@/home/joy $
```

```
joy@/home/joy $ su - root
Password:
[root@linux ~]# cd /util
[root@linux util]# cd hadoop
```

su - root

cd /util/hadoop

ifconfig로 확인

ssh myid@172.18.55.150

ssh 설정은, 73페이지참조

2-2. Hadoop설치

■ 네임노드 초기화

```
[root@linux hadoop]# ./bin/hadoop namenode -format
WARNING: Use of this script to execute namenode is deprecated.
WARNING: Attempting to execute replacement "hdfs namenode" instead.

WARNING: /util/hadoop/logs does not exist. Creating.
2023-09-10 11:15:45,462 INFO namenode.NameNode: STARTUP_MSG:
/*****
```

./bin/hadoop namenode -format

아래의 메시지가 나오면, 정상적으로 완료됨

```
2023-09-10 17:39:14,673 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2023-09-10 17:39:14,673 INFO util.GSet: VM type = 64-bit
2023-09-10 17:39:14,673 INFO util.GSet: 0.029999999329447746% max memory 1.9 GB = 602.1 KB
2023-09-10 17:39:14,673 INFO util.GSet: capacity = 2^16 = 65536 entries
2023-09-10 17:39:14,688 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1379072242-127.0.1.1-1694335154683
2023-09-10 17:39:14,710 INFO common.Storage: Storage directory /tmp/hadoop-root/dfs/name has been successfully formatted.
2023-09-10 17:39:14,728 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-root/dfs/name/current/fsimage.
g no compression
2023-09-10 17:39:14,790 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-root/dfs/name/current/fsimage.ckpt_00
bytes saved in 0 seconds
2023-09-10 17:39:14,819 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-09-10 17:39:14,842 INFO namenode.FSNamesystem: Stopping services started for active state
2023-09-10 17:39:14,843 INFO namenode.FSNamesystem: Stopping services started for standby state
2023-09-10 17:39:14,846 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2023-09-10 17:39:14,846 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-28CEK70/127.0.1.1
*****/
```

3. Spark설치

1

다운로드/설치

- <https://spark.apache.org/downloads.html>

다운로드 | 아파치 스파크

spark.apache.org/downloads.html

Apache Spark™ 다운로드

1. Spark 릴리스를 선택하세요. 3.4.1(2023년 6월 23일) ▾
2. 패키지 유형을 선택하세요: Apache Hadoop 3.3 이상용으로 사전 구축됨 ▾
3. Spark 다운로드: Spark-3.4.1-bin-hadoop3.tgz
4. 다음 절차에 따라 3.4.1 서명, 체크섬 및 프로젝트 릴리스 KEYS를 사용하여 이 릴리스를 설치하십시오.

Spark 3은 일반적으로 Scala 2.12로 사전 구축되었으며 Spark 3.2+는 Scala 2.13으로 사전 구축된 추가 배포판을 제공합니다.

스파크와 연결

Spark 아티팩트는 [Maven Central](#)에서 호스팅됩니다. 다음 좌표를 사용하여 Maven 종속성을 추가할 수 있습니다.

THE APACHE SOFTWARE FOUNDATION ESTABLISHED 1999

커뮤니티 주도 개발 "THE APACHE"

프로젝트 ▾ 사람들 ▾ 지역 사회 ▾ 특 ▾

다운로드하려면 다음 사이트를 권장합니다.

<https://dlcdn.apache.org/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz>

대체 다운로드 위치는 아래에 제안되어 있습니다.

PGP 서명(파일) 또는 해시(또는 파일)를 사용하여 다운로드

HTTP

<https://dlcdn.apache.org/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz>

새 탭에서 링크 열기
새 창에서 링크 열기
시크릿 창에서 링크 열기
다른 이름으로 링크 저장...
링크 주소 복사
검사

3. Spark설치

```
[root@linux ~]#  
[root@linux ~]#  
[root@linux ~]# cd /util  
[root@linux util]#  
[root@linux util]# pwd  
/util  
[root@linux util]#  
[root@linux util]#
```

cd /util

pwd

```
[root@linux util]#  
[root@linux util]# wget https://dlcdn.apache.org/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz  
--2023-09-10 18:16:19-- https://dlcdn.apache.org/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz  
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644  
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 388341449 (370M) [application/x-gzip]  
Saving to: 'spark-3.4.1-bin-hadoop3.tgz'  
  
spark-3.4.1-bin-hadoop3.tgz      100%[=====>]
```

wget https://dlcdn.apache.org/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz

3. Spark설치

```
[root@linux util]#  
[root@linux util]# tar -zxvf spark-3.4.1-bin-hadoop3.tgz  
spark-3.4.1-bin-hadoop3/  
spark-3.4.1-bin-hadoop3/R/  
spark-3.4.1-bin-hadoop3/R/lib/  
spark-3.4.1-bin-hadoop3/R/lib/sparkr.zip
```

tar -zxvf spark-3.4.1-bin-hadoop3.tgz

```
[root@linux util]#  
[root@linux util]# mv spark-3.4.1-bin-hadoop3 spark  
[root@linux util]#  
[root@linux util]# ls -al  
total 1279760  
drwxr-xr-x  4 root root    4096 Sep 10 18:23 .  
drwxr-xr-x 21 root root    4096 Sep 10 05:22 ..  
drwxr-xr-x 11 joy  joy     4096 Sep 10 11:15 hadoop  
-rw-r--r--  1 root root 730107476 Jun 26 08:35 hadoop-3.3.6.tar.gz  
-rw-r--r--  1 root root 192003505 Jun 16 03:35 jdk-20_linux-x64_bin.tar.gz  
drwxr-xr-x 13 joy  joy     4096 Jun 20 08:23 spark  
-rw-r--r--  1 root root 388341449 Jun 20 08:25 spark-3.4.1-bin-hadoop3.tgz  
[root@linux util]#
```

mv spark-3.4.1-bin-hadoop3 spark

3. Spark설치

2

환경설정

- 스팍의 기본 변수 및 웹페이지 설정

```
[root@linux util]#
[root@linux util]# cd spark
[root@linux spark]# cd conf
[root@linux conf]# ls
fairscheduler.xml.template  metrics.properties.template  spark-env.sh.template
log4j2.properties.template  spark-defaults.conf.template  workers.template
[root@linux conf]#
[root@linux conf]# cp spark-env.sh.template spark-env.sh
[root@linux conf]#
[root@linux conf]#
```

cd spark
cd conf

cp spark-env.sh.template spark-env.sh

3. Spark설치

- 스팍의 기본 변수 및 웹페이지 설정

```
[root@localhost conf]#  
[root@localhost conf]# vi spark-env.sh
```

vi spark-env.sh

또는

gnome-text-editor spark-env.sh

```
Open ▾ + spark-env.sh  
/util/spark/conf  
#!/usr/bin/env bash  
export SPARK_WORKER_INSTANCES=3  
export SPARK_HOME=/util/spark  
export SPARK_CONF_DIR=/util/spark/conf  
export JAVA_HOME=/usr/jdk-20.0.2  
export SPARK_MASTER_WEBUI_PORT=8080  
#export HADOOP_HOME=/util/hadoop  
#export HADOOP_CONF_DIR=/util/hadoop/etc/Hadoop  
  
#  
# Licensed to the Apache Software Foundation (ASF) under one  
# contributor license agreements. See the NOTICE file distributed
```

<vi 명령어 사용>

j 로 아래로 한칸 이동
o 누르면, 한줄 아래부터 키 입력 가능

```
export SPARK_WORKER_INSTANCES=3  
export SPARK_HOME=/util/spark  
export SPARK_CONF_DIR=/util/spark/conf  
export JAVA_HOME=/usr/jdk-20.0.2  
export SPARK_MASTER_WEBUI_PORT=8080  
#export HADOOP_HOME=/util/hadoop  
#export HADOOP_CONF_DIR=/util/hadoop/etc/Hadoop
```

ESC 콜론(:) wq EnterKey 입력

3. Spark설치

- 스팍의 마스터 웹페이지와 로그 설정

```
[root@localhost conf]# ls
fairscheduler.xml.template  metrics.properties.template  spark-env.sh  workers.template
log4j2.properties.template  spark-defaults.conf.template  spark-env.sh.template

[root@localhost conf]# cp spark-defaults.conf.template spark-defaults.conf

[root@localhost conf]# vi spark-defaults.conf
```

cp spark-defaults.conf.template spark-defaults.conf

vi spark-defaults.conf

```
# Default system properties included when running spark-submit.
# This is useful for setting default environmental settings.

# Example:
spark.master                spark://master:7077
spark.eventLog.enabled      true
spark.eventLog.dir          /util/spark/logs
# spark.eventLog.dir        hdfs://namenode:8021/directory
# spark.serializer
```

<vi 명령어 사용>

j 로 아래로 이동 ➔ # spark 문자 까지
x 를 두번 눌러서 # 을 삭제
마지막줄 추가

```
spark.master                spark://master:7077
spark.eventLog.enabled      true
spark.eventLog.dir          /util/spark/logs
```

ESC 콜론(:) wq EnterKey 입력

3. Spark설치

■ 워크 환경 설정

```
[root@linux conf]#  
[root@linux conf]# cp workers.template workers  
[root@linux conf]#
```

```
[root@linux conf]#  
[root@linux conf]# cd ~  
[root@linux ~]#  
[root@linux ~]# vi .bashrc
```

cp workers.template workers

cd ~

vi .bashrc

source .bashrc



■ 셸 설정

```
#----- SPARK -----  
export SPARK_HOME=/util/spark  
export PATH=$PATH:$SPARK_HOME/bin  
export PATH=$PATH:$SPARK_HOME/sbin
```

<vi 명령어 사용>

j 로 아래로 이동 → #---- HADOOP--- 설정 아래
o 누르고, EnterKey 입력 → 두줄 아래부터 키 입력 가능

```
#----- SPARK -----  
export SPARK_HOME=/util/spark  
export PATH=$PATH:$SPARK_HOME/bin  
export PATH=$PATH:$SPARK_HOME/sbin
```

ESC 콜론(:) wq EnterKey 입력

3. Spark설치

```
[root@linux ~]# echo $PATH
/usr/jdk-19.0.1/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/util/spark/bin
[root@linux ~]#
```

echo \$PATH

3. Spark설치

3

방화벽 설정

- 콘솔과 마스터, 제플린 Port 방화벽 열기

```
[root@linux sbin]# apt install firewalld
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
[root@linux sbin]# firewall-cmd --permanent --add-service=http
success
[root@linux sbin]# firewall-cmd --permanent --add-port=80/tcp
success
[root@linux sbin]# firewall-cmd --permanent --add-port=9090/tcp
success
[root@linux sbin]# firewall-cmd --permanent --add-port=7077/tcp
success
[root@linux sbin]# firewall-cmd --permanent --add-port=10101/tcp
success
[root@linux sbin]#
[root@linux sbin]# firewall-cmd --reload
Error: COMMAND_FAILED: 'python-nftables' failed: internal:0:0:0:
```

```
[root@linux sbin]# firewall-cmd --list-all
public
target: default
icmp-block-inversion: no
interfaces:
sources:
services: dhcpv6-client http ssh
ports: 80/tcp 9090/tcp 7077/tcp 10101/tcp
protocols:
forward: yes
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
[root@linux sbin]#
```

apt install firewalld -y

firewall-cmd --permanent --add-service=http

firewall-cmd --permanent --add-port=80/tcp

firewall-cmd --permanent --add-port=8080/tcp

firewall-cmd --permanent --add-port=7077/tcp

firewall-cmd --permanent --add-port=9090/tcp

firewall-cmd --list-all

아파치용

Spark 콘솔

Spark Master

zeppelin notebook

3. Spark설치

▪ 워크 Port 방화벽 열기

```
[root@linux ~]#
[root@linux ~]#
[root@linux ~]# firewall-cmd --permanent --add-port=8081/tcp
success
[root@linux ~]# firewall-cmd --permanent --add-port=8082/tcp
success
[root@linux ~]# firewall-cmd --permanent --add-port=8083/tcp
success
[root@linux ~]# firewall-cmd --permanent --add-port=4040/tcp
success
[root@linux ~]# firewall-cmd --reload
success
[root@linux ~]# firewall-cmd --list-all
public (active)
target: default
icmp-block-inversion: no
interfaces: ens33
sources:
services: dhcpv6-client http ssh
ports: 80/tcp 8080/tcp 7077/tcp 9090/tcp 8081/tcp 8082/tcp 8083/tcp 4040/tcp
protocols:
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
[root@linux ~]#
```

firewall-cmd --permanent --add-port=8081/tcp

worker-1

firewall-cmd --permanent --add-port=8082/tcp

worker-2

firewall-cmd --permanent --add-port=8083/tcp

worker-3

firewall-cmd --permanent --add-port=4040/tcp

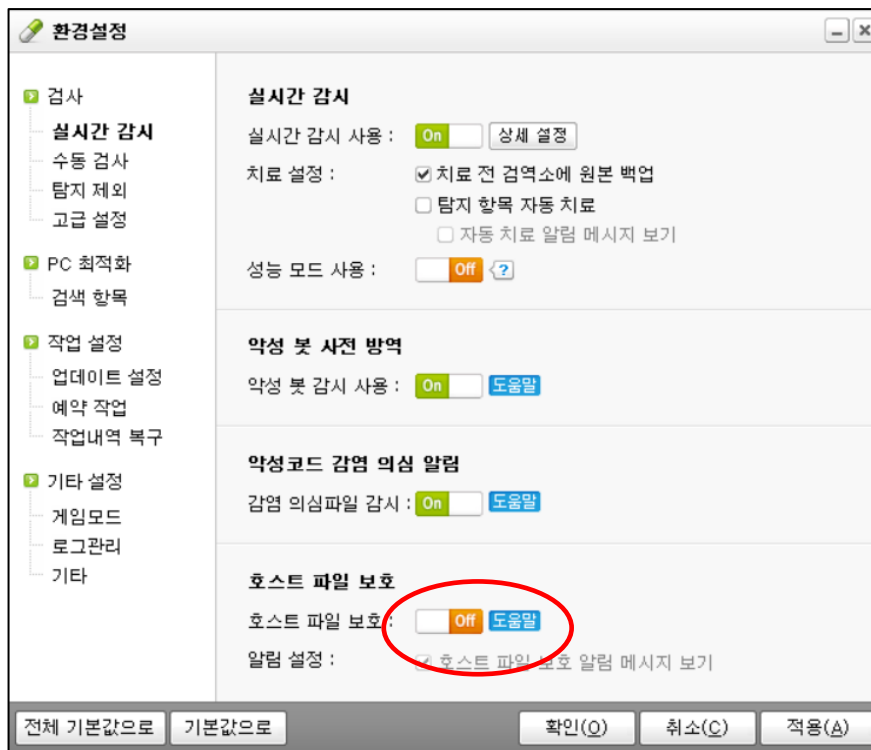
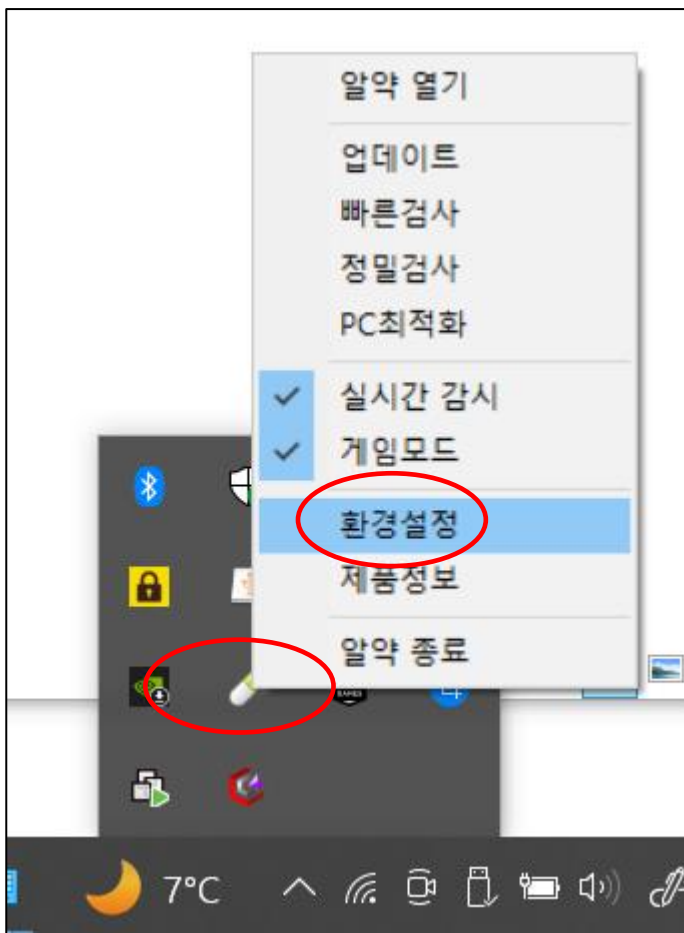
공유 프로세스

firewall-cmd --list-all

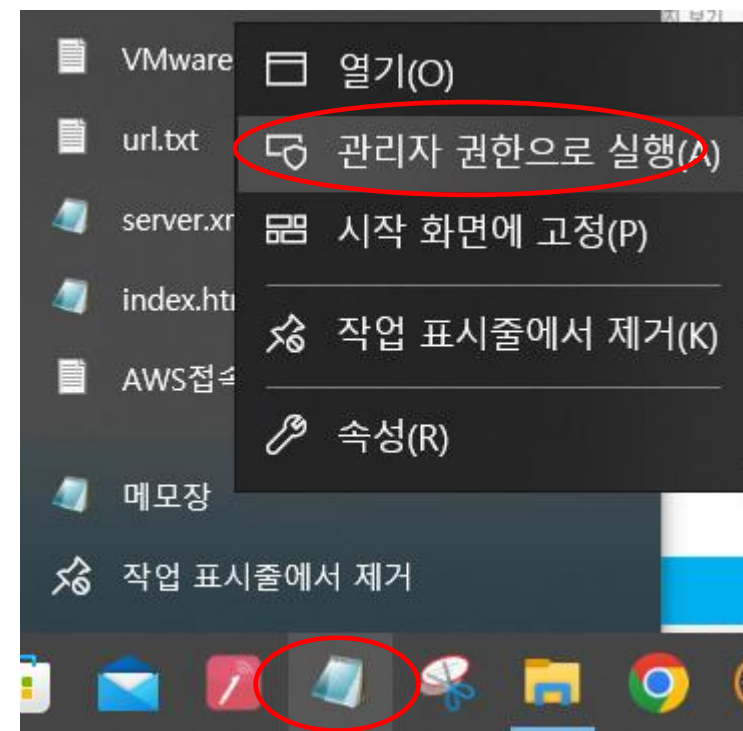
3. Spark설치

4

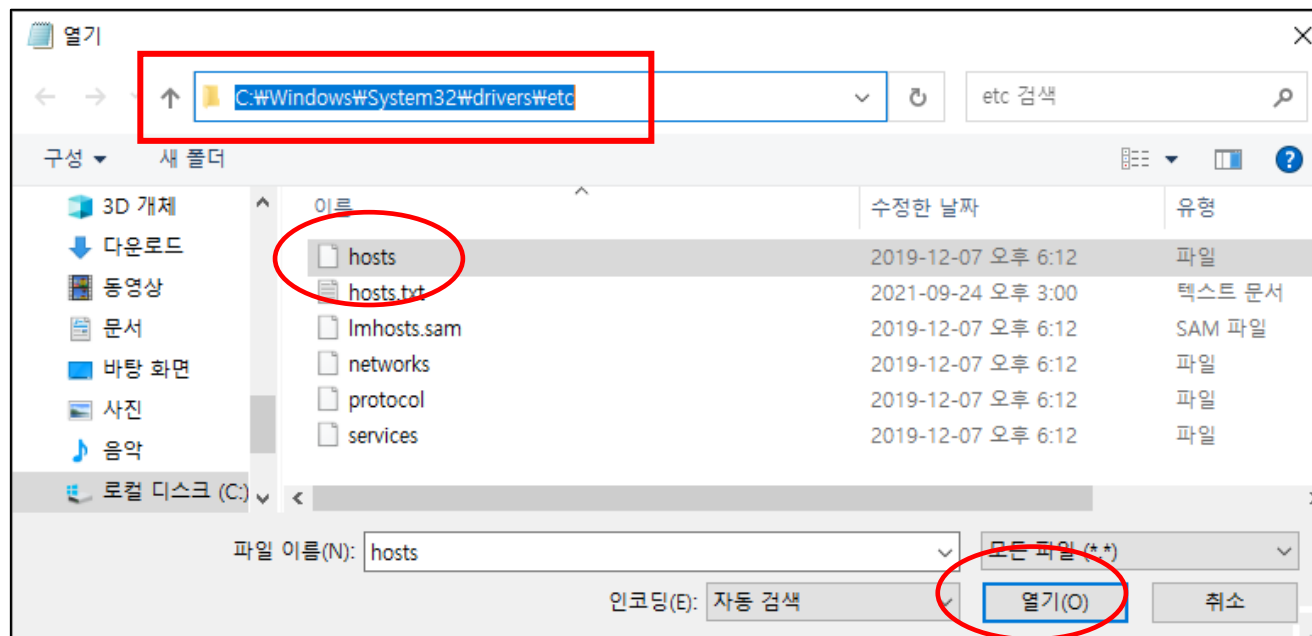
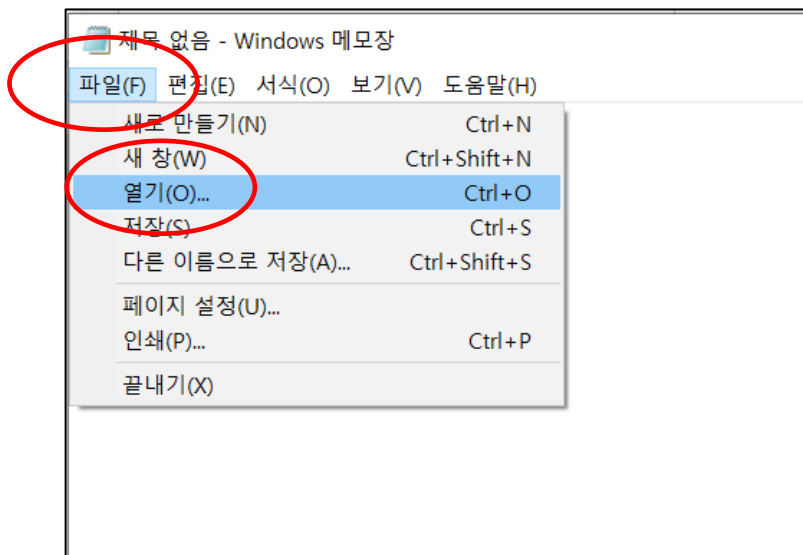
PC 환경설정



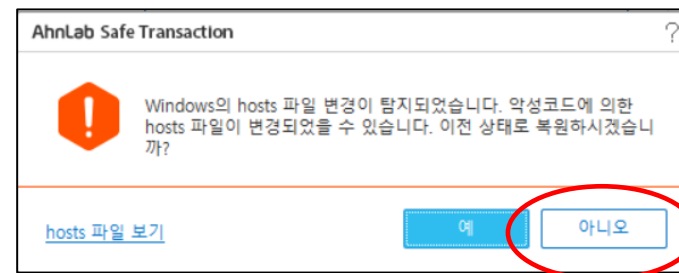
- 호스트 파일 보호 를 OFF 함



3. Spark설치



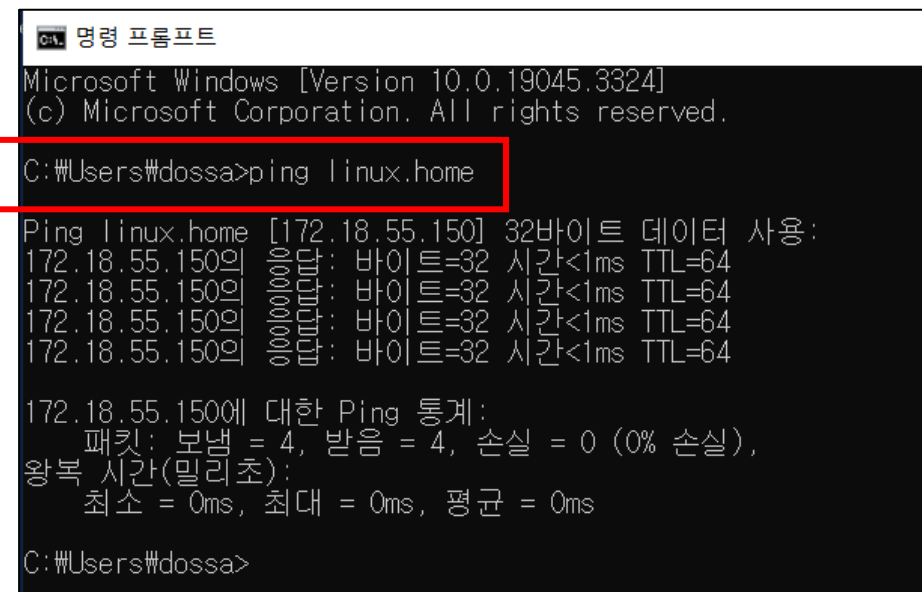
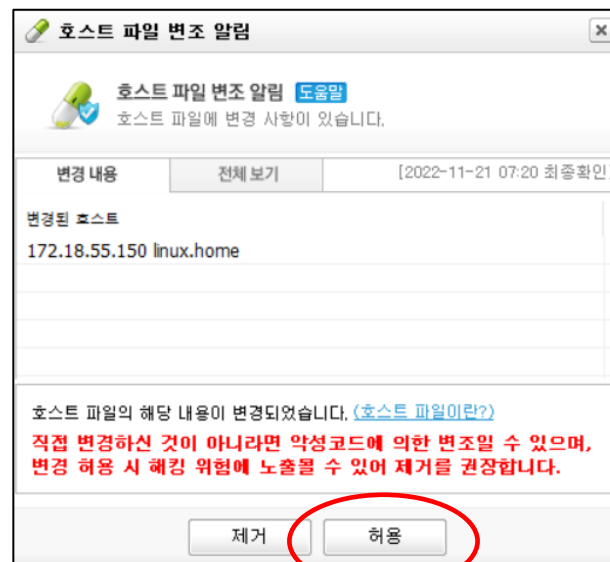
```
# localhost name resolution is handled within DNS itself.
#          127.0.0.1      localhost
#          ::1            localhost
172.18.55.150  linux.home
```



3. Spark설치



- 호스트 파일 보호 를 ON 함



- cmd 창을 띄워서, 확인
- ping linux.home

3. Spark설치

5

Spark 구동

▪ Master 실행

```
[root@linux ~]#  
[root@linux ~]# cd /util/spark/sbin  
[root@linux sbin]# ls  
decommission-slave.sh  start-all.sh  start-slave.sh  stop-connect-server.sh  stop-slaves.sh  
decommission-worker.sh start-connect-server.sh start-slaves.sh stop-history-server.sh stop-thriftserver.sh  
slaves.sh start-history-server.sh start-thriftserver.sh stop-master.sh stop-worker.sh  
spark-config.sh start-master.sh start-worker.sh stop-mesos-dispatcher.sh stop-workers.sh  
spark-daemon.sh start-mesos-dispatcher.sh start-workers.sh stop-mesos-shuffle-service.sh workers.sh  
spark-daemons.sh start-mesos-shuffle-service.sh stop-all.sh stop-slave.sh  
[root@linux sbin]#  
[root@linux sbin]#  
[root@linux sbin]# ./start-master.sh  
starting org.apache.spark.deploy.master.Master, logging to /util/spark/logs/spark-root-org.apache.spark.deploy.master.Master-1-DESKTOP-28CEK70.out  
[root@linux sbin]#
```

cd /util/spark/sbin

ls

./start-master.sh

3. Spark설치

- PC의 크롬에서 우분투의 Spark Master 접속 확인

<http://linux.home:8080>

Spark Master at **spark://DESKTOP-28CEK7O.:7077**

URL: spark://DESKTOP-28CEK7O.:7077
 Alive Workers: 0
 Cores in use: 0 Total, 0 Used
 Memory in use: 0.0 B Total, 0.0 B Used
 Resources in use:
 Applications: 0 Running, 0 Completed
 Drivers: 0 Running, 0 Completed
 Status: ALIVE

▼ Workers (0)

Worker Id	Address	State	Cores	Memory	Resources

▼ Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

▼ Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

이 부분을 마우스로 클릭해서 복사함

아직 Workers 는 살아 있지 않음

3. Spark설치

▪ Slave (Worker) 실행

```
[root@linux sbin]# ls
decommission-slave.sh  start-connect-server.sh  start-worker.sh  stop-slave.sh
decommission-worker.sh start-history-server.sh  start-workers.sh  stop-slaves.sh
run.sh                start-master.sh          stop-all.sh       stop-thriftserver.sh
slaves.sh             start-mesos-dispatcher.sh stop-connect-server.sh stop-worker.sh
spark-config.sh        start-mesos-shuffle-service.sh stop-history-server.sh stop-workers.sh
spark-daemon.sh        start-slave.sh            stop-master.sh     workers.sh
spark-daemons.sh      start-slaves.sh           stop-mesos-dispatcher.sh
start-all.sh          start-thriftserver.sh     stop-mesos-shuffle-service.sh

[root@linux sbin]# ./start-worker.sh spark://DESKTOP-28CEK70.:7077 -m 2g -c 1
starting org.apache.spark.deploy.worker.Worker, logging to /util/spark/logs/spark-root-org.apache.spark.deploy.worker.Wo
rker-1-DESKTOP-28CEK70.out
starting org.apache.spark.deploy.worker.Worker, logging to /util/spark/logs/spark-root-org.apache.spark.deploy.worker.Wo
rker-2-DESKTOP-28CEK70.out
starting org.apache.spark.deploy.worker.Worker, logging to /util/spark/logs/spark-root-org.apache.spark.deploy.worker.Wo
rker-3-DESKTOP-28CEK70.out
[root@linux sbin]#
[root@linux sbin]#
```

./start-worker.sh spark://DESKTOP-28CEK70.:7077 -m 2g -c 1

마스터지정

메모리

CPU

브라우저에서 복사해온 링크

3. Spark설치

■ 재접속 확인

<http://linux.home:8080>

The screenshot shows the Spark Master web interface at `spark://DESKTOP-28CEK7O.:7077`. The interface displays the following information:

- URL:** `spark://DESKTOP-28CEK7O.:7077`
- Alive Workers:** 3
- Cores in use:** 3 Total, 0 Used
- Memory in use:** 6.0 GiB Total, 0.0 B Used
- Resources in use:**
- Applications:** 0 Running, 0 Completed
- Drivers:** 0 Running, 0 Completed
- Status:** ALIVE

The **Workers (3)** section is highlighted with a red circle. It contains a table with the following data:

Worker Id	Address	State	Cores	Memory	Resources
worker-20230911194907-172.18.55.150-33839	172.18.55.150:33839	ALIVE	1 (0 Used)	2.0 GiB (0.0 B Used)	
worker-20230911194910-172.18.55.150-42099	172.18.55.150:42099	ALIVE	1 (0 Used)	2.0 GiB (0.0 B Used)	
worker-20230911194913-172.18.55.150-32945	172.18.55.150:32945	ALIVE	1 (0 Used)	2.0 GiB (0.0 B Used)	

The **Running Applications (0)** section is also visible below the workers table.

Workers 가 3 개 살아남

3. Spark설치

6

Spark 구동확인(Log확인)

- ssh 로 로그인 ※ cmd 창을 새롭게 띄우고 접속

```

C:\Users\dossa>ssh joy@172.18.55.150
joy@172.18.55.150's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

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

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge
Last login: Mon Sep 11 19:58:50 2023 from 172.18.48.1
joy@/home/joy $ su - root
Password:
[root@linux ~]# cd /util/spark/logs
[root@linux logs]# ls
spark-root-org.apache.spark.deploy.master.Master-1-DESKTOP-28CEK70.out
spark-root-org.apache.spark.deploy.worker.Worker-1-DESKTOP-28CEK70.out
spark-root-org.apache.spark.deploy.worker.Worker-2-DESKTOP-28CEK70.out
spark-root-org.apache.spark.deploy.worker.Worker-3-DESKTOP-28CEK70.out
[root@linux logs]#
    
```

ifconfig로 확인

ssh myid@172.18.55.150

su - root

cd /util/spark/logs

3. Spark설치

```
[root@linux logs]#  
[root@linux logs]# tail -f spark-root-org.apache.spark.deploy.master.Master-1-DESKTOP-28CEK70.out  
23/09/10 19:11:23 INFO SecurityManager: Changing view acls groups to:  
23/09/10 19:11:23 INFO SecurityManager: Changing modify acls groups to:  
23/09/10 19:11:23 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view  
w permissions: EMPTY; users with modify permissions: root; groups with modify permissions: EMPTY  
23/09/10 19:11:23 INFO Utils: Successfully started service 'sparkMaster' on port 7077.  
23/09/10 19:11:23 INFO Master: Starting Spark master at spark://DESKTOP-28CEK70.:7077  
23/09/10 19:11:23 INFO Master: Running Spark version 3.4.1  
23/09/10 19:11:23 INFO JettyUtils: Start Jetty 0.0.0.0:9090 for MasterUI  
23/09/10 19:11:23 INFO Utils: Successfully started service 'MasterUI' on port 9090.  
23/09/10 19:11:23 INFO MasterWebUI: Bound MasterWebUI to 0.0.0.0, and started at http://linux.home:9090  
23/09/10 19:11:23 INFO Master: I have been elected leader! New state: ALIVE
```

tail -f spark-root-org.apache.spark.deploy.master.Master-1-DESKTOP-28CEK70.out

3. Spark설치

7

Spark 구동확인 (파이썬 코딩)

```
[root@linux sbin]#  
[root@linux sbin]# cd /util  
[root@linux util]#  
[root@linux util]# mkdir test
```

```
[root@linux util]#  
[root@linux util]# cd test  
[root@linux test]#  
[root@linux test]#
```

```
[root@linux test]#  
[root@linux test]#  
[root@linux test]# vi pyspark-test.py
```

```
cd /util  
mkdir test  
cd test  
vi pyspark-test.py
```

- pyspark-test.py

```
from pyspark import SparkContext, SparkConf  
  
conf = SparkConf()  
conf.setMaster("spark://DESKTOP-28CEK7O.:7077")  
conf.setAppName("seokill")  
sc = SparkContext(conf=conf)  
  
print("="*50, "\n")  
print("안녕하세요~스파크님~")  
print(99 * 1000000)  
print(sc)  
print("="*50, "\n")
```

3. Spark설치

```
[root@linux test]#  
[root@linux test]#  
[root@linux test]# vi run.sh
```

vi run.sh

root@linux:/util/test

```
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop  
spark-submit --master yarn --deploy-mode client pyspark-test.py
```

```
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

```
spark-submit --master yarn --deploy-mode client pyspark-test.py
```

sh run.sh

```
[root@linux test]#  
[root@linux test]# sh run.sh  
22/11/29 20:42:17 INFO SparkContext:  
22/11/29 20:42:17 WARN NativeCodeLoad  
asses where applicable
```

```
99000000  
<SparkContext master=spark://linux.home:7077 appName=seokill>
```

- 정상적으로 프로그램이 실행 됨

3. Spark설치

8

Spark 구동 Shell 만들어 두기

```
[root@linux ~]# cd /util/spark/sbin
[root@linux sbin]#
[root@linux sbin]#
[root@linux sbin]# vi run.sh
```

cd /util/spark/sbin

vi run.sh

Open +

run.sh

/util/spark/sbin

auto run

===== Spark Start =====

cd /util/spark/sbin

./start-master.sh

./start-slave.sh spark://DESKTOP-28CEK70.:7077 -m 2g -c 1

auto run

===== Spark Start =====

cd /util/spark/sbin

./start-master.sh

./start-slave.sh spark://DESKTOP-28CEK70.:7077 -m 2g -c 1

```
[root@linux sbin]#
```

```
[root@linux sbin]# chmod 755 run.sh
```

chmod 755 run.sh

```
[root@linux sbin]#
```

```
[root@linux sbin]# cp run.sh /root/run-spark.sh
```

cp run.sh /root/run-spark.sh

```
[root@linux sbin]#
```

4. 전달 사항



교재

주교재

- PowerPoint 로 만든 pdf 자료
- 데이터 과학 기반의 파이썬 빅데이터 분석 (이지영 지음, 한빛아카데미)

부교재

- 필요 시, 영상 공유



4. 전달 사항

RoadMap

Hadoop설치

- ✓ VM 셋업
- ✓ JDK
- ✓ Python
- ✓ Hadoop Engine
- ✓ Spark Engine
- ✓ Zeppelin

빅데이터분석

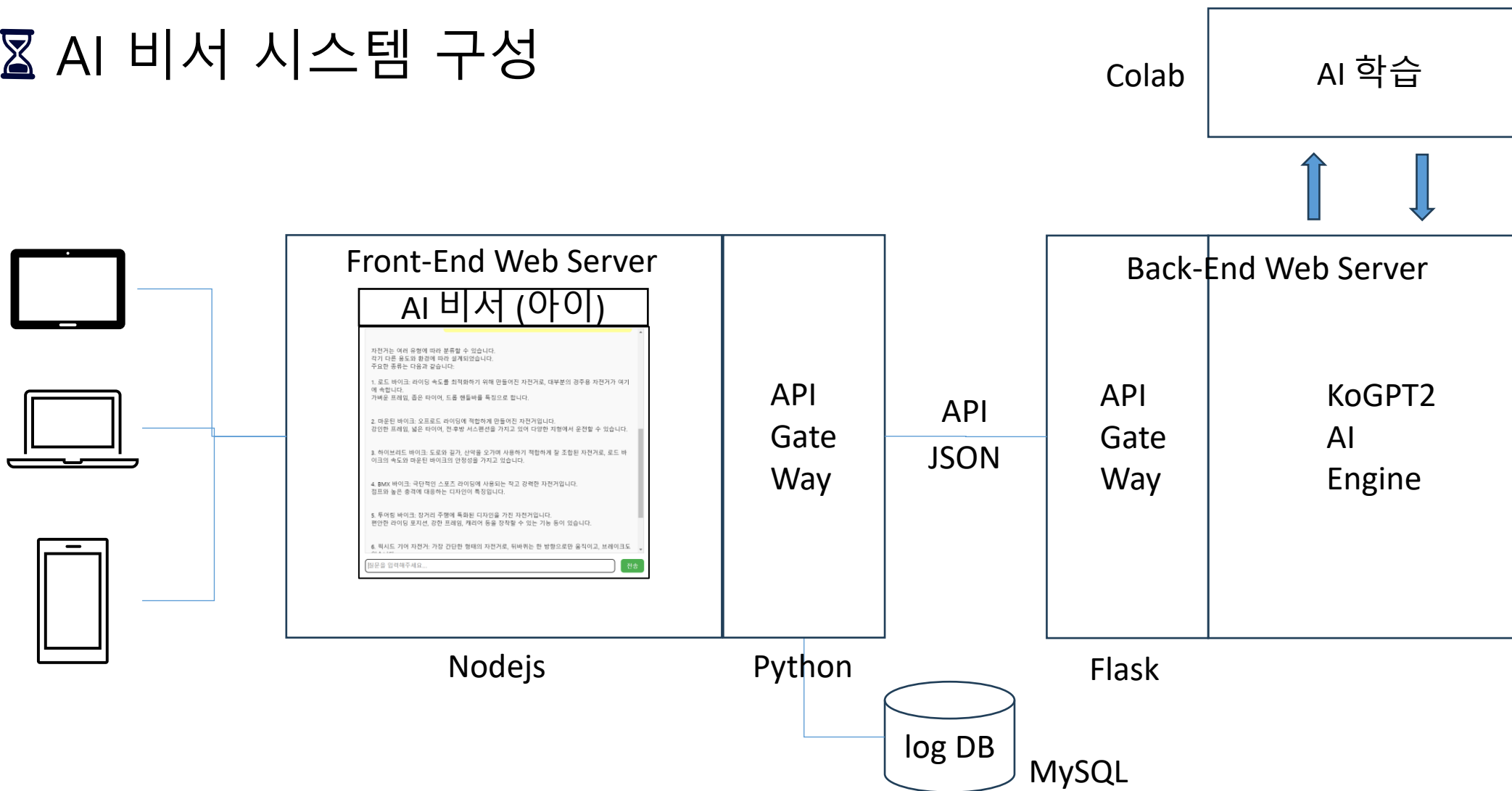
- ✓ 빅데이터 산업의 이해
- ✓ 파이썬 프로그래밍
- ✓ 크롤링
- ✓ 통계분석
- ✓ 텍스트빈도분석
- ✓ 지리정보분석
- ✓ 회귀분석/분류분석
- ✓ 텍스트마이닝

AI 비서학습

- ✓ 챗봇 데이터 수집
- ✓ Flask 웹서버
- ✓ Nodejs API 연동
- ✓ KoGPT2 환경구성
- ✓ Colab을 이용한 학습
- ✓ 말풍선생성기 활용
- ✓ MySQL
- ✓ 챗봇 비서 만들기

4. 전달 사항

⌚ AI 비서 시스템 구성



참고 자료

- 자바와 파이썬으로 만드는 빅데이터시스템(제이펍, 황세규)
- 위키독스(<https://wikidocs.net/22654>)
- 네이버블로그(<https://blog.naver.com/classmethodkr/222822485338>)
- 데이터분석과 인공지능 활용 (NOSVOS, 데이터분석과인공지능활용편찬위원회 편)

참고 사이트

유튜버 : 빅공잼 : <https://www.youtube.com/watch?v=bnYxO2XRCQ0>

네이버 블로그 : 빅공잼

<https://biggongjam.notion.site/3-Hadoop-cd6944182da74edf8d2339b654e0bfb9>

<https://biggongjam.notion.site/4-Spark-2c341ddc8715411484cb2f0254b60126>

Q n A

* wsl 실행 시 오류

#Case-1 wsl 이 실행이 되지 않으면,

#PowerShell을 관리자 권한으로 실행하고 다음 명령을 입력하십시오:

`Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All`

#Case-2 Error code: Wsl/Service/CreateInstance/CreateVm/0x80370102

#와 같은 오류가 발생하면, 다음 명령을 입력하십시오:

`bcdedit /set hypervisorlaunchtype Auto`

#Case-3 `wsl -l -v` 명령을 수행 시, 아직도 우분투가 Installing 중으로 나오면, 아래의 명령을 수행하십시오:

`wsl --unregister Ubuntu-22.04`

* SSH 설정

1) Linux (우분투) 서버에 ssh 데몬 설치 및 실행

```
sudo apt-get install openssh-server
```

```
sudo apt-get install ssh
```

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

```
ps -ef | grep ssh
```

* SSH 설정

2) sshd 설정 변경

`sudo vi /etc/ssh/sshd_config`

```
Port 22
AddressFamily any
ListenAddress 0.0.0.0
ListenAddress ::
```

- 맨 앞의 # 를 제거하고 저장

`sudo service ssh restart`

- ssh 데몬을 재시작

* SSH 설정

3) Linux(우분투) 방화벽 설정 변경

`sudo ufw enable`

- 방화벽에서 22번 포트 허용

`sudo ufw allow 22`

4) Linux(우분투) 의 IP 주소 확인

`ifconfig -a`

- ens33 에서 inet 에 등록된 IP 주소 확인

➔ 예시) 192.168.121.128

* SSH 설정

5) Windows PowerShell 에서 SSH 접속

ssh seokill@192.168.121.128

```
PS C:\Users\dossa> ssh seokill@192.168.121.128
The authenticity of host '192.168.121.128 (192.168.121.128)' can't be established.
ECDSA key fingerprint is SHA256:7Ymko23VvqHHX27lqhPeyxFrquCtDHR7tvYTZ3Bdl0s.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.121.128' (ECDSA) to the list of known hosts.
seokill@192.168.121.128's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-40-generic x86_64)

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

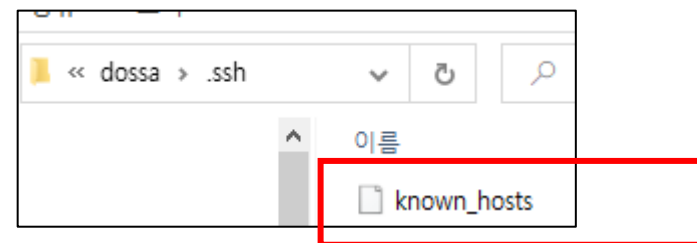
Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
추가 업데이트를 확인하려면 apt list --upgradable 을 실행하세요.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Sun May 7 17:42:13 2023
seokill@/home/seokill $
seokill@/home/seokill $
```

- yes : 기본 인증키 발급 허용
- 접속 ID 의 비밀번호 입력



- 인증키가 생성됨
- 사용자\ssh\known_hosts

* gnome-text-editor 활용

1) x-window (xming) 설치-pc

<https://sourceforge.net/projects/xming/>

2) gnome-text-editor 설치

```
[root@linux Downloads]#  
[root@linux Downloads]# apt install gnome-text-editor -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
aspell aspell-en dictionaries-common emacsen-common enchant-2  
libcairo-script-interpreter2 libenchant-2-2 libgraphene-1.0-0  
libgtksourceview-5-0 libgtksourceview-5-common libhunspell-1.4-0
```

`sudo apt update`

`sudo apt install gnome-text-editor -y`

3) .bashrc 에 내용 추가

`cd ~`
`vi .bashrc`

```
export DISPLAY=:0  
export LIBGL_ALWAYS_INDIRECT=0
```

추가 후, 반드시 오른쪽 명령 실행 `source .bashrc`

4) 수정을 원하는 파일이름 입력

```
[root@linux ~]# gnome-text-editor .bashrc  
Gsk-Message: 06:26:46.844: Failed to realize renderer of type 'GskGLRe  
ot available  
  
Gsk-Message: 06:27:28.288: Failed to realize renderer of type 'GskGLRe  
available
```

`gnome-text-editor .bashrc`