

빅데이터시스템

제플린(Zeppelin)

01 Zeppelin 설치

다운로드, 서버환경설정, 구동, 활용, Shell만들기

목 차

--

전달 사항

전수업리뷰, 로드맵, 교재

*. 전수업리뷰

⌚ 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

* . 전수업리뷰

⌚ 빅데이터의 이해

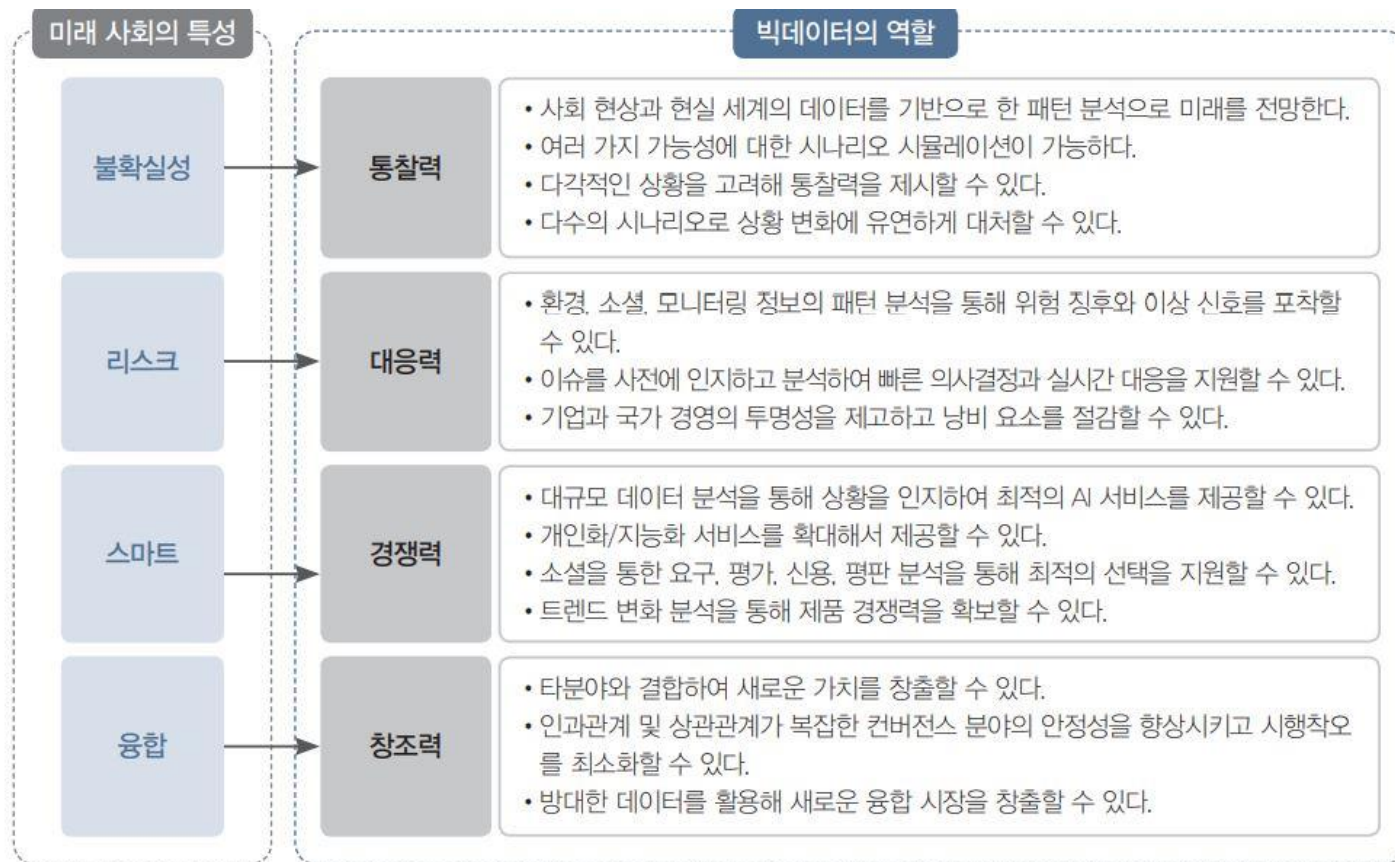
■ 빅데이터의 분석환경

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

*. 전수업리뷰

⌚ 빅데이터의 활용

■ 빅데이터의 역할



*. 전수업리뷰

⌚ 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.

*. 전수업리뷰

⌚ 빅데이터의 활용

■ 처리 단계별 기술영역

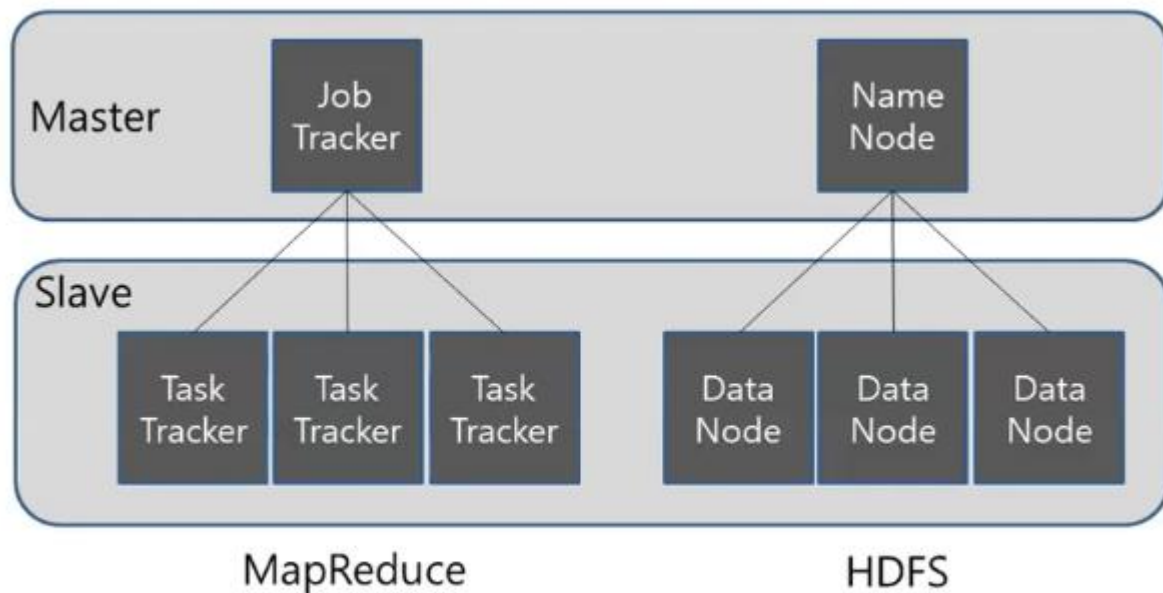
단계	기술 영역	내용
데이터 소스	내부 데이터	데이터베이스, 파일 관리 시스템
	외부 데이터	파일, 멀티미디어, 스트리밍
수집	크롤링 <small>crawling</small>	검색 엔진 로봇을 이용한 데이터 수집
	ETL: 추출 <small>Extraction</small> , 변환 <small>Transformation</small> , 적재 <small>Loading</small>	소스 데이터의 추출, 전송, 변환, 적재
저장	데이터 관리: NoSQL	비정형 데이터 관리
	저장소	빅데이터 저장
	서버	초경량 서버
처리	맵리듀스 <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

*. 전수업리뷰

⌚ 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

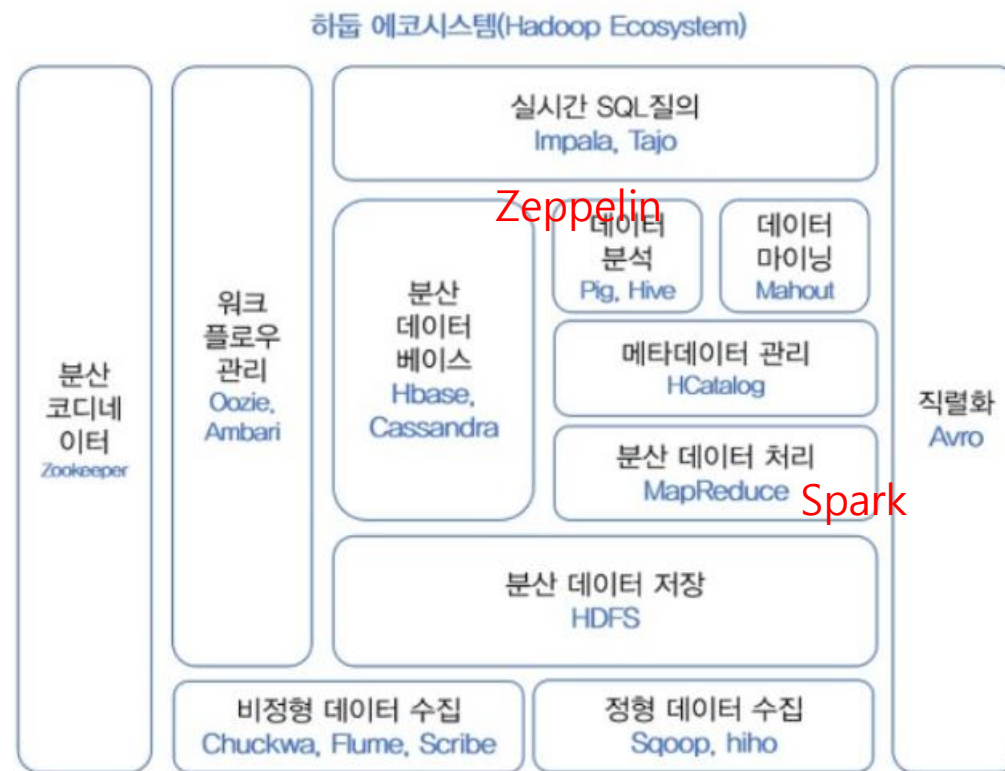
기본골격



한기철, K-ICT 빅데이터 교육교재 중 발췌

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

기능확장



하둡 프로그래밍(위키북스)

*. 전수업리뷰

⌚ 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>
```

*. 전수업리뷰

⌚ wsl

wsl --update

wsl --import Ubuntu-22.04 .\Ubuntu-22.04\ .\u-22.04.tar

메모장 열고

wsl --set-default Ubuntu-22.04

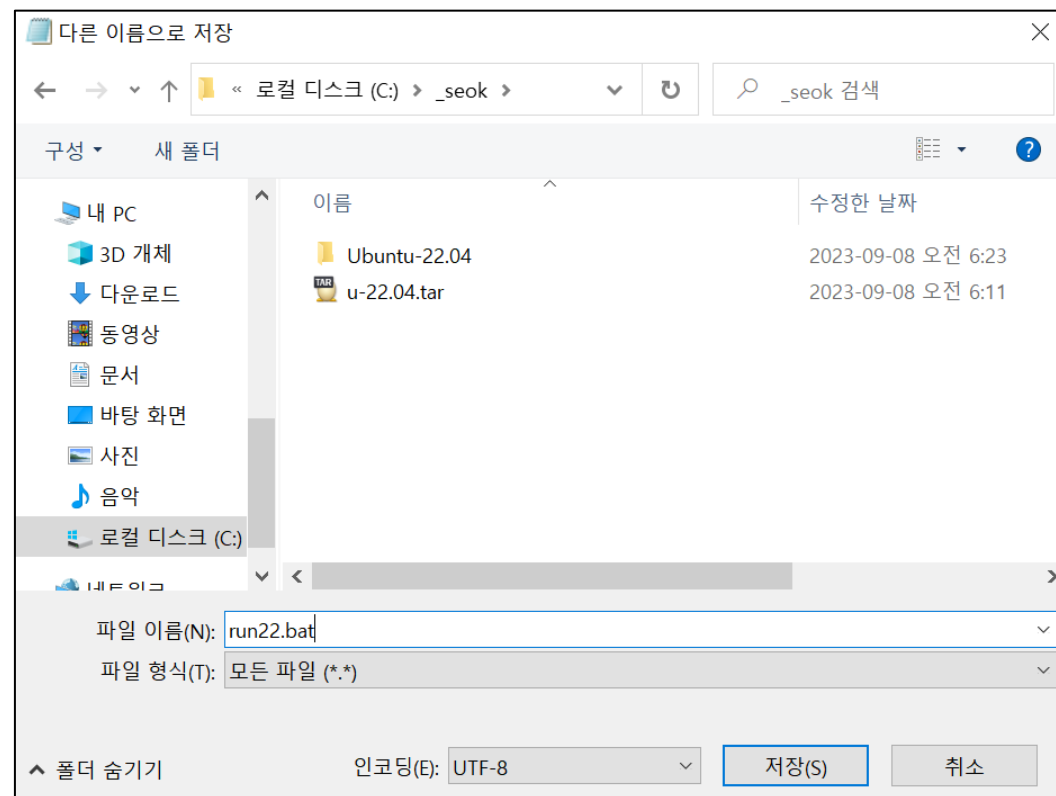
wsl --distribution Ubuntu-22.04

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



run22.bat 으로 저장

*. 전수업리뷰

⌚ jvm

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

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

*. 전수업리뷰

⌚ 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

*. 전수업리뷰

⌚ 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+dfsg-1ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 52 not upgraded.
[root@linux ~]#
```

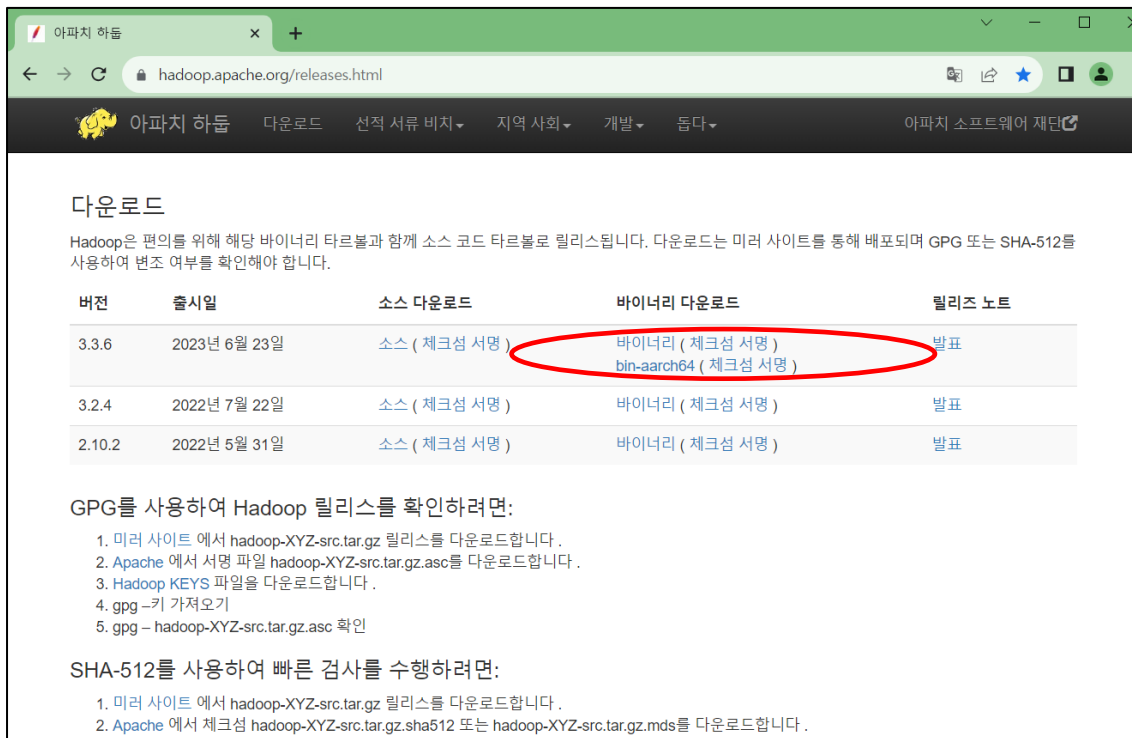
python -V
python3 -V

apt-get install -y python3-pip

*. 전수업리뷰

⌚ Hadoop

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



다운로드

Hadoop은 편의를 위해 해당 바이너리 타르볼과 함께 소스 코드 타르볼로 릴리스됩니다. 다운로드를 미래 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. 미래 사이트에서 `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. 미래 사이트에서 `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.

- 최신버전의 <바이너리(체크섬 서명)> 클릭



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

HTTP

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

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

- 링크에서 오른쪽 마우스 <링크 주소 복사> 클릭

*. 전수업리뷰

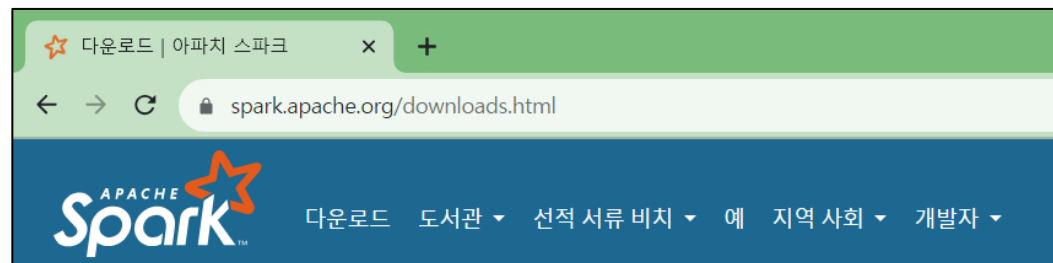
⌚ 환경설정파일

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 	

*. 전수업리뷰

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

⌚ Spark



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 검증하십시오.

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.



*. 전수업리뷰

⌚ Spark

```
[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")
```


*. 전수업리뷰

⌚ Spark

```
[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

*. 전달 사항



교재

주교재

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

부교재

- 필요 시, 영상 공유



*. 전달 사항

RoadMap

Hadoop설치

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

빅데이터분석

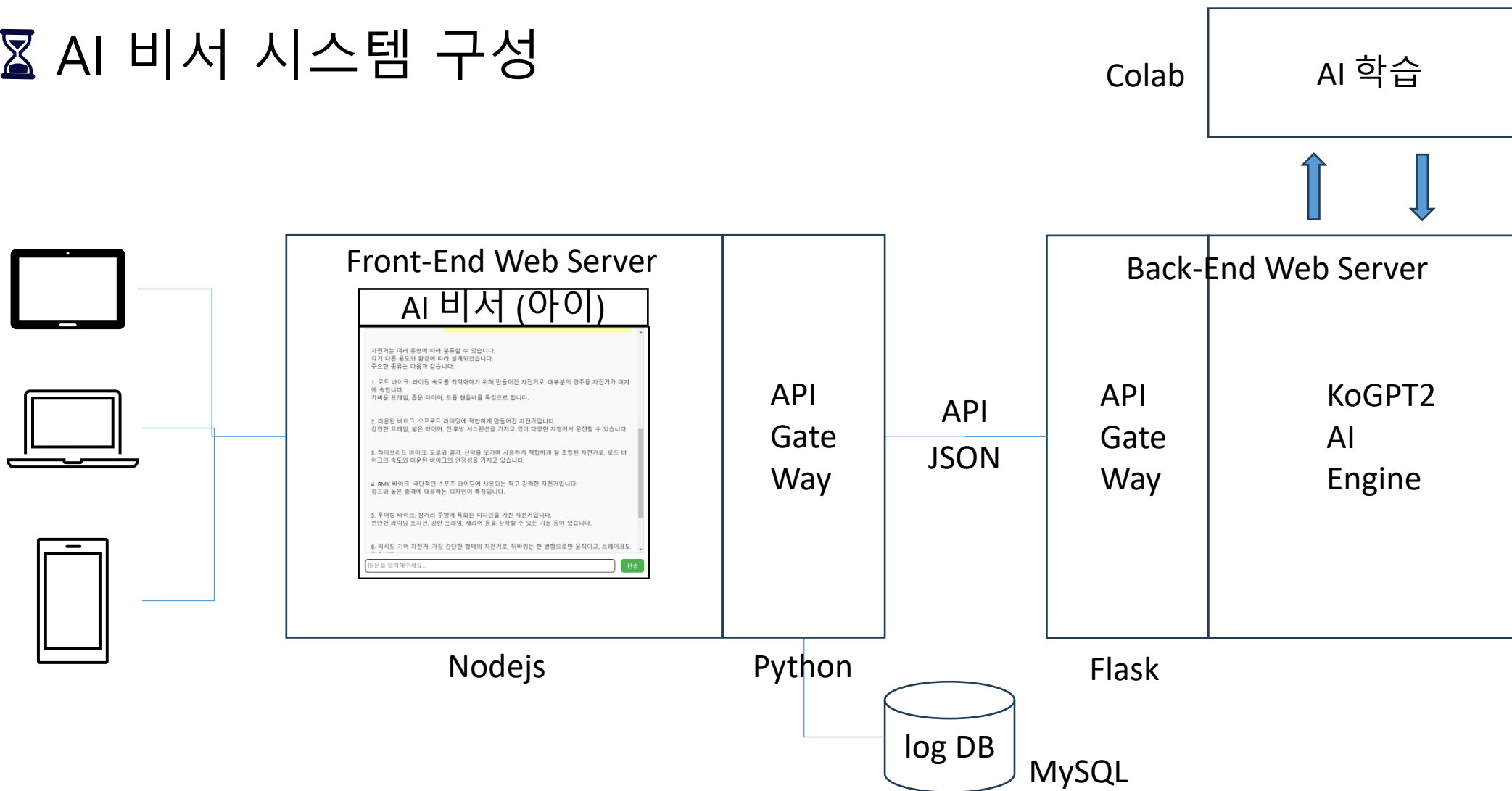
- ✓ 빅데이터 산업의 이해
- ✓ 파이썬 프로그래밍
- ✓ 크롤링
- ✓ 통계분석
- ✓ 텍스트빈도분석
- ✓ 지리정보분석
- ✓ 회귀분석/분류분석
- ✓ 텍스트마이닝

AI 비서학습

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

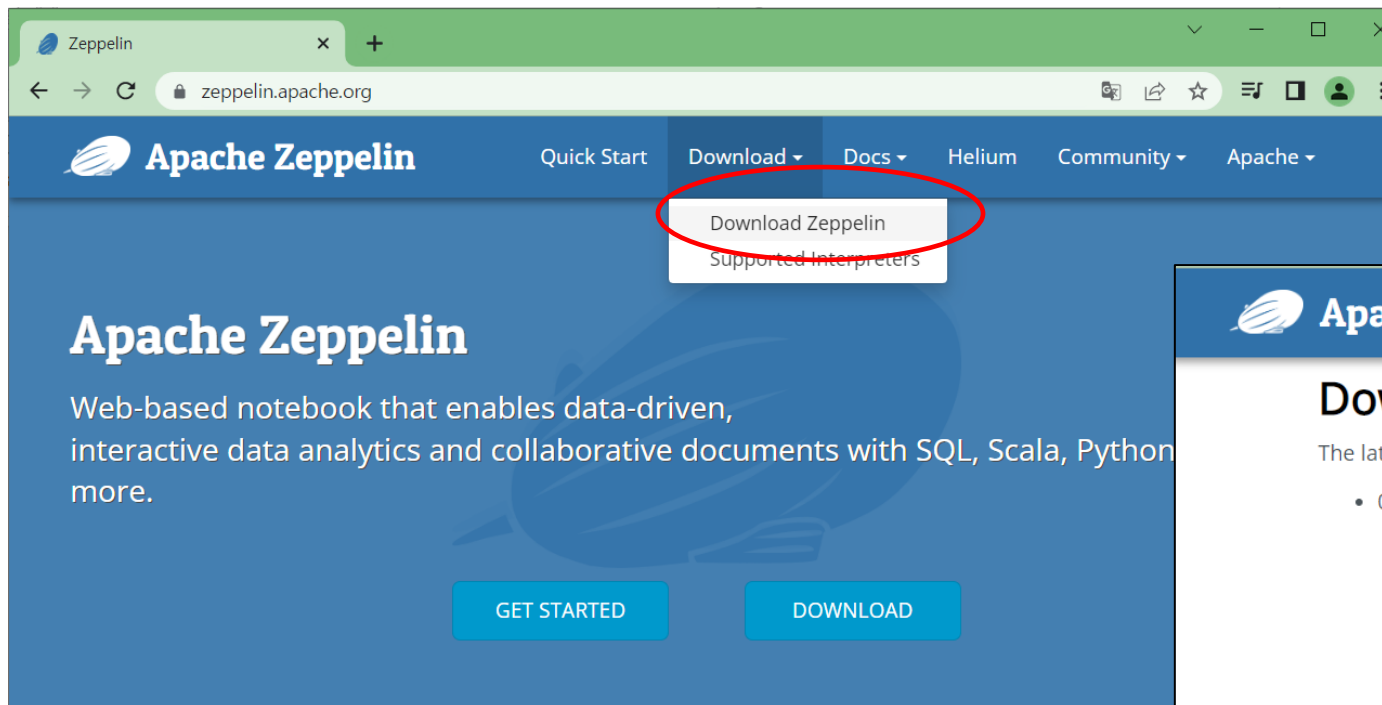
*. 전달 사항

⌚ AI 비서 시스템 구성



1. Zeppelin설치

1 다운로드



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



1. Zeppelin설치



1. Zeppelin설치

```
[root@localhost conf]# cd /util
[root@localhost util]# ls -al
drwxr-xr-x. 3 root root 54 11월 21 03:57 .
dr-xr-xr-x. 18 root root 236 11월 21 03:50 ..
drwxr-xr-x. 14 110302528 dorocy 223 11월 21 04:04 spark
-rw-r--r--. 1 root root 299350810 10월 15 19:53 spark-3.3.1-bin-hadoop3.tgz

[root@localhost util]#
[root@localhost util]# wget https://dlcdn.apache.org/zeppelin/zeppelin-0.10.1/zeppelin-0.10.1-bin-all.tgz
--2022-11-21 04:57:07-- https://dlcdn.apache.org/zeppelin/zeppelin-0.10.1/zeppelin-0.10.1-bin-all.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: 1680577910 (1.6G) [application/x-gzip]
Saving to: 'zeppelin-0.10.1-bin-all.tgz'

100%[=====>] 1,680,577,910

2022-11-21 05:00:14 (8.73 MB/s) - 'zeppelin-0.10.1-bin-all.tgz' saved [1680577910/1680577910]

[root@localhost util]# ls
spark spark-3.3.1-bin-hadoop3.tgz zeppelin-0.10.1-bin-all.tgz
[root@localhost util]# tar -zxvf zeppelin-0.10.1-bin-all.tgz
[root@localhost util]#
[root@localhost util]#
```

cd /util
ls -al

tar -zxvf zeppelin-0.10.1-bin-all.tgz

wget https://dlcdn.apache.org/zeppelin/zeppelin-0.10.1/zeppelin-0.10.1-bin-all.tgz

1. Zeppelin설치

```
[root@localhost util]#
[root@localhost util]# ls
spark  spark-3.3.1-bin-hadoop3.tgz  zeppelin-0.10.1-bin-all  zeppelin-0.10.1-bin-all.tgz
[root@localhost util]#
[root@localhost util]# mv zeppelin-0.10.1-bin-all zeppelin
[root@localhost util]#
[root@localhost util]# ls
spark  spark-3.3.1-bin-hadoop3.tgz  zeppelin  zeppelin-0.10.1-bin-all.tgz
[root@localhost util]#
[root@localhost util]#
```

mv zeppelin-0.10.1-bin-all zeppelin

1. Zeppelin설치

2

서버 환경설정

```
[root@localhost util]#  
[root@localhost util]# cd zeppelin  
[root@localhost zeppelin]#  
[root@localhost zeppelin]# ls  
LICENSE  README.md  conf      k8s    licenses  plugins  zeppelin-web-0.10.1.war  
NOTICE   bin        interpreter  lib    notebook  scripts  zeppelin-web-angular-0.10.1.war  
[root@localhost zeppelin]#  
[root@localhost zeppelin]# cd conf  
[root@localhost conf]#  
[root@localhost conf]# ls  
configuration.xml  log4j.properties  log4j2.properties  shiro.ini.template  zeppelin-env.sh.template  
interpreter-list   log4j.properties2  log4j_yarn_cluster.properties  zeppelin-env.cmd.template  zeppelin-site.xml.template  
[root@localhost conf]#  
[root@localhost conf]# cp zeppelin-site.xml.template zeppelin-site.xml  
[root@localhost conf]#  
[root@localhost conf]# vi zeppelin-site.xml
```

```
cd zeppelin  
cd conf  
cp zeppelin-site.xml.template zeppelin-site.xml  
vi zeppelin-site.xml
```

1. Zeppelin설치

```
<configuration>
<property>
<name>zeppelin.server.addr</name>
<value>127.0.0.1</value>
<description>Server binding address</description>
</property>
<property>
<name>zeppelin.server.port</name>
<value>8080</value>
<description>Server port.</description>
</property>
```

```
<configuration>
<property>
<name>zeppelin.server.addr</name>
<value>172.18.55.150</value>
<description>Server binding address</description>
</property>
<property>
<name>zeppelin.server.port</name>
<value>9090</value>
<description>Server port.</description>
</property>
```

ifconfig 명령어로
우분투의 ip 확인

<vi 명령어 사용>

j 로 아래로 이동
l 로 오른쪽으로 → 127 문자 까지
x 로 문자 삭제 → 127.0.0.1 모두 삭제
i 를 누르고, 172.18.55.150 입력

ESC 누르고,

j, h, l 키로 아래로 이동 → 8080 문자까지
x 로 문자 삭제 → 8080 모두 삭제
i 를 누르고, 9090 입력

ESC 콜론(:) wq EnterKey 입력

1. Zeppelin설치

```
[root@localhost conf]#
```

```
[root@localhost conf]# cp zeppelin-env.sh.template zeppelin-env.sh
```

```
[root@localhost conf]#
```

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

cp zeppelin-env.sh.template zeppelin-env.sh

vi zeppelin-env.sh

1. Zeppelin설치

- 맨 아래로 이동 하여 아래의 내용 입력 (맨 아래로 이동 vi에서는 \$G)

본인의 Spark
마스터 주소

```

Open ▾ + • zeppelin-env.sh /util/zeppelin/conf
Zeppelin with authentication.

#### Zeppelin
# export ZEPPELIN_JAVA_HOME=/usr/jdk-20.0.2
# export ZEPPELIN_SPARK_HOME=/util/spark
# export ZEPPELIN_HADOOP_HOME=/util/hadoop
# export ZEPPELIN_YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
# export ZEPPELIN_HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
# export ZEPPELIN_MASTER=spark://DESKTOP-28CEK7O:7077
# export ZEPPELIN_ZEPPELIN_PORT=9090
# export ZEPPELIN_PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9.7-src.zip:/usr/bin/python3
# export ZEPPELIN_PYSARK_PYTHON=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9.7-src.zip:$SPARK_HOME/python/lib/pyspark.zip:/usr/bin/python3
# export ZEPPELIN_PYSARK_DRIVER_PYTHON=/usr/bin/python3

export JAVA_HOME=/usr/jdk-20.0.2
export SPARK_MASTER=spark://DESKTOP-28CEK7O:7077
export SPARK_HOME=/util/spark
export HADOOP_HOME=/util/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export MASTER=spark://DESKTOP-28CEK7O:7077
export ZEPPELIN_PORT=9090
export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9.7-src.zip:/usr/bin/python3
export PYSARK_PYTHON=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9.7-src.zip:$SPARK_HOME/python/lib/pyspark.zip:/usr/bin/python3
export PYSARK_DRIVER_PYTHON=/usr/bin/python3
  
```

```

export JAVA_HOME=/usr/jdk-20.0.2
export SPARK_MASTER=spark://DESKTOP-28CEK7O:7077
export SPARK_HOME=/util/spark
export HADOOP_HOME=/util/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export MASTER=spark://DESKTOP-28CEK7O:7077
export ZEPPELIN_PORT=9090
export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9.7-src.zip:/usr/bin/python3
export PYSARK_PYTHON=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9.7-
src.zip:$SPARK_HOME/python/lib/pyspark.zip:/usr/bin/python3
export PYSARK_DRIVER_PYTHON=/usr/bin/python3
  
```

1. Zeppelin설치

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

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

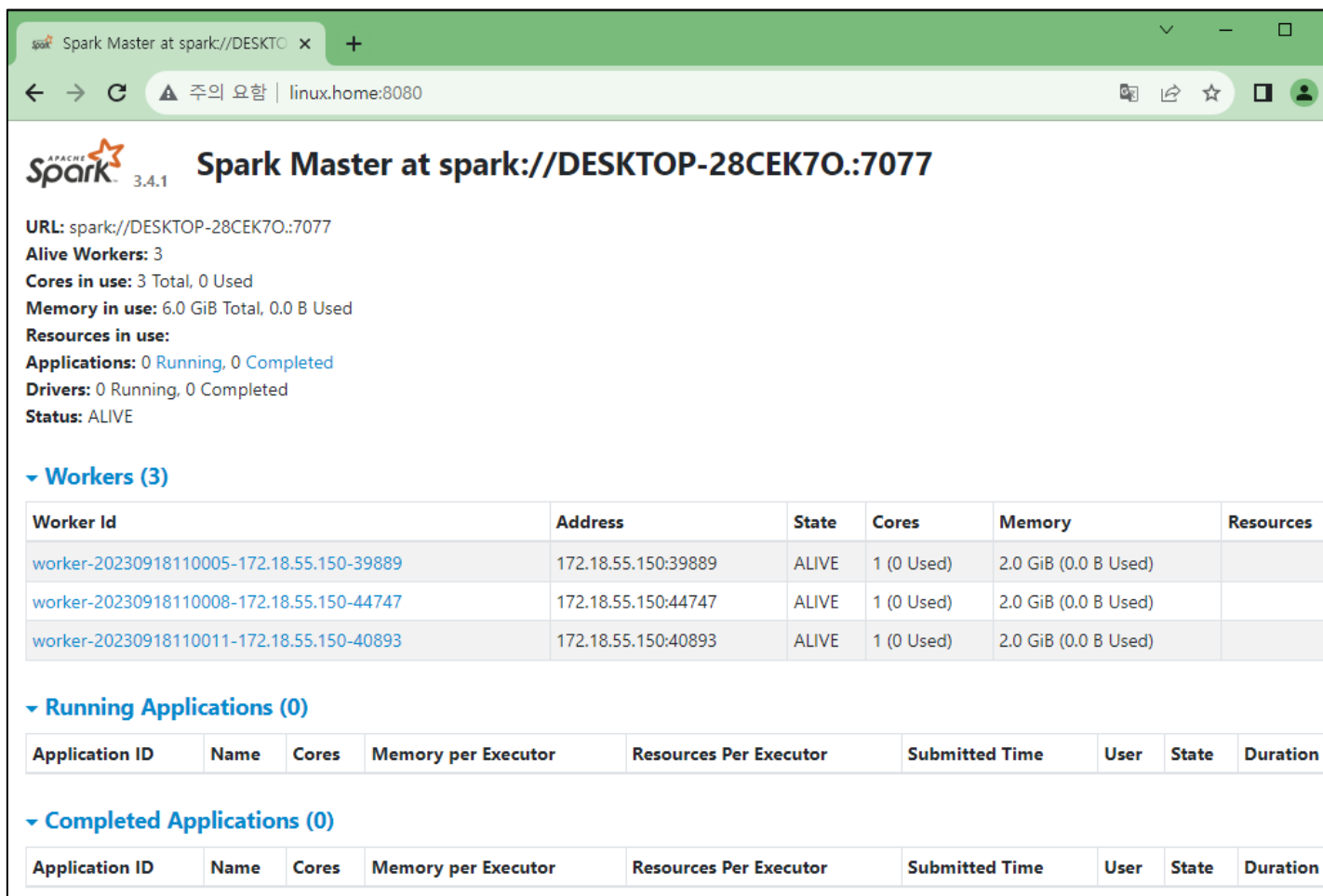


source .bashrc

```
#----- ZEPPELIN -----  
export ZEPPELIN_HOME=/util/zeppelin  
export PATH=$PATH:$ZEPPELIN_HOME/bin
```


1. Zeppelin설치

3 zeppelin 구동



The screenshot shows the Spark Master web interface at the URL `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

Below the summary, there are three expandable sections:

- Workers (3):** A table showing 3 workers in an ALIVE state, each with 1 core and 2.0 GiB of memory.
- Running Applications (0):** A table showing 0 running applications.
- Completed Applications (0):** A table showing 0 completed applications.

Worker Id	Address	State	Cores	Memory	Resources
worker-20230918110005-172.18.55.150-39889	172.18.55.150:39889	ALIVE	1 (0 Used)	2.0 GiB (0.0 B Used)	
worker-20230918110008-172.18.55.150-44747	172.18.55.150:44747	ALIVE	1 (0 Used)	2.0 GiB (0.0 B Used)	
worker-20230918110011-172.18.55.150-40893	172.18.55.150:40893	ALIVE	1 (0 Used)	2.0 GiB (0.0 B Used)	

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
No running applications.								

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
No completed applications.								

`http://linux.home:8080/`

Spark 이 살아 있는 지 확인

1. Zeppelin설치

```
[root@linux ~]#
[root@linux ~]# cd /util/zeppelin/bin
[root@linux bin]#
[root@linux bin]# pwd
/util/zeppelin/bin
[root@linux bin]#
[root@linux bin]# ./zeppelin-daemon.sh stop
Zeppelin stop [ OK ]
[root@linux bin]#
[root@linux bin]# ./zeppelin-daemon.sh start
Zeppelin start [ OK ]
[root@linux bin]#
[root@linux bin]# _
```

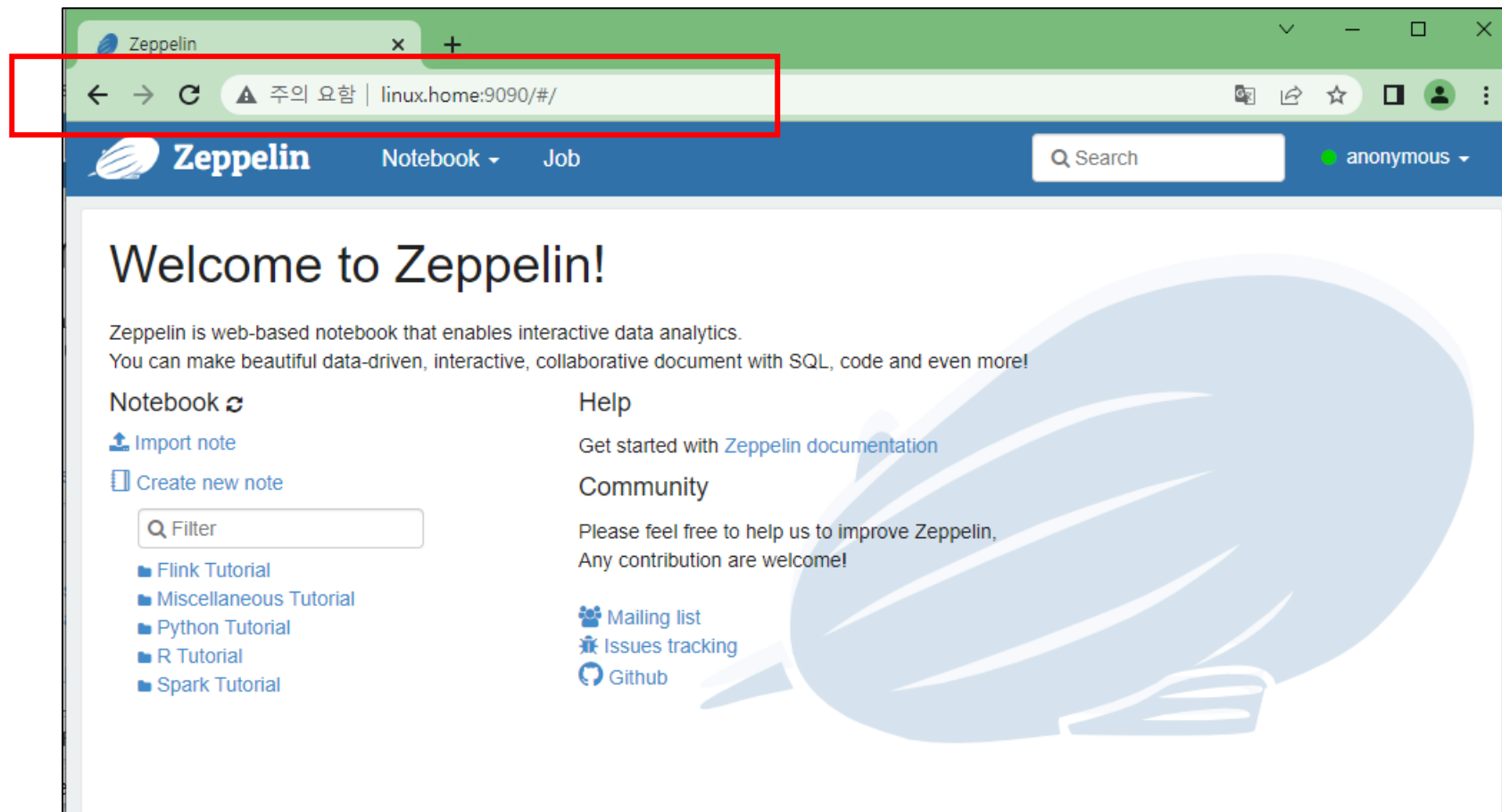
```
cd /util/zeppelin/bin
pwd
./zeppelin-daemon.sh start
```

1. Zeppelin설치

4

zeppelin 접속

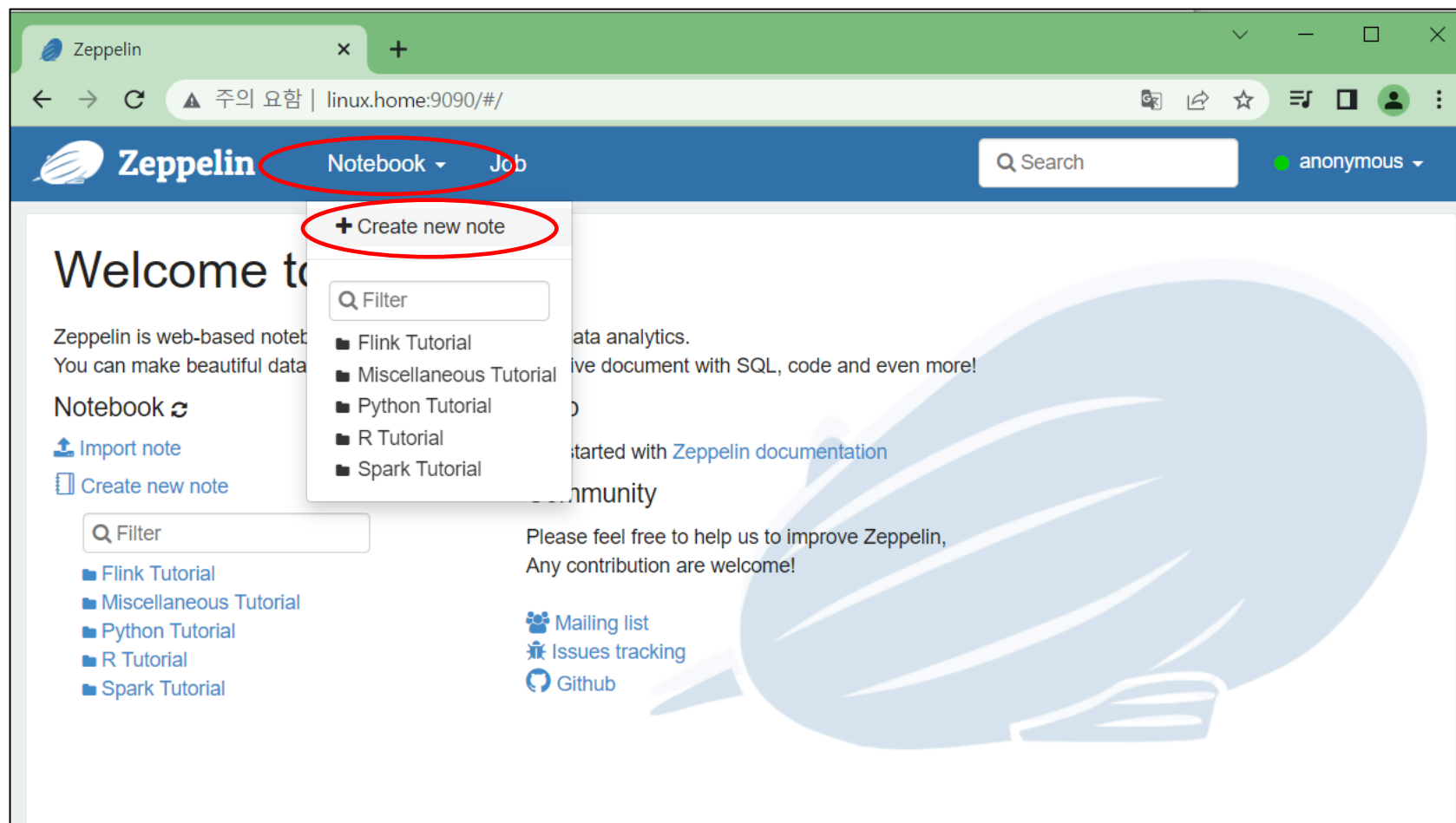
<http://linux.home:9090>



1. Zeppelin설치

5

zeppelin 활용



1. Zeppelin설치

Create New Note

Note Name

exam01

Default Interpreter

spark

Use '/' to create folders. Example: /NoteDirA/Note1

Create

- exam01 로 입력

Zeppelin Notebook Job

Search anonymous

exam01

sc

ERROR

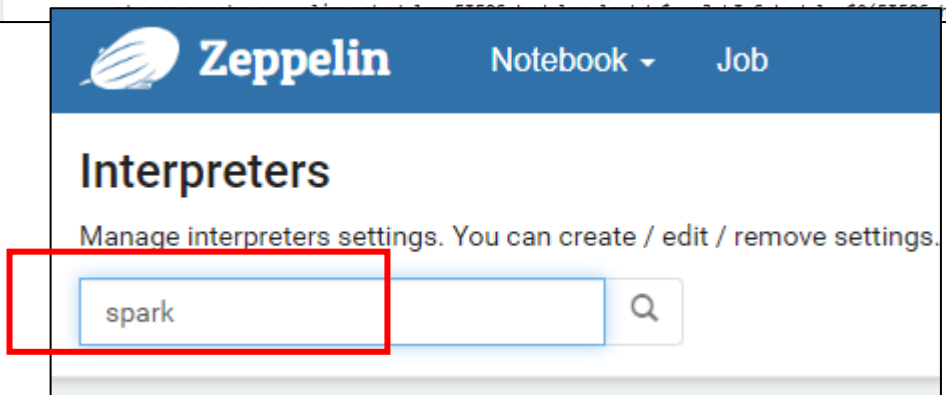
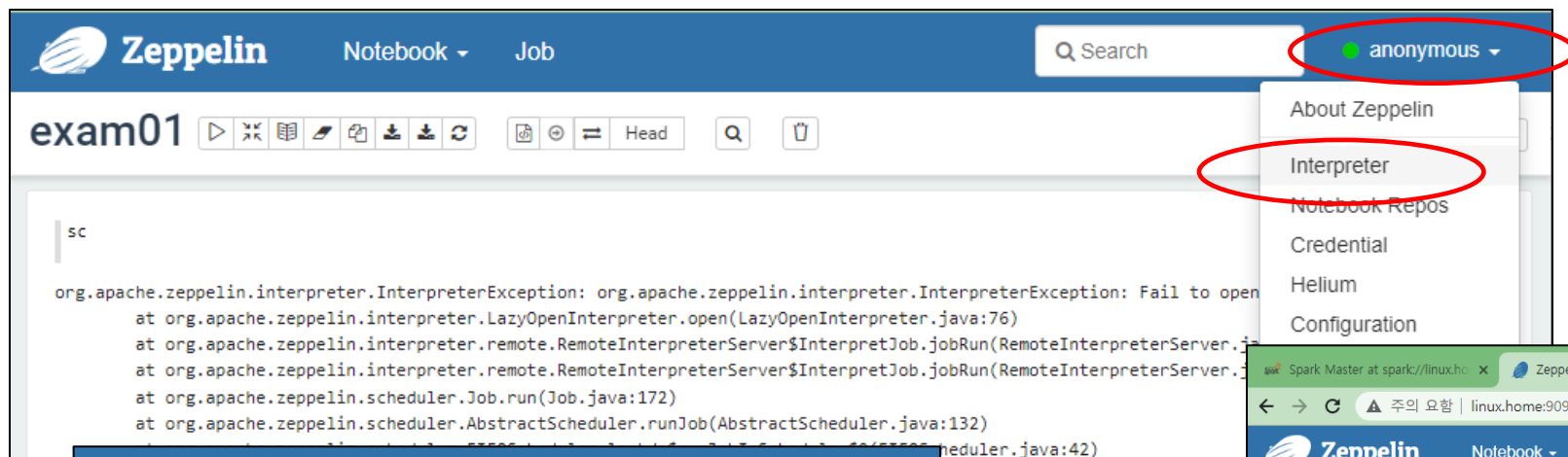
```

org.apache.zeppelin.interpreter.InterpreterException: org.apache.zeppelin.interpreter.InterpreterException: Fail to open SparkInterpreter
    at org.apache.zeppelin.interpreter.LazyOpenInterpreter.open(LazyOpenInterpreter.java:76)
    at org.apache.zeppelin.interpreter.remote.RemoteInterpreterServer$InterpretJob.jobRun(RemoteInterpreterServer.java:844)
    at org.apache.zeppelin.interpreter.remote.RemoteInterpreterServer$InterpretJob.jobRun(RemoteInterpreterServer.java:752)
    at org.apache.zeppelin.scheduler.Job.run(Job.java:172)
    at org.apache.zeppelin.scheduler.AbstractScheduler.runJob(AbstractScheduler.java:132)
    at org.apache.zeppelin.scheduler.FIFOScheduler.lambda$runJobInScheduler$0(FIFOScheduler.java:42)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:750)
Caused by: org.apache.zeppelin.interpreter.InterpreterException: Fail to open SparkInterpreter
    at org.apache.zeppelin.spark.SparkInterpreter.open(SparkInterpreter.java:138)
    at org.apache.zeppelin.interpreter.LazyOpenInterpreter.open(LazyOpenInterpreter.java:70)
    ... 8 more
Caused by: java.lang.Exception: This is not officially supported spark version: 3.3.1
You can set zeppelin.spark.enableSupportedVersionCheck to false if you really want to try this version of spark.
    at org.apache.zeppelin.spark.SparkInterpreter.open(SparkInterpreter.java:128)
    ... 1 more
        
```

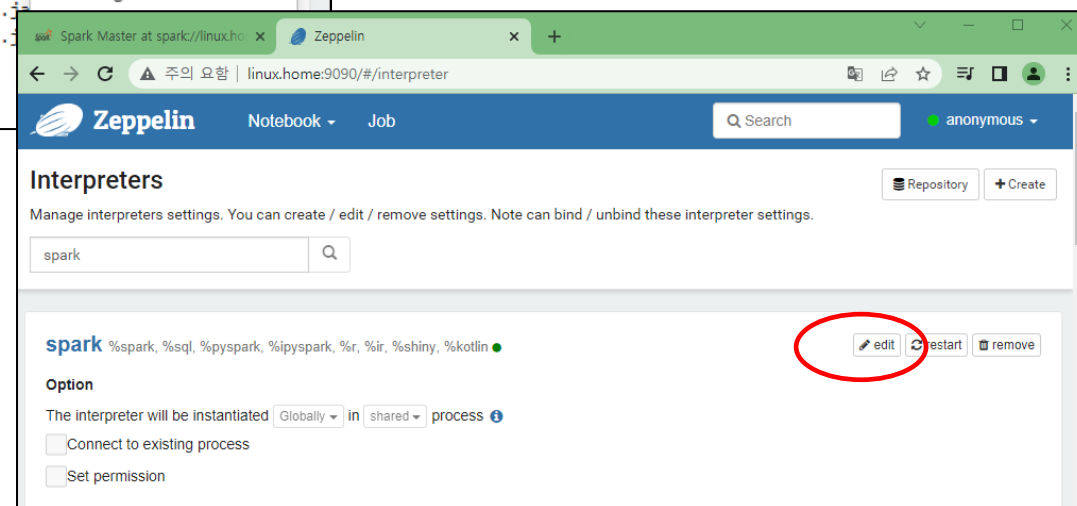
Took 41 sec. Last updated by anonymous at November 21 2022, 1:40:24 PM.

- sc (Spark Context) 를 입력, Shift + Enter 를 누르고 잠시 기다림
- 우측 메시지에 PENDDDING -> RUNNING 후 결과 보임
- 오류가 나면 다음을 수행

1. Zeppelin설치



- spark 로 검색



- [edit] 클릭

1. Zeppelin설치

Properties	
Name	Value
SPARK_HOME	/util/spark
spark.master	spark://DESKTOP-28CEK7O.:7077
spark.submit.deployMode	client
spark.app.name	
spark.driver.cores	1
spark.driver.memory	1g
spark.executor.cores	1
spark.executor.memory	1g
spark.executor.instances	3

본인의 Spark
마스터 주소

/util/spark

spark://DESKTOP-28CEK7O.:7077

client

3

1. Zeppelin설치

zeppelin.spark.useHiveContext	<input checked="" type="checkbox"/>
zeppelin.spark.run.asLoginUser	<input checked="" type="checkbox"/>
zeppelin.spark.printREPLOutput	<input checked="" type="checkbox"/>
zeppelin.spark.maxResult	1000
zeppelin.spark.enableSupportedVersionCheck	<input type="checkbox"/>
zeppelin.spark.uiWebUrl	

체크를 해제함...
버전 체크를 계속 하지 않도록 함

1. Zeppelin설치

zeppelin.spark.sql.interpolation	<input type="checkbox"/>
PYSPARK_PYTHON	/usr/bin/python3
PYSPARK_DRIVER_PYTHON	/usr/bin/python3
zeppelin.pyspark.usePython	<input checked="" type="checkbox"/>
zeppelin.R.knitr	<input checked="" type="checkbox"/>
zeppelin.R.cmd	R
zeppelin.R.image.width	100%

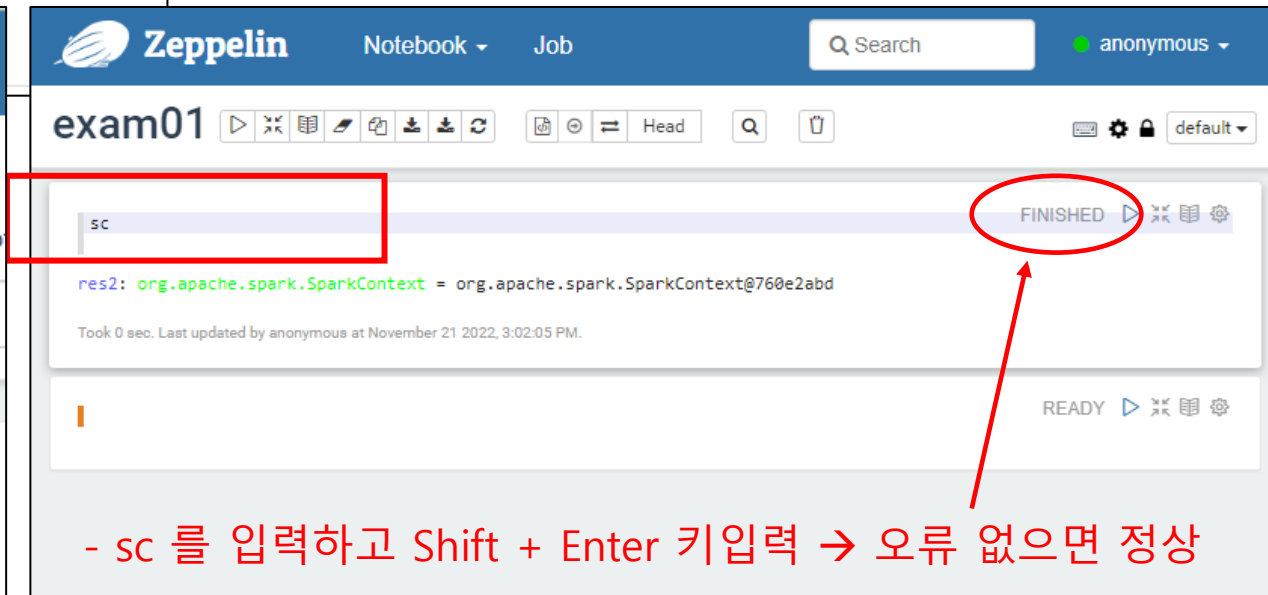
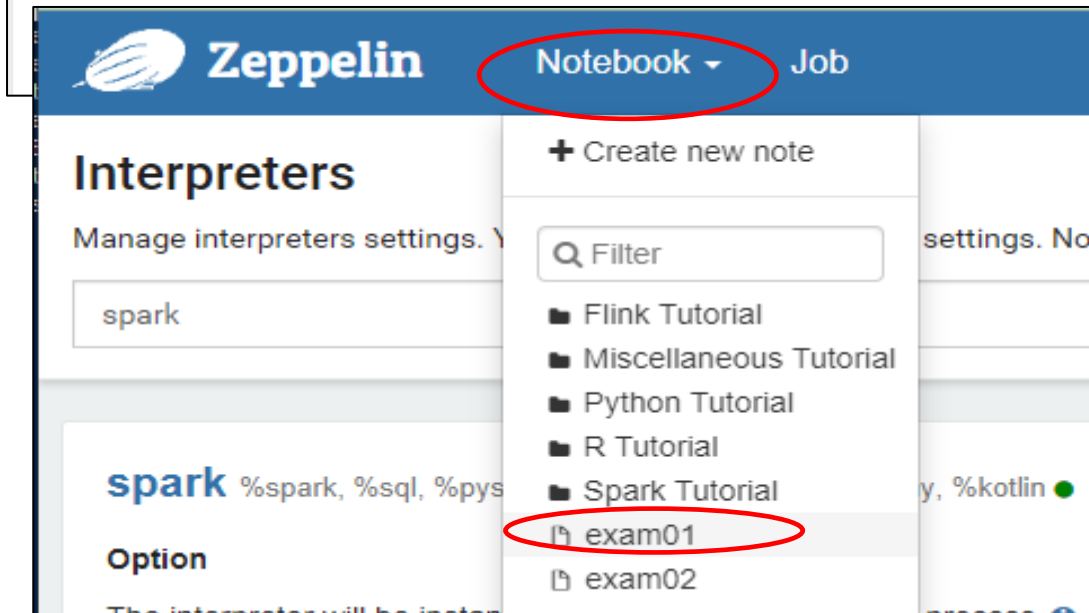
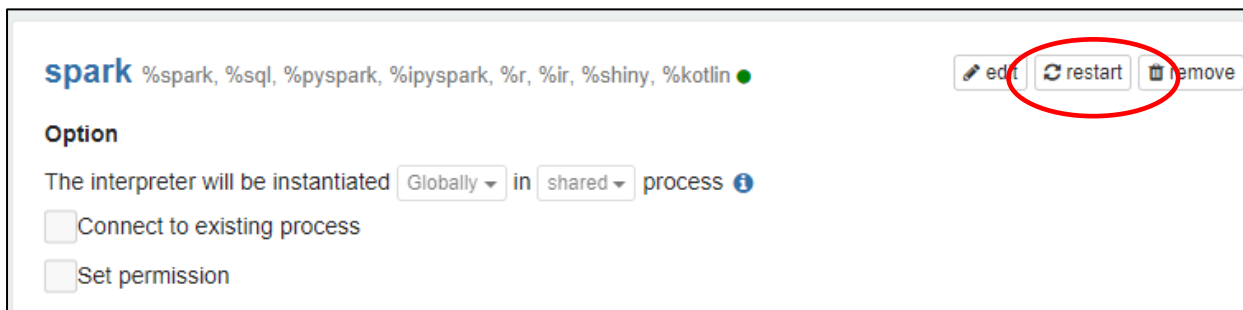
/usr/bin/python3

/usr/bin/python3

Artifact

1. Zeppelin설치

Spark 모듈을 리스타트함



- sc 를 입력하고 Shift + Enter 키입력 → 오류 없으면 정상

1. Zeppelin설치

- 우분투로 돌아 와서 git-hub 을 설치

```
[root@linux ~]#
[root@linux ~]# cd /util
[root@linux util]#
[root@linux util]# yum install git
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.kakao.com
* extras: mirror.kakao.com
* updates: mirror.kakao.com
```

cd /util
yum install git

```
[root@linux util]#
[root@linux util]# git clone https://github.com/sEOKiLL-jEONG/mydata.git data
Cloning into 'data'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
[root@linux util]#
```

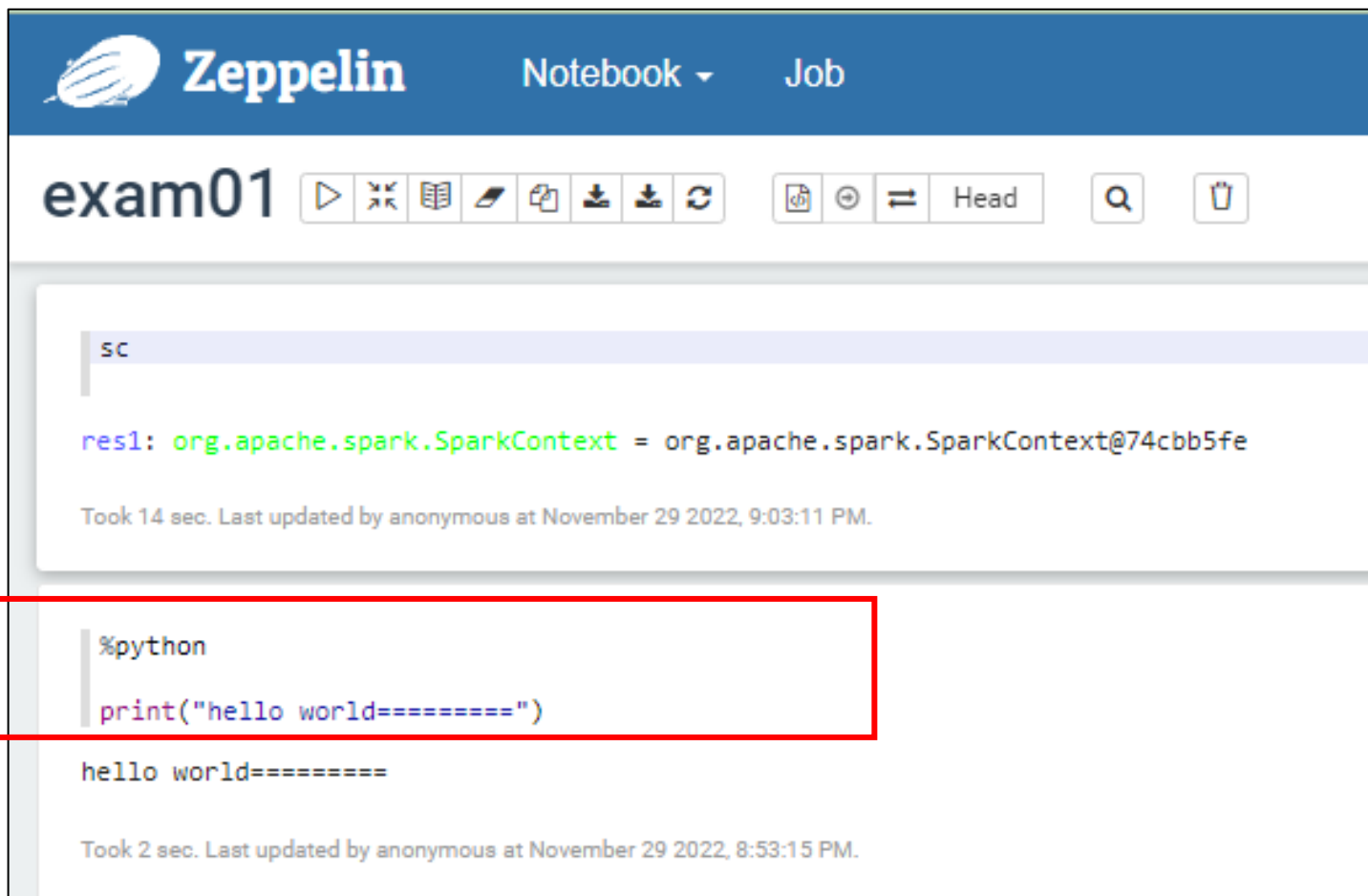
yum 이 없으면,
yum4 install git

```
[root@linux util]#
[root@linux util]# ls -al data/mydata
ls: cannot access 'data/mydata': No such file or directory
[root@linux util]#
```

ls -al data/mydata

git clone https://github.com/sEOKiLL-jEONG/mydata.git data

1. Zeppelin설치



The screenshot shows the Zeppelin Notebook interface. The top navigation bar includes the Zeppelin logo, 'Notebook', and 'Job' tabs. Below the navigation bar, the notebook title 'exam01' is displayed along with various action icons. The main content area shows two code blocks. The first block is a Scala (SC) code snippet that initializes a SparkContext. The second block is a Python code snippet that prints 'hello world' and is highlighted with a red box.

```
SC
res1: org.apache.spark.SparkContext = org.apache.spark.SparkContext@74cbb5fe

Took 14 sec. Last updated by anonymous at November 29 2022, 9:03:11 PM.
```

```
%python
print("hello world=====")

hello world=====

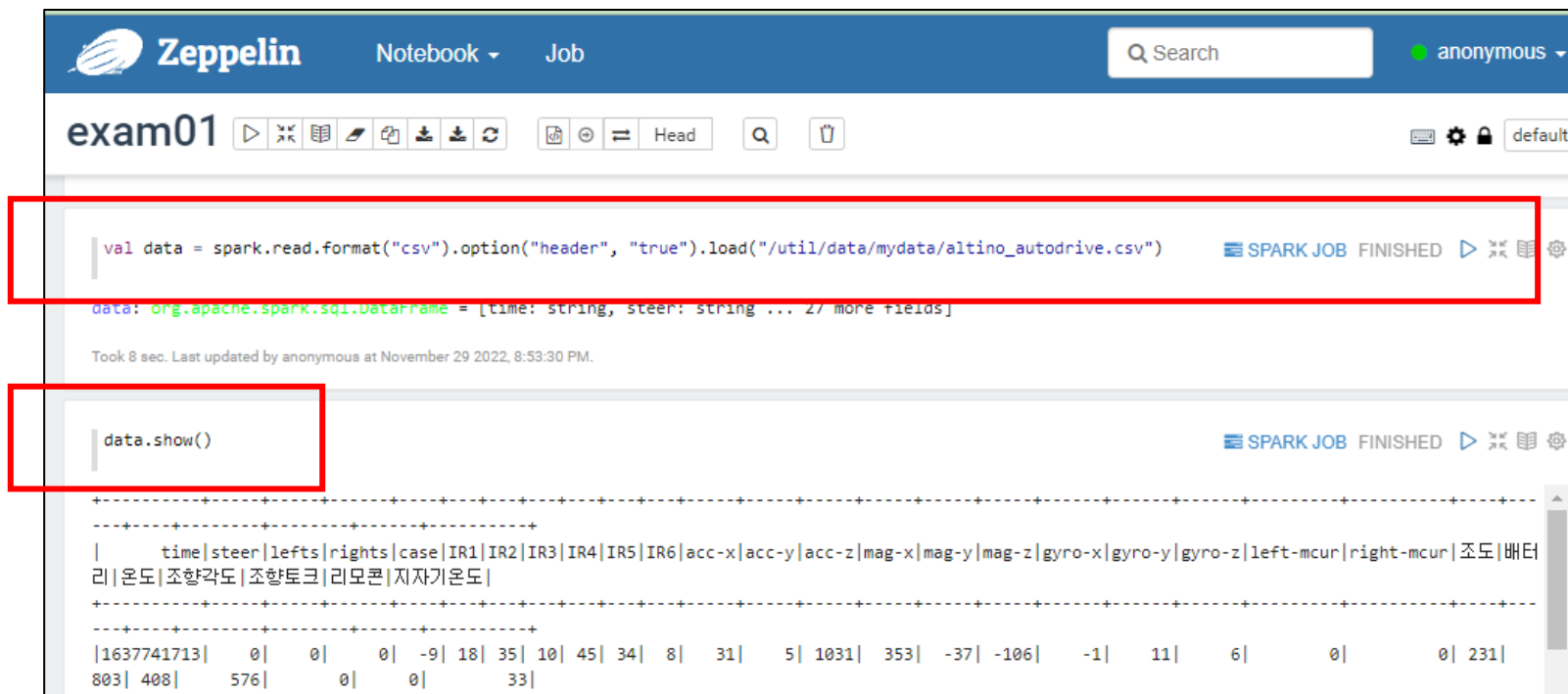
Took 2 sec. Last updated by anonymous at November 29 2022, 8:53:15 PM.
```

%python

print("hello world=====")

- 입력하고 Shift + EnterKey누름

1. Zeppelin설치



The screenshot shows the Zeppelin Notebook interface. The top bar includes the Zeppelin logo, 'Notebook' and 'Job' tabs, a search bar, and a user profile dropdown showing 'anonymous'. The notebook title is 'exam01'. Below the title bar, there are two code blocks, each highlighted with a red rectangle. The first code block contains the following Spark SQL command:

```
val data = spark.read.format("csv").option("header", "true").load("/util/data/mydata/altino_autodrive.csv")
```

The second code block contains the following command:

```
data.show()
```

Below the code blocks, the output of the first command is displayed as a table. The table has 20 columns: time, steer, lefts, rights, case, IR1, IR2, IR3, IR4, IR5, IR6, acc-x, acc-y, acc-z, mag-x, mag-y, mag-z, gyro-x, gyro-y, gyro-z, left-mcur, right-mcur, 조도, and 배터리. The first row of data is:

time	steer	lefts	rights	case	IR1	IR2	IR3	IR4	IR5	IR6	acc-x	acc-y	acc-z	mag-x	mag-y	mag-z	gyro-x	gyro-y	gyro-z	left-mcur	right-mcur	조도	배터리
1637741713	0	0	0	-9	18	35	10	45	34	8	31	5	1031	353	-37	-106	-1	11	6	0	0	231	
803	408	576	0	0		33																	

```
val data = spark.read.format("csv").option("header", "true").load("/util/data/mydata/altino_autodrive.csv")

data.show()
```

1. Zeppelin설치

6

zeppelin 구동 shell 만들어 두기

```
[root@linux bin]#
[root@linux bin]# cd /util/zeppelin/bin
[root@linux bin]#
[root@linux bin]# pwd
/util/zeppelin/bin
[root@linux bin]#
[root@linux bin]# ls
common.cmd      functions.sh      interpreter.sh    zeppelin-daemon.sh    zeppelin.sh
common.sh        install-interpreter.sh  stop-interpreter.sh  zeppelin-systemd-service.sh
functions.cmd    interpreter.cmd    upgrade-note.sh    zeppelin.cmd
```

cd /util/zeppelin/bin

vi run.sh

```
> root@linux:/util/zeppelin/bin
```

```
# auto run
```

```
echo ===== zeppelin start =====
```

```
cd /util/zeppelin/bin
```

```
./zeppelin-daemon.sh start
```

```
# auto run
```

```
echo ===== zeppelin start =====
```

```
cd /util/zeppelin/bin
```

```
./zeppelin-daemon.sh start
```

1. Zeppelin설치

```
[root@linux bin]#  
[root@linux bin]# cp run.sh /root/run-zepp.sh  
[root@linux bin]#
```

cp run.sh /root/run-zepp.sh

chmod 755 /root/run-zepp.sh

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

* gedit 활용

1) x-window (xming) 설치-pc

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

2) gedit 설치

```
[root@linux ~]#
[root@linux ~]# sudo apt install gedit -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bubblewrap dbus-user-session docbook-xml gedit-common gir1.2-gtk-3.0 gir1.2-gtksource-4 gir1.2-harfbuzz-0.0 gir1.2-glib-2.0
  glib-networking-common glib-networking-services gstreamer1.0-x libaa1 libasynclns0 libavc1394-0 libcac0
```

sudo apt install gedit -y

오류가 발생하면,
아래의 명령 수행 후, 재시도

sudo apt-get update

3) .bashrc 에 내용 추가

vi ~/.bashrc

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

4) 수정을 원하는 파일이름 입력

```
[root@linux ~]#
[root@linux ~]# gedit .bashrc

(gedit:3526): dconf-WARNING **: 07:26:22.629: failed to commit changes
ctory
```

gedit .bashrc

* kiro 활용

사용법 : <https://github.com/rhysd/kiro-editor>

1) 설치

```
sudo apt install cargo -y
```

```
cargo install kiro-editor
```

오류가 발생하면,
아래의 명령 수행 후, 재시도

```
sudo apt-get update
```

2) 수정을 원하는 파일이름 입력

반드시 roor 로 로그인

```
cd ~
```

```
vi .bashrc
```

맨 아래 부분에 추가

```
export PATH=$PATH:/root/.cargo/bin
```

저장하고,

```
source .bashrc
```

3) 수정을 원하는 파일이름 입력

```
kiro test.txt
```

간단사용법

저장 : ctrl + s

탈출 : ctrl + q 두 번

* Linux용 Windows 하위 시스템 Linux GUI 앱 실행

<https://learn.microsoft.com/ko-kr/windows/wsl/tutorials/gui-apps>