

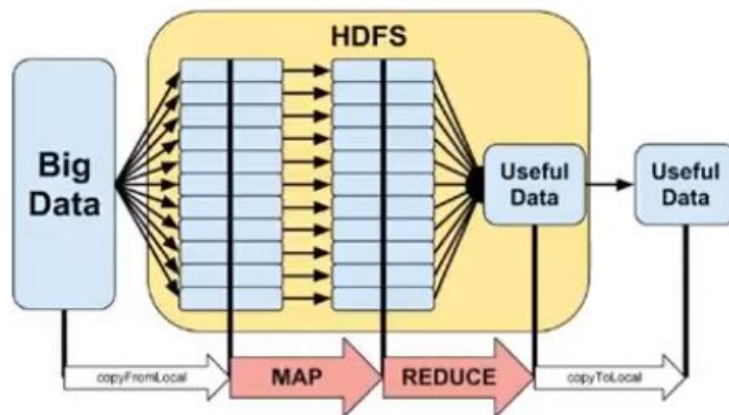
## 하둡이란?

- 1.대용량 데이터를 분산처리 할 수 있는 자바 기반의 오픈소스 Framework
- 2.더그 커팅이 구글의 논문(2003년, "The Google File system", 2004년, "MapReduce: Simplified Data Processing on Large Cluster")을 참조하여 구현함
- 3.더그 커팅의 아들이 노란 코끼리 장난감 인형을 hadoop이라고 부르는 것을 듣고 명명함
- 4.공식사이트: <http://hadoop.apache.org>

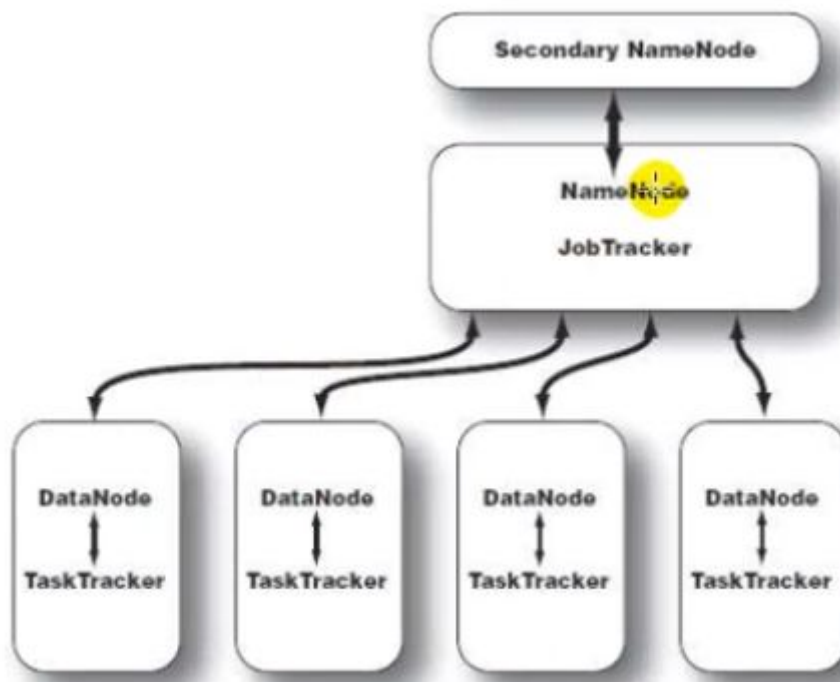
## Hadoop 관련 용어 정리

### 1)HDFS(Hadoop Distributed File System)

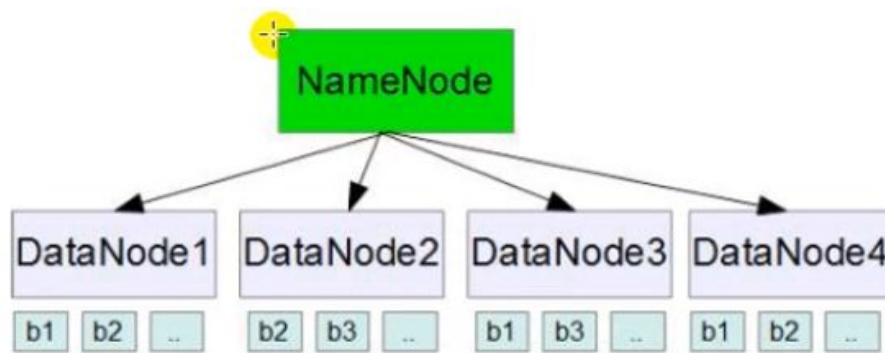
대용량 파일을 분산된 서버에 설치하고 많은 클라이언트가 저장된 데이터를 빠르게 처리할 수 있게 설계된 파일 시스템



큰 데이터가 있을 때, HDFS에 넣어주게되면 MAP과정을 통해 분산 처리를 하고, Reduce과정을 통해 취합한다.



## 2)NameNode:가장 중요한(main)역할을 하는 노드



HDFS의 모든 메타데이터를 관리하고 클라이언트가 HDFS에 저장된 파일에 접근할 수 있도록 처리하는 노드

## 3)Secondary NameNode : NameNode의 보조 노드

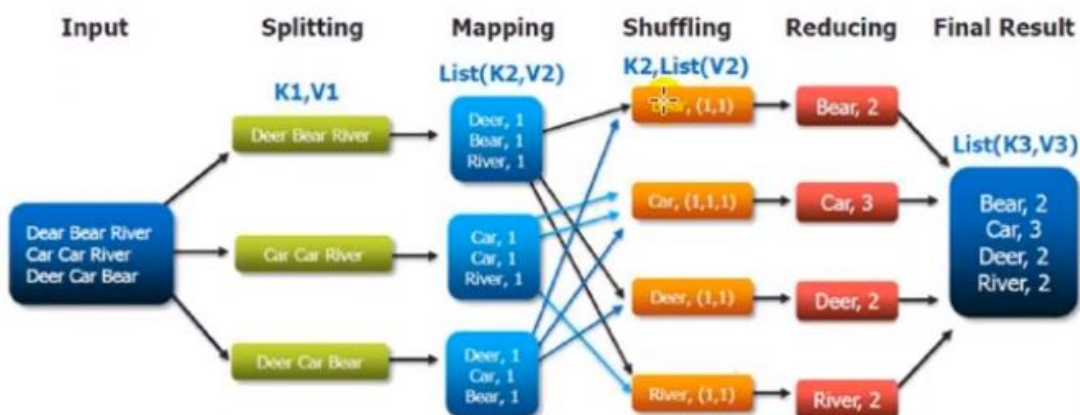
주기적으로 네임노드의 파일 시스템 이미지 파일을 갱신하는 역할을 수행하는 노드

## 4)DataNode:실제로 데이터를 분산해서 처리하는 노드

HDFS에 데이터를 입력하면 입력 데이터는 32MB의 블록으로 나뉘어져서 여러대의 데이터노드에 분산되어 저장된다.

## 5)MapReduce

map과 reduce라는 두 개의 method로 구성됨. 대규모 분산 컴퓨팅 혹은 단일 컴퓨팅 환경에서 대량의 데이터를 병렬로 분석할 수 있는 알고리즘



## 6)JobTracker

하둡 클러스터에 등록된 전체 job의 스케줄링을 관리하고 모니터링하는 노드  
전체 하둡 클러스터에서 하나의 JobTracker가 실행됨. 보통 하둡 클러스터의 네임 노드(마스터)에서 실행됨

## 7)TaskTracker

JobTracker의 작업을 요청받고 JobTracker가 요청한 맵과 리듀스 개수만큼 Map Task와 Reduce Task를 생성함. 하둡 클러스터의 데이터노드에서 실행됨.

## 8)Mapper

맵리듀스 프로그래밍 모델에서 map method의 역할을 수행하는 클래스. 키와 값으로 구성된 입력 데이터를 전달받아 이 데이터를 가공하고 분류해서 새로운 데이터를 생성함

## 9)Reducer

맵리듀스 프로그래밍 모델에서 reduce method의 역할을 수행하는 클래스. map task의 출력 데이터를 입력 데이터로 전달받아 집계 연산을 수행

## 10)YARN(Yet Another Resource Negotiator)

맵리듀스의 차세대 기술, 맵리듀스의 확장성과 속도 문제를 해소하기 위해 개발된 프로젝트

## 11)SSH(Secure Shell)

: NameNode는 하나지만, DataNode를 여러개이다. 따라서 노드간 안전한 통신이 되어야한다. 여러대의 PC들이 안전하게 데이터를 주고받기 위해서는 접속을 해야한다.

접속하려면, Login을 해야하고, 통신간 보안이 유지되어야 하기 때문에, SSH를 이용한다.

네트워크상의 다른 컴퓨터에 로그인하거나 원격 시스템에서 명령을 실행하고 다른 시스템으로 파일을 복사할 수 있게 해주는 응용 프로토콜, 기존의 telnet을 대체하기 위해 설계되었으며 암호화 기법을 사용하여 강력한 인증 방법 및 안전하지 못한 네트워크에서 안전하게 통신할 수 있는 기능을 제공함, 기본적으로 22번 포트를 사용함.

하둡에서는 SSH 프로토콜을 이용하여 하둡 클러스터 간의 내부 통신을 수행한다. 이 때 SSH를 이용할 수 없다면 하둡을 실행할 수 없게 된다. 따라서 네임노드에서 SSH 공개키를 설정하고 이 공개키를 전체 서버에 복사하는 작업을 진행한다.

## 12)NoSQL(Not Only SQL)

관계형 데이터 모델과 SQL 문을 사용하지 않는 데이터베이스 시스템 혹은 데이터 저장소. 기존 RDBMS가 분산 환경에 적합하지 않기 때문에 이를 극복하기 위해 고안됨. row 단위가 아닌 집합 형태로 저장됨. 또한 Sharding(샤딩)이라는 기능이 있어서 데이터를 여러 서버에 분산하여 저장함. 기존 RDBMS처럼 완벽한 데이터 무결성을 제공하지는 않음. 기업의 핵심 데이터는 RDBMS를 이용하고 핵심은 아니지만 데이터를 보관하고 처리를 해야 하는 경우 NoSQL 이용.

MonaoDB, HBase 등 다양한 솔루션이 있음.

- 하둡 설치

## 2. 하둡 설치



### 가. VMPlayer 설치

<http://vmware.com>

VMPlayer에는 Hadoop Multi Cluster 구축 실습에 필요한 네트워크 설정 기능이 없으므로 VMware workstation Pro의 30일 평가판을 다운로드받아 실습

VMware workstation Pro(정식버전)을 설치하면 기존에 설치한 VMPlayer는 삭제됨 (VMPlayer가 포함되어 있음)

멀티 클러스터를 구축해야 하므로 에러가 나거나 가상머신을 복제하여 작업해야 할 경우가 생기므로 Centos를 설치한 후 백업본을 만들어둘 것

VM player만이 아니라 VMware workstation Pro를 받아야한다. 왜냐하면, NameNode 1대 DataNode3대 해서 4개의 서버를 동시운용하려고 한다. 이때분산처리 시스템이기 때문에, 서버를 여러개 돌려야한다.  
그래서 30일 평가판으로 진행을 한다.



1) vmware : <http://vmware.com>

가) VMware Workstation Pro : 유료

<https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>

나) VMware Player :

상업적인 목적이 아닌 경우 무료

[https://my.vmware.com/en/web/vmware/free#desktop\\_end\\_user\\_computing/vmware\\_workstation\\_player/15\\_0](https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/15_0)

2) Virtual Box : 오라클에서 개발, 상업적인 목적이 아닌 경우 무료

<http://virtualbox.org>

vmware®

VMware Cloud

제품

지원

솔루션

전문 서비스

다운로드

파트너

회사

다운로드

📁 제품 다운로드

📁 무료 제품 평가판 및 데모

📁 무료 제품 다운로드

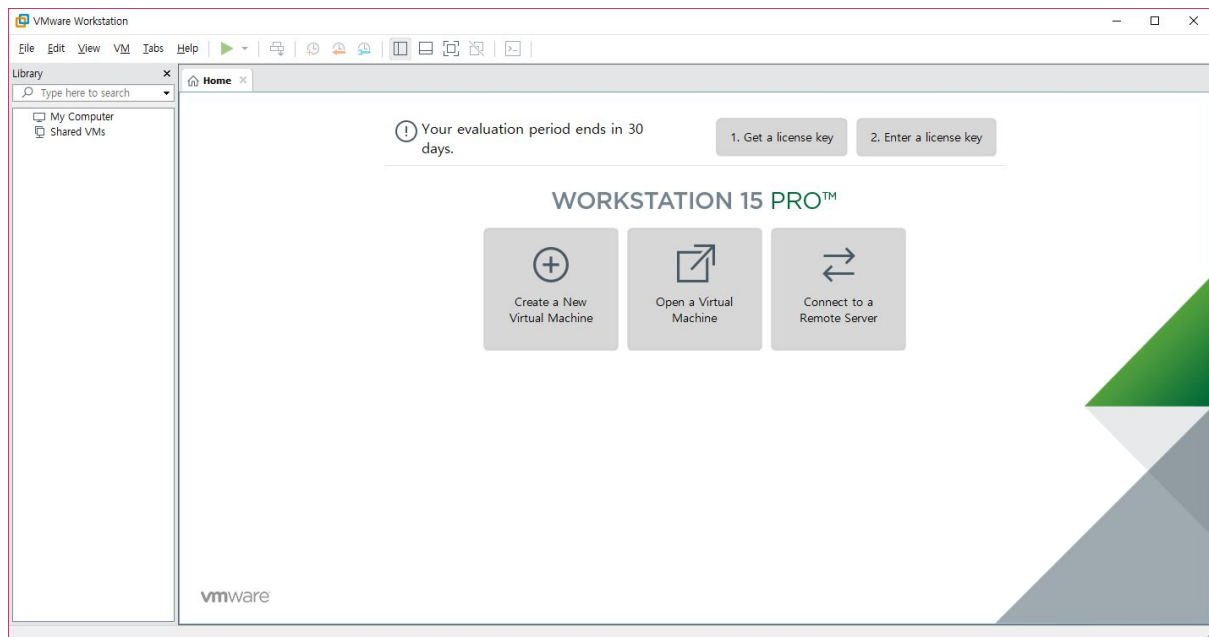
무료 제품 평가판 및 데모

vCloud Suite ↗

Virtual SAN ↗

Horizon 7

Workstation Pro





New Virtual Machine Wizard

**Specify Disk Capacity**  
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB):

Recommended size for CentOS 7 64-bit: 20 GB

☒ Store virtual disk as a single file

☐ Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Help < Back Next > Cancel

## 2) Multi Cluster 구성 계획

CPU i5, RAM 16GB 기준

master 1대 - 메모리 1~2GB, HDD 40GB

slave 3대 - 메모리 1GB, HDD 40GB

CPU i5, RAM 8GB 기준

master 1대 - 메모리 512~1024GB, HDD 40GB

slave 3대 - 메모리 512MB, HDD 40GB

가상머신이 설치된 원본 디렉토리를 복사하여 만든다.

d:\Centos\backup - Centos 7 백업용 디렉토리(리눅스 설정에 에러가 있을 경우에 대비한 백업 디렉토리)

d:\Centos\Master - 네임노드

d:\Centos\Slave1 - 데이터노드1

d:\Centos\Slave2 - 데이터노드2

d:\Centos\Slave3 - 데이터노드3



## 메모리 수정

Device	Summary
Memory	512 MB
Processors	1
Hard Disk (SCSI)	40 GB
CD/DVD (IDE)	Auto detect

### Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

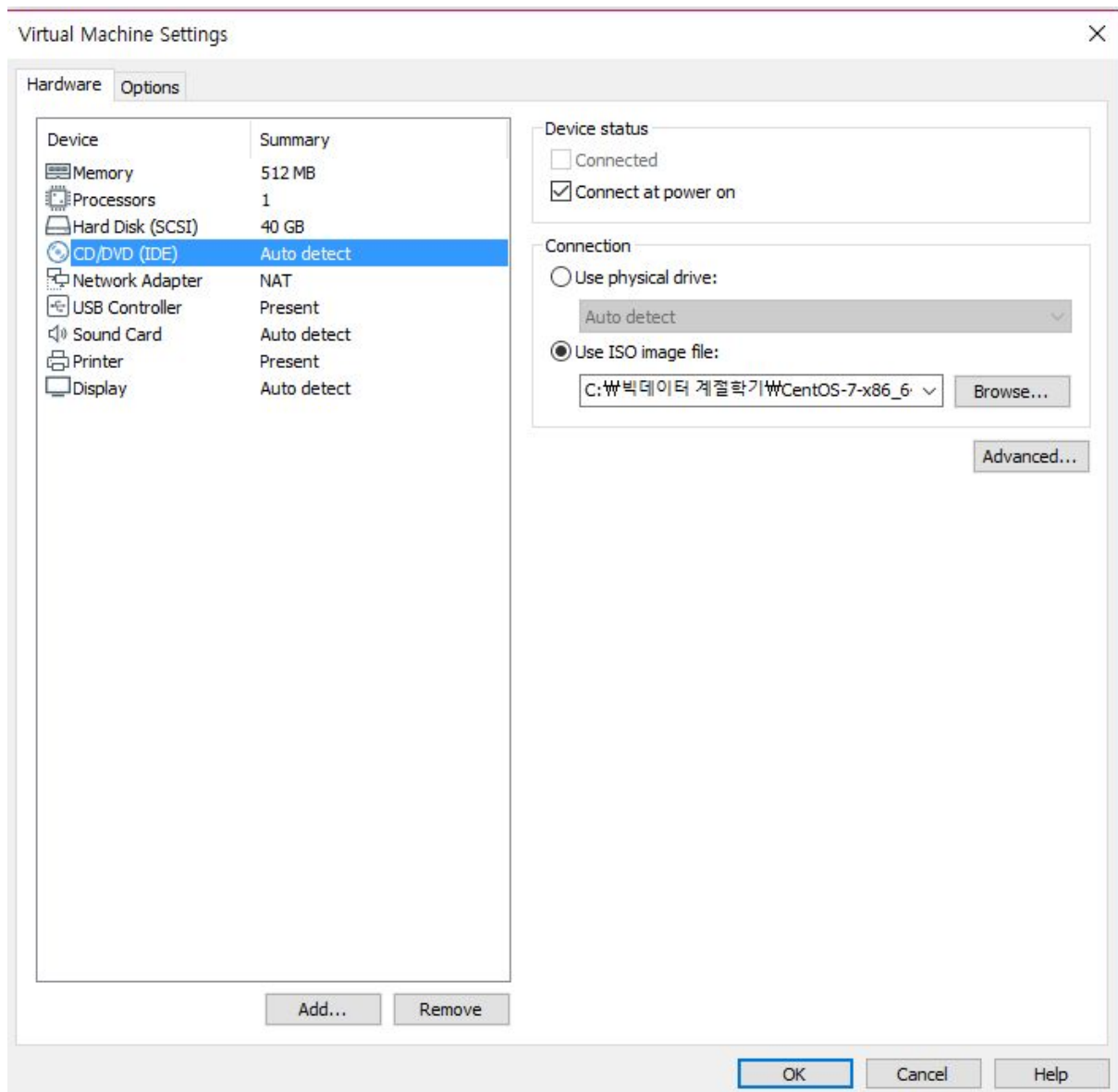
Memory for this virtual machine:  MB

## centos iso파일 다운

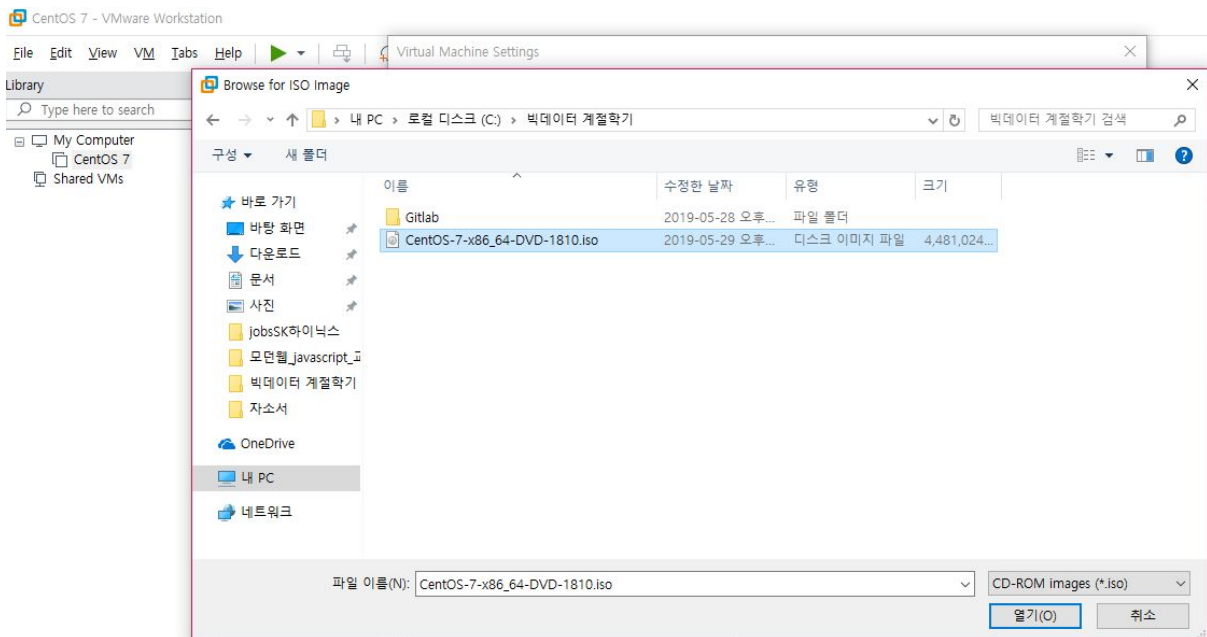
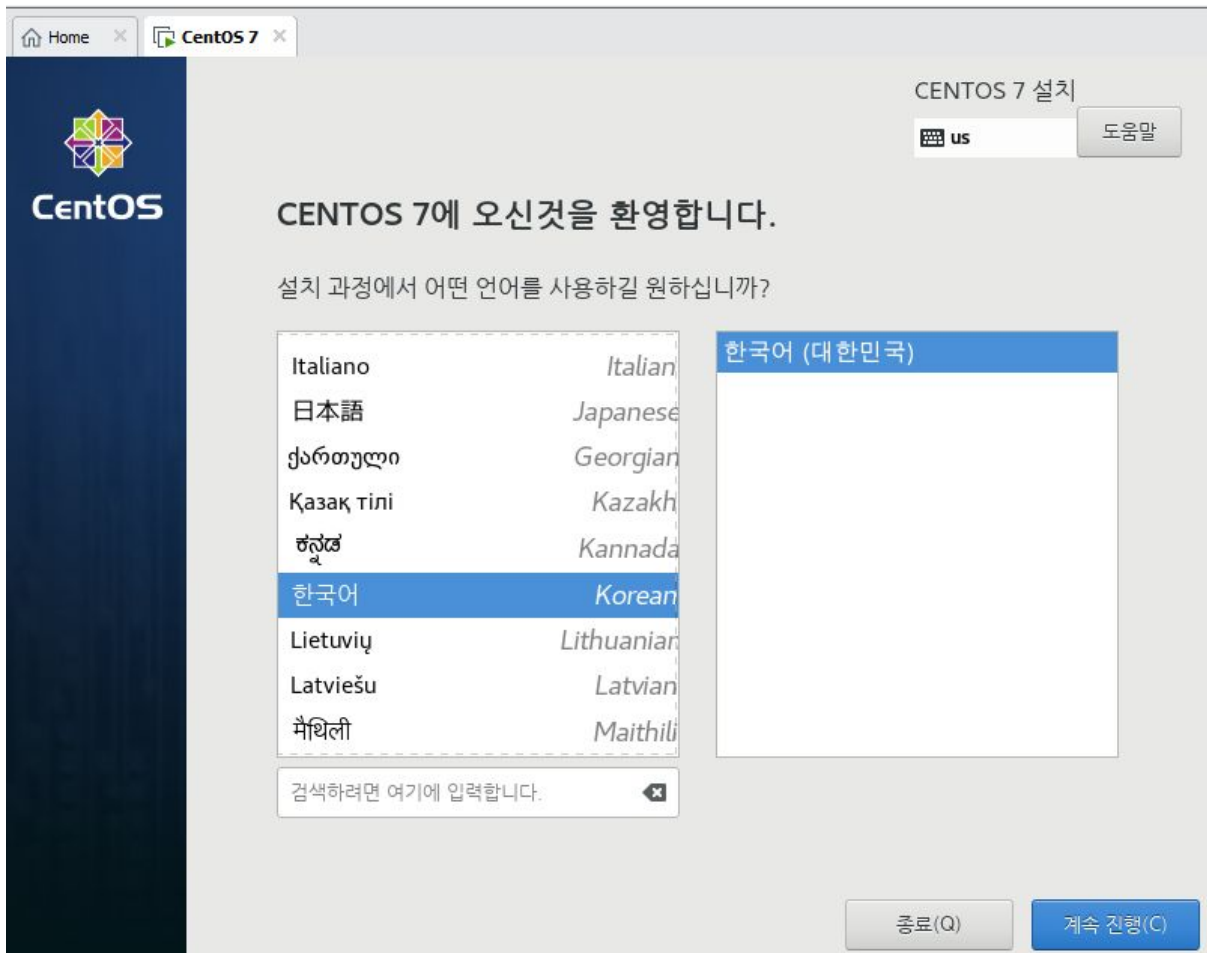
1. <https://wiki.centos.org/kr/Download>

2. [CentOS-7-x86\\_64-DVD-1810.iso](#)

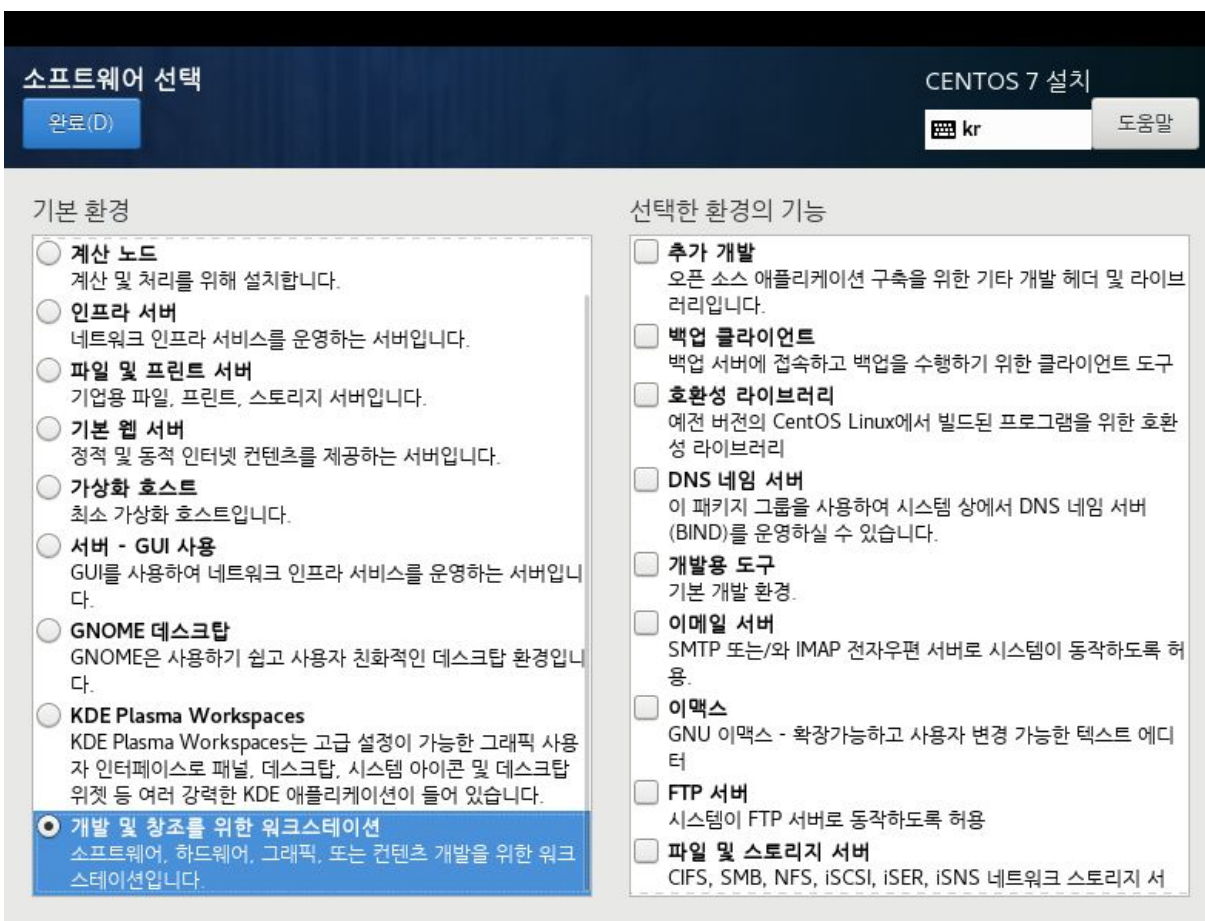
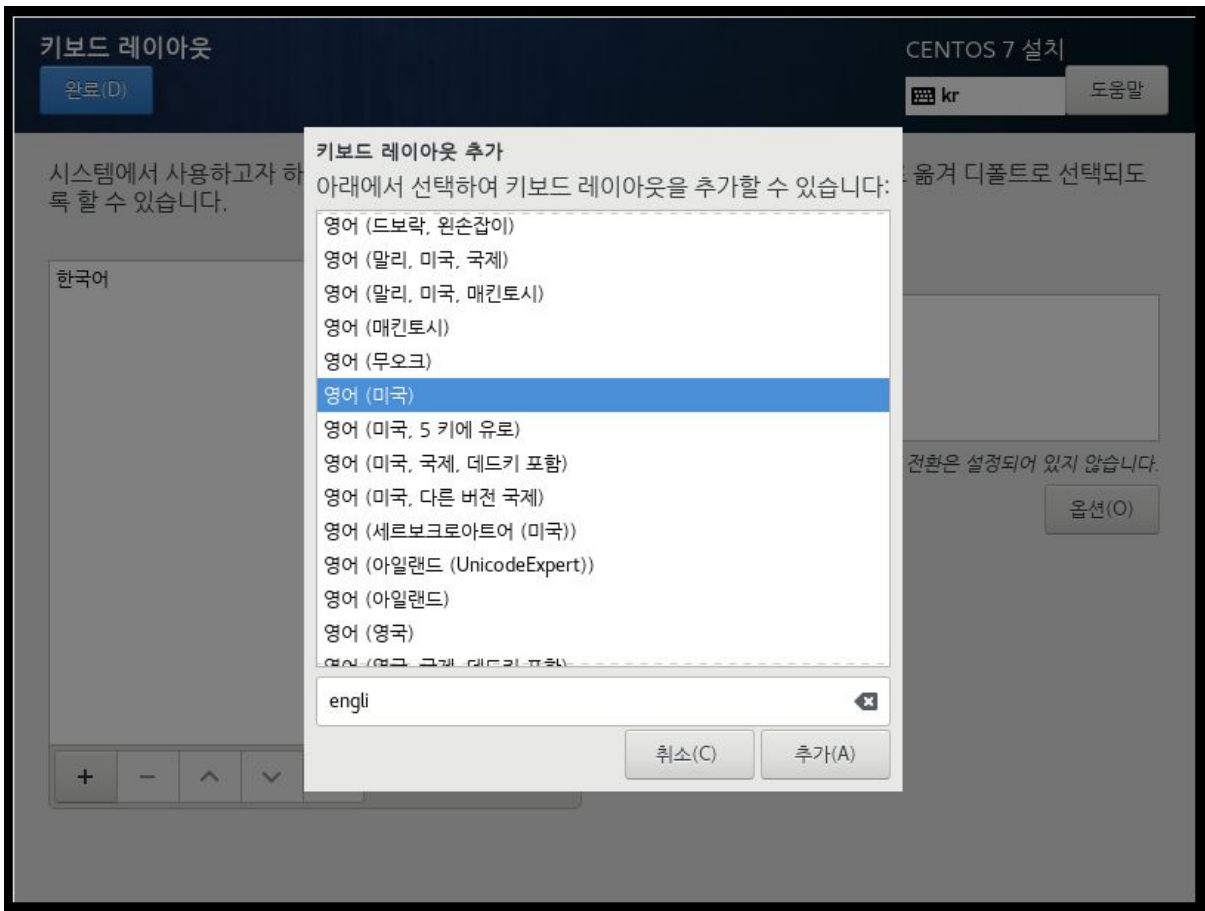
26-Nov-2018 08:55 4.3G

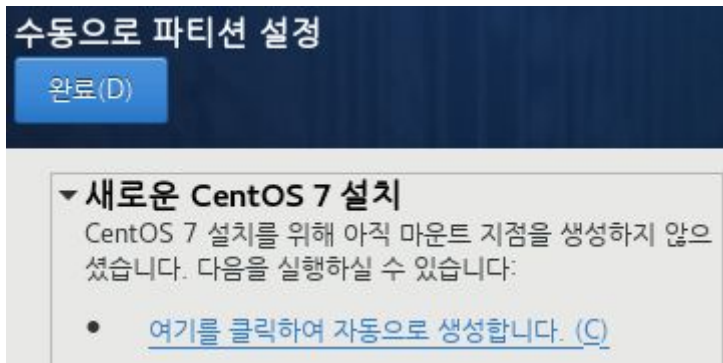


파워를 키고 os를 설치한다.



CD/DVD centos iso 파일 지정

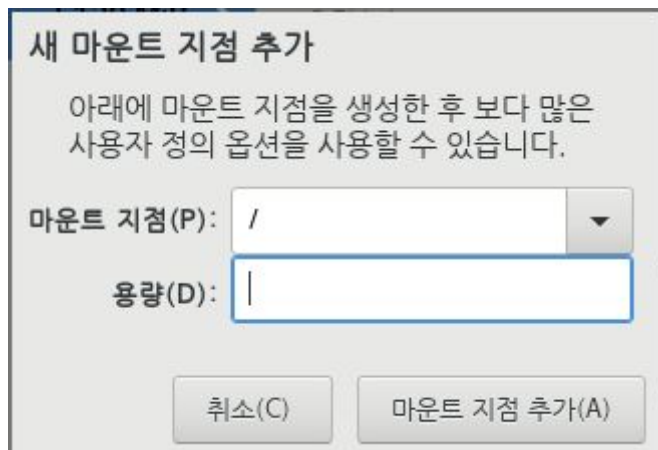




파티션 설정에서 자동생성 클릭



가상메모리 영역을 5g로 설정



남은 용량을 모두 root에 할당

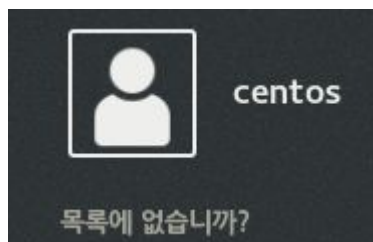
네트워크 컴



설치시작

암호 : 1234

사용자명 및 암호 : centos



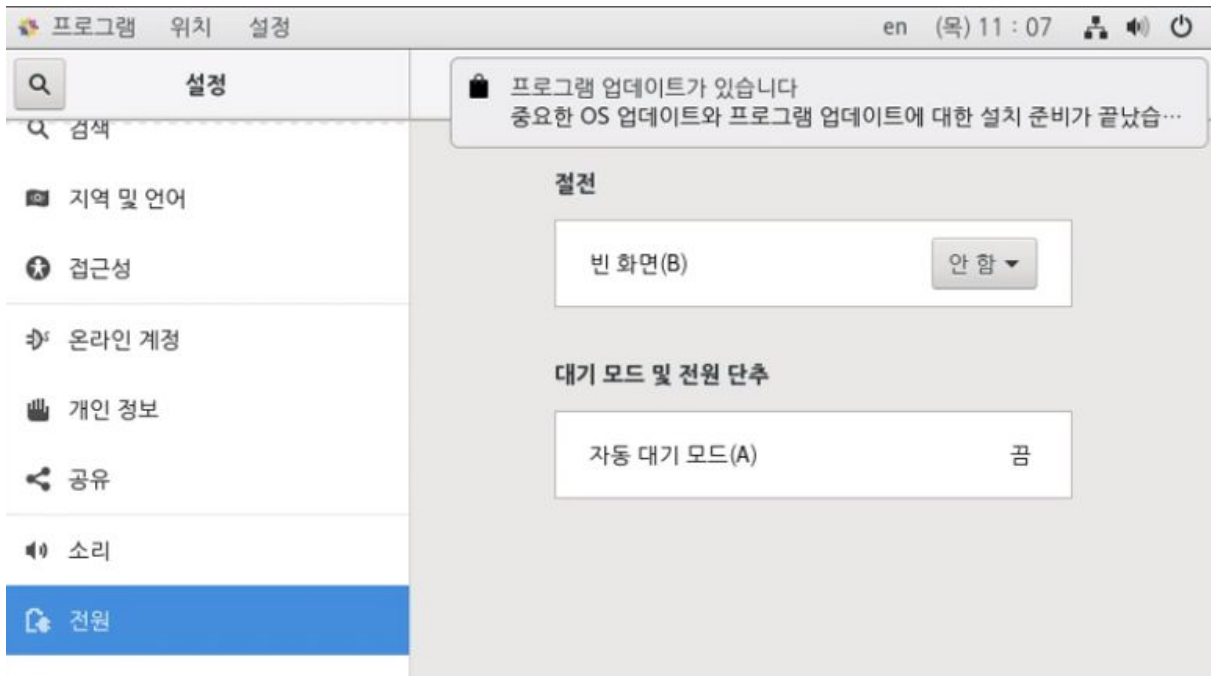
목록이 없습니까?클릭!

이름 : root

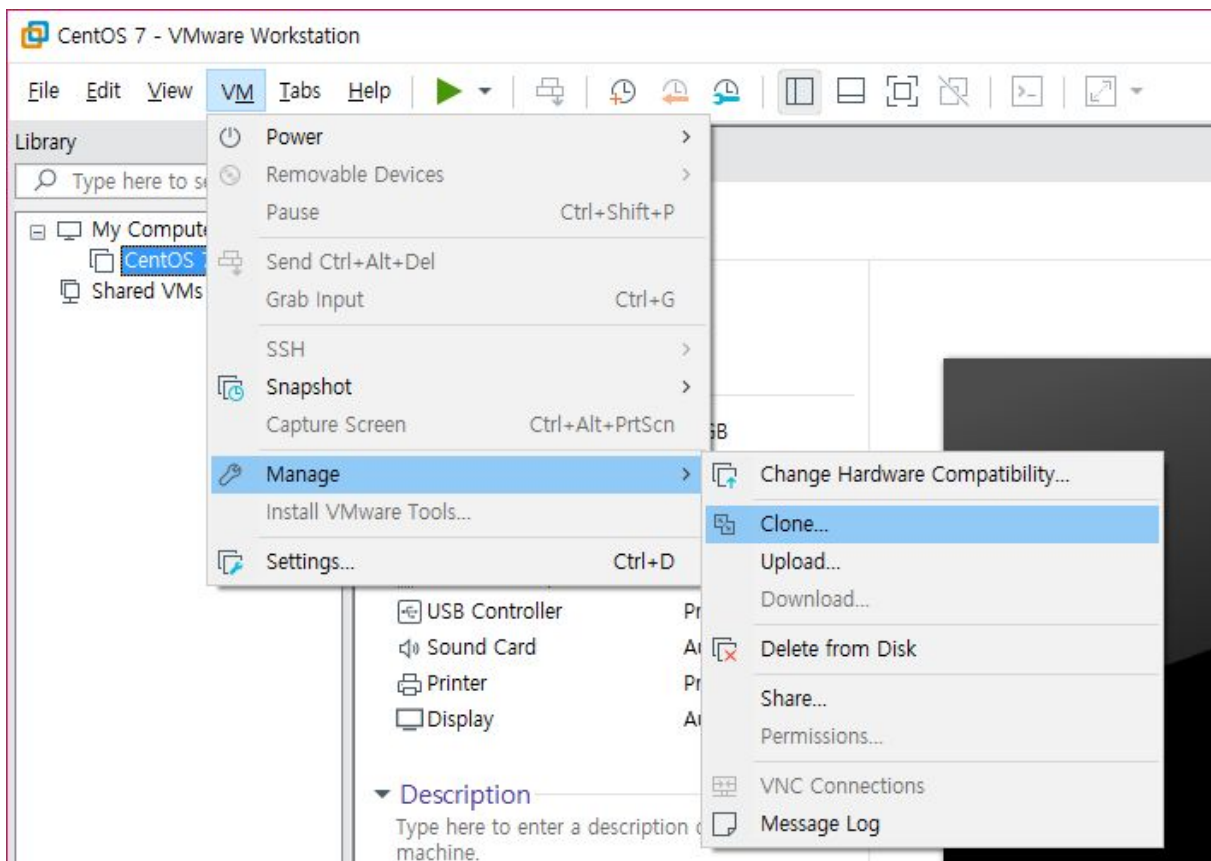
패스워드 : 1234

centos는 계속 사용하지 않으면 자동으로 잠기는 특성이 있기 때문에,  
시스템 도구-설정-개인-사생활-한면 잠금 끄  
을 설정하도록 하자.





추가적으로 편의를 위해 전원 빈화면 사용안함도 추가해 주도록하자



을 통해서 지금까지 설치하고 설정한 CentOS 7을 복제해서 여러대를 바로 만들어줄 수 있다.



위에 linked clone은 바로가기를 생성하는 것이기 때문에, 복제 가상머신을 만들때는 밑에 것을 설정하는 것이 옳다.

Master계정을 하나 Clone을 하고 하둡을 이제 설치할 것이다.

하지만 앞서서, 하둡은 Java기반이기 때문에, JDK를 설치해주어야하는데, 이는

<https://www.youtube.com/watch?v=yJoTmwwEx2Q>

다음 URL강의를 통해서 진행한다.

#### 1.버전 확인

```
Home x CentOS 7 x Master x
프로그램 위치 터미널

root@localhost:~

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

[root@localhost ~]# java -version
openjdk version "1.8.0_181"
OpenJDK Runtime Environment (build 1.8.0_181-b13)
OpenJDK 64-Bit Server VM (build 25.181-b13, mixed mode)
[root@localhost ~]#
```

현재는 8버전이지만, oracle에 최신버전은 12까지 나온상태이기 때문에, 업데이트 하는 방법을 알아 보도록하자.

- 1) rpm : 새로운 패키지를 설치하거나 업그레이드하는 자동 설치 프로그램, 윈도우의 **셋업 프로그램에** 해당됨

-q 질의옵션

-a 시스템에 설치된 모든 패키지의 목록

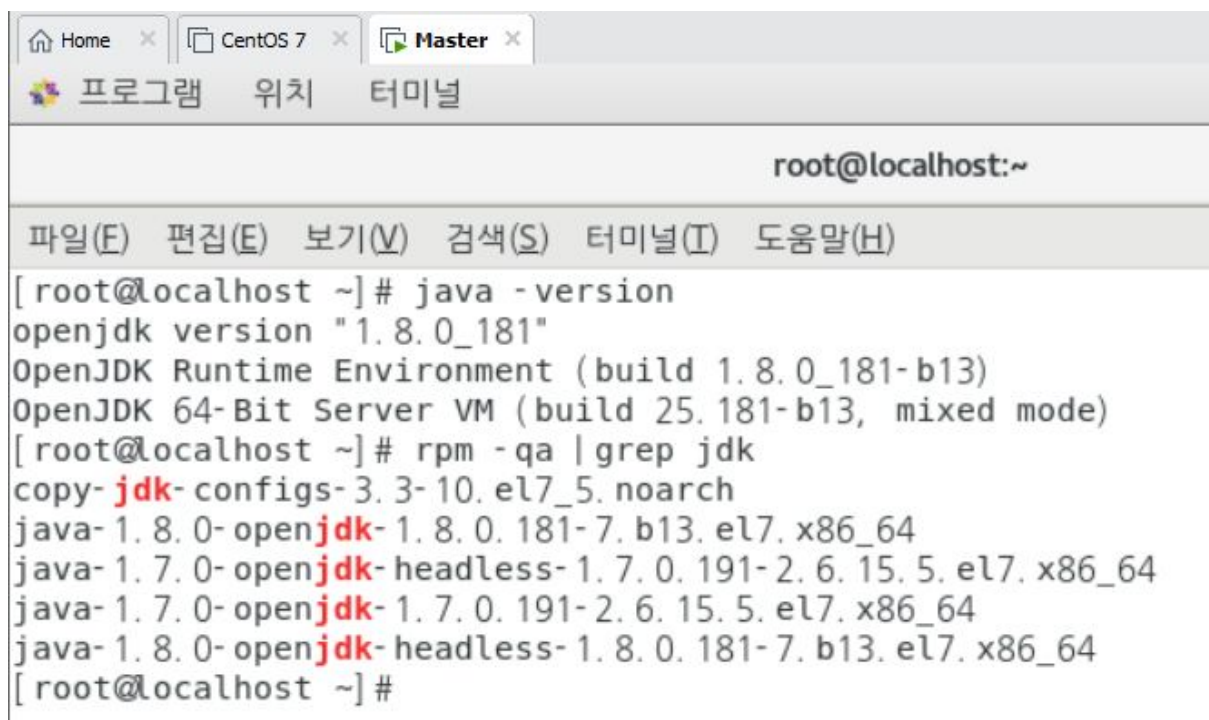
grep : 파일에서 특정한 패턴(문자열)을 찾는 명령어

```
rpm -qa | grep jdk
```

- 2) yum : 프로그램의 다운로드 및 설치, 삭제 등의 작업을 자동으로 해주는 프로그램

java로 시작하는 모든 프로그램 제거

```
yum remove java*
```



The screenshot shows a terminal window with the following content:

```
root@localhost:~  
파일(E) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[root@localhost ~]# java -version  
openjdk version "1.8.0_181"  
OpenJDK Runtime Environment (build 1.8.0_181-b13)  
OpenJDK 64-Bit Server VM (build 25.181-b13, mixed mode)  
[root@localhost ~]# rpm -qa | grep jdk  
copy-jdk-configs-3.3-10.el7_5.noarch  
java-1.8.0-openjdk-1.8.0.181-7.b13.el7.x86_64  
java-1.7.0-openjdk-headless-1.7.0.191-2.6.15.5.el7.x86_64  
java-1.7.0-openjdk-1.7.0.191-2.6.15.5.el7.x86_64  
java-1.8.0-openjdk-headless-1.8.0.181-7.b13.el7.x86_64  
[root@localhost ~]#
```

인데, 이 jdk를 삭제하고 다시 최신버전을 깔아보도록하자.

```
[root@localhost ~]# yum remove java*
```

지워진 것을 확인하자.

```
[root@localhost ~]# rpm -qa | grep jdk
copy-jdk-configs-3.3-10.el7_5.noarch
[root@localhost ~]#
```

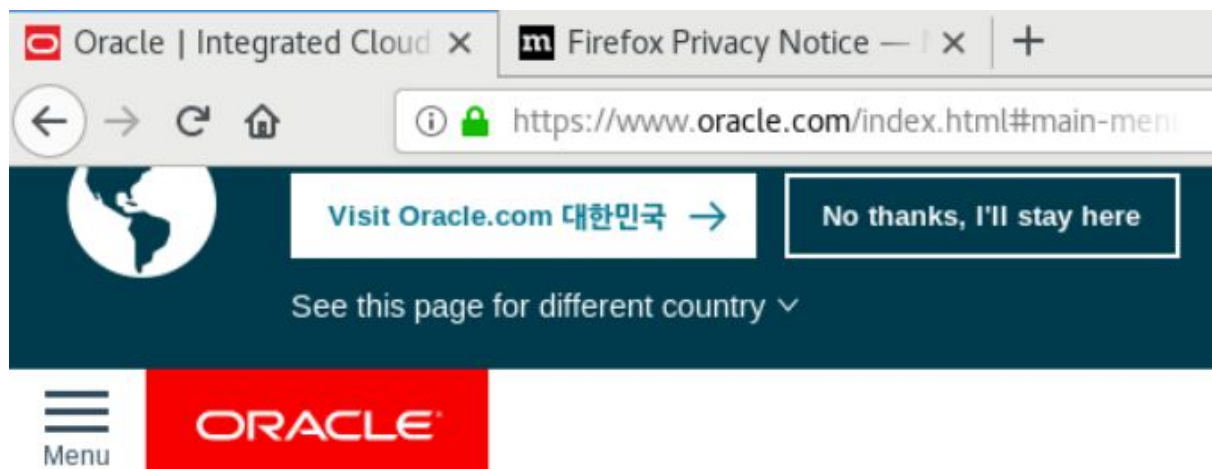
## 다. jdk 다운로드 및 설치

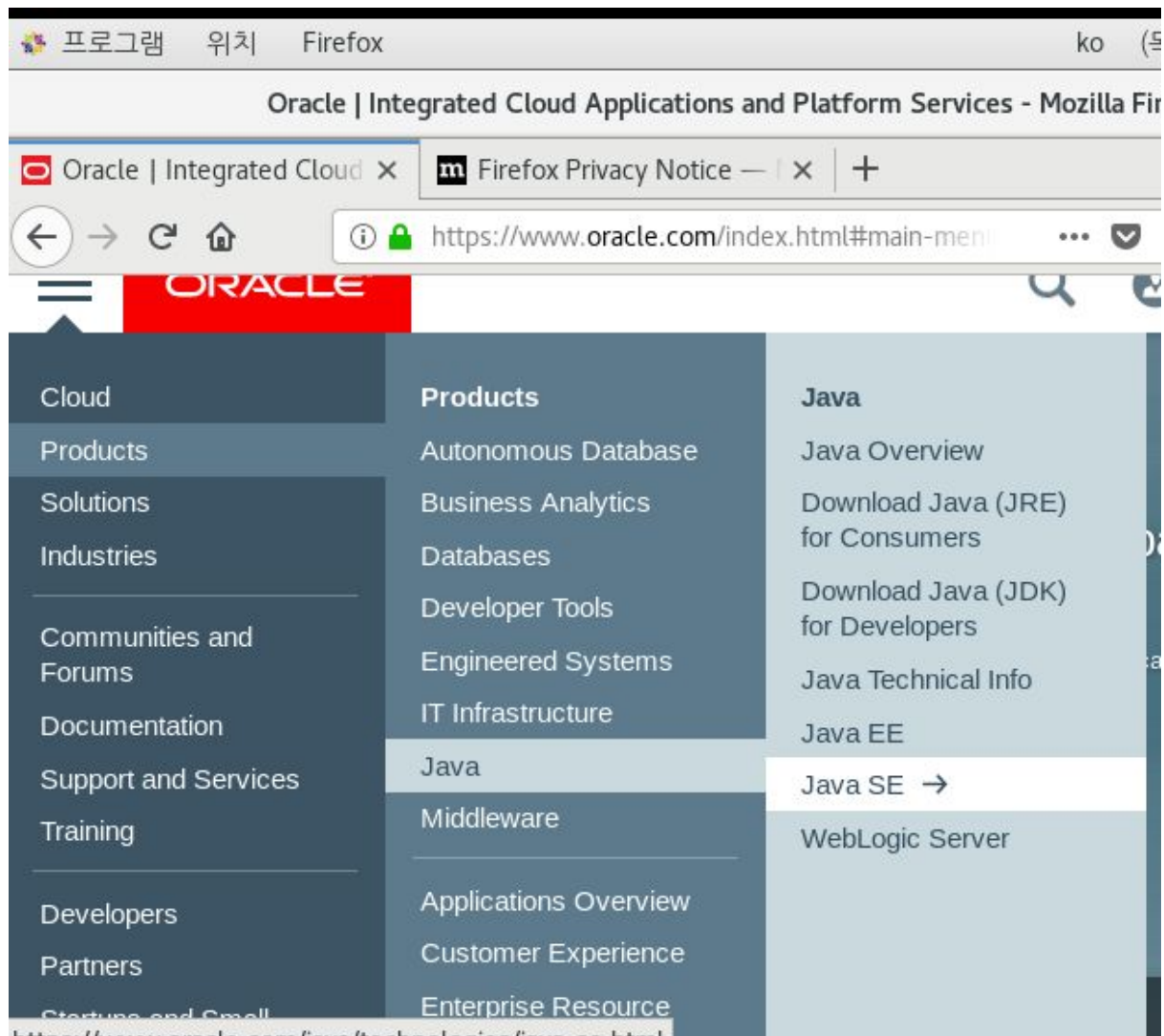
Firefox 웹 브라우저를 실행한 후 오라클 홈페이지(<http://oracle.com>)에서 jdk 64bit 리눅스용 tar.gz 버전을 다운로드한 후 압축을 해제한 후 `/usr/local/jdk1.8` 디렉토리로 옮긴다

자바의 Linux를 받아보자.

1.firefox실행

2.oracle.com 접속





자바 SE로 이동

## Get Started

[Download Java SE for Developers →](#)

[Learn about Oracle Java SE subscription →](#)

[Start Oracle Cloud for free →](#)

[Contact Sales →](#)



<a href="#">DOWNLOAD</a>	<a href="#">DOWNLOAD</a>
Java Platform (JDK) 12	NetBeans with JDK 8

### Java Platform, Standard Edition

**Java SE 12.0.1**  
 Java SE 12.0.1 is the latest release for the Java SE Platform  
[Learn more](#)

- Installation Instructions
- Release Notes
- Oracle JDK License
- Java SE Licensing Information User Manual
  - Includes Third Party Licenses
- Certified System Configurations
- Readme

**Oracle JDK**

[DOWNLOAD](#)

Looking for Oracle OpenJDK builds?

현재보면, 12버전까지 jdk새로운 버전이 나와있는 것을확인할 수 있다.  
 하지만, 최신버전 같은 경우는 호환성 측면에 있어서 문제가 발생할 확률이 높기 때문에  
 현업에서는 많이 선호되지 않는다.  
 따라서 우리는 1.8버전으로 down grade한 상태로 진행하도록 한다.

**Java SE 8u211 / Java SE 8u212**  
 Java SE 8u211 / Java SE 8u212 includes important bug fixes. Oracle strongly recommends that  
 all Java SE 8 users upgrade to this release.  
[Learn more](#)

- Installation Instructions
- Release Notes
- Oracle License

**JDK**

[DOWNLOAD](#)

## Java SE Development Kit 8u211

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software.

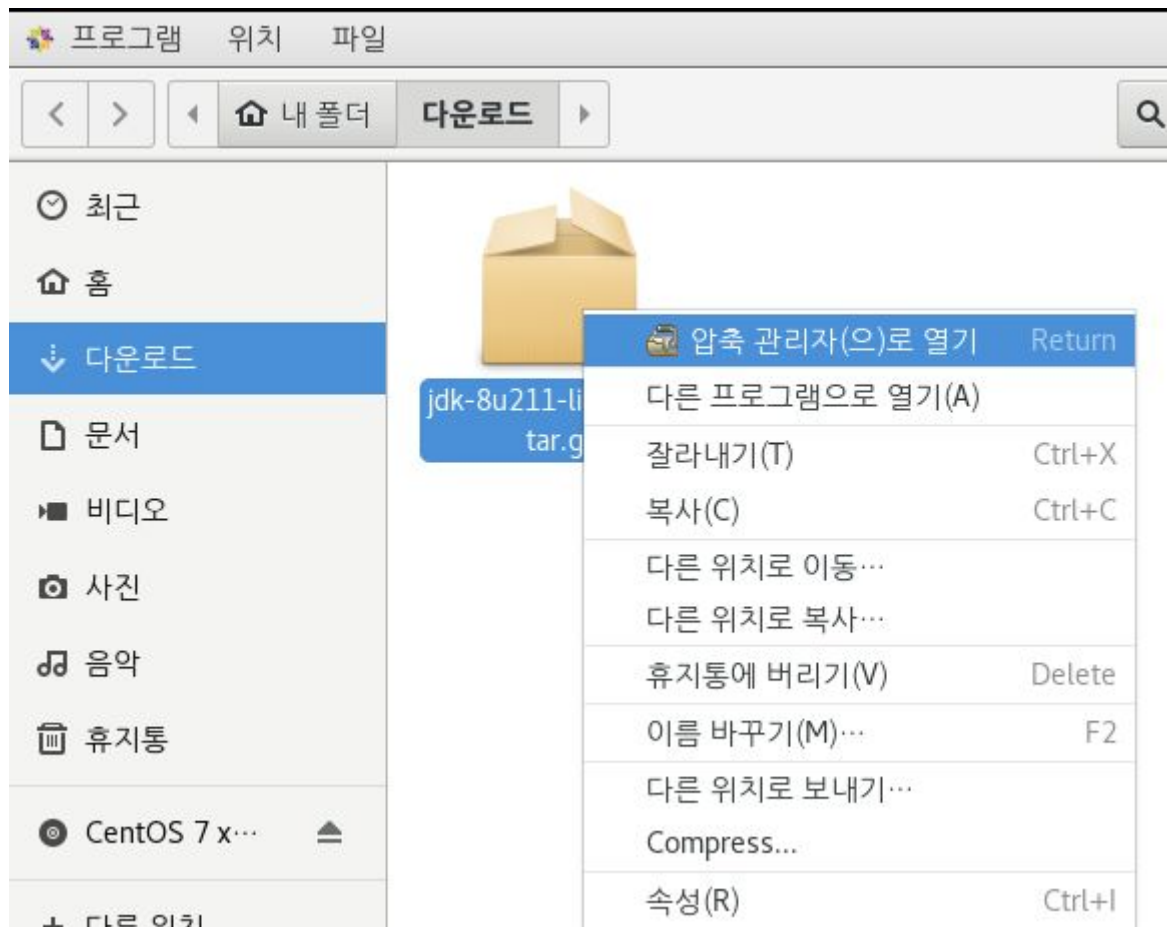
☐ Accept License Agreement
 ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.86 MB	<a href="#">jdk-8u211-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	69.76 MB	<a href="#">jdk-8u211-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	174.11 MB	<a href="#">jdk-8u211-linux-i586.rpm</a>
Linux x86	188.92 MB	<a href="#">jdk-8u211-linux-i586.tar.gz</a>
Linux x64	171.13 MB	<a href="#">jdk-8u211-linux-x64.rpm</a>
Linux x64	185.96 MB	<a href="#">jdk-8u211-linux-x64.tar.gz</a>
Mac OS X x64	252.23 MB	<a href="#">jdk-8u211-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	132.98 MB	<a href="#">jdk-8u211-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	94.18 MB	<a href="#">jdk-8u211-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	133.57 MB	<a href="#">jdk-8u211-solaris-x64.tar.Z</a>
Solaris x64	91.93 MB	<a href="#">jdk-8u211-solaris-x64.tar.gz</a>
Windows x86	202.62 MB	<a href="#">jdk-8u211-windows-i586.exe</a>
Windows x64	215.29 MB	<a href="#">jdk-8u211-windows-x64.exe</a>

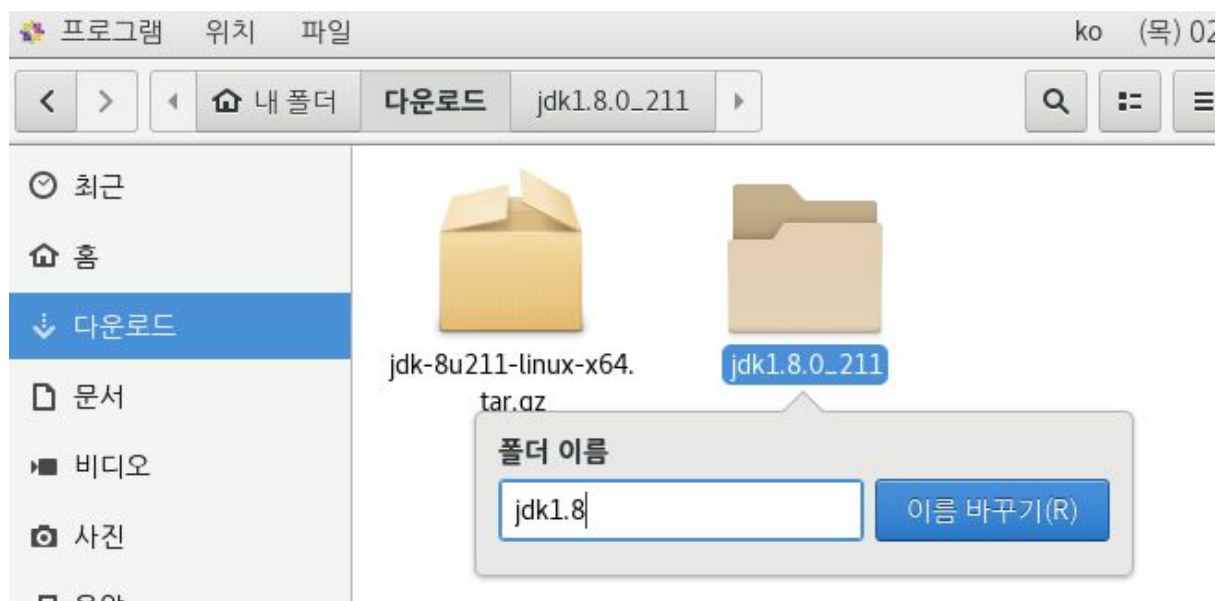
다음과 같은 화면에서 Linux ARM은 소형기기를 위한 jdk이기 때문에, 우리는 Linux x64를 설치하도록 한다.

그중에서도 tar.gz를 설치.

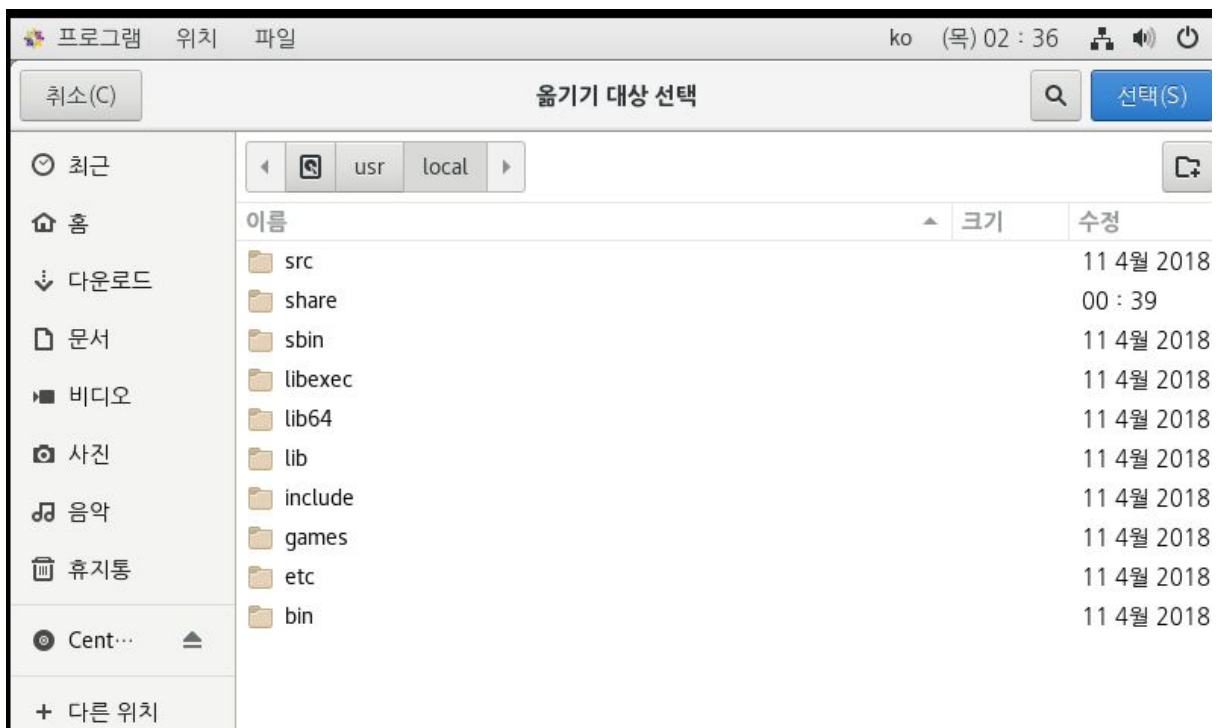
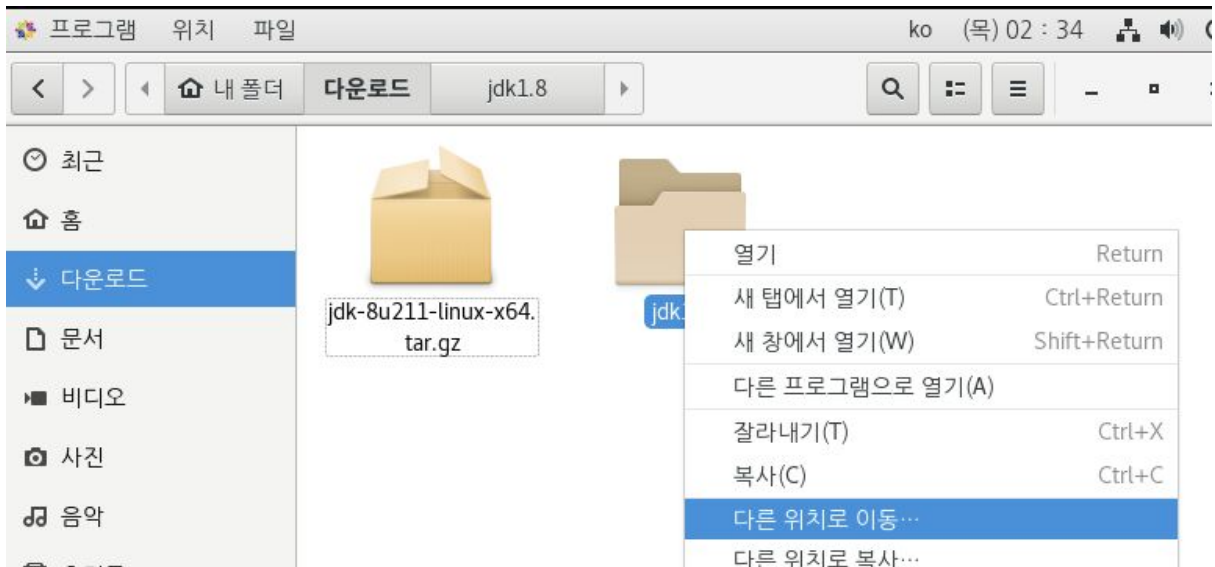




압축풀기



편의를 위해 이름을 짧게 변경해 준다.



다른위치->usr->local  
로 선택

## 라. 자바 환경변수 설정

### 1) 자동실행 스크립트 파일

- 가) 사용자가 로그인하면 자동으로 실행되는 스크립트, 윈도우즈의 autoexec.bat과 비슷한 기능
- 나) 사용자가 로그인하면 /etc/profile 과 .profile 스크립트가 자동으로 실행됨
- 다) /etc/profile : 모든 사용자에게 공통적으로 적용되는 스크립트
- 라) .profile : 사용자별로 적용되는 스크립트

### 2) 자동실행 스크립트 파일의 실행 순서

/etc/profile

/etc/bashrc

~/.bashrc

~/.bash\_profile

### 3) /etc/profile 스크립트 파일 편집

터미널에서 실행

```
gedit /etc/profile
```

맨 아래에 내용 추가

```
export JAVA_HOME=/usr/local/jdk1.8
export PATH=$PATH:$JAVA_HOME/bin
export JAVA_OPTS="-Dfile.encoding=UTF-8"
export CLASSPATH=."
```

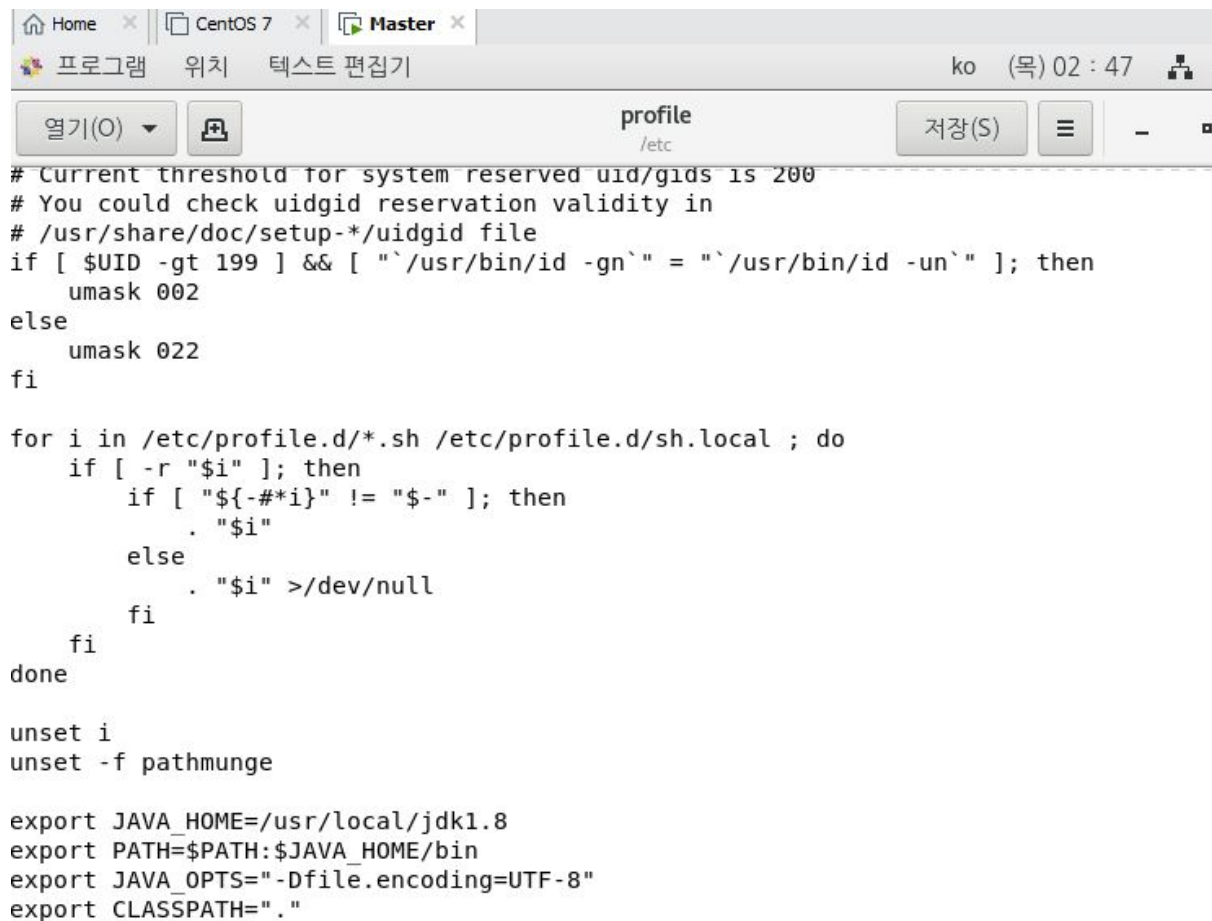
저장하고 파일을 닫음



```
[root@localhost ~]# cd /usr/local/jdk1.8/bin
[root@localhost bin]# ls -la java*
-rwxr-xr-x. 1 root root 8464 4월 2 12:48 java
-rwxr-xr-x. 1 root root 1809 4월 2 12:48 java-rmi.cgi
-rwxr-xr-x. 1 root root 8584 4월 2 12:48 javac
-rwxr-xr-x. 1 root root 8584 4월 2 12:48 javadoc
-rwxr-xr-x. 1 root root 2293 3월 14 17:37 javafxpackager
-rwxr-xr-x. 1 root root 8584 4월 2 12:48 javah
-rwxr-xr-x. 1 root root 8584 4월 2 12:48 javap
-rwxr-xr-x. 1 root root 2293 3월 14 17:37 javapackager
-rwxr-xr-x. 1 root root 140296 4월 2 12:48 javaws
[root@localhost bin]#
```

```
[root@localhost bin]# gedit /etc/profile
```

맨 밑에 작성



```
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge

export JAVA_HOME=/usr/local/jdk1.8
export PATH=$PATH:$JAVA_HOME/bin
export JAVA_OPTS="-Dfile.encoding=UTF-8"
export CLASSPATH="."
```

저장하고 닫기

이후에 시스템에 적용시키고 재부팅을 하자

4) 변경된 사항을 시스템에 적용

```
source /etc/profile
```

5) 재부팅

```
reboot
```

```
프로그램  위치  터미널 ko
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# java -version
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
[root@localhost ~]#
```

재부팅 후 확인.

JDK설치 끝.

## 하둡설치


Home CentOS 7 Master

프로그램 위치 Firefox ko (금) 00 : 12

Apache Hadoop - Mozilla Firefox

Restore Session Apache Hadoop

hadoop.apache.org

 Apache Hadoop

Download Documentation Community Development Help Old site

Apache Software Foundation

storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

Learn more » Download » Getting started »

#### 4) 하둡 설치

2019.1월 현재 하둡의 최신 버전은 3.0.3 이며 여기서는 2.9.2 로 실습

<http://hadoop.apache.org> 사이트에서 hadoop-2.9.2.tar.gz 다운로드(binary 버전)

<http://mirror.apache-kr.org/hadoop/common/hadoop-2.9.2/hadoop-2.9.2.tar.gz>

압축을 풀어서 /home/centos/hadoop-2.9.2 로 옮김

#### 5) 하둡 환경설정

HADOOP\_HOME을 환경변수에 추가

```
gedit /etc/profile
```

하둡 관련 설정을 추가한다.

```
export JAVA_HOME=/usr/local/jdk1.8
export JAVA_OPTS="-Dfile.encoding=UTF-8"
export CLASSPATH="."
export HADOOP_HOME=/home/centos/hadoop-2.9.2
export
PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

export HDFS_NAMENODE_USER="root"
export HDFS_DATANODE_USER="root"
export HDFS_SECONDARYNAMENODE_USER="root"
export YARN_RESOURCEMANAGER_USER="root"
export YARN_NODEMANAGER_USER="root"
```

```
export JAVA_HOME=/usr/local/jdk1.8
```

```
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

2개의 디렉토리를 추가

```
export JAVA_OPTS="-Dfile.encoding=UTF-8"
```

```
export CLASSPATH="."
```

```
export HADOOP_HOME=/home/centos/hadoop-2.9.2
```

하둡의 홈이 어디냐

```
export HDFS_NAMENODE_USER="root"
```

```
export HDFS_DATANODE_USER="root"
```

```
export HDFS_SECONDARYNAMENODE_USER="root"
```

```
export YARN_RESOURCEMANAGER_USER="root"
```

```
export YARN_NODEMANAGER_USER="root"
```

노드들의 유저를 기본값으로 root로 설정한다.

```
export JAVA_HOME=/usr/local/jdk1.8
```

```
export JAVA_OPTS="-Dfile.encoding=UTF-8"
```

```
export CLASSPATH="."
```

```
export HADOOP_HOME=/home/centos/hadoop-2.9.2
```

```
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

```
export HDFS_NAMENODE_USER="root"
```

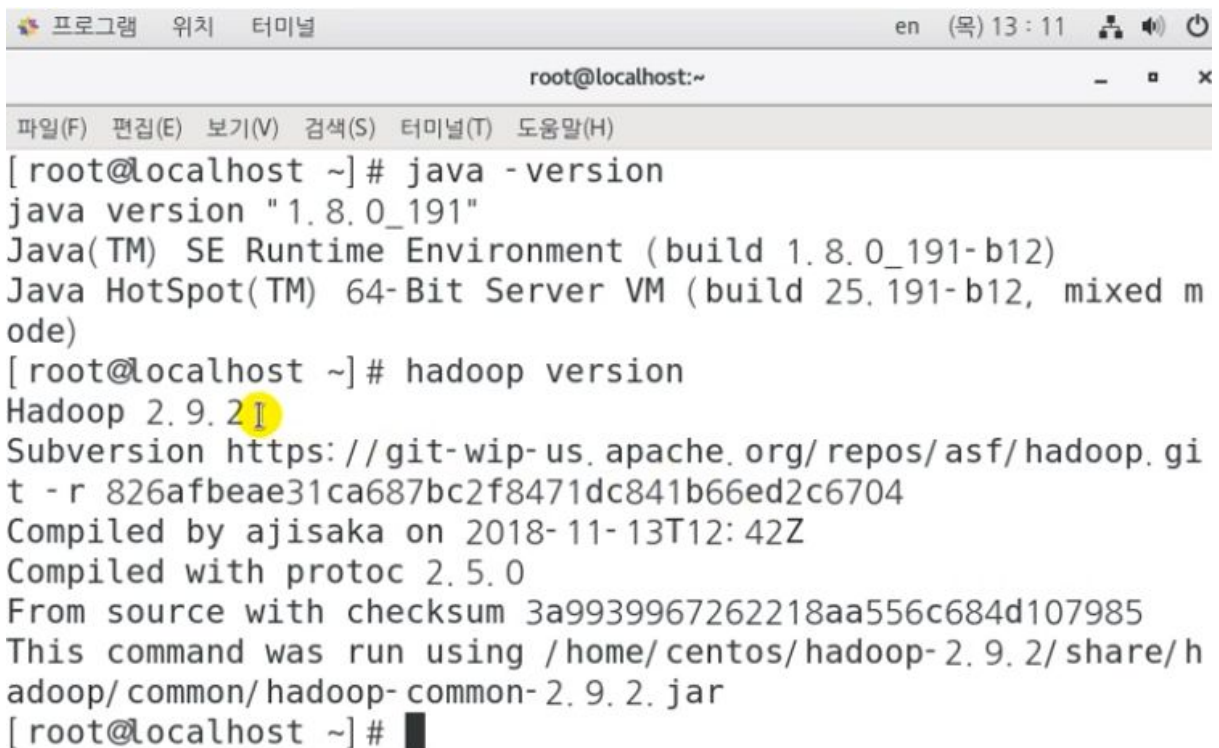
```
export HDFS_DATANODE_USER="root"
```

```
export HDFS_SECONDARYNAMENODE_USER="root"
```

```
export YARN_RESOURCEMANAGER_USER="root"
```

```
export YARN_NODEMANAGER_USER="root"
```

```
[root@localhost ~]# gedit /etc/profile
[root@localhost ~]# source /etc/profile
[root@localhost ~]# reboot
```



```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
[root@localhost ~]# hadoop version
Hadoop 2.9.2
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git - r 826afbeae31ca687bc2f8471dc841b66ed2c6704
Compiled by ajisaka on 2018-11-13T12:42Z
Compiled with protoc 2.5.0
From source with checksum 3a9939967262218aa556c684d107985
This command was run using /home/centos/hadoop-2.9.2/share/hadoop/common/hadoop-common-2.9.2.jar
[root@localhost ~]#
```

## 하둡 실행 실습

### 단독모드

### 의사분산모드

### 완전분산모드

의사 분산모드는 가짜 분산모드이다. 여러대를 깔았지만 자원을 공유하고 있지 않은 상태

#### 단독모드 실습

##### 다. 실습1 - Single Node(단독모드)

하둡의 설정을 변경하지 않은 상태에서 하둡에 내장된 예제 jar 파일을 이용한 테스트

wordcount 프로그램을 실행하여 hadoop-env.sh 파일의 단어 갯수 분석

```
hadoop jar
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.
jar wordcount $HADOOP_HOME/etc/hadoop/hadoop-env.sh wordcount_output
```

실행결과 파일 목록 확인

```
ls -la wordcount_output/*
```

실행결과 확인

```
cat wordcount_output/part-r-00000
또는
cat wordcount_output/*
```

```
hadoop jar
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar
wordcount $HADOOP_HOME/etc/hadoop/hadoop-env.sh wordcount_output
```

wordcount라는 class를 실행하는데 이 class가 뭐냐하면, 파일의 wordcount를 세는 class이다.

근데 어떤 파일을 셀것이나. etc/hadoop/hadoop-env.sh



실행결과 파일 목록 확인

```
ls -la wordcount_output/*
```



실행결과 확인

```
cat wordcount_output/part-r-00000
```

또는

```
cat wordcount_output/*
```

다시 한번 실행하면 에러가 발생함(output 디렉토리가 중복되면 안됨)

다시 실행하려면 wordcount\_output 디렉토리 삭제 후 재실행

```
rm -rf wordcount_output
```

```

[root@localhost ~]# ls -la wordcount_output/*
-rw-r--r--. 1 root root 0 5월 31 01:00 wordcount_output/_SUCCESS
-rw-r--r--. 1 root root 3905 5월 31 01:00 wordcount_output/part-r-00000
[root@localhost ~]# cat wordcount_output/*
"
1
"$HADOOP_CLASSPATH" 1
"$HADOOP_HEAPSIZE" 1
"AS 1
"License"); 1
# 61
### 6
#HADOOP_JAVA_PLATFORM_OPTS="-XX:-UsePerfData 1
#export 5
$HADOOP_CLIENT_OPTS" 1
$HADOOP_DATANODE_OPTS" 1
$HADOOP_HOME/contrib/capacity-scheduler/*.jar; 1
$HADOOP_HOME/logs 1
$HADOOP_JAVA_PLATFORM_OPTS" 1
$HADOOP_NAMENODE_OPTS" 1
$HADOOP_PORTMAP_OPTS" 1
$HADOOP_SECONDARYNAMENODE_OPTS" 1
$USER 1
(ASF) 1
(fs, 1
(the 1
**MUST 1
**MUST** 1
-Dhdfs.audit.logger=${HDFS_AUDIT_LOGGER:-INFO,NullAppender} 2
-Djava.net.preferIPv4Stack=true" 1
-Dsun.security.krb5.debug=true 1
-Dsun.security.spnego.debug" 1
/tmp 1
1000. 1
2.0 1
= 1
A 1
ANY 1
ASF 1
Advanced 1
All 1
Apache 2
Automatically 1
BASIS, 1
CLASSPATH 1
CONDITIONS 1
Command 1
Default 1
Empty 1
Enable 1
Extra 2
Federation 1
For 1
Foundation 1

```

---