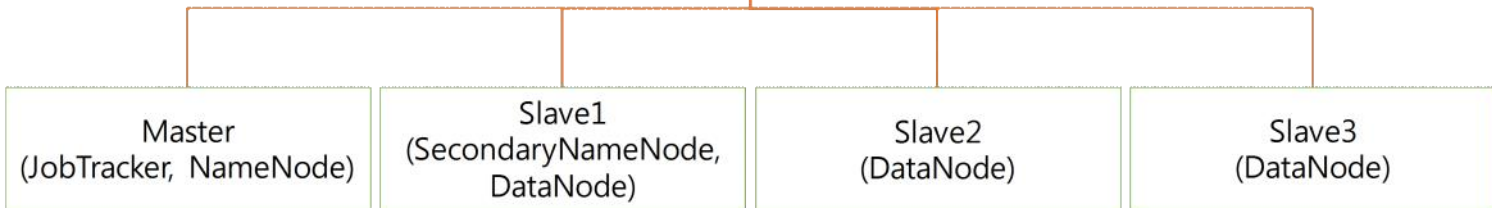


## 공통 설치 파일

우분투 설치 : 랩 1GB, 하드디스크 20GB  
기타 util 설치 : SSH, Vim  
네트워크 설정 / 호스트 파일 수정  
자바설치 : JDK 8.x  
하둡설치 : 2.9.0  
그룹 및 계정 생성  
- group : Hadoop  
- account : manager  
하둡 폴더 소유자 변경

복제



2017년 9월 30일 토요일      오전 11:12

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)



# VirtualBox

# Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 [virtualization](#) solution. VirtualBox is an extremely feature rich, high performance product. VirtualBox is a solution that is freely available as Open Source Software under the terms of the [GNU GPL](#). See "[About VirtualBox](#)" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and **systems** including but not limited to Windows (NT 4.0, 2000

## About

## Screenshots

## Downloads

## Documentation

다운로드 클릭

## Download VirtualBox

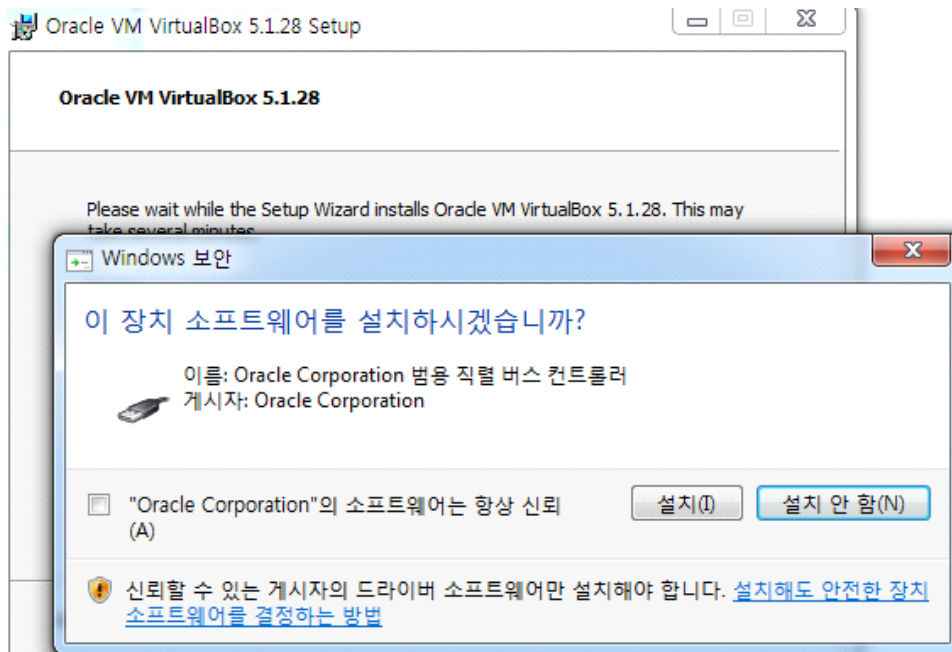
Here, you will find links to VirtualBox binaries and its s

## VirtualBox binaries

By downloading, you agree to the terms and conditions

- **VirtualBox 5.1.28 platform packages.** The binaries are available for:
  - [Windows hosts](#)
  - [OS X hosts](#)
  - [Linux distributions](#)
  - [Solaris hosts](#)
- **VirtualBox 5.1.28 Oracle VM VirtualBox Extension Pack**  
Support for USB 2.0 and USB 3.0 devices, Virtual Machine Monitors, and other features. This is the first introduction to this Extension Pack.  
The Extension Pack binaries are released under a

원도우 호스트 클릭

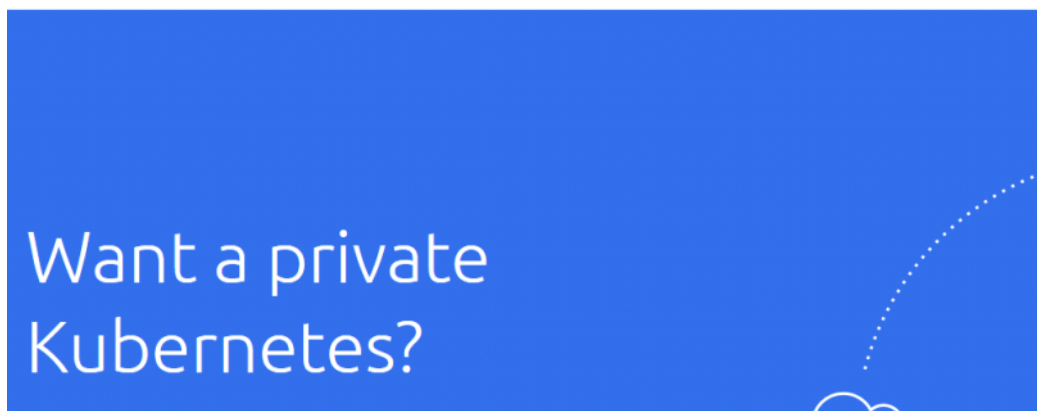
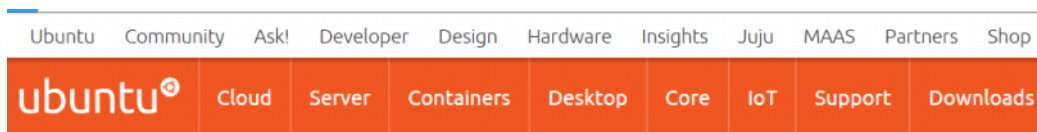


# 우분투 설치

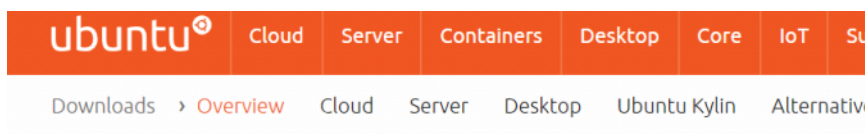
2017년 9월 30일 토요일    오전 11:31

## 다운로드

2017년 9월 30일 토요일 오전 11:12

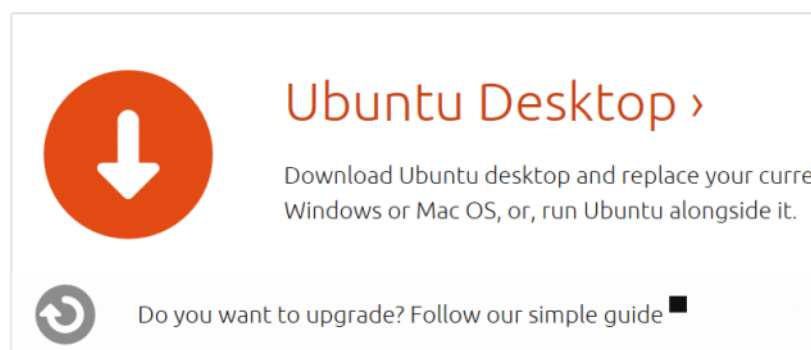


위의 메뉴 끝 Download 클릭



# Get Ubuntu

Ubuntu is completely free to download, use and share.



위의 화면 클릭

# Download Ubuntu Desktop

## Ubuntu 16.04.3 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years of free security and maintenance updates, guaranteed.

[Ubuntu 16.04 LTS release notes](#) ■

Recommended system requirements:

✓ 2 GHz dual core processor or better

[Download](#)

[Alternative downloads and torrents ›](#)

Download 클릭



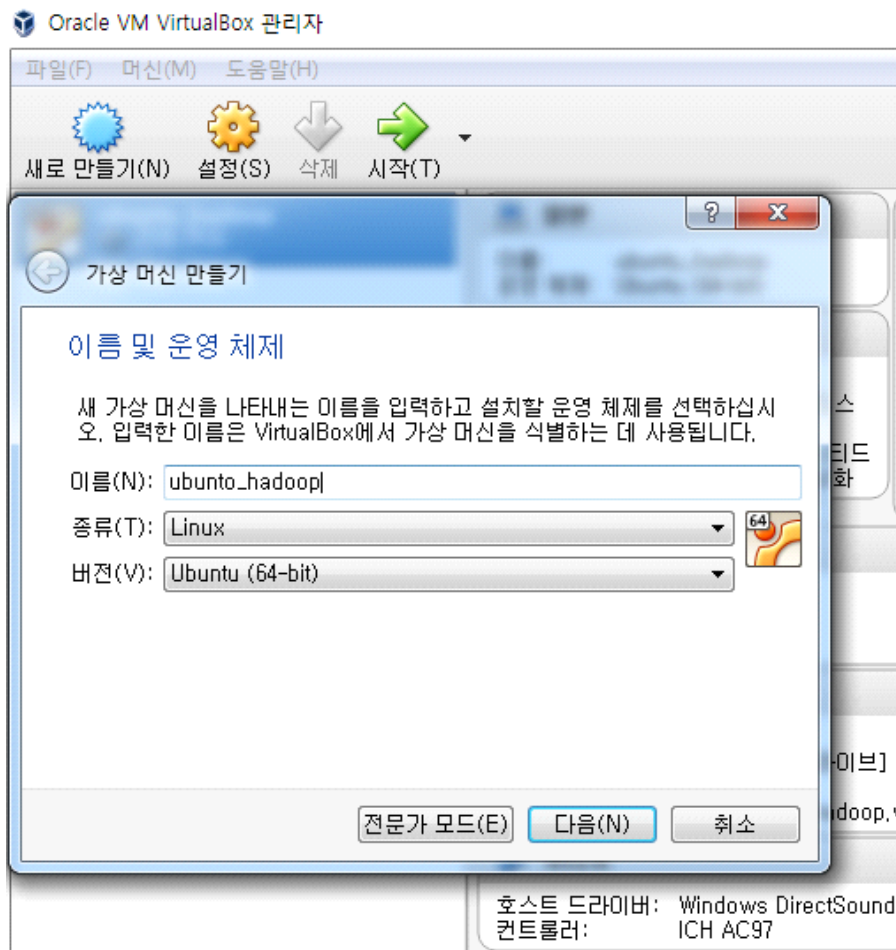
## Help shape the future of Ubuntu

Tell us what is most important to you, or [you can continue to the download ›](#)

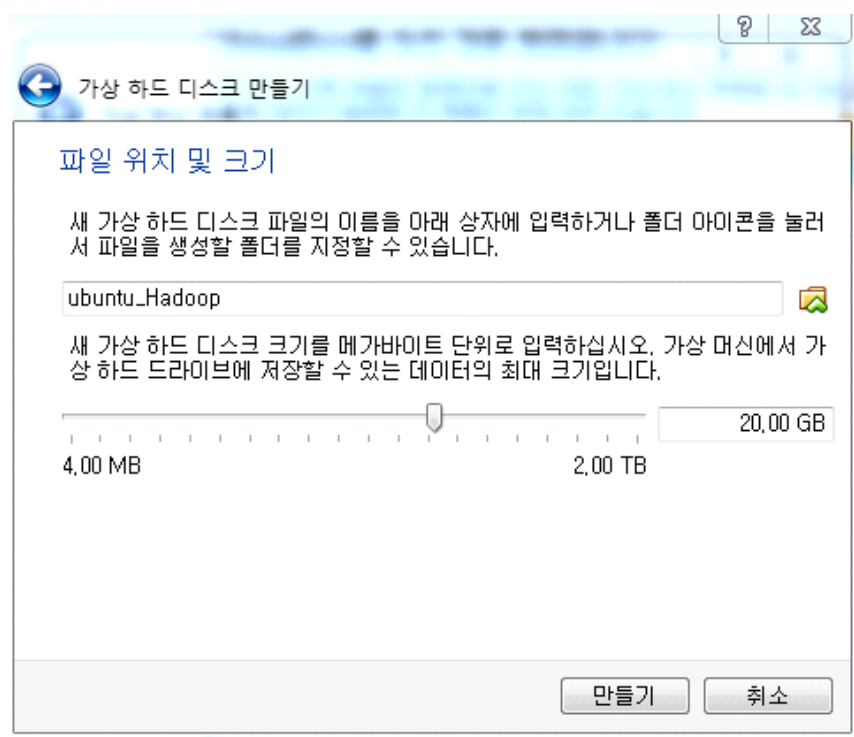
빨간색 부분 you can continue to the download 클릭

# 가상머신에 설치

2017년 9월 30일 토요일 오전 11:32

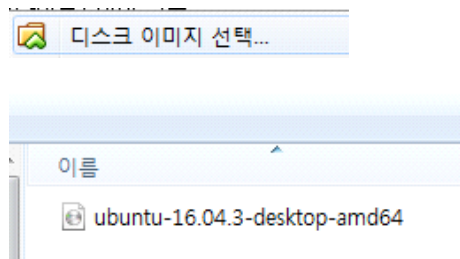


새로 만들기 클릭



하드 디스크의 크기를 20GB로 설정한다.

나머지 기본값 설정



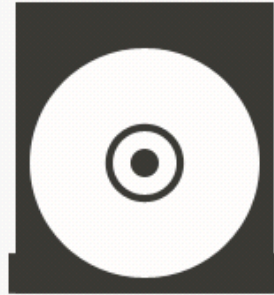
위와 같이 광학드라이브 -> 디스크 이미지 선택 -> ISO 선택

이미지 선택 후 시작버튼을 클릭하면 설치가 시작된다.



## 환영합니다

हिन्दी  
বাংলা  
ਪੰਜਾਬੀ  
ગુજરાતી  
தமிழ்  
பெரிய  
മലയാളം  
සිංහල  
ภาษาไทย  
ລາວ  
ବଙ୍ଗଳା  
မြန်မာစာ  
**한국어**  
中文(简体)  
中文(繁體)  
日本語



Ubuntu 체험하기



Ubuntu 설치

컴퓨터를 전혀 바꾸지 않고 CD에서 바로 Ubuntu을(를) 사용할 수 있습니다.

준비가 되어 있다면 현재 사용하는 운영 체제를 지우지 않고 Ubuntu을(를) 설치할 수 있습니다. 오랜 시간이 필요하지 않습니다.

[릴리즈 노트](#)를 읽을 수 있습니다.

## Ubuntu 설치 준비 중

☐ Ubuntu 설치 중 업데이트 다운로드

설치 후 작업 시간을 줄일 수 있습니다.

☐ 그래픽과 와이파이 하드웨어, 플래시, MP3 등의 추가 소프트웨어를 설치

This software is subject to license terms included with its documentation. Some is proprietary.

Fluendo MP3 플러그인은 Fraunhofer IIS와 Technicolor SA로부터 인증 받은 MPEG 레이어-3 기술을 사용합니다.

끝내기(Q)

뒤로(B)

계속

## 설치 형식

이 컴퓨터에는 설치한 운영 체제가 없습니다. 어떻게 하시겠습니까?

☒ 디스크를 지우고 Ubuntu 설치

**경고:** 이 작업은 모든 운영 체제에서 프로그램, 문서, 사진, 음악, 그리고 기타 파일을 삭제합니다.

☐ 설치한 새 Ubuntu을(를) 암호화하여 보안을 높임  
다음 단계에서 보안 키를 선택할 수 있습니다.

☐ LVM을 사용해 Ubuntu 설치  
논리 볼륨 관리를 설정합니다. 이 작업을 하면 스냅샷을 만들거나 파티션 크기를 쉽게 바꿀 수 있습니다.

☐ 기타  
Ubuntu이(가) 사용할 파티션을 만들거나 크기를 바꾸거나 여러 파티션을 선택할 수 있습니다.

끝내기(Q)

뒤로(B)

지금 설치(I)

❌ 바뀐 점을 디스크에 쓰시겠습니까?

계속하시면 아래의 바뀐 사항을 디스크에 씁니다. 계속하지 않으시면 나중에 수동으로 설정을 바꿀 수 있습니다.

다음 장치의 파티션 테이블이 바뀌었습니다:  
SCSI3 (0,0,0) (sda)

다음 파티션을 포맷합니다:  
SCSI3 (0,0,0) (sda) 장치의 #1 파티션에 있는 ext4  
SCSI3 (0,0,0) (sda) 장치의 #5 파티션에 있는 스왑

뒤로 이동

계속

설치 (as superuser)

어디에 살고 계신가요?



Seoul

뒤로(B)

계속

기본값으로 설치하다 다음과 같은 화면이 나타나면 한국어를 선택한다.

## 키보드 배치

키보드 배치 선택:

프랑스어(캐나다)

프랑스어(콩고 민주 공화국)

핀란드어

필리핀어

한국어

헝가리어

히브리어

한국어

한국어 - 한국어(101/104기 호환)

선택한 키보드 배치를 확인하려면 이 곳을 사용하십시오.

키보드 배치 확인

## 당신은 누구십니까?

이름: jin ✓

컴퓨터 이름: jin-VirtualBox ✓

다른 컴퓨터에서 보여지는 이름

사용자 이름 선택: jin ✓

암호 선택: ●●●●●● 양호한 암호

암호 확인: ●●●●●● ✓

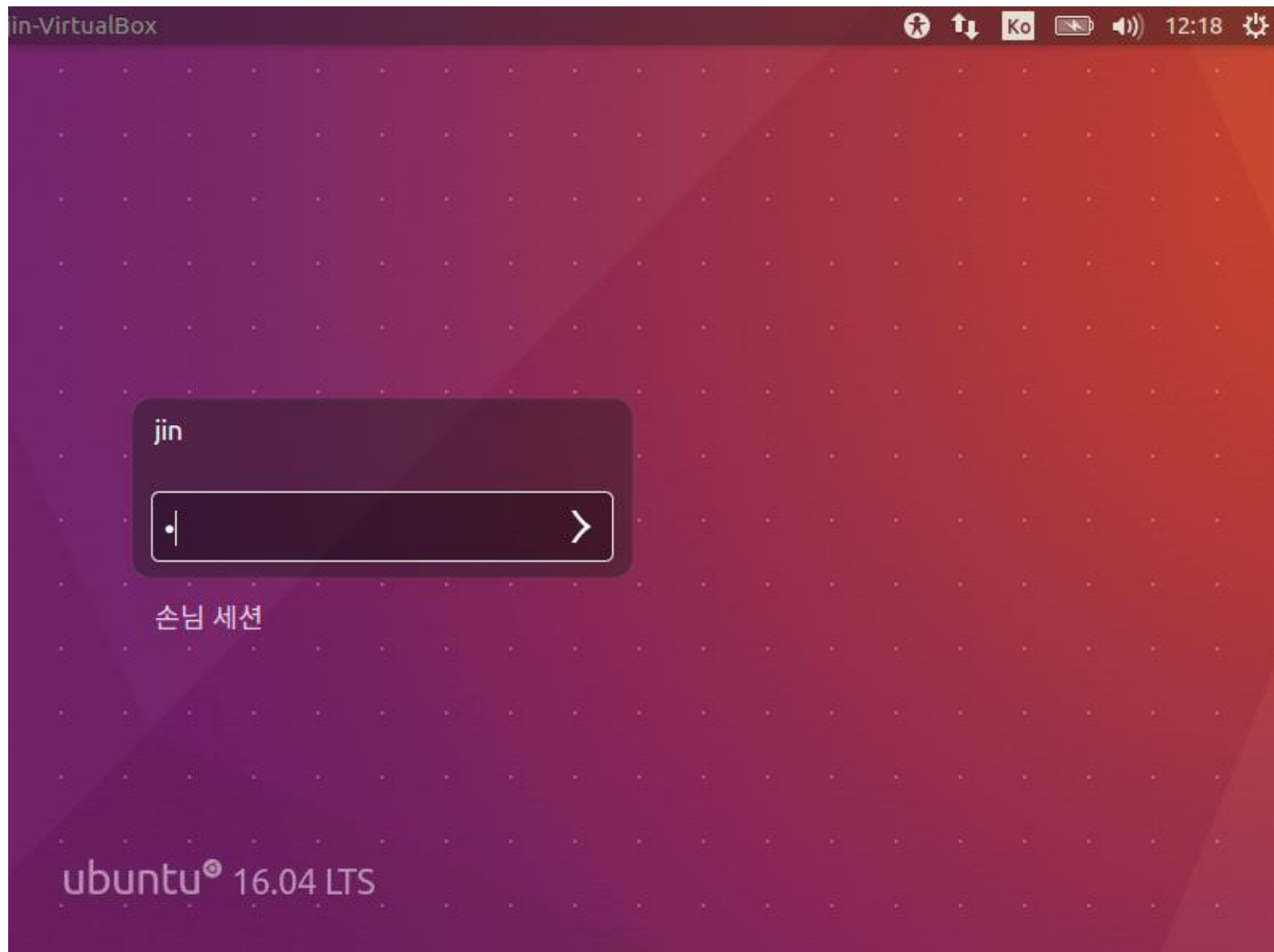
☐ 자동으로 로그인

☒ 로그인할 때 암호 입력

☐ 개인 폴더 암호화

뒤로(B)

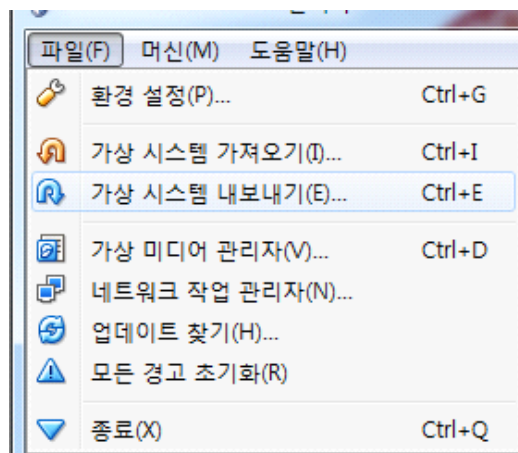
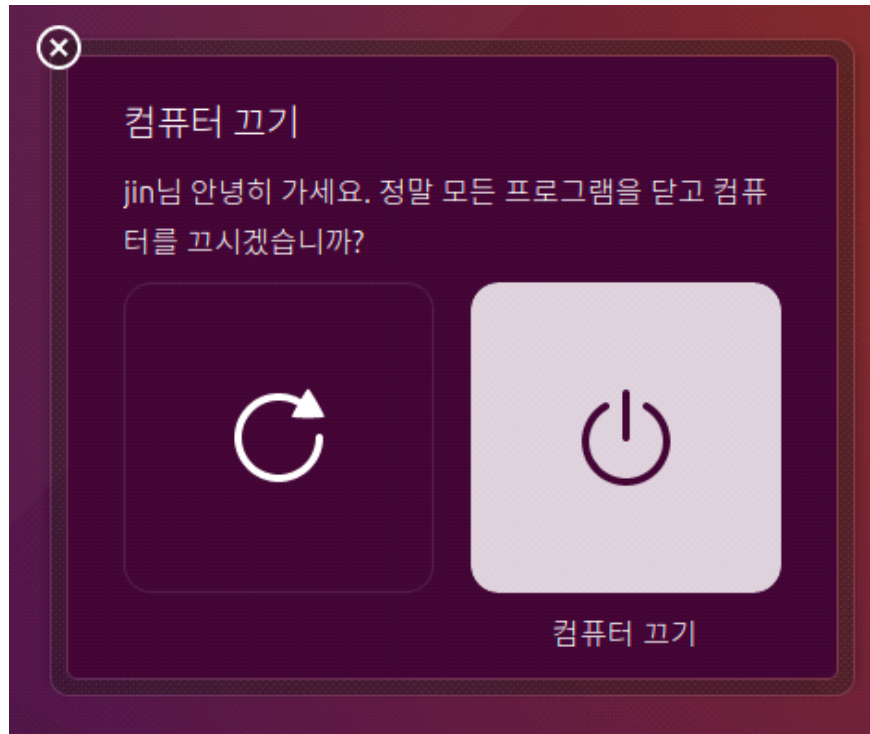
계속



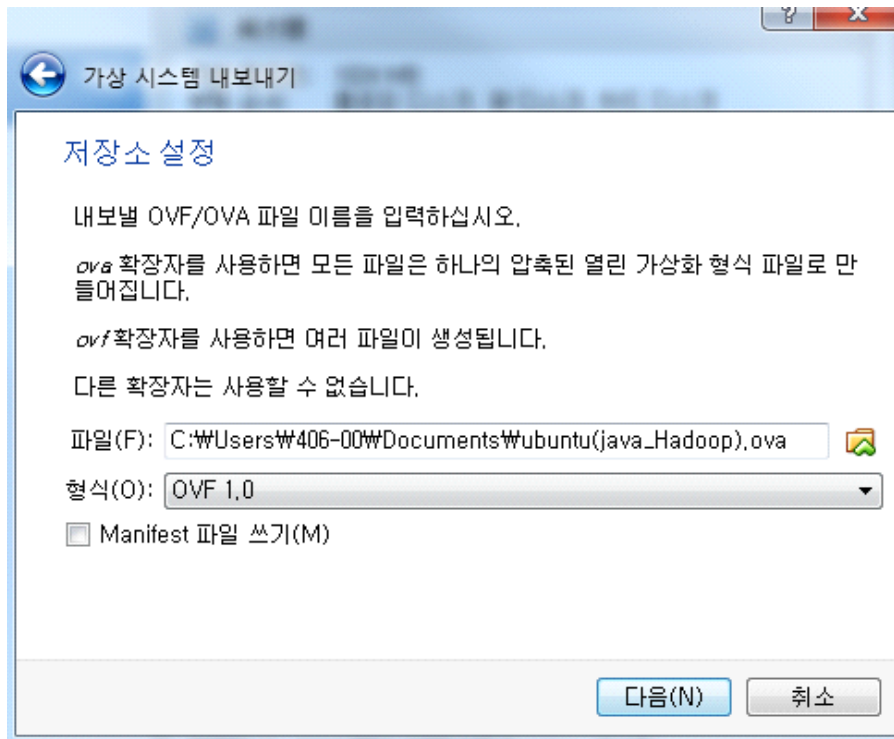
정상적으로 부팅 완료

# 복제하기

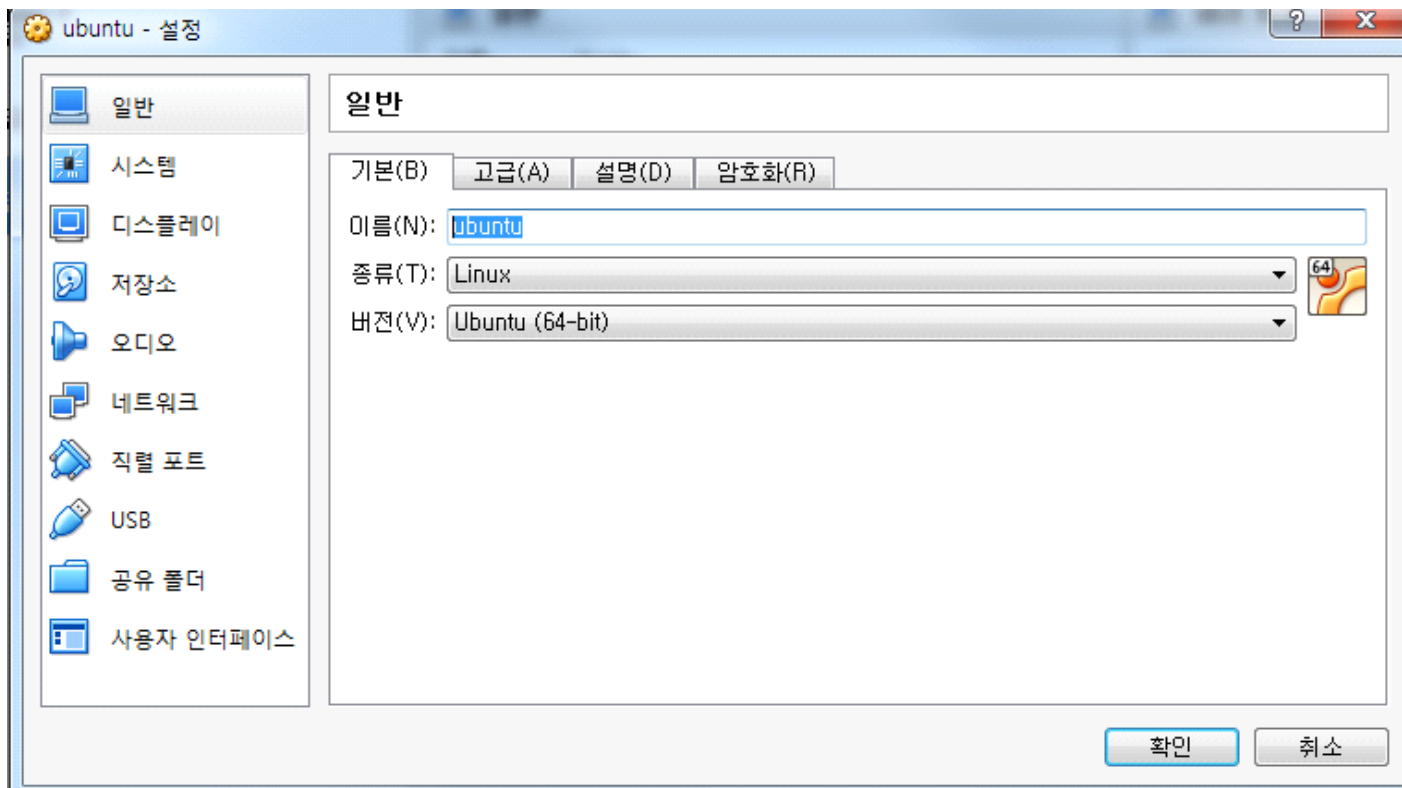
2017년 11월 21일 화요일    오후 2:02



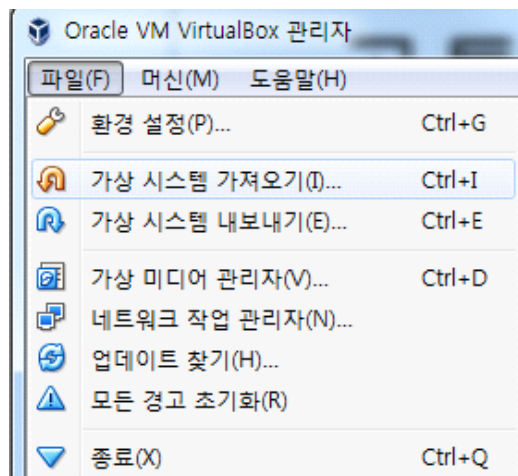




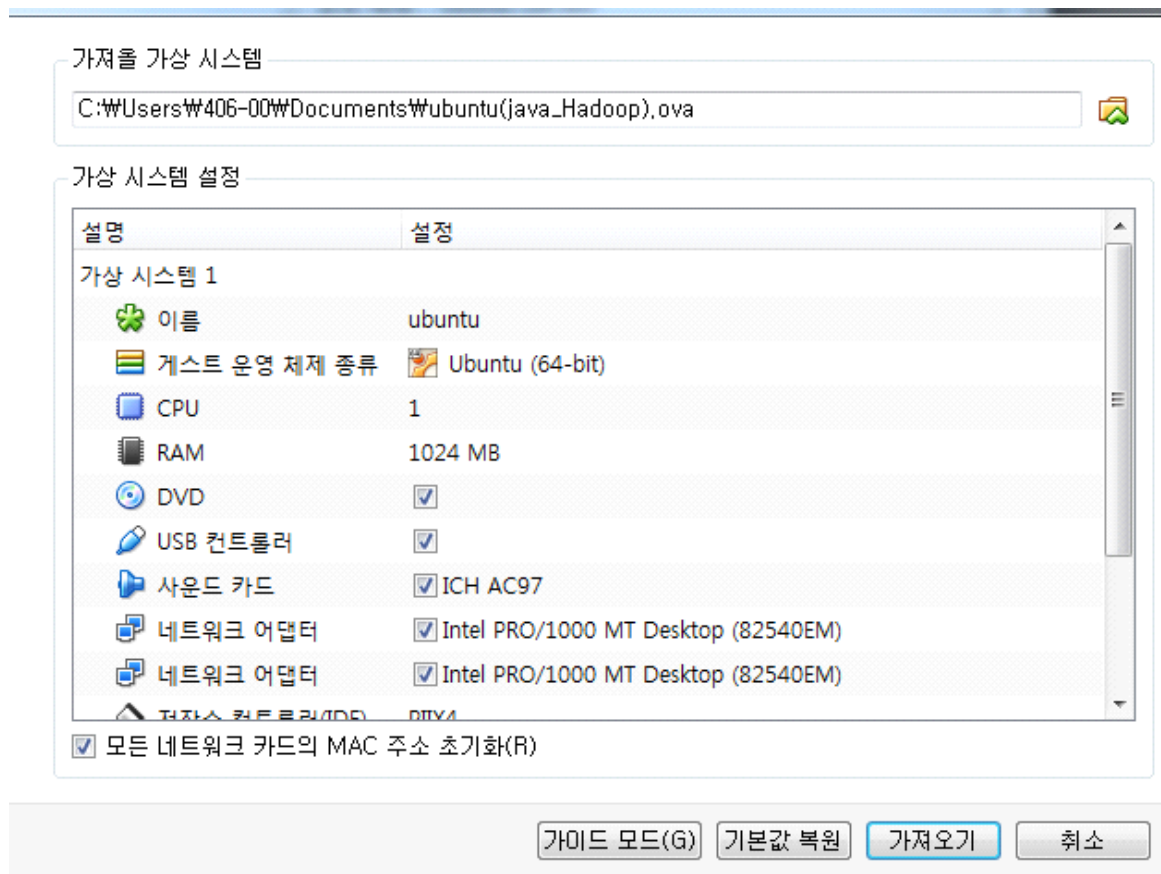
기본값으로 지금까지 설정한 내용을 내보내기 한다.



내보내기 완료 후 설정으로 이동하여 이름을 각 서버에 맞게 변경(master, slave1, slave2, slave3)



가져오기



모든 네트워크 카드의 MAC 주소 초기화 클릭



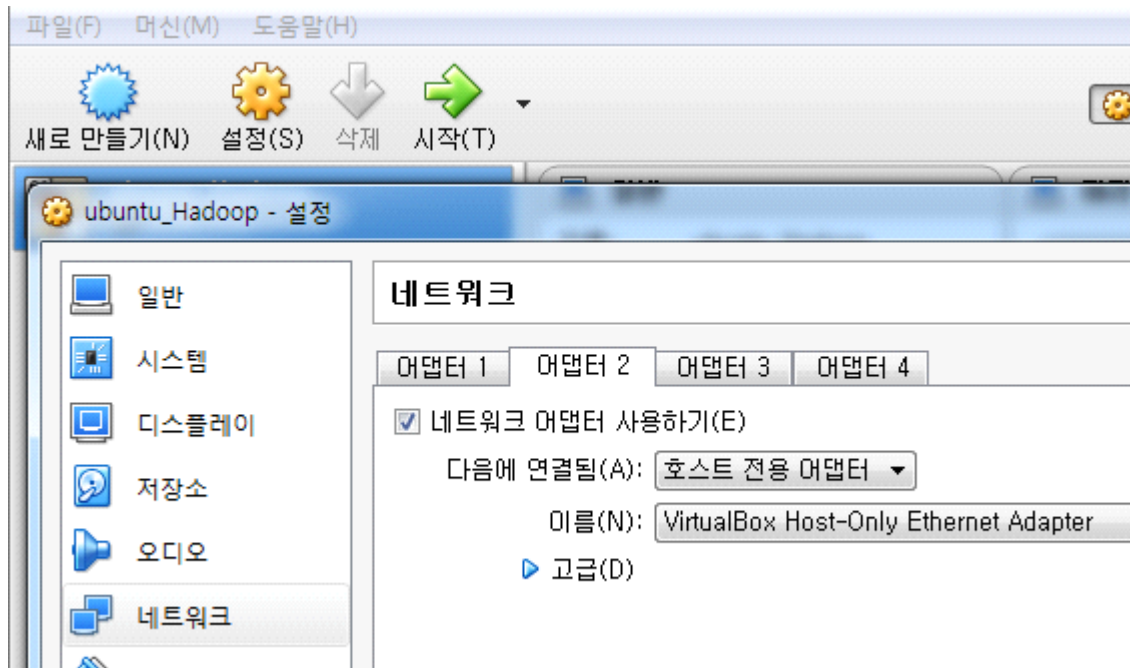
# 단일 노드 설정

2017년 11월 21일 화요일    오전 11:26



# 네트워크 설정

2017년 10월 3일 화요일    오후 3:13



1. 톱니바퀴 모양의 설정 클릭
2. 설정창에서 네트워크 클릭
3. 어댑터 2 선택하여 호스트 전용 어댑터 선택

```
jln@jln-VirtualBox:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:71:99:21
        inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::68be:903b:1eb1:1c58/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:10 errors:0 dropped:0 overruns:0 frame:0
        TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:2079 (2.0 KB)  TX bytes:7550 (7.5 KB)

enp0s8  Link encap:Ethernet  HWaddr 08:00:27:48:cd:d0
        inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
        inet6 addr: fe80::d1d2:1e73:2246:575/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:8 errors:0 dropped:0 overruns:0 frame:0
        TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:1559 (1.5 KB)  TX bytes:7485 (7.4 KB)
```

위와 같이 진행하면 두 개의 랜카드가 설정되었던 것을 확인 할 수 있다. 첫 번째는 외부 인터넷을 하기 위한 NAT이고 두 번째는 내부 통신을 위한 호스트 전용 이다. 내부에서만 처리할 것이기 때문에 호스트 전용의 IP를 사용할 것이다.

## 유틸 설치

2017년 9월 30일 토요일 오후 2:41

Putty 설치



putty



전체

이미지

동영상

지도

뉴스

더보기

설정

도구

검색결과 약 32,600,000개 (0.44초)

### Download PuTTY: latest release (0.70) - Chiark

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> ▼ 이 페이지 번역하기

2017. 7. 8. - This page contains download links for the latest released version of **PuTTY**. Currently this is 0.70, released on 2017-07-08. When new releases ...

[PuTTY FAQ](#) · [PuTTY Documentation Page](#) · [Download PuTTY: release 0.70](#) · [Links](#)

### PuTTY: a free SSH and Telnet client - Chiark

<https://www.chiark.greenend.org.uk/~sgtatham/putty/> ▼ 이 페이지 번역하기

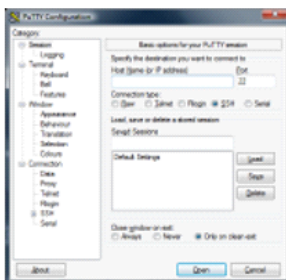
2017. 7. 8. - **PuTTY** is a free implementation of SSH and Telnet for Windows and Unix platforms, along with an xterm terminal emulator. It is written and ...

### Download PuTTY - a free SSH and telnet client for Windows

[www.putty.org/](http://www.putty.org/) ▼ 이 페이지 번역하기

**PuTTY** is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. **PuTTY** is open source software that is available with source ...

세 번째 위치한 [www.putty.org](http://www.putty.org) 접속



## Download PuTTY

PuTTY is an SSH and telnet client, developed originally that is available with source code and is developed

You can download PuTTY [here](http://www.putty.org/).

Here 클릭

This page contains download links for the latest released version of PuTTY. Currently this is 0.70, released When new releases come out, this page will update to contain the latest, so this is a good page to bookmark Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

## Package files

You probably want one of these. They include all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

### MSI ('Windows Installer')

32-bit: [putty-0.70-installer.msi](#) [\(or by FTP\)](#) [\(signature\)](#)

64-bit: [putty-64bit-0.70-installer.msi](#) [\(or by FTP\)](#) [\(signature\)](#)

### Unix source archive

.tar.gz: [putty-0.70.tar.gz](#) [\(or by FTP\)](#) [\(signature\)](#)

64비트 선택

Ssh 설치

```
jln@jln-VirtualBox:~$ sudo apt-get install ssh
패키지 목록을 읽는 중입니다... 완료
의존성 트리들을 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
The following additional packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
제안하는 패키지:
  ssh-askpass rssh molly-guard monkeysphere
다음 새 패키지를 설치할 것입니다:
```

에러 발생

```
jln@jln-VirtualBox:~$ sudo apt-get install ssh -y
[sudo] password for jln:
E: /var/lib/dpkg/lock 잠금 파일을 얻을 수 없습니다 - open (11: 자원이 일시적으로
사용 불가능함)
E: 관리 디렉터리를 (/var/lib/dpkg/) 잠글 수 없습니다. 다른 프로세스가 사용하고
있지 않습니까?
jln@jln-VirtualBox:~$ sudo rm /var/lib/dpkg/lock
jln@jln-VirtualBox:~$ sudo apt-get update
```

```
jln@jln-VirtualBox:~$ which ssh
/usr/bin/ssh
jln@jln-VirtualBox:~$ which sshd
/usr/sbin/sshd
```

Vim 설치

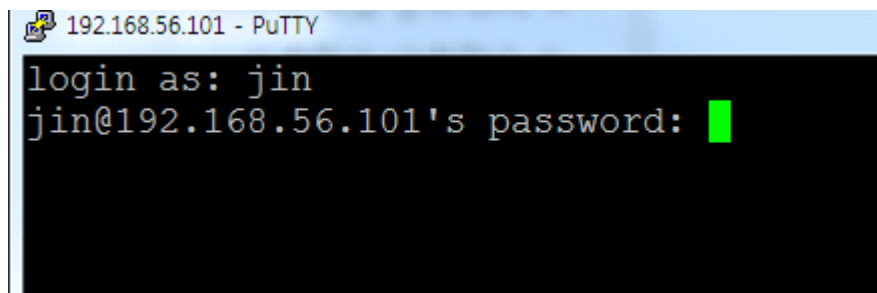
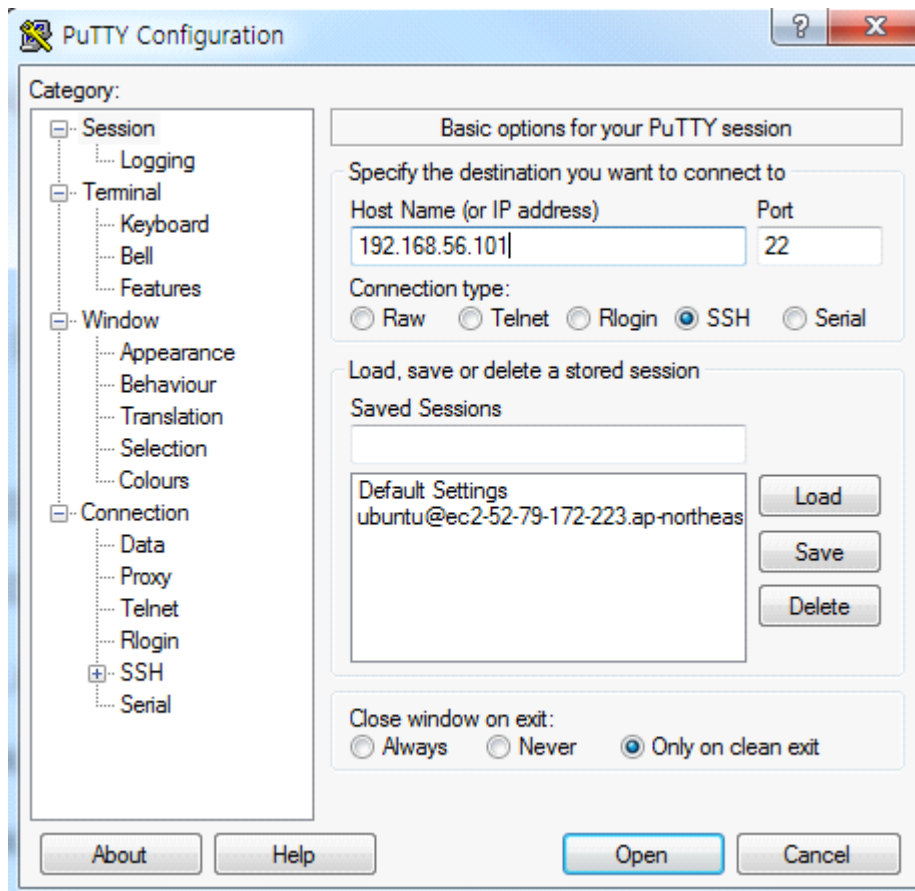
```
jln@jln-VirtualBox:~$ sudo apt-get install vim -y
```

sudo apt-get install ssh -y

sudo apt-get install vim -y

# Putty 접속

2017년 11월 21일 화요일    오후 1:41



# JDK 설치

2017년 9월 30일 토요일    오후 1:56

```
jin@jin-VirtualBox:~$ sudo add-apt-repository ppa:webupd8team/java
Oracle Java (JDK) Installer (automatically downloads and installs Or
JDK8 / JDK9). There are no actual Java files in this PPA.

Important -> Why Oracle Java 7 And 6 Installers No Longer Work: http:
d8.org/2017/06/why-oracle-java-7-and-6-installers-no.html
```

자바 저장소 추가

```
jin@jin-VirtualBox:~$ sudo apt-get update
받기:1 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease [17.5 k
B]
기존:2 http://security.ubuntu.com/ubuntu xenial-security InRelease
받기:3 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main amd64 Packag
es [2,908 B]
```

```
jin@jin-VirtualBox:~$ sudo apt-get install ora
oracle-java6-installer          oracle-java8-unlimited-jce-policy
oracle-java6-set-default        oracle-java9-installer
oracle-java7-installer          oracle-java9-set-default
oracle-java7-set-default        oracle-java9-unlimited-jce-policy
oracle-java7-unlimited-jce-policy oracle-jdk7-installer
oracle-java8-installer          orage
oracle-java8-set-default
```

위처럼 ora 입력 후 탭키를 누르면 관련 파일들이 나타난다.

```
jin@jin-VirtualBox:~$ sudo apt-get install oracle-java8-installer
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
선택된 패키지를 읽는 중입니다... 완료
```





### oracle-java8-installer 설정 중입니다

In order to install this package, you must accept the license terms, the "Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX ". Not accepting will cancel the installation.

Do you accept the Oracle Binary Code license terms?

<예>

<아니오>

### 설치 확인

```
jin@jin-VirtualBox:~$ java -version
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
```

```
jin@jin-VirtualBox:~$ sudo update-alternatives --display java
java - 수동 모드
  link best version is /usr/lib/jvm/java-8-oracle/jre/bin/java
링크가 현재 /usr/lib/jvm/java-8-oracle/jre/bin/java 가리킴
  link java is /usr/bin/java
  slave java.1.gz is /usr/share/man/man1/java.1.gz
/usr/lib/jvm/java-8-oracle/jre/bin/java - 우선순위 1081
슬레이브 java.1.gz: /usr/lib/jvm/java-8-oracle/man/man1/java.1.gz
```

단축 아이콘 설정되어 있으나 java\_home은 없음.

```
jin@jin-VirtualBox:~$ sudo vim /etc/environment
```

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin
JAVA_HOME=/usr/lib/jvm/java-8-oracle
CLASSPATH=$JAVA_HOME/lib
```

파일이 열리면 터미널을 열어 경로를 찾고 이를 복사하여 위처럼 입력한다.

```
jin@jin-VirtualBox:~$ source /etc/environment
jin@jin-VirtualBox:~$ cat $CLASSPATH
cat: /usr/lib/jvm/java-8-oracle/lib: 디렉터리입니다
```

현재 설정된 내용을 적용하기 위해 source명령을 처리한다.

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer -y
```

```
java -version
sudo update-alternatives --display java
```

```
sudo vim /etc/environment
```

```
JAVA_HOME=/usr/lib/jvm/java-8-oracle
CLASSPATH=$JAVA_HOME/lib
```

```
source /etc/environment
cat $CLASSPATH
```

## [참고]/var/lib/dpkg/lock 잠금 파일

2017년 10월 2일 월요일 오전 7:03

```
jln@jln-VirtualBox:~$ sudo apt-get install oracle-java9-installer
E: /var/lib/dpkg/lock 잠금 파일을 얻을 수 없습니다 - open (11: 자원이 일시적으로
사용 불가능함)
E: 관리 디렉터리를 (/var/lib/dpkg/) 잠글 수 없습니다. 다른 프로세스가 사용하고
있지 않습니까?
jln@jln-VirtualBox:~$ sudo rm /var/lib/apt/lists/lock
jln@jln-VirtualBox:~$ sudo rm /var/cache/apt/archives/lock
jln@jln-VirtualBox:~$ sudo apt-get install oracle-java9-installer
E: /var/lib/dpkg/lock 잠금 파일을 얻을 수 없습니다 - open (11: 자원이 일시적으로
사용 불가능함)
E: 관리 디렉터리를 (/var/lib/dpkg/) 잠글 수 없습니다. 다른 프로세스가 사용하고
있지 않습니까?
jln@jln-VirtualBox:~$ sudo rm /v
var/      vmlinuz
jln@jln-VirtualBox:~$ sudo rm /var/lib/dpkg/lock
jln@jln-VirtualBox:~$ sudo rm /var/lib/dpkg
rm: '/var/lib/dpkg'를 지울 수 없음: 디렉터리입니다
jln@jln-VirtualBox:~$ sudo apt-get install oracle-java9-installer
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
The following additional packages will be installed:
  gsfontr-x11 java-common oracle-java9-set-default
제안하는 패키지:
```


일단 위처럼 잠금 파일을 삭제하였더니 잘 진행됨 검색해 보니 많이 나옴.

# JDK9 문제

2017년 10월 2일 월요일    오후 8:28

```
jin@jin-VirtualBox:~$ cat /home/jin/eclipse-workspace/.metadata/.log
SESSION 2017-10-02 20:24:31.396 -----
eclipse.buildId=4.7.1.M20170906-1700
java.version=9
java.vendor=Oracle Corporation
Bootloader constants: OS=linux, ARCH=x86_64, WS=gtk, NL=ko_KR
|널 |ork arguments: -product org.eclipse.epp.package.jee.product
|ommand-line arguments: -os linux -ws gtk -arch x86_64 -product org.eclipse.epp
package.jee.product

ENTRY org.eclipse.osgi 4 0 2017-10-02 20:24:44.096
MESSAGE Application error
STACK 1
org.eclipse.e4.core.di.InjectionException: java.lang.NoClassDefFoundError: javax
'annotation/PostConstruct
    at org.eclipse.e4.core.internal.di.InjectorImpl.internalMake(InjectorImpl
..java:410)
    at org.eclipse.e4.core.internal.di.InjectorImpl.make(InjectorImpl.java:3
8)
    at org.eclipse.e4.core.contexts.ContextInjectionFactory.make(ContextInje
ctionFactory.java:162)
    at org.eclipse.e4.ui.internal.workbench.swt.E4Application.createDefaultH
```



DOWNLOAD    GETTING STARTED    MEMBERS    PROJECTS    MORE

**Bugzilla - Bug 493761**    Platform won't launch on Java 9-ea builds (InjectionException: NoClassDefFoundError: javax/annotation/PostConstruct)    Last modified: 2017-10-02 05:06:22

[Home](#) | [New](#) | [Browse](#) | [Search](#) |   [\[?\] | Reports](#) | [Requests](#) | [Help](#) | [Log In](#) | [Forgot Password](#) | [Terms of Use](#) | [Copyright Agent](#)

First Last Prev Next    This bug is not in your last search results.

**Bug 493761 - Platform won't launch on Java 9-ea builds (InjectionException: NoClassDefFoundError: javax/annotation/PostConstruct)**

**Status:** RESOLVED WONTFIX

**Product:** Equinox


**Component:** Launcher

**Version:** 4.6.0 Neon

**Hardware:** All All

**Importance:** P3 major ([vote](#))

**Target Milestone:** Oxygen M7

**Assigned To:** Arun Thondapu 

**QA Contact:**

**URL:**


**Whiteboard:** workaround: add vmarg --add-modules=A...

**Keywords:**

**Duplicates:** [497595](#) [516050](#) ([view as bug list](#))













**Depends on:**

**Blocks:** [493759](#) [515127](#) [516349](#) [516911](#)  
[Show dependency tree](#)

**Reported:** 2016-05-16 21:23 EDT by Wayne Beaton 

**Modified:** 2017-10-02 05:06 EDT ([History](#))

**CC List:** 29 users ([show](#))

**See Also:**  [514939](#)  
 [Gerrit Change](#)  
 [Gerrit Change](#)  
 [Git Commit](#)  
 [Git Commit](#)  
 [Git Commit](#)  
 [Gerrit Change](#)  
 [Git Commit](#)  
 [Git Commit](#)  
 [Git Commit](#)  
 [516143](#)  
 [516349](#)

2017-10-02일 현재 문제 발생



# [참고]리눅스 환경 변수 : 네이버 블로그

2017년 11월 21일 화요일 오전 9:36

환경 변수란 프로세스가 컴퓨터에서 동작하는 방식에 영향을 미치는 동적인 값들의 모임으로 셸에서 정의되고 실행하는 동안 프로그램에 필요한 변수를 나타냄.

리눅스 상에서 환경 변수를 이해하려면 동작 범위에 따른 환경 변수와 환경 변수 구성 파일 그리고 환경 변수 명령어에 대해서 알아봄.

리눅스 상에서 자바라든지 이클립스 같은 특정 프로그램에 대한 환경 변수를 설정할 때 어떤 관점에서 설정해야 할지 명확히 알 필요가 있음.

그래서 리눅스 환경 변수에 대해서 알아보자!!!

## ( 1 ) 동작 범위에 따른 환경 변수

크게 세 가지 관점으로 나눌 수 있음.

### 1. 로컬 환경 변수

현재 세션에서만 동작하는 환경 변수임.

### 2. 사용자 환경 변수

특정 사용자에게 대해서만 정의된 환경 변수로 로컬 터미널 세션 또는 원격 로그인 세션을 사용하여 로그인할 때마다 로드됨.

관련 파일은 특정 사용자의 홈 디렉토리에 존재하는 파일로 `.bashrc`, `.bash_profile`, `bash_login`, `.profile` 임.

### 3. 시스템 전체 환경 변수

해당 시스템에 존재하는 모든 사용자가 사용할 수 있는 환경 변수로 시스템 전원이 켜져 있고 모든 사용자가 로컬 또는 원격으로 로그인할 때마다 로드됨.

관련 파일은 아래와 같음.

`/etc/environment`

`/etc/profile`

`/etc/profile.d/`

`/etc/bash.bashrc`

## ( 2 ) 환경 변수 구성 파일

### `.bashrc`

특정 사용자가 새로운 로컬 세션을 생성할 때마다 로드되는 파일로 별칭(alias)과 bash 가 실행될 때 실행되는 함수 등으로 구성됨.

이 파일은 전역적인 설정 파일인 `/etc/bashrc` 이 수행된 다음 바로 수행됨.

모든 사용자에게 영향을 주는 `/etc/bashrc` 와는 달리 `~/.bashrc` 는 오직 bash 를 실행하는 그 사용자에게만 영향을 줌.

### .bash\_profile

특정 사용자의 원격 로그인 파일로 이 파일에 있는 환경 변수는 사용자가 원격 로그인 세션이 이루어질 시에 호출됨.

이 파일이 존재하지 않으면 시스템은 .bash\_login 이나 .profile 파일을 검색함.

이 파일은 전역적인 설정 파일인 /etc/profile 이 수행된 다음 바로 수행됨.

모든 사용자에게 영향을 주는 /etc/profile 과는 달리 ~/.bash\_profile 는 오직 bash 를 실행하는 그 사용자에게만 영향을 줌.

### /etc/environment

전반적인 시스템을 제어하는 파일로 필요한 환경 변수를 작성하거나 편집 또는 제거함.

이 파일에서 만든 환경 변수는 로컬 및 원격으로 접속한 모든 사용자가 액세스할 수 있음.

### /etc/bash.bashrc

시스템 전체의 bashrc 파일로 모든 사용자가 로컬 터미널 세션을 열 때마다 로드됨.

이 파일에서 만든 환경 변수는 모든 사용자가 액세스할 수 있지만 로컬 터미널 세션에서만 가능함.

### /etc/profile

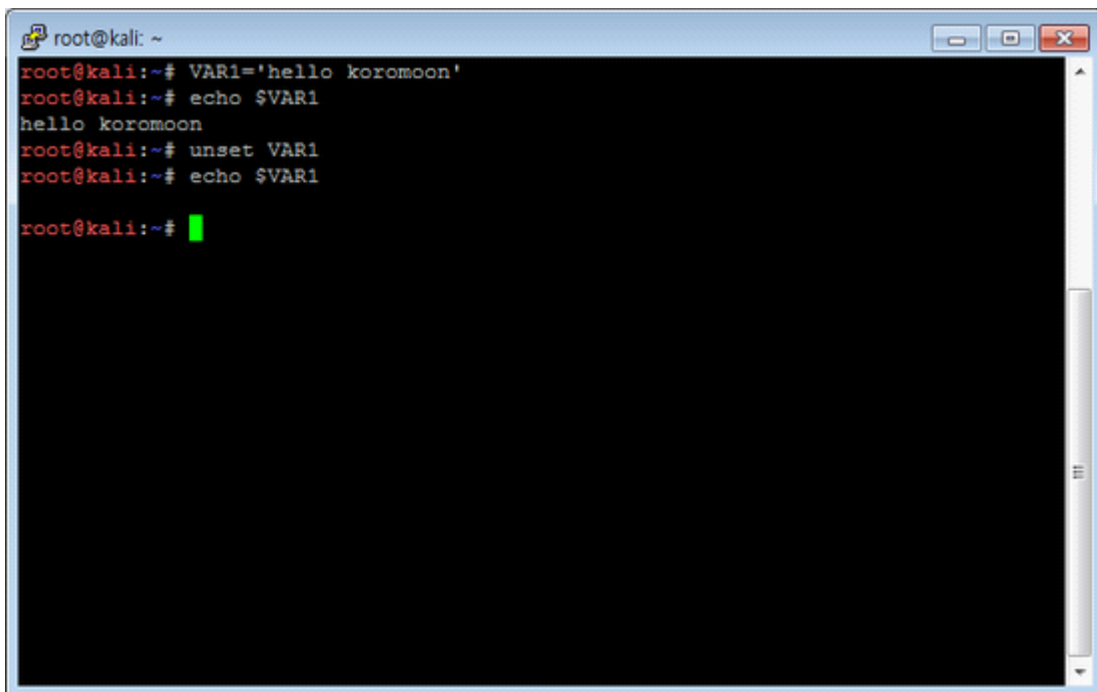
시스템 전체의 profile 파일로 모든 사용자가 원격 로그인 세션이 이루어질 시에 호출됨.

이 파일에서 만든 환경 변수는 모든 사용자가 액세스할 수 있지만 원격 로그인 세션에서만 가능함.

파일 내용을 수정 후에 재부팅/재실행 없이 즉시 적용하고자 한다면 아래 명령어와 같이 입력함.  
source [파일 이름]

## ( 3 ) 로컬 환경 변수 설정

2 가지 방법이 있음.



```
root@kali: ~
root@kali:~# VAR1='hello koromoon'
root@kali:~# echo $VAR1
hello koromoon
root@kali:~# unset VAR1
root@kali:~# echo $VAR1

root@kali:~#
```

1. 가변적인 변수명을 써서 설정함.

기본 형식 : [가변적인 변수명]=[환경변수값]

```
root@kali: ~  
root@kali:~# export VAR2='koromoon world'  
root@kali:~# echo $VAR2  
koromoon world  
root@kali:~# env -i bash  
root@kali:/root# echo $VAR2  
  
root@kali:/root# exit  
exit  
root@kali:~# echo $VAR2  
koromoon world  
root@kali:~# echo $VAR1  
hello koromoon  
root@kali:~#
```

2. export 명령어를 이용하여 환경 변수를 설정함.

기본 형식 : `export [변수명]=[환경변수값]`

여기서 주의! 변수명과 데이터값 사이에 있는 '=' 과 띄워쓰기를 하면 안됨.  
띄워쓰기를 하면 '=' 을 데이터로 인식하여 오류가 남.

입력한 환경 변수를 확인할 때는 'echo [\$변수명]' 명령을 입력해서 확인함.

로컬 환경 변수를 삭제할 때는 'unset [가변적인 변수명]' 입력하거나 'env -i bash' 명령을 입력함.

참고로 그 후에 exit 명령어를 입력하면 모든 변수명이 복원되어서 다시 사용할 수 있음.

아래는 export 명령어에 대한 추가 정보임.

export

환경변수 리스트를 보여줌

`export [변수명]=[$변수명]:[환경변수값]`

ex. `export PATH=$PATH:/home/koromoon/hack1/bin:/home/koromoon/hack2/bin`

변수명에 환경변수값을 추가함

## ( 4 ) 사용자 환경 변수 설정

.bashrc 이나 .bash\_profile 파일 하단에 'export [변수명]=[환경변수값]' 명령줄을 추가한 후 source [파일 이름] 명령어를 입력해서 설정함.

삭제할 때는 역으로 'export [변수명]=[환경변수값]' 명령줄을 삭제한 후 source [파일 이름] 명령어를 입력해서 설정함.

.bashrc (특정 사용자가 새로운 로컬 세션으로 접근 시도시)

.bash\_profile (특정 사용자가 원격 로그인 세션으로 접근 시도시)

## ( 5 ) 시스템 전체 환경 변수

/etc/bash.bashrc 이나 /etc/profile 파일 하단에 'export [변수명]=[환경변수값]' 명령줄을 추가한 후 source [파일 이름] 명령어를 입력해서 설정함.

삭제할 때는 역으로 'export [변수명]=[환경변수값]' 명령줄을 삭제한 후 source [파일 이름] 명령어를 입력해서 설정함.

.bashrc (모든 사용자가 새로운 로컬 세션으로 접근 시도시)

.bash\_profile (모든 사용자가 원격 로그인 세션으로 접근 시도시)

## ( 6 ) 일반적으로 사용되는 환경변수 목록

HOME	: 사용자의 홈디렉토리
PATH	: 실행파일을 찾는 경로
LANG	: 프로그램 사용시 기본 지원되는 언어
PWD	: 사용자의 현재 작업하는 디렉토리
TERM	: 로그인 터미널 타입
SHELL	: 로그인해서 사용하는 셸
USER	: 사용자의 이름
DISPLAY	: X 디스플레이 이름
VISUAL	: visual 편집기의 이름
EDITOR	: 기본 편집기의 이름
COLUMNS	: 현재 터미널이나 윈도우 터미널의 컬럼수
PS1	: 명령프롬프트변수
PS2	: 2차 명령프롬프트임. 명령행에서 사용하여 명령행을 연장했을 때 나타냄.
BASH	: 사용하는 bash 셸의 경로
BASH_VERSION	: bash의 버전
HISTFILE	: history 파일의 경로
HISTFILESIZE	: history 파일의 크기
HISTSIZE	: history에 저장되는 갯수
HISTCONTROL	: 중복되어지는 명령에 대한 기록 유무를 지정하는 변수
HOSTNAME	: 호스트의 이름
LINES	: 터미널의 라인 수
LOGNAME	: 로그인이름
LS_COLORS	: ls 명령의 색상관련 옵션
MAIL	: 메일을 보관하는 경로
MAILCHECK	: 메일확인시간
OSTYPE	: 운영체제 타입
SHLVL	: 셸의 레벨
TERM	: 터미널종류
UID	: 사용자의 UID
USERNAME	: 사용자이름

## ( 7 ) 환경변수 확인 명령어

set 명령어 : 로컬 환경변수 조회 명령어

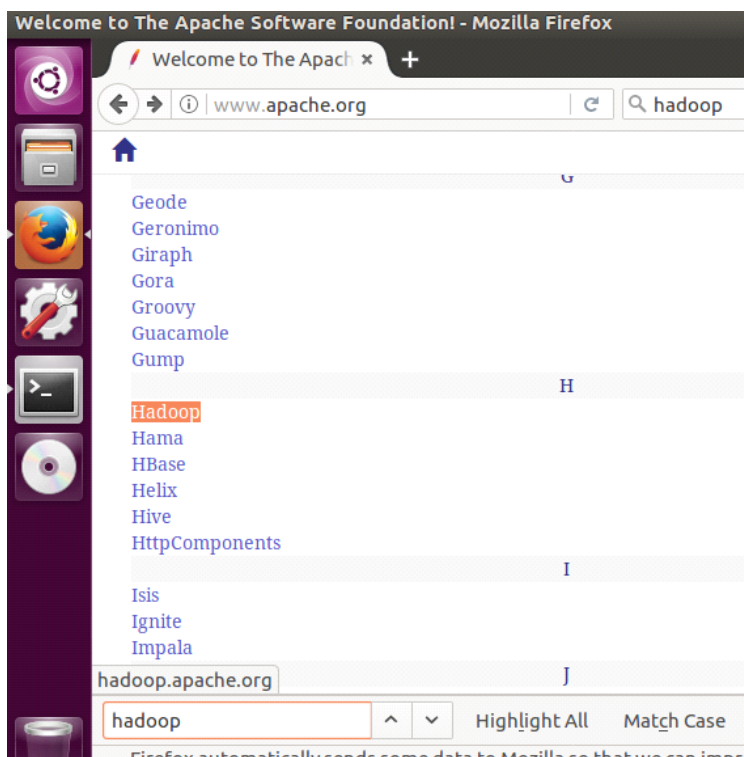
env 명령어 : 글로벌 환경변수 조회 명령어

<<http://blog.naver.com/koromoon/220793570727>>에서 삽입

# 하둡 설치

2017년 9월 30일 토요일 오후 3:02

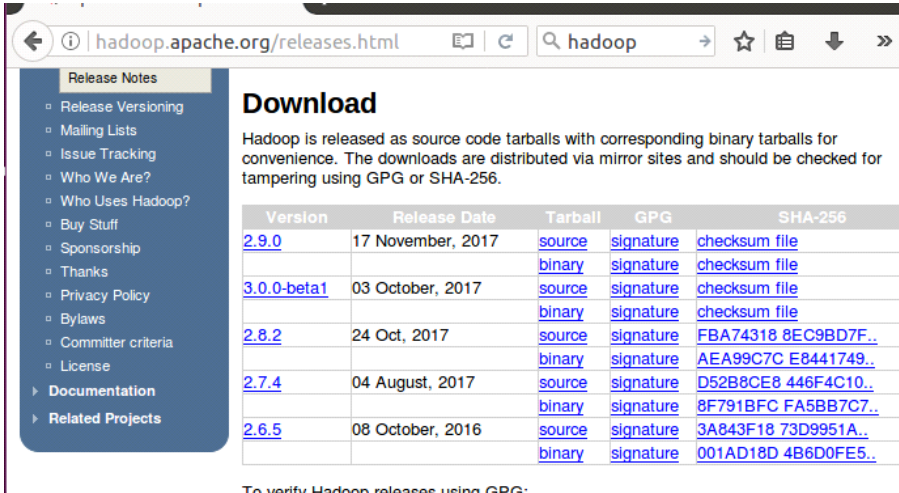
<http://www.apache.org/>



Ctrl + F로 hadoop을 검색 하면 위처럼 나타난다.



위에서 Releases를 클릭한다.



To verify Hadoop releases using GPG:

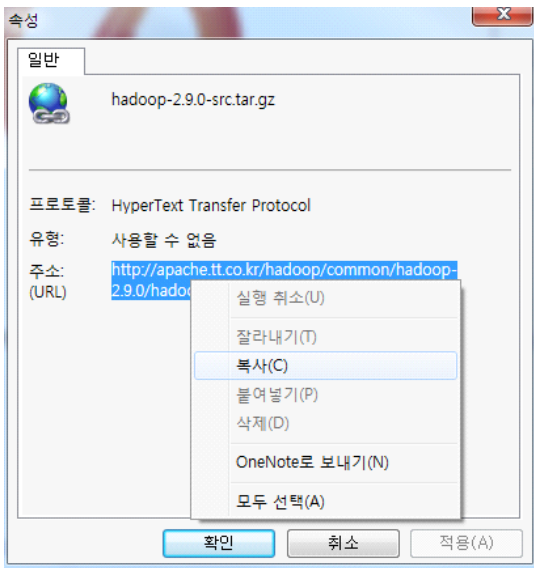
3.0.0은 beta임으로 2.9 설치

binary를 클릭하면 다음과 같은 화면으로 넘어간다.



첫 번째 나타나는 정보가 추천 사이트 이고 HTTP 아래 부분에 나타난 정보가 Hadoop 2.9.0정보를 가지고 있는 사이트 이다.

첫 번째 사이트에서 마우스 오른쪽 클릭하여 속성을 선택한다.



이렇게 나타난 정보에서 주소를 복사하자

```
jin@jin-VirtualBox:~$ wget http://mirror.apache-kr.org/hadoop/common/hadoop-2.8.2/hadoop-2.8.2.tar.gz
```

```
jin@jin-VirtualBox:~$ tar zxvf hadoop-2.9.0.tar.gz
```

```
jin@jin-VirtualBox:~$ sudo mkdir -p /usr/local/hadoop
```

```
jin@jin-VirtualBox:~$ ls
examples.desktop  hadoop-2.9.0.tar.gz  다운로드  바탕화면  사진
hadoop-2.9.0      공개                 문서      비디오    음악
jin@jin-VirtualBox:~$ cd hadoop-2.9.0/
jin@jin-VirtualBox:~/hadoop-2.9.0$ sudo mv * /usr/local/hadoop/
```

```
jin@jin-VirtualBox:~/hadoop-2.9.0$ ls
jin@jin-VirtualBox:~/hadoop-2.9.0$ cd ..
jin@jin-VirtualBox:~$ rm -rf ha*
jin@jin-VirtualBox:~$ ls
examples.desktop  공개  다운로드  문서
```

```
wget http://mirror.apache-kr.org/hadoop/common/hadoop-2.9.0/hadoop-2.9.0.tar.gz
tar zxvf hadoop-2.9.0.tar.gz
sudo mkdir -p /usr/local/hadoop
cd hadoop-2.9.0/
sudo mv * /usr/local/hadoop/
cd
rm -rf ha*
```

# 사용자 등록 및 하둡 소유권 변경

2017년 11월 22일 수요일    오후 1:12

하둡을 관리할 그룹과 계정을 만든다.

추후 데이터 처리를 해야 함으로 sudo 권한도 부여한다.

```
jin@jin-VirtualBox:~$ sudo addgroup hadoop
그룹 'hadoop' (GID 1001) 추가 ...
완료.
jin@jin-VirtualBox:~$ sudo adduser --ingroup hadoop manager
'manager' 사용자를 추가 중...
새 사용자 'manager' (1001) 을(를) 그룹 'hadoop' (으)로 추가 ...
'/home/manager' 홈 디렉터리를 생성하는 중...
'/etc/skel'에서 파일들을 복사하는 중...
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
```

```
jin@jin-VirtualBox:~$ groups manager
manager : hadoop
jin@jin-VirtualBox:~$ sudo adduser manager sudo
[sudo] password for jin:
'manager' 사용자를 'sudo' 그룹에 추가 중...
사용자 manager을(를) sudo 그룹에 등록 중
완료.
```

```
jin@jin-VirtualBox:~$ sudo chown -R manager:hadoop /usr/local/hadoop/
jin@jin-VirtualBox:~$ ls -l /usr/local
합계 36
drwxr-xr-x 2 root    root    4096  8월   1  2017 bin
drwxr-xr-x 2 root    root    4096  8월   1  2017 etc
drwxr-xr-x 2 root    root    4096  8월   1  2017 games
drwxr-xr-x 9 manager hadoop 4096  2월   9 16:20 hadoop
drwxr-xr-x 2 root    root    4096  8월   1  2017 include
drwxr-xr-x 4 root    root    4096  8월   1  2017 lib
lrwxrwxrwx 1 root    root      9 11월 21 09:22 man -> share/man
drwxr-xr-x 2 root    root    4096  8월   1  2017 sbin
drwxr-xr-x 8 root    root    4096  8월   1  2017 share
drwxr-xr-x 2 root    root    4096  8월   1  2017 src
```

```
sudo addgroup hadoop
sudo adduser --ingroup hadoop manager
groups manager
```

```
sudo adduser manager sudo
```

```
sudo chown -R manager:hadoop /usr/local/hadoop/
ls -l /usr/local
```



패스워드 1111

# Static IP 설정

2017년 11월 21일 화요일    오후 2:42

```
enp0s8    Link encap:Ethernet  HWaddr 08:00:27:69:22:d0
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:
          inet6 addr: fe80::6d4e:23b5:2b11:93/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:344 errors:0 dropped:0 overruns:0 frame:0
          TX packets:215 errors:0 dropped:0 overruns:0 carrier:
          collisions:0 txqueuelen:1000
          RX bytes:49269 (49.2 KB)  TX bytes:36088 (36.0 KB)
```

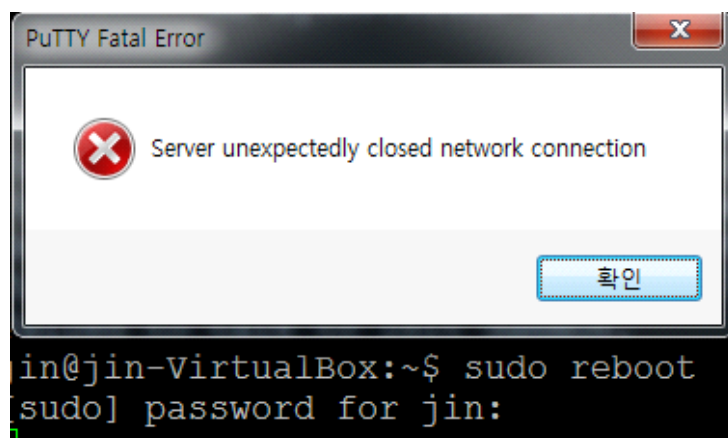
고정으로 둘 LAN 카드 확인

```
jin@jin-VirtualBox:~$ sudo vim /etc/network/interfaces
```

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto enp0s8
iface enp0s8 inet static
address 192.168.56.101
netmask 255.255.255.0
gateway 192.168.56.1
dns-nameservers 168.126.63.1
```

```
jin@jin-VirtualBox:~$ sudo /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
```



```
login as: jin
jin@192.168.56.101's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)

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

패키지 242개를 업데이트할 수 있습니다.
120 업데이트는 보안 업데이트입니다.

Last login: Tue Nov 21 14:45:03 2017 from 192.168.56.1
jin@jin-VirtualBox:~$
```

Putty 재시작

Ifconfig

```
sudo vim /etc/network/interfaces
```

```
auto enp0s8
iface enp0s8 inet static
address 192.168.56.101
netmask 255.255.255.0
gateway 192.168.56.1
dns-nameservers 168.126.63.1
```

```
sudo /etc/init.d/networking restart
```

위와 같은 방법으로 나머지 3개의 서버 IP 변경

# ~/.bashrc 수정

2017년 9월 30일 토요일    오후 3:21

```
manager@master:~$ sudo vim .bashrc
```

```
#HADOOP VARIABLES START
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#HADOOP VARIABLES END
```

sudo vim .bashrc

```
#HADOOP VARIABLES START
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
#HADOOP VARIABLES END
```

source .bashrc

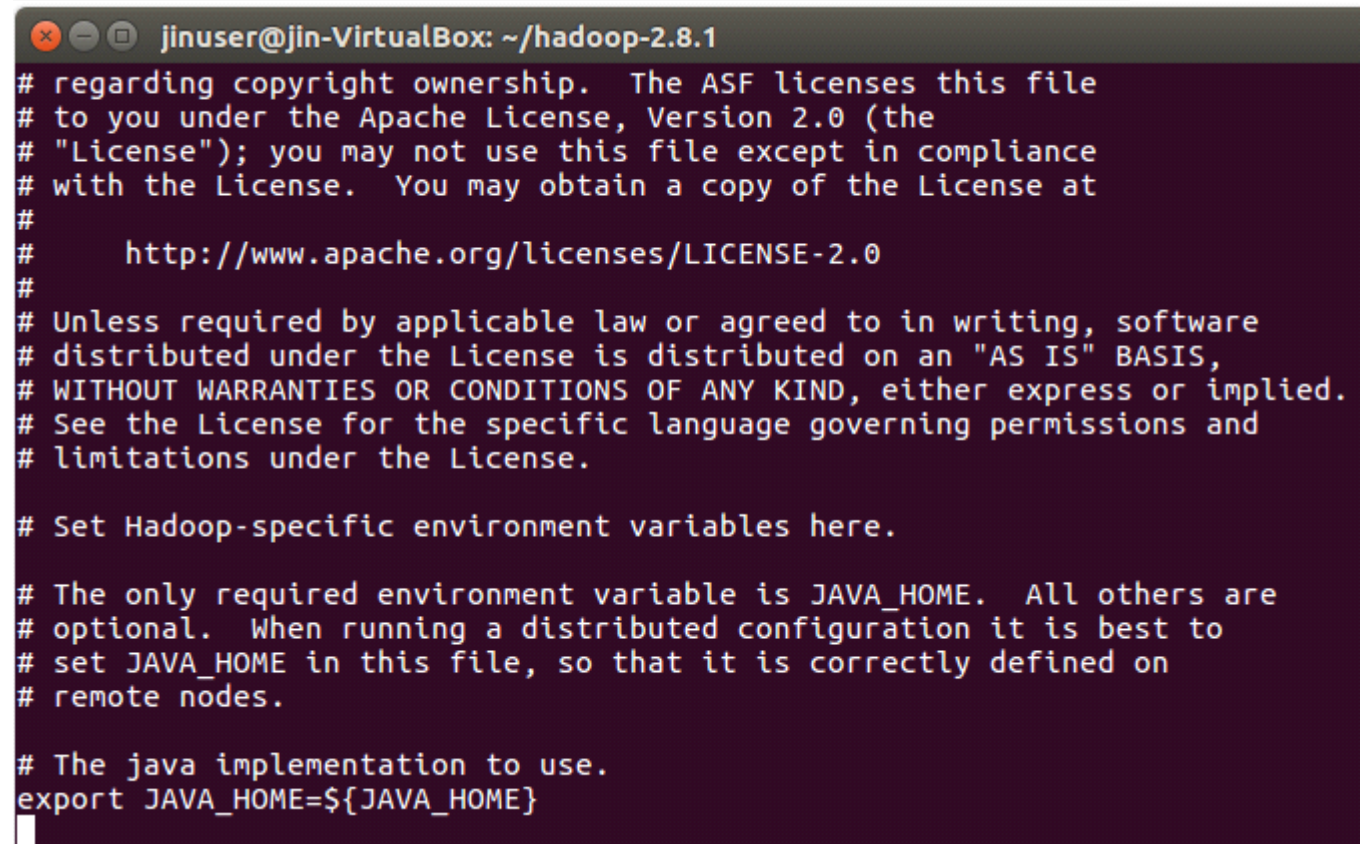
아래 정보는 필요하면 추가

```
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#HADOOP VARIABLES END
```

# java\_home 확인

2017년 9월 30일 토요일    오후 3:24

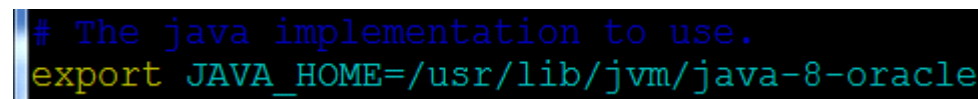
```
hduser@laptop:~$ vi /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```



```
jinuser@jin-VirtualBox: ~/hadoop-2.8.1
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Set Hadoop-specific environment variables here.
#
# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.
#
# The java implementation to use.
export JAVA_HOME=${JAVA_HOME}
```

JAVA\_HOME 이 설정되어 있음으로 별도의 설정은 필요 없음.

JAVA\_HOME이 없다고 나타나는 경우 아래와 같이 수정



```
# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

## core-site

2017년 9월 30일 토요일    오후 3:30

### /usr/local/hadoop/etc/hadoop/core-site.xml:

```
manager@master:~$ sudo mkdir -p /usr/local/hadoop/tmp
```

```
manager@master:~$ sudo chown -R manager:hadoop /usr/local/hadoop/tmp/  
manager@master:~$ sudo vim /usr/local/hadoop/etc/hadoop/core-site.xml
```

```
<configuration>  
  <property>  
    <name>hadoop.tmp.dir</name>  
    <value>/usr/local/hadoop/tmp</value>  
  </property>  
  
  <property>  
    <name>fs.default.name</name>  
    <value>hdfs://localhost:62350</value>  
  </property>  
</configuration>
```

1. Hadoop.tmp.dir  
하둡에서 발생하는 임시 데이터를 저장하는 공간
2. Fs.default.name  
HDFS의 기본 이름  
URI 형태로 사용  
기본값은 "[file://](#)"로 돼 있지만 실제로 "HDFS://"형식으로 사용됨  
데이터노드는 여러 작업을 진행하기 위해 반드시 네임노드의 주소를 알고 있어야 함(결국 slave에도 위와 같은 내용이 배포되어야 됨.

```
sudo mkdir -p /usr/local/hadoop/tmp  
sudo chown -R manager:hadoop /usr/local/hadoop/tmp/  
sudo vim /usr/local/hadoop/etc/hadoop/core-site.xml
```

```
<property>  
  <name>hadoop.tmp.dir</name>  
  <value>/usr/local/hadoop/tmp</value>  
</property>  
  
<property>  
  <name>fs.default.name</name>  
  <value>hdfs://master:62350</value>  
</property>
```

## mapred-site

2017년 9월 30일 토요일    오후 6:17

### /usr/local/hadoop/etc/hadoop/mapred-site.xml

```
master@master:~$ cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
master@master:~$ sudo vim /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

템플릿 복사

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>master:62351</value>
  </property>
</configuration>
```

#### 1. Mapred.job.tracker

JobTracker 데몬의 주소를 의미

데이터노드에서 이 주소를 보고 맵리듀스 작업을 요청(결국 이 부분도 Slave에 설정해야 됨)

```
sudo cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/etc/hadoop/mapred-site.xml
```

```
sudo vim /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

```
<property>
  <name>mapred.job.tracker</name>
  <value>master:62351</value>
</property>
```

# Master 또는 slave hadoop 저장

2017년 11월 27일 월요일 오전 9:34

```
manager@jin-VirtualBox:~$ tar -cvf hadoop.tar /usr/local/hadoop
```

```
manager@jin-VirtualBox:~$ sudo reboot
```

tar -cvf hadoop.tar /usr/local/hadoop  
sudo reboot





=== 완전 분산 모드 ===

2018년 2월 9일 금요일      오후 4:01

# Master 설정

2018년 2월 9일 금요일    오후 4:02

# Hosts 파일 변경

2017년 11월 21일 화요일    오후 1:42

```
jin@jin-VirtualBox:~$ sudo vim /etc/hosts
```

```
127.0.0.1      localhost
127.0.1.1      jin-VirtualBox

192.168.56.101 master
192.168.56.102 slave1
192.168.56.103 slave2
192.168.56.104 slave3

# The following lines are desirable for IPv6 capable hosts
::1           ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
```

sudo vim /etc/hosts

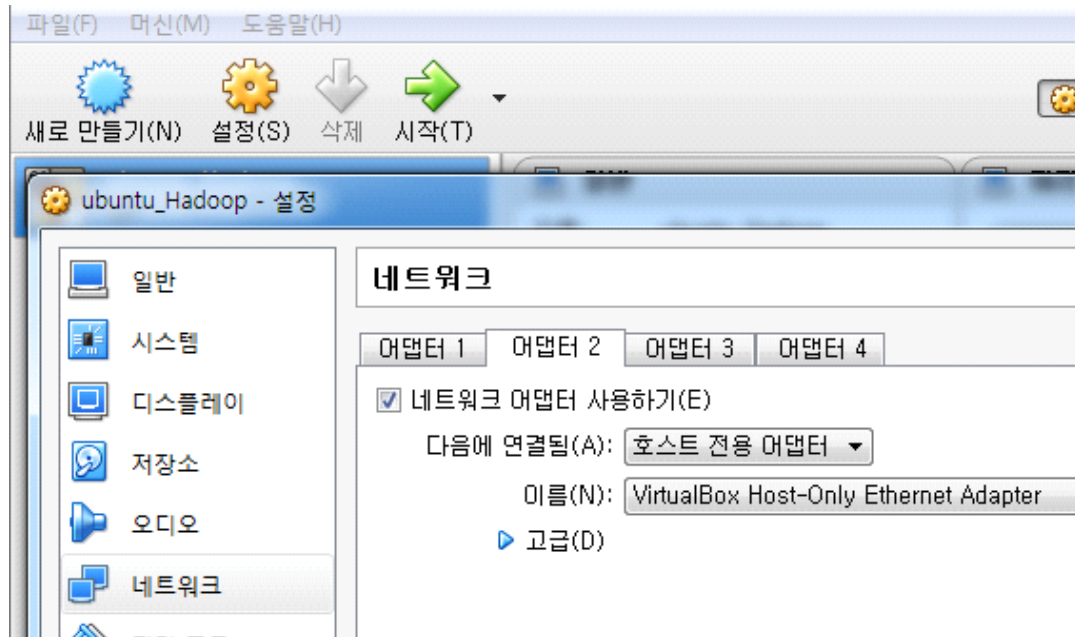
```
192.168.56.101 master
192.168.56.102 slave1
192.168.56.103 slave2
192.168.56.104 slave3
```

# Slave 만들기

2017년 9월 30일 토요일    오후 3:20

# 네트워크 설정

2017년 10월 3일 화요일    오후 3:13



1. 톱니바퀴 모양의 설정 클릭
2. 설정창에서 네트워크 클릭
3. 어댑터 2 선택하여 호스트 전용 어댑터 선택

```
jin@jin-VirtualBox:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:71:99:21
        inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::68be:903b:1eb1:1c58/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:10  errors:0  dropped:0  overruns:0  frame:0
        TX packets:65  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:2079 (2.0 KB)  TX bytes:7550 (7.5 KB)

enp0s8  Link encap:Ethernet  HWaddr 08:00:27:48:cd:d0
        inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
        inet6 addr: fe80::d1d2:1e73:2246:575/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:8  errors:0  dropped:0  overruns:0  frame:0
        TX packets:58  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:1559 (1.5 KB)  TX bytes:7485 (7.4 KB)
```

위와 같이 진행하면 두 개의 랜카드가 설정되었던 것을 확인 할 수 있다. 첫 번째는 외부 인터넷을 하기 위한 NAT이고 두 번째는 내부 통신을 위한 호스트 전용 이다.

내부에서만 처리할 것이기 때문에 호스트 전용의 IP를 사용할 것이다.

```
jin@jin-VirtualBox:~$ sudo ifconfig enp0s8 192.168.56.102 up
```

# 유틸 설치

## Ssh 설치

```
jln@jln-VirtualBox:~$ sudo apt-get install ssh
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
The following additional packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
제안하는 패키지:
  ssh-askpass rssh molly-guard monkeysphere
다음 새 패키지를 설치할 것입니다:
```

```
jln@jln-VirtualBox:~$ which ssh
/usr/bin/ssh
jln@jln-VirtualBox:~$ which sshd
/usr/sbin/sshd
```

## Vim 설치

```
jln@jln-VirtualBox:~$ sudo apt-get install vim -y
```

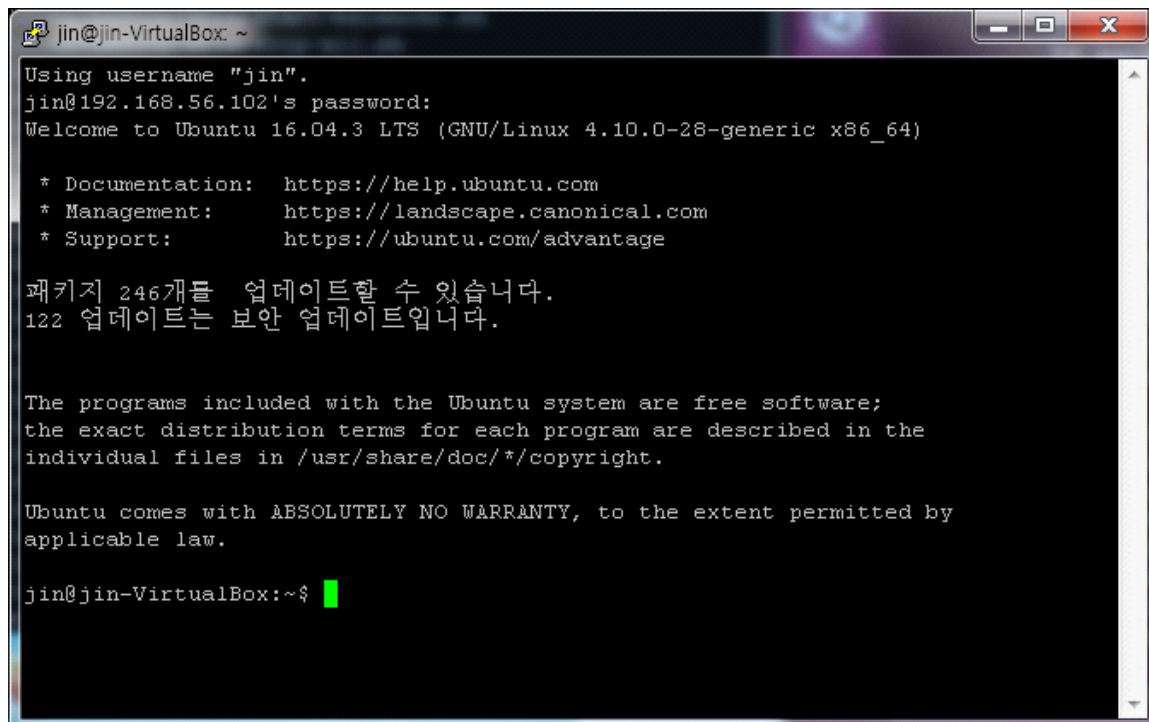
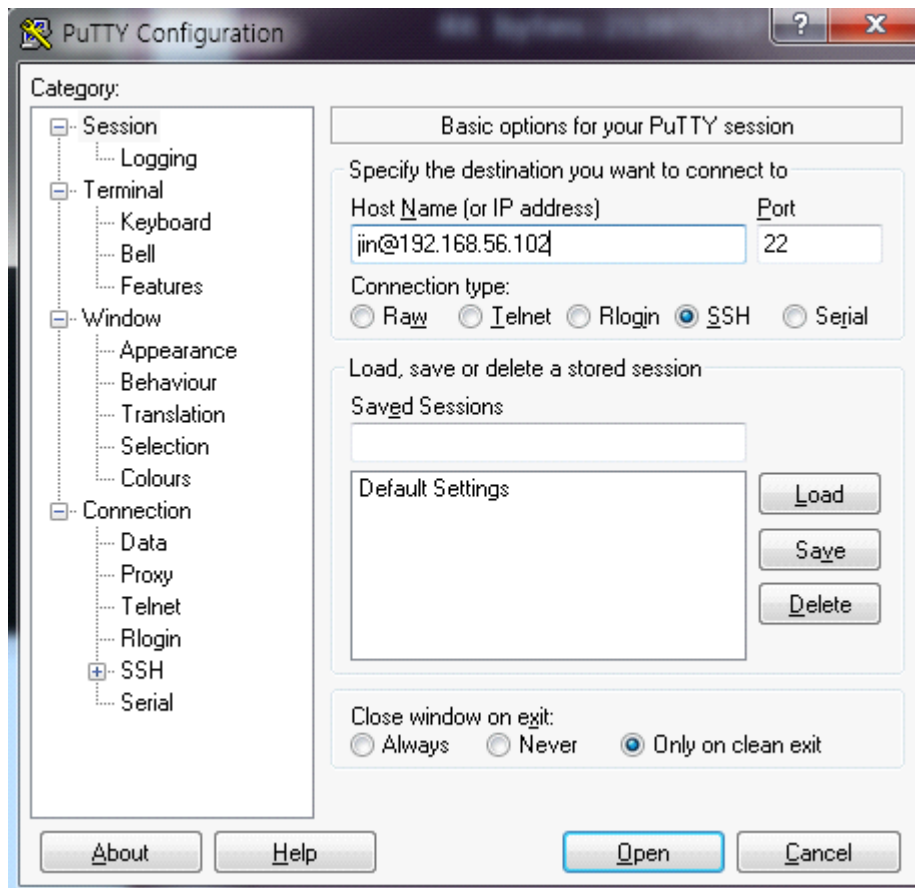
```
sudo apt-get install ssh -y
```

```
sudo apt-get install vim -y
```



# Putty 접속

2017년 11월 21일 화요일    오후 1:41



# JDK 설치

2017년 9월 30일 토요일    오후 1:56

```
jin@jin-VirtualBox:~$ sudo add-apt-repository ppa:webupd8team/java
Oracle Java (JDK) Installer (automatically downloads and installs Oracle JDK7
JDK8 / JDK9). There are no actual Java files in this PPA.

Important -> Why Oracle Java 7 And 6 Installers No Longer Work: http://www.web
d8.org/2017/06/why-oracle-java-7-and-6-installers-no.html

Ubuntu 16.10 Yakkety Yak is no longer supported by Canonical (and thus, Launch
d and this PPA). The PPA supports Ubuntu 17.10, 17.04, 16.04, 14.04 and 12.04.

More info (and Ubuntu installation instructions):
- for Oracle Java 7: http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-
-ubuntu-via.html
- for Oracle Java 8: http://www.webupd8.org/2012/09/install-oracle-java-8-in-u
ntu-via-ppa.html
```

자바 저장소 추가

```
jin@jin-VirtualBox:~$ sudo apt-get update
기존:1 http://kr.archive.ubuntu.com/ubuntu xenial InRelease
기존:2 http://kr.archive.ubuntu.com/ubuntu xenial-updates InRelease
기존:3 http://kr.archive.ubuntu.com/ubuntu xenial-backports InRelease
받기:4 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease
받기:5 http://security.ubuntu.com/ubuntu xenial-security InRelease
받기:6 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease
받기:7 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease
```

```
jin@jin-VirtualBox:~$ sudo apt-get install ora
oracle-java6-installer
oracle-java6-set-default
oracle-java7-installer
oracle-java7-set-default
oracle-java7-unlimited-jce-policy
oracle-java8-installer
oracle-java8-set-default
oracle-java8-unlimited-jce-policy
oracle-java9-installer
oracle-java9-set-default
oracle-java9-unlimited-jce-policy
oracle-jdk7-installer
orage
```

위처럼 ora 입력 후 탭키를 누르면 관련 파일들이 나타난다.

```
jin@jin-VirtualBox:~$ sudo apt-get install oracle-java8-installer
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
선택된 패키지를 읽는 중입니다... 완료
```

## oracle-java8-installer 설정 중입니다

Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX

You MUST agree to the license available in <http://java.com/license> if you want to use Oracle JDK.

<확인>

## oracle-java8-installer 설정 중입니다

In order to install this package, you must accept the license terms, the "Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX ". Not accepting will cancel the installation.

Do you accept the Oracle Binary Code license terms?

<예>

<아니오>

## 설치 확인

```
jjin@jjin-VirtualBox:~$ java -version
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
```

```
jjin@jjin-VirtualBox:~$ sudo update-alternatives --display java
java - 수동 모드
  link best version is /usr/lib/jvm/java-8-oracle/jre/bin/java
  링크가 현재 /usr/lib/jvm/java-8-oracle/jre/bin/java 가리킴
  link java is /usr/bin/java
  slave java.1.gz is /usr/share/man/man1/java.1.gz
/usr/lib/jvm/java-8-oracle/jre/bin/java - 우선순위 1081
  슬레이브 java.1.gz: /usr/lib/jvm/java-8-oracle/man/man1/java.1.gz
```

단축 아이콘 설정되어 있으나 java\_home은 없음.

```
jjin@jjin-VirtualBox:~$ sudo vim /etc/environment
```

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
JAVA_HOME=/usr/lib/jvm/java-8-oracle
CLASSPATH=$JAVA_HOME/lib
```

파일이 열리면 터미널을 열어 경로를 찾고 이를 복사하여 위처럼 입력한다.

```
jin@jin-VirtualBox:~$ source /etc/environment
jin@jin-VirtualBox:~$ cat $CLASSPATH
cat: /usr/lib/jvm/java-8-oracle/lib: 디렉터리입니다
```

현재 설정된 내용을 적용하기 위해 source명령을 처리한다.

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer -y
```

```
java -version
sudo update-alternatives --display java
```

```
sudo vim /etc/environment
```

```
JAVA_HOME=/usr/lib/jvm/java-8-oracle
CLASSPATH=$JAVA_HOME/lib
```

```
source /etc/environment
cat $CLASSPATH
```

# Static IP 설정

2017년 11월 21일 화요일    오후 2:42

```
enp0s8    Link encap:Ethernet  HWaddr 08:00:27:69:22:d0
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:
          inet6 addr: fe80::6d4e:23b5:2b11:93/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:344 errors:0 dropped:0 overruns:0 frame:0
          TX packets:215 errors:0 dropped:0 overruns:0 carrier:
          collisions:0 txqueuelen:1000
          RX bytes:49269 (49.2 KB)  TX bytes:36088 (36.0 KB)
```

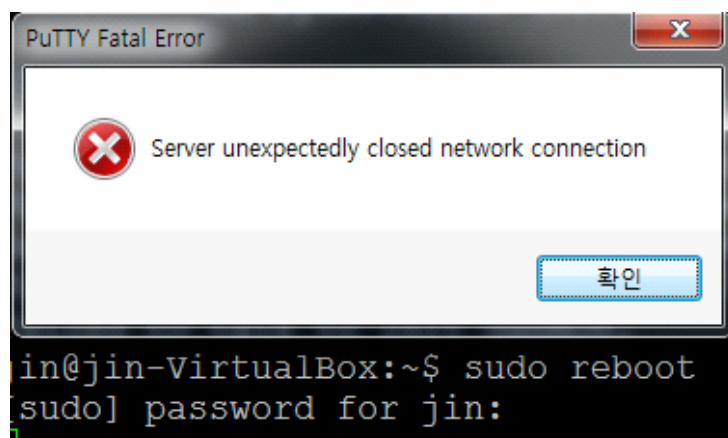
고정으로 둘 LAN 카드 확인

```
jin@jin-VirtualBox:~$ sudo vim /etc/network/interfaces
```

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto enp0s8
iface enp0s8 inet static
address 192.168.56.101
netmask 255.255.255.0
gateway 192.168.56.1
dns-nameservers 168.126.63.1
```

```
jin@jin-VirtualBox:~$ sudo /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
```



```
login as: jin
jin@192.168.56.101's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)

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

패키지 242개를 업데이트할 수 있습니다.
120 업데이트는 보안 업데이트입니다.

Last login: Tue Nov 21 14:45:03 2017 from 192.168.56.1
jin@jin-VirtualBox:~$
```

```
jin@jin-VirtualBox:~$ sudo reboot
```

Ifconfig

sudo vim /etc/network/interfaces

```
auto enp0s8
iface enp0s8 inet static
address 192.168.56.101
netmask 255.255.255.0
gateway 192.168.56.1
dns-nameservers 168.126.63.1
```

sudo reboot

위와 같은 방법으로 나머지 3개의 서버 IP 변경



# 사용자 등록

2017년 11월 22일 수요일    오후 1:12

하둡 실행 시 datanode를 마스터와 같은 계정으로 접속을 시도함으로 계정은 통일해야 한다.  
(별도의 설정 위치는 모름)

```
jin@jin-VirtualBox:~$ sudo addgroup hadoop
[sudo] password for jin:
그룹 'hadoop' (GID 1001) 추가 ...
완료.
jin@jin-VirtualBox:~$ sudo adduser --ingroup hadoop manager
'manager' 사용자를 추가 중...
새 사용자 'manager' (1001) 을(를) 그룹 'hadoop' (으)로 추가 ...
'/home/manager' 홈 디렉터리를 생성하는 중...
'/etc/skel'에서 파일들을 복사하는 중...
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
manager의 사용자의 정보를 바꿉니다
새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요
이름 []:
방 번호 []:
직장 전화번호 []:
집 전화번호 []:
기타 []:
정보가 올바릅니까? [Y/n]
jin@jin-VirtualBox:~$ groups manager
manager : hadoop
jin@jin-VirtualBox:~$ sudo adduser manager sudo
'manager' 사용자를 'sudo' 그룹에 추가 중...
사용자 manager을(를) sudo 그룹에 등록 중
완료.
```

```
sudo addgroup hadoop
sudo adduser --ingroup hadoop manager
groups manager
```

```
sudo adduser manager sudo
```

패스워드 1111



# Master 설정

2017년 11월 27일 월요일    오후 4:34

# Masters 및 slaves 파일 수정

2017년 11월 22일 수요일    오전 10:36

Slaves파일은 데이터노드를 설정하기 위한 파일이다.

```
master@master:~$ sudo vim /usr/local/hadoop/etc/hadoop/slaves
```

```
slave1  
slave2  
slave3
```

```
sudo vim /usr/local/hadoop/etc/hadoop/slaves
```

```
slave1
```

```
slave2
```

```
slave3
```

# hdfs-site

2017년 9월 30일 토요일    오후 6:21

## /usr/local/hadoop/etc/hadoop/hdfs-site.xml

```
manager@master:~$ sudo mkdir -p /usr/local/hadoop/hdfs/namenode
manager@master:~$ sudo chown -R manager:hadoop /usr/local/hadoop/hdfs

manager@master:~$ sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.http.address</name>
    <value>master:50070</value>
  </property>
  <property>
    <name>dfs.secondary.http.address</name>
    <value>slave1:50090</value>
  </property>
</configuration>
```

1. dfs.replication  
HDFS에 저장할 데이터의 복제본 개수 지정  
1일 경우 가상분산 모드를 의미함
2. dfs.namenode.name.dir  
Namenode의 위치 지정
3. dfs.http.address  
Namenode의 위치 지정
4. dfs.secondary.http.address  
보조 Namenode의 위치 지정

```
sudo mkdir -p /usr/local/hadoop/hdfs/namenode
sudo chown -R manager:hadoop /usr/local/hadoop/hdfs
```

```
sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/usr/local/hadoop/hdfs/namenode</value>
</property>
<property>
  <name>dfs.http.address</name>
  <value>master:50070</value>
</property>
```

```
<property>  
  <name>dfs.secondary.http.address</name>  
  <value>slave1:50090</value>  
</property>
```

# 인증키 생성 및 배포

2017년 10월 1일 일요일    오후 12:59

```
master@master:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/master/.ssh/id_rsa):
Created directory '/home/master/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/master/.ssh/id_rsa.
Your public key has been saved in /home/master/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:xtl5OP2Oay2LNdymbe55ApM1X+XIDpLU10XicJlLFac master@master
The key's randomart image is:
+---[RSA 2048]---+
|      ...EO+|
|      . .+*O.|
|      . . +.+.|
|      . oo+. * o|
|      S =.O= o.|
|      .  ++o. .|
|      +++   |
|      ooB+..|
|      ..=B*O |
+-----[SHA256]-----+
```

```
manager@master:~$ ssh-copy-id -i .ssh/id_rsa.pub manager@slave1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_rsa.pub"
The authenticity of host 'slave1 (192.168.56.102)' can't be established.
ECDSA key fingerprint is SHA256:CR2ZcVni0W+B8hD3yBIR5npVLucsoF9NBmWEja+qHvI.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
lled
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
w keys
manager@slave1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'manager@slave1'"
and check to make sure that only the key(s) you wanted were added.
```

```
manager@master:~$ ssh manager@slave1
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)

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

패키지 243개를 업데이트할 수 있습니다.
121 업데이트는 보안 업데이트입니다.
```

```
ssh-keygen -t rsa
ssh-copy-id -i .ssh/id_rsa.pub manager@slave1
ssh-copy-id -i .ssh/id_rsa.pub manager@slave2
```

```
ssh-copy-id -i .ssh/id_rsa.pub manager@slave3  
ssh-copy-id -i .ssh/id_rsa.pub manager@master
```

# 배포하기

2017년 11월 27일 월요일    오후 4:59



# 하둡 배포하기1(master)

2017년 11월 27일 월요일 오전 10:03

```
manager@jin-VirtualBox:~$ scp hadoop.tar manager@slave1:/home/manager
hadoop.tar                                22% 146MB 67.7MB/s
```

```
manager@jin-VirtualBox:~$ ssh manager@slave1 "cd /home/manager;tar xf hadoop.tar
;rm hadoop.tar"
```

scp hadoop.tar manager@slave1:/home/manager

ssh manager@slave1 "cd /home/manager;tar xf hadoop.tar;rm hadoop.tar"

# 하둡 배포하기2(slave)

2017년 11월 27일 월요일 오전 10:03

```
manager@jin-VirtualBox:~$ sudo mkdir /usr/local/hadoop
[sudo] password for manager:
manager@jin-VirtualBox:~$ cd usr/local/hadoop/
manager@jin-VirtualBox:~/usr/local/hadoop$ sudo mv * /usr/local/hadoop/
manager@jin-VirtualBox:~/usr/local/hadoop$ sudo chown -R manager:hadoop /usr/local/hadoop/
```

sudo mkdir /usr/local/hadoop

cd usr/local/hadoop/

sudo mv \* /usr/local/hadoop/

sudo chown -R manager:hadoop /usr/local/hadoop/

# slave 설정

2017년 11월 27일 월요일    오후 4:34

# ~/.bashrc 수정

2017년 9월 30일 토요일    오후 3:21

```
manager@master:~$ sudo vim .bashrc
```

```
#HADOOP VARIABLES START
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#HADOOP VARIABLES END
```

sudo vim .bashrc

```
#HADOOP VARIABLES START
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
#HADOOP VARIABLES END
```

source .bashrc

아래 정보는 필요하면 추가

```
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#HADOOP VARIABLES END
```

# hdfs-site

2017년 9월 30일 토요일    오후 6:21

## /usr/local/hadoop/etc/hadoop/hdfs-site.xml

```
manager@master:~$ sudo mkdir -p /usr/local/hadoop/hdfs/namenode
manager@master:~$ sudo chown -R manager:hadoop /usr/local/hadoop/hdfs
manager@master:~$ sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.http.address</name>
    <value>master:50070</value>
  </property>
  <property>
    <name>dfs.secondary.http.address</name>
    <value>slave1:50090</value>
  </property>
```

1. dfs.replication  
HDFS에 저장할 데이터의 복제본 개수 지정  
1일 경우 가상분산 모드를 의미함
2. dfs.namenode.name.dir  
Namenode의 위치 지정
3. dfs.http.address  
Namenode의 위치 지정
4. dfs.secondary.http.address  
보조 Namenode의 위치 지정

```
sudo mkdir -p /usr/local/hadoop/hdfs/datanode
sudo chown -R manager:hadoop /usr/local/hadoop/hdfs
sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/usr/local/hadoop/hdfs/datanode</value>
</property>
```

# 프롬프트 변경

2017년 9월 30일 토요일    오후 2:38

프롬프트의 jin-VirtualBox를 master로 변경하기 위한 작업이다. 이는 추후 salve 접속을 구분짓기 위한 것이다.

```
jin@jin-VirtualBox:~$ sudo vim /etc/hostname
jin@jin-VirtualBox:~$ sudo vim /etc/hosts
```

위처럼 변경하고 재부팅한다.

```
login as: jin
jin@192.168.56.101's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-62-generic x86_64)

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

패키지 242개를 업데이트할 수 있습니다.
120 업데이트는 보안 업데이트입니다.

Last login: Tue Nov 21 14:56:23 2017 from 192.168.56.101
jin@master:~$
```

```
sudo vim /etc/hostname
sudo vim /etc/hosts
```

# 하둡 파일 시스템 포맷하기

2017년 10월 1일 일요일    오전 9:31

```
jinuser@jin-VirtualBox:~$ hadoop namenode -format
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
```

hadoop namenode -format



# 하둡 실행하기

2017년 10월 1일 일요일 오전 9:57

```
manager@master:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-man
slave3: starting datanode, logging to /usr/local/hadoop/logs/hadoop-man
slave2: starting datanode, logging to /usr/local/hadoop/logs/hadoop-man
slave1: starting datanode, logging to /usr/local/hadoop/logs/hadoop-man
Starting secondary namenodes [slave1]
slave1: starting secondarynamenode, logging to /usr/local/hadoop/logs/h
out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-manage
slave2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ma
slave3: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ma
slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ma
manager@master:~$ jps
10439 ResourceManager
10510 Jps
10158 NameNode
```

```
manager@slave1:~$ jps
5585 DataNode
5685 SecondaryNameNode
5868 Jps
5774 NodeManager
```

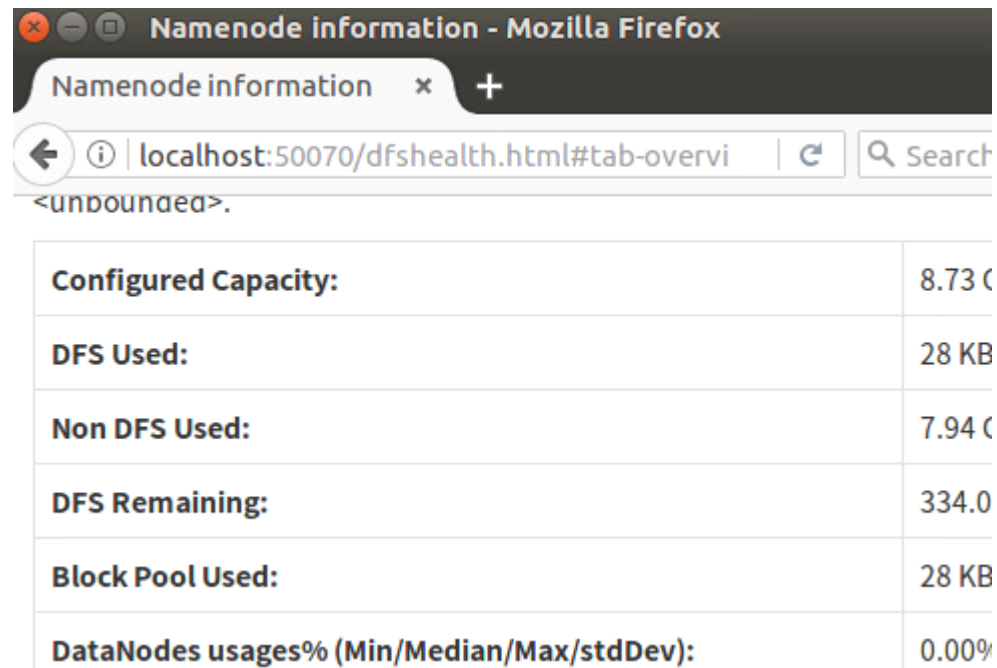
```
manager@slave2:~$ jps
4275 NodeManager
4170 DataNode
4349 Jps
```

```
manager@slave3:~$ jps
4624 NodeManager
4519 DataNode
4717 Jps
```

```
manager@master:~$ netstat -plten | grep java
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 127.0.1.1:50070      0.0.0.0:*
    LISTEN  1001      54884      10158/java
tcp        0      0 127.0.1.1:62350      0.0.0.0:*
    LISTEN  1001      54895      10158/java
```

# 웹 브라우저 접속 및 확인

2017년 10월 1일 일요일 오전 10:21



The screenshot shows a Mozilla Firefox browser window with the title 'Namenode information - Mozilla Firefox'. The address bar displays 'localhost:50070/dfshealth.html#tab-overvi'. Below the address bar, there is a table with the following data:

Configured Capacity:	8.73 C
DFS Used:	28 KB
Non DFS Used:	7.94 C
DFS Remaining:	334.0
Block Pool Used:	28 KB
DataNodes usages% (Min/Median/Max/stdDev):	0.00%

# [참고]

2017년 10월 7일 토요일    오후 2:21

<link href='https://www.blogger.com/dyn-css/authorization.css?targetBlogID=9080564510080817825&zx=9aaaaea36-d59a-4e99-8363-e27fe4d84811'  
rel='stylesheet'/>

## Archive for CE

Thursday, April 23, 2015

### 하둡 2.6.0 설치법 (멀티 노드)

#### 0. 사용 환경

- ubuntu 서버용 14.10 여러대 (ubuntu 14.04 LTS를 추천한다.)
- Java 1.8.0\_40 (2015년 4월 24일 기준 최신버전이다.)
- Hadoop 2.6.0 (최신버전은 2015년 4월 22일에 나온 2.7.0 버전이다.)
- NFS와 NIS를 사용하여 10대의 노드들이 /home 디렉터리와 사용자 계정을 공유중
- \* NFS와 NIS를 사용하면 간편하겠지만, 꼭 그럴 필요는 없다.

#### 1. 설치 개념 설명

- 하둡은 자바를 기반으로 만들어진 framework 이다. 따라서 자바가 반드시 설치되어 있어야 한다.
- 하둡은 한 대의 마스터 노드와 여러대의 슬레이브 노드로 구성된다. 마스터와 슬레이브를 한 대의 노드(=컴퓨터)에 설치해도 작동은 한다.
- 하둡은 파일의 저장 및 관리를 담당하는 HDFS(Hadoop File System)와 실제로 map & reduce job을 실행하는 Yarn으로 나눌 수 있다. 실제로도 둘은 각각 따로 실행 및 종료 된다.
- 이 글에서는 SSH Key 등록, Java 설치, 하둡 설치 및 설정, 하둡 실행 및 테스트 등 기초부터 실제 사용까지 상세히 다룰 것이다.
- 질문과 오류 지적은 언제든지 환영한다.

\* NFS와 NIS 설정 방법은 아래 링크를 참조.

NFS : [http://www.server-world.info/en/note?os=Ubuntu\\_12.04&p=nfs](http://www.server-world.info/en/note?os=Ubuntu_12.04&p=nfs)

NIS : [http://www.server-world.info/en/note?os=Ubuntu\\_12.04&p=nis](http://www.server-world.info/en/note?os=Ubuntu_12.04&p=nis)

(둘 다 우분투 12.04버전 기준이지만 14버전에서도 작동한다. NIS 설치하는 페이지 우측 상단에 있는 NIS Client도 클릭해서 설치 해줘야 한다)

#### 2. SSH Key 등록

- 한 노드에서 다른 노드로 접속할 때 비밀번호 없이 바로 로그인할 수 있도록 SSH Key라는 것을 생성 후 등록해야 한다.
- 하둡을 실행시킬 때 모든 노드에서 로그인을 한 뒤 각 노드에서 하둡을 실행하게 된다. 이 때 일일이 비밀번호를 치고 로그인을 할 수는 없기 때문에 SSH Key 등록 과정을 반드시 해주어야 한다.
- 단일 노드에서 설치를 할 경우에는 필요가 없는 과정이다.
- <http://opentutorials.org/module/432/3742> 에 정말 이해도 쉽고 따라하기 좋게 설명이 되어있으며, 여기서는 이를 참고하여 간략히 과정을 설명하겠다.

(1) 본인 계정의 home 디렉터리에서 `ssh-keygen -t rsa` 명령어로 키를 생성한다.

(2) 여러가지 입력할 것들이 나오는데, 그냥 엔터를 몇 번 쳐주면 default로 설정된다.

(3) `cd .ssh` 명령어로 .ssh 디렉터리에 들어간다.

(4) `ls -al`을 쳐서 권한이 다음과 같이 설정되었는지 확인한다.

```
drwx----- 2 inthound inthound 4096 Apr  9 16:34 ./
drwxr-xr-x 5 inthound inthound 4096 Apr 16 21:19 ../
-rw-r--r-- 1 inthound inthound 1985 Apr  9 16:40 authorized_keys
-rw----- 1 inthound inthound 1675 Apr 24 10:07 id_rsa
-rw-r--r-- 1 inthound inthound 397 Apr 24 10:07 id_rsa.pub
-rw-r--r-- 1 inthound inthound 4810 Apr 16 21:20 known_hosts
```

잘 모르겠다면 그냥 다음과 같이 명령어를 쳐준다.

```
chmod 700 ./
```

```
chmod 755 ../
```

```
chmod 600 id_rsa
```

```
chmod 644 id_rsa.pub
```

```
chmod 644 authorized_keys
```

```
chmod 644 known_hosts
```

\* `known_hosts` 파일이 없다면 `touch known_hosts` 로 파일을 만들어준다.

\* SSH는 권한 설정이 매우 중요하다. 여기서 대부분의 오류가 발생한다.

(5) `scp id_rsa.pub 아이디@서버이름:id_rsa.pub` 명령어로 `id_rsa.pub` 파일을 보내고자 하는 슬레이브 노드들에 복사해준다.

\* NFS를 사용중인 경우는 마스터노드와 슬레이브노드 모두 같은 디렉터리를 사용하므로 scp를 해줄 필요가 없다.

(6) 해당 노드에 접속해서 `cat ~/id_rsa.pub >> ~/.ssh/authorized_keys` 명령어로 공개키를 등록해준다.

\* NFS를 사용중인 경우는 현재 노드에서 바로 `cat ~/id_rsa.pub >> ~/.ssh/authorized_keys` 명령을 한번만 해주면 모든 노드에 적용이 되므로 편하다.

(7) `ssh 아이디@서버이름` 으로 접속할 때 비밀번호가 없이 접속이 되면 성공.

### 3. Java 설치

- 자바가 이미 설치되어 있는 경우에는 Java 경로를 PATH에 등록하는 것만 따라하면 된다.

(1) <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 에서 Accept License Agreement 에 체크를 하고 해당 리눅스 버전에 맞는 Java SE Development Kit을 다운받는다. 아마 `jdk-8u45-linux-x64.tar.gz` 이라고 써진 것을 다운 받으면 될 것이다.

(2) WinSCP 등의 파일 전송 프로그램을 사용하여 해당 파일을 각 서버의 본인 계정으로 복사해준다.

\* NFS를 사용중인 경우는 역시 하나의 노드에 자바를 설치하면 나머지 노드들이 자바를 쓸 수 있어서 편리하다.

(3) `tar -xvzf jdk-8u45-linux-x64.tar.gz` 명령어로 압축을 풀어준다.

(4) `mv jdk1.8.0_40 java` 명령어로 디렉터리 이름을 java로 바꿔준다.

(5) vi .bashrc 명령어로 .bashrc 파일을 연 후 맨 하단에 다음을 입력한다.

```
#####  
### Java  
#####  
export JAVA_HOME=$HOME/java  
export PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
```

\* 위에 있는 # 부분은 해당 PATH가 자바 설정을 위한 것임을 나타내는 것이며, 주석처리되어있으므로 당연히 설정에는 영향을 미치지 않는다. 보기 싫다면 #부분을 모두 빼버려도 좋다.

(6) 저장 후 종료한 뒤, source .bashrc 명령어로 .bashrc를 적용시킨다.

(7) jps 를 쳤을 때 23190 Jps 처럼 나오면 성공. (숫자는 바뀌어도 상관 없다.)

\* jps 명령어는 나중에 매우 자주 사용될 것이다.

\* java -version 명령어로도 현재 설치된 자바의 버전을 알 수 있다. 1.8.0\_40 버전이라고 뜨면 잘 설치된 것이다.

(8) 나머지 노드들에도 동일하게 설정을 해준다.

\* 역시나 NFS를 사용할 경우 한 노드에서만 설정을 해도 나머지가 모두 적용된다.

#### 4. 하둡 설치

(1) <http://apache.mirror.cdnetworks.com/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz> 에서 하둡 2.6.0버전을 받은 후 WinSCP등을 통해 본인의 홈 디렉터리로 옮겨준다.

\* 하둡의 다운로드 페이지는 <http://hadoop.apache.org/#Download+Hadoop> 이다.

(2) tar -xvzf hadoop-2.6.0.tar.gz 명령어로 압축을 풀어준다.

(3) mv hadoop-2.6.0 hadoop 명령어로 디렉터리 이름을 hadoop으로 바꿔준다.

(4) vi .bashrc 명령어로 .bashrc 파일을 연 후 맨 하단에 다음을 입력한다.

```
#####  
### Hadoop  
#####  
export HADOOP_HOME=$HOME/hadoop  
export PATH=$PATH:$HADOOP_HOME/bin  
export PATH=$PATH:$HADOOP_HOME/sbin  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export YARN_HOME=$HADOOP_HOME  
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native  
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"  
export HADOOP_CLASSPATH=$JAVA_HOME/lib/tools.jar  
export CLASSPATH=$HADOOP_HOME/share/hadoop/common/hadoop-common-2.6.0.jar:  
$HADOOP_HOME/share/hadoop/mapreduce/:$HADOOP_HOME/share/hadoop/common/lib/commons-  
cli-1.2.jar
```

\* 위의 HADOOP\_HOME과 PATH 부분이 있는 3줄은 반드시 해줘야 하는 것들이고, 그 밑으로는 오류를 수정할 때 하나씩 추가한 것들이어서 좀 난잡하다. 필요가 없는 것들이 있을지 모르겠지만 일단 놔두겠다.

(5) source .bashrc 명령어로 .bashrc 파일의 변경 사항을 적용해준다.

\* 나머지 노드들도 모두 같게 설정해준다.

(6) 홈 디렉터리에서 mkdir -p hadoop\_tmp/hdfs/namenode 명령어와

mkdir -p hadoop\_tmp/hdfs/datanode 명령어를 입력한다.

\* NFS를 사용할 경우 홈디렉터리가 아니라 각 서버의 로컬 디렉터리에 위의 디렉터를 만들어줘야 한다. 그러자면 sudo권한이 필요할 수도 있으며, 다 만든 후에 소유권 설정(chown)이 제대로 되어있나 `ls -al`으로 확인해야 한다.

\* NFS를 사용한다 하더라도 로컬 디렉터리 생성은 서로 공유가 안 되기 때문에 각각의 노드에서 해주어야 한다.

## 5. 하둡 설정

- 위의 4단계까지 했다면 하둡을 설치하는 일은 어느정도 완료한 것이다. 그러나 가장 중요한 몇몇 설정들이 남아 있다. 하둡 설치의 핵심이라고도 할 수 있으며, 하둡 실행시 오류가 대부분 이곳에서 발생한다.

- `cd ~/hadoop/etc/hadoop` 로 이동한 후, 해당 디렉터리에 있는 `hadoop-env.sh`, `slaves`, `core-site.xml`, `hdfs-site.xml`, `mapred-site.xml`, `yarn-site.xml` 만 손보면 된다.

(1) vi slave 로 파일을 연 후 마스터 노드와 슬레이브 노드를 다음과 같이 적는다.

서버1

서버2

서버3

...

서버10

\* 이 때, 서버1~서버10은 vi `/etc/hosts` 에 등록되어있는 닉네임을 사용하였으며, 닉네임 대신에 아이피를 직접 입력해도 무관하다.

(2) `hadoop-env.sh`의 25번째줄 근처에 있는 `JAVA_HOME`을 다음과 같이 변경 및 추가하고 저장 및 종료한다.

```
export JAVA_HOME=${JAVA_HOME}
export HADOOP_PREFIX=${HADOOP_HOME}
```

\*\* 추가: log파일 퍼미션 에러가 난다면

```
$ vi hadoop-env.sh
```

상단에 환경변수를 추가하고 디렉터리도 해당위치에 만들어준다.

```
export HADOOP_LOG_DIR=${HADOOP_HOME}/logs/hadoop/core
export HADOOP_ROOT_LOGGER=INFO,console
export HADOOP_PID_DIR=${HADOOP_LOG_DIR}
export HADOOP_SECURITY_LOGGER=INFO,NullAppender
export HDFS_AUDIT_LOGGER=INFO,NullAppender
```

출처: <https://beyondi2ee.wordpress.com/2013/05/08/hadoop%EC%9D%98-%EB%A1%9C%EA%B7%B8%ED%8C%8C%EC%9D%BC%EC%9D%80-%EC%96%B4%EB%94%94%EC%97%90-%EC%9E%88%EB%8A%94%EA%B0%80/>

(3) `core-site.xml`에 다음을 작성한다.

```
< configuration>
  < property>
    < name>fs.defaultFS< /name>
    < value>hdfs://서버1:9000/< /value>
  < /property>
< /configuration>
```

(4) `hdfs-site.xml`에 다음을 작성한다.

```
< configuration>
  < property>
    < name>dfs.replication< /name>
    < value>3< /value>
  < /property>
  < property>
    < name>dfs.namenode.name.dir< /name>
    < value>file:~/hadoop_tmp/hdfs/namenode< /value>
  < /property>
  < property>
    < name>dfs.datanode.data.dir< /name>
    < value>file:~/hadoop_tmp/hdfs/datanode< /value>
```

```
</property>
< /configuration>
```

\* name.dir과 data.dir의 위치는 본인이 hadoop\_tmp의 namenode와 datanode 디렉터리를 만든 곳의 위치로 설정해줘야 한다. 여기서는 홈디렉터리에 hadoop\_tmp를 만들었다고 가정하였으며, NFS를 사용할 경우 해당 로컬디렉터리의 위치로 설정해주어야 한다.

(5) mapred-site.xml에 다음을 작성한다.

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
< /configuration>
```

(6) yarn-site.xml에 다음을 작성한다.

```
<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>서버.ac.kr</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>서버.ac.kr:8025</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>서버.ac.kr:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>서버.ac.kr:8050</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>서버.ac.kr:8055</value>
  </property>
</configuration>
```

\* resourcemanager 부분에서는 외부에서 접속을 해야하기 때문에, 외부 접속이 가능한 도메인 주소나 아이피 주소를 써야한다. 여기서는 서버1.univ.ac.kr 등으로 적었다.

\* 8025 등의 숫자는 포트 번호를 뜻하며, 해당 노드에서 사용중이지 않은 포트 번호를 사용해야 한다.

(7) 위에서 수정한 모든 파일들을 scp slave 아이디@서버2:~/hadoop/etc/hadoop 과 같은 명령어로 모든 노드에 복사해준다.

\* 혹은 하둡 자체를 복사해줘도 된다.

```
scp -r hadoop 아이디@서버2:
scp -r hadoop 아이디@서버3:
...
```

```
(8) scp .bashrc 아이디@서버2:
scp .bashrc 아이디@서버3:
...
```

등으로 .bashrc 파일을 모두 뿌려준다.

\* NFS를 사용할 경우, 이 과정을 해줄 필요가 없다.

## 6. 하둡 실행

- NFS를 사용하든 안 하든, 아래 과정은 마스터 노드에서만 하면 된다.

(1) `hadoop namenode -format` 명령어로 네임노드를 포맷한다.

(2) `start-dfs.sh` 명령어로 hdfs를 실행시킨다.

(3) 잠시 기다린 후 `jps` 명령어를 쳤을 때, 마스터 노드에서는

```
10690 NameNode
10873 DataNode
12060 Jps
11119 SecondaryNameNode
```

데이터 노드에서는

```
10873 DataNode
12060 Jps
```

과 같이 떠야 정상이다. 숫자는 다를 수 있다.

(4) `start-yarn.sh` 명령어로 yarn(map-reduce 프로그램)을 실행시킨다.

(5) 잠시 기다린 후 `jps` 명령어를 쳤을 때, 마스터노드에서는

```
10690 NameNode
11475 NodeManager
12060 Jps
11292 ResourceManager
11119 SecondaryNameNode
```

데이터노드에서는

```
11475 NodeManager
10873 DataNode
12060 Jps
```

과 같이 떠야 정상이다. 숫자는 다를 수 있다.

(6) 웹 브라우저 주소창에 서버1.univ.ac.kr:50070 을 쳤을 때, HDFS 상태를 알려주는 화면이 뜨며, live node 개수가 서버 총 개수와 같으면 HDFS는 정상이다.

(7) 웹 브라우저 주소창에 서버1.univ.ac.kr:8055 를 쳤을 때, Resource 상태를 알려주는 화면이 뜨며, active node 개수가 서버 총 개수와 같으면 Resource manager 또한 정상이다.

(8) 하둡을 끝 때는 `stop-all.sh`를 사용해서 끝 수 있으며,  
`start-all.sh`로 hdfs와 yarn을 한꺼번에 켤 수도 있다.

## 7. 하둡 테스트

- 하둡 테스트 방법은 <http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-common/SingleCluster.html>을 참고하였다. 단일 노드 뿐 아니라 다중 노드에서도 정상 작동한다.

(1) `hdfs dfs -mkdir input` 을 쳐서 input 디렉토리를 만든다.



- (2) `hdfs dfs -lsr` 을 쳐서 `input` 디렉토리가 제대로 생성되었는지 확인한다.
- (3) `hdfs dfs -put ~/hadoop/etc/hadoop input` 명령어로 `input` 디렉토리에 데이터로 사용할 `hadoop` 파일들을 업로드한다. 딱히 큰 의미가 있는 것은 아니며, 그냥 여러 파일들이 있으니까 쓰는거다. 본인이 직접 만든 `data set`을 사용해도 된다.
- (4) `hadoop jar ~/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar grep input/hadoop/ output 'dfs[a-z.]+'`  
명령어로 하둡 Job을 실행한다.
- (5) `map`과 `reduce`의 퍼센트가 조금씩 올라갈 것이다. <http://서버1.univ.ac.kr:8055> 에도 Job의 진행 경과를 살펴볼 수 있다.
- (6) Job이 종료되면 `hdfs dfs -lsr` 명령어로 `output` 디렉토리가 제대로 생성되었는지 확인해본다.
- \* 확인 후 `hdfs dfs -rmr output` 명령어로 `output` 디렉토리를 지워줘야 다음번 Job 에서 `already existed` 오류가 안생긴다.

## 8. 오류 해결

- (1) 하둡을 실행시킨 후 `jps`를 쳐보면 `DataNode`가 안 나온다.  
-> `hadoop_tmp/hdfs/datanode` 디렉토리를 삭제하고 새로 만든 후 `hadoop namenode -format` 을 다시 해주고 (Y/N 을 물을 경우 Y를 해준다), 다시 하둡을 실행해본다.
- (2) 하둡을 실행시켰는데 `jps`에서 제대로 뜨는게 별로 없다.  
-> 다른 하둡이 실행되고 있는지 확인해본다. 실행중이라면 그 하둡을 끄거나, 아니면 `hdfs-site.xml`과 `yarn-site.xml` 파일의 여러 포트번호를 새로 설정해서 포트간 충돌이 나지 않게 해야 한다.

`hdfs-site.xml` 에서

```
<property>
  <name>dfs.http.address</name>
  <value>http://서버1.univ.ac.kr:50160</value>
</property>
<property>
  <name>dfs.datanode.address</name>
  <value>0.0.0.0:50161</value>
</property>
<property>
  <name>dfs.datanode.http.address</name>
  <value>0.0.0.0:50162</value>
</property>
<property>
  <name>dfs.datanode.https.address</name>
  <value>0.0.0.0:50163</value>
</property>
<property>
  <name>dfs.datanode.ipc.address</name>
  <value>0.0.0.0:50164</value>
</property>
```

를 추가해주고, 포트 번호들은 안 쓰는 포트 번호로 지정해준다.

`netstat -a -t --numeric-ports -p` 명령어로 사용중인 포트 번호를 확인할 수 있다. `yarn-site.xml`은 아마 고칠 필요가 없을 것이다. 여러 계정이 서버를 사용중이라면, 사용중인 하둡 포트번호를 문서화해서 관리하는 것도 하나의 방법이다.

- (3) 하둡이 안 되는데 뭐가 문제인지 모르겠다.

-> 리눅스 방화벽 문제일 수도 있다. `/etc/init.d/iptables stop` 명령어로 방화벽을 꺼본다. 다시 켤 때는 `/etc/init.d/iptables start` 명령어를 쓰면 된다.

- (4) `jps` 명령어가 안 먹는다.

-> JAVA\_HOME이 제대로 설정되지 않은 문제이다. .bashrc에서 JAVA\_HOME을 제대로 변경했는지, 그리고 source .bashrc를 해주었는지 확인해본다.

(5) Input path error가 뜬다.

-> 하둡은 디스크 용량이 90% 이상 사용중일 경우 제대로 작동하지 않는다. 사용중인 용량을 90% 이하로 만들고 다시 실행하면 된다.

(6) Live node 설정이 제대로 안 되어 있다.

-> vi /etc/hosts 에서 127.0.0.1에 등록 되어 있는 본인 서버 이름을 없앤다.

(그냥 127.0.0.1 localhost 만 남아있게 설정해줌. 모든 slave 노드에 적용)

## 9. 출처 및 유용한 사이트

[SSH Key - 비밀번호 없이 로그인] <http://opentutorials.org/module/432/3742>

[서버 재시작 시 namenode 시작 안되는 현상] <http://egloos.zum.com/entireboy/v/4821605>

[HOW TO INSTALL APACHE HADOOP 2.6.0 IN UBUNTU (SINGLE NODE SETUP)] <http://pingax.com/install-hadoop2-6-0-on-ubuntu/>

[20141029 하둡2.5와 hive설치 및 예제] <http://www.slideshare.net/TaeYoungLee1/20141029-25-hive>

[BindException] - <http://wiki.apache.org/hadoop/BindException>

[HDFS Ports] - [https://ambari.apache.org/1.2.3/installing-hadoop-using-ambari/content/reference\\_chap2\\_1.html](https://ambari.apache.org/1.2.3/installing-hadoop-using-ambari/content/reference_chap2_1.html)

\* 하둡의 port관련 정보가 있다.

[Apache Hadoop] - <http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-common/SingleCluster.html>

\* 하둡의 single node 설치 튜토리얼. 좌측의 Cluster Setup에서는 Wordcount 예제와 port별 설명을 자세하게 알 수 있다.

[Hadoop 2.x : jar file location for wordcount example] - <http://ac31004.blogspot.kr/2013/11/hadoop-2x-jar-file-location-for.html>

\* jar 실행 관련 문제가 생겨서 참고하였다.

[리눅스(Linux) 포트 열기, 방화벽 설정/해제 등.] - [http://www.v-eng.co.kr/bbs/board.php?bo\\_table=faq&wr\\_id=40](http://www.v-eng.co.kr/bbs/board.php?bo_table=faq&wr_id=40)

출처: <<http://inthound.blogspot.kr/2015/04/how-to-install-hadoop-260-in-ubuntu.html>>

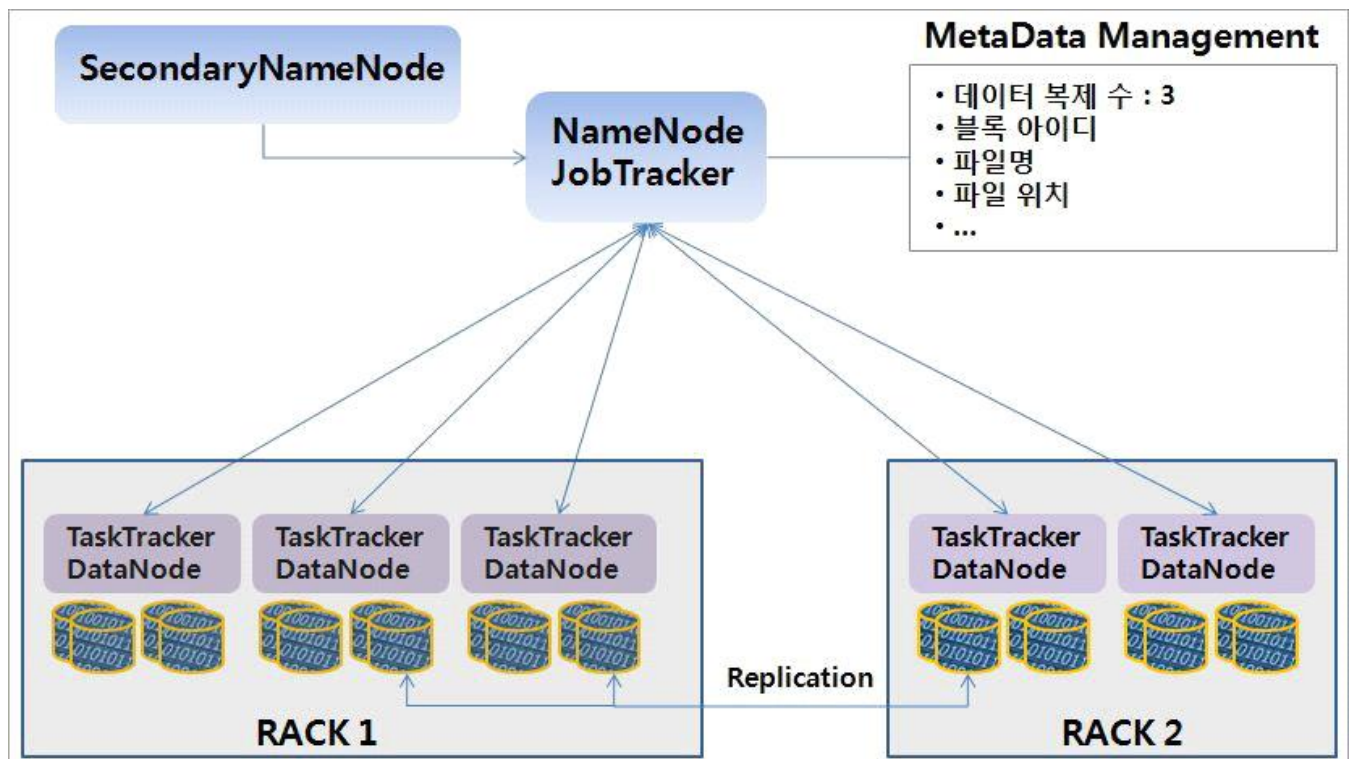
## [참고]

2017년 10월 19일 목요일 오후 5:09

## Hadoop - 분산 클러스터 구성(Hadoop v1.1.2)

Posted 03 27, 2013 17:15, Filed under: [BigData/Hadoop](#)

### # Hadoop 분산 클러스터 구성



노드 구분명	서버 IP	호스트명
Name Node	192.168.1.100	master.namenode
Secondary Node	192.168.1.101	secondary.namenode
Data Node 01	192.168.1.111	slave.datanode01
Data Node 02	192.168.1.112	slave.datanode02
Data Node 03	192.168.1.113	slave.datanode03

### # 시스템 인코딩 방식 확인

- 하둡은 인코딩 방식으로 UTF-8을 사용함.
- 한글로된 파일을 처리할 경우, 한글이 모두 깨질 수 있음
- 시스템 인코딩 방식을 확인후에 UTF-8이 아닌경우 인코딩을 변경

```
1 [hadoop@master bin]$ echo $LANG
```

```

2      ko_KR.eucKR
3
4      // 인코딩 변경
5      [hadoop@master bin]$ vi /etc/sysconfig/i18n
6      LANG="ko_KR.UTF-8"
7
8      // 변경한 설정 적용
9      [hadoop@master bin]$ source /etc/sysconfig/i18n
10
11     // 변경된 정보 확인
12     [hadoop@master bin]$ locale
13     LANG=ko_KR.UTF-8
14     LC_CTYPE="ko_KR.UTF-8"
15     LC_NUMERIC="ko_KR.UTF-8"
16     LC_TIME="ko_KR.UTF-8"
17     LC_COLLATE="ko_KR.UTF-8"
18     LC_MONETARY="ko_KR.UTF-8"
19     LC_MESSAGES="ko_KR.UTF-8"
20     LC_PAPER="ko_KR.UTF-8"
21     LC_NAME="ko_KR.UTF-8"
22     LC_ADDRESS="ko_KR.UTF-8"
23     LC_TELEPHONE="ko_KR.UTF-8"
24     LC_MEASUREMENT="ko_KR.UTF-8"
25     LC_IDENTIFICATION="ko_KR.UTF-8"

```

## # 그룹 및 계정 생성, 각 서버별 공통(namenode, secondary\_namenode, datanode)

```

1      groupadd hadoop
2      useradd -g hadoop -d /home/hadoop -c /bin/bash hadoop

```

## # master.namenode 서버의 hosts 설정 추가

※각 서버의 기본 호스트(localhost)명을 아래와 같은 호스트명으로 구분되도록 변경(웹페이지 확인시 필요)

```

1      /etc/hosts
2      192.168.1.100  master.namenode
3      192.168.1.101  secondary.namenode
4
5      192.168.1.111  slave.datanode01
6      192.168.1.112  slave.datanode02
7      192.168.1.113  slave.datanode03

```

# 호스트명 이용시 각각의 서버에 master.namenode 호스트 정보와 해당 서버의 호스트 정보를 등록해줌

예1) secondary.namenode 일경우

```
1      cat /etc/hosts
2      192.168.1.100  master.namenode
3      192.168.1.101  secondary.namenode
```

예2) slave.datanode01 일경우

```
1      cat /etc/hosts
2      192.168.1.100  master.namenode
3      192.168.1.111  slave.datanode01
```

## iptables 방화벽 포트 접근 허용

# master.namenode, secondary\_namenode

```
1      [root@master ~]# vi /etc/sysconfig/iptables
2      -A INPUT -m state --state NEW -m tcp -p tcp --dport 9000 -j ACCEPT
3      -A INPUT -m state --state NEW -m tcp -p tcp --dport 9001 -j ACCEPT
4      -A INPUT -m state --state NEW -m tcp -p tcp --dport 50030 -j ACCEPT
5      -A INPUT -m state --state NEW -m tcp -p tcp --dport 50070 -j ACCEPT
6      -A INPUT -m state --state NEW -m tcp -p tcp --dport 50090 -j ACCEPT
```

## # 테이터노드(datanode) : 각 datanode 서버들 별로 포트 접근 허용

```
1      [root@master ~]# vi /etc/sysconfig/iptables
2
3      -A INPUT -m state --state NEW -m tcp -p tcp --dport 50010 -j ACCEPT
4      -A INPUT -m state --state NEW -m tcp -p tcp --dport 50020 -j ACCEPT
5      -A INPUT -m state --state NEW -m tcp -p tcp --dport 50060 -j ACCEPT
6      -A INPUT -m state --state NEW -m tcp -p tcp --dport 50075 -j ACCEPT
```

# Hadoop 에서 사용할 디렉토리 생성

# master.namenode

```
1      [root@master data]# mkdir /data/name
2      [root@master data]# chown -R hadoop.hadoop /data
```

# secondary.namenode

```
1      [root@master ~]# mkdir /data
2      [root@master ~]# chown -R hadoop.hadoop /data
```

# 각각의 slave.datanode(디렉토리 권한 755)

```
1      [root@slave ~]# mkdir -p /data/node01
```

```
2 [root@slave ~]# mkdir -p /data/node02
3 [root@slave ~]# chown -R hadoop.hadoop /data
```

## # Hadoop 시스템간의 데이터 전송을 위하여 SSH 인증 키 생성 및 서버별 배포

# namenode에서 모든 시스템 접근

```
1 [hadoop@master ~]$ ssh-keygen -t rsa
2 Generating public/private rsa key pair.
3 Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
4 Enter passphrase (empty for no passphrase):
5 Enter same passphrase again:
6 Your identification has been saved in /home/hadoop/.ssh/id_rsa.
7 Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
8 The key fingerprint is:
9 df:1f:3d:1d:00:37:02:a7:d7:4c:19:4b:cd:71:e3:03 hadoop@master
10 The key's randomart image is:
11 +--[ RSA 2048]-----+
12 |      ..+ E=oo|
13 |      o O.=oo|
14 |      . . = o |
15 |      .  . .|
16 |      S    . |
17 |      . . .o|
18 |      . . ..o|
19 |      .  ..|
20 |      .  . |
21 +-----+
```

## 인증키 배포

```
1 // master.namenode 서버 자신을 접속하기 위해 인증키 복사
2 [hadoop@master ~]$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
3
4 // secondary.namenode
5 [hadoop@master ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@secondary.namenode
6
7 // datanode
8 [hadoop@master ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@slave.datanode01
9 [hadoop@master ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@slave.datanode02
10 [hadoop@master ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@slave.datanode03
11
12 // 인증키를 이용한 접근 테스트
13 [hadoop@master ~]$ ssh hadoop@secondary.namenode
```

```
14 [hadoop@master ~]$ ssh hadoop@slave.datanode01
15 [hadoop@master ~]$ ssh hadoop@slave.datanode02
16 [hadoop@master ~]$ ssh hadoop@slave.datanode03
```

## Hadoop 설치

# master.namenode 서버의 홈디렉토리에 설치

```
1 [hadoop@master ~]$ tar -xzf hadoop-1.1.2.tar.gz
```

# Java 위치 확인

/home/hadoop/jdk1.7.0\_17

## 환경 설정

```
1 [hadoop@master ~]$ cd hadoop-1.1.2/conf/
```

### hadoop-env.sh

Hadoop 을 구동하는 스크립트에서 사용되는 환경 변수를 정의한 파일

```
1 [hadoop@master conf]$ vi hadoop-env.sh
2 # Set Hadoop Home
3 export HADOOP_HOME=/home/hadoop/hadoop-1.1.2
4 export HADOOP_HOME_WARN_SUPPRESS="TRUE"
5
6 # The java implementation to use. Required.
7 export JAVA_HOME=/home/hadoop/jdk1.7.0_17
8
9 # Extra Java CLASSPATH elements. Optional.
10 # export HADOOP_CLASSPATH=
11
12 # The maximum amount of heap to use, in MB. Default is 1000.
13 # export HADOOP_HEAPSIZE=2000
14
15 # Extra Java runtime options. Empty by default.
16 export HADOOP_OPTS=-server
```

### core-site.xml

HDFS 와 Map/Reduce 에 공통적으로 사용되는 I/O 설정 같은 Hadoop 코어를 위한 설정파일

```
1 [hadoop@master conf]$ vi core-site.xml
2 <?xml version="1.0"?>
3 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
4
5 <!-- Put site-specific property overrides in this file. -->
6
7 <configuration>
```

```

8      <property>
9          <name>fs.default.name</name>
10         <value>hdfs://master.namenode:9000</value>
11     </property>
12     <property>
13         <name>hadoop.tmp.dir</name>
14         <value>/tmp/hadoop-${user.name}</value>
15     </property>
16 </configuration>

```

## hdfs-site.xml

Name Node, Secondary Node, Data Node 등과 같은 HDFS Daemon 을 위한 환경설정 파일

```

1      [hadoop@master conf]$ vi hdfs-site.xml
2      <?xml version="1.0"?>
3      <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
4
5      <configuration>
6          <property>
7              <name>fs.default.name</name>
8              <value>hdfs://master.namenode:9000</value>
9          </property>
10         <property>
11             <name>dfs.name.dir</name>
12             <value>/data/name,/data/backup</value>
13         </property>
14         <property>
15             <name>dfs.data.dir</name>
16             <value>/data/node01,/data/node02</value>
17         </property>
18         <property>
19             <name>dfs.replication</name>
20             <value>2</value>
21         </property>
22         <property>
23             <name>fs.trash.interval</name>
24             <value>30</value>
25         </property>
26         <property>
27             <name>dfs.webhdfs.enabled</name>
28             <value>true</value>
29         </property>
30         <property>
31             <name>dfs.support.append</name>

```



```
32         <value>true</value>
33     </property>
34     <property>
35         <name>dfs.support.broken.append</name>
36         <value>true</value>
37     </property>
38     <property>
39         <name>dfs.permissions</name>
40         <value>true</value>
41     </property>
42     <property>
43         <name>dfs.web.ugi</name>
44         <value>hadoop,supergroup</value>
45     </property>
46     <property>
47         <name>dfs.permissions.supergroup</name>
48         <value>supergroup</value>
49     </property>
50     <property>
51         <name>dfs.upgrade.permission</name>
52         <value>0777</value>
53     </property>
54     <property>
55         <name>dfs.umaskmode</name>
56         <value>022</value>
57     </property>
58     <property>
59         <name>dfs.http.address</name>
60         <value>master.namenode:50070</value>
61     </property>
62     <property>
63         <name>dfs.secondary.http.address</name>
64         <value>secondary.namenode:50090</value>
65     </property>
66     <property>
67         <name>fs.checkpoint.dir</name>
68         <value>/data/checkpoint</value>
69     </property>
70     <property>
71         <name>fs.checkpoint.period</name>
72         <value>360</value>
73     </property>
74     <property>
75         <name>dfs.backup.address</name>
```

```

76         <value>secondary.namenode:50100</value>
77     </property>
78     <property>
79         <name>dfs.backup.http.address</name>
80         <value>secondary.namenode:50105</value>
81     </property>
82 </configuration>

```

## mapred-site.xml

Job Tracker, Task Tracker 같은 Map/Reduce Daemon 을 위한 환경설정 파일

mapred.system.dir : HDFS 상의 디렉토리로 설정합니다.

mapred.local.dir : 실제 Data Node 의 로컬 디렉토리를 지정하며, 콤마(,) 로 여러개의 경로를 지정할 수 있습니다.

```

1    [hadoop@master conf]$ vi mapred-site.xml
2    <?xml version="1.0"?>
3    <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
4
5    <configuration>
6        <property>
7            <name>mapred.job.tracker </name>
8            <value>hdfs://master.namenode:9001</value>
9        </property>
10       <property>
11           <name>mapred.system.dir</name>
12           <value>/tmp/hadoop/mapred/system</value>
13       </property>
14       <property>
15           <name>mapred.local.dir</name>
16           <value>/data/mapred/local</value>
17       </property>
18       <property>
19           <name>dfs.hosts.exclude</name>
20           <value>/home/hadoop/hadoop-1.1.2/conf/hosts.exclude</value>
21       </property>
22       <property>
23           <name>mapred.hosts.exclude</name>
24           <value>/home/hadoop/hadoop-1.1.2/conf/hosts.exclude</value>
25       </property>
26   </configuration>

```

## masters

보조네임노드 (Secondary Node) 를 구동시킬 서버의 목록을 지정하여 줍니다.

```
1 [hadoop@master conf]$ vi masters
2 secondary.namenode
```

## slaves

Data Node 와 Task Tracker 를 구동시킬 서버의 목록을 지정하여 줍니다.

```
1 [hadoop@master conf]$ vi slaves
2 slave.datanode01
3 slave.datanode02
4 slave.datanode03
```

## ### 서버별 배포

# Java 배포 : 서버별 Java 설정이 필요할 경우

# 서버별 Bit 체크 : 서버의 Bit에 맞는 Java 설치

```
1 // secondary.namenode
2 scp -r /home/hadoop/jdk1.7.0_17 secondary.namenode:/home/hadoop/jdk1.7.0_17
3
4 // slave.datanode
5 scp -r /home/hadoop/jdk1.7.0_17 slave.datanode01:/home/hadoop/jdk1.7.0_17
6 scp -r /home/hadoop/jdk1.7.0_17 slave.datanode02:/home/hadoop/jdk1.7.0_17
7 scp -r /home/hadoop/jdk1.7.0_17 slave.datanode03:/home/hadoop/jdk1.7.0_17
8
9 // jps 정보 확인이 필요한 경우, 프로파일 배포
10 scp -r /home/hadoop/.bash_profile secondary.namenode:/home/hadoop/.bash_profile
11 scp -r /home/hadoop/.bash_profile slave.datanode01:/home/hadoop/.bash_profile
12 scp -r /home/hadoop/.bash_profile slave.datanode04:/home/hadoop/.bash_profile
```

## # 최초 배포

master.namenode 에 설치되어 있는 Hadoop 디렉토리 전체를 배포

```
1 // secondary.namenode
2 scp -r /home/hadoop/hadoop-1.1.2 secondary.namenode:/home/hadoop/hadoop-1.1.2
3
4 // slave.datanode
5 scp -r /home/hadoop/hadoop-1.1.2 slave.datanode01:/home/hadoop/hadoop-1.1.2
6 scp -r /home/hadoop/hadoop-1.1.2 slave.datanode02:/home/hadoop/hadoop-1.1.2
7 scp -r /home/hadoop/hadoop-1.1.2 slave.datanode03:/home/hadoop/hadoop-1.1.2
```

## # 변경 설정 배포

master.namenode 의 설정이 변경되었을 경우 각 서버별 변경된 파일만 배포

```
1 // secondary.namenode
2 rsync -av /home/hadoop/hadoop-1.1.2/conf
```

```

3      secondary.namenode:/home/hadoop/hadoop-1.1.2
4
5      // slave.datanode
6      rsync -av /home/hadoop/hadoop-1.1.2/conf
7      slave.datanode01:/home/hadoop/hadoop-1.1.2
      rsync -av /home/hadoop/hadoop-1.1.2/conf
      slave.datanode02:/home/hadoop/hadoop-1.1.2
      rsync -av /home/hadoop/hadoop-1.1.2/conf
      slave.datanode03:/home/hadoop/hadoop-1.1.2

```

### ### namenode 포맷

```

1      [hadoop@master ~]$ cd hadoop-1.1.2/bin/
2      [hadoop@master bin]$ ./hadoop namenode -format
3      13/03/27 11:45:27 INFO namenode.NameNode: STARTUP_MSG:
4      /*****
5      STARTUP_MSG: Starting NameNode
6      STARTUP_MSG:  host = master/192.168.1.100
7      STARTUP_MSG:  args = [-format]
8      STARTUP_MSG:  version = 1.1.2
9      STARTUP_MSG:  build =
10     https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.1 -r 1440782;
11     compiled by 'hortonfo' on Thu Jan 31 02:03:24 UTC 2013
12     *****/
13     Re-format filesystem in /data/name ? (Y or N) Y
14     Re-format filesystem in /data/backup ? (Y or N) Y
15     13/03/27 11:45:33 INFO util.GSet: VM type      = 32-bit
16     13/03/27 11:45:33 INFO util.GSet: 2% max memory = 19.33375 MB
17     13/03/27 11:45:33 INFO util.GSet: capacity    = 2^22 = 4194304 entries
18     13/03/27 11:45:33 INFO util.GSet: recommended=4194304, actual=4194304
19     13/03/27 11:45:34 INFO namenode.FSNamesystem: fsOwner=hadoop
20     13/03/27 11:45:34 INFO namenode.FSNamesystem: supergroup=supergroup
21     13/03/27 11:45:34 INFO namenode.FSNamesystem: isPermissionEnabled=true
22     13/03/27 11:45:34 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
23     13/03/27 11:45:34 WARN namenode.FSNamesystem: The dfs.support.append option is
24     in your configuration, however append is not supported. This configuration option is no
25     longer required to enable sync.
26     13/03/27 11:45:34 INFO namenode.FSNamesystem: isAccessTokenEnabled=false
27     accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
28     13/03/27 11:45:34 INFO namenode.NameNode: Caching file names occuring more than
29     10 times
30     13/03/27 11:45:36 INFO common.Storage: Image file of size 112 saved in 0 seconds.
31     13/03/27 11:45:36 INFO namenode.FSEditLog: closing edit log: position=4,
32     editlog=/data/name/current/edits
33     13/03/27 11:45:36 INFO namenode.FSEditLog: close success: truncate to 4,

```

```

34 editlog=/data/name/current/edits
35 13/03/27 11:45:36 INFO common.Storage: Storage directory /data/name has been
    successfully formatted.
    13/03/27 11:45:36 INFO common.Storage: Image file of size 112 saved in 0 seconds.
    13/03/27 11:45:36 INFO namenode.FSEditLog: closing edit log: position=4,
    editlog=/data/backup/current/edits
    13/03/27 11:45:36 INFO namenode.FSEditLog: close success: truncate to 4,
    editlog=/data/backup/current/edits
    13/03/27 11:45:36 INFO common.Storage: Storage directory /data/backup has been
    successfully formatted.
    13/03/27 11:45:36 INFO namenode.NameNode: SHUTDOWN_MSG:
    /*****
    SHUTDOWN_MSG: Shutting down NameNode at master/192.168.1.100
    *****/

```

## # DataNode 제거

```

1  /*
2    mapped-site.xml 파일에 아래처럼 제외할 호스트 파일정보가 있을경우
3    hosts.exclude 파일에 DataNode 명을 추가해줌으로 제거 가능
4    dfs.hosts.exclude, mapred.hosts.exclude 가 기재되어 있지 않을경우, 설정후 전체를 재시
5    작
6    */
7    <property>
8        <name>dfs.hosts.exclude</name>
9        <value>/home/hadoop/hadoop-1.1.2/conf/hosts.exclude</value>
10    </property>
11    <property>
12        <name>mapred.hosts.exclude</name>
13        <value>/home/hadoop/hadoop-1.1.2/conf/hosts.exclude</value>
    </property>

```

### 제외할 DataNode를 NameNode 서버의 hosts.exclude에 등록

```

1    echo slave.datanode01 > /home/hadoop/hadoop-1.1.2/conf/hosts.exclude
2
3    // DataNode 반영
4    [hadoop@master bin]$ ./hadoop dfsadmin -refreshNodes

```

## Hadoop 설치 정보 콘솔 확인

```

1    [hadoop@master bin]$ ./hadoop dfsadmin -report
2    Configured Capacity: 162174115840 (151.04 GB)
3    Present Capacity: 104111394816 (96.96 GB)
4    DFS Remaining: 104111194112 (96.96 GB)
5    DFS Used: 200704 (196 KB)

```

```

6      DFS Used%: 0%
7      Under replicated blocks: 0
8      Blocks with corrupt replicas: 0
9      Missing blocks: 0
10
11     -----
12     Datanodes available: 3 (3 total, 0 dead)
13
14     Name: 192.168.1.111:50010
15     Decommission Status : Normal
16     Configured Capacity: 44776726528 (41.7 GB)
17     DFS Used: 69632 (68 KB)
18     Non DFS Used: 39449726976 (36.74 GB)
19     DFS Remaining: 5326929920(4.96 GB)
20     DFS Used%: 0%
21     DFS Remaining%: 11.9%
22     Last contact: Wed Mar 27 14:33:29 KST 2013
23
24
25     Name: 192.168.1.113:50010
26     Decommission Status : Normal
27     Configured Capacity: 11708014592 (10.9 GB)
28     DFS Used: 69632 (68 KB)
29     Non DFS Used: 5046538240 (4.7 GB)
30     DFS Remaining: 6661406720(6.2 GB)
31     DFS Used%: 0%
32     DFS Remaining%: 56.9%
33     Last contact: Wed Mar 27 14:33:27 KST 2013
34
35
36     Name: 192.168.1.112:50010
37     Decommission Status : Normal
38     Configured Capacity: 105689374720 (98.43 GB)
39     DFS Used: 61440 (60 KB)
40     Non DFS Used: 13566455808 (12.63 GB)
41     DFS Remaining: 92122857472(85.8 GB)
42     DFS Used%: 0%
43     DFS Remaining%: 87.16%
44     Last contact: Wed Mar 27 14:33:26 KST 2013

```

## # 사용자 디렉토리 생성

```

1      // slave.datanode 에 생성됨
2      [hadoop@master bin]$ ./hadoop fs -mkdir /user/hadoop
3

```

```

4      // 디렉토리 정보 확인
5      [hadoop@master bin]$ ./hadoop fs -ls /
6      Found 2 items
7      drwxr-xr-x  - hadoop supergroup      0 2013-03-27 14:07 /tmp
8      drwxr-xr-x  - hadoop supergroup      0 2013-03-27 14:39 /user
9      [hadoop@master bin]$ ./hadoop fs -ls /user
10     Found 1 items
11     drwxr-xr-x  - hadoop supergroup      0 2013-03-27 14:39 /user/hadoop

```

## # 파일 업로드

```

1      [hadoop@master bin]$ ./hadoop fs -put ../cloud_computing_primer.pdf .
2      [hadoop@master bin]$ ./hadoop fs -
put ../oracle_plsql_language_pocket_reference_fourth_edition.pdf .

```

## # 서버별 프로세스 확인

```

1      // master.namenode
2      [hadoop@master bin]$ jps
3      14422 Jps
4      13714 NameNode
5      13877 JobTracker
6
7      // secondary.namenode
8      [hadoop@localhost data]$ jps
9      15764 Jps
10     14765 SecondaryNameNode
11
12     // slave.datanode 동일(TaskTracker, DataNode)
13     [hadoop@localhost data]$ jps
14     30475 TaskTracker
15     30900 Jps
16     30365 DataNode

```

## # Hadoop 구성 웹페이지 확인

<http://master.namenode:50070>

Live Nodes 가 연결된 DataNode의 수

# NameNode 'master:9000'

**Started:** Wed Mar 27 15:52:42 KST 2013  
**Version:** 1.1.2, r1440782  
**Compiled:** Thu Jan 31 02:03:24 UTC 2013 by hortonfo  
**Upgrades:** There are no upgrades in progress.

[Browse the filesystem](#)  
[Namenode Logs](#)

## Cluster Summary

10 files and directories, 3 blocks = 13 total. Heap Size is 48.12 MB / 966.69 MB (4%)

Configured Capacity	:	151.04 GB
DFS Used	:	3.07 MB
Non DFS Used	:	54.08 GB
DFS Remaining	:	96.95 GB
DFS Used%	:	0 %
DFS Remaining%	:	64.19 %
<a href="#">Live Nodes</a>	:	3
<a href="#">Dead Nodes</a>	:	0
<a href="#">Decommissioning Nodes</a>	:	0
Number of Under-Replicated Blocks	:	0

## NameNode Storage:

Storage Directory	Type	State
/data/name	IMAGE_AND_EDITS	Active
/data/backup	IMAGE_AND_EDITS	Active

This is [Apache Hadoop](#) release 1.1.2

## # DataNode 노드 리스트 확인

<http://master.namenode:50070/dfsnodelist.jsp?whatNodes=LIVE>



## NameNode 'master:9000'

**Started:** Wed Mar 27 15:52:42 KST 2013  
**Version:** 1.1.2, r1440782  
**Compiled:** Thu Jan 31 02:03:24 UTC 2013 by hortonfo  
**Upgrades:** There are no upgrades in progress.

[Browse the filesystem](#)  
[Namenode Logs](#)  
[Go back to DFS home](#)

Live Datanodes : 3

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
<a href="#">localhost</a>	2	In Service	41.7	0	36.74	4.96	0	<div></div>	11.89	3
<a href="#">localhost</a>	0	In Service	98.43	0	12.64	85.79	0	<div></div>	87.16	1
<a href="#">slave</a>	2	In Service	10.9	0	4.7	6.2	0.01	<div></div>	56.88	2

This is [Apache Hadoop](#) release 1.1.2

## # DataNode 노드 정보 확인

<http://slave.datanode01:50075/browseDirectory.jsp?namenodeInfoPort=50070&dir=%2F>

### Contents of directory [/](#)

Goto :  × go

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
<a href="#">tmp</a>	dir				2013-03-27 15:53	rwxr-xr-x	hadoop	supergroup
<a href="#">user</a>	dir				2013-03-27 15:54	rwxr-xr-x	hadoop	supergroup

[Go back to DFS home](#)

### Local logs

[Log](#) directory

This is [Apache Hadoop](#) release 1.1.2

## # 노드의 파일 정보 확인

Contents of directory [/user/hadoop](#)

Goto :

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
<a href="#">cloud computing primer.pdf</a>	file	629.43 KB	2	64 MB	2013-03-27 15:55	rw-r--r--	hadoop	supergroup
<a href="#">oracle plsql language pocket reference fourth edition.pdf</a>	file	849.04 KB	2	64 MB	2013-03-27 15:55	rw-r--r--	hadoop	supergroup

[Go back to DFS home](#)

Local logs

[Log](#) directory

This is [Apache Hadoop](#) release 1.1.2

Goto :

[Advanced view/download options](#)[View Next chunk](#)

Chunk size to view (in bytes, up to file's DFS block size):

```
4      Total number of blocks: 1
5      -1479047574059270437: 192.168.1.113:50010 192.168.1.112:50010
6
7      Total number of blocks: 1
8      -8675812117651072288: 192.168.1.112:50010 192.168.1.111:50010
```

# 한개의 파일의 사이즈가 64MB 를 초과할 경우, 블록을 64MB 단위로 나누어 저장  
> 기본 1블록(64MB)

```
1      // 3개의 블록에 2개씩의 복제
2      Total number of blocks: 3
3
4      -527122170720419961: 192.168.1.111:50010 192.168.1.113:50010
5      8606886155927734038: 192.168.1.112:50010 192.168.1.113:50010
6      3788523866326037174: 192.168.1.111:50010 192.168.1.113:50010
```

[# 모든 설정 후 start-all.sh 구동시 각각의 DataNode 서버에서 TaskTracker, DataNode 구동이 되지 않을 경우 참고](#)

참고 사이트 : <http://blog.beany.co.kr/archives/1373>

출처: <<http://develop.sunshiny.co.kr/868>>

## [참고]방화벽 설정

2017년 11월 23일 목요일 오전 10:07

```
manager@master:~$ sudo ufw enable
명령은 존재하는 ssh 연결에 피해를 줄 수 있습니다. 이 작업과 함께 진행하시겠습니까(y|n)? y
방화벽이 활성화 상태이며 시스템이 시작할 때 사용됩니다
manager@master:~$ sudo ufw allow 22
규칙이 추가되었습니다
규칙이 추가되었습니다 (v6)
manager@master:~$ sudo ufw allow 50070
규칙이 추가되었습니다
규칙이 추가되었습니다 (v6)
manager@master:~$ sudo ufw status verbose
상태: 활성
로깅: on (low)
기본 설정: allow (내부로 들어옴), allow (외부로 나감), disabled (라우팅 된)
새 프로필: skip

목적                동작                출발
--                --                --
22                  ALLOW IN            Anywhere
50070                ALLOW IN            Anywhere
22 (v6)             ALLOW IN            Anywhere (v6)
50070 (v6)          ALLOW IN            Anywhere (v6)
```

```
manager@master:~$ sudo reboot
```

```
sudo ufw enable
sudo ufw allow 22
sudo ufw allow 50070
sudo ufw status verbose
sudo reboot
```

## [참고]

2017년 11월 24일 금요일    오후 12:52

우분투에서 apt upadte 시 아래와 같은 메시지를 만난다면...

~> sudo apt-get update E: /var/lib/apt/lists/lock 잠금 파일을 얻을 수 없습니다 - open (11: 자원이 일시적으로 사용 불가능함) E: /var/lib/apt/lists/ 디렉터리를 잠글 수 없습니다 E: /var/lib/dpkg/lock 잠금 파일을 얻을 수 없습니다 - open (11: 자원이 일시적으로 사용 불가능함) E: 관리 디렉터리를 (/var/lib/dpkg/) 잠글 수 없습니다. 다른 프로세스가 사용하고 있지 않습니까?

```
$sudo rm -rf /var/lib/apt/lists/lock
```

```
$sudo rm -rf /var/lib/dpkg/lock
```

문제는 apt list 진행 중 강제로 종료할 때 많이 발생한다.

이럴 경우 lock이라는 파일을 제거해주고 다시 진행하면 된다.

apt install 할 때도 비슷 맥락의 에러가 나타난다면

메시지를 보고 lock의 경로를 제거해주면 된다.

출처: [http://blog.naver.com/PostView.nhn?blogId=chandong83&logNo=220806174408&beginTime=0&jumpingVid=&from=search&redirect=Log&widgetTypeCall=true&topReferer=http%3A%2F%2Fsearch.naver.com%2Fsearch.naver%3Fwhere%3Dnsearch%26sm%3Dtop\\_hy%26fbm%3D0%26ie%3Dutf8%26query%3D%25EC%259E%2590%25EC%259B%2590%25EC%259D%25B4%2B%25EC%259D%25BC%25EC%258B%259C%25EC%25A0%2581%25EC%259C%25BC%25EB%25A1%259C%2B%25EC%2582%25AC%25EC%259A%25A9%2B%25EB%25B6%2588%25EA%25B0%2580%25EB%258A%25A5%25ED%2595%25A8%26url%3Dhttp%253A%252F%252Fchandong83.blog.me%252F220806174408%26ucs%3Dk1DxuPqxtY%252FY&directAccess=false](http://blog.naver.com/PostView.nhn?blogId=chandong83&logNo=220806174408&beginTime=0&jumpingVid=&from=search&redirect=Log&widgetTypeCall=true&topReferer=http%3A%2F%2Fsearch.naver.com%2Fsearch.naver%3Fwhere%3Dnsearch%26sm%3Dtop_hy%26fbm%3D0%26ie%3Dutf8%26query%3D%25EC%259E%2590%25EC%259B%2590%25EC%259D%25B4%2B%25EC%259D%25BC%25EC%258B%259C%25EC%25A0%2581%25EC%259C%25BC%25EB%25A1%259C%2B%25EC%2582%25AC%25EC%259A%25A9%2B%25EB%25B6%2588%25EA%25B0%2580%25EB%258A%25A5%25ED%2595%25A8%26url%3Dhttp%253A%252F%252Fchandong83.blog.me%252F220806174408%26ucs%3Dk1DxuPqxtY%252FY&directAccess=false)