



빅데이터 시스템 구축 및 딥러닝 분석

- 분산환경 구축 -

2022. 3. 29

정 준 수 PhD

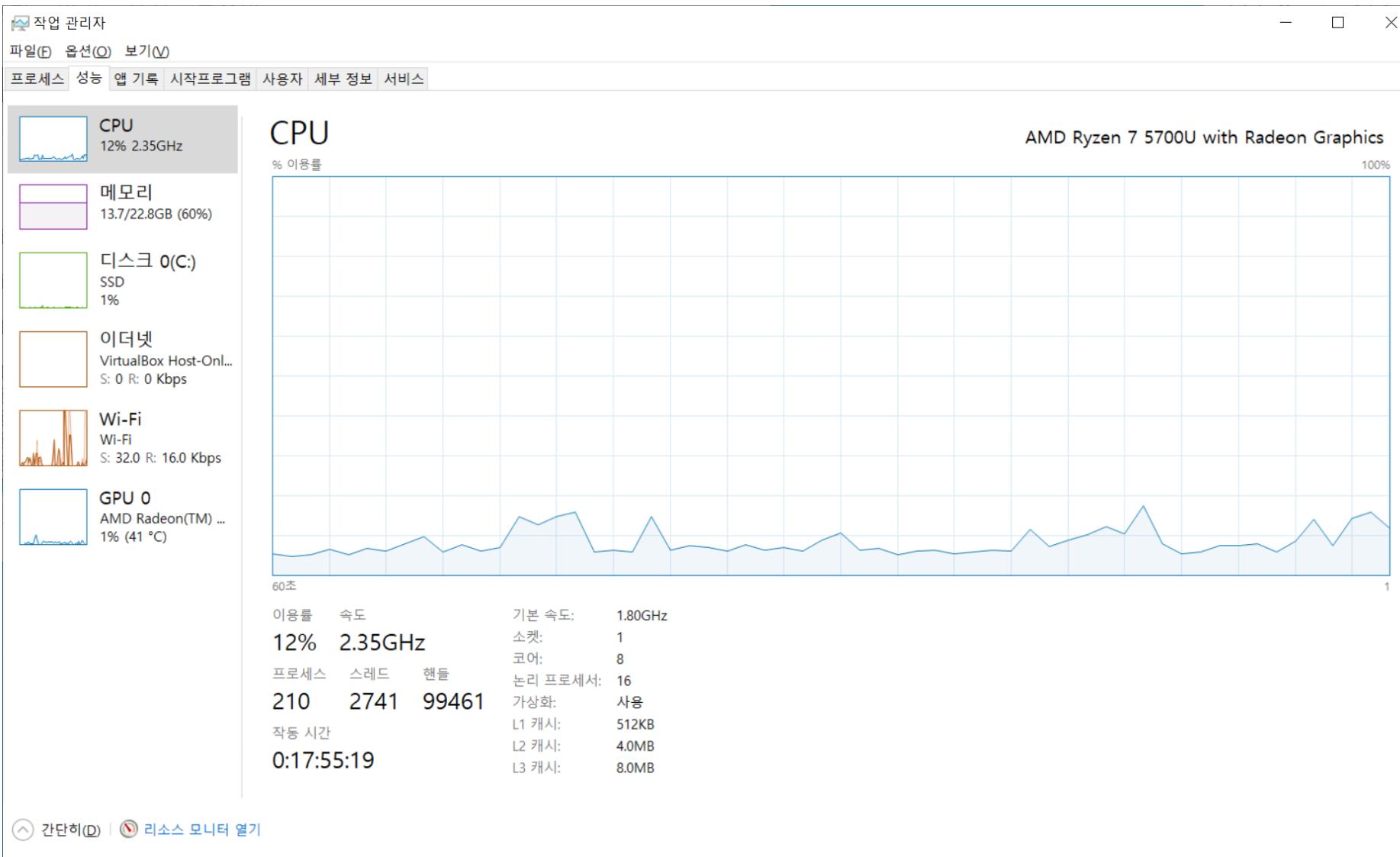
파일럿 프로젝트의 하드웨어 환경

파일럿 환경	CPU	메모리	디스크	비고
저사양 PC	듀얼코어 이상	8GB 이상(여유7GB)	90GB 이상	디스크 SSD 권장
고사양 PC	i5 이상	16GB 이상(여유 15GB)	120GB 이상	

고사양 파일럿 아키텍처는 오라클 버추얼 박스(Oracle Virtual Box)를 이용해 3대의 가상 머신을 만들고 CentOS 리눅스 서버로 구성하고, 저사양의 아키텍처는 2대의 가상 머신 (Server01, Server02)만을 이용하게 된다.

파일럿 프로젝트의 환경을 구성하기 위해 3대의 가상머신을 만들고, 가상머신에 총 17개의 소프트웨어를 설치한다.

시스템 요구 사항



파일럿 프로젝트 환경 구축

빅데이터 개발 환경 구성: 개인 윈도우 PC

1. 자바 설치
2. 이클립스 설치
3. 버추얼 박스 설치
4. 가상 머신에 CentOS 설치
5. 기타 도구 설치: PuTTY, 파일질라, 크롬, 예제 코드

빅데이터 서버 환경 구성: 개인 윈도우 PC 위의 리눅스 가상 머신 3대

6. 클라우데라 매니저(Cloudera Manager) 설치
7. 빅데이터 에코시스템 설치: 하둡, 주키퍼(Zookeeper) 등 기본 구성

1. 자바(JAVA) 설치

01. 먼저 웹 브라우저로 아래의 URL로 접속한다.

- 자바다운로드 페이지: <https://www.oracle.com/downloads/index.html>

02 다운로드 페이지에서 [Java] 선택 → [Java(JDK) for Developers] 선택 [Download] 선택한다(오라클 사이트가 리뉴얼되면 JDK 다운로드 위치와 버전이 변경될 수 있다. 파일럿 프로젝트 [Java SE 8uxxxx]에서 [JDK]에서는 Java SE 8uxxxx 버전이면 문제 없이 진행할 수 있다).

03. 설치가 완료되면 간단히 JAVA_HOME 환경변수를 설정해 보자(여기서는 윈도우 7 기준으로 설명 한다).

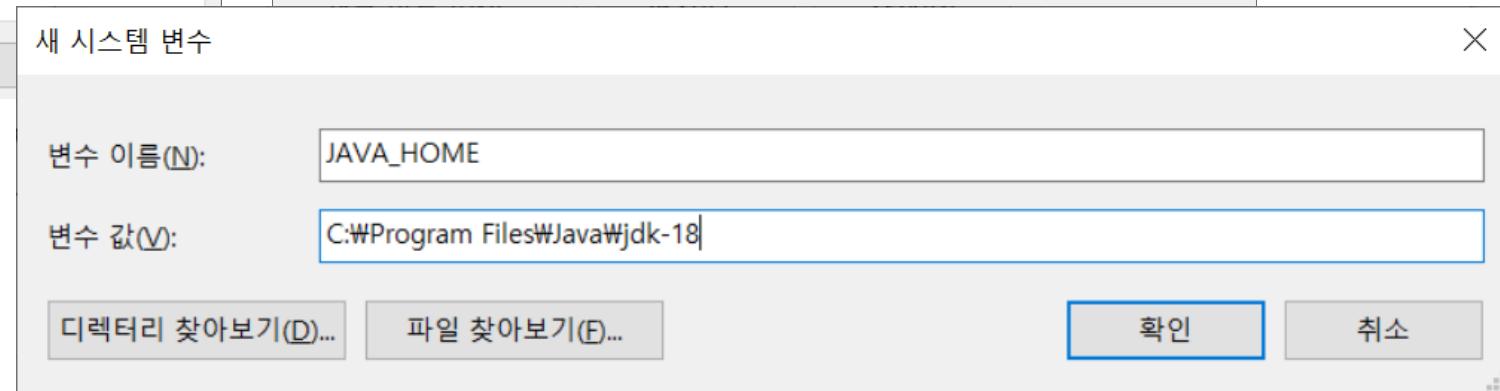
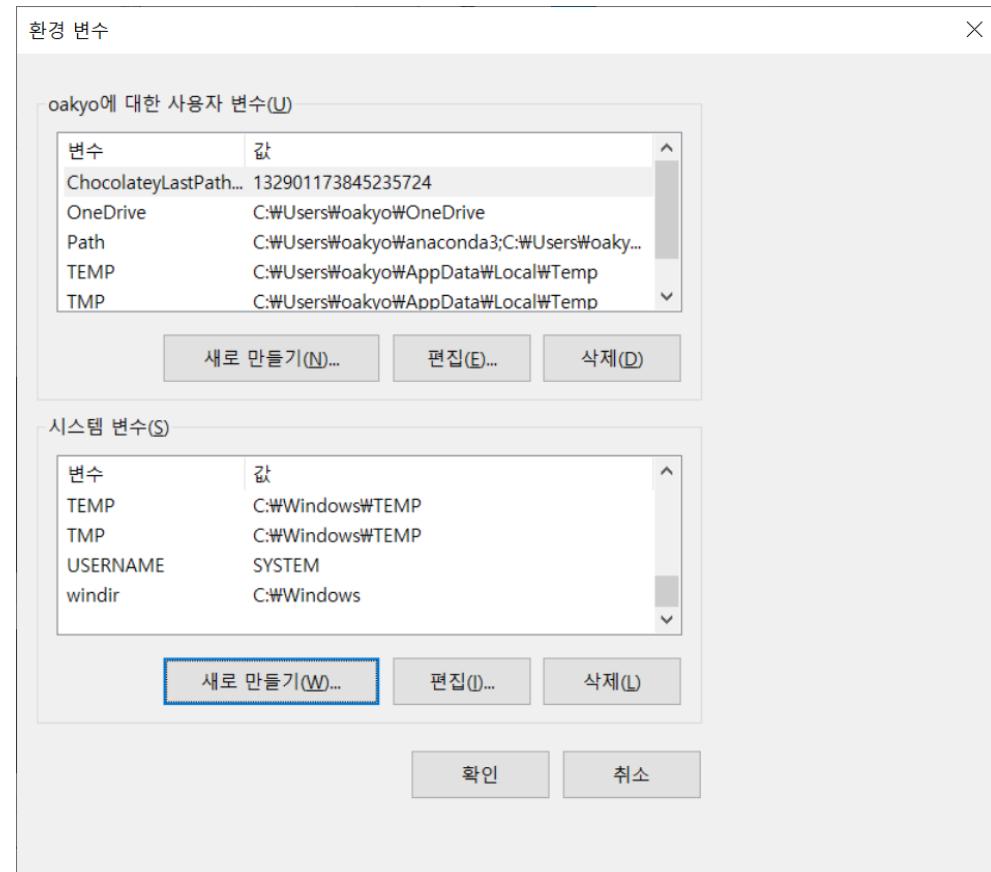
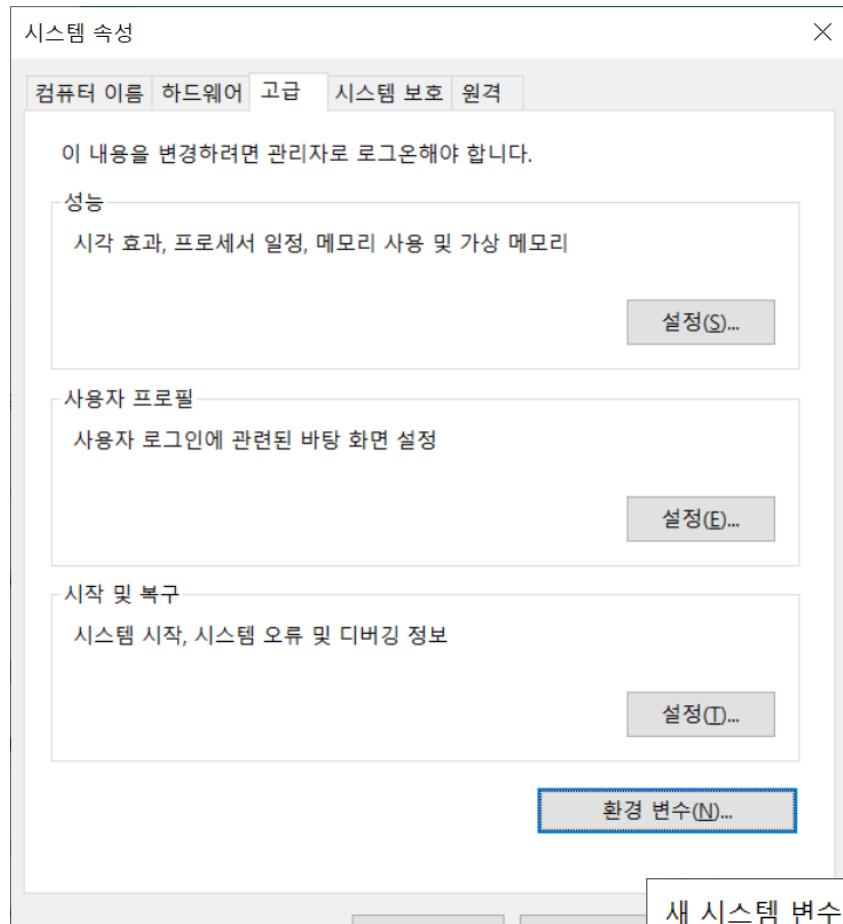
[제어판]- [시스템] → [고급 시스템 설정] [환경변수] 버튼을 차례로 클릭한다.

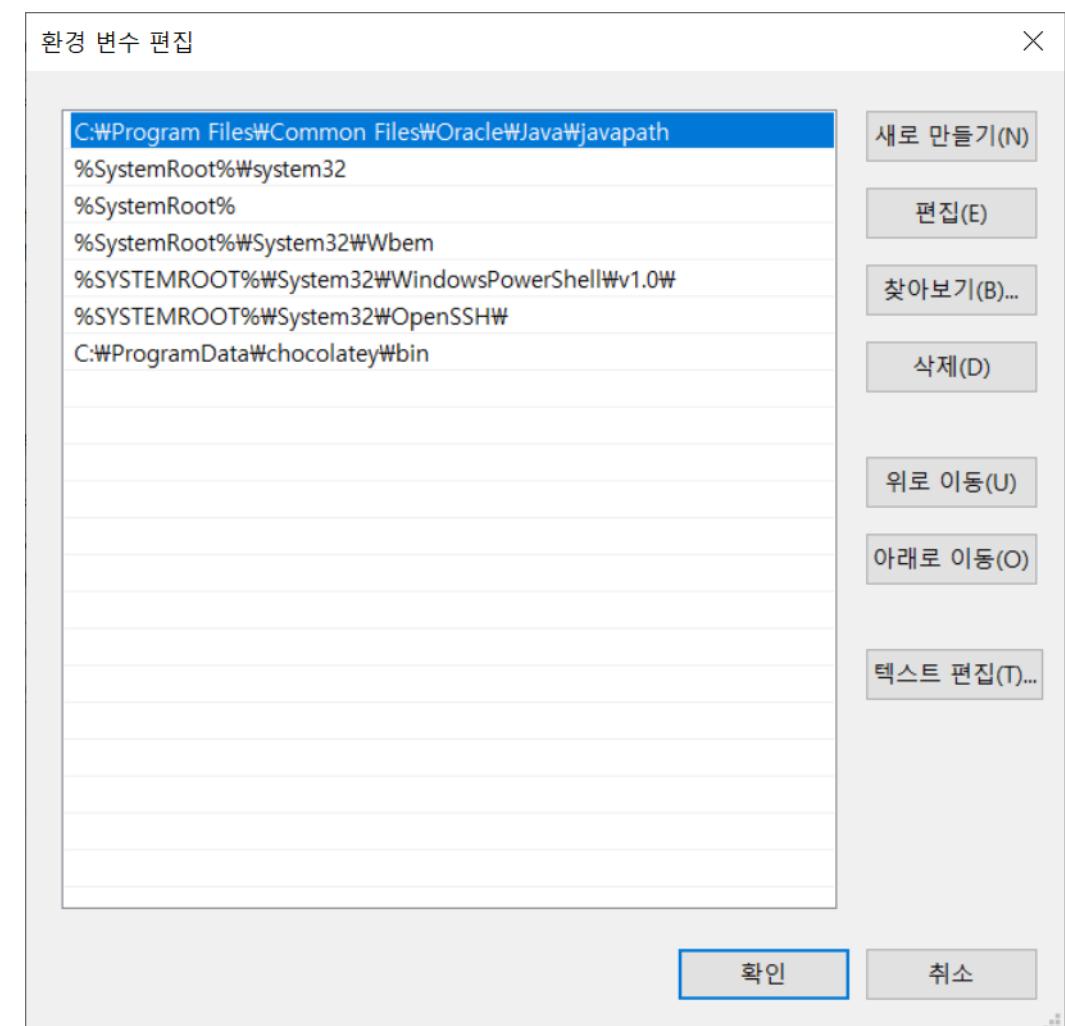
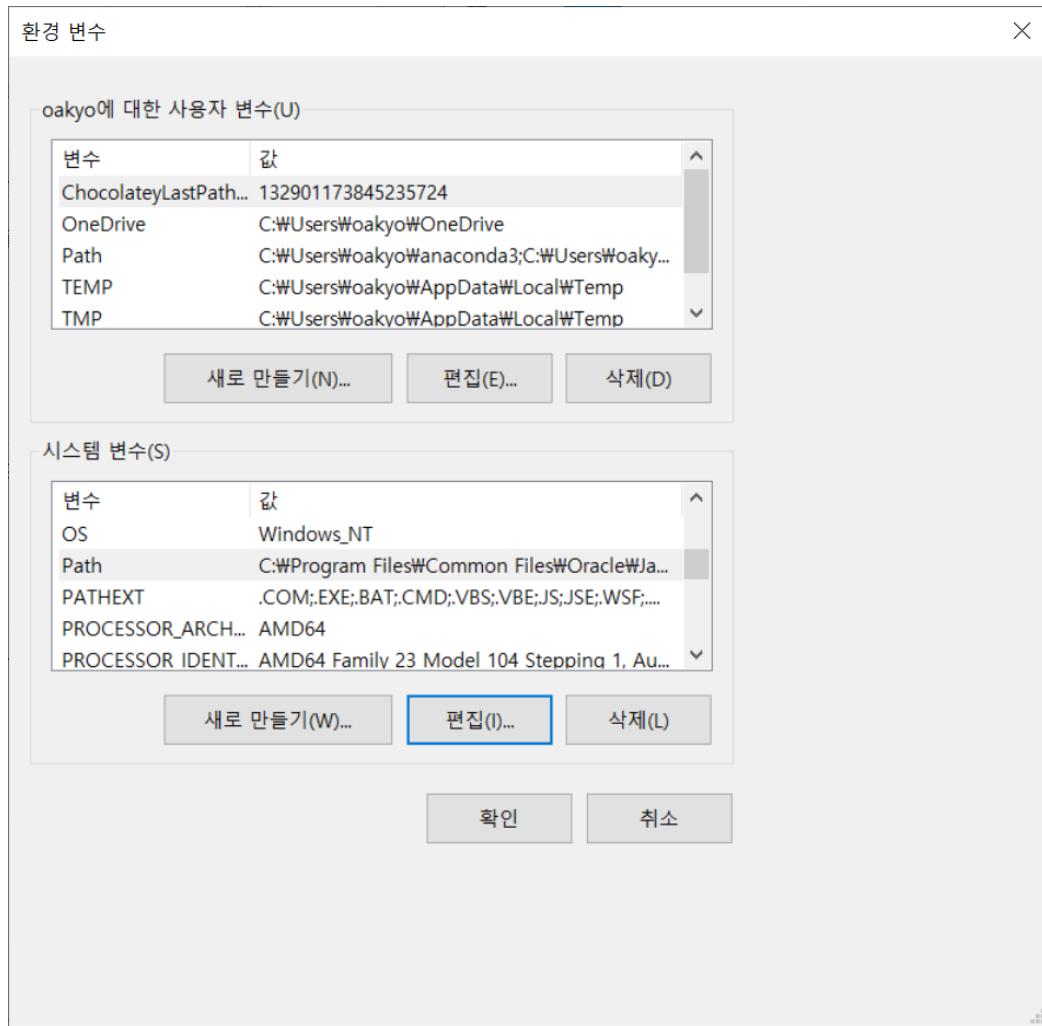
The screenshot shows a web browser window with the Oracle Java Downloads page open. The URL in the address bar is <https://www.oracle.com/java/technologies/downloads/#jdk18-windows>. The page header includes the Oracle logo and navigation links for Products, Industries, Resources, Customers, Partners, Developers, and Events. Below the header, there are links for Java downloads, Tools and resources, and Java archive. A banner at the top states "Java 18 will receive updates under reserved builds until at least September 2024". The main content area is titled "Java SE Development Kit 18 downloads" and includes a brief description of the JDK. It lists download options for Linux, macOS, and Windows. The Windows section is currently selected, showing three download links:

Product/file description	File size	Download
x64 Compressed Archive	172.54 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.zip (sha256)
x64 Installer	153.2 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.exe (sha256)
x64 MSI Installer	152.08 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.msi (sha256)

Below the download table, there is a section titled "JDK Script-friendly URLs" with a note: "The URLs listed above will remain the same for JDK update releases to allow their use in scripts." It also includes a link to "Learn more about automating the downloads of JDK". At the bottom of the page, a file download progress bar shows "jdk-18_windows-x...exe" at 108/153MB, 3초 남음.

<https://www.oracle.com/java/technologies/downloads/#jdk18-windows>





JAVA_HOME 환경변수를 설정한다. [시스템 변수의 새로 만들기]를 클릭해서 아래와 같이 설정한다.

- 변수 이름: JAVA_HOME
- 변수 값: C:\Program Files\Java\jdk1.8.0_241

변수 값의 경우 반드시 앞서 설치한 자바의 설치 디렉터리를 지정해야 한다.

- 변수 값에 추가할 내용 : %JAVA_HOME%\bin

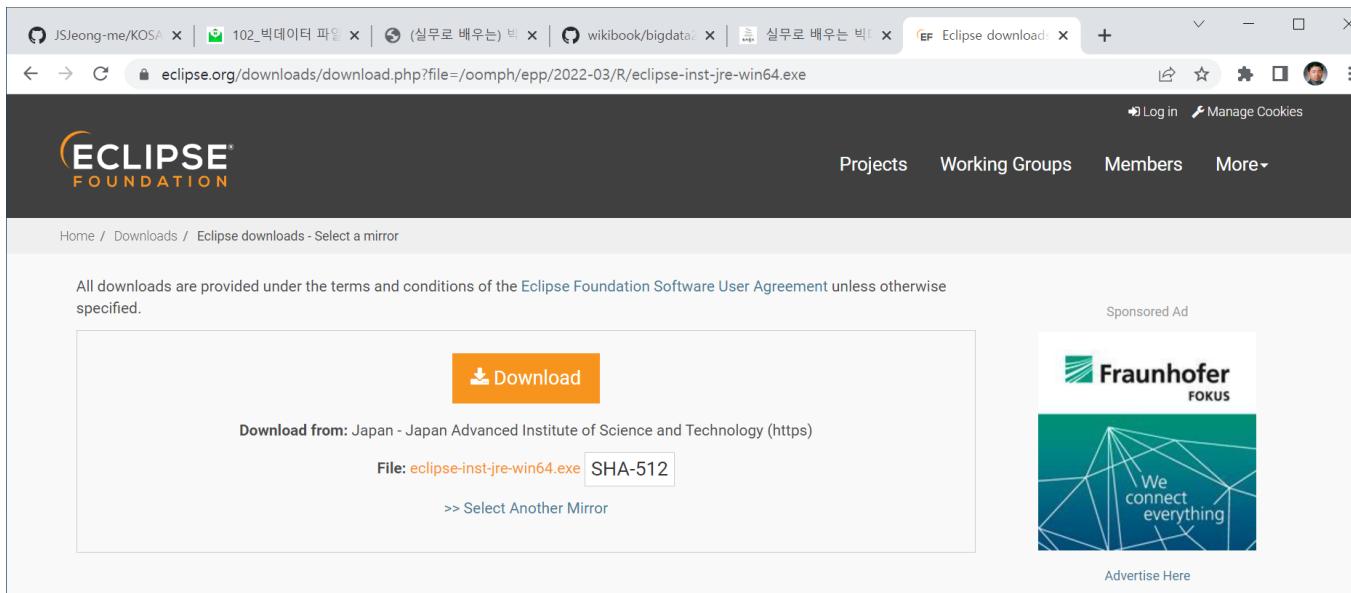
자바 버전 확인

C:> java --version

이클립스 설치

이클립스를 설치하려면 먼저 아래의 URL로 이동해서 PC 환경에 맞는 eclipse Eclipse IDE for Java EE Developers를 내려받는다. 이클립스를 설치 하려면 아카이빙 (zip 파일) 버전의 경우 압축 파일을 풀고 인스톨 버전(eclipse-inst-xxx.exe)의 경우 설치 파일을 실행하기만 하면 된다. 설치 방법은 간단하므로 지면상 상세한 내용은 생략한다. 인터넷 검색을 통해 많은 자료를 찾아볼 수 있으니 참고하기 바란다.

- 이클립스 다운로드
- <https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2022-03/R/eclipse-inst-jre-win64.exe>



오라클 버추얼 박스 설치

버추얼 박스 역시 설치 방법이 간단하다. 아래의 URL을 통해 설치 VirtualBox 파일을 다운로드한 후 실행하면 안내에 따라 쉽게 설치할 수 있으므로 버추얼 박스의 상세 설치 과정도 생략한다.

오라클 버추얼 박스 다운로드 페이지: <https://www.virtualbox.org/>

(VirtualBox는 하드웨어와 OS의 특성을 많이 타는 소프트웨어다. 파일럿 환경에서 문제가 발생할 경우 많은 해결 사례들을 인터넷 상에서 찾아볼 수 있으니 참고하기 바란다.)

-  일반
-  입력
-  업데이트
-  언어
-  디스플레이
-  네트워크
-  확장
-  프록시

네트워크

NAT 네트워크(N)

활성화됨 | 이름



NatNetwork

네트워크 이름: NatNetwork

네트워크 CIDR: 10.0.2.0/24

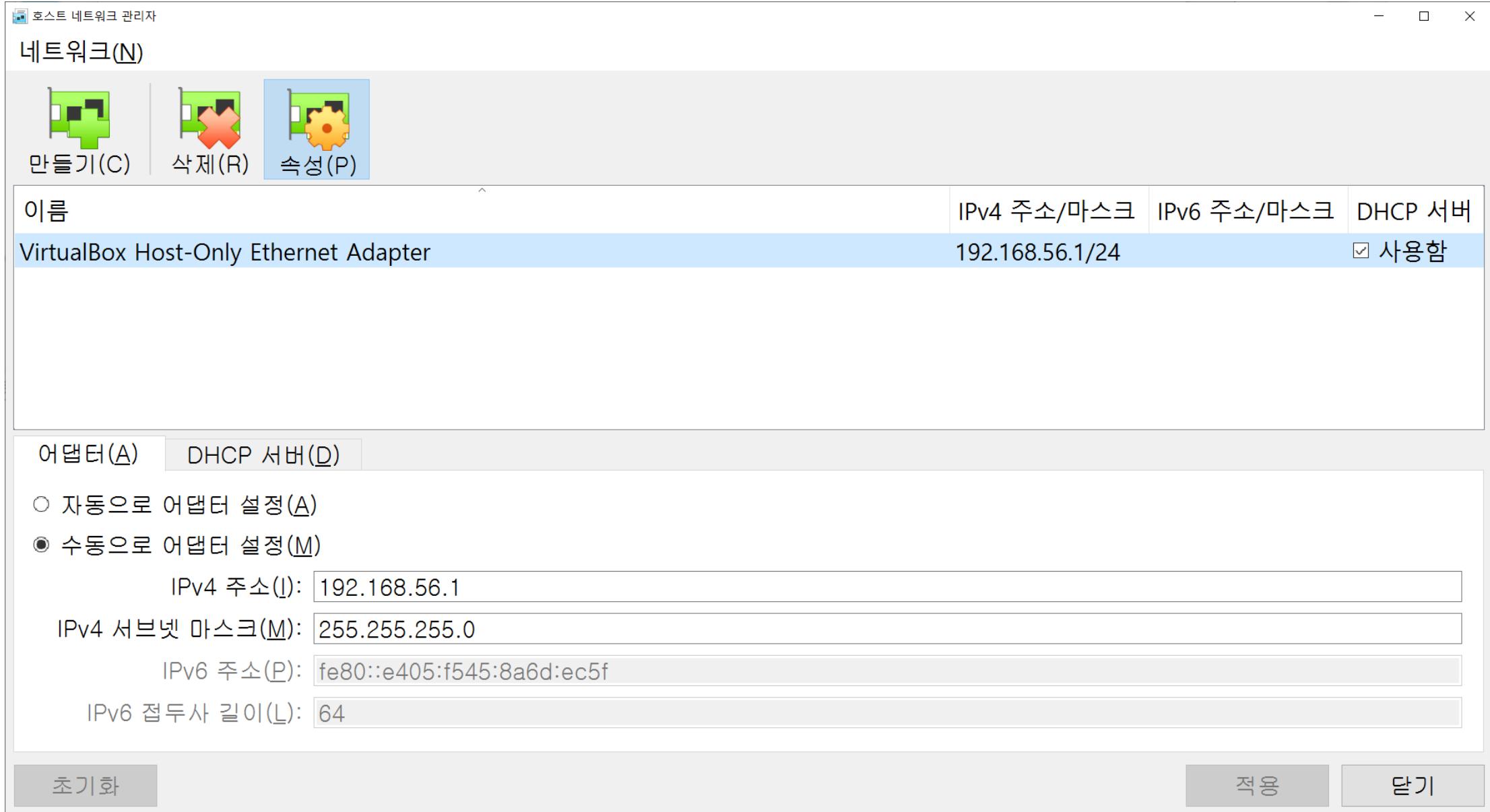
DHCP 지원: 예

IPv6 지원: 아니요



확인

취소



호스트 네트워크 관리자

네트워크(N)

만들기(C) 삭제(R) 속성(P)

이름	IPv4 주소/마스크	IPv6 주소/마스크	DHCP 서버
VirtualBox Host-Only Ethernet Adapter	192.168.56.1/24		<input checked="" type="checkbox"/> 사용함

어댑터(A) DHCP 서버(D)

서버 사용함(E)

서버 주소(R): 192.168.56.100

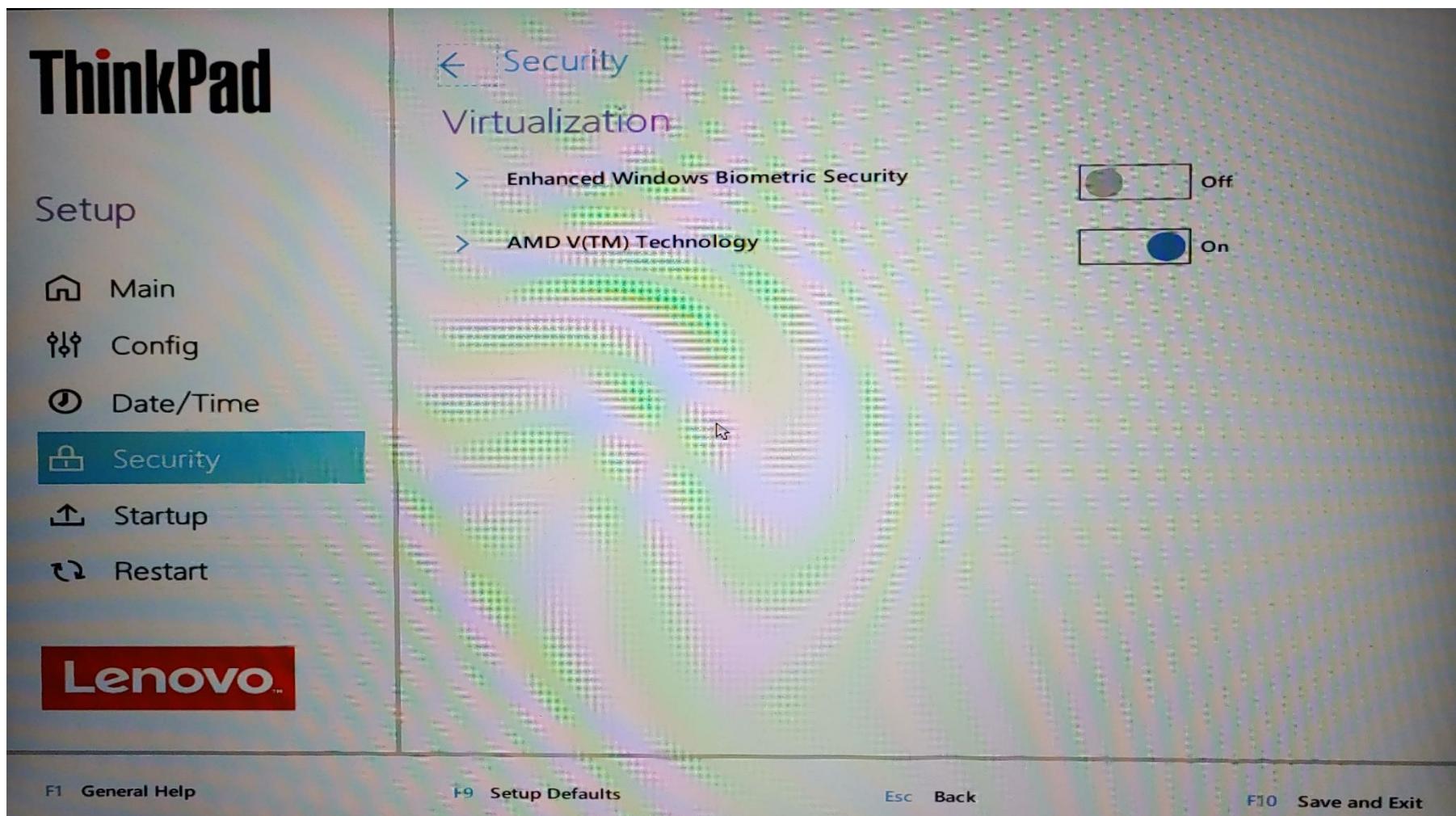
서버 마스크(M): 255.255.255.0

최저 주소 한계(L): 192.168.56.101

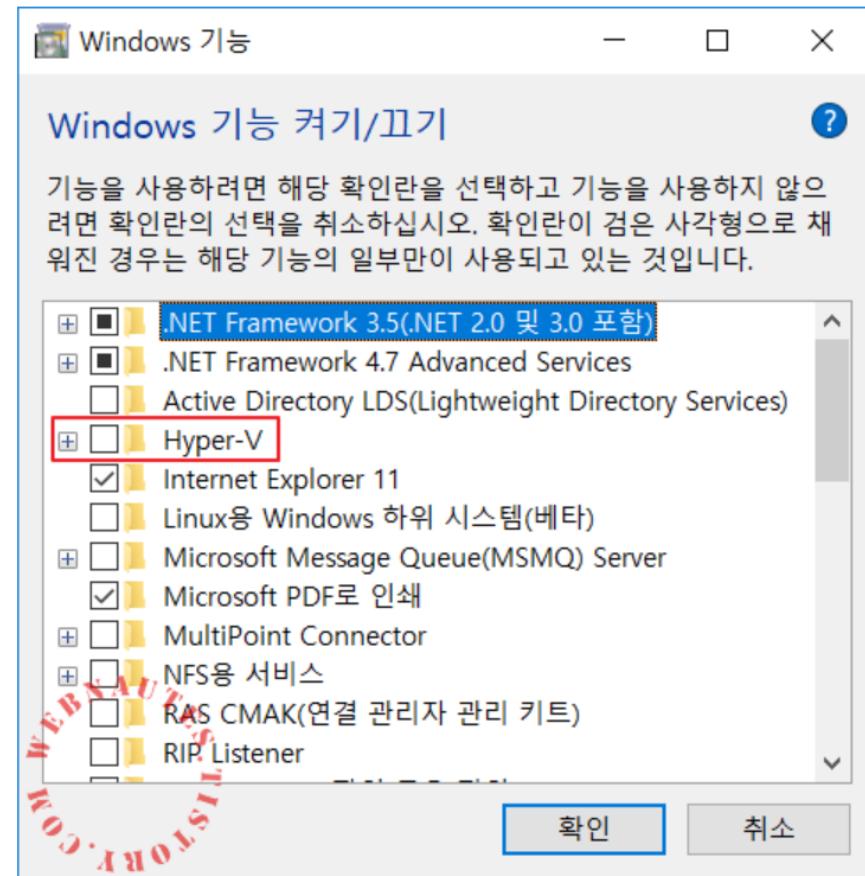
최고 주소 한계(U): 192.168.56.254

초기화 적용 닫기

PC Bios 가상화 활성 확인



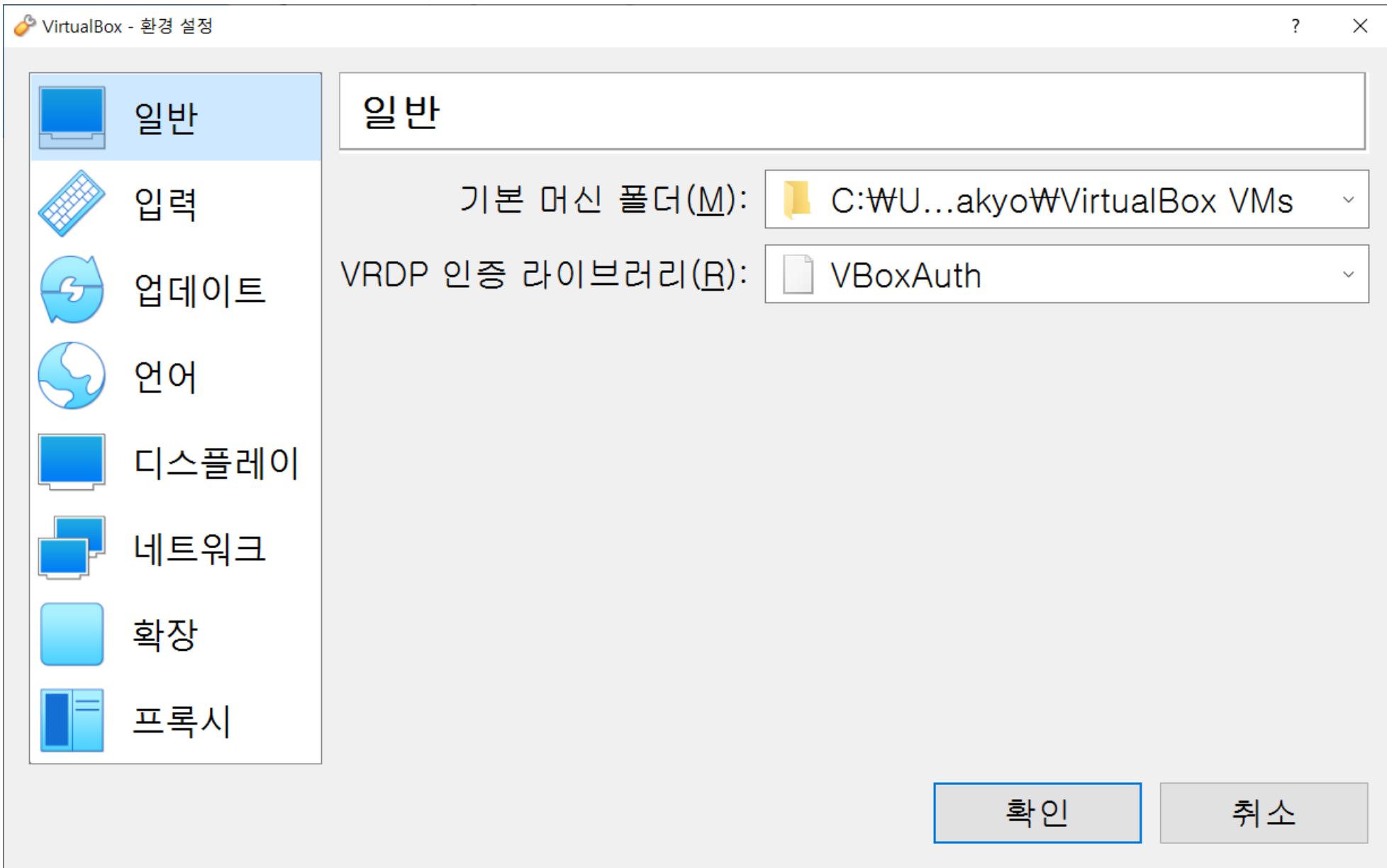
4. Hyper-V 항목을 체크 해제하고 확인을 클릭합니다. 윈도우를 재부팅을 해주어야 적용이 됩니다.

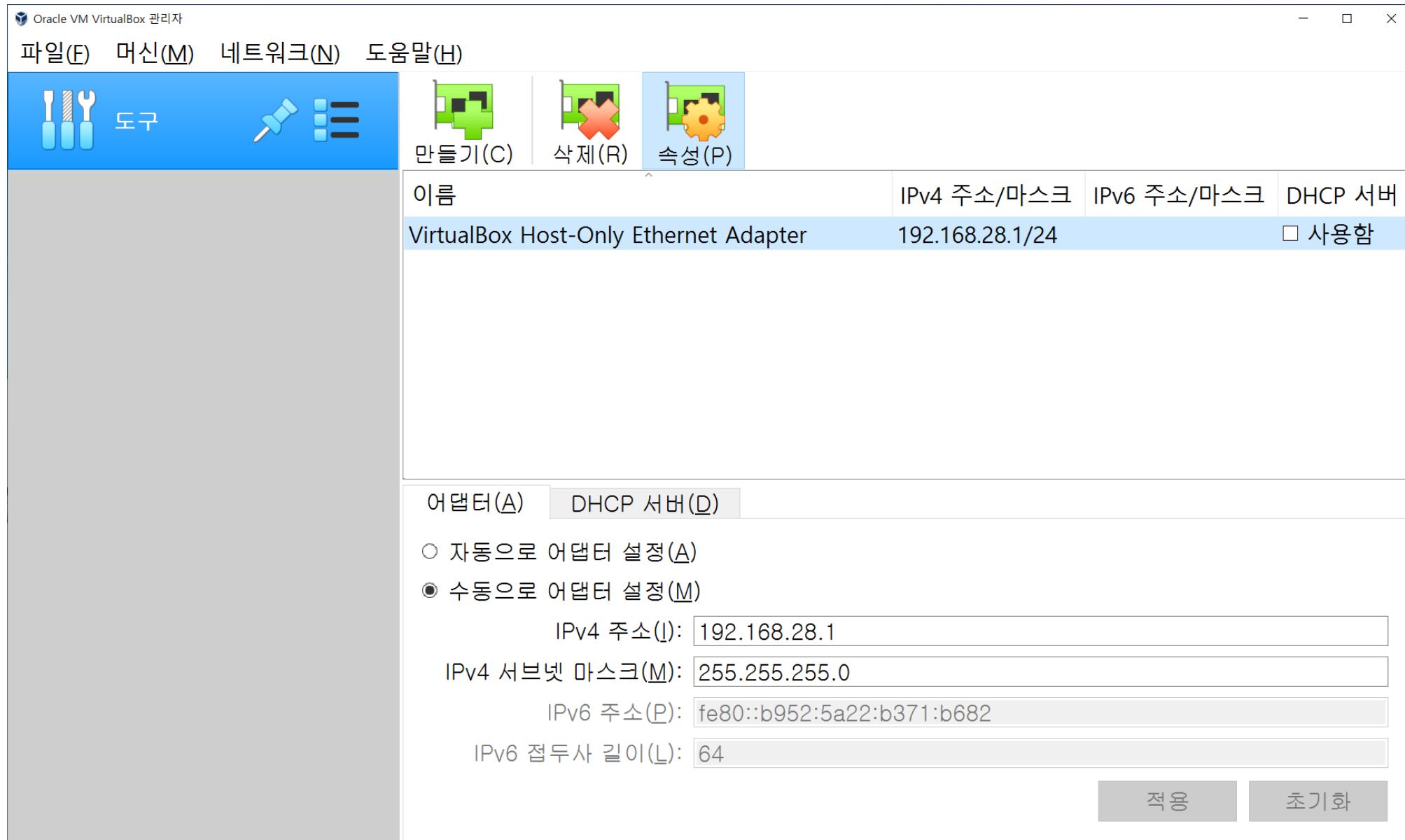


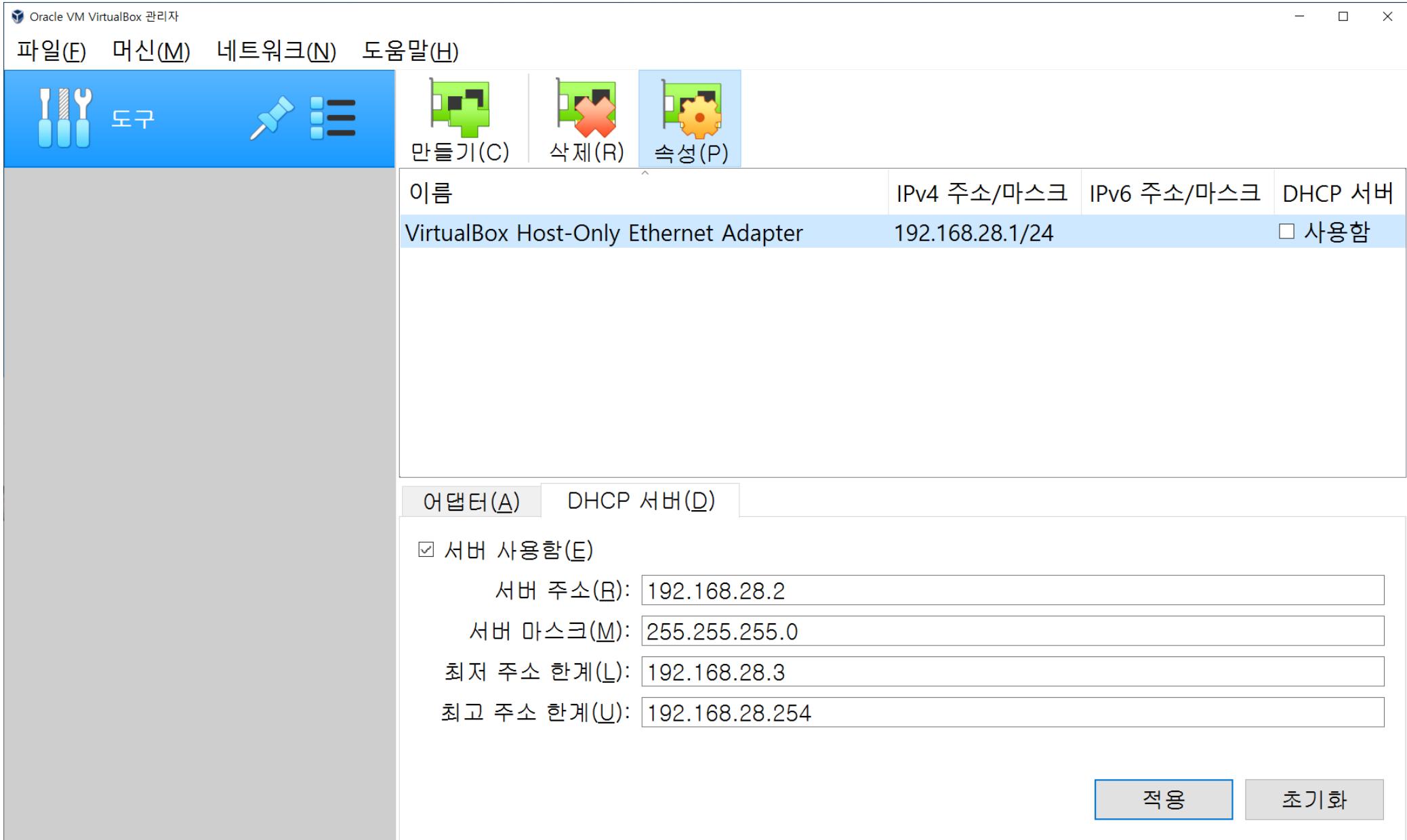
리눅스 가상 머신 환경 구성

05. CentOS 7.x 설치 파일을 아래 URL에서 내려받는다. 파일 크기가 1GB에 가까우므로 다운로드가 완료될 때까지 시간이 좀 걸릴 수 있다(파일럿 환경에서는 7.9 버전을 사용한다)

- CentOS 7.x 다운로드 페이지: <https://www.centos.org/centos-linux/>
- Debian <http://ftp.harukasan.org/debian-cd/11.3.0-live/i386/iso-hybrid/>









가상 머신 만들기

이름 및 운영 체제

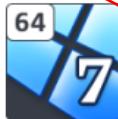
새 가상 머신을 나타내는 이름과 저장할 대상 폴더를 입력하고 설치할 운영 체제를 선택하십시오. 입력한 이름은 VirtualBox에서 가상 머신을 식별하는 데 사용됩니다.

이름:

머신 폴더:

 C:\Users\Woakyow\VirtualBox VMs

종류(I):

 Microsoft Windows

버전(V):

 Windows 7 (64-bit)

Linux

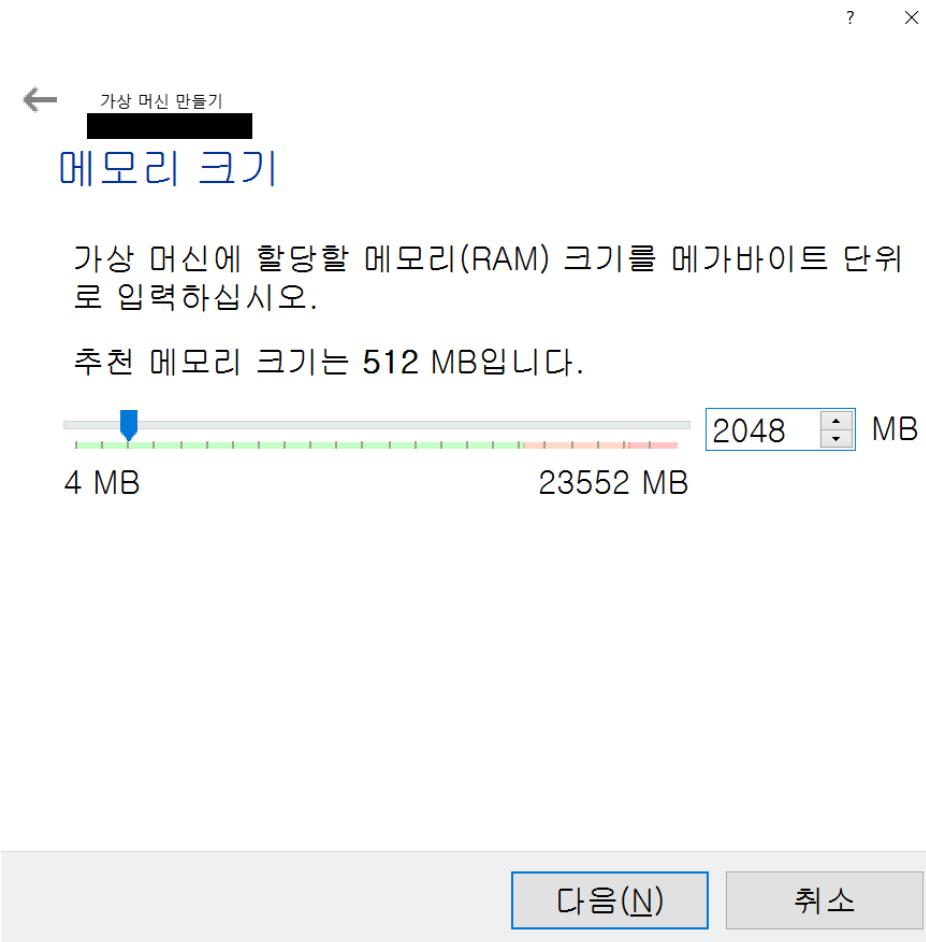
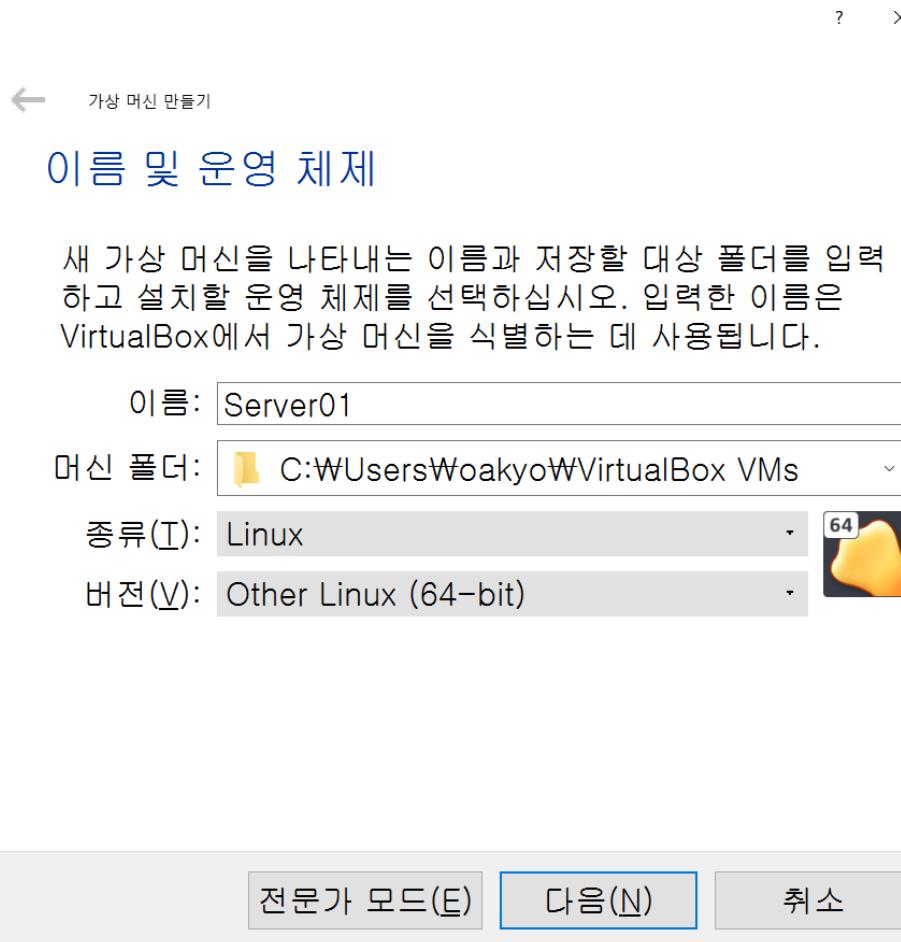
Other Linux (64x)

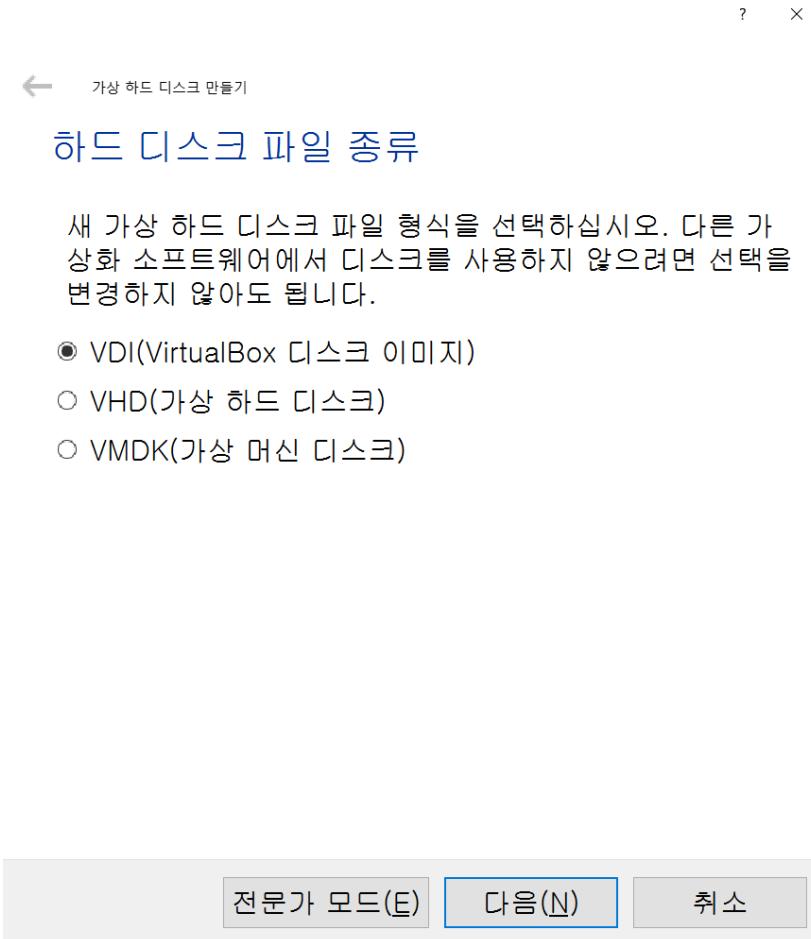
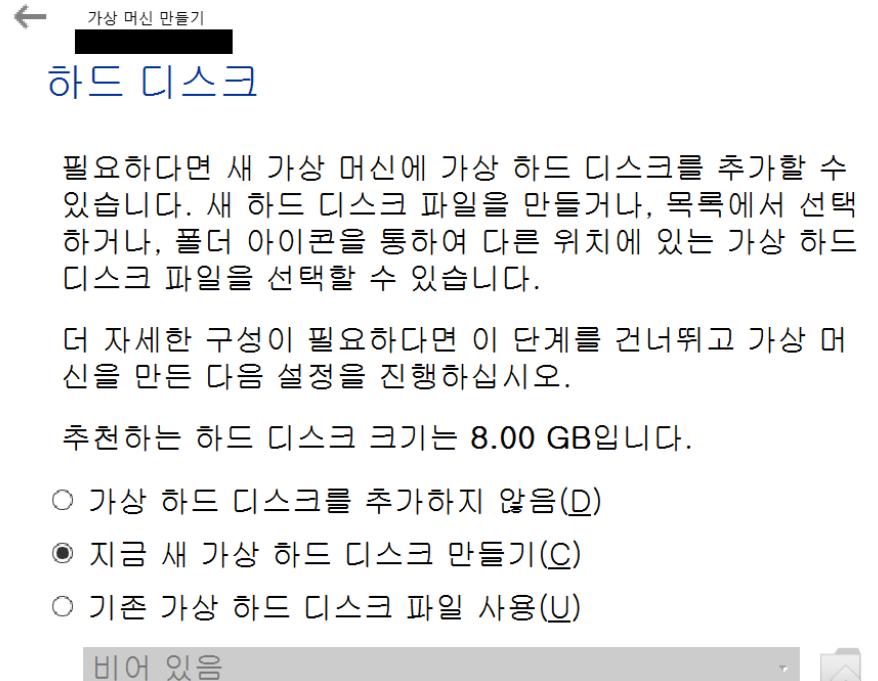
로 변경

전문가 모드(E)

다음(N)

취소





? X

← 가상 하드 디스크 만들기

물리적 하드 드라이브에 저장

새 가상 하드 디스크 파일을 사용하는 대로 커지게 할 것인지(동적 할당) 최대 크기로 만들 것인지(정적 할당) 선택하십시오.

동적 할당 하드 디스크 파일은 가상 디스크를 사용할 때 고정된 최대 크기까지 파일 크기가 커지지만, 사용량이 줄어들어도 자동적으로 작아지지는 않습니다.

고정 크기 하드 디스크 파일은 만드는 데 더 오래 걸리지만 사용할 때 더 빠릅니다.

동적 할당(D)

고정 크기(F)

다음(N) 취소

? X

← 가상 하드 디스크 만들기

파일 위치 및 크기

새 가상 하드 디스크 파일의 이름을 아래 상자에 입력하거나 폴더 아이콘을 클릭해서 파일을 생성할 폴더를 지정할 수 있습니다.

s:\oakyo\VirtualBox VMs\Server01\Server01.vdi

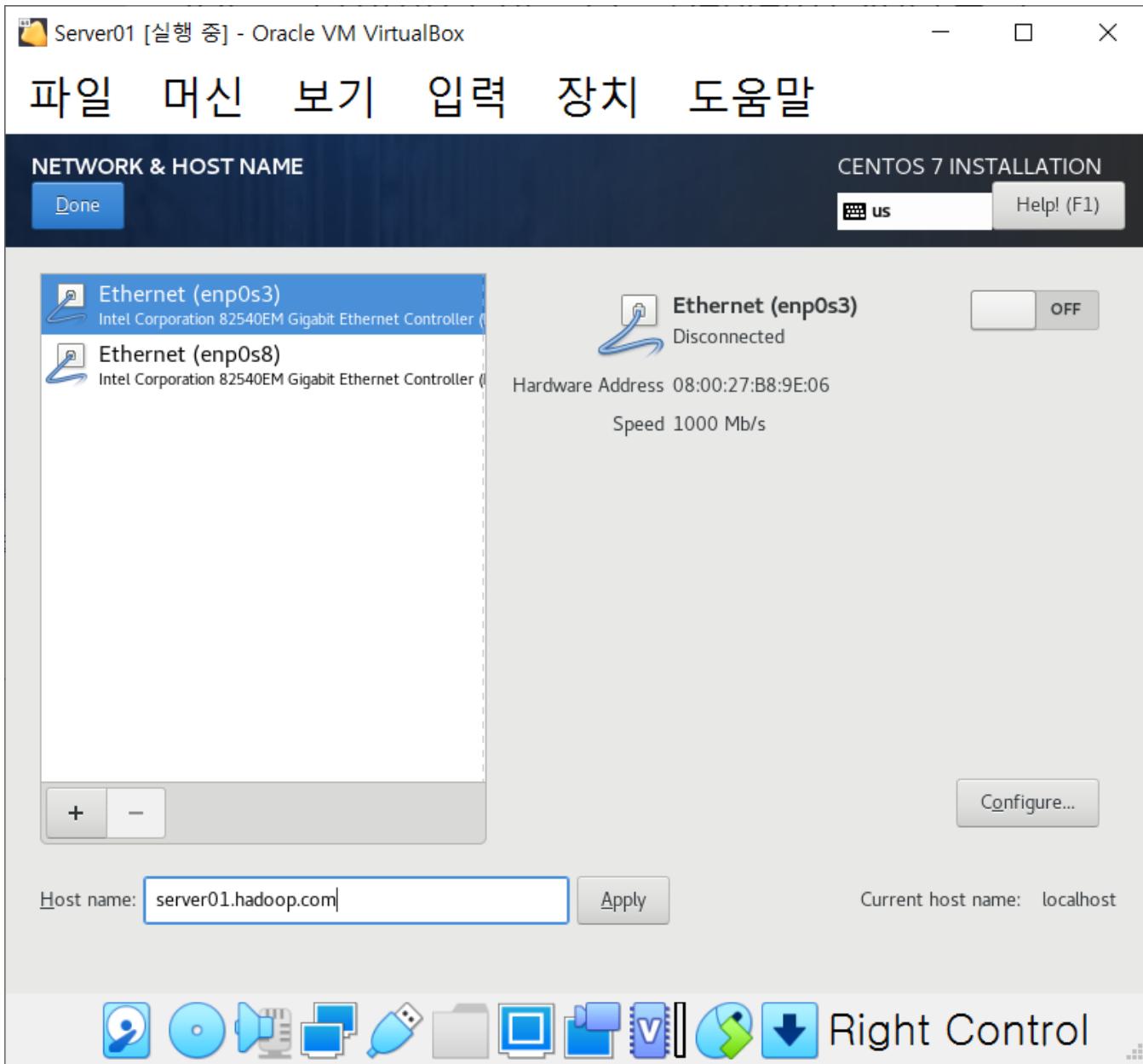
새 가상 하드 디스크 크기를 메가바이트 단위로 입력하십시오. 가상 머신에서 가상 하드 드라이브에 저장할 수 있는 데이터의 최대 크기입니다.



4.00 MB 30.00 GB 2.00 TB

만들기 취소





-  일반
-  시스템
-  디스플레이
-  저장소
-  오디오
-  네트워크
-  직렬 포트
-  USB
-  공유 폴더
-  사용자 인터페이스

네트워크

어댑터 1 어댑터 2 어댑터 3 어댑터 4

네트워크 어댑터 사용하기(E)

다음에 연결됨(A): NAT

이름(N):

▼ 고급(D)

어댑터 종류(I): Intel PRO/1000 MT Desktop(82540EM)

무작위 모드(P): 거부

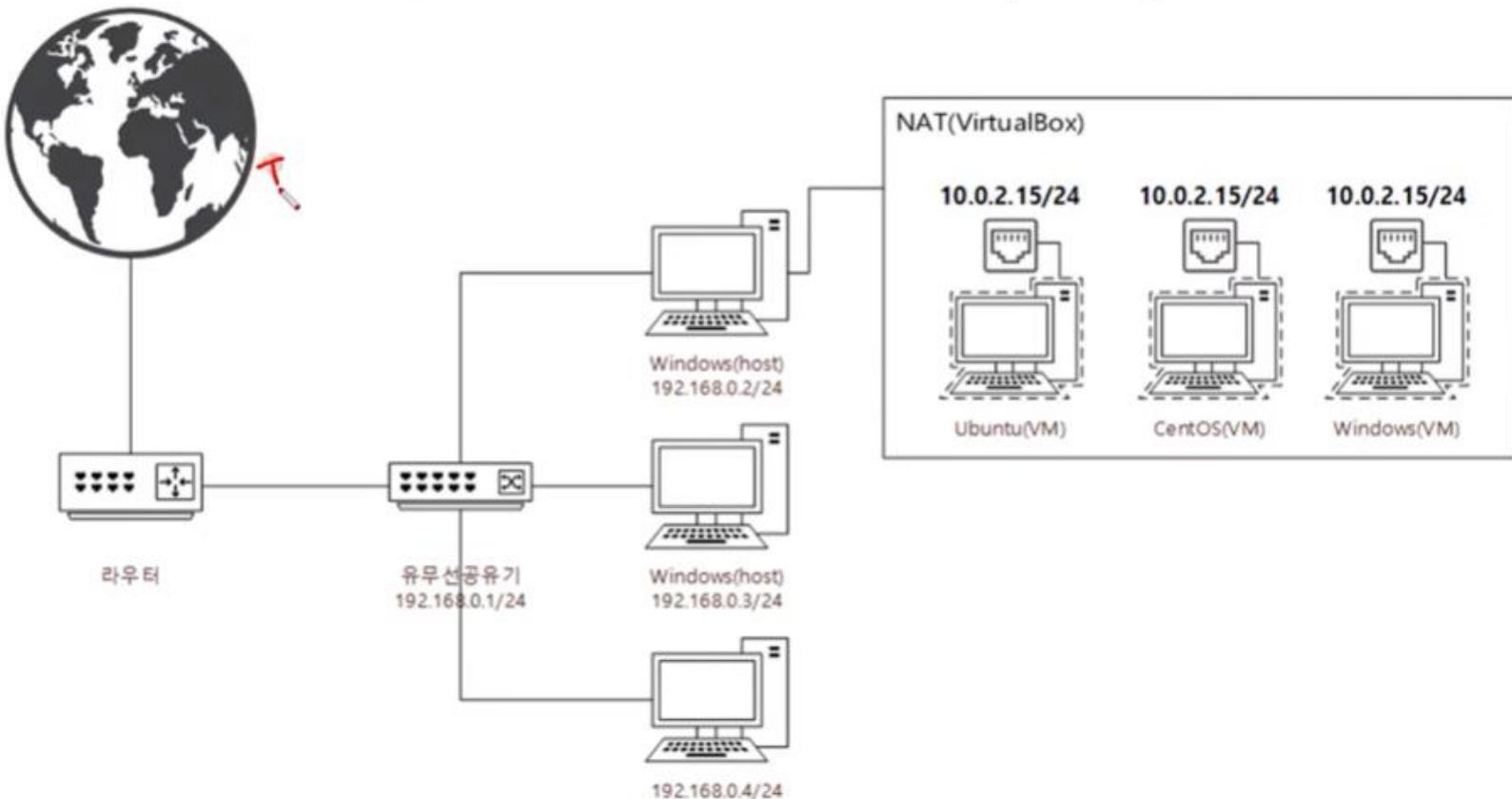
MAC 주소(M): 08002780A1FC

케이블 연결됨(C)

포트 포워딩(P)

확인 취소

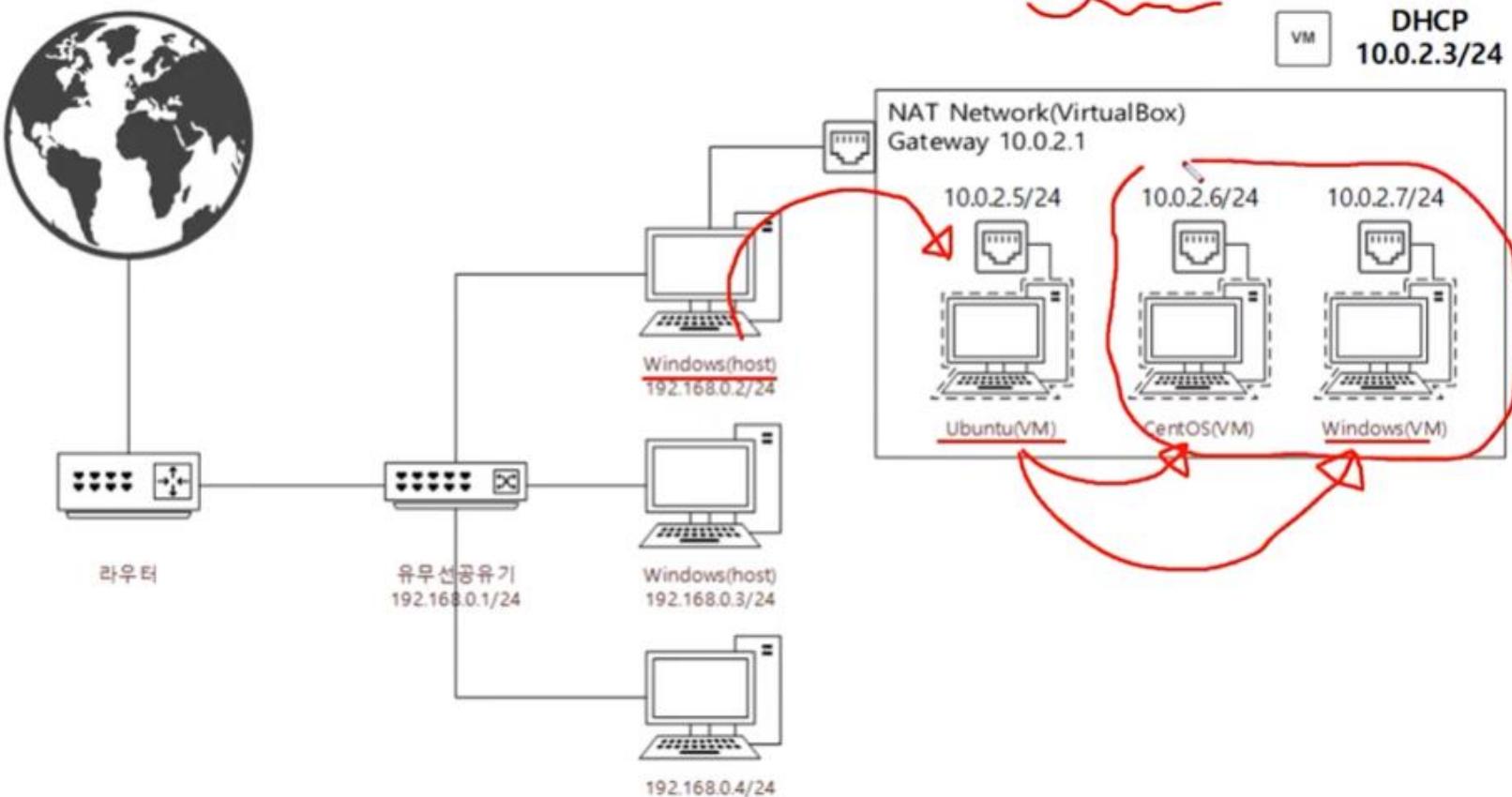
VirtualBox Network (NAT)



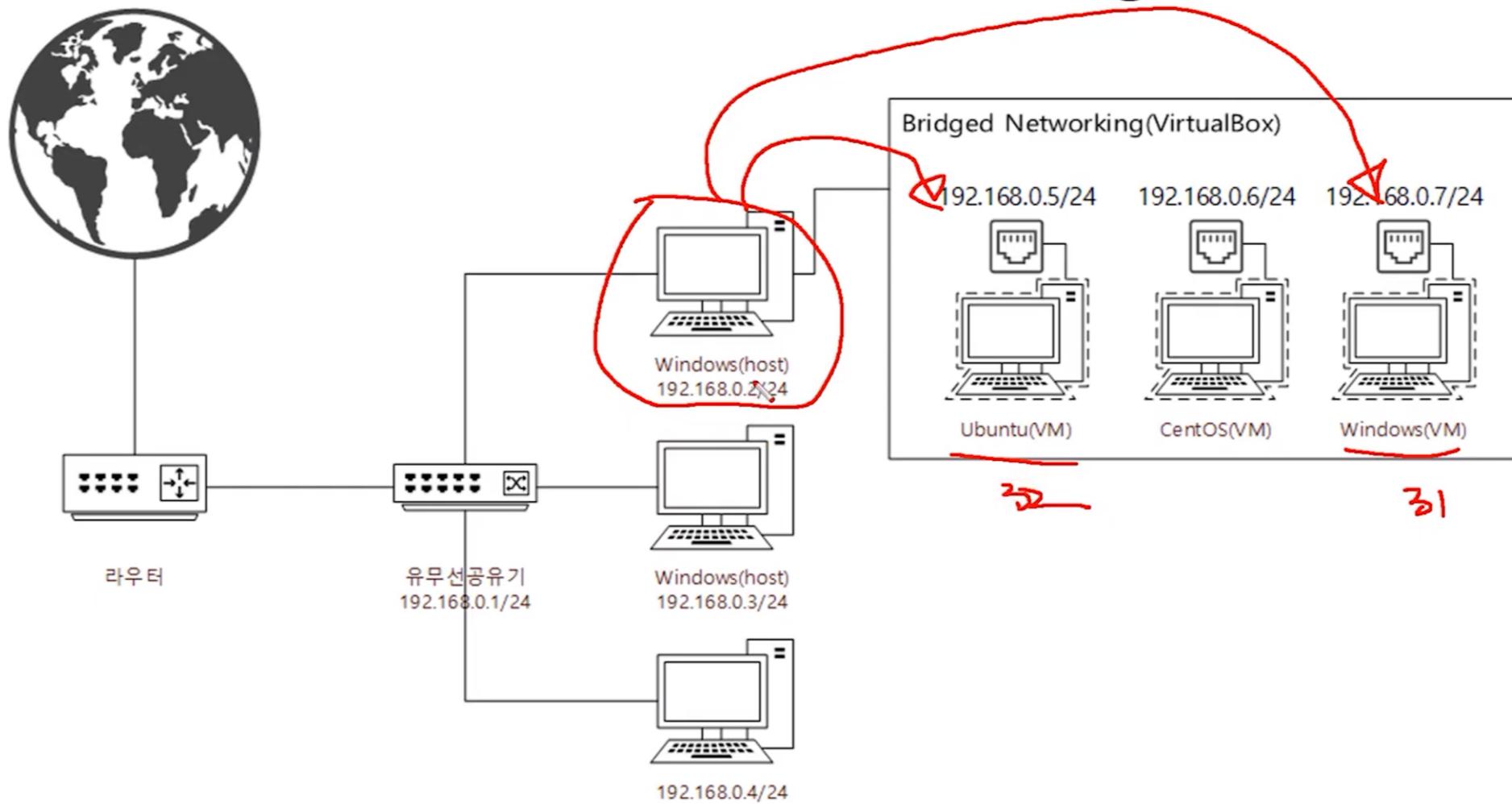
```
$ nmcli d
```

```
$ nmcli d show enp0s3
```

VirtualBox Network (NAT Network)



VirtualBox Network (Bridge)



-  일반
-  시스템
-  디스플레이
-  저장소
-  오디오
-  네트워크
-  직렬 포트
-  USB
-  공유 폴더
-  사용자 인터페이스

네트워크

어댑터 1 어댑터 2 어댑터 3 어댑터 4

네트워크 어댑터 사용하기(E)

다음에 연결됨(A): 호스트 전용 어댑터

이름(N): VirtualBox Host-Only Ethernet Adapter

▼ 고급(D)

어댑터 종류(I): Intel PRO/1000 MT Desktop(82540EM)

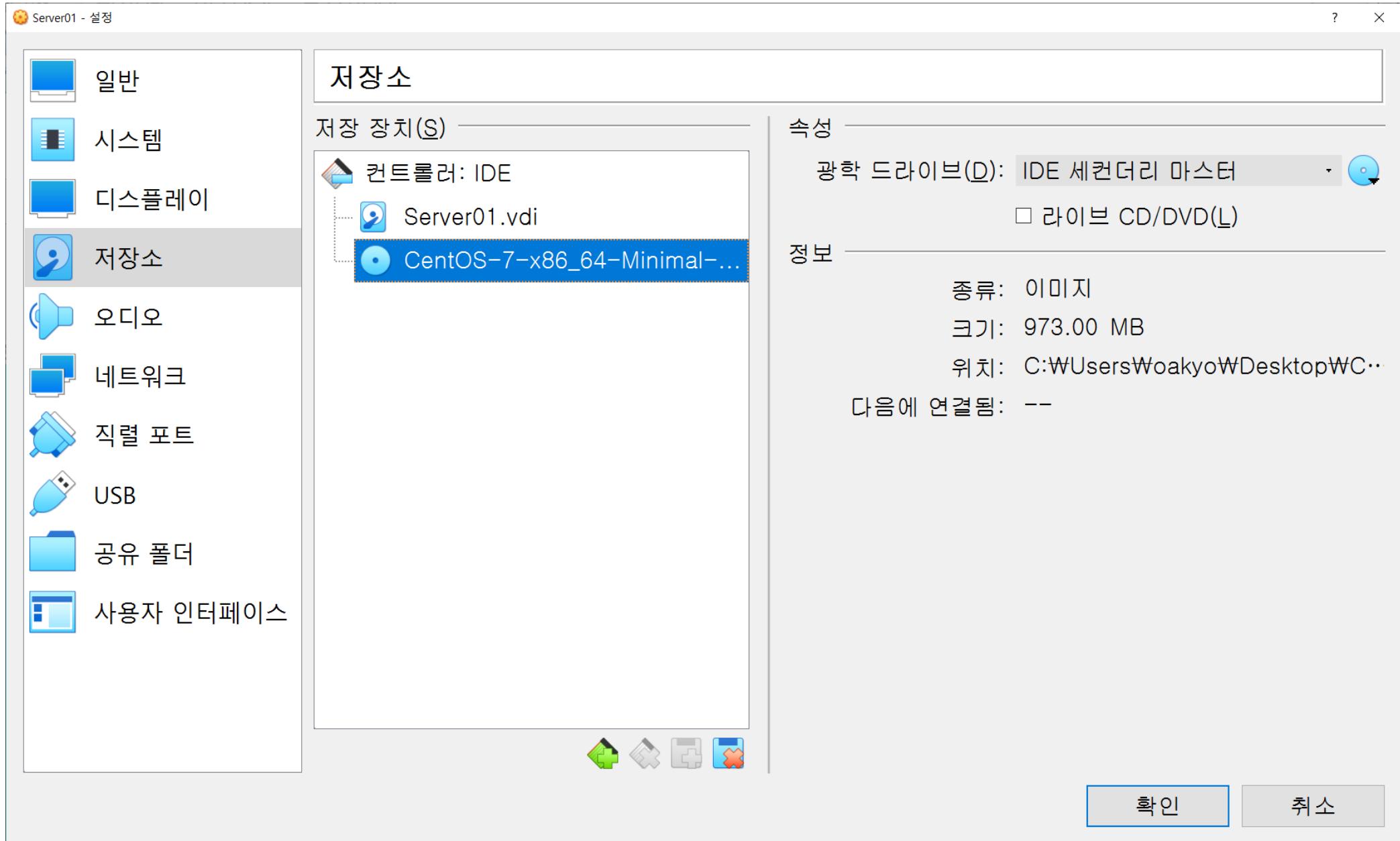
무작위 모드(P): 모두 허용

MAC 주소(M): 080027139F35

케이블 연결됨(C)

포트 포워딩(P)

확인 취소





Server01 [실행 중] - Oracle VM VirtualBox



파일 머신 보기 입력 장치 도움말

CentOS 7

Install CentOS 7
Test this media & install CentOS 7

Troubleshooting >

Press Tab for full configuration options on menu items.



Network Host name : server01.Hadoop.com

Root Password: adminuser

사용자(User) 등록

adduser username

passwd username

```
선택 명령 프롬프트
C:\Users\Hoakyo>ipconfig

Windows IP 구성

이더넷 어댑터 이더넷:
  미디어 상태 . . . . . : 미디어 연결 끊김
  연결별 DNS 접미사 . . . . :

이더넷 어댑터 VirtualBox Host-Only Network:
  연결별 DNS 접미사 . . . . :
    링크-로컬 IPv6 주소 . . . . . : fe80::e405:f545:8a6d:ec5f%48
    IPv4 주소 . . . . . : 192.168.56.1
    서브넷 마스크 . . . . . : 255.255.255.0
    기본 게이트웨이 . . . . . :

무선 LAN 어댑터 로컬 영역 연결* 1:
  미디어 상태 . . . . . : 미디어 연결 끊김
  연결별 DNS 접미사 . . . . :

무선 LAN 어댑터 로컬 영역 연결* 10:
  미디어 상태 . . . . . : 미디어 연결 끊김
  연결별 DNS 접미사 . . . . :

무선 LAN 어댑터 Wi-Fi:
```

putty SSH 접속 확인

\$ service network restart # network restart

2. network 재시작

```
[root@eloquence ~]# systemctl restart network
```

Bonding 구성 확인

1. 인터페이스 확인

```
[root@eloquence ~]# nmcli d  
DEVICE  TYPE      STATE      CONNECTION  
bond0   bond      connected   bond0  
eno1    ethernet  connected  bond-slave-eno1  
eno2    ethernet  connected  bond-slave-eno2
```

SSH 설치 방법 (Program Download)

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

클라우데라 매니저 설치

Step 1: Download and Install Virtual box and download Cloudera VM

Download Links -

VirtualBox - <https://www.virtualbox.org/wiki/Downloads>

Cloudera quickstart VM - https://downloads.cloudera.com/demo_vm/virtualbox/cloudera-quickstart-vm-5.12.0-0-virtualbox.zip

INSTALL Virtual box and Import Cloudera quickstart VM

Step 2: Find the Oozie Examples that came with System and unzip them

To unzip-

```
tar -xvf /usr/share/doc/oozie-4.1.0+cdh5.12.0+442/oozie-examples.tar.gz -C /home/cloudera/Desktop
```

Step 3: Changes to do in Job.Properties-

nameNode=hdfs://quickstart.cloudera:8020

jobTracker=quickstart.cloudera:8032

Step 4: Upload Examples to HDFS

Create a directory structure as shown in Job.Properties file

And put examples folder in hdfs

```
$ hadoop fs -put /home/cloudera/Desktop/examples /user/cloudera
```

Step 5: Run Oozie Job

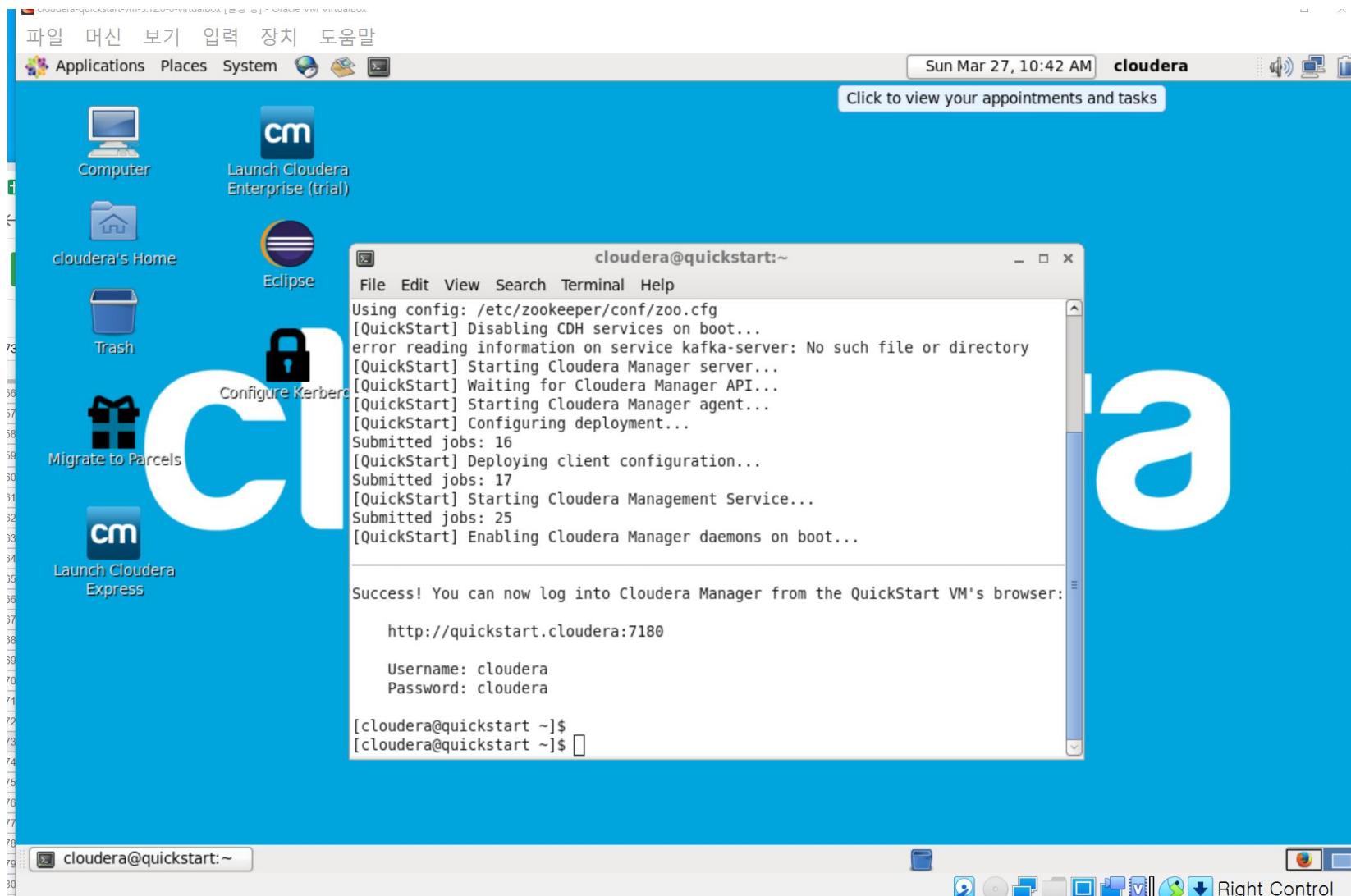
```
oozie job -oozie http://localhost:11000/oozie -config /home/cloudera/Desktop/examples/apps/java-main/job.properties -run
```

Step 6: Check the status of Oozie Job

```
oozie job -oozie http://localhost:11000/oozie -info
```

Browser 연결 -> <http://quickstart.cloudera:7180/>

클라우데라 설치 완료



Apache Oozie 소개

Oozie는 [정식 홈페이지](#)에 나와 있듯이 Hadoop ecosystem에서 사용하는 Workflow Scheduler(혹은 orchestration) 프레임워크입니다. Oozie에서 제공하는 기능은 크게 아래의 3가지와 같습니다.

- Scheduling

- 특정 시간에 액션 수행
- 주기적인 간격 이후에 액션 수행
- 이벤트가 발생하면 액션 수행

- Coordinating

- 이전 액션이 성공적으로 끝나면 다음 액션 시작

- Managing

- 액션이 성공하거나 실패했을 때 이메일 발송
- 액션 수행시간이나 액션의 단계를 저장

<https://eoriented.github.io/post/introduction-to-oozie/>

<https://www.youtube.com/watch?v=rU30SebJG6I>

```
$ hadoop fs -ls /user/cloudera
```

```
$ hadoop fs -put /home/cloudera/Desktop/examples /user/cloudera
```

```
$ hadoop fs -ls /user/cloudera/examples
```

```
$ oozie job -oozie http://localhost:11000/oozie -info <oozie job ID>
```

- Hadoop은 다음 세 가지로 구성됩니다.

- 노드 클러스터에 광범위한 데이터를 저장하기 위한 **HDFS** 파일 시스템
- 분산 계산을 위해 개발 된 **MapReduce** 프레임 워크
- 요청 된 작업에 사용 가능한 리소스를 할당하기 위한 **YARN**

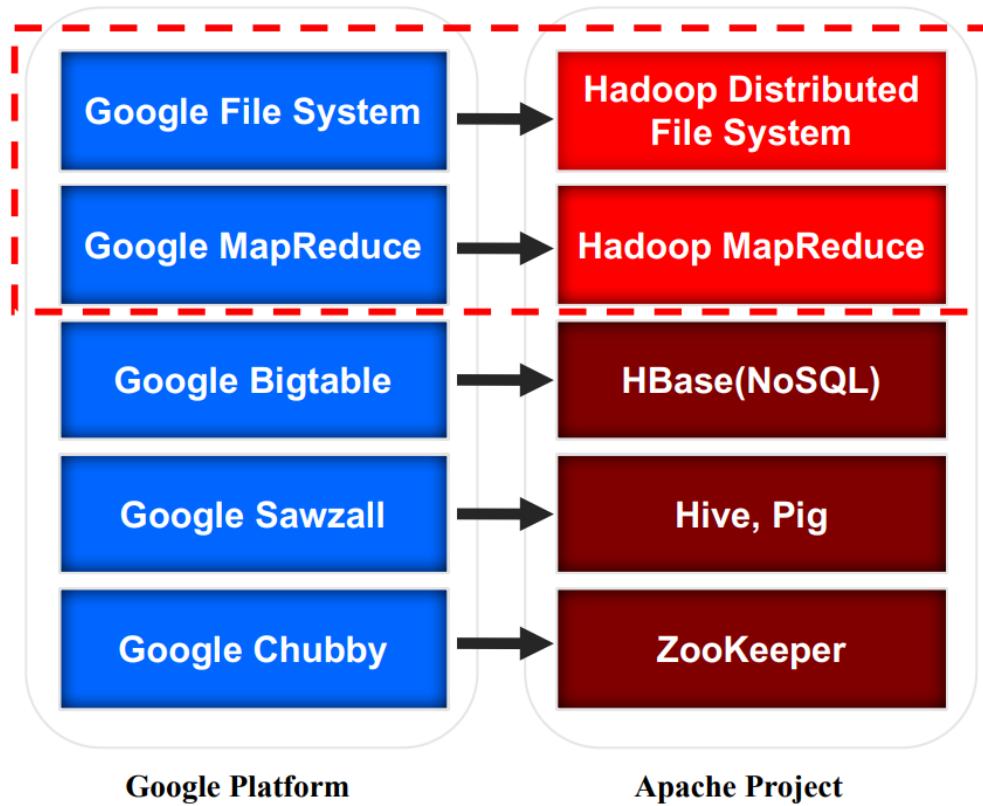
Hadoop 모듈

- Common
 - 다른 하둡 모듈을 지원하는 유틸리티
- HDFS
 - Hadoop Distributed File System
 - 어플리케이션 데이터에 고성능 접근을 지원하는 분산 파일 시스템
- MapReduce
 - 대용량 데이터의 병렬처리를 위한 얀 기반 시스템
- YARN
 - 잡 스케줄링과 클러스터 리소스 관리를 위한 프레임워크
- HBase
 - 분산 데이터베이스

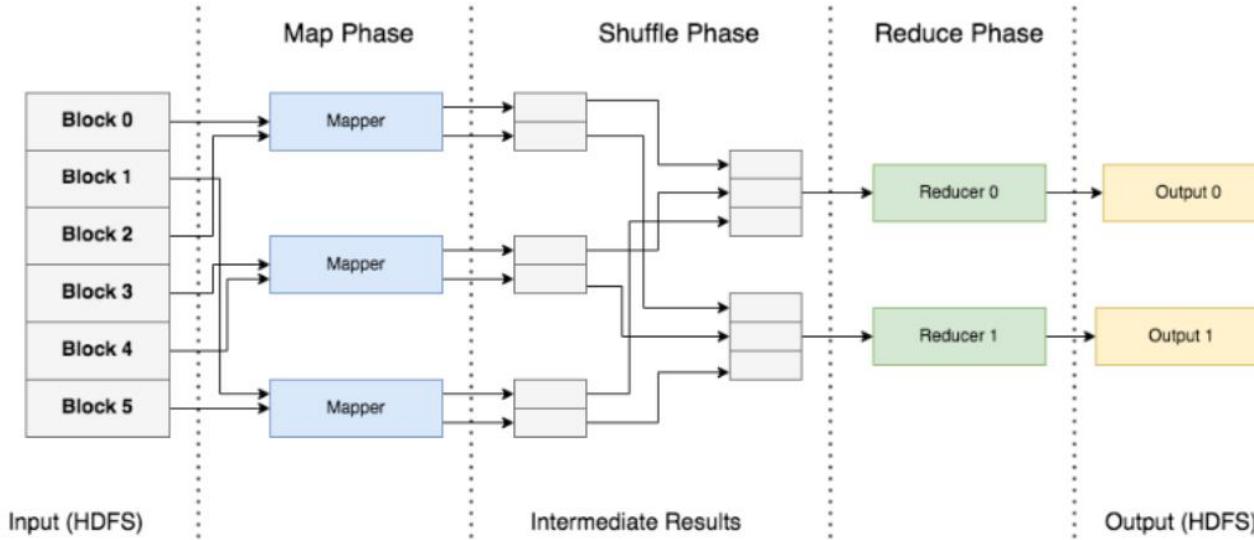
HDFS(Hadoop Distributed File System)

- 어플리케이션 기반 파일시스템
- 파일의 분산 저장이 목적
- NameNodes와 DataNodes로 구성
 - Master NameNode : 파일시스템 이미지(fsimage)와 변경기록(edits)을 저장
 - fsimage는 in-memory로 관리됨
 - Secondary NameNode : Master NameNode의 fsimage 파일과 edits 파일의 사본을 저장
 - DataNode : 데이터파일의 블록을 저장, 디폴트 블록의 크기는 128MB
- 저렴한 컴퓨터로 대용량 데이터를 저장할 수 있는 시스템
 - 네트워크 Raid와 같이 연결된 것 처럼 사용하는 하드디스크
 - Scale Out
- Block(Chunk) 단위로 파일관리 (저장/복제/삭제)
 - Default Size는 128M(134217728)
- 복제기능을 통해 안전성/신뢰성을 보장
- 1대의 Master서버에 4000+이상의 DataNodes를 운영할 수 있음.
- API지원
 - 하둡 코어는 Python, Java, C/C++

하둡



MapReduce



• 위 그림은, MapReduce 흐름이며

• 첫 번째로, HDFS 의 각 노드들에 있는 Data Block 이 Mapper 에 할당됩니다.

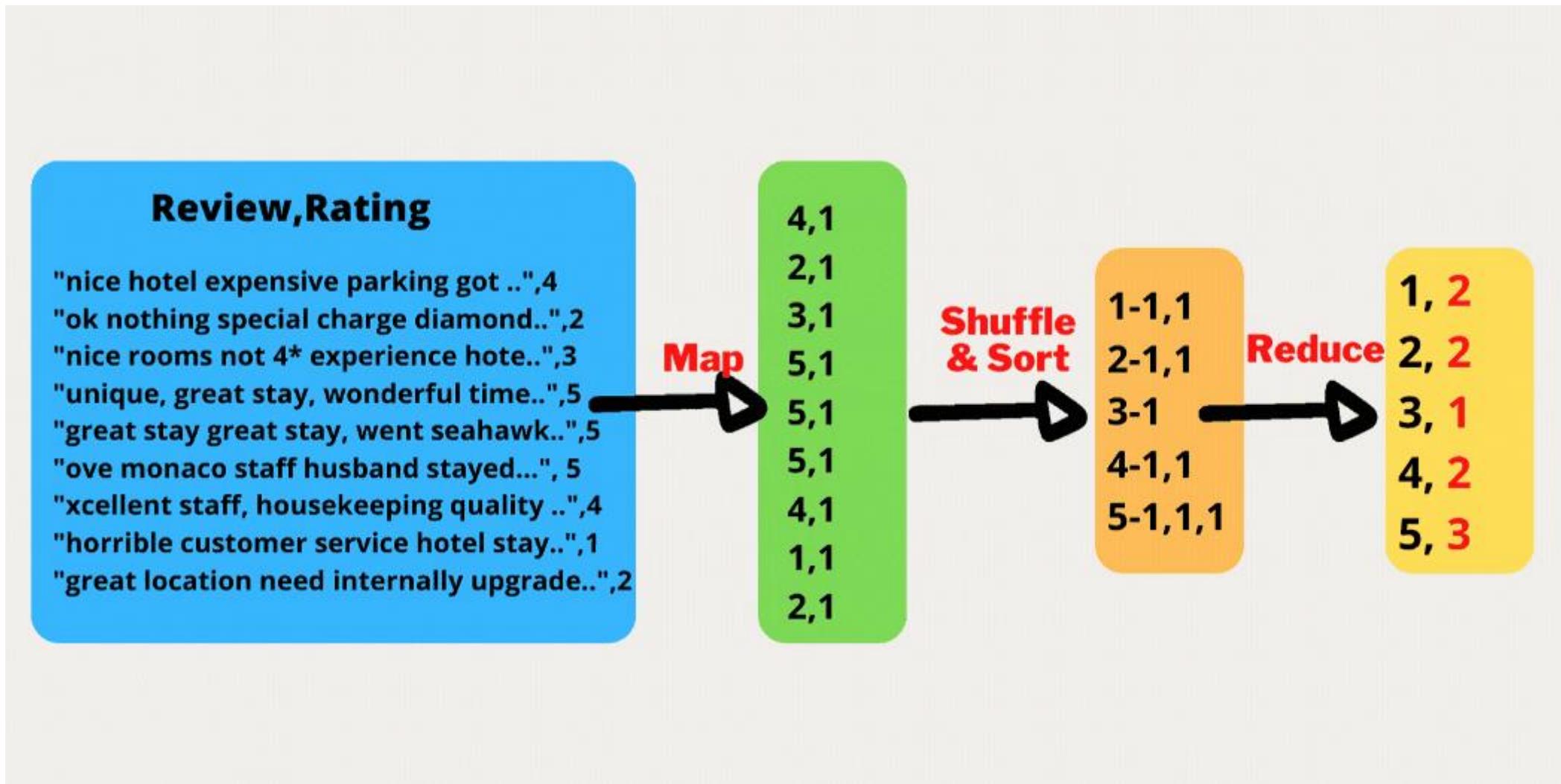
• 두 번째로, 계산 된 중간 결과가 Shuffle 됩니다.

- 계산은 관련 Data Block 이 있는 노드에서 이루어지며, 이는 데이터 지역성이라고 불립니다.
- 각 Data Block 이 저장된 노드의 로컬 리소스를 활용하여 계산을 실행하면, 계산을 위해 노드 간에 데이터를 이동할 필요가 없게 됩니다.

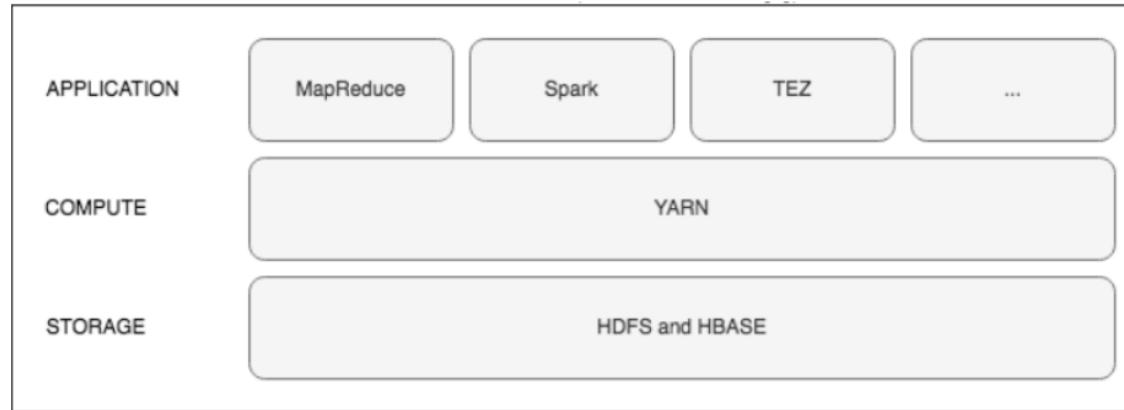
• 세 번째로, Shuffle 된 데이터를 Reducer에게 전달해 마지막 계산을 진행합니다.

• 마지막으로, 계산 된 결과가 HDFS에 저장됩니다.

MapReduce



YARN



- YARN은 "Yet Another Resource Negotiator"의 약자이며,
- Storage(HDFS)와 Application(MapReduce) 간의 자원을 조절해주는 역할을 합니다.

HDFS(파일시스템) 명령

- hdfs dfs -명령어 -옵션 명령행인자
 - ex) hdfs dfs -mkdir -p /user/hadoop
 - hadoop fs -명령어 명령행인자 : 1.x 명령
- 명령어
 - ls [-d][-h][-R] : 파일 또는 디렉토리 목록
 - du [-s][-h] : 파일 용량 확인
 - cat, text : 파일 내용 보기
 - mkdir [-p] : 디렉토리 생성
 - put, get : 파일 복사(로컬 <-> HDFS)
 - getmerge [-nl] : 병합해서 로컬에 저장(nl은 각 파일 끝에 개행문자 포함)
 - cp, mv : 파일 복사, 이동(HDFS <-> HDFS)
 - rm [-R][-skipTrash] : 파일 삭제, 디렉토리 삭제, 완전 삭제
 - count [-q] : 카운트 값 조회
 - tail : 파일의 마지막 내용 확인
 - chmod, chown, chgrp : 권한, 소유주, 그룹 변경
 - touchz : 0바이트 파일 생성
 - stat [-R] <format> : 통계 정보 조회
 - 포맷 : %b(바이트수) %F(파일인지디렉토리인지) %u(소유주) %g(그룹) %n(이름) %o(블록크기) %r(복제수) %y(날짜 및 시간) %Y(유닉스타임스탬프)
 - setrep : 복제 수 변경
 - expunge : 휴지통 비우기
 - test -[edz]: 파일 형식 확인(empty, zero, dir)

1.X에는 명령어에 옵션이 포함되는 형식이었음.
아래 두 명령은 동일

hdfs dfs -ls -R / ←2.x, 3.x 명령
hadoop fs -lsr / ←1.x 명령

HDFS(파일시스템) 명령

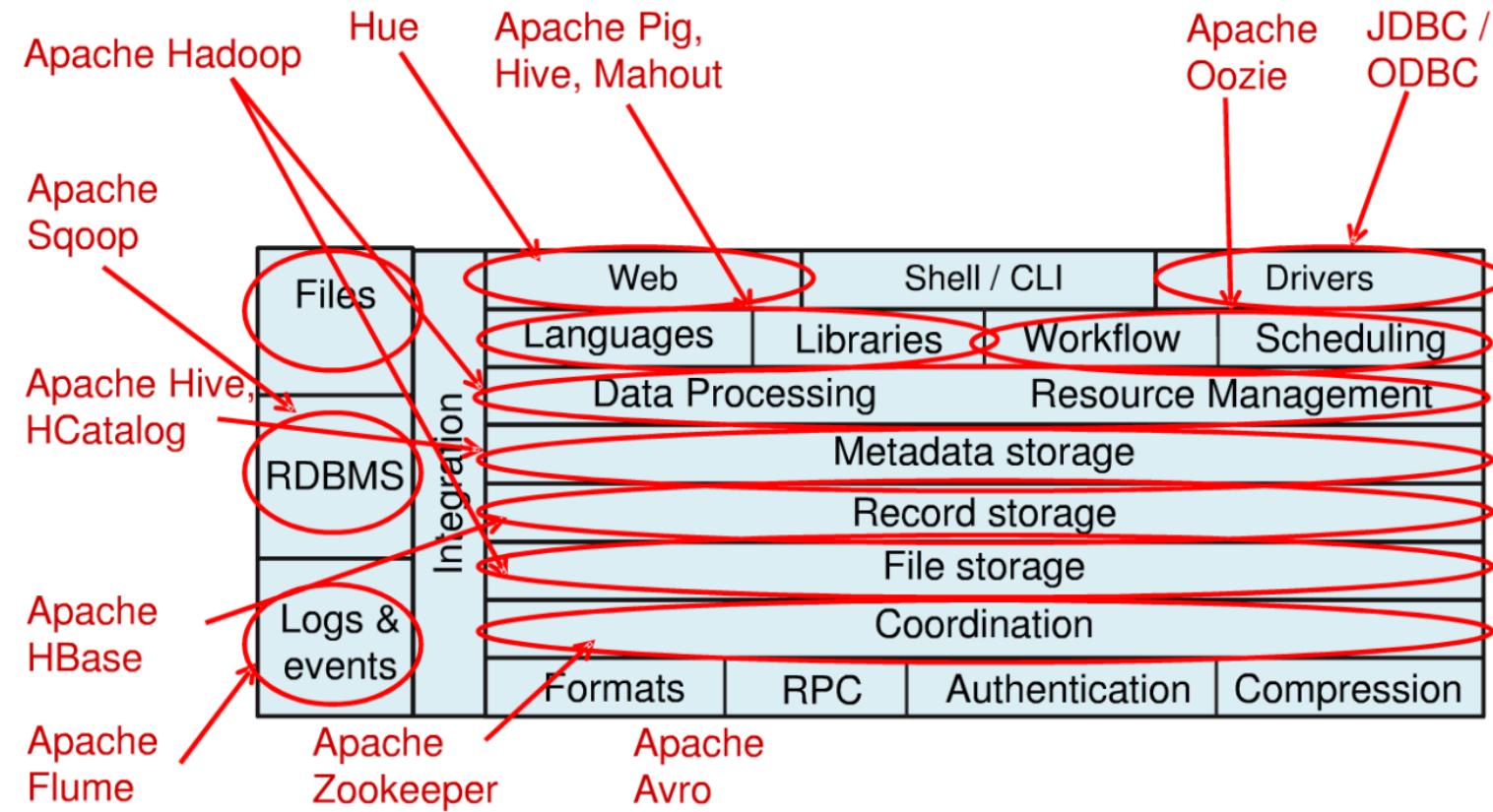
- 사용자의 홈디렉토리를 생성하세요.
hdfs dfs -mkdir -p /user/nova
- 사용자 홈디렉토리에 airline 디렉토리를 생성하세요.
hdfs dfs -mkdir airline
- airline 디렉토리에 2008.csv 파일을 업로드하세요.
hdfs dfs -put 2008.csv airline/
- airline 디렉토리에 2007.csv 파일을 업로드하세요.
hdfs dfs -put 2007.csv airline/
- 로컬의 2008.csv 파일을 삭제하세요.
rm 2008.csv
- HDFS의 2008.csv 파일을 로컬에 저장하세요.
hdfs dfs -get airline/2008.csv
- airline 디렉토리를 삭제하세요.
hdfs dfs -rm -R airline/
- 루트에 airline 디렉토리를 생성하세요.
hdfs dfs -mkdir /airline
- /airline 디렉토리에 2008.csv 파일을 업로드하세요.
hdfs dfs -put 2008.csv /airline/
- 2008.csv 파일의 처음 5라인을 출력하세요.
hdfs dfs -cat /airline/2008.csv | head -5
- 2008.csv 파일의 마지막 1KB를 출력하세요.
hdfs dfs -tail /airline/2008.csv
- 2008.csv 파일의 통계 정보를 조회하세요.
hdfs dfs -stat "%b %F %n %o %r %y" /airline/2008.csv
- 2008.csv 파일의 복제 데이터 개수를 변경하세요.
hdfs dfs -setrep 1 /airline/2008.csv
- 2008.csv 파일의 복제 수를 확인하세요.
hdfs dfs -stat %r /airline/2008.csv
- 2008.csv 파일의 복제 수를 1로 변경하세요.
hdfs dfs -setrep 2 /airline/2008.csv

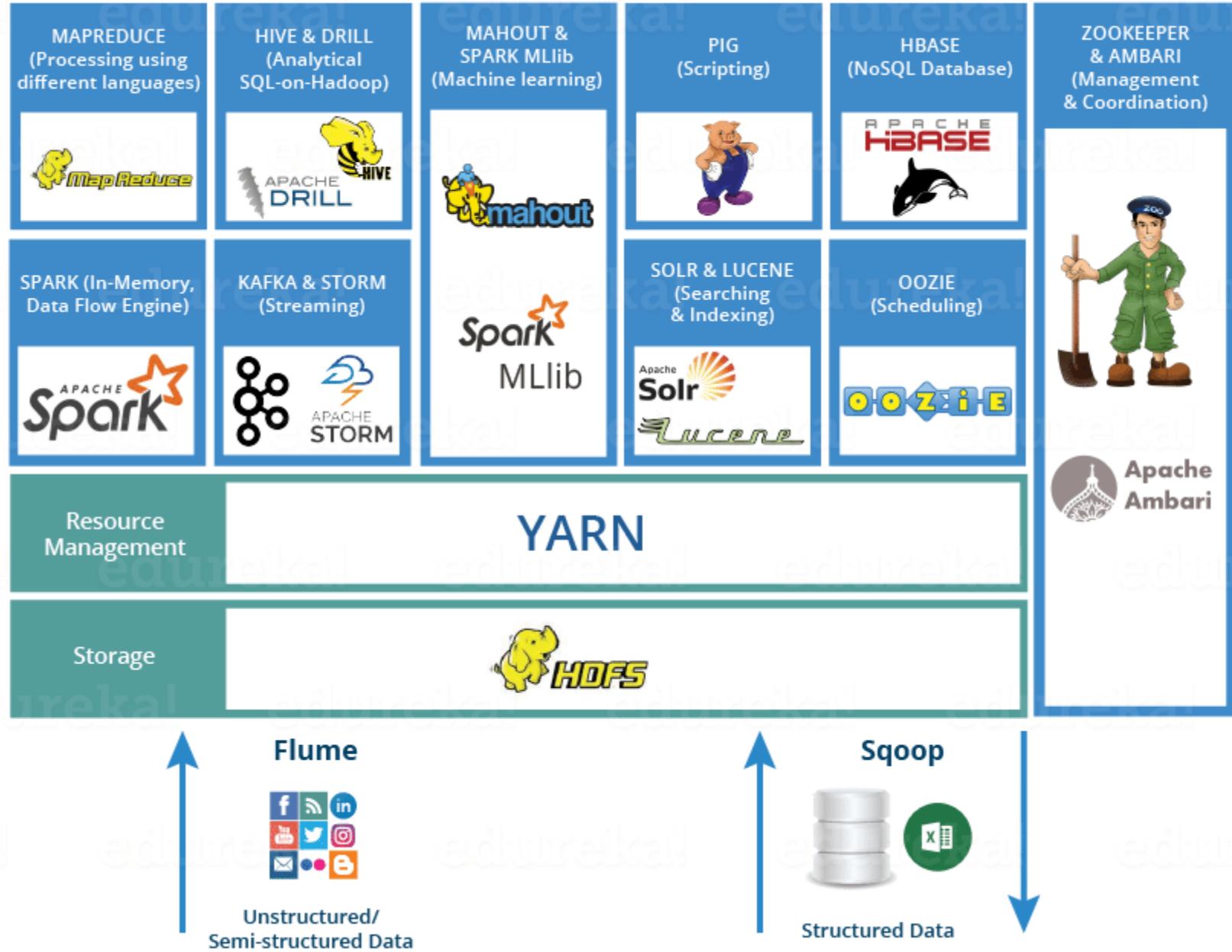
HDFS(파일시스템) 명령

- hdfs dfs -명령어 -옵션 명령행인자
 - ex) hdfs dfs -mkdir -p /user/hadoop
 - hadoop fs -명령어 명령행인자 : 1.x 명령
- 명령어
 - ls [-d][-h][-R] : 파일 또는 디렉토리 목록
 - du [-s][-h] : 파일 용량 확인
 - cat, text : 파일 내용 보기
 - mkdir [-p] : 디렉토리 생성
 - put, get : 파일 복사(로컬 <-> HDFS)
 - getmerge [-nl] : 병합해서 로컬에 저장(nl은 각 파일 끝에 개행문자 포함)
 - cp, mv : 파일 복사, 이동(HDFS <-> HDFS)
 - rm [-R][-skipTrash] : 파일 삭제, 디렉토리 삭제, 완전 삭제
 - count [-q] : 카운트 값 조회
 - tail : 파일의 마지막 내용 확인
 - chmod, chown, chgrp : 권한, 소유주, 그룹 변경
 - touchz : 0바이트 파일 생성
 - stat [-R] <format> : 통계 정보 조회
 - 포맷 : %b(바이트수) %F(파일인지디렉토리인지) %u(소유주) %g(그룹) %n(이름) %o(블록크기) %r(복제수) %y(날짜 및 시간) %Y(유닉스타임스탬프)
 - setrep : 복제 수 변경
 - expunge : 휴지통 비우기
 - test -[edz]: 파일 형식 확인(empty, zero, dir)

1.X에는 명령어에 옵션이 포함되는 형식이었음.
아래 두 명령은 동일

hdfs dfs -ls -R / ←2.x, 3.x 명령
hadoop fs -lsr / ←1.x 명령





DT 와 AI

오프라인

온라인



기록되지 않던 사람들의 생활이 데이터로 남기 시작

IoT의 핵심 중 하나 → 많은 센서(sensor)



센서의 특징

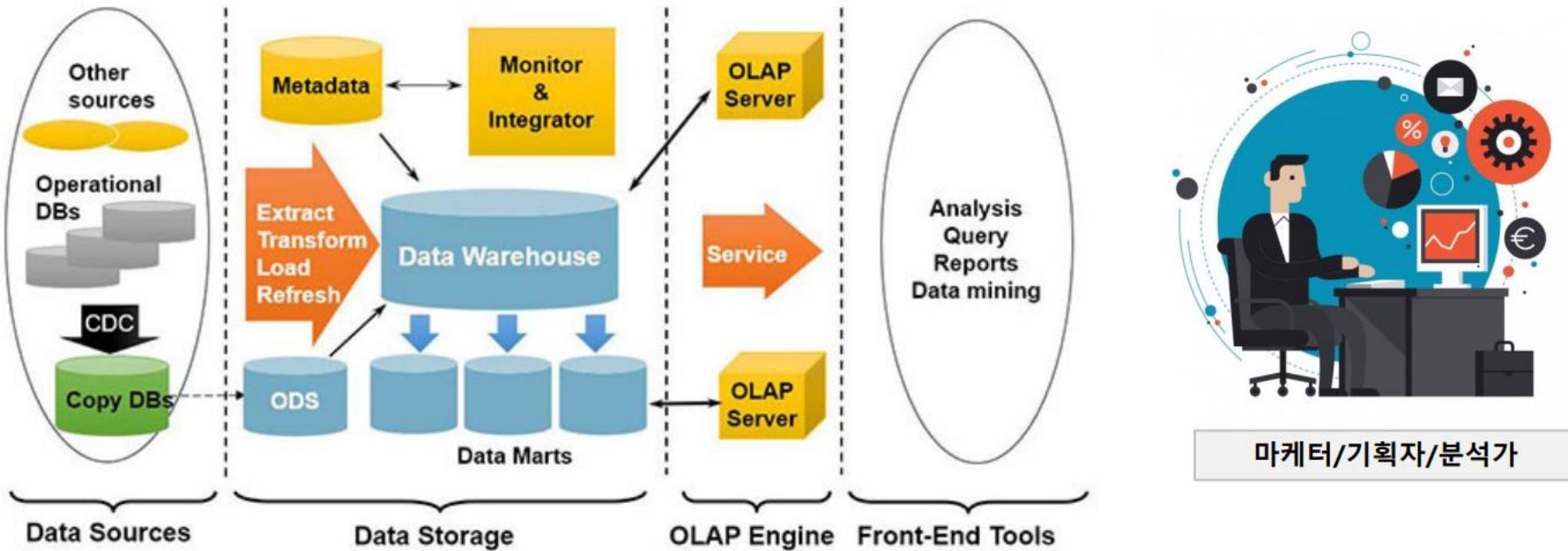
숨만 쉬어도 데이터를 뿜어낸다.

IoT → 모든 기기가 데이터를 뿜어내고 주고 받는다.

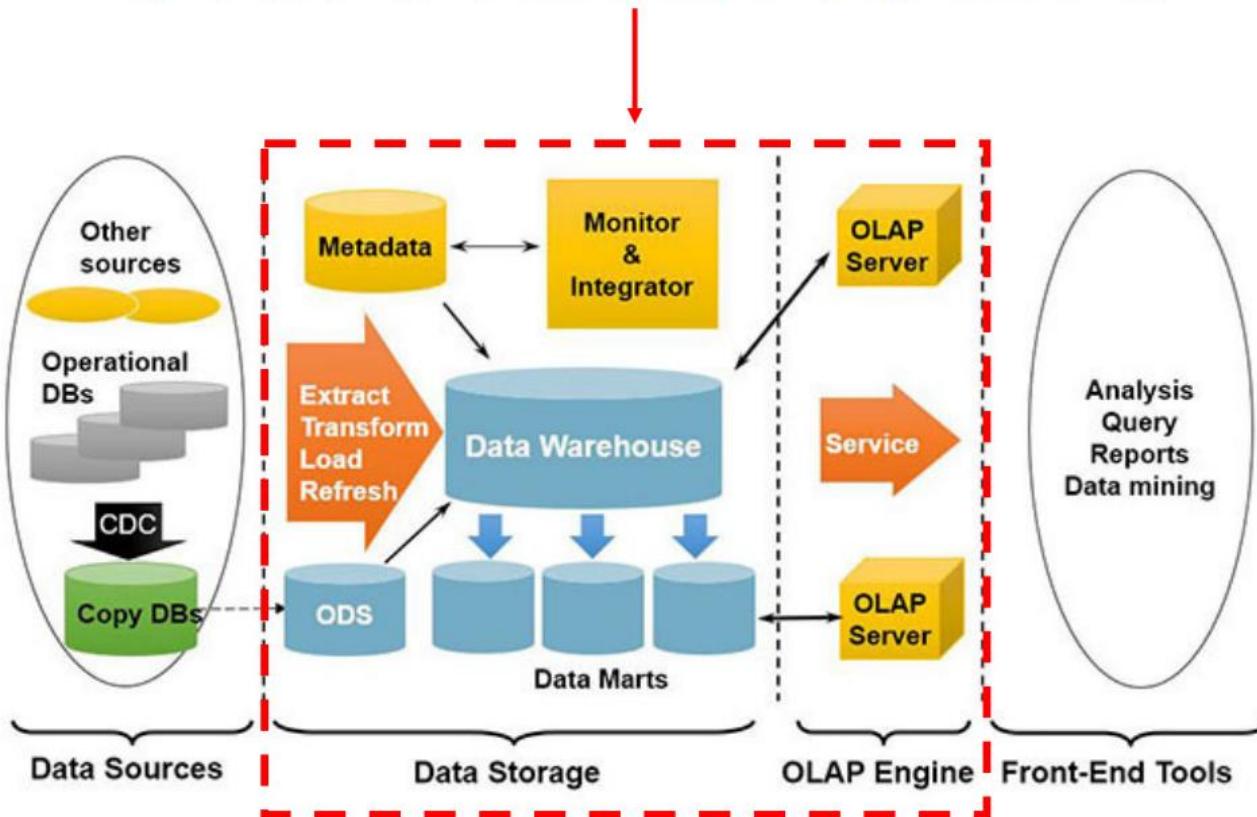
분석해야 할 데이터가 점점 많아진다...

좋은 것인가? ^^;

데이터 분석 환경 구축



주로 하둡 에코시스템으로 이 영역을 구축



마케터/기획자/분석가

NoSQL 데이터베이스 선정 기준 -> HBase

이번 포스트에서는 애플리케이션을 개발할 때 관계형 데이터베이스가 아닌 NoSQL 데이터베이스를 선택할 때 참고할 수 있는 기준들을 살펴보면 도움이 될 것입니다. RDBMS와 NoSQL 데이터베이스의 대표적인 차이점은 스키마와 트랜잭션 속성이나 실제로 데이터를 저장하는 구조에서 살펴볼 수 있습니다. 애플리케이션을 개발할 때 애플리케이션이 가진 특징을 명확히 이해하고 있으면 다양한 데이터베이스 중에 알맞는 데이터베이스를 선택할 수 있을 것입니다.

데이터 모델

데이터를 저장하는 방식에 따라 구분할 수 있습니다. 데이터를 저장하는 방식은 다음과 같은 것들이 있습니다.

- 키/값 방식(일종의 거대한 해시맵 구조)
 - 반구조적(semistructured 방식)
 - 컬럼 지향 방식
 - 문서 지향 방식
- 애플리케이션에서 데이터를 어떻게 접근할 것인지 고민하고, 스키마가 시간이 지남에 따라 변할 수 있는지 생각해보고 데이터 모델을 결정할 수 있을 것입니다.

<https://eoriented.github.io/post/nosql-dimensions/>



정 준 수 / Ph.D (jsjeong@hansung.ac.kr)

- 前) 삼성전자 연구원
- 前) 삼성의료원 (삼성생명과학연구소)
- 前) 삼성SDS (정보기술연구소)
- 現) (사)한국인공지능협회, AI, 머신러닝 강의
- 現) 한국소프트웨어산업협회, AI, 머신러닝 강의
- 現) 서울디지털재단, AI 자문위원
- 現) 한성대학교 교수(겸)
- 전문분야: Computer Vision, 머신러닝(ML), RPA
- <https://github.com/JSeong-me/>