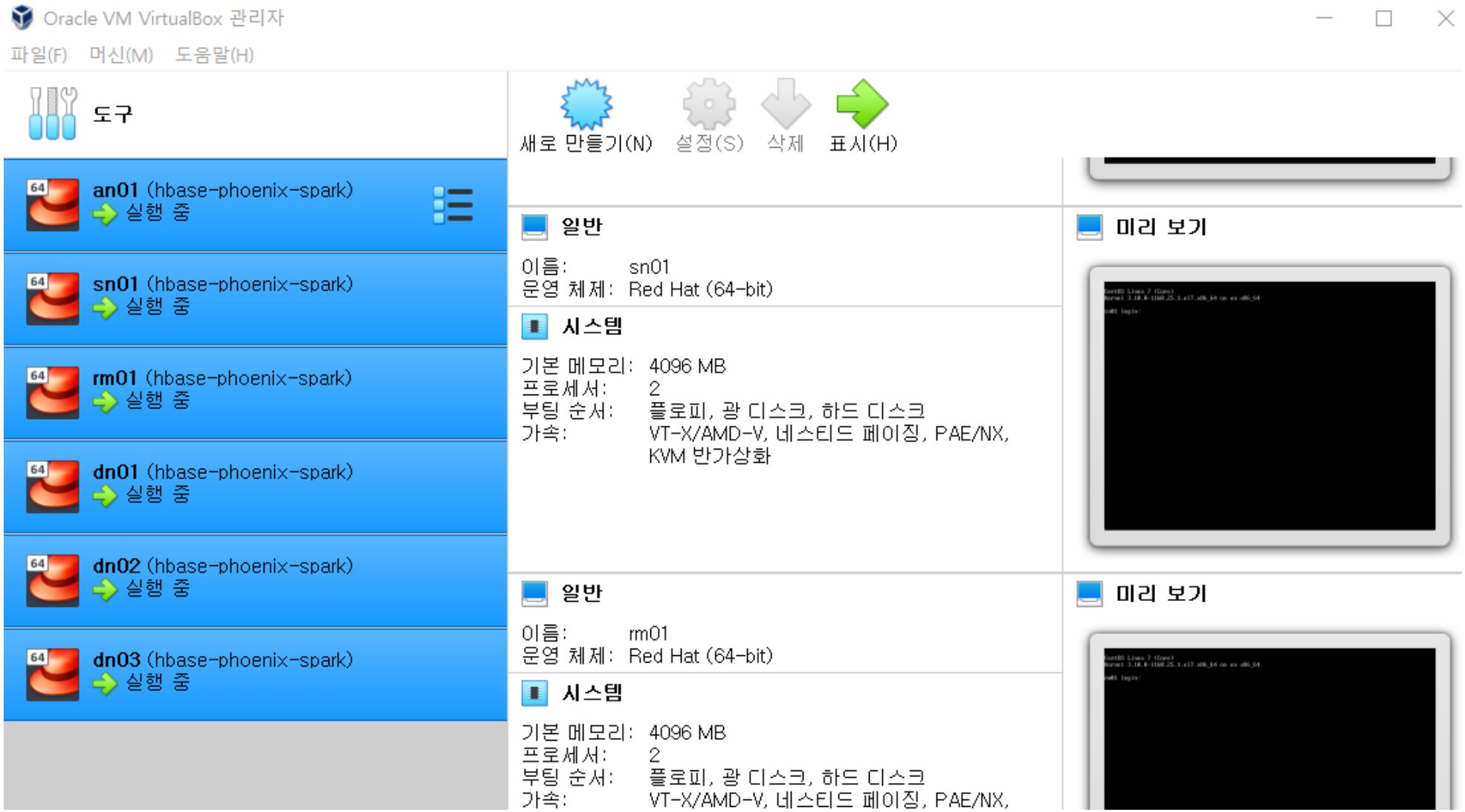
Hadoop cluster install
(server 6)

- 가상 서버 6대의 설정이 모두 마친 상태임을 가정



1. Zookeeper install

- 주키퍼 계정 생성
- 주키퍼 클러스터는 an01, sn01, rm01 3대의 서버에만 설치함

```
action server: an01, sn01, rm01
pwd: /root

adduser zookeeper
passwd zookeeper // 비밀번호 설정, 편의상 zookeeper
su - zookeeper // 주키퍼 계정 접속
```

- 주키퍼 계정간 ssh 통신

action server: an01, sn01, rm01
pwd: /home/zookeeper

```
ssh-keygen
ssh-copy-id -i .ssh/id_rsa.pub zookeeper@an01
ssh-copy-id -i .ssh/id_rsa.pub zookeeper@sn01
ssh-copy-id -i .ssh/id_rsa.pub zookeeper@rm01
ssh [서버명]을 통해 비밀번호 여부 없이 접속 확인
```

1. Zookeeper install

- 주키퍼 설치
- an01서버에서 설치, 설정 후 재압축 및 타 sn01, rm01 서버에 배포

```
action server: an01
pwd: /home/zookeeper

wget https://archive.apache.org/dist/zookeeper/zookeeper-3.4.10/zookeeper-3.4.10.tar.gz
tar xvfz zookeeper-3.4.10.tar.gz
cd zookeeper-3.4.10
```

- 주키퍼 파일 설정

```
action server: an01
pwd: /home/zookeeper/zookeeper-3.4.10

cp cp conf/zoo_sample.cfg conf/zoo.cfg
vi conf/zoo.cfg
```

```
tickTime=2000
initLimit=10
syncLimit=5
dataDir=/home/zookeeper/data
clientPort=2181
maxClientCnxns=0
maxSessionTimeout=180000
server.1=an01:2888:3888
server.2=sn01:2888:3888
server.3=rm01:2888:3888
```

```
[zookeeper@an01 ~]$ cd zookeeper-3.4.10
[zookeeper@an01 zookeeper-3.4.10]$ cat ./conf/zoo.cfg
tickTime=2000
initLimit=10
syncLimit=5
dataDir=/home/zookeeper/data
clientPort=2181
maxClientCnxns=0
maxSessionTimeout=180000
server.1=an01:2888:3888
server.2=sn01:2888:3888
server.3=rm01:2888:3888

[zookeeper@an01 zookeeper-3.4.10]$
```

```
cd ..
```

1. Zookeeper install

- 재압축 후 배포

```
action server: an01
pwd: /home/zookeeper

tar cvfz zookeeper.tar.gz zookeeper-3.4.10
scp zookeeper.tar.gz zookeeper@sn01:/home/zookeeper
scp zookeeper.tar.gz zookeeper@rm01:/home/zookeeper
```

- 압축 해제

```
action server: sn01, rm01
pwd: /home/zookeeper

tar xvfz zookeeper.tar.gz
```

1. Zookeeper install

- 각 서버별 myid 설정

```
action server: an01,sn01, rm01
pwd: /home/zookeeper

mkdir data
cd data
vi myid // an01에는 1, sn01에는 2, rm01에는 3 입력 후 저장
cd /home/zookeeper/zookeeper-3.4.10
```

- 주키퍼 실행

```
action server: an01, sn01, rm01
pwd: /home/zookeeper/zookeeper-3.4.10

./bin/zkServer.sh start // 서버 시작
./bin/zkServer.sh status // 상태 확인
```

```
[zookeeper@an01 zookeeper-3.4.10]$ ./bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/zookeeper/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: follower
[zookeeper@an01 zookeeper-3.4.10]$
```

```
[zookeeper@sn01 zookeeper-3.4.10]$ ./bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/zookeeper/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: follower
[zookeeper@sn01 zookeeper-3.4.10]$
```

```
[zookeeper@rm01 zookeeper-3.4.10]$ ./bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/zookeeper/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: leader
[zookeeper@rm01 zookeeper-3.4.10]$
```

한 서버는 leader, 두 서버는 follower

exit // root 계정으로 변환

- 주키퍼 종료

action server: an01, sn01, rm01
pwd: /home/zookeeper/zookeeper-3.4.10

./bin/zkServer.sh stop // 곧바로 하둡 설치 후 실행할 것 이기에 종료하지 않음.

2. Hadoop install

- 하둡 관련 계정 생성, 폴더 생성 및 권한 부여

```
action server: all
pwd: /root

adduser hadoop
passwd hadoop // 편의상 Hadoop

mkdir /dfs
mkdir /pids
mkdir /yarn

chown hadoop /dfs
chown hadoop /pids
chown hadoop /yarn
chgrp hadoop /dfs
chgrp hadoop /pids
chgrp hadoop /yarn

su - hadoop // 하둡 계정 접속
```

2. Hadoop install

- 하둡 계정간 ssh 통신

action server: all
pwd: /home/hadoop

```
ssh-keygen
ssh-copy-id -i .ssh/id_rsa.pub hadoop@an01
ssh-copy-id -i .ssh/id_rsa.pub hadoop@sn01
ssh-copy-id -i .ssh/id_rsa.pub hadoop@rm01
ssh-copy-id -i .ssh/id_rsa.pub hadoop@dn01
ssh-copy-id -i .ssh/id_rsa.pub hadoop@dn02
ssh-copy-id -i .ssh/id_rsa.pub hadoop@dn03

ssh [서버명] // 비밀번호 여부 없이 접속 확인
```

2. Hadoop install

- 하둡 설치
- 마찬가지로 an01서버에서 다운로드 및 설정 후 타 서버들에 배포

```
action server: an01
pwd: /home/hadoop

wget https://archive.apache.org/dist/hadoop/core/hadoop-3.1.2/hadoop-3.1.2.tar.gz
tar xvfz hadoop-3.1.2.tar.gz
cd ./hadoop-3.1.2/etc/hadoop
```

- 데이터 노드로 사용할 서버 지정

```
action server: an01
pwd: /home/hadoop/hadoop-3.1.2/etc/hadoop

vi workers
dn01
dn02
dn03
```

- hdfs와 맵리듀스에서 공통적으로 사용할 환경 설정

action server: an01
pwd: /home/hadoop/hadoop-3.1.2/etc/hadoop

vi core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed 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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->
```

```
<configuration>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://NNHA</value>
        </property>
        <property>
                <name>ha.zookeeper.quorum</name>
                <value>an01:2181,sn01:2181,rm01:2181</value>
        </property>
</configuration>
```

- hdfs에서 사용할 환경 설정

```
action server: an01
pwd: /home/hadoop/hadoop-3.1.2/etc/hadoop

vi hdfs-site.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed 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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>dfs.client.failover.proxy.provider.NNHA</name>
                <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
        </property>
        <property>
                <name>dfs.namenode.name.dir</name>
                <value>/dfs/namenode</value>
        </property>
        <property>
                <name>dfs.datanode.data.dir</name>
                <value>/dfs/datanode</value>
        </property>
```

```
<property>
        <name>dfs.journalnode.edits.dir</name>
        <value>/dfs/journalnode</value>
</property>
<property>
        <name>dfs.nameservices</name>
        <value>NNHA</value>
</property>
<property>
        <name>dfs.ha.namenodes.NNHA</name>
        <value>an01,sn01</value>
</property>
<property>
        <name>dfs.namenode.rpc-address.NNHA.an01</name>
        <value>an01:8020</value>
</property>
<property>
        <name>dfs.namenode.rpc-address.NNHA.sn01</name>
        <value>sn01:8020</value>
</property>
<property>
        <name>dfs.namenode.http-address.NNHA.an01</name>
        <value>an01:9870</value>
</property>
<property>
        <name>dfs.namenode.http-address.NNHA.sn01</name>
        <value>sn01:9870</value>
</property>
```

```
        <property>
                <name>dfs.namenode.shared.edits.dir</name>
                <value>qjournal://an01:8485;sn01:8485;rm01:8485/NNHA</value>
        </property>
        <property>
                <name>dfs.ha.automatic-failover.enabled</name>
                <value>true</value>
        </property>
        <property>
                <name>dfs.ha.fencing.methods</name>
                <value>sshfence</value>
        </property>
        <property>
                <name>dfs.ha.fencing.ssh.private-key-files</name>
                <value>/home/hadoop/.ssh/id_rsa</value>
        </property>
</configuration>
```

- 맵리듀스에서 사용할 환경 설정

```
action server: an01
pwd: /home/hadoop/hadoop-3.1.2/etc/hadoop

vi mapred-site.xml
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed 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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

- yarn에서 사용할 환경 설정

action server: an01
pwd: /home/hadoop/hadoop-3.1.2/etc/hadoop

vi yarn-site.xml

```
<?xml version="1.0"?>
<!--
  Licensed 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. See accompanying LICENSE file.
-->

<configuration>
        <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.nodemanager.local-dirs</name>
        <value>/yarn/nm-local-dir</value>
</property>
<property>
        <name>yarn.resourcemanager.fs.state-store.uri</name>
        <value>/yarn/system/rmstore</value>
</property>
<property>
        <name>yarn.resourcemanager.hostname</name>
        <value>rm01</value>
</property>
<property>
        <name>yarn.resourcemanager.address</name>
        <value>rm01:8032</value>
</property>
<property>
        <name>yarn.web-proxy.address</name>
        <value>0.0.0.0:8089</value>
</property>
```

```
        <property>
                <name>yarn.application.classpath</name>
                <value>
                        /home/hadoop/hadoop-3.1.2/etc/hadoop,
                        /home/hadoop/hadoop-3.1.2/share/hadoop/common/*,
                        /home/hadoop/hadoop-3.1.2/share/hadoop/common/lib/*,
                        /home/hadoop/hadoop-3.1.2/share/hadoop/hdfs/*,
                        /home/hadoop/hadoop-3.1.2/share/hadoop/hdfs/lib/*,
                        /home/hadoop/hadoop-3.1.2/share/hadoop/mapreduce/*,
                        /home/hadoop/hadoop-3.1.2/share/hadoop/mapreduce/lib/*,
                        /home/hadoop/hadoop-3.1.2/share/hadoop/yarn/*,
                        /home/hadoop/hadoop-3.1.2/share/hadoop/yarn/lib/*
                </value>
        </property>
        <property>
                <name>yarn.nodemanager.pmem-check-enabled</name>
                <value>false</value>
        </property>
        <property>
                <name>yarn.nodemanager.vmem-check-enabled</name>
                <value>false</value>
        </property>
</configuration>
```

- hadoop-env.sh 설정

```
action server: an01
pwd: /home/hadoop/hadoop-3.1.2/etc/hadoop

vi hadoop-env.sh

[54번째 줄]
export JAVA_HOME=/opt/apps/jdk8/

[211번째 줄]
export HADOOP_PID_DIR=/pids

cd /home/hadoop // 경로 이동
```

- 재압축 후 배포

```
action server: an01
pwd: /home/hadoop

tar cvfz hadoop.tar.gz hadoop-3.1.2
scp hadoop.tar.gz hadoop@sn01:/home/hadoop
scp hadoop.tar.gz hadoop@rm01:/home/hadoop
scp hadoop.tar.gz hadoop@dn01:/home/hadoop
scp hadoop.tar.gz hadoop@dn02:/home/hadoop
scp hadoop.tar.gz hadoop@dn03:/home/hadoop
```

- 압축 해제

```
action server: sn01, rm01, dn01, dn02, dn03
pwd: /home/hadoop

tar xvfz hadoop.tar.gz

exit // 모든 서버 root계정 변환
```

- 하둡 환경변수 설정

```
action server: all
pwd: /root

vi /etc/profile.d/hadoop.sh

export HADOOP_HOME=/home/hadoop/hadoop-3.1.2
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

source /etc/profile.d/hadoop.sh

su - hadoop // 하둡 계정 재접속
cd hadoop-3.1.2 // 경로 변경
```

2. Hadoop install

- 하둡 클러스터 실행
- 각 서버마다 실행할 명령어가 다름. 명령어 옆 주석 확인!
- 최초 실행 명령어는 설치 직후 한 번만 실행하고 다음 실행 시에는 생략
- zookeeper 서버가 실행되어있어야 함.

action server: an01, sn01, rm01
pwd: /home/hadoop/hadoop-3.1.2

./bin/hdfs zkfc -formatZK // an01, 최초 실행 1번만

./bin/hdfs --daemon start journalnode // an01, sn01, rm01, 저널 노드 실행

./bin/hdfs namenode -format NNHA // an01, 최초 실행 1번만, 네임노드 초기화

./bin/hdfs --daemon start namenode // an01, 네임 노드 실행

./bin/hdfs --daemon start zkfc // an01, 주키퍼 장애 컨트롤러 실행

./sbin/hadoop-daemons.sh start datanode // an01, 데이터 노드들 실행
* daemon.sh, daemons.sh 명령어가 둘 다 있어 daemons.sh 확인

```
./bin/hdfs namenode -bootstrapStandby // sn01, 스탠바이 네임노드 초기화, 최초 실행 1번만
* 액티브 네임노드의 메타데이터가 스탠바이 네임노드로 복사됨

./bin/hdfs --daemon start namenode // sn01, 네임 노드 실행

./bin/hdsfs --daemon start zkfc // sn01, 주키퍼 장애 컨트롤러 실행

./sbin/start-yarn.sh // rm01, 얀 클러스터, 리소스 매니저 실행
```

2. Hadoop install

- 실행 중인 프로세스 확인

```
action server: all
pwd: /home/hadoop/hadoop-3.1.2

jps
```

an01

```
[hadoop@an01 hadoop-3.1.2]$ jps
1520 JournalNode
1682 DFSZKFailoverController
7721 Jps
1598 NameNode
[hadoop@an01 hadoop-3.1.2]$
```

sn01

```
[hadoop@sn01 hadoop-3.1.2]$ jps
4499 Jps
1524 JournalNode
1896 DFSZKFailoverController
1791 NameNode
[hadoop@sn01 hadoop-3.1.2]$
```

rm01

```
[hadoop@rm01 hadoop-3.1.2]$ jps
1527 JournalNode
2471 Jps
2057 WebAppProxyServer
1678 ResourceManager
[hadoop@rm01 hadoop-3.1.2]$
```

dn01

```
[hadoop@dn01 hadoop-3.1.2]$ jps
1921 DataNode
1429 NodeManager
3350 Jps
[hadoop@dn01 hadoop-3.1.2]$
```

dn02

```
[hadoop@dn02 hadoop-3.1.2]$ jps
1952 DataNode
1426 NodeManager
3803 Jps
[hadoop@dn02 hadoop-3.1.2]$
```

dn03

```
[hadoop@dn03 hadoop-3.1.2]$ jps
1425 NodeManager
3620 Jps
1917 DataNode
[hadoop@dn03 hadoop-3.1.2]$
```

2. Hadoop install

- 액티브, 스탠바이 네임 노드 확인

action server: an01
pwd: /home/hadoop/hadoop-3.1.2

./bin/hdfs haadmin -getServiceStandby an01
./bin/hdfs haadmin -getServiceStandby sn01

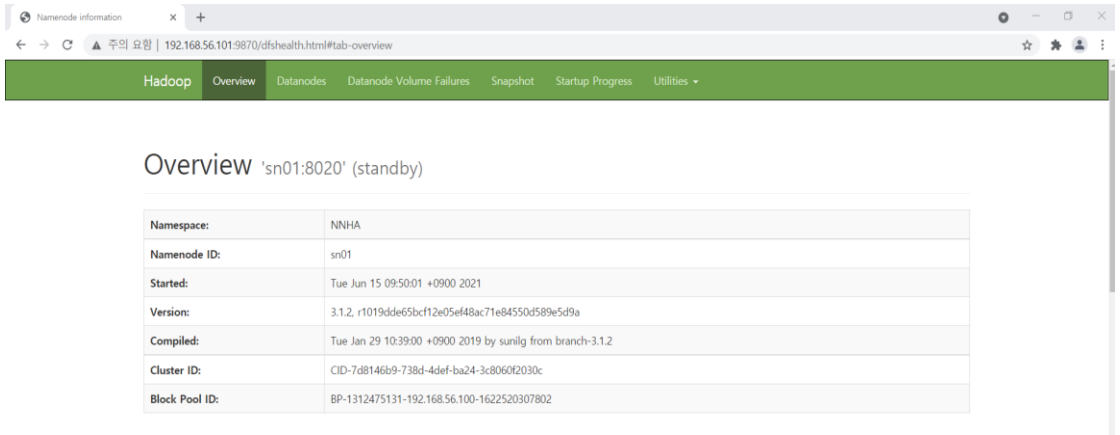
```
[hadoop@an01 hadoop-3.1.2]$ ./bin/hdfs haadmin -getServiceState an01
active
[hadoop@an01 hadoop-3.1.2]$ ./bin/hdfs haadmin -getServiceState sn01
standby
[hadoop@an01 hadoop-3.1.2]$
```

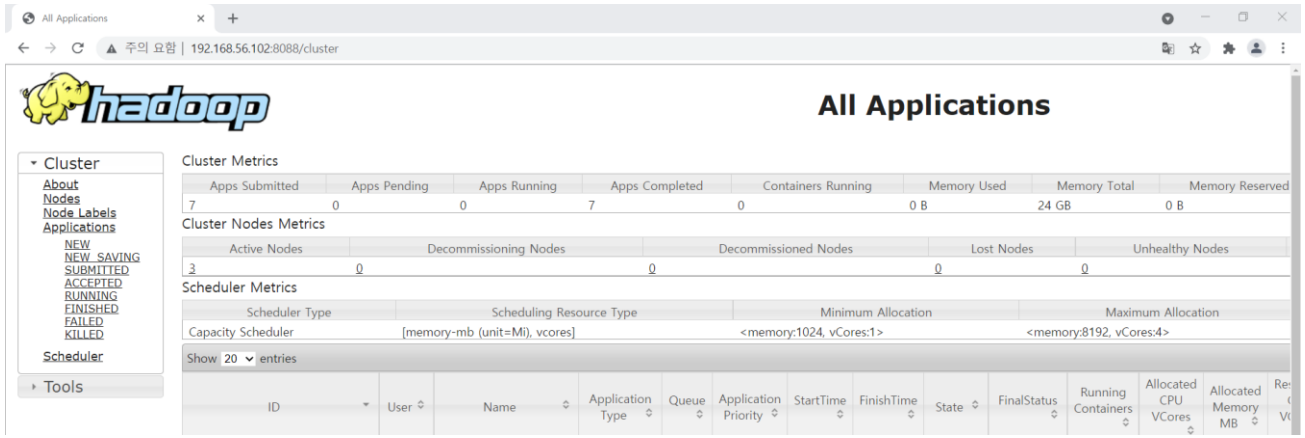
2. Hadoop install

- 웹 확인

action server: an01, sn01
pwd: /home/hadoop/hadoop-3.1.2

http://192.168.56.100:9870 // 액티브 네임 노드 웹
http://192.168.56.101:9870 // 스탠바이 네임노드 웹
http://192.168.56.102:8088 // 리소스 매니저 웹





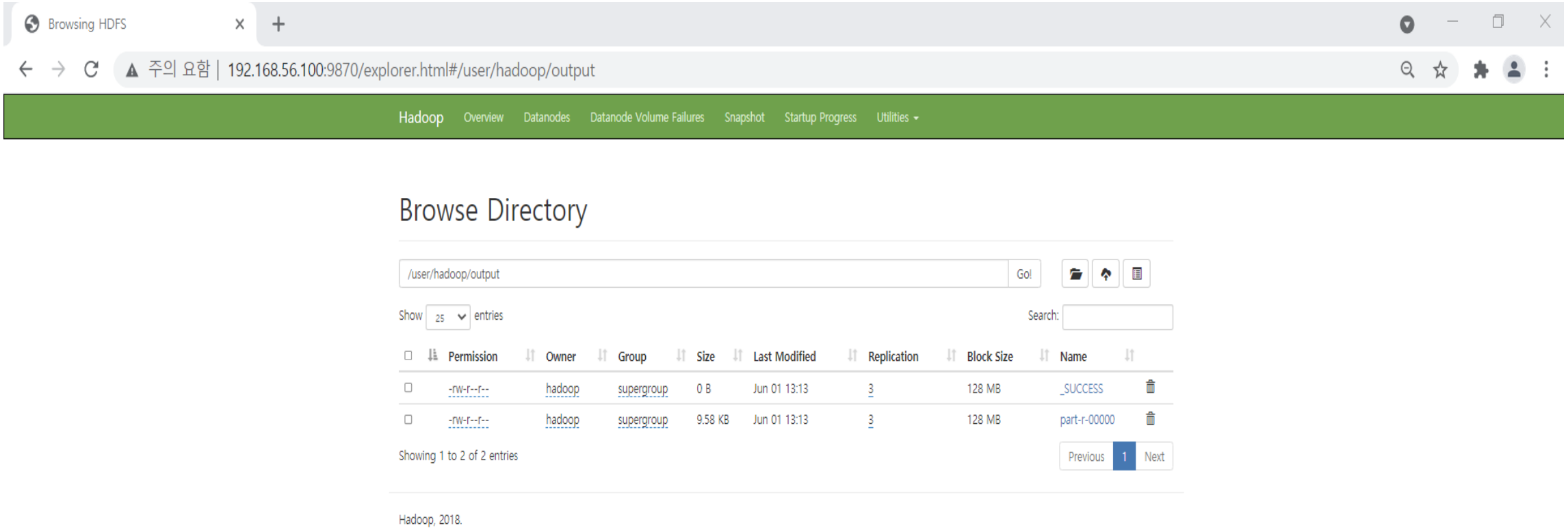
2. Hadoop install

- 맵 리듀스 example
- hadoop-env.sh 파일 활용

```
action server: an01
pwd: /home/hadoop/hadoop-3.1.2

hdfs dfs -ls /
hdfs dfs -mkdir /user
hdfs dfs -mkdir /user/hadoop
hdfs dfs -mkdir /user/hadoop/conf

hdfs dfs -put /home/hadoop/hadoop-3.1.2/etc/hadoop/hadoop-env.sh /user/hadoop/conf/ // 예제 파일 hdfs에 삽입
hdfs dfs -ls /user/hadoop/conf // 파일이 이동되었는지 조회

yarn jar /home/hadoop/hadoop-3.1.2/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2jar wordcount
conf output // 맵 리듀스 활용 wordcount 실행

hdfs dfs -ls /user/hadoop/output/ // 결과 파일 생성 확인
hdfs dfs -cat /user/hadoop/output/part-r-00000 // 워드 카운트 결과 확인
```

```
[hadoop@an01 hadoop-3.1.2]$ hdfs dfs -cat /user/hadoop/output/part-r-00000
"AS     1
"License");     1
"log    1
#       304
##      12
###     28
#export 1
$USER   1
${HADOOP_HOME}/logs     1
${HOME}/.hadooprc       1
'-'     1
'.'     1
'hadoop 1
'mapred 1
'yarn   1
(       1
(ASF)   1
(BUT    1
(Java   2
(Note   1
(command)_(subcommand)_USER.    1
(e.g.,  1
(file/dir       1
(i.e.,  2
(period)        1
(primarily)     1
(such   1
(superficially) 1
(the    1
)       1
**MUST  1
**MUST**        1
*NOT*   1
+'%Y%m%d%H%M')" 2
--config)       1
--daemon        1
-Dcom.sun.management.jmxremote.authenticate=false       1
-Dcom.sun.management.jmxremote.port=1026"       1
-Dcom.sun.management.jmxremote.ssl=false        1
-Dhadoop.security.logger=foo).  1
-Dsun.security.krb5.debug=true  1
-Dsun.security.spnego.debug"    1
-XX:+PrintGCDateStamps  1
-XX:+PrintGCDateStamps" 1
-XX:+PrintGCDetails     2
-XX:+PrintGCTimeStamps  2
-Xloggc:${HADOOP_LOG_DIR}/gc-rm.log-$(date      2
-Xms).  1
-Xmx).  1
-blah). 1
-f      1
-ls     1
-o      2
-s)}    1
```

- 웹 페이지로 파일 확인 가능
- 상단 Utilities -> Browse the file system



2. Hadoop install

- 하둡 클러스터 종료
- 시작의 역순으로 실행

action server: an01, sn01, rm01
pwd: /home/hadoop/hadoop-3.1.2

./sbin/stop-yarn.sh // rm01, 얀 종료

./bin/hdfs --daemon stop zkfc // sn01, 주키퍼 장애 컨트롤러 종료

./bin/hdfs --daemon stop namenode // sn01, 스탠바이 네임노드 종료

./sbin/hadoop-daemons.sh stop datanode // an01, 데이터 노드 종료

./bin/hdfs --daemon stop zkfc // an01, 주키퍼 장애 컨트롤러 종료

./bin/hdfs --daemon stop namenode // an01, 액티브 네임노드 종료

./bin/hdfs --dameon stop journalnode // an01, sn01, rm01, 저널 노드 종료

2. Hadoop install

- 최초 실행 이후 두 번째 부터의 실행

action server: an01, sn01, rm01
pwd: /home/hadoop/hadoop-3.1.2

./bin/hdfs --daemon start journalnode // an01, sn01, rm01, 저널 노드 실행

./bin/hdfs --daemon start namenode // an01, 액티브 네임노드 실행

./bin/hdfs --daemon start zkfc // an01, 주키퍼 장애 컨트롤러 실행

./sbin/hadoop-daemons.sh start datanode // an01, 데이터 노드 실행

./bin/hdfs --daemon start namenode // sn01, 스탠바이 네임노드 실행

./bin/hdfs --daemon start zkfc // sn01, 주키퍼 장애 컨트롤러 실행

./sbin/start-yarn.sh // rm01, 얀 실행