


# 하둡 클러스터 설치 과정

## 참고 사이트

How To Set Up a Hadoop 3.2.1 Multi-Node Cluster on Ubuntu 18.04 (2 Nodes)

To start: What is Hadoop?

 [https://medium.com/@jootorres\\_11979/how-to-set-up-a-hadoop-3-2-1-multi-node-cluster-on-ubuntu-18-04-2-nodes-567ca44a3b12](https://medium.com/@jootorres_11979/how-to-set-up-a-hadoop-3-2-1-multi-node-cluster-on-ubuntu-18-04-2-nodes-567ca44a3b12)



## 1. SSH 설치

```
sudo apt-get update
(pw : )

sudo apt install ssh
(y)
```

### ▼ pdsh란?

**pdsh**를 설치하면 원격 컴퓨터에 동시에 명령을 실행하거나 클러스터 환경에서 효율적인 작업 관리를 수행하는 데 사용할 수 있다. 예를 들어, 여러 대의 서버에 동일한 명령을 실행하거나 로그 파일을 수집하는 등의 작업을 수행할 때 유용하다.

```
vim .bashrc

# 가장 마지막 줄에 아래 코드 추가
export PDSH_RCMD_TYPE=ssh
```

### ▼ bashrc란?

**.bashrc**는 Bash 셸의 설정 파일이다. Bash는 대부분의 Linux 및 macOS 시스템에서 기본적으로 사용되는 셸이다. **.bashrc** 파일을 수정하여 새로운 환경 변수를 정의하거나, 특정 명령을 자동으로 실행하도록 설정할 수 있다. 또한, 터미널에서 사용하기 편리한 단축키(alias)를 설정하거나, 프롬프트의 외관을 변경하는 등의 작업을 수행할 수 있다.

### ▼ PDSH\_RCMD\_TYPE란?

**export PDSH\_RCMD\_TYPE=ssh**는 환경 변수 **PDSH\_RCMD\_TYPE**을 설정하는 명령이다. 이 명령은 현재 셸 세션에서 **PDSH\_RCMD\_TYPE** 변수를 **ssh**로 설정하도록 지정한다.

**PDSH\_RCMD\_TYPE** 환경 변수는 **pdsh** 도구에서 사용되는 원격 명령 실행 프로토콜을 정의한다. 여기서 **ssh**는 Secure Shell(SSH) 프로토콜을 사용하여 원격 명령을 실행하도록 설정하는 것을 의미한다.

일반적으로 `pdsh` 는 원격 시스템에 접속하여 명령을 실행하는 데 사용된다. `PDSH_RCMD_TYPE` 변수를 `ssh` 로 설정하면 `pdsh` 는 SSH를 통해 원격 시스템에 접속하고 원격으로 명령을 실행할 수 있다. 이를 통해 여러 대의 원격 시스템에 동시에 명령을 전달하거나 클러스터 환경에서 작업을 관리하는 등의 작업을 수행할 수 있다.

따라서 `export PDSH_RCMD_TYPE=ssh` 명령은 `pdsh` 도구를 사용하여 SSH 프로토콜을 통해 원격 명령을 실행하도록 설정하는 것을 의미한다.

```
# SSH키 쌍을 생성하는 명령어 (추가 옵션 : 비밀번호 없이 키를 생성하도록 지정)
ssh-keygen -t rsa -P ""

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

ssh localhost
(yes)
```

## JAVA 설치

```
sudo apt install openjdk-8-jdk
java -version
```

## Hadoop 설치

```
sudo wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.5/hadoop-3.3.5

tar -xvzf hadoop-3.3.5.tar.gz

# hadoop-3.3.5.tar.gz 폴더의 이름을 hadoop으로 변경하는 명령어 (쉬운 명령을 위해)
mv hadoop-3.3.5 hadoop
```

## hadoop-env.sh (for JAVA\_HOME)

```
vim ~/hadoop/etc/hadoop/hadoop-env.sh

# 맨 아래 코드 붙여넣기
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

## hadoop 폴더 경로 변경

```
sudo mv hadoop /usr/local/hadoop
```

#### ▼ 경로를 변경해주는 이유

"sudo mv hadoop /usr/local/hadoop" 명령은 "hadoop" 폴더의 디렉토리를 "/usr/local/hadoop"으로 변경하는 명령입니다.

일반적으로 "hadoop"과 같은 소프트웨어는 보통 시스템에서 `/usr/local` 디렉토리 아래에 설치됩니다. "/usr/local"은 시스템 전체에서 공유되는 로컬 소프트웨어의 설치 위치로 약속된 디렉토리입니다.

위의 명령을 실행하면 "hadoop" 폴더가 현재 위치에서 "/usr/local/hadoop" 디렉토리로 이동됩니다. 이렇게 폴더를 이동시키면 "hadoop" 디렉토리는 "/usr/local/hadoop"에 위치하게 됩니다.

"/usr/local/hadoop" 디렉토리로 이동시키는 이유는 일반적으로 Hadoop과 관련된 파일과 설정은 이 디렉토리에 위치시키기 때문입니다. 이렇게 하면 Hadoop을 사용하는 다른 프로그램이나 스크립트에서 Hadoop의 위치를 기대하고 참조하기 쉬워지며, 관리 및 설정이 용이해집니다. 또한, 이렇게 표준 디렉토리 구조를 따르면 Hadoop의 경로를 변경하거나 관리하는 데 더 편리하고 일관된 방식으로 처리할 수 있습니다.

## environment 수정

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

```
# 기존의 path 지우고 아래 path 붙여넣기
```

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/ga
```

## hadoopuser라는 user 추가하기

```
sudo adduser --gecos "" hadoopuser
```

```
# 아래의 코드 돌리기
```

```
sudo usermod -aG hadoopuser hadoopuser
```

```
sudo chown hadoopuser:root -R /usr/local/hadoop/
```

```
sudo chmod g+rwX -R /usr/local/hadoop/
```

```
sudo adduser hadoopuser sudo
```

#### ▼ 각 명령어 설명

1. `sudo usermod -aG hadoopuser hadoopuser` : 이 명령은 `hadoopuser` 사용자를 `hadoopuser` 그룹에 추가하는 명령입니다. `usermod` 명령은 사용자 계정의 속성을 변경하는 데 사용됩니다. `aG` 옵션은 사용자를 추가 그룹에 추가하는 옵션입니다. 즉, `hadoopuser` 사용자를 `hadoopuser` 그룹에 추가합니다.
2. `sudo chown hadoopuser:root -R /usr/local/hadoop/` : 이 명령은 `/usr/local/hadoop/` 디렉토리의 소유자를 `hadoopuser` 로, 그룹을 `root` 로 변경하는 명령입니다. `chown` 명령은 파일 또는 디렉토리의 소유자

와 그룹을 변경하는 데 사용됩니다. **R** 옵션은 하위 디렉토리와 파일에 대해서도 재귀적으로 변경을 적용하라는 옵션입니다.

3. `sudo chmod g+rwX -R /usr/local/hadoop/` : 이 명령은 `/usr/local/hadoop/` 디렉토리의 그룹에게 읽기, 쓰기, 실행 권한을 부여하는 명령입니다. `chmod` 명령은 파일 또는 디렉토리의 권한을 변경하는 데 사용됩니다. `g+rwX` 는 그룹에게 읽기(r), 쓰기(w), 실행(x) 권한을 부여하는 옵션입니다. **R** 옵션은 하위 디렉토리와 파일에 대해서도 재귀적으로 변경을 적용하라는 옵션입니다.
4. `sudo adduser hadoopuser sudo` : 이 명령은 `hadoopuser` 사용자를 `sudo` 그룹에 추가하는 명령입니다. `adduser` 명령은 사용자를 추가하는 데 사용됩니다. `sudo` 그룹에 사용자를 추가하면 해당 사용자에게 관리자 권한이 부여됩니다. 따라서 `hadoopuser` 사용자를 관리자로 추가하려는 목적으로 사용됩니다.

위의 명령어들은 Hadoop 관련 디렉토리와 사용자 권한을 설정하는 과정을 수행합니다. 이를 통해 `hadoopuser` 사용자에게 Hadoop 디렉토리에 대한 적절한 권한을 부여하고, 필요한 작업을 수행할 수 있도록 설정합니다.

## 연결할 서버 IP 확인

```
203.252.206.*** master***
203.252.206.*** slave01***
.
.
.
203.252.206.*** slave13***
203.252.206.*** slave14***
203.252.206.*** slave15***
```

## host파일을 열어서 네트워크 구성 삽입

```
sudo vim /etc/hosts

# 편집기에 아래 내용 삽입
203.252.206.*** master***
203.252.206.*** slave01***
.
.
.
203.252.206.*** slave13***
203.252.206.*** slave14***
203.252.206.*** slave15***
```

이제 **slave node**를 만들어야 하는데, **master**와 동일한 환경과 설정을 가지게 세팅 (중요)

---

## 마스터 노드 이름 변경하기

```
sudo vim /etc/hostname

#smu-server를 지우고 아래껄로 바꾸기
hadoop-master
```

## 슬래브 노드 이름 변경하기

```
sudo nano /etc/hostname

#각 슬레이브 노드마다 다르게 변경해야 함.
slave01***
```

## 리부팅

```
sudo reboot
```

- 
- 여기까지가 우선 마스터 작업 1차 종료
  - 15대 slave 동일한 작업 (자바 하둡 설치 호스트 이름 변경 등)
  - 완료 후, 아래 작업으로 넘어가기
- 

## 마스터 노드에 접속해서 SSH 서로 공유하기

```
su - hadoopuser

ssh-keygen -t rsa
(enter)
(enter)
(enter)

ssh-copy-id hadoopuser@hadoop-master
(yes)
```

```
ssh-copy-id hadoopuser@hadoop-slave01
(yes)
```

```
ssh-copy-id hadoopuser@hadoop-slave02
ssh-copy-id hadoopuser@hadoop-slave03
ssh-copy-id hadoopuser@hadoop-slave04
ssh-copy-id hadoopuser@hadoop-slave05
ssh-copy-id hadoopuser@hadoop-slave06
ssh-copy-id hadoopuser@hadoop-slave07
ssh-copy-id hadoopuser@hadoop-slave08
ssh-copy-id hadoopuser@hadoop-slave09
ssh-copy-id hadoopuser@hadoop-slave10
ssh-copy-id hadoopuser@hadoop-slave11
ssh-copy-id hadoopuser@hadoop-slave12
ssh-copy-id hadoopuser@hadoop-slave13
ssh-copy-id hadoopuser@hadoop-slave14
ssh-copy-id hadoopuser@hadoop-slave15
```

→ 모든 슬레이브에 대해서 복사 해줌

```
hadoopuser@hadoop-master:~$ ssh-copy-id hadoopuser@hadoop-slave08
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/hadoopuser/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
hadoopuser@hadoop-slave08's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'hadoopuser@hadoop-slave08'"
and check to make sure that only the key(s) you wanted were added.
```

## core-site.xml

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

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://hadoop-master:9000</value>
</property>
</configuration>
```

## hdfs-site.xml

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

```

<configuration>
<property>
<name>dfs.namenode.name.dir</name>
<value>/usr/local/hadoop/data/nameNode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/usr/local/hadoop/data/dataNode</value>
</property>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>

```

## worker file 수정 - slave 노드 이름 입력해주기

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

```

# 아래 내용 추가
hadoop-slave01
hadoop-slave02
hadoop-slave03
hadoop-slave04
hadoop-slave05
hadoop-slave06
hadoop-slave07
hadoop-slave08
hadoop-slave09
hadoop-slave10
hadoop-slave11
hadoop-slave12
hadoop-slave13
hadoop-slave14
hadoop-slave15

```

## 마스터의 구성 파일을 slave 로 복사하기

```

scp /usr/local/hadoop/etc/hadoop/* hadoop-slave01:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave02:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave03:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave04:/usr/local/hadoop/etc/ha

```

```

scp /usr/local/hadoop/etc/hadoop/* hadoop-slave05:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave06:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave07:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave08:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave09:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave10:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave11:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave12:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave13:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave14:/usr/local/hadoop/etc/ha
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave15:/usr/local/hadoop/etc/ha

```

#### ▼ 코드 설명

`/usr/local/hadoop/etc/hadoop/` 디렉토리에 있는 모든 파일을 `SWR04` 서버의 `/usr/local/hadoop/etc/hadoop/` 디렉토리로 복사하는 명령입니다. `scp` 명령은 로컬 시스템과 원격 시스템 간에 파일을 복사하는 데 사용됩니다. `*` 는 모든 파일을 의미합니다.

즉, 위의 명령어들은 마스터 머신에 있는 Hadoop의 구성 파일을 슬레이브 머신들로 복사하여 동일한 구성을 가지도록 합니다. Hadoop의 구성 파일은 클러스터 설정과 관련된 중요한 정보를 포함하고 있으므로, 이를 슬레이브 머신들로 복사함으로써 마스터와 슬레이브 간에 일관된 구성을 유지할 수 있습니다.

## hdfs format

```

source /etc/environment
hdfs namenode -format

```

## 하둡 분산 시스템 시작

```

/usr/local/hadoop/sbin/start-dfs.sh
/usr/local/hadoop/sbin/stop-dfs.sh

```



```
hadoopuser@hadoop-master:~$ /usr/local/hadoop/sbin/start-dfs.sh
Starting namenodes on [hadoop-master]
Starting datanodes
hadoop-slave03: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave09: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave05: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave07: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave04: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave06: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave02: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave08: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave10: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave11: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave12: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave14: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave15: WARNING: /usr/local/hadoop/logs does not exist. Creating.
hadoop-slave13: WARNING: /usr/local/hadoop/logs does not exist. Creating.
Starting secondary namenodes [hadoop-master]
```

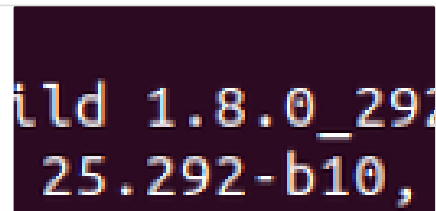
실행했을 때 permission denied 나오면서 실행 안되면 ?

```
echo "ssh" | sudo tee /etc/pdsh/rcmd_default
```

#### Hadoop Installation on Ubuntu (나의 하둡 설치 삼질기)

하둡? Hadoop은 여러 클러스터에서 대규모 데이터 셋을 분산 처리할 수 있게 하는 프레임워크입니다. 아마 빅데이터 수업이나 데이터마이닝 수업을 하실 때 하둡을 설치할 일이 생기실텐데, 저 또한 이러한 이유로 설치를 하게 되었습니다. 이 포스팅을 보는 분들은 적

 <https://minutemaids.tistory.com/88>



위의 코드 작업이 제대로 수행되었는지 확인하려면 아래의 명령어를 입력하십시오.

```
jps
```

#### ▼ jps?

"jps"는 Java 프로세스 상태 도구로, 현재 실행 중인 Java 프로세스 목록을 보여주는 명령어입니다. 이 명령어를 실행하면 현재 실행 중인 프로세스 중에서 Hadoop 관련 프로세스를 확인할 수 있습니다. 이를 통해 Hadoop 서비스가 시작되었는지, 어떤 리소스가 초기화되었는지 등을 파악할 수 있습니다.

따라서 "jps" 명령어를 실행하여 초기화된 리소스와 실행 중인 Hadoop 관련 프로세스 목록을 확인할 수 있습니다. 이를 통해 Hadoop 클러스터의 상태와 서비스가 제대로 시작되었는지를 확인할 수 있습니다.

```
hadoopuser@hadoop-master:~$ jps
4138 Jps
3771 NameNode
4014 SecondaryNameNode
hadoopuser@hadoop-master:~$
```

Now we need to do the same in the slaves:

```
hadoopuser@hadoop-slave1:~$ jps
1808 DataNode
2024 Jps
hadoopuser@hadoop-slave1:~$
```

```
hadoopuser@hadoop-slave2:~$ jps
1814 DataNode
2031 Jps
hadoopuser@hadoop-slave2:~$
```

```
hadoopuser@hadoop-master:~$ /usr/local/hadoop/sbin/start-dfs.sh
Starting namenodes on [hadoop-master]
pdsh@hadoop-master: hadoop-master: rcmd: socket: Permission denied
Starting datanodes
pdsh@hadoop-master: localhost: rcmd: socket: Permission denied
pdsh@hadoop-master: hadoop-slave01: rcmd: socket: Permission denied
Starting secondary namenodes [hadoop-master]
pdsh@hadoop-master: hadoop-master: rcmd: socket: Permission denied
hadoopuser@hadoop-master:~$ echo "ssh" | sudo tee /etc/pdsh/rcmd_default
ssh
hadoopuser@hadoop-master:~$ jps
6710 Jps
hadoopuser@hadoop-master:~$ /usr/local/hadoop/sbin/start-dfs.sh
Starting namenodes on [hadoop-master]
Starting datanodes
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
hadoop-slave01: WARNING: /usr/local/hadoop/logs does not exist. Creating.
Starting secondary namenodes [hadoop-master]
hadoopuser@hadoop-master:~$ jps
7745 Jps
7026 NameNode
7272 DataNode
7562 SecondaryNameNode
```

## YARN 구성하기

```
export HADOOP_HOME="/usr/local/hadoop"
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_HDFS_HOME=$HADOOP_HOME
```

```
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
```

## 각 slave node에서 yarn-site.xml 파일 열기

```
su - hadoopuser
```

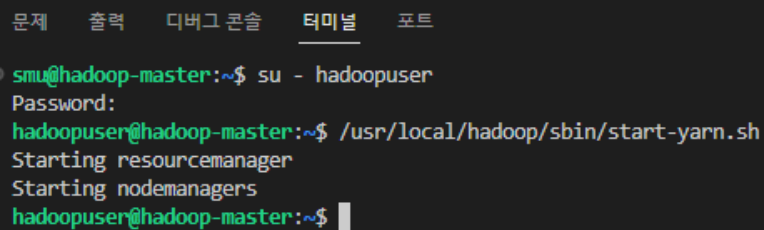
```
sudo vim /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

```
<property>
<name>yarn.resourcemanager.hostname</name>
<value>hadoop-master</value>
</property>
```

## master node에서 yarn을 시작해보자

```
/usr/local/hadoop/sbin/start-yarn.sh
/usr/local/hadoop/sbin/stop-yarn.sh
```

# 정상적으로 작동한다면 (아래 사진과 같이 나옴)



```
문제   출력   디버그 콘솔   터미널   포트
smu@hadoop-master:~$ su - hadoopuser
Password:
hadoopuser@hadoop-master:~$ /usr/local/hadoop/sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoopuser@hadoop-master:~$
```