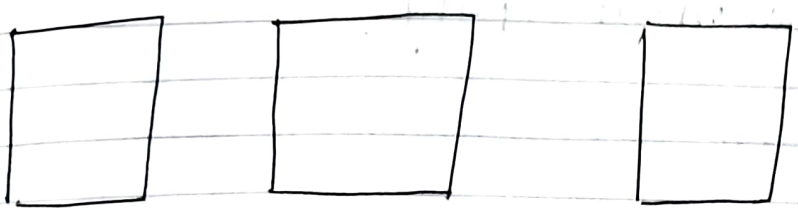
MultiNode



* Masternode :-

i) create mic (create user)
ii) download java, ssh
iii) Enable ssh & start
iv) open putty
v) download hadoop. tar it
vi) ~~generate Key-gen~~
vii) then /etc/hosts → add ip of that master node, dn1, dn2, d3
viii) add .bashrc file & edit
ix) generate key-gen
x) copy that key (ssh-copy-id username@masternode1)
xi) login to check
xii) it will not ask password
xiii) create /usr/local/hadoop folder
xiv) give ownership permission
xv) mv ~~folder~~ file (hadoop) in hadoop folder
xvi) create folder nn(namenode) & tmp
↳ usr/local/hadoop/hd-data/nn
xviii) sudo init 0

← after any change run command source

* masternode (mic)

i) create 2 clone
(a) datanode 1
(b) datanode 2

Imp* start all mic
masternode (data 1
datanode 2

In masternode & /etc/hosts - add dn1, w dn2 ip

Page No.: YOUVA
Date:

add

In n

* Masternode (putty).

Edit the files
1]

imp* make sure you copy file from master which is given by github link (hadoop-main - master)

- hadoop-env.sh. - ~~[illegible]~~ uploaded data from single node installation file

changes
* (1) Core-site.xml - In <value> masternode </Valu
↓
name of your namenod

(2) yarn-site.xml - ~~NO CHANGE~~ edit (masternode)
(3) ~~hdfs~~ hdfs.xml - NO CHANGE
4) mapred.xml NO CHANGE
5) workers - dn1 dn2 dn3

- check nn folder path
- check permission
- check $HADOOP_HOME /usr/local/hadoop
echo $PATH

- 1) format namenode.
2) hadoop namenode - formate
3) start-all.sh
4) jps →

* dn1 :-

1] Hostname

2] check ip address cat /etc/hosts

3] ~~mkdir~~

3) Creat Folder

• mkdir -p /usr/local/hadoop/hd-data/yarn/data

• mkdir -p /usr/local/hadoop/hd-data/yarn/logs

4] Edit ~~5~~ file :-

(1) core-fisik.xml - remember whatver your s name in namenode etc/hosts
e.g masternode

(2) yarn-sike.xml - ~~edooooge~~ ~~mooooo~~.

(3) Hadoop → from single node data

(4) .hdfs

mapred

* check nano coorker file

* dn1 :-

i) give hostname
ii) check .bashrc file & do source ~/.bashrc
iii) nano workers (add dn1, dn2)
iv) /etc/hosts (add ip & name)

- create 2 folder :-
  mkdir -p /usr/local/hadoop/hd-data/yarn/data
  /logs

- Edit 5 files :-
  hadoop → upload single node data
  core →
  mapred
  hdfs
  yarn
  } remember in this (in masternode what name give in all this file, that Exact name give)

* dn2 :- all same as dn1

* masternode :-

i) format Namenode
ii) start-all.sh
iii) jps

dn1 :-

i) start-dfs.sh
ii) jps
or
start-all.sh

dn2 :-

i) start-dfs.sh
ii) jps

- Imp :- if any node or anything not work then give

for [redacted] :- $HDOOP_HOME/sbin/hadoop-daemon.sh
/ HDFS :- start namenode

$HADOOP_HOME/sbin/hadoop-daemon.sh
start datanode
—11— start secondarynamenode

yarn :- $HDOOP_HOME/sbin/yarn-daemon.sh start resourcemanager

—11— start nodemanger

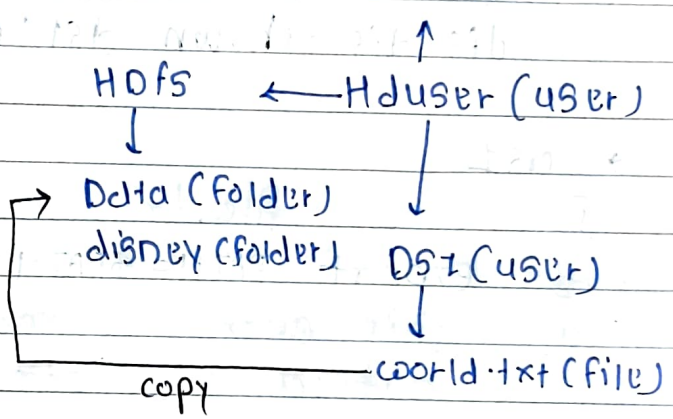
Mapredue :- $HDOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver

$HDOOP_HOME/sbin/start-dfs.sh
/start-yarn.sh

upload in disney
↑ (tom.filez)
↑ (jerry.filez)
(tom, jerry) user add in group
↑
client (group)

ACL

Assignment :-



- create dsi user
  create dsi-data directory in ~~dsi user~~ HDfs
  make dsi owner of dsi-data directory
  make file in dsi & upload it in dsi-data directory
  in hduser { create group by client (this group will create in hduser
  add tom & jerry user in client group
  Now make ~~sure~~ directory disney in HDfs
  Now make sure tom & jelly can upload the file in disney folder → to upload folder/file in disney tom & jelly must have their group also, so Now change group Name of client into ~~tom~~ & ~~jerry~~
  Disney also disney have (read-write) permission

* whenver you poweroff on machine there is need to start-all sh

creat hello.txt file in ds1
append hello.txt data in earlier file of ds1 that uploadid in HDFS

* Master :-

- ① Creat user ds1
② creat ds1-data dire. in HDFS
#: hdfs dfs -mkdir /[redacted]/ds-data

③ make owner ds1 of ds-data dire.
hdfs dfs -chown ds1:ds1 /ds-data

* ds1
1] creat file.txt
2] export PATH=$PATH:/usr/local/hadoop/bin
give this above command in .bashrc also outside (temporary)

* [redacted]

3] Now copy file.txt to hdfs
hdfs dfs -put file.txt /ds1-data

4) su - hduser

* hduser
i) add group client
ii) Now add Tom, jerry user to client group
sudo adduser --ingroup client tom
sudo adduser --ingroup client jerry

* whenver you poweroff on machine there is need to start-all sh

creat hello.txt file in ds1
append hello.txt data in earlier file of ds1 that uploadid in HDFS

* Master :-

• ① Creat user ds1
② creat ds1-data dire. in HDFS
#: hdfs dfs -mkdir /[redacted]/ds-data

③ make owner ds1 of ds-data dire.
hdfs dfs -chown ds1:ds1 /ds-data

* ds1
1] creat file.txt
2] export PATH=$PATH:/usr/local/hadoop/bin
give this above command in .bashrc also outside (temporary)

* [redacted]

3] Now copy file.txt to hdfs
hdfs dfs -put file.txt /ds1-data

4) su - hduser

* hduser
i) add group client
ii) Now add Tom, jerry user to client group
sudo adduser --ingroup client tom
sudo adduser --ingroup client jerry

iii) Creat disney folder in hdfs
iv] mkdir dfs -mkdir /disney

iv) To access disney folder for tom, jerry change client group name to group-disney
hdfs dfs -chgrp client /disney

v] Now change give read, write permissie-to disney.
hdfs dfs -chmod 775 /disney

vi) log in Tom

* Tom
i) Give .bashrc
export PATH = $PATH : /usr/local/hadoop/bin
also give same command to outside.
source ~/.bashrc

ii) ~~Upload~~ creat tom.txt file
iii) upload to hdfs
hdfs dfs -put tom.txt /disney

iv) su jerry

* Jerry
i) Give .bashrc
export PATH = $PATH : /usr/local/hadoop/bin
also give same command to outside
source ~/.bashrc

ii) creat jerry.txt file & upload to
hdfs dfs -put jerry.txt /disney

* append

* i) ds1 :-
  i] creeat file. hello.txt
  ii) hello.txt append to file1.txt which is uploaeleed on hdfs folder ds-data
- hdfs dfs -appendTofile hello.txt /ds1-data/file

- hdfs dfs -cat /ds1-data/file1.txt

* DECOMMISION *

* master
① go to hadoop.

cd /usr/local/hadoop/etc/hadoop

② go creeat file dfs.host.exclude
nano dfs.host.exclude
in this file add the datanode which you want to decommision (name same as hostname)

3) open hdfs-site.xml add property
<property>
<name> dfs.hosts.exclude </name>
<value> /usr/local/hadoop/etc/hadoop/dfs.hosts.exclude </value>
</property>

4] Now stop namenode
hdfs --daemon stop namenode

(5) Now start
hdfs --daemon start namenode

(6) ~~Restart~~ Refresh the node

hdfs --=do

hdfs dfsadmin -refreshNodes.

7] Report

hdfs dfsadmin -report

~~There all~~ so
There dll command make datanode 4 inactive

8] To Active the dny

- Remove dny from nano dfs.hosts.exclude file
- Stop Namenode (hdfs --daemon stop namenode
- Start Namenode (hdfs -- daemon start namenode
- Refresh (-hdfs dfsadmin -refreshNodes)
- Report (hdfs dfsadmin -report)

TRASH

* master :-

(1) /usr/local/hadoop/etc/hadoop/ nano core-site.xml

```
<property>
<name> fs.trash.interval</name>
<value> 30 </value>
</property>
```

(2) hdfs --daemon stop Namenode
start Namenode

(3) create folder
dsy-data (hdfs dfs -mkdir /dsy-data)

(4) create file in dsy-data
nano file.txt

(5) To trash file.txt
[hdfs dfs -rm -f /dsy-data/file.txt
if this give error then ~~off~~ the safemode
- hdfs dfs admin -safemode (off) leave
- then Now this command
hdfs dfs -rm -f /dsy-data/file.txt

- on safemode
hdfs dfs admin -safemode get
—ll— enter

Recover Trash file

(6) To check Trash file path

hdfs dfs -ls /user

i) Now check path add that in next command upto to get trash to ~~dsi-data folder~~ file1.txt

hdfs dfs -ls /user/hduser

-ll /.Trash/current/dsi-data/file1.txt

~~Edureka~~

(7) Now ~~re~~move this trash file dsi-

hdfs dfs -mv /usr/hduser/.Trash/current/dsi-data1/file1.txt /dsi-data

(8) check

hdfs dfs -ls /

RACK

* hduser

(1) go to hadoop

cd /usr/local/hadoop/etc/hadoop

(2) Now make t1.py file and past python code

3] again make t2.txt file and copy text from github but change IP addr of datanode 3 and datanode 4.

(make sure your IP-address)

4) open core-site.xml file

<property>
<name> net.topology.script.file.name</name>

<value> /usr/local/hadoop/etc/hadoop/tl.py</value>

</property>

- nano tl.py → python file
  ↓ in this file change file name (give txt file name

3. stop-dfs.sh
4. start-dfs.sh
5. hdfs dfsadmin -PrintTopology

<property>
<name> Net. topology. script. file. name </name>
<value> /usr/local/hadoop/etc/hadoop/tl.py
topology.py </value>
</property>

- nano tl.py → python file
  ↓ in this file change file name (give txt file name)

3. Stop-dfs.sh
4. Start-dfs.sh
5. hdfs dfsadmin - Print Topology