

Exp No: 9

Date:

HADOOP

SET UP A SINGLE HADOOP CLUSTER AND SHOW THE PROCESS USING WEB UI

AIM:

To set-up one node Hadoop cluster.

PROCEDURE:

1. System Update
2. Install Java
3. Add a dedicated Hadoop user
4. Install SSH and setup SSH certificates
5. Check if SSH works
6. Install Hadoop
7. Modify Hadoop config files
8. Format Hadoop filesystem
9. Start Hadoop
10. Check Hadoop through web UI
11. Stop Hadoop

THEORY

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from a single server to thousands of machines, each offering local computation and storage.

HADOOP ARCHITECTURE

Hadoop framework includes following four modules:

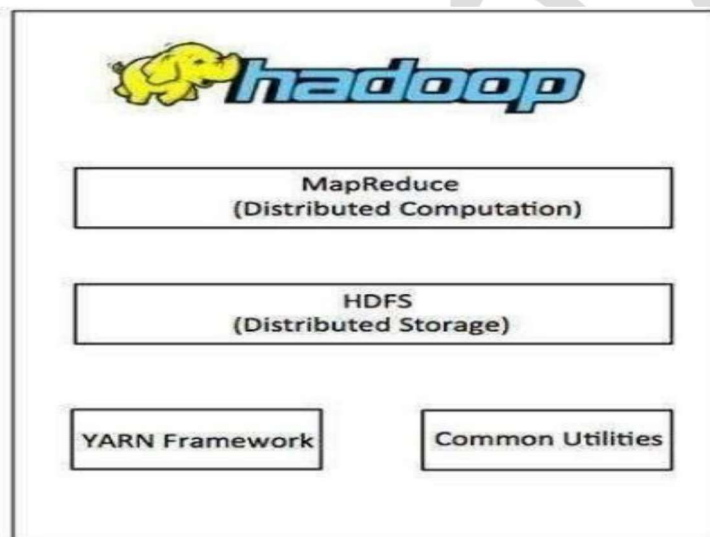
Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provide filesystem and OS level abstractions and contain the necessary Java files and scripts required to start Hadoop.

Hadoop YARN: This is a framework for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data.

Hadoop MapReduce: This is a YARN-based system for parallel processing of large data sets.

We can use following diagram to depict these four components available in Hadoop framework.



PROCEDURE

\$ nano ~/.bashrc

```
hadoop@kali: ~
File Actions Edit View Help
GNU nano 7.2 .bashrc

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_202
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=~/.hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_STREAMING=$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar
export HADOOP_LOG_DIR=$HADOOP_HOME/logs
export PDSH_RCMD_TYPE=ssh
export HADOOP_COMMON_LIB_NATIVE_DIR=~/.hadoop/lib/native
export HADOOP_OPTS="-Djava.library.path=~/.hadoop/lib/native"
```

\$ nano \$HADOOP_HOME/etc/hadoop/hadoop-env.sh

```
kali-linux-2023.4-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
hadoop@kali: ~
File Actions Edit View Help
GNU nano 7.2 /home/hadoop/.hadoop/etc/hadoop/hadoop-env.sh

#
# Hadoop: Setting the JAVA_HOME variable
#
# Many of the options here are built from the perspective that users
# may want to provide overrides/overrides on the command line.
# For example:
#
# JAVA_HOME=/usr/java/testing-hdfs-etc
#
# Therefore, the vast majority (NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferred, modify this file accordingly.
#
# General settings for HADOOP
#
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
#
# The Java implementation to use. By default, this environment
# variable is REQUIRED on all platforms except OS X.
# export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_202
#
# The language environment in which Hadoop runs. Use the English
# environment to ensure that logs are printed as expected.
# export LANG=en_US.UTF-8
#
# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=
#
# Location of Hadoop's configuration information. I.e., where this
# file is living. If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#
# NOTE: It is recommended that this variable not be set here but in
# /etc/profile.d or equivalent. Some options (such as
# --config) may react strangely otherwise.
#
# export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

\$nano \$HADOOP_HOME/etc/hadoop/core-site.xml

```
File Actions Edit View Help
GNU nano 7.2 /home/hac
<?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://localhost:9000</value> </property>
<property>
<name>hadoop.proxyuser.dataflair.groups</name> <value>*</value>
</property>
<property>
<name>hadoop.proxyuser.dataflair.hosts</name> <value>*</value>
</property>
<property>
<name>hadoop.proxyuser.server.hosts</name> <value>*</value>
</property>
<property>
<name>hadoop.proxyuser.server.groups</name> <value>*</value>
</property>
</configuration>
```

\$nano \$HADOOP_HOME/etc/hadoop/hdfs-site.xml

```
File Actions Edit View Help
GNU nano 7.2 /home/hadoop/hadoop/etc/hadoop
<?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.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
</property>
</configuration>
```

\$nano \$HADOOP_HOME/etc/hadoop/mapred-site.xml

```
hadoop@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /home/hadoop/hadoop/etc/hadoop/mapred-  
?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>  
    <property>  
      <name>mapreduce.application.classpath</name>  
  
      <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>  
    </property>  
  </configuration>
```

\$ nano \$HADOOP_HOME/etc/hadoop/yarn-site.xml

```
hadoop@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /home/hadoop/hadoop/etc/hadoop/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>  
  <!-- Site specific YARN configuration properties -->  
  <property>  
    <name>yarn.nodemanager.aux-services</name>  
    <value>mapreduce_shuffle</value>  
  </property>  
  <property>  
    <name>yarn.nodemanager.env-whitelist</name>  
  
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPENDED_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>  
  </property>  
</configuration>
```

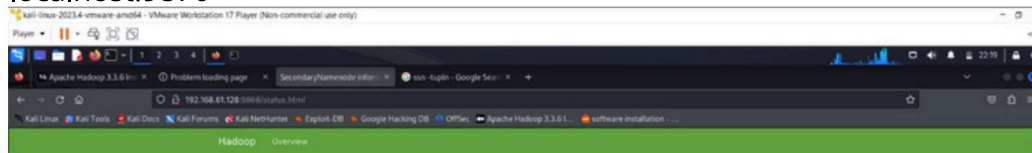
\$ start-all.sh

```
hadoop@kali: ~  
File Actions Edit View Help  
(hadoop@kali)~  
$ start-all.sh  
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.  
WARNING: This is not a recommended production deployment configuration.  
WARNING: Use CTRL-C to abort.  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [kali]  
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true  
2024-09-11 04:59:16,429 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
Starting resourcemanager  
Starting nodemanagers
```

\$ jps

```
(hadoop@kali)-[~]
$ jps
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
14436 NodeManager
16772 Jps
13830 SecondaryNameNode
14311 ResourceManager
13597 DataNode
13471 NameNode
(hadoop@kali)-[~]
```

localhost:9870



Overview

Version	3.3.6
Compiled	2023-06-18T08:22Z by ubuntu from (HEAD detached at release-3.3.6-RC1)
NameNode Address	localhost:9000
Started	Wed Aug 14 21:51:32 -0400 2024
Last Checkpoint	Never
Checkpoint Period	3600 seconds
Checkpoint Transactions	1000000

Checkpoint Image URI

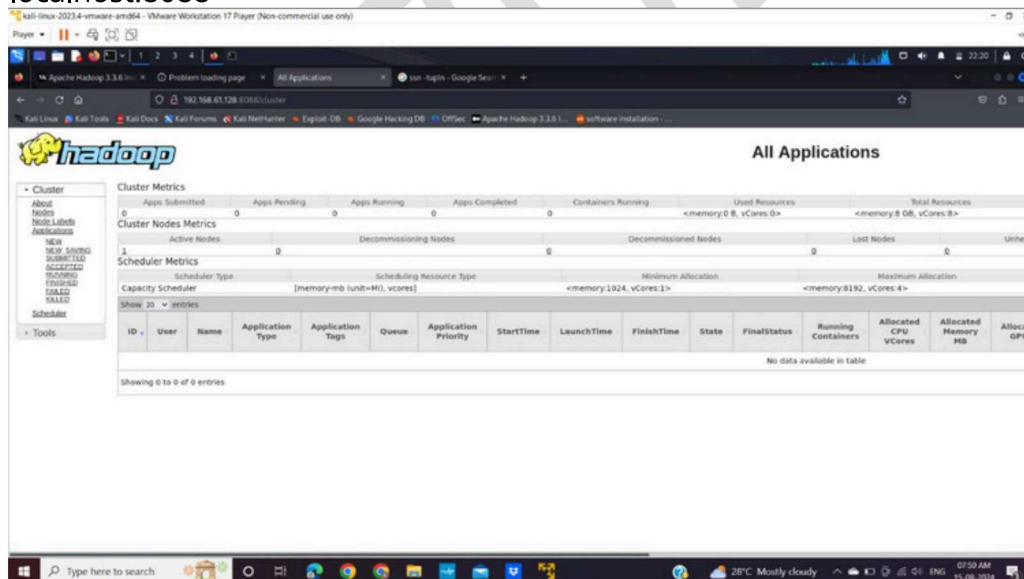
• file:///tmp/hadoop-kali/vfs/secondary

Checkpoint Editlog URI

• file:///tmp/hadoop-kali/vfs/secondary

Hadoop, 2023.

localhost:8088



RESULT:

Thus, Hadoop has been successfully installed.