Guide: HADOOP v3 / Centos 7 Installation

(Pseudo Distributed Mode)

(Feel free to update)

Note: Use as guide only. The SABA tutorial is more thorough.

1. Ensure user account has sudo access
2. Check: sudo ls
3. Add to sudoers:
4. Login as root: su
5. Execute: usermod -aG wheel **username**
6. Update / Change Hostname

<https://www.tecmint.com/set-change-hostname-in-centos-7/>

1. SSH Hostname

* See SABA tutorial

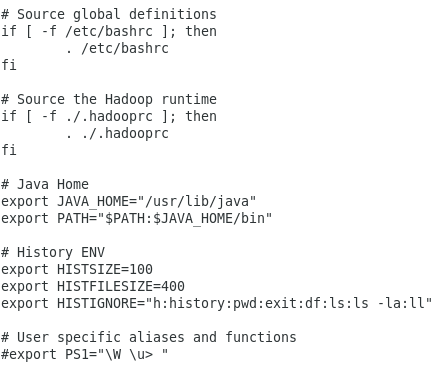
1. Install Java
2. Confirm if already installed  
   javac -version
3. Yum installation

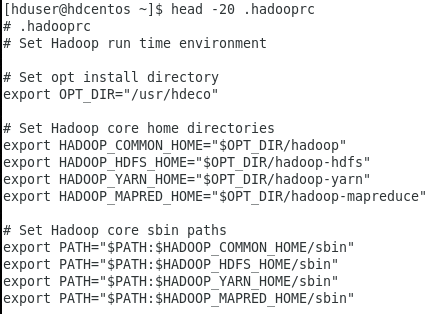
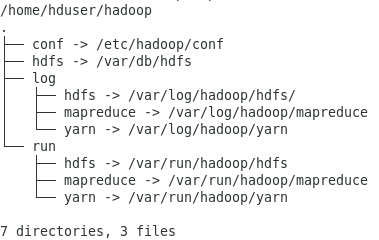
yum install java-1.8.0-openjdk

1. Set softlink same in .bashrc for JAVA\_HOME

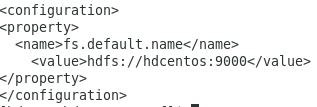
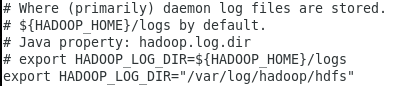


1. Modify .bashrc in home directory

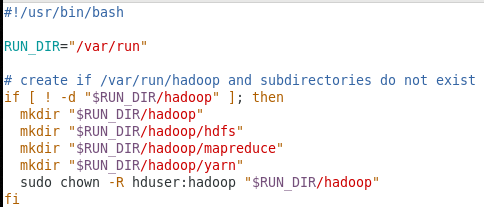


1. HADOOP Core
2. [BASIC]
3. Download repo:   
   **sudo curl -O** [**http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/hdp.repo**](http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/hdp.repo)
4. Move downloaded **hdp.repo** file to yum repository ( overwrite )  
   **mv /home/hduser/Downloads/hdp.repo /etc/yum.repos.d**
5. Install with yum:  
   **yum install hadoop hadoop-client**
6. Create **.hadooprc** for hadoop-core  
   
7. Create user-home’s hadoop tree below and necessary softlinks  
   
8. Locate **conf.empty** and copy hadoop-core files (hdfs, yarn, mapreduce) to **/etc/hadoop/conf** or **~/hadoop/conf**
9. cd /usr/hdp
10. cd **`**find . -name conf.empty**`**; ls # Note the back-quote in red
11. cp **hdfs\*** **yarn\*** **mapred\*** /etc/hadoop/conf
12. cd /etc/hadoop/conf; ls
13. Hdeco softlinks (core hadoop):

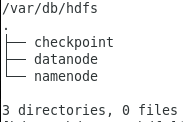


1. ~/hadoop/conf/core-site.xml  
   
2. ~/hadoop/conf/hadoop-env.sh  
   
3. Create the following directories:  
   /var/run/hadoop/hdfs  
   /var/run/hadoop/mapreduce  
   /var/run/hadoop/yarn  
   sudo chown -R **hduser:hadoop** /var/run/hadoop

**Note: /var/run/XXX** custom directories need to be re-created on shutdown / restart

**[Optional]**: Write a bash script that re-creates **/var/run/hadoop** and **sub-directories** and run as sudo (ex. sudo ./create\_rundir.sh )  


1. **HDFS** (Already installed in 3.1)
2. Conf: ~/hadoop/conf/hdfs-site.xml  
   
3. HDFS DB tree: /var/db/hdfs



1. Log: /var/log/hadoop/hdfs
2. Run: /var/run/hadoop/hdfs

* Re-run **mkdir** on run directory for every reboot

1. Format the hadoop file system for the very first time:  
   hdfs namenode -format
2. Start:  
   **hdfs --daemon start namenode**

**hdfs --daemon start datanode**

**hdfs --daemon start secondarynamenode**

1. Stop (reverse order):

**hdfs --daemon stop secondarynamenode**

**hdfs --daemon stop datanode**

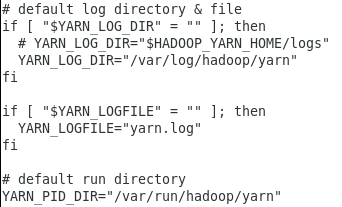
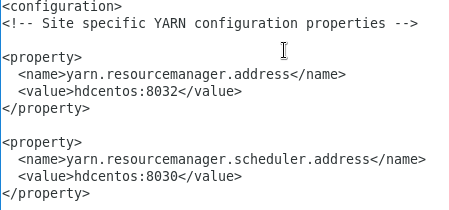
**hdfs --daemon stop namenode**

1. Check:

**jps**

or

**http://localhost:9870**

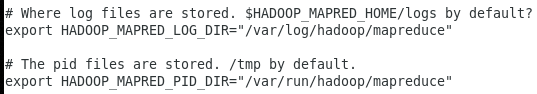
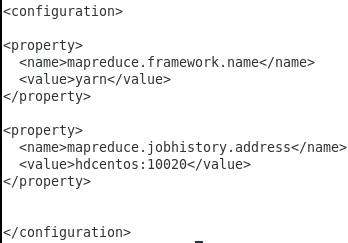
1. **YARN** (Already installed in 3.1)
2. Conf:  
   ~/hadoop/conf/yarn-env.sh  
     
     
   ~/hadoop/conf/yarn-site.xml (follow SABA tutorial for complete conf)  
   
3. DB: N/A
4. Log: /var/log/hadoop/yarn
5. Run: /var/run/hadoop/yarn
6. Start:

**yarn --daemon start resourcemanager**

**yarn --daemon start nodemanager**

1. Check for errors:

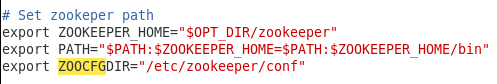
**grep ERR /var/log/hadoop/yarn/\*/\***

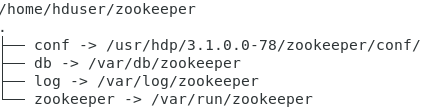
1. Stop: ( Reverse order of start )
2. WEB UI: hdcentos:8088
3. **MAPREDUCE** (Already installed in 3.1)
4. Conf:  
   ~/hadoop/conf/mapred-env.sh  
     
   ~/hadoop/conf/mapred-site.xml  
   
5. DB: N/A
6. Log : /var/log/hadoop/mapreduce
7. Run: /var/run/hadoop/mapreduce
8. Start:

**mr-jobhistory-daemon.sh start historyserver**

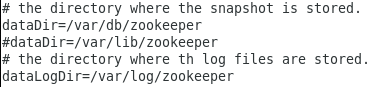
1. Test Wordcount application  
   \* Ensure **input file** below is in hdfs, **last argument** is the output directory of the wordcount mapreduce job

* **cd $HADOOP\_MAPRED\_HOME**
* **hadoop jar ./hadoop-mapreduce-examples.jar wordcount /webpages/index.html /webpages/out**
* **hdfs dfs –ls /webpages/out**
* **hdfs dfs –cat /webpages/out/part-r-XXXXX**

1. ZOOKEEPER (Already installed in 3.1)
2. Update .hadooprc  
   
3. Home Dir:



1. Hdeco:  
   
2. conf:

/etc/zookeeper/conf/zoo.cfg  


/etc/zookeeper/conf/zookeeper-env.sh  


1. DB: /var/db/zookeeper
2. Log: /var/log/zookeeper
3. Run: /var/run/zookeeper
4. Start:

**zkServer.sh start**

1. Login and check:

**zkCli.sh -server hdcentos:2181**

**ls /**

**quit**

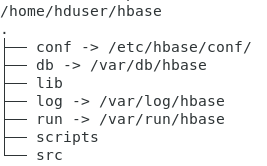
1. Stop:

**zkServer.sh stop**

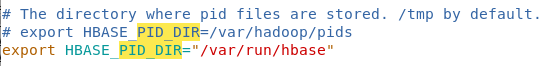
1. HBASE (Already installed in 3.1)
2. Update .hadooprc



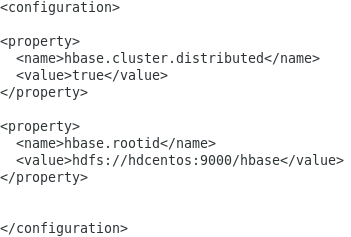
1. Home Dir:



1. Hdeco:  
   
2. Conf:

/etc/hbase/conf/hbase-env.sh  




/etc/hbase-site.xml  


/etc/hbase/regionservers



1. DB: /var/db/hbase
2. Log: /var/log/hbase
3. Run: /var/run/hbase
4. Start:

**hbase-daemon.sh start master**

**hbase-daemon.sh start regionserver**

1. Stop:

**hbase-daemon.sh stop regionserver**

**hbase-daemon.sh stop master**

1. WebUI:

**hdcentos:16010/master-status**

1. Hbase shell and commands:

**hbase shell**

**status ‘detailed’**

**version**

**whoami**

**help ‘create’**

**help ‘get’**

**list**

# Create: create <table\_name>, <column\_family1>, <column\_family2>

**create ‘customers’, ‘Name’, ‘Contact’**

**list**

# Put: put <table\_name>, <row\_id>, <column\_family:column>, <value>

**put ‘customers’, ‘1’, ‘Contact:FirstName’, ‘Joe’**

**put ‘customers’, ‘1’, ‘Contact:Email’, ‘joe@google.com’**

**put ‘customers’ ,’2’, ‘Contact:FirstName’, ‘Sarah’**

**put ‘customers’, ‘2’, ‘Contact:Email’, ‘sarah@yahoo.com’**

# Display table data

**scan ‘customers’**

**scan ‘customers’, {‘LIMIT’ => 1}**

# Update row: re-run put on row data

# Check if table exist: exists <table\_name>

**exists ‘customers’**

# Disable table: disable <table\_name>

**disable ‘customers’**

**list**

**scan**

# Re-enable table:

**enable ‘customers’**

# Describe table

**describe ‘customers’**

# Change permission

**alter ‘customers’, READONLY**

# Delete column family

**alter ‘customers’, ‘delete’ => ‘Contact’**

# Drop table: disable + drop

**disable ‘table\_name’**

**drop ‘table\_name’**

**exists ‘table\_name’**

# Select specific columns only: **scan <table\_name>, <parameters>**

**scan ‘customer’, {COLUMNS => [‘Contact:FirstName’]}**

**scan ‘customer’, {COLUMNS => [‘Contact:FirstName’], LIMIT => 1}**

# Delete column value

**delete ‘customers’**, ‘2’, ‘Name:FirstName’

# Delete entire row

**deleteall ‘customers’, ‘1’**

# Count rows

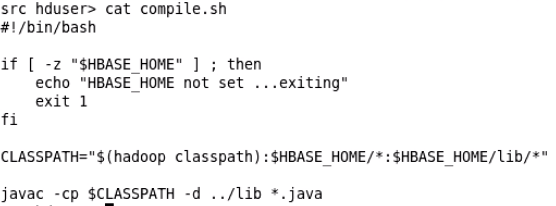
**count ‘customers’**

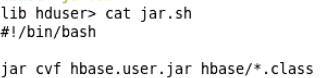
# Delete all table entries with truncate

**truncate ‘customers’**

1. Execute a hbase shell script: hbase shell <script\_name>

**hbase shell ./customers.hb**

1. Execute hbase java file **(incomplete)** :
   1. Compile java file  
      
   2. Run the **jar** file



1. MYSQL Installation

Note: Centos7 may have **MariaDB** installed by default. You may use MariaDB instead of MySQL. Optionally, MariaDB can be replaced by installing MySQL using the following instructions.

1. Remove MariaDB:  
   <https://stackoverflow.com/questions/33362904/completely-remove-mariadb-or-mysql-from-centos-7-or-rhel-7>
2. Install MySQL:

<https://www.linode.com/docs/databases/mysql/how-to-install-mysql-on-centos-7/>

1. Installation Issues Troubleshooting:
2. All errors in general:

* Confirm correct **Java** version:  
  **java –version  
  echo $JAVA\_HOME**
* Confirm correct **hadoop** version:  
  **hadoop version**
* Always check the log files first in **/var/log/hadoop, /var/log/zookeeper,** etc.
* Use **grep ERR** or **grep FATAL** on log directories
* Check for correct configuration in .hadooprc
* Check correct spelling and value in **/etc/hadoop/conf, /etc/zookeeper,** etc.configuration files
* Command **source .hadooprc** is ran after each update
* Check **jps** if dependent daemons are running

1. HDFS

* Check **/var/log/hadoop/hdfs**, hdfs-site.xml, **jps**

1. Yarn Class Path error or Resource / Node manager not running:

* Confirm correct yarn home: **echo $HADOOP\_YARN\_HOME**
* Check **yarn.application.classpath** property in **/etc/hadoop/conf/**