12: Docker Tutorial: Hadoop Big Data configuration files on Cloudera quickstart

Posted on May 28, 2019

The “etc” (i.e etcetera) folder is mainly for configuration files. It’s purpose is to host various **configuration files**. For instance, to add a new hard drive to your system and have Linux auto-mount it on boot, you’d have to edit /etc/fstab. Key “Hadoop Cluster Configuration” files are:

```
[root@quickstart /]# ls -ltr /etc/hadoop/conf/
total 40
-rwxr-xr-x 1 root root  2375 Feb 23  2016 yarn-si
-rwxr-xr-x 1 root root  1104 Feb 23  2016 README
```

300+ Java Interview FAQs

300+ Java FAQs

16+ Java Key Areas Q&As

150+ Java Architect FAQs

80+ Java Code Quality Q&As

150+ Java Coding Q&As

300+ Big Data Interview FAQs

300+ Big Data FAQs

Tutorials - Big Data

TUT - Starting Big Data

TUT - Starting Spark & Scala

```
-rwxr-xr-x 1 root root  2890 Feb 23  2016 hadoop-r
-rwxr-xr-x 1 root root  1366 Feb 23  2016 hadoop-e
-rwxr-xr-x 1 root root 11291 Mar 23  2016 log4j.pr
-rw-rw-r-- 1 root root  1546 Apr  5  2016 mapred-s
-rw-rw-r-- 1 root root  3739 Apr  5  2016 hdfs-sit
-rw-rw-r-- 1 root root  1915 Apr  5  2016 core-sit

```

hadoop-env.sh

This file specifies environment variables that affect the JDK used by Hadoop Daemon (i.e. /usr/bin/hadoop). As Hadoop framework is written in Java and uses Java Runtime environment, one of the important environment variables for Hadoop daemon is $JAVA_HOME.

core-site.xml

This file tells where NameNode runs in the cluster. It contains the configuration settings for Hadoop Core. The commonly used port is **8020** and you can also specify IP address rather than hostname.

```
[root@quickstart /]# cat /etc/hadoop/conf/core-sit
.....
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://quickstart.cloudera:8020</value>
  </property>
.....
</configuration>

```

hdfs-site.xml

This file contains the configuration settings for HDFS daemons.

800+ Java Interview Q&As

300+ Core Java Q&As
300+ Enterprise Java Q&As
150+ Java Frameworks Q&As
120+ Companion Tech Q&As
Tutorials - Enterprise Java

```
[root@quickstart /]# cat /etc/hadoop/conf/hdfs-si
.....
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  .....
  <property>
     <name>dfs.permissions</name>
     <value>false</value>
  </property>
  .....
</configuration>

```

yarn-site.xml

Configurations for **ResourceManager** and **NodeManager**.

mapred-site.xml

```
[root@quickstart /]# cat /etc/hadoop/conf/mapred-s

<configuration>
  ...
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>0.0.0.0:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</nar
    <value>0.0.0.0:19888</value>
  </property>
  .....
</configuration>

```

slaves

The ‘slaves’ file at Master node contains a list of hosts, one per line, that are to host Data Node and Task Tracker servers. The “slaves” file at Slave node contains only its own IP address and not of any other

Data Nodes in the cluster. In Hadoop 3.0 all worker hostnames or IP addresses will be in /etc/hadoop/**workers** file.

Masters

This file informs about the Secondary Namenode location to hadoop daemon. The 'masters' file at Master server contains a hostname Secondary Name Node servers. In Hadoop 3.0 you will have one active name node & more than one passive name nodes.

/etc/init.d services

init.d is the sub-directory of **/etc** directory in Linux file system. init.d basically contains the bunch of start/stop/reload/restart/status scripts which are used to control the Hadoop ecosystem daemons whilst the system is running or during boot. If you look at /etc/init.d then you will notice all the scripts for different services

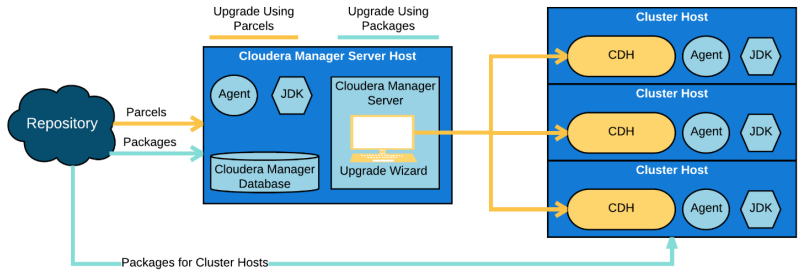
```
[root@quickstart /]# cd /etc/init.d
[root@quickstart init.d]# ls
atd                           hadoop-httpfs
cloudera-quickstart-init      hadoop-mapreduce-ht
cloudera-scm-agent            hadoop-yarn-nodemar
cloudera-scm-server           hadoop-yarn-proxyse
crond                         hadoop-yarn-resourc
flume-ng-agent                halt
functions                     hbase-master
hadoop-hdfs-datanode          hbase-regionserver
hadoop-hdfs-journalnode       hbase-rest
hadoop-hdfs-namenode          hbase-solr-indexer
hadoop-hdfs-secondarynamenode hbase-thrift
[root@quickstart init.d]# service hbase-master sta
HBase master daemon is running
[root@quickstart init.d]# service mysqld status
mysqld (pid  169) is running...
[root@quickstart init.d]# service impala-server st
Impala Server is running

```

Cloudera Manager Agent/Server Architecture

Cloudera Manager runs a central server, which is aka “SCM Server” that hosts the UI Web Server and the application logic for managing CDH. Everything related to installing CDH, configuring services, and starting and stopping services is managed by the Cloudera Manager Server. You can also configure all the configuration parameters listed on config files like hdfs-site.xml, core-site.xml, yarn-site.xml, etc.

The Cloudera Manager Agents are installed on every managed host. They are responsible for starting and stopping Linux processes, unpacking configurations, triggering various installation paths, and monitoring the host.

The Cloudera Manager Server is the master service that manages the data model of the entire cluster in a database. The data model contains information regarding the services, roles, and configurations assigned for each node in the cluster. You can also upgrade the services via parcels & packages.



Cloudera Manager – Server & Agents

CDH – stands for Cloudera Distribution Hadoop. CDH upgrades contain updated versions of the Hadoop software and other components. You can use

Cloudera Manager to upgrade CDH for major, minor, and maintenance upgrades.

Cloudera Manager Client Vs. Server configurations

Novice Cloudera Manager administrators are often surprised that modifying **/etc/hadoop/conf** and then restarting HDFS has no effect. This is because service instances started by Cloudera Manager do not read configuration files from the default locations. Cloudera Manager distinguishes between server and client configuration. In the case of HDFS, the file /etc/hadoop/conf/hdfs-site.xml contains only configuration relevant to an HDFS client.

Let's start the Cloudera Manager on Docker as described in Docker Tutorial: Cloudera BigData on Docker via DockerHub

```
1 [root@quickstart /]# cd /home/cloudera/
2 [root@quickstart cloudera]# ./cloudera-manager --e
3
```

http://localhost:7180/ to access the Cloudera manager, where services can be not only stopped and started, but also configured.

Cloudera Manager obtain their configurations from a private per-process directory, under **/var/run/cloudera-scm-agent/process/unique-process-name**. Giving each process its own private execution and configuration environment allows Cloudera Manager to control each process independently. Here is an example:

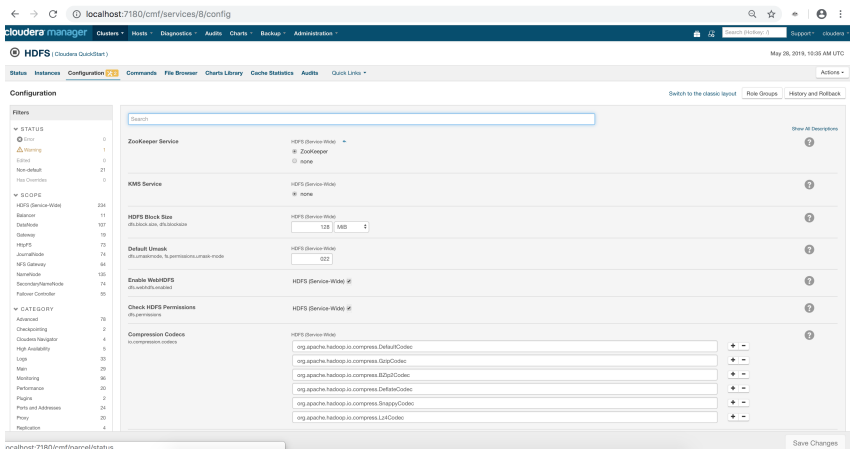
```
[root@quickstart cloudera]# ls -ltr /var/run/cloud
total 0
drwxr-x--x 3 zookeeper    zookeeper    260 May 28
drwxr-x--x 5 sqoop2       sqoop        260 May 28
drwxr-xr-x 4 root         root         100 May 28
drwxr-xr-x 7 root         root         200 May 28
drwxr-xr-x 4 root         root         100 May 28
drwxr-xr-x 6 root         root         140 May 28
drwxr-xr-x 4 root         root         100 May 28
drwxr-xr-x 4 root         root         100 May 28
drwxr-xr-x 4 root         root         100 May 28
drwxr-x--x 3 cloudera-scm cloudera-scm 240 May 28
drwxr-x--x 3 cloudera-scm cloudera-scm 300 May 28
drwxr-x--x 3 cloudera-scm cloudera-scm 220 May 28
drwxr-x--x 3 cloudera-scm cloudera-scm 320 May 28
drwxr-x--x 3 cloudera-scm cloudera-scm 260 May 28
drwxr-x--x 3 cloudera-scm cloudera-scm 340 May 28
drwxr-x--x 3 cloudera-scm cloudera-scm 320 May 28

```

```
[root@quickstart cloudera]# ls -ltr /var/run/cloud
total 36
-rw-r----- 1 root root 1510 May 28 10:17 topology
-rw-r----- 1 root root  201 May 28 10:17 topology
-rw-r----- 1 root root  315 May 28 10:17 ssl-clie
-rw-r----- 1 root root  314 May 28 10:17 log4j.pr
-rw-r----- 1 root root 1772 May 28 10:17 hdfs-site
-rw-r----- 1 root root 2696 May 28 10:17 hadoop-e
-rw-r----- 1 root root 3549 May 28 10:17 core-site
-rw-r--r-- 1 root root   26 May 28 10:17 __cloude
-rw-r--r-- 1 root root   21 May 28 10:17 __cloude

```

http://localhost:7180/

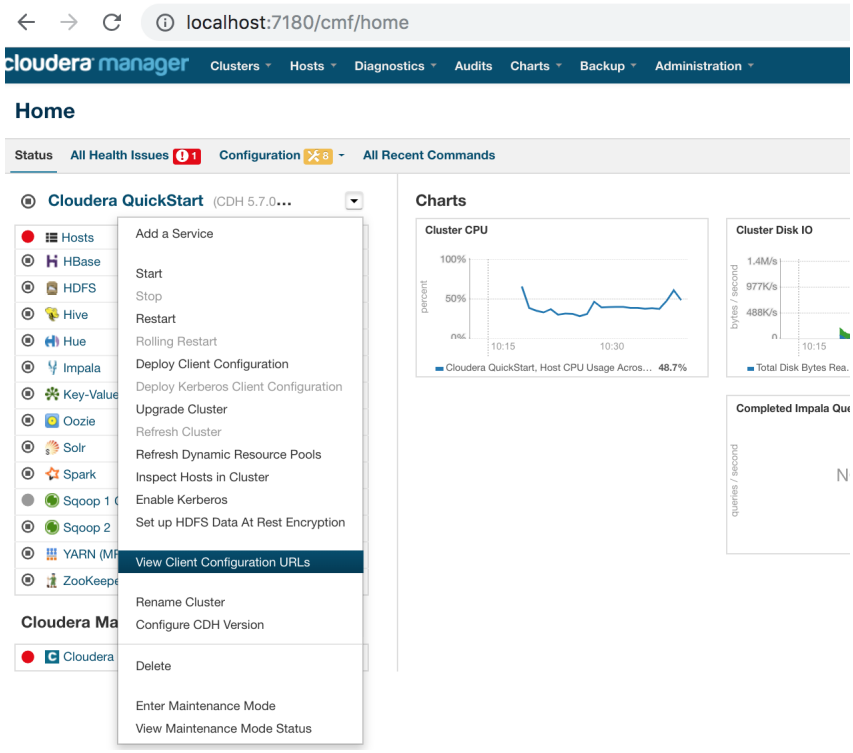
Login with cloudera/cloudera.

Click on **HDFS** –> **Configuration**, where you can modify the server configuration values.



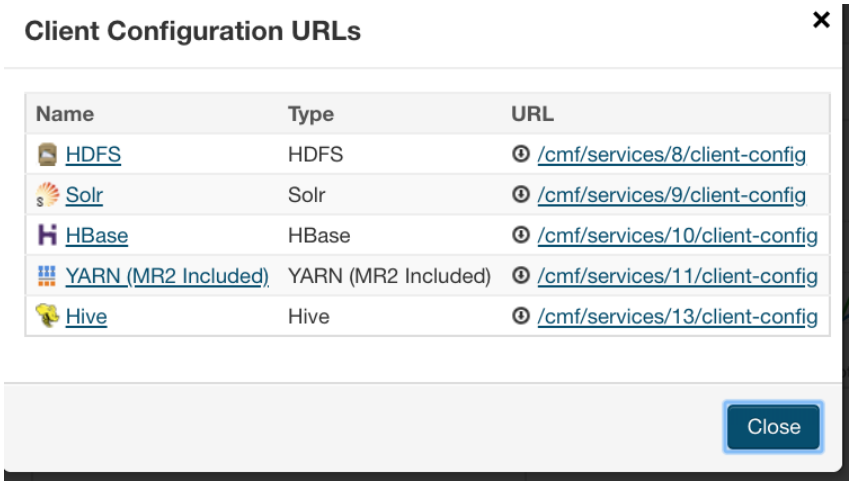
Cloudera Manager – HDFS Configurations

Cloudera Manager Client Configurations



Cloudera Manager – Client Configurations

Cloudera Manager Client Configurations – download URLs



Cloudera Manager – Client Configuration download URLs

‹ 11: Docker Tutorial: Hadoop Big Data CLIs on Cloudera quickstart

13: Docker Tutorial: Apache Spark (spark-shell & pyspark) on Cloudera quickstart ›

Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty ltd. The EmpoweringTech pty ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. Privacy Policy

© 2022 java-success.com