# Installing Cloudera Hadoop on UCB W205 Base Image

There are 2 approaches to setting up and installing Cloudera’s Distribution including Apache Hadoop using the AWS AMI for W205: manual and via the Cloudera Manager GUI application. This document provides instructions for the manual method.

When launching your AWS image, add additional open ports before starting the instance. Specifically, open 50070, 8080, 8088, and 4040.

When you’ve completed the steps in this document, you may want to make a new AMI for your personal use! (You can do this from the EC2 web console.)

## Recommended, Make a Working User

It is ALWAYS a bad idea to run as root. You need to make a user for yourself and using that for day to day work on the AMI. Once you’ve set up a user you like, you can always make a new, personal AMI to save your configurations.

## Make a user for yourself

useradd <username>

## Setting Up HDFS

As root:

yum install hadoop-conf-pseudo

As root, make a Hadoop user:

useradd –g hadoop hadoop

passwd hadoop

Add your personal user to the Hadoop group

usermod –G hadoop <user>

As root, setup password-less ssh

su - hadoop

ssh-keygen -t rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

chmod 0600 ~/.ssh/authorized\_keys

As root, mount your extra storage:

1. Find the extra storage
   1. Run df –h and notice that only a 10GB partition is mounted
   2. Find your available EBS storage or Ephemeral Storage
      1. Run fdisk –l
      2. Locate the device that looks like /dev/<something>**b**
2. Mount that storage as /data
   1. mkdir /data
   2. mkfs.ext3 /dev/<something>
   3. mount –t ext3 /dev/<something> /data
   4. chown hadoop:hadoop /data
   5. chmod -R g+rwx /data
3. As root, make a directory for your personal user on /data
   1. mkdir /data/<user>
   2. chown <user>:hadoop /data/<user>

As root edit the HDFS and Yarn config files:

1. cd /etc/Hadoop/conf
   1. edit hdfs-site.xml
      1. Edit the following properties to match the following:

<property>

<name>hadoop.tmp.dir</name>

<value>/data/hadoop-hdfs/cache/${user.name}</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///data/hadoop-hdfs/cache/${user.name}/dfs/name</value>

</property>

<property>

<name>dfs.namenode.checkpoint.dir</name>

<value>file:///data/hadoop-hdfs/cache/${user.name}/dfs/namesecondary</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///data/hadoop-hdfs/cache/${user.name}/dfs/data</value>

</property>

* 1. Edit yarn-site.xml
     1. Edit properties to match the following

<property>

<description>List of directories to store localized files in.</description>

<name>yarn.nodemanager.local-dirs</name>

<value>/data/hadoop-yarn/cache/${user.name}/nm-local-dir</value>

</property>

<property>

<description>Where to store container logs.</description>

<name>yarn.nodemanager.log-dirs</name>

<value>/data/hadoop-yarn/containers</value>

</property>

<property>

<description>Where to aggregate logs to.</description>

<name>yarn.nodemanager.remote-app-log-dir</name>

<value>/data/hadoop-yarn/apps</value>

</property>

As the “hdfs” user, format the HDFS NameNode:

sudo -u hdfs hdfs namenode -format

Start HDFS (this is a bash for loop which starts every service named hadoop-hdfs-\*)

for x in `cd /etc/init.d ; ls hadoop-hdfs-\*` ; do sudo service $x restart ; done

As root, setup the HDFS directory structure:

/usr/lib/hadoop/libexec/init-hdfs.sh

As hdfs, make sure HDFS looks ok:

sudo -u hdfs hadoop fs -ls -R /

As hdfs, set up space for your personal user:

sudo -u hdfs hdfs dfs –mkdir /user/<user>

sudo –u hdfs hdfs dfs –chown <user> /user/<user>

## Starting YARN

Start YARN, as root

service hadoop-yarn-resourcemanager start

service hadoop-yarn-nodemanager start

service hadoop-mapreduce-historyserver start

Test that hadoop is working. As your named user:

hdfs dfs –mkdir test\_input

hdfs dfs –put /etc/hadoop/conf/\*.xml test\_input

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar grep test\_input test\_output 'dfs[a-z.]+'

hdfs dfs –ls test\_output

## Set up Hive

As root, yum install hive

As root, edit /etc/hive/conf/hive-site.xml

Set the following property:

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:derby:;databaseName=/data/${user.name}/hive/metastore/metastore\_db;create=true</value>

<description>JDBC connect string for a JDBC metastore</description>

</property>

As your personal user, start Hive

su <user>

hive

## Set up Spark 1.5

Open <https://spark.apache.org/downloads.html> in your browser

Select a release as follows:

Spark 1.5.0

Pre-built for Hadoop 2.6 or later

Direct download

Copy the URL to download spark

As your personal user,

wget <url for spark>

tar xvzf spark-1.5.0-bin-hadoop2.6.tgz

mv spark-1.5.0-bin-hadoop2.6 spark15

export SPARK\_HOME=$HOME/spark15

export HADOOP\_CONF\_DIR=/etc/hadoop/conf

You can start pyspark as follows:

$SPARK\_HOME/bin/pyspark --master yarn