

Lecture 05

Creating Virtual Machine and Hadoop Installation

Zoran B. Djordjević

@Zoran B. Djordjevic

1

Objectives

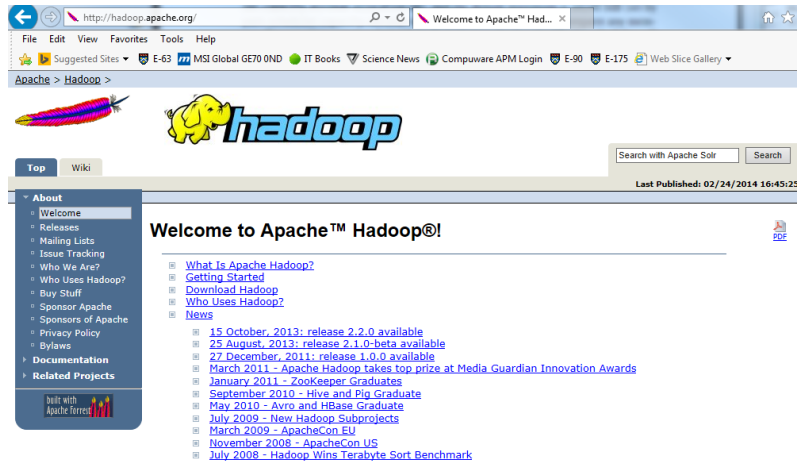
- Install recent release of Hadoop on a recent release of a popular Linux OS.
- When selecting the version of Linux on which to work you have a few things to consider:
 - \$30 or \$60 for the annual academic license for a Red Hat is not bad. Advantage is that major manufacturers, like IBM, like Red Hat.
 - Another option is CentOS OS. CentOS is free. CentOS follows Red Hat in features and API-s. Cloudera is publishing its downloadable VMs on CentOS.
 - Yet another option is Ubuntu. It is free.
 - Another free version is Virtual Box and Oracle VM.
 - Fedora is open source, free version of Red Hat. Fedora is like an experimental sandbox and its code is ahead of Red Hat's API-s.
 - SUSE is not free but is moderately popular.
 - There are a few other versions of Linux.

@Zoran B. Djordjevic

2

Hadoop

- Hadoop could be fetched from `hadoop.apache.org`



What Is Apache Hadoop?

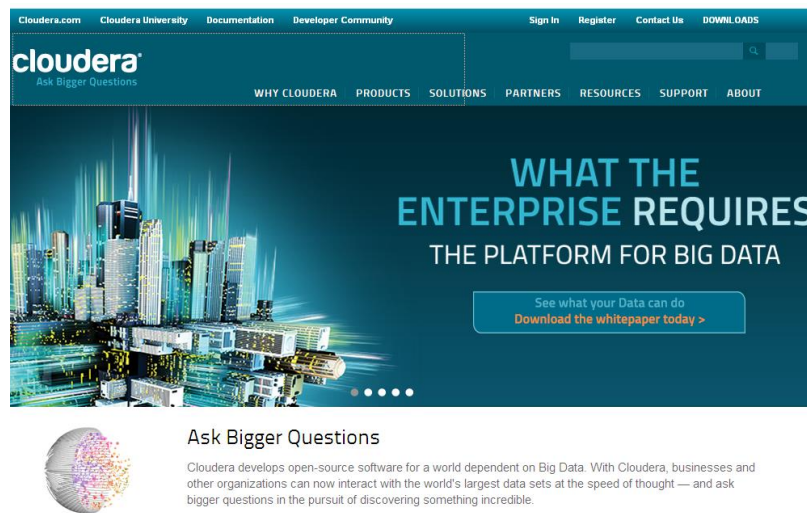
The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

@Zoran B. Djordjevic

3

Cloudera

- People who invented Hadoop are mostly at Cloudera.
- If we want the latest and the best, we go there, presumably



@Zoran B. Djordjevic

4

Cloudera CDH


- Cloudera offers CDH (Cloudera Distribution Apache Hadoop).
 - The latest non-beta version appears to be CDH 4.6
- <https://ccp.cloudera.com/display/SUPPORT/CDH+Downloads>
- We read documentation first. We always do.
 - If you have a 32 bit machine, you better do.
 - Many of Cloudera tools and packages are available for 64 bit machines only.


Version: CDH 4.6 ▼

CDH 4.6


Last update: 27 Feb 2014

Installation Options

 [Linux Packages](#)

 [Quick Start VM](#)

Notice
These are 64-bit VMs, and require a 64-bit host OS and a virtualization product that can support a 64-bit guest OS.

 [Tarballs](#)

[Release Notes](#)
[CDH4 Documentation](#)

@Zoran B. Djordjevic

5

CDH Requirements and Supported Versions

<https://ccp.cloudera.com/display/CDH4DOC/CDH4+Documentation>

Operating Systems

CDH4 provides packages for Red-Hat-compatible, SLES, Ubuntu, and Debian systems as described below.

Operating System	Version	Packages
Red Hat compatible		
Red Hat Enterprise Linux (RHEL)	5.7	64-bit
	6.2	64-bit, 32-bit
CentOS	5.7	64-bit
	6.2	64-bit, 32-bit
Oracle Linux with Unbreakable Enterprise Kernel	5.6	64-bit
SLES		
SLES Linux Enterprise Server (SLES)	11 with Service Pack 1 or later	64-bit
Ubuntu/Debian		
Ubuntu	Lucid (10.04) - Long-Term Support (LTS)	64-bit
	Precise (12.04) - Long-Term Support (LTS)	64-bit
Debian	Squeeze (6.03)	64-bit

Notes

- For production environments, 64-bit packages are recommended. Except as noted above, CDH4 provides **only 64-bit packages**.

@Zoran B. Djordjevic

6

32 vs 64 bit

- My laptop is 32 bit and, I suppose, many of yours.
- It would only be fair to run an installation with a 32 bit Hadoop on a 32 bit OS.
- The only choice is a Red Hat 6.2 or CentOS 6.2.
- If you have a new machine, it is most probably 64 bit and you are free to work with any OS that supports CHD4.6.
- Fedora is a free version of OS that is ahead of Red Hat in releases and serves as a development platform for what will eventually be packaged as Red Hat.
- CentOS 6.2 is a “repackaged” Red Hat 6.2. Let us go with it.

@Zoran B. Djordjevic

7

CDH 4.6 Packaging and Tarball Information

- Each CDH release series is made up of a collection of CDH project packages (tarballs) that are known to work together

Component	Package Version
DataFu	pig-udf-datafu-0.0.4+11
Apache Flume	flume-ng-1.4.0+96
Apache Hadoop	hadoop-2.0.0+1554
Apache HBase	hbase-0.94.15+86
Apache HCatalog	hcatalog-0.5.0+13
Apache Hive	hive-0.10.0+237
Hue	hue-2.5.0+217
Apache Mahout	mahout-0.7+15
Apache Oozie	oozie-3.3.2+100
Parquet	parquet-1.2.5+7
Apache Pig	pig-0.11.0+42
Apache Sentry (incubating)	sentry-1.1.0+20
Apache Sqoop	sqoop-1.4.3+92
Apache Sqoop2	sqoop2-1.99.2+99
Apache Whirr	whirr-0.8.2+15
Apache ZooKeeper	zookeeper-3.4.6+25

@Zoran B. Djordjevic

8

CDH 5.0 Beta 2 Packaging and Tarball Information

- CDH5.0 has a richer set of features and tarballs.
- Most of our work will rely on CDH4.6, however, we might use CDH5 for some analysis.
- It appears that CDH5 is currently available only in 64 bit version.

Component	Package Version
Apache Avro	avro-1.7.5+cdh5.0.0b2+8
Apache Crunch	crunch-0.9.0+cdh5.0.0b2+19
DataFu	pig-udf-datafu-1.1.0+cdh5.0.0b2+8
Apache Flume	flume-ng-1.4.0+cdh5.0.0b2+90
Apache Hadoop	hadoop-2.2.0+cdh5.0.0b2+1610
Apache HBase	hbase-0.96.1.1+cdh5.0.0b2+23
HBase-Solr	hbase-solr-1.3+cdh5.0.0b2+39
Apache Hive	hive-0.12.0+cdh5.0.0b2+265
Hue	hue-3.5.0+cdh5.0.0b2+186
Cloudera Impala	impala-1.2.3+cdh5.0.0b2+0
Kite SDK	kite-0.10.0+cdh5.0.0b2+69
Llama	llama-1.0.0+cdh5.0.0b2+0
Apache Mahout	mahout-0.8+cdh5.0.0b2+28
Apache Oozie	oozie-4.0.0+cdh5.0.0b2+144
Parquet	parquet-1.2.5+cdh5.0.0b2+29
Parquet-format	parquet-format-1.0.0+cdh5.0.0b2+4
Apache Pig	pig-0.12.0+cdh5.0.0b2+20
Cloudera Search	search-1.0.0+cdh5.0.0b2+0
Apache Sentry (incubating)	sentry-1.2.0+cdh5.0.0b2+64
Apache Solr	solr-4.4.0+cdh5.0.0b2+165
Apache Spark	spark-0.9.0+cdh5.0.0b2+6
Apache Sqoop	sqoop-1.4.4+cdh5.0.0b2+40
Apache Sqoop2	sqoop2-1.99.3+cdh5.0.0b2+19
Apache Whirr	whirr-0.8.2+cdh5.0.0b2+20
Apache ZooKeeper	zookeeper-3.4.6+cdh5.0.0b2+27

Select a mirror near you

- Please go to <http://wiki.centos.org/Download>. Select version 6.5.
- If you are installing a 32 bit OS, choose Cd and DVD ISO images `i386`. If you have a 64 bit laptop and are installing a 64 bit OS, as you should go to `x86_64`. Either link will bring you to list of mirrors. Choose your mirror.
- There, please read the `README.txt`. It will tell you that you do not know what you are doing and that you should go somewhere else.
- Do as you are told.
- Select
 - `CentOS-6.5-i386-bin-DVD1.iso` and then
 - `CentOS-6.5-i386-bin-DVD2.iso` for 32 bit OS or
 - `CentOS-6.5-x86_64-bin-DVD1.iso` and then
 - `CentOS-6.5-x86_64-bin-DVD2.iso` for 64 bit OS.
- The second DVD is not used, at least not for the initial installation.
- With some downloads I got an impression that I have to burn the DVD and that CentOS would not install from an `iso` image. That might not be true.
- VMWare Workstation would install most Linux OS-s from an `iso` image.

@Zoran B. Djordjevic

10

6.5 i386 folder

Which images are in this directory

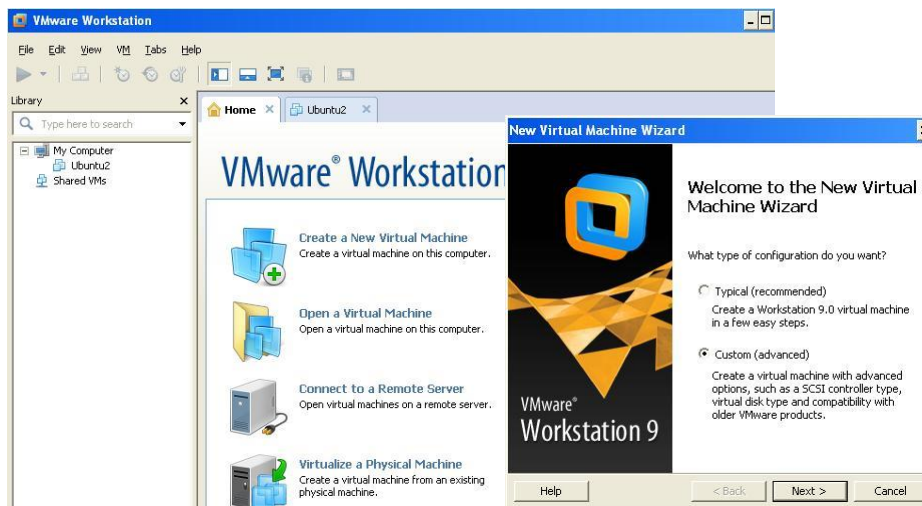
- [CentOS-6.5-i386-netinstall.iso](#) This is the network install and rescue image. This image is designed to be burned onto a CD. You then boot your computer off the CD CentOS-6.5-i386-minimal.iso The aim of this image is to install a very basic CentOS 6.3 system, with the minimum of packages needed to have a functional system. Please burn this image onto a CD and boot your computer off it. A preselected set of packages will be installed on your system. Everything else needs to be installed using yum. Please read <http://wiki.centos.org/Manuals/ReleaseNotes/CentOSMinimalCD6.3> for more details about this image. Beware that the set of packages installed by this image is NOT identical to the one installed when choosing the group named "Minimal" from the full DVD image.
- [CentOS-6.5-i386-bin-DVD1.iso](#) [CentOS-6.5-i386-bin-DVD2.iso](#) These two dvd images contain the entire base distribution. Please burn DVD1 onto a DVD and boot your computer off it. A basic install will not need DVD2. After the installation is complete, please run "yum update" in order to update your system.
- [CentOS-6.5-i386-LiveCD.iso](#) This is a CD live image of CentOS 6.3 designed to be burned onto a CD. You then boot your computer using that CD. Please read <http://wiki.centos.org/Manuals/ReleaseNotes/CentOSLiveCD6.3> for more details about this image. The disk can also be used to install CentOS 6.3 onto your computer.
- [CentOS-6.5-i386-LiveDVD.iso](#) This is a DVD live image of CentOS 6.3 designed to be burned onto a DVD. You then boot your computer using that DVD. Please read <http://wiki.centos.org/Manuals/ReleaseNotes/CentOSLiveDVD6.3> for more details about this image. The disk can also be used to install CentOS 6.3 onto your computer. Remember that in order to be able to partition your disk you will need to run the GUI installer which in turns needs enough RAM. The same is true for the network setup step. (<http://wiki.centos.org/Manuals/ReleaseNotes/CentOS6.3>) provide more details about these aspects.

@Zoran B. Djordjevic

11

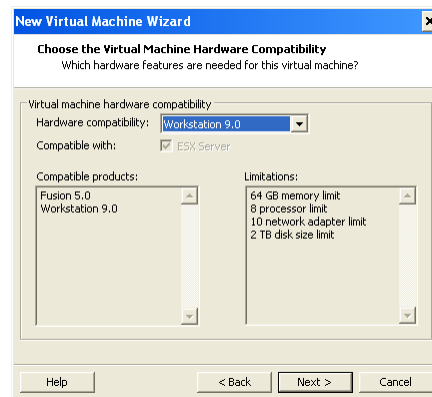
Start VMware Workstation

- Select "Create a New Virtual Machine".
- Accept Custom, Next >



If on 64 bit PC Choose Workstation 10 or 9

- You could have also gone to File > “New Virtual Machine”.
- If on Mac, select Fusion 6 rather than Workstation 10.
- If on 32 bit PC, select VMware Workstation 7.
- If on 32 bit Mac, select VMware Fusion 3.0.

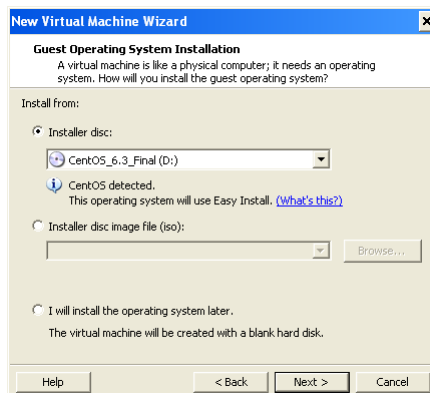


@Zoran B. Djordjevic

13

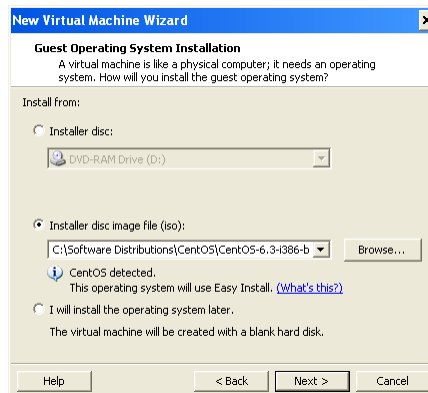
Select the Distribution, Create User

- Accept selected Installer disk and click Next >
- If VMWare Workstation accepts to read your `iso` file, it is faster and safer to use that `iso` file, as depicted on in the image on the right, than to burn the CD first



@Zoran B. Djordjevic

14



Create a Linux User

- Name your Linux instance and create new user.
- We will create user `hadoop`. If you do it here, user `hadoop` will belong to the Linux group `hadoop`. Typically, system administrators would create users and groups later on the command line.
- Please, note that you are creating the initial password for user `root`, as well.
- In normal circumstances, you will change those passwords as soon as you open the system.
- If you do not plan to use this VM too often, stick with `hadoop/hadoop` credentials.

@Zoran B.

Name the VM, Select Directory

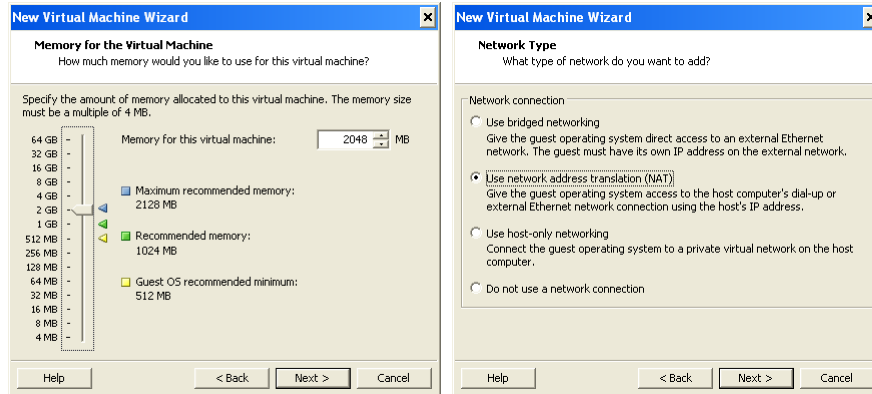
- You can place your VM anywhere, including a thumb drive or a USB external drive.
- Select the number of processors your machine has.

@Zoran B. Djordjevic

16

Select Memory, Network Type

- Assign memory to your VM. More memory, faster performance. You have to leave some for underlying OS.



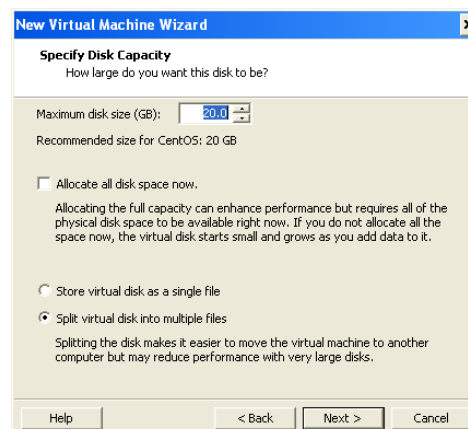
- Accept default under "I/O Controller Types"
- Accept "Create a new virtual disk" under "Select a Disk"
- Accept "SCSI" under "Select a Disk Type"

@Zoran B. Djordjevic

17

Important, Split disk into multiple files

- Select the size of your disk. You can add disk space later.
- Most importantly, make sure that you selected "Split virtual disk into multiple files"
- If you store disk as a single file, you will not be able to copy it around (easily).
- Next>

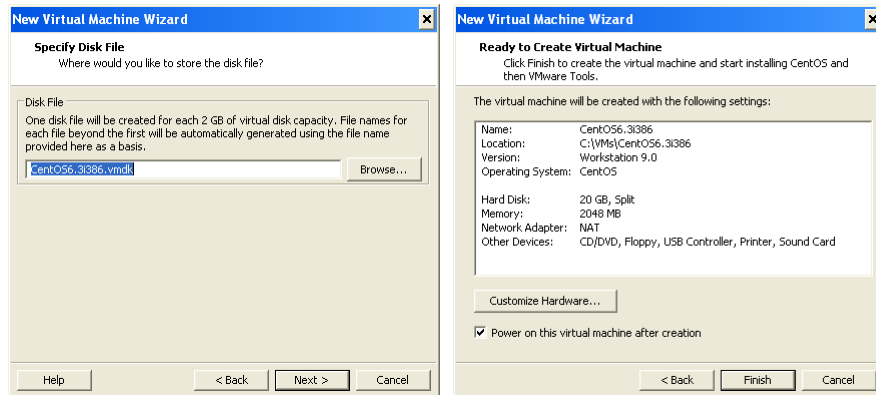


@Zoran B. Djordjevic

18

Accept Disk File Name

- Disk file name is not particularly important. It might be convenient to leave the default. Click Finish

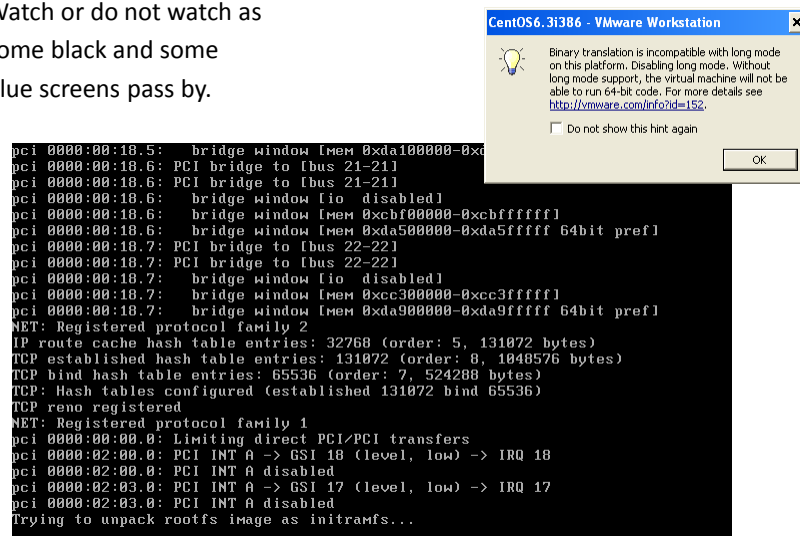


@Zoran B. Djordjevic

19

Incompatible Binary Translation

- Click OK if you get binary translation incompatibility warning.
- Watch or do not watch as some black and some blue screens pass by.

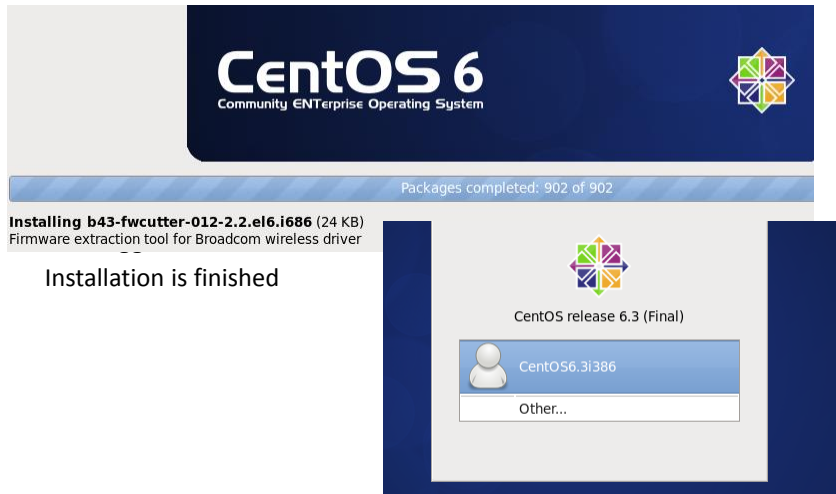


@Zoran B. Djordjevic

20

Eventually all packages are installed

- On a fast machine it might take 30 minutes. On a slow machine it might take 2 hours.

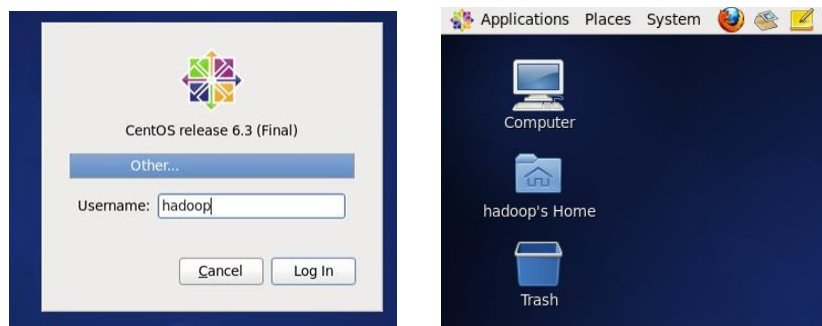


@Zoran B. Djordjevic

21

Login into the Linux VM

- If you remember hadoop's password, you can log into your Linux box:

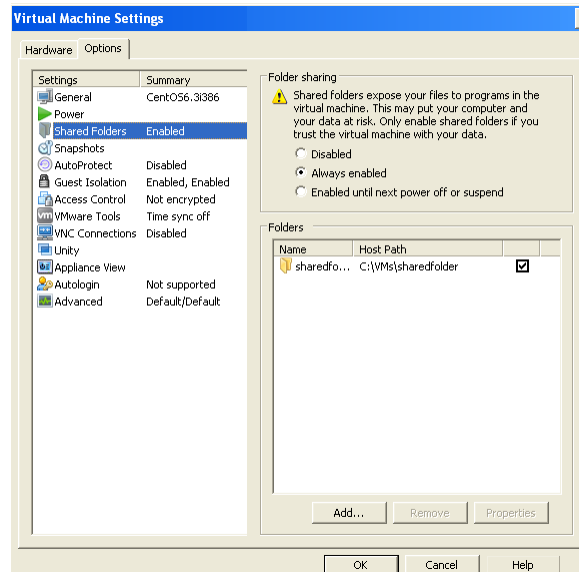


@Zoran B. Djordjevic

22

Enable Shared Folders

- In order to be able to share files with the host OS, we need to enable Shared Folders.
- Power down VM. Right click on the VM, select Edit virtual machine settings > Options
- Select Shared Folders > Always enables > Add
- Add folder
c:\VMs\sharedfolder
- Check Always enable
Finish, OK
- Power up VM
- Login as `hadoop`.
- Shared folder shows as
`/mnt/hgfs/sharefolder`

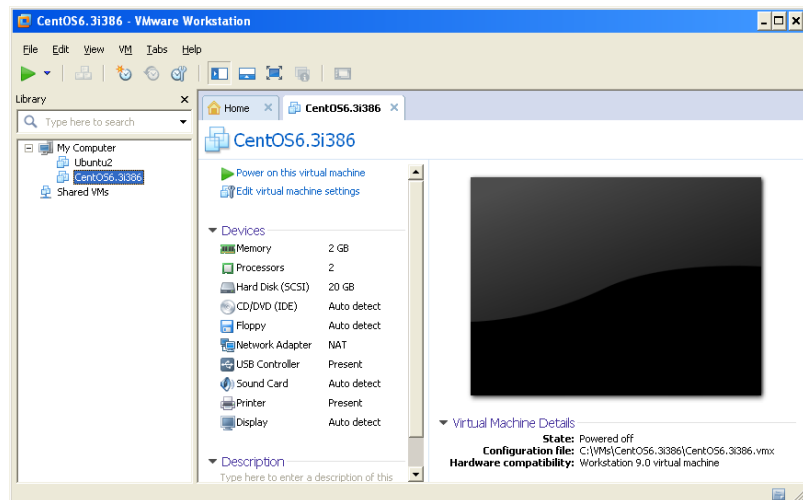


@Zoran B. Djordjevic

23

Power on the Virtual Machine

- If you are restarting the VM, you open the Workstation, select the VM you want to run and then hit the green triangle.
- If you have enough memory, you could run several VMs simultaneously.

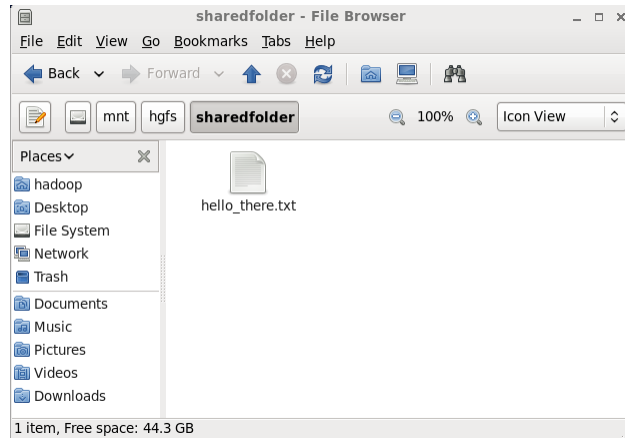
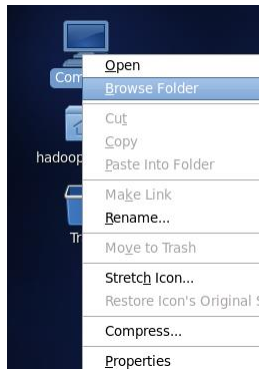


@Zoran B. Djordjevic

24

Browse Folder

- Right click on Computer
- Browse Folder > File System > mnt > hgfs > sharefolder



You can use
sharefolder

to share files back and forth between the operating system of your host machine and the operating system of your new VM

@Zoran B. Djordjevic

25

Select Terminal, whoami, Shell



- Select Applications > System Tools > Terminal.
- Find out who you are, `$ whoami`
- Examine `/etc/passwd` file
- Examine `/etc/group` file
- User `hadoop` belongs to group `hadoop` and has `bash` shell.

```
[hadoop@localhost ~]$ whoami
hadoop
[hadoop@localhost ~]$ cat /etc/passwd | grep hadoop
hadoop:x:500:500:CentOS6.3i386:/home/hadoop:/bin/bash
[hadoop@localhost ~]$ cat /etc/passwd | grep hadoop
hadoop:x:500:500:CentOS6.3i386:/home/hadoop:/bin/bash
[hadoop@localhost ~]$ pwd
/home/hadoop
```

@Zoran B. Djordjevic

26

Giving sudo privileges to user hadoop

- We need `hadoop` to be very powerful user. This is done by user `root` who grants “`sudo`” privilege to user `hadoop`.
- On the top menu, select `System` and “`Log Out hadoop`”
- On the next widget select `Switch User`.
- On the following widget select `Other (user)`.
- Enter `root` as the username.
- `root`'s password is still the same as the password of user `hadoop`.
- As user `root` open the terminal window and change permissions on file `/etc/sudoers`.
`$chmod a+w /etc/sudoers.`
- Add the following line to `/etc/sudoers`,
`root$ vi /etc/sudoers`
`hadoop ALL=NOPASSWD: ALL`
- If you do not want user `hadoop` to be asked for password after a `sudo` command, change that line to
`hadoop ALL=(ALL) NOPASSWD:ALL`

@Zoran B. Djordjevic

27

Giving sudo privileges to user hadoop

- Allowing user `hadoop` to run commands without checking its password, is a security issue but is a great convenience.
- On some Linux systems, CentOS included, `sudo` command clears the environmental variables.
- In order to preserve some of those, you need to add lines to `/etc/sudoers` that read like :
`Defaults env_keep+= "JAVA_HOME"`
- Then change permissions of `/etc/sudoers` back to read only (440 mode)
`$chmod 440 /etc/sudoers.`
`$ls -ls /etc/sudoers`
`-r-r----- . 1 root root 4035 Mar 8 06:56 /etc/sudoers`
- Once you install Java JDK, you will be able to verify that `sudo` command does not remove `JAVA_HOME` environmental variable by typing:
`$sudo env | grep JAVA_HOME`

@Zoran B. Djordjevic

28

While root, Check for Java

- Check for version of Java. Hadoop needs Java 6 or 7.

```
[root@localhost ~]# which java
/usr/bin/which: no java in
(/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin:/root
/bin)
```

- Since Java is not there, we need to install it.
- Cloudera recommends using `jdk1.7_15` or later. You can fetch it at:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
- Accept the license agreement and then select `jdk-7u51-linux-i586.rpm`, if downloading for a 32 bit VM.
- Select `jdk-7u51-linux-x64.rpm`, if downloading for a 64 bit VM.
- When you click on the download link, you will be asked to save the file.
- If, on your VM, you are the user `hadoop`, the file will go into the directory `/home/hadoop/Downloads`

@Zoran B. Djordjevic

29

Installing Java

- The name of the downloaded file for 32 bit system has the following format:
`jdk-7u51-linux-i586.rpm`
- 1. **If not root, become root** by running the `$ su` command and entering root's password.
- 2. **Install JDK by running rpm command**
`$ rpm -ivh jdk-7u51-linux-i586.rpm`
- To uninstall this package type:
`$ rpm -e jdk-7`
- On some installations, the script displays a binary license agreement, which you are asked to agree to before installation can proceed. Once you have agreed to the license, `rpm` runs the file `jdk-7u51-linux-i586.rpm` in the current directory.
- **NOTE:** `rpm` process sends `jdk-7` files to `/usr/java/jdk1.7.0_51` directory.
- Java executable is sent to `/usr/bin`, as you could see by `$which java`
- 3. **Delete the rpm file** if you want to save disk space.
- 4. **To see which packages are installed, type:**
`$rpm -qa | grep jdk` or just `$rpm -qa`

@Zoran B. Djordjevic

30

Set JAVA_HOME for user root

- If you are root, go to root's home directory (~ represents current user's home)
\$ cd ~
- As root open file /root/.bash_profile using vi editor and add JAVA_HOME.
- Before adding a line in vi, hit Esc(ape) key and then lower case i, for inserting. Your .bash_profile should at least have the following lines:

```

JAVA_HOME=/usr/java/jdk1.7.0_51
export JAVA_HOME
PATH=$PATH:/$HOME/bin:$JAVA_HOME/bin
export PATH

```
- When done editing .bash_profile, exit by typing Esc and then :wq , for write and quit.
- In order for your session to become aware of new values, or new variables, you need to source .bash_profile file.
- In the directory where .bash_profile resides (e.g. /root) type:
\$ source .bash_profile
- To verify new values, type
\$ echo \$PATH
\$ cd \$JAVA_HOME

@Zoran B. Djordjevic

31

Set JAVA_HOME for user hadoop

- If you are root, type \$ exit and verify that you are user hadoop:
\$whoami
hadoop
- As hadoop open file /home/hadoop/.bash_profile using vi editor and add JAVA_HOME.
- User hadoop's .bash_profile should at least have the following lines:

```

JAVA_HOME=/usr/java/jdk1.7.0_51
export JAVA_HOME
PATH=$PATH:/$HOME/bin:$JAVA_HOME/bin
export PATH

```
- When done editing .bash_profile, exit by typing Esc and then :wq , for write and quit.
- In order for your session to become aware of new values, or new variables, you need to source .bash_profile file.
- In the directory where .bash_profile resides (e.g. /root) type:
\$ source .bash_profile
- To verify new values, type
\$ echo \$PATH
\$ cd \$JAVA_HOME

@Zoran B. Djordjevic

32

Versions of Hadoop, MapReduce

- It appears that Apache (Cloudera) Hadoop has two major versions, older Hadoop 1 and more modern Hadoop 2.
- Hadoop 2 is more scalable and more reliable than Hadoop 1. For example in Hadoop 2 we could have more than one name node.
- The major new component or framework with Hadoop 2 is called Yarn.
- **Yarn** is a new framework that facilitates writing arbitrary distributed processing frameworks and applications, and not only MapReduce.
- YARN provides the daemons and APIs necessary to develop generic distributed applications of any kind, and not only MapReduce, handles and schedules resource requests (such as memory and CPU) from such applications, and supervises their execution.
- YARN's execution model (MRv2) is more generic than the earlier MapReduce implementation. YARN can run applications that do not follow the MapReduce model, unlike the original Apache Hadoop MapReduce (also called MR1).

@Zoran B. Djordjevic

33

MR2

What is MR2 or MRv2?

- With YARN, there is no longer a single JobTracker to run jobs and a TaskTracker to run tasks of the jobs.
- The old MR1 framework was rewritten and named MR2 or MRv2, MapReduce version 2.
- MR2 utilizes the familiar MapReduce execution underneath, except that each job now controls its own destiny via its own ApplicationMaster taking care of execution flow (such as scheduling tasks, handling speculative execution and failures, etc.).
- MR2 is a more isolated and scalable model than the MR1 system where a singular JobTracker does all the resource management, scheduling and task monitoring work.
- MR2 and a new proof-of-concept application called the DistributedShell are the first two applications using the YARN API in CDH4.

@Zoran B. Djordjevic

34

Install CDH 4.6 with Yarn (Mrv2)

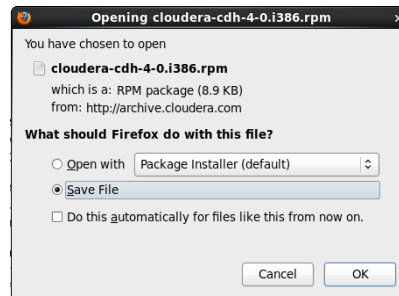
- Hadoop eco system comes in several (many) packages.
- For the basic Map Reduce we need Hadoop itself and could download it separately.
- For other tools like Hive, Pig, Hbase, etc., we need additional packages.
- If you know what you are doing, you could download the source code for all of those packages, improve on them and then deploy them.
- We will use prepackaged versions of the code which comes as the so called tar balls.
- The main product of Cloudera is the Cloudera Hadoop Distribution or CDH. Currently CDH is at version 4.6
- We will first install newer version, i.e. Hadoop with Yarn.

@Zoran B. Djordjevic

35

Installing CDH4 on a Single Pseudo-Distributed Node

- In order to download CDH4.6 package, in your VM, open Mozilla, and for 32 bit Redhat (CentOS) system, go to <http://archive.cloudera.com/cdh4/one-click-install/redhat/6/i386/cloudera-cdh-4-0.i386.rpm>
- Download file `cloudera-cdh-4-0.i386.rpm`



- File will end up in `/home/hadoop/Downloads`
- For 64 bit Redhat (CentOS) system, go to http://archive.cloudera.com/cdh4/one-click-install/redhat/6/x86_64/cloudera-cdh-4-0.x86_64.rpm

@Zoran B. Djordjevic

36

In case you need to stop Hadoop and uninstall

- If you have YARN (MRv2) already install, stop all the daemons by typing:

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

- To remove Hadoop on Red Hat-compatible systems type:

```
$ sudo yum remove hadoop-conf-pseudo hadoop-mapreduce-*
$ sudo yum remove hadoop-conf-pseudo hadoop-hdfs-*
$ sudo yum remove hadoop-conf-pseudo hadoop-yarn-*
```

- If you have MRv1 already install, stop all the daemons by typing:

```
$ for x in `cd /etc/init.d ; ls hadoop-hdfs-*`; do sudo service $x stop ; done
$ for x in `cd /etc/init.d ; ls hadoop-0.20-mapreduce-*`; do sudo service $x stop ; done
```

- To remove Hadoop on Red Hat-compatible systems type:

```
$ sudo yum remove hadoop-0.20-conf-pseudo hadoop-mapreduce-*
$ sudo yum remove hadoop-0.20-conf-pseudo hadoop-hdfs-*
```

@Zoran B. Djordjevic

37

Install the RPM and Install CDH4

- To install RPM type:

```
$ sudo yum --nogpgcheck localinstall cloudera-cdh-4-0.i386.rpm
```

- Accept all defaults when asked. After a while, you will get:

```
Installed: cloudera-cdh.i386 0:4-0
```

- Optionally, we could add the Cloudera Public GPG Key (GNU Privacy Guard) to the local repository by executing, all on one line:

```
$ sudo rpm --import
http://archive.cloudera.com/cdh4/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
```

- That key will be used for encrypting traffic between Hadoop nodes.

- Next, install Hadoop with YARN in pseudo-distributed mode:

```
$ sudo yum install hadoop-conf-pseudo
```

- To install Hadoop with MRv1 in pseudo-distributed mode, type

```
$ sudo yum install hadoop-0.20-conf-pseudo
```

- yum will list a fairly large number of packages and their combined download size and ask a few times whether you want to proceed. You should always say y(es)

@Zoran B. Djordjevic

38

Verifying Pseudo-Distributed Configuration

- Pseudo-distributed YARN installation has a single node running five daemons called: namenode, secondarynamenode, resourcemanager, datanode and nodemanager.
- Pseudo-distributed Hadoop MRv1 installation consists of one node running five Hadoop daemons called: namenode, secondarynamenode, jobtracker, datanode and tasktracker.
- To view configuration files for YARN on Red Hat (CenOS), type :
\$ rpm -ql hadoop-conf-pseudo
- To view configuration files on Red Hat (CenOS), for MRv1 version, type:
\$ rpm -ql hadoop-0.20-conf-pseudo
/etc/hadoop/conf.pseudo.mrl
/etc/hadoop/conf.pseudo.mrl/README
/etc/hadoop/conf.pseudo.mrl/core-site.xml
/etc/hadoop/conf.pseudo.mrl/hadoop-metrics.properties
/etc/hadoop/conf.pseudo.mrl/hdfs-site.xml
/etc/hadoop/conf.pseudo.mrl/log4j.properties
/etc/hadoop/conf.pseudo.mrl/mapred-site.xml
- The configuration of our Hadoop installation is all contained in /etc/hadoop/conf.pseudo directory.
- All Hadoop components search for the Hadoop configuration in /etc/hadoop/conf.

@Zoran B. Djordjevic

39

Format NameNode

- Before starting the NameNode for the first time we must format the HDFS file system. The installation process did not do that for us.
 - Formatting of namenode must be performed as user hdfs. You can do that using command `hdfs namenode -format` with an additional `sudo -u hdfs` as the part of the command string, as below:
- ```
$ sudo -u hdfs hdfs namenode -format
14/03/05 11:25:12 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = localhost.localdomain/127.0.0.1
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 2.0.0-cdh4.6.0
STARTUP_MSG: classpath = /etc/hadoop/conf:/usr/lib/hadoop/lib/jetty-
6.1.26.jar
. . .
14/03/05 11:25:18 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at
localhost.localdomain/127.0.0.1
*****/
```
- In earlier releases of CHD hadoop-config-pseudo package performed this formatting automatically.

@Zoran B. Djordjevic

40

## Start HDFS

- Before we start HDFS, list all hadoop executables in `/etc/init.d` directory that contain `-hdfs-` and then start them as services.

```
[hadoop@localhost init.d]$ ls -la hadoop-*
-rwxr-xr-x. 1 root root 4335 Nov 20 15:27 hadoop-hdfs-datanode
-rwxr-xr-x. 1 root root 4469 Nov 20 15:27 hadoop-hdfs-namenode
-rwxr-xr-x. 1 root root 4202 Nov 20 15:27 hadoop-hdfs-secondarynamenode
-rwxr-xr-x. 1 root root 4221 Nov 20 15:27 hadoop-mapreduce-historyserver
-rwxr-xr-x. 1 root root 4138 Nov 20 15:27 hadoop-yarn-nodemanager
-rwxr-xr-x. 1 root root 4182 Nov 20 15:27 hadoop-yarn-resourcemanager
$ for x in `cd /etc/init.d ; ls hadoop-hdfs-*` ;
do sudo service $x start ;
done
```

- For MRv1 type:

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

- To verify that services have started, we could visit <http://localhost:50070/> where NameNode provides a web report on Distributed File System (DFS) capacity, number of DataNodes and logs.

@Zoran B. Djordjevic

41

## NameNode, DFS Status Page, localhost:50070

- NameNode has its status reporting web site at port 50070, where one could inquire about the health of the nameNode and HDFS file system.

**NameNode 'localhost:8020' (active)**

**Started:** Wed Feb 26 20:22:20 PST 2014  
**Version:** 2.0.0-cdh4.5.0, 8e266e052e423af592871e2dfe09d54c03f6a0e8  
**Compiled:** Wed Nov 20 15:09:55 PST 2013 by jenkins from (no branch)  
**Upgrades:** There are no upgrades in progress.  
**Cluster ID:** CID-88a75c43-4925-41b7-98cd-c0cacecc80d1  
**Block Pool ID:** BP-1714929148-127.0.0.1-1393473987419

[Browse the filesystem](#)  
[NameNode Logs](#)

---

**Cluster Summary**

Security is OFF  
 1 files and directories, 0 blocks = 1 total.  
 Heap Memory used 60.30 MB is 46% of Committed Heap Memory 128.75 MB. Max Heap Memory is 889 MB.  
 Non Heap Memory used 19.54 MB is 87% of Committed Non Heap Memory 22.25 MB. Max Non Heap Memory is 112 MB.

|                     |   |          |
|---------------------|---|----------|
| Configured Capacity | : | 15.49 GB |
| DFS Used            | : | 24 KB    |
| Non DFS Used        | : | 3.95 GB  |
| DFS Remaining       | : | 11.54 GB |
| DFS Used%           | : | 0.00%    |
| DFS Remaining%      | : | 74.49%   |

@Zoran B. Djordjevic

42

## Create /tmp, staging and log directories

- First recursively remove /tmp directory, if there.

```
$ sudo -u hdfs hadoop fs -rm -r /tmp
```

- For Hadoop with YARN, create new /tmp directory and set permissions

```
$ sudo -u hdfs hadoop fs -mkdir -p /tmp/hadoop-yarn/staging/done_intermediate
```

```
$ sudo -u hdfs hadoop fs -chown -R mapred:mapred /tmp/hadoop-yarn/staging
```

```
$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp
```

- We need to create /var/log/hadoop/yarn because it is the parent of /var/log/hadoop-yarn/apps which is explicitly configured in yarn-site.xml.

- For Hadoop with MRv1, create /tmp and MapReduce system directories

```
$ sudo -u hdfs hadoop fs -mkdir /tmp
```

```
$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp
```

```
$ sudo -u hdfs hadoop fs -mkdir -p /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
```

```
$ sudo -u hdfs hadoop fs -chmod 1777 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
```

```
$ sudo -u hdfs hadoop fs -chown -R mapred /var/lib/hadoop-hdfs/cache/mapred
```

@Zoran B. Djordjevic

43

## Verify HDFS File Structure on YARN

- Run the following command

```
[hadoop@localhost ~]$ sudo -u hdfs hadoop fs -ls -R /
```

- On YARN, we will see a list of directories

```
drwxrwxrwt - hdfs supergroup 0 2014-02-28 09:13 /tmp
```

```
drwxrwxrwt - hdfs supergroup 0 2014-02-28 09:13 /tmp/hadoop-yarn
```

```
drwxrwxrwt - mapred mapred 0 2014-02-28 09:13 /tmp/hadoop-yarn/staging
```

```
drwxrwxrwt - mapred mapred 20. .09:13 /tmp/hadoop-yarn/staging/done_intermediate
```

```
drwxr-xr-x - hdfs supergroup 0 20. . 09:21 /var
```

```
drwxr-xr-x - hdfs supergroup 0 20. . 09:21 /var/log
```

```
drwxr-xr-x - yarn mapred 0 20. . 09:21 /var/log/hadoop-yarn
```

- On MRv2 start YARN, by typing:

```
$ sudo service hadoop-yarn-resourcemanager start
```

```
$ sudo service hadoop-yarn-nodemanager start
```

```
$ sudo service hadoop-mapreduce-historyserver start
```

@Zoran B. Djordjevic

44

## On MRv1, Verify File Structure and Start MapReduce

- To verify HDFS file structure on MRv1, type:

```
$ sudo -u hdfs hadoop fs -ls -R /
drwxrwxrwt - hdfs supergroup 0 2014-03-05 12:29 /tmp
drwxr-xr-x - hdfs supergroup 0 2014-03-05 12:45 /var
drwxr-xr-x - hdfs supergroup 0 2014-03-05 12:45 /var/lib
drwxr-xr-x - hdfs supergroup 0 2014-03-05 12:45 /var/lib/hadoop-hdfs
drwxr-xr-x - hdfs supergroup 0 2014-03-05 12:45 /var/lib/hadoop-hdfs/cache
drwxr-xr-x - hdfs supergroup 0 2014-03-05 12:45 /var/lib/hadoop-hdfs/cache/mapred
drwxrwxrwt - mapred supergroup 0 2014-03-05 12:45 /var/lib/hadoop-hdfs/cache/mapred/staging
```

- On MRV1, start MapReduce by typing:

```
for x in `cd /etc/init.d ; ls hadoop-0.20-mapreduce-*` ; do sudo
service $x start ; done
```

- To verify that services have started, we can check the Web console.
- JobTracker provides a web console <http://localhost:50070/> for viewing running, completed and failed jobs.

@Zoran B. Djordjevic

45

## JobTracker's status site, localhost:50030

**localhost Hadoop Map/Reduce Administration**

State: RUNNING  
 Started: Thu Mar 06 16:29:30 PST 2014  
 Version: 2.0.0-mr1-cdh4.6.0, Unknown  
 Compiled: Wed Feb 26 02:11:55 PST 2014 by Jenkins from Unknown  
 Identifier: 201403061629

**Cluster Summary (Heap Size is 104.50 MB/889 MB)**

| Running Map Tasks | Running Reduce Tasks | Total Submissions | Nodes | Occupied Map Slots | Occupied Reduce Slots | Reserved Map Slots | Reserved Reduce Slots | Map Task Capacity | Reduce Task Capacity | Avg. Tasks/Node | Blacklisted Nodes |
|-------------------|----------------------|-------------------|-------|--------------------|-----------------------|--------------------|-----------------------|-------------------|----------------------|-----------------|-------------------|
| 0                 | 0                    | 0                 | 0     | 0                  | 0                     | 0                  | 0                     | 0                 | 0                    | -               | 0                 |

**Scheduling Information**

| Queue Name | State   | Scheduling Information |
|------------|---------|------------------------|
| default    | running | N/A                    |

Filter (Jobid, Priority, User, Name)   
 Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

**Running Jobs**

@Zoran B. Djordjevic

46

## hdfs dfsadmin

- A very useful command is `hdfs dfsadmin` command, which allows you to quickly review status of your daemons. For example, if daemons are not working, you get something like

```
$ sudo -u hdfs hdfs dfsadmin -report
Configured Capacity: 0 (0 B)
Present Capacity: 0 (0 B)
DFS Remaining: 0 (0 B)
DFS Used: 0 (0 B)
DFS Used%: NaN%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
```

@Zoran B. Djordjevic

47

## Create User, User Directory

- In order to create a new Linux user account `chuck` logged in as user `root`, or type: `$su -` and enter `root`'s password. Then, type:

```
$ useradd -g mapred chuck
```

- The above will create new account/user `chuck`, as a member of group `mapred`. Please note that a user running MapReduce programs must be a member of `mapred` group. To create password for new user, type:

```
$ passwd chuck
```

- At the `New password:` prompt, enter a password for user `chuck`, press [Enter].
- At the `Retype new password:` prompt, enter the same password to confirm.
- Next create the home directory for a new MapReduce user, e.g. `chuck`. By the way in a truly distributed cluster you will do all of this on the NameNode. Type:

```
$ sudo -u hdfs hadoop fs -mkdir /user/chuck
```

```
$ sudo -u hdfs hadoop fs -chown chuck /user/chuck
```

- Alternatively, if the Linux user already exist, you can login as that user and create the home directory as follows:

```
$ sudo -u hdfs hadoop fs -mkdir /user/$USER
```

```
$ sudo -u hdfs hadoop fs -chown $USER /user/$USER
```

- To remove a Linux user type:

```
$ sudo userdel chuck
```

@Zoran B. Djordjevic

48



## Running an example with YARN

- Login as user `chuck` or switch the account `chuck` by typing:  

```
$ su - chuck
```

 Password: \*\*\*\*\*
- Subsequently make a directory in HDFS called `input` and copy some XML files into it by running the following commands:  

```
$ hadoop fs -mkdir input
```

```
$ hadoop fs -put /etc/hadoop/conf/*.xml input
```

```
$ hadoop fs -ls input
```

```
-rw-r--r-- 1 chuck mapred 1458 2014-02-28 10:33 input/core-site.xml
```

```
-rw-r--r-- 1 chuck mapred 1875 2014-02-28 10:33 input/hdfs-site.xml
```

```
-rw-r--r-- 1 chuck mapred 1549 2014-02-28 10:33 input/mapred-site.xml
```

```
-rw-r--r-- 1 chuck mapred 2361 2014-02-28 10:33 input/yarn-site.xml
```
- Set `HADOOP_MAPRED_HOME` for user `chuck`. Best, enter it into `.bash_profile` file.  

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
```

@Zoran B. Djordjevic

49

## Running an example with YARN

- As user `chuck`, run example Hadoop job to grep with a regular expression on input files  

```
$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar grep input output23 'dfs[a-z.]+'
```
- After a lot of console output, the job completes.
- We can find the output in the HDFS directory named `output23` because we specified that output directory to Hadoop. Type:  

```
$ hadoop fs -ls
```
- You will see directories `input` and `output23`  

```
$ hadoop fs -ls output23
```

 will produce the content of `output23`  

```
Found 2 items
```

```
drwxr-xr-x - joe supergroup 0 2014-02-28 10:33 /user/joe/output23/_logs
```

```
-rw-r--r-- 1 joe supergroup 1068 2014-02-28 10:33 /user/joe/output23/part-00000
```

```
-rw-r--r-- 1 joe supergroup 0 2014-02-28 10:33 /user/joe/output23/_SUCCESS
```
- The content of the output file `part-00000` can be seen using `fs -cat` command:  

```
$ hadoop fs -cat output23/part-00000 | head
```

```
1 dfs.datanode.data.dir
```

```
1 dfs.namenode.checkpoint.dir
```

```
1 dfs.namenode.name.dir
```

```
1 dfs.replication
```

```
1 dfs.safemode.extension
```

```
1 dfs.safemode.min.datanodes
```

@Zoran B. Djordjevic

50

## Running an example with MRv1

- As user `chuck`, run example Hadoop job to `grep` with a regular expression on input files  
`$ hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-mapreduce-examples.jar grep input output 'dfs[a-z.]+'`
- After a lot of console output, the job completes.
- We can find the output in the HDFS directory named `output23` because we specified that output directory to Hadoop. Type:

```
$ hadoop fs -ls
```

- You will see directories `input` and `output`

`$hadoop fs -ls output` will produce the content of output

```
Found 2 items
```

```
drwxr-xr-x - joe supergroup 0 2014-02-28 10:33 /user/joe/output/_logs
-rw-r--r-- 1 joe supergroup 1068 2014-02-28 10:33 /user/joe/output/part-00000
-rw-r--r- 1 joe supergroup 0 2014-02-28 10:33 /user/joe/output/_SUCCESS
```

- The content of the output file `part-00000` can be seen using `fs -cat` command:

```
$ hadoop fs -cat output/part-00000 | head
```

```
1 dfs.datanode.data.dir
1 dfs.namenode.checkpoint.dir
1 dfs.namenode.name.dir
1 dfs.replication
1 dfs.safemode.extension
1 dfs.safemode.min.datanodes
```

@Zoran B. Djordjevic

51

## Removing YARN CDH4 Installation

- Stop the daemons:

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

- Remove `hadoop-conf-pseudo`:
- On Red Hat-compatible systems:

```
$ sudo yum remove hadoop-conf-pseudo hadoop-mapreduce-*
```

@Zoran B. Djordjevic

52

## Running an example with MVR1

- To switch user from `hadoop` or `cloudera` to `chuck` type:  

```
$ su - chuck
```

 Password : **\*\*\*\*\***
- Now, as user `chuck` make a directory in HDFS called `input` and copy some XML files into that directory by running the following commands:  

```
$ hadoop fs -mkdir input
```

```
$ hadoop fs -put /etc/hadoop/conf/*.xml input
```

```
$ hadoop fs -ls input
```

```
-rw-r--r-- 1 chuck supergroup 1458 2014-02-28 10:33 input/core-site.xml
```

```
-rw-r--r-- 1 chuck supergroup 1875 2014-02-28 10:33 input/hdfs-site.xml
```

```
-rw-r--r-- 1 chuck supergroup 1549 2014-02-28 10:33 input/mapred-site.xml
```

```
-rw-r--r-- 1 chuck supergroup 2361 2014-02-28 10:33 input/yarn-site.xml
```
- Set `HADOOP_MAPRED_HOME` for user `chuck`. Best, enter it into `.bash_profile` file.  

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
```

@Zoran B. Djordjevic

53

## Running an example with YARN

- As user `chuck`, run example Hadoop job to `grep` with a regular expression on input files  

```
$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar grep input output23 'dfs[a-z.]+'
```
- After a lot of console output, the job completes.
- We can find the output in the HDFS directory named `output23` because we specified that output directory to Hadoop. Type:  

```
$ hadoop fs -ls
```
- You will see directories `input` and `output23`  

```
$ hadoop fs -ls output23
```

 will produce the content of `output23`  

```
Found 2 items
```

```
drwxr-xr-x - joe supergroup 0 2014-02-28 10:33 /user/joe/output23/_logs
```

```
-rw-r--r-- 1 joe supergroup 1068 2014-02-28 10:33 /user/joe/output23/part-00000
```

```
-rw-r--r-- 1 joe supergroup 0 2014-02-28 10:33 /user/joe/output23/_SUCCESS
```
- The content of the output file `part-00000` can be seen using `fs -cat` command:  

```
$ hadoop fs -cat output23/part-00000 | head
```

```
1 dfs.datanode.data.dir
```

```
1 dfs.namenode.checkpoint.dir
```

```
1 dfs.namenode.name.dir
```

```
1 dfs.replication
```

```
1 dfs.safemode.extension
```

```
1 dfs.safemode.min.datanodes
```

@Zoran B. Djordjevic

54

## Running an example with MRv1

- As user `chuck`, run example Hadoop job to `grep` with a regular expression on input files  
`$ hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-mapreduce-examples.jar grep input output 'dfs[a-z.]+'`
- After a lot of console output, the job completes.
- We can find the output in the HDFS directory named `output23` because we specified that output directory to Hadoop. Type:

```
$ hadoop fs -ls
```

- You will see directories `input` and `output`

`$hadoop fs -ls output` will produce the content of output

```
Found 2 items
```

```
drwxr-xr-x - joe supergroup 0 2014-02-28 10:33 /user/joe/output/_logs
-rw-r--r-- 1 joe supergroup 1068 2014-02-28 10:33 /user/joe/output/part-00000
-rw-r--r- 1 joe supergroup 0 2014-02-28 10:33 /user/joe/output/_SUCCESS
```

- The content of the output file `part-00000` can be seen using `fs -cat` command:

```
$ hadoop fs -cat output/part-00000 | head
```

```
1 dfs.datanode.data.dir
1 dfs.namenode.checkpoint.dir
1 dfs.namenode.name.dir
1 dfs.replication
1 dfs.safemode.extension
1 dfs.safemode.min.datanodes
```

@Zoran B. Djordjevic

55

## .bash\_profile File

```
.bash_profile
Get the aliases and functions
if [-f ~/.bashrc]; then
 . ~/.bashrc
fi
User specific environment and startup programs
HADOOP_HOME=/usr/local/hadoop
export HADOOP_HOME
JAVA_HOME=/usr/java/jdk1.6.0_31
export JAVA_HOME
HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
Export HADOOP_MAPPRED_HOME
PATH=$PATH:$HOME/bin:$HADOOP_HOME/bin:$JAVA_HOME/bin
export PATH
```

@Zoran B. Djordjevic

56

## Create new Linux user joe

- First create a Linux user group and assign to it a name and group id (gid). Group id has to be a number greater than 500.
- To see existing group ids, do `$ cat /etc/group`.
- To create new group `hadoopusers` with group id 505, type:  
`$ groupadd hadoopusers - - gid 505`
- Next add a user `joe` who is a member of that group  
`$ useradd -u 504 -g 505 -m -s /bin/bash joe`
- Check that `joe` is created  
`$ cat /etc/passwd | grep joe`  
`joe:x:504:505::/home/joe:/bin/bash`
- Add a password to `joe`  
`$ passwd joe`  
New password:xxxxxxxxx  
Retype new password:xxxxxxxxx
- Add `joe` to sudo users, just like you did it for user `hadoop`.
- Switch the user. Become `joe`  
`# su - joe`  
`[joe@localhost root]$`

@Zoran B. Djordjevic

57

## References

- Hadoop the Definitive Guide, 3<sup>rd</sup> Edition, by Tom White, O'Reilly 2012
- Hadoop in Action, by Chuck Lam, Manning 2011
- Hadoop in Practice, by Alex Holmes, Manning 2012
- Cloudera, CDH4 Quick Start Guide

@Zoran B. Djordjevic

58