Homework 2

Big Data Course 2016-B

Group Name: G-City

Team Members:

Ilan Godik
316315332
ilan3580@gmail.com

Yuval Alfassi
318401015
u67v67@gmail.com

Gil Henkin
315046805
gil7788@gmail. Com

Charlie Mubariky
316278118
Charlie_mabariky.1996@hotmail.com

Tony Tannous
205735046
ttannous@campus.haifa.ac.il

Hadoop Cluster installation on Vagrant

We split our implementation of the installation script to multiple shell scripts, some inside the Vagrantfile, the provided ones in /vagrant/ and additional ones in /vagrant/scripts.

Additionally, we placed all our configuration files in /vagrant/config/, and copied them to the appropriate location in the installation scripts.

Vagrantfile:

1. We allow all our VMs to use 4096MB of memory.
2. We create a private local network using vagrant, with the calculated IP's of 37.51.23.10 for Master node (nodeA), and 37.51.23.11 for the Slave node (nodeB).
3. We forward ports 50070 for the HDFS WebUI and 8088 for the Yarn WebUI from the master node.
4. We set hostnames for the master node and the slave nodes to master, slave respectively. Not doing so would result in a name collision, and a problem detecting both node managers at the same time.
5. We use "vagrant-cachier" to cache apt-get dependencies.
6. We set the hostnames manually, and not via the first 2 lines in the Vagrantfile provision script (these are just for showing you what we've done at a first glace), but by overwriting the /etc/hosts file by the one we provide in /vagrant/config/hosts, and that's because Vagrant by default sets the local hostname to 127.0.0.1, which is problematic, because we need it to be the local network address.
7. We then install all the necessary dependencies from apt-get, and in addition we install dos2unix to ensure we have no problems with line endings.
8. We then convert all the scripts and all the config files to Unix line endings.
9. We install Java, we modified the script to download to /vagrant/ and we avoid redownloading if the file exists. (We also made the extraction silent)
10. We use the provided commands to setup ssh keys between the machines.
11. We copy all the scripts from /vagrant/scripts to the user's directory.
12. We install Hadoop via "source ./install-hadoop.sh" (source so that the current running shell will also have the exports from the spawned shell)
13. And for the Master node only, we complete the setup with "setup-master.sh"

etc/hosts:

1. We set the "master" hostname to 37.51.23.10
2. We set the "slave" hostname to 37.51.23.11
3. We include the default hosts file
4. And we exclude Vagrant's hostname management (e.g. 127.0.0.1 master)

.bashrc:

1. We export JAVA_HOME
2. We export HADOOP_PREFIX to the location of hadoop's installation folder
3. We add $JAVA_HOME/bin:$HADOOP_PREFIX and $HADOOP_PREFIX/sbin to the path, for convenience of using hadoop_daemon/yarn_daemon/stop_all/jps

install-hadop.sh:

1. We download the Hadoop 2.7.2 archive to /vagrant/, avoiding redownloads
2. We extract it to the user's directory
3. We overwrite ~/.bashrc for commands to run on startup of a shell and execute it in the currect shell with "source ~/.bashrc"
4. We set the owner & permissions of the Hadoop folder
5. We copy the configuration files to the correct location
6. We copy the hosts file to the correct location
7. We clean hdfs from previous runs of the machine by removing $HADOOP_PREFIX/hdfs/

setup-master.sh:

1. We format the namenode, forcing reformatting if needed, so that it won't ask any questions during vagrant provisioning.
2. We re-set the ownership & permissions of the Hadoop directory, as formatting the namenode created the namenode directory as root.

after-startup.sh:

1. We left it unchanged, it configures the ssh keys between the two vm's.

start.sh:

1. We call "vagrant up" to bring up the vm's.
2. For the Master node, we call "./after_startup.sh && ./start-master.sh"
3. For the Slave node, we call "./after_startup.sh && ./start-slave.sh"

start-master.sh:

1. We start a NameNode, ResourceManager, DataNode & NodeManager on the Master node.

start-slave.sh:

1. We start a DataNode & NodeManager on the Slave node.

core-site.xml:

1. We set the location of the namenode (fs.defaultFS) to “hdfs://master/”, master being the hostname of the Master node.

hdfs-site.xml:

1. We set the location of the namenode and datanode to $HADOOP_PREFIX/hdfs/<namenode|datanode> respectively
2. We set the replication to 2, to be able to test both Data Notes together & test replication.

yarn-site.xml:

1. We set the maximum memory physical to 4096MB
2. We don’t limit the virtual memory usage.
3. We set the hostname of the resourcemanager to master.
4. We make the Resource Manager WebUI to listen on all IPs on port 8088. (So that we can access it from the outside via localhost:8088)

Work Distribution:

We met up several times for a couple of hours at a time,

And we worked on it all together at the same time, made progress together.