Prerequisites

Install CentOS 6.4 on two machines setup static IP addresses, hostnames and login with root user on both of the machines and add below entries in /etc/hosts file

```
192.168.1.11 spark1.guavus.com
192.168.1.12 spark2.guavus.com
```

Setup keyless ssh on both the nodes usong root user.

Filter below ports from firewall

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8044 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8042 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8088 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9000 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9001 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 40034 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 50070 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 50030 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 50010 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 50075 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 50060 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 19888 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 7077 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 7078 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 18080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 18081 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 53411 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9000 -j ACCEPT
```

Setup Apache Hadoop 2.4.0 (Multi Node Cluster) on CentOS

Step 1. Install JAVA/JDK

Download and Install Java 7 using below commands

- `mkdir /data`
- `cd /data`
- `wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com" "http://download.oracle.com/otn-pub/java/jdk/7u25-b15/jdk-7u25-linux-x64.rpm"`
- `rpm -ivh /data/jdk-7u25-linux-x64.rpm`

Setting java, javaw, javac, jar, jps commands in alternatives

- `alternatives --install /usr/bin/java java /usr/java/latest/jre/bin/java 50000`
- `alternatives --install /usr/bin/javaws javaws /usr/java/latest/jre/bin/javaws 50000`
- `alternatives --install /usr/bin/javac javac /usr/java/latest/bin/javac 50000`
- `alternatives --install /usr/bin/jar jar /usr/java/latest/bin/jar 50000`
- `alternatives --install /usr/bin/jps jps /usr/java/latest/bin/jps 50000`

Setting Java path for current user

- `echo "" >> ~/.bash_profile`
- `echo "export JAVA_HOME=/usr/java/jdk1.7.0_25" >> ~/.bash_profile`
- `echo "export PATH=$PATH:$JAVA_HOME/bin" >> ~/.bash_profile`

Step 2. Download Hadoop 2.4.0

Execute below commands

- `mkdir /home`
- `cd /home`
- `wget http://supergsego.com/apache/hadoop/common/hadoop-2.4.0/hadoop-2.4.0.tar.gz`
- `tar -xvf hadoop-2.4.0.tar.gz`
- `mv  hadoop-2.4.0 hadoop`

Set environment variable uses by hadoop. Edit **~/.bash_profile** file and append following values at end of file by executing below commands.

- `echo "" >> ~/.bash_profile`
- `echo "export HADOOP_HOME=/home/hadoop" >> ~/.bash_profile`
- `echo "export HADOOP_INSTALL=$HADOOP_HOME" >> ~/.bash_profile`
- `echo "export HADOOP_MAPRED_HOME=$HADOOP_HOME" >> ~/.bash_profile`
- `echo "export HADOOP_COMMON_HOME=$HADOOP_HOME" >> ~/.bash_profile`
- `echo "export HADOOP_HDFS_HOME=$HADOOP_HOME" >> ~/.bash_profile`
- `echo "export YARN_HOME=$HADOOP_HOME" >> ~/.bash_profile`
- `echo "export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native" >> ~/.bash_profile`
- `echo "export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin" >> ~/.bash_profile`

Reload Configuration

- `source ~/.bash_profile`

Create hadoop data directories

- `mkdir -p /home/hadoop-data/nn  /home/hadoop-data/snn  /home/hadoop-data/dn /home/hadoop-data/mapred/system  /home/hadoop-data/mapred/local`

Now edit **$HADOOP_HOME/etc/hadoop/hadoop-env.sh** file and set JAVA_HOME environment variable in $HADOOP_HOME/etc/hadoop/hadoop-env.sh

- `export JAVA_HOME=/usr/java/jdk1.7.0_25/`

Edit Configuration Files in **$HADOOP_HOME/etc/hadoop/** directory

Append below content to hdfs-site.xml

```
<configuration>
	<property>
		<name>dfs.replication</name>
		<value>1</value>
	</property>
	<property>
		<name>dfs.name.dir</name>
		<value>file:///home/hadoop-data/nn</value>
	</property>
	<property>
		<name>dfs.data.dir</name>
		<value>file:///home/hadoop-data/dn</value>
	</property>
	<property>
		<name>dfs.namenode.checkpoint.dir</name>
		<value>file:///home/hadoop-data/snn</value>
	</property>
</configuration>
```

Append below content to core-site.xml

```
<configuration>
	<property>
		<name>fs.default.name</name>
		<value>hdfs://spark1.guavus.com:9000</value>
	</property>
</configuration>
```

Append below content to yarn-site.xml

```
<configuration>
	<property>
		<name>yarn.nodemanager.aux-services</name>
		<value>mapreduce_shuffle</value>
	</property>
	<property>
		<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
		<value>org.apache.hadoop.mapred.ShuffleHandler</value>
	</property>
</configuration>
```

Append below content to mapred-site.xml

```
<configuration>
	<property>
		<name>mapreduce.framework.name</name>
		<value>yarn</value>
	</property>
</configuration>
```

Append host names of all the slave nodes in slaves file

```
spark1.guavus.com
spark2.guavus.com
```

[Follow all same STEP 1 and STEP 2 on other node (spark2.guavus.com)]

Format NameNode

After Installation is done format namenode from the master node (spark1.guavus.com) using below command.

- `hdfs namenode -format`

Start/Stop Hadoop Cluster

Start/Stop HDFS using below commands

- `sh $HADOOP_HOME/sbin/start-dfs.sh`
- `sh $HADOOP_HOME/sbin/stop-dfs.sh`

Start/Stop YARN services using below commands

- `sh $HADOOP_HOME/sbin/start-yarn.sh`
- `sh $HADOOP_HOME/sbin/stop-yarn.sh`

Access Hadoop Services in Browser

- Name Node : http://spark1.guavus.com:50070/
- YARN Services : http://spark1.guavus.com:8088/
- Secondary Name Node : http://spark1.guavus.com:50090/
- Data Node 1 : http://spark1.guavus.com:50075/
- Data Node 2 : http://spark2.guavus.com:50075/

Setup Scala on CentOS

Download scala using below commands

- `cd /home/`
- `wget http://www.scala-lang.org/files/archive/scala-2.10.4.tgz`
- `tar -xvf scala-2.10.4.tgz`

Add below lines to ~/.bash_profile

```
export SCALA_HOME=/home/scala-2.10.4

export PATH=$PATH:$SCALA_HOME/bin
```

Reload ~/.bash_profile

- `source ~/.bash_profile`

[Repeat same above steps for other node (spark2.guavus.com)]

Setup Apache Spark 1.0.1 (Multi Node Cluster) on CentOS

Download Apache Spark using below commands

- `cd /home/`
- `wget http://d3kbcqa49mib13.cloudfront.net/spark-1.0.1.tgz`
- `tar -xvf spark-1.0.1.tgz`

Configuration in spark-env.sh

Create /home/spark-1.0.1-bin-hadoop2/conf/spark-env.sh and add below lines to the file

```
SPARK_JAVA_OPTS=-Dspark.driver.port=53411
HADOOP_CONF_DIR=$HADOOP_HOME/conf
SPARK_MASTER_IP=spark1.guavus.com
```

Create /home/spark-1.0.1-bin-hadoop2/conf/spark-defaults.conf and add below lines to the file

```
spark.master            spark://spark1.guavus.com:7077
spark.serializer        org.apache.spark.serializer.KryoSerializer
```

Append hostnames of all the slave nodes in /home/spark-1.0.1-bin-hadoop2/conf/slaves file

```
spark1.guavus.com
spark2.guavus.com
```

[Repeat same above steps for other node (spark2.guavus.com)]

Start/Stop Spark using below commands

- `sh /home/spark-1.0.1-bin-hadoop2/sbin/start-all.sh`
- `sh /home/spark-1.0.1-bin-hadoop2/sbin/stop-all.sh`

Start Spark shell using YARN

- `cd /home/spark-1.0.1-bin-hadoop2`
- `./bin/spark-shell --master yarn-client`

Submit Job using YARN cluster

- `cd /home/spark-1.0.1-bin-hadoop2`
- `./bin/spark-submit --class my.main.Class --master yarn-cluster`

Access SPARK UI in Browser

- Spark Master : http://spark1.guavus.com:8080/