**AWS Hadoop Cluster Installation**

**Creating AWS Account:**

**· Go to aws.amazon.com and sign in/sign up using existing Amazon account**

**· Fill contact information using ‘Personal Account’**

**· Fill Credit/Debit card details (avoid putting PAN card number)**

**· Choose ‘Basic Plan’**

**· Sign in to console > AWS Service > EC2**

**Creating EC2 instance, and selecting Ubuntu 14.04 as our Linux Server:**

**● click on 'launch instance' button**

**● quick launch > ubuntu 14.04 > select**

**● select t2.micro > next: configure instance details**

**● provide 2 instances > next: add storage**

**● next: tag instances**

**● type - hadoop, hadoop in key and value > next**

**● select an existing security group > review and launch**

**● launch**

**● 2 instances running - click**

**● create a new key pair > key pair name: hadoop > download key pair**

**● store .pem key with name as hadoop.pem**

**● Launch instances**

**● Scroll down ‘View Instances’**

**Rename both instances as ‘master’ and ‘slave’ under name column.**

**copy public DNS one will be master and second as slave:**

**master****[ec2-52-90-30-236.compute-1.amazonaws.com](http://ec2-52-90-30-236.compute-1.amazonaws.com/)**

**slave****[ec2-54-234-141-102.compute-1.amazonaws.com](http://ec2-54-234-141-102.compute-1.amazonaws.com/)**

**Edit security groups**

**goto Network and security > Security Groups**

**Change inbound and outbound rules and allow everything to connect to system as this is a non-prod environment.**

**Creating private key:**

**As we will be using Putty to login and Putty works with private key. Therefore using puttygen.exe we will be creating private key using .pem key we downloaded earlier.**

**open puttygen**

**load > select hadoop.pem key > save private key > yes**

**save hadoop.ppk file**

**Connecting to EC2 instance from Windows:**

**open putty**

**In hostname type ubuntu@ec2-52-90-30-236.compute-1.amazonaws.com**

**Type ‘master’ in saved session > save**

**connection > SSH > auth > browse > select hadoop.ppk**

**Click 'open'**

**Right click on the window > change setting > behavior > master/slave**

**Setup after logging into EC2 instance(both):**

**Changing hostname**

**hostname**

**sudo hostname****[ec2-52-90-30-236.compute-1.amazonaws.com](http://ec2-52-90-30-236.compute-1.amazonaws.com/)**

**hostname**

**sudo nano /etc/hosts**

**Replace ‘127. …localhost …’ with public DNS of master**

**Ctrl+x > y > enter**

**cat /etc/hosts**

**Copy pem key into master and slave using WinSCP:**

**open winscp (both)**

**Username: ubuntu**

**Hostname: public DNS of master/slave**

**Click on Advanced > SSH > Authentication.**

**Browse and select hadoop.ppk file.**

**Drag and drop hadoop.pem file into /home/ubuntu**

**Open putty:**

**To update and install Java**

**sudo apt-get update**

**sudo apt install openjdk-8-jre-headless**

**java -version**

**ls /usr/lib/jvm/**

**Downloading Hadoop**

**wget https://dist.apache.org/repos/dist/release/hadoop/common/hadoop-2.7.1/hadoop-2.7.1.tar.gz**

**tar -xzvf hadoop-2.7.1.tar.gz**

**mv hadoop-2.7.1 hadoop**

**Configure Hadoop and Java path in .bashrc**

**cd**

**nano .bashrc**

**<scroll to the bottom>**

**export HADOOP\_CONF= /home/ubuntu/hadoop/etc/hadoop**

**export HADOOP\_PREFIX=/home/ubuntu/hadoop**

**#Set JAVA\_HOME**

**export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64**

**# Add Hadoop bin/ directory to path**

**export PATH=$PATH:$HADOOP\_PREFIX/bin**

**ctrl+x > y > enter**

**source .bashrc**

**echo $HADOOP\_PREFIX**

**echo $HADOOP\_CONF**

**rm -fr hadoop-2.7.1.tar.gz**

**For passwordless connection:**

**chmod 644 .ssh/authorized\_keys**

**chmod 400 hadoop.pem**

**eval `ssh-agent`**

**ssh-add hadoop.pem**

**Above commands need to be executed everytime we start exit bash**

**To check if passwordless connection is setup successfully or not**

**from master**

**ssh ubuntu@<public DNS of slave instance>**

**hostname (should give public DNS of slave)**

**exit**

**from slave**

**ssh ubuntu@<public DNS of master instance>**

**hostname (should give public DNS of master)**

**exit**

**SETUP DONE!!!**

**Editing config files in Hadoop - hadoop-env.sh, core-site.xml, hdfs-site.xml, yarn-site.xml, mapred-site.xml – ONLY MASTER:**

**nano $HADOOP\_CONF/hadoop-env.sh**

**export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64**

**ctrl+x > y > enter**

**nano $HADOOP\_CONF/core-site.xml**

**<configuration>**

**<property>**

**<name>**[**fs.default.name**](http://fs.default.name/)**</name>**

**<value>hdfs://<public DNS of master instance>:8020</value>**

**</property>**

**<property>**

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

**<value>/home/ubuntu/hdfstmp</value>**

**</property>**

**</configuration>**

**ctrl+x > y > enter**

**cd**

**mkdir hdfstmp**

**nano $HADOOP\_CONF/hdfs-site.xml**

**<configuration>**

**<property>**

**<name>dfs.replication</name>**

**<value>2</value>**

**</property>**

**</configuration>**

**ctrl+x > y > enter**

**cp $HADOOP\_CONF/mapred-site.xml.template $HADOOP\_CONF/mapred-site.xml**

**nano $HADOOP\_CONF/mapred-site.xml**

**<configuration>**

**<property>**

**<name>mapreduce.framework.name</name>**

**<value>yarn</value>**

**</property>**

**</configuration>**

**ctrl+x > y > enter**

**nano $HADOOP\_CONF/yarn-site.xml**

**<configuration>**

**<property>**

**<name>yarn.nodemanager.aux-services</name>**

**<value>mapreduce\_shuffle</value>**

**</property>**

**ctrl+x > y > enter**

**cd $HADOOP\_CONF**

**scp hadoop-env.sh core-site.xml yarn-site.xml hdfs-site.xml mapred-site.xml ubuntu@<public DNS of slave instance>: /home/ubuntu/hadoop/etc/hadoop**

**Edit master and slave files in master:**

**nano /home/ubuntu/hadoop/etc/hadoop/master**

**<public dns of master>**

**nano /home/ubuntu/hadoop/etc/hadoop/slaves**

**<public dns of slave>**

**Edit master and slave files in slave:**

**nano /home/ubuntu/hadoop/etc/hadoop/master**

**blank**

**nano /home/ubuntu/hadoop/etc/hadoop/slaves**

**<public dns of slave>**

**Starting Hadoop from master:**

**hadoop namenode -format**

**cd /home/ubuntu/Hadoop/sbin**

**./start-all.sh**

**[http://ec2-52-90-30-236.compute-1.amazonaws.com:50070/dfshealth.html](http://ec2-52-90-30-236.compute-1.amazonaws.com:50070/dfshealth.jsp)**

**[http://ec2-52-90-30-236.compute-1.amazonaws.com:8088](http://ec2-52-90-30-236.compute-1.amazonaws.com:50070/dfshealth.jsp)**

**Type jps is master and slave to check if everything is good:**

**Master:**

**Namenode, Resource Manager and Secondary Namenode**

**Slave:**

**Datanode, Node Manager**