Build a Hadoop Cluster on AWS EC2

(Elastic Cloud Computing)

# CONTENTS

1. Introduction to Amazon EC2

2. Create an AWS Account

3. Setup the Instances

4. Setup connectivity using a SSH Client

5. First time Login – Create User, Configure Network settings

6. Download and Install Oracle Java

7. Setup Password less connectivity between the Nodes

8. Download and Install Apache Hadoop

9. Start Hadoop Services and Configure First time settings

10. Run a Sample MR program in the Cluster

# Introduction to Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage.

EC2 provides the following features:

* Virtual computing environments, known as instances.
* Preconfigured templates for your instances, known as Amazon Machine Images (AMIs) with OS and Software choices.
* Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types.
* Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place).
* Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses.

# 2. What is AWS Free Tier?

You can try some AWS services free of charge within certain usage limits. AWS calls this the [AWS Free Tier](https://aws.amazon.com/free/). The free tier is designed to give you hands-on experience with a range of AWS services at no charge.

* When you create an AWS account, you're automatically signed up for the free tier for 12 months. Your free tier eligibility expires at the end of the 12-month period. When your free tier expires, AWS starts charging the regular rates for any AWS services and resources that you're using.
* To avoid charges while on the free tier, you must keep your usage below the free tier limits. You are charged for any usage that exceeds the limits. If you don't use the full benefits provided by the free tier in a given month, the benefits don't roll over to the next month.

**Amazon EC2 Free Tier Limits per Month:**

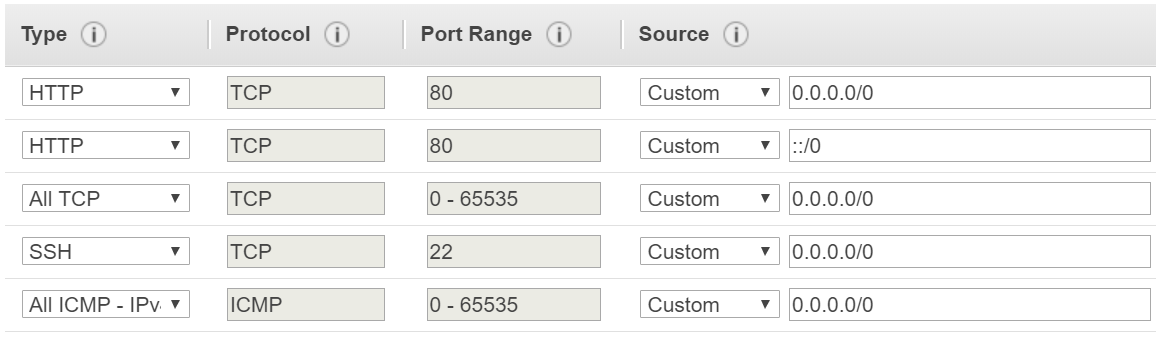
* **750 Hours** of **t2.micro** Instance
* **30 GB** of Elastic Block Storage (Magnetic or SSD)
* **2K** PUT Requests to Amazon S3
* **20K** GET Requests from Amazon S3

# Setup the Instances

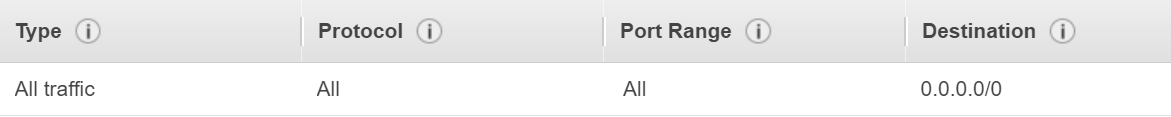
* Choose an Amazon Machine Image (AMI)
* Choose an Instance Type
* Configure Instance Details
* Add Elastic Block Storage
* Add Tags (Optional)
* Add Security Group
* Create new Key pair and Download the Private Key
* Clone the rest of them from the first Instance
* Convert the Amazon Private Key (.pem) to .ppk

# Configure Security Group

* Inbound Rule :



* Outbound Rule :



# Create User and Configure Network Settings

Default user for RHEL instances is ec2-user. You should always login using this account.

**Update the Installation in each Node:**

sudo yum update

**Create new user for Hadoop in each Node:**

sudo groupadd hadoop

sudo useradd hduser -g hadoop

sudo usermod -aG wheel hduser

sudo passwd hduser

**Update the Cluster Hosts in each Node (/etc/hosts):**

<private ip of namenode-1> namenode-01.compute.amazonaws.com

<private ip of datanode-1> datanode-01.compute.amazonaws.com

<private ip of datanode-2> datanode-02.compute.amazonaws.com

**Update the individual Hostnames (/etc/hostname):**

namenode-01.compute.amazonaws.com

datanode-01.compute.amazonaws.com

datanode-02.compute.amazonaws.com

# Download and Install Oracle Java in each Node

**Download Java from the following location:**

sudo wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u191-b12/2787e4a523244c269598db4e85c51e0c/jdk-8u191-linux-x64.tar.gz"

**Install Java and set Alternatives for the current version:**

sudo mkdir /opt/java

sudo mv jdk-8u191-linux-x64.tar.gz /opt/java

sudo tar xzf jdk-8u191-linux-x64.tar.gz

sudo alternatives --install /usr/bin/java java /opt/java/jdk1.8.0\_191/bin/java 1

sudo alternatives --install /usr/bin/javac javac /opt/java/jdk1.8.0\_191/bin/javac 1

sudo alternatives --install /usr/bin/jar jar /opt/java/jdk1.8.0\_191/bin/jar 1

**Test Installation:**

javac -version

java –version

# Setup Password less connectivity from the Namenode to the Datanodes

Hadoop requires the Namenode to talk to the Datanodes very frequently. There should be a mechanism to connect to the Datanodes from the Namenode without any Password requirement. To achieve this, you have to use SSH public-key authentication. This authentication technique generates a pair of separate keys; one private and the other public. You keep the private key a secret and store it on the Namenode and conceivably share the public key with the Datanodes. In general, if you want to connect to a destination system, the source system’s public key has to be transferred to the destination system’s authorized keys.

**Create private-public keypair and transfer the Public key to the Datanodes:**

ssh-keygen -t rsa -P ""

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

ssh-copy-id -i ~/.ssh/id\_rsa.pub hduser@datanode-01.compute.amazonaws.com

ssh-copy-id -i ~/.ssh/id\_rsa.pub hduser@datanode-02.compute.amazonaws.com

**Test Passwordless Connectivity:**

ssh namenode-01.compute.amazonaws.com

ssh datanode-01.compute.amazonaws.com

ssh datanode-02.compute.amazonaws.com

# Download and Install Apache Hadoop (Only in Namenode)

**Download Hadoop from the following location:**

sudo wget http://www.us.apache.org/dist/hadoop/common/hadoop-2.8.5/hadoop-2.8.5.tar.gz

**Create necessary folders and Install Hadoop:**

sudo mkdir -p /opt/hadoop/metadata/yarn/local

sudo mkdir -p /opt/hadoop/metadata/yarn/log

sudo mkdir -p /opt/hadoop/metadata/hdfs/datanode

sudo mkdir -p /opt/hadoop/metadata/hdfs/namenode

sudo mv hadoop-2.8.5.tar.gz /opt/hadoop

sudo tar xzf hadoop-2.8.5.tar.gz

sudo chown -R hduser:hadoop /opt/hadoop

**Modify the Hadoop Configuration files:** core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml

**Modify Slaves files:** List all slave hostnames or IP addresses in the **slaves** file, one per line.

**Copy Hadoop installation to the Datanodes:**

cd /opt

scp -r hadoop datanode-01.compute.amazonaws.com:/home/hduser

scp -r hadoop datanode-02.compute.amazonaws.com:/home/hduser

# Download and Install Apache Hadoop (Only in Namenode)

**Create and Modify Environment variables:**

export JAVA\_HOME=/opt/java/jdk1.8.0\_191

export PATH=$PATH:$JAVA\_HOME/bin

export HADOOP\_HOME=/opt/hadoop/hadoop-2.8.5

export PATH=$PATH:$HADOOP\_HOME/bin

export PATH=$PATH:$HADOOP\_HOME/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib"

export CLASSPATH=$CLASSPATH:$HADOOP\_HOME/lib/\*:.

export HADOOP\_OPTS="$HADOOP\_OPTS -Djava.security.egd=file:///dev/../dev/urandom"

**Do the below steps only in the Datanodes:**

sudo mv /home/hduser/hadoop /opt

# Empty the slaves configuration file

hadoop version

# Start Hadoop Services and Configure First time Settings

**Format the Namenode:**

hdfs namenode -format

**Start HDFS and YARN Services:**

$HADOOP\_HOME/sbin/start-dfs.sh

$HADOOP\_HOME/sbin/start-yarn.sh

**Check Services:**

jps

**Create HDFS Directory Structure:**

hdfs dfs -mkdir /tmp

hdfs dfs -mkdir /user

hdfs dfs -mkdir /user/app

hdfs dfs -mkdir /user/hduser

hdfs dfs -chmod -R 1777 /tmp

hdfs dfs -chmod -R 1777 /user

hdfs dfs -chmod -R 1777 /user/app

hdfs dfs -chmod -R 1777 /user/hduser

# Run a Sample MapReduce program in the Cluster

**Run a Sample MR Program:**

hadoop jar $HADOOP\_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.5.jar pi 2 4

**Cluster Monitoring:**

[http://<namenode public url>:50070](http://ec2-13-127-196-101.ap-south-1.compute.amazonaws.com:50070/)

**Stop HDFS and YARN Services:**

$HADOOP\_HOME/sbin/stop-yarn.sh

$HADOOP\_HOME/sbin/stop-dfs.sh