



**Data Foundation Installation Play Book**

**Authors:**

IBM Software Group, Lab Services Team

Date Created: 12 –Apr-2017

Last Updated: 03 –Mar-YYYY

VERSION: 1.00

The information contained herein is proprietary to YNAP and IBM. The recipient of this document, by its retention and use, agrees to protect the information contained herein.

Readers are advised that this project may be subject to the terms of a non-disclosure agreement.

DO NOT DISCLOSE ANY OF THIS INFORMATION WITHOUT OBTAINING PERMISSION FROM THE MANAGEMENT RESPONSIBLE FOR THIS DOCUMENT.

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Revision Date** | **Revision Description** | **Authors** |
| 0.1 | 12-04-2017 | Draft Version Initial copy | Nirmal Ramesan |
| 0.1 | 12-04-2017 | Hadoop Installation details | Nirmal Ramesan |
| 0.2 | 13-04-2017 | Spark Installation details | Sunil M |
| 0.3 | 14-04-2017 | Cassandra Installation and Clustering | Nagalakshmi Vellaichamy |
| 0.3 | 14-04-2017 | Elastic search and clustering | Charls Chacko |
| 0.4 | 15-04-2017 | Node JS details | Himanshu Mech |
| 1.0 | 17-04-2017 | Reviewed 1st version and comments fixed | Sudhanshu Sar |

**Document Approval History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Description** | **Owner** | **Location** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Reference Documents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Description** | **Owner** | **Location** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 4](#_Toc480195905)

[2 Business Background 4](#_Toc480195906)

[3 Scope 4](#_Toc480195907)

[3.1 In Scope 4](#_Toc480195908)

[4 Architecture 5](#_Toc480195909)

[4.1 High level Architecture on Data Foundation Tier 5](#_Toc480195910)

[5 HDFS Installation with Clustering 5](#_Toc480195911)

[Step 1: Verifying Java Installation 5](#_Toc480195912)

[Step 2: Creating User Account 6](#_Toc480195913)

[Step 3: Mapping the nodes 6](#_Toc480195914)

[Step 4: Configuring Key Based Login 6](#_Toc480195915)

[Step 5: Installing Hadoop 6](#_Toc480195916)

[Step 6: Configuring Hadoop 7](#_Toc480195917)

[Step 7: Installing Hadoop on Slave Servers 8](#_Toc480195918)

[Step 8: Configuring Hadoop on Master Server 8](#_Toc480195919)

[Step 9: Starting Hadoop Services 9](#_Toc480195920)

[Step 10: Verify the Installation 9](#_Toc480195921)

[6 Spark Streaming Installation 11](#_Toc480195922)

[Step 1: Verifying Java Installation 11](#_Toc480195923)

[Step 2: Scala installation 12](#_Toc480195924)

[Step 3: Verify Scala installation 12](#_Toc480195925)

[Step 4: Installing Spark 12](#_Toc480195926)

[Step 5: Verifying Spark Installation 13](#_Toc480195927)

[SPARK Streaming 14](#_Toc480195928)

[7 Cassandra Installation and Clustering 14](#_Toc480195929)

[Step 1: Verifying Java Installation 14](#_Toc480195930)

[Step 2: Creating User Account 15](#_Toc480195931)

[Step 3: Cassandra installation 15](#_Toc480195932)

[Step 4: Configure Cassandra Cluster 15](#_Toc480195933)

[Step 5: Starting the Nodes in the Cluster 16](#_Toc480195934)

[Step 6: Verifying the Cassandra Cluster 17](#_Toc480195935)

[Step 7: Connecting to CQLSH 17](#_Toc480195936)

[8 Elastic Search Installation and Clustering 17](#_Toc480195937)

[Step 1: Verifying Java Installation 18](#_Toc480195938)

[Step 2: Elastic Search installation 18](#_Toc480195939)

[Step 3: Configure and Start Elasticsearch 18](#_Toc480195940)

[Step 4: Setup and Elastic Search Cluster 19](#_Toc480195941)

[9 Node JS Installation 20](#_Toc480195942)

[Step 1: Install Node.js using Yum 20](#_Toc480195943)

[Step 2: Verify Node.js Installation 20](#_Toc480195944)

[10 Reference 21](#_Toc480195945)

# Introduction

YNAP has IBM OMS & IBM WCS to support their retail business. These two applications have Oracle databases. Currently the data is not available in real time from these databases.

YNAP requires a data foundation that will bring data from their commerce applications in real-time and into one destination. This data foundation will drive the cognitive analytical solutions.

Along with the real-time data replication solution we will move the historical data from legacy YOOX & NAP systems into the same destination.

This Play Book covers the Installation and configuration of the application and components in the Data Foundation Tier.

# Business Background

YNAP has embarked on a cognitive journey that will provide features such as Real Time Personalization, Order Optimization etc. One of the building blocks for this cognitive journey is Data Foundation.

To develop Data Foundation, data from multiple sources needs to be transported into one single destination. Some of the data sources such as Order Management Systems will require non-intrusive methods so that production servers do not get unduly burdened. Data Foundation will require data replication in real time as well as a one-time transportation of the historical data. Destination is typically a NoSQL environment given the volume, variety, and velocity of data pumped into the destination.

# Scope

## In Scope

Installation and Configuration of following systems in YNAP’s Data Foundation Tier environment

• Hadoop

• SPARK

• Cassandra

• Elastic Search

* Node JS

# Architecture

## High level Architecture on Data Foundation Tier

**Figure 4.1.1**



***Node JS***

**Data Foundation Tier**

***Elastic Search***

# HDFS Installation with Clustering

The below section covers the Installation of Hadoop – Multi Node Cluster on a distributed environment. The steps cover the cluster environment with 1 master node and 2 slave nodes which can be extended according to the implementation.

**Pre-requisites:**

* JDK 64 Bit
* Hadoop Installable hadoop-2.7.3.tar.gz

**Installation Steps:**

## **Step 1: Verifying Java Installation**

Java is the main prerequisite for Hadoop. You should verify the existence of java in your system using “java -version”

# java -version

If java is not installed in your system, then follow the steps for installing java.

## **Step 2: Creating User Account**

Create a system user account on both master and slave systems to use the Hadoop installation.

# useradd hadoop

# passwd hadoop

## **Step 3: Mapping the nodes**

You must edit hosts file in /etc/ folder on all nodes as **root** user, specify the IP address of each system followed by their host names.

# vi /etc/hosts

enter the following lines in the /etc/hosts file.

<Master node host IP> hadoop-master

<Slave node 1 host IP> hadoop-slave-1

<Slave node 2 host IP> hadoop-slave-2

Ex:

192.168.1.109 hadoop-master

192.168.1.145 hadoop-slave-1

192.168.56.1 hadoop-slave-2

## **Step 4: Configuring Key Based Login**

Setup ssh in every node such that they can communicate with one another without any prompt for password.

# su hadoop

$ ssh-keygen -t rsa

$ ssh-copy-id -i ~/.ssh/id\_rsa.pub hadoop@hadoop-master

$ ssh-copy-id -i ~/.ssh/id\_rsa.pub hadoop @hadoop-slave-1

$ ssh-copy-id -i ~/.ssh/id\_rsa.pub hadoop @hadoop-slave-2

$ chmod 0600 ~/.ssh/authorized\_keys

$ exit

## **Step 5: Installing Hadoop**

In the Master server, install Hadoop using the following commands.

# mkdir /opt/hadoop

# cd /opt/hadoop/

Copy the Hadoop installables mentioned in the pre-requisite to the folder /opt and untar it

# tar -xzf hadoop-2.7.3.tar.gz

# mv hadoop-2.7.3 hadoop

# chown -R hadoop /opt/hadoop

# cd /opt/hadoop

## **Step 6: Configuring Hadoop**

You must configure Hadoop server by making the following changes as given below files located at /opt/hadoop/etc/hadoop

core-site.xml

Open the core-site.xml file and edit it as shown below.

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://hadoop-master:9000/</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

hdfs-site.xml

Open the hdfs-site.xml file and edit it as shown below.

<configuration>

<property>

<name>dfs.data.dir</name>

<value>/opt/hadoop /dfs/name/data</value>

<final>true</final>

</property>

<property>

<name>dfs.name.dir</name>

<value>/opt/hadoop/dfs/name</value>

<final>true</final>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

mapred-site.xml

Open the mapred-site.xml file and edit it as shown below.

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>hadoop-master:9001</value>

</property>

</configuration>

hadoop-env.sh

Open the hadoop-env.sh file and edit JAVA\_HOME, HADOOP\_CONF\_DIR, and HADOOP\_OPTS as shown below.

Note: Set the JAVA\_HOME as per your system configuration.

export JAVA\_HOME=/opt/jdk1.7.0\_17

export HADOOP\_OPTS=-Djava.net.preferIPv4Stack=true

export HADOOP\_CONF\_DIR=/opt/hadoop/etc/hadoop

## **Step 7: Installing Hadoop on Slave Servers**

Install Hadoop on all the slave servers by following the given commands.

# su hadoop

$ cd /opt/hadoop

$ scp -r hadoop hadoop-slave-1:/opt

$ scp -r hadoop hadoop-slave-2:/opt

## **Step 8: Configuring Hadoop on Master Server**

Open the master server and configure it by following the given commands.

# su hadoop

$ cd /opt/hadoop

Configuring Master Node

$ vi etc/hadoop/masters

hadoop-master

Configuring Slave Node

$ vi etc/hadoop/slaves

hadoop-slave-1

hadoop-slave-2

Format Name Node on Hadoop Master

# su hadoop

$ cd /opt/hadoop/

$ bin/hadoop namenode –format

System should response with success message for formatting the name node.

## **Step 9: Starting Hadoop Services**

The following command is to start all the Hadoop services on the Hadoop-Master.

$ cd /opt/hadoop /sbin

$ ./start-all.sh

start-all.sh & stop-all.sh

Used to start and stop hadoop daemons all at once. Issuing it on the master machine will start/stop the daemons on all the nodes of a cluster.

## **Step 10: Verify the Installation**

Run the following command in the Master node and slave nodes separately to check the running daemons:

#jps

Hadoop daemons in Master node are:

NameNode

SecondaryNameNode

ResourceManager

JobHistoryServer

Hadoop daemons in Slave node are:

DataNode

NodeManager

**start-dfs.sh & stop-dfs.sh**

start/stop only HDFS daemons separately on all the nodes from the master machine. (HDFS Daemons are NameNode, SecondaryNameNode and DataNode)

Daemons in master node

NameNode

SecondaryNameNode

Daemons in slave node

DataNode

**start-yarn.sh & stop-yarn.sh**

start/stop YARN daemons separately on all the nodes from the master machine. (YARN daemons are ResourceManager and NodeManager)

Daemons in master node

ResourceManager

Daemons in slave node

NodeManager

**Start individual Hadoop daemons**

To start individual daemons on an individual machine manually. You need to go to a particular node and issue these commands from the sbin folder.

./hadoop-daemon.sh namenode - Start namenode

./hadoop-daemon.sh datanode - Start datanode

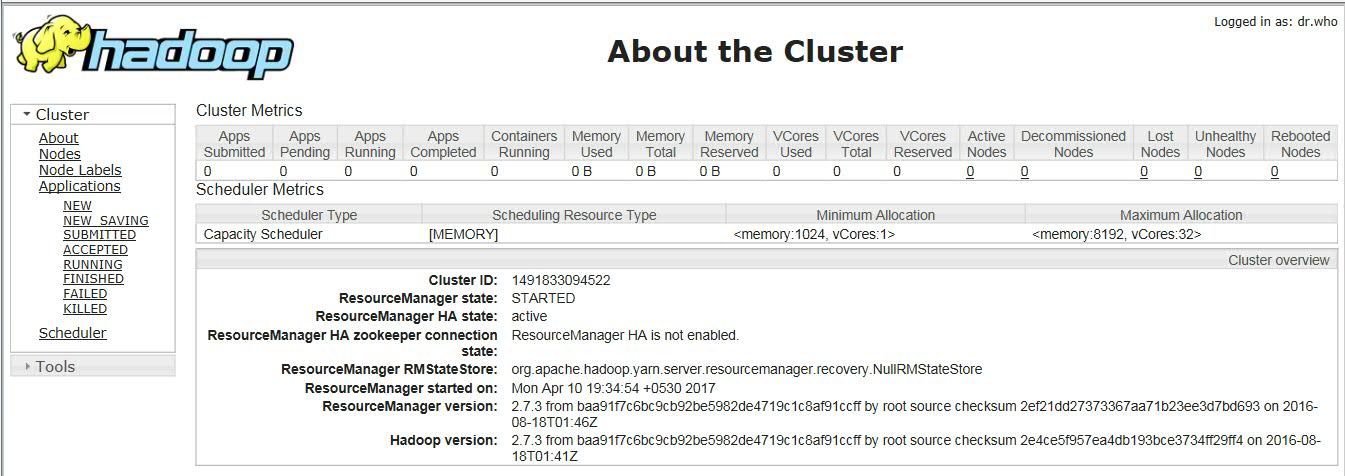
./yarn-deamon.sh resourcemanager - Start resourcemanager

./yarn-deamon.sh nodemanager - Start nodemanager

./sbin/hadoop-daemons.sh start datanode - Start History Server

**Default URL**

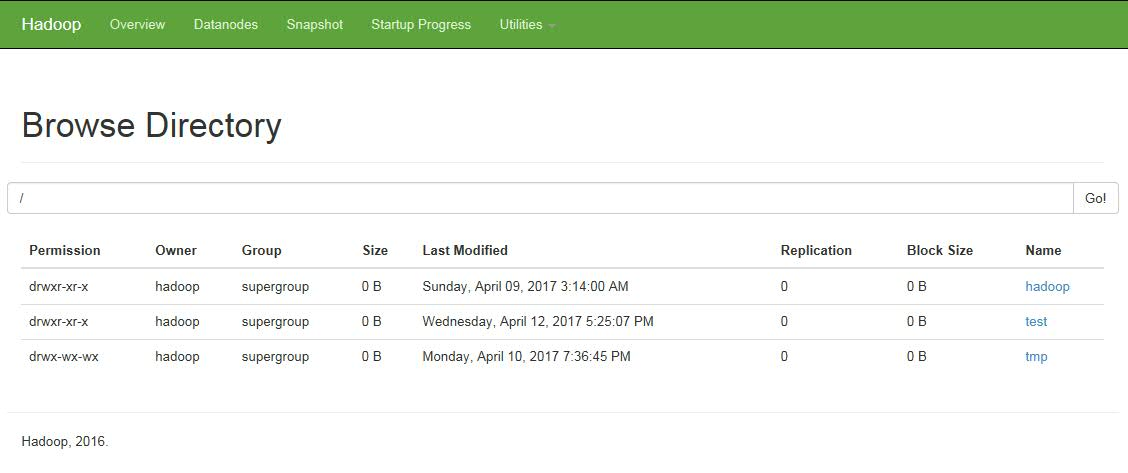
Cluster Information:[http:// <masterHost>:8088/cluster/cluster](http://9.109.117.40:8088/cluster/cluster)



**Job History Server -** http://<masterHost>:19888/jobhistory



FS Bowser: http://<masterHost>:50070/explorer.html#/



**How to Put a file into HDFS to test**

1. Go to /opt/hadoop/bin in the master node
2. Create a new directory ‘test’ in the File System

#hadoop fs -mkdir /test

1. Push a file to the ‘test’ directory using the following command

#hadoop fs -put /opt/softwares/testfile.txt /test

1. Open the FS Bowser URL and verify the new file is successfully placed.

# Spark Streaming Installation

The below section covers the Installation of Spark on multiple Nodes of Hadoop Cluster (Master- Slave) on a distributed environment.

**Pre-requisites:**

* JDK 64 Bit
* Scala Installable
* Spark Installable compatible with Hadoop version

**Installation Steps:**

## **Step 1: Verifying Java Installation**

Java is the main prerequisite for Spark. Verify the existence of java in your system using “java -version”

# java -version

If java is not installed in your system, then follow the steps for installing java.

## **Step 2: Scala installation**

Download the latest version of Scala and copy the Scala tar file in the download folder (ex: /opt/downloads).

Change to Hadoop user created for the Hadoop Installation in the previous section

#su – hadoop

Extract the Scala tar file

# tar xvf scala-2.11.6.tgz

Copy the Scala file to the respective directory

# mv scala-2.11.6 /opt/scala

Set Path and Home for Scala.

vi ~/.bashrc

Add the following for setting the PATH and SCALA\_HOME

export SCALA\_HOME=/opt/scala

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

Use the following command for sourcing the ~/.bashrc file.

# source ~/.bashrc

## **Step 3: Verify Scala installation**

Use the following command to verify the scala installation.

$scala -version

Scala code runner version 2.11.6 -- Copyright 2002-2013, LAMP/EPFL

## **Step 4: Installing Spark**

Download the latest version of Spark specific to the Hadoop version you have installed and copy the Spark tar file in the download folder (ex: /opt/downloads).

### Extracting Spark tar

#tar xvf spark-1.3.1-bin-hadoop2.6.tgz

### Moving Spark software files

#mv spark-1.3.1-bin-hadoop2.6 /opt/spark

### Setting up the environment for Spark

vi ~/.bashrc

Add the following for setting the PATH and SPARK\_HOME

export SPARK\_HOME=/opt/spark

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

Use the following command for sourcing the ~/.bashrc file.

# source ~/.bashrc

## **Step 5: Verifying Spark Installation**

Write the following command for opening Spark shell.

$spark-shell

If spark is installed successfully then you will find the following output.

Spark assembly has been built with Hive, including Datanucleus jars on classpath

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

15/06/04 15:25:22 INFO SecurityManager: Changing view acls to: hadoop

15/06/04 15:25:22 INFO SecurityManager: Changing modify acls to: hadoop

15/06/04 15:25:22 INFO SecurityManager: SecurityManager: authentication disabled;

ui acls disabled; users with view permissions: Set(hadoop); users with modify permissions: Set(hadoop)

15/06/04 15:25:22 INFO HttpServer: Starting HTTP Server

15/06/04 15:25:23 INFO Utils: Successfully started service 'HTTP class server' on port 43292.

Welcome to

\_\_\_\_ \_\_

/ \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_

\_\ \/ \_ \/ \_ `/ \_\_/ '\_/

/\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 1.4.0

/\_/

Using Scala version 2.10.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0\_71)

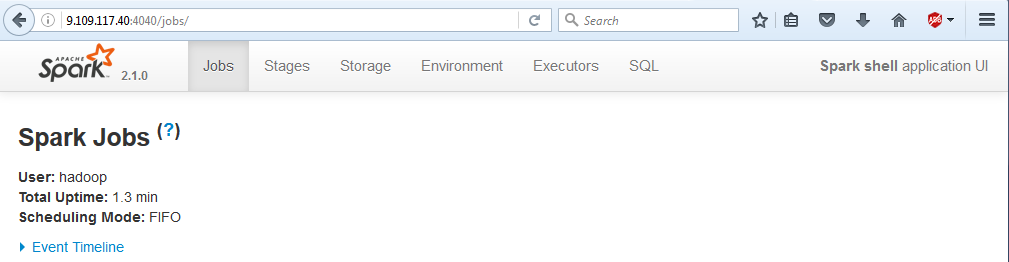
Type in expressions to have them evaluated.

Spark context available as sc

scala>

Now check the Spark UI from the browser using the below URI:

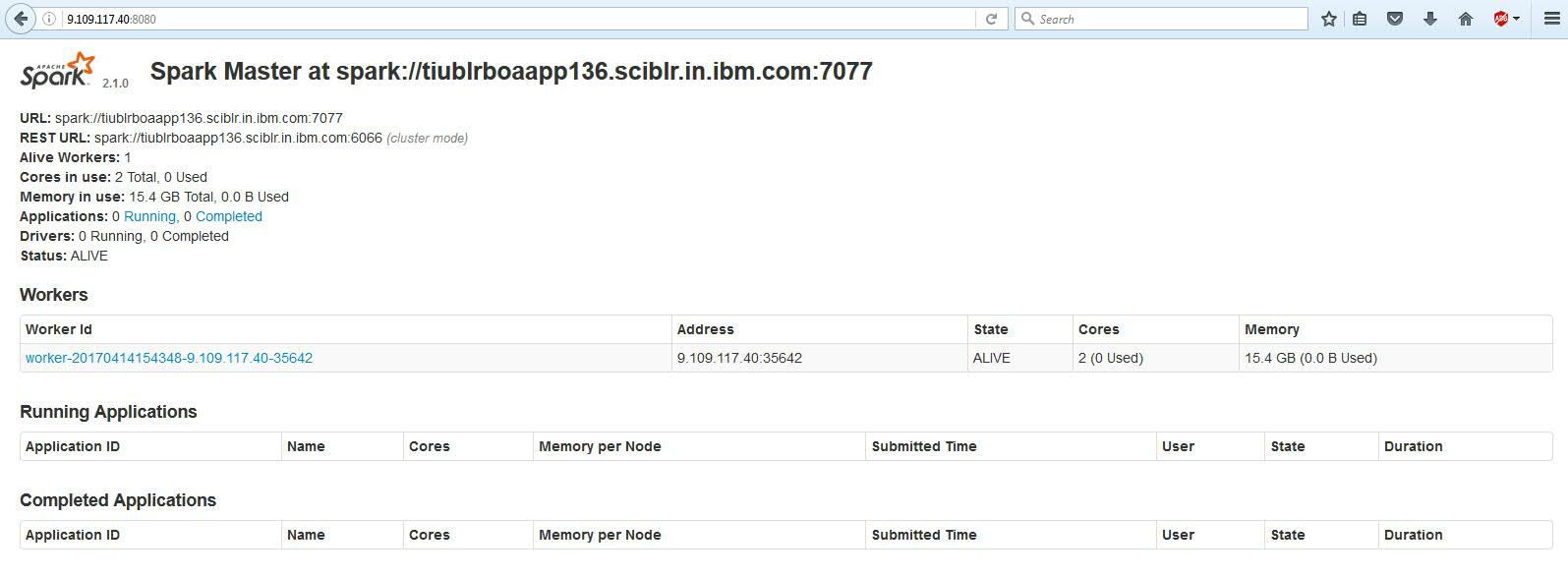
http://<hostname>:4040/jobs/



**sbin/start-all.sh** - Starts both a master and number of slaves.

Check the Master details from UI:

http://<hostaname>:8080/



## **SPARK Streaming**

<TBD>

# Cassandra Installation and Clustering

The below section covers the Installation of Cassandra on multiple Nodes and clustering the nodes on a distributed environment.

**Pre-requisites:**

* JDK 64 Bit
* Cassandra Installables

**Installation Steps:**

## **Step 1: Verifying Java Installation**

Java is the main prerequisite for Cassandra. Verify the existence of java in your system using “java -version”

# java -version

If java is not installed in your system, then follow the steps for installing java.

## **Step 2: Creating User Account**

Create a system user account on both nodes to use the Cassandra installation.

# useradd cassandra

# passwd cassandra

## **Step 3: Cassandra installation**

Download the latest version of Cassandra and copy the Cassandra tar file in the download folder (ex: /opt/downloads).

Change to cassandra user created for the Hadoop Installation in the previous section

#su – cassandra

Extract the Cassandra tar file

# tar xvf apache-cassandra-3.10-bin.tar.gz

Copy the Cassandra file to the respective directory

# mv apache-cassandra-3.10 /opt/Cassandra

Set Path and Home for Cassandra.

vi ~/.bashrc

Add the following for setting the PATH and CASSANDRA\_HOME

export CASSANDRA\_HOME=/opt/cassandra

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

Use the following command for sourcing the ~/.bashrc file.

# source ~/.bashrc

## **Step 4: Configure Cassandra Cluster**

Open the **cassandra.yaml** file, which will be available in the **config** directory of Cassandra.

Verify the following configurations. By default, these values will be set to the specified directories.

* data\_file\_directories “/var/lib/cassandra/data”
* commitlog\_directory “/var/lib/cassandra/commitlog”
* saved\_caches\_directory “/var/lib/cassandra/saved\_caches”

Create the two directories **/var/lib/cassandra** and **/var/log/cassandra** into which Cassandra writes its data and give permissions.

mkdir /var/lib/cassandra

mkdir /var/log/cassandra

chmod 777 /var/lib/cassandra

chmod 777 /var/log/cassandra

Properties to set:

* num\_tokens: *recommended value: 256*
* -seeds: *internal IP address of each seed node*

Seed nodes do not [bootstrap](https://docs.datastax.com/en/glossary/doc/glossary/.html), which is the process of a new node joining an existing cluster. For new clusters, the bootstrap process on seed nodes is skipped.

* listen\_address:
* If not set, Cassandra asks the system for the local address, the one associated with its hostname. In some cases Cassandra doesn't produce the correct address and you must specify the listen\_address.
* endpoint\_snitch: *name of snitch* . If you are changing snitches, see [Switching snitches](https://docs.datastax.com/en/cassandra/2.1/cassandra/operations/ops_switch_snitch.html).
* auto\_bootstrap: false (Add this setting **only** when initializing a fresh cluster with no data.)

Note: If the nodes in the cluster are identical in terms of disk layout, shared libraries, and so on, you can use the same copy of the cassandra.yaml file on all of them.

Example:

cluster\_name: 'MyCassandraCluster'

num\_tokens: 256

seed\_provider:

- class\_name: org.apache.cassandra.locator.SimpleSeedProvider

parameters:

- seeds: "110.82.155.0,110.82.155.3"

listen\_address:

rpc\_address: 110.82.155.0

endpoint\_snitch: GossipingPropertyFileSnitch

 In the cassandra-rackdc.properties file, assign the data center and rack names you determined in the Prerequisites. (Optional Step)

For example:

# indicate the rack and dc for this node

dc=DC1

rack=RAC1

Repeat the same steps for other nodes in different VM’s

## **Step 5: Starting the Nodes in the Cluster**

After you have installed and configured Cassandra on all nodes, start the seed nodes one at a time, and then start the rest of the nodes.

Note: If the node has restarted because of automatic restart, you must first stop the node and clear the data directories

Starting Cassandra Node:

# cd /opt/Cassandra/bin

# cassandra

## **Step 6: Verifying the Cassandra Cluster**

To check that the ring is up and running, run:

Package installations:

# nodetool status

Each node should be listed and it's status and state should be UN (Up Normal).

Datacenter: dc1

===============

Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

-- Address Load Tokens Owns Host ID Rack

UN 9.194.240.84 23.71 MB 256 ? 4cd9e3fc-f3a9-4f09-a2cb-23e20560282f rack1

UN 9.109.116.225 24.69 MB 256 ? faa774b2-46f8-4c79-b056-34c73cbfcf3b rack1

Note: Non-system keyspaces don't have the same replication settings, effective ownership information is meaningless

## **Step 7: Connecting to CQLSH**

Connect to a specific node using cqlsh

# cqlsh 9.109.116.225

Connected to MasterCluster at 9.109.116.225:9042.

[cqlsh 5.0.1 | Cassandra 3.0.9 | CQL spec 3.4.0 | Native protocol v4]

Use HELP for help.

cqlsh>

# Elastic Search Installation and Clustering

The below section covers the Installation of Elastic search on multiple Nodes of Elastic search Cluster (Master- Slave) on a distributed environment.

**Pre-requisites:**

Elasticsearch requires java to run. Install Oracle java or OpenJDK. Check the Elasticsearch support matrix to make sure it will work.

Elasticsearch support matrix: https://www.elastic.co/support/matrix#show\_jvm

**Installation Steps:**

## **Step 1: Verifying Java Installation**

Java is the main prerequisite for Elastic Search. Verify the existence of java in your system using “java -version”

# java -version

If java is not installed in your system, then follow the steps for installing java.

## **Step 2: Elastic Search installation**

first import the Elasticsearch GPG key with:

rpm --import http://packages.elastic.co/GPG-KEY-elasticsearch

Next, open a blank text file in your favourite text editor (be sure to run the editor as root, so that you have the necessary save permissions) and fill it with these lines:

name=Elasticsearch repository for 2.x packages

baseurl=http://packages.elastic.co/elasticsearch/2.x/centos

gpgcheck=1

gpgkey=http://packages.elastic.co/GPG-KEY-elasticsearch

enabled=1

Save the file as /etc/yum.repos.d/elasticsearch.repo

Next, install the Elasticsearch package with:

sudo yum -y install elasticsearch

## **Step 3: Configure and Start Elasticsearch**

We should tweak Elasticsearch a little and start it up.

Specifically, we want to prevent outside connections to the Elasticsearch HTTP API. To do this, open the file /etc/elasticsearch/elasticsearch.yml in an editor, set the below configuration. Under network section, change the “network.host” parameter with the IP address of your VM (or localhost). Then save the file and exit.

You can now start/stop/check status Elasticsearch using the following commands:

* sudo service elasticsearch start
* sudo service elasticsearch stop
* sudo service elasticsearch status

Verify the elasticsearch service by sending a HTTP request to port 9200. By default, elasticsearch run on port 9200.

*curl* <http://localhost:9200>

[root@pcs\_virtual\_01 ~]# curl http://localhost:9200

{

"name" : "es-master-01",

"cluster\_name" : "ynap\_cluster",

"cluster\_uuid" : "bu4WI6PbSmGhT69Ku64RXg",

"version" : {

"number" : "2.4.4",

"build\_hash" : "fcbb46dfd45562a9cf00c604b30849a6dec6b017",

"build\_timestamp" : "2017-01-03T11:33:16Z",

"build\_snapshot" : false,

"lucene\_version" : "5.5.2"

},

"tagline" : "You Know, for Search"

}

## **Step 4: Setup and Elastic Search Cluster**

The below steps are to setup a basic Elasticsearch cluster of 2 nodes (one Master node and one Data node). For this setup to work, as a prerequisite, you need two virtual machines with enough memory.

Do the following before we start configuring the server for elasticsearch.

1. Create 2 VMs (say the IPs are: 192.168.4.1, 192.168.4.2) installed with Elasticsearch.
2. Setting up the Master Node:

* All the elasticsearch configurations are present in elasticsearch.yml file, which is located in /etc/elasticsearch folder. Now, the elasticsearch.yml file has to be edited for the configuring the node as a Master node. The configuration file has many sections like cluster, node, paths etc.
* Under the cluster section, change the cluster name parameter.

cluster.name: es\_cluster

* Under node section, change the node name parameter and add other parameters as shown below.

node.name: es-master-01

node.master: true

node.data: false

* Under network section, change the “network.host” parameter with the IP address of your master node.

network. host: 192.168.4.1

* Under discover section add the following:

discovery.zen.ping.multicast.enabled: false

discovery.zen.ping.unicast.hosts: ["192.168.4.1", "192.168.4.2"]

The above parameters disable the multicast and send unicast message to the specified hosts.

1. Setting up the Data node:

* Follow all the steps we used to setup the Master node for the data node. Only while configuring the elasticsearch.yml file just uses the data given below. All the other steps are same for all the nodes.
* Under node section, add the following

node.name: es-data-01

node.client: false

node.data: true

If you do not specify a node name in the configuration, elasticsearch assigns a random name on every restart.

* Under network section, change the “network.host” parameter with the IP address of your data node.

network.host: 192.168.4.2

1. Once you configure all the three nodes, restart the elasticsearch service on all the three nodes.

sudo service Elasticsearch restart

1. Now you will have working elasticsearch cluster.

# Node JS Installation

The below section covers the Installation of Node JS instance in an environment.

## **Step 1: Install Node.js using Yum**

Install the Node.js using yum as shown below. This will also ask you to review and accept the Nodesource GPG key before installing nodejs.

# yum install nodejs

...

...

Public key for nodejs-0.10.40-1nodesource.el7.centos.x86\_64.rpm is not installed

nodejs-0.10.40-1nodesource.el7.centos.x86\_64.rpm | 4.5 MB 00:01

Retrieving key from file:///etc/pki/rpm-gpg/NODESOURCE-GPG-SIGNING-KEY-EL

Importing GPG key 0x34FA74DD:

Userid : "NodeSource <gpg-rpm@nodesource.com>"

Fingerprint: 2e55 207a 95d9 944b 0cc9 3261 5ddb e8d4 34fa 74dd

Package : nodesource-release-el7-1.noarch (installed)

From : /etc/pki/rpm-gpg/NODESOURCE-GPG-SIGNING-KEY-EL

Is this ok [y/N]: y

..

..

Installing : nodejs-0.10.40-1nodesource.el7.centos.x86\_64

Complete!

## **Step 2: Verify Node.js Installation**

Verify that the node.js and npm are successfully installed.

# node -v

v0.10.40

# npm version

{ http\_parser: '1.0',

node: '0.10.40',

v8: '3.14.5.9',

ares: '1.9.0-DEV',

uv: '0.10.36',

zlib: '1.2.8',

modules: '11',

openssl: '1.0.1p',

npm: '1.4.28' }

In this method, you’ll see that node and npm executable are installed under /usr/bin directory.

# whereis node

node: /usr/bin/node /usr/share/node /usr/share/man/man1/node.1.gz

# whereis npm

npm: /usr/bin/npm

# Reference

Hadoop - <https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm>

Spark - <https://www.tutorialspoint.com/apache_spark/apache_spark_installation.htm>

Cassandra - <https://docs.datastax.com/en/cassandra/3.0/cassandra/initialize/initSingleDS.html>

Elastic search - <https://developers.redhat.com/blog/2016/06/07/how-to-install-elastic-stack-elk-on-red-hat-enterprise-linux-rhel/>

Node JS - <http://www.thegeekstuff.com/2015/10/install-nodejs-npm-linux/>