BASIC HADOOP CONFIGURATION & SETUP

CLASS-III

Instructor: Palash Gupta

Re-Cap

Hello Everyone!

Last Week, we went through the working principle of HDFS and MapReduce.

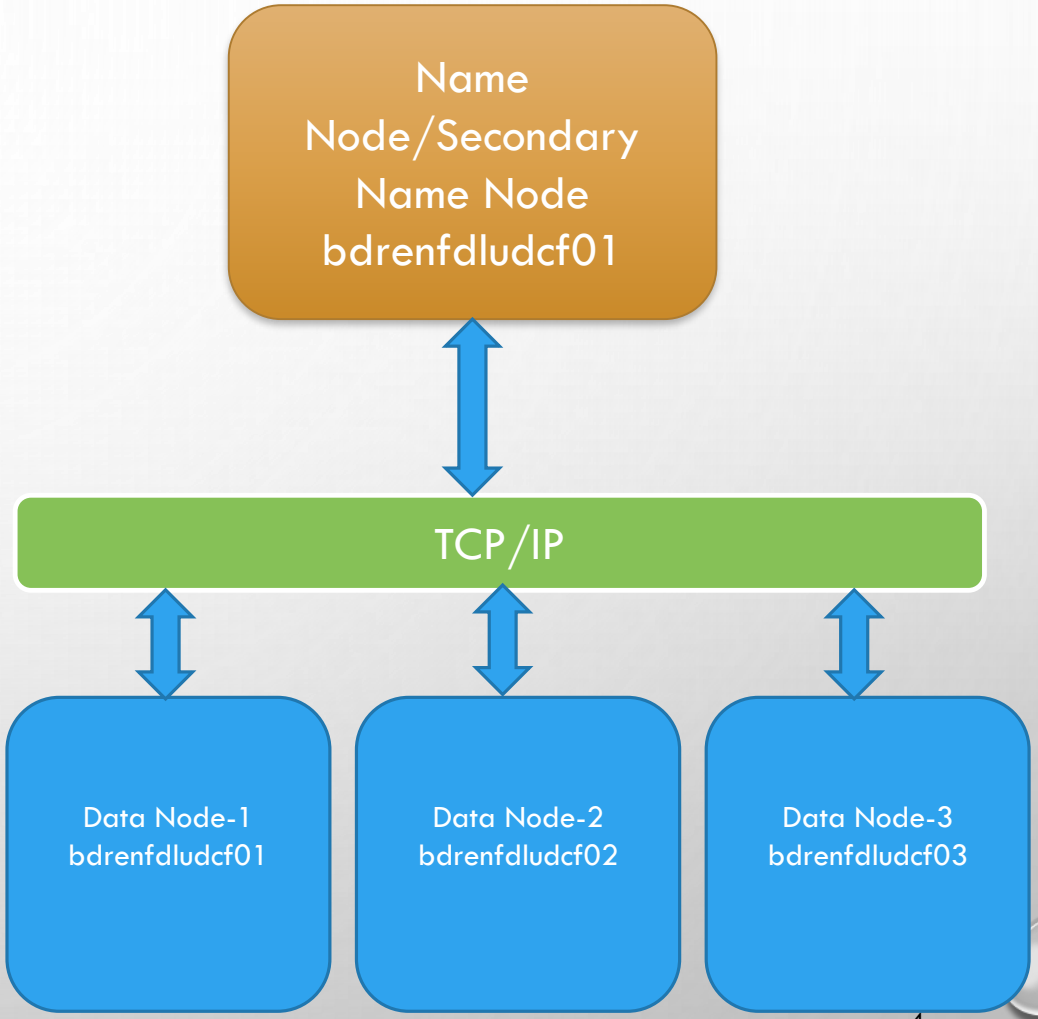
- HDFS Concept
- HDFS Architecture
- Introduction to Map Reduce
- Working Procedure of Map Reduce

What we are going to Cover today?

- Design a Hadoop HDFS Cluster
- Key Procedure Setting up a cluster
- Configuring a Cluster
- Basic Administration
- Reference Commands

Design a Hadoop HDFS Cluster

SN	Hostname	SSH Port	IP	Operating System	vCPU	vRAM	vHDD
1	bdrenfdludcf01	2200	103.28.121.5	Cent OS 7 64 bit	2	8 GB	100 GB
2	bdrenfdludcf02	2200	103.28.121.7	Cent OS 7 64 bit	2	8 GB	100 GB
3	bdrenfdludcf03	2200	103.28.121.30	Cent OS 7 64 bit	2	8 GB	100 GB



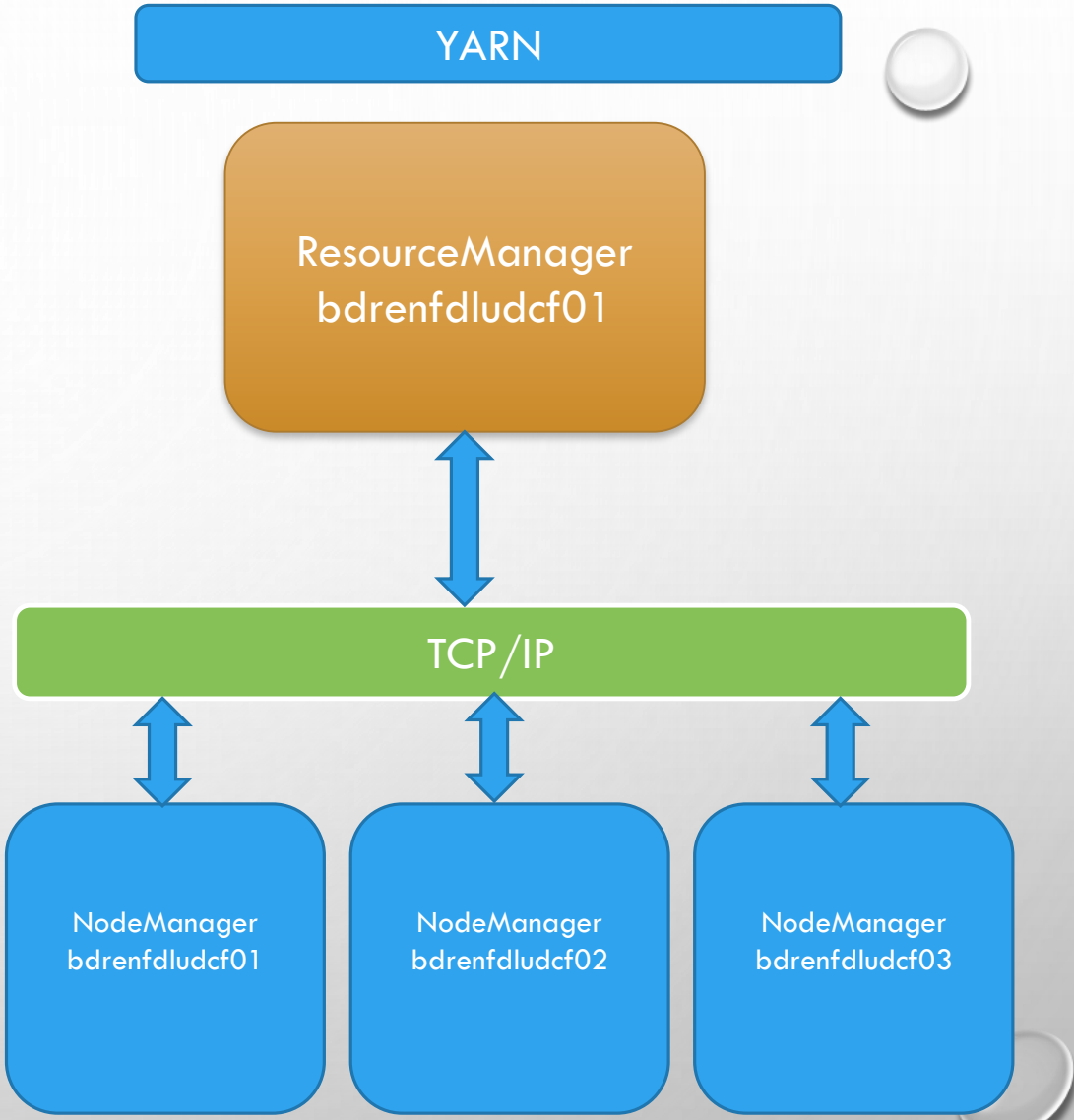
Design a Hadoop HDFS Cluster Contd.

- **Assumptions**
  - Replication Factor: 3
  - HDFS Storage Partition: /
  - SSH Port: 2200
  - Hadoop Binaries: Hadoop 2.7.1
  - JDK: 1.7.0
  - MapReduce Framework Name: YARN
  - YARN Resource Manager: NameNode
  - YARN Node Manager: All DataNodes
  - Name Node Directory: /usr/local/hadoop/hadoop_data/hdfs/namenode
  - Data Node Directory: /usr/local/hadoop/hadoop_data/hdfs/datanode

```
[root@bdrenfdludcf01 ~]# df -h
Filesystem                Size  Used Avail Use% Mounted on
devtmpfs                  3.9G     0  3.9G   0% /dev
tmpfs                     3.9G     0  3.9G   0% /dev/shm
tmpfs                     3.9G  257M  3.6G   7% /run
tmpfs                     3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/mapper/centos-root    50G  3.0G   48G   6% /
/dev/sda1                  42G  192M   42G   1% /boot
tmpfs                     783M     0  783M   0% /run/user/0
[root@bdrenfdludcf01 ~]#
```

Design a Hadoop HDFS Cluster Contd.

- **Apache Hadoop YARN** is the resource management and job scheduling technology in the open source Hadoop distributed processing framework. One of Apache Hadoop's core components, YARN is responsible for allocating system resources to the various applications running in a Hadoop cluster and scheduling tasks to be executed on different cluster nodes.
- **Why YARN as we have MapReduce?** YARN stands for Yet Another Resource Negotiator, but it's commonly referred to by the acronym alone. The addition of YARN significantly expanded Hadoop's potential uses. The original incarnation of Hadoop closely paired the Hadoop Distributed File System (HDFS) with the batch-oriented MapReduce programming framework and processing engine, which also functioned as the big data platform's resource manager and job scheduler. As a result, Hadoop 1.0 systems could only run MapReduce applications -- a limitation that Hadoop YARN eliminated. We will use Apache Spark over YARN.



Key Procedure Configuring a Cluster Contd.

SN	Step	Impacted Nodes	Type	Remarks
1	Know your machine resources & their IP and credential	All	Mandatory	We will use two hosts.
2	Configure Hostname	All	Mandatory	Naming convention is preferable.
3	Create a Repository	All	Optional	
4	Check & fix your date time	All	Mandatory	
5	Setup a NTP	All	Optional	
6	Stop Firewall	All	Mandatory	This is only for our Lab, not for a production system.
7	Step SELINUX	All	Mandatory	This is only for our Lab, not for a production system.
8	Create Hadoop User	All	Mandatory	
9	Create Password less login within cluster nodes	All	Mandatory	

Key Procedure Configuring a Cluster Contd.

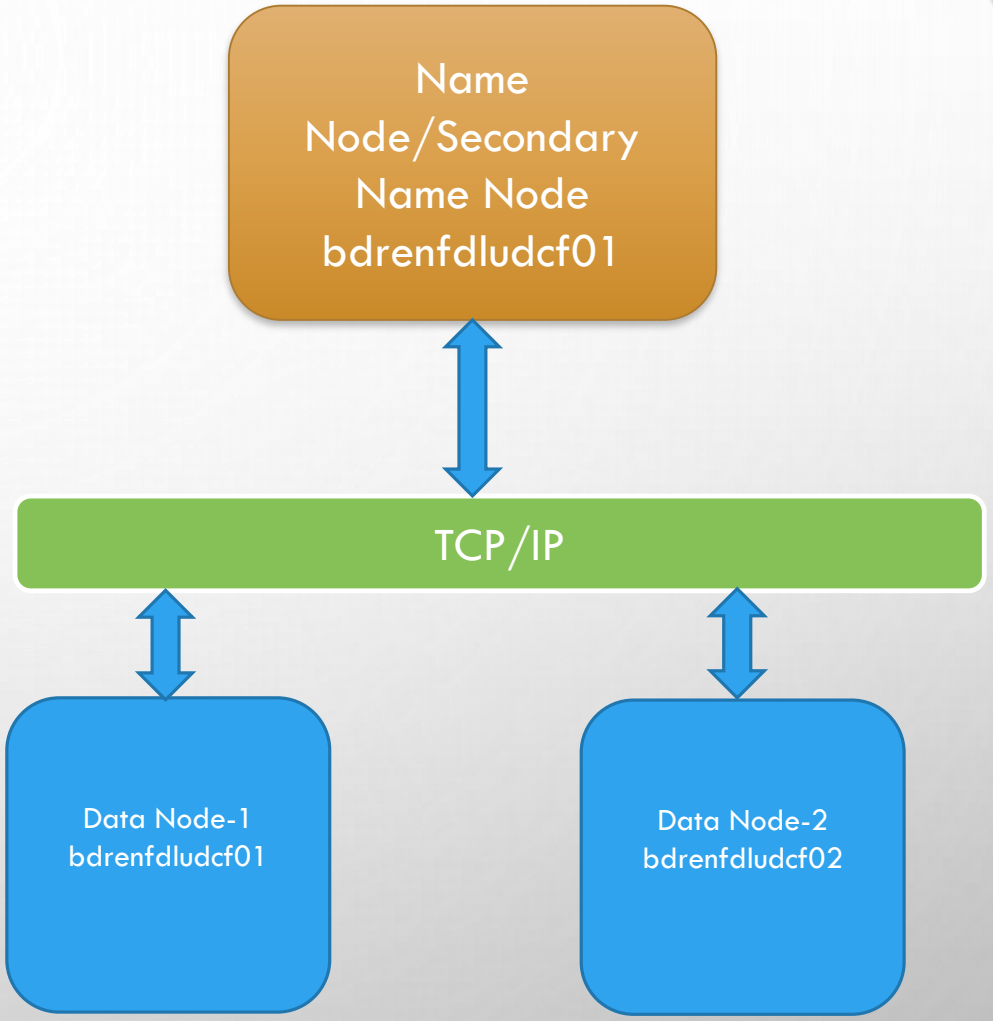
SN	Step	Impacted Nodes	Type	Remarks
10	Install targeted Java version	All	Mandatory	
Start Main Hadoop Installation				
1	Upload Hadoop Binaries	All	Mandatory	
2	Set Hadoop user environment	All	Mandatory	
3	Common Hadoop Configurations	All	Mandatory	•$HADOOP_CONF_DIR/hadoop-env.sh •$HADOOP_CONF_DIR/core-site.xml •$HADOOP_CONF_DIR/yarn-site.xml •$HADOOP_CONF_DIR/mapred-site.xml

Key Procedure Configuring a Cluster Contd.

SN	Step	Impacted Nodes	Type	Remarks
4	Name Node specific configuration	Name Node	Mandatory	/etc/hosts $HADOOP_CONF_DIR/hdfs-site.xml $HADOOP_CONF_DIR/masters $HADOOP_CONF_DIR/slaves
5	Change ownership of Hadoop Directory	Name Node	Mandatory	
6	Data Node specific configuration	Data Nodes	Mandatory	
7	Change ownership of Hadoop Directory	Data Nodes	Mandatory	
7	Format Name Node & then Start Hadoop Cluster (HDFS, YARN, Job History Server)	Name Node	Mandatory	Our main focus will be HDFS

Configuring a Cluster

- **Assumptions**
  - Replication Factor: 2
  - One Name Node & Two Data Nodes
  - HDFS Storage Partition: /
  - SSH Port: 2200
  - Hadoop Binaries: Hadoop 2.7.1
  - JDK: 1.7.0
  - MapReduce Framework Name: YARN
  - YARN Resource Manager: NameNode
  - YARN Node Manager: All DataNodes
  - Name Node Directory: /usr/local/hadoop/hadoop_data/hdfs/namenode
  - Data Node Directory: /usr/local/hadoop/hadoop_data/hdfs/datanode
  - Will Follow a Method of Procedure



- **We will explore benefits practically of using Hadoop in real life in late class**
- **We will also focus on Apache Spark for Distributed Computing Framework**

Basic Administration

SN	Topic	Command Syntax
1	Start and Stop HDFS	$HADOOP_HOME/sbin/start-dfs.sh HADOOP_HOME/sbin/stop-dfs.sh
2	Start and Stop Yarn	$HADOOP_HOME/sbin/start-yarn.sh $HADOOP_HOME/sbin/stop-yarn.sh
3	Start and Stop Job History	$HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver $HADOOP_HOME/sbin/mr-jobhistory-daemon.sh stop historyserver
4	Leaving Safe mode	$hadoop dfsadmin -safemode leave
5	Start or Stopping a specific data node manually	$cd /home/hadoop/hadoop/sbin $hadoop-daemon.sh stop datanode $hadoop-daemon.sh start datanode
6	Overall HDFS Status	$hadoop dfsadmin –report OR http://<name_node_IP>:50070/
7	Hadoop Balancer	$hadoop balancer
8	HDFS default block size	$hdfs getconf -confKey dfs.blocksize #(In Bytes)
9	Knowing a file block	$ hadoop fsck /mydir/mysecfile.log -blocks

Reference Command

SN	Topic	Command Syntax
1	Start and Stop HDFS	$HADOOP_HOME/sbin/start-dfs.sh HADOOP_HOME/sbin/stop-dfs.sh
2	Browsing HDFS	$hdfs dfs –ls / $hadoop fs –ls /
3	Putting a file into HDFS	$hdfs dfs –copyFromLocal my_file.txt / $hadoop fs –put my_file.txt /
4	Getting a file from HDFS	$hdfs dfs –copyToLocal /my_file.txt /home/hadoop $hadoop fs –get /my_file.txt /home/Hadoop
5	Removing a file from HDFS	$hdfs dfs -rm /user/my_file $Hadoop fs –rm /user/my_file
6	Overwriting a file in HDFS	$hadoop fs –put –f my_file.txt /
7	Append to a file	$echo "Line-to-add" | hdfs dfs -appendToFile - /my_file.txt
8	View a file	$hdfs dfs -cat /mydir/mysecfile.log

QUESTION & ANSWER

THANKS FOR ATTENDING THE CLASS & YOUR CO-OPERATION

References

- https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/
- http://elephantscale.com/
- https://www.hadoop.apache.org
- https://www.guru99.com/
- https://techvidvan.com/tutorials
- https://www.tutorialspoint.com/
- https://hadoop.apache.org/docs/r3.2.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
- https://searchdatamanagement.techtarget.com/definition/Apache-Hadoop-YARN-Yet-Another-Resource-Negotiator