Instructions to set up a BigTop Hadoop/Spark Cluster on IBM Power 8:

* Choose an installer platform (it could be your laptop)
* Download the script from github for single node or cluster in the link below

<https://github.com/OpenPOWER-BigData/solution-builder>

           or from the linux shell:

                git clone [**https://github.com/OpenPOWER-BigData/solution-builder.git**](https://github.com/OpenPOWER-BigData/solution-builder.git)

* Refer to the README.md file to build your Hadoop cluster

**Note:** The script assumed that the targeted user has background in Linux, hdfs, and running Java applications in such an environment.

**Steps:**

* **Create User Account on All Nodes**

**On Ubuntu:**

                sudo useradd bd\_user -U -G sudo -m

                sudo passwd bd\_user

* **Set passwordless login**

**On Installer system:**

                ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

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

                chmod 0600 ~/.ssh/authorized\_keys

From the installer node copy the public key to all nodes**:**

                ssh-copy-id -i ~/.ssh/id\_rsa.pub user@host

                ssh bd\_user@host

\*\*IMPORTANT - The username must match service's username defined in the solution definition file \*\*

* Ensure the root password is the same on all cluster nodes
* Ensure the username (i.e. bd\_user) has the same password on all cluster node.
* Ensure SSH daemon is running on all the nodes.
* Ensure the nodes are set for password-less SSH both ways (master<->slaves).
* It is not necessary to mount the HDDs as the solution-builder can do this.
* **Edit  Solution Definition file and customize it**
  + Edit **solution-builder/solutions/solution\_definition\_template**
  + **For example: To deploy Apache Bigtop 1.2 + optimized OPenJDK 1.8 for Power (** must run **build\_power\_opt\_openjdk** first **) with user: bd\_user** and **master node: cpobroad1** and**one data node: spocfire6**
  + The IP address of these nodes could be specified in this template instead of the hostnames, cpobroad1 and spocfire6.

# A Solution Definition File is a collection of services and relationships, designed to

# give you an entire deployment in one easy to use step. Defines the topology of the solution.

# Each line represents a service and is consist of comma-separated fields:

# 1) Service Name

# 2) Space separated list of additional required services to be installed on the same node

# 3) Target node's IP/Hostname

# 4) Service's user name (not root)

# 5) Configuration and connections values for the service, must be comma-separated.

# Example: Apache Bigtop Deployment

#################### Master node #############################################################

#### Hadoop master node includes namenode, resourcemanager, and spark-master services      ###

#### All Hadoop services have dependency on hadoop-client service                            ###

#### Next three arguments must be the hostname of the master node                          ###

##############################################################################################

#hadoop-namenode,hadoop-client,<masterIP>,bd\_user,<masterIP>,<masterIP>,<masterIP>

hadoop-namenode,hadoop-client,**cpobroad1**,bd\_user,**cpobroad1,cpobroad1,cpobroad1**

hadoop-resourcemanager,hadoop-client,**cpobroad1**,bd\_user,**cpobroad1,cpobroad1,cpobroad1**

spark-master,hadoop-client,**cpobroad1**,bd\_user,**cpobroad1,cpobroad1,cpobroad1**

#################### Worker Node 1 ###########################################################

#### Includes datanode, nodemanager, and spark worker services                             ###

#### Next three arguments must be the hostname of the master node                          ###

##############################################################################################

hadoop-datanode,hadoop-client,**spocfire6**,bd\_user,**cpobroad1,cpobroad1,cpobroad1**

hadoop-nodemanager,hadoop-client,**spocfire6**,bd\_user,**cpobroad1,cpobroad1,cpobroad1**

spark-worker,hadoop-client,**spocfire6**,bd\_user,**cpobroad1,cpobroad1,cpobroad1**

#################### Worker Node 2 #############################

#hadoop-datanode,hadoop-client,172.17.0.4,bd\_user,master,master,master

#hadoop-nodemanager,hadoop-client,172.17.0.4,bd\_user,master,master,master

#spark-worker,hadoop-client,172.17.0.4,bd\_user,master,master,master

#################### Worker Node 3 #############################

#hadoop-datanode,hadoop-client,172.17.0.5,bd\_user,master,master,master

#hadoop-nodemanager,hadoop-client,172.17.0.5,bd\_user,master,master,master

#spark-worker,hadoop-client,172.17.0.5,bd\_user,master,master,master

#################### Apache Zeppelin Serivce  #####################

#zeppelin,hadoop-client,172.17.0.6,bd\_user,master,master,master

#################### Apache Hive Service  #########################

* **Mapping the nodes**

                Edit /etc/hosts file on ALL nodes, specify the IP address of each of the nodes followed

                by their host names. e.g.,

**# sudo vim /etc/hosts**

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

                            192.168.1.1 **cpobroad1**

                            192.168.1.2 **spocfire6**

                            …

                            ...

                            …

* Change the hostname prompt appropriately based on your /etc/hosts file. Then logout and log back in.

$ sudo hostnamectl set-hostname <<your\_hostname>>

* **Mount Disks**

If it is needed for Hadoop to be installed across multiple disks, make a  copy of /services/hadoop-namenode/disk\_list.example

cd services/hadoop-namenode/

cp disk\_list.example disk\_list

sudo vim disk\_list

Add/remove the appropriate disks according to your cluster setup. (tip: use lsblk in the command console to check the number of physical drives you have)

sdb

sdc

sdd

sde

sdf

sdg

sdh

sdi

sdj

sdk

sdl

sdm

* **Deploy the solution and manage cluster as the exemple user “bd\_user” (Note: installer node can use a different installation username)**

bd\_user$ ./deploy\_solution --sd solutions/solution\_definition\_template  // deploy solution for the first time

bd\_user$ ./solution\_status --sd solutions/solution\_definition\_template // check which services are active

bd\_user $ ssh **bd\_user@cpobroad1** "bash -s" < test/hadoopTest.sh  // Test Hadoop Deployment using Terasort (takes longer)

bd\_user$ ssh **bd\_user@cpobroad1** "bash -s" < test/sparkTest.sh  // Test Spark Deployment

bd\_user$ ./stop\_solution --sd solutions/solution\_definition\_template  // Stop the solution

bd\_user$ ./restart\_solution --sd solutions/solution\_definition\_template  // Restart the solution

bd\_user$ ./remove\_solution --sd solutions/solution\_definition\_template  // Remove the solution

* **Create History\_log Directory ( only if the command is missing in spark solution-builder/services/spark-master/config.sh file)**
  + - Create history log directory for spark.eventLog.dir hdfs:///history\_logs as configured in /usr/lib/spark/conf/spark-default.conf

MasterNode$ hdfs dfs –mkdir /history\_logs

* **Addresses & Ports**
  + - HDFS web address : <http://your_masternode:50070> //Replace Local host with the master node IP address or FQDN in your yarn-site.xml file
    - Spark webUI : <http://spark_masternode:18080> //Replace Local host with the spark master node IP address or FQDN in your spark-default.conf file
    - Spark History Server : <http://spark_master_node:18082> //Replace Local host with the master node IP address or FQDN in your spark-default.conf file
* **Log files location**
* /var/log/hadoop-yarn/userlogs
* /var/log/spark
* /var/log/hadoop-hdfs
* /var/log/hadoop-mapreduce