SMOKE TEST DOCUMENT

CDH and HDP Smoke Test Cases

Date Prepared: August 2019

Document Information

Project Name	CDH and HDP Smoke Test Document		
Project Owner		**Document Version No**	1.0
Quality Review Method	By email/HP SharePoint		
Prepared By		**Preparation Date**	August 2019
Reviewed By	Refer to version history	**Review Date**	

Table of Contents

1 TESTING BASIC FUNCTIONALITY OF YARN

- Sample command to run the yarn test

  hadoop --config /usr/hdp/3.1.0.0-78/hadoop/conf jar /usr/hdp/3.1.0.0-78/hadoop-
  mapreduce/hadoop-mapreduce-examples-3.*.jar wordcount /user/ambari-qa/mapredsmokeinput
  /user/ambari-qa/mapredsmokeoutput

- Word Count will run on input data file in the below path

  /user/ambari-qa/mapredsmokeinput

- Output will be written to this file

  /user/ambari-qa/mapredsmokeoutput

2 TESTING BASIC FUNCTIONALITY OF HBASE

- Following command is only for kerberos

  kinit -kt {keytab path} {principal}

- Run in hbase shell

  > hbase shell

- Execute the below command to disable database

  disable 'smoketest'

- Execute the below command to drop database

  drop 'smoketest'

- Execute the below command to create table

  create 'smoketest','family'

- Execute the below command to insert data in to table

  put 'smoketest','row01','family:col01','id000a2001_date232319'

- Execute the below command to view the data in HTable

  scan 'smoketest'

  exit

3 TESTING BASIC FUNCTIONALITY OF HIVE

- Execute the below command only for Kerberos

  kinit -kt {keytab path} {principal}

- Execute the below command to run beeline

  > beeline -u "{hive jdbc url}"

- Execute the below command to drop database

  drop database if exists smoketest;

- Execute the below command to create database

  create database if not exists smoketest;

- Following command is to use the database

  use smoketest;

- Following is an example to insert the data in to table

  create table smoketable(number int, name string);

  insert into smoketable values(1,'test1');

  insert into smoketable values(2,'test2');

- Fetching the records from table

  select count(*) from smoketable;

- Execute the below command to drop table

  drop table smoketable;

  !quit;

4 TESTING BASIC FUNCTIONALITY OF IMPALA

- Execute the below command only for Kerberos

  kinit -kt {keytab path} {principal}

- Execute the below command to get impala shell

  > impala-shell -k -i {impala deamon url}

- Execute the below command to drop database

  drop database if exists smoketest;

- Execute the below command to create database

  create database if not exists smoketest;

- Execute the below command to use database

  use smoketest;

- Following is an example to insert the data in to table

  create table smoketable(number int, name string);

  insert into smoketable values(1,'test1');

  insert into smoketable values(2,'test2');

- Fetching the records from table

  select count(*) from smoketable;

- Execute the below command to drop table

  drop table smoketable;

  quit;

5 TESTING BASIC FUNCTIONALITY OF ZOOKEEPER

- Execute the below command only for Kerberos

  kinit -kt {keytab path} {principal}

- Execute zookeeper-client-server

  > zookeeper-client -server {zk-host:2181}

- Execute the below command to return the associated data in znode

  get /zk_smoketest

- Execute the below command to create data in znode

  create /zk_smoketest []

- Execute the below command to set data in specific path of znode

  set /zk_smoketest testdata

  get /zk_smoketest

- Execute the below command to delete the znode in specific path

  delete /zk_smoketest

  quit

6 TESTING BASIC FUNCTIONALITY OF PIG

- Execute the below command only for Kerberos

  kinit -kt {keytab path} {principal}

- Execute the below command to put hosts information into etc/passwd /tmp file

  hdfs dfs -put /etc/passwd /tmp

- Run pig shell

  A = load '/tmp/passwd' using PigStorage(':');

  B = foreach A generate \$0 as id;

- Execute the below command to stores output in separate file

  store B into 'pigsmoke.out';

  quit;

7 TESTING BASIC FUNCTIONALITY OF SPARK

- Execute the below command only for Kerberos

  kinit -kt {keytab path} {principal}

- Run PySpark shell

  > pyspark

- Source code

  >>> a = [1,2,3,4]

  >>> b = sc.parallelize(a)

  >>> c = b.map(lambda x: x*x)

  >>> c.collect()

- Output

  [1, 4, 9, 16]