**#!/bin/bash**

**<< details**

**This is a project to create a data pipeline, which does the following things.**

1. **Takes some file(s) from the local file system and copies them to a staging folder in HDFS.**
2. **Runs a MapReduce job on the files contained in the staging folder.**
3. **Takes the output files, renames them, and moves them to a target folder.**
4. **Deletes the temporary (MR-generated) output folder.**
5. **Moves the content of the staging folder to an archive folder in HDFS.**

**The MapReduce programs are for word count and were written in Python.**

**details**

**echo -e "\n\e[0;31m --------------------------------------------------------------------------------------------------------------------------------------------"**

**echo -e "\n\e[0;31m HDFS MAP REDUCE PROJECT SCRIPT TO COPY FILES INTO HDFS AND RUN MAP REDUCE"**

**echo -e "\n\n\e[0;32m Arguments can be passed while executing like: bash project.sh -i <input\_file\_path> -t <target\_file\_path> -a <archive\_file\_path> -m <mapper\_file\_path> -r <reducer\_file\_path> -p <target\_file\_prefix>"**

**while getopts i:t:a:m:r:p: flag**

**do**

**case "${flag}" in**

**i) inp=${OPTARG};;**

**t) target=${OPTARG};;**

**a) archive=${OPTARG};;**

**m) map=${OPTARG};;**

**r) red=${OPTARG};;**

**p) prefix=${OPTARG};;**

**esac**

**done**

**#Checking if the backup folder has been given as argument**

**echo -e "\n\e[0;31mChecking if the backup folder has been given as argument\n"**

**if [ -z "$archive" ]**

**then**

**echo -e "\nArchive file path not given so, default path has been chosen"**

**archive="jasim.mohamed/project/backup"**

**fi**

**#Checking if the given backup folder exists**

**hadoop fs -test -d $archive**

**if [ $? -eq 0 ]**

**then**

**echo -e "\nBackup Directory Exists"**

**else**

**echo -e "\nBackup Directory does not exist!! Default backup chosen"**

**archive="jasim.mohamed/project/backup"**

**fi**

**#Checking if target folder has been given as argument**

**echo -e "\n\e[0;31mChecking if target folder has been given as argument\n"**

**if [ -z "$target" ]**

**then**

**echo -e "\nTarget folder not specified, so going with the default target path"**

**target="jasim.mohamed/project/target"**

**fi**

**#Checking if target folder exists**

**hadoop fs -test -d $target**

**if [ $? -eq 0 ]**

**then**

**echo -e "\nTarget Directory Exists\n"**

**else**

**echo -e "\nTarget Directory does not exist!! Default Target chosen\n"**

**target="jasim.mohamed/project/target"**

**fi**

**#Checking if mapper file path has been given**

**echo -e "\n\e[0;31mChecking if mapper file path has been given\n"**

**if [ -z "$map" ]**

**then**

**echo -e "\nMapper file path not specified!! Default mapper chosen\n"**

**map="workspace/Mapreduce/mapper.py"**

**elif [ ! -e $map ]**

**then**

**echo -e "\nMapper file does not exist!! Default has been chosen\n"**

**map="workspace/Mapreduce/mapper.py"**

**fi**

**#Checking if reducer file path has been given**

**echo -e "\n\e[0;31mChecking if reducer file path has been given\n"**

**if [ -z "$red" ]**

**then**

**echo -e "\nMapper file path not specified!! Default mapper chosen\n"**

**red="workspace/Mapreduce/reducer.py"**

**elif [ ! -e $red ]**

**then**

**echo -e "\nMapper file does not exist!! Default has been chosen\n"**

**red="workspace/Mapreduce/reducer.py"**

**fi**

**#Checking if target file prefix was given**

**if [ -z "$prefix" ]**

**then**

**echo -e "\nTarget file prefix not given, Going with default\n"**

**val=1**

**elif [[ $prefix =~ ['!><$%^\:;+='] ]]**

**then**

**echo -e "\nPrefix does not support naming convention, Going with default\n"**

**val=1**

**else**

**val=2**

**fi**

**#Checking if the input file path has been given**

**if [ -z "$inp" ]**

**then**

**read -p "No argument given for input file path, Would you like to give now (y/n) " ans**

**case $ans in**

**y|Y|yes) read -p "Please give input file path : " new**

**inp=$new**

**;;**

**n|\*) echo -e "\nTaking default path"**

**inp="workspace/testfiles"**

**;;**

**esac**

**fi**

**if [ -d $inp ]**

**then**

**echo -e "\nGiven file path is a directory\n"**

**#Creating the staging folder**

**hdfs dfs -mkdir jasim.mohamed/project/staging**

**#For loop to copy files from the directory into staging folder**

**for i in $(ls $inp)**

**do**

**hadoop fs -put $inp/$i jasim.mohamed/project/staging**

**temp=$(date +"%H-%M-%S\_%d.%m.%Y")**

**suffxd=$(echo $i"\_"$temp)**

**hdfs dfs -cp jasim.mohamed/project/staging/$i $archive**

**hdfs dfs -mv $archive/$i jasim.mohamed/project/backup/$suffxd**

**echo -e "\n$i File was copied and archived"**

**done**

**# Running mapreduce on the txt files in Staging**

**for i in $(hdfs dfs -ls jasim.mohamed/project/staging | rev | cut -d "/" -f '1' | rev | tail -n +2)**

**do**

**if [ $val -eq 2 ]**

**then**

**temp=$prefix**

**elif [ $val -eq 1 ]**

**then**

**temp=$(date +"%H-%M-%S\_%d.%m.%Y")**

**fi**

**prfxd=$(echo $temp"\_"$i)**

**hadoop fs -cat jasim.mohamed/project/staging/$i | $map | sort -k1,1 | $red >> workspace/$prfxd**

**hadoop fs -put workspace/$prfxd $target**

**rm workspace/$prfxd**

**done**

**#Removing the staging folder**

**hdfs dfs -rm -r jasim.mohamed/project/staging**

**elif [ -f $inp ]**

**then**

**echo -e "\nPath given is of a single file!\n"**

**#Copy file to staging folder**

**hdfs dfs -mkdir jasim.mohamed/project/staging**

**hadoop fs -put $inp jasim.mohamed/project/staging**

**echo -e "\nFile has been copied to HDFS\n"**

**if [ $val -eq 2 ]**

**then**

**temp=$prefix**

**elif [ $val -eq 1 ]**

**then**

**temp=$(date +"%H-%M-%S\_%d.%m.%Y")**

**fi**

**temp2=$(date +"%H-%M-%S\_%d.%m.%Y")**

**suffxd=$(echo $inp"\_"$temp2)**

**prfxd=$(echo $temp"\_"$inp)**

**#Copy file to backup and rename with timestamp as suffix**

**hdfs dfs -cp jasim.mohamed/project/staging/$inp $archive**

**hdfs dfs -mv $archive/$inp jasim.mohamed/project/backup/$suffxd**

**#Running Mapreduce on the file and saving the output to target with prefix**

**hadoop fs -cat jasim.mohamed/project/staging/$inp | $map | sort -k1,1 | $red >> workspace/$prfxd**

**hadoop fs -put workspace/$prfxd $target**

**rm workspace/$prfxd**

**#Removing Staging Folder**

**hdfs dfs -rm -r jasim.mohamed/project/staging**

**else**

**echo -e "\n\e[0;31mInvalid input file!! Check your input"**

**fi**