# PHHD Cluster Build Document

Version 1.0

EMR 6.0

EMR 5.x

1. **Build EMR** (version 6.0) - Create Cloud Formation template using the below link

<https://github.build.ge.com/PowerDigital/Engineering-Data-Ingestion/blob/master/EMR_Scripts/src/main/cf_templates/MnD-intermediate-emr.yaml>

* Identify size, name and other details as needed
* For EMR 5.8 make appropriate change to template before build

ReleaseLabel:

Description: Must be a valid EMR release version

Default: emr-5.8.0

Type: String

* Update appropriate master, core and Task instances. Task instance count can be left 0 as we can add instance group later after build
* Build steps -> AWS console (Power-Services) -> Cloud Formation -> use the template and build



**Parameters** - Sample

|  |  |
| --- | --- |
| **Key** | **Value** |
| AdditionalMasterSecurityGroups | sg-29398352,sg-2c914c56 |
| CoreInstanceType | d2.4xlarge |
| EMRClusterName | MnD-Intermediate-EMR-Cluster-5-2002-2010 |
| KeyName | EMR-power-services |
| LogUri | s3://aws-logs-564772463473-us-east-1/elasticmapreduce/ |
| MasterInstanceType | m4.10xlarge |
| Name | MnD-Intermediate |
| NumberOfCoreInstances | 5 |
| NumberOfTaskInstances | 0 |
| ReleaseLabel | emr-5.8.0 |
| S3DataUri | s3://ge-engineering/ |
| SubnetID | subnet-cd377295 |
| TaskInstanceType | r4.16xlarge |
| env | prd |
| role | hadoop-dev-app |
| uai | UAI2005971 |

* Make sure the CF name and EMR cluster name to be unique (try to keep both same)
* Also the template has bootstrap to add users. To include any new user, update their key to the exe-user.sh file and upload to same location on S3

Name: Install Users

ScriptBootstrapAction:

Path: 's3://ge-engineering/powerservices-setup/mnd-prd/scripts/exe-user.sh'

* You can include another bootstrap to update SSM Agent as this helps with compliance

Name: ssm agent

ScriptBootstrapAction:

Path: 's3://ge-engineering/powerservices-setup/mnd-prd/scripts/emr-ssm-agent.sh'

1. PHHD DCI Bootstrap:

As soon as EMR cluster is ready login to master node and run the PHHD DCI bootstrap script using the link below

<https://github.build.ge.com/PowerDigital/Engineering-Data-Ingestion/blob/master/EMR_Scripts/src/main/emr/PHHD_DCI.sh>



1. Login to EMR, sudo to root (sudo su) and use screen (being in screen helps to get back to same screen incase of network disconnect)
2. create "phhd\_dci.sh" (copy content from above git path) (note: do not use /mnt/phhd/ folder for log, create log.txt in same folder as script)
3. Update/verify
   * hadoop fs -copyToLocal s3a://dev-pd-inbound/database/pwp\_dci\_master\_47.dump /$DIR/dumptmp/
   * pg\_restore -d pwp\_dci /$DIR/dumptmp/pwp\_dci\_master\_47.dump &

The DB dump file varies and need to be updated to latest or as needed (reach out to Prasanth/Vamshee)

Note: you can comment the ssm agent instance line if you add it as part of template bootstrap

1. In case of EMR 5.8 make the following changes in the script file
   * Add this line for EMR 5.8 -> aws configure set s3.signature\_version s3v4
   * Command Syntax for postgresql on EMR 5.8 changes as below

#postgresql-setup initdb >> log.txt

/etc/init.d/postgresql92 initdb >> log.txt

mkdir /$DIR/postgresql/

mv /var/lib/pgsql92/data /$DIR/postgresql/

ln -s /$DIR/postgresql/data /var/lib/pgsql92/data

mkdir /$DIR/dumptmp

chmod 777 /$DIR/dumptmp/

rm -rf /$DIR/dumptmp/pwp\_dci\_latest.\*

hadoop fs -copyToLocal s3a://dev-pd-inbound/database/pwp\_dci\_master\_44.dump /$DIR/dumptmp/

aws s3 cp s3://dci-phhd-setup-files/postgresql.conf .

mv postgresql.conf /var/lib/pgsql92/data

aws s3 cp s3://dci-phhd-setup-files/pg\_hba.conf .

mv pg\_hba.conf /var/lib/pgsql92/data/

#service postgresql start

/etc/init.d/postgresql92 startA

1. Make phhd\_dci.sh executable (chmod 755 phhd\_dci.sh)
2. Execute phhd\_dci.sh (./phhd\_dci.sh)

(Note: As this takes time to create DB, you can open new session for next steps)

1. Copy script and hadoop folder setup
   1. Got to /mnt/phhd/ folder & Create copyscript.sh script

( use month if needed else only year)

1. (#! /bin/bash

year=$1

{month=$2}

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/defcbac5-0158-411c-b45b-4251e9cd2e26/$year/$month/ --dest=hdfs:///PHHD/PWP\_Archives/defcbac5-0158-411c-b45b-4251e9cd2e26/$year/$month/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/dc3c0cdc-c086-47ba-a8c9-eb18cd2933ea/$year/$month/ --dest=hdfs:///PHHD/PWP\_Archives/dc3c0cdc-c086-47ba-a8c9-eb18cd2933ea/$year/$month/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/c06ff0f8-1c79-4b25-97dd-c3e95e8e4d11/$year/$month/ --dest=hdfs:///PHHD/PWP\_Archives/c06ff0f8-1c79-4b25-97dd-c3e95e8e4d11/$year/$month/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/25a5a333-ced9-4f78-b55c-c0ce9608adc1/$year/$month/ --dest=hdfs:///PHHD/PWP\_Archives/25a5a333-ced9-4f78-b55c-c0ce9608adc1/$year/$month/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/b8d3cf5d-3c3f-4407-b27b-f30e3f747514/$year/$month/ --dest=hdfs:///PHHD/PWP\_Archives/b8d3cf5d-3c3f-4407-b27b-f30e3f747514/$year/$month/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/a0cbbf5c-f703-4fb2-a3f8-8bfef6bf8e0f/$year/$month/ --dest=hdfs:///PHHD/PWP\_Archives/a0cbbf5c-f703-4fb2-a3f8-8bfef6bf8e0f/$year/$month/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/b450eb9c-dc17-4d2c-b5c5-3b19cf75b3b4/$year/$month/ --dest=hdfs:///PHHD/PWP\_Archives/b450eb9c-dc17-4d2c-b5c5-3b19cf75b3b4/$year/$month/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/e6a8d557-0f7f-47a3-875d-f9908c413631/$year/$month/ --dest=hdfs:///PHHD/PWP\_Archives/e6a8d557-0f7f-47a3-875d-f9908c413631/$year/$month/)

#! /bin/bash

year=$1

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/4ba6fec8-dd5e-419e-97fc-50852472563c/$year/ --dest=hdfs:///PHHD/PWP\_Archives/4ba6fec8-dd5e-419e-97fc-50852472563c/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/60969ca3-75a3-46ab-8ff9-62d6ac604b09/$year/ --dest=hdfs:///PHHD/PWP\_Archives/60969ca3-75a3-46ab-8ff9-62d6ac604b09/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/7e8db425-6548-4d34-adf8-d3fb2f7295b4/$year/ --dest=hdfs:///PHHD/PWP\_Archives/7e8db425-6548-4d34-adf8-d3fb2f7295b4/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/7fdb2dec-31d3-487b-b765-c61c3a11cd2d/$year/ --dest=hdfs:///PHHD/PWP\_Archives/7fdb2dec-31d3-487b-b765-c61c3a11cd2d/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/88dcf466-38b4-4994-821e-89c0da68c3f4/$year/ --dest=hdfs:///PHHD/PWP\_Archives/88dcf466-38b4-4994-821e-89c0da68c3f4/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/9205f129-a85f-4dde-9889-95da9d70c459/$year/ --dest=hdfs:///PHHD/PWP\_Archives/9205f129-a85f-4dde-9889-95da9d70c459/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/c43d153c-d6e9-4fa5-8f79-fd5a846250fe/$year/ --dest=hdfs:///PHHD/PWP\_Archives/c43d153c-d6e9-4fa5-8f79-fd5a846250fe/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/e52af41f-8526-4e9c-be6e-c7b9b4d14804/$year/ --dest=hdfs:///PHHD/PWP\_Archives/e52af41f-8526-4e9c-be6e-c7b9b4d14804/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/defcbac5-0158-411c-b45b-4251e9cd2e26/$year/ --dest=hdfs:///PHHD/PWP\_Archives/defcbac5-0158-411c-b45b-4251e9cd2e26/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/dc3c0cdc-c086-47ba-a8c9-eb18cd2933ea/$year/ --dest=hdfs:///PHHD/PWP\_Archives/dc3c0cdc-c086-47ba-a8c9-eb18cd2933ea/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/c06ff0f8-1c79-4b25-97dd-c3e95e8e4d11/$year/ --dest=hdfs:///PHHD/PWP\_Archives/c06ff0f8-1c79-4b25-97dd-c3e95e8e4d11/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/25a5a333-ced9-4f78-b55c-c0ce9608adc1/$year/ --dest=hdfs:///PHHD/PWP\_Archives/25a5a333-ced9-4f78-b55c-c0ce9608adc1/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/b8d3cf5d-3c3f-4407-b27b-f30e3f747514/$year/ --dest=hdfs:///PHHD/PWP\_Archives/b8d3cf5d-3c3f-4407-b27b-f30e3f747514/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/a0cbbf5c-f703-4fb2-a3f8-8bfef6bf8e0f/$year/ --dest=hdfs:///PHHD/PWP\_Archives/a0cbbf5c-f703-4fb2-a3f8-8bfef6bf8e0f/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/b450eb9c-dc17-4d2c-b5c5-3b19cf75b3b4/$year/ --dest=hdfs:///PHHD/PWP\_Archives/b450eb9c-dc17-4d2c-b5c5-3b19cf75b3b4/$year/

s3-dist-cp --src=s3://dev-pd-inbound/PHHD/PWP\_Archives/e6a8d557-0f7f-47a3-875d-f9908c413631/$year/ --dest=hdfs:///PHHD/PWP\_Archives/e6a8d557-0f7f-47a3-875d-f9908c413631/$year/

1. Make copyscript.sh executable.
2. Run these steps

sudo su hdfs

hadoop fs -mkdir /PHHD

hadoop fs -df -h /

hadoop fs -chown phhd:phhd /PHHD

hadoop fs -mkdir /user/phhd

hadoop fs -chown phhd:phhd /user/phhd

exit

sudo su phhd

hadoop fs -mkdir /tmp/parquetGen/

hadoop fs -mkdir /tmp/phhdFailedParquet/

1. Got to Screen mode, if user is not phhd then Sudo su phhd and execute -> copyscripts.sh {year}

(./copyscript 2010 or for year and month ./copyscript.sh 2019 09) (also can run all copy script simultaneously within another shell)

1. Create ssh tunnel (create tunnel to bastion host), setup foxyproxy and monitor through browser using cluster URL (<http://ip-10-228-147-109.ec2.internal:8088/cluster>)

Refer ink for foxyproxy setup - <https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-ssh-tunnel.html>

1. Verify
   1. How to verify Archive

hadoop fs -ls -R /PHHD/

1. Verify postgres DB

sudo su postgres

psql -d pwp\_dci

\dt dci.\* --> this lists db

select \* from dci.datastore\_hdfs\_table limit 5; --> select and verify datastore

select \* from dci.tag\_table limit 5;

1. Configure:
   1. Postgresql conf update (sudo root)

/mnt/postgresql/data/postgresql.conf

shared\_buffers = 50032MB # min 128kB

maintenance\_work\_mem = 20GB # min 1MB

1. Modify xml files (sudo root)

/etc/hadoop/conf

1. #Add property not to limit timeout to default 600ms

File -> mapred-site.xml

<property>

   <name>mapreduce.task.timeout</name>

   <value>0</value>

</property>

**Edit values for rest ->**

<property>

<name>mapreduce.map.java.opts</name>

<value>-Xmx7918m</value>

</property>

<property>

<name>mapreduce.reduce.java.opts</name>

<value>-Xmx9836m</value>

</property>

<property>

<name>mapreduce.map.memory.mb</name>

<value>9648</value>

</property>

<property>

<name>yarn.app.mapreduce.am.command-opts</name>

<value>-Xmx9837m</value>

</property>

<property>

<name>mapreduce.reduce.memory.mb</name>

<value>13296</value>

</property>

<property>

<name>yarn.app.mapreduce.am.resource.mb</name>

<value>17296</value>

</property>

1. #Reducing the hdfs thread count so that it doesn’t cause the jobs to fail

File - > hdfs-site.xml (add these to the end)

  <property>

<name>dfs.datanode.handler.count</name>

<value>20</value>

</property>

<property>

<name>dfs.client.file-block-storage-locations.num-threads</name>

<value>20</value>

</property>

<property>

<name>dfs.datanode.balance.max.concurrent.moves</name>

<value>24</value>

</property>

1. Run the following commands

sudo systemctl stop hadoop-hdfs-namenode

sudo systemctl start hadoop-hdfs-namenode

sudo systemctl stop hadoop-yarn-resourcemanager

sudo systemctl start hadoop-yarn-resourcemanager

service postgresql restart

1. Use phhd user (sudo su phhd)

Modify properties in /var/DataScienceCLIENT

1. App.properties file

Change the port to 8091, edit activemq IP (amqIP), phhdGeneratorThreads & dciIgnoredScopes Update JMS settings & PHHD Settings with relevant into

conversionDir=./conversion

dciEndpoint=http://localhost:8091/DataScienceREST/rest/DCI/

phhdEndpoint=http://localhost:8091/DataScienceREST/rest/PHHD/

hiveEndPoint=http://localhost:8091/DataScienceREST/rest/hive/

sparkEndpoint=http://localhost:8091/DataScienceREST/rest/spark/

hdfsEndpoint=http://localhost:8091/DataScienceREST/rest/HDFS/

userAgent=Mozilla/5.0

###JMS Settings###

amqIP=10.228.144.249

amqPort=61616

amqPrefetch=1

phhdParquetGenQueue=PHHD\_Parquet\_Gen\_2010\_2014

parquetMergeQueue=Merge\_Parquet\_2010\_2014

phhdErrorQueue=PHHD\_Error\_2010\_2014

###Thread Settings###

parquetCombinerThreads=10

phhdGeneratorThreads=40

dciTagThreads=10

#Allowed=PWPGALPHIUS02

dciIgnoredScopes=PWPGALPHIAR01,PWPGALPHIIN01,PWPGALPHIIN02,PWPGALPHIUS01,PWPGALPHIUS02

batchedAssetsPerScope=30

###PHHD Settings### replace the whole section below

finalBaseDir=/tmp/finalParquet/

phhdOutputBase=/tmp/parquetGen/

phhdFailureBase=/tmp/phhdFailedParquet/

conversionIntervalHours=240

maxRetries=120

dciTagThreads=6

dciIgnoredScopes=testing

numCollection=5000

enforceScope=false

jobPrefix=job\_

simultaneousConversion=1

Note: the jobPrefix changes according to the version of EMR/Application requirement (EMR 6.0 uses application\_)

1. Timeline.txt --> edited the files for appropriate time  (duration of the data)

2010-01-01 00:00:00.000;2015-01-01 00:00:00.000

1. PHHD Tomcat
   1. Modify Config file eh.ini

/var/PHHD\_Tomcat/apache-tomcat-7.0.70/PHHD

Update/Replace the master node IP

Sample :

fs.defaultFS=hdfs://10.228.146.166:8020

mapred.jobtracker.address=http://10.228.146.166:20888/

#yarn.resourcemanager.address=10.228.146.88:8032

yarn.resourcemanager.address=10.228.146.166:8032

jobtracker.baseurl=http://10.228.146.166:20888/

namenode.browseurl=http://10.228.146.166:50070/

fs.default.name=hdfs://10.228.146.166:8020/

mapred.job.tracker=10.228.146.166:20888

1. /var/PHHD\_Tomcat/apache-tomcat-7.0.70/bin --> start the tomcat here

./startup.sh

Monitor catalina.out in ../logs/

if you see port in use error, restart the tomcat (shutdown and start)

1. PDS Tomcat
   1. Change properties file db.properties

/var/PDS\_Tomcat/apache-tomcat-7.0.70/appProps/

jobStatusUrl= to masetrnode IP

database to localhost or edge node IP (if postgres is running on different edge server)

Also verify --> dbUrl=jdbc:postgresql://localhost:5432/pwp\_dci

Sample :

dbUrl=jdbc:postgresql://ip-10-228-146-166.ec2.internal:5432/pwp\_dci

jobStatusUrl=http://10.228.146.166:8088/ws/v1/cluster/apps/

1. spark-defaults.conf

/var/PDS\_Tomcat/apache-tomcat-7.0.70/appProps/

spark.yarn.historyServer.address to master node ip

Sample :

spark.yarn.historyServer.address 10.228.146.166:18080

1. Copy jar under /var/DataScienceCLIENT/ from s3://ge-engineering/vamsheetest/DataScienceClient-jar-with-dependencies.jar

(aws s3 cp s3://ge-engineering/vamsheetest/DataScienceClient-jar-with-dependencies.jar .)

Note: Ignore this step if you are using EMR 5.8 and use the existing old jar

1. /var/PDS\_Tomcat/apache-tomcat-7.0.70/bin --> start the tomcat here

./startup.sh

Monitor catalina.out in ../logs/

if you see port in use error, restart the tomcat (shutdown and start)

1. To add instance task group

In AWS console -> emr -> Cluster created (MnD-Intermediate) -> verify cluster details -> hardware -> Add task instance group --> 30 (or any number of instances as needed) spot instances with Set max price /instance /hr to $ (any recommend rate or max on demand)

* Also add 300 to 500 GB of EBS Volume

Machine generated alternative text:
Cluster: MnD-lntermediate Waiting Cluster ready to run steps. 
Summary Application user interfaces Monitoring 
Hardware Configurations 
Add task instance group 
Instance groups 
Filter 
Clone 
Terminate 
No scaling enabled 
AWS CLI export 
Events 
Steps 
Add task instance group 
Name 
Task 
EBS-Optimized instance 
Volume type O 
General Purpose SSD (GP2) 
Add ESS volumes 
EC2 instance type 
r5_ 16xlarge 
Instance count 
30 
Request Spot O 
Size (GiB) O 
500 
Min: 1 GiB; Max: 16384 GiB 
'OPS O 
1500/3000 
Bootstrap actions 
$ 1.75 
Set max price per instance/hr v 
O 
Volumes per instance 
Cancel 
x 
x 
Add 
Clus 

Recommended R5 for EMR 6.0 and R4 for EMR 5.8

1. To initiate tagging messages ( This step is during initial setup, ignore if you are restarting the phhd jobs even on new cluster)

Go to folder (as phhd user) /var/DataScienceCLIENT

1. rm taglist.txt from mastertags folder -- remove the folder
2. sudo su postgres
3. bash-4.2$ psql -d pwp\_dci
4. db=>\o /tmp/taglist.txt
5. db=>select "Scope"::text||'::'||"TagName"::text from dci.tag\_table where "Scope" in('pwpgalgaphiar01','pwpgalgaphiin01','pwpgalgaphius01','pwpgalgaphiin02','pwpgalgaphius02','PWPGALGAPHIIN03') and "TagId" in(select distinct "TagId" from dci.tag\_datastore\_table where "StartTime" > '1/3/2017');
6. move the /tmp/taglist.txt tp mastertags
7. Run command -> ./genMQmsgs.sh -- start the message

Sample at the end :

Message count: 1264530 Total Assets: 35587

This step adds the Parquet names in the ActiveMQ and updates the messages. <http://10.228.144.249:8161/admin/queues.jsp>

Machine generated alternative text:
18416 
20760 
2344 
Browse Active Consumers 
Active Producers 
atom rss 
Send To Purge Delete 

1. To star the phhd jobs

./genPHHDjobs.sh -- start jobs

You can monitor Status using cluster URL - <http://ip-10-228-147-42.ec2.internal:8088/cluster>

You can monitor the progress through ActiveMQ admin url - <http://10.228.144.249:8161/admin/queues.jsp>

1. When the phhd jobs are complete or incase you need to stop due to space issue, you need copy the parquet files to S3 location with below command

s3-dist-cp --src=hdfs:///tmp/parquetGen/ --dest=s3://pd-dev-phhd-output/tmp/parquetGen/ --s3ServerSideEncryption

1. Taking the /tmp/parquetGen backup

sudo su hadoop

hdfs dfs -ls /bckup/parquetGen (if folder exists remove the parquetGen folder and recreate)

hdfs dfs -rm -R /bckup/parquetGen

hdfs dfs -mkdir /bckup/parquetGen-2020\_C3\_Merge\_Parquet\_New

hdfs dfs -mv /tmp/parquetGen/ /bckup/parquetGen-2020\_C3\_Merge\_Parquet\_New

hdfs dfs -ls /bckup/parquetGen-2020\_C3\_Merge\_Parquet\_New

hdfs dfs -mkdir /tmp/parquetGen

hadoop fs -chown phhd:hadoop /tmp/parquetGen

## Outage/Restart Steps

Activities on any Outage or to just restart all process

1. Login to EMR Master
2. Sudo su (root)
3. Restart Postgres

(steps for EMR 5.8)

* 1. /etc/init.d/postgresql92 status
  2. /etc/init.d/postgresql92 start

(use stop first if you need to restart)

(steps for EMR 6.0)

1. systemctl stop postgresql
2. systemctl start postgresql

1. Screen (if there was outage/reboot of server, old screens will be lost)
2. Sudo su phhd
3. Restart Tomcat
   1. cd /var/PHHD\_Tomcat/apache-tomcat-7.0.70/bin
   2. ./startup.sh
   3. tail -f ../logs/catalina.out (until you see service up)
   4. cd /var/PDS\_Tomcat/apache-tomcat-7.0.70/bin
   5. ./startup.sh
   6. tail -f ../logs/catalina.out (until you see service up)
4. Restart phhd jobs
   1. cd /var/DataScienceCLIENT/
   2. ./genPHHDjobs.sh

* In case of unhealthy nodes, logs files need to be cleaned

Through script run has Hadoop user (file path may vary for 5.x)

<https://github.build.ge.com/PowerDigital/Engineering-Data-Ingestion/blob/master/EMR_Scripts/src/main/emr/cleanup_unhealthy_nodes.sh>

#!/usr/bin/env bash

export HADOOP\_CLIENT\_OPTS="-Xmx6g"

for x in `hdfs dfs -ls /var/log/spark/apps/ | awk '!/inprogress/{print $8}'`; do

echo Deleting $x

hdfs dfs -rm -r -skipTrash $x;

done

for x in `hdfs dfs -ls /var/log/hadoop-yarn/apps/phhd/logs-tfile/ | awk '!/inprogress/{print $8}'`; do

echo Deleting $x

hdfs dfs -rm -r -skipTrash $x;

done

* Unhealthy node even after cleaning up then

1. Stop jobs
2. Yarn restart –

5.x

sudo /sbin/stop hadoop-yarn-resourcemanager

sudo /sbin/start hadoop-yarn-resourcemanager

6.0

sudo systemctl stop hadoop-yarn-resourcemanager

sudo systemctl start hadoop-yarn-resourcemanager

1. S3cp -> s3-dist-cp --src=hdfs:///tmp/parquetGen/ --dest=s3://pd-dev-phhd-output/tmp/parquetGen/ --s3ServerSideEncryption
2. Cleanup (Delete /tmp/parquetGen)

hadoop fs -rm -r -h /tmp/parquetGen/

1. Re-Start jobs

* You can verify hdfs report

To get HDFS report command: hdfs dfsadmin -report

* You can also run balancer in separate screen (long running process) as Hadoop user

Command: hdfs balancer

* Options for Balancer

[HDFS Balancer (2): Configurations & CLI Options - Cloudera Community](https://community.cloudera.com/t5/Community-Articles/HDFS-Balancer-2-Configurations-CLI-Options/ta-p/246687)

[HDFS Balancers | 6.3.x | Cloudera Documentation](https://docs.cloudera.com/documentation/enterprise/6/latest/topics/admin_hdfs_balancer.html)

Add to map-site.xml under /etc/hadoop/conf  (value 3 to 4 times of core nodes)

<property>

<name>dfs.datanode.balance.max.concurrent.moves</name>

<value>36</value>

</property>

dfsadmin -setBalancerBandwidth <bandwidth in bytes per second>

hdfs dfsadmin -setBalancerBandwidth 10737418240

dfs.balancer.moverThreads

hdfs balancer -Ddfs.balancer.moverThreads=20000

* In case of server reboot or namenode restarts, Hadoop will be in safe mode

You can wait for it leave safe mode or urgent case you can force leave safe more

hadoop dfsadmin -safemode leave

* In case of S3Distcp crash due to Out of memory error (GC overhead issue while S3cp)

Try to increase size

cd /etc/hadoop/conf/hadoop-env.sh

export HADOOP\_CLASSPATH="$HADOOP\_CLASSPATH:/usr/share/aws/emr/security/conf:/usr/share/aws/emr/security/lib/\*"

export HADOOP\_OPTS="-XX:NewRatio=12 -Xmx8192m -XX:MaxHeapFreeRatio=40 -XX:MinHeapFreeRatio=15 -XX:+UseGCOverheadLimit -XX:+UseConcMarkSweepGC"

export HADOOP\_OPTS="$HADOOP\_OPTS -server -XX:OnOutOfMemoryError='kill -9 %p'"

export HADOOP\_NAMENODE\_HEAPSIZE=8088

export HADOOP\_DATANODE\_HEAPSIZE=5727

export HADOOP\_JOB\_HISTORYSERVER\_HEAPSIZE=5365

In case of too many open files error, increase file size

To verify > ulimit -a

To increase temporary for one user > ulimit -n <size>

To increase for all user permanently

Update file /etc/security/limits.conf

\* hard nofile 65536

\* soft nofile 65536

# Final Parquet

**Setup – one time/first time server setup -> can be reused for final parquet**

sudo su

sudo su hdfs

hadoop fs -mkdir /user/phhd

hadoop fs -chown phhd:phhd /user/phhd

exit

adduser phhd

chmod 777 /mnt

cd /var

ln -s /mnt/phhd\_spark/Clients /var/Clients

ln -s /mnt/phhd\_spark/PDS\_Tomcat /var/PDS\_Tomcat

sudo su phhd

cd /mnt

mkdir phhd\_spark

cd phhd\_spark

aws s3 cp --recursive s3://ge-engineering/vamsheetest/NewSparkbkup/ .

unzip Clients.zip

unzip PDS\_Tomcat.zip

rm \*.zip

cd /var/PDS\_Tomcat/apache-tomcat-7.0.70/appProps

rm \*.xml

ln -s /etc/hadoop/conf/hdfs-site.xml hdfs-site.xml

ln -s /etc/hadoop/conf/core-site.xml core-site.xml

ln -s /etc/hadoop/conf/mapred-site.xml mapred-site.xml

ln -s /etc/hadoop/conf/yarn-site.xml yarn-site.xml

vi spark-defaults.conf

Update: spark.yarn.historyServer.address ip-10-228-146-29.ec2.internal:18080 { Hostname:18080}

vi db.properties

Update: phhds3Bucket=s3a://pd-dev-phhd-output (s3a://BucketName/FolderName)

Update: jobStatusUrl=http://10.228.46.29:8088/ws/v1/cluster/apps/ (<http://HostIP:8088/ws/v1/cluster/apps/>)

cd /var/PDS\_Tomcat/apache-tomcat-7.0.70/bin

rm \*.log

./startup.sh

exit

tail -50f /var/PDS\_Tomcat/apache-tomcat-7.0.70/logs/catalina.out

**Final parquet process (Cluster 1 sample) -> modify for next cluster final parquet**

Login Final Praquet server

Sudo su

screen

sudo su phhd

cd /var/Clients/ClientApp

update the app.properties

amqIP=10.228.144.249 {ActiveMQ IP}

parquetMergeQueue=Merge\_Parquet\_sep19\_dec19 {Cluster for FinalParquet}

finalBaseDir=s3a://pd-dev-ge-pqt-derived-temporary/,s3a://pd-dev-ge-pqt-osm-temporary/,s3a://pd-dev-ge-pqt-unknown/,s3a://gep-pr-pqt-derived-us-east-2/,s3a://gep-pr-pqt-osm-us-east-2/

parquetCombinerThreads=100

jobPrefix=application\_ (job\_ for emr 5 and older)

rm -f ParquetMerge\*

Verify - DataScienceClient-jar-with-dependencies.jar {if EMR 5 leave old version)

Copy jar from s3://ge-engineering/vamsheetest/DataScienceClient-jar-with-dependencies.jar

(aws s3 cp s3://ge-engineering/vamsheetest/DataScienceClient-jar-with-dependencies.jar .)

Update -> generatePDSParquet.sh

Run -> ./generatePDSParquet.sh

-----

* ActiveMQ.DLQ - Reprocess failed messages

Goto ActiveMQ server

/mnt/apache-activemq-5.14.0/bin

./activemq browse --amqurl tcp://localhost:61616 ActiveMQ.DLQ > SepDec19FinalParquet.txt

cat SepDec19FinalParquet.txt |grep JMS\_BODY\_FIELD:JMSText > SepDec19FinalParquet\_jmsbody.txt

sed -ie 's/JMS\_BODY\_FIELD:JMSText = //g' SepDec19FinalParquet\_jmsbody.txt

aws s3 cp SepDec19FinalParquet\_jmsbody.txt s3://ge-engineering/vamsheetest/queues/

Next Steps - Back to Final Parqet Cluster Master node

Sudo su phhd

cd /var/Clients/InsertMessages

Download file from S3 -> aws s3 cp s3://ge-engineering/vamsheetest/queues/SepDec19FinalParquet\_jmsbody.txt .

Update file -> insertScript.sh

java -cp DataScienceCLIENT-jar-with-dependencies.jar com.ge.pw.driver.InsertMessages messages.txt 10.228.144.249 61616 "$Queue\_Name" true > file

mv SepDec19FinalParquet\_jmsbody.txt messages.txt

Run -> ./insertScript.sh Merge\_Parquet\_sep19\_dec19

Verify queue in ActiveMQ site

Delete ActiveMQ.DLQ from ActiveMQ

Stop and start ./generatePDSParquet.sh

--

Redo reprocess failed messages

again ActiveMQ if needed

--

* Processing Error Queue

ActiveMQ server -> Pull Error messages

/mnt/apache-activemq-5.14.0/bin

./activemq browse --amqurl tcp://localhost:61616 PHHD\_Error\_sep19\_dec19 > PHHD\_Error\_sep19\_dec19.txt

aws s3 cp PHHD\_Error\_sep19\_dec19.txt s3://ge-engineering/vamsheetest/queues/

Login to FinalParquet instance for insert:

Cd /mnt/phhd\_spark/Clients/InsertMessages

aws s3 cp s3://ge-engineering/vamsheetest/queues/PHHD\_Error\_sep19\_dec19.txt .

sed '/No IHA files are found to contain time series data specified in the queries/{N;N;N;d;}' PHHD\_Error\_sep19\_dec19.txt > PHHD\_Error\_sep19\_dec19\_Filtered.txt

cat PHHD\_Error\_sep19\_dec19\_Filtered.txt | grep JMS\_BODY\_FIELD:JMSText > PHHD\_Error\_sep19\_dec19\_Filtered\_JMSText.txt

sed -ie 's/JMS\_BODY\_FIELD:JMSText = //g' PHHD\_Error\_sep19\_dec19\_Filtered\_JMSText.txt

Update file -> insertScript.sh

java -cp DataScienceCLIENT-jar-with-dependencies.jar com.ge.pw.driver.InsertMessages messages.txt 10.228.144.249 61616 "$Queue\_Name" true > file

--update the ip of MQ in the above file

cp PHHD\_Error\_sep19\_dec19\_Filtered\_JMSText.txt messages.txt

Run -> ./insertScript.sh PHHD\_Parquet\_Gen\_sep19\_dec19

Verify queue in ActiveMQ site

---

* You can also split a queue

Split (PHHD\_Parquet\_Gen\_jan20\_may20\_split2)

./activemq browse --amqurl tcp://localhost:61616 PHHD\_Parquet\_Gen\_jan20\_may20\_split2 > PHHD\_Parquet\_Gen\_jan20\_may20\_split2.txt

cat PHHD\_Parquet\_Gen\_jan20\_may20\_split2.txt | grep JMS\_BODY\_FIELD:JMSText > PHHD\_Parquet\_Gen\_jan20\_may20\_split2\_jmsbody.txt

sed -ie 's/JMS\_BODY\_FIELD:JMSText = //g' PHHD\_Parquet\_Gen\_jan20\_may20\_split2\_jmsbody.txt

aws s3 cp PHHD\_Parquet\_Gen\_jan20\_may20\_split2\_jmsbody.txt s3://ge-engineering/vamsheetest/queues/

Login to server with insert messages ( FinalParquet)

cd /var/Clients/InsertMessages

Download file from S3 -> aws s3 cp s3://ge-engineering/vamsheetest/queues/PHHD\_Parquet\_Gen\_jan20\_may20\_split2\_jmsbody.txt .

Split files as needed (split -l [lines] file name --verbose) - (split -l 306855 PHHD\_Parquet\_Gen\_2002\_2010\_jms.txt --verbose)

Rename files mv xaa PHHD\_Parquet\_Gen\_jan20\_may20\_split1.txt

mv xab PHHD\_Parquet\_Gen\_jan20\_may20\_split2.txt

Update file -> insertScript.sh

java -cp DataScienceCLIENT-jar-with-dependencies.jar com.ge.pw.driver.InsertMessages messages.txt 10.228.145.117 61616 "$Queue\_Name" true > file

1. For first file : cp PHHD\_Parquet\_Gen\_jan20\_may20\_split1.txt messages.txt
2. Run -> ./insertScript.sh PHHD\_Parquet\_Gen\_jan20\_may20\_split1

Verify queue in ActiveMQ site

Repeat a and b for split2

**Error In Intermediate**

Caused by: Output directory hdfs://ip-10-228-145-205.ec2.internal:8020/tmp/parquetGen/2013-01-01-270T450-0032 already exists

2021-05-20 02:43:56,554 WARN conf.Configuration: file:/var/PDS\_Tomcat/apache-tomcat-7.0.70/appProps/core-site.xml:an attempt to override final parameter: fs.s3.buffer.dir; Ignoring.

May 20, 2021 2:43:56 AM com.sun.jersey.spi.container.ContainerResponse mapMappableContainerException

SEVERE: The RuntimeException could not be mapped to a response, re-throwing to the HTTP container

java.lang.IllegalArgumentException: Invalid rule: /L

check the configuration of below item

cd /var/PDS\_Tomcat/apache-tomcat-7.0.70/appProps -- edit vi db.properties

--dbUrl=jdbc:postgresql://loclahost:5432/pwp\_dci chk this in "pwp\_dci" db.properties

Check for Final Parquet Errors

cd /var/Clients/ClientApp

cat ParquetMerge\*.txt > ParquetMerge\_full\_2012.txt

awk '$1=="POST" {print f} {f=$3}' ParquetMerge\_full\_2012.txt > PDSFinalParquetQueueState\_Errors\_2012.txt

Check for Bad Parquet error in Final Parquet Process

cat /var/PDS\_Tomcat/apache-tomcat-7.0.70/logs/catalina.out | grep 'is not a Parquet file'

Steps for clearing unhealthy nodes in Final Parquet Process

* 1. Sudo su Hadoop
  2. hdfs dfs -rm -R /var/log/hadoop-yarn/apps/phhd/logs-tfile/
  3. hdfs dfs -rm -r -skipTrash /var/log/spark/apps/
  4. hdfs dfs -mkdir /var/log/spark/apps/
  5. hdfs dfs -chmod 777 /var/log/spark/apps/
  6. hadoop fs -chown spark:spark /var/log/spark/apps

Connect to DCi Database - Test

psql --host=ip-10-228-144-252.ec2.internal --port=5432 --username=pguser --password --dbname=pwp\_dci

pwp\_dci=# select count(\*) from dci.datastore\_hdfs\_table;

count

-------

52898

(1 row)

**DCI Update Process**

1. Login with super user as below

psql -U 'pguser' -d "postgres"

use password -----> P0wer5ervic3s

2. Run the following Queries drop the database

SELECT \* FROM pg\_stat\_activity WHERE datname = 'pwp\_dci';

DROP DATABASE IF EXISTS pwp\_dci;

3. copy the lastest DCI dump

hadoop fs -copyToLocal s3a://dev-pd-inbound/database/pwp\_dci\_master\_55.dump /mnt/dumptmp/

4. restoring the DB

psql -c "create database pwp\_dci;"

pg\_restore -d pwp\_dci /mnt/dumptmp/pwp\_dci\_master\_55.dump &

wait

psql -d pwp\_dci -c "GRANT ALL ON SCHEMA dci TO pguser;"

psql -d pwp\_dci -c "GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA dci TO pguser;"

su postgres <<'EOF'

echo /mnt/ >> log.txt

psql -c "CREATE USER pguser SUPERUSER PASSWORD 'P0wer5ervic3s';"

psql -c "CREATE USER gpadmin WITH PASSWORD 'gpadmin';"

psql -c "create database pwp\_dci;"

pg\_restore -d pwp\_dci /mnt/dumptmp/pwp\_dci\_master\_55.dump &

wait

psql -d pwp\_dci -c "GRANT ALL ON SCHEMA dci TO pguser;"

psql -d pwp\_dci -c "GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA dci TO pguser;"

EOF

5. Remove the dumpfolder

rm -rf /mnt/dumptmp/pwp\_dci\_\*

6. Test the pwp\_dci DATABASE

psql -d pwp\_dci;

SELECT \* FROM dci.hhd\_tags\_to\_hdfs(5, null::varchar[], array[row('282125.\*')]::varchar[], '2019-01-01 00:00:00.000'::timestamp, '2019-01-01 23:59:59.999'::timestamp, 0) where (input\_index = -1 or name in ('282125.HRTBT\_LHIST', '282125.TNH'));

DCI Update Process

1. Login with super user as below

psql -U 'pguser' -d "postgres"

use password -----> P0wer5ervic3s

2. Run the following Queries drop the database

SELECT \* FROM pg\_stat\_activity WHERE datname = 'pwp\_dci';

DROP DATABASE IF EXISTS pwp\_dci;

3. copy the lastest DCI dump

hadoop fs -copyToLocal s3a://dev-pd-inbound/database/pwp\_dci\_master\_55.dump /mnt/dumptmp/

4. restoring the DB

psql -c "create database pwp\_dci;"

pg\_restore -d pwp\_dci /mnt/dumptmp/pwp\_dci\_master\_55.dump &

wait

psql -d pwp\_dci -c "GRANT ALL ON SCHEMA dci TO pguser;"

psql -d pwp\_dci -c "GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA dci TO pguser;"

su postgres <<'EOF'

echo /mnt/ >> log.txt

psql -c "CREATE USER pguser SUPERUSER PASSWORD 'P0wer5ervic3s';"

psql -c "CREATE USER gpadmin WITH PASSWORD 'gpadmin';"

psql -c "create database pwp\_dci;"

pg\_restore -d pwp\_dci /mnt/dumptmp/pwp\_dci\_master\_55.dump &

wait

psql -d pwp\_dci -c "GRANT ALL ON SCHEMA dci TO pguser;"

psql -d pwp\_dci -c "GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA dci TO pguser;"

EOF

5. Remove the dumpfolder

rm -rf /mnt/dumptmp/pwp\_dci\_\*

6. Test the pwp\_dci DATABASE

psql -d pwp\_dci;

SELECT \* FROM dci.hhd\_tags\_to\_hdfs(5, null::varchar[], array[row('282125.\*')]::varchar[], '2019-01-01 00:00:00.000'::timestamp, '2019-01-01 23:59:59.999'::timestamp, 0)

where (input\_index = -1 or name in ('282125.HRTBT\_LHIST', '282125.TNH'));

Intermediate HDFS Validation

Download the below file to as Hadoop user to /home/hadoop

aws s3 cp s3://ge-engineering/vamsheetest/validation/validation.zip .

unzip validation.zip

./validation\_hdfs.sh > $Month\_$Year.log

awk '/Elements/{nr[NR]; nr[NR+7]}; NR in nr' Feb\_2020.log -> gives the count or corrupted folder!

Historian

Starting 2002 – Aug 2019 🡪 Historian

25a5a333-ced9-4f78-b55c-c0ce9608adc1 AR01

4ba6fec8-dd5e-419e-97fc-50852472563c IN01

60969ca3-75a3-46ab-8ff9-62d6ac604b09 IN03

7e8db425-6548-4d34-adf8-d3fb2f7295b4 AR01

7fdb2dec-31d3-487b-b765-c61c3a11cd2d IN03

88dcf466-38b4-4994-821e-89c0da68c3f4 IN03

9205f129-a85f-4dde-9889-95da9d70c459 IN03

a0cbbf5c-f703-4fb2-a3f8-8bfef6bf8e0f IN03

b450eb9c-dc17-4d2c-b5c5-3b19cf75b3b4 IN03

b8d3cf5d-3c3f-4407-b27b-f30e3f747514 US01

c06ff0f8-1c79-4b25-97dd-c3e95e8e4d11 IN01

c43d153c-d6e9-4fa5-8f79-fd5a846250fe AR01

dc3c0cdc-c086-47ba-a8c9-eb18cd2933ea IN02

defcbac5-0158-411c-b45b-4251e9cd2e26 US02

e52af41f-8526-4e9c-be6e-c7b9b4d14804 IN01

e6a8d557-0f7f-47a3-875d-f9908c413631 IN03