

**UCB Pharmaceuticals**

OneTruth Development HDP Cluster

Raha Naseri · Big Industries NV

31.01.2017

# Table of Contents

[1Table of Contents 3](#__RefHeading___Toc3680_928472187)

[2System Inventory 5](#__RefHeading___Toc3682_928472187)

[3Prepare the hosts 5](#__RefHeading___Toc3684_928472187)

[3.1Disable selinux 5](#__RefHeading___Toc3686_928472187)

[3.2Install and run ntpd 5](#__RefHeading___Toc3688_928472187)

[3.3Set swappiness 5](#__RefHeading___Toc3690_928472187)

[3.4Disable transparent huge pages 6](#__RefHeading___Toc3692_928472187)

[3.5Disable firewalld 6](#__RefHeading___Toc3694_928472187)

[3.6Reboot 6](#__RefHeading___Toc3696_928472187)

[3.7Passwordless ssh 6](#__RefHeading___Toc3698_928472187)

[3.8install oracle client (sqlplus) 6](#__RefHeading___Toc3700_928472187)

[4Install and setup Ambari-server,HDP 7](#__RefHeading___Toc3702_928472187)

[4.1Download the tarballs 7](#__RefHeading___Toc3704_928472187)

[4.2Create repository 7](#__RefHeading___Toc3706_928472187)

[4.3install ambari-server 9](#__RefHeading___Toc3708_928472187)

[4.4Run the installation wizard 10](#__RefHeading___Toc3710_928472187)

[4.5Configure services 10](#__RefHeading___Toc3712_928472187)

[4.6Install Ranger 11](#__RefHeading___Toc3714_928472187)

[5Setup Kerberos using KDC 12](#__RefHeading___Toc3716_928472187)

[5.1Install and configure kerberos server and client 12](#__RefHeading___Toc3718_928472187)

[5.2create the Kerberos database 13](#__RefHeading___Toc3720_928472187)

[5.3start the service 13](#__RefHeading___Toc3722_928472187)

[5.4create an admin principal for KDC 13](#__RefHeading___Toc3724_928472187)

[5.5install and locate JCE files 14](#__RefHeading___Toc3726_928472187)

[5.6Enable kerberos from ambari user interface 14](#__RefHeading___Toc3728_928472187)

[5.7Create and authorize user accounts for kerberos 14](#__RefHeading___Toc3730_928472187)

[5.8Setup Kerberos for ambari-server 15](#__RefHeading___Toc3732_928472187)

[6Setup views in Ambari 16](#__RefHeading___Toc3734_928472187)

[6.1Files view 16](#__RefHeading___Toc3736_928472187)

[6.2Hive view 16](#__RefHeading___Toc3738_928472187)

[6.3Tez view 17](#__RefHeading___Toc3740_928472187)

[7performance improvement configuration 17](#__RefHeading___Toc3742_928472187)

[7.1Tez configs 17](#__RefHeading___Toc3744_928472187)

[7.2Hive config 18](#__RefHeading___Toc3746_928472187)

[8Enabling ACID transactions on hive 18](#__RefHeading___Toc3748_928472187)

# System Inventory

|  |  |
| --- | --- |
| Hostname | Specifications |
| gdcdrwhap802.dir.ucb-group.com  (10.1.76.166) | Red Hat Enterprise Linux 7.3  62 GB RAM  220 GB Disk  10 core |
| gdcdrwhap802.dir.ucb-group.com  (10.1.76.166) | Red Hat Enterprise Linux 7.3  62 GB RAM  1 TB Disk  10 core |

# Prepare the hosts

## Disable selinux

Disable selinux on all nodes of the cluster :

open /etc/selinux/config file and edit this line:

SELINUX=disabled

Run sestatus to make sure selinux is disabled.

## Install and run ntpd

Install and run ntpd on all nodes in the cluster :

yum install -y ntp

systemctl enable ntpd

systemctl start ntpd

## Set swappiness

open /etc/sysctl.conf with an editor and add this line:

vm.swappiness=10

## Disable transparent huge pages

echo never > /sys/kernel/mm/transparent\_hugepage/defrag

echo never > /sys/kernel/mm/transparent\_hugepage/enabled

## Disable firewalld

systemctl disable firewalld

yum install -y httpd telnet

## Reboot

Reboot for all changes to take effect.

reboot

## Passwordless ssh

To enable Passwordless ssh from ambari-server host to other hosts

On GDCDRWHAP802 (ambari-server host) login with the linux user you want to run ambari installation wizard with, then run:

ssh-keygen

now two key files are generated in ~/.ssh

~/.ssh/id\_rsa

~/.ssh/id\_rsa.pub

copy these two files into ~/.ssh directory of all the nodes in the cluster and from this directory in each node run:

cat id\_rsa.pub >> authorized\_keys

sudo chmod 644 –R ~/.ssh/

to verify the passwordless ssh ,from the ambari-server host (gdcdrwhap802) run:

ssh e610617a@gdcdrwhap803.dir.ucb-group.com

## install oracle client (sqlplus)

Install oracle client (sqlplus) to access oracle db remotely

from the following page accept the terms and download the rpm packages of basic and sql plus (latest versions):

http://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html

transfer the files to the server and install them using rpm:

rpm –i oracle-instantclient12.1-basic-12.1.0.2.0-1.x86\_64.rpm

rpm -i oracle-instantclient12.1-sqlplus-12.1.0.2.0-1.x86\_64.rpm

(install oracle-xe-11.2.0-1.0.x86\_64.rpm as well to use ‘impdp’ and ‘expdp’ commands to backup the oracle database).

configure oracle client as below :

LD\_LIBRARY\_PATH=/usr/lib/oracle/12.1/client64/lib:${LD\_LIBRARY\_PATH}

export LD\_LIBRARY\_PATH

PATH=/usr/bin:${PATH}

export PATH

to access oracle using the username ambari run:

sqlplus64 'ambari@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=gdcqwhmap801)(Port=1521))(CONNECT\_DATA=(SID=QWHM801M)))'

after running this command you can access the oracle command shell and run the oracle script to create the tables and indexes for ambari:

@/var/lib/ambari-server/resources/Ambari-DDL-Oracle-CREATE.sql

Make sure all the queries are successful.

exit the shell .

# Install and setup Ambari-server,HDP

# Download the tarballs

download the tarballs for HDP,AMBARI,HDP utilities from the following links into your local machine and transfer them to ambari-server host (gdcdrwhap802):

http://public-repo-1.hortonworks.com/HDP/centos7/2.x/updates/2.5.3.0/HDP-2.5.3.0-centos7-rpm.tar.gz

http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21/repos/centos7/HDP-UTILS-1.1.0.21-centos7.tar.gz

http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.4.2.0/ambari-2.4.2.0-centos7.tar.gz

## Create repository

Create a directory:

sudo mkdir -p /var/www/10.1.76.166/local-repo

cd /var/www/10.1.76.166/local-repo

move the tarballs you just downloaded to this directory.

Now set this path as http Document root:

cat >> /etc/httpd/conf/httpd.conf <<EOF

<VirtualHost \*:80>

DocumentRoot "/var/www/10.1.76.166/local-repo"

ServerName 10.1.76.166

</VirtualHost>

EOF

Give the proper permissions to this directory:

chmod -R 755 /var/www/

Start the http service :

service httpd start

now create the repository files:

cat > /etc/yum.repos.d/ambari.repo <<EOF

#VERSION\_NUMBER=2.4.2.0-136

[Updates-ambari-2.4.2.0]

name=ambari-2.4.2.0 - Updates

baseurl= http://10.1.76.166/AMBARI-2.4.2.0/centos7/2.4.2.0-136/

gpgcheck=0

enabled=1

priority=1

EOF

cat > /etc/yum.repos.d/HDP.repo <<EOF

#VERSION\_NUMBER=2.5.3.0-37

[HDP-2.5.3.0]

name=HDP Version - HDP-2.5.3.0

baseurl= http://10.1.76.166/HDP/centos7

gpgcheck=0

enabled=1

priority=1

[HDP-UTILS-1.1.0.21]

name=HDP-UTILS Version - HDP-UTILS-1.1.0.21

baseurl= http://10.1.76.166/HDP-UTILS-1.1.0.21/repos/centos7

gpgcheck=0

enabled=1

priority=1

EOF

# install ambari-server

find the compatible version of jdk to the hdp and ambari server version and install it using rpm from the following link:

http://www.oracle.com/technetwork/java/javase/downloads/

set java home

JAVA\_HOME=/path/to/java/home:${JAVA\_HOME}

export JAVA\_HOME

download the compatible ojdbc jar file (ojdbc6.jar in this case) from oracle and place it in the following directories:

/usr/java/share/bin

/usr/lib/oracle/12.1/client64/lib

setup ambari server with the following configs:

sudo ambari-server setup

ambari account: root

jdk : choose ‘ 3.custom jdk ’

and specify the java home path (the path where you have jdk installed)

database:choose 3.oracle

username:ambari

password: FgAD82yVDv

host: bracdbdb802

sid: DRWH801M

finish the configuration and run the following command:

ambari-server setup --jdbc-db=oracle --jdbc-driver=/usr/lib/oracle/12.1/client64/lib/ojdbc6.jar

run ambari-server

sudo ambari-server start

## Run the installation wizard

Access ambari-server UI from the browser via

http://10.1.76.166:8080

Name the cluster 'UcbCluster'

choose to install HDP using the local repository and in the section for specifying the path keep only the one for the operating system of the servers( in this case Centos 7 )

find the path from the repo files for HDP and HDP utilities

uncheck the 'skip verifying the repositories' and click next

insert the hostnames of the cluster and copy the content of ~/.ssh/id\_rsa in private key section

and specify which user (linux user) is going to run ambari tasks

in the next page the hosts should be verified installed and registered.

**Bugfix**

if the installation of the hosts is not successful you can do it manually by running these commands in all the nodes of the cluster and retry to register them

yum install ambari-agent

sudo nano /etc/ambari-agent/conf/ambari-agent.ini

add the following to this file :

[server]

hostname=10.1.76.166

url\_port=8440

secured\_url\_port=8441

now retry registering hosts via ambari.

## Configure services

in the next page we have to configure following services with their database credentials

**hive:**

use an existing oracle database

database name: hive

database username: hive

password: h96fDMB6QH

database URL: jdbc:oracle:thin:@//bracdbdb802.dir.ucb-group.com:1521/DRWH801M

**oozie:**

database name: oozie

database username: oozie

password: FgAD82yVDv

database URL : jdbc:oracle:thin:@//bracdbdb802.dir.ucb-group.com:1521/DRWH801M

**grafana :**

username: grafana

password: UT5D2#uqU(

for oozie and hive test the connections before proceeding to verify that ambari can access and connect to the remote oracle database

in the next page check all the services required for the cluster and assign the components to the master nodes and after reviewing the whole architecture confirm it to be installed and started.

Before installing the services make sure that the servers do have access to ambari ,hdp and hdp utilities repositories by running sudo yum repolist.

**Bugfix**

The installation of the services might be aborted due to an error about snappy,in that case run the following command on the nodes running data nodes (workers)

sudo yum downgrade snappy

And retry the installation from the wizard.

After a successful installation of the services you can launch the cluster’s dashboard and have the list of the services with their status on the left side of the page.

## Install Ranger

Before installing Ranger you should make sure that all the services are running,green and without critical alert.

From Actions below the service names, choose add service then proceed with the wizard:

During the procedure configure Ranger as follows:

In Ranger admin tab

Ranger database name: onetruth

Database name: ranger

Database username: ranger

Database password: S7BNaYxaN

Database host : bracdbdb802:1521:DRWH801M

Setup Database and Database User : No

In Ranger Audit tab

Audit solr : no

In advanced tab: insert the following

Ranger admin username for ambari: admin

Password: admin

In the same tab expand ‘advanced ranger-env’ and insert the same username and password (admin/admin).

And start the installation of Ranger.

# Setup Kerberos using KDC

Before proceeding to this stage make sure all the services are running green and without critical alert.

## Install and configure kerberos server and client

yum install krb5-server krb5-libs krb5-workstation

edit krb5.conf **:**

sudo nano /etc/krb5.conf

add the following to this file:

[libdefaults]

renew\_lifetime = 7d

forwardable = true

default\_realm = DEV.ONETRUTH.LOCAL

ticket\_lifetime = 24h

dns\_lookup\_realm = false

dns\_lookup\_kdc = false

default\_ccache\_name = /tmp/krb5cc\_%{uid}

#default\_tgs\_enctypes = aes des3-cbc-sha1 rc4 des-cbc-md5

#default\_tkt\_enctypes = aes des3-cbc-sha1 rc4 des-cbc-md5

[domain\_realm]

dev.onetruth.local = DEV.ONETRUTH.LOCAL

.dev.onetruth.local = DEV.ONETRUTH.LOCAL

[logging]

default = FILE:/var/log/krb5kdc.log

admin\_server = FILE:/var/log/kadmind.log

kdc = FILE:/var/log/krb5kdc.log

[realms]

DEV.ONETRUTH.LOCAL = {

admin\_server = gdcdrwhap802.dir.ucb-group.com

kdc = gdcdrwhap802.dir.ucb-group.com

}

## create the Kerberos database

kdb5\_util create -s

## start the service

systemctl enable krb5kdc

systemctl enable kadmin

systemctl start krb5kdc

systemctl start kadmin

## create an admin principal for KDC

kadmin.local -q "addprinc admin/admin"

check /var/kerberos/krb5kdc/kadm5.acl to confirm your principal is added as an entry in this file.

Restart the service

systemctl restart kadmin

## install and locate JCE files

download the appropriate JCE policy file :

http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

in all hosts add these jar files to $JAVA\_HOME/jre/lib/security

Restart ambari-server for change to take effect.

sudo ambari-server restart

## Enable kerberos from ambari user interface

In Ambari navigate to admin -> Kerberos -> enable kerberos

In 'get started' page choose existing MIT KDC and confirm you have met all the prerequisites.

on the Kerberos configuration page insert the following values:

KDC type: existing KDC

KDC-host: gdcdrwhap802.dir.ucb-group.com

Realm name: DEV.ONETRUTH.LOCAL

Domains : DEV.ONETRUTH.LOCAL,. DEV.ONETRUTH.LOCAL

Then test the KDC connection, if it is ok then enter the credentials for the admin principal that you just created :

Kadmin host: gdcdrwhap802.dir.ucb-group.com

Admin principal: admin/admin@dev.onetruth.local

Admin password : 2fUWqtUGry

In the next pages proceed with the wizard and let ambari install and test Kerberos and enable it.

When the wizard is finished ambari has generated all the necessary principals for the services.

## Create and authorize user accounts for kerberos

To access the hadoop components with a specific username on the cluster the username has to get a ticket to authorize on Kerberos.

For example to authenticate user infa (the user used for informatica tools)

1. The linux user “infa” has to be added to the servers on the cluster :

sudo adduser -g hdfs infa

1. Create the principal:

Kadmin.local –q ‘addprinc -randkey infa@DEV.ONETRUTH.LOCAL’

1. Generate the keytab for this principal:

Kadmin.local –q ‘xst -k /etc/security/keytabs/infa.keytab infa@DEV.ONETRUTH.LOCAL’

1. Give proper permissions on keytab file:

sudo chmod 644 /etc/security/keytabs/infa.keytab

This keytab file has to be located somewhere in the local machine of windows users as well (ask the informatica admin for the proper path)

1. Get a ticket for the user:

kinit -k -t /etc/security/keytabs/infa.keytab infa@DEV.ONETRUTH.LOCAL

1. Add the proxyuser properties for the corresponding user to 'core-site.xml'

In ambari by navigating to hdfs -> config -> advanced -> Custom core-site -> add these properties:

Hadoop.proxyuser.infa.hosts= gdcdrwhap801.dir.ucb-group.com

Hadoop.proxyuser.infa.groups= \*

The infa user is accessing the cluster from the server ‘gdcdrwhap801’ so in the hosts the FQDN of this server has to be inserted.

1. login into one of the servers using “infa” user and create home directory for the user on hdfs

Hadoop fs –mkdir /user/infa

and make sure this directory is owned by infa:hdfs

1. Now you can authenticate to hadoop services using “infa” user.

## Setup Kerberos for ambari-server

Kadmin.local ‘addprinc -randkey ambari-server@DEV.ONETRUTH.LOCAL’

xst -k /etc/security/keytabs/ambari.server.keytab ambari-server@DEV.ONETRUTH.LOCAL’

sudo chmod 644 /etc/security/keytabs/ambari.server.keytab

sudo ambari-server stop

sudo ambari-server setup-security

choose option ‘Setup Ambari kerberos JAAS configuration’

insert ambari-server@DEV.ONETRUTH.LOCALas the principal and

insert the path of the keytab

Sudo ambari-server start

Make sure the following properties exist in core-site.xml by navigating to hdfs-> config-> advanced -> Custom core-site

Hadoop.proxyuser.ambari-server.hosts= gdcdrwhap802.dir.ucb-group.com

Hadoop.proxyuser.ambari-server.groups= \*

# Setup views in Ambari

Configure the cluster for views

Add these 2 properties to the core-site.xml by navigating to

hdfs -> config -> advanced -> Custom core-site

hadoop.proxyuser.root.groups=\*

hadoop.proxyuser.root.hosts=\*

## Files view

navigate to manage ambari -> views ->Files -> create instance

configure the following fields as below:

Hive.Authentication : auth=KERBEROS;principal=hive/gdcdrwhap802@DIR.UCB-GROUP.COM;hive.server2.proxy.user=${username}

webHDFS Authentication: auth=KERBEROS;proxyuser= ambari-server@DEV.ONETRUTH.LOCAL

cluster configuration:

‘custom’

HiveServer2 Host: gdcdrwhap802.dir.ucb-group.com

webHDFS Filesystem URI: webhdfs://gdcdrwhap802.dir.ucb-group.com:50070

Yarn Application Timeline Server: http://gdcdrwhap802.dir.ucb-group.com:8188

Yarn resourcemanagement URL: http://gdcdrwhap802.dir.ucb-group.com:8088

## Hive view

navigate to manage ambari -> Views -> Hive -> create instance

configure the following as below:

choose version 1.0.0

Hive Authentication: auth=KERBEROS;principal=hive/gdcdrwhap802@DIR.UCB-GROUP.COM;hive.server2.proxy.user=${username}

WebHDFS: auth=KERBEROS;proxyuser= ambari-server@DEV.ONETRUTH.LOCAL

Cluster Configuration:

Choose Custom

HiveServer2 Host: gdcdrwhap802.dir.ucb-group.com

WebHDFS FileSystem URI: webhdfs://gdcdrwhap802.dir.ucb-group.com:50070

Yarn application Timeline Server URL: http://gdcdrwhap802.dir.ucb-group.com:8188

Yarn ResourceManager URL: http://gdcdrwhap802.dir.ucb-group.com:8088

Save and Grant Permission to “admin” user .

## Tez view

navigate to manage ambari -> Views -> Tez -> create instance

configure the following as below:

cluster configuration:

Choose Custom

Yarn application Timeline Server URL: http://gdcdrwhap802.dir.ucb-group.com:8188

Yarn ResourceManagaer URL: http://gdcdrwhap802.dir.ucb-group.com:8088

note: Make sure in the tez configuration the value for the property tez.tez-ui.history-url.base is :

http://gdcdrwhap802.dir.ucb-group.com:8080/#/main/views/TEZ/0.7.0.2.5.3.0-136/TEZ\_CLUSTER\_INSTANCE

# performance improvement configuration

to improve the performance in the cluster the following configurations should be done for hive and tez:

## Tez configs

tez.session.am.dag.submit.timeout.secs =600

tez.task.resource.memory.mb = 4096

tez.am.resource.memory.mb = 5120

tez.am.container.reuse.enabled = true

tez.runtime.io.sort.mb = 675

## Hive config

set hive.vectorized.execution.enabled = true;

set hive.vectorized.execution.reduce.enabled = true;

Allow dynamic numbers of reducers = true

Enable Reduce Vectorization = true

Restart the affected services

# Enabling ACID transactions on hive

Configure the following properties in hive settings :

Hive.support.concurrency= true

Hive.exec.dynamic.partition.mode = nonstrict

Hive.txn.manager = org.apache.hadoop.hive.ql.lockmgr.DbTxnManager

ACID Trnasactions Runcompactor = true

Hive.compactor.worker.threads = 1

Oracle user credentials

|  |  |
| --- | --- |
| Service Username | Password |
| ambari | FgAD82yVDv |
| hive | h96fDMB6QH |
| oozie | FgAD82yVDv |
| ranger | S7BNaYxaN |