Hadoop Manual Installation:

**PreRequisties:** Take min 3 Vm. 1 Vm for Master node and other two for worker node, Change name of all 3 machines to “golden-node1, golden-node2, golden-node3”

**Ist Machine:**

[hpcsa@localhost ~]$ su –

Password:

[root@localhost ~]# hostnamectl set-hostname golden-node1

[root@localhost ~]# ip a

**IInd Machine:**

[hpcsa@localhost ~]$ su –

Password:

[root@localhost ~]# hostnamectl set-hostname golden-node2

[root@localhost ~]# ip a

**IIIrd Machine:**

[hpcsa@localhost ~]$ su –

Password:

[root@localhost ~]# hostnamectl set-hostname golden-node3

[root@localhost ~]# ip a

[root@localhost ~]# vim /etc/hosts

**On the Ist Machine:**

[root@master ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.248.140 golden-node1

192.168.248.141 golden-node2

192.168.248.142 golden-node3

[root@localhost ~]# scp /etc/hosts root@golden-node2:/etc/hosts

[root@localhost ~]# scp /etc/hosts root@golden-node3:/etc/hosts

[root@localhost ~]# ping golden-node2

[root@localhost ~]# ping golden-node3

[root@localhost ~]# systemctl stop firewalld [Also Do on Other two nodes]

[root@localhost ~]# systemctl disable firewalld [Also Do on Other two nodes]

[root@localhost ~]# systemctl status firewalld [Also Do on Other two nodes]

* firewalld.service - firewalld - dynamic firewall daemon

[root@localhost ~]# sestatus [Also Do on Other two nodes]

[root@localhost ~]# vim /etc/selinux/config [Also Do on Other two nodes]

* Change selinux to “disabled”

[root@localhost ~]# init 6 [Also Do on Other two nodes]

[root@localhost ~]# sestatus [Also Do on Other two nodes]

[root@localhost ~]# systemctl status firewalld [Also Do on Other two nodes]

[root@golden-node1 ~]# ssh-keygen

* It will generate a private and public key pair for this “rsa algorithm” used
* Default file for saving public key, identification is “/root/.ssh/id\_rsa”

[root@golden-node1 ~]# ssh-copy-id root@golden-node2

[root@golden-node1 ~]# ssh-copy-id root@golden-node3

[root@golden-node1 ~]# yum install httpd\* [Also Do on Other two nodes]

[root@golden-node1 ~]# yum install -y apache\* [Also Do on Other two nodes]

[root@golden-node1 ~]# yum install webserver\* [Also Do on Other two nodes]

[root@golden-node1 ~]# systemctl start httpd [Also Do on Other two nodes]

[root@golden-node1 ~]# systemctl enable http [Also Do on Other two nodes]

[root@golden-node1 ~]# systemctl status httpd [Also Do on Other two nodes]

[root@golden-node1 ~]# cd /var/www/html/

[root@golden-node1 html]# mkdir ambari

[root@golden-node1 html]# cd ambari/

[root@golden-node1 ambari]# cp /root/ambari-2.7.3.0-centos7.tar.gz /var/www/html/ambari/

[root@golden-node1 ambari]# ls

[root@golden-node1 ambari]# tar -xvf ambari-2.7.3.0-centos7.tar.gz

[root@golden-node1 ambari]# cp /var/www/html/ambari/ambari/centos7/2.7.3.0-139/ambari.repo /etc/yum.repos.d/

[root@golcen-node1 ambari]# cd

[root@golden-node1 ~]# cd /etc/yum.repos.d

[root@golden-node1 yum.repos.d]# ls

[root@golden-node1 yum.repos.d]# mv ambari.repo ambary

[root@golden-node1 yum.repos.d]# yum install –y createrepo

[root@golden-node1 yum.repos.d]# mv ambary ambary.repo

[root@golden-node1 yum.repos.d]# createrepo /var/www/html/ambari/ambary

[root@golden-node1 yum.repos.d]# ping golden-node2

[root@golden-node1 yum.repos.d]# ping golden-node3

[root@golden-node1 yum.repos.d]# vim ambari.repo

* Find url: /var/www/html/ambari/ambari/centos7/2.7.3.0-139/
* Change baseurl with above path
* Change gpgcheck to 0
* Comment gpgkey

[root@golden-node1 yum.repos.d]# yum repolist

[root@golden-node1 yum.repos.d]# scp ambari.repo [root@golden-node2:/etc/yum.repos.d/](mailto:root@golden-node2:/etc/yum.repos.d/)

[root@golden-node1 yum.repos.d]# scp ambari.repo [root@golden-node3:/etc/yum.repos.d/](mailto:root@golden-node3:/etc/yum.repos.d/)

[root@golden-node1 yum.repos.d]# yum install -y ambari-server

[root@golden-node1 yum.repos.d]# ambari-server setup –s

* Using python /usr/bin/python Setup ambari-server
* Configuring database, Creating schema and user, Creating tables
* Ambari repo file contains latest json url http://public-repo-1.hortonworks.com/HDP/hdp\_urlinfo.json, updating stacks repoinfos with it
* Adjusting ambari-server permissions and ownership
* About to start PostgreSQL, Configuring local database..., Configuring PostgreSQL..., Restarting PostgreSQL

[root@golden-node1 yum.repos.d]# ambari-server start

[root@golden-node1 yum.repos.d]# ambari-server status

[root@golden-node1 yum.repos.d]# yum install -y ambari-agent [Also Do on Other two nodes]

[root@golden-node1 yum.repos.d]# ambari-agent start [Also Do on Other two nodes]

[root@golden-node1 yum.repos.d]# ambari-agent status [Also Do on Other two nodes]

[root@golden-node1 yum.repos.d]# vim /etc/ambari-agent/conf/ambari-agent.ini

* Change localhost to “golden-node1”

[root@golden-node1 yum.repos.d]# ambari-agent restart [Also Do on Other two nodes]

[root@golden-node1 yum.repos.d]# ambari-agent status [Also Do on Other two nodes]

[root@golden-node1 yum.repos.d]# scp /etc/ambari-agent/conf/ambari-agent.ini [root@golden-node2:/etc/ambari-agent/conf/ambari-agent.ini](mailto:root@golden-node2:/etc/ambari-agent/conf/ambari-agent.ini)

[root@golden-node1 yum.repos.d]# scp /etc/ambari-agent/conf/ambari-agent.ini [root@golden-node2:/etc/ambari-agent/conf/ambari-agent.ini](mailto:root@golden-node2:/etc/ambari-agent/conf/ambari-agent.ini)

Go to Browser 🡪 golden-node1:8080

Username: admin

Password: admin

[root@golden-node1 ~]# yum install –y ntp

[root@golden-node1 ~]# systemctl start ntpd

[root@golden-node1 ~]# systemctl enable ntpd

* Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to /usr/lib/systemd/system/ntpd.service.

[root@golden-node1 ~]# systemctl status ntpd

[root@golden-node1 ~]# cd /var/www/html/

[root@golden-node1 html]# ls

[root@golden-node1 html]# mkdir hdp

[root@golden-node1 html]# cd hdp/

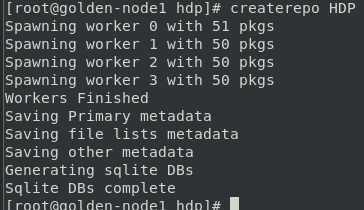
[root@golden-node1 hdp]# cp /root/HDP-3.1.0.0-centos7-rpm.tar.gz /var/www/html/

[root@golden-node1 hdp]# tar -xvf HDP-3.1.0.0-centos7-rpm.tar.gz

[root@golden-node1 hdp]# cp /root/HDP-UTILS-1.1.0.22-centos7.tar.gz /var/www/html/hdp/

[root@golden-node1 hdp]# tar -xvf HDP-UTILS-1.1.0.22-centos7.tar.gz

[root@golden-node1 hdp]# createrepo HDP

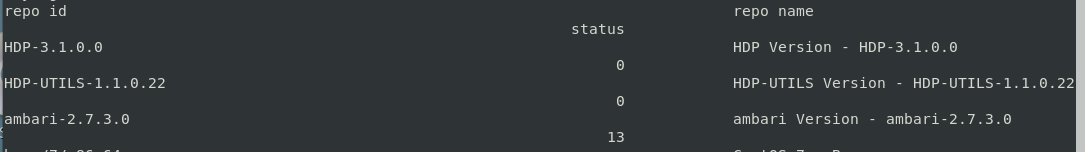


[root@golden-node1 hdp]# createrepo HDP-UTILS

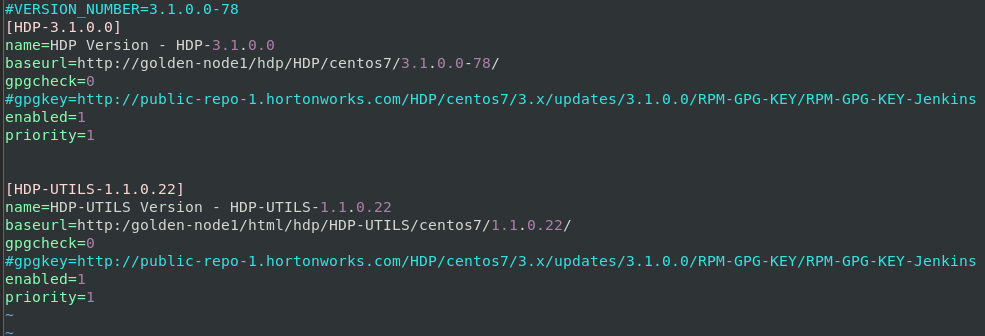
[root@golden-node1 hdp]# cd

[root@golden-node1 ~]# cp /var/www/html/hdp/HDP/centos7/3.1.0.0-78/hdp.repo /etc/yum.repos.d/

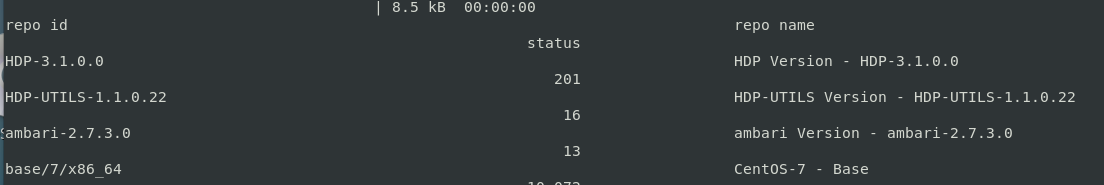
[root@golden-node1 ~]# yum repolist



[root@golden-node1 ~]# vim /etc/yum.repos.d/hdp.repo



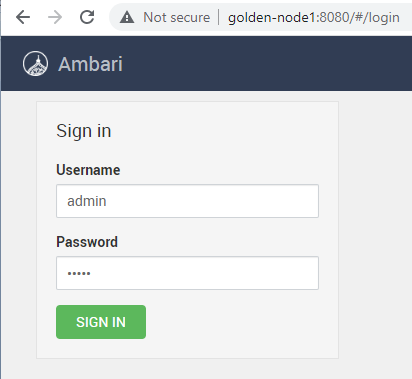
[root@golden-node1 ~]# yum repolist



Go to Browser: golden-node1:8080

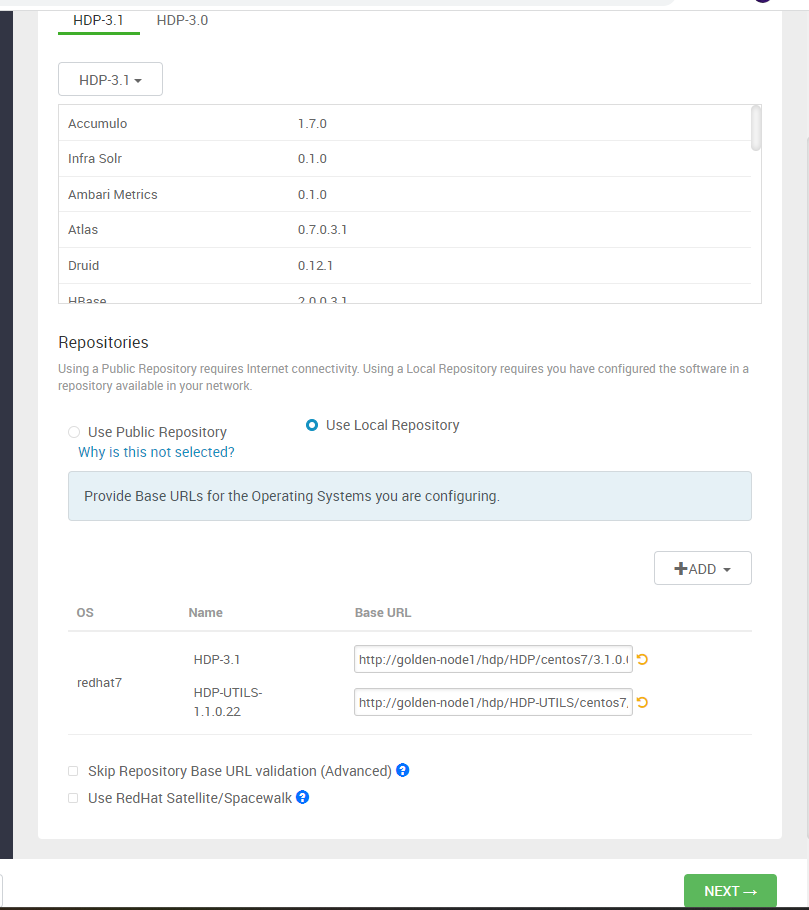
Userbane: admin

Password: admin

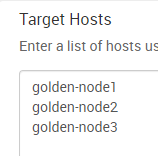


Click on LAUNCH INSTALL WIZARD

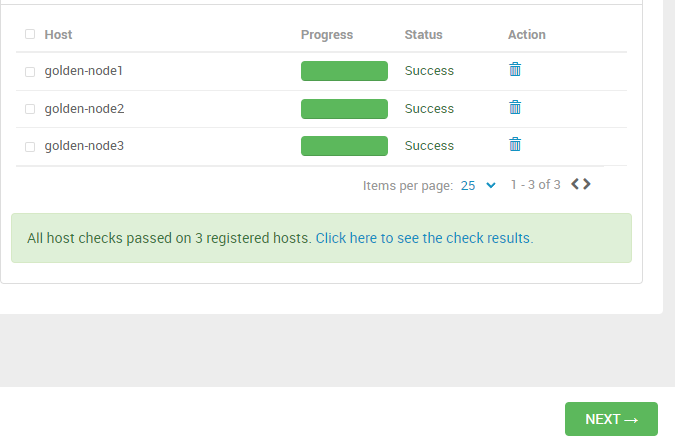
* Get Started
  + Name your cluster: golden (give any name)
* Select version:
  + Click on Use Local Repository
  + Copy baseurl from /etc/yum.repos.d/hdp.repo for both HDP and HDP-Utils



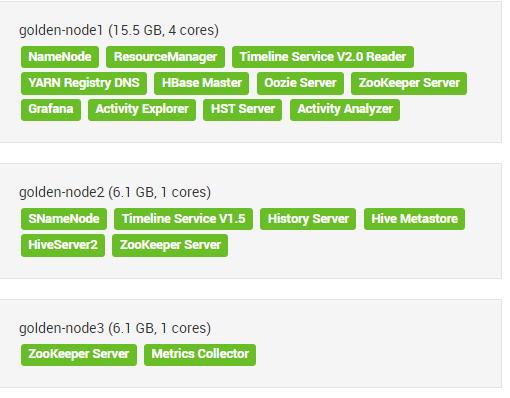
* Install Options
  + Give Target Host



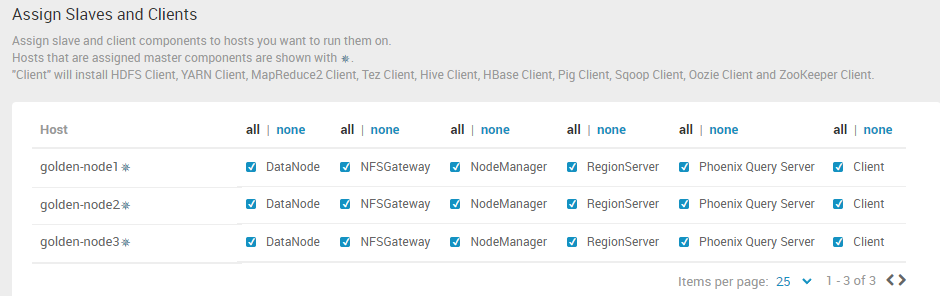
* + Click on Perform [manual registration](javascript:void(null)) on hosts and do not use SSH
* Confirm Hosts



* Choose Services
  + Select HDFS, YARN, MapReduce2, Tez, Hive, Hbase, Pig Sqoop, Oozie, Ambari Metrics (you can select as much services as you want)
* Assign Masters
  + Leave default page



* Assign Slaves and Clients
  + Select all for all hosts



* Customizes Services
  + CREDENTIALS
    - Give “hpcsa” in password for “Grafana Admin, Hive Database, Oozie Database” and, give “admin” as password in “Activity Explorer’s Admin”
  + DATABASES
    - Click on the link of downloading mysql & It will redirect to next page 🡪 Select Operating System: “Red Hat Enterprise Linux /Oracle Linux” and, Select OS version: Linux 7





* + Then click on “No thanks, just start my download”.

Then Go to terminal:

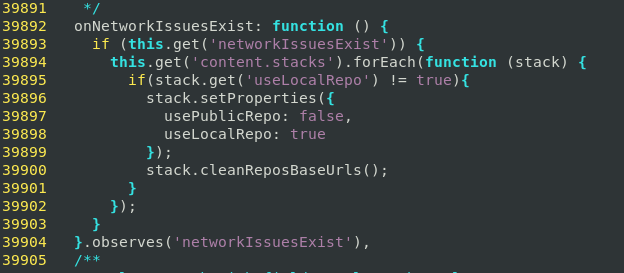
[root@golden-node1 ~]# cd /usr/lib/ambari-server/web/javascripts/

[root@golden-node1 javascripts]# cp app.js app.js\_backup

[root@golden-node1 javascripts]# vim app.js

Go to browser: <https://www.cnblogs.com/nshuai/p/13404273.html>

find out the line(39892) in “app.js” file then change the content like this





[root@golden-node1 javascripts]# ambari-server reset

[root@golden-node1 javascripts]# ambari-server stop

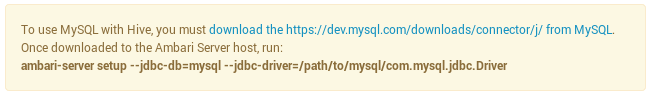
[root@golden-node1 javascripts]# ambari-server reset

[root@golden-node1 javascripts]# ambari-server start

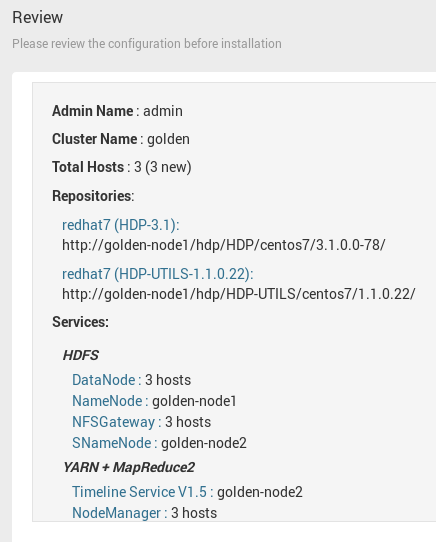
[root@golden-node1 javascripts]# cd

[root@golden-node1 ~]# rpm -ivh /home/hpcsa/Downloads/mysql-connector-j-8.0.31-1.el7.noarch.rpm

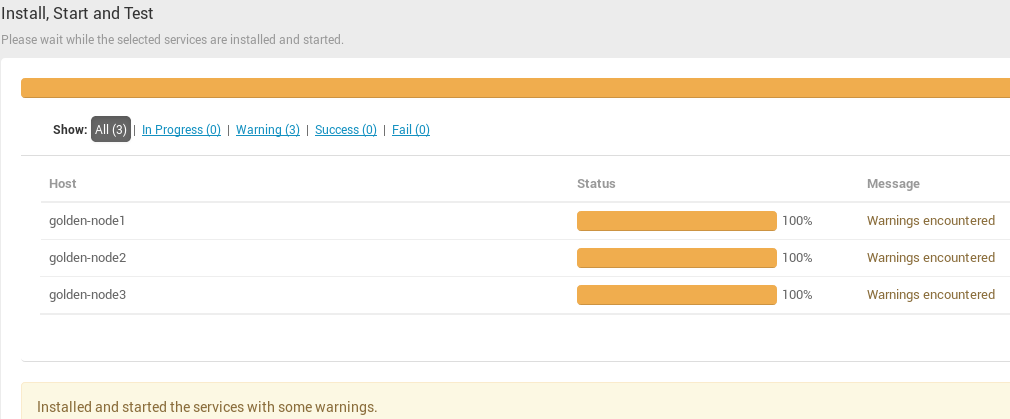
[root@golden-node1 ~]# ambari-server setup --jdbc-db=mysql --jdbc driver=/usr/share/java/mysql-connector-j.jar



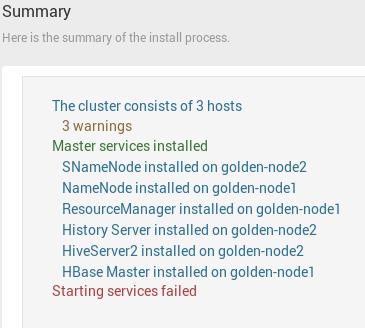
* + DIRECTORIES
    - Leave Deafaul page 🡪 Click on Next untill reach to Review page
* Review 🡪 Deploy



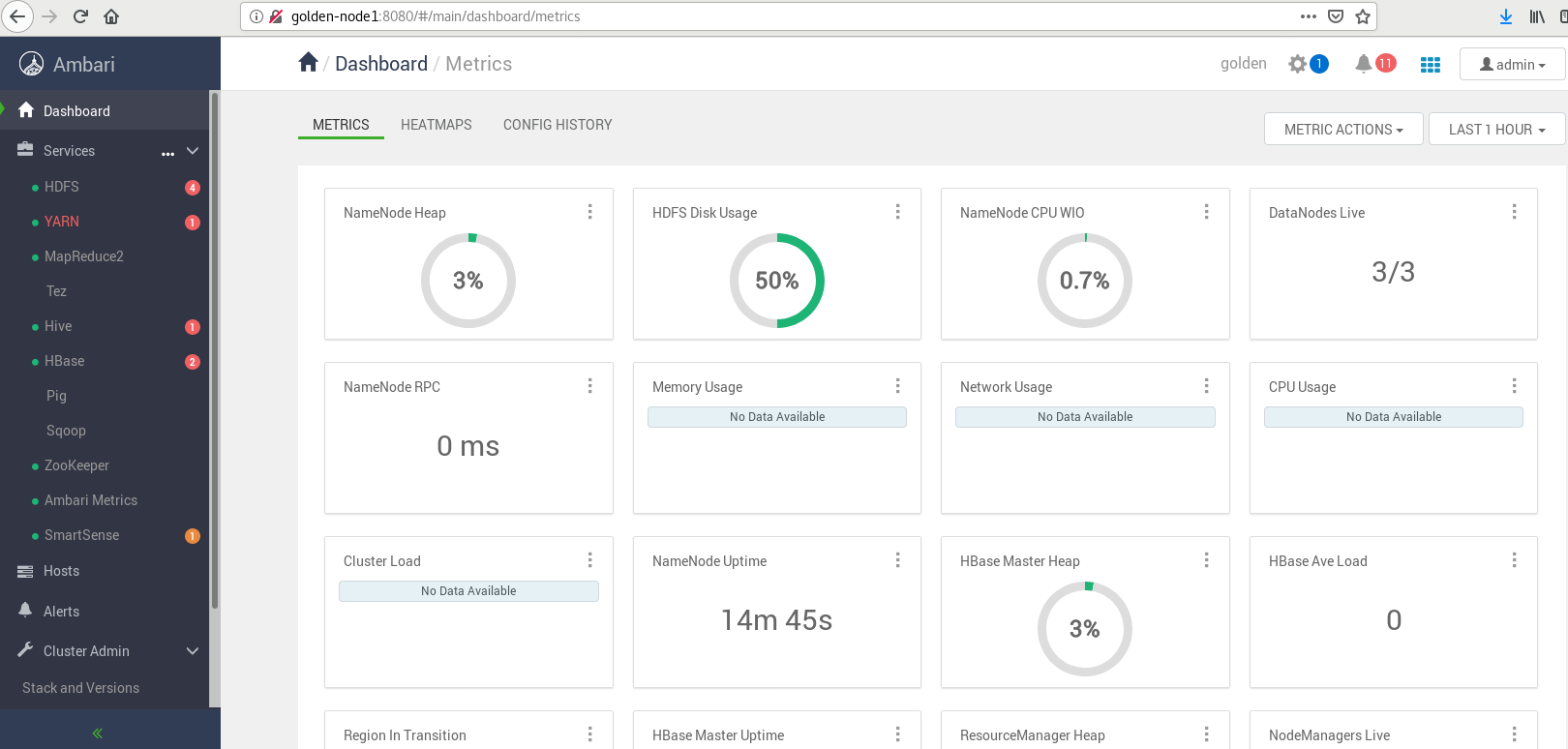
* Install, Start and Test



* Summary 🡪 COMPLETE



Then, Dashboard should looklike this.



Important Points:

1. **HDFS:**

|  |  |
| --- | --- |
| 1. **DATA DIR.** | **PATH** |
| DataNode Directories  NameNode Directories  SecondaryNameNode Checkpoint Dir.  NFSGateway dump dir.  NameNode Backup dir.  JournalNode Edits directory  NameNode Checkpoint Edits directory | /hadoop/hdfs/data  /hadoop/hdfs/namenode  /hadoop/hdfs/namesecondary  /tmp/.hdfs-nfs  /tmp/upgrades  /hadoop/hdfs/journalnode  ${dfs.namenode.checkpoint.dir} |
| 1. **LOG DIR**. | **PATH** |
| Hadoop Log Dir Prefix | /var/log/hadoop |
| 1. **PID DIR.** | **PATH** |
| Hadoop PID Dir Prefix | /var/run/hadoop |