**Step1: Create a local repo**

#Download the ambari and HDP repo from the public

Here is the link to get the public repo

<https://docs.cloudera.com/HDPDocuments/Ambari-2.6.2.0/bk_ambari-installation/content/ambari_repositories.html>

1. Download the ambari repo[This is with respective to Operating system]

#wget -b <http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.6.2.0/ambari-2.6.2.0-centos7.tar.gz>

#wget -b <http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.6.2.0/ambari.repo>

1. Download the HDP repository form Public

#wget -b <http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7/HDP-UTILS-1.1.0.22-centos7.tar.gz>

#wget -b <http://public-repo-1.hortonworks.com/HDP/centos7/2.x/updates/2.6.5.0/HDP-2.6.5.0-centos7-rpm.tar.gz>

#wget -b <http://public-repo-1.hortonworks.com/HDP/centos7/2.x/updates/2.6.5.0/hdp.repo>

1. After Downloaded the HDP and Ambari packages just need to Extract the tar

#tar -xvzf HDP-2.6.5.0-centos7-rpm.tar.gz

#tar -xvzf HDP-UTILS-1.1.0.22-centos6.tar.gz

#tar -xvzf ambari-2.6.2.2-centos7.tar.gz

1. After extracted the above packages we need to move this to webserver

#systemctl start httpd

#chkconfig httpd on

#mkdir -p /var/www/html/repo/

# mv HDP /var/www/html/repo/

#mv HDP-UTILS /var/www/html/repo/

#mv ambari /var/www/html/repo/

#mv ambari.repo /etc/yum.repo.d/

#mv hdp.repo /etc/yum.repo.d/

1. Once you have moved this to webserver we need to change the baseurl which is pointing to the public to local[change the baseurl and gpgkey]

#vim /etc/yum.repo.d/ambari.repo

#VERSION\_NUMBER=2.6.2.2-1

[ambari-2.6.2.2]

name=ambari Version - ambari-2.6.2.2

baseurl=http://10.20.3.35/repo/centos7/ambari/centos7/2.6.2.2-1/

gpgcheck=1

gpgkey=http://10.20.3.35/repo/centos7/ambari/centos7/2.6.2.2-1/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins

enabled=1

priority=1

1. We need to do the same for hdp.repo as well

Here we have changed the baseurl and gpgkey in HDP and HDP-UTILS

#vim /etc/yum.repo.d/hdp.repo

#VERSION\_NUMBER=2.6.5.0-292

[HDP-2.6.5.0]

name=HDP Version - HDP-2.6.5.0

baseurl=http://10.20.3.35/repo/centos7/HDP/centos7/2.6.5.0-292/

gpgcheck=1

gpgkey=http://10.20.3.35/repo/centos7/HDP/centos7/2.6.5.0-292/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins

enabled=1

priority=1

[HDP-UTILS-1.1.0.22]

name=HDP-UTILS Version - HDP-UTILS-1.1.0.22

baseurl=http://10.20.3.35/repo/centos7/HDP-UTILS/centos7/1.1.0.22/

gpgcheck=1

gpgkey=http://10.20.3.35/repo/centos7/HDP-UTILS/centos7/1.1.0.22/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins

enabled=1

priority=1

1. After pointing to the local we need to run the below command

#yum clean all

#yum repolist

1. #Hostnamectl set-hostname master.local

#bash

1. Systemctl stop firewalld

**Step 2: Install the Docker in the machine**

#yum install docker

#systemctl start docker

#systemctl status docker

**Step3: Download the mysql connector and mysql packages and move to ambari-server resource folder**

#wget <https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.20-1.el7.noarch.rpm>

#wget <http://repo.mysql.com/mysql57-community-release-el7.rpm>

#rpm -ivh mysql57-community-release-el7.rpm

#rpm -ivh mysql-connector-java-8.0.20-1.el7.noarch.rpm

#ls -al /usr/share/java/mysql-connector-java.jar

#mv /usr/share/java/mysql-connector-java /var/lib/ambari-server/resources/

#cd /var/lib/ambari-server/resources/

#ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar

**Step4: we need to move towards passwordless ssh as ambari-server want to connect to all machines for registering the slave and master**

1. In ambari-server macine[master machine], as a root execute the below command[Press enter for all by default it will generate 2 file private-id\_rsa and public-id\_rsa.pub]

#ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa):

Created directory '/root/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

c6:66:93:16:73:0b:bf:46:46:28:7d:a5:38:a3:4d:6d root@master

The key's randomart image is:

+--[ RSA 2048]----+

| . |

| . + o |

| . @ E |

| \* & . |

| . S = |

| = + . |

| o |

| . |

| |

+-----------------+

This will generate the following files:

1. We can go and check the same.

#cd /root/.ssh/

-rw-r--r-- 1 root root 399 Apr 30 01:53 id\_rsa.pub

-rw------- 1 root root 1679 Apr 30 01:53 id\_rsa

c. Create an authorized\_keys and copy the content of id\_rsa.pub to it.

#vi authorized\_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCroX/JeGDudQ0McGuV6pbU+DgnrGaEiYVWeKo/8sups/+MxRUmGCfEYY1dQbj+LDhXz90GXWGK/2LQwxx+t+fQMRP7G/g/SZ/7/Nxfr9MxchHPfkIKnU1oQscvOTxIhZbnNK1IcBNkb8jiOuf3JR/rLlZXb9ikm9T2dUuKUqOYEya3fgDlRWQdExn7SIxX1lkiByKZfdcUPruMyBBXuKlQmMqN7H1Z6IbofcS36gc+4OgQAZgmDL1LsIFYQUsXMby/HGMK+EZ+KtW0sz/tPigikwO+uXpcDrPa3QuNwsUIpUm8ZfSrWmMckNpZuyEsI1UiPWEuA8eMKsKPBRhsON39 [root@master.local](mailto:root@master.local)

Step3: Create the docker file and by using that dockerfile create the docker image and by using that docker image we can create multiple docker slave containers…

Move to opt folder and create the Docker file

#cd /opt/

1. Create the dockerfile

#vi Dockerfile

FROM centos:7

MAINTAINER Nagesh

ENV container docker

RUN yum update -y && yum install systemd tar wget unzip -y && yum clean all && \

yum install epel-release -y && \

yum repolist && \

yum clean all

RUN mkdir -p /hadoop/hdfs/data

RUN mkdir -p /opt/localdata

RUN yum -y install openssh openssh-server openssh-clients openssl-libs sudo install initscripts net-tools ntp

RUN systemctl start sshd

RUN systemctl start ntpd

RUN (cd /lib/systemd/system/sysinit.target.wants && for i in \*; do [ $i == systemd-tmpfiles-setup.service ] || rm -vf $i; done) && \

rm -vf /lib/systemd/system/multi-user.target.wants/\* && \

rm -vf /etc/systemd/system/\*.wants/\* && \

rm -vf /lib/systemd/system/local-fs.target.wants/\* && \

rm -vf /lib/systemd/system/sockets.target.wants/\*udev\* && \

rm -vf /lib/systemd/system/sockets.target.wants/\*initctl\* && \

rm -vf /lib/systemd/system/basic.target.wants/\* && \

rm -vf /lib/systemd/system/anaconda.target.wants/\*

VOLUME ["/sys/fs/cgroup"]

CMD ["/usr/sbin/init"]

1. By using the above docker file we need to build the docker image..

#docker build --rm -t new\_hadoop\_image - < Dockerfile

1. Once the above command is executed successfully we docker images containers are ready, we can have look by exectuting the below command

#docker images

[root@master ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

new\_hadoop\_image latest 6e42ffc8a5bd 1 days ago 480 MB

docker.io/centos 7 5e35e350aded 6 months ago 203 MB

1. By using the docker images we can create the docker container by executing the below command

#docker run --privileged -e container=slave1 --name slave1 --hostname slave1.local --add-host master.local:10.20.3.35 --add-host slave1.local:172.17.0.2 --add-host slave2.local:172.17.0.3 --add-host slave3.local:172.17.0.4 --add-host slave4.local:172.17.0.5 -v /hadoop/data:/haoop/hdfs/data -v /opt/localdata:/root new\_hadoop\_image /usr/sbin/init

1. You came out by contr+c In master shell
2. We can start the container

#docker start slave1

1. We want to make passwardless ssh with slave1 and slave2 container login to slave1 and slave2 container and start the sshd and ntpd service and create authorized\_keys file in /root/.ssh/

#docker exec -it slave1 /bin/bash

#systemctl start sshd

#systemctl start ntpd

#ssh-keygen

Once you run the above command it will create the 2 file with is private and public…move to /root/.ssh/

#cd /root/.ssh/

#rm -rf id\_rsa\*

#vi authorized\_keys

Copy the content from id\_rsa.pub of master to here and save and exit

1. We need to enable the sshd and ntpd in slave

#systemctl start sshd

#systemctl start ntpd

1. We can check whether from master can slave1 can able to login without password.

In master server

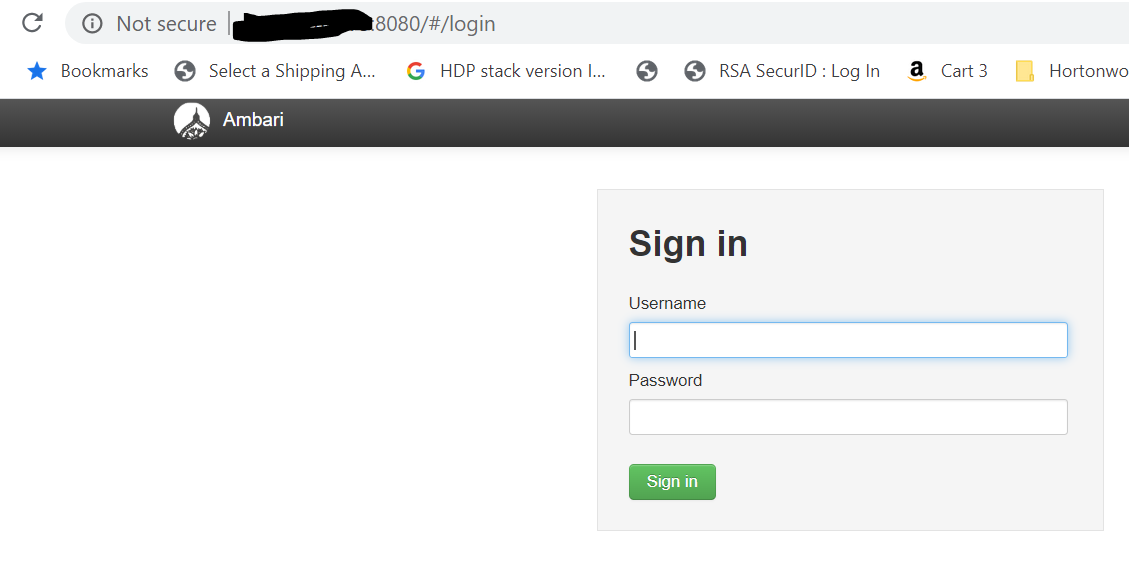
#ssh slave1

It should go to the slave1 shell witout asking the password…

1. Above should be same for all the containers, even if you add the new container to the existing cluster also.

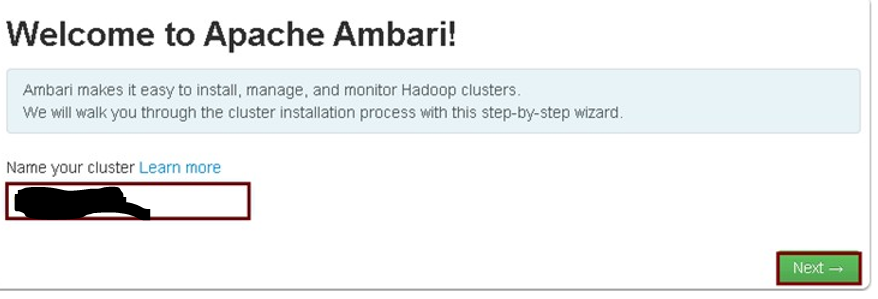
**Step4: Instalation of ambari server and ambari-agent in master**

1. #yum install ambari-server [press enter for default instalation]
2. #ambari-server setup [Press Enter for Default instalation]
3. #systemctl ambari-server start
4. #systemctl ambari-agent start
5. After starting the ambari server, we can check that ambari-server is lauching in webui with 8080 port it will look like the below



Default username and password will be admin and admin

1. Once you login we can able to find launch screen and Enter the clustername



1. Select the default stack and select the local repository and add the target machines that will automatically install the ambari-agent In all the machines.

Master.local

Slave1.local

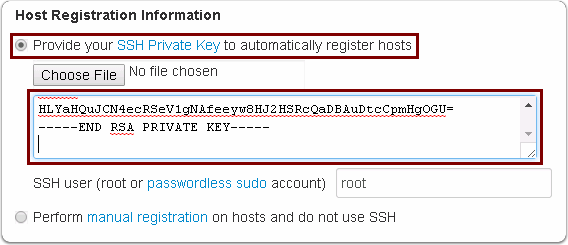
Save2.local

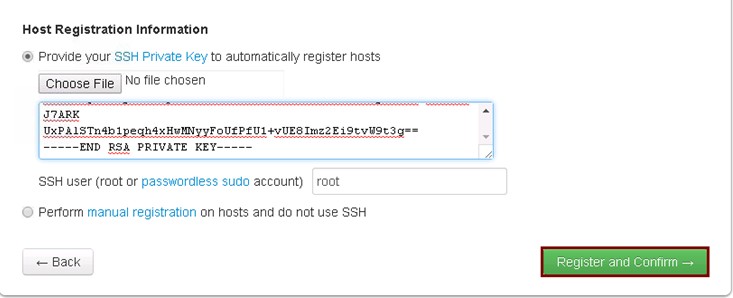
#add the baseurl of hdp and hdp-utils from the hdp.repo which is present in ambari-server[Master-- /etc/yum.repo.d/hdp.repo]

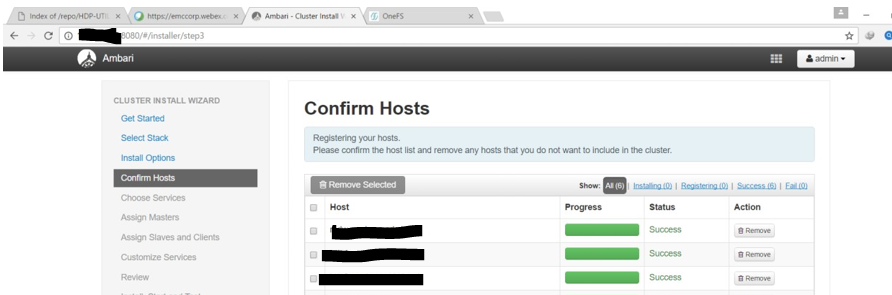
1. Copy the private key from the master server where we have installed the ambari-server

#cat ~/.ssh/id\_rsa



Enter the same in the ssh private key

1. Than click register and confirm
2. Confirm the hosts that need to be register it looks like below



K. Choose the server

# HDFS

#YARN+Mapreduce

#Tez

#Sqoop

#Zookeeper

#Hive

1. Assign the masters and slave based on the load between the master and slave1
2. Customize services: In this section we need to add the hive database password by default hive:hive
3. We can review the configuration and deploy and once you click the deploy button..it will start the Instalation.
4. Once the installation is done we can go to ambari webui and start the services which is not started during the installation..

Step4: Adding the node to the Existing cluster.

1. We need to create the container by using the step3 d. commands, In That command we need to change the container name and hostname name to slave2.local..

#docker run --privileged -e container=slave2 --name slave2 --hostname slave2.local --add-host master.local:10.20.3.35 --add-host slave1.local:172.17.0.2 --add-host slave2.local:172.17.0.3 --add-host slave3.local:172.17.0.4 --add-host slave4.local:172.17.0.5 -v /hadoop/data:/haoop/hdfs/data -v /opt/localdata:/root new\_hadoop\_image /usr/sbin/init

1. To start the container slave2.

#docker start slave2

1. To make this to password less we need to copy the content from id\_rsa.pub from the ambari-server[master] to this container in /root/.ssh/authorized\_keys..to do that we have follow the below steps

#docker exec -it slave1 /bin/bash

#cd /root/.ssh/

#vi authorized\_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCroX/JeGDudQ0McGuV6pbU+DgnrGaEiYVWeKo/8sups/+MxRUmGCfEYY1dQbj+LDhXz90GXWGK/2LQwxx+t+fQMRP7G/g/SZ/7/Nxfr9MxchHPfkIKnU1oQscvOTxIhZbnNK1IcBNkb8jiOuf3JR/rLlZXb9ikm9T2dUuKUqOYEya3fgDlRWQdExn7SIxX1lkiByKZfdcUPruMyBBXuKlQmMqN7H1Z6IbofcS36gc+4OgQAZgmDL1LsIFYQUsXMby/HGMK+EZ+KtW0sz/tPigikwO+uXpcDrPa3QuNwsUIpUm8ZfSrWmMckNpZuyEsI1UiPWEuA8eMKsKPBRhsON39 [root@master.local](mailto:root@master.local)

#save and exit and we can validate the same by doing the ssh from master to slave2

#ssh slave2 [in ambari-server – it should be able to login without asking for password]

1. Then go to Ambari webui. Click on the host on the right hand side there is an action button click on it chose the add node button.
2. Once you get adding node pop up screen add the hostname to it, I mean hostname of new container to be added to the cluster here it is slave2.local

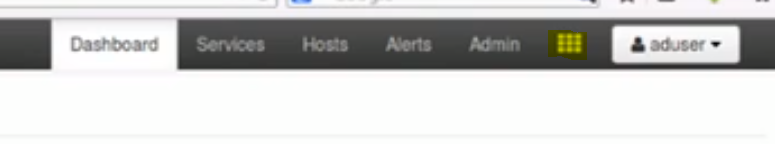
#To get the file view in Ambari, you need to go to HDFS Service and Advance configuration and go to customcoresite in that search for “hadoop.proxyuser.root.groups=\*” and try to create an admin user in haddop

#sudo -u hdfs hdfs dfs -mkdir -p /user/admin

#sudo -u hdfs hdfs dfs -chown admin:admin /user/admin

#sudo -u hdfs hdfs dfs -chmod 755 /user/admin

Try to login to ambari and in you right corner we can find the ickon as below click on that and select the file view and try to upload the file



**Know Issues:**

1. Ambari-agent while registering, it will through up an ssl issue for cento 7.5 above to resolve this issue.

If you are using the centos7.5, below are the mandatory to do otherwise ambari-agent will not register itsell to overcome that we need to add this “force\_https\_protocol=PROTOCOL\_TLSv1\_2” in /etc/ambari-agent/conf/ambari-agent.ini under Security below “verify ssl\_verify\_cert=0”

In Ambari-server, master machines we need to disable the cert-verification

#vim /etc/python/cert-verification.cfg

verify=disable

1. If hive not starts while default installation need to be validated

#ls -al /usr/share/java/mysql-connector-java.jar

#mv /usr/share/java/mysql-connector-java /var/lib/ambari-server/resources/

#cd /var/lib/ambari-server/resources/

#ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar