**Installing Hadoop using cloudera & ansible**

1. Choose aws servers for your Hadoop cluster

A.1 I am choosing t3a.xlarge as it suits my requirement of memory and CPU vcores with minimum hourly rate as compared to its peers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t3a.xlarge | 4 | 96 | 16 | EBS-Only | Up to 5 |

A.2 login to aws ec2 dashboard and in marketplace select centos 7, select centos7 with updated HMV.

A.3 From panel choose t3a.xlarge and click on configure instance details.

A.4 My POC need six such instance hence add 6 instances and click on add storage

A.5 Choose size of volume as 40 gig and choose column type as magnetic. Make sure you check Delete on Termination.

A.6 Add tag and then create a new security group and then click on option continue magnetic as boot volume.

A.7 create new key and download and save it on a some secure location.

A.8 click on launch instances.

A.9 Your instances are ready now rename them as per understanding .

1. Configure and provision your servers

B.1 try login into aws server using git bash /putty/Cygwin

ssh -i Documents/AWS\_keys/centos7.pem root@ec2-52-54-144-200.compute-1.amazonaws.com

Please login as the user "centos" rather than the user "root".

B.2 Lets change hostname.

B.2.1 update hostname in /etc/hostname : mm1.vigdatos.com

B.2.2 Now run command : sudo hostnamectl set-hostname --static mm1.vigdatos.com

B.2.3 Update sudo vi /etc/cloud/cloud.cfg

B.2.4 type line in end of file : preserve\_hostname: true

Now reboot.

1. Install web server on master1

sudo yum -y install httpd

sudo mkdir -p /var/www/html

[root@mm1 ~]# sudo systemctl start httpd

[root@mm1 ~]# sudo systemctl enable httpd

I have domain listed in godaddy so I will add an entry of type A with host as extension (mm1) and Point as elastic ip of server

Also add entries in route 53

1. Install and configure Ansible

D.1 update: sudo yum update

D.2 install ansible sudo yum install ansible

D.3 check version ansible –version

D.4 Now add hosts name in /etc/ansible/hosts file

[hadoop-servers]

ip-172-31-5-67.ec2.internal ansible\_user=root

ip-172-31-13-61.ec2.internal ansible\_user=root

ip-172-31-12-228.ec2.internal ansible\_user=root

ip-172-31-7-59.ec2.internal ansible\_user=root

ip-172-31-2-105.ec2.internal ansible\_user=root

D.5 sample playbook

---

- hosts: hadoop-servers

tasks:

- name: install java 8 on all servers

command: sudo yum -y install java-1.8.0-openjdk

- name: install open jdk 8

command: sudo yum -y install java-1.8.0-openjdk-devel

# User specific environment and startup programs

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk

PATH=$PATH:$HOME/bin:$JAVA\_HOME/bin

export PATH

D6. Let ansible.cfg as default

D7. Test with few commands

ansible hadoop-servers -i hosts -a "hostname -f"

ansible hadoop-servers -i hosts -a "sudo systemctl restart sshd"

D8.Anisble sample command to run update java

ansible-playbook java\_update.yml -i /etc/ansible/hosts --become

1. Install SSH on all servers and enable password less access to each other .

E.1 create password for root on all servers.

Passwd centos

Passwd root

E.2 check users and groups currently present

awk -F: '{print $1}' /etc/passwd

awk -F: '{print $1}' /etc/group

awk -F ":" '{print $1" "$3}' /etc/passwd

awk -F ":" 'NR==FNR {h[$3] = $1; next} {print $1" "h[$3]}' /etc/group /etc/passwd

E.3 Create new users and groups

[root@mm1 ~]# sudo groupadd hadoop

[root@mm1 ~]# sudo useradd hadoop\_admin

[root@mm1 ~]# usermod -a -G hadoop hadoop\_admin

[root@mm1 ~]# passwd hadoop

passwd: Unknown user name 'hadoop'.

[root@mm1 ~]# passwd hadoop\_admin

Changing password for user hadoop\_admin.

New password:

BAD PASSWORD: The password is shorter than 8 characters

Retype new password:

passwd: all authentication tokens updated successfully.

E.4 Now log into all servers and generate ssh key with user root

ssh-keygen

E.5 Now change the sshd\_config file in /etc/ssh : Permitrootlogin as yes and PasswordAuthentication as Yes , PubkeyAuthentication yes

ssh-copy-id -i ~/.ssh/id\_rsa.pub [root@ec2-3-89-228-190.compute-1.amazonaws.com](mailto:root@ec2-3-89-228-190.compute-1.amazonaws.com)

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

1. Installing Kerberos

F.0 setup and install ntp

yum -y install ntp

ntpdate 0.rhel.pool.ntp.org

systemctl start ntpd.service

systemctl enable ntpd.service

F.1 create yml playbook for installing Kerberos server and Kerberos client

---

- hosts: mm1.vigdatos.com

tasks:

- name: install kerberos server on master server.

command: sudo yum -y install krb5-server

- hosts: hadoop-servers

tasks:

- name: install kerberos client on all server

yum:

name: krb5-libs

state: latest

- name: install kerberos workstation

yum:

name: krb5-workstation

state: latest

F2. Execute the playbook

ansible-playbook install\_kerberos.yml -i /etc/ansible/hosts –become

F3.Set up default realm to your domain name

Vi /etc/krb5.conf

Sample ::

# Configuration snippets may be placed in this directory as well

#includedir /etc/krb5.conf.d/

[libdefaults]

default\_realm = MM1.VIGDATOS.COM

dns\_lookup\_realm = false

dns\_lookup\_kdc = false

ticket\_lifetime = 24h

forwardable = true

udp\_preference\_limit = 1000000

default\_tkt\_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

default\_tgs\_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

permitted\_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

[realms]

MM1.VIGDATOS.COM = {

kdc = mm1.vigdatos.com:88

admin\_server = mm1.vigdatos.com:749

default\_domain = mm1.vigdatos.com

}

[domain\_realm]

.mm1.vigdatos.com = MM1.VIGDATOS.COM

mm1.vigdatos.com = MM1.VIGDATOS.COM

[logging]

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

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

default = FILE:/var/log/krb5lib.log

Now /var/kerberos/krb5kdc/kdc.conf

[kdcdefaults]

kdc\_ports = 88

kdc\_tcp\_ports = 88

[realms]

VIGDATOS.COM = {

#master\_key\_type = aes256-cts

acl\_file = /var/kerberos/krb5kdc/kadm5.acl

dict\_file = /usr/share/dict/words

admin\_keytab = /var/kerberos/krb5kdc/kadm5.keytab

supported\_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal arcfour-hmac:normal camellia256-cts:normal camellia128-cts:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal

}

vi /var/kerberos/krb5kdc/kadm5.acl

example ::

default\_realm = MM1.VIGDATOS.COM

[kdcdefaults]

v4\_mode = nopreauth

kdc\_ports = 0

[realms]

MM1.VIGDATOS.COM = {

kdc\_ports = 88

admin\_keytab = /var/kerberos/krb5kdc/kadm5.keytab

database\_name = /var/kerberos/krb5kdc/principal

acl\_file = /var/kerberos/krb5kdc/kadm5.acl

key\_stash\_file = /var/kerberos/krb5kdc/stash

max\_life = 10h 0m 0s

max\_renewable\_life = 7d 0h 0m 0s

master\_key\_type = des3-hmac-sha1

supported\_enctypes = arcfour-hmac:normal des3-hmac-sha1:normal des-cbc-crc:normal des:normal des:v4 des:norealm des:onlyrealm des:afs3

default\_principal\_flags = +preauth

}

~

add vigdatos.com from example.com

F4. Create KDC database :

kdb5\_util create -r MM1.VIGDATOS.COM -s

[root@mm1 krb5kdc]# kadmin.local

Authenticating as principal root/admin@vigdatos.com with password.

kadmin.local: addprinc root/admin

WARNING: no policy specified for root/admin@vigdatos.com; defaulting to no policy

Enter password for principal "root/admin@vigdatos.com":

Re-enter password for principal "root/admin@vigdatos.com":

Principal "root/admin@vigdatos.com" created.

kadmin.local: addprinc centos

WARNING: no policy specified for centos@vigdatos.com; defaulting to no policy

Enter password for principal "centos@vigdatos.com":

Re-enter password for principal "centos@vigdatos.com":

Principal "centos@vigdatos.com" created.

kadmin.local: ktadd -k /var/kerberos/krb5kdc/kadm5.keytab kadmin/admin

Entry for principal kadmin/admin with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.

Entry for principal kadmin/admin with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.

kadmin.local: ktadd -k /var/kerberos/krb5kdc/kadm5.keytab kadmin/changepw

Entry for principal kadmin/changepw with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.

Entry for principal kadmin/changepw with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.

F5. Start kadmin and krb5kdc deamons

systemctl start krb5kdc.service

systemctl start kadmin.service

systemctl enable krb5kdc.service

systemctl enable kadmin.service

F6. Create a user

kadmin.local

Authenticating as principal root/admin@vigdatos.com with password.

kadmin.local: addprinc -randkey hadoop/mm1.vigdatos.com

WARNING: no policy specified for hadoop/mm1.vigdatos.com@vigdatos.com; defaulting to no policy

Principal "hadoop/mm1.vigdatos.com@vigdatos.com" created.

kadmin.local: ktadd hadoop/mm1.vigdatos.com

Entry for principal hadoop/mm1.vigdatos.com with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.

Entry for principal hadoop/mm1.vigdatos.com with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.

kadmin.local:

F7. Copy krb5.conf file on all client server

scp /etc/krb5.conf ip-172-31-2-105.ec2.internal://etc/

F8. Remove Kerberos

Complete!

[root@mm1 ~]# sudo rm -r /var/kerberos

[root@mm1 ~]# sudo rm -r /etc/krb5.conf

[root@mm1 ~]# sudo rm -r /usr/lib64/krb5

* yum remove krb5-server
* yum remove krb5-libs
* yum remove krb5-workstation

F9. Now kinit root/admin

F10.we can get kdc does not have realm, check network .iptables and firewalld .add the ports ,check telnet mm1.vigdatos.com 88

F11. kadmin -p root/admin

addprinc -randkey root/ip-172-31-5-67.ec2.internal

ktadd root/ip-172-31-5-67.ec2.internal

kinit -kt /etc/krb5.keytab root/ip-172-31-5-67.ec2.internal

1. Make directory /var/www/html

G.1 sudo mkdir -p /var/www/html

G.2 wget <https://archive.cloudera.com/cm5/repo-as-tarball/5.14.1/cm5.14.1-centos7.tar.gz>

G.3 tar -xvzf cm5.14.1-centos7.tar.gz

G.4 tar xvzf cm5.14.1-centos7.tar.gz

G5. sudo mv cm /var/www/html

G6. Sudo mkdir /var/www/html/parcels/chd5

G7. wget <https://archive.cloudera.com/cdh5/parcels/5.14.2/CDH-5.14.2-1.cdh5.14.2.p0.3-el6.parcel.sha1>

G8. wget <https://archive.cloudera.com/cdh5/parcels/5.14.2/CDH-5.14.2-1.cdh5.14.2.p0.3-el7.parcel>

G9. wget <https://archive.cloudera.com/cdh5/parcels/5.14.2/manifest.json>

G10. sudo mkdir impala

G11. cd impala/

G12. wget <https://archive.cloudera.com/impala/parcels/2.1.0/IMPALA-2.1.0-1.impala2.0.0.p0.1995-el6.parcel>

G13. wget <https://archive.cloudera.com/impala/parcels/2.1.0/manifest.json>

G14. sudo mkdir solr

G15. wget https://archive.cloudera.com/search/parcels/latest/SOLR-1.3.0-1.cdh4.5.0.p0.9-el6.parcel

G16. wget <https://archive.cloudera.com/search/parcels/latest/manifest.json>

G17. wget <https://archive.cloudera.com/search/parcels/latest/SOLR-1.3.0-1.cdh4.5.0.p0.9-el6.parcel.sha1>

G18. Sudo mkdir accumulo

G19. wget <https://archive.cloudera.com/accumulo/parcels/latest/ACCUMULO-1.6.0-1.cdh4.6.0.p0.273-el6.parcel>

G20. wget <https://archive.cloudera.com/accumulo/parcels/latest/ACCUMULO-1.6.0-1.cdh4.6.0.p0.273-el6.parcel.sha1>

G21. wget https://archive.cloudera.com/accumulo/parcels/latest/manifest.json

G22. Sudo mkdir kafka

G23. wget https://archive.cloudera.com/kafka/parcels/4/KAFKA-4.1.0-1.4.1.0.p0.4-el7.parcel

G24.

wget https://archive.cloudera.com/kafka/parcels/4/KAFKA-4.1.0-1.4.1.0.p0.4-el7.parcel.sha1

G 25. wget https://archive.cloudera.com/kafka/parcels/4/manifest.json

G 26. Sudo mkdir spark

G27. wget <https://archive.cloudera.com/spark/parcels/latest/SPARK-0.9.0-1.cdh4.6.0.p0.98-el6.parcel>

G28. wget <https://archive.cloudera.com/spark/parcels/latest/SPARK-0.9.0-1.cdh4.6.0.p0.98-el6.parcel.sha1>

G30. wget https://archive.cloudera.com/spark/parcels/latest/manifest.json

G31. Sudo mkdir sqoop-connectors

G32. wget <https://archive.cloudera.com/sqoop-connectors/parcels/latest/SQOOP_TERADATA_CONNECTOR-1.7c5-el7.parcel>

G33. wget <https://archive.cloudera.com/sqoop-connectors/parcels/latest/SQOOP_NETEZZA_CONNECTOR-1.5.1c5-el7.parcel.sha1>

G34. wget https://archive.cloudera.com/sqoop-connectors/parcels/latest/manifest.json

G35. cd /etc/yum.repos.d/

G36. Sudo vi cloudera-manager.repo

G37. [cloudera-manager]

# Packages for Cloudera Manager, Version 5, on RedHat or CentOS 7 x86\_64

name=Cloudera Manager

baseurl=http://mm1.vigdatos.com:80/cm/5/

gpgkey =http://mm1.vigdatos.com:80/cm/RPM-GPG-KEY-cloudera

gpgcheck = 1

G.38 wget <http://archive.cloudera.com/cm5/installer/5.14.2/cloudera-manager-installer.bin>

G39. sudo chmod -R 777 cloudera-manager-installer.bin

Unistall cloudera :: <https://techanswersweb.wordpress.com/2018/03/13/uninstalling-cloudera-manager-and-manager-softwares-5-13-x/>

H Now Install cloudera manager using cloudera manager installer bin

H.1 <http://mm1.vigdatos.com:7180/cmf/login>

H.2 hosts : 3.210.215.20, ip-172-31-1-47.ec2.internal, ip-172-31-13-61.ec2.internal, ip-172-31-12-228.ec2.internal, ip-172-31-7-59.ec2.internal, ip-172-31-2-105.ec2.internal

H3. Login into cmf

H4. Check custom repository : <http://mm1.vigdatos.com/cm/5/>

H5. Add key address : <http://mm1.vigdatos.com/cm/RPM-GPG-KEY-cloudera>

H6. In use parcels options check more options and add remote repositories links :: <http://mm1.vigdatos.com/parcels/sqoop-connectors/>

H7. Also in custom repository :: <http://mm1.vigdatos.com/cm/5/>

H.8 in custom GPG : <http://mm1.vigdatos.com/cm/RPM-GPG-KEY-cloudera>

H.9 <https://techanswersweb.wordpress.com/2018/03/13/uninstalling-cloudera-manager-and-manager-softwares-5-13-x/>

H.10 re.findall('[a-zA-Z0-9.\_]+@[a-zA-Z0-9.\_]+',text)

Install oracle java

wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" " https://download.oracle.com/otn/java/jdk/8u231-b11/5b13a193868b4bf28bcb45c792fce896/jdk-8u231-linux-i586.tar.gz"

I got bad connection refused error while accessing server when I changed acl of var folder,hence I needed to deatch the volume and attach it to other server and make amends.

First thing is to deatch volume ,then attach volume with name /dev/xvdf to other server working.

Lsblk

mount /dev/nvme1n1 /mnt

mount: /dev/nvme1n1 is write-protected, mounting read-only

mount: unknown filesystem type '(null)'

dmesg | tail

sudo mount -o nouuid /dev/nvme1n1p1 /mnt

cd /mnt

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html>

<https://serverfault.com/questions/779438/error-on-ec2-volume-mount>

<https://stackoverflow.com/questions/51091712/how-to-remove-acl-from-a-directory-and-back-to-usual-access-control>

setfacl -bn foobar

sudo umount -l /mnt