**Confluent Kafka Administration and Operations**

**Exercise 1 – Setting Basic Pre-Requisites for Kafka Cluster**

1. Kindly use the below cloud machines provided to you for the Hands-on

Cloud machines are having Public DNS, Public IP, Private DNS, Private IP

1. With the provided Cloud machines details and private key. Login to all the aws instances using the user name – ec2-user
2. Always get into sudo user immediately once you get into the cloud until the document mention you to login as any other user
3. Using Sudo user privilege. Execute the below command to change the hostname from private DNS.

*hostnamectl set-hostname <your-hostname> --static*

*shutdown –r*

The reason to reboot the machine is for making the hostname changes happen effective.

Do the same in all the machine with different hostname, so you can identify the machines. Also this is a good practice to set the unique hostnames for every machines instead of identifying those with IPs. Suggested is to keep confluent-1, confluent-2 etc. Do not use (\_) underscore in the hostname. That is an illegal character.

1. Disable SELinux to allow the ports.

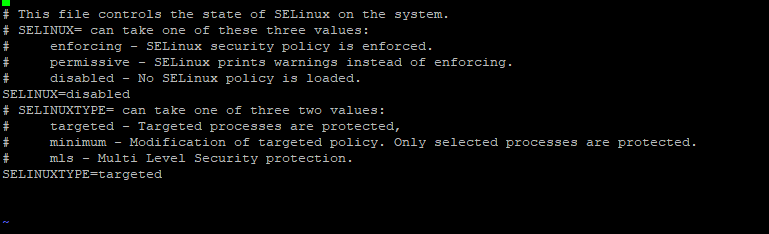
          Reason - One common reason to disable the firewall is, as we know HDFS maintains replication in different nodes/racks but it shouldn't take any extra time for that. Setting firewall using SElinux may disturb this (or) lead to performance issue. So the general recommendation is to disable the firewall.

a.  Open the file /etc/selinux/config

vi /etc/selinux/config

b. Change the config SELINUX=disabled from enforcing

Do the same in all the machines



1. Install NTP Daemon

If you are using RHEL 7 disable chrony daemon before you install ntp using the below command

*systemctl disable chronyd*

Reason for NTP in linux

                    It is used to synchronize the time on your Linux system with a centralized NTP server.

*yum install ntp*

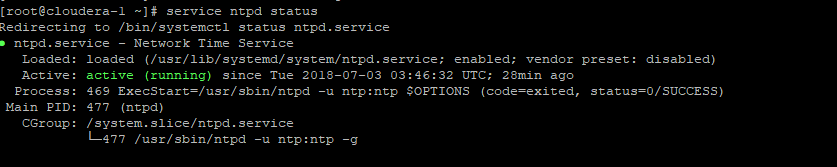
Verify the ntp using

*chkconfig ntpd on*

Start the ntpd using the following command

*service ntpd start*

Do these on all the machines



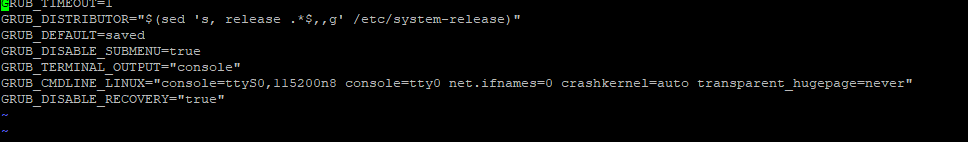
1. Disable Transparent Huge page in all the machines

      Reason - In the Linux kernel, the standard size for a block of addressable memory is four kilobytes. This is called a page. Every system has a finite number of addressable pages, based on how much physical memory the system has installed. A [page table](http://en.wikipedia.org/wiki/Page_table) is used to keep track of the location of each page in the system. This table can get gigantic. So disabling it in hadoop to handle the linux block size

                 vi /etc/default/grub

               Add the following line to the GRUB\_CMDLINE\_LINUX options

               transparent\_hugepage=never



Run the below command to make the changes reflected

*grub2-mkconfig -o /boot/grub2/grub.cfg*

1. Set vm.swappiness to 1 using the below process in all the machines

Reason - The Linux kernel parameter, vm.swappiness, is a value from 0-100 that controls the swapping of application data (as anonymous pages) from physical memory to virtual memory on disk. The higher the value, the more aggressively inactive processes are swapped out from physical memory. The lower the value, the less they are swapped, forcing filesystem buffers to be emptied.

*echo "vm.swappiness=1" >> /etc/sysctl.conf*

Do this on all the machines.

Restart the machine using shutdown –r

1. .     Setup password less authentication for root  to access other VMs

Login your first machine

*sudo su*

vi /etc/hosts this file will host the hostname and alias details for the private IP and DNS.

Please do not use \_ for hostname

Enter the Private IP and Hostname in the below format

* + - 1. confluent-1
      2. confluent-2

172.31.17.98     confluent-3

Copy these lines which you have added and paste it in rest of the VMs /etc/hosts file using sudo access

Once the hostname is set, proceed to establish password less authentication.

Switch to the first machine. Execute

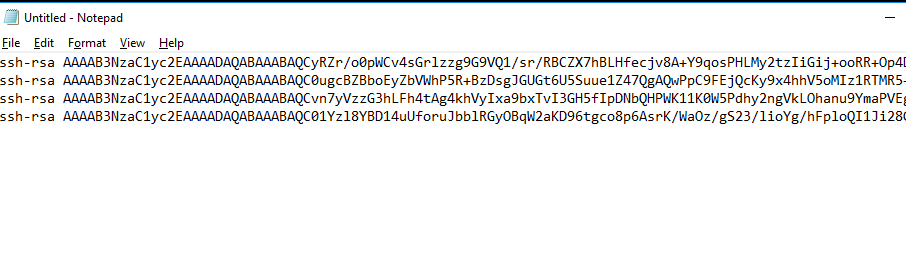
*ssh-keygen -t rsa*

Enter the passphrase and location with empty value. Just press enter. For additional security we would be using passphrase. Then about the location, default location of id\_rsa.pub will be ~/.ssh/



*cat ~/.ssh/id\_rsa.pub*

Copy the previous command output and paste it in a notepad.



Similarly copy the contents from rest of the machines and paste it in notepad.

*cd ~/.ssh/*

*vi authorized\_keys*

Paste all the copied id\_rsa.pub from the notepad to

authorized\_keys files.



Do the same in all the machines

ssh each other machines and check the connectivity using the below commands

      ssh <first\_machine-hostname>

      ssh <second\_machine-hostname>

      ssh <third\_machine-hostname>

**Exercise – 2 – Install Java 1.8.x and configure it**

1. Check the installation of java using the command **Java –version**. At this point java will not be installed in the machine.
2. To install Java, execute **yum install java-1.8.0-openjdk\*.** Press Y on the prompts
3. Do the same on all the machines.
4. Configure the java using **sudo /usr/sbin/alternatives --config java** and press 1
5. Do the same in all the machines
6. Configure javac using **sudo /usr/sbin/alternatives --config javac** and press 1
7. Do the same in all the machines
8. Set JAVA\_HOME. Open **vi ~/.bashrc** file
9. Append **export JAVA\_HOME="/usr/lib/jvm/jre-1.8.0-openjdk"** to the file and save it.
10. Execute **source ~/.bashrc** which will update the changes.
11. Now check the JAVA\_HOME setup by executing **echo $JAVA\_HOME**

Do the same in all the machines

**Exercise 3 – Install and Configure Python**

1. Install GCC using **yum install gcc openssl-devel bzip2-devel**
2. Prompt Y while running and do the same in all the machines
3. Go to /usr/src folder. Execute **cd /usr/src**
4. Verify the folder using pwd. It should show /usr/src
5. Install wget using **yum install wget**
6. Download python using **wget** [**https://www.python.org/ftp/python/2.7.14/Python-2.7.14.tgz**](https://www.python.org/ftp/python/2.7.14/Python-2.7.14.tgz)
7. Type **ls** . You will see a file name Python-2.7.14.tgz
8. Extract python using the command **tar –xvf Python-2.7.14.tgz**
9. Navigate to Python directory using **cd Python-2.7.14**
10. Verify the present working director using **pwd.** It should possess output /usr/src/Python-2.7.14
11. Execute the command **./configure --enable-optimizations** to configure python
12. Execute **make altinstall** to prevent replacing the default python lib file
13. Verify the python installation using **/usr/local/bin/python2.7 –V**
14. Do all these steps in other machines

**Exercise 4 - Install Scala**

1. Navigate to /usr/src folder using **cd /usr/src**
2. Download scala rpm packages using **wget** [**https://downloads.lightbend.com/scala/2.12.2/scala-2.12.2.rpm**](https://downloads.lightbend.com/scala/2.12.2/scala-2.12.2.rpm)
3. Install scala using **sudo yum install scala-2.12.2.rpm**
4. Verify scala installation using **scala –version**
5. Do the same in all the machines

**Exercise 5 – Install Confluent Kafka 5.3**

1. Install **curl** and **which** tools on all the machines

sudo yum install curl which

1. Install the Confluent Platform public key. This key is used to sign packages in the YUM repository

sudo rpm --import <https://packages.confluent.io/rpm/5.3/archive.key>

1. Create a repo file for confluent inside /etc/yum.repos.d in the name of confluent.repo

cd /etc/yum.repos.d/

vi confluent.repo

Paste the below lines in confluent.repo on all the machines

[Confluent.dist]

name=Confluent repository (dist)

baseurl=https://packages.confluent.io/rpm/5.3/7

gpgcheck=1

gpgkey=https://packages.confluent.io/rpm/5.3/archive.key

enabled=1

[Confluent]

name=Confluent repository

baseurl=https://packages.confluent.io/rpm/5.3

gpgcheck=1

gpgkey=https://packages.confluent.io/rpm/5.3/archive.key

enabled=1

1. Clear the yum cache and install confluent platform

sudo yum clean all && sudo yum install confluent-platform-2.12

**Exercise 6 - Configure the zookeeper, Kafka and control centre properties**

1. Go to /etc/kafka/zookeeper.properties
2. Configure the zookeeper server properties with id, leader port and election port in the below format on all the machines

tickTime=2000

dataDir=/var/lib/zookeeper

# the port at which the clients will connect

clientPort=2181

initLimit=5

syncLimit=2

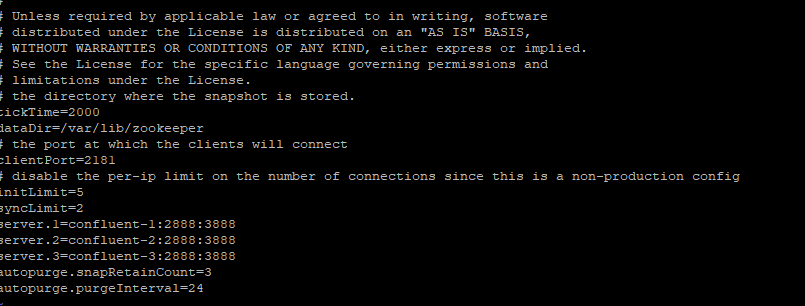
server.1=confluent-1:2888:3888

server.2=confluent-2:2888:3888

server.3=confluent-3:2888:3888

autopurge.snapRetainCount=3

autopurge.purgeInterval=24



1. Create a file myid inside /var/lib/zookeeper and assign the machine ID. For example confluent-1 should have 1, confluent-2 should have 2 and confluent-3 should have 3. Make sure you are cross verifying it once again.
2. Now let’s start configuring kafka properties. Open the file /etc/kafka/server.properties
3. Change the zookeeper.connect property to the below format

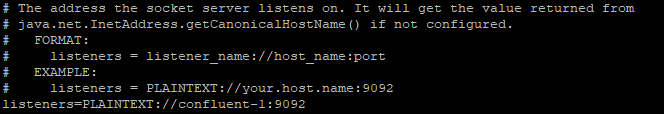


1. Comment out the config broker.id = 0 and append the below config

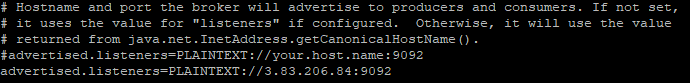
broker.id.generation.enable=true



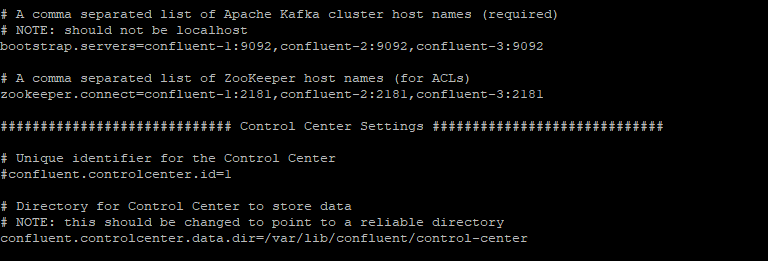
1. Configure the listener using the hostname:9092 or private\_ip:9092



1. Configure the advertised.listeners using the public IP address, so that it communicates with the local and external agents

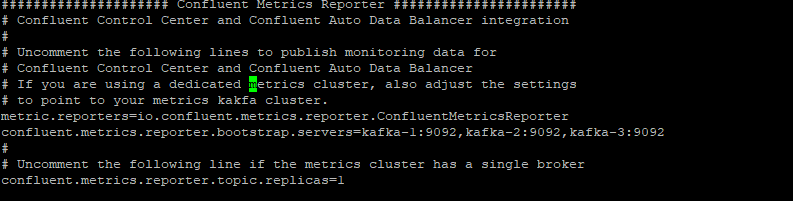


1. Next step is to configure confluent control centre
2. Go to the file /etc/confluent-control-center/control-center-production.properties
3. Configure the kafka bootstrap and zookeeper properties in the below mentioned format



1. Go to /etc/kafka/server.properties file and change the confluent.metrics.reporter.bootstrap.servers config to all the three hosts and uncomment the confluent.metrics.reporter.topic.replicas line

Also please uncomment the line metric.reporters=io.confluent.metrics.reporter.ConfluentMetricsReporter

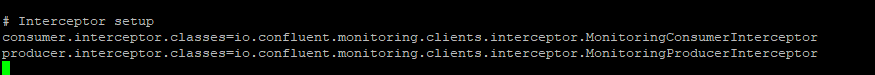


1. Add the below lines in/etc/kafka/connect-distributed.properties

# Interceptor setup

consumer.interceptor.classes=io.confluent.monitoring.clients.interceptor.MonitoringConsumerInterceptor

producer.interceptor.classes=io.confluent.monitoring.clients.interceptor.MonitoringProducerInterceptor



1. Start the zookeeper, kafka and confluent control centre one by one.
2. Start Zookeeper on all the machines using the below command

sudo systemctl start confluent-zookeeper

1. Start Confluent kafka using the below command on all the machines

sudo systemctl start confluent-kafka

1. Execute the below command to start the confluent control center

systemctl start confluent-control-center

Verify the status of the control center using the below command

service confluent-control-center status

1. Once the confluent control center s started. Use the url to view control center

https://<host-1 PublicIP>:9021