CDP Production Deployment

****

CDP production deployment

****

-----*** Create a VPC ***-----

## Create a elastic ip

## Create VPC with a public and a private subnet

## Create two security groups

-> In default security group

All traffic	your-ip/32
All traffic	private-subnet-cidr/24

-> in second security group (allowing communication only from edge)

All traffic	default-sec-group-id
All traffic	private-subnet-cidr/24

-----*** Launch public nodes ***-----

-----*** Update the server ***-----

```
$ sudo apt-get update && sudo apt-get dist-upgrade -y
```

-----*** Install NTP ***-----

```
$ sudo apt-get install ntp -y
$ sudo service ntp status
## If it isn't running
$ sudo service ntp start
```

-----*** Set Swappiness ***-----

```
sudo sysctl -a | grep vm.swappiness
sudo sysctl vm.swappiness=1
echo 'vm.swappiness=1' | sudo tee --append /etc/sysctl.conf
```

```
$ sudo nano /etc/init.d/disable-transparent-hugepages
```

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          disable-transparent-hugepages
# Required-Start:    $local_fs
# Required-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Disable Linux transparent huge pages
# Description:       Disable Linux transparent huge pages, to improve
#                    database performance.
### END INIT INFO

case $1 in
  start)
    if [ -d /sys/kernel/mm/transparent_hugepage ]; then
      thp_path=/sys/kernel/mm/transparent_hugepage
    elif [ -d /sys/kernel/mm/redhat_transparent_hugepage ]; then
      thp_path=/sys/kernel/mm/redhat_transparent_hugepage
    else
      return 0
    fi

    echo 'never' > ${thp_path}/enabled
    echo 'never' > ${thp_path}/defrag

    unset thp_path
    ;;
esac
```

```
$ sudo chmod 755 /etc/init.d/disable-transparent-hugepages
```

```
$ sudo update-rc.d disable-transparent-hugepages defaults
```

>>Restart server

-----*** Setting root reserved space ***-----

```
$ sudo lsblk
$ sudo tune2fs -l /dev/nvme0n1p1
$ sudo tune2fs -l /dev/nvme0n1p1 | egrep "Block count|Reserved block count"
$ sudo tune2fs -m 1 /dev/nvme0n1p1
$ sudo tune2fs -l /dev/nvme0n1p1 | egrep "Block count|Reserved block count"
```

### Note : Now, save the instance to an image, call it “Cloudera Manager” Make sure to check “No reboot”

-----*** Launch cluster nodes ***----

Launch reamining instances in private subnet

Select second security group for these instances

##### Preparing external database for cloudera #####

-----*** Install MySQL ***-----

```
sudo apt-get install mysql-server -y
```

-----*** Stop mysql ***-----

```
sudo service mysql stop

sudo service mysql status
```

-----*** Move innodb log file ***-----

```
mkdir mysqlbup

sudo su

cd /var/lib/mysql

 mv ib_logfile0 /home/ubuntu/mysqlbup/

 mv ib_logfile1 /home/ubuntu/mysqlbup/
```

```
exit
```

-----*** Edit the option file mysqld.cnf to cloudera recommended settings ***-----

```
cd /etc/mysql/mysql.conf.d/

sudo nano mysqld.cnf
```

===

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security risks;
# to do so, uncomment this line:
symbolic-links = 0

key_buffer_size = 32M
max_allowed_packet = 16M
thread_stack = 256K
thread_cache_size = 64
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1

max_connections = 550
#expire_logs_days = 10
#max_binlog_size = 100M

#log_bin should be on a disk with enough free space.
#Replace '/var/lib/mysql/mysql_binary_log' with an appropriate path for your
#system and chown the specified folder to the mysql user.
log_bin=/var/lib/mysql/mysql_binary_log

#In later versions of MySQL, if you enable the binary log and do not set
#a server_id, MySQL will not start. The server_id must be unique within
#the replicating group.
server_id=1

binlog_format = mixed

read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
```

```
join_buffer_size = 8M

# InnoDB settings
innodb_file_per_table = 1
innodb_flush_log_at_trx_commit  = 2
innodb_log_buffer_size = 64M
innodb_buffer_pool_size = 4G
innodb_thread_concurrency = 8
innodb_flush_method = O_DIRECT
innodb_log_file_size = 512M

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

sql_mode=STRICT_ALL_TABLES
```

==

-----*** Make mysql start on boot ***----

```
sudo update-rc.d mysql defaults
```

-----*** Start mysql ***-----

```
sudo service mysql start
```

-----*** Mysql secure install ***----

```
$ sudo /usr/bin/mysql_secure_installation
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] y
New password:
Re-enter new password:
Remove anonymous users? [Y/n] Y
[...]
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
All done!
```

-----*** Install mysql JDBC driver ***-----

```
sudo apt-get install libmysql-java
```

#### Establish Your Cloudera Manager Repository #####

```
$ wget https://archive.cloudera.com/cm7/7.1.4/ubuntu1804/apt/cloudera-manager-trial.list

$ sudo mv cloudera-manager-trial.list /etc/apt/sources.list.d/

$ sudo apt-get update

$ sudo apt-key adv --recv-key --keyserver keyserver.ubuntu.com 73985D43B0B19C9F

$ sudo apt-get update
```

##### Install Cloudera Manager server software #####

---** Install Oracle JDK **---

```
$ sudo apt-get install openjdk-8-jdk -y
```

---** Install Cloudera Manager packages **---

```
$ sudo apt-get install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server
-y
```

##### Setting up the Cloudera Manager Server Database #####

```
sudo mysql -u root

mysql>CREATE DATABASE cmdb;
mysql> CREATE USER 'cm'@'localhost' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'cm'@'localhost' IDENTIFIED BY 'password' WITH GRANT
OPTION;
mysql> exit;

sudo /opt/cloudera/cm/schema/scm_prepare_database.sh -p mysql cmdb cm password
```

##### Start the Cloudera Manager Server #####

```
$ sudo service cloudera-scm-server start

$ sudo service cloudera-scm-agent start

$ sudo service cloudera-scm-agent status
```

## Login to CM web UI on 7180

After assigning service roles to hosts, when you are on database setup page, do following

##### Create database for Cloudera services #####

```
sudo mysql -u root
```

>> For reports manager

```
mysql> create database rman DEFAULT CHARACTER SET utf8;
mysql> CREATE USER 'rman'@'ip-172-31-30-152.us-west-1.compute.internal' IDENTIFIED BY
'password';
```

(Note : Enter pri-dns of host where the particular service is running)

```
mysql> grant all on rman.* TO 'rman'@'ip-172-31-30-152.us-west-1.compute.internal' IDENTIFIED BY
'password';
```

>> For Hive

```
mysql> create database metastore DEFAULT CHARACTER SET utf8;
mysql> CREATE USER 'hive'@'ip-172-31-27-181.us-west-1.compute.internal' IDENTIFIED BY
'password';
mysql> grant all on metastore.* TO 'hive'@'ip-172-31-27-181.us-west-1.compute.internal'
IDENTIFIED BY 'password';
```

>> For Oozie

```
mysql> create database oozie default character set utf8;
mysql> CREATE USER 'oozie'@'ip-172-31-27-181.us-west-1.compute.internal' IDENTIFIED BY
'password';
mysql> grant all privileges on oozie.* to 'oozie'@'ip-172-31-27-181.us-west-1.compute.internal'
identified by 'password';
```

== Add mysql driver jar

```
sudo cp /usr/share/java/mysql-connector-java-5.1.38.jar /opt/cloudera/parcels/CDH/lib/oozie/lib/
```

##### Install Mysql jdbc jar on hosts where services are running #####

```
sudo apt-get install libmysql-java

sudo cp /usr/share/java/mysql-connector-java-5.1.45.jar /opt/cloudera/parcels/CDH/lib/oozie/lib/
```

##### On database host

```
sudo su

cd /etc/mysql/mysql.conf.d/

sudo nano mysqld.cnf

bind-address=(private-ip-of-cm) OR (0.0.0.0)

$ sudo service mysql restart
```

## Now continue installation on CM web ui

-----* Set up Edge node ***-----**

Add gateway roles on CM node

test whether read and write is working

## Run a test job

```
yarn jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 100
```

## Set up Jumper node to access webui of daemons