**Cloudera Hadoop**

**Instalação do Cluster CDH PathB no CentOS7.x**

**Apresentação e considerações**

Nesse documento vamos abordar a instalação inicial de um Cluster Cloudera Hadoop com todos os passos para vê-lo funcionando, iremos fazer toda a preparação das máquinas até o Login no Cloudera Manager no browser!

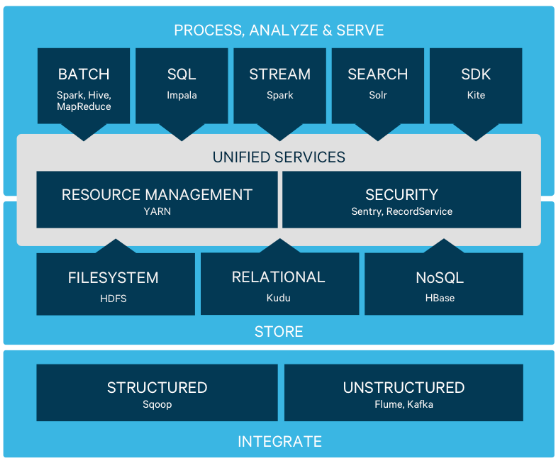
**O que é o Hadoop**

O Hadoop é uma plataforma de software em Java de computação distribuída voltada para clusters e processamento de grandes massas de dados. Foi inspirada no MapReduce e no GoogleFS (GFS). Trata-se de um projeto da Apache de alto nível, que vem sendo construído por uma comunidade de contribuidores e também por grandes empresas.

Ele é projetado para escalar de servidores individuais para milhares de máquinas, cada uma oferecendo computação e armazenamento local.

Para oferecer disponibilidade a própria biblioteca é projetada para detectar e lidar com falhas na camada do aplicativo, ele se aproveita de hardwares convencionais de baixo custo, como componentes principais temos o HDFS, Yarn e MapReduce.

Como as ferramentas do ecossistema Hadoop se encaixam e se relacionam:



**O que é Cloudera?**

Empresa fundada em 2008, é uma das principais fornecedoras do Hadoop a nível mundial, digamos que a Cloudera está para o Hadoop, assim como a Red Hat está para o Linux, ela possui uma interface de gerenciamento e orquestração do Cluster e de todos os serviços que o compõe de forma mais intuitiva com o CDH, existe uma versão gratuita, com suas limitações, com a aquisição da subscrição usufruímos do suporte e das inúmeras features de administração que só estão disponíveis na versão enterprise.

**Ponto de atenção – DNS Server**

**Nota 1:** É importante um DNS Server configurado adequadamente, inclusive o mesmo deve ter a Zona Reversa (PTR) configurada e responsiva, porém se não tiver, pode-se configurar o arquivo /etc/hosts para que todas as máquinas do cluster se conversem por nome, lembrando que a configuração precisa estar em TODAS!

**Nota 2:** As máquinas precisam estar com seu FQDN (Fully Qualified Domain Name) configurado adequadamente, ao digitar o comando hostname -f precisa retornar o nome completo, faça essa validação

**Cloudera Like a Boss**

Link interessante para ler sobre topologia e algumas recomendações do ambiente:

<http://blog.cloudera.com/blog/2015/01/how-to-deploy-apache-hadoop-clusters-like-a-boss/>

**Preparação do Sistema Operacional**

Antes de iniciar a instalação do cluster propriamente dito, realizar alguns ajustes no sistema operacional seguindo as recomendações e melhores práticas recomendadas pela Cloudera.

Os procedimentos realizados nessa sessão devem ser realizados em TODOS os NODES do Cluster, tem alguns itens não necessariamente obrigatórios, mesmo assim fazer em todos.

**Ajuste dos parâmetros do Kernel no /etc/sysctl.conf**

Mudar a prioridade de utilização da memória **swap**, para que ela não seja utilizada e setar alguns parâmetros de tunning do S.O para melhor desempenho dos servidores:

~# echo "vm.max\_map\_count=262144" >> /etc/sysctl.conf

~# echo "fs.file-max=262144" >> /etc/sysctl.conf

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

~# echo 1 > /proc/sys/vm/swappiness

**Ajuste dos limits**

O Hadoop trabalha com uma massa extremamente grande dados, é importante ajustar os limites soft e hard para números de arquivos abertos (nofile) e números de processos em execução (nproc) para os usuários.

~# vim /etc/security/limits.conf

\* soft nofile 64000

\* hard nofile 64000

\* soft nproc 64000

\* hard nproc 64000

**Desabilitar o “tuned” Service**

O Tuned é um daemon que monitora o uso de componentes do sistema e melhora dinamicamente as configurações do sistema com base nessas informações de monitoramento, esse será configurado posteriormente via Cloudera Manager

~# systemctl stop tuned; systemctl disable tuned

**Desabilitando o Firewall e o SE Linux**

É necessário desabilitar o SELinux para o funcionamento do Cloudera, o Firewall não é necessariamente obrigatório desativar, porém utilizá-lo com o Cloudera Hadoop acaba deixando a administração extremamente mais complexa, seguimos somente desativando e cuidamos da segurança em outras camadas.

~# systemctl disable firewalld; systemctl stop firewalld

~# sed -ri 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config

~# setenforce 0

**Instalar NTP**

Os clusters precisam estar com os horários devidamente sincronizados, para isso vamos ajustar o NTP

~# yum install ntp -y

Vamos editar o arquivo de configuração, aqui você pode definir o NTP Server da sua rede ou um público de sua confiança

~# vim /etc/ntp.conf

Adicionar essa linha apontando para um servidor público confiável de NTP, caso possua servidores NTP dentro do seu local de instalação opte por utilizá-los.

server pool.ntp.br iburst

Iniciar o serviço, habilitar na inicialização e validar

~# systemctl start ntpd; systemctl enable ntpd

~# ntpq -pn

**Instar o NSCD**

O NSCD é para acelerar a resolução de nomes fazendo cache.

~# yum install nscd -y

~# systemctl start nscd; systemctl enable nscd

**Instalação do Oracle Java JDK 8**

Consulte a documentação para validar a última versão recomendada e homologada de acordo com a versão do CDH.

<https://www.cloudera.com/documentation/enterprise/release-notes/topics/rn_consolidated_pcm.html#concept_ihg_vf4_j1b>

Seguindo a documentação, para a última versão “Recommended/Latest version tested”, utilize o link abaixo:

wget -c --header "Cookie: oraclelicense=accept-securebackup-cookie" http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.rpm

~# rpm -ivh jdk-8u131-linux-x64.rpm

**Desabilitar transparent hugepage**

~# echo never > /sys/kernel/mm/transparent\_hugepage/defrag

~# echo never > /sys/kernel/mm/transparent\_hugepage/enabled

Para desabilitar o Transparent HugePage de forma permanente, basta editar o arquivo vim /etc/grub2.cfg, adicione transparent\_hugepage=never ao

Final da linha 102 ficará da seguinte forma:

|  |  |
| --- | --- |
| 1 | linux16 /boot/vmlinuz-3.10.0-693.5.2.el7.x86\_64 root=UUID=6f15c206-f516-4ee8-a4b7-89ad880647db ro console=tty0 console=ttyS0,115200n8 crashkernel=auto console=ttyS0,115200 LANG=en\_US.UTF-8 transparent\_hugepage=never |

**Instalando o driver JDBC para o MariaDB/MySQL**

*Nota: Não use o comando yum install para instalar o pacote do driver do MySQL/MariaDB, porque ele instala o openJDK e usa o comando Linux alternatives para configurar o JDK do sistema como openJDK.*

~# wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.42.tar.gz

~# tar xf mysql-connector-java-5.1.42.tar.gz

~# mkdir -p /usr/share/java/

~# cp mysql-connector-java-5.1.42/mysql-connector-java-5.1.42-bin.jar /usr/share/java/

~# ln -s /usr/share/java/mysql-connector-java-5.1.42-bin.jar /usr/share/java/mysql-connector-java.jar

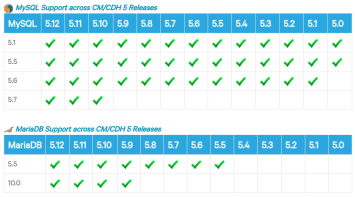
~# rm -rfv mysql-connector-java-5.1.42

**Finalizando esses passos “reboot” o servidor para ter certeza que está tudo ok!**

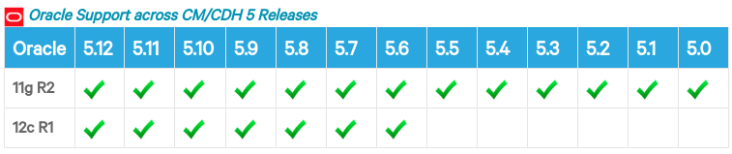
**>> Preparação do Banco de Dados**

Vamos precisar de um banco relacional para instalar os componentes do Hadoop, e nesse banco também ficarão armazenados todos os metadados referentes aos apontamentos dos namenodes para a localização dos arquivos nos datanodes e por ai vai.

Nessas imagens abaixo temos as versões de Banco de Dados suportadas pelas Cloudera e que você pode optar por utilizar.

[](https://churrops.io/2017/05/10/%ef%bb%bfcloudera-hadoop-instalacao-do-cluster-cdh5-pathb-no-centos7-x-parte1/chh-databases-mysq-mariadb/)

[](https://churrops.io/2017/05/10/%ef%bb%bfcloudera-hadoop-instalacao-do-cluster-cdh5-pathb-no-centos7-x-parte1/chh-databases-postgres/)

[](https://churrops.io/2017/05/10/%ef%bb%bfcloudera-hadoop-instalacao-do-cluster-cdh5-pathb-no-centos7-x-parte1/chh-databases-oracle/)

Em nossa instalação vamos utilizar o MariaDB antes de iniciar segue algumas recomendações

* Fazer backup do database é indispensável;
* É recomendável que se utilize no mínimo a feature de replicação Master/Slave, ou algum tipo de clusterização do seu database para que seu ambiente de fato possua alta disponibilidade;
* O Banco de dados pode ser instalado nos nodes nomeados como master1 e master2 do seu ambiente, e é isso que vamos fazer!

Estando logado no servidor elegido como Master, vamos iniciar a instalação, e aqui em nosso Lab vamos utilizar uma single instance de MariaDB mesmo, para maiores informações sobre o replication clique [aqui](https://churrops.io/2017/05/05/mariadb-configurando-o-replication-masterslave/).

O MariaDB já vem por padrão nos repositórios RHEL/CentOS.

~# yum install mariadb-server -y

Antes de iniciar o serviço vamos sobrepor o arquivo de configuração padrão do MariaDB com o conteúdo abaixo e em seguida iniciá-lo.

~# > /etc/my.cnf

~# vim /etc/my.cnf

[mysqld]

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 = 16M

key\_buffer\_size = 32M

max\_allowed\_packet = 32M

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

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

### Esse parâmetro deve ser ajustado de acordo com a memória RAM do sistema

### em um Cluster de produção recomenda-se pelo menos 4GB

#innodb\_buffer\_pool\_size = 1G

#innodb\_buffer\_pool\_size = 512M

#innodb\_thread\_concurrency = 8

#innodb\_flush\_method = O\_DIRECT

#innodb\_log\_file\_size = 512M

[mysqld\_safe]

log-error=/var/log/mariadb/mariadb.log

pid-file=/var/run/mariadb/mariadb.pid

~# systemctl enable mariadb; systemctl start mariadb

Nesse passo é importante definir uma senha de root para o seu banco.

~# mysql\_secure\_installation

Set root password? [Y/n] y

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

Criando as bases

Nota: O parâmetro “IDENTIFIED BY” é onde definimos a senha dos bancos que serão criados, portanto defina uma senha específica para o seu ambiente, documente-a pois precisaremos delas para configurar os serviços

Criei o arquivo abaixo, e em seguida copie e cole o conteúdo pois esse será o nosso script de criação dos databases necessários para prosseguirmos

~# vim /tmp/create-cloudera-db.sql

# Role: Activity Monitor

create database amon DEFAULT CHARACTER SET utf8;

grant all on amon.\* TO 'amon'@'%' IDENTIFIED BY 'amon\_pass';

# Role: Reports Manager

create database rman DEFAULT CHARACTER SET utf8;

grant all on rman.\* TO 'rman'@'%' IDENTIFIED BY 'rman\_pass';

# Role: Hive Metastore Server

create database metastore DEFAULT CHARACTER SET utf8;

grant all on metastore.\* TO 'hive'@'%' IDENTIFIED BY 'hive\_pass';

# Role: Sentry Server

create database sentry DEFAULT CHARACTER SET utf8;

grant all on sentry.\* TO 'sentry'@'%' IDENTIFIED BY 'sentry\_pass';

# Role: Cloudera Navigator Audit Server

create database nav DEFAULT CHARACTER SET utf8;

grant all on nav.\* TO 'nav'@'%' IDENTIFIED BY 'nav\_pass';

# Role: Cloudera Navigator Metadata Server

create database navms DEFAULT CHARACTER SET utf8;

grant all on navms.\* TO 'navms'@'%' IDENTIFIED BY 'navms\_pass';

# Role: Cloudera Hue

create database hue DEFAULT CHARACTER SET utf8;

grant all on hue.\* TO 'hue'@'%' IDENTIFIED BY 'hue\_pass';

# Role: Cloudera Oozie

create database oozie default character set utf8;

grant all privileges on oozie.\* to 'oozie'@'%' identified by 'oozie\_pass';

Vamos criar as bases, basta executar o comando abaixo e inserir a senha do usuário root do banco de dados quando solicitado

~# mysql -uroot -p < /tmp/create-cloudera-db.sql

~# rm -f /tmp/create-cloudera-db.sql

Verificando as bases que foram criadas

~# mysql -uroot -p

MariaDB [(none)]> show databases;

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

| Database |

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

| information\_schema |

| amon |

| hue |

| metastore |

| mysql |

| nav |

| navms |

| oozie |

| performance\_schema |

| rman |

| sentry |

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

11 rows in set (0.00 sec)

**>> Instalando o Cloudera Manager Server**

Agora vamos instalar de fato o Cloudera Manager que é a interface responsável pela orquestração e gerenciamento do nosso Cluster

~# yum clean all

~# cd /etc/yum.repos.d/ ; wget https://archive.cloudera.com/cm5/redhat/7/x86\_64/cm/cloudera-manager.repo

~# yum install cloudera-manager-daemons cloudera-manager-server -y

Preparando o Cloudera Manager Server External Database, para isso vamos criar um usuário temp com permissão total no banco

~# echo "grant all on \*.\* to 'temp'@'%' identified by 'temp' with grant option;" | mysql -uroot -p

Agora vamos criar o database, no parâmetro -h e --scm-host são passados o endereço do banco de dados, como ele está na máquina master estou passando localhost mesmo

~# /usr/share/cmf/schema/scm\_prepare\_database.sh mysql -h 127.0.0.1 -utemp -ptemp --scm-host 127.0.0.1 scm scm scm

Feito isso, vamos dropar o usuário temp

~# echo "drop user 'temp'@'%';" | mysql -uroot -p

Com todos os passos acima realizados, agora só precisamos iniciar o serviço

~# service cloudera-scm-server start

~# systemctl enable cloudera-scm-server

Acompanhando nos Logs o start do Serviço, se estiver tudo ok, aparecerá no final uma saída semelhante a essa:

~# tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log

...

....

....

2017-05-18 00:43:00,891 INFO SearchRepositoryManager-0:com.cloudera.server.web.cmf.search.components.SearchRepositoryManager: Constructing repo:2017-05-18T03:43:00.891Z

2017-05-18 00:43:03,102 INFO SearchRepositoryManager-0:com.cloudera.server.web.cmf.search.components.SearchRepositoryManager: Finished constructing repo:2017-05-18T03:43:03.102Z

2017-05-18 00:43:04,015 INFO WebServerImpl:org.mortbay.log: jetty-6.1.26.cloudera.4

2017-05-18 00:43:04,054 INFO WebServerImpl:org.mortbay.log: Started SelectChannelConnector@0.0.0.0:7180

2017-05-18 00:43:04,054 INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty server.

Vis

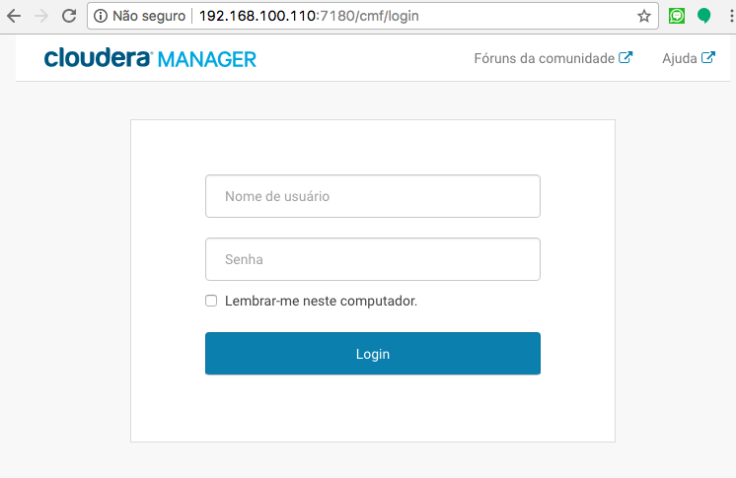
ualizando a porta em LISTEN no sistema

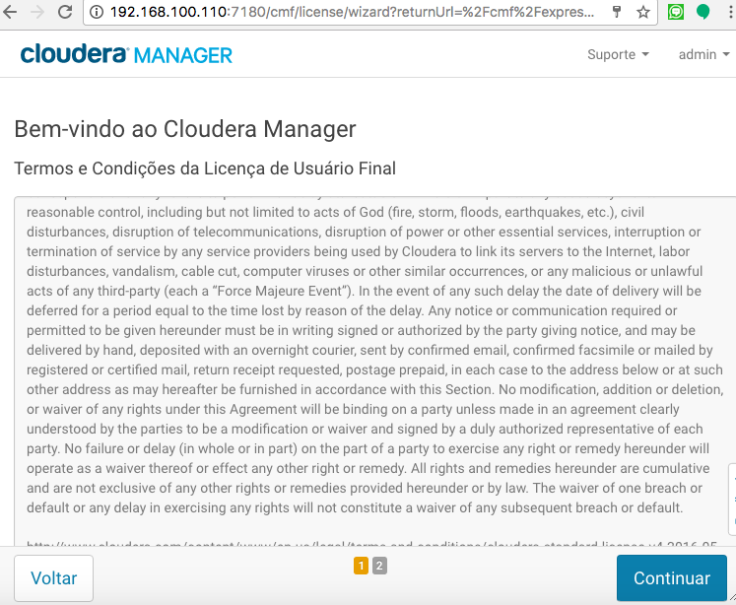
~# netstat -anput |grep 7180 |grep LIST

tcp 0 0 0.0.0.0:7180 0.0.0.0:\* LISTEN 16130/java

**Acessando via browser**

[http://192.168.100.110:7180](http://192.168.100.110:7180/)  
User: admin  
Pass: admin





No nosso caso vamos selecionar o “Cloudera Express”que é a versão FREE

