# Instalação do Cloudera Manager – Debian 8 com postgresql 9.6

## Versão Unix

lsb\_release -a

No LSB modules are available.

Distributor ID: Debian

Description: Debian GNU/Linux 8.9 (jessie)

Release: 8.9

Codename: jessie

## Atualizar o sistema operacional

*apt-get update*

## Instalação Cloudera master node

*apt-get install cloudera-manager-daemons cloudera-manager-server cloudera-manager-agent*

*dpkg -l |grep cloudera*

*ii cloudera-manager-agent 5.16.1-1.cm5161.p0.1~jessie-cm5 amd64 The Cloudera Manager Agent*

*ii cloudera-manager-daemons 5.16.1-1.cm5161.p0.1~jessie-cm5 all Provides daemons for monitoring Hadoop and related tools.*

*ii cloudera-manager-server 5.16.1-1.cm5161.p0.1~jessie-cm5 all The Cloudera Manager Server*

## Instalação dos slave nodes

*dpkg -l |grep cloudera*

*ii cloudera-manager-agent 5.16.1-1.cm5161.p0.1~jessie-cm5 amd64 The Cloudera Manager Agent*

*ii cloudera-manager-daemons 5.16.1-1.cm5161.p0.1~jessie-cm5 all Provides daemons for monitoring Hadoop and related tools.*

## Instalação oracle jdk

*apt-get install cloudera-manager-daemons cloudera-manager-agent*

Feito na mão download do (jdk-8u201-linux-x64.tar.gz) do site da oracle <https://www.oracle.com/technetwork/es/java/javase/downloads/jdk8-downloads-2133151.html?printOnly=1>

Descompactar o tar.gz

*tar -xzvf jdk-8u171-linux-x64.tar.gz*

*Editar o arquivo .bashrc*

*vi ~/.bashrc*

*export JAVA\_HOME=/usr/java/jdk1.8,0\_181*

*export PATH=$PATH:/usr/java/jdk1.8.0\_181/bin*

*source ~/.bashrc*

*dpkg -l |grep oracle*

*ii oracle-j2sdk1.7 1.7.0+update67-1 amd64 no description given*

## Instalação postgresql

*apt-get install postgresql-server-9.6*

*root@c3n1:/etc/ssh# dpkg -l |grep postgresql-*

*ii postgresql-9.6 9.6.12-1.pgdg80+1 amd64 object-relational SQL database, version 9.6 server*

*ii postgresql-client-9.6 9.6.12-1.pgdg80+1 amd64 front-end programs for PostgreSQL 9.6*

*ii postgresql-client-common 199.pgdg80+1 all manager for multiple PostgreSQL client versions*

*ii postgresql-common 199.pgdg80+1 all PostgreSQL database-cluster manager*

*ii postgresql-contrib-9.6 9.6.12-1.pgdg80+1 amd64 additional facilities for PostgreSQL*

*ii postgresql-server-dev-9.6 9.6.12-1.pgdg80+1 amd64 development files for PostgreSQL 9.6 server-side programming*

*root@c3n1:/etc/ssh#*

*checar conexão*

*su –postgres*

*psql*

## Alterar o arquivo de configuração postgresql

*vi /etc/postgresql/9.6/main/pg\_hba.conf*

*host all all 0.0.0.0/0 trust*

*start no serviço do postgresql*

*systemctl restart postgresql*

*apt-get update*

*apt-get install postgresql-9.6*

*zypper install python-psycopg2*

*apt-get install python-pip*

*pip install psycopg2*

## Alterar hosts

*more /etc/hosts*

*127.0.0.1 localhost*

*#127.0.1.1 c3n1.lab.local c3n1*

*#All nodes*

*192.168.100.249 c3n1.compwire.local c3n1*

*192.168.100.248 c3n2.compwire.local c3n2*

*192.168.100.247 c3n3.compwire.local c3n3*

*192.168.100.246 c3n4.compwire.local c3n4*

## Alterar sshd\_config para permitir o login do root por ssh

*PermitRootLogin yes*

## Instalar o NTP

Instalar os pacotes

* apt-get install ntp
* apt-get install ntp ntpdate
* Editar o arquivo /etc/ntp.conf e alterar os servidores do NTP para:
* server a.ntp.br
* server b.ntp.brdp
* server c.ntp.br
* Habilitar o serviço
* systemctl enable ntpd.service
* Ativar o serviço NTP
* systemctl start ntpd.service
* Sincronizar o node
  + ntpdate -u a.ntp.org
* Remover o Chrony do SO
  + yum erase chrony

## Alterar o swapiness

* Alterar em tempo de execução
  + sysctl vm.swappiness=1
* Incluir no arquivo /etc/sysctl.conf
  + vi /etc/sysctl.conf
  + vi /usr/lib/tuned/virtual-guess/tuned.conf

## Conectar na base de dados e criar as tabelas e grants pros serviços

(antes da primeira conexão no bd postgresql foi feito o reset da senha ALTER USER postgres with password 'password')

CREATE ROLE HIVE LOGIN PASSWORD 'PASSWORD';

CREATE DATABASE METASTORE OWNER HIVE ENCODING 'UTF8';

CREATE ROLE HUE LOGIN PASSWORD 'PASSWORD';

CREATE DATABASE HUE OWNER HUE ENCODING 'UTF8';

CREATE ROLE SENTRY LOGIN PASSWORD 'PASSWORD';

CREATE DATABASE SENTRY OWNER SENTRY ENCODING 'UTF8';

CREATE ROLE OOZIE LOGIN PASSWORD 'PASSWORD';

CREATE DATABASE OOZIE OWNER OOZIE ENCODING 'UTF8';

CREATE ROLE RMAN LOGIN PASSWORD 'PASSWORD';

CREATE DATABASE RMAN OWNER RMAN ENCODING 'UTF8';

CREATE ROLE RMAN LOGIN PASSWORD 'PASSWORD';

CREATE DATABASE RMAN OWNER RMAN ENCODING 'UTF8';

ALTER DATABASE METASTORE SET STANDARD\_CONFORMING\_STRINGS=OFF;

ALTER DATABASE OOZIE SET STANDARD\_CONFORMING\_STRINGS=OFF;

GRANT ALL PRIVILEGES ON DATABASE HUE TO HUE;

GRANT ALL PRIVILEGES ON DATABASE METASTORE TO HIVE;

GRANT ALL PRIVILEGES ON DATABASE SENTRY TO SENTRY;

GRANT ALL PRIVILEGES ON DATABASE OOZIE TO OOZIE;

GRANT ALL PRIVILEGES ON DATABASE RMAN TO RMAN;

GRANT ALL PRIVILEGES ON DATABASE RMAN TO RMAN;

Executar o script /usr/share/cmf/schema/scm\_prepare\_database.sh postgresql -upostgres -p --verbose scm scm scm

Arquivos sql para execução de criação e grants das tabelas nos banco de dados mariadb e mysql