# CDH6部署文档

[宝付网络科技(上海)有限公司] [上海市浦东新区居里路 99 号]

# 关于本文档

文档信息 文档名称 CDH6部署文档 作者 杨泽 审批者 说明 文件名称 修订历史 (REVISION HISTORY) 版本 章节 类型 日期 作者 备注 2018年11月19 杨泽 日 1.0 所有 创建 文档建立

# 内部资料 注意保密

1一、安装前准备4
 配置ssh免登陆5
 安装ansible5
 安装ansible/hosts5
 挂载磁盘5
 1.格式化:6
 2. 硬盘挂载:6
 3.查询是否成功:6
 修改所有节点主机名6
 修改所有节点主机名和IP映射关系6
 系统优化8
 关闭防火墙10

目录

<u>配置NTP服务 11</u> vi /etc/ntp.conf 11 vi /etc/ntp.conf 11 <u>JDK安装 12</u> vi /etc/profile 12 MYSQL安装 13 <u>3 MYSQL配置13</u> 4 创建库(后续安装服务等使用) 14 二、离线安装CM 14 <u>2.1 上传rpm文件到主节点(cdh-85-29)14</u> 2.2 安装Cloudera Manager Server 14 2.3 安装Cloudera Manager Agent 15 vi /etc/yum. repos. d/cloudera-manager. repo 16 3.3 登录管理界面 18 <u>3.1 上传文件 20</u> <u>3.4 配置服务角色 22</u> 四、Kerberos安装 24 4.1 Kerberos服务端安装 24 4.2 Kerberos客户端安装 25 4.3 关于AES-256加密 25 4.4 创建数据库 25 <u>4.5 启动服务 26</u> <u>4.6 创建Kerveros管理员 26</u> 4.7 CDH启用Kerberos 26 1.集群中有多少个节点,每个账户都会生成对应个数的principal 29 <u>2.为每个对应的principal创建keytab</u>29 3. 部署keytab文件到指定的节点中 29 4.在每个服务的配置文件中加入有关Kerberos的配置 29 4.8 创建HDFS超级用户 29 4.9 确认HDFS可以正常使用 30 五、 LDAP安装 30 o: baofoo Company 33 5.2 服务端配置 37 5.3 测试并生成配置文件 37 5.4 配置完成重启服务 38 <u>5.5 Kerberos和Ldap集成 38</u> 5.6 配置并迁移系统用户 39 5.7 LDAP客户端配置 39 <u>5.8 配置HIVE集成LDAP 40</u> 5.9 配置IMPALA集成LDAP 42 <u>5.10 配置HUE集成LDAP 42</u>

# 1一、安装前准备

PS: 此安装文档适用于CDH6.0版本,操作系统版本7以上。

# 文件准备

文件	存放路径	
Cloudera Manager 包	/opt/cm6/	
parcels	/opt/parcels	
JDK1.8	/opt/cm6/oracle-j2sdk1.8-1.8.0+update141- 1.x86_64.rpm	所有节点
MYSQL5.7	/opt/mysql/mysql-5.7.19-1.el7.x86_64.rpm- bundle.tar	
MYSQL_JDBC	/opt/mysql-jdbc/mysql-connector-java.jar	所有节点

#### 配置dns

Vi /etc/resolv.conf

nameserver 114.114.114.114

# 配置ssh免登陆

```
在172.20.85.29、172.20.85.39 上分别生成一对钥匙
ssh-keygen -t rsa
...
#将公钥拷贝到其他节点,包括自己
ssh-copy-id 172.20.85.29
...
```

ssh-copy-id -i ~/.ssh/id\_rsa.pub user@server

# 安装ansible

Yum install ansible

vi /etc/ansible/hosts

[cdh] 172. 20. 85. [10:59] [hive] 172.20.85.[10:44]

[hbase] 172.20.85.[45:59]

# 挂载磁盘

 格式化:
 mkfs -t xfs /dev/sdb
 硬盘挂载:
 主节点 : mount /dev/sdb /opt
 计算节点 : mount /dev/sdb /dfs/data1 mount /dev/sdc /dfs/data2

3. 查询是否成功:

df -HT

# 修改所有节点主机名

 ${\tt hostnamect1 \ set-hostname \ cdh85-29}$ 

ansible 172.20.85.10 -m shell -a "hostnamectl set-hostname cdh85-10 "

# 修改所有节点主机名和IP映射关系

vim /etc/hosts
172. 20. 85. 10 cdh85-10
172. 20. 85. 11 cdh85-11
172. 20. 85. 12 cdh85-12
172. 20. 85. 13 cdh85-13
172. 20. 85. 14 cdh85-14
172. 20. 85. 15 cdh85-15
172. 20. 85. 16 cdh85-16
172. 20. 85. 17 cdh85-17
172. 20. 85. 18 cdh85-18
172. 20. 85. 19 cdh85-19
172. 20. 85. 20 cdh85-20
172. 20. 85. 21 cdh85-21

172.20.85.22 cdh85-22 172.20.85.23 cdh85-23 172.20.85.24 cdh85-24 172.20.85.25 cdh85-25 172.20.85.26 cdh85-26 172.20.85.27 cdh85-27 172.20.85.28 cdh85-28 172.20.85.29 cdh85-29 172.20.85.30 cdh85-30 172.20.85.31 cdh85-31 172.20.85.32 cdh85-32 172.20.85.33 cdh85-33 172.20.85.34 cdh85-34 172.20.85.35 cdh85-35 172.20.85.36 cdh85-36 172.20.85.37 cdh85-37 172.20.85.38 cdh85-38 172.20.85.39 cdh85-39 172.20.85.40 cdh85-40 172.20.85.41 cdh85-41 172.20.85.42 cdh85-42 172.20.85.43 cdh85-43 172.20.85.44 cdh85-44 172.20.85.45 cdh85-45 172.20.85.46 cdh85-46 172.20.85.47 cdh85-47 172.20.85.48 cdh85-48 172.20.85.49 cdh85-49 172.20.85.50 cdh85-50 172.20.85.51 cdh85-51 172.20.85.52 cdh85-52 172.20.85.53 cdh85-53 172.20.85.54 cdh85-54 172.20.85.55 cdh85-55 172.20.85.56 cdh85-56 172.20.85.57 cdh85-57 172.20.85.58 cdh85-58 172.20.85.59 cdh85-59

Vi cpfile.yml

\_\_\_\_\_

- hosts: cdh tasks: - name: copy files

copy:

src: /etc/hosts
dest: /etc/hosts

ansible-playbook cpfile.yml

# 系统优化

Vi jiagu-baoxin-9-30.sh

#!/bin/bash echo "# 关闭防火墙和selinux" sed -i s'/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config setenforce 0 iptables -F iptables -X systemctl disable firewalld echo "# 更新yum源" cd /etc/yum.repos.d/ mv /etc/yum.repos.d/\*.repo /tmp/ rpm -ivh http://192.168.25.200/olinux/7/ol7\_u4\_base/getPackage/wget-1.14-15.el7.x86\_64.rpm wget http://192.168.25.200/file/baofoo-centos.repo -0 /etc/yum.repos.d/baofoo-centos.repo yum clean all yum makecache echo " # 调整时区" ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime yum install net-tools ntpdate -y echo " # 安装开发包组" yum groupinstall "Development Tools" -y echo ″ # 安装基础环境和常用工具包" yum -y make cmake gcc-c++ gcc zib zlib-devel lrzsz iftop dstat wget net-tools # 修改网卡为eth #cd /etc/sysconfig/network-scripts/ #mv ifcfg-eno16777728 ifcfg-eth0 //NAME=ethO #name修改为eth0 #sed -i s/"crashkernel=auto rhgb"/"crashkernel=auto rhgb net.ifnames=0 biosdevname=0"/g /etc/sysconfig/grub

#grub2-mkconfig -o /boot/grub2/grub.cfg echo ″ # 禁用不必要的服务" systemctl stop libvirtd. service rpcbind. service rpcbind. socket avahi-daemon. service avahidaemon. socket cups. path cups. service cups. socket postfix. service systemctl disable libvirtd. service rpcbind. service rpcbind. socket avahi-daemon. service avahidaemon. socket cups. path cups. service cups. socket postfix. service echo " # 安全加固" ###设置会话超时 echo "TMOUT=900">>/etc/profile ###设置umask echo "umask 027">>/etc/profile ###pam限制su echo "auth /lib64/security/pam wheel.so group=wheel">>/etc/pam.d/su required ###SSH配置 # echo "PermitRootLogin no">>/etc/ssh/sshd\_config #echo "Ciphers aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, aes192cbc, aes256-cbc, rijndael-cbc@lysator.liu.se">>/etc/ssh/sshd\_config sed -i 's/GSSAPIAuthentication yes/GSSAPIAuthentication no/' /etc/ssh/sshd\_config sed -i '/#UseDNS yes/a\UseDNS no' /etc/ssh/sshd\_config echo ″###SSH版本升级" yum -y update openssh-server openssh-clients openssl systemctl restart sshd echo " #内核优化" echo -e "\nnet.ipv4.tcp\_tw\_reuse = 1 \nnet.ipv4.tcp\_tw\_recycle = 1 \nnet.ipv4.tcp\_keepalive\_time = 1200 \nnet.ipv4.ip\_local\_port\_range = 10000 65000 \nnet.ipv4.tcp\_max\_syn\_backlog = 8192 \nnet.ipv4.tcp max tw buckets = 5000nfs.file-max = 655350\nnet.ipv4.route.gc\_timeout = 100 \nnet.ipv4.tcp\_syn\_retries = 1 \nnet.ipv4.tcp\_synack\_retries = 1 \nnet.core.netdev\_max\_backlog = 16384 \nnet.ipv4.tcp\_max\_orphans = 16384 \nnet.ipv4.tcp\_fin\_timeout = 2 \net.core.somaxconn=32768 \kernel.threads-max=196605 \kernel.pid\_max=196605 \vm.max\_map\_count=393210 \nvm.swappiness =  $0'' \rightarrow /etc/sysctl.conf$  && echo 'yes' /sbin/sysctl -p echo ″ #设置最大文件打开数 ulimit -a" sed -i '\$ a\\* soft nofile 196605' /etc/security/limits.conf sed -i '\$ a\\* hard nofile 196605' /etc/security/limits.conf echo "\* soft nproc 196605" >> /etc/security/limits.conf

echo "\* hard nproc 196605" >> /etc/security/limits.conf

```
echo ″ #关闭大页面"
echo never > /sys/kernel/mm/transparent_hugepage/defrag
echo 'echo never > /sys/kernel/mm/transparent_hugepage/defrag' >> /etc/rc.local
chmod +x /etc/rc.d/rc.local
echo " #时间同步"
  /usr/sbin/ntpdate 192.168.23.108
  echo '3 */1 * * * /usr/sbin/ntpdate time3.aliyun.com | logger -t NTP' >> /var/spool/cron/root
  echo '13 */1 * * * /usr/sbin/ntpdate 192.168.23.108 | logger -t NTP' >> /var/spool/cron/root
  cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
  /usr/sbin/ntpdate us.pool.ntp.org
  echo "30 22 * * * /usr/sbin/ntpdate us.pool.ntp.org" >> /var/spool/cron/root
  systemctl restart crond
echo ″ #多余用户和组清理"
  for user in adm lp sync shutdown halt uucp operator games gopher ftp;do passwd -1 $user;done
echo ″ #创建数据文件目录"
  for i in {1..10} ;do mkdir /dfs/data$i -p;done
  for i in {a..j};do /usr/sbin/mkfs.xfs /dev/sd$i;done
  echo -e "/dev/sdb /dfs/data1
                                     xfs defaults 0 0
  /dev/sdc /dfs/data2
                            xfs defaults 0 0
  /dev/sdd /dfs/data3
                            xfs defaults 0 0
  /dev/sde /dfs/data4
                             xfs defaults 0 0
  /dev/sdf /dfs/data5
                             xfs defaults 0 0
  /dev/sdg /dfs/data6
                             xfs defaults 0 0
  /dev/sdh /dfs/data7
                             xfs defaults 0 0
  /dev/sdi /dfs/data8
                            xfs defaults 0 0
  /dev/sdj /dfs/data9
                             xfs defaults 0 0" >> /etc/fstab
  mount -a
```

# 关闭防火墙

#查看防火墙状态 firewall-cmd --state

> #关闭防火墙 systemctl stop firewalld.service

#关闭防火墙开启启动

systemctl disable firewalld.service

#关闭SELINUX

setenforce 0 (临时生效)

修改/etc/selinux/config 下的 SELINUX=disabled (重启后生效)。

# 配置NTP服务

停止系统自带的ntp服务: systemctl stop chronyd 关闭开机自启动ntp服务: systemctl disable chronyd 查看系统自带ntp服务状态: systemctl status chronyd

所有节点安装相关组件: yum -y install ntp 配置开机启动: systemctl enable ntpd 检查是否设置成功: systemctl list-unit-files |grep ntpd

### 主节点配置(cdh85-29)

vi /etc/ntp.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
#server 101.231.72.162 prefer
#server time3.aliyun.com iburst
server 192.168.23.108 iburst
server ntpl.aliyun.com iburst

启动ntp服务: systemctl start ntpd

### 配置ntp客户端(其他所有节点)

vi /etc/ntp.conf

# Use public servers from the pool.ntp.org project. # Please consider joining the pool (http://www.pool.ntp.org/join.html). #server 0.rhel.pool.ntp.org iburst #server 1.rhel.pool.ntp.org iburst #server 2.rhel.pool.ntp.org iburst #server 3.rhel.pool.ntp.org iburst server cdh85-29 prefer server 192.168.23.108

ok保存退出,请求服务器前,请先使用ntpdate手动同步一下时间: ntpdate -u n1 (主节点ntp服务器) 启动服务: systemctl start ntpd 查看服务状态: ntpstat

# JDK安装

卸载系统自带的jdk:

查看: rpm -qa|grep -i jdk

卸载: rpm -e java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.x86\_64 -nodeps

安装jdk并配置环境变量,版本为1.8

rpm -ivh /opt/cm6/oracle-j2sdk1.8-1.8.0+update141-1.x86\_64.rpm

vi /etc/profile

JAVA\_HOME=/usr/java/jdk1.8.0\_141-cloudera CLASSPATH=\$JAVA\_HOME/lib/tools.jar PATH=\$JAVA\_HOME/bin:\$PATH

export JAVA\_HOME CLASSPATH PATH

source /etc/profile

# MYSQL安装

# https://www.cloudera.com/documentation/enterprise/latest/topics/cm ig mysql.html

wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm sudo yum update sudo yum install mysql-server sudo systemctl start mysqld

# https://blog.csdn.net/u014539401/article/details/78138292

cd /opt/mysql/ tar -xvf ./mysql-5.7.19-1.el7.x86\_64.rpm-bundle.tar

rpm -ivh mysql-community-common-5.7.19-1.el7.x86\_64.rpm --force --nodeps

```
rpm -ivh mysql-community-libs-5.7.19-1.el7.x86_64.rpm --force --nodeps
rpm -ivh mysql-community-client-5.7.19-1.el7.x86_64.rpm --force --nodeps
rpm -ivh mysql-community-server-5.7.19-1.el7.x86_64.rpm --force --nodeps
rpm -ivh mysql-community-libs-compat-5.7.19-1.el7.x86 64.rpm --force --nodeps
3 MYSQL配置
yum install -y libaio
mysqld -- defaults-file=/etc/my.cnf --initialize --user=mysql # 初始化mysql使mysql目录的拥有者为mysql用
户
cat /var/log/mysqld.log # 最后一行将会有随机生成的密码
systemctl start mysqld.service # 设置mysql服务自启
mysql -uroot -p
# 输入临时密码
ALTER USER 'root'@'localhost' IDENTIFIED BY 'root'; # 修改密码
4 创建库(后续安装服务等使用)
CREATE DATABASE cmserver DEFAULT CHARACTER SET utf8;
GRANT ALL on cmserver.* TO 'cmserveruser'@'%' IDENTIFIED BY 'password';
CREATE DATABASE metastore DEFAULT CHARACTER SET utf8;
GRANT ALL on metastore. * TO 'hiveuser'@'%' IDENTIFIED BY 'password';
CREATE DATABASE amon DEFAULT CHARACTER SET utf8;
GRANT ALL on amon.* TO 'amonuser'@'%' IDENTIFIED BY 'password';
CREATE DATABASE rman DEFAULT CHARACTER SET utf8;
GRANT ALL on rman. * TO 'rmanuser'@'%' IDENTIFIED BY 'password';
CREATE DATABASE oozie DEFAULT CHARACTER SET utf8;
GRANT ALL on oozie.* TO 'oozieuser'@'%' IDENTIFIED BY 'password';
CREATE DATABASE hue DEFAULT CHARACTER SET utf8;
GRANT ALL on hue. * TO 'hueuser'@'%' IDENTIFIED BY 'password';
```

# **安装Mysql客户端** 除主节点以外的其他节点 先卸载系统自带的mysql: yum remove -y mysql-community-common 安装mysql客户端: yum install -y mariadb

CentOS 安装MariaDB

# 二、离线安装CM

下载对应CDH版本的CM, 下载地址: http://archive.cloudera.com/cm5/redhat/

### 2.1 上传rpm文件到主节点(cdh-85-29)

#### 2.2 安装Cloudera Manager Server

在cdh85-29上运行:

yum -y localinstall cloudera-manager-daemons-6.0.0-530873.e17.x86\_64.rpm yum -y localinstall cloudera-manager-server-6.0.0-530873.e17.x86\_64.rpm yum -y localinstall yum -y localinstall cloudera-manager-agent-6.0.0-530873.e17.x86\_64.rpm

### 2.3 安装Cloudera Manager Agent

除cdh85-29以外的其他节点

yum -y localinstall cloudera-manager-daemons-5.12.1-1.cm5121.p0.6.el7.x86\_64.rpm
yum -y localinstall cloudera-manager-agent-5.12.1-1.cm5121.p0.6.el7.x86\_64.rpm

配置/etc/cloudera-scm-agent/config.ini修改server\_host= (Name of the host where Cloudera Manager Server is running.)

# # 执行初始化脚本

/opt/cloudera/cm/schema/scm\_prepare\_database.sh mysql -h 192.168.85.29 -uroot -p baofoo@64 --scmhost 192.168.85.29 cmserver root baofoo@64

# 启动进程

启动进程, 主节点(cdh85-29): systemctl start cloudera-scm-server systemctl start cloudera-scm-agent

启动进程,各子节点(除cdh85-29以外的所有节点): systemctl start cloudera-scm-agent

# 配置Cloudera Manager包yum源(主节点)

mkdir -p /var/www/html/cloudera-repos 将下载的cm包文件移到此目录下 创建repodata

 $[{\tt root@cdh85-29\ cm6}]\#\ createrepo$  .

创建.repo

systemctl enable|disable httpd.service #开机启动与否 systemctl {start|stop|restart|status} httpd.service #单次操作状态

• <u>Step 1: Configure a Repository</u>

vi /etc/yum.repos.d/cloudera-manager.repo
[cloudera-manager]
name = Cloudera Manager, Version
baseur1 = http://172.20.85.29/cloudera-repos/cm6/
gpgcheck = 0

<u>Step 2: Install JDK</u>

<u>Step 3: Install Cloudera Manager Server</u>

sudo yum install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server

配置外置数据库 在cdh85-29上传mysql驱动包到/usr/share/java/目录,并做软连接

创建/usr/share/java目录,将mysql-jdbc包放过去 mkdir -p /usr/share/java cd /usr/share/java && ln -s /usr/share/java/mysql-connector-java-5.1.44.jar mysql-connectorjava.jar

指定CM外部数据源:

/opt/cloudera/cm/schema/scm\_prepare\_database.sh mysql scm scm baofoo@64

#### Step 4: Install Databases

## Installing the MySQL Server

#### Note:

•

- If you already have a MySQL database set up, you can skip to the section Configuring and Starting the MySQL Server to verify that your MySQL configurations meet the requirements for Cloudera Manager.
- For MySQL 5.6 and 5.7, you must install the MySQL-shared-compat or MySQL-shared package. This is
  required for the Cloudera Manager Agent package installation.
- It is important that the datadir directory, which, by default, is /var/lib/mysql, is on a partition that has sufficient free space.

Cloudera Manager installation fails if GTID-based replication is enabled in MySQL.

. . . . . . . . . . . .

#### 注意 开启GTID会安装失败

wget <u>http://repo.mvsql.com/mvsql-community-release-el7-5.noarch.rpm</u> sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm sudo yum update sudo yum install mysql-server sudo systemctl start mysqld

• <u>Step 5: Set up the Cloudera Manager D</u>	atabase	
sudo mkdir -p /usr/share/java/		
chmod 755 /usr/share/java/	# 巨坑!	没改权限连接不上mysq1

cd mysql-connector-java-5.1.46 sudo cp mysql-connector-java-5.1.46-bin.jar /usr/share/java/mysql-connector-java.jar

• Step 6: Install CDH and Other Software

# 3.3 登录管理界面

等待cm元数据库初始化完成,大概需要几分钟,随后浏览器访问: 192.168.85.59:7180,默认帐号admin,默认密码 admin

Hive集群 172.20.85.[10-44]

# Hbase集群 172.20.85.[45-59]

# 

勾选主机,点击继续

# http://172.20.85.29/cloudera-repos/cm6/



点击继续

Cloudera Manag	er				
群集安装					
		▲ 王彤兴裕	IP 地址	連織	秋命
	✓ #32	cdh05-22	172.20.05.22		✓ 已成功完成
	C. Setup Autor II S	cdh05-23	172.20.05.23		✓ 已成功完成
	Geographics	cd105-24	172.20.85.24		✔ 已成功完成
	Specify Hosts	cd105-25	172.20.85.25		✔ 已成功完成
		cd185-26	172.20.85.26		✔ 已成功完成
	● 选择存储库	cd185-27	172.20.85.27		✔ 已成功完成
	JDK 安装送版	cdh85-28	172.20.85.28		✔ 已成功完成
		cd185-29	172.20.85.29		✔ 已成功完成
	✓ 提供 SSH 整景凭 捐。	cdh85-30	172.20.85.30		✓ 已成功完成

← → C ▲ 不安全   ≥	177://172.20.85.50:7183/cmf/expreza	wizard/wizard#step=parcelInstallStep		
Cloudera Manag	ger			
群集安装				
		* 正在安装选定 Parcel 选出的 Parcel 正在下载并安装在群集的所有当	=01_E.	
	Setup Auto-TLS	✓ CDH 6.0.0-1.edh6.0.0.p0.537114	日下断: 0%	83號 0/0
	Specify Hosts			
	✓ 选择存储库			
	🕑 ЈОК 安朝(田道			
	受供55H 登录兆 索。			

# 离线安装CDH

# 3.1 上传文件

在cdh81-59上,将CHD5相关的Parcel包放到主节点的/opt/cloudera/parcel-repo/

相关文件如下:

CDH-6.0.0-1.cdh6.0.0.p0.537114-e17.parcel

CDH-6.0.0-1.cdh6.0.0.p0.537114-e17.parcel.sha256

manifest.json

下载地址: https://archive.cloudera.com/cdh6/6.0.0/parcels/

最后将CDH-6.0.0-1.cdh6.0.0.p0.537114-e17.parcel.sha256

, 重命名为CDH-6.0.0-1.cdh6.0.0.p0.537114-e17.parcel.sha

,这点必须注意,否则,系统会重新下载

文件。

	95742
Specify Hosts	他推翻在例前 15 个主机上运行。
	◇ 个强主机正确地解析了自己的主机需称。
	查查符存在中央的初始線中35米划版物源。
) (239479960	◎ 检查 /etc./horts 时未返现错误。
	※前生机均將 locahost 編行为 127.0.0 1。
JOK STREET	经重应的所有主机均正确且及时均解析了偏远的主机杂制。
	主机8种几乎同步(10分钟内)。
	◇ 整个教练中的主机时区一般。
使供SSH叠使用	无用户1008缺失。
遇.	◎ 软件包和 parcal 之吗未检测别中央。
	沒有存在已知情識的內緒該本在运行。
Install Agents	※前主机上的./proc/ego//m/swappinecs 都未送现问题。
Theory Wilson Proved	● 目前用于時人工調田信,可能会得放重人性語可認,適応行"ethe news > /nys/kernel/mm/transparent_trageoge/defing KT/scho news > /nys/kernel/mm/transparent_trageoge/ 用於記述,然所同一中心能加引./ntcn local 等到的代謝中中,以提在系统重要的扩充记录量,以下当时尽受到影响 3 考虑于######
) IEEEsowatta Paros	<ul> <li>P器F COM d Hua Python 版本收换发展。</li> </ul>
Inspect Hosts	・ Starting with CDH & Pestgreid CL-backed Hue requires the Pigropp1 version to be at least 2.5.4, see the documentation for more information. This warring can be ignored if hosts v CDH 6, or will not run Hue with PostgreidQL. The following hosts have an incompatible Pigropp2 version of 2.5.1: <
	O hosts are running COH 6 and 16 hosts are running CDH 6.
	每个教编中检查过的所有主机均在运行相同版本的相伴。
	<ul> <li>⑤ 所有托置的主机都拥有不一致的Java版本。</li> </ul>

RHEL 7 Compatible 1 Install the python-pip package: sudo yum install python-pip

2 Install psycopg2 2.7.5 using pip:

sudo pip install psycopg2==2.7.5 ---ignore-installed

# 点击继续,等待安装,安装完成如下图:

### 点击继续

验证	
۲	检查器在所有 22 个主机上运行。
۲	个别主机正确地解析了自己的主机名称。
۲	查询存在冲突的初始脚本时未发现错误。
۲	检查 /etc/hosts 时未发现错误。
۲	所有主机均将 localhost 解析为 127.0.0.1。
۲	检查过的所有主机均正确且及时地解析了彼此的主机名称。
۲	主机时钟几乎同步(10分钟内)。
۲	整个群集中的主机时区一致。
۲	无用户或组缺失。
۲	软件包和 parcal 之间末检测到冲突。
۲	没有存在已知错误的内核版本在运行。
۲	所有主机上的 /proc/sys/vm/swappiness 都未发现问题。
۲	没有任何性能与"透明大页面"设置有关。
۲	已满足 CDH 5 Hue Python 版本依赖关系。
۲	Hue Psycopg2 version for PostgreSQL is satisfied for both CDH 5 and CDH 6.
۲	0 hosts are running CDH 5 and 22 hosts are running CDH 6.
۲	每个群集中检查过的所有主机均在运行相同版本的组件。
۲	所有托管的主机都拥有不一致的 Java 版本。
۲	所检查的所有 Cloudera Management Daemon 版本与跟务器一致。
۲	所检查的所有 Cloudera 管理代理版本与服务器一致。

根据提示修改,点击完成

我们这里选含Impala的内核安装,点击继续 3.4 配置服务角色

创建hive、oozie、sentry、monitor、hue库并授权: baofoo\_hive、baofoo\_oozie、baofoo\_sentry、 baofoo\_monitor、baofoo\_hue

拷贝mysql驱动到hive jar包(cdh85-29)

cd /opt/cloudera/parcels/CDH/lib/hive/lib && ln -s /usr/share/java/mysql-connector-java-5.1.44-bin.jar mysql-connector-java.jar

拷贝mysql驱动到oozie jar包(cdh85-29) cd /opt/cloudera/parcels/CDH/lib/oozie/libtools && ln -s /usr/share/java/mysqlconnector-java-5.1.44-bin.jar mysql-connector-java.jar

cd /opt/cloudera/parcels/CDH/lib/oozie/libserver && ln -s /usr/share/java/mysqlconnector-java-5.1.44-bin.jar mysql-connector-java.jar

FS 現大小	Cluster 1 > HDFS(服卵馆图)	
block size, dfs.blocksize	128 兆字节 •	
定的 DataNode 失败的密	Cluster 1 > DataNode Default Group 🐂	
datanode.failed.volumes.tolerated	3	
aNode 数据目录	Cluster 1 > DataNode Default Group 🕤	
deta.dit, dfa.detanoole.deta.dir	/dfa/data1/dfa/dn	⊜⊕
	/dfs/data2/dfs/dn	⊝⊕
	/dfs/data3/dfs/dn	⊕⊞
	/dfs/data4/dfs/dn	⊕⊕
	/dfs/data5/dfs/dn	⊜⊕
	/dfs/data6/dfs/dn	ÐÆ
	/dts/oatab/dts/on	<b>BH</b>
	/0Tty/datab/dTb/0n	88
NameNade 政策目录 ds.name.di, dfs.namenade.name.dir	Zata-Zata-Zata-Zata-Zata-Zata-Zata-Zata	88
NameNode 政府目录 dfs.name.dir, dfs.namenode.name.dir	zatszoiataozatszan Cluster 1 > NameNode Default Group ♠ Zopt/dfs/nn	68
NameNode 股原目录 dts.nerre.dt; dfs.namende.nerre.dtr HDFS 检查点目录	Zatszolatacization Cluster 1 > NameNode Default Group • ZogUdfa/nn Cluster 1 > SecondaryNameNode Default Group •	68
NamaNieda 取服目現 dls.name.dl; dls.namende.name.dlr HDF5 检查点目現 fss.namende.deselparit.dl;	Zatszaszarzan Cluster 1 - NameNode Default Group - Zogt/dfa/nn Cluster 1 - SecondaryNameNode Default Group - Zogt/dfa/ann	
NameNode 政策目录 dls.name.dl; dls.namenode.name.dlr HDF5 信告点目录 fs.checipose.dl; dls.namenode.checiposet.dlr HHve 合序目录	/dts/data//dts/dn Cluster 1 > NameNode Default Group ● /opt/dfs/nn Cluster 1 > SacondaryNameNode Default Group ● /opt/dfs/ann Cluster 1 > Hive (副的范围)	
NameNode 政策目現 dh.name.dk, dh.namenode.name.dk HDFS 役会立目現 fa.cheolopart.dk, dh.namenode.cheolopaint.dk HNve 在本日登 Nve metastore.warthouse.dk	//dts/datad/dts/dts/dts/ Cluster 1 > NameNode Default Group ● /opt/dfs/tm Cluster 1 > SecondaryNameNode Default Group ● /opt/dfs/ann Cluster 1 > Hwe ( 起外范囲 ) //user/hive/warehouse	
NameNade 政策目录 dls.name.dls, dls.namende.name.dlr HDFS 检查查目录 fsc.theologisted.dl dls.namenade.cheologisted.dl HHme 仓库目录 Hive metastates watchouse.dlr HHme Matastates 服务器端口	Zatszolatab/atta/ Cluster 1 > NameNode Default Group ● Zogt/dfs/nn Cluster 1 > SacondaryNameNode Default Group ● Zogt/dfs/ann Cluster 1 > Hive (副的范围) Zuster 1 > Hive (副的范围) Zuster 1 > Hive (Mathatore Server Default Group	
NameNode 股份目現 dis sene di, dis namenode some dir HDFS 检查点目現 fas seenedae sheelgasit.dir Hive chel日発 Hive metastore Northouse.dir Hive Metastore 股份股限口 Nive metastore port	Zatsvastab/ats/a Cluster 1 > NameNode Default Group Zogt/dfs/nn Cluster 1 > Have ( 题外范面 ) Zuster 1 > Have ( 题外范面 ) Zuster 1 > Have ( 题外范面 ) Zuster 1 > Have Metaatore Server Default Group 0063	
NameNicole IRVIER dissene dis disseneroide nome dir HDFS Régistration fascheroport dis disseneroide scheroport dir Hive metastore IRVF/RIGO Hive Metastore IRVF/RIGO Nise metastore port Kadu IIB/F	Zatsvastab/atsvan Cluster 1 > NameNode Default Group  Zopt/dfs/nn Cluster 1 > Have ( 副時元間 ) Zupt/dfs/ann Cluster 1 > Have ( 副時元間 ) Cluster 1 > Have ( 副時元間 ) Cluster 1 > Have Metastore Server Default Group 0083 Cluster 1 > Impale ( 副時元間 ) Cluster 1 > Impale ( 副時元間 )	

### 点击继续

mpala Daemon 暂存目录	Cluster 1 > Impala Daemon Default Group和其他2个 🐂	
oratch_dirs 编辑单个值	/dfa/data1/impala/impalad	
	/dfs/data2/impala/impalad	⊝⊕
	/dfs/data3/impala/impalad	8€
	/dfa/data4/impala/impalad	⊖⊕
	/dfs/data5/impala/impalad	
	/dfs/data6/impala/impalad	

NodeManager 本地目录	Cluster 1 > NodeManager Default Group和助化 2 个 🕤	
yarn.nodemanager.local-dirs 编辑单个值	/dfs/data1/yarn/nm	
	/dfs/data2/yarn/nm	
	/dfs/data3/yarn/nm	⊖⊕
	/dfs/data4/yarn/nm	⊟⊕
	/dfs/dats5/yarn/nm	⊟⊞
	/dfs/data6/yarn/nm	⊟⊕

# 点击继续,等待启动。

Show All Steps      Show Only Failed Steps     Show Only Running Steps		
C Ensuring that the expected software releases are installed on hosts. 已成功把成 1 个步骤。	12月 6, 3:54:44 下午	43ma
> ● 正在部署集户講館選 C Cluster 1 C Successfully deployed all client configurations.	12月 6, 3:54:44 下午	16.05s
》 意識 Cloudera Management Service, ZooKeeper 已成功的成 2 个步骤。	12月 6, 3:55:00 下午	24.278
> ● 启动HDF3 已成功完成1 个步撰。	12月 6, 3.55:24 下午	39.169
> ● 应应 YARN (MR2 Included) 已成功完成1 个步骤。	12月 6, 3:56:03 下午	25.54s
> ② 自动 Hive 日成功的成1 个步展。	12月 6, 3 56:29 下午	39.85a
▶ ● 启动 impela, Ocole 已成功完成 2 个步骤。	12月 6, 3:57:09 下午	36.83s
> の 訳記 Hue 已成況的成1 个步骤。	12月 6, 357.46 下午	22.64s
		_
		继续

点击继续即可完成。

四、Kerberos安装 4.1 Kerberos服务端安装 在cdh85-59上安装: yum install -y krb5-server krb5-libs krb5-workstation

# 修改/etc/krb5.conf

[logging] default = FILE:/var/log/krb5libs.log kdc = FILE:/var/log/krb5kdc.log admin\_server = FILE:/var/log/kadmind.log

```
[libdefaults]
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
default_realm = master
renewable = true
```

```
[realms]
```

```
master = {
   kdc = cdh85-59
   admin_server = cdh85-59
}
```

```
[domain_realm]
```

```
.master = master
master = master
```

# 修改/var/kerberos/krb5kdc/kdc.conf

[kdcdefaults]
kdc\_ports = 88
kdc tcp ports = 88

```
[realms]
master = {
    #master_key_type = aes256-cts
    acl_file = /var/kerberos/krb5kdc/kadm5.acl
    dict_file = /usr/share/dict/words
```

 $max\_renewable\_life = 30d$ 

 $max_life = 30d$ 

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
```

}

# 修改/var/kerberos/krb5kdc/kadm5.acl

\*/admin@master \*

以上三个文件配置完毕后,只需拷贝krb5.conf到集群中其他机器上即可。 scp /etc/krb5.conf cdh81-30:/etc/

#### 4.2 Kerberos客户端安装

在其他几点上安装: yum install -y krb5-libs krb5-workstation 或者在cdh85-29上调用脚本: sh /app/shell/exe\_command\_on\_all\_nodes\_1.sh "scp /app/krb5-\*" "yum -y localinstall krb5\*.rpm && rm -rf ~/krb5\*.rpm"

#### 4.3 关于AES-256加密

oracle官网下载jce\_policy-8.zip, 解压, 将local\_policy.jar、US\_export\_policy.jar拷贝 到 \$JAVA\_HOME/jre/lib/security目录下

#### 4.4 创建数据库

在 cdh85-29 上运行初始化数据库命令。其中 -r 指定对应 realm,初始密码\*\*\* \$ kdb5\_util create -r master -s

出现 Loading random data 的时候另开个终端执行点消耗CPU的命令如 cat /dev/sda > /dev/urandom 可以 加快随机数采集。该命令会在 /var/kerberos/krb5kdc/ 目录下创建 principal 数据库。 如果遇到数据库已经存在的提示,可以把 /var/kerberos/krb5kdc/ 目录下的 principal 的相关文件都删除掉。 默认的数据库名字都是 principal。可以使用 -d 指定数据库名字。

#### 4.5 启动服务

在cdh85-59上执行如下命令: systemctl enable krb5kdc.service systemctl enable kadmin.service systemctl start krb5kdc systemctl start kadmin

#### 4.6 创建Kerveros管理员

关于 kerberos 的管理,可以使用 kadmin.local 或 kadmin,至于使用哪个,取决于账户和访问权限: 如果有访问 kdc 服务器的 root 权限,但是没有 kerberos admin 账户,使用 kadmin.local 如果没有访问 kdc 服务器的 root 权限,但是用 kerberos admin 账户,使用 kadmin 在 cdh85-29 上创建远程管理的管理员: 在KDC server主机上,创建一个名为『hadoop』的principal,并将其密码设为『\*\*\*』。执行命令: [root@cdh85-29 /]# kadmin.local Authenticating as principal root/admin@master with password. kadmin.local: addprinc -pw \*\*\* hadoop/admin@master

通过执行kadmin.local中的listprincs命令可以看到创建了一个名为『hadoop/admin@master』的principal: kadmin.local: listprincs K/M@master hadoop/admin@master kadmin/admin@master kadmin/cdh85-29@master kadmin/changepw@master kiprop/cdh85-29@master principal的名字的第二部分是admin,那么该principal就拥administrative privileges 这个账号将会被CDH用来生成其他用户/服务的principal

#### 4.7 CDH启用Kerberos

在CM的界面上点击启用Kerberos, 启用的时候需要确认几个事情:

1.KDC已经安装好并且正在运行

2.将KDC配置为允许renewable tickets with non-zerolifetime

- 在之前修改kdc.conf文件的时候已经添加了kdc\_tcp\_ports、max\_life和max\_renewable\_life这个三个选项

3.在Cloudera Manager Server上安装open1dap-clients

PS:为了使Kerberos能够绑定到OpenLDAP服务器,需要创建一个管理员用户和一个principal,并生成keytab文件,所以这里先安装openldap-clients

4.为Cloudera Manager创建一个principal,使其能够有权限在KDC中创建其他的principals,就是上面创建的 Kerberos管理员账号.

确定完了之后点击continue,进入下一页进行配置,要注意的是:这里的『Kerberos Encryption Types』必须跟 KDC实际支持的加密类型匹配(即kdc.conf中的值)

这里使用了默认的aes256-cts

新定有关 KDC 的氛围、Cloudera Manag	per 使用下面的漏性生成在群集中运行的 CDH 守护理序的主体。		
KDC 类型	MIT KDC		0
	Active Directory		
Kerberos 安全領域	master	e	0
default_realm			
KDC Server 主机	odh81-50	e	0
KDC Admin Server Host			0
and the part of the			
Domain Name(s)	Ð		0
			0
Kerberos 加密类型	aes256 cts	ΘĦ	
	0		
Kerberos Principal 最大可更新生命期	5 天 •		0
返回			-
			(in) =
启用 Kerberos 用于 CDH-P	PRO-NEW		
KRB5 配置			
编织为群集牛成 krb5.conf 所需的漏件。同	T以使用"空全概"本段指示意识 KDC 设置的配置:乐约,使用路径加升份验证。		
Wet Clouders Manager 1978			0
krb5.conf			

注意,如果勾选了这个选项就可以通过CM的管理界面来部署krb5.conf,但是实际操作过程中发现有些配置仍然需要手动修改该文件并同步

启用 Kerberos 用于 CDH-PRO-NEW

KDC Account Manager 凭据

输入有权限创建其他用户的帐	户的凭据。Cloudera M	anager #	乳以加密形式存储该	先据并在需要生成新主体	时使用它。	
用户名	hadoop/admin	0	master			
惠码	******					

点击continue,进入下一页,输入Cloudera Manager Principal的管理员账号和密码,注意输入账号的时候要使用@前要使用全称,hadoop/admin

启用 Kerberos	用于	CDH-PRO	-NEW
-------------	----	---------	------

导入 KDC Account Manager 凭据 命令

状态 📀 已完成 🛗 12月 7, 1:30:54 下午 🛛 5.02s

Successfully imported KDC Account Manager credentials.

## 点击continue,进入下一页,导入KDC Account Manager Credentials

指定群集中每个服务使用的	Kerberos Principal,如果包决定使用联认值更改这些 Principal,可能需要执行其他步骤。请先阅读关于	自定义 Principal 的文档,然后对此页真做出更改。
Kerberos 主体	Flume ( 服务范围 )	0
	flume	
	HBase ( 服务范围 )	
	hbase	
	HDFS ( 服务范围 )	
	hdis	
	Hive ( 嚴惩范围 )	
	hive	
	Hue ( 服务范围 )	
	hua	
	Impala (服务范围)	

# 点击continue,进入下一页, restart cluster并且enable Kerberos

DataNode 收发器编目	1004 DataNode 원 XCeiver 1602的編集, 전관 DataNode 원出 티노동(法)(法)(此),			
DataNode HTTP Web UI (유드)	1006 DataNode HTTP Web UI SDBAEL, 영남 DataNode 양止年礼的秘密立며 HTTP 地址。			
書要重启群集以使更改生物	x.			
2 是,我现在已准备好重	台灣處。			

之后CM会自动重启集群服务,启动之后会会提示Kerberos已启用

这个过程中CM会自动在Kerberos的数据库中创建各个节点中各个账户对应的principle 可以使用 kadmin.local -q "list\_principals"查看,,格式为username/hostname@<u>XIAOHEI.INFO</u>,例如 hdfs/hadoop-10-0-8-124@<u>XIAOHEI.INFO</u>

在CM上启用Kerberos的过程中,CM会自动做以下的事情:

- 1.集群中有多少个节点,每个账户都会生成对应个数的principal
- 2.为每个对应的principal创建keytab
- 3. 部署keytab文件到指定的节点中

4.在每个服务的配置文件中加入有关Kerberos的配置 其中包括Zookeeper服务所需要的jaas.conf和keytab文件都会自动设定并读取,如果用户仍然手动修改了 Zookeeper的服务,要确保这两个文件的路径和内容正确性 keytab是包含principals和加密principal key的文件 keytab文件对于每个host是唯一的,因为key中包含hostname keytab文件用于不需要人工交互和保存纯文本密码,实现到kerberos上验证一个主机上的principal 启用之后访问集群的所有资源都需要使用相应的账号来访问,否则会无法通过Kerberos的authenticatin

## 4.8 创建HDFS超级用户

我们使用yarn作为hadoop集群的超级用户,在集群所有节点上创建supergroup组并加入yarn用户。 此时直接用CM生成的principal访问HDFS会失败,因为那些自动生成的principal的密码是随机的,用户并不知道, 而通过命令行的方式访问HDFS需要先使用kinit来登录并获得ticket 用户可以通过创建一个yarn@master的principal并记住密码从命令行中访问HDFS # 需要输入两遍密码 kadmin.local -q "addprinc yarn" 先使用 kinit <u>varn@master</u> 登录之后就可以通过认证并访问HDFS

# 查看principals

\$ kadmin: list\_principals

# 添加一个新的 principal

kadmin: addprinc user1

WARNING: no policy specified for user1@JAVACHEN.COM; defaulting to no policy Enter password for principal "user1@JAVACHEN.COM": Re-enter password for principal "user1@JAVACHEN.COM": Principal "user1@JAVACHEN.COM" created.

# 删除 principal

kadmin: delprinc user1

Are you sure you want to delete the principal "user1@JAVACHEN.COM"? (yes/no): yes Principal "user1@JAVACHEN.COM" deleted. Make sure that you have removed this principal from all ACLs before reusing.

kadmin: exit

# 4.9 确认HDFS可以正常使用

登录到某一个节点后,用kinit来获取yarn用户的credentials 现在用'hadoop fs -ls /'应该能正常输出结果 用kdestroy销毁credentials后,再使用hadoop dfs -ls /会发现报错

获取了yarn的证书后,提交一个PI程序,如果能正常提交并成功运行,则说明Kerberized Hadoop cluster在正常 工作 hadoop jar /opt/cloudera/parcels/CDH/jars/hadoop-examples.jar pi 10 1000

```
beeline连接hive、impala
```

```
hive: beeline -u 'jdbc:hive2://cdh85-29:10000/;principal=hive/cdh85-29@master'
impala: beeline -u 'jdbc:hive2://cdh81-51:21050/;principal=impala/cdh81-51@master'
```

五、LDAP安装 线上安装 https://blog.csdn.net/fanren224/article/details/79707206

Connection	n proj	perties	×
Connection	name:	local-cdh6-idap	
General	Option	as <u>Attributes</u>	
Connec	tion:		
Host:	10.	0.221.73 Port: 389 Version: 3 ~	
Base:	dc-	dc-baldu,dc-com Fetch DNs	
	© :	imple authentication SSL 11.5 SSS-API SASL	
Account	t		
Usernar	ne:	cn=admin,dc=baidu,dc=com	
Passwo	rd:	•••••	
Anor	nymou	s connection	
_			

yum install openldap openldap-servers openldap-clients compat-openldap

cp /usr/share/openldap-servers/DB\_CONFIG.example /var/lib/ldap/DB\_CONFIG  $\ensuremath{\mathsf{CONFIG}}$ 

```
chown ldap. /var/lib/ldap/DB_CONFIG
```

systemctl enable slapd

systemctl start slapd

netstat -tunlp |grep slapd

Slappasswd

New password: Re-enter new password: {SSHA}Ltmskub54M7W30yGI5Z91+G00DtvUGKe

vim chrootpw.ldif

dn: olcDatabase={0} config, cn=config
changetype: modify
add: olcRootPW
olcRootPW: {SSHA}Ltmskub54M7W30yGI5Z91+G00DtvUGKe

ldapadd -Y EXTERNAL -H ldapi:/// -f chrootpw.ldif

我们需要向 LDAP 中导入一些基本的 Schema。这些 Schema 文件位于 /etc/open1dap/schema/ 目录中, schema控制着条目拥有哪些对象类和属性

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif

配置 LDAP 的根域(以 dc=baofoo, dc=com 为例)及其管理域:

vim chdomain.ldif

```
dn: olcDatabase={1}monitor, cn=config
changetype: modify
replace: olcAccess
olcAccess: {0} to * by dn.base="gidNumber=0+uidNumber=0, cn=peercred, cn=external, cn=auth"
    read by dn.base="cn=admin, dc=baofoo, dc=com" read by * none
```

dn: olcDatabase={2}hdb, cn=config
changetype: modify
replace: olcSuffix

```
olcSuffix: dc=baofoo, dc=com
dn: olcDatabase={2}hdb, cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=admin, dc=baofoo, dc=com
dn: olcDatabase={2}hdb, cn=config
changetype: modify
add: olcRootPW
olcRootPW: {SSHA} Ltmskub54M7W30yGI5Z91+G00DtvUGKe
dn: olcDatabase={2}hdb, cn=config
changetype: modify
add: olcAccess
olcAccess: {0} to attrs=userPassword, shadowLastChange by
    dn="cn=admin, dc=baofoo, dc=com" write by anonymous auth by self write by * none
olcAccess: \{1\} to dn. base="" by * read
olcAccess: {2} to * by dn="cn=admin, dc=badiu, dc=com" write by * read
```

ldapmodify -Y EXTERNAL -H ldapi:/// -f chdomain.ldif

在上述基础上,我们来创建一个叫做 baofoo company 的组织,并在其下创建一个 admin 的组织角色 (该角色内的用户具有管理整个 LDAP 的权限)和 People 和 Group 两个组织单元:

dn: dc=baofoo, dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: baofoo Company
dc: baofoo

dn: cn=admin, dc=baofoo, dc=com

objectClass: organizationalRole
cn: admin

dn: ou=People, dc=baofoo, dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Group, dc=baofoo, dc=com
objectClass: organizationalRole
cn: Group

ldapadd -x -D cn=admin, dc=baofoo, dc=com -W -f basedomain.ldif

vim ldapuser.ldif

dn: uid=tom, ou=People, dc=baofoo, dc=com objectClass: inetOrgPerson objectClass: posixAccount objectClass: shadowAccount uid: tom cn: tom sn: tom userPassword: {SSHA}KzC51n00VkpXxajXetcyYYq5VcHhfAuM uidNumber: 1100 gidNumber: 1100 homeDirectory: /home/tom

dn: cn=SRE, ou=Group, dc=baofoo, dc=com
objectClass: posixGroup
cn: SRE
gidNumber: 1100
memberUid: SRE

# 修改密码:

https://blog.csdn.net/developerinit/article/details/76141065

# 统一账户管理

# 软件安装 yum install -y nss-pam-ldapd openldap-clients

#### 配置openLDAP-client

cp /etc/nsswitch.conf /etc/nsswitch.conf.old sed -i '/^passwd:.\*\$/s//& ldap/g' /etc/nsswitch.conf sed -i '/^shadow:.\*\$/s//& ldap/g' /etc/nsswitch.conf

cp /etc/openldap/ldap.conf /etc/openldap/ldap.conf.old cat >> /etc/openldap/ldap.conf <<EOF BASE dc=baofoo,dc=com URI ldap://172.20.15.13 ssl off EOF

启用LDAP身份验证机制 cp /etc/sysconfig/authconfig /etc/sysconfig/authconfig.old cat > /etc/sysconfig/authconfig <<EOF IPADOMAINJOINED=no USEMKHOMEDIR=no USEPAMACCESS=no CACHECREDENTIALS=yes USESSSDAUTH=no USESHADOW=yes USEWINBIND=no USEDB=no USEFPRINTD=yes FORCESMARTCARD=no PASSWDALGORITHM=sha512 USELDAPAUTH=yes USEPASSWDQC=no IPAV2NONTP=no USELOCAUTHORIZE=yes USECRACKLIB=yes USEIPAV2=no USEWINBINDAUTH=no USESMARTCARD=no USELDAP=yes USENIS=no USEKERBEROS=no USESYSNETAUTH=no USESSSD=no USEHESIOD=no USEMD5=yes FORCELEGACY=no EOF

nslcd 服务 cat >> /etc/nslcd.conf <<EOF uri ldap://172.20.85.29 base dc=baofoo,dc=com EOF

编辑系统认证文件,保证使用LDAP来认证 cp /etc/pam.d/system-auth /etc/pam.d/system-auth.old cat > /etc/pam.d/system-auth <<EOF #%PAM-1.0 # This file is auto-generated. # User changes will be destroyed the next time authconfig is run. auth required pam env. so auth sufficient pam fprintd.so auth sufficient pam\_unix.so nullok try\_first\_pass auth requisite pam\_succeed\_if.so uid >= 500 quiet auth sufficient pam ldap.so use first pass auth required pam\_deny.so account required pam\_unix.so account sufficient pam localuser.so account sufficient pam\_succeed\_if.so uid < 500 quiet account [default=bad success=ok user\_unknown=ignore] pam\_ldap.so account required pam permit.so password requisite pam\_cracklib.so try\_first\_pass retry=3 type= password sufficient pam unix.so sha512 shadow nullok try first pass use authtok password sufficient pam\_ldap.so use\_authtok password required pam deny.so session optional pam keyinit.so revoke session required pam limits.so session [success=1 default=ignore] pam\_succeed\_if.so service in crond quiet use\_uid session required pam unix.so session optional pam\_ldap.so session required pam mkhomedir.so skel=/etc/skel/ umask=0022 EOF

在cdh85-29上安装

\$ yum install -y db4 db4-utils db4-devel cyrus-sasl\* krb5-server-ldap 安装ldap服务 #!/bin/bashecho "install ldap rpm" yum install -y openldap-servers openldap-clients cp /usr/share/openldap-servers/DB\_CONFIG.example /var/lib/ldap/DB\_CONFIG chown ldap. /var/lib/ldap/DB\_CONFIG systemctl start slapd

systemctl enable slapd

有两个文件要复制: slapd的配置文件和数据库文件,将openldap-servers自带的example复制到相应目录: PS: centos7 slapd.conf.obsolete 并不存在,所以我从centos6 里拷贝了一个过来

cp /usr/share/openldap-servers/slapd.conf.obsolete /etc/openldap/slapd.conf

cp /usr/share/openldap-servers/DB\_CONFIG.example /var/lib/ldap/DB\_CONFIG

#### 5.2 服务端配置

使用slappasswd创建LDAP管理员密码,这个命令不会直接将密码写入配置,运行slappasswd后输入两次密码,会返回一串密文,复制下这个 密文。 [root@cdh85-29 ~]# slappasswd New password: Re-enter new password: {SSHA}PL+FAWxVd7uLGcqBCbwaq/ET3yqaQx7E

编辑/etc/openldap/slapd.conf, 找到"database bdb", 按照自己的需求更改下面的: suffix "dc=baofoo,dc=com" rootdn "cn=admin,dc=baofoo,dc=com" //管理员为rootpw {SSHA}PL+FAWxVd7uLGcqBCbwaq/ET3yqaQx7E //复制的管理员的密码,也支持明文

<del>添加一些基本配置,并引入\_kerberos\_和\_openldap\_的\_schema:\_</del> <del>\$\_cp\_/usr/share/doc/krb5-server=ldap=1.15.1/kerberos.schema\_/etc/openldap/schema/</del>

在/etc/openldap/slapd.conf加入 include /etc/openldap/schema/kerberos.schema

#### 更改目录权限:

chown -R ldap:ldap /etc/openldap chown -R ldap:ldap /var/lib/ldap

#### 5.3 测试并生成配置文件

rm -rf /etc/openldap/slapd.d/\* //删除原文件
systemctl start slapd //生成bdb文件
slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d //生成配置文件
chown -R ldap:ldap /etc/openldap/slapd.d

#### 5.4 配置完成重启服务

systemctl restart slapd
systemctl enable slapd //设置开机启动
systemctl list-unit-files slapd.service //查看开机启动状态
经过上面的配置后, openldap server就配置好了。

#### 查看状态,验证服务端口:

[root@cdh85-29 open1dap]# ps aux | grep slapd | grep -v grep ldap 58094 0.0 0.0 533648 9748 ? Ssl 10:11 0:00 /usr/sbin/slapd -u ldap -h ldapi:/// ldap:/// [root@cdh85-29 open1dap]# ss -tun1p | grep :389

```
tcp LISTEN 0 128
                                                          *:*
                                 *:389
                                                                                users:
(("slapd",pid=58094,fd=8))
      LISTEN
                0
                        128
                                 :::389
                                                         :::*
                                                                                users:
tcp
(("slapd",pid=58094,fd=9))
查看LDAP数据库结构:
[root@cdh85-29 openldap]# ldapsearch -x -H ldap://127.0.0.1 -b 'dc=baofoo,dc=com'
# extended LDIF
#
# LDAPv3
# base <dc=baofoo,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
# search result
search: 2
result: 32 No such object
# numResponses: 1
```

### 5.5 Kerberos和Ldap集成

为了使Kerberos能够绑定到OpenLDAP服务器,需要创建一个管理员用户和一个principal,并生成keytab文件 设置该文件的权限为LDAP服务运行用户可读(一般为ldap): kadmin.local -q "addprinc -randkey ldap/cdh85-29@master"

kadmin.local -q "ktadd -k /etc/openldap/ldap.keytab ldap/cdh85-29@master"

chown ldap:ldap /etc/openldap/ldap.keytab && chmod 640 /etc/openldap/ldap.keytab

```
确保LDAP启动时使用上一步中创建的keytab文件,在/etc/sysconfig/ldap增加KRB5_KTNAME配置:
PS: centos7 /etc/sysconfig/ldap 并不存在,所以我从centos6 里拷贝了一个过来
export KRB5_KTNAME=/etc/openldap/ldap.keytab
```

重启 systemctl restart slapd

### 5.6 配置并迁移系统用户

配置好的LDAP数据库是空的,需要将系统上的用户导入到LDAP数据库中。需要用migrationtools将系统用户转换为 LDAP能识别的ldif文件。

安装migrationtools: yum install -y migrationtools

配置migrationtools: 编辑/usr/share/migrationtools/migrate\_common.ph ,按需更改下面两行: \$DEFAULT\_MAIL\_DOMAIN = "baofoo.com"; \$DEFAULT BASE = "dc=baofoo,dc=com";

### 生成模板文件:

/usr/share/migrationtools/migrate\_base.pl > /opt/base.ldif 生成ldif文件: /usr/share/migrationtools/migrate\_passwd.pl /etc/passwd >/opt/passwd.ldif /usr/share/migrationtools/migrate\_group.pl /etc/group >/opt/group.ldif

# 将生成的ldif导入到LDAP数据库:

ldapadd -x -D "cn=admin,dc=baofoo,dc=com" -W -f /opt/base.ldif ldapadd -x -D "cn=admin,dc=baofoo,dc=com" -W -f /opt/passwd.ldif ldapadd -x -D "cn=admin,dc=baofoo,dc=com" -W -f /opt/group.ldif

# 5.7 LDAP客户端配置

在其他节点上运行 yum install openldap-clients -y 或者在cdh85-29上调用脚本: sh /app/shell/exe\_command\_on\_all\_nodes\_1.sh "scp /app/ openldap-clients-2.4.44-5.el7.x86\_64.rpm" "yum -y localinstall openldap-clients-2.4.44-5.el7.x86\_64.rpm && rm rf ~/ openldap-clients-2.4.44-5.el7.x86\_64.rpm"

修改 /etc/openldap/ldap.conf 以下两个配置 BASE dc=baofoo,dc=com URI ldap://cdh85-29

然后,运行下面命令测试: #先删除 ticket \$ kdestroy [root@cdh85-29 shell]# ldapsearch -b 'dc=baofoo,dc=com' SASL/GSS-SPNEGO authentication started ldap\_sasl\_interactive\_bind\_s: Local error (-2) additional info: SASL(-1): generic failure: GSSAPI Error: Unspecified GSS failure. Minor code may provide more information (SPNEGO cannot find mechanisms to negotiate)

重新获取 ticket: [root@cdh85-29 shell]# kinit yarn Password for yarn@master: [root@cdh85-29 shell]# klist Ticket cache: FILE:/tmp/krb5cc\_0 Default principal: yarn@master

Valid starting Expires Service principal 12/08/2017 13:33:58 12/09/2017 13:33:58 krbtgt/master@master renew until 12/15/2017 13:33:58

```
$ ldapsearch -x -b 'dc=baofoo,dc=com'
#没有报错
```

# numEntries: 128

# ldap导出:

ldapsearch -x -b 'dc=baofoo, dc=com' > ldapbackup.ldif

ldap导入: ldapadd -x -D cn=admin,dc=baofoo,dc=com -W -f ldapbackup.ldif

#### 5.8 配置HIVE集成LDAP

在hive-site.xml中加入以下配置: <property> <name>hive.server2.authentication</name> <value>LDAP</value> </property> <property> <name>hive.server2.authentication.ldap.url</name> <value>ldap://cdh85-29</value> </property> <property> <name>hive.server2.authentication.ldap.baseDN</name> <value>ou=people,dc=baofoo,dc=com</value> </property> 重启Hive和Yarn服务,进入beeline测试: LDAP认证: [root@cdh85-29 ~] # beeline Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support was removed in 8.0 Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support was removed in 8.0 Beeline version 1.1.0-cdh5.12.1 by Apache Hive beeline> !connect jdbc:hive2://cdh85-29:10000/default scan complete in 1ms

Connecting to jdbc:hive2://cdh85-29:10000/default Enter username for jdbc:hive2://cdh85-29:10000/default: yarn Enter password for jdbc:hive2://cdh85-29:10000/default: \*\*\*\*\*\*\*\*\*\*\* Connected to: Apache Hive (version 1.1.0-cdh5.12.1) Driver: Hive JDBC (version 1.1.0-cdh5.12.1) Transaction isolation: TRANSACTION\_REPEATABLE\_READ 0: jdbc:hive2://cdh85-29:10000/default>

### Kerberos认证:

[root@cdh85-29 ~]# kinit yarn
Password for yarn@master:
[root@cdh85-29 ~]# klist
Ticket cache: FILE:/tmp/krb5cc\_0
Default principal: yarn@master

Valid starting Service principal Expires 12/08/2017 15:18:53 12/09/2017 15:18:53 krbtgt/master@master renew until 12/15/2017 15:18:53 [root@cdh85-29 ~]# beeline -u 'jdbc:hive2://cdh85-29:10000/;principal=hive/cdh85-29@master' Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support was removed in 8.0 Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support was removed in 8.0 scan complete in 2ms Connecting to jdbc:hive2://cdh85-29:10000/;principal=hive/cdh85-29@master Connected to: Apache Hive (version 1.1.0-cdh5.12.1) Driver: Hive JDBC (version 1.1.0-cdh5.12.1) Transaction isolation: TRANSACTION REPEATABLE READ Beeline version 1.1.0-cdh5.12.1 by Apache Hive 0: jdbc:hive2://cdh85-29:10000/>

# 5.9 配置IMPALA集成LDAP

Impala中可以同时使用Kerberos+LDAP的认证方式,所以在已经启用Kerberos的情况下启用LDAP可以正常工作在Impala配置页中:

- 启用 LDAP 身份验证选项设置为true
- 启用 LDAP TLS 选项设置为true
- LDAP URL 设置为1dap://cdh85-29
- LDAP BaseDN 设置为ou=people,dc=baofoo,dc=com

重启Impala服务

在chd81-51上执行,使用impala-shell测试LDAP账号: [root@cdh81-51 ~]# impala-shell -l -u yarn --auth\_creds\_ok\_in\_clear Starting Impala Shell using LDAP-based authentication LDAP password for yarn: Connected to cdh81-51:21000 Server version: impalad version 2.9.0-cdh5.12.1 RELEASE (build 5131a031f4aa38c1e50c430373c55ca53e0517b9)

ALL PASSWORDS WILL BE SENT IN THE CLEAR TO IMPALA.
[cdh81-51:21000] >

使用beeline测试LDAP账号: beeline -u "jdbc:hive2://cdh81-51:21050/default;" -n yarn -p 密码

# 5.10 配置HUE集成LDAP

在Hue中配置LDAP可以让Hue直接使用LDAP所管理的账号而不必在Hue中重新管理 在Hue的配置页面中修改

- 身份验证后端/backend为desktop.auth.backend.LdapBackend
- 登录时创建 LDAP 用户/create\_users\_on\_login 设置为True
- 修改ldap\_url=ldap://cdh85-29, ldap\_username\_pattern=uid=<username>, ou=people, dc=baofoo, dc=com
- 使用搜索绑定身份验证/search\_bind\_authentication 设置为False

#### 滚动重启步骤.png

STOW ALL	Steps O Show Only Failed Steps	O Show Running Steps			
」 运行課 正在明	<b>动重信</b> 待克威多令 (Reload (15427))			11月 23, 1:45:58 下午	
~0	重迫工作者角色之前执行所需的操作。 已成功完成命令(Toggle Balancer (1528	7))		11月 23, 1:45:58 下午	18.58
	> 📀 将 Balancer 从 Master (cdh85-59) 状态关	<i>त</i> .	Master (cdh85-59) 28	11月 23, 1:45:58 下午	18.58
•0	步骤数: 13 已完成: 12 成功: 12 (承动重启 ed 正在時待完成命令 (Reload (15427))	h85-58 上的工作者角色)		11月 23, 1:46:16 下午	
	◎ 滾動動廠 edn85-45 上的工作者角色	◎ 滾動重向 cdh85-46 上的工作者角色	◎ 滚动重启 edh85-47 上的	工作者角色	
	◎ 涼动動画 cdh85-48 上的工作者角色	◎ 涼动重启 cdh85-50 上的工作者角色	◎ 滾动胍自 cdh85-51 上的	工作者角色	
	◎ 滾動豐高 edh85-52 上的工作者角色	◎ 涼动重启 cdh85-53 上的工作者角色	◎ 深动重向 cdh85-54 上的	工作者角色	
	◎ 滾动動會 edh85-55 上的工作者角色	◎ 涼动重肩 odh85-56 上的工作者角色	◎ 滾动重肩 cdh85-57 上的	工作者角色	
	○ 滚动服白 odh85-58 上的工作者角色				
<b>v</b> 0	重点工作者角色之后执行清理操作。				