# 基于kylin4的浪潮云HD部署

## 第一章 硬件信息

操作系统: 银河kylin4.0.2

芯片：ARM64

## 第二章 部署安装

### 2.1 准备环境

1. 安装源： <http://10.221.129.22/kylin-arm64/>
2. 资源列表：用户密码均为inspur/inspur123 (sudo -i 以使用管理员权限)

| **序号** | **服务器mgtip（外部访问）** | **data数据网ip（内部访问）** | **主机名** |
| --- | --- | --- | --- |
| 1 | 10.110.56.220 | 172.16.15.220 | manager.bigdata.com |
| 2 | 10.110.56.221 | 172.16.15.221 | master1.bigdata.com |
| 3 | 10.110.56.222 | 172.16.15.222 | master2.bigdata.com |
| 4 | 10.110.56.223 | 172.16.15.223 | worker1.bigdata.com |
| 5 | 10.110.56.224 | 172.16.15.224 | worker2.bigdata.com |
| 6 | 10.110.56.225 | 172.16.15.225 | worker3.bigdata.com |

#### 2.1.1 修改主机名 (集群所有节点均需执行此操作)

1. 通过Shell连接主机

ssh -q -l root -p 22 10.110.56.220

1. 修改主机名，如:

sudo -i hostnamectl set-hostname manager.bigdata.com

1. 查看主机名:

cat /etc/hostname

#### 2.1.2 修改ip与域名的对应关系 (集群所有节点均需执行此操作)

1. 运行命令 vi /etc/hosts

添加集群内所有节点的ip以及对应主机名：

172.16.15.220 manager.bigdata.com manager

172.16.15.221 master1.bigdata.com master1

172.16.15.222 master2.bigdata.com master2

172.16.15.223 worker1.bigdata.com worker1

172.16.15.224 worker2.bigdata.com worker2

172.16.15.225 worker3.bigdata.com worker3

1. 查看修改是否成功: cat /etc/hosts
2. 将修改好的配置文件分发到集群中其他节点:
3. scp /etc/hosts root@master1.bigdata.com:/etc
4. scp /etc/hosts root@master2.bigdata.com:/etc
5. scp /etc/hosts root@worker1.bigdata.com:/etc
6. scp /etc/hosts root@worker2.bigdata.com:/etc
7. scp /etc/hosts root@worker3.bigdata.com:/etc
8. ...
9. 运行 reboot 命令重启

#### 2.1.3 关闭seLinux (集群所有节点均需执行此操作)

1. 暂时关闭:

setenforce 0

1. 永久关闭:

vi /etc/selinux/config

SELinux设置为:

SELINUX=disabled

#### 2.1.4 关闭Linux透明大页 (集群所有节点均需执行此操作)

1. 暂时关闭:

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

1. 开机关闭:

vi /etc/rc.local

加入代码:

if test -f /sys/kernel/mm/transparent\_hugepage/enabled; then

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

fi

1. 查看透明大页是否已经关闭:

cat /sys/kernel/mm/transparent\_hugepage/enabled

显示如下内容说明透明大页已经关闭:

always madvise [never]

#### 2.1.5 关闭防火墙 (集群所有节点均需执行此操作)

1. 禁用防火墙:

ufw disable

1. 查看防火墙状态:

ufw status

#### 2.1.6 打通SSH (mannager节点需执行此操作)

server端要通过ssh协议将软件包分发到集群中各节点上，所以要打通server端到集群中各节点的ssh。

1. Shell登录Manager节点

ssh -q -l root -p 22 10.110.56.220

1. 打通ambari-server端到集群中所有节点的ssh:
2. sudo -i ssh-keygen
3. sudo -i ssh-copy-id root@manager.bigdata.com
4. sudo -i ssh-copy-id root@master1.bigdata.com
5. sudo -i ssh-copy-id root@master2.bigdata.com
6. sudo -i ssh-copy-id root@worker1.bigdata.com
7. sudo -i ssh-copy-id root@worker2.bigdata.com
8. sudo -i ssh-copy-id root@worker3.bigdata.com

### 2.2 源配置 (集群所有节点均需执行此操作)

#### 2.2.1 源文件修改

1. 修改源列表文件 运行命令:

vi /etc/apt/sources.list

加入:

deb http://10.221.129.22/kylin-arm64/Kylin-4.0.2/ juniper main

deb http://10.221.129.22/kylin-arm64/kylin4 juniper main

注释掉文件中的其他内容。

1. ambari安装源 写入 /etc/apt/sources.list.d/ambari.list 文件中:
2. echo 'deb http://10.221.129.22/kylin-arm64/ambari/kylin4/2.7.3.0-0/ Ambari main' > /etc/apt/sources.list.d/ambari.list

#### 2.2.2 源更新

1. 运行命令:

apt-get update

1. 安装源后所存放的目录:

/var/cache/apt/archives

#### 2.2.3 添加密钥:

1. 获取密钥文件:

wget http://10.221.129.22/kylin-arm64/ambari/kylin4/2.7.3.0-0/gpg/Inspur\_HD\_Manager\_Signing\_Key\_2019.pub

1. 添加密钥文件:

apt-key add Inspur\_HD\_Manager\_Signing\_Key\_2019.pub

### 2.3 ntp时间同步 (集群所有节点均需执行此操作)

集群中各节点从managerNode节点同步时间，managerNode节点从标准时钟服务器同步时间。当managerNode节点出现故障时，各节点从masterNode节点同步时间。

#### 2.3.1 安装ntp服务

1. 卸载冲突包（如果存在）

apt autoremove cifs-utils localechooser-data user-setup

1. 安装ntp，若存在未安装的依赖包则会提示安装

apt-get install -y ntp

1. 按照提示内容安装ntp依赖的包（此处为以下3个）

apt install -y ntpdate ntp ntp-doc ntpstat

1. 检查ntp是否安装成功:

dpkg --get-selections ntp

#### 2.3.2 修改ntp配置文件

输入命令修改ntp配置L vi /etc/ntp.conf

集群各类节点ntp.conf配置文件内容如下所示：

1. manager节点:
2. restrict -4 default kod notrap nomodify nopeer limited #删除noquery
3. restrict -6 default kod notrap nomodify nopeer limited #删除noquery
4. restrict 192.168.6.0 mask 255.255.255.0 nomodify notrap
5. server 192.168.0.1 prefer
6. server 127.127.1.0
7. fudge 127.127.1.0 stratum 8

参数说明:

192.168.6.0 和 255.255.255.0 是集群所在网段的网关和子网掩码。

192.168.0.1 是主时钟源的IP地址，请根据实际情况替换，prefer 表示优先选择的时钟源。

1. master节点:
2. restrict -4 default kod notrap nomodify nopeer limited #删除noquery
3. restrict -6 default kod notrap nomodify nopeer limited #删除noquery
4. restrict 192.168.6.0 mask 255.255.255.0 nomodify notrap
5. server manager.bigdata.com prefer
6. server 127.127.1.0
7. fudge 127.127.1.0 stratum 9
8. worker节点和slave节点:
9. server manager.bigdata.com prefer
10. server master1.bigdata.com
11. server master2.bigdata.com

#### 2.3.3 使master、worker、slave节点从manager节点同步时间

在启动ntp服务之前先手工同步下集群中各节点时间，ntp服务启动后，此命令将不能再执行：

service ntp stop

ntpdate manager.bigdata.com

#### 2.3.4 启动ntp服务

systemctl start ntp

#### 2.3.5 设置开机启动ntp服务

systemctl enable ntp

/lib/systemd/systemd-sysv-install enable ntp

#### 2.3.6 查看ntp状态

ntpq -p

### 2.4 openjdk安装 (集群所有节点均需执行此操作)

1. 创建并切换到jdk64文件夹

mkdir /usr/jdk64 cd /usr/jdk64

1. 获取openjdk包:

wget http://10.221.129.22/kylin-arm64/openjdk1.8.0\_aarch/openjdk-1.8.0-internal\_aarch64.tar.gz

1. 解压openjdk包:

tar -zxvf openjdk-1.8.0-internal\_aarch64.tar.gz

将解压得到的openjdk-1.8.0-internal文件夹改名为jdk1.8.0\_221方便后续配置:

mv openjdk-1.8.0-internal jdk1.8.0\_221

1. 配置环境变量

编辑环境变量配置文件:

vi /etc/profile

添加内容:

export JAVA\_HOME=/usr/jdk64/jdk1.8.0\_221

CLASSPATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/lib/mysql-connector-java.jar

export PATH=$PATH:$JAVA\_HOME/bin

1. 运行命令:

source /etc/profile

### 2.5 安装JDBC驱动文件 (集群所有节点均需执行此操作)

1. 下载mysql-connector-java-5.1.47.jar

cd /usr/jdk64/jdk1.8.0\_221/lib wget http://10.221.129.22/kylin-arm64/Kylin-4.0.2/pool/main/m/mysql-5.7/mysql-connector-java-5.1.47.jar

1. 查看文件:

ll

1. 创建软链接:

ln -s mysql-connector-java-5.1.47.jar mysql-connector-java.jar

### 2.6 python-dev安装 (集群所有节点均需执行此操作)

1. 获取python-dev:

wget http://10.221.129.22/kylin-arm64/kylin4/pool/main/p/python-dev/python-dev\_2.7\_all.deb

1. 解压

dpkg -i python-dev\_2.7\_all.deb

1. 若解压不成功则强制执行：

dpkg -i --force-overwrite python-dev\_2.7\_all.deb

### 2.7 安装配置MySQL (数据库节点需执行此操作，此处manager为主节点)

#### 2.7.1 安装和初始化

版本：MySQL5.7

1. MySQL使用的是系统盘的源，所以直接运行安装命令:

apt-get install mysql-server

1. 启动数据库:

systemctl stop mysql.service systemctl start mysql.service

1. 查看是否启动成功:

systemctl status mysql.service4

1. mysql5.7安装时会提示设置root密码，默认密码为空。将root用户密码修改为bigdata:

mysql -uroot -p use mysql; update mysql.user set authentication\_string=password('bigdata') where user='root';

1. 授权root用户在其他节点访问数据库的权限:

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'bigdata' WITH GRANT OPTION; FLUSH PRIVILEGES;

1. 重启数据库:

systemctl restart mysql.service

1. 登录数据库，执行如下命令验证数据库编码是否为utf-8:

show variables like 'char%';

若数据库编码是否不是utf-8，运行如下命令：

vi /etc/mysql/mysql.conf.d/mysqld.cnf

如果需要修改为utf8，在 lc-messages-dir = /usr/share/mysql 语句后加入:

character-set-server = utf8

1. 将数据库配置为允许远程连接

vi /etc/mysql/mysql.conf.d/mysqld.cnf

配置修改如下：

skip-external-locking

bind-address = 0.0.0.0

#### 2.7.2 MySQL主从配置

1. 在数据库主节点创建同步用户sync

create user 'sync'@'%' identified by 'bigdata';

1. 给sync用户赋予主从复制权限

grant replication slave on \*.\* to 'sync'@'%' identified by 'bigdata' with grant option;

1. 在数据库从节点验证sync用户登录主节点mysql数据库 登录数据库从节点所在的主机，连接主节点mysql数据库：

mysql -usync -pbigdata -hmanager.bigdata.com

1. 配置主从、节点的 /etc/my.cnf 文件 主节点 /etc/my.cnf 文件配置内容如下所示：
2. [mysqld]
3. datadir=/var/lib/mysql
4. #socket=/var/lib/mysql.sock
5. user=mysql
6. symbolic-links=0
7. log\_bin\_trust\_function\_creators=1
8. log-bin=mysql-bin
9. binlog-do-db=ambari
10. binlog-do-db=hue
11. binlog-do-db=hive
12. binlog-do-db=ranger
13. binlog-do-db=ranger\_audit
14. binlog-do-db=oozie
15. binlog-ignore-db=mysql
16. binlog\_format=mixed
17. server-id = 1
18. character-set-server=utf8
19. init\_connect='SET NAMES utf8'
20. [client]
21. default-character-set=utf8
22. [mysql]
23. no-auto-rehash
24. default-character-set=utf8
25. [mysqld\_safe]
26. log-error=/var/log/mysqld.log
27. pid-file=/var/lib/mysql/manager.bigdata.pid
28. replicate-do-db=all

说明：需要做备份的数据库通过“binlog-do-db=ambari”参数进行配置。

从节点 /etc/my.cnf 文件配置内容如下所示：

[mysqld]

datadir=/var/lib/mysql

#socket=/var/lib/mysql.sock

user=mysql

symbolic-links=0

log\_bin\_trust\_function\_creators=1

log-bin=mysql-bin

replicate-do-db=ambari

replicate-do-db=hue

replicate-do-db=hive

replicate-do-db=ranger

replicate-do-db=ranger\_audit

replicate-do-db=oozie

replicate-ignore-db=mysql

binlog\_format=mixed

read\_only=1

server-id = 2

default-storage-engine=InnoDB

character-set-server=utf8

init\_connect='SET NAMES utf8'

[client]

default-character-set=utf8

[mysql]

no-auto-rehash

default-character-set=utf8

[mysqld\_safe]

log-error=/var/log/mysqld.log

pid-file=/var/lib/mysql/master1.bigdata.pid

1. 分别登录主、从节点，重启MySQL服务

systemctl restart mysql

1. 登录主节点的mysql数据库，对主节点上锁并查看状态

mysql -uroot -pbigdata flush tables with read lock; show master status;

1. 登录从节点的mysql

mysql -uroot -pbigdata

1. 在从节点关闭slave

stop slave;

1. 在从节点配置从节点要同步的主节点信息

change master to master\_host='172.16.15.220', master\_user = 'sync', master\_password = 'bigdata', master\_log\_file = 'mysql-bin.000001', master\_log\_pos = 6437;

参数解释说明如下：

master\_user为同步用户； master\_log\_file和master\_log\_pos为主节点执行完show master status后系统返回的参数。

1. 在从节点启动mysql从节点

start slave;

1. 在从节点验证mysql从节点的状态是否成功

show slave status;

返回结果中以下两项都为yes，说明配置成功：

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

1. 主节点解锁 unlock tables;

### 2.8 在manager上安装ambari

#### 2.8.1 准备环境

创建ambari数据库及ambari用户并赋权

1. 通过shell登录集群的manager节点

ssh -q -l root -p 22 10.110.56.220

1. 连接mysql数据库

mysql -uroot -p

1. 创建ambari数据库

create database ambari;

1. 创建ambari用户并赋权，manager.bigdata.com 根据实际情况改为对应的hostname

grant all privileges on \*.\* to 'ambari'@'manager.bigdata.com' identified by 'bigdata' with grant option;

FLUSH PRIVILEGES;

#### 2.8.2 安装ambari-server

运行命令:

apt-get install ambari-server

#### 2.8.3 初始化ambari

1. 运行命令:

ambari-server setup

配置过程中选项如下:

Customize user ... daemon: n

JDK: 2

JAVA\_HOME:/usr/jdk64/jdk1.8.0\_221

GPL: y

database configuration: y

3 (mysql选项)

Hostname: 主机的hostname(如:kylin2.bigdata.com)

Port: 3306

Database name: ambari

Username: ambari

Password: bigdata

jdbc: n

输入: /usr/jdk64/jdk1.8.0\_221/lib/mysql-connector-java.jar

若在配置时没有jdbc选项，需要继续运行如下命令配置jdbc:

ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/jdk64/jdk1.8.0\_221/lib/mysql-connector-java.jar

1. 执行sql脚本

登录数据库:

mysql -uroot -pbigdata

运行命令:

use ambari;

source /var/lib/ambari-server/resources/Ambari-DDL-MySQL-CREATE.sql;

#### 2.8.4 启动ambari-server

1. 启动ambari:

ambari-server start

/etc/ambari-server/conf/ambari.properties

1. 测试登录 打开浏览器，以manager(10.110.56.220)为例，输入 <http://10.110.56.220:8080/> 登陆web页面成功表示manager安装成功

### 2.9 部署组件

#### 2.9.1 准备环境

1. 配置MySQL

提前配置好 Hive、 Oozie 、Ranger所对应的数据库，并对相应用户授予权限：

root用户登录MySQL：

(1) Hive组件:

create database hive;

use hive;

grant all privileges on hive.\* to 'hive'@'%' identified by 'bigdata';

flush privileges;

(2) Oozie组件:

create database oozie;

use mysql;

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

flush privileges;

(3) Ranger组件:

create database ranger;

use mysql;

grant all privileges on ranger.\* to 'ranger'@'%' identified by 'bigdata';

flush privileges;

1. 开启 innodb 表索引字符串最大长度限制（如若不执行此步骤，则 ranger 启动失败）
2. mysql -uroot -pbigdata
3. use mysql;
4. set global innodb\_file\_format = BARRACUDA;
5. set global innodb\_large\_prefix = ON;

注意：各组件依赖关系如下图所示，在安装部署过程中，有依赖关系的组件注意先部署被依赖的组件。

建议各组件安装顺序如下：

Zookeeper --> HDFA --> HD Metrics --> Yarn + MapReduce --> (...其他组件...) --> Kerberos

安装过程中出现的异常会有提示信息，请根据提示信息做相应操作。

可以在 /var/log/组件名/ 中查看安装日志，便于排查错误。

访问快速链接查看安装成功的WebUI，如果登录WebUI的主机没有配置 IP 和 hostname 的映射关系，在链接到组件WebUI界面时需要把域名更换成IP地址。

#### 2.9.2 安装源信息

http://10.221.129.22/kylin-arm64/ambari/kylin4/2.7.3.0-0/

http://10.221.129.22/kylin-arm64/hortonworks/HDP-GPL/kylin4/3.1.0.0-78/

http://10.221.129.22/kylin-arm64/hortonworks/HDP-UTILS/kylin4/1.1.0.22/

http://10.221.129.22/kylin-arm64/hortonworks/HDP/kylin4/3.1.0.0-78/

#### 2.9.3 常规组件部署（HDFS、ZooKeeper、HD Metrics、YARN + MapReduce2）

1. 登录Web页面

在浏览器中输入 <http://10.200.72.2:8080/> 进入登录页面（ip地址修改为对应值）。

登录账密: admin admin

1. 点击“启动安装向导”按钮，进行Insight HD部署。
2. 在“开始”向导页输入要创建的集群名称，点击“下一步”。
3. 进入“选择版本”向导页，版本信息选择: HDP-3.1.0.0，并选择使用本地存储库。输入本地源地址，操作系统选择: kylin4 （其他操作系统版本移除）。单击“下一步”。

URL对应输入源地址:

HDP: http://10.221.129.22/kylin-arm64/hortonworks/HDP/kylin4/3.1.0.0-78/

HDP-GPL: http://10.221.129.22/kylin-arm64/hortonworks/HDP-GPL/kylin4/3.1.0.0-78/

HDP-UTILS: http://10.221.129.22/kylin-arm64/hortonworks/HDP-UTILS/kylin4/1.1.0.22/

1. 在“安装选项”向导页中注册及确认主机。

(1) 在目标主机文本框中输入ambari-agent各节点hostname，如:manager.bigdata.com（每行输入一个节点的hostname即可）。

(2) 在主机注册信息中输入SSH私钥信息。

单击“选择文件”，选择路径“/root/.ssh/id\_rsa”来配置私钥。

单击“注册和确认”。

(安装完成后查看问题列表，最好解决掉所有警告)

1. 在“确认主机”向导页中，确认主机信息并安装ambari-agent相关进程。

安装完成后，进度条会呈现绿色，状态栏信息会显示“Success”。

点击提示信息“点击此处查看警告”，出现详细问题列表，建议将警告全部解决，便于此后集群的后续安装。

1. 在“选择服务”向导页中选择需要安装的组件。

建议先同时安装HDFS、ZooKeeper、HD Metrics和YARN + MapReduce2，勾选完成后单击“下一步”，后续其他组件单独安装。

说明：对于有依赖关系的组件，会有相关提示信息；同时，对未勾选的组件，集群安装完成后也可进行后续安装。

1. 在“指定Master”向导页中指定所选组件Master进程所在节点。

可完全采用默认设置，同时也可自定义进程要安装的节点。

点击每个进程对应的下拉框，在下拉框显示的所有agent节点中进行选择。设置完成后点击“下一步”。

说明：对于后面有绿色加号的组件，可以增加集群节点进行部署。

1. 在“指定Slaves和Client”向导页中指定所选组件Slaves或Client进程的所在节点。

可完全采用默认设置，也可进行自定义设置。

“主机”一栏显示当前可用的agent节点，对每个agent节点，勾选要安装的服务（说明：NFSGateway不需要安装）设置完成后单击“下一步”。

说明：slave和client的可选组件在右边显示，请将水平滚动条拖动到最右端进行查看。

1. 在“自定义服务”向导页检查各个组件需要的配置是否完备。

对于需要用户手工配置或者配置存在问题的组件会在“CREDENTIALS”中创建。

1. 进入“检查”向导页，检查配置信息，审查无误后，单击“部署”，进行组件安装。

注意: “检查”页签的“库”区域下的 **”redhat7(HDP-3.1)“** 和 **”redhat7(HDP-UTILS-1.1.0.22)“** 的值不能为空，如果为空，不要单击”部署“，刷新一下此页，此处的值便可以正常写入，然后单击”部署“即可。

1. 进入“安装、启动和测试”向导页，此过程中组件自动安装、启动和调试。

安装完成后节点状态显示100%，消息提示“成功”。点击【下一步】查看概览。

1. “概览”向导页提供了集群安装过程的概况，如果服务安装过程无异常，集群安装即成功。
2. 点击“完成”进入集群的控制台页面可监控集群。

同时，如果启动或测试过程出现问题，也会提示相关警告信息。

#### 2.9.4 HBase部署

HMaster 服务部署在Master节点上，RegionServer 服务部署在Worker节点上，如果有其他服务则部署在manager节点上。

1. 单击页面左侧“…”->“添加服务”，选择HBase并单击“下一步”。
2. 根据安装向导提示信息，执行安装步骤。
3. 安装成功后，有Success标识。
4. 若安装失败或者启动失败，请根据提示信息或者后台日志定位问题。

#### 2.9.5 Hive部署 (Manager)

Hive部署在manager节点上，Hive依赖HDFS、Yarn+MapReduce、Tez。其中HDFS、Yarn+MapReduce的已安装，Tez组件在安装Hive时会提示自动安装。

1. 单击页面左侧“…”->“添加服务”，选择Hive并单击“下一步”。
2. 根据安装向导提示信息，执行安装步骤（请提前创建Hive数据库和用户及赋权）。
3. 在“自定义服务”向导页，根据提示信息，配置数据库相关信息。

数据库选项选择 Exciting MySQL/MariaDB

输入数据库的账号密码（根据之前的设置，均为bigdata）

运行命令:

ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/jdk64/jdk1.8.0\_221/lib/mysql-connector-java.jar

点击TEST CONNECTION测试是否能连上数据库。

1. 安装成功后，有Success标识。
2. 安装完成后，执行如下语句对Hive数据库执行utf8编码:
3. mysql -uhive -pbigdata
4. use hive;
5. alter table COLUMNS\_V2 modify column COMMENT varchar(256) character set utf8;
6. alter table TABLE\_PARAMS modify column PARAM\_VALUE varchar(4000) character set utf8;
7. alter table PARTITION\_KEYS modify column PKEY\_COMMENT varchar(4000) character set utf8;
8. alter table TBLS modify column VIEW\_ORIGINAL\_TEXT mediumtext character set utf8;
9. alter table TBLS modify column VIEW\_EXPANDED\_TEXT mediumtext character set utf8;
10. 若安装失败或者启动失败，请根据提示信息或者后台日志定位问题。

#### 2.9.6 Spark2部署 (Manager)

Spark部署时，根据安装规划，Spark相关服务部署在manager节点上。

1. 单击页面左侧“…”->“添加服务”，选择Spark并单击“下一步”。
2. 根据安装向导提示信息，执行安装步骤。
3. 安装成功后，有Success标识。
4. 若安装失败或者启动失败，请根据提示信息或者后台日志定位问题。

#### 2.9.7 Oozie部署 (Manager)

Oozie部署时，根据安装规划，Oozie相关服务部署在 manager 节点上。

1. 单击页面左侧“…”->“添加服务”，选择 Oozie 并单击“下一步”。
2. 根据安装向导提示信息，执行安装步骤（请提前创建**Oozie数据库和用户及赋权**）。

根据之前的设置，数据库账号密码为: bigdata bigdata

点击 TEST CONNECTION 测试是否能连上数据库。

1. 安装成功后，有Success标识。
2. 若安装失败或者启动失败，请根据提示信息或者后台日志定位问题。

#### 2.9.8 Ranger部署(Manager)

Ranger部署时，根据安装规划，Ranger相关服务部署在manager节点上。

1. 单击页面左侧“…”->“添加服务”，选择 Ranger 并单击“下一步”。
2. 根据安装向导提示信息，执行安装步骤（请提前创建**Ranger数据库和用户及赋权**）。

以 manager 为例:

DB Flavor: MYSQL

Ranger DB host: manager.bigdata.com

Ranger DB Password: bigdata

DBA username: root

DBA password: bigdata

1. 安装过程中，请将 “Ranger Plugin” 页签中，各组件的插件 plugin 打开。
2. 安装过程中，请将 “Ranger Audit” 页签的 “Audit to Solr” 和 “Audit to HDFS” 选项关掉。
3. 配置 “ADVANCED” 页签中 “高级设置ranger-env” 区域的内容。
4. 安装成功后，有 Success 标识。
5. 若安装失败或者启动失败，请根据提示信息或者后台日志定位问题。

#### 2.9.9 Sqoop部署 (Manager)

Sqoop部署时，根据安装规划，Sqoop相关服务部署在manager节点上。

1. 单击页面左侧“…”->“添加服务”，选择Sqoop并单击“下一步”。
2. 根据安装向导提示信息，执行安装步骤。
3. 安装成功后，有Success标识。
4. 若安装失败或者启动失败，请根据提示信息或者后台日志定位问题。

#### 2.9.10 Zeppelin部署 (Slave)

Zeppelin部署时，根据安装规划，Zeppelin相关服务部署在slave节点上。

1. 单击页面左侧“…”->“添加服务”，选择Zeppelin并单击“下一步”。
2. 根据安装向导提示信息，执行安装步骤。
3. 安装成功后，有Success标识。
4. 若安装失败或者启动失败，请根据提示信息或者后台日志定位问题。

### 2.10 开启HA

目前Insight HD中HDFS、YARN、HBase可以开启HA模式，其中，以HDFS开启HA过程最为典型，对于开启YARN HA操作请参考本章开启HDFS HA，其他组件则更为简单直接点击添加按钮即可，本文将不再赘述。

1. 进入设置向导操作。点击“服务”菜单中的“HDFS”组件。
2. 在“服务操作”中单击“启用NameNode HA”。
3. 进入“开始设置”向导页，如果Hbase正在运行，则需要停止Hbase服务，在文本框中输入Nameservice ID（例如，mycluster），点击“下一步”。
4. 进入“选择主机”向导页，指定附加NameNode以及JournalNode部署节点， 配置好后单击“下一步”按钮。
5. 在“检查”概览界面显示了节点的变化情况，删除Secondary NameNode，添加 Additional NameNode、Journal Node。
6. 进入“创建检查点”向导页，进行创建checkpoint 操作，此操作需要手工进行，根据提示信息，登录当前NameNode节点，执行显示在页面上的命令，执行完后单击 “下一步”。

手工执行命令及返回结果如下所示，表示命令执行成功。

1. 进入“配置组件”向导页进行相关配置（系统自动完成）。

等待1-3min，配置完成后单击“下一步”。

1. 在“启动NameNode HA向导”页进行初始化Journal Node操作，此操作需要根据提示信息手工进行，登录当前NameNode节点，执行显示在页面上的命令，执行完后，单击“下一步”。
2. 进入“启动组件”向导页，启动ZooKeeper和NameNode（系统自动完成），完成后单击“下一步”。
3. 进入“初始化NameNode HA元数据”向导页，初始化NameNode HA 元数据，此操作需要根据提示信息手工进行，登录当前NameNode节点，执行显示在页面上的命令，执行完后单击“下一步”。
4. 进入“完成HA设置”向导页，启动HA。

等待HA设置完成，单击“完成”。

1. 查看HDFS服务概要界面，会显示HA相关信息，HA安装结束。

### 2.11 开启安全模式

Kerberos KDC安装采用主从模式，主库master位于kylin2.bigdata上，从库slave位于kylin30.bigdata上，安装及配置过程如下。

#### 2.11.1 KDC Master安装

1. 安装Kerberos: apt-get install krb5-kdc krb5-admin-server

以manager为例:

* 1. 弹出relam对话框，默认为当前机器名称后缀大写，如:manager.bigdata.com，默认值为BIGDATA.COM，保留此默认值，按回车键
  2. kerberos servers for your realm, 填写kdc主机名，如:manager.bigdata.com，按回车键
  3. Administrative servers for your kerberos realm, 填写kdc主机名:manager.bigdata.com，按回车键
  4. 确定，按回车键

1. 创建新域: sudo krb5\_newrealm
2. 修改配置文件
   1. **修改krb5.conf**

运行命令 vi /etc/krb5.conf

修改参数:

[libdefaults]

renew\_lifetime = 7d

forwardable = true

default\_realm = BIGDATA.COM

ticket\_lifetime = 24h

dns\_lookup\_realm = false

dns\_lookup\_kdc = false

default\_ccache\_name = /tmp/krb5cc\_%{uid}

#default\_tgs\_enctypes = aes des3-cbc-sha1 rc4 des-cbc-md5

#default\_tkt\_enctypes = aes des3-cbc-sha1 rc4 des-cbc-md5

[domain\_realm]

bigdata.com = BIGDATA.COM

[logging]

default = FILE:/var/log/krb5kdc.log

admin\_server = FILE:/var/log/kadmind.log

kdc = FILE:/var/log/krb5kdc.log

[realms]

BIGDATA.COM = {

admin\_server = kylin2.bigdata.com

kdc = kylin2.bigdata.com

kdc = kylin30.bigdata.com

}

```

* 1. **修改kdc.conf**

运行命令 vi /etc/krb5kdc/kdc.conf

修改参数:

[kdcdefaults]

kdc\_ports = 750,88

[realms]

BIGDATA.COM = {

database\_name = /var/lib/krb5kdc/principal

admin\_keytab = FILE:/etc/krb5kdc/kadm5.keytab

acl\_file = /etc/krb5kdc/kadm5.acl

key\_stash\_file = /etc/krb5kdc/stash

kdc\_ports = 750,88

max\_life = 10h 0m 0s

max\_renewable\_life = 7d 0h 0m 0s

master\_key\_type = des3-hmac-sha1

supported\_enctypes = aes256-cts:normal arcfour-hmac:normal des3-hmac-sha1:normal des-cbc-crc:normal des:normal des:v4 des:norealm des:onlyrealm des:afs3

default\_principal\_flags = +preauth

}

```

* 1. **修改kadm5.acl**

运行命令 vi /etc/krb5kdc/kadm5.acl

修改参数:

admin/admin@BIGDATA.COM

1. 创建数据库

kdb5\_util create -r BIGDATA –s

1. 创建KDC超级管理员，根据提示信息此处需要输入管理员密码

kadmin.local -q "addprinc admin/admin@BIGDATA"

1. 启动:
2. sudo systemctl unmask krb5-admin-server
3. sudo systemctl enable krb5-admin-server

启动并查看状态:

sudo systemctl start krb5-admin-server

sudo systemctl status krb5-admin-server

1. 测试

运行命令:

kinit admin/admin

password为admin

查看当前认证用户:

klist

#### 2.11.2 辅助KDC安装

1. 安装

sudo apt install krb5-kdc krb5-admin-server

1. 修改配置文件

分别修改如下文件（操作与主KDC相同）:

/etc/krb5.conf /etc/krb5kdc/kdc.conf /etc/krb5kdc/kadm5.acl

#### 2.11.3 主从配置

1. 登录主KDC，运行命令:

kadmin.local

1. 添加用户

运行命令:

addprinc -randkey host/manager.bigdata.com

addprinc -randkey host/master1.bigdata.com

addprinc -randkey host/master2.bigdata.com

1. 主KDC生成keytab文件

运行命令（注意大小写）:

ktadd -norandkey -k /etc/krb5.keytab host/manager.bigdata.com@BIGDATA.COM host/master1.bigdata.com@BIGDATA.COM host/master1.bigdata.com@BIGDATA.COM

1. 运行 quit 退出
2. 将keytab发送到辅助KDC

以发送到 master1 为例，在主KDC节点运行命令:

scp /etc/krb5.keytab root@master1.bigdata.com:/etc

1. 列出keytab

运行命令:

sudo klist -k /etc/krb5.keytab

1. 主从机均编辑 kpropd.acl 文件

运行命令: vi kpropd.acl

加入:

host/kylin2.bigdata.com@BIGDATA.COM

host/kylin30.bigdata.com@BIGDATA.COM

1. 辅助KDC上创建空数据库

运行命令:

sudo kdb5\_util -s create

1. 启动kpropd守护程序

运行命令:

sudo kpropd -S

1. 创建主体数据库的转储文件

运行命令:

sudo kdb5\_util dump /var/lib/krb5kdc/dump

1. 主KDC 数据库转推到 辅助KDC

运行命令:

sudo kprop -r BIGDATA.COM -f /var/lib/krb5kdc/dump master1.bigdata.com

运行成功会有 SUCCEEDED 信息:

Database propagation to master1.bigdata.com: SUCCEEDED

1. 定时推送数据库到辅助KDC

运行命令:

crontab -e

加入内容:

2 \* \* \* \* /usr/sbin/kdb5\_util dump /var/kerberos/krb5kdc/dump && /usr/sbin/kprop -r BIGDATA -f /var/kerberos/krb5kdc/dump master1.bigdata.com

1. 在辅助KDC创建一个存储文件以保存Kerberos主密钥

运行命令:

sudo kdb5\_util stash

结果: Using existing stashed keys to update stash file.

1. 在辅助KDC上启动krb5-kdc守护程序

运行命令:

sudo systemctl start krb5-kdc.service

#### 2.11.4 Kerberos客户端

1. 安装krb5-user和libpam-krb5等包

运行命令:

sudo apt install krb5-user libpam-krb5 libpam-ccreds auth-client-config

1. 配置

运行命令:

sudo dpkg-reconfigure krb5-config

确保 /etc/krb5.conf 中有以下内容:

[libdefaults] default\_realm = BIGDATA.COM... [realms] BIGDATA.COM = { admin\_server = kylin2.bigdata.com kdc = kylin2.bigdata.com kdc = kylin30.bigdata.com }

1. 测试

运行命令:

kinit admin/admin@BIGDATA.COM

查看详情:

klist

1. 使用auth-client-config来配置libpam-krb5模块，使得在登录时请求票据:

运行命令:

sudo auth-client-config -a -p kerberos\_example

## 第三章 集群高可用测试

### 3.1 HDFS测试

#### 3.1.1 主从替换测试

##### 3.1.1.1查看节点状态

hdfs haadmin -getServiceState nn1

active

hdfs haadmin -getServiceState nn2

standby

都为standby时，需要手动将一个节点从standby切换为active，以nn1为例:

hdfs haadmin -transitionToActive --forcemanual nn1

##### 3.1.1.2 关闭nn1进程

在nn1主机运行命令:

jps

找到NameNode进程号并关闭

kill -9 进程号

查看nn2进程号，发现从standby转为active

hdfs haadmin -getServiceState nn2

active

##### 3.1.1.3 重新开启nn1

find / -name hadoop-daemon.sh

找到文件位置并运行:

/usr/hdp/3.1.0.0-78/hadoop/sbin/hadoop-daemon.sh start namenode

##### 3.1.1.4 再次查看节点状态:

hdfs haadmin -getServiceState nn1

standby

hdfs haadmin -getServiceState nn2

active

#### 3.1.2 NameNode测试

##### 3.1.2.1 切换到hdfs用户

su hdfs

##### 3.1.2.2 本地准备用于传输的文件，存放在路径 /tmp/uploadtest 中

##### 3.1.2.3 集群上新建测试文件夹datatest

hdfs dfs -mkdir /datatest

##### 3.1.2.4 本地文件传到集群

hdfs dfs -put /tmp/uploadtest/ /datatest

##### 3.1.2.5 手动停止namenode后，报错信息如下:

##### 3.1.2.6 查看集群文件

运行命令:

hdfs dfs -ls /datatest

hdfs dfs -ls /datatest/uploadtest

##### 3.1.2.7 本地新建文件夹 /tmp/localtest，将传输到集群中的文件下载到新文件夹

mkdir /tmp/localtest

hdfs dfs -copyToLocal /datatest/uploadtest /tmp/localtest

##### 3.1.2.8 查看下载文件

cd /tmp/localtest

ll

cd uplaodtest

ll

##### 3.1.2.9 比较新文件与上传文件的差异

diff /tmp/localtest/uploadtest /tmp/uploadtest

结果显示无差异

#### 3.1.3 DataNode测试

##### 3.1.3.1 切换到hdfs用户

su hdfs

##### 3.1.3.2 本地准备用于传输的文件，存放在路径 /tmp/uploadtest 中

##### 3.1.3.3 集群上新建测试文件夹datatest1，本地文件传到集群

hdfs dfs -mkdir /datatest1

hdfs dfs -put /tmp/uploadtest/ /datatest1

以3个DataNode的集群为例。

##### 3.1.3.4 手动关闭一个DataNode节点，UI界面如下所示:

##### 3.1.3.5 后台报错信息如下:

##### 3.1.3.6 UI最终显示文件

##### 3.1.3.7 运行命令查看集群文件

hdfs dfs -ls /datatest1

hdfs dfs -ls /datatest1/uploadtest

##### 3.1.3.8 本地新建文件夹 /tmp/localtest1，将传输到集群中的文件下载到新文件夹

创建文件夹：

mkdir /tmp/localtest1

hdfs dfs -copyToLocal /datatest1/uploadtest /tmp/localtest1

##### 3.1.3.9 查看下载文件

cd /tmp/localtest1

ll

cd uploadtest1

ll

##### 3.1.3.10 比较新文件与上传文件的差异

diff /tmp/localtest1/uploadtest /tmp/uploadtest

结果显示无差异

### 3.2 YARN主备节点切换测试

#### 3.2.1 切换到yarn用户

su yarn

#### 3.2.2 查看节点状态

yarn rmadmin -getServiceState rm1

active

yarn rmadmin -getServiceState rm2

standby

#### 3.2.3 关闭YARN进程

找到ResourceManager进程号

jps

kill -9 进程号

查看状态:

yarn rmadmin -getServiceState rm2

active

#### 3.2.4 重启YARN进程

找到并运行yarn-daemon.sh文件启动ResourceManager

find / -name yarn-daemon.sh

/usr/hdp/3.1.0.0-78/hadoop-yarn/sbin/yarn-daemon.sh start resourcemanager

#### 3.2.5 再次查看节点状态:

yarn rmadmin -getServiceState rm1

standby

yarn rmadmin -getServiceState rm2

active

## 第四章 常见问题 FAQ

### 4.1 MySql相关

#### 4.1.1 重置root密码时，报错 Plugin 'auth\_socket' is not loaded

需要将auth插件更改为

mysql\_native\_password

免密登录数据库后，运行如下命令:

```

use mysql;

update user set authentication\_string=PASSWORD('bigdata') where User='root';

update user set plugin="mysql\_native\_password" where User='root';

flush privileges;

exit

```

重启mysql登录即可。

#### 4.1.2 Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock'

原因在于 /var/run 下面没有mysqld目录。

解决方法:

```

mkdir -p /var/run/mysqld

chown mysql /var/run/mysqld/

service mysql restart

```

#### 4.1.3 mysql主从配置slave节点slave start报错: ERROR 1872 (HY000): Slave failed to initialize relay log info structure from the repository

由于 mysql.slave\_relay\_log\_info 表中保留了以前的复制信息，导致新从库启动时无法找到对应文件，需要清理掉该表中的记录。

运行命令:

slave reset;

再重新运行change master...

cat id\_rsa.pub >> .ssh/authorized\_keys

#### 4.1.4 组件数据库配置有误

安装组件时没有配置正确的数据库（默认为 new database）。

应提前配置数据库，然后在安装组件时选择 existing MySql/MariaDB 并输入配置好的数据库信息：

其他组件安装时也应该选择类似的数据库配置。

### 4.2 ambari-server相关

#### 4.2.1 ambari-servers 启动失败

运行 ambari-server start时报错 no valid keyboard

原因: 缺少 /var/lib/ambari-server/keys/db/newcerts

运行命令:

mkdir -p /var/lib/ambari-server/keys/db/newcerts

#### 4.3.2 REASON: Unable to detect a system user for Ambari Server.

运行命令:

vi /etc/ambari-server/conf/ambari.properties

添加内容:

ambari-server.user=root

### 4.3 HDFS 安装相关

#### 4.3.1 安装向导源丢失

1. 开启新的tab页刷新浏览器
2. 将源手动写入数据库
3. 重试

#### 4.3.2 HDFS NameNode启动报错: Execution of '/usr/hdp/current/hadoop-hdfs-namenode/bin/hdfs dfsadmin -fs hdfs://kylin2.bigdata.com:8020 -safemode get | grep 'Safe mode is OFF'' returned 1.

原因:

NameNode Server threads 参数值400

修改为200后正常启动。

#### 4.3.3 HDFS NameNode启动报错: Exiting with status 1: java.lang.OutOfMemoryError: unable to create new native thread

原因: cgroup 限制，需要修改系统参数。

运行命令:

vi /etc/systemd/system.conf

DefaultTasksMax默认值512，修改为:

DefaultTasksMax=65535

vi /etc/systemd/logind.conf

UserTasksMax默认值12288，修改为:

UserTasksMax=65535

### 4.4 其他问题

#### 4.4.1 DNS错误:

修改host:

vi /etc/hosts

#### 4.4.2 报错 UnicodeDecodeError: 'ascii' codec can't decode byte 0xe6 in position 1: ordinal not in range(128)

在报错的.py文件开头添加:

import sys

reload(sys)

sys.setdefaultencoding('utf-8')

#### 4.4.3 登录时不是root用户，则需要开启root：

ssh -q -l inspur -p 22 10.110.56.220

1. 初始化root密码： sudo passwd root
2. 将PermitRootLogin项prohibit-password改为yes: sudo vi /etc/ssh/sshd\_config PermitRootLogin=yes
3. 重启sshd服务即可: service sshd restart

#### 4.4.4 安装ntp时80端口被apache2占用：

journalctl -xe netstat -anp|grep 80 systemctl status apache2 systemctl stop apache2

root@master1:# service ntpdate stop Failed to stop ntpdate.service: Unit ntpdate.service not loaded. root@master1:# service ntp stop root@master1:~# ntpdate manager.bigdata.com 25 Nov 17:04:27 ntpdate[4930]: adjust time server 172.16.15.220 offset 0.001752 sec

#### 4.4.5 票据问题

边栏 -> Cluster Admin -> 安全认证 -> zeppelin.server.kerberos.principal参数

修改参数：