## 目录

[目录 0](#_Toc394855855)

[1. 基础环境 1](#_Toc394855856)

[2. 准备工作 2](#_Toc394855857)

[2.1 配置环境 2](#_Toc394855858)

[2.1.1 配置hosts文件 2](#_Toc394855859)

[2.1.2 SSH无密码登入 2](#_Toc394855860)

[2.1.3 NTP 时间同步 3](#_Toc394855861)

[2.1.4 SELinux & iptables 关闭 3](#_Toc394855862)

[2.2 Java环境安装 4](#_Toc394855863)

[2.2.1 安装JDK 4](#_Toc394855864)

[2.2.2 配置环境变量 4](#_Toc394855865)

[3. Ambari安装配置 6](#_Toc394855866)

[3.1 配置本地源 6](#_Toc394855867)

[3.1.1 建立本地资源库 6](#_Toc394855868)

[3.1.2 配置repo文件 7](#_Toc394855869)

[3.1.3 配置Media源 8](#_Toc394855870)

[3.1.4 安装必要工具 8](#_Toc394855871)

[3.1.5 配置Media的http源 9](#_Toc394855872)

[3.1.6 安装ambari-server服务 9](#_Toc394855873)

[3.1.7 安装ambari客户端 10](#_Toc394855874)

[3.2 ambari服务器配置与管理 10](#_Toc394855875)

[4. 常见问题 19](#_Toc394855876)

[4.1 mapreduce 19](#_Toc394855877)

[4.2 oozie安装 20](#_Toc394855878)

# 基础环境

**本人配置**

操作系统：redhat6.4

内核版本：

内存大小：

处理器：

Ambari版本：ambari-1.6.0

HDP版本：HDP-2.1-latest-centos6-rpm.tar.gz

HDP-UTILS版本：HDP-UTILS-1.1.0.17-centos6.tar.gz

JDK版本：jdk-7u45-linux-x64

**Ambari安装的环境路径（选择安装所有服务的情况）：**

|  |  |  |  |
| --- | --- | --- | --- |
| **服务** | **安装路径**  **/usr/lib** | **日志路径**  **/var/log** | **配置文件**  **/etc** |
| ambari-server | **√** | **√** | **√** |
| ambari-agent | **√** | **√** | **√** |
| falcon | **√** | **√** | **√** |
| ganglia |  |  | **√** |
| hadoop | **√** | **√** | **√** |
| hbase | **√** | **√** | **√** |
| hive | **√** | **√** | **√** |
| hcatalog | **√** | **√** |  |
| oozie | **√** | **√** | **√** |
| tez | **√** | **√** | **√** |
| webhcat |  | **√** |  |
| zookeeper | **√** | **√** | **√** |
|  |  |  |  |

# 准备工作

本次配置使用hdp-m2作为主master节点

## 配置环境

### 配置hosts文件

所有机器都得执行，使用root用户

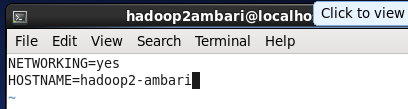
1. @ hostname hdp-m2（该命令可用于临时修改主机名）
2. @ vi /etc/hosts（该命令可用于配置主机名和IP的对应信息）

10.242.157.115 hdp-m1

10.242.157.117 hdp-m2

10.242.157.122 hdp-s1

1. @ vi /etc/sysconfig/network（该命令可用于修改网络主机名）



### SSH无密码登入

所有机器都得执行，使用root用户

@ yum install ssh （安装SSH协议）

@ yum install rsync （rsync是一个远程数据同步工具，可通过LAN/WAN快速同步多台主机间的文件）

@ service sshd restart （启动服务）

注：如果系统中没有安装SSH，需要进行以上操作。

@ ssh-keygen（该命令生成指定公私秘钥的名字，id\_dsa及id\_dsa.pub）

@ cat /root/.ssh/id\_dsa.pub >> /root/.ssh/authorized\_keys（把id\_rsa.pub追加到授权的key里面去,如果有多个master节点则需把所有的master节点上id\_dsa.pub文件内容整合到authorized\_keys文件中）

@ ll /root/.ssh/（查看共私钥、授权key）

@ ssh-copy-id -i id\_dsa.pub root@hdp-m1（将公钥copy到所有的机器上）

@ssh hdp-m1（测试SSH无密码登入是否成功）

### NTP 时间同步

Hdp-m2节点

将server上/etc/ntp.conf文件中以下两行的注释打开

@ vim /etc/ntp.conf

server 127.127.1.0

fudge 127.127.1.0 stratum 10

保存并重启ntpd服务（注意关闭防火墙，遇到no server suitable for synchronization found问题，则需要在从节点上service ntpd start）：

@ service ntpd start

@ chkconfig ntpd on

@ /etc/init.d/ntpd restart

@ service ntpd status

进入其他从节点进行时间同步：

@ ssh hdp-m1

@ vim /etc/ntp.conf

server hdp-m2

fudge 127.127.1.0 stratum 10

@ ntpdate hdp-m2

@ exit

### SELinux & iptables 关闭

Hdp-m2节点

@ setenforce 0（临时关闭selinux）

@ vi /etc/yum/pluginconf.d/refresh-packagekit.conf

enabled=0

所有节点

@ chkconfig iptables off（关闭防火墙关闭开机启动）

@ service iptables stop（关闭防火墙）

至此环境配置完成，关闭防火墙后方可SSH无密码登入。

## Java环境安装

所有的机器上都要安装JDK，现在就先在hdp-m2服务器安装，然后其他服务器按照步骤重复进行即可。安装JDK以及配置环境变量，需要以"**root**"的身份进行。

### 安装JDK

首先用**root**身份在"/usr"下创建"java"文件夹，再复制"/home/hadoop/"下的" jdk-7u45-linux-x64.gz"复制到"/usr/java"文件夹中。

@ mkdir /usr/java

@ cp /home/hambari/hadoop/jdk-7u45-linux-x64.gz /usr/java

@ tar -zxvf jdk-7u45-linux-x64.gz

然后查看"**/usr/java**"下面会发现多了一个名为"**jdk1.7.0\_45**"文件夹，说明我们的JDK安装结束，进入下一个"配置环境变量"环节。

### 配置环境变量

编辑"/etc/profile"文件，在后面添加Java的"JAVA\_HOME"、"CLASSPATH"以及"PATH"内容。

@ vim /etc/profile

在"**/etc/profile**"文件的**尾部**添加以下内容：

# set java environment

export JAVA\_HOME=/usr/java/jdk1.7.0\_45/

export JRE\_HOME=/usr/java/jdk1.7.0\_45/jre

export CLASSPATH=.:$CLASSPATH:$JAVA\_HOME/lib:$JRE\_HOME/lib

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

保存并退出，执行下面命令使其配置立即生效。

@ source /etc/profile

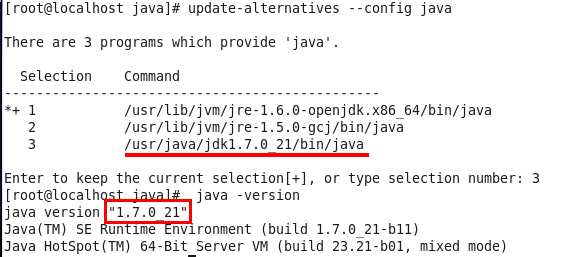
@ java -version

如果java -version命令显示的结果依然是openjdk，表示JDK还未更新，需要自己手动选择新版本的JDK，首先安装 java alternative ---jdk

@ update-alternatives --install /usr/bin/java java $JAVA\_HOME/bin/java 60

然后配置JDK：

@ update-alternatives --config java（选择最新的JDK即可）



# Ambari安装配置

首先要下载ambari及相关软件包的安装包，别尝试着在yum来下，特别慢，最好的方式是通过建立本地资源的方式来安装：

**ambari-1.6.0-centos6.tar.gz**

**HDP-2.1-latest-centos6-rpm.tar.gz**

**HDP-UTILS-1.1.0.17-centos6.tar.gz**

## 配置本地源

### 建立本地资源库

在hdp-m2机器上建立本地资源库，在var/www/html下建立一个hdp目录，如果没有该目录，表示httpd服务未安装，请先安装httpd服务yum install httpd （一般的LiveCD和DVD版本的IOS安装都是默认有的，但是如minimal版本的安装是没有的，目录地址根据实际情况设定）。

@ yum install httpd

如页面不能访问[http://hdp-m2](http://hdp-m2/)则看下面文件配置：



@ service httpd restart

@ mkdir /var/www/html/hdp

把下到的三个资源解压到/software/hdp之后，移动到hdp目录下：

@ cd /software/hdp

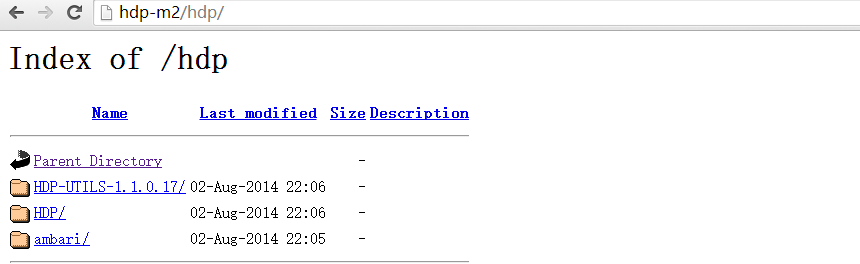
@ mv ambari /var/www/html/hdp

@ mv HDP /var/www/html/hdp

@ mv HDP-UTILS-1.1.0.17 /var/www/html/hdp

在网页上输入如下地址进行验证：

http://hdp-m2/hdp



### 配置repo文件

在hdp-m2机器上准备ambari.repo文件,，然后把这个文件复制到集群中的所有机器的/etc/yum.repos.d/目录下：

@ cd /etc/yum.repos.d

@ vim ambari.repo

**ambari.repo**

[ambari-1.x]

name=Ambari 1.x

baseurl=http://hdp-m2/hdp/ambari/centos6/1.x/updates/1.6.1

gpgcheck=1

gpgkey=http://hdp-m2/hdp/ambari/centos6/1.x/updates/1.6.1/RPM-GPG-KEYa/RPM-GPG-KEY-Jenkins

enabled=0

priority=1

[HDP-UTILS-1.1.0.16]

name=Hortonworks Data Platform Utils Version - HDP-UTILS-1.1.0.16

baseurl=http://hdp-m2/hdp/HDP-UTILS-1.1.0.17/repos/centos6

gpgcheck=1

gpgkey=http://hdp-m2/hdp/HDP-UTILS-1.1.0.17/repos/centos6/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins

enabled=0

priority=1

[HDP-UTILS-1.1.0.17]

name=Hortonworks Data Platform Utils Version - HDP-UTILS-1.1.0.17

baseurl=http://hdp-m2/hdp/HDP-UTILS-1.1.0.17/repos/centos6

gpgcheck=0

gpgkey=http://hdp-m2/hdp/HDP-UTILS-1.1.0.17/repos/centos6/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins

enabled=1

priority=1

[Updates-ambari]

name=ambari - Updates

baseurl=http://hdp-m2/hdp/ambari/centos6/1.x/updates/1.6.1

gpgcheck=1

gpgkey=http://hdp-m2/hdp/ambari/centos6/1.x/updates/1.6.1/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins

enabled=1

priority=1

### 配置Media源

进入hdp-m2机器上/etc/yum.repos.d文件夹还可能看到已存在四个文件：

CentOS-Debuginfo.repo CentOS-Vault.repo

CentOS-Base.repo CentOS-Media.repo  
rhel-source.repo

把这些文件都删除

### 安装必要工具

强制替换安装rpm -ivh --force 文件名(顺序执行) , 都可以在光盘的Package目录找到：

所有机器都运行：

rpm –ivh –-force deltarpm-3.5-0.5.20090913git.el6.x86\_64.rpm

rpm –ivh –-force python-deltarpm-3.5-0.5.20090913git.el6.x86\_64.rpm

rpm –ivh –-force createrepo-0.9.8-5.el6.noarch.rpm

rpm –ivh –-force openssl-1.0.1e-16.el6\_5.x86\_64.rpm

### 配置Media的http源

挂载iso中的Package到hdp-m2将机器上的/var/www/html/rhel下，然后：

@ mkdir /var/www/html/rhel

@ cp -r /media/cdrom/Packages /var/www/html/rhel

@ createrepo /var/www/html/rhel（生成repodata文件夹）

[root@hambari html]# createrepo /var/www/html/rhel

3720/3720 - Packages/rootfiles-8.1-6.1.el6.noarch.rpm

Saving Primary metadata

Saving file lists metadata

Saving other metadata

在所有机器上的/etc/yum.repos.d文件夹中添加一个myiso.repo文件，内容如下：

@ vim myiso.repo

[myiso]

name=myiso repo

baseurl=http://hdp-m2/rhel

gpgcheck=0

enabled=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

### mysql卸载

使用root用户登陆hdp-m2执行以下命令，使用的mysql安装文件为提供的MySQL-server-5.5.28-1.linux2.6.x86\_64.rpm，MySQL-client-5.5.28-1.linux2.6.x86\_64.rpm

1. 查找以前是否装有mysql

命令：rpm -qa|grep -i mysql



1. 删除mysql

删除命令：rpm -e --nodeps 包名



查看是否删除成功：



删除成功

1. 删除老版本mysql的开发头文件和库

命令：rm -fr /usr/lib/mysql

rm -fr /usr/include/mysql

注意：卸载后/var/lib/mysql中的数据及/etc/my.cnf不会删除，如果确定没用后就手工删除

rm -f /etc/my.cnf

rm -fr /var/lib/mysql

### mysql安装

1. 执行命令安装mysql服务端(文件参见附件清单中，mysql安装程序

/mysql-server-\*\*\*\*\*\*\*\*)

rpm -ivh MySQL-server-5.5.28-1.linux2.6.x86\_64.rpm

1. 执行命令安装mysql客户端(文件参见附件清单中，mysql安装程序

/mysql-client-\*\*\*\*\*\*\*\*)

rpm -ivh MySQL-client-5.5.28-1.linux2.6.x86\_64.rpm

1. 配置mysql

设置服务为开机自启动：chkconfig mysql on

开启服务：service mysql start

1. 启动设置root密码

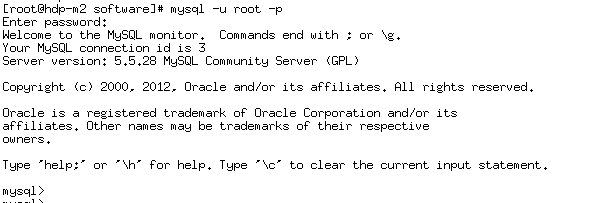
登陆命令行：mysql -u root mysql

更改密码：Update user set password=PASSWORD('123456') where user = 'root';

退出：exit;

1. 设置远程可访问root用户的权限，并指定密码

命令行执行：mysql –u root –p，第一次时，无需输入密码，直接按回车键



进入mysql的shell命令行，依次输入以下内容：

mysql> set PASSWORD = PASSWORD('123456');

提示成功



mysql> grant all on \*.\* to root@'%' identified by '123456';

提示成功



mysql> flush privileges;

提示成功



1. 创建用户密码和远程访问的权限

* hive用户，执行以下命令

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

grant all privileges on \*.\* to 'hive'@'localhost' identified by 'hive' with grant option;

grant all privileges on \*.\* to 'hive'@'hdp-m2' identified by 'hive' with grant option;

grant all privileges on \*.\* to 'hive'@'hdp-m1' identified by 'hive' with grant option;

flush privileges;

* ambari用户，执行以下命令

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

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

grant all privileges on \*.\* to 'ambari'@'hdp-m2' identified by 'ambari' with grant option;

flush privileges;

* bigdata用户，执行以下命令

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

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

grant all privileges on \*.\* to 'bigdata'@'hdp-m2' identified by 'bigdata' with grant option;

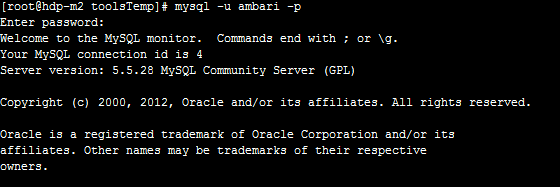
flush privileges;

退出：exit;

1. 测试用户是否已经创建

mysql -u ambari –p

输入密码ambari



表明用户已成功创建

### 数据库初始化

1. 创建数据库
2. 执行命令：mysql –u root –p 123456进入mysql shell窗口
3. 依次执行以下mysql语句

create database ambari default charset='UTF8';

create database hive default charset='UTF8';

create database bigdata default charset='UTF8';

1. 初始化数据 (这一步会在安装ambari-server时有提示，到时候根据提示执行即可)
2. 执行命令：mysql -u ambari -p ambari进入ambari用户操作窗口
3. 执行语句：source 路径/ Ambari-DDL-MySQL-CREATE.sql(路径视具体情况而定)

### 安装ambari-server服务

在hdp-m2机器上,从配置好的本地源安装ambari服务，如遇到postgresql版本太低，用CDROM源进行高版本的安装即可：

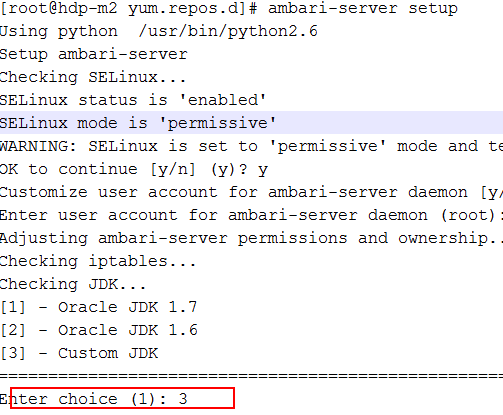
@ yum clean all

@ yum -y install ambari-server（输入y回车）

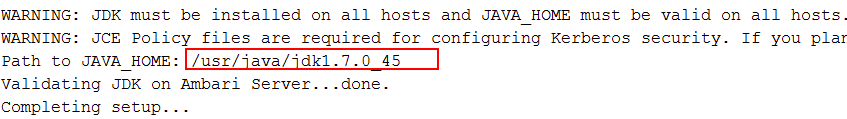
然后server上执行ambari-server setup，一路回车，安装ambari-server，它会自动安装配置PostgreSQL，同时要求输入用户名和密码，如果按n，它用默认的用户名/密码值：ambari-server/bigdata，如果输入有误可在/etc/ambari-server/conf/修改ambari.properties文件即可。

@ ambari-server setup

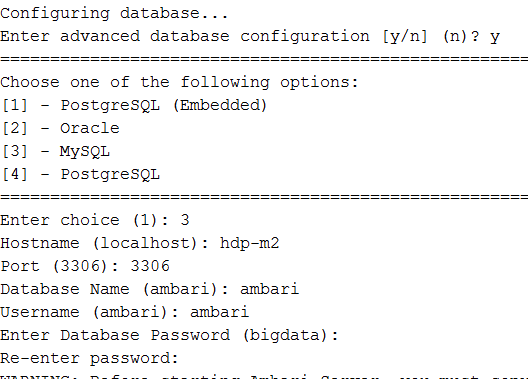
* 选择自定义的jdk



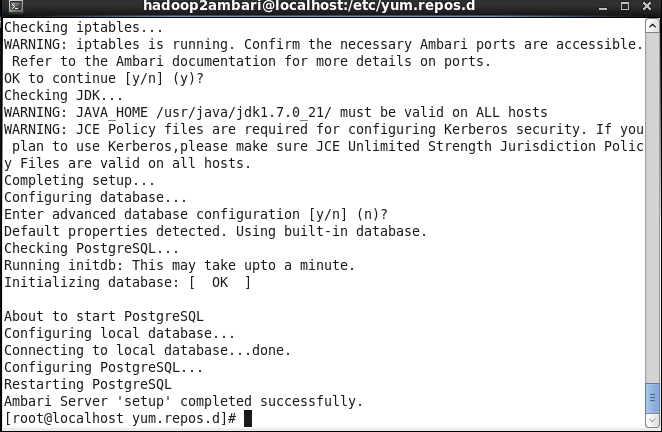
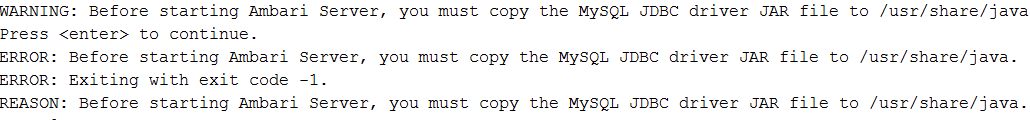
* 配置JAVA\_HOME目录地址



* 配置数据库



* 将mysql的jdbc驱动包放到对应文件夹下



server启动ambari-server：ambari-server start

@ ambari-server start

@ ps -ef | grep Ambari（查看Ambari服务是否启动）

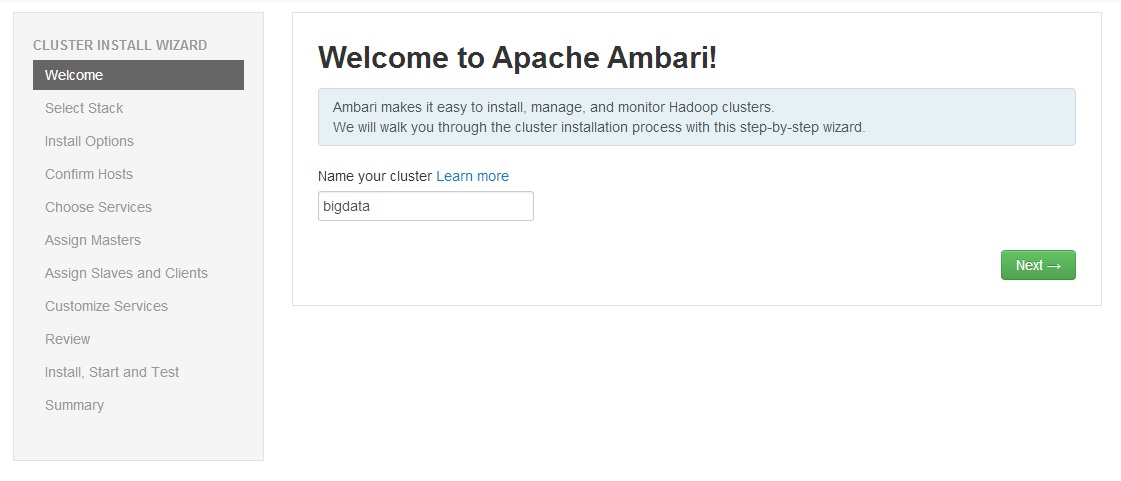
## ambari服务器配置与管理

使用浏览器输入http://hdp-m2:8080进入登陆页面如下，用户名/密码：admin/admin

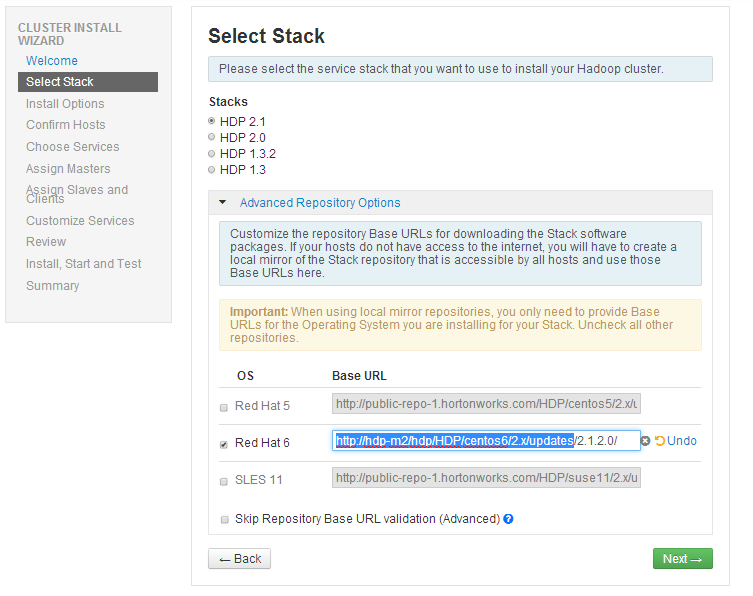
* 集群服务信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **服务 机器名** | **hdp-m1** | **hdp-m2** | **hdp-s1** | **hdp-s2** | **hdp-s3** |
| **HDFS** | Active NameNode | StandbyNameNode | DataNode | DataNode | DataNode |
| **YARN** | YarnClient、ResourceManager、App Timeline Server | YarnClient | NodeManager | NodeManager | NodeManager |
| **MapReduce2** | MapReduce2Client | HistoryServer MapReduce2Client |  |  |  |
| **Tez** | √ | √ |  |  |  |
| **HBase** | Active HBaseMaster | StandbyHBaseMaster | RegionServer | RegionServer | RegionServer |
| **Hive** | Hive Client | Hive Metastore、Hive Server2、MysqlServer、Hive Client |  |  |  |
| **WebHCat** |  | [WebHCat Serve](http://hdp-m2:8080/" \o "hdp-m2) |  |  |  |
| **Ganglia** | Ganglia Monitor | [Ganglia Server](http://hdp-m2:8080/" \o "hdp-m2) | Ganglia Monitor | Ganglia Monitor | Ganglia Monitor |
| **Nagios** |  | [Nagios Server](http://hdp-m2:8080/" \o "hdp-m2) |  |  |  |
| **ZooKeeper** | √ | √ | √ |  |  |
| **Pig** | √ | √ |  |  |  |
| **Sqoop** | √ | √ |  |  |  |
| **Oozie** | Oozie Client | Oozie Server、Oozie Client |  |  |  |
| **Falcon** |  |  |  |  |  |
| **Storm** |  |  |  |  |  |

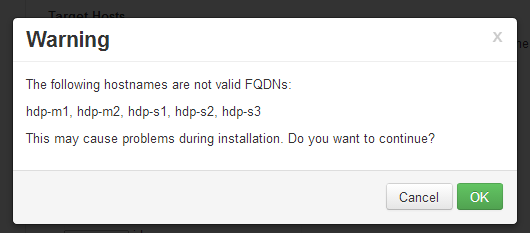
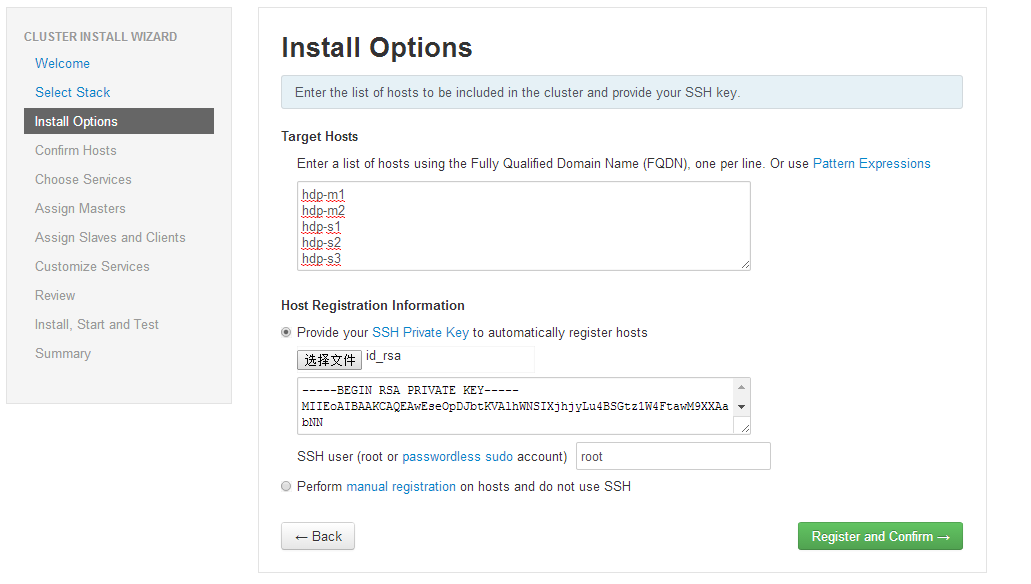
* 集群名称，随便取：



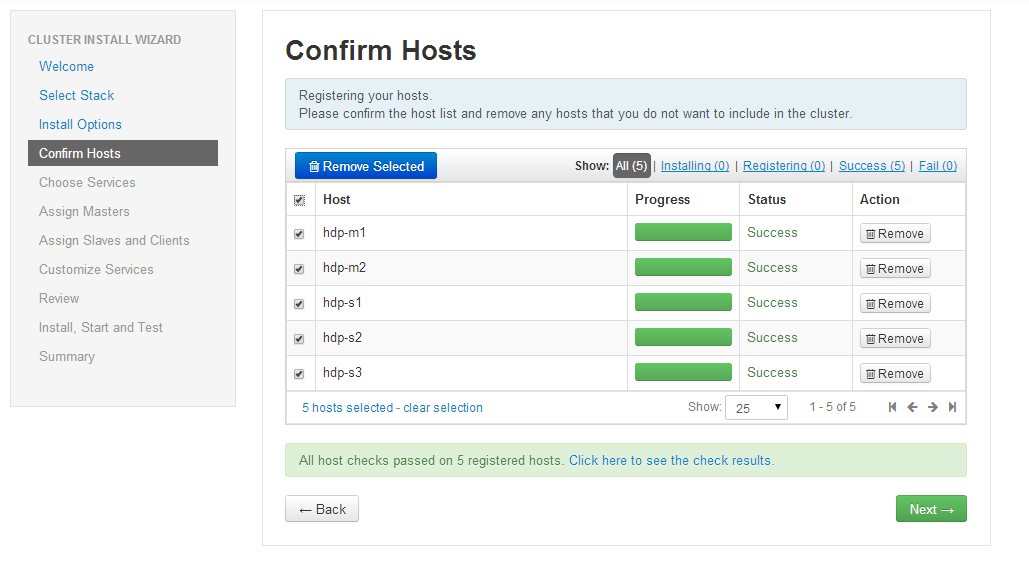
* 指定hdp版本和hdp的yum 仓库位置，centos6对应项的仓库位置配置需要修改，其他的不选：



* 节点列表和server的id\_dsa私钥文件，弹出的警告直接点OK，host最好是主机名和域名的结合，记得在/etc/hosts中增加带有域名的IP映射，id\_dsa私钥文件要讲文件内容复制到空白版中(最好选择私钥文件)：

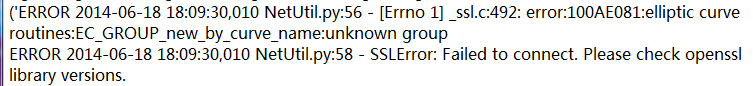


* 确认节点：



主要有2个问题：openssl和server主机无法访问的问题。

**问题一解决方案：**



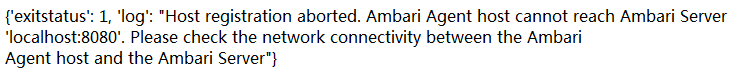
在部署ambari的时候会出现如题的错误。这个错误是在配置host的时候出现的错误。谷歌这个是说centos6.5的一个bug，不过已经修复了。如果是使用在线的yum源的话，应该是不会出现这个问题的。不过我使用的是本地源，所以就出现了这个问题，这个应该是版本兼容性不行导致的。可以下载新的openssl-1.0.1e-16.el6\_5.x86\_64文件并进行安装即可，以前的版本是openssl-1.0.1e-15.el6\_5.x86\_64， 由于centos已经含有这个软件，所以需要使用命令：

@ rpm -ivh --replacefiles openssl-\*.rpm

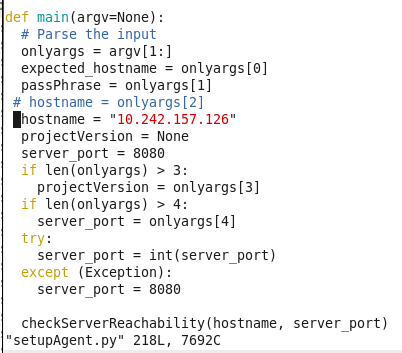
当然也可以自己下载openssl-1.0.1e-16.el6\_5.x86\_64文件进行安装，因为已经安装过openssl-1.0.1e-15.el6\_5.x86\_64，所以要先卸载：

@ rpm -i --force --nodeps openssl-1.0.1e-16.el6\_5.x86\_64.rpm

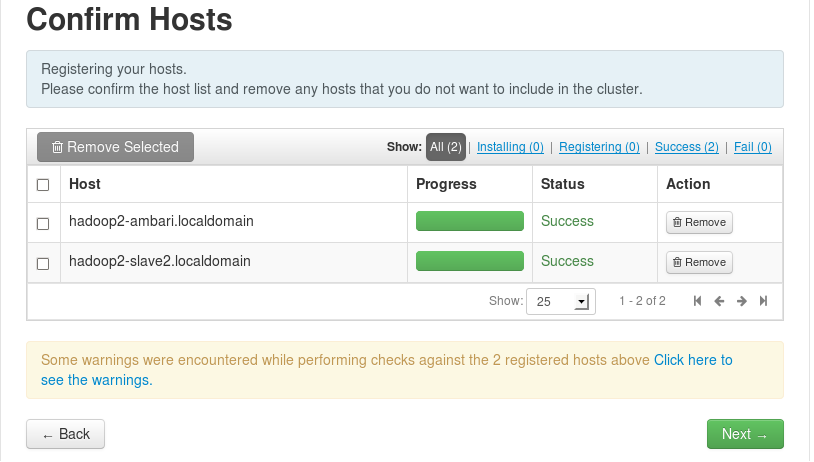
**问题二解决方案：**



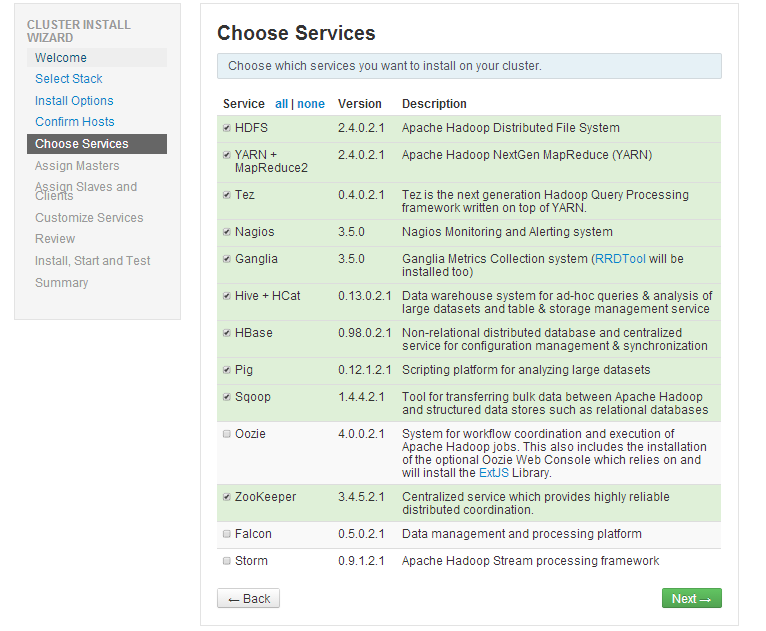
在安装完ambari－server并配置之后，选定客户端进行指定安装的时候，发现总是遇到这一错误，看了一下错误日志发现在从ambari－server传送到客户端的setupAgent.py脚本的main函数中的参数设置老是不对，如下所示：



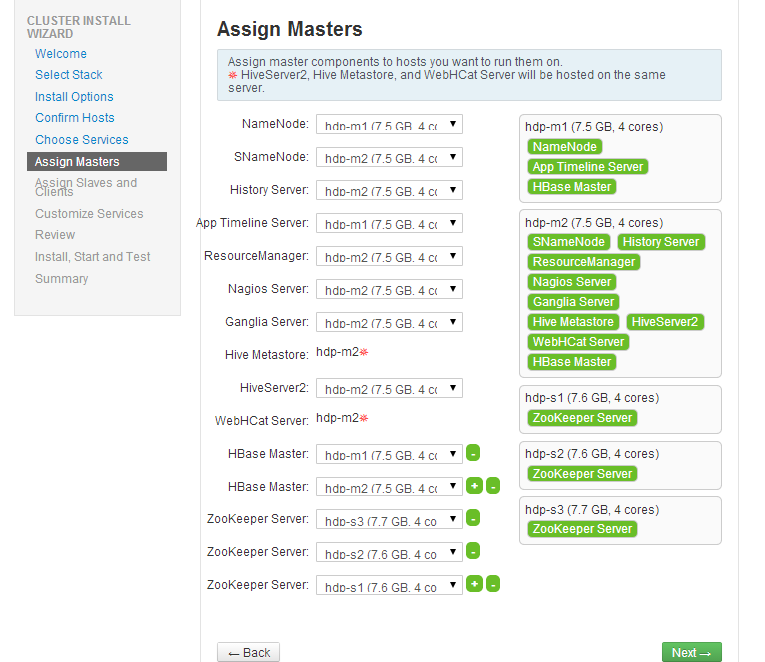
原先的代码不对，如果按照原先的代码执行就会一直出现上述问题。而手动指定了ambari－server所在机器的host和service port，修改之后执行就可以通过。



* 选择服务，全选，直接next：

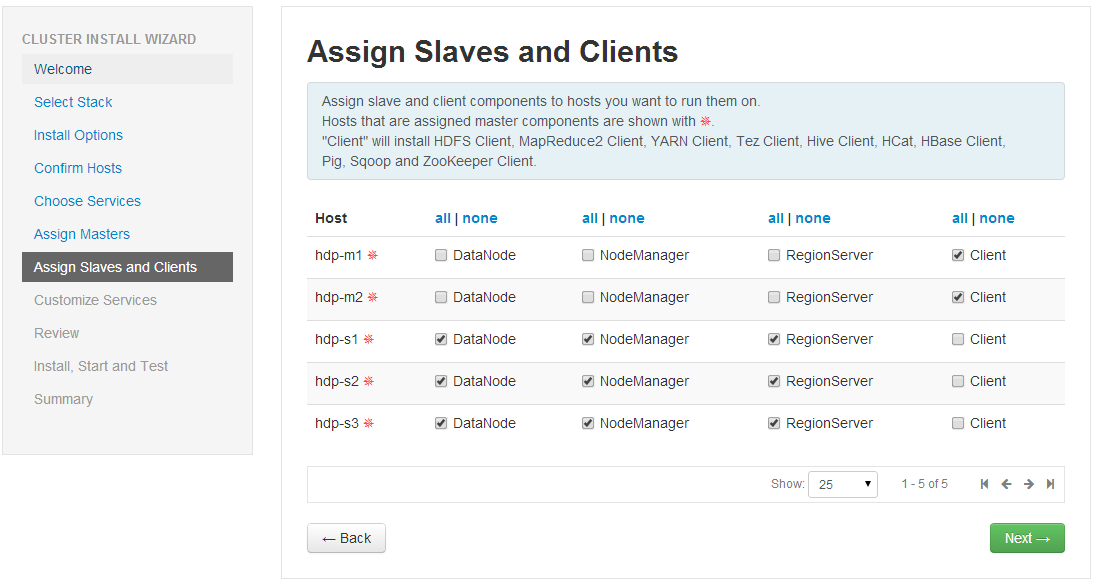


* 指定masters

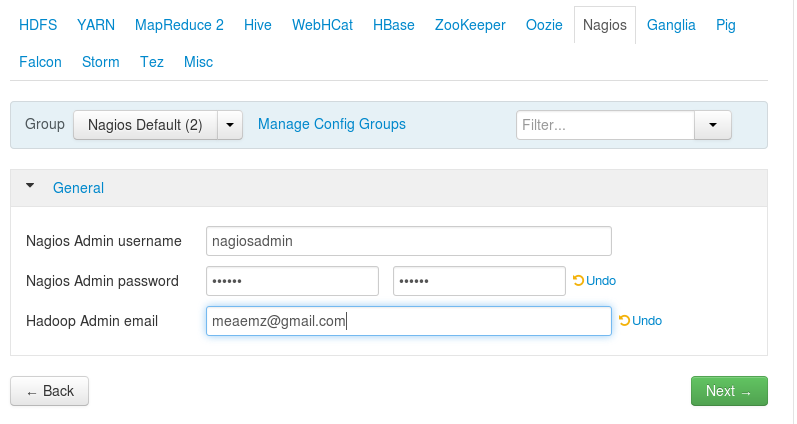
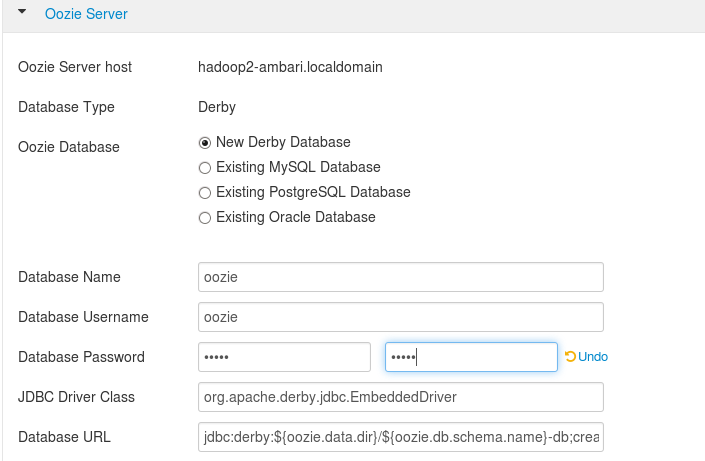
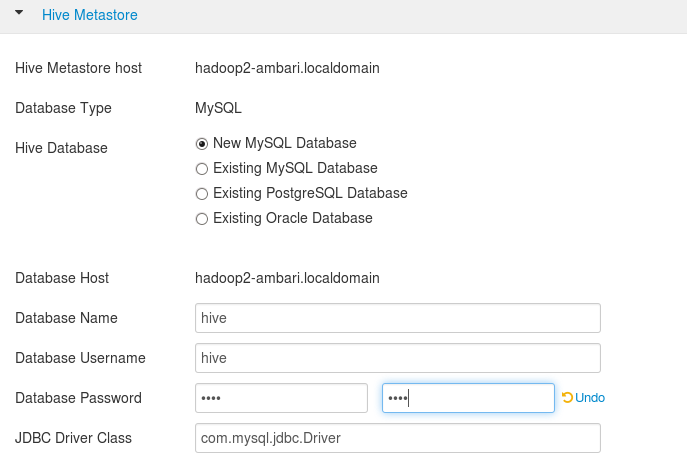


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **服务 机器名** | **Hdp-m1** | **Hdp-m2** | **Hdp-s1** | **Hdp-s2** | **Hdp-s3** |
| HDFS | Active NameNode | StandbyNameNode | DataNode | DataNode | DataNode |
| YARN | YarnClient | ResourceManager、YarnClient、App Timeline Server | NodeManager | NodeManager | NodeManager |
| MapReduce2 |  | History Server | MapReduce2Client | MapReduce2Client | MapReduce2Client |
| Tez |  |  | √ | √ | √ |
| HBase | Active HBaseMaster | Standby HBaseMaster | RegionServer | RegionServer | RegionServer |
| Hive | Hive Client | Hive Metastore、Hive Server2、MysqlServer、Hive Client |  |  |  |
| Oozie | Oozie Client | Oozie Server | Oozie Client | Oozie Client | Oozie Client |
| WebHCat | √ | √ |  |  |  |
| Falcon |  |  |  | √ | √ |
| Storm | √ | √ | √ | √ | √ |
| Ganglia | √ | √ | √ | √ | √ |
| Nagios | √ | √ | √ | √ | √ |
| ZooKeeper | √ | √ | √ |  |  |
| Pig | √ | √ |  |  |  |
| Sqoop |  |  | √ | √ | √ |

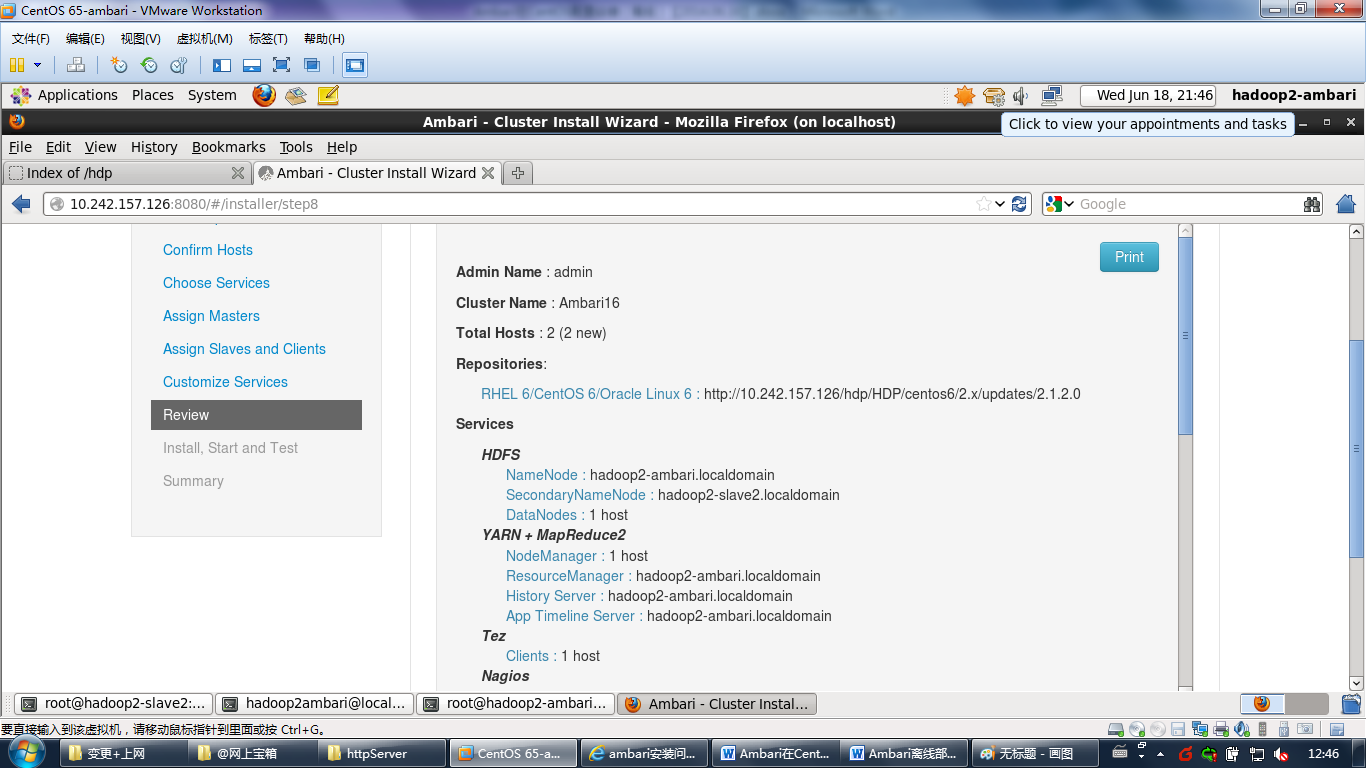
* 指定slaves和客户端



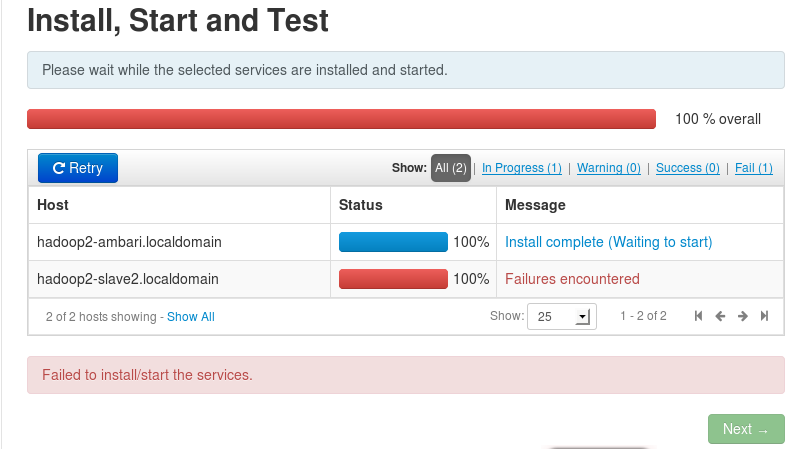
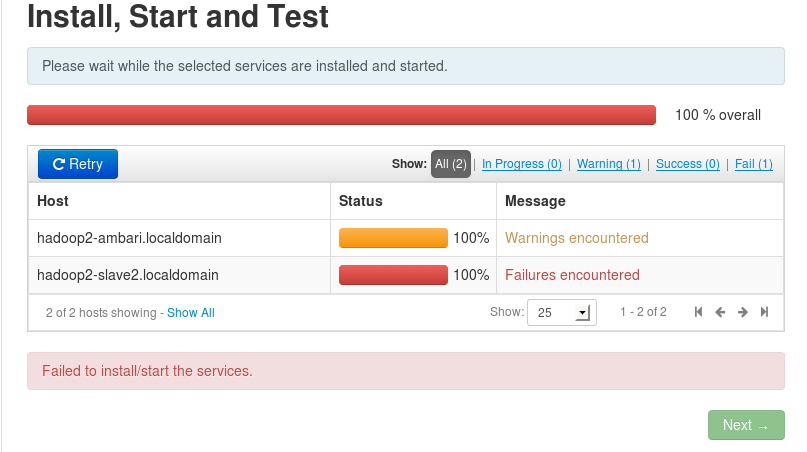
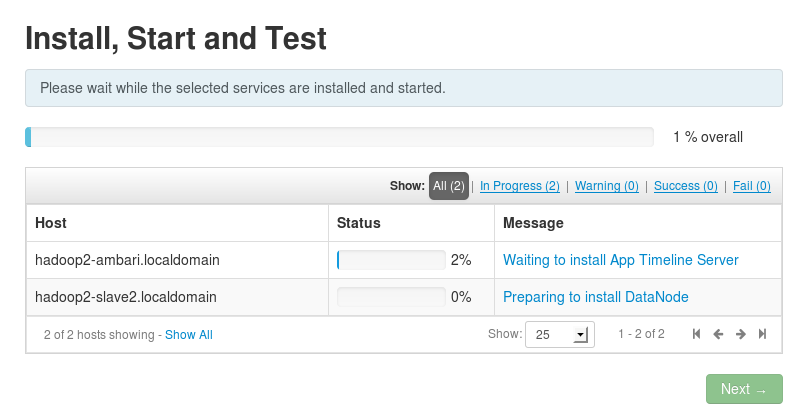
* 配置服务，将hive的数据库password配置成abcd1234，oozie的数据库password配置成abcd1234，nagios的password配置成abcd1234，email配置成meaemz@gmail.com



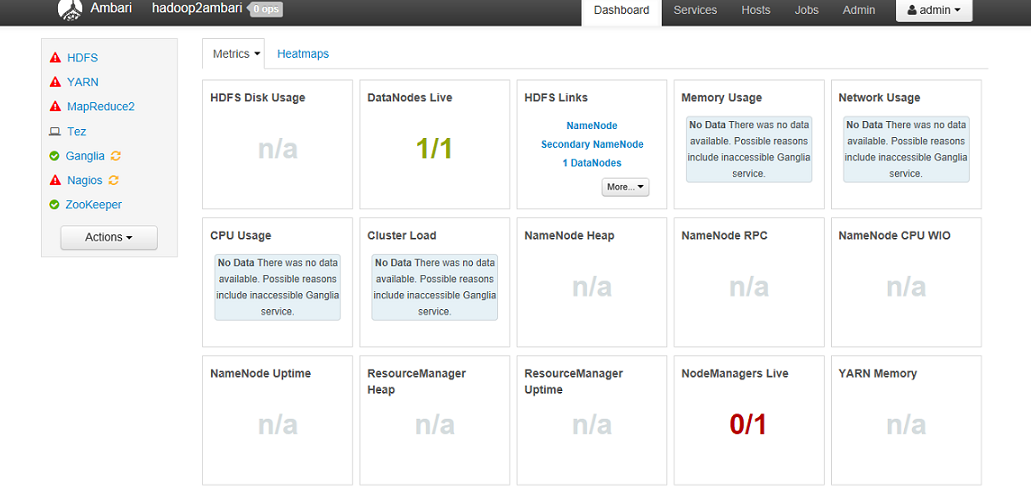
* 预览配置情况



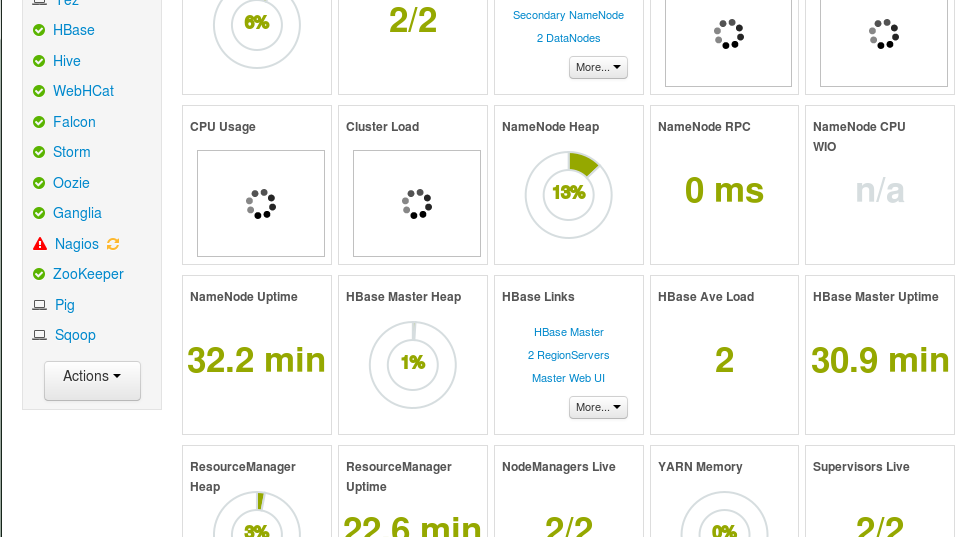
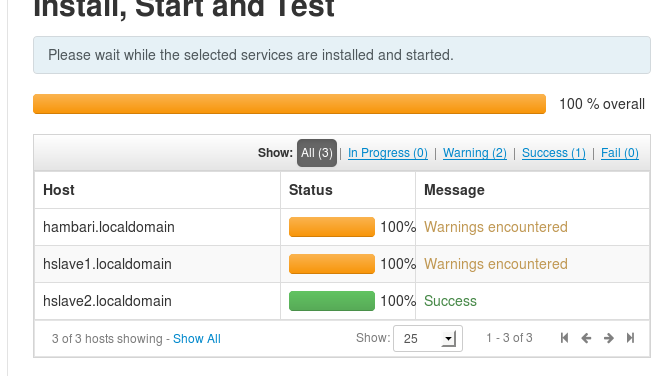
* 开始安装



* 只安装一个节点时才成功，而且不少服务没有启动：



* 以下为三个节点：



### 高可用配置

在完成NameNode配置之后，Hive需要实现高可用，对Hive中的hdfs存储路径指向进行修改。在hive服务的安装机器上进行。命令如下：

**1)检查当前的hive中的HDFS数据存储配置**

**hive --config /etc/hive/conf.server --service metatool –listFSRoot**

**结果如下：**

**Listing FS Roots..**

**hdfs://<namenode-host>/apps/hive/warehouse**

**2)修改hive的hdfs文件存储目录为namenode高可用方式：**

**$ hive --config /etc/hive/conf.server --service metatool -updateLocation**

**<new-location> <old-location>**

**如：hive --config /etc/hive/conf.server --service metatool -updateLocation hdfs://mycluster/apps/hive/warehouse hdfs://hdp-m1:8020/apps/hive/warehouse**

**3)再次检查修改结果：**

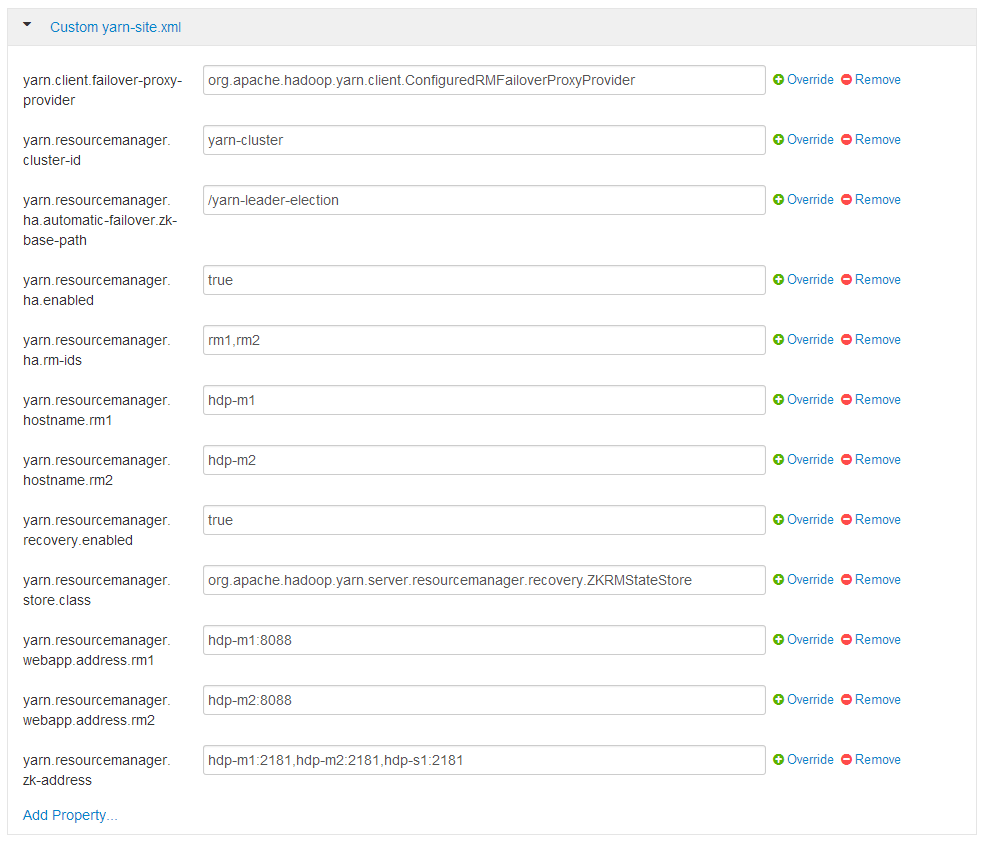
**hive --config /etc/hive/conf.server --service metatool –listFSRoot**

### Resource Manager高可用配置

Hortonworks2.1版本需要手动配置Resource Manager高可用，参考bk\_system-admin-g

uide-xxx.pdf

1. 修改yarn-site.xml文件：



<!-- RM HA Configurations-->

<property>

<name>yarn.resourcemanager.cluster-id</name>

<value>yarn-cluster</value>

</property>

<property>

<name>yarn.resourcemanager.ha.enabled</name>

<value>true</value>

</property>

<property>

<name>yarn.resourcemanager.ha.rm-ids</name>

<value>rm1,rm2</value>

</property>

<property>

<name>yarn.resourcemanager.hostname.rm1</name>

<value>${rm1 address}</value>

</property>

<property>

<name>yarn.resourcemanager.hostname.rm2</name>

<value>${rm2 address}</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address.rm1</name>

<value>rm1\_web\_address:port\_num</value>

<description>We can set rm1\_web\_address separately. If not, it will use

${yarn.resourcemanager.hostname.rm1}:DEFAULT\_RM\_WEBAPP\_PORT</description>

</property>

<property>

<name>yarn.resourcemanager.webapp.address.rm2</name>

<value>rm2\_web\_address:port\_num</value>

</property>

<property>

<name>yarn.resourcemanager.recovery.enabled</name>

<value>true</value>

</property>

<property>

<name>yarn.resourcemanager.store.class</name>

<value>org.apache.hadoop.yarn.server.resourcemanager.recovery.

ZKRMStateStore</value>

</property>

<property>

<name>yarn.resourcemanager.zk-address</name>

<value>${zk1.address,zk2.address}</value>

</property>

<property>

<name>yarn.client.failover-proxy-provider</name>

<value>org.apache.hadoop.yarn.client.ConfiguredRMFailoverProxyProvider</

value>

</property>

<property>

<name>yarn.resourcemanager.ha.automatic-failover.zk-base-path</name>

<value>/yarn-leader-election</value>

<description>Optional setting. The default value is /yarn-leader-election</

description>

</property>

1. **重启zookeeper和resourcemanager**
2. **Ambari Server界面只能启动默认安装的，其它的需手动启动：**

su - yarn -c "export HADOOP\_LIBEXEC\_DIR=/usr/lib/hadoop/libexec && /usr/lib/hadoop-yarn/sbin/yarn-daemon.sh --config /etc/hadoop/conf start resourcemanager"

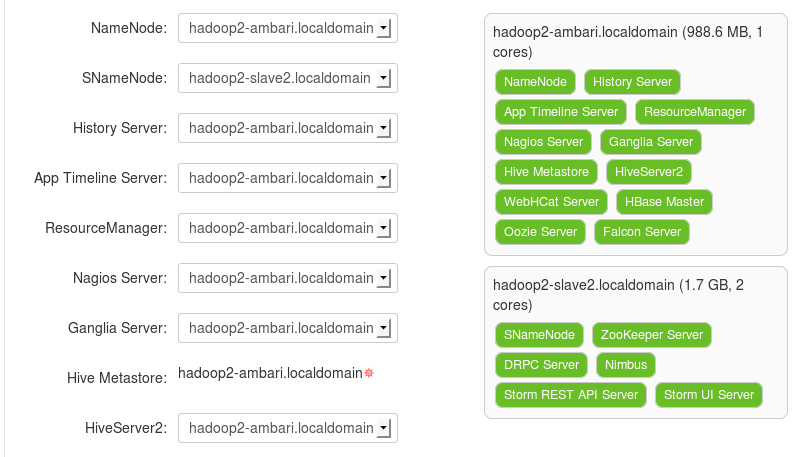
1. **切换状态：**

yarn rmadmin -transitionToActive m1

yarn rmadmin -transitionToStandby m2

### HBase HA配置

在选择服务所在节点的页面，可以增加HBase Master的节点数，就行Zookeeper一样增加节点即可，如下图：



### 增加新节点

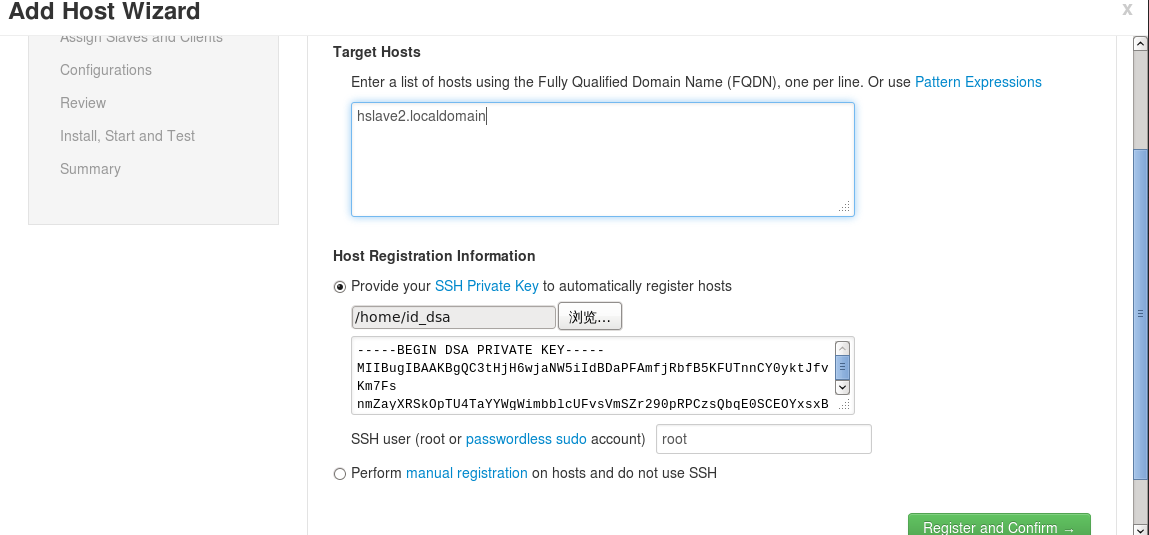
* yum remove hcatalog\\*
* yum remove hive\\*
* yum remove hbase\\*
* yum remove zookeeper\\*
* yum remove oozie\\*
* yum remove pig\\*
* yum remove snappy\\*
* yum remove hadoop-lzo\\*
* yum remove knox\\*
* yum remove hadoop\\*
* yum remove extjs-2.2-1 mysql-connector-java-5.0.8-1\\*

如果之前有安装过，则执行以上命令

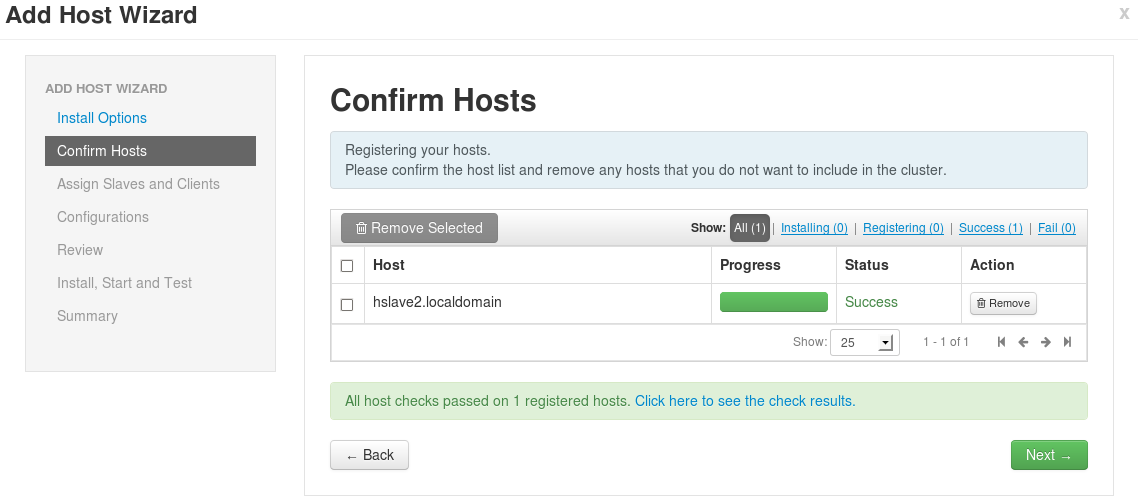
* 将其他正常节点下的/etc/hadoop/conf/\* 拷贝到新节点的/etc/hadoop/conf下
* 在Web管理页面进入Hosts->Add New Hosts：



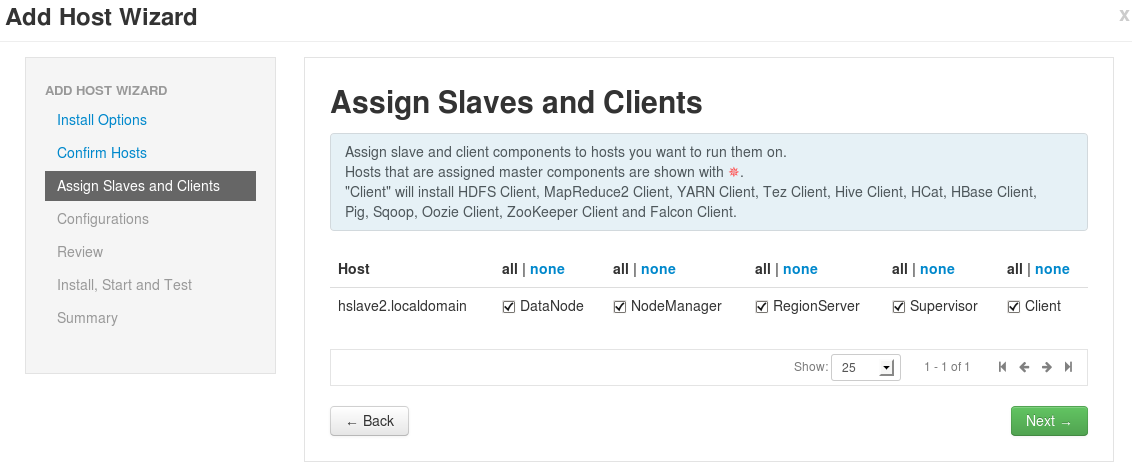
* 进入新增节点界面：



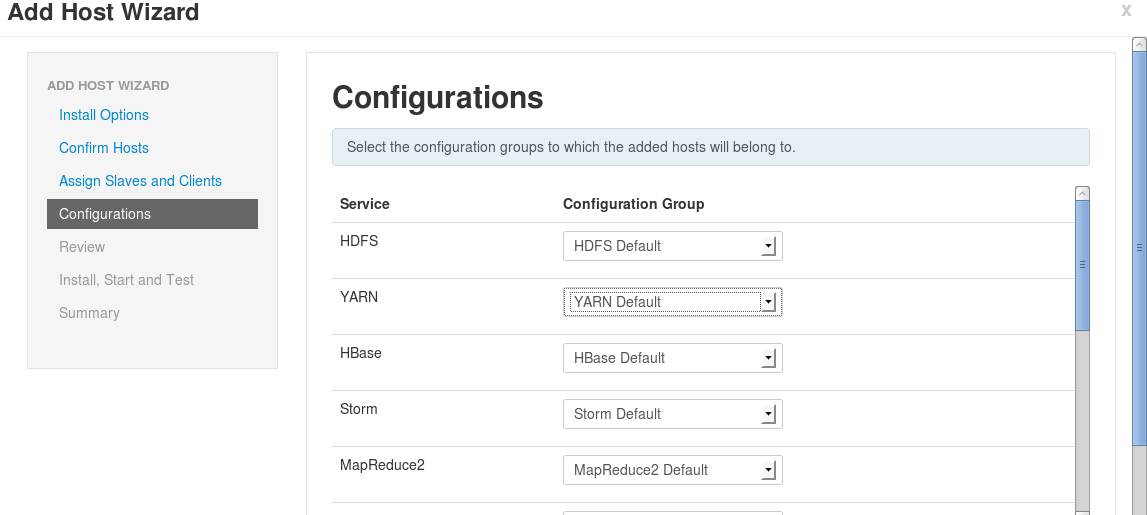
* 注册新节点：



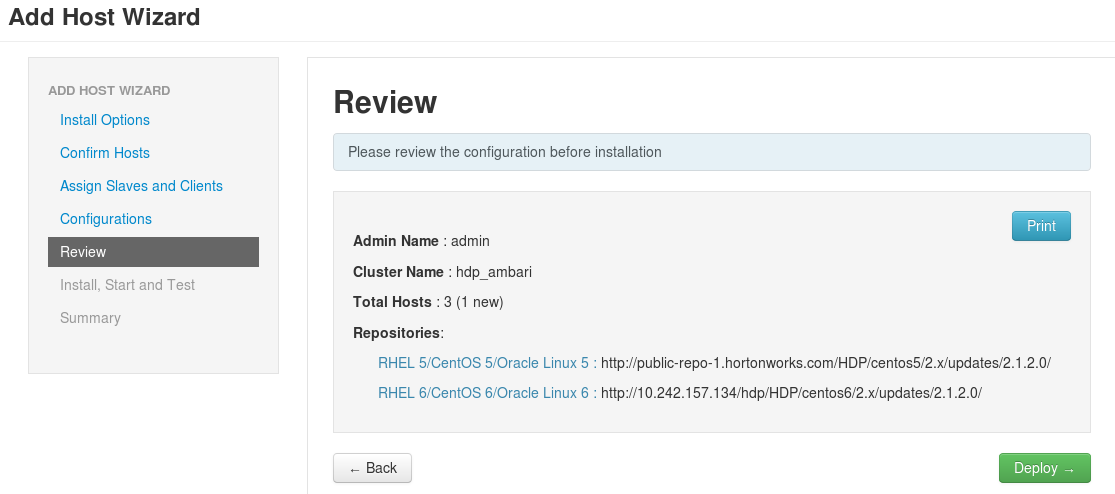
* 确认为slavve节点：



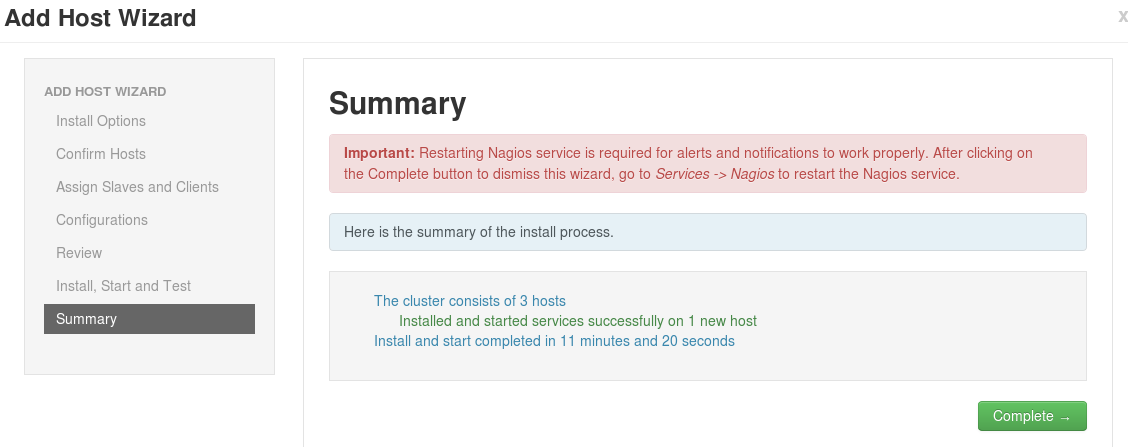
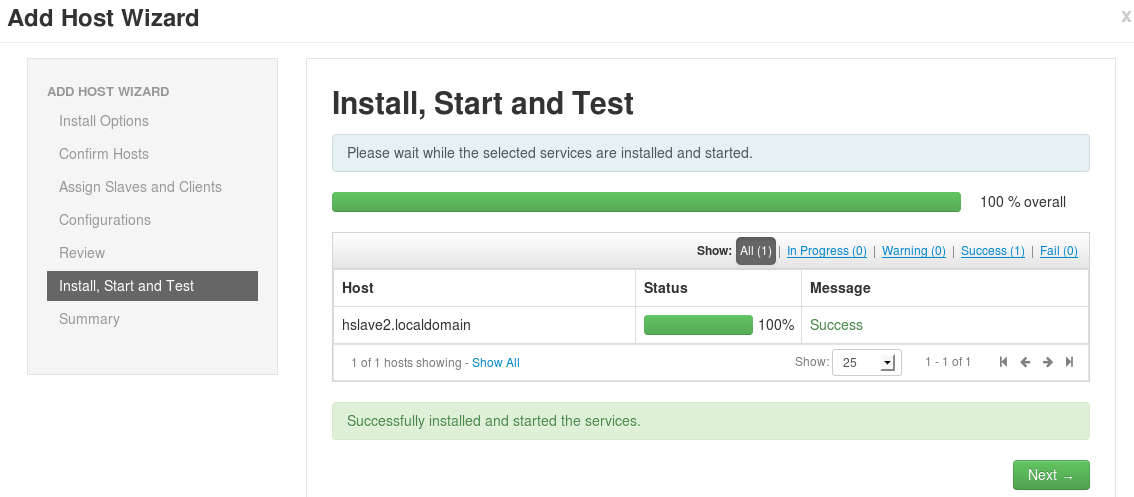
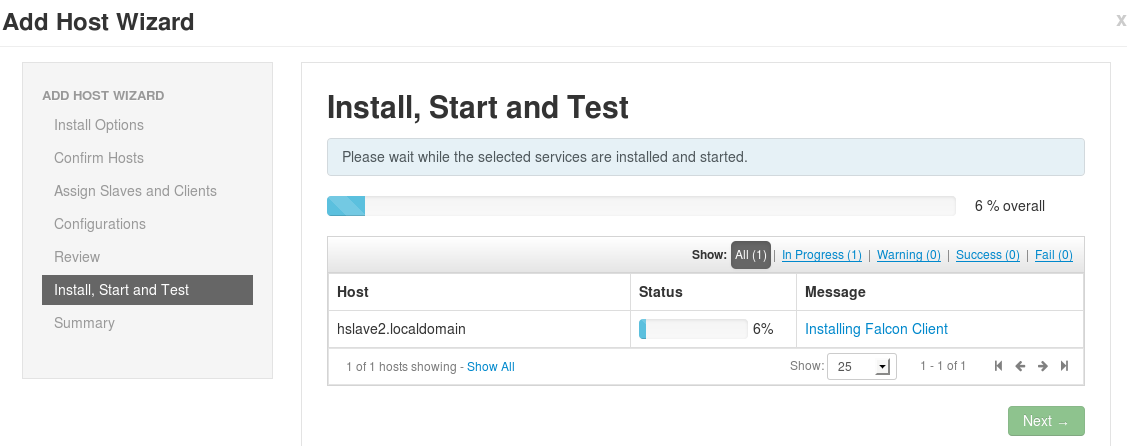
* 配置服务：



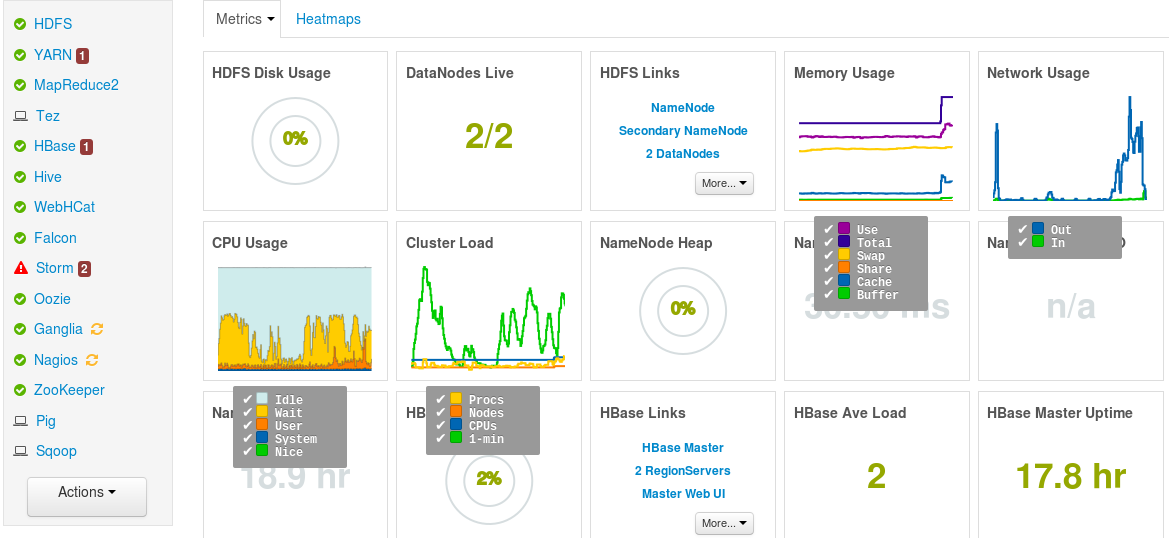
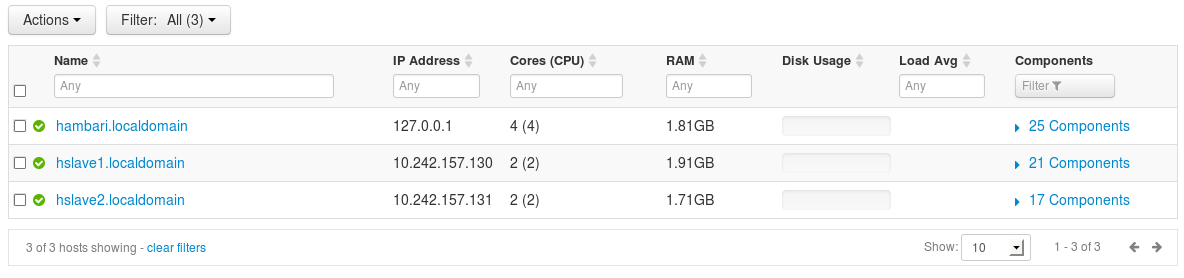
* 查看配置：



* 安装与完成：



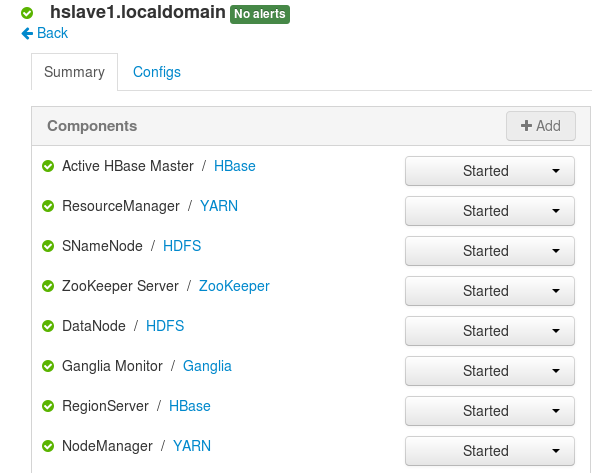
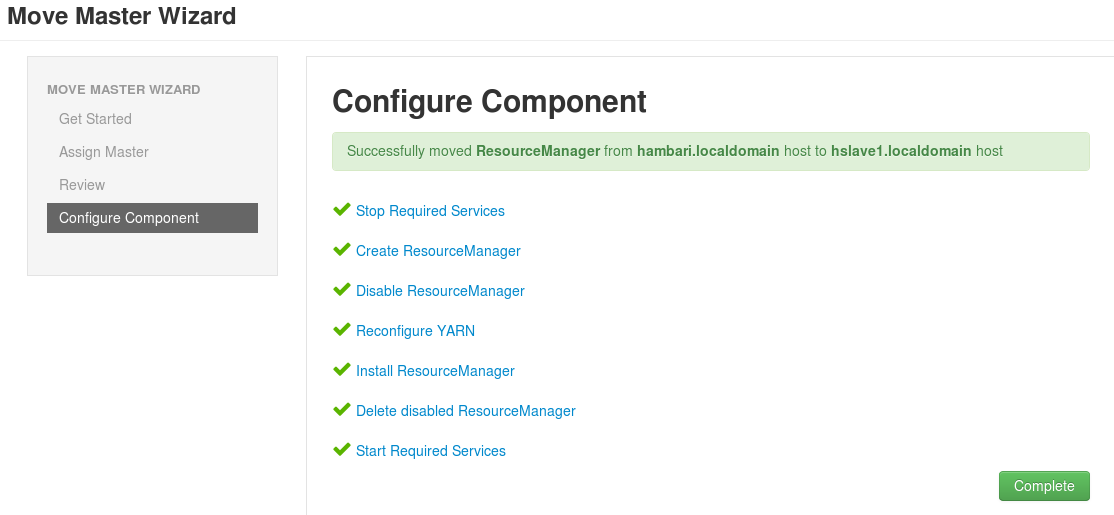
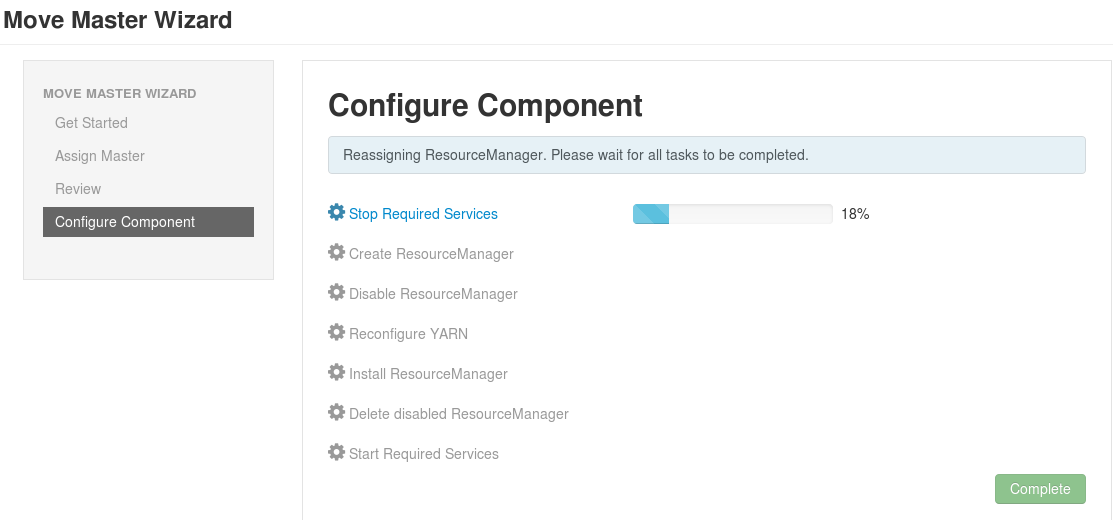
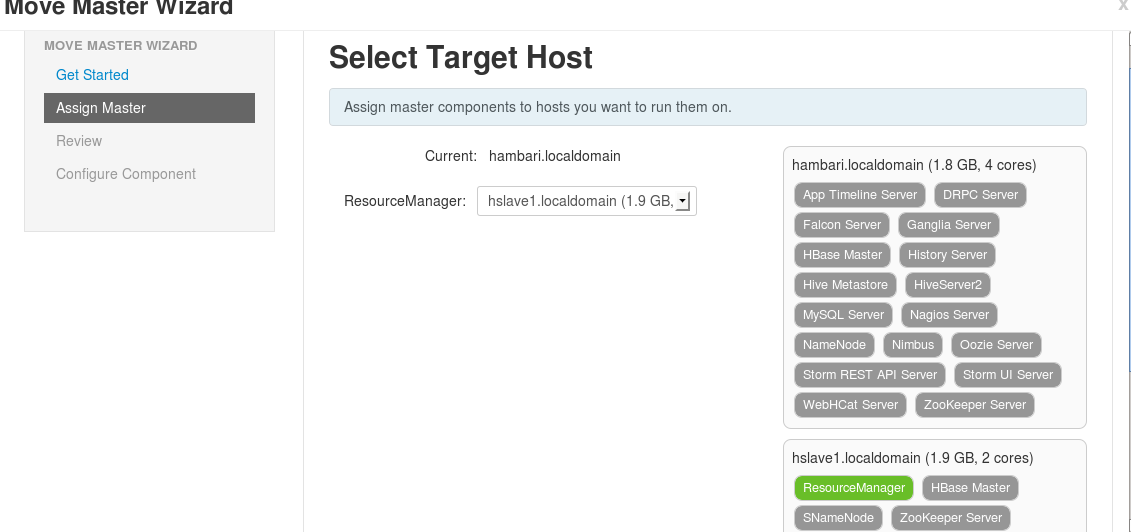
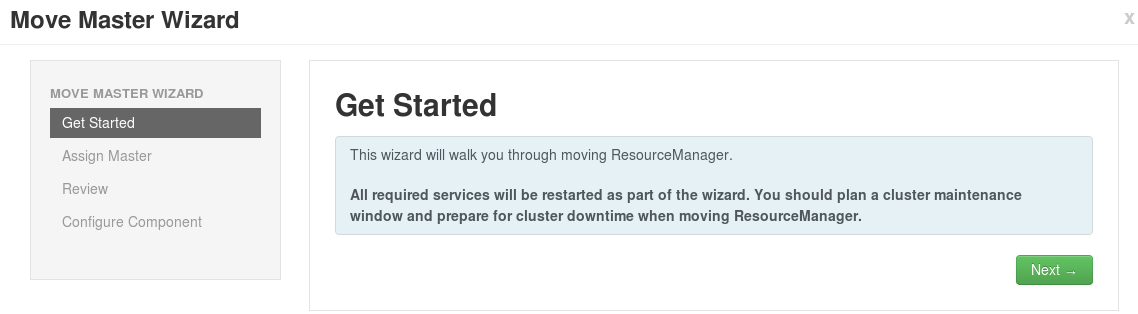
* Web页面上查看新节点：



### 服务迁移到另一节点

服务迁移是将主服务迁移到其他节点的功能，只支持NameNode、SNameNode和ResourceManager的迁移。在此，以ResourceManager为例：

在Services -> Yarn -> Service Actions -> Move ResourceManager



### Ambari Server移动

将ambari Server移动到其它机器，命令如下：

1. **停止ambari server：**

**ambari-server stop**

1. **创建备份目录：**

**mkdir /tmp/dbdumps**

**cd dbdumps**

1. **备份postgres数据库：**

**pg\_dump -U ambari ambari > ambari.sql Password:bigdata**

**pg\_dump -U mapred ambarirca > ambarirca.sql Password:mapred**

**ambari默认账号密码：ambari-server/bigdata**

**ambarirca默认账号密码：mapred/mapredss**

1. **停止所有ambari agent服务：**

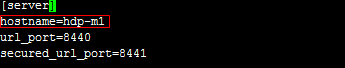
**ambari-agent stop**

1. **删除所有机器上旧的代理证书：**

**rm /var/lib/ambari-agent/keys/\***

1. **修改所有代理机器上的/etc/ambari-agent/conf/ambari-agent.ini文件：**

**将hostname值修改成新的ambari server的机器名**



1. **下面所有操作都在新的ambari server机器上执行**
2. **如果新机器上没有ambari server则先安装：**

**参考3.1.6**

1. **停止新的ambari server,以便复制旧的数据库到新的服务器：**

**Ambari-server stop**

1. **重启postgre数据库：**

**service postgresql restart**

1. **打开PostgreSQL交互式终端：**

**su - postgres**

**psql**

1. **删除原有的数据库，创建新的数据库：**

**drop database ambari;**

**drop database ambarirca;**

**create database ambari;**

**create database ambarirca;**

**退出交互终端：\q**

1. **将postgres备份文件copy至新的ambari server机器上并执行：**

**scp /tmp/dbdumps/\*.sql @hdp-m1:/tmp/dbdumps/**

**su - postgres**

**psql -d ambari -f /tmp/dbdumps/ambari.sql**

**psql -d ambarirca -f /tmp/dbdumps/ambarirca.sql**

1. **启动ambari server：**

**ambari-server start**

1. **启动所有ambari agent服务：**

**ambari-agent start**

1. **默认的web端口是8080**
2. **卸载原有的ambari Server**

**yum remove ambari-server**

### Ambari 组件卸载

1. **在控制台停止服务**
2. **备份postgres数据库：**

**pg\_dump -U ambari ambari > ambari.sql Password:bigdata**

**pg\_dump -U mapred ambarirca > ambarirca.sql Password:mapred**

**ambari默认账号密码：ambari-server/bigdata**

**ambarirca默认账号密码：mapred/mapredss**

1. **删除ambari脚本(将#service\_name#改成如：OOZIE)：**

delete from servicedesiredstate where service\_name = '#service\_name#';

delete from clusterconfig where upper(type\_name) like '%#service\_name#%';

delete from clusterconfigmapping where upper(type\_name) like '%#service\_name#%';

delete from hostcomponentdesiredstate where service\_name = '#service\_name#';

delete from hostcomponentstate where service\_name = '#service\_name#';

delete from servicecomponentdesiredstate where service\_name = '#service\_name#';

delete from clusterservices where service\_name = '#service\_name#';

delete from execution\_command where task\_id in (select task\_id from host\_role\_command where upper(role) like '%#service\_name#%');

delete from host\_role\_command where upper(role) like '%#service\_name#%';

delete from requestresourcefilter where service\_name = '#service\_name#';

commit;

1. **卸载组件(如：oozie)**

**yum remove oozie\\***

1. **重启ambari server和ambari agents服务：**

**ambari-server restart**

**ambari-agent restart**

### 安装ambari客户端

首先确保hdp-m2机器上的/etc/yum.repos.d/ambari.repo文件上传到了所有机器上，在所有从节点上运行如下命令：

@ yum install ambari-agent

　　这个步骤要等一会，它需要上网下载，6M左右，然后启动客户端服务：

@ ambari-agent start

@ ps -ef | grep Ambari（查看Ambari服务是否启动）

# HUE手动安装

参考（bk\_installing\_manually\_bookxxx.pdf）文件

1. 修改hdfs-site.xml文件：

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

1. 修改core-site.xml文件：

<property>

<name>hadoop.proxyuser.hue.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.hue.groups</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.hcat.groups</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.hcat.hosts</name>

<value>\*</value>

</property>

1. 修改webhcat-site.xml文件：

<property>

<name>webhcat.proxyuser.hue.hosts</name>

<value>\*</value>

</property>

<property>

<name>webhcat.proxyuser.hue.groups</name>

<value>\*</value>

</property>

1. 修改oozie-site.xml文件：

<property>

<name>oozie.service.ProxyUserService.proxyuser.hue.hosts</name>

<value>\*</value>

</property>

<property>

<name>oozie.service.ProxyUserService.proxyuser.hue.groups</name>

<value>\*</value>

</property>

1. 修改文件后重启相关联的服务
2. 安装hue:

**yum install hue**

1. 修改/etc/hue/hue.ini文件，把相关联的属性值改成实际环境的
2. 启动命令：

/etc/init.d/hue start

/etc/init.d/hue stop

/etc/init.d/hue restart

# Phoenix手动安装

1. 安装Phoenix(RHEL/CentOS/Oracle Linux):

yum install phoenix

1. 将phoenix-core-xx.jar拷贝到所有的Hbase节点:

scp /usr/lib/phoenix/lib/phoenix-core-4.0.0.2.1.1.0-365.jar **@hostName**:/usr/lib/

hbase/lib/

1. 修改hbase-site.xml文件:

<property>

<name>hbase.defaults.for.version.skip</name>

<value>true</value>

</property>

<property>

<name>hbase.regionserver.wal.codec</name>

<value>org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec</value>

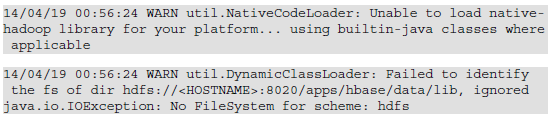
</property>

1. 将core-site.xml和hbase-site.xml拷贝到**<PHOENIX\_HOME>/bin**目录：

ln -sf /etc/hadoop/conf/core-site.xml /usr/lib/phoenix/bin/

ln -sf /etc/hbase/conf/hbase-site.xml /usr/lib/phoenix/bin/

1. 出现以下错误可忽略：



1. 运行测试脚本

/usr/lib/phoenix/bin/psql.py hdp-s1:2181 /usr/share/doc/phoenix-4.0.0.2.1.2.0/examples/WEB\_STAT.sql /usr/share/doc/phoenix-4.0.0.2.1.2.0/examples/WEB\_STAT.csv /usr/share/doc/phoenix-4.0.0.2.1.2.0/examples/WEB\_STAT\_QUERIES.sql

1. 重启hbase
2. 进入控制台：

/usr/lib/phoenix/bin/sqlline.py hdp-s1:2181

# 常见问题

## mapreduce

在hadoop中执行mapreduce，访问Hbase中的表时，regions会报错。

此错误在Hbase 0.96版本以上才会出现

报错描述：

Exception in thread "main" java.lang.IllegalAccessError: class

com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass

com.google.protobuf.LiteralByteString

at java.lang.ClassLoader.defineClass1(Native Method)

at java.lang.ClassLoader.defineClass(ClassLoader.java:792)

at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)

at java.net.URLClassLoader.defineClass(URLClassLoader.java:449)

at java.net.URLClassLoader.access$100(URLClassLoader.java:71)

at java.net.URLClassLoader$1.run(URLClassLoader.java:361)

at java.net.URLClassLoader$1.run(URLClassLoader.java:355)

at java.security.AccessController.doPrivileged(Native Method)

at java.net.URLClassLoader.findClass(URLClassLoader.java:354)

at java.lang.ClassLoader.loadClass(ClassLoader.java:424)

at java.lang.ClassLoader.loadClass(ClassLoader.java:357)

at

org.apache.hadoop.hbase.protobuf.ProtobufUtil.toScan(ProtobufUtil.java:818)

at

问题解决：

在所有master节点上添加如下环境变量

HADOOP\_CLASSPATH=$HADOOP\_CLASSPATH:/use/lib/hbase/hbase-protocol-xxx-hadoop2.jar

## oozie安装

问题描述：

启动oozie服务端的时候报SQLException：

问题分析：

Oozie必须我们手动创建用户和数据库

问题解决：

--创建用户，做两遍

insert into user(host,user,password) values('localhost','oozie',PASSWORD('oozie'));

insert into user(host,user,password) values('hdp-m2','oozie',PASSWORD('oozie'));

flush privileges;

--创建数据库

create database oozie;

--授权，做两遍·

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

grant all privileges on oozie.\* to oozie@'hdp-m2' identified by 'oozie';

flush privileges;

## 启动异常

问题描述：服务启动出错

问题解决：

将相关的进程服务手动kill掉

## Oozie启动

访问oozieweb页面时会出现权限文件。默认会创建oozie用户来安装，而我们使用的是hadoop用户启动所以没有权限访问。

报错描述：

org.apache.jasper.JasperException: java.lang.IllegalStateException: No output folder

问题解决：使用root用户将/var/lib/oozie文件夹下所有文件授予hadoop用户读写权限

## Hive使用JDBC连接

1. Hive1 Server使用以下配置：

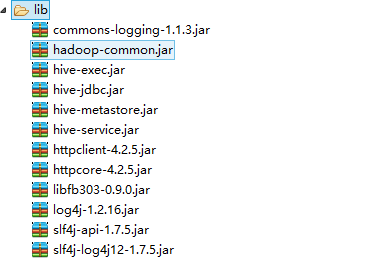
Class.forName("org.apache.hadoop.hive.jdbc.HiveDriver");

Connection conn = DriverManager.getConnection("jdbc:hive://hdp-m2:10000/default", "hadoop", "hadoop");

1. Hive2 Server使用以下配置：

Class.forName("org.apache.hive.jdbc.HiveDriver");

Connection conn = DriverManager.getConnection("jdbc:hive2://hdp-m2:10000/default", "hadoop", "hadoop");



## 启动ZKFailoverController错误

问题描述：服务启动出错

Exception in thread "main" java.lang.IllegalArgumentException: Missing required configuration 'ha.zookeeper.quorum' for ZooKeeper quorum

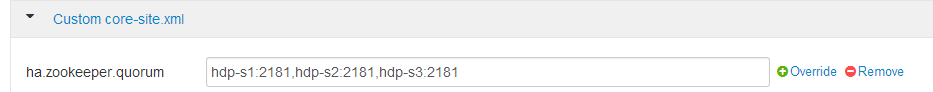
at com.google.common.base.Preconditions.checkArgument(Preconditions.java:115)

at org.apache.hadoop.ha.ZKFailoverController.initZK(ZKFailoverController.java:334)

at org.apache.hadoop.ha.ZKFailoverController.doRun(ZKFailoverController.java:185)

at org.apache.hadoop.ha.ZKFailoverController.access$000(ZKFailoverController.java:59)

问题解决,修改core-site.xml文件：



## 启动HMaster错误

HMaster自动挂掉的问题，并且master的log里出现“TableExistsException: hbase:namespace”字样，很可能是更换了Hbase的版本过后zookeeper还保留着上一次的Hbase设置，所以造成了冲突。

解决方案：

1.切换到zookeeper的bin目录；

2.执行$sh zkCli.sh

输入‘ls /’

4.输入‘rmr /hbase’

5.退出

重启hbase即可。

## hortonworks安装hive报mysql错误

可能是因为那台机器上安装了跟本地源不一致的mysql版本

解决：将安装的mysql版本替换iso本地源中的mysql版本，并重新创建createrepo

## 解决mysql冲突

服务器默认安装版本是5.1，实际安装版本是5.3，导致hortonworks在安装过程中产生冲突导致报错，服务器执行以下命令卸载原先的mysql相关lib文件

yum remove mysql-libs-5.1.66-2.el6\_3.x86\_64

## 缺少mysql lib文件

安装mysql-libs-5.1.66-2.el6\_3.x86\_64.rpm 以及 openssl-1.0.1e-16.el6\_5.x86\_64.rpm文件

nagios-plugins-1.4.9-1.x86\_64 (HDP-UTILS-1.1.0.17)

Requires: libmysqlclient.so.16(libmysqlclient\_16)(64bit)

Error: Package: nagios-plugins-1.4.9-1.x86\_64 (HDP-UTILS-1.1.0.17)

Requires: libssl.so.10(libssl.so.10)(64bit)

Error: Package: nagios-plugins-1.4.9-1.x86\_64 (HDP-UTILS-1.1.0.17)

Requires: libcrypto.so.10(libcrypto.so.10)(64bit)

Error: Package: nagios-plugins-1.4.9-1.x86\_64 (HDP-UTILS-1.1.0.17)

Requires: libmysqlclient.so.16()(64bit)