

目录

目录

[目录 2](#_Toc505346009)

[1. 项目介绍 3](#_Toc505346010)

[2. 技术架构 3](#_Toc505346011)

[3. 集群部署架构 4](#_Toc505346012)

[4. 单机版部署 6](#_Toc505346013)

[4.1 前期准备 6](#_Toc505346014)

[4.2 一键部署 6](#_Toc505346015)

[4.3 环境测试 7](#_Toc505346016)

[5. 分布式集群部署 7](#_Toc505346017)

[5.1 前期准备 7](#_Toc505346018)

[**5.1.1主机配置要求** 7](#_Toc505346019)

[**5.1.2 组件安装规划及准备** 8](#_Toc505346020)

[5.2 选择部署方式 9](#_Toc505346021)

[5.3集群部署 9](#_Toc505346022)

[**5.3.1 一键部署** 9](#_Toc505346023)

[**5.3.2 独立部署** 12](#_Toc505346024)

[**5.3.3 集成已有的Kafka和Hadoop集群** 27](#_Toc505346025)

[5.4 分布式集群测试 31](#_Toc505346026)

[6 集群管理 33](#_Toc505346027)

[6.1 启动集群 33](#_Toc505346028)

[6.2 更新服务 34](#_Toc505346029)

[6.3 增删Ambari-Agent 37](#_Toc505346030)

[**6.3.1 增加Agent** 37](#_Toc505346031)

[**6.3.2 迁移服务** 38](#_Toc505346032)

[6.4 切换HDFS 40](#_Toc505346033)

[7. 附录： 42](#_Toc505346034)

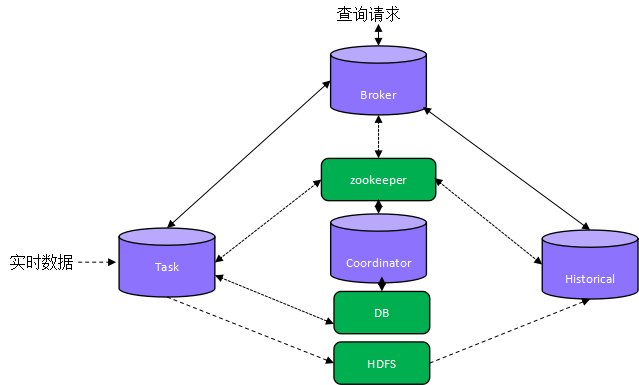
[7.1 ASTRO配置说明 42](#_Toc505346035)

**1. 项目介绍**

数果Tindex平台是数果智能自主研发的产品，用于解决大数据行业中实现大规模（PB级）行为数据实时接入、高效存储和快速查询，满足数据分析和数据挖掘等领域的需求。

1. **技术架构**

数果Tindex平台核心组件包含Broker, Coordinator, MiddleManager, Overlord, Historical，整体架构如如图1所示：



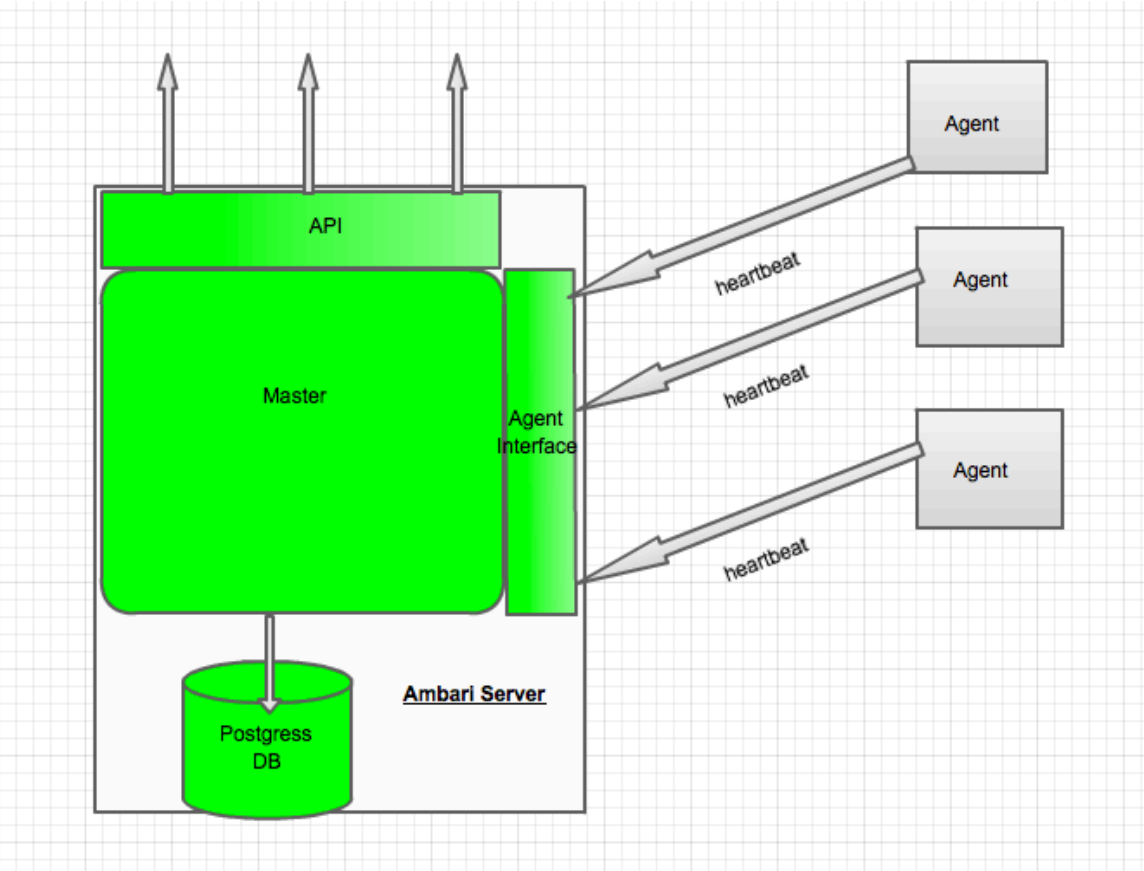
**图1 Tindex技术架构图**

* Broker节点：接受客户端请求，分发至各历史/实时节点，然后汇总返回客户端的节点
* Coordinator节点：用于在平衡各历史节点负载的节点。
* Overlord节点：用于调度实时/离线数据接入任务的节点。
* MiddleManager节点：用于启动实时/离线数据接入任务的节点。
* Historical节点：用于加载并提供历史数据查询功能的节点。

**3. 集群部署架构**

分布式集群借助Ambari进行部署，并利用Amabri对集群进行统一管理和监控，技术架构中的所有组件均通过Ambari进行部署、监控和管理，这里以3节点为例，介绍分布式集群的部署架构。

数果智能将Ambari汉化及添加自研组件后，改名为Alaska，整体框架基本不变，所以Alaska在此文档中仍称为Amabri，Ambari基本架构如图2所示

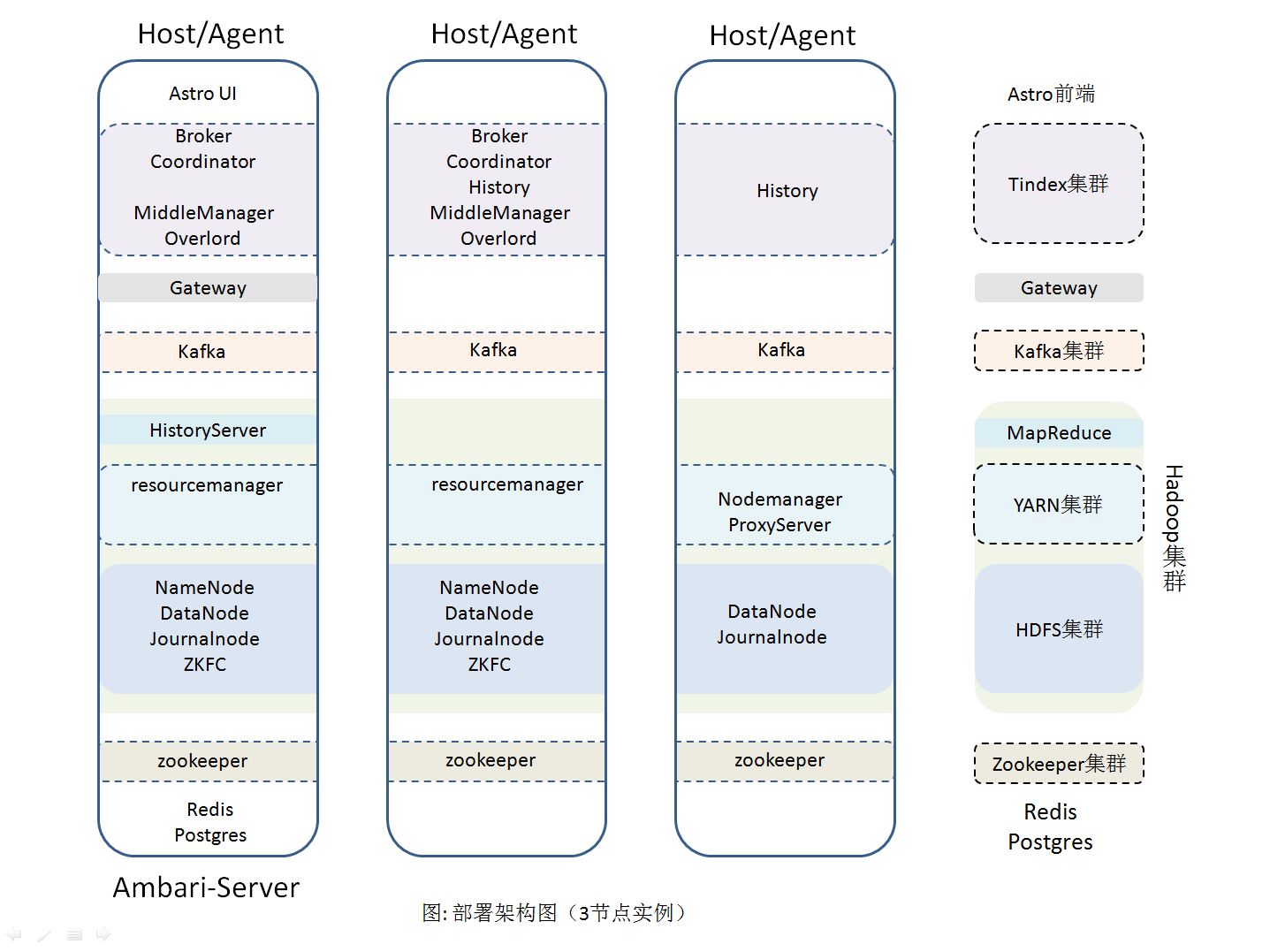


**图2 Ambari架构图**

Apache Ambari是一种基于Web的工具，支持Apache Hadoop集群的供应、管理和监控。Ambari目前已支持大多数Hadoop组件，包括HDFS、MapReduce、Hive、Pig、 Hbase、Zookeper、Sqoop和Hcatalog等。

Ambari主要包括Ambari Server和Amabri Agent两部分，Ambari Server在整个架构中只有一个，元数据信息存储在Postgres数据库中，通过心跳信息的传递对多个Ambari Agent进行管理，Ambari Agent与主机一一对应，Ambari通过Ambari Agent对该主机上的组件进行管理。

下图3为基于Ambari的部署架构：



**图3 Ambari的部署架构**

3台主机对应3个Ambari Agent，其中一台主机安装Ambari Server，每台主机上安装、运行的进程如图，每台主机安装多个组件，节点上安装的进程由该节点主机性能等因素决定，可灵活调整。主机部署规划可咨询数果智能。组件之间具有依赖关系，所以组件的安装需基于上图的顺序（从下往上）进行安装配置。

具体安装流程如下：

1）安装Ambari-Server，注册主机

2）部署集群

3）修改配置、启动组件

4）集群测试

部署集群有两种方式供选择，一键部署和独立部署。

1）一键部署：主要在终端操作，修改安装所需的配置文件后，运行安装脚本；

2）独立部署：主要是通过Ambari的Web界面进行操作。

一键部署适合对Linux系统熟悉，对Json语法规则有所了解的人员。独立部署操作界面更友好，但耗时一些。建议非运维人员通过独立部署方式进行安装，具体操作在分布式集群部署部分有详细介绍，可先查看再决定部署方式。

**4. 单机版部署**

本部分描述的单机版部署，所有服务均安装在同一台主机上，是在全新安装的主机上进行的，不建议在已有其它服务的主机上进行部署。由于部分环境内网无法访问，需要通过公网进行访问，则需要绑定公网IP(需开放端口号: 80、8000、8080、8887和8090)，并且在启动部署脚本时会有些不同。

4.1 前期准备

**主机配置要求**

|  |  |  |
| --- | --- | --- |
| **项目** | **要求** | **备注** |
| 系统 | CentOS 6.8 | 其它系统未经测试，不建议使用 |
| CPU | 4核以上 |  |
| 总内存 | 16G以上 |  |
| 磁盘 |  | 根据数据量、存储周期决定，不做RAID，单盘500G到2TB |

**表1 单机版主机配置要求**

也可根据实际情况对主机配置进行调整，但不建议降低配置，主机安装系统后，修改静态IP地址，下载单机版安装包，如果无法连接网络，请安装本地yum源。

4.2 一键部署

解压单机版安装包，启动安装脚本，如果内网无法访问，需要通过公网IP对服务进行访问时，请绑定公网IP，且启动安装脚本时，参数选择会略有不同。

|  |
| --- |
| tar –zxvf single\_deploy.tar.gz  cd single\_deploy  1)无需公网IP时执行：  source single-deploy.sh –IP 192.168.0.120  2)需要公网IP时执行：  source single-deploy.sh –IP 192.168.0.120 -public\_IP 192.168.0.121 |

此时，所有服务安装完成并启动，端口80、8000、8887和8090都能被访问表示正常。

如果安装是通过无需公网IP的方式执行部署脚本，而安装完成后因其它原因需要通过公网IP才能访问时，请开放端口，且将以下参数的IP地址修改为公网IP：

|  |
| --- |
| cd /opt/apps/astro\_sugo/analytics  vim config.js  collectGateway: 'http://192.168.0.122'  sdk\_ws\_url: 'ws://192.168.0.122:8887'  websdk\_api\_host: '192.168.0.122:8000' |

4.3 环境测试

测试方法与分布式集群测试一致，具体请参考[分布式集群测试](#分布式集群测试)步骤。

**5. 分布式集群部署**

5.1 前期准备

**5.1.1主机配置要求**

|  |  |  |
| --- | --- | --- |
| **项目** | **要求** | **备注** |
| 系统 | CentOS 6.8 | 其它系统未经测试，不建议使用 |
| 总CPU | 8核以上 |  |
| 总内存 | 32G以上 |  |
| 主机数量 | 3台及以上 |  |
| 磁盘 |  | 根据数据量、存储周期决定，不做RAID，单盘500G到2TB |
| 其它 | 静态IP、二级域名类型的hostname，yum源，网络 | hostname格式: test1.sugo.vm  主机之间网络互通 |

**表2 主机配置要求**

也可根据实际情况对主机配置进行调整，但不建议降低配置

**5.1.2 组件安装规划及准备**

根据技术架构、部署架构及主机配置，合理规划组件的安装节点，表3为3节点的安装规划（3台主机配置均为4核16G），供参考，具体可咨询数果智能。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Services** | **Components** | **Host1** | **Host2** | **Host3** |
| PostgresQL | Postgres server | √ |  |  |
| Redis | Redis server | √ |  |  |
| Zookeeper | ZooKeeper Server | √ | √ | √ |
| HDFS | Namenode | √ | √ |  |
| Datanode | √ | √ | √ |
| ZKFailoverController | √ | √ |  |
| Journalnode | √ | √ | √ |
| Yarn | Resourcemanager | √ | √ |  |
| ProxyServer |  |  | √ |
| Nodemanager |  |  | √ |
| MapReduce | History Server |  |  | √ |
| Kafka | Kafka Broker | √ | √ | √ |
| Gateway | Gateway Server | √ |  |  |
| Tindex(DruidIO) | Broker | √ | √ |  |
| Historical |  | √ | √ |
| Overlord | √ | √ |  |
| MiddleManager | √ | √ |  |
| Coordinator | √ | √ |  |
| Astro | ASTRO UI | √ |  |  |

**表3 安装配置要求**

做好主机规划后，配置好各主机的静态IP，修改hostname，注意hostname需要为二级域名，如：test1.sugo.vm，如果是离线主机，需要配置本地yum安装源库。这部分的具体操作不做描述。

5.2 选择部署方式

**环境部署的不同场景：**

* 如果您的主机是新安装的，在配置静态IP、修改hostname后，可通过脚本一键部署环境，这也是我们推荐的方式；
* 如果您的主机上有其它服务（部署架构中不包含该服务），需手动安装，暂由数果人员安装；
* 如果需要将集群与已有的其它集群结合使用，请联系数果人员，由数果人员安装。

生产环境已有其它服务时，请与数果人员沟通或由数果人员进行安装，勿利用此脚本独自安装！

5.3集群部署

**5.3.1 一键部署**

一键部署脚本包含了各个主机上需要安装的组件(即：如[表3 安装配置要求](#表3安装配置要求)所示的规划)，且默认按照该规划进行安装，如果需要自定义各主机上所安装的组件，可通过csv格式的文本进行编辑，具体格式请参考解压后的脚本sugo\_yum/deploy\_scripts/centos6/conf/ hosts.csv，格式如下：

POSTGRES\_SUGO,SUGO\_POSTGRES\_SERVER,test01.sugo.vm,,

需要注意的是：

* 当前此一键部署脚本只能安装该文件内包含的服务(即第一列的所有值)，如需其它服务，请在一键部署完成后手动安装其它服务
* HDFS需要2个NAMENODE，2个ZKFC且与NAMENODE在相同主机
* YARN需要2个RESOURCEMANAGER

**1）安装部署**

联系数果智能获取安装包的下载链接。

操作步骤：

第1步：进入数据存储目录(注意: 不能将安装包存储于/root/目录下的任何目录)，下载本地安装服务的安装包；

第2步：解压安装包，进入部署脚本目录${数据存储目录}/sugo\_yum/deploy\_scripts/centos6/ambari-server/，修改host和ip.txt文件。host文件为各主机ip与hostname的映射， ip.txt文件为各主机的root用户密码+ hostname，以空格分割。如果配置了ambari-server到其它主机的ssh免密码登录，则不需要ip.txt文件；

(注意： 如果是自定义安装或者节点数不为3，则需编辑sugo\_yum/deploy\_scripts/centos6/conf/hosts.csv文件进行自定义，或在excel等工具中编辑好后替换该文件，文件名必须为hosts.csv)

第3步：修改完成后保存，执行脚本start.sh（http的端口号建议不设置为80，因为安装网关时可能会占用端口号80），具体操作如下：

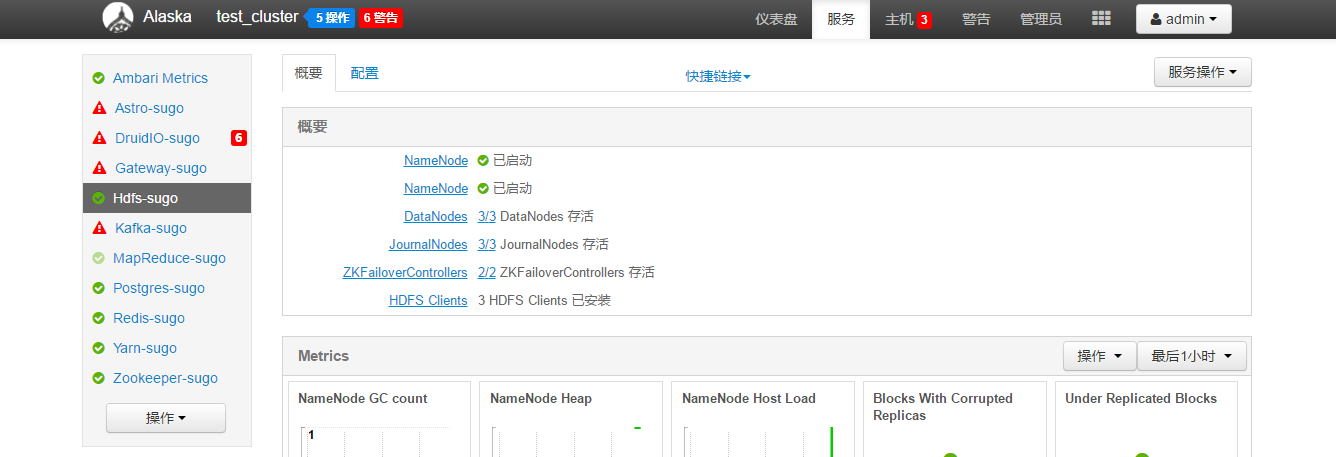
|  |
| --- |
| mkdir ${数据存储目录}  cd ${数据存储目录}  wget ${安装包下载链接}  tar -zxf sugo\_yum.tar.gz  cd ${数据存储目录}/sugo\_yum/deploy\_scripts/centos6/ambari-server/  vi host  192.168.10.1 test1.sugo.vm  192.168.10.2 test2.sugo.vm  192.168.10.3 test3.sugo.vm  ...  vi ip.txt  123456 test1.sugo.vm  123456 test2.sugo.vm  123456 test3.sugo.vm  ... |

启动安装脚本：

可通过./start.sh -help查看脚本使用方法，如无特殊需求，可按如下格式启动脚本:

|  |
| --- |
| ./start.sh -ambari\_ip 192.168.0.1  注: 默认httpd服务端口号为81，集群名称为sugo\_cluster，需要设置为其它时，请在执行脚本时添加对应参数及值，建议将执行脚本的日志信息重定向要文件中，方便对日志信息进行查阅  如果已经配置了ssh免密码登录，则需要加上参数-skip\_ssh |

等待脚本执行完成，脚本执行完成大概需要10分钟，完成后打开Web界面，如下图所示：



**图4 正在安装服务界面**

上图显示正在启动相关组件，等待启动完成，完成后如下图所示，即可开始修改配置。



**图5 成功安装服务界面**

**2）安装失败解决办法**

利用一键部署脚本安装时，可能会由于配置文件输入错误等原因造成安装失败，安装过程中如果失败，会有相关提示信息，或者脚本中断，安装失败后，检查并修改配置信息，然后重新启动一键部署脚本

**3）修改配置**

表4为需要修改的服务和参数(其中Astro的参数，如果无需公网IP即可访问，则无需修改)：

|  |  |  |  |
| --- | --- | --- | --- |
| **Services** | **Parameters** | **Value(example)** | **备注** |
| DruidIO | druid.license.signature |  | 联系数果智能获取秘钥 |
| Astro | site.collectGateway | http://test1.sugo.vm | 数据上报网关 |
| site.websdk\_app\_host | test1.sugo.vm:8000 | Web获取埋点事件服务端 |

**表4 修改的服务和参数**

1. **Tindex**

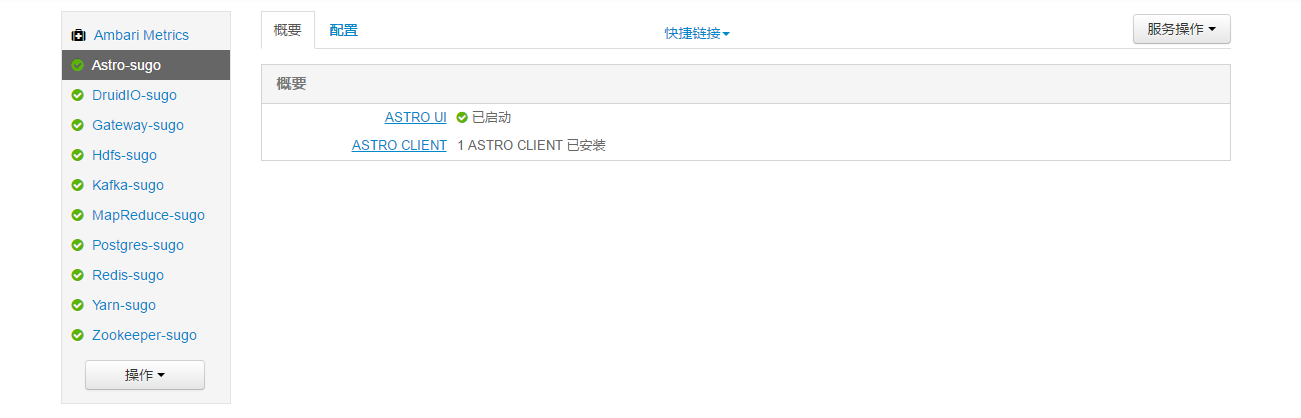
点击DruidIO-sugo，点击”配置”，点击”高级 common.runtime”修改参数druid.license.signature值，根据表4修改参数，保存后点击”服务操作”的“重启所有“



**图6 Tindex启动成功**

1. **Astro**

如果无需通过端口转发利用公网 IP进行访问，则无需修改，否则根据表4修改参数，保存后重启，重启完成后通过快捷链接打开前端访问页面，联系数果人员，发送注册码获取验证码，即完成安装



**图7 Astro启动成功**

**5.3.2 独立部署**

**1）Ambari-server的安装**

ambari-server可通过脚本进行安装，也可以手动安装，在生产环境中，不建议轻易使用脚本进行安装！

下载或上传安装包，上传或下载到数据存储目录(注意: 不能将安装包存储于/root/目录下的任何目录)，打开终端，解压安装包。

**此部分为脚本安装ambari-server:**

* 创建数据存储目录，下载本地安装的yum源安装包（联系数果智能获取下载链接）；
* 解压安装包，进入部署脚本目录${yum源下载目录}/sugo\_yum/deploy\_scripts/centos6/ambari-server/，修改host和ip.txt文件。host文件为各主机ip与hostname的映射，ip.txt文件为所有主机的root用户密码+hostname，以空格分割。如果配置了ambari-server到其它主机的ssh免密码登录，则不需要ip.txt文件；
* 修改完成后保存，执行脚本start.sh（http的端口号建议不设置为80，因为安装网关时会占用端口号80），具体操作如下：

|  |
| --- |
| mkdir ${数据存储目录}  cd ${数据存储目录}  wget ${yum源下载链接}  tar -zxf /sugo\_yum.tar.gz  cd sugo\_yum/deploy\_scripts/centos6/ambari-server/  vi host  192.168.10.1 test1.sugo.vm  192.168.10.2 test2.sugo.vm  192.168.10.3 test3.sugo.vm  ...  vi ip.txt  123456 test1.sugo.vm  123456 test2.sugo.vm  123456 test3.sugo.vm  ... |

启动安装脚本：

可通过./start.sh –help查看脚本使用方法，如无特殊需求，可按如下格式启动脚本:

|  |
| --- |
| ./start.sh -ambari\_ip 192.168.0.1 -skip\_cluster\_services |

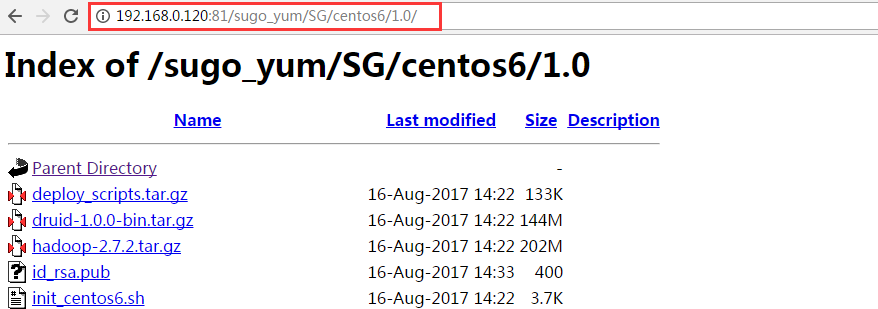
等待脚本执行完成，脚本执行完成大概需要3分钟，完成后打开Web界面，进行主机注册及services的安装，通过Web界面登录后会提示注册集群、主机，ssh秘钥及安装Ambari Metrics，具体操作如下：

**第1步：**启动安装向导（如下图所示）；



**图8 Ambari安装向导页面**

**第2步：**命名集群，选择服务栈（此处仅勾选redhat6），浏览器打开http服务地址（如图9所示），选择1.0所在的目录，复制链接粘贴到Ambari界面的”基础URL”内（如图10所示）。



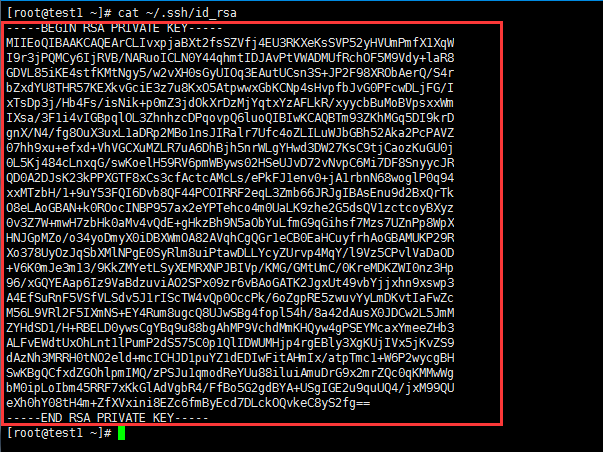
**图9 基础URL链接**



**图10 选择服务栈**

**第3步：**点击下一步后，如图11所示，获取秘钥（在Ambari-server所在节点的终端查询），复制粘贴到安装选项界面（如图12所示），同时在界面填写目标主机：

|  |
| --- |
| cat ~/.ssh/id\_rsa |



**图11 SSH秘钥**



**图12 安装选项界面**

**第4步：**确认注册，Ambari-Server便开始注册主机并检测环境，图13表明主机注册成功，点击下一步，勾选Ambari Metrics，点击下一步，分配主从节点，可打开下拉框选择将Ambari Metrics安装在您规划的主机上，按照提示填写参数Grafana Admin Password（admin,admin。也可自行填写其它密码），下一步，部署，然后等待安装完成。







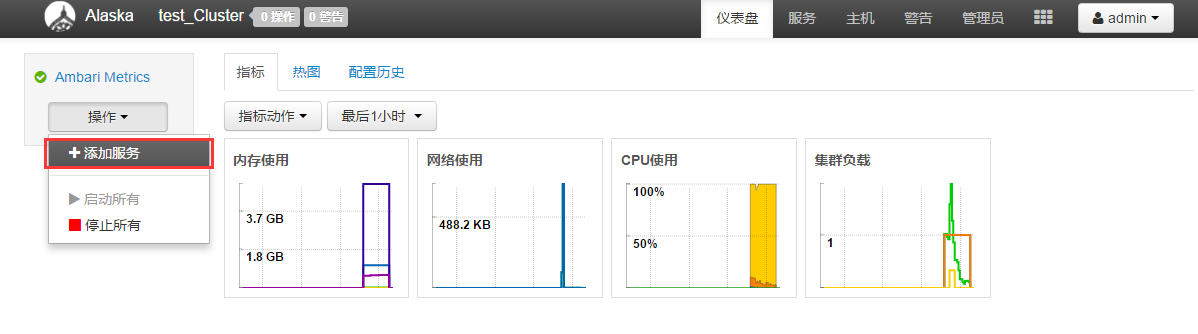
**图13 主机注册及Ambari Metrics安装成功界面**

**2）其它服务(services)的安装**

独立部署在安装各个服务（service, 如HDFS、YARN）时，需要按照一定的顺序进行安装！

1. **Postgres**

第1步：添加服务（如图14所示），选择服务，勾选Postgres-sugo，点击下一步



**图14 添加服务**

第2步：分配主从节点，选择Postgres安装的节点（如下图所示），点击下一步



**图15 分配主节点**

正常启动后，创建其它服务的依赖库：

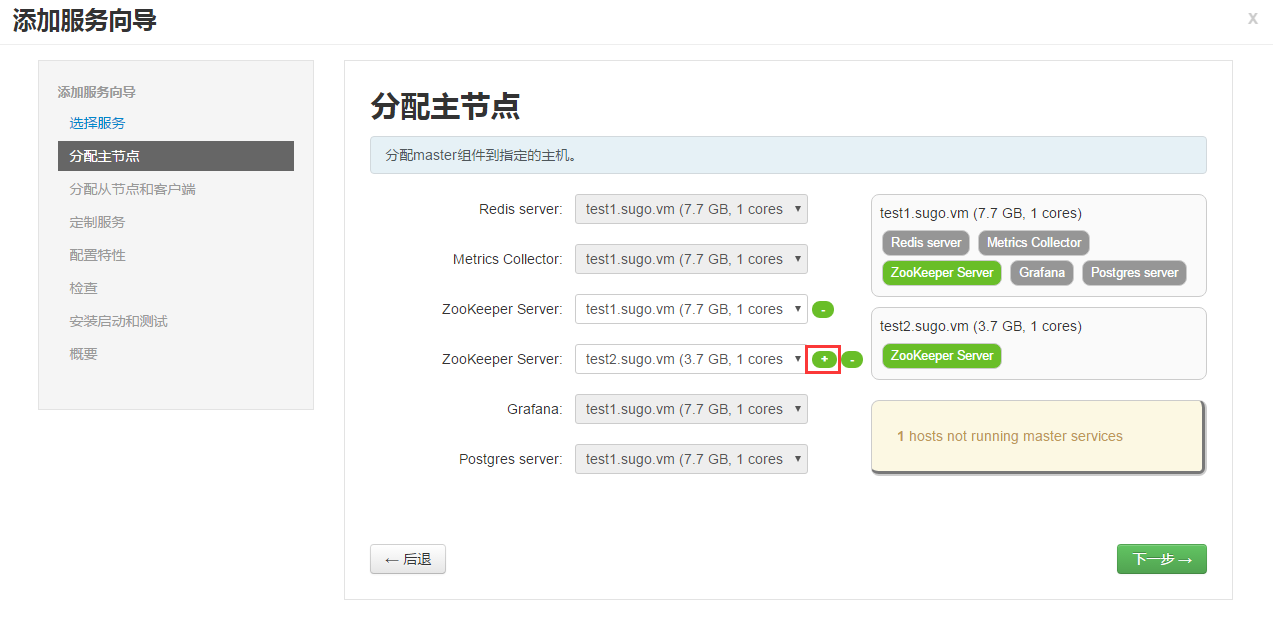
|  |
| --- |
| cd /opt/apps/postgres\_sugo  bin/psql -p 15432 -U postgres -d postgres -c "CREATE DATABASE druid WITH OWNER = postgres ENCODING = UTF8;"  bin/psql -p 15432 -U postgres -d postgres -c "CREATE DATABASE sugo\_astro WITH OWNER = postgres ENCODING = UTF8;"  bin/psql -p 15432 -U postgres -d postgres -c "CREATE DATABASE monitor WITH OWNER = postgres ENCODING = UTF8;"  bin/psql -p 15432 -U postgres -d postgres -c "CREATE DATABASE pio WITH OWNER = postgres ENCODING = UTF8;"  bin/psql -p 15432 -U postgres -d postgres -c "select datname from pg\_database" |

1. **Redis**

安装步骤与Postgresql基本相同，且无需修改配置信息，按照提示操作即可

1. **Zookeeper**

添加服务，选择服务，分配主节点，此处有多个Zookeeper Server时，需要点击加号按钮选择多个节点



**图16添加Zookeeper主节点**

按照提示完成Zookeeper的安装，配置文件无需修改

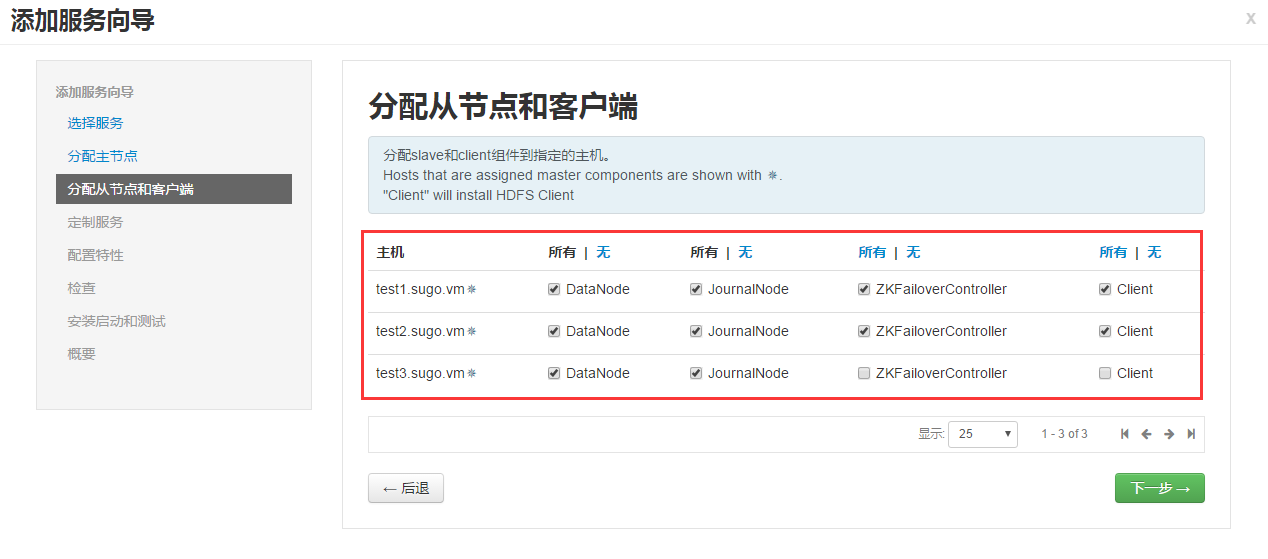
1. **HDFS**

第1步：按提示进行操作，分配主节点时，需要通过加号按钮添加一个NameNode（如图17所示），实现HDFS集群的高可用



**图17 添加NameNode主节点**

第2步：点击下一步，分配从节点和客户端（如下图所示），下一步，部署



**图18 选择从节点和客户端**

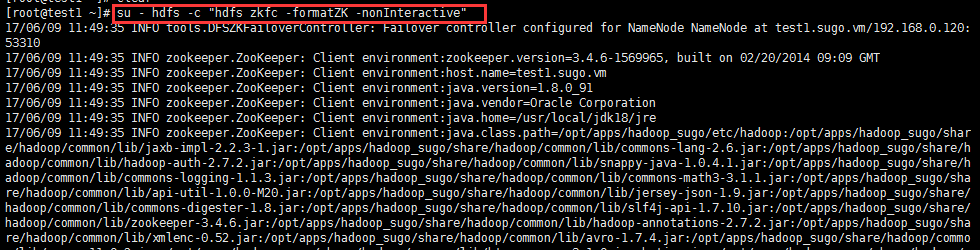
注意：在安装完成后会出现报错信息（如下图所示），为正常现象，点击下一步，完成，然后开始HDFS的格式话操作



**图19 独立部署HDFS报错信息**

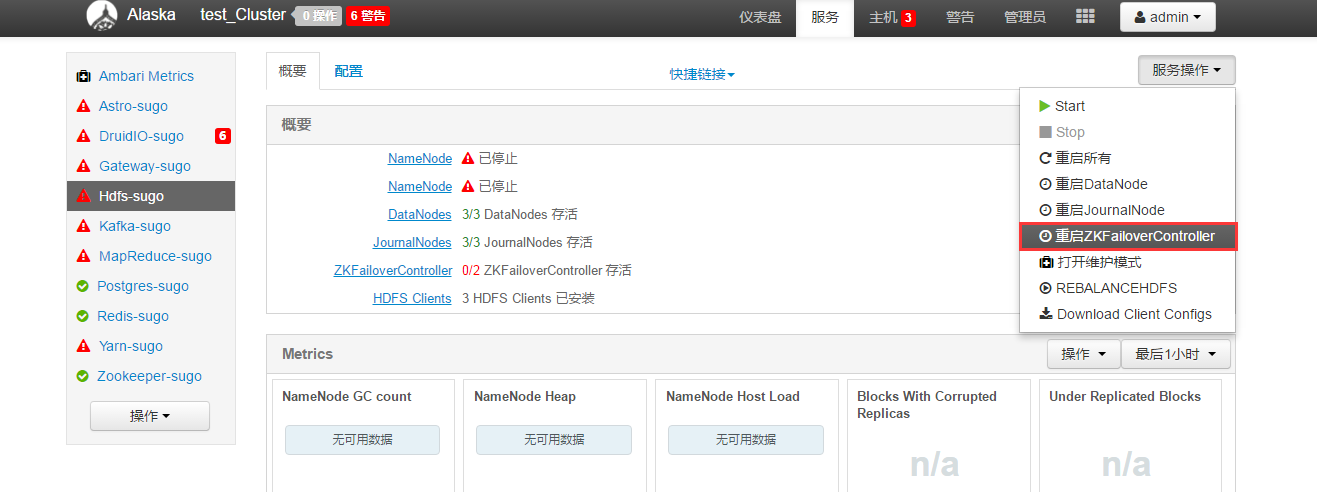
第3步：在NameNode1节点上执行zkfc格式化（如图20所示）：

|  |
| --- |
| su - hdfs -c "hdfs zkfc -formatZK -nonInteractive" |



**图20 zkfc格式化**

第4步：格式化完成后，在界面启动所有zkfc（如图21所示）。



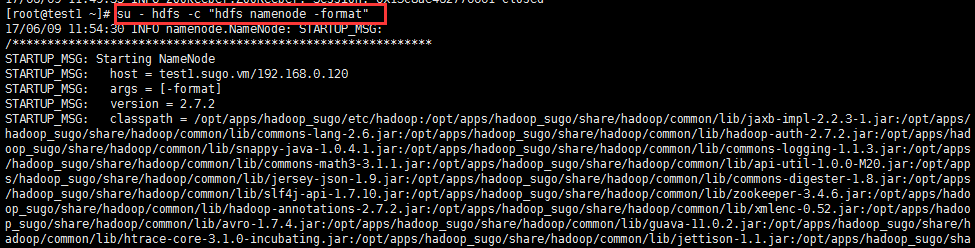
**图21 ZKFC启动界面**



**图22 ZKFC重启界面**

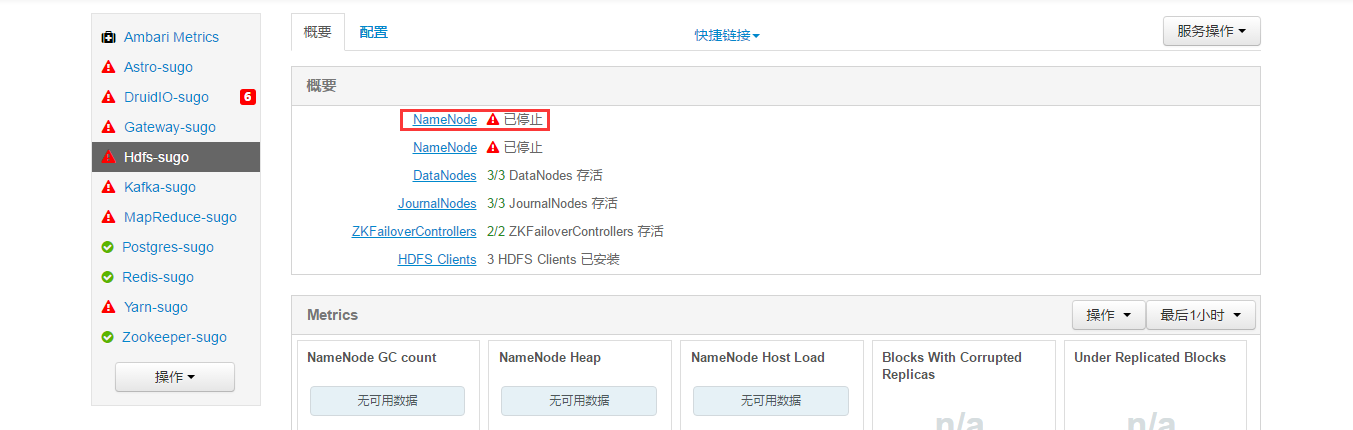
第5步：zkfc重启完成后，界面会显示zkfc处于启动状态，然后在NameNode1上执行格式化操作：

|  |
| --- |
| su - hdfs -c "hdfs namenode -format" |



**图23 NameNode1执行格式化**

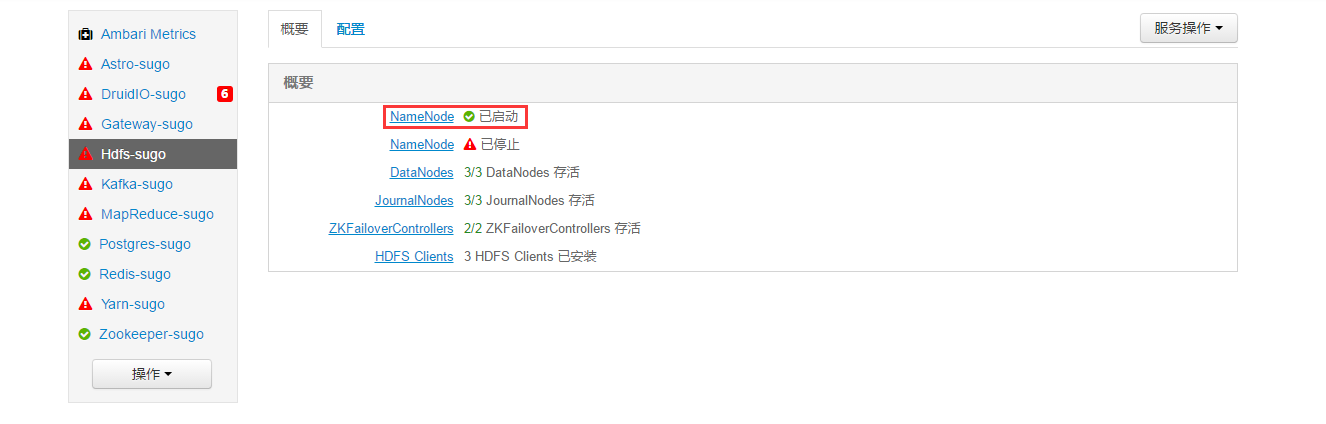
第6步：格式化完成后，在界面重启NameNode1（如图24）。





**图24重启 NameNode1**

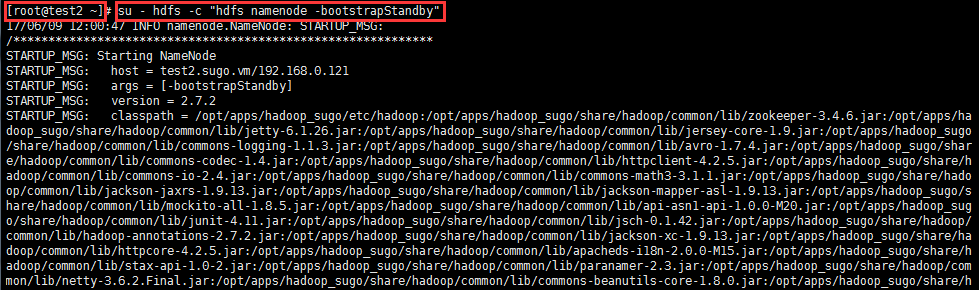
第7步：启动完成后，如下图所示，界面会显示test1.sugo.vm上的NameNode已启动。



**图25 成功启动NameNode1**

第8步：在NameNode2节点执行格式化后的数据同步（如下图所示）。

|  |
| --- |
| su - hdfs -c "hdfs namenode -bootstrapStandby" |



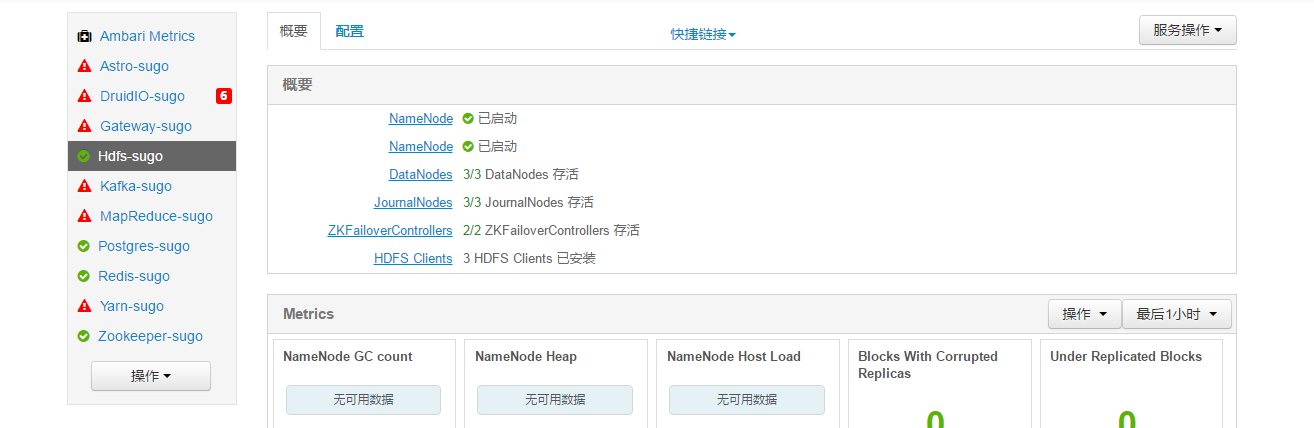
**图26 NameNode2执行格式化后的数据同步**

第9步：数据同步命令执行完成后，启动NameNode2



**图27重启 NameNode2**

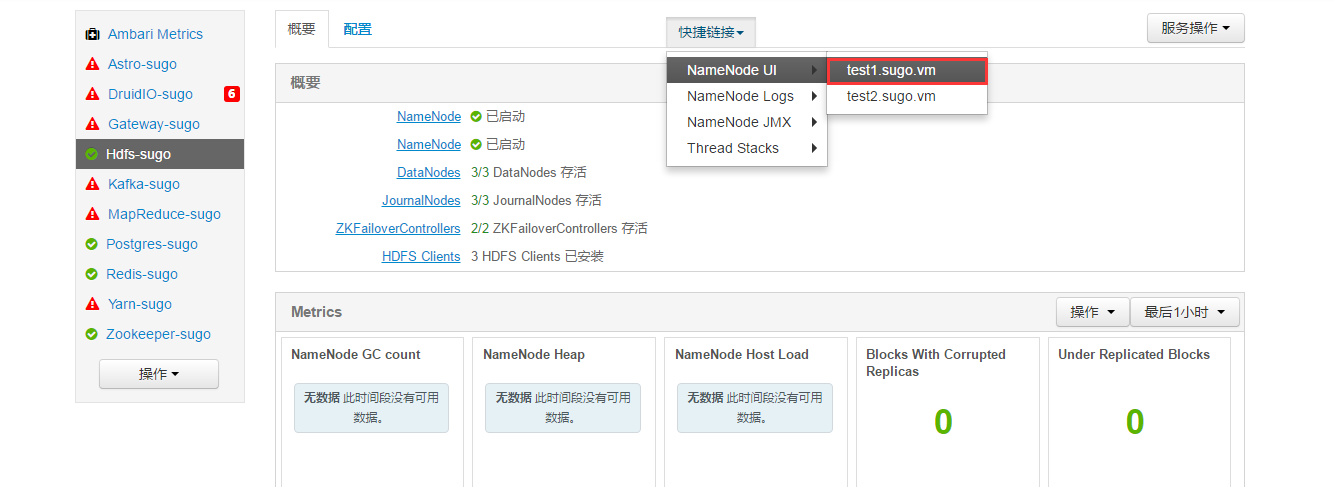
启动完成后不再显示报错信息



**图28 HDFS安装启动成功**

第10步：在浏览器上打开NameNode的IP:50070页面，通过Amabari主机名打开页面时，需要在windows的host文件中配置IP与hostname的映射

|  |
| --- |
| 打开文件C:\Windows\System32\drivers\etc\host  复制Linux下/etc/hosts文件内的映射，追加到Windows的host文件末尾  192.168.10.1 test1.sugo.vm  192.168.10.2 test2.sugo.vm  192.168.10.3 test3.sugo.vm |



**图29 打开HDFS文件目录界面**

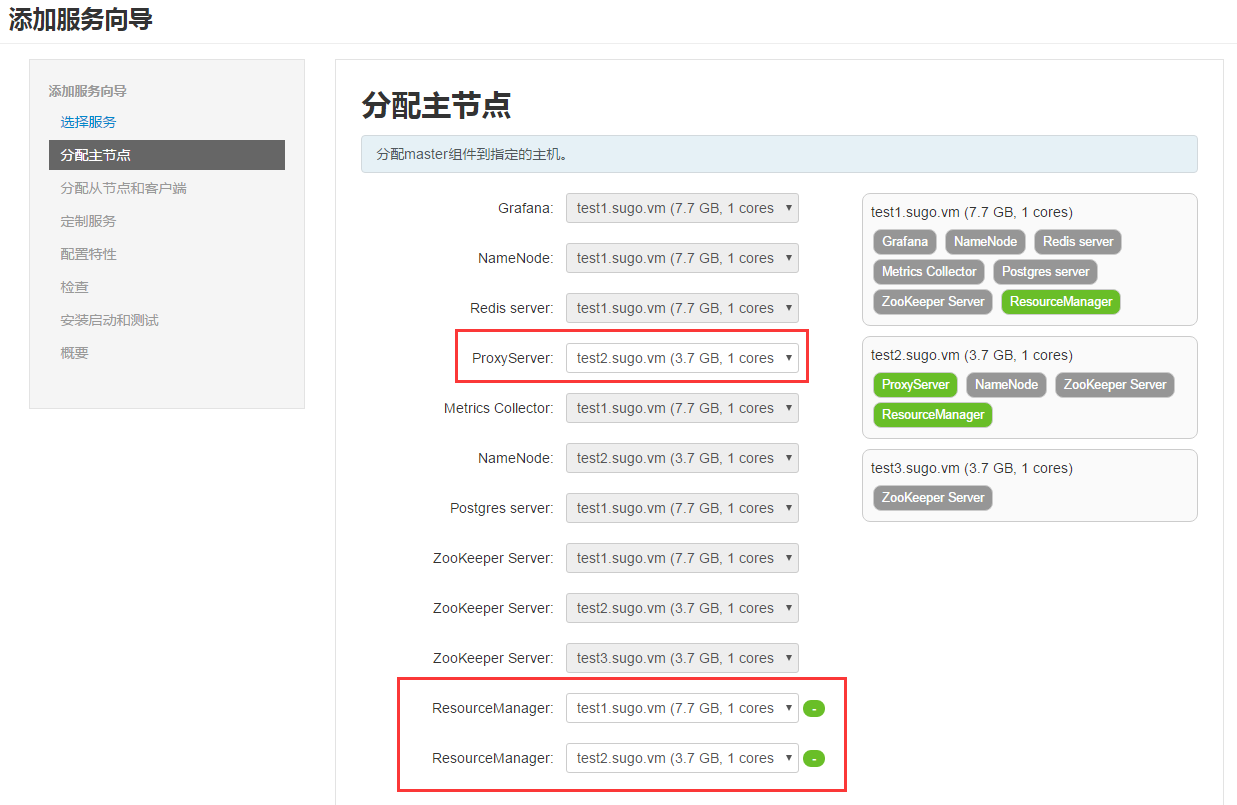
通过主机名打开HDFS界面：test1.sugo.vm:50070

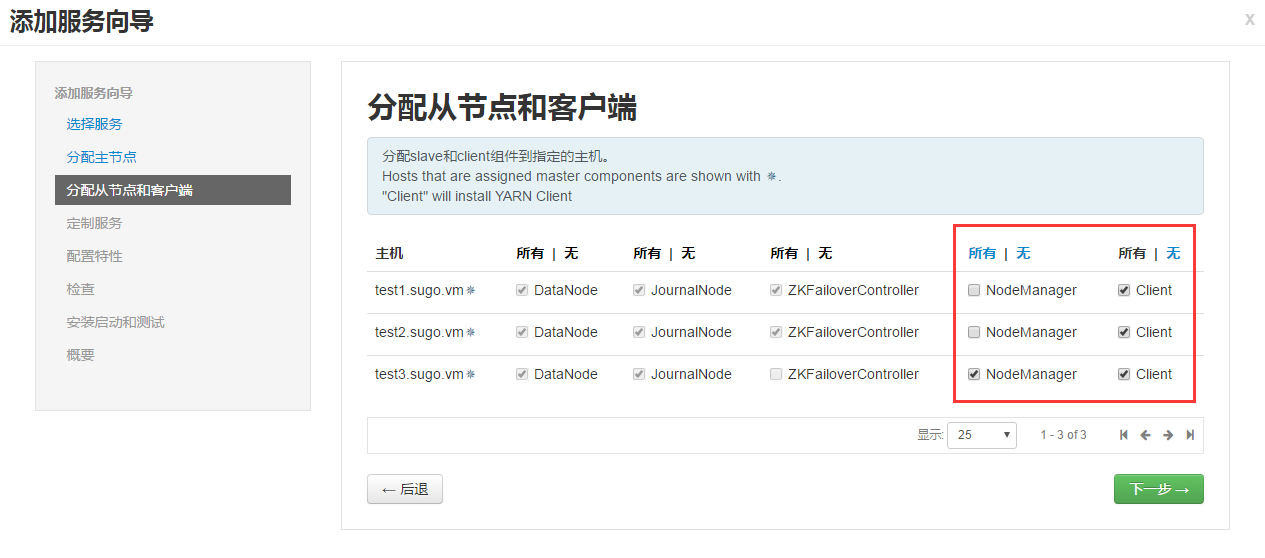
第11步：配置NameNode1和NameNode2在 hdfs用户下的免密码登录，启动脚本：

|  |
| --- |
| cd ${数据存储目录}/deploy\_scripts/centos6/service  ./passwdless.sh {namenode1} {namenode2} {user} {group}  例：./passwdless.sh test1.sugo.vm test2.sugo.vm hdfs hadoop |

1. **YARN**

按照提示操作即可，注意ResourceManager需要两个（如下图所示），实现YARN的高可用





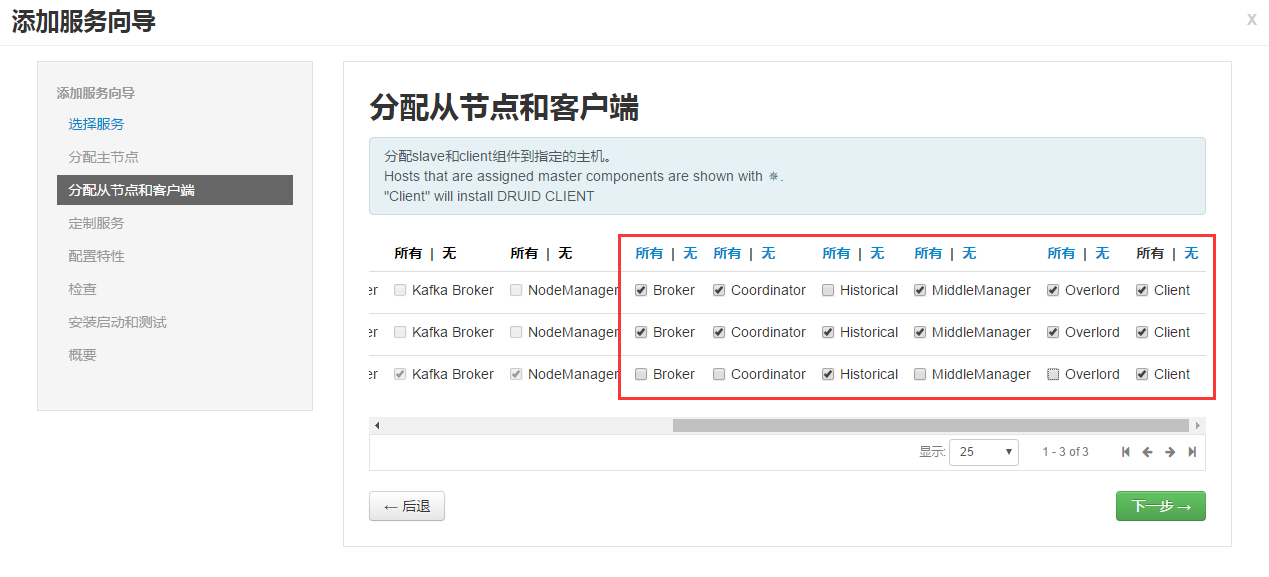
**图30 分配YARN主节点、从节点和客户端**

1. **MapReduce、Kafka、Gateway**

按照提示操作即可

1. **Tindex**

第1步：按照提示进行操作（如下图所示）

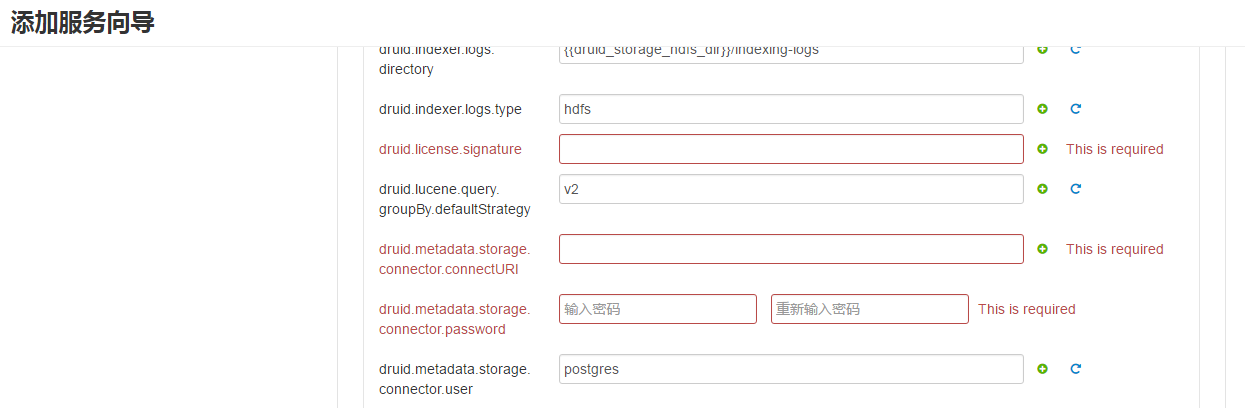


**图31 选择druid从节点和客户端**

第2步：修改配置文件（需修改参数如表5所示），然后按提示完成安装

|  |  |  |
| --- | --- | --- |
| **配置项（参数）** | **参数值** | **备注** |
| druid.license.signature |  | 联系数果智能获取秘钥 |
| druid.metadata.storage.connector.connectURI | jdbc:postgresql://test1.sugo.vm:15432/druid | Postgresql数据库的druid库 |

**表5 Tindex需修改的参数**



**图32 修改druid参数界面**

1. **Astro**

第1步：按提示安装，修改配置文件（需修改参数如表6所示），修改完成后按提示完成安装

|  |  |  |
| --- | --- | --- |
| **配置项（参数）** | **参数值** | **备注** |
| db.database | test1.sugo.vm | Postgresql数据库所在主机 |
| redis.host | test1.sugo.vm | Redis所在主机 |
| site.collectGateway | http://test1.sugo.vm | 数据上报网关 |
| site.websdk\_app\_host | test1.sugo.vm:8000 | Web获取埋点事件服务端 |

**表6 Astro需修改的参数**

**5.3.3 集成已有的Kafka和Hadoop集群**

需要注意的是：

* 需要Hadoop的配置文件信息

hdfs-site.xml

core-site.xml

yarn-site.xml

mapred-site.xml

* Kafka集群信息，如： test1.sugo.vm:9092,test2.sugo.vm:9092

**1）安装部署**

|  |
| --- |
| mkdir ${数据存储目录}  cd ${数据存储目录}  wget ${安装包下载链接}  tar -zxf sugo\_yum.tar.gz  cd ${数据存储目录}/sugo\_yum/deploy\_scripts/centos6/ambari-server/  vi host  192.168.10.1 test1.sugo.vm  192.168.10.2 test2.sugo.vm  192.168.10.3 test3.sugo.vm  ...  vi ip.txt  123456 test1.sugo.vm  123456 test2.sugo.vm  123456 test3.sugo.vm  ... |

启动安装脚本：

可通过./start.sh -help查看脚本使用方法，如无特殊需求，可按如下格式启动脚本:

|  |
| --- |
| ./start.sh -ambari\_ip 192.168.0.1 -skip\_hadoop -skip\_kafka  注: 默认httpd服务端口号为81，集群名称为sugo\_cluster，需要设置为其它时，请在执行脚本时添加对应参数及值，建议将执行脚本的日志信息重定向要文件中，方便对日志信息进行查阅  如果已经配置了ssh免密码登录，则需要加上参数-skip\_ssh |

等待脚本执行完成，脚本执行完成大概需要10分钟，完成后打开Web界面，如下图所示：



**图33 正在安装服务界面**

上图显示正在启动相关组件，等待启动完成，完成后如下图所示，即可开始修改配置。



**图34 成功安装服务界面**

**2）安装失败解决办法**

利用一键部署脚本安装时，可能会由于配置文件输入错误等原因造成安装失败，安装过程中如果失败，会有相关提示信息，或者脚本中断，安装失败后，检查并修改配置信息，然后重新启动一键部署脚本

**3）修改配置**

表7为需要修改的服务和参数(其中Astro的参数，如果无需公网IP即可访问，则无需修改)：

|  |  |  |  |
| --- | --- | --- | --- |
| **Services** | **Parameters** | **Value(example)** | **备注** |
| DruidIO | core-site.xml |  | 自定义hadoop.config，添加 |
| hdfs-site.xml |  | 自定义hadoop.config，添加 |
| yarn-site.xml |  | 自定义hadoop.config，添加 |
| mapred-site.xml |  | 自定义hadoop.config，添加 |
| supervisor.kafka.zkHost | test1.sugo.vm:2181,test2.sugo.vm:2181,test3.sugo.vm:2181/kafka | Kafka集群依赖的Zookeeper集群信息。自定义supervisor.properties，添加 |
| Gateway | bootstrap.servers | test1.sugo.vm:9092, test2.sugo.vm:9092 | Kafka集群信息。自定义kafka.properties, Add Property…添加该参数 |
| Astro | site.collectGateway | http://test1.sugo.vm | 数据上报网关 |
| site.websdk\_app\_host | test1.sugo.vm:8000 | Web获取埋点事件服务端 |

**表7 修改的服务和参数**

1. **Gateway**

点击Gateway-sugo，点击”配置”，点击”自定义kafka.properties”Add property新增参数bootstrap.servers值，根据表4修改参数，保存后点击”服务操作”的“重启所有“

1. **Tindex**

点击DruidIO-sugo，点击”配置”，点击” 自定义 supervisor.properties”，新增参数supervisor.kafka.zkHost及值，点击”高级 hadoop.config”, 根据表4修改参数值，保存后点击”服务操作”的“start“

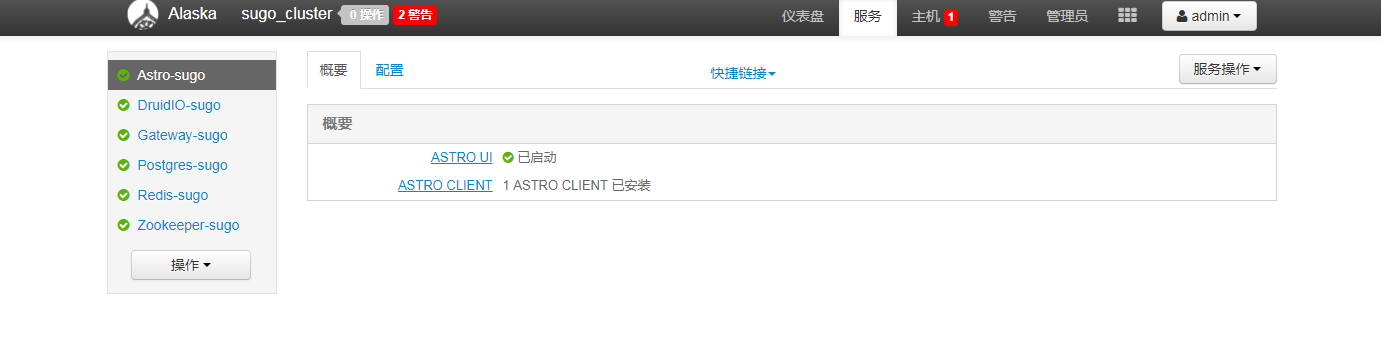




**图35 Tindex启动成功**

1. **Astro**

如果无需通过端口转发利用公网 IP进行访问，则无需修改，但是需要重启，否则根据表4修改参数，保存后重启，重启完成后通过快捷链接打开前端访问页面，即完成安装



**图36 Astro启动成功**

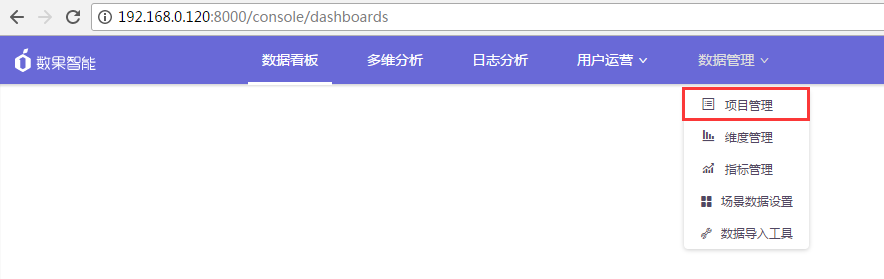
5.4 分布式集群测试

第1步：打开Astro前端Web界面IP:8000（如下图所示，帐号密码分别为：admin,admin123456)



**图37前端Astro登录界面**

第2步：进入数据管理，项目管理（如下图所示）



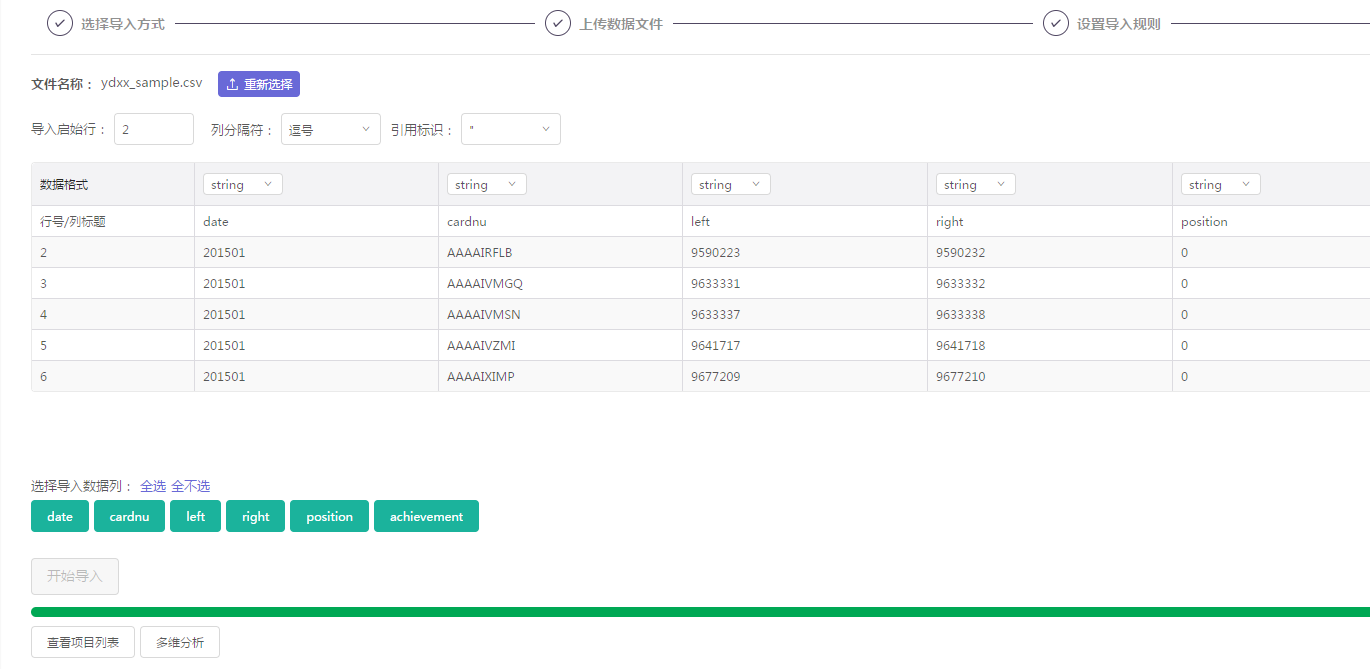
**图38前端Astro进入项目管理操作**

第3步：新建项目，输入项目名称，选择文件导入方式，执行下一步（如下图所示）



**图39选择接入数据类型**

第4步：选择文件，进入下一步（如下图所示），选择维度字段（全选），开始导入，查看是否成功



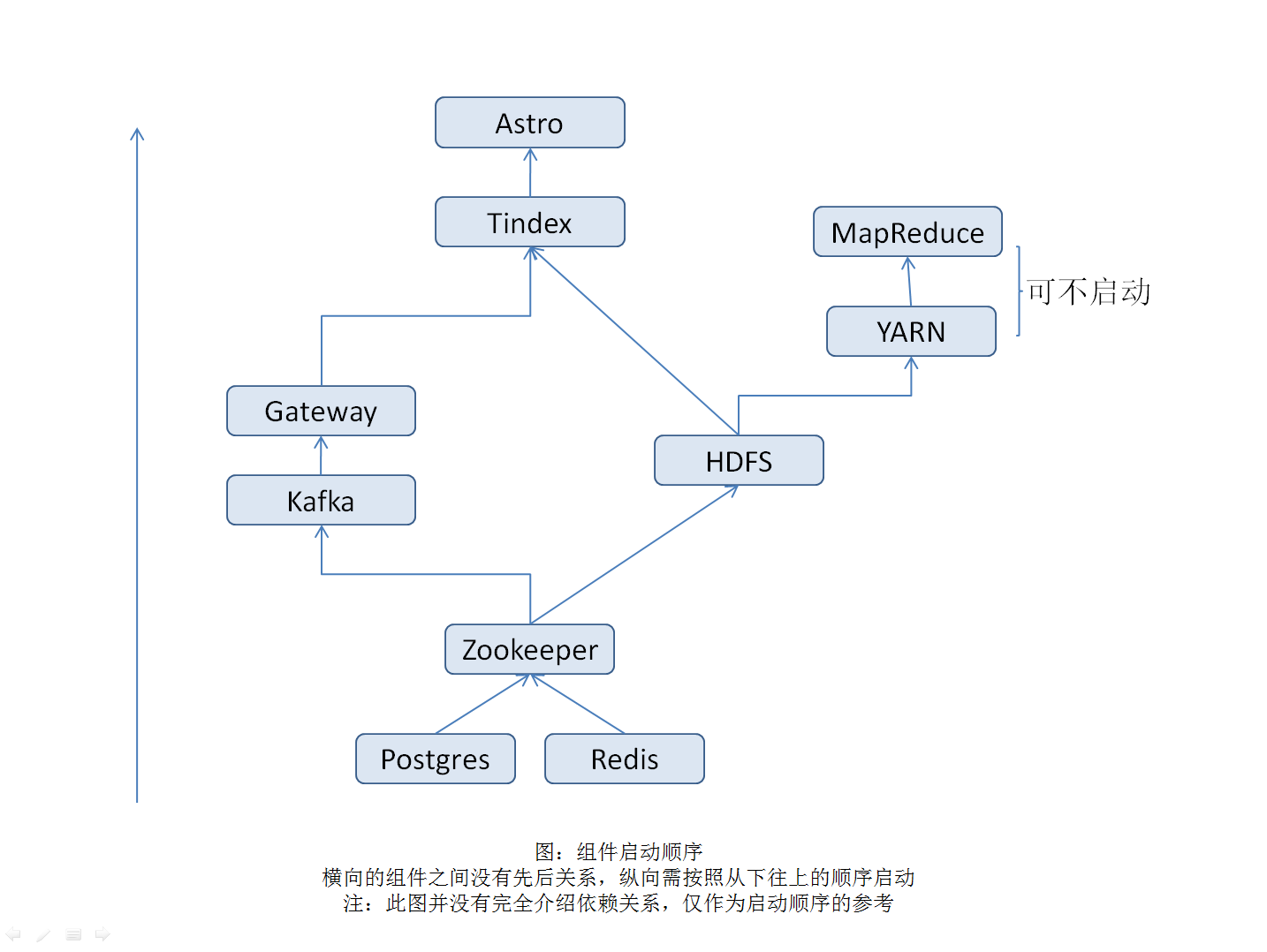
**图40配置接入参数**

第5步：点击多维分析，执行查询，出现总记录数且与源数据相同，证明部署成功

**6 集群管理**

6.1 启动集群

启动集群时，由于各组件之间具有依赖关系，需要按照一定的顺序启动各组件，可以**按照下图所示的顺序启动各组件**:



**图41组件启动顺序参考图**

启动集群后，便于更新服务等操作，需要启动httpd服务:

|  |
| --- |
| service httpd start |

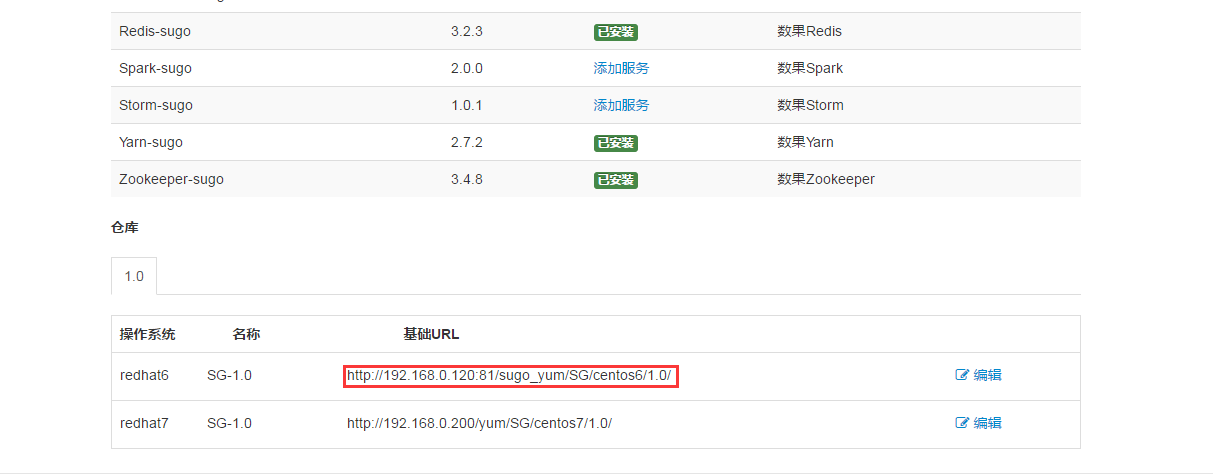
6.2 更新服务

通过Ambari界面更新部分组件时，无需重复配置参数。

更新服务具体步骤如下:

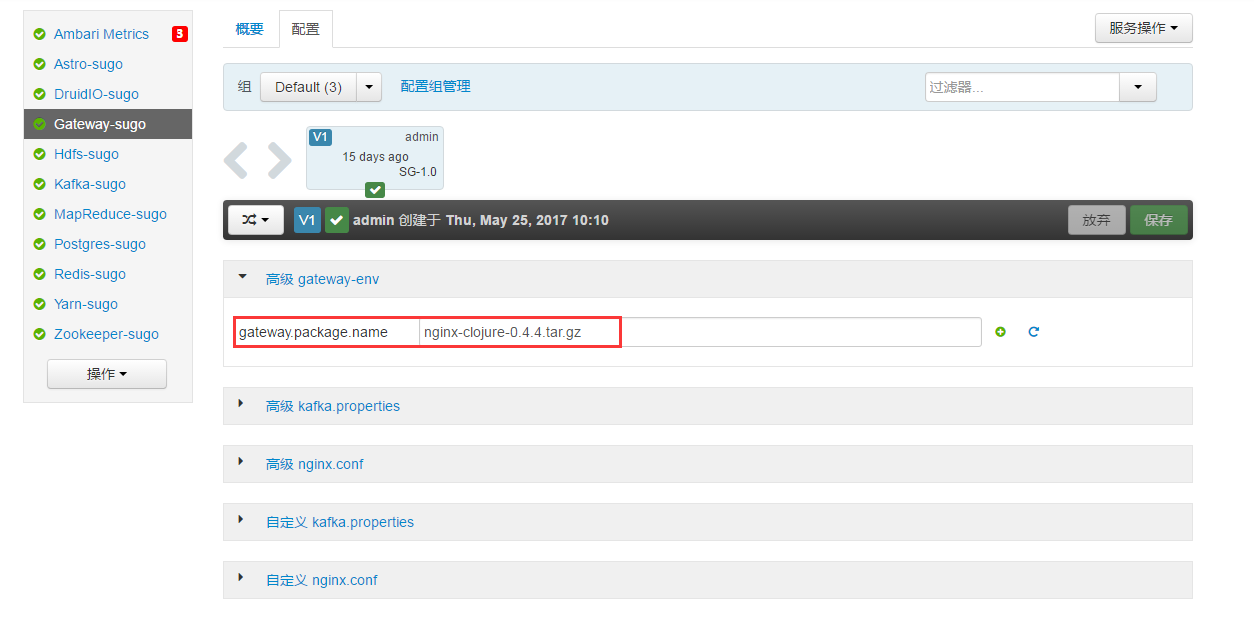
第1步：下载更新安装包到http服务所在目录，即Ambari指向的基础URL地址（如图40所示），在Ambari界面的管理员中查看，Linux系统上的位置一般为：/var/www/html/sugo\_yum/SG/centos6/1.0，操作命令如下:

|  |
| --- |
| cd /var/www/html/sugo\_yum/SG/centos6/1.0  wget {安装包链接（联系数果智能获取）}  service httpd start |



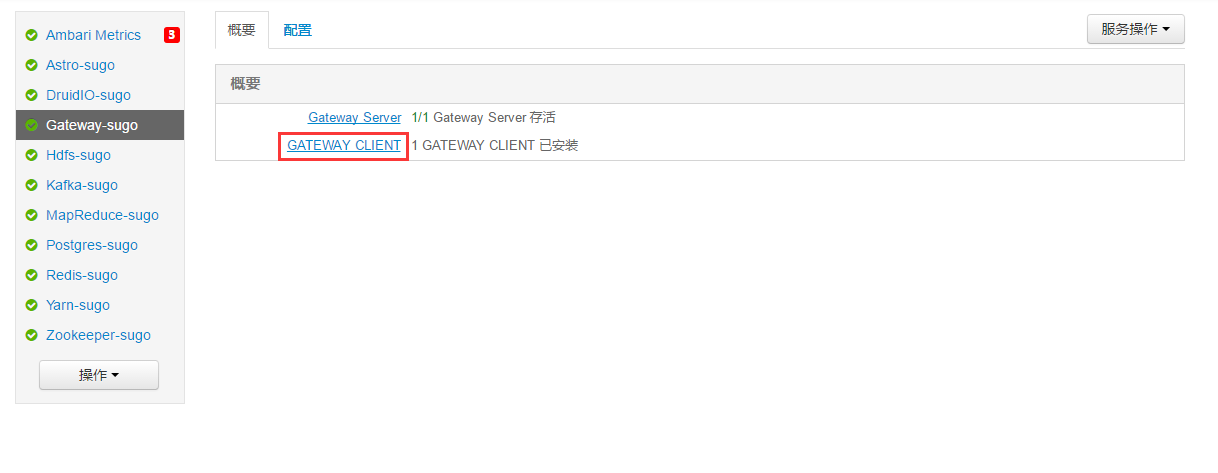
**图42查看Ambari的基础URL地址**

第2步：修改安装包包名，需与该服务配置文件中的package.name保持一致，之前的安装包可修改名称作为备份；



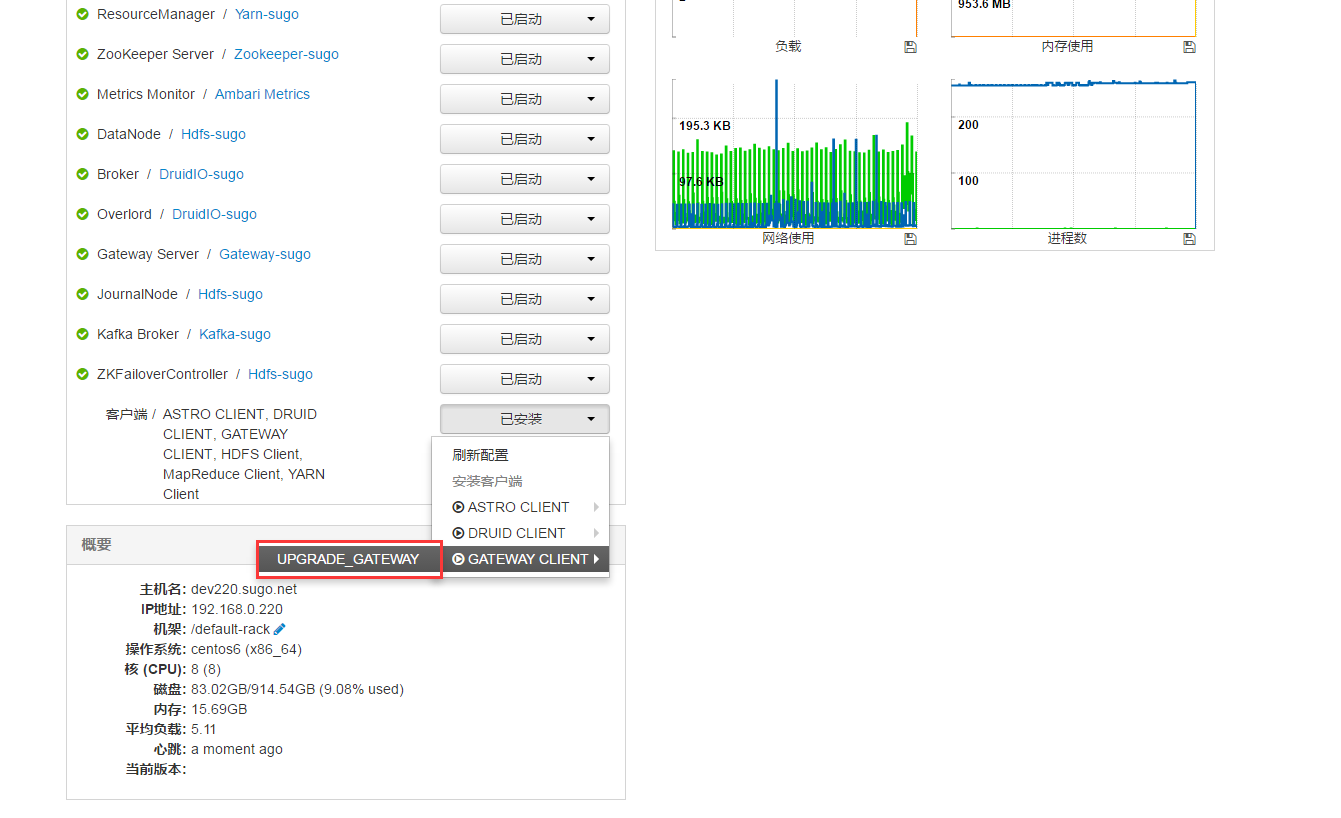
**图43查看该服务配置文件中的package name**

第3步：点击该服务的Client，选择该服务所在主机（如果该服务部署在多台主机上，每台主机都需要更新操作，即第4步操作）



**图44选择该服务的客户端**

第4步：选择主机后，下拉页面到最底端，点击更新按钮（如下图所示），即可一键更新服务



**图45一键更新服务操作**

6.3 增删Ambari-Agent

在已有Ambari的情况下，可能会遇到添加机器、扩展集群和迁移服务的需求，为了方便对集群的统一管理，增加Ambari-Agent则是一种比较合适的选择，增加Ambari-Agent后，通过Ambari对各服务进行迁移、增删等操作更加简便、友好。

增添Ambari-Agent主要分为主机准备、主机注册两个步骤，在完成主机注册后，即可在Ambari集群上对服务进行增删、迁移等。注：以下终端操作部分全部在Ambari-Server节点主机执行

**6.3.1 增加Agent**

**第1步：主机准备**

配置待添加主机的静态IP，修改hostname，注意hostname需要为二级域名，如：test1.sugo.vm，如果是离线主机，需要配置本地yum安装源库。

**第2步：Agent环境准备**

进入ambari-server所在主机终端，进入脚本目录/{安装包存储目录}/sugo\_yum/deploy\_scripts/centos6/ambari-agent(注：此处您的主机上可能不存在此目录，或者sugo\_yum目录实际为yum，可以在github上下载脚本：<https://github.com/Datafruit/deploy_scripts>)，修改hosts文件，hosts文件用于写入需要添加的agent的主机名, root用户的密码，该主机的IP地址 按行写入，每行代表一个主机(ambari-agent)，各项目之间以空格“ ”分割，例：

|  |
| --- |
| test3.sugo.vm 123456789 192.168.0.122 |

修改完成后保存，执行脚本pre\_add\_agent.sh，此脚本用于安装agent前的主机准备，包括安装jdk、配置ambari-server到该agent的ssh免密码登录、系统优化等，且jdk、ssh免密码可选择性安装，具体使用可通过如下命令进行查看

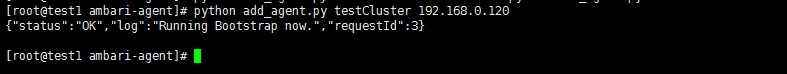
|  |
| --- |
| ./pre\_add\_agent.sh –help  例：  ./pre\_add\_agent.sh -http\_port 81 -ambari\_ip 192.168.0.120 |

**第3步：注册主机**

运行脚本add\_agent.py，此脚本用于安装、注册ambari-agent，会在hosts文件中列出的主机上安装注册agent，运行此脚本需输入参数：集群名称 ambari-server的IP地址 例：

|  |
| --- |
| python add\_agent.py testCluster 192.168.0.120 |

注：此脚本包含安装和注册两个部分，由于主机的配置等因素不同，安装所需时间也会不同，而注册需要安装完成后才可执行，此处设置安装等待时间为5秒，若因配置原因，脚本注册部分执行失败，重复执行该脚本即可



**图46注册Ambari-Agent**

注册如果没有报错，则表明Agent注册成功，查看Web管理界面，如下图显示test3.sugo.vm注册成功，即可在此基础上管理相关服务。



**图47 Ambari-Agent注册成功后的Web管理界面**

**6.3.2 迁移服务**

在迁移服务之前，请注意，需要将下线节点所在的服务的数据转移到其它节点，所以需要先添加服务，此处以kafka为例：

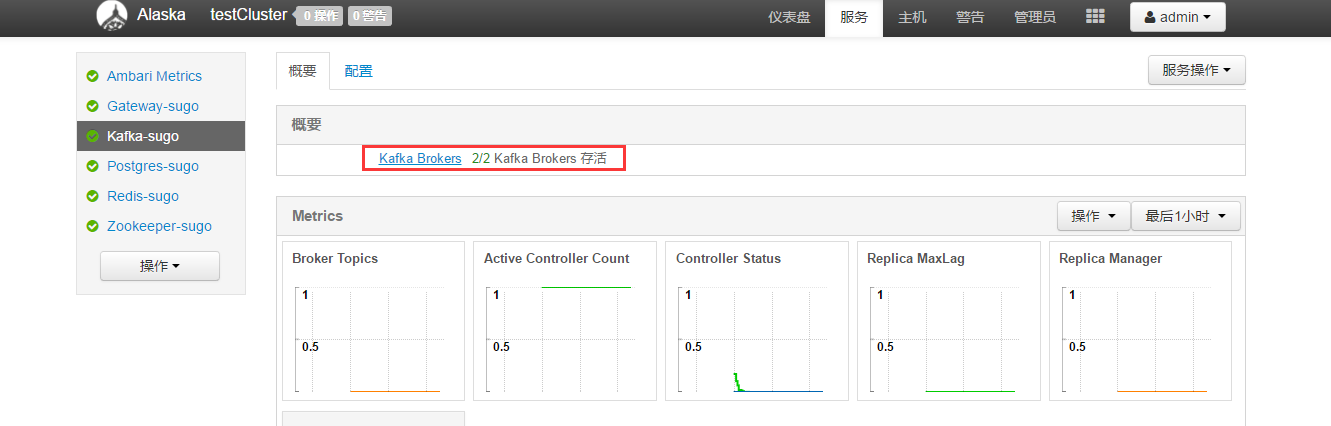
**第1步：添加kafka服务的组件kafka broker**

选择新添加的主机，如此处的test3.sugo.vm，点击增加，选择Kafka Broker，如下图所示：



**图48 增加Kafka Broker**

此时，返回kafka主界面，会看到Kafka Broker从1个变成了2个，如下图所示：



**图49 增加Kafka Broker后的Kafka主界面**

第2步：迁移数据

手工将test1.sugo.vm主机上Kafka的数据迁移至test3.sugo.vm上

第3步：删除旧的Kafka Broker

进入到Ambari界面旧主机（此处为test1.sugo.vm），停止该主机的Kafka Broker，如下图所示：



**图50 停止旧主机的Kafka Broker**

正确停止该Kafka Broker后，点击该选项的删除按钮，删除完成后即将Kafka从test1.sugo.vm迁移到test3.sugo.vm上。如果只需要添加Kafka Broker，省略第3步。

6.4 切换HDFS

在集群上安装了数果的HDFS，想要将数据存储到已有的HDFS集群的情况下，可以切换HDFS。需要注意的是：切换HDFS前，DruidIO存储在原来的HDFS上的数据需要导入到新的HDFS上，此过程需要手动操作；切换后的HDFS集群与已有的DruidIO集群网络需要能够互通。切换HDFS的具体操作如下：

第1步：准备

配置HDFS集群的IP与hostname的映射关系到DruidIO集群各主机的/etc/hosts文件

检查新的HDFS集群上是否存在目录DruidIO的相关目录，没有则创建，并将原HDFS上的数据导入到新的HDFS集群对应的目录下，DruidIO相关目录如下(权限755，用户及组druid:druid)：

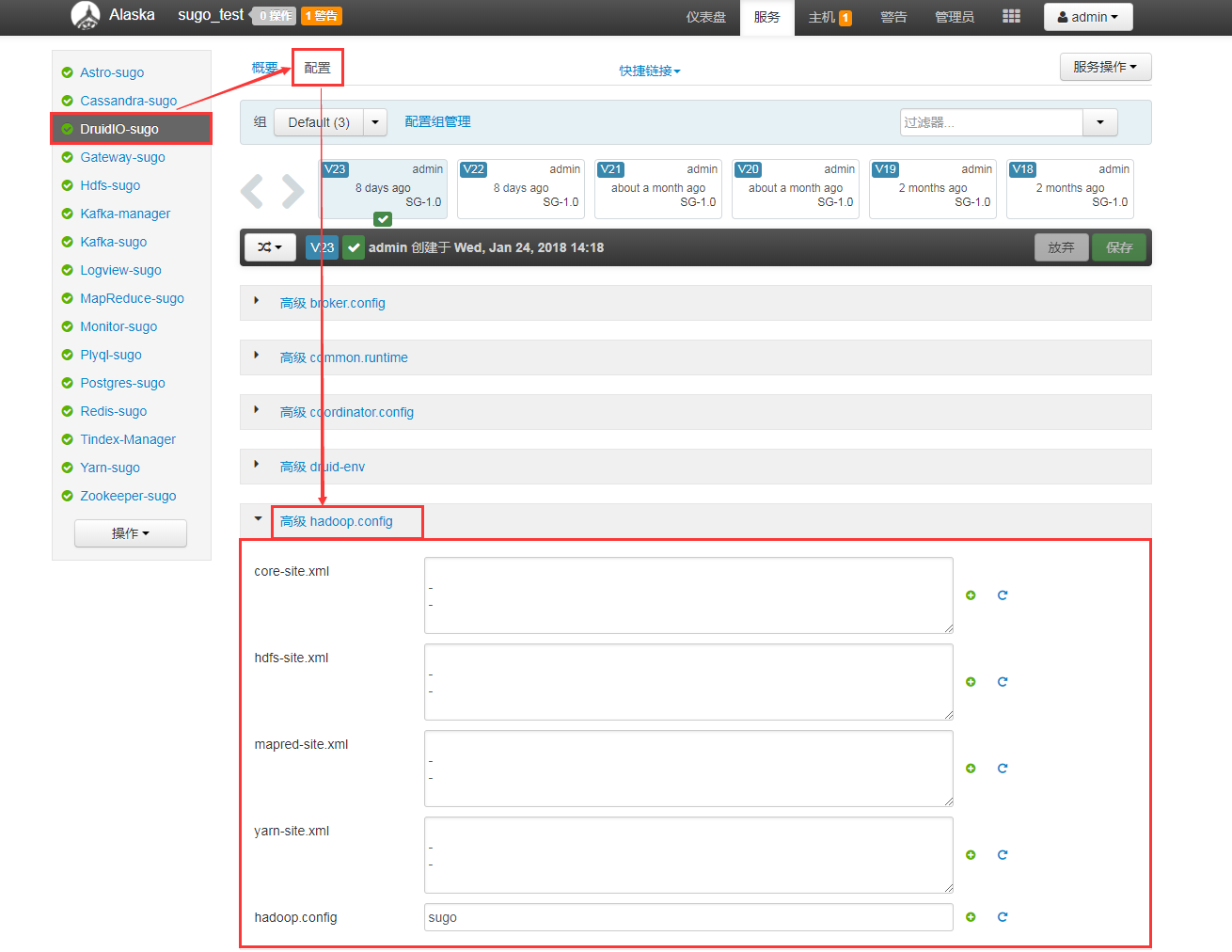
/druid/

/user/druid

第2步：停止服务DruidIO

第3步：修改配置

修改DruidIO的配置“高级hadoop.config”，修改配置项hadoop.config的值为“sugo”以外的其它值，如公司名称，修改core-site.xml、hdfs-site.xml、mapred-site.xml、yarn-site.xml，修改为切换后的HDFS的相关配置内容，如下图所示。



**图51 修改druidio配置**

第4步：启动服务DruidIO

**7. 附录：**

7.1 ASTRO配置说明

Alaska界面配置参数时，如下所示，如果参数有多层级，每一级以点(.)表示，如：

参数docUrl需填写为：site. docUrl=<http://docs.sugo.io>

site: {

//\* 通常无需特别配置，环境变量，由运行程序决定，

env: process.env.NODE\_ENV || 'development',

//\*\* 通常由扩展包决定，帮助网站链接

docUrl: 'http://docs.sugo.io',

//\*\* 通常由扩展包决定，版权信息，为空则不显示

copyrightText: '© 广东数果',

//\*\* 通常由扩展包决定，登录页版权信息

copyrightTextLogin: '© 2016 ~ 2017 广东数果科技',

//\*\* 通常由扩展包决定，siteName

siteName: '数果智能',

//\*\* 通常由扩展包决定，网站图标，例子 favicon: 'static/images/temp-wxj-favicon.ico',

favicon: 'favicon.ico',

//\*\* 通常由扩展包决定，登录页面logo文件名字,

loginLogoName: '/\_bc/sugo-analytics-static/assets/images/logo.png',

//\*\* 通常由扩展包决定，左侧菜单logo名字

mainLogoName: '/\_bc/sugo-analytics-static/assets/images/logo.svg',

//\*\* 通常由扩展包决定，csv上传限制, 默认10g

csvUploadLimit: 1024 \* 1024 \* 1024 \* 10,

//\*\*\* 需要根据环境配置，数据采集网关，csv上传和sdk数据上传的地址

collectGateway: 'http://collect.sugo.io',

//\*\*\* 需要根据环境配置， sdk 静态资源地址，通常要使用ip端口或者域名以供对外访问

websdk\_app\_host: 'localhost:8080',

//\*\*\* 需要根据环境配置 websocket 端口（埋点使用）

sdk\_ws\_port: 8887,

//\* 通常无需特别配置，cdn网站静态资源url，例如 `http://cdn.sugo.io`, 默认不使用

//cdn: ,

//websdk相关

//\* 通常无需特别配置，sdk数据上报地址，如果不配置 默认使用collectGateway

websdk\_api\_host: '@{site.collectGateway#getHost}',

//\* 通常无需特别配置，可视化埋点接口地址，如果为空，则使用 `websdk\_app\_host`

websdk\_decide\_host: '@{site.websdk\_app\_host}',

//\* 通常无需特别配置 web\_sdk cdn配置，如果为空，则使用 site.cdn || site.websdk\_app\_host

//, modifier cdnOrWebsdkAppHost 参考 src/server/config/config-template-parser.js

websdk\_js\_cdn: '@{#cdnOrWebsdkAppHost}',

//\* 通常无需特别配置 进入自助分析后首先根据 maxTime 查询数据

show\_newest\_data\_at\_first: true,

//\* 通常无需特别配置 iOS可视化埋点渲染器, 可选项有'Standard', 'Infinitus'等

iOS\_renderer: 'Standard',

//\* 通常无需特别配置 sdk埋点用到的 websocket url，默认使用 websdk\_app\_host和sdk\_ws\_port端口自生成配置

//modifier noPort 参考 src/server/config/config-template-parser.js

sdk\_ws\_url: 'ws://@{site.websdk\_app\_host#noPort}:@{site.sdk\_ws\_port}',

//\* 通常无需特别配置 业务通用websocket url，默认使用 websdk\_app\_host和ws\_port端口自动生成配置

ws\_url: 'ws://@{site.websdk\_app\_host#noPort}:@{site.ws\_port}',

//\*\* 通常由扩展包决定 自助分析下载限制

downloadLimit: '100,500',

//\*\* 通常由扩展包决定 批量下载限制

batchDownloadLimit: '100,500,1000',

//\*\* 通常由扩展包决定 自助分析默认时间，只能填相对时间 '-1 day', '-15 min' 等

analyticDefaultTime: '-1 day',

//\*\* 通常由扩展包决定 qq客服链接 没有则不显示

qqCustomerServiceUrl: '',

//\*\* 通常由扩展包决定 客服电话号码 没有则不显示

customerServicePhoneNumber: '',

//\*\* 通常由扩展包决定 动态帮助图标，默认不显示

showHelpLink: false,

//\*\* 通常由扩展包决定 自助分析书签功能

analyticBookmarkEnable: false,

//\*\* 通常由扩展包决定 维度 distinct 下拉框的查询限制

distinctDropDownFirstNLimit: 10

//\*\*\* 需要根据环境配置 程序运行监听的地址，如果在nginx反向代理后面，可以只监听localhost

host: '0.0.0.0',

//\*\*\* 需要根据环境配置 程序运行监听的端口

port: process.env.PORT || 8000,

//\*\*\* 需要根据环境配置 redis配置，用于共享session等

redis: {

host: '192.168.0.212',

port: 6379

},

// postgres数据库配置

//\*\*\* 需要根据环境配置, 请仅仅配置 host,database,username,password,port

//

db: {

dialect: 'postgres',

host: '192.168.0.210',

database: 'dev\_sugo\_astro',

username: 'postgres',

password: '123456',

pool: {

max: 5,

min: 0,

idle: 10000

},

port: 5432,

define: {

charset: 'utf8',

timestamps: false // true by default

}

},

//\*\*\* 需要根据环境配置 druid查询配置

//请仅仅配置 host, supervisorHost, lookupHost

//ipLibSpec要看实际情况是否需要

druid: {

host: '192.168.0.212:8082', // Tindex => brokder查询服务(可配置多个逗号隔开)

engine: 'sugoDruid',

timeout: 60000, // Tindex查询超时时间设定；默认一分钟，减小后端队列

verbose: '', // true=是否开启Tindex查询json日志

exactResultsOnly: true, // 开启Tindex精准查询，group by

groupByStrategy: 'v2',

firstN: true, // 默认用firstN查询替换topN查询

supervisorHost: 'http://192.168.0.212:8090', // Tindex => overload服务

lookupHost: 'http://192.168.0.212:8081', // Tindex => coordinator服务

ipLibSpec: 'none' // 是否启用ip库解析默认为none=不启用，ip=启用

},

//\*\*\* 需要根据环境配置 kafka配置

kafka: {

zookeeperHost: '192.168.0.216:2181/kafka',

kafkaServerHost: '192.168.0.214:9092,192.168.0.216:9092'

},

//\* 通常无需特别配置 默认使用redis配置

dataConfig: {

hostAndPorts: '@{#getRedisConfig}',

clusterMode: false,

password: '@{#getRedisPassword}',

type: 'redis'

},

//\* 通常无需特别配置 后端请求超时时间 15秒默认

timeout: 15000,

//\*\* 通常由扩展包决定 是否记录访问日志到数据库。默认不记录

log2db: false,

//\*\* 通常由扩展包决定 redis缓存过期时间 默认60秒

redisExpire: 60,

//\*\* 通常由扩展包决定 druid查询缓存时间 默认六十分钟

druidCacheTime: 60 \* 60,

//\* 通常无需特别配置

//sdk 缓存大小

sdk\_ws\_max\_buffer\_byte: 4 \* 1024 \* 1024,

//\*\* 通常由扩展包决定

//项目容量，druid可以支持的项目数量(集群允许运行的最大任务数)

projectCapacity: 10