Apache Knox

# 1.Apache knox简介

Apache konx是和Apache hadoop集群交互的REST API Gateway。Knox Gateway为所有的Hadoop集群中所有的REST交互提供单点访问控制，为企业提供集群的管理、集成、监控、管理自动化及分析等，包括下面的功能：

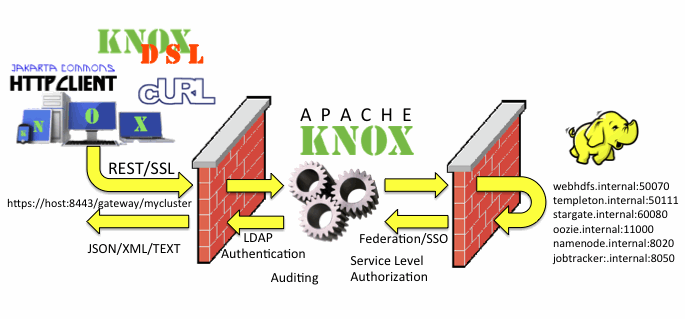
认证，包括LDAP和Active Directory 两种认证provider

Federation/SSO，统一认证系统，登录一次即可访问服务

Authorization，授权管理，Service level的授权

Auditing

访问流程如下所示：



部署不安全的Apache Hadoop Cluster的有很多的优势，Knox Gateway可以完善Kerberos Hadoop，Apache Knox为企业提供的方案有以下好处：

和企业已有的认证管理方案较好的结合

为用户隐藏集群的host和端口，包含集群部署的细节

简化客户端和多种服务的交互

Knox API Gateway以反向代理的形式向终端用户提供后端的服务，Policy定义在Topology deployment descriptor中，这些描述文件存放在topology目录中。

当前已经支持的服务包括Ambari、WebHDFS、Templeton(HCatalog)、Stargate(HBase)、Oozie、Hive/JDBC、Yarn RM和Storm。

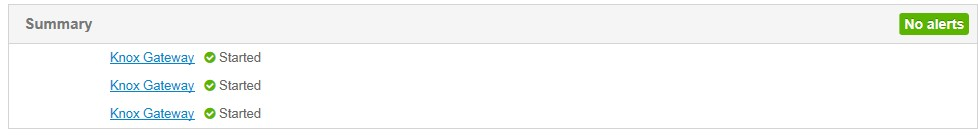
支持的Apache Hadoop UIs包括，Name Node UI、Job History UI、Oozie UI、HBase UI、YARN UI、Spark UI、Ambari UI及Ranger Admin Console。

Apache knox GateWay实现目标如下：

1. 在外层提供认证和token verification
2. 与Enterprise和Cloud identity 管理系统结合
3. 在外围提供service level的授权机制
4. 将Hadoop cluster的Aggregate REST APIs融合成单URL hierarchy，类似于Firewall holes的功能，限制Network endpoints来访问Hadoop cluster，使Cluster集群避免潜在的attackes

# 2.安装

1）使用Apache Ambari安装Knox后，服务如下：



2）配置如下：



主要的配置为：

1. gateway-site.xml

2.topologies

admin.xml

default.xml

3.users.ldif

3）安装目录为/usr/hdp/current/knox-server

|— bin  *//执行脚本和jar包*

|— conf —>/etc/knox/2.4.2.0-258/0  *//knox的全局配置*

|— conf/topologies *//用于部署集群代理的topology文件*

|— conf/users.ldif *//保存用户及密码*

|— conf/gateway-site.xml  *//knox的配置文件*

|— data —> /var/lib/knox/data-2.4.2.0-258 *//保存安全和topology specifice artifacts的信息*

|— /data/security *//放置master secret和keystore目录*

|— /data/security/keystores *//放置gateway和部署cluster toplogy的identity keystore 和credential store*

|—data/deployments *//放置部署的cluster topologies，用于控制hadoop cluster 的访问*

|—data/service，*//knox中部署的service，其中包括url的映射关系等*

|— /var/log/knox *//knox的运行日志*

|—pids -> /var/run/knox  *//knox的pid文件*

|—dep *//knox运行所依赖的jar包*

|— lib  *//knox服务组成的jar包*

|—ext *//knox的扩展jar包*

|—samples *//knox的样例，用于描述其运行流程*

|—templates *//默认配置文件及模板*

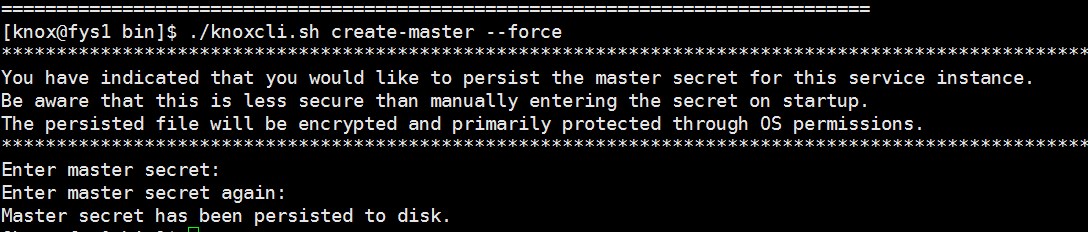
Apache Knox默认使用端口8443，Knox的使用首先启动embedded的LDAP。

# 3.Apache knox使用初探

1）创建Master Secret

bin/knoxcli.sh create-master

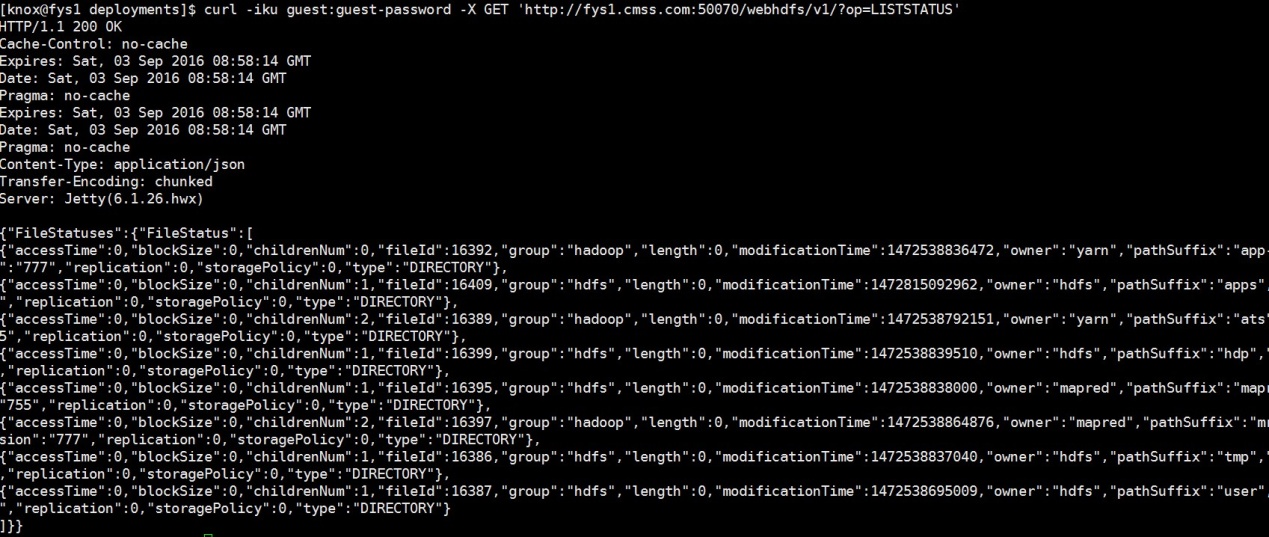
该命令用于将gateway instance的key和credential的master secret持久化到系统中，该命令的执行会提示输入master secret。



在测试中，输入密码为: qwe123

2）Webhdfs的访问

hadoop正常启动后，通过50070端口获取文件系统，如下所示：

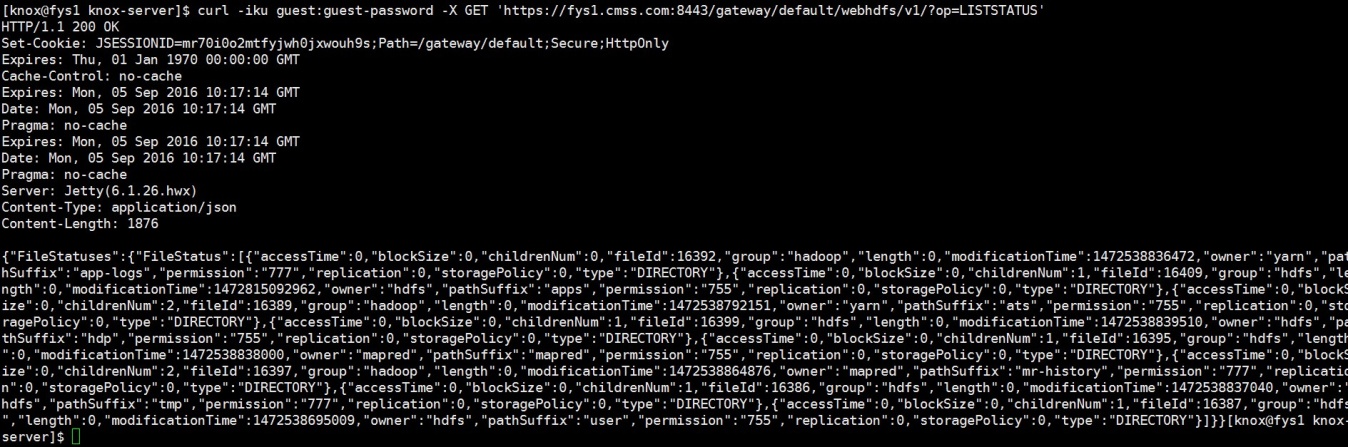


通过apache knox访问，使用8443端口作为代理，命令如下：

curl -iku guest:guest-password

-X GET 'https://fys1.cmss.com:8443/gateway/default/webhdfs/v1/?op=LISTSTATUS'

结果如下：



3) 运行机制分析

Gateway的URL格式如下：

https://{GATEWAY-HOST}:{PORT}/gateway/{SERVICE}/{URL}

1. GATEWAY-HOST:PORT

Knox Server启动主机和端口，端口默认是8443

1. SERVICE

服务名

1. URL

服务使用的URL

例子如下：https://fys1.cmss.com:8443/gateway/default/webhdfs/v1/?op=LISTSTATUS

webhdfs相关的service数据如下

1. /etc/knox/conf/topologies/default.xml，该文件为服务集default的配置文件，default看到是一个集群的名字，会在knox/data/deployments生成最新的war包，配置描述如下所示：

<topology> *//描述service集的topology*

<gateway>

<provider>....</provider> //保证web app的security provider，通过params来定义

<provider>....</provider>

</gateway>

<service>....</service> *//具体的服务*

<service>....</service>

</topology>

provider的定义案例如下(需要细化)：

<provider>

<role>authentication</role>

<name>ShiroProvider</name>

<enabled>true</enabled>

<param>

<name>sessionTimeout</name>

<value>30</value>

</param>

<param>

<name>main.ldapRealm</name>

<value>org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm</value>

</param>

<param>

<name>main.ldapRealm.userDnTemplate</name>

<value>uid={0},ou=people,dc=hadoop,dc=apache,dc=org</value>

</param>

<param>

<name>main.ldapRealm.contextFactory.url</name>

<value>ldap://fys1.cmss.com:33389</value>

</param>

<param>

<name>main.ldapRealm.contextFactory.authenticationMechanism</name>

<value>simple</value>

</param>

<param>

<name>urls./\*\*</name>

<value>authcBasic</value>

</param>

</provider>

service的案例如下：

<service>

<role>WEBHDFS</role>

<url>http://fys1.cmss.com:50070/webhdfs</url> //服务的具体URL

</service>

1. data/services/{app-name}/{version}，该目录用于描述url的代理关系，包括两个文件

a) service.xml，具备以下内容(以webhdfs为例)：

<service role="" name="" version="">

<routers>

<route path=""> <rewrite></rewrite></route>

<route>......</route>

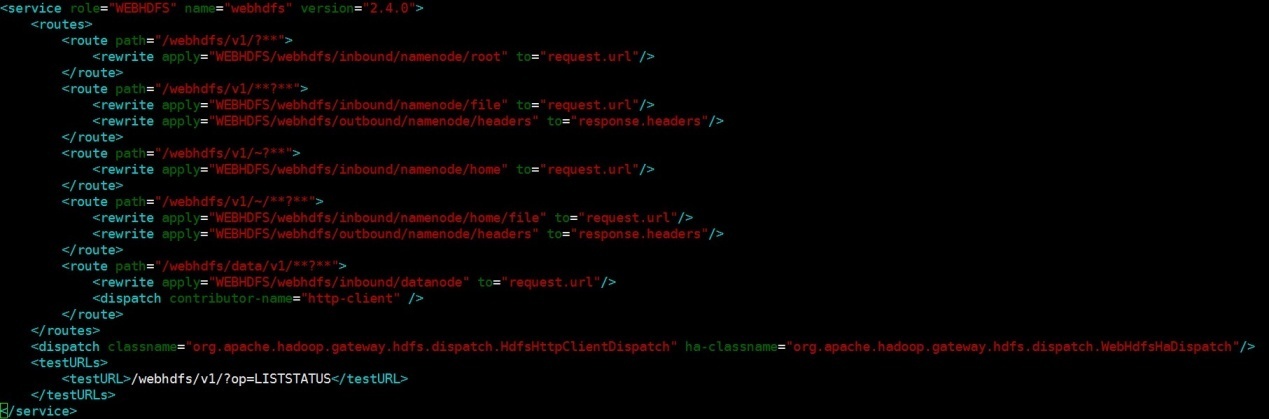
</route>

<dispatch></dispatch>

<testURLs><testURLs>

</routes>

webhdfs的案例如下：



b) rewrite.xml

<rules>

<rule dir="OUT" name="" pattern="">

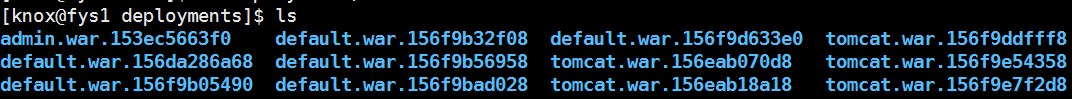
<rewrite></rewrite>

</rule>

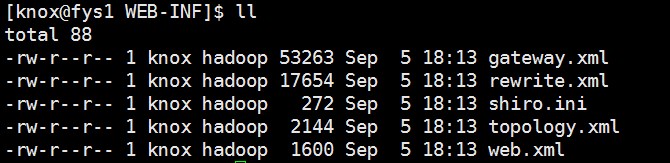
</rules>

3. /data/deployments，定义了上面的描述文件后，会在这个目录下生成war文件

如下所示：



war包中的文件如下所示：



# 4.通过Apache knox向RM中提交程序

# 5.Apache knox作为HTTP代理

Apache knox代理任何http服务或者web app，不仅仅局限于代理hadoop的服务或者Web应用/界面。例如可以把tomcat装在一台集群上，然后使用Knox作为代理，作为tomcat的访问点

5

# 5. proxy 机制源码解析

3种接口，hdfs,rpc,http，举例如下：

<service>

<role>NAMENODE</role>

<url>hdfs://fys1.cmss.com:8020</url>

</service>

<service>

<role>JOBTRACKER</role>

<url>rpc://fys2.cmss.com:8050</url>

</service>

<service>

<role>WEBHDFS</role>

<url>http://fys1.cmss.com:50070/webhdfs</url>

</service>

# 6. High Availability with Apache HTTP Server + mod\_proxy + mod\_proxy\_balancer

参考文献：

http://pivotalhd.docs.pivotal.io/docs/knox-gateway-administration-guide.html

http://knox.apache.org/books/knox-0-9-1/user-guide.html#Quick+Start

http://blog.csdn.net/tonyhuang\_google\_com/article/details/50038165 支持HTTP代理