

基于Cloudera的 Hive+Impala+Hue 集成 Ldap和Sentry 完成认证和授权

V_1.0

一. 安装LDAP服务端

1. 安装

```
# yum install -y openldap-*
```

2. 配置

拷贝ldap配置文件到ldap目录

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

创建ldap管理员密码

```
# slappasswd
New password:
Re-enter new password:
{SSHA}scEXP4oMUugxo9v0hgDpNkoMMDageLuV
```

输入保存管理员密码，返回的是加密后的一串密文

编辑配置文件

```
# vim /etc/openldap/slapd.conf
```

注：这里组织的域为 bigdata.com，修改对应如下内容my-domain替换为bigdata，rootpw中设置密码：

```
# enable server status monitoring (cn=monitor)
database monitor
access to *
    by dn.exact="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read
    by dn.exact="cn=Manager,dc=bigdata,dc=com" read
    by * none

#####
# database definitions
#####

database      bdb
suffix        "dc=bigdata,dc=com"
checkpoint    1024 15
rootdn        "cn=Manager,dc=bigdata,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
# rootpw      secret
rootpw        {SSHA}scEXP4oMUugxo9v0hgDpNkoMMDageLuV #加密后的管理员密码
```

拷贝DB_CONFIG文件到指定目录

```
# cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

删除默认/etc/openldap/slapd.d下面的所有内容

```
# rm -rf /etc/openldap/slapd.d/*
```

赋予配置目录相应权限

```
# chown -R ldap:ldap /var/lib/ldap
# chown -R ldap:ldap /etc/openldap/
# service slapd start
```

生成配置文件并赋值

```
# slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d
```

config file testing succeeded

```
# chown -R ldap:ldap /etc/openldap/slapd.d/*
# service slapd restart
```

查看状态，验证服务端口：

```
# ps aux | grep slapd | grep -v grep
ldap      9225  0.0  0.2 581188 44576 ?        Ssl  15:13   0:00 /usr/sbin/slapd -h ldap:/// -u ldap

$ netstat -tunlp | grep :389
tcp        0      0 0.0.0.0:389          0.0.0.0:*           LISTEN      8510/slapd
tcp        0      0 :::389              :::*                 LISTEN      8510/slapd
```

配置开机启动 LDAP 服务：

```
chkconfig --add slapd
chkconfig --level 345 slapd on
```

3. LDAP 的使用

导入系统用户

接下来你可以从 /etc/passwd, /etc/shadow, /etc/groups 中生成 ldif 更新 ldap 数据库，这需要用到 migrationtools 工具。

安装：

```
# yum install migrationtools -y
```

利用迁移工具生成模板，先修改默认的配置：

```
# vim /usr/share/migrationtools/migrate_common.ph

# default DNS domain (line 71)
$DEFAULT_MAIL_DOMAIN = "bigdata.com";
# default base (line 74)
$DEFAULT_BASE = "dc=bigdata,dc=com";
```

生成模板文件：

```
# /usr/share/migrationtools/migrate_base.pl > /tmp/base.ldif
然后，可以修改该文件，再执行导入命令：

# ldapadd -x -D "cn=manager,dc=bigdata,dc=com" -W -f /tmp/base.ldif
```

将当前节点上的用户导入到 ldap 中，可以有选择的导入指定的用户：

```
先添加用户
# useradd appuser hive
查找系统上的 appuser、hive 等用户
# grep -E "appuser|impala|hue|hive" /etc/passwd >/tmp/passwd.txt

转换
# /usr/share/migrationtools/migrate_passwd.pl /tmp/passwd.txt /tmp/passwd.ldif
```

导入

```
# ldapadd -D "cn=manager,dc=bigdata,dc=com" -W -f /tmp/passwd.ldif
```

将用户组导入到 ldap 中:

生成用户组的 ldif 文件, 然后导入到 ldap

```
# grep -E "appuser|impala|hue|hive" /etc/group >/tmp/group.txt
```

转换

```
# /usr/share/migrationtools/migrate_group.pl /tmp/group.txt /tmp/group.ldif
```

导入

```
# ldapadd -D "cn=manager,dc=bigdata,dc=com" -W -f /tmp/group.ldif
```

查询

```
# ldapsearch -LLL -x -D "cn=manager,dc=bigdata,dc=com" -W -b 'dc=bigdata,dc=com' 'uid=appuser'
```

修改, 用户添加好以后, 需要给其设定初始密码, 运行命令如下:

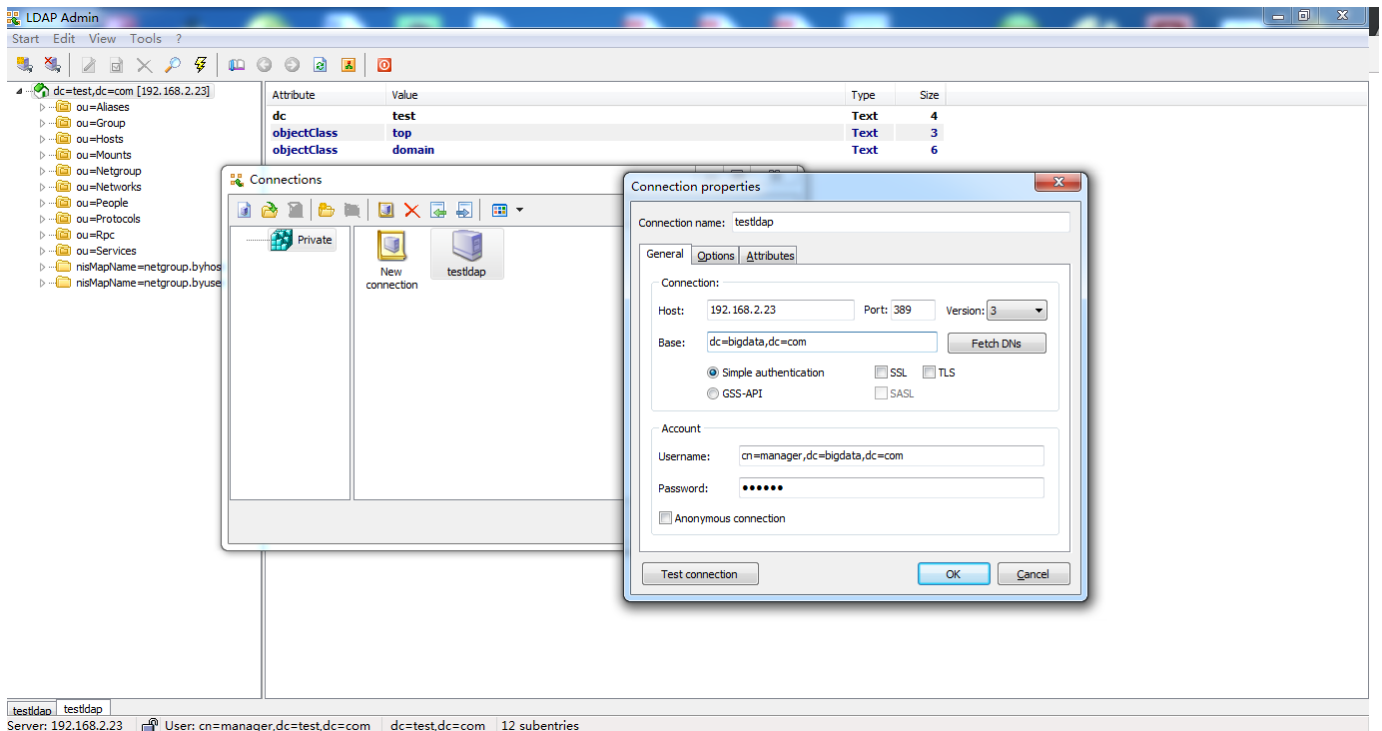
```
# ldappasswd -x -D "cn=manager,dc=bigdata,dc=com" -W "uid=appuser,ou=people,dc=bigdata,dc=com" -S
```

删除, 删除用户或组条目:

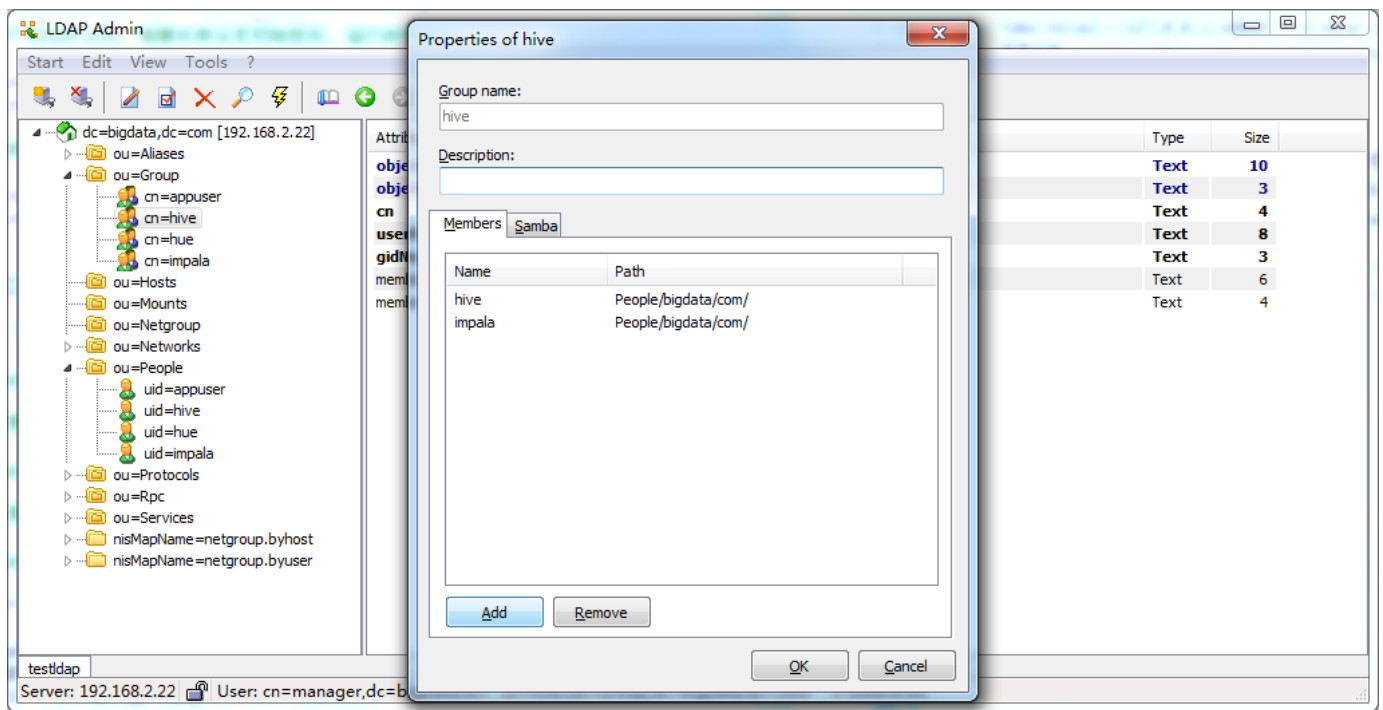
```
# ldapdelete -x -W -D "cn=manager,dc=bigdata,dc=com" "uid=hive,ou=people,dc=bigdata,dc=com"
```

4. ldap管理工具 ldapadmin

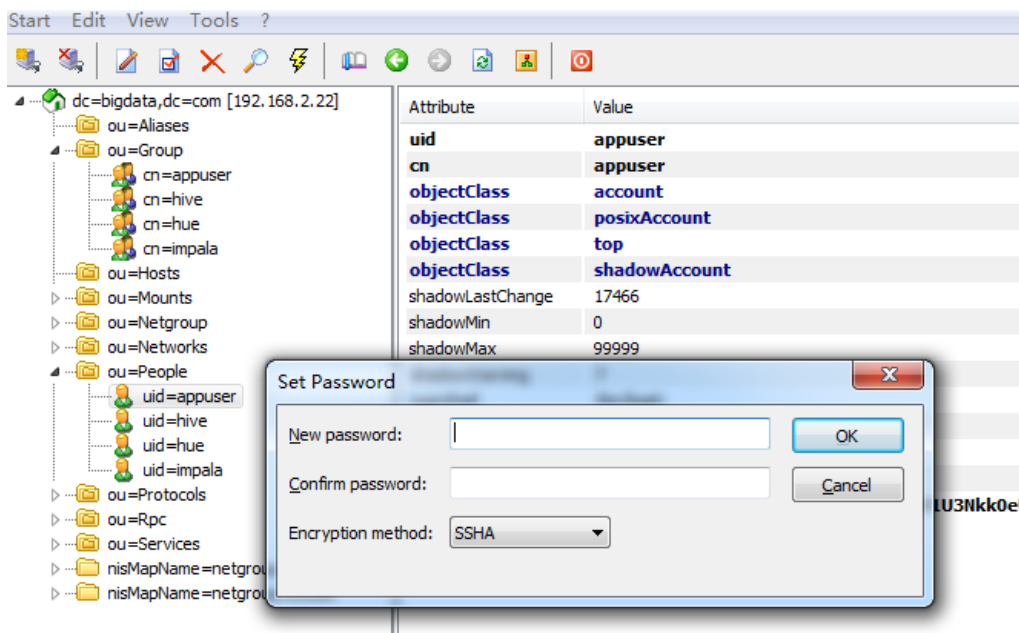
下载地址 <http://www.ldapadmin.org/download/ldapadmin.html>



在组中添加用户



为用户设置密码



二. LDAP认证实现

对HUE, impala-shell和beeline都要进行访问认证,所以需要分别在HUE, Impala和Hive中分别实现LDAP的集成。但不管那个系统与LDAP的集成,不外乎都要配置三个基本的属性: 1) 告诉系统我要与LDAP集成。2) LDAP服务器的地址。3) LDAP的baseDN。

1. LDAP和Hive的集成

ldap

显示 2 个已抑制的警告

启用 LDAP 身份验证

Hive (服务范围)

LDAP URL

hive.server2.authentication.ldap.url

Hive (服务范围)

ldap://hadoop22.test.com

Active Directory 域

hive.server2.authentication.ldap.Domain

Hive (服务范围)

LDAP BaseDN

hive.server2.authentication.ldap.baseDN

Hive (服务范围)

ou=People,dc=bigdata,dc=com

抑制参数验证 : LDAP URL

Hive (服务范围)

抑制参数验证 : Active Directory Domain

Hive (服务范围)

抑制参数验证 : LDAP BaseDN

Hive (服务范围)

抑制配置验证程序 : Client TLS/SSL In Use With LDAP Authentication Validator

Hive (服务范围)

完成上面的配置后重新启动Hive，配置就生效了，以后如果通过beeline来链接hive，就需要提供用户名和密码了

验证

```
# beeline -u "jdbc:hive2://127.0.0.1:10000" -n hive -p 123qwe
正确用户和密码，登录成功

# beeline -u "jdbc:hive2://127.0.0.1:10000"
无用户和密码，登录失败，Unknown HS2 problem when communicating with Thrift server.

# beeline -u "jdbc:hive2://127.0.0.1:10000" -n hive -p 123456
错误用户和密码，登录失败，Unknown HS2 problem when communicating with Thrift server.
```

2. LDAP与Impala的集成

LDAP和Impala的集成与和Hive的集成非常的类似，除了Hive中提到的3个配置项之外，还需要多配一个配置项（-ldap_passwords_in_clear_ok=true），以告诉Impala密码可以通过明文来传播（如果你的环境中没有配置TSL并且又没有设置这个配置，impala将无法启动）

ldap

显示 2 个已抑制的警告

启用 LDAP 身份验证

enable_ldap_auth

☒ Impala (服务范围)

LDAP URL

ldap_uri

Impala (服务范围)

ldap://hadoop22.test.com

启用 LDAP TLS

ldap_tls

☐ Impala (服务范围)

Active Directory 域

ldap_domain

Impala (服务范围)

LDAP BaseDN

ldap_baseDN

Impala (服务范围)

ou=People,dc=bigdata,dc=com

LDAP 模式

ldap_bind_pattern

Impala (服务范围)

LDAP 服务器 CA 证书

ldap_ca_certificate

编辑单个值

Impala Daemon Default Group ...和另 1 个

抑制参数验证：LDAP Pattern

☐ Impala (服务范围)

抑制参数验证：LDAP URL

☒ Impala (服务范围)

抑制参数验证：Active Directory Domain

☐ Impala (服务范围)

抑制参数验证：LDAP BaseDN

☐ Impala (服务范围)

抑制配置验证程序：LDAP Configuration Validator

☒ Impala (服务范围)

抑制参数验证：LDAP Server CA Certificate

编辑单个值

☐ Impala Daemon Default Group ...和另 1 个

Impala 命令行参数高级配置代码段 (安全阀)

Impala (服务范围)

-ldap_passwords_in_clear_ok=true

验证命令：

```
impala-shell -i impalad-server -u impala -l --auth_creds_ok_in_clear
-i 集群中任意一台impalad服务器都可以
-u 登录用户
-l 使用ldap
--auth_creds_ok_in_clear 由于没有使用ssl，需要添加该参数。
```

如果需要免密登录，需要参数 --ldap_password_cmd

```
impala-shell -i 127.0.0.1 -u impala -l --ldap_password_cmd="echo -n '123qwe'" --auth_creds_ok_in_clear
```

3. LDAP和HUE的集成

LDAP和HUE的集成依然必须包含那三个关键的要素：告诉HUE要启动LDAP认证，LDAP的Server地址以及baseDN

告诉Hue使用LDAP来做认证

backend=desktop.auth.backend.LdapBackend

backend

用户扩增器

user_augmentor

Hue (服务范围)

desktop.auth.backend.DefaultUserAugmentor

身份验证后端

backend

Hue (服务范围) ↺

desktop.auth.backend.LdapBackend

ldap登陆用户的模板，username运行时被替换

```
ldap_username_pattern="uid=<username>,ou=people,dc=bigdata,dc=com"
```

ldap|

身份验证后端

backend

Hue (服务范围) ↺

desktop.auth.backend.LdapBackend

LDAP URL

ldap_url

Hue (服务范围) ↺

ldap://hadoop22.test.com

LDAP 服务器 CA 证书

ldap_cert

Hue (服务范围)

启用 LDAP TLS

use_start_tls

☐ Hue (服务范围) ↺

LDAP 用户名模式

ldap_username_pattern

Hue (服务范围) ↺

uid=<username>,ou=people,dc=bigdata,dc=com

使用搜索绑定身份验证

search_bind_authentication

☐ Hue (服务范围)

登录时创建 LDAP 用户

create_users_on_login

☒ Hue (服务范围)

LDAP 搜索基础

base_dn

Hue (服务范围) ↺

dc=bigdata,dc=com

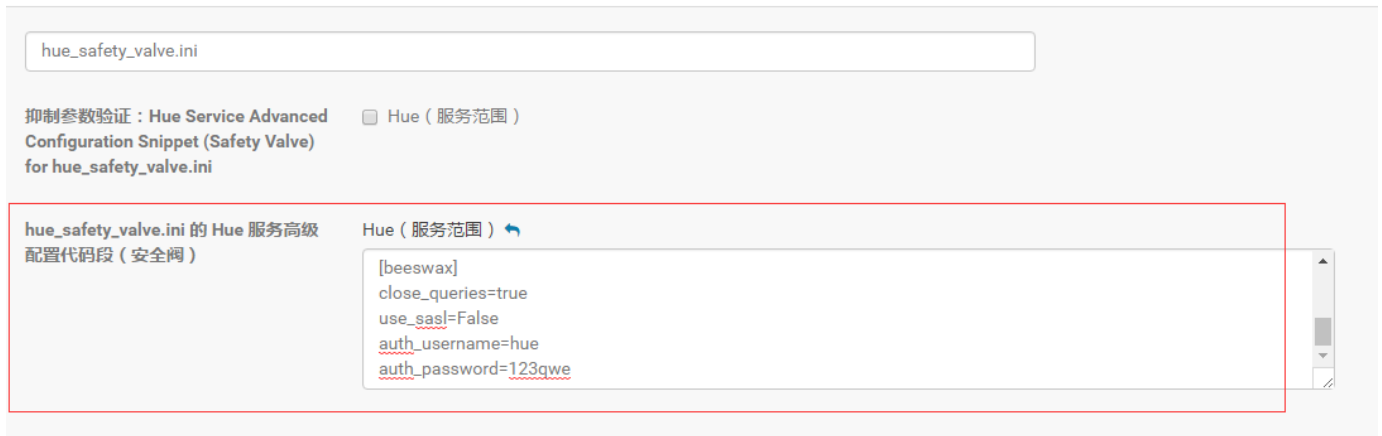
其他snippet，用户登录Hue时，需要使用一个预设的用户名和密码去连接hive/impala，连接成功之后，当真正执行QL的时候，还使用登录时的账号来做鉴权。

hue_safety_valve.ini 的 Hue 服务高级配置代码段（安全阀）中添加以下内容

```
[impala]
impersonation_enabled=true
auth_username=hue
auth_password=123qwe

[beeswax]
close_queries=true
use_sasl=false
auth_username=hue
auth_password=123qwe
```

截图如下

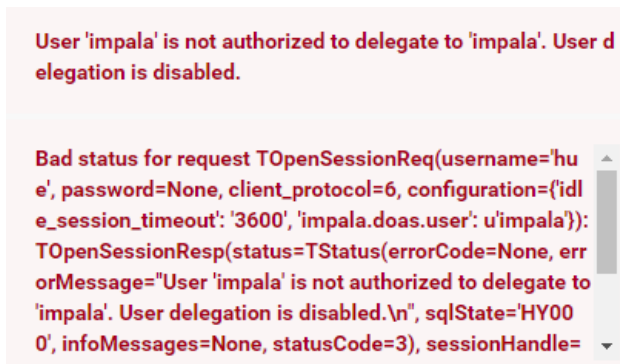


如果登录Hue后出现以下错误提示:

Then you might hit this error:

User 'hue' is not authorized to impersonate 'impala'. User impersonation is disabled.
This is because Hue is not authorized to be a proxy. To fix it, startup Impala with this flag:

如图:



需要到impala中配置项“Impala Daemon 命令行参数高级配置代码段（安全阀）”添加参数

```
--authorized_proxy_user_config=hue=*
```

如图:

clouderaMANAGER

群集

主机

诊断

审核

图表

管理

Impala (Cluster 1)

操作

状态

实例

配置

命令

查询

图表库

最佳做法

审核

Web UI

快速链接

筛选器

范围

Impala (服务范围)1

Impala Catalog Server0

Impala Daemon2

Impala LLama ApplicationMaster0

Impala StateStore0

类别

Admission Control0

主要1

基于政策文件的 Sentry0

堆栈集合0

安全性0

性能0

抑制0

日志0

监控0

端口和地址0

资源管理0

高级2

Impala Daemon 命令行

显示 2 个已抑制的警告

Impala Daemon 查询选项高级配置代码段 (安全阀)

default_query_options

编辑单个值

Impala Daemon Default Group ...和另 1 个

Impala 命令行参数高级配置代码段 (安全阀)

Impala (服务范围)

-ldap_passwords_in_clear_ok=true

Impala Daemon 命令行参数高级配置代码段 (安全阀)

Impala Daemon Default Group ...和另 1 个

-use_local_tz_for_unix_timestamp_conversions=true

-authorized_proxy_user_config=impala=*
























至此，LDAP和HUE，Impala，Hive的整合就完成了。用户访问HUE，Impala以及Hive都需要提供用户名和密码了。

三. SENTRY授权实现

1. 安装sentry

将服务添加到 Cluster 1

选择您要添加的服务类型。

服务类型	说明
 Accumulo	The Apache Accumulo sorted, distributed key/value store is a robust, scalable, high performance data storage and retrieval system. This service only works with releases based on Apache Accumulo 1.6 or later.
 Flume	Flume 从几乎所有来源收集数据并将这些数据聚合到永久性存储（如 HDFS）中。
 HBase	Apache HBase 提供对大型数据集的随机、实时的读/写访问权限（需要 HDFS 和 ZooKeeper）。
 HDFS	Apache Hadoop 分布式文件系统 (HDFS) 是 Hadoop 应用程序使用的主要存储系统。HDFS 创建多个数据块副本并将它们分布在整个群集的计算机上，以后用可靠且极其快速的计算功能。
 Hive	Hive 是一种数据仓库系统，提供名为 HiveQL 的 SQL 类语言。
 Hue	Hue 是与包括 Apache Hadoop 的 Cloudera Distribution 配合使用的图形用户界面(需要 HDFS、MapReduce 和 Hive)。
 Impala	Impala 为存储在 HDFS 和 HBase 中的数据提供了一个实时 SQL 查询接口。Impala 需要 Hive 服务，并与 Hue 共享 Hive Metastore。
 Isilon	EMC Isilon is a distributed filesystem.
 Java KeyStore KMS	The Hadoop Key Management Service with file-based Java KeyStore. Maintains a single copy of keys, using simple password-based protection. Requires CDH 5.3+. Not recommended for production use.
 Kafka	Apache Kafka is publish-subscribe messaging rethought as a distributed commit log. Before adding this service, ensure that either the Kafka parcel is activated or the Kafka package is installed.
 Key-Value Store Indexer	键/值 Store Indexer 侦听 HBase 中所含表内的数据变化，并使用 Solr 为其创建索引。
 Kudu	Kudu is a true column store for the Hadoop ecosystem.
 MapReduce	Apache Hadoop MapReduce 支持对整个群集中的大型数据集进行分布式计算（需要 HDFS）。 建议改用 YARN（包括 MapReduce 2）。包括 MapReduce 用于向后兼容性。
 Oozie	Oozie 是群集中管理数据处理作业的工作协调服务。
 S3 Connector	The S3 Connector Service securely provides a single set of AWS credentials to Impala and Hue. This enables Hue administrators to browse the S3 filesystem and define Impala tables backed by S3 data authorized to that AWS identity, and also enables Impala users to query S3-backed tables without directly providing AWS credentials, subject to having the proper permissions defined via Sentry. The S3 Connector only supports the S3A protocol.
 Sentry	Sentry 服务存储身份验证凭证元数据并为客户提供对该元数据的开发安全访问。
 Solr	Solr 是一个分布式服务，用于编制存储在 HDFS 中的数据的索引并搜索这些数据。
 Spark	Apache Spark is an open source cluster computing system. This service runs Spark as an application on YARN.
 Spark (Standalone)	Apache Spark is an open source cluster computing system. This is the standalone version of the service which does not use YARN for resource management. Cloudera recommends using Spark on YARN instead of this standalone version.
 Sqoop 1 Client	Configuration and connector management for Sqoop 1.
 Sqoop 2	Sqoop 是一个设计用于在 Apache Hadoop 和结构化数据存储（如关系数据库）之间高效地传输大批量数据的工具。Cloudera Manager 支持的版本为 Sqoop 2。
 YARN (MR2 Included)	Apache Hadoop MapReduce 2.0 (MRv2) 或 YARN 是支持 MapReduce 应用程序的数据计算框架（需要 HDFS）。
 ZooKeeper	Apache ZooKeeper 是用于维护和同步配置数据的集中服务。

将 Sentry 服务添加到 Cluster 1

自定义 Sentry 的角色分配

您可以在此处自定义新服务的角色分配，但请注意，如果分配不正确（例如，分配到某个主机上的角色太多），性能受到影响。

还可以按主机查看角色分配。[按主机查看](#)

SS Sentry Server × 1 新建

hadoop22.test.com

G Gateway

选择主机

将 Sentry 服务添加到 Cluster 1

数据库设置

配置和测试数据库连接。首先根据[Installation Guide](#)的[Installing and Configuring an External Database](#)小节创建数据库。

Sentry

Successful

数据库主机名称: *

数据库类型:

数据库名称: *

用户名: *

密码:

hadoop22.test.com

MySQL

sentry

sentry

☐ 显示密码

测试连接

备注:

- 创建数据库时，数据库主机名称 字段中的值必须与您用于主机名称的值匹配。[了解更多](#)
- 如数据库未在其默认端口运行，请使用 数据库主机名称 字段中的 host:port 指定端口号。
- 强烈建议将各个数据库与相应角色实例置于同一主机上。

将 Sentry 服务添加到 Cluster 1

首次运行 命令

状态 ✔ 已完成 11月 1, 5:17:09 下午 63.08s

Finished First Run of the following services successfully: Sentry.

✔ 已完成 4 个步骤 (共 4 个)。

☒ Show All Steps ☐ Show Only Failed Steps ☐ Show Only Running Steps

✔ Ensuring that the expected software releases are installed on hosts. 已成功完成 1 个步骤。		11月 1, 5:17:09 下午	370ms
✔ 正在部署客户端配置 Successfully deployed all client configurations.	Cluster 1	11月 1, 5:17:11 下午	20.16s
✔ 正在创建 Sentry 数据库表 Successfully created Sentry database tables.	Sentry	11月 1, 5:17:31 下午	16.14s
✔ 启动 Sentry Successfully started service.	Sentry	11月 1, 5:17:47 下午	25.09s

[返回](#)

1 2 3 4 5 6

[继续](#)

2. 配置Sentry服务

1). 设置hive数据仓库目录权限, 目前使用的是默认路径

```
$ sudo -u hdfs hdfs dfs -chmod -R 771 /user/hive/warehouse
$ sudo -u hdfs hdfs dfs -chown -R hive:hive /user/hive/warehouse
```

2) 禁用 HiveServer2 的模拟功能

cloudera MANAGER

群集 ▾ 主机 ▾ 诊断 ▾ 审核 图表 ▾ 管理 ▾

✔ Hive (Cluster 1) 操作 ▾

状态 实例 配置 命令 图表库 审核 [HiveServer2 Web UI](#) 快速链接 ▾

筛选器 清除全部

▼ 范围 清除

Hive (服务范围) 11

Gateway 1

Hive Metastore Server 2

HiveServer2 2

WebHCat Server 0

▼ 类别 清除

Hive Metastore 数据库 0

Sentry HDFS 同步缓存 0

主要 2

搜索

显示 2 个已抑制的警告

HiveServer2 Load Balancer

HiveServer2 启用模拟

hive.server2.enable.impersonation,
hive.server2.enable.doAs

HiveServer2 Default Group

☐ HiveServer2 Default Group ↻

显示 25 每页

3) 在HUE, Hive和Impala中都做如下的配置:

Sentry 服务

Hive (服务范围)

☒ Sentry
 ☐ none

抑制参数验证 : Hive Service Advanced Configuration Snippet (Safety Valve) for sentry-site.xml

☐ Hive (服务范围)

sentry-site.xml 的 Hive 服务高级配置代码段 (安全阀)

Hive (服务范围)

名称

sentry.hive.testing.mode

值

true

说明

说明

☐ 最终

+

以

这两个配置中，第一个配置告诉系统（hue，hive，impala）要使用sentry来做权限管理 第二个配置把testing mode设置为true，如果为false，hive会要求必须使用ssl来进行密码传输，我们的环境并没有配置ssl

4) 配置LDAP组映射

```
hadoop.security.group.mapping.ldap.url=ldap://hadoop22.test.com
hadoop.security.group.mapping.ldap.search.filter.user=(&(objectClass=posixAccount)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group=(objectClass=posixGroup)
hadoop.security.group.mapping.ldap.search.attr.member=memberUid
hadoop.security.group.mapping.ldap.search.attr.group.name=cn
hadoop.security.group.mapping.ldap.bind.user=cn=Manager,dc=bigdata,dc=com
hadoop.security.group.mapping.ldap.bind.password=123qwe
hadoop.security.group.mapping.ldap.base=dc=bigdata,dc=com
hadoop.security.group.mapping=org.apache.hadoop.security.LdapGroupsMapping
```

如图：

筛选器

范围

HDFS (服务范围)	22
Balancer	0
DataNode	0
Gateway	0
HttpFS	0
JournalNode	0
NFS Gateway	0
NameNode	0
SecondaryNameNode	0
Fallover Controller	0

类别

High Availability	0
主要	0
代理	0
堆栈集合	0
复制	0
安全性	12
性能	0
其他	10

日志	0
检查点	0
监控	0
端口和地址	0
资源管理	0
高级	0

状态

错误	0
警告	0
已编辑	0
非默认	8
包含覆盖项	0

ldap

Hadoop 用户组映射实现
hadoop.security.group.mapping

HDFS (服务范围)

- ☐ org.apache.hadoop.security.JniBasedUnixGroupsMapping
- ☐ org.apache.hadoop.security.ShellBasedUnixGroupsMapping
- ☒ org.apache.hadoop.security.LdapGroupsMapping

Hadoop 用户组 进程ping LDAP URL
hadoop.security.group.mapping.ldap.url

HDFS (服务范围)

ldap://hadoop22.test.com

Hadoop 用户组映射 LDAP TLS/SSL 已启用
hadoop.security.group.mapping.ldap.use.ssl☐ HDFS (服务范围)**Hadoop 用户组映射 LDAP TLS/SSL Truststore**
hadoop.security.group.mapping.ldap.ssl.keystore

HDFS (服务范围)

Hadoop 用户组映射 LDAP TLS/SSL Truststore 密码
hadoop.security.group.mapping.ldap.ssl.keystore.password

HDFS (服务范围)

Hadoop 用户组映射 LDAP 绑定用户可分辨名称
hadoop.security.group.mapping.ldap.bind.user

HDFS (服务范围)

cn=manager,dc=bigdata,dc=com

Hadoop 用户组 进程ping LDAP 绑定用户密码
hadoop.security.group.mapping.ldap.bind.password

HDFS (服务范围)

Hadoop 用户组 进程ping 搜索基础
hadoop.security.group.mapping.ldap.base

HDFS (服务范围)

dc=bigdata,dc=com

Hadoop 用户组 进程ping LDAP 用户搜索过滤器
hadoop.security.group.mapping.ldap.search.filter.user

HDFS (服务范围)

(&(objectClass=posixAccount)(uid={0}))

Hadoop 用户组 进程ping LDAP 组搜索过滤器
hadoop.security.group.mapping.ldap.search.filter.group

HDFS (服务范围)

(objectClass=posixGroup)

Hadoop 用户组 进程ping LDAP 组成员身份属性
hadoop.security.group.mapping.ldap.search.attribute.member

HDFS (服务范围)

memberUid

Hadoop 用户组 进程ping LDAP 组名称属性
hadoop.security.group.mapping.ldap.search.attribute.group.name

HDFS (服务范围)

cn

sentry授权命令

```
create role admin_role;
grant all on server server1 to role admin_role;
grant role admin_role to group impala;

create database testdb1;
create database testdb2;

create role test_role;
grant all on database testdb1 to role test_role;
grant role test_role to group appuser;
```

```
grant select on database testdb2 to role test_role;  
grant all on uri 'hdfs://hadoop22.test.com:8020/user/appuser/' to role test_role;
```

参考文章

<http://blog.csdn.net/u014728303/article/details/53908412>