Transwarp Data Hub Version 4.3.3
安全手册

星环信息科技（上海）有限公司

Transwarp Data Hub v4.3.3

安全手册

版本：1.0v

发布日期： 2016-03-26

版本号：T001433-09-010

免责声明

本说明书依据现有信息制作，其内容如有更改，恕不另行通知。星环信息科技（上海）有限公司在编写该说明书的时候已尽最大努力保证期内容准确可靠，但星环信息科技（上海）有限公司不对本说明书中的遗漏、不准确或印刷错误导致的损失和损害承担责任。具体产品使用请以实际使用为准。

注释：Hadoop® 和 SPARK® 是Apache™ 软件基金会在美国和其他国家的商标或注册的商标。

修订历史记录

文档版本T001433-09-010 (2016-03) 第一次发布。

目录

1. 简介

Transwarp Data Hub提供全系统的认证机制，同时提供服务对用户的授权机制，确保您的集群在多用户使用的情况下依然安全，并确保各用户之间资源的隔离。

1.1. 认证

从Transwarp Data Hub 4.2开始，我们推出一套新的安全认证系统：Kerberos with LDAP。相对于TDH4.2之前版本中的Kerberos认证，Kerberos with LDAP实现了Kerberos的HA（高可用性），同时Kerberos with LDAP还做到了Kerberos和LDAP中的用户使用同一个密码认证。

这里，为后文的介绍做准备，我们需要区分本手册中“认证系统”和“认证方式”两个词汇的使用。

- 认证系统： 我们指认证使用的协议和程序，比如Kerberos，LDAP或Kerberos with LDAP。
- 认证方式： 我们指通过某种认证系统认证的方法。比如要通过Kerberos认证系统的认证，您需要提供Kerberos认证所需的信息——Kerberos prinicpal和密码（或者keytab）。要通过LDAP认证，您需要提供LDAP认证所需的信息——LDAP用户名和密码。

在本手册中，我们将称部署并启用了认证系统（包括Kerberos，LDAP或者Kerberos with LDAP）的集群为“ **安全模式** ”下的集群。在安全模式下的集群，用户需要通过认证才能够使用服务。如果集群没有部署或者部署了但没有启用任何认证系统，那么我们称其为“非安全模式”下的集群。

需要指出的是，在Kerberos with LDAP认证系统中，除Inceptor外各服务认证方式的选择和TDH4.2之前版本的相同，即一个服务可以启用或者不启用Kerberos。但是Inceptor可以选择：

- 不启用安全认证（既不启用Kerberos也不启用LDAP）；
- 仅启用LDAP
- 仅启用Kerberos
- 既启用Kerberos，也启用LDAP。用户必须通过LDAP认证来使用Inceptor——即在此设置下，用户必须提供LDAP认证信息而不是Kerberos认证信息来通过认证。

对于整个TDH集群来说，安全认证设置有下面几种常见选择：

1. 集群使用简单认证模式——不启用任何认证系统，用户无需提供任何认证信息。
2. 集群使用Kerberos with LDAP认证系统，所有服务都启用Kerberos并且所有服务都使用Kerberos认证——即用户要提供Kerberos principal和密码（或者keytab）。
3. 集群使用Kerberos with LDAP认证系统，所有服务都启用Kerberos同时Inceptor启用LDAP认证。所有服务除Inceptor外使用Kerberos认证——即用户要提供Kerberos principal和密码（或者keytab），Inceptor使用LDAP认证——用户要提供LDAP用户名和密码。

1.2. 授权

一个Transwarp Data Hub集群中，提供授权机制的服务有三个：HDFS，Hyperbase和Inceptor，其中：

1. HDFS提供文件和目录的“rwx”权限控制。
2. Hyperbase提供不同用户在Hyperbase中的“RWC(Read, Write, Create)”权限控制。
3. Inceptor提供基于SQL的SELECT/CREATE/DELETE/UPDATE/INSERT权限控制。

由于Inceptor会和HDFS及Hyperbase交互，Hyperbase会和HDFS交互，这三个服务的授权机制也互相影响。

本手册中我们将介绍如何在不同的认证和授权场景下使用集群中的服务，以及如何利用我们提供的认证和授权机制更好地保护您集群中的资源。

2. Kerberos基本概念

本节我们介绍一些Kerberos最基础的概念。

2.1. Ticket

Ticket是在部署了Kerberos的集群中访问服务所需的凭证。Kerberos凭借Ticket中的信息来验证访问服务者的身份。用户通过Kerberos认证访问服务的过程中会需要两张ticket：

- Ticket-Granting-Ticket(TGT)：这张ticket需要用户手动获取，这张ticket是用户获取Service Ticket的凭证。
- Service Ticket：访问集群中各个服务的凭证。Kerberos会自动根据TGT来向用户发放Service Ticket，用户手动获取了TGT后 **无需** 自己获取Service Ticket。

我们可以做这样一个比喻：

集群中的每一个服务都是一个公园，要进入这些公园（访问服务）你需要各个公园的门票（Service Ticket）。而有了公园联票（Ticket-Granting-Ticket），你就可以不需要单独获取各个公园的门票（Service Ticket），只需要出示联票(Ticket-Granting-Ticket)，便可进入各个公园（访问服务）。

2.2. Principal

在Kerberos认证系统中，principal相当于用户名，是Kerberos给予Ticket-Granting-Ticket的对象。

Principal一般的形式是name/instance@realm或者name@realm。

- Name是用户名或者服务名。例如张三的用户名可以是zhangsan；Inceptor服务的服务名是hive。
- Instance用来对用户名或服务名进行限定。在集群中，用户和服务都会有各自对应的principal。有时，一个用户或服务会有多个principal：
  # 对于一个用户来说，拥有多个principal可以让她拥有对Kerberos的数据库（Kerberos存储principal和密码或者keytab的数据库）不同的管理权限。比如一个用户Alice可以有两个principal：alice@realm和alice/admin@realm。其中alice@realm可以让她如普通用户一般访问集群中的服务，而alice/admin@realm可以给她Kerberos数据库管理员的权限。
  # 对于一个服务来说，它可能分布在集群中的多个节点上，那么它在每个节点上都需要有一个自己对应的principal，这时可以用节点的hostname作为instance来区分。比如Inceptor在TDH集群中的principal的形式为：hive/hostname@TDH。

Name相同但是instance不同的principal在Kerberos中是完全不同的principal——它们一般会有不一样的密码和权限。

- Realm用于区分principal所在的Kerberos数据库。Realm相同的principal属于同一个Kerberos数据库，同一个realm的用户和服务可以互相访问。譬如一个TDH集群中的服务和用户的realm都是TDH。

2.3. 密码和Keytab文件

要通过Kerberos的认证需要提供principal及其对应的密码。密码可以手动输入，也可以存放在一个keytab文件中。“Keytab”是“key table”的简写，它用于存放一个或多个principal的密码。进行Kerberos认证时，一个用户可以提供principal和密码，或者principal和keytab文件。如果使用principal和keytab文件认证，那么Kerberos会去keytab文件中读取principal密码。通常，用户会使用principal加密码的形式通过认证，而服务会使用principal加keytab文件的形式通过认证。一般keytab文件的文件名为*.keytab。

管理员可以生成新的keytab文件，也可以向已有的keytab文件中添加某个principal的密码。

3. HDFS文件和目录权限简介

HDFS提供目录和文件级别的权限管理。每个文件和目录都有一个owner（所有者）和一个group（用户组）。一个文件或目录可以对owner、所属group中的用户和其他所有用户开放不同的权限。

- 对于一个文件，“r”代表读权限，“w”代表写权限。
- 对于一个目录，“r”代表查看目录下内容的权限，“w”代表在目录中创建或删除新的文件或目录的权限，“x”代表访问该目录的子目录的权限。

当您用hdfs dfs -ls /… 查看HDFS下的目录时，您通常会看到类似下面的输出：

```
drwxr-xr-x   - hdfs       hbase           0 2015-08-05 17:45 /test1
```

和权限有关的是“drwxr-xr-x”、“hdfs”和“hbase”这三个字段。其中：

- 第一个字段“drwxr-xr-x”包含了下面信息：
  # 第一个字符显示该行末尾的路径是文件还是目录：如果第一个字符是“d”代表该路径对应一个目录；如果第一个字符是“-”则代表该路径对应一个文件。
  # 后九个字符可分为三组三个的字符：第一组的三个字符代表该路径的owner的权限；第二组的三个字符代表该路径所属的group的权限；第三组的三个字符代表所有其他用户对该路径拥有的权限。
  # 每组三个字符中的第一个对应“r”,第二个对应“w”，第三个对应“x”，如果对应的位置显示是字母，则代表对应用户有字母所代表的权限，如果是“-”则代表没有权限。
- HDFS中“rwx”权限可以像操作系统下文件系统的“rwx”权限一样用数字表示——比如“rwxrwxrwx” = 777，“rwxr-xr-x” = 755等等。
- 第二个字段“hdfs”对应着该行末尾路径的owner。
- 第三个四段“hbase”对应该行末尾路径所属的group。

HDFS自己并不维护一套用户和用户组的信息，而是通过下面两种方式判断使用者的身份和用户组信息：

- 当HDFS没有启用Kerberos认证时，HDFS通过使用者在服务器操作系统上的用户名判断她的身份，她所在的组即她在操作系统上所属的组。
- 当HDFS启用了Kerberos认证，它通过使用者的principal判断她的身份。目前，当TDH集群上的HDFS服务需要获取用户所在的组信息时，HDFS会将用户的principal映射到NameNode所在服务器的操作系统上的用户并通过服务器操作系统上的用户组信息判断用户所在的组。映射方式如下：alice/instance@realm和alice@realm都会被映射到操作系统上的alice用户。此时，如果alice在NameNode所在服务器的操作系统中没有对应用户，那么alice对HDFS来说则没有组信息。

3.1. HDFS权限管理操作

HDFS的文件和木权限管理和Linux文件系统操作非常相似，只需在相同操作前加 `hdfs dfs` 即可。例如 `hdfs dfs -chmod`。下面举一些简单例子。

修改目录/文件owner

```
hdfs dfs -chown alice /test
```

修改目录/文件group

```
hdfs dfs -chgrp hadoop /test
```

修改目录/文件权限

```
hdfs dfs -chmod 777 /test
hdfs dfs -chmod g-r /test
hdfs dfs -chmod o=r /test
```

4. Inceptor权限管理简介

Inceptor使用基于SQL标准的用户权限管理——您可以使用熟悉的SQL语法来进行权限操作。Inceptor的管理员为hive。普通用户在经过hive授权之前在Inceptor中只能看到数据库下所有表的表名，而不能对表进行读写操作。

4.1. Inceptor中的权限

Inceptor中的权限如下：

- SELECT： 用于读表中的数据或用于查看数据库中的表；
- INSERT： 用于向表添加数据
- UPDATE： 用于更新表；
- DELETE： 用于删除表中数据；
- ALL：包括SELECT, INSERT, UPDATE和DELETE权限。
- CREATE： 用于建表或者建数据库。

4.2. 权限作用的对象

上面的权限可以作用于三个Inceptor中的对象：table, view和database。上面的权限全部都可以作用于table和view，但是只有SELECT和CREATE权限可以作用于database。也就是说，管理员可以向一个用户授予对table或view的SELECT, INSERT, UPDATE, DELETE,ALL和CREATE权限，但是仅可以向一个用户授予database的SELECT和CREATE权限。

当一个用户创建一个table, view或者database，该用户即成为这个table, view或者database的owner，拥有它所有的权限。

4.3. Inceptor中的用户(user)和角色(role)

如果Inceptor以Kerberos认证，Inceptor通过登陆者的TGT判断登录者的principal,进而得出登录者的身份。如果Inceptor以LDAP认证，Inceptor通过登陆者的LDAP用户名来判断登录者的身份。

Inceptor还管理角色（“role”）信息，通过向角色授权来批量管理用户权限。Table, view和database的权限既可以被授予给用户也可以被授予给角色。Inceptor中有两个特殊角色：public和admin。所有用户都有public角色；未经授权，只有hive用户有admin角色。一个用户可以同时拥有多个角色，除了admin之外的角色都在用户的当前角色（current roles）中。但是用户也可以使用set role选择某个特定角色作为当前角色。

创建新角色、给用户和角色授权等权限是admin角色独有的，拥有admin角色的用户必须先将她当前的角色设为admin才能行使admin独有的权利。

4.4. Inceptor行级权限

Inceptor支持行级别的权限管理。在数据敏感行业,用户希望有更细粒度的权限控制。如银行数据仓库,操作人员可能属于不同的支行,他们只应该看到其所属支行的数据。比较传统的做法是,将一张大表按支行切分成多个小表或者创建多个VIEW,通过对表或者VIEW的表级权限(PRIVILGE)管理来实现这个功能。但是这种做法在表或者VIEW的管理上相当繁琐。在TDH4.3及以后,通过一条简单的授权SQL,管理员或者表的Owner可以灵活地设定行级的访问权限规则。

Inceptor中的权限管理在《Inceptor使用手册》下的《Inceptor Guardian手册》中有详细介绍。

5. Hyperbase权限管理简介

安全模式下，Hyperbase通过Kerberos认真，所以Hyperbase通过用户的principal判断用户身份并使用Access Control Labels （ACLs）对客户端进行授权管理。Hyperbase的超级用户为hbase，可以向其他用户授权。经hbase授权之前，一个普通用户在Hyperbase没有任何权限，她只能list Hyperbase中的表，而不能进行诸如查看表内容、建表、修改表等操作，这些操作权限需要hbase授予。

在Hyperbase中一个用户可以拥有“RWC”权限，即：

- “R”：读权限，用来进行Get, Scan读操作；
- “W”：写权限，用来进行Put, Delete等写操作；
- “C”：建表权限，用来进行Create, Alter等操作；

默认情况下，一个用户对自己建的表有全部“RWC”权限。

Hyperbase中没有数据库中常见的“角色”概念，而是通过对用户组权限的管理来实现批量授权。Hyperbase没有自己管理的用户组信息，而是将用户的身份映射到服务器的操作系统上的用户并通过服务器操作系统上的用户组信息判断用户所在的组。映射方式如下：alice/instance@realm和alice@realm都会被映射到操作系统上的alice用户。此时，如果alice在操作系统中没有对应用户，那么alice对Hyperbase来说则没有组信息。

Hyperbase的权限管理操作在《Hyperbase使用手册》的“Hyperbase安全”章节有详细介绍。

6. 快速入门

我们在本章介绍如何进行一些最基本的操作来通过您集群上的安全认证开始使用服务。

6.1. 安全认证系统的判定

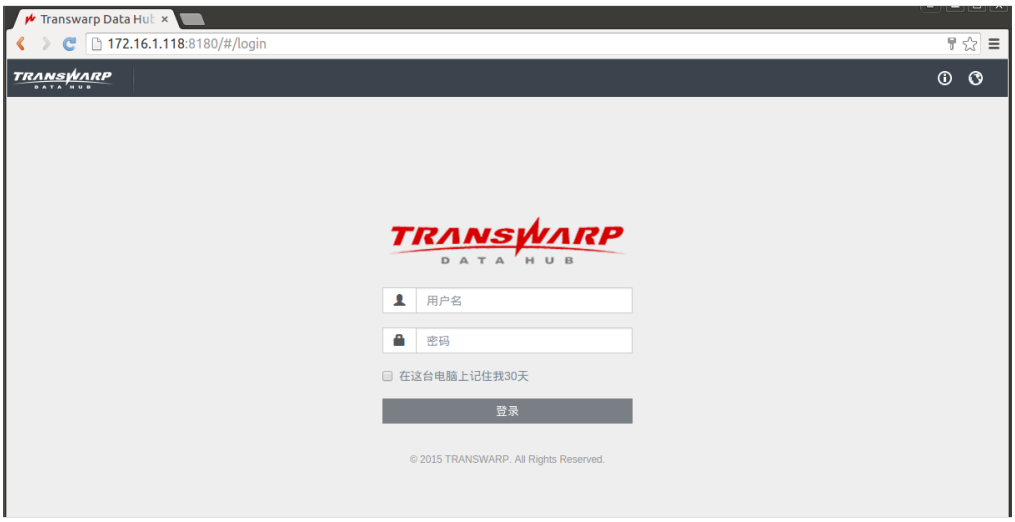
对集群的安全认证设置有下面几种选择：

1. 集群使用简单认证模式——不启用任何认证系统。
2. 集群使用Kerberos with LDAP认证系统，所有服务都启用Kerberos。
3. 集群使用Kerberos with LDAP认证系统，所有服务都启用Kerberos同时Inceptor启用LDAP。

下面我们介绍如何判定您集群的认证系统。

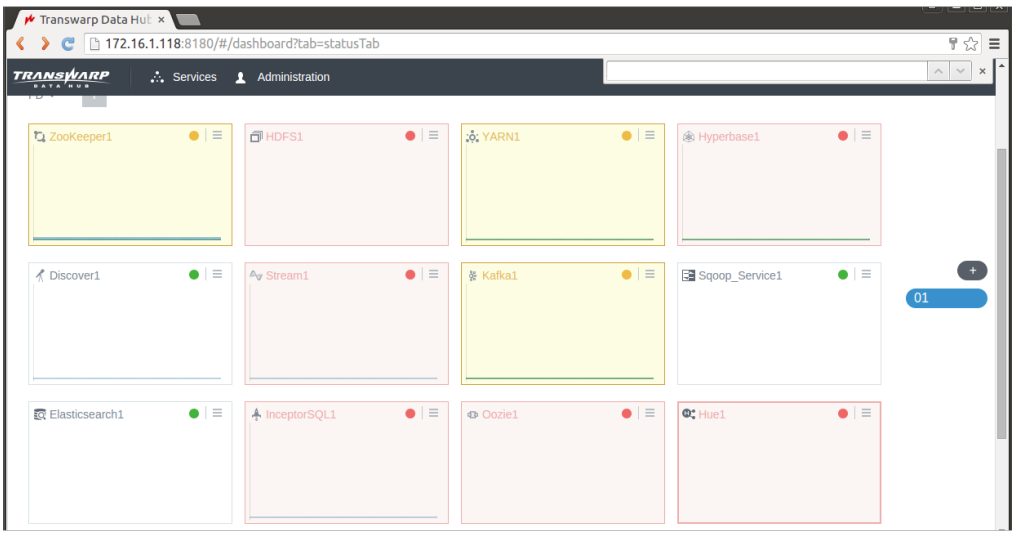
6.1.1. 通过管理界面

1. 用浏览器访问http://<manager_node_ip>:8180来登陆您集群的Transwarp Manager，这里，<manager_node_ip>是您管理节点的ip：

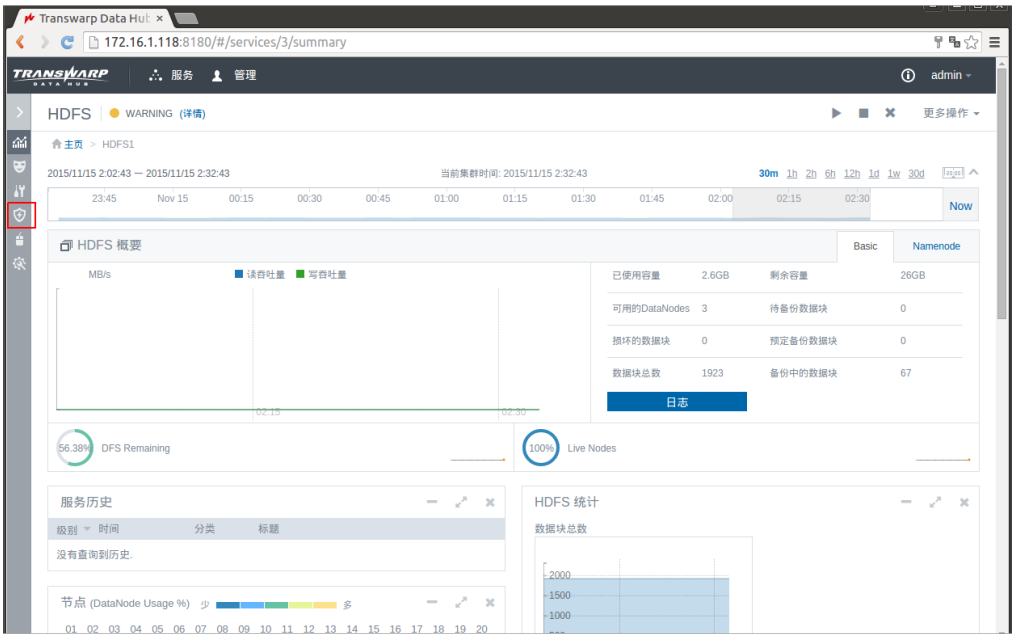


以您的用户名和密码登陆。如果您没有管理界面的用户名和密码，请您联系集群的管理员获取用户名和密码。

2. 在管理界面主页选择Zookeeper, HDFS, YARN, Hyperbase, Stream, Kafka, Sqoop, Oozie和Hue中的（这些服务的认证方式可以用管理界面控制）任意一个服务：



3. 我们以HDFS为例：



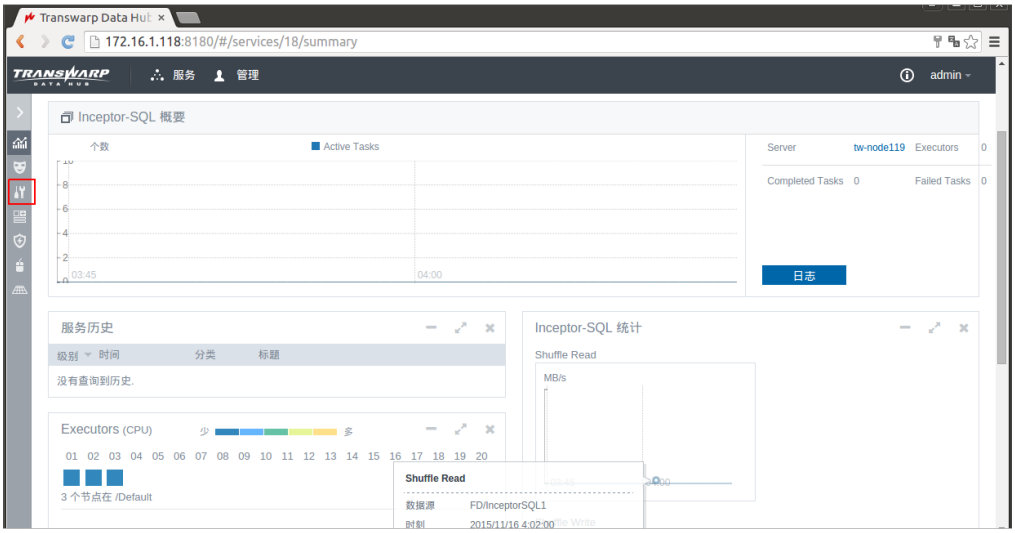
点击服务概况页面左边的盾牌图标，进入安全设置页面。

如果您看到页面显示“当前服务已经开启Kerberos”，则说明HDFS已经启用Kerberos，您在使用该服务前，必须通过Kerberos的认证——您需要提供Kerberos principal和密码（或keytab）。

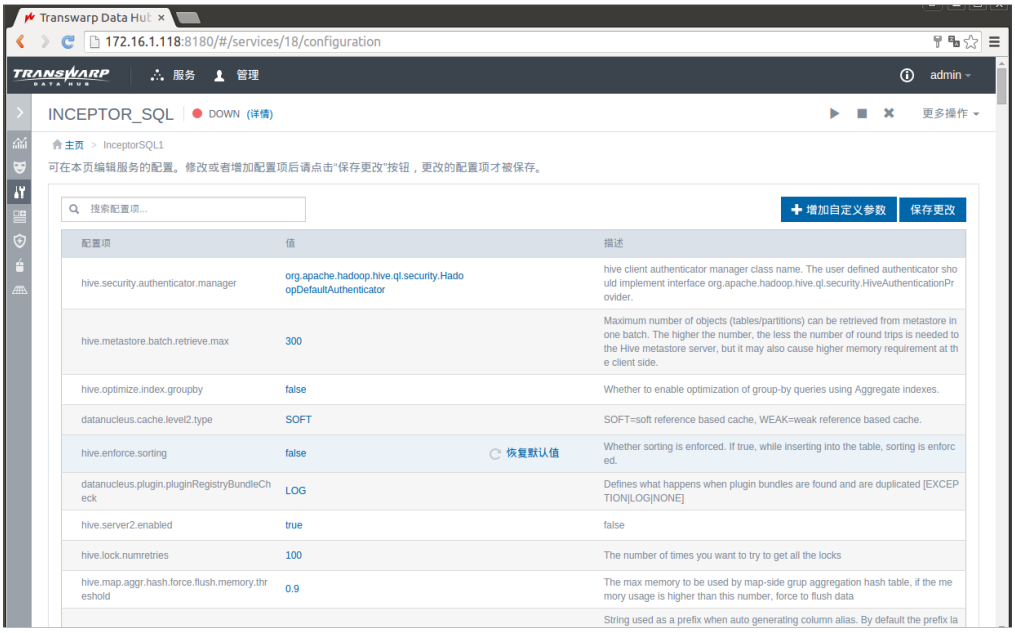
如果您看到页面显示“当前服务使用了基于用户的简单认证协议，可以开启Kerberos来达到更高的安全等级”，则说明HDFS没有开启Kerberos，您无需在使用服务前通过认证：

这里我们提供的是一个初步判定集群安全认证方式的方法。但是一个服务启用了Kerberos不代表整个集群中所有的服务都启用了Kerberos，您可以用同样的方式查看各服务是否都启用了Kerberos。因为服务之间互相依赖，我们推荐系统管理员为所有服务启用Kerberos或所有服务都不启用Kerberos。具体集群的配置请咨询您的系统管理员。

1. 要判定Inceptor是否启用了LDAP，您则需要进入Inceptor的服务页面：



点击服务页面左边的工具图标，进入Inceptor的配置页面：



在页面中寻找hive.server2.authentication参数。如果这个参数显示“LDAP”，则说明Inceptor使用LDAP认证方式。您在使用该服务前必须通过LDAP的认证——您需要提供LDAP的用户名和密码。

hive.server2.authentication	LDAP	Client authentication types. NONE: no authentication check LDAP: LDAP/AD based authentication KERBEROS: Kerberos/GSSAPI authentication CUSTOM: Custom authentication provider (Use with property hive.server2.custom.authentication.class)

如果这个参数显示“NONE”，则说明Inceptor没有使用LDAP认证方式：

hive.server2.authentication	NONE	Client authentication types. NONE: no authentication check LDAP: LDAP/AD based authentication KERBEROS: Kerberos/GSSAPI authentication CUSTOM: Custom authentication provider (Use with property hive.server2.custom.authentication.class)

这时，如果Inceptor启用了Kerberos（您可以参照上面查看HDFS是否启用Kerberos的方法查看），那么您在使用Inceptor之前需通过Kerberos认证——您需要提供Kerberos principal和密码（或keytab）。

如果Inceptor没有启用Kerberos也没有启用LDAP，您无需通过认证使用Inceptor。

6.1.2. 通过命令行

- 登陆您集群中的任意一个节点，登陆到集群中的服务器要求您有进入该服务器操作系统的用户名和密码，如果您没有，请联系您的集群管理员获取。
- 登陆集群后，执行klist命令。如果您看到下面输出：

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: usera@TDH

Valid starting     Expires            Service principal
11/16/15 04:52:04  11/16/15 14:52:04  krbtgt/TDH@TDH
 renew until 11/23/15 04:52:04
```

或者下面输出：

```
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

则说明您的集群安装了Kerberos with LDAP系统。但是暂时不能说明您系统中的服务是否使用了Kerberos认证。您需要进行下一步。

- 尝试使用一个服务，以HDFS为例，用下面指令尝试查看HDFS根目录下的内容：

```
hdfs dfs -ls /
```

如果您看到大量的错误输出，其中包含下面信息：

```
[Caused by GSSException: No valid credentials provided (Mechanism level: Failed to
 find any Kerberos tgt)];
```

则说明您的HDFS服务启用了Kerberos，您在使用该服务前，必须通过Kerberos的认证——您需要提供Kerberos principal和密码（或keytab）。

- 如果您需要在命令行使用Inceptor，您系统的管理员会给您连接到Inceptor的连接串。如果您的连接串类似下面，则说明Inceptor使用Kerberos认证：

```
"jdbc:hive2://<inceptor_server_ip/hostname>:10000/default;principal=hive/tw-
node119@TDH"
```

这里，<inceptor_server_ip/hostname>是您Inceptor服务所在节点的ip或者hostname；"hive/tw-node119@TDH"是您要访问的Inceptor服务的principal。

如果您从管理员处获取的Inceptor连接串类似如下，则说明Inceptor使用LDAP认证：

```
"jdbc:hive2://<inceptor_server_ip/hostname>:10000/default" -n <username> -p
 <password>
```

这里，<username>和<password>分别是您在LDAP中的用户名和密码，也应当从管理员处获得。

6.2. 通过认证使用服务

本节我们介绍如何在您集群的服务器上在命令行中通过认证使用服务。

6.2.1. 通过Kerberos认证

对于任意一个除了Inceptor外的服务，它只会有两种认证方式，启用了Kerberos或者没有启用Kerberos。如果它没有启用Kerberos，则您无需通过认证即可开始使用。如果它启用了Kerberos，那么您需要先获取一张有效的TGT，之后您便可以使用服务。您能获取有效TGT的前提是您在Kerberos数据库中有principal和密码（或者keytab）。如果您没有，请联系管理员获取。

- 如果您有Kerberos principal和密码，获取TGT的指令为：

```
kinit <principal>
```

这里<principal>处，您需要提供您的Kerberos principal。此时，系统会要求您输入您的密码，如果您的密码正确，您即获得一张有效的TGT。

- 如果您有Kerberos principal和keytab文件，获取TGT的指令为：

```
kinit -kt <keytab_path> <principal>
```

这里<keytab_path>出您需要提供您的keytab文件的路径，它可以是绝对路径或者相对路径。<principal>处则请提供您的Kerberos principal。如果您的keytab文件路径正确且keytab文件正确对应您的principal，您即获得一张有效的TGT。

- 您可以通过klist指令查看当前有效的TGT。如果执行klist得到类似下面的输出：

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: usera@TDH

Valid starting     Expires            Service principal
11/16/15 04:52:04  11/16/15 14:52:04  krbtgt/TDH@TDH
```

```
  renew until 11/23/15 04:52:04
```

且有效日期显示当前有效，则您已经有一张有效的TGT，您可以开始访问集群中的服务。如果执行klist得到下面输出：

```
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

或者类似下面的输出显示TGT已经过期

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: usera@TDH

Valid starting     Expires            Service principal
11/16/15 04:52:04  11/16/15 14:52:04  krbtgt/TDH@TDH
 renew until 11/23/15 04:52:04
```

则说明当前没有有效的TGT。您还不能访问集群中的服务。如果您的Kerberos principal和密码有误，请您联系您的管理员。

6.2.2. 命令行连接Inceptor

1. 如果您的Inceptor使用Kerberos认证，您需要先按照“通过Kerberos认证”中的步骤获取一张有效的TGT。然后执行下面的指令：

   ```
   [$host] beeline -u "jdbc:hive2:///<inceptor_server_ip/hostname>:10000/
   default;principal=<princpal_name>"
   ```

   这里<inceptor_server_ip/hostname>处提供您的Inceptor服务所在节点的ip或者hostname；<principal_name>处填写您想要连接的Inceptor server的principal，它一般的形式是hive/<inceptor_server_hostname>@TDH。

   - 如果您的Inceptor使用LDAP认证，您需要执行下面的命令通过LDAP的认证连接到Inceptor:

     ```
     beeline -u "jdbc:hive2:///<inceptor_server_ip/hostname>:10000/default" -n
      <username> -p <password>
     ```

     这里，<username>和<password>分别是您在LDAP中的用户名和密码。

部分 I. 管理员安全手册

7. 管理员安全手册简介

我们在管理员安全手册中将介绍如何使用脚本部署认证系统以及如何进行用户管理。我们定义管理员为拥有下面身份或角色的用户：

1. Transwarp Manager的admin用户
2. 集群服务器操作系统的root用户
3. Kerberos with LDAP认证系统的admin
4. Kerberos数据库管理员(用户的principal是admin@TDH或者*/admin@TDH)
5. 各项服务的超级用户,例如Inceptor的超级用户hive、Hyperbase的超级用户hbase、HDFS的超级用户hdfs等等。

从Transwarp Data Hub 4.3.2开始，Transwarp Manager支持通过图形化界面进行Guardian组件的安装和使用。从TDH4.3.2开始，Guardian组件是Transwarp Data Hub中认证和授权的中心，而通过Transwarp Manager图形界面进行的安全管理简单易用，本手册将要介绍的下面内容能够完全通过Guardian的安装和使用来实现：

- 第 8 章 Kerberos with LDAP部署对应“Guardian的安装和配置”，在《Transwarp Data Hub安装手册》中的“Guardian的安装和配置”部分有详细介绍。
- 第 9 章 Kerberos with LDAP用户管理对应“通过Guardian的用户管理”，在《Transwarp Data Hub运维手册》中的“用户管理”部分有详细介绍。
- 第 11 章 Kerberos管理系统常用指令仅介绍如何进行Kerberos数据库，而不影响LDAP中的信息，在多数情况下我们 **不推荐** 单独使用，因为会导致LDAP和Kerberos中信息不对称。

由于Guardian组件加强了统一管理，可以完全避免命令行误操作的信息不一致，我们推荐您通过Transwarp Manager使用Guardian组件进行用户管理。细节请参考《Transwarp Data Hub运维手册》中的“用户管理”章节。

8. Kerberos with LDAP部署

- 本章中，我们假设您的集群 **从未** 启用过Kerberos。如果您的集群之前启用过Kerberos，但是您希望转为使用新的Kerberos with LDAP认证系统，请先参考本章最后一节，“Kerberos数据的导入和导出”。
- 我们推荐的安全设置是整个集群部署Kerberos with LDAP并为每个服务启用Kerberos。在这个基础上，您可以选择是否为Inceptor启用LDAP。

下面，我们将以tw-node118, tw-node119和tw-node120三台机器组成的集群为例，演示如何安装Kerberos with LDAP认证系统。集群中，tw-node118是管理节点（Transwarp Manager安装的节点）。tw-node119和tw-node120将作为Kerberos with LDAP服务器，tw-node118将作为Kerberos with LDAP客户端。

8.1. 安装部署前的准备

1. 您需要确定您 **安装服务** 的时候为Inceptor服务选用了HiveServer2，并且将认证方式选为了LDAP。也就是说安装Inceptor时进行如下设置：
   - hive.server2.enabled 设置为 true
2. 如果您想要用LDAP做Inceptor的认证，您需要在安装服务的时候将Inceptor认证方式选为LDAP。也就是说安装Inceptor时进行如下设置：
   - hive.server2.authentication 设置为 LDAP

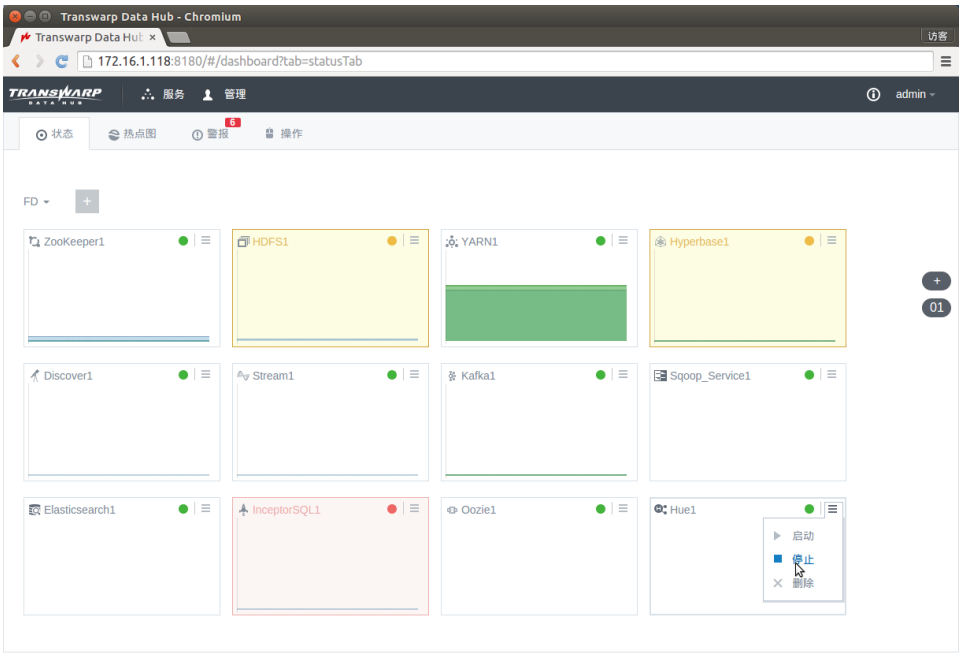
   如果您不想用LDAP做Inceptor的认证，您无需进行这步操作。更多关于Inceptor-SQL安装的细节和注意事项请参考《TDH安装手册》。
3. 管理节点（这里为tw-node118）的/root/transwarp/support/script/krb5-ldap-setup目录下将有所有您需要的脚本。该目录的结构如下：

```
|- client-install.sh （Kerberos客户端程序安装和配置脚本）
|- install.sh （Kerberos服务器部署脚本）
|- MigrateKerberos.sh （KDC数据库迁移脚本）
|- UserManager.sh （用户管理脚本）
|- krb5
    |- krb5-ldap-install.sh
|- logs
    |- client-install.log
    |- install.log
    |- UserManager.log
|- ldap
    |- ldap-ha-install.sh
    |- conf
        |- ldap.template.conf
        |- slapd.conf.obsolete
    |- ldif
        |- base.template.ldif
```

```
|- configSync.template.ldif
|- dbSync.template.ldif
|- enableTLS-SUSE.ldif
|- logLevel.ldif
|- syncProvider.ldif
|- configSyncProv.ldif
|- dbSyncProv.ldif
|- enableTLS.ldif
|- initDatabase.ldif
|- olcServerID.ldif
|- syncUser.ldif
```

如果您没有上述文件，请和我们联系。

4. 关闭iptables、SELinux、SuSEFirewall2，如果安装了TDH这些组件会被自动关闭。
5. 安装openssh-clients 用来scp传输文件
6. 修改/etc/hosts，直接使用IP地址安装会导致安装失败
7. 时间同步，如果安装了TDH会自动配置NTP时间同步
8. 需要配置运行脚本的机器与安装服务器或者客户端的机器ssh无密码登陆，安装完krb5-ldap服务和客户端之后，无密码登陆就可以删除了。
9. 通过管理界面停止集群中的所有服务。



8.2. 服务器ssh无密码登陆设置

我们将在tw-node118上运行install.sh脚本，将服务安装到tw-node119和tw-node120上，所以要设置tw-node118免密码登陆到tw-node119以及tw-118免密码登陆到tw-120。这里，我们演示设置tw-node118免密码登陆到tw-node119。

1. 在tw-node118下执行下面指令，生成公钥/私钥对：

```
ssh-keygen -t rsa -P ''
```

-P参数指定密码，''下为空表示空密码。执行该指令后在系统提示输入保存密钥的文件时直接回车。密钥文件会存在/root/.ssh/id_rsa.pub中。下面是操作记录：

```
[root@tw-node118 ~]# ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
59:c7:90:fd:b3:e0:52:0a:c1:af:dc:28:69:df:89:06 root@tw-node120
The key's randomart image is:
+--[ RSA 2048]----+
|        .  .o    |
|         o .o.   |
|          o. o.  |
|         .o..o o |
|        oS= + . o|
|       E + + . . |
|      . + o o    |
|         + o     |
|        .        |
+-----------------+
```

这个指令会在/root/.ssh下生成id_rsa和id_rsa.pub文件。看一下/root/.ssh下的文件：

```
[root@tw-node118 .ssh]# ls /root/.ssh
authorized_keys  config  id_rsa  id_rsa.pub  known_hosts  _PWD_TEMP_
```

2. 将tw-node118:/root/.ssh/id_rsa.pub复制到tw-node119:/root/.ssh目录下：

```
[root@tw-node118 ~]# scp /root/.ssh/id_rsa.pub root@tw-node119:/root/.ssh
root@tw-node119's password:
id_rsa.pub                                                    100%
 397     0.4KB/s   00:00
```

由于免密码登陆目前还没有设置好，您还需要输入tw-node119的密码才能scp。

3. 将id_rsa.pub中的密码写进tw-node119下的/root/.ssh/authorized_keys文件中：

```
[root@tw-node119 ~]# cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys
```

4. 将/root/.ssh/authorized_keys文件的权限改为600：

```
[root@tw-node119 ~]# chmod 600 /root/.ssh/authorized_keys
```

5. 现在从tw-node118免密码登陆到tw-node119已经设置完成。我们可以从tw-node118 ssh 到tw-node119：

```
[root@tw-node118 ~]# ssh tw-node119
```

8.3. 安装和配置

我们使用tw-node118:/root/transwarp/support/script/krb5-ldap-setup下的install.sh脚本安装Kerbero with LDAP服务，用同目录下的client-install.sh安装Kerberos with LDAP客户端。

8.3.1. install.sh脚本

使用方法: ./install.sh [options] <servers_hostname/ip>

<servers_hostname/ip> 为必要参数，指定安装Kerberos with LDAP服务的节点。例如下面指令将Kerberos with LDAP服务安装到tw-node119和tw-node120上。

```
./install tw-node119 tw-node120
```

Options:

- -s：如果使用-s参数，那么安装的OpenLDAP服务会启用TLS，那么再连接OpenLDAP时需要正确信任OpenLDAP服务的CA证书。
- -d：指定创建的ldap的域的后缀，如果不指定则使用dc=tdh作为后缀，例如后续添加的用户都会默认放在ou=people的子树下，如uid=testuser,ou=people,dc=tdh。
- -r：用来只能Kerberos的realm，默认情况为TDH。
- -l：用来设置OpenLDAP的log level，一般情况下不需要设置，如果发生故障需要知道Openldap的状态可以设置这个值，-1表示输出所有的日志，0表示没有任何输出。
- -a：使用这个参数指定 cn=Manager,dc=tdh 的密码，如果不指定会在脚本运行中提示。
- -m：使用这个参数设置Master Key的密码，如果不指定会在脚本运行中提示。
- -p：设置一个管理员用户用来管理其他用户，默认用户是admin，后面的-w参数指明这个管理员的密码。
- -w：设置管理员的密码，如果不指定会在脚本运行中提示。

8.3.2. client-install.sh脚本

使用方法： ./client-install.sh <servers_hostname/ip>

<servers_hostname/ip> 为必要参数，指定安装Kerberos with LDAP客户端的节点。例如下面指令将Kerberos with LDAP客户端安装到tw-node118上：

```
./client-install.sh tw-node118
```

安装过Kerberos with LDAP服务的节点无需再安装Kerberos with LDAP客户端，即使安装也会被跳过，因为安装服务的时候已经安装了客户端。

8.3.3. 操作记录

1. 部署Kerberos with LDAP服务器。这里，我们将Kerberos with LDAP服务器安装在tw-node119和tw-node120上。在tw-node118:/root/transwarp/support/script/krb5-ldap-setup下运行：

```
./install.sh tw-node119 tw-node120
```

这时，您需要设置三个密码：

- cn=Manager,dc=tdh的密码，这个用来管理LDAP自身
- Master Key，用来加密Kerberos数据库
- admin密码，Kerberos with LDAP的管理员密码

这个命令最终会完成Kerberos with LDAP服务的安装，并生成krb5-ldap.conf的配置文件。

2. 安装客户端。这里我们将客户端安装在tw-node118上。在tw-node118:/root/transwarp/support/script/krb5-ldap-setup下运行：

```
./client-instsall.sh localhost
```

这个脚本会自动安装Kerberos和LDAP的客户端软件，并配置好相关的选项。这里不需要将安装过Kerberos with LDAP服务器软件的机器加到参数中，因为这些机器已经配置好客户端，即使加入到参数中也会被跳过。

请注意，这个脚本应该在运行install.sh的机器上运行，如果你希望移动到其他地方运行，需要将krb5-ldap.conf配置文件一同拷贝过去。

8.3.4. 为Inceptor配置LDAP

如果您想要使用LDAP来对Inceptor进行认证，您需要进行下面的设置。

在启动服务之前，我们需要为Inceptor配置新装的LDAP服务。登陆管理界面， 在Inceptor的配置页面进行如下设置：

- hive.server2.authentication.ldap.baseDN 设置为 ou=people,dc=tdh
- hive.server2.authentication.ldap.url 设置为 ldap://tw-node119 ldap://tw-node120

在Inceptor配置页面进行操作的细节请参考《TDH运维手册》。

8.3.5. 为服务启用Kerberos

至此，Kerberos with LDAP认证系统安装结束。您可以去管理界面为各服务依次启用Kerberos。

注意，给服务启用Kerberos需要按照如下顺序，否则服务的启动将会遇到问题：

1. Zookeeper
2. HDFS
3. YARN
4. Hyperbase
5. Inceptor-SQL
6. Kafka
7. Stream
8. Sqoop
9. Oozie
10. HUE

8.4. Kerberos数据的导入和导出

如果您的TDH集群配置过Kerberos，但是您希望使用新的Kerberos with LDAP认证系统，您需要在安装和部署Kerberos with LDAP认证系统之前将原来Kerberos数据库中的数据转移到新的Kerberos with LDAP数据库中。您将需要用到MigrateKerberos.sh脚本。这个脚本在您集群的管理节点（Transwarp Manager安装的节点）的/root/transwarp/support/script/krb5-ldap-setup目录下。

我们以tw-node118, tw-node119和tw-node120三台机器组成的集群为例，介绍如何迁移Kerberos数据库中的数据。集群中，tw-node118是管理节点。假设该集群中原来的Kerberos服务器部署在tw-node118上，而新的Kerberos with LDAP认证系统计划在tw-node119和tw-node120上部署Kerberos with LDAP服务器，在tw-node118上部署Kerberos with LDAP客户端。

请注意，MigrateKerberos.sh脚本必须在 **本地** 运行，不能远程执行该脚本。

1. 进入/root/transwarp/support/script/krb5-ldap-setup目录，执行MigrateKerberos.sh脚本，导出原来Kerberos数据库中的数据到krb5.dump文件中：

```
[root@tw-node118] cd /root/transwarp/support/script/krb5-ldap-setup
[root@tw-node118] ./MigrateKerberos.sh export -m <MasterKey> -o krb5.dump    ❶
```

❶ 这里的MasterKey用于加密您 **将要** 部署的新Kerberos with LDAP数据库。

请确保这个MasterKey与您将要创建的Kerberos with LDAP数据库的Master Key一致，否则会导致迁移的信息无法使用。

2. 按照本章前文的“Kerberos with LDAP部署”在tw-node119和tw-node120上部署Kerberos with LDAP服务器， 在tw-node118上部署Kerberos with LDAP客户端。部署完之后先不要为集群启用Kerberos，而是直接进行下一步。

3. 现在我们需要将krb5.dump中的信息导入到tw-node119和tw-node120中任意一台上的Kerberos with LDAP数据库中。这里我们选择tw-node119。由于MigrateKerberos.sh脚本必须在本地执行，先要将tw-node118:/root/transwarp/support/script/krb5-ldap-setup下的MigrateKerberos.sh脚本和krb5.dump文件拷贝到tw-node119上。

```
[root@tw-node118] cd /root/transwarp/support/script/krb5-ldap-setup
[root@tw-node118] scp MigrateKerberos.sh krb5.dump root@tw-node119:~/kerberos-
setup
```

4. 登陆到tw-node119上，执行MigrateKerberos.sh脚本，导入数据到新的Kerberos with LDAP数据库中：

```
[root@tw-node118] ssh root@tw-node119
[root@tw-node119] cd kerberos-setup
[root@tw-node119] ./MigrateKerberos.sh import -h <krb5-ldap-server> -i
 krb5.dump ❶
```

❶ 这里的krb5-ldap-server处要填写一个Kerberos with LDAP所在的节点名。由于MigrateKerberos.sh必须在本机执行，所以这里要填本机节点名。

5. 卸载tw-node118上原有的Kerberos服务器krb5-server

- CentOS系统下：

```
[tw-node118] yum list installed | grep krb ❶
krb5-libs.x86_64                  1.10.3-10.el6_4.6
 @anaconda-CentOS-201311272149.x86_64/6.5
krb5-server.x86_64                1.10.3-10.el6_4.6                @os ❷
krb5-workstation.x86_64           1.10.3-10.el6_4.6                @os
```

❶ 查看当前安装的Kerberos相关软件包
❷ 这将是我们需要卸载的Kerberos服务器

执行下面指令，卸载原Kerberos服务器：

```
[tw-node118] yum remove krb5-server
```

- SUSE系统下：

```
[tw-node118] rpm -qa | grep krb ❶
krb5-client-1.6.3-133.49.54.1
krb5-1.6.3-133.49.54.1
krb5-server-1.6.3-133.49.54.1      ❷
krb5-plugin-kdb-ldap-1.6.3-133.49.54.1
krb5-32bit-1.6.3-133.49.54.1
```

❶ 查看当前安装的Kerberos相关软件包
❷ 这将是我们需要卸载的Kerberos服务器

执行下面指令，卸载原Kerberos服务器：

```
[tw-node118] zypper remove krb5-server
```

6. 删除tw-node118上原有的Kerberos数据库

- CentOS系统下：

```
[tw-node118] rm -rf /var/kerberos/krb5kdc
```

- SUSE系统下：

```
[tw-node118] rm -rf /var/lib/kerberos/krb5kdc
```

我们建议您卸载原有的Kerberos服务器和删除原有的Kerberos数据库。否则管理员可能会混淆新老Kerberos服务器和数据库的使用。

7. 到这里，从原Kerberos数据库迁移数据到新的Kerberos with LDAP数据库的工作就全部完成了。您可以为集群中的服务依次启用Kerberos，要注意为各服务启用Kerberos的顺序（见本章“安装过程”一节的末尾）。

9. Kerberos with LDAP用户管理

TDH中的一些服务（比如HDFS和Hyperbase）会将Kerberos principal映射到服务器操作系统中的用户（比如alice@TDH会被映射到系统用户alice），这个映射对这些服务的正常运行至关重要，所以我们建议管理员在进行用户管理时，确保每一个Kerberos with LDAP中的用户都能对应到服务器操作系统的用户，且操作系统中各服务器上用户所在的组信息保持一致。

我们为Kerberos with LDAP提供了用户管理工具：UserManager.sh。这个脚本在管理节点的root/transwarp/support/script/krb5-ldap-setup目录下。

使用UserManager.sh可以执行的操作如下：

1. useradd：添加用户
2. groupadd：添加用户组
3. usermod：修改用户
4. passwd：修改用户密码
5. userdel：删除用户
6. groupdel：删除组
7. ktadd：为用户生成keytab

- 除了passwd和ktadd以外，其他操作都需要管理员权限才能执行。普通用户可以使用passwd和ktadd来修改自己的密码和为自己生成keytab。
- 为了使普通用户能够使用UserManager.sh脚本执行passwd和ktadd，管理员需要设置好UserManager.sh脚本以及相关文件（例如krb5-ldap.conf配置文件和）和目录的权限，让普通用户可以正常使用。

您可以直接运行下面的指令查看关于一个command的详细用法。

```
./UserManager.sh command
```

下面，我们列举这些指令的详细用法。我们推荐在运行过Kerberos with LDAP的服务安装脚本install.sh的目录下运行UserManager.sh脚本，因为install.sh运行时生成的krb5-ldap.conf文件可以帮助减少参数的输入。您也可将krb5-ldap.conf拷贝到运行UserManager.sh脚本的目录下。

krb5-ldap.conf中的内容：

下面是tw-node118中krb5-ldap.conf中的内容：

```
[root@tw-node118 krb5-ldap-setup]# cat krb5-ldap.conf
DOMAIN=dc=tdh
```

```
SECURE=off
REALM=TDH
LOGLEVEL=0
ADMIN_PRINC=admin
SERVERS=(tw-node119 tw-node120)
```

所以在该文件目录下运行UserManager.sh就无需提供以上信息。上一章中，我们在tw-node118: /root/transwarp/support/script/krb5-ldap-setup/目录下执行了install.sh脚本，所以以下命令都在该目录下执行。

9.1. useradd （添加用户）

用法

```
./UserManager.sh useradd [options] -u <username> [<servers_hostname/ip>]
```

-u <username> 指定新建用户的用户名。

Options:

- -s：指定是否启用TLS，默认是不启用。
- -d：指定LDAP中的domain name，比如“dc=tdh”。默认是“dc=tdh”。
- -r：指定Kerberos中的realm。
- -a：指定了管理员用户的用户名，默认是admin。
- -w：指定了管理员用户的密码，如果不指定脚本运行时会提示输入。
- -p：指定新建用户的密码，如果不指定则在脚本运行中会提示输入。
- -g：指明用户所属的组，如果不指明，默认用户属于default组。
- -k：产生该用户的keytab，默认是不生成keytab。
- -f：指定生成的keytab的存放的位置，如果不指明，默认为$HOME/.keytab。
- -R：在本地操作系统上创建相同的用户（在操作系统上的密码和在Kerberos with LDAP系统中密码相同），并生成keytab存放在该用户的$HOME/.keytab中（这个选项是了兼容RStudio，因为RStudio需要在本地创建一个用户）。
- [<servers_hostname/ip>]：指定命令作用的服务器，默认为krb5-ldap.conf中SERVERS的值。这个参数必须放在命令的末尾。

实例操作：添加用户testuser

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh useradd -u testuser ❶
Using the servers: tw-node119 tw-node120
Input the password for the adminstrator user - "admin": ❷

Please set a password for testuser ❸
New password for testuser:
New password for testuser again:
Adding principals in KDC ❹
```

```
adding new entry "uid=testuser,ou=people,dc=tdh"    ❺

Add the testuser to group default    ❻
Adding user testuser successfully.
```

❶ 添加一个名为testuser的用户。
❷ 需要输入Kerberos with LDAP的管理员admin的密码。
❸ 设置添加的用户testuser的密码，该密码将是testuser在Kerberos和LDAP中公用的。
❹ UserManager.sh脚本会向Kerberos添加testuser的principal，该principal的名字将是testuser@TDH。
❺ UserManager.sh脚本同时会向LDAP添加testuser的信息。在默认参数下，testuser在LDAP中的DN (Distinguished Name)是：uid = testuser, ou = people, dc = tdh。
❻ 默认情况下testuser的用户组是default。

因为testuser同时被添加进了Kerberos数据库和LDAP数据库中，我们可以通过查看Kerberos数据库和LDAP数据库中的任意一个来验证testuser是否添加成功。

- 通过Kerberos

  执行kinit testuser@TDH来尝试为testuser@TDH获取一张ticket。如果获取成功，则证明testuser添加成功：

```
[root@tw-node118 krb5-ldap-setup]# kinit testuser@TDH    ❶
Password for testuser@TDH:                               ❷
[root@tw-node118 krb5-ldap-setup]# klist                 ❸
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: testuser@TDH

Valid starting     Expires            Service principal
10/21/15 23:28:19  10/22/15 09:28:19  krbtgt/TDH@TDH
 renew until 10/28/15 23:28:19
```

  ❶ 为testuser@TDH获取ticket。
  ❷ 输入testuser@TDH这个principal在Kerberos中的密码 （即UserManager.sh运行时设置的密码）。
  ❸ 查看当前的有效ticket。下面的输出证明testuser添加成功。

- 通过LDAP

  执行ldapsearch来在LDAP数据库中搜索testuser：

```
[root@tw-node118 krb5-ldap-setup]# ldapsearch -x -D uid=testuser,ou=people,dc=tdh
 -W -b dc=tdh
Enter LDAP Password:    ❶
```

  ❶ 这里填写testuser在LDAP中的密码 （即UserManager.sh运行时设置的密码）。

  如果您看到下面输出，则说明添加成功：

```
...
```

```
...
# testuser, people, tdh
dn: uid=testuser,ou=people,dc=tdh
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
cn: testuser@TDH
sn: testuser@TDH
uid: testuser
userPassword:: e1NBU0x9dGVzdHVzZXJAVERI

# search result
search: 2
result: 0 Success

# numResponses: 55
# numEntries: 54
```

9.2. userdel （删除用户）

用法

```
./UserManager.sh userdel [options] -u <username> [<servers_hostname/ip>]
```

-u <username> 指定被删除用户的用户名。

Options:

- -s：指定是否启用TLS，默认是不启用。
- -d：指定LDAP中的domain name，比如“dc=tdh”。默认是“dc=tdh”。
- -r：指定Kerberos中的realm。
- -a：指定了管理员用户的用户名，默认是admin。
- -w：指定了管理员用户的密码，如果不指定脚本运行时会提示输入。
- [<servers_hostname/ip>]：指定命令作用的服务器，默认为krb5-ldap.conf中SERVERS的值。这个参数必须放在命令的末尾。

实例操作

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh userdel -u testuser
Using the servers: tw-node119 tw-node120
Input the password for the adminstrator user - "admin":
Deleting principals in KDC
Delete user testuser successfully.
```

9.3. groupadd （添加用户组）

用法

```
./UserManager.sh groupadd [options] -g <groupname> [<servers_hostname/ip>]
```

-g <groupname> 指定添加的用户组的组名。

Options:

- -s: 指定是否启用TLS，默认是不启用。
- -d: 指定LDAP中的domain name，比如“dc=tdh”。默认是“dc=tdh”。
- -r: 指定Kerberos中的realm。
- -a: 指定了管理员用户的用户名，默认是admin。
- -w: 指定了管理员用户的密码，如果不指定脚本运行时会提示输入。
- [<servers_hostname/ip>]: 指定命令作用的服务器，默认为krb5-ldap.conf中SERVERS的值。这个参数必须放在命令的末尾。

实例操作

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh groupadd -g testgroup
Using the servers: tw-node119 tw-node120
Input the password for the adminstrator user - "admin":
Adding group testgroup successfully.
```

9.4. groupdel （删除用户组）

用法

```
./UserManager.sh groupdel [options] -g <groupname> [<servers_hostname/ip>]
```

-g <groupname> 指定删除的用户组的组名。

Options

- -s: 指定是否启用TLS，默认是不启用。
- -d: 指定LDAP中的domain name，比如“dc=tdh”。默认是“dc=tdh”。
- -r: 指定Kerberos中的realm。
- -a: 指定了管理员用户的用户名，默认是admin。
- -w: 指定了管理员用户的密码，如果不指定脚本运行时会提示输入。
- [<servers_hostname/ip>]: 指定命令作用的服务器，默认为krb5-ldap.conf中SERVERS的值。这个参数必须放在命令的末尾。

实例操作

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh groupdel -g testgroup
Using the servers: tw-node119 tw-node120
```

```
Input the password for the adminstrator user - "admin":
Delete group testgroup successfully.
```

9.5. usermod （修改用户的用户组）

用法

```
./UserManager.sh usermod [options] -u <username> (-g <groupname>|-x <groupname>)
 [<servers_hostname/ip>]
```

使用-g参数将指定用户添加进<groupname>指定的用户组。使用-x参数将指定用户从<groupname>指定的用户组中删除。

Options

- -s：指定是否启用TLS，默认是不启用。
- -d：指定LDAP中的domain name，比如“dc=tdh”。默认是“dc=tdh”。
- -r：指定Kerberos中的realm。
- -a：指定了管理员用户的用户名，默认是admin。
- -w：指定了管理员用户的密码，如果不指定脚本运行时会提示输入。
- [<servers_hostname/ip>]：指定命令作用的服务器，默认为krb5-ldap.conf中SERVERS的值。这个参数必须放在命令的末尾。

实例操作

将用户添加进一个用户组：

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh usermod -u testuser -g testgroup
Using the servers: tw-node119 tw-node120
Input the password for the adminstrator user - "admin":
Add the testuser to group testgroup
Change user testuser successfully.
```

将用户从一个用户组中删除：

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh usermod -u testuser -x testgroup
Using the servers: tw-node119 tw-node120
Input the password for the adminstrator user - "admin":
Delete the testuser from group testgroup
Change user testuser successfully.
```

9.6. passwd （修改用户的密码）

用法1：管理员修改用户的密码

```
/UserManager.sh passwd [options] -a admin -u <username> [<servers_hostname/ip>]
```

-a 指定管理员用户的用户名；-u 指定修改密码的用户的用户名。

用法2：用户修改自己的密码

```
/UserManager.sh passwd [options] -u <username> [<servers_hostname/ip>]
```

-u 指定修改密码的用户的用户名。

Options

- -s：指定是否启用TLS，默认是不启用。
- -d：指定LDAP中的domain name，比如“dc=tdh”。默认是“dc=tdh”。
- -r：指定Kerberos中的realm。
- -w：指定了管理员用户的密码，如果不指定脚本运行时会提示输入。
- -o：没有管理员密码的情况下，也可以修改密码，只需要输入原来的密码和新的密码，这个选项可以不指定会在脚本中提示。
- -n：指定新的密码。
- -k：指定需要产生新的keytab信息。
- -f：在使用-k参数的情况下，keytab信息存放的路径，默认存放路径是$HOME/.keytab。
- -R：更改用户在Kerberos with LDAP中的密码，同时更新用户对应的 **操作系统用户** 的密码以及更新$HOME/.keytab下的信息（这个选项是为了兼容RStudio，RStudio需要用到$HOME/.keytab中的信息）。
- [<servers_hostname/ip>]：指定命令作用的服务器，默认为krb5-ldap.conf中SERVERS的值。这个参数必须放在命令的末尾。

实例操作

admin修改testuser的密码：

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh passwd -a admin -u testuser
Using the servers: tw-node119 tw-node120
Input the password for the adminstrator user admin:

Please set a password for testuser
New password for testuser:
New password for testuser again:
Change password for testuser successfully.
```

testuser修改自己的密码：

```
[testuser@tw-node118 krb5-ldap-setup]# ./UserManager.sh passwd -u testuser
Using the servers: tw-node119 tw-node120
Input the current password for the user testuser:

Please set a password for testuser
New password for testuser:
```

```
New password for testuser again:
Change password for testuser successfully.
```

- 用passwd指令修改密码并不会修改用户对应的keytab文件中的密码，所以用passwd修改密码后原来的keytab文件就不能使用了。所以在修改用户密码的时候需要注意这一点，保证更新keytab。要保证keytab信息的同步您需要使用-k参数指定生成对应新密码的keytab信息。使用-k参数的同时使用-f可以指定新生成的keytab信息存放的路径，如果不用-f则默认将生成的keytab信息放在$HOME/.keytab下。此时，如果新生成的keytab信息和原keytab信息存放路径不同，请确保用户使用新的keytab路径。
- 如果被修改密码的用户创建时用了-R参数（./UserManager.sh useradd -u -R <username>），修改密码时也要加上-R参数，保证用户对应的 **操作系统用户** 家目录下的keytab文件和密码同步。

9.7. ktadd （为指定用户生成keytab文件）

用法

```
./UserManager.sh ktadd [options] -u <username>
```

-u 指定生成keytab的用户。

Options

- -k 指明了keytab文件的路径，如果不指定则$HOME/.keytab被默认使用。
- -r 指定了Kerberos的realm，如果不指定，则使用TDH或者是krb5-ldap.conf中的REALM的值。
- -w 指定了要生成keytab的用户的密码，如果不指定，脚本会在运行时提示输入。

运行ktadd时，系统会将用户的密码（通过-w指定或者在运行时输入）添加进keytab文件中。运行ktadd时提供的密码必须和用户密码相同，否则生成的keytab将无法使用。如果为一个不存在的用户执行ktadd，系统也会生成keytab，但是不能使用。

管理员可以为用户生成keytab文件，用户也可以为自己生成keytab文件。如果管理员为用户生成keytab文件，请管理员设置好这个keytab文件的权限保证用户可以使用。

实例操作

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh ktadd -u testuser
Using the servers: tw-node119 tw-node120
Generate keytab for testuser

Please set a password for testuser
Enter password for testuser:
Enter password for testuser again:
```

```
Keytab for testuser@TDH generated in /root/.keytab successfully ❶
```

❶ 默认设置下，生成的keytab信息会存在/root/.keytab文件下。如果您想要自己设置keytab的路径和文件名，使用-k参数：

```
[root@tw-node118 krb5-ldap-setup]# ./UserManager.sh ktadd -k /root/testuserkt -u
 testuser
Using the servers: tw-node119 tw-node120
Generate keytab for testuser

Please set a password for testuser
Enter password for testuser:
Enter password for testuser again:
Keytab for testuser@TDH generated in /root/testuserkt successfully
```

10. Kerberos管理系统简介

Kerberos数据库中存放了所有principal和密码/keytab信息，您可以通过kadmin或者kadmin.local命令行来进入Kerberos管理系统来管理其中的信息，比如添加和删除用户、修改用户密码，生成keytab文件等等。

10.1. kadmin和kadmin.local

kadmin和kadmin.local都是进入Kerberos管理系统的命令行工具，支持同样的管理指令。它们的区别在于，kadmin.local只能在Kerberos server所在节点使用，而且使用者必须以root用户身份登陆节点。

10.1.1. kadmin.local

如果您以root身份登陆，您无需在服务器上有有效的TGT便可以进入kadmin.local命令行：

```
[root@tw-node119 ~]# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
[root@tw-node119 ~]# kadmin.local
Authenticating as principal root/admin@TDH with password.
kadmin.local:
```

“Authenticating as principal root/admin@TDH with password.”显示kadmin.local会将您当做root/admin@TDH来认证，但是值得注意的是root/admin@TDH这个principal并不在Kerberos数据库中。

假设您以root身份登陆Kerberos server所在节点，但是以userc@TDH身份获取了一张TGT，那么kadmin.local会将您当做userc/admin@TDH来认证：

```
[root@tw-node119 ~]# kinit userc@TDH
Password for userc@TDH:
[root@tw-node119 ~]# kadmin.local
Authenticating as principal userc/admin@TDH with password.
kadmin.local:
```

在Kerberos管理系统中，所有principal形式为*/admin@TDH的用户都有Kerberos管理系统的所有权限。也就是说，Kerberos server所在节点的root用户拥有Kerberos管理系统的所有管理权限。

10.1.2. kadmin

kadmin是Kerberos管理 系统的远程客户端，使用权限受Kerberos管理系统权限ACL控制。一个用户可以在集群的任何一个节点上使用kadmin，而且无需是操作系统的root用户。kadmin需要用户必须以一个principal的身份登陆（登陆指令为 kadmin -p <principal>）：

```
[userc@tw-node118 ~]$ kadmin -p userc@TDH
```

```
Couldn't open log file /var/log/kadmind.log: Permission denied ❶
Authenticating as principal userc@TDH with password.
Password for userc@TDH:
kadmin:
```

<1>您无需担心这条信息——kadmin.log是服务端的log，客户端不需要去编辑此文件。

10.2. Kerberos管理系统权限

用户在Kerberos管理系统中有不同权限。Kerberos管理系统中用户所拥有的权限用一个ACL管理，信息保存在Kerberos server所在节点的kadm5.acl文件中（文件路径：kerbers_server_node:/var/kerberos/krb5kdc/kadm5.acl）。我们可以ll一下这个文件：

```
[root@tw-node119 ~]# ll /var/kerberos/krb5kdc/kadm5.acl
-rw------- 1 root root 28 Oct 18 01:29 /var/kerberos/krb5kdc/kadm5.acl
```

可以看到文件的owner和group都是root，权限是600。只有操作系统的root用户才可以改这个文件。这个文件的格式是：

```
Kerberos_principal      permissions     [target_principal] [restrictions]
```

- Kerberos_principal：授予permissions中权限的对象。
- permissions： Kerberos_principal的权限，由一串对应权限的字母表示。下面是Kerberos管理系统中的所有权限。将字母大写代表 **没有** 对应的权限。

a (add)	添加principal或policy的权限
d (delete)	删除principal或policy的权限
m (modify)	修改principal或policy的权限
c (changepw)	修改所有principal密码的权限
i (inquire)	inquire数据库的权限
l (list)	列出principal或policy的权限
s (set)	设置principal key的权限
*	所有权限
x	所有权限（和*相同）

- target_principal（可选）：设置permissions中权限可以作用的principal。
- restrictions（可选）：一个

Kerberos_principal和可选的target_principal中的任意一部分：name, instance和realm都可以包含通配符：比如*/admin@TDH表示所有instance是admin,realm为TDH的principal。默认设置下，您的kadm5.acl文件的内容如下：

```
*/admin@TDH  *
admin@TDH  *
```

即在默认设置下，principal形式为admin@TDH和*/admin@TDH的用户在Kerberos管理系统中有所有权限，其他principal只有修改自己密码的权限。从kadmin.local进入管理系统的用户会被Kerberos管理系统自动认为是管理员，因而有所有管理权限。

> 如果kadm5.acl有改动，kadmind必须要重启才能让改动生效。

11. Kerberos管理系统常用指令

如果您的集群部署了Kerberos with LDAP安全认证系统，建议您使用UserManager.sh脚本来进行用户管理，以保证Kerberos和LDAP中用户信息同步。使用kadmin或者kadmin.local的修改只对Kerberos数据库生效。

本章中介绍的指令是用于管理Kerberos数据库中信息的指令，只能在kadmin或kadmin.local中运行。和Kerberos指令（例如klist, kinit等）有别。

首先，进入Kerberos管理系统：您可以以操作系统用户root身份在Kerberos server所在节点使用kadmin.local：

```
[root@tw-node119 ~]# kadmin.local
Authenticating as principal userc/admin@TDH with password.
kadmin.local:
```

或者在集群中任意一个节点以一个有Kerberos管理员权限的principal使用kadmin：

```
[root@tw-node118 ~]# kadmin -p alice/admin@TDH
Authenticating as principal alice/admin@TDH with password.
Password for alice/admin@TDH:
kadmin:
```

通过这两个命令行工具能够执行的指令相同。只是通过kadmin执行的指令的权限会受kadm5.acl中的规定限制。

下面是所有可以在kadmin或kadmin.local中执行的指令，除非特殊情况，我们将不做kadmin和kadmin.local的区分。

11.1. add_principal

用法：add_principal [options] <new_principal>

执行权限：add

化名：addprinc, ank

添加名为new_principal的新principal，系统会提示您输入两次设置密码。

Options:

- -randkey

  为新添加的principal生成一个随机密码。注意如果为principal生成一个随机密码，那么必须要将生成的随机密码放在keytab文件中才能使用（参见ktadd指令）。

- -pw <password>

将新添加的principal的密码设置为指定的字符串。使用这个选项系统就不会提示您输入两次来设置密码，但是容易将密码暴露。

实例操作

- 添加bob@TDH这个principal：

```
kadmin:  add_principal bob@TDH
WARNING: no policy specified for bob@TDH; defaulting to no policy
Enter password for principal "bob@TDH":
Re-enter password for principal "bob@TDH":
Principal "bob@TDH" created.
```

- 添加carol@TDH这个principal同时指定密码为123456：

```
kadmin:  add_principal -pw 123456 carol@TDH
WARNING: no policy specified for carol@TDH; defaulting to no policy
Principal "carol@TDH" created.
```

11.2. rename_principal

用法： rename_principal [-force] <old_principal> <new_principal>

执行权限： add和delete

化名： renprinc

将old_principal重命名为new_principal。如果不加“-force”选项，系统会提示您对重命名操作进行确认。

11.3. delete_principal

用法： delete_principal [-force] <principal>

执行权限： delete

化名： delprinc

将指定的principal删除。如果不加“-force”选项，系统会提示您对删除操作进行确认。

11.4. change_password

用法： change_password [options] <principal>

执行权限： changepw或者执行者是principal本人

化名： cpw

修改指定principal的密码。如果不加“-randkey”或者“-pw”选项，系统会提示您输入两次设置新的密码。

Options:

- -randkey

  将principal的密码设置为一个随机字符串。注意，如果您这么做，那么必须要将随机密码放在keytab文件中才能使用（参见ktadd指令）。

- -pw <password>

  将principal的密码设置为指定的字符串。使用这个选项系统就不会提示您输入两次来设置密码，但是容易将密码暴露。

实例操作

- 修改bob@TDH的密码：

```
kadmin:  cpw bob@TDH
Enter password for principal "bob@TDH":
Re-enter password for principal "bob@TDH":
Password for "bob@TDH" changed.
```

- 修改carol@TDH的密码：

```
kadmin:  cpw -pw 123 carol@TDH
Password for "carol@TDH" changed.
```

11.5. list_principals

用法：list_principals [expression]

执行权限：list

化名：listprincs, get_principals, get_princs

列出所有和名字和expression匹配的principal。如果不指定[expression]则列出Kerberos数据库中的所有principal。

实例操作

```
kadmin:  listprincs */admin@TDH
kadmin/admin@TDH
alice/admin@TDH
```

11.6. ktadd

用法1：ktadd -k [-norandkey] <keytab_path> <principal>

执行权限：inquire和changepw，或者执行者是principal本人。

将指定的principal的密码加进一个keytab_path指定的keytab文件中。

用法2： ktadd -k [-norandkey] <keytab_path> -glob <principal_expression>

执行权限： inquire, changepw和list。

将和<principal_expression>匹配的全部principal的密码加入keytab_path指定的keytab文件中。

- -norandkey选项只有在kadmin.local命令行中可以使用
- 如果不加-norandkey选项，Kerberos将改掉principal的原密码并用一个新生成的随机密码代替，再将新生成的随机密码放入指定的keytab中，这样一来principal就不能使用原有的密码认证。所以我们建议您为用户principal生成keytab时，总是使用-norandkey选项。

实例操作

用法1

```
kadmin.local:  ktadd -k /tmp/bob.keytab -norandkey bob@TDH
Entry for principal bob@TDH with kvno 7, encryption type aes256-cts-hmac-sha1-96
 added to keytab WRFILE:/tmp/bob.keytab.
Entry for principal bob@TDH with kvno 7, encryption type aes128-cts-hmac-sha1-96
 added to keytab WRFILE:/tmp/bob.keytab.
Entry for principal bob@TDH with kvno 7, encryption type des3-cbc-sha1 added to
 keytab WRFILE:/tmp/bob.keytab.
Entry for principal bob@TDH with kvno 7, encryption type arcfour-hmac added to keytab
 WRFILE:/tmp/bob.keytab.
```

用法2

```
kadmin.local:  ktadd -k /tmp/user.keytab -norandkey -glob user*@TDH
Entry for principal user1@TDH with kvno 1, encryption type aes256-cts-hmac-sha1-96
 added to keytab WRFILE:/tmp/user.keytab.
Entry for principal user1@TDH with kvno 1, encryption type aes128-cts-hmac-sha1-96
 added to keytab WRFILE:/tmp/user.keytab.
Entry for principal user1@TDH with kvno 1, encryption type des3-cbc-sha1 added to
 keytab WRFILE:/tmp/user.keytab.
Entry for principal user1@TDH with kvno 1, encryption type arcfour-hmac added to
 keytab WRFILE:/tmp/user.keytab.
Entry for principal user2@TDH with kvno 1, encryption type aes256-cts-hmac-sha1-96
 added to keytab WRFILE:/tmp/user.keytab.
Entry for principal user2@TDH with kvno 1, encryption type aes128-cts-hmac-sha1-96
 added to keytab WRFILE:/tmp/user.keytab.
Entry for principal user2@TDH with kvno 1, encryption type des3-cbc-sha1 added to
 keytab WRFILE:/tmp/user.keytab.
Entry for principal user2@TDH with kvno 1, encryption type arcfour-hmac added to
 keytab WRFILE:/tmp/user.keytab.
kadmin.local: Principal user*@TDH does not exist.
```

部分 II. 用户安全手册

12. 用户安全手册简介

在本手册中,我们将称部署并启用了认证系统(包括Kerberos,LDAP或者Kerberos with LDAP)的集群为“ **安全模式** ”下的集群。在安全模式下的集群,用户需要通过认证才能够使用服务。如果集群没有部署或者部署了但没有启用任何认证系统,那么我们称其为“非安全模式”下的集群。

我们在用户安全手册中将介绍普通用户在 **安全模式** 下的集群中的用户信息管理和各项服务的使用。如果您想要迅速通过认证开始使用集群,请阅读 第 6 章 快速入门。

我们定义 **普通用户** 为没有管理员权限的用户,除非特别设置,普通用户没有下面用户或角色的权限:

- 集群服务器操作系统的root用户
- Kerberos with LDAP认证系统的admin
- Kerberos数据库管理员(用户的principal不是admin@TDH或者*/admin@TDH)
- 各项服务的超级用户,例如Inceptor的超级用户hive、Hyperbase的超级用户hbase、HDFS的超级用户hdfs等等。作为普通用户,在开始使用集群服务前,您应该从管理员处获取以下信息:
- 您在集群的操作系统中的用户名和密码
- 如果您的集群部署了Kerberos,您应该有您在Kerberos中的principal和密码/keytab。
- 如果您的Inceptor启用了LDAP,您需要LDAP的用户名和密码来访问Inceptor服务。
- 如果您的集群部署了Kerberos with LDAP系统,您需要有这个系统的用户名和密码。假设您的用户名是user_name,密码是password。那么:

  # 您的Kerberos principal将是user_name@TDH,Kerberos密码是password;

  # 您的LDAP用户名是user_name,密码是password。

我们将以一位名为Alice的用户为例进行介绍和演示,她的用户信息为:

- 集群操作系统用户名:alice
- Kerberos with LDAP用户名:alice,所以Kerberos principal为alice@TDH。Alice将以alice身份登陆集群进行操作。

从Transwarp Data Hub 4.3.2开始，Transwarp Manager支持通过图形化界面进行Guardian组件的安装和使用。从TDH4.3.2开始，Guardian组件是Transwarp Data Hub中认证和授权的中心，而通过Transwarp Manager图形界面进行的安全管理简单易用，本手册将要介绍的下面内容能够完全通过Guardian的安装和使用来实现。并且，由于Guardian组件加强了统一管理，可以完全避免命令行误操作的信息不一致，我们推荐您通过Transwarp Manager使用Guardian组件进行用户管理。细节请参考《Transwarp Data Hub运维手册》中的“用户管理”章节。

13. Kerberos with LDAP用户密码管理

在Kerberos with LDAP系统中进行密码管理需要使用Kerberos with LDAP的UserManager.sh脚本。您需要向管理员咨询该脚本所在的路径。 您作为普通用户使用UserManager.sh能够进行的操作有

- 使用 `ktadd` 指令生成keytab信息。
- 使用 `passwd` 指令修改密码。

13.1. ktadd

用法

```
./UserManager.sh ktadd [options] -u <username>
```

`-u` 指定生成keytab的用户。

Options

- `-k` 指明了keytab文件的路径,如果不指定则$HOME/.keytab被默认使用。
- `-r` 指定了Kerberos的realm,如果不指定,则使用TDH或者是krb5-ldap.conf中的REALM的值。
- `-w` 指定了要生成keytab的用户的密码,如果不指定,脚本会在运行时提示输入。

执行ktadd时需要提供生成keytab的用户的密码——提供的密码必须和用户在Kerberos with LDAP系统中的密码一致,否则生成的keytab信息将无法使用。

实例操作:Alice为自己生成keytab信息

```
[alice@tw-node118 krb5-ldap-setup]$ ./UserManager.sh ktadd -u alice
Using the servers: tw-node119 tw-node120
Generate keytab for alice
Please set a password for alice
Enter password for alice:
Enter password for alice again:
Keytab for alice@TDH generated in /home/alice/.keytab successfully
```

实例操作:Alice为自己生成keytab信息,并将其放在/home/alice/credentials/keytabs中

```
[alice@tw-node118 krb5-ldap-setup]$ ./UserManager.sh ktadd -k ~/credentials/keytabs -
u alice
Using the servers: tw-node119 tw-node120
Generate keytab for alice

Please set a password for alice
Enter password for alice:
Enter password for alice again:
```

```
Keytab for alice@TDH generated in /home/alice/credentials/keytabs successfully
```

13.2. passwd

用法:

```
/UserManager.sh passwd [options] -u <username> [<servers_hostname/ip>]
```

`-u` 指定修改密码的用户的用户名。

Options

- `-s` 指定是否启用TLS,默认是不启用。
- `-d` 指定LDAP中的domain name,比如“dc=tdh”。默认是“dc=tdh”。
- `-r` 指定Kerberos中的realm。
- `-n` 指定新的密码。
- `-k` 指定需要产生新的keytab信息。
- `-f` 在使用-k参数的情况下,keytab信息存放的路径,默认存放路径是$HOME/.keytab。
- `[<servers_hostname/ip>]` 指定命令作用的服务器,默认为krb5-ldap.conf中SERVERS的值。这个参数必须放在命令的末尾。

用passwd指令修改密码并不会修改用户对应的keytab文件中的密码,所以用passwd修改密码后原来的keytab文件就不能使用了。所以在修改密码的时候需要注意这一点,保证更新keytab。要保证keytab信息的同步您需要使用-k参数指定生成对应新密码的keytab信息。使用-k参数的同时使用-f可以指定新生成的keytab信息存放的路径,如果不用-f则默认将生成的keytab信息放在$HOME/.keytab下。此时,如果新生成的keytab信息和原keytab信息存放路径不同,请确保您使用新的keytab路径。

14. Kerberos用户密码管理

本章操作只对Kerberos数据库中保存的密码生效。如果您的集群部署了Kerberos with LDAP系统,使用本章操作会导致Kerberos和LDAP中密码不同步。所以如果您的集群部署了Kerberos with LDAP系统,请参考第 13 章 Kerberos with LDAP用户密码管理一章。

修改您自己在Kerberos数据库中的密码使用 `kpasswd` 指令。

举例:Alice修改自己在Kerberos数据库中的密码

```
[alice@tw-node118 ~]$ kpasswd
Password for alice@TDH:
Enter new password:
Enter it again:
Password changed.
```

15. Kerberos Ticket管理

本章介绍如何管理您的Kerberos Ticket,这里的Ticket是指Ticket-Granting-Ticket(TGT),是您访问集群中服务的凭证。我们假设您已经有自己的principal和密码(或者keytab),如果您还没有这些信息,请联系您的系统管理员获取。下面的指令您可以在安装了Kerberos或Kerberos with LDAP系统的集群中任意一台机器上执行。

15.1. 获取Ticket: kinit

如果您当前的session中没有ticket或者ticket已过期,您都需要使用kinit指令来获取ticket。您只需执行:

```
kinit <your_principal>
```

这里<your_principal>处提供您的principal并在系统提示下输入密码,即可获取一张ticket。

举例:Alice用户获取ticket:

```
[root@tw-node118 ~]# kinit alice@TDH
Password for alice@TDH:
```

如果您的密码存在keytab文件中并想要提供keytab文件进行认证,您需要执行:

```
kinit -kt <keytab_path> <your_principal>
```

这里<keytab_path>处提供您keytab文件的路径。这个路径可以是绝对路径也可以是相对路径。因为提供keytab文件就是在提供密码,所以您无需再输入密码。

举例:Alice用户通过提供keytab获取ticket:

```
[root@tw-node118 ~]# kinit -kt /root/.keytab alice@TDH
```

15.2. 查看Ticket: klist

要查看您当前的session是否有ticket以及ticket的有效期,您只需要在命令行执行klist。

举例:当前session没有ticket:

```
[root@tw-node118 ~]# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

举例:当前session有ticket:

```
[root@tw-node118 ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: alice@TDH
Valid starting
```

```
Expires
11/26/15 19:04:19 11/27/15 05:04:19
renew until 12/03/15 19:04:19
Service principal
krbtgt/TDH@TDH
```

在这里您可以看到ticket的过期时间,如果根据这个信息当前session中的ticket已经过期,您是无法访问服务的。您需要使用kinit指令重新获取ticket。

15.3. 销毁Ticket: kdestroy

当您结束对集群服务的使用,可以用kdestroy指令手动销毁您的ticket,以防别人持您的ticket来使用您的数据和应用。

举例

```
[root@tw-node118 ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: alice@TDH
Valid starting
Expires
11/26/15 19:11:31 11/27/15 05:11:31
renew until 12/03/15 19:11:31
[root@tw-node118 ~]# kdestroy
[root@tw-node118 ~]# klist
Service principal
krbtgt/TDH@TDH
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

当前session没有ticket说明ticket已经被销毁。

客户服务

技术支持

感谢你使用星环信息科技（上海）有限公司的产品和服务。如您在产品使用或服务中有任何技术问题，可以通过以下途径找到我们的技术人员给予解答。

Email：support@transwarp.io

技术支持热线电话：4008 079 976

技术支持QQ专线：3221723229, 3344341586

官方网址：www.transwarp.io

意见反馈

如果你在系统安装，配置和使用中发现任何产品问题，可以通过以下方式反馈：

Email：support@transwarp.io

感谢你的支持和反馈，我们一直在努力！

关于我们：

星环信息科技（上海）有限公司是一家大数据领域的高科技公司，致力于大数据基础软件的研发。星环科技目前掌握的企业级Hadoop和Spark核心技术在国内独树一帜，其产品Transwarp Data Hub (TDH)的整体架构及功能特性堪比硅谷同行，在业界居于领先水平，性能大幅领先Apache Hadoop，可处理从GB到PB级别的数据。星环科技的核心开发团队参与部署了国内最早的Hadoop集群，并在中国的电信、金融、交通、政府等领域的落地应用拥有丰富经验，是中国大数据核心技术企业化应用的开拓者和实践者。星环科技同时提供存储、分析和挖掘大数据的高效数据平台 和服务，立志成为国内外领先的大数据核心技术厂商。

行业地位：

来自知名外企的创业团队，成功完成近千万美元的A轮融资，经验丰富的企业级Hadoop发行版开发团队，国内最多落地案例。

核心技术：

高性能、完善的SQL on Hadoop、R语言的并行化支持，为企业数据分析与挖掘提供优秀选择。

应用案例：

已成功部署多个关键行业领域，包括电信、电力、智能交通、工商管理、税务、金融、广电、电商、物流等。

地址：上海市徐汇区桂平路481号18幢3层301室（漕河泾新兴技术开发区）
邮编：200233
电话：4008-079-976
网址：www.transwarp.io