
**Global Architecture & Technology Enablement Practice**

# Hadoop with Kerberos – Deployment Considerations

**Document Type: Best Practice**

**Note: The content of this paper refers exclusively to the second maintenance release (M2) of SAS 9.4.**

**Contact Information**

Name: Stuart Rogers

Title: Principal Technical Architect

Phone Number: +44 (0) 1628 490613

E-mail address: stuart.rogers@sas.com

Name: Tom Keefer

Title: Principal Solutions Architect

Phone Number: +1 (919) 531-0850

E-mail address: Tom.Keefer@sas.com

THE POWER TO KNOW®

# Table of Contents

# 1 Introduction

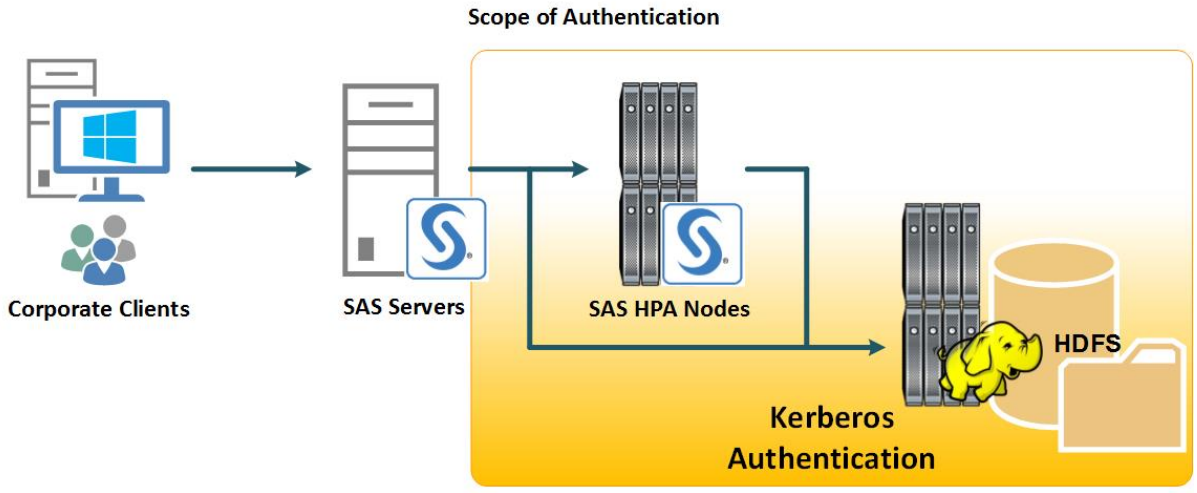
Note: The content of this paper refers exclusively to the second maintenance release (M2) of SAS 9.4.

## 1.1 Purpose of the Paper

This paper addresses the deployment of secure Hadoop environments with SAS products and solutions. In a secure Hadoop deployment, you enable Kerberos to provide strong authentication for the environment. This paper provides gives an overview of this process.

The paper also describes how to ensure that the SAS software components interoperate with the secure Hadoop environment. The SAS software components covered are SAS/ACCESS to Hadoop and SAS Distributed In-Memory Processes.

This paper is focuses on the Kerberos-based access to the Hadoop environment and does not cover using Kerberos authentication to access the SAS environment.




## 1.2 Deployment Considerations Overview

- Cloudera CDH requires an instance of the MIT Kerberos Key Distribution Center (KDC). The provided automated scripts will fail with other KDC distributions. Cloudera can interoperate with other Kerberos distributions, via a configured trust from the MIT Kerberos KDC to the other Kerberos distribution.
- Hortonworks requires more manual steps to configure Kerberos authentication than Cloudera. However, Hortonworks provides more flexibility in the Kerberos distribution, which can be directly used with Hortonworks.

- SAS does not directly interact with Kerberos. SAS relies on the underlying operating system and APIs to handle requesting tickets, managing ticket caches, and authenticating users.

- The operating system of the hosts, where either SAS Foundation or the SAS High-Performance Analytics root node will be running, must use Kerberos authentication. The Kerberos authentication used by these hosts either must be the same Kerberos realm as the secure Hadoop environment or have a trust that is configured against that Kerberos realm.

- The Kerberos Ticket-Granting Ticket (TGT), which is generated at the initiation of the user's session, is stored in the Kerberos ticket cache. The Kerberos ticket cache must be available to the SAS processes that connect to the secure Hadoop environment. Either the jproxy process started by SAS Foundation or the SAS High-Performance Analytics Environment root node need to access the Kerberos ticket cache.

- SAS Foundation on UNIX hosts must be configured for Pluggable Authentication Modules (PAM).

- On Linux and most UNIX platforms, the Kerberos ticket cache will be a file. On Linux, by default, this will be /tmp/krb5cc_<uid>_<rand>. By default on Windows, the Kerberos ticket cache that is created by standard authentication processing is in memory. Windows can be configured to use MIT Kerberos and then use a file for the Kerberos ticket cache.

- Microsoft locks access to the Kerberos Ticket-Granting Ticket session key when using the memory Kerberos Ticket Cache. To use the Ticket-Granting Ticket for non-Windows processes, you must add a Windows registry key in the Registry Editor.

- The SAS Workspace Server or other server started by the SAS Object Spawner might not have the correct value set for the KRB5CCNAME environment variable. This environment variable points to the location of the Kerberos ticket cache. Code can be added to the WorkspaceServer_usermods.sh to correct the value of the KRB5CCNAME environment variable.

- Kerberos attempts to use the highest available encryption strength for the Ticket-Granting Ticket. (In most cases, this is 256-bit AES.) Java, by default, cannot process 256-bit AES encryption. To enable Java processes to use the Ticket-Granting Ticket, you must download the Unlimited Strength Jurisdiction Policy Files and add them to the Java Runtime Environment. Due to import regulations in some countries, you should verify that the use of the Unlimited Strength Jurisdiction Policy Files is permissible under local regulations.

- There can be three different Java Runtime Environments (JRE) in use in the complete system. There is the JRE used by the Hadoop Distribution, the SAS Private JRE used by SAS Foundation, and the JRE used by the SAS High-Performance Analytics Environment. All of these JREs might require the Unlimited Strength Jurisdiction Policy Files.

- You need to regenerate the Hadoop configuration file (an XML file that describes the Hadoop environment) after Kerberos is enabled in Hadoop. The XML file used by

SAS merges several configuration files from the Hadoop environment. Which files are merged depends on the version of MapReduce that is used in the Hadoop environment.

- The SAS LIBNAME statement and PROC HADOOP statement have different syntax when connecting to a secure Hadoop environment. In both cases, user names and passwords are not submitted.
- An additional MPI option is required during the installation of SAS High-Performance Analytics infrastructure for environments that use Kerberos.
- The GRIDRSHCOMMAND option enables SAS Foundation to use an alternative SSH command to connect to the SAS High-Performance Analytics environment. You must use an alternative command when using Kerberos via GSSAPI. Using an alternative command such as /usr/bin/ssh can also provide more debug options.

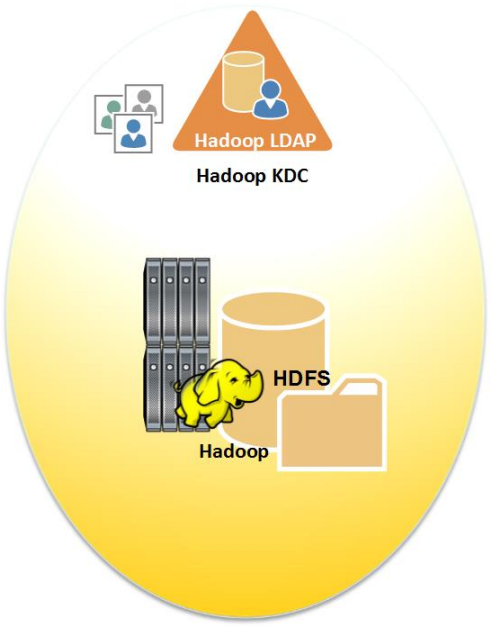
# 2 Hadoop Security Described

Currently Hadoop security is an evolving field. Most major Hadoop distributors are developing competing projects. Some examples of such projects are Cloudera Sentry and Apache Knox Gateway. A common feature of these security projects is that they have Kerberos enabled for the Hadoop environment.

The non-secure configuration relies on client-side libraries. As part of the protocol, these libraries send the client-side credentials as determined from the client-side operating system. While not secure, this configuration is sufficient for many deployments that rely on physical security. Authorization checks through ACLs and file permissions are still performed against the client-supplied user ID.

After Kerberos is configured, Kerberos authentication is used to validate the client-side credentials. This means that, when connecting to the client, you must request a Service Ticket that is valid for the Hadoop environment. The client submits this Service Ticket as part of the client connection. Kerberos provides strong authentication. Tickets are exchanged between client and server, and validation is provided by a trusted third party in the form of the Kerberos Key Distribution Center.

To create a new Kerberos Key Distribution Center specifically for the Hadoop environment, follow the standard instructions from the Cloudera or Hortonworks. See the following figure.



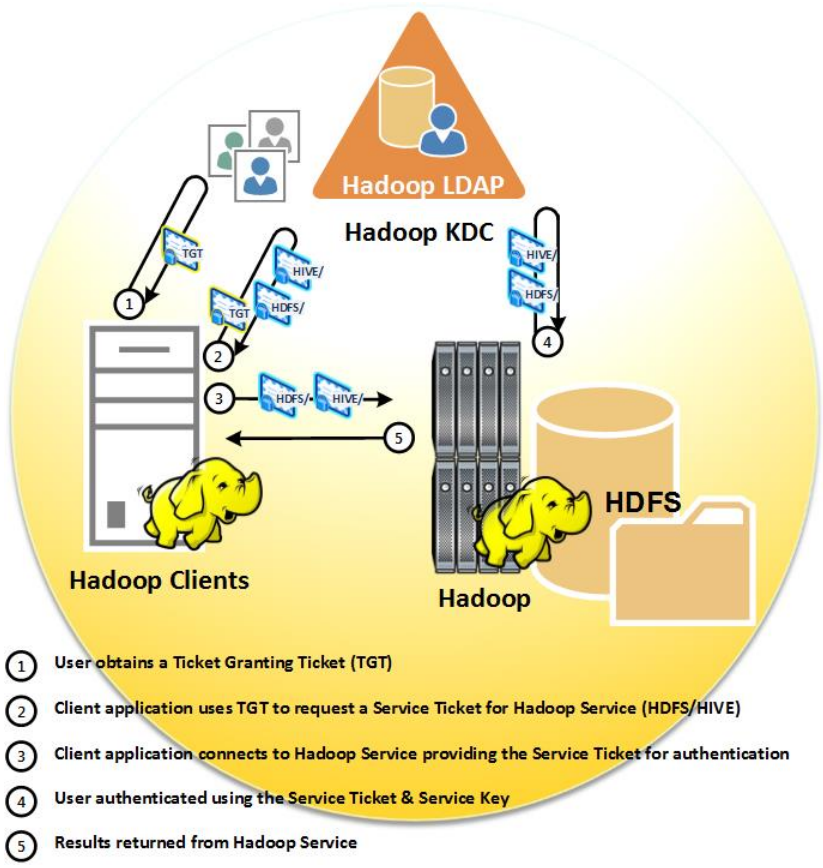

This process is used to authenticate both users and server processes. For example, with Cloudera 4.5, the management tools include all the required scripts to configure Cloudera to use Kerberos. Running these scripts after you register an administrator principal causes Cloudera to use Kerberos. This

process can be completed in minutes after the Kerberos Key Distribution Center is installed and configured.

## 2.1 Kerberos and Hadoop Authentication Flow

The process flow for Kerberos and Hadoop authentication is shown in the figure below. The first step, where the end user obtains a Ticket-Granting Ticket (TGT), does not necessarily occur immediately before the second step where the Service Tickets are requested. There are different mechanisms that can be used to obtain the TGT. Some customers have users run a kinit command after accessing the machine that is running the Hadoop clients. Other customers will integrate the Kerberos configuration in the host operating system setup. In this case, the action of logging onto the machine that is running the Hadoop clients generates the TGT.

After the user has a Ticket-Granting Ticket, the client application provides access to Hadoop services and initiates a request for the Service Ticket (ST). This ST request corresponds to the Hadoop service that the user is accessing. The ST is first sent, as part of the connection, to the Hadoop service. The Hadoop service then authenticates the user. The service decrypts the ST using the Service Key, which is exchanged with the Kerberos Key Distribution Center. If this decryption is successful, the end user is authenticated to the Hadoop Service.



## 2.2 Configuring a Kerberos Key Distribution Center

All supported Hadoop distributions recommend a separate Kerberos deployment. The key part of a Kerberos deployment is the Kerberos Key Distribution Center (KDC). With Microsoft Active Directory, Kerberos is tightly integrated into the Active Directory domain services. Each Active Directory domain already includes a Kerberos KDC. Alternatively, you can use either the MIT or Heimdal distributions of Kerberos to run a separate Kerberos KDC.

## 2.3 Cloudera CDH 4.5 Hadoop Configuration

Cloudera Manager can automatically complete most of the configuration for you. Cloudera does not provide instructions for the complete manual configuration of Kerberos, only for the automated approach that uses the Cloudera Manger. This means that if you don't use the specific approach detailed by Cloudera, you are left without documentation.

Cloudera expects the customer to use the MIT Kerberos, Release 5. Cloudera's solution for customers who want to integrate into a wider Active Directory domain structure is to implement a separate MIT Kerberos KDC for the Cloudera cluster. Then you implement the required trusts to integrate the KDC into the Active Directory. Using an alternative Kerberos distribution or even a locked-down version of the MIT distribution, as found in the Red Hat Identity Manager product, is not supported. The Cloudera scripts issue MIT Kerberos specific commands and fail if the MIT version of Kerberos is not present.

The Cloudera instructions tell the user to manually create a Kerberos administrative user for the Cloudera Manager Server. Then subsequent commands that are issued to the KDC are driven by the Cloudera scripts. The following principals are created by these scripts:

- HTTP/fullyqualified.node.names@REALM.NAME
- hbase/fullyqualified.node.names@REALM.NAME
- hdfs/fullyqualified.node.names@REALM.NAME
- hive/fullyqualified.server.name@REALM.NAME
- hue/fullyqualified.server.name@REALM.NAME
- impala/fullyqualified.node.names@REALM.NAME
- mapred/fullyqualified.node.names@REALM.NAME
- oozie/fullyqualified.server.name@REALM.NAME
- yarn/fullyqualified.node.names@REALM.NAME
- zookeeper/fullyqualified.server.name@REALM.NAME

Multiple principals are created for services that are running on multiple nodes, as shown in the list above by “fullyqualified.node.names." For example, with three HDFS nodes running on hosts chd01.exmaple.com, cdh02.exmaple.com, and chd03.example.com, there will be three principals: hdfs/ch01.example.com@EXAMPLE.COM, hdfs/cdh02.example.com@EXMAPLE.COM, and hdfs/ch03.example.com@EXAMPLE.COM.

In addition, the automated scripts create Kerberos Keytab files for the services. Each Kerberos Keytab file contains the resource principal’s authentication credentials. These Keytab files are then distributed across the Cloudera installation on each node. For example, for most services, on a data node, the following Kerberos Keytab files and locations are used:

- /var/run/cloudera-scm-agent/process/189-impala-IMPALAD/impala.keytab
- /var/run/cloudera-scm-agent/process/163-impala-IMPALAD/impala.keytab
- /var/run/cloudera-scm-agent/process/203-mapreduce-TASKTRACKER/mapred.keytab
- /var/run/cloudera-scm-agent/process/176-mapreduce-TASKTRACKER/mapred.keytab
- /var/run/cloudera-scm-agent/process/104-hdfs-DATANODE/hdfs.keytab
- /var/run/cloudera-scm-agent/process/109-hbase-REGIONSERVER/hbase.keytab
- /var/run/cloudera-scm-agent/process/121-impala-IMPALAD/impala.keytab
- /var/run/cloudera-scm-agent/process/192-impala-IMPALAD/impala.keytab
- /var/run/cloudera-scm-agent/process/200-hbase-REGIONSERVER/hbase.keytab
- /var/run/cloudera-scm-agent/process/216-impala-IMPALAD/impala.keytab
- /var/run/cloudera-scm-agent/process/173-hbase-REGIONSERVER/hbase.keytab
- /var/run/cloudera-scm-agent/process/182-yarn-NODEMANAGER/yarn.keytab
- /var/run/cloudera-scm-agent/process/168-hdfs-DATANODE/hdfs.keytab
- /var/run/cloudera-scm-agent/process/128-yarn-NODEMANAGER/yarn.keytab
- /var/run/cloudera-scm-agent/process/195-hdfs-DATANODE/hdfs.keytab
- /var/run/cloudera-scm-agent/process/209-yarn-NODEMANAGER/yarn.keytab
- /var/run/cloudera-scm-agent/process/142-hdfs-DATANODE/hdfs.keytab
- /var/run/cloudera-scm-agent/process/112-mapreduce-TASKTRACKER/mapred.keytab
- /var/run/cloudera-scm-agent/process/156-yarn-NODEMANAGER/yarn.keytab
- /var/run/cloudera-scm-agent/process/147-hbase-REGIONSERVER/hbase.keytab
- /var/run/cloudera-scm-agent/process/150-mapreduce-TASKTRACKER/mapred.keytab

All of these tasks are well managed by the automated process. The only manual steps are as follows:

- Create the initial administrative user.
- Create the HDFS Super User Principal.
- Get or create a Kerberos Principal for each user account.
- Prepare the cluster for each user account.

## 2.4 Hortonworks Data Platform 2.0 Configuration

Hortonworks does not automate the Kerberos configuration in the same way as Cloudera. Hortonworks provides a CSV-formatted file of all the required principal names and keytab files that are available from the Ambari Web GUI. The Service Principals for Hortonworks are as follows:

| Service | Component | Mandatory Principal Name |
|---|---|---|
| HDFS | NameNode | nn/*$FQDN* |
| HDFS | NameNode HTTP | HTTP/*$FQDN* |
| HDFS | SecondaryNameNode | nn/*$FQDN* |
| HDFS | SecondaryNameNode HTTP | HTTP/*$FQDN* |
| HDFS | DataNode | dn/*$FQDN* |
| MR2 | History Server | jhs/*$FQDN* |
| MR2 | History Server HTTP | HTTP/*$FQDN* |
| YARN | ResourceManager | rm/*$FQDN* |
| YARN | NodeManager | nm/*$FQDN* |
| Oozie | Oozie Server | oozie/*$FQDN* |
| Oozie | Oozie HTTP | HTTP/*$FQDN* |
| Hive | Hive Metastore<br>HiveServer2 | hive/*$FQDN* |
| Hive | WebHCat | HTTP/*$FQDN* |
| HBase | MasterServer | hbase/*$FQDN* |
| HBase | RegionServer | hbase/*$FQDN* |
| ZooKeeper | ZooKeeper | zookeeper/*$FQDN* |
| Nagios Server | Nagios | nagios/*$FQDN* |
| JournalNode Server | JournalNode | jn/*$FQDN* |

The principal names must match the values that are provided in the table. In addition, four special principals are required for Ambari:

| User | Mandatory Principal Name |
|---|---|
| Ambari User | ambari |
| Ambari Smoke Test User | ambari-qa |
| Ambari HDFS User | hdfs |
| Ambari HBase User | hbase |

The Kerberos Keytab file names required by Hortonworks are as follows:

| Component | Principal Name | Mandatory Keytab File Name |
|---|---|---|
| NameNode | nn/*$FQDN* | nn.service.keytab |
| NameNode HTTP | HTTP/*$FQDN* | spnego.service.keytab |
| SecondaryNameNode | nn/*$FQDN* | nn.service.keytab |
| SecondaryNameNode HTTP | HTTP/*$FQDN* | spnego.service.keytab |
| DataNode | dn/*$FQDN* | dn.service.keytab |
| MR2 History Server | jhs/*$FQDN* | jhs.service.keytab |
| MR2 History Server HTTP | HTTP/*$FQDN* | spnego.service.keytab |
| YARN | rm/*$FQDN* | rm.service.keytab |
| YARN | nm/*$FQDN* | nm.service.keytab |
| Oozie Server | oozie/*$FQDN* | oozie.service.keytab |
| Oozie HTTP | HTTP/*$FQDN* | spnego.service.keytab |
| Hive Metastore<br>HiveServer2 | hive/*$FQDN* | hive.service.keytab |
| WebHCat | HTTP/*$FQDN* | spnego.service.keytab |
| HBase Master Server | hbase/*$FQDN* | hbase.service.keytab |
| HBase RegionServer | hbase/*$FQDN* | hbase.service.keytab |
| ZooKeeper | zookeeper/*$FQDN* | zk.service.keytab |
| Nagios Server | nagios/*$FQDN* | nagios.service.keytab |
| Journal Server | jn/*$FQDN* | jn.service.keytab |
| Ambari User | ambari | ambari.keytab |
| Ambari Smoke Test User | ambari-qa | smokeuser.headless.keytab |
| Ambari HDFS User | hdfs | hdfs.headless.keytab |
| Ambari HBase User | hbase | hbase.headless.keytab |

Hortonworks expects the keytab files to be located in the /etc/security/keytabs directory on each host in the cluster. The user must manually copy the appropriate keytab file to each host. If a host runs more than one component (for example, both NodeManager and DataNode), the user must copy

keytabs for both components. The Ambari Smoke Test User, the Ambari HDFS User, and the Ambari HBase User keytabs should be copied to all hosts on the cluster. These steps are covered in the Hortonworks documentation under the first step entitled “Preparing Kerberos.”

The second step from the Hortonworks documentation is “Setting up Hadoop Users." This step covers creating or setting the principals for the users of the Hadoop environment. After all of the steps have been accomplished, Kerberos Security can be enabled in the Ambari Web GUI. Enabling Kerberos Security is the third and final step in the documentation.

# 3 SAS and Hadoop with Kerberos

This section deals with how SAS interoperates with the secure Hadoop environment. This document does not cover using Kerberos to authenticate into the SAS environment. The document only covers using Kerberos to authenticate from the SAS environment to the secure Hadoop environment.

## 3.1 User Kerberos Credentials

SAS does not directly interact with Kerberos. SAS relies on the underlying operating system and APIs to handle requesting tickets, managing ticket caches, and authenticating users. Therefore, the servers that host the SAS components must be integrated into the Kerberos realm, which has been configured for the secure Hadoop environment. This involves configuring the operating system's authentication processes to use either the same KDC as the secure Hadoop environment or a KDC with a trust relationship to the secure Hadoop environment.

### 3.1.1 Operating Systems and Kerberos Credentials

Linux environments are the supported operating systems for distributed SAS High-Performance Analytics environments. Integrating Linux operating systems into a Kerberos infrastructure requires that you configure Pluggable Authentication Modules (PAM). One recommended approach for both Red Hat Enterprise Linux and the SUSE Linux Enterprise Server is to use System Security Services Daemon (SSSD). SSSD provides a single interface into multiple sub-systems. An example of such a subsystem is NSS LDAP to retrieve user identity information from LDAP and pam_krb5, which allows for user authentication. With SSSD configured correctly, users logging into the servers are authenticated using Kerberos, and their Kerberos credentials are available to further processes through the Kerberos Ticket Cache.

With Windows systems, the authentication processing is tightly integrated with Active Directory and Microsoft's implementation of Kerberos. There is one difference to note between Windows and Linux: The Kerberos Ticket Cache on Windows is memory-based, but on Linux, it is file-based. Other operating systems such as AIX and Solaris have configuration options similar to those of Linux and provide a file-based Kerberos Ticket Cache.

The SAS processes that access the secure Hadoop environment must access the each user's Kerberos Ticket Cache. In Linux or UNIX environments, this typically means having access to the KRB5CCNAME environment variable, which points to a valid file. For Linux, the Kerberos Ticket Cache is typically /tmp/krb5cc_<uid>_<rand>.

In Windows environments, Microsoft manages access to the Kerberos Ticket Cache in memory. Microsoft does not allow access to the session key of the Ticket-Granting Ticket (TGT) to non-

Windows processes. So a Window Registry update is required for SAS to access the session key and hence use the TGT. The REG_DWORD key AllowTgtSessionKey registry key must be added to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters with a value of 1.

There is an alternative on Windows. Rather than using the integrated authentication processes tightly linked with Active Directory, you can use a separate deployment of MIT Kerberos to authenticate the user and then access the secure Hadoop environment. When using MIT Kerberos for Windows, the Kerberos Ticket Cache can either reference the memory location managed by Windows or be stored in a file as on Linux systems.

To configure MIT Kerberos to use the file system and for the file system Kerberos Ticket Cache to be used, complete the following steps:

1. Define environment variables for the MIT Kerberos so that it knows where to find the Kerberos configuration and where to put the Ticket Cache.

```
KRB5_CONFIG=C:\ProgramData\MIT\Kerberos5\krb5.ini
KRB5CCNAME=FILE:%USERPROFILE%\krb5cc_<username>
```

2. Run the following Kerberos commands from the bin subdirectory of the MIT Kerberos install directory.

   - kinit to create a ticket for the user
   - kinit –R to format the ticket correctly for Java

3. Create a JAAS configuration file with the following contents:

```
com.sun.security.jgss.initiate {
  com.sun.security.auth.module.Krb5LoginModule required
  useTicketCache=true
  doNotPrompt=true
  useKeyTab=false
  debug=true
  ;
};
```

4. Tell Java where the JAAS configuration file is located by doing either of the following:

   a. Set the property java.security.auth.login.config to the JAAS configuration file.

   - Example: Set the following property on the command line: Djava.security.auth.login.config="C:\ProgramData\MIT\Kerberos5\jaas.conf"

   b. Set the login.config.url.1 property in the java.security file for the JRE:

   - Example: login.config.url.1=file:C:\ProgramData\MIT\Kerberos5\jaas.conf

5. Tell Java where the Kerberos configuration file is located by doing either of the following:

a. Set the property java.security.krb5.conf to the Kerberos configuration file.

- Example: Set the property on the command line.
  Djava.security.krb5.conf="C:\ProgramData\MIT\Kerberos5\krb5.ini"

b. Place the kerberos configuration file in a known location. Here are two examples:

   - 'C:\Windows\krb5.ini'
   - '<JRE_HOME>\krb5.conf'

6. For debug purposes, you can add the following JVM parameters:

   - -Dsun.security.krb5.debug=true
   - -Dsun.security.jgss.debug=true

## 3.1.2 SAS Foundation Authentication Configuration

SAS Foundation on UNIX hosts must be configured to use Pluggable Authentication Modules (PAM). This ensures that the configured processes that link the operating system into the Kerberos realm are used by SAS Foundation when it authenticates the user. Details on configuring PAM are in the *Configuration Guide for SAS 9.4 Foundation for UNIX Environments*.

SAS Foundation can either be configured when you run the SAS Deployment Wizard to initially deploy the SAS server or configured as a manual step after the deployment is complete.

## 3.1.3 SAS Processes Accessing the Ticket Cache

To confirm that the Kerberos Ticket Cache is available, the following code can be submitted in SAS. This example is for submitting via SAS Studio:

```
%let krb5env=%sysget(KRB5CCNAME);
%put &KRB5ENV;
```

The SAS log contains the value of the KRB5CCNAME environment variable for the current user's SAS session. Here is an example of the output:

```
43          %let krb5env=%sysget(KRB5CCNAME);
44          %put &KRB5ENV;
FILE:/tmp/krb5cc_100001_ELca0y
```

This file can then be checked on the operating system to confirm that the correct Kerberos Ticket Cache is identified. If the incorrect Kerberos Ticket Cache is being passed in the KRB5CCNAME environment variable (or it is not being passed at all), code can be added to the start-up of the SAS session to correctly set the environment variable. For example, adding the following to `<SAS_CONFIG>/SASApp/WorkspaceServer/WorkspaceServer_usermods.sh` searches the /tmp directory for a valid Kerberos Ticket Cache for the user and sets the environment variable.

```
if [ -z ${KRB5CCNAME+x} ]; then
workspace_user=$(whoami)
workspace_user_ccaches=$(find /tmp -maxdepth 1 -user ${workspace_user} -
type f -name "krb5cc_*" -printf '%T@ %p\n' | sort -k 1nr | sed 's/^[^ ]*
//' | head -n 1)

if test ! -z "$workspace_user_ccaches"; then
            echo "Most recent krb5 ccache found for '${workspace_user}' at
'${workspace_user_ccaches}'."
            echo "Cache last modified: $(stat -c%y
${workspace_user_ccaches})"
            export KRB5CCNAME=$workspace_user_ccaches
            echo "KRB5CCNAME has been set to ${KRB5CCNAME}."
else
            echo "No krb5 credentials caches were found in /tmp for
'${workspace_user}'."
fi
fi
```

There are two different types of SAS processes that require access to the Kerberos Ticket Cache. First for LIBNAME statements, SAS Foundation launches a jproxy Java process. This process loads the Hadoop JAR files that are specified by the SAS_HADOOP_JAR_PATH environment variable. The jproxy process is then the client that connects to Hadoop. So it is this Java process that needs access to the Kerberos Ticket Cache.

The second type of process is a SAS High-Performance Analytics infrastructure process. It is launched by the SAS High-Performance Analytics Environment when a SAS In-Memory solution accesses the secured Hadoop environment. When you use the SAS Embedded Process, only the process that runs on the root node makes a connection to the secured Hadoop environment. The SAS Embedded Process then connects back from the secured Hadoop environment to the worker nodes. When you access HDFS directly, to read or write a sashdat file, the root node connects to the secured Hadoop environment using Kerberos. The worker nodes connect to the data nodes using internal Hadoop tokens.

## 3.2 Encryption Strength

Kerberos strong authentication relies on encryption. Different strengths of encryption can be used with the Kerberos tickets. By default, Kerberos will attempt to use 256-bit AES encryption with the Kerberos Ticket Granting Ticket. However, Java cannot process this strength of encryption, which means that the SAS LIBNAME statement will fail.

Enabling Java to process the 256-bit AES TGT requires the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. These files can be downloaded from Oracle for most operating systems. However, for AIX, because the IBM JRE is used, the Policy Files can be

downloaded from IBM. The files must then be copied into the Java Runtime Environment lib/security subdirectory. Due to import regulations in some countries, you should verify that the use of the Unlimited Strength Jurisdiction Policy Files is permissible under local regulations.

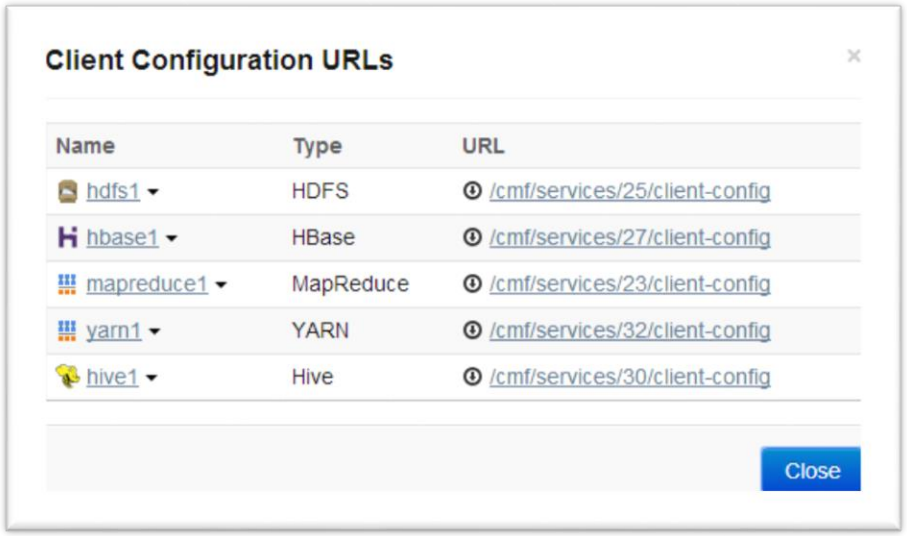
The Unlimited Strength Jurisdiction Policy Files are required by the SAS Private Java Runtime Environment. This environment is used by SAS Foundation for issuing a LIBNAME statement to the secure Hadoop environment. If the SAS Grid Manager is licensed, the JCE Unlimited Strength Policy files will be needed on all machines. They are required for all instances of SAS Foundation that might issue a LIBNAME statement. In addition, the SAS High-Performance Analytics Environment, when accessing the SAS Embedded Process, uses a Java Runtime Environment. This also requires the Unlimited Strength Jurisdiction Policy Files.

## 3.3 Hadoop Configuration File

SAS requires a number of configuration options to pass to the SAS processes that connect to the secure Hadoop environment. The best practice is to provide these configuration options in the form of a configuration file with the LIBNAME statement or PROC HADOOP code. The best practice mechanism to generate the configuration file is from the Hadoop distributions management interface or directly from the Hadoop environment.

To generate the configuration file for Cloudera:

1. For Cloudera, from the Status page of the Cloudera Manager application, select the required cluster and select **View Client Configuration URLs**.




2. From the pop-up window, select the link to the appropriate service to download a ZIP file that contains the service configuration files.

- If you use MapReduce 1, merge the properties from the Hadoop core, Hadoop HDFS, and MapReduce configuration files into a single configuration.
- If you use MapReduce 2, merge the properties from the Hadoop core, Hadoop HDFS, MapReduce 2, and YARN configuration files into a single configuration file.

3. After the merged configuration file is available, place it in a location that can be read by all users running either the LIBNAME statement or PROC HADOOP code. If SAS Grid Manager is used to access the secure Hadoop environment, then every grid node that runs SAS Code requires access to the file.

To generate the configuration file for Hortonworks:

1. For Hortonworks, the Ambari interface does not provide a simple mechanism to collect the client configuration files. The configuration files should be found under the /etc folder structure in the Hadoop environment. Retrieve the Hadoop core, Hadoop HDFS, and MapReduce 1 configuration files from the Hadoop environment and merge into a single configuration file.
2. After the merged configuration file is available, place it in a location that can be read by all users running either the LIBNAME statement or PROC HADOOP code. If SAS Grid Manager is used to access the secure Hadoop environment, then every grid node running SAS Code requires access to the file.

## 3.4 SAS LIBNAME with Secured Hadoop

The SAS LIBNAME for Hadoop must be updated so that it works in a secure Hadoop environment. The original configuration XML file does not contain the Kerberos parameters. So the configuration file must be updated after Kerberos is enabled in Hadoop. The configuration file location is provided to the LIBNAME statement via the environment variable SAS_HADOOP_CONFIG_PATH.

In the default setup, the USER and PASSWORD options are provided on the connection. These are not valid for Kerberos connections and must be removed from the LIBNAME statement. Instead the HDFS_PRINCIPAL and/or HIVE_PRINCIPAL are specified.

Example LIBNAME statements:

```
/* HIVE Server 2 Libname */
libname HIVE hadoop server="gatecdh01.gatehadoop.com"
HDFS_PRINCIPAL="hdfs/_HOST@GATEHADOOP.COM"
HIVE_PRINCIPAL="hive/_HOST@GATEHADOOP.COM"
subprotocol=hive2;

/* HDFS Libname */
libname HDFS hadoop server="gatecdh01.gatehadoop.com"
HDFS_PRINCIPAL="hdfs/_HOST@GATEHADOOP.COM"
HIVE_PRINCIPAL="hive/_HOST@GATEHADOOP.COM"
```

```
HDFS_TEMPDIR="/user/sasdemo/temp"
HDFS_METADIR="/user/sasdemo/meta"
HDFS_DATADIR="/user/sasdemo/data";
```

The example above illustrates the two types of Hadoop LIBNAME statements that can be used. The first connects via HiverServer2. The second connects directly to HDFS. Use the second format when SAS maintains an XML-based metadata description of HDFS files and tables. You can create XML-based metadata with PROC HDMD. The filetype for an XML-based metadata description that PROC HDMD produces is SASHDMD (for example, product_table.sashdmd). Another name for this metadata is a SASHDMD descriptor.

To debug issues with the LIBNAME statement or SAS/ACCESS, include the following option statement at the beginning of your submitted code:

```
option SASTRACE = "d,d,d,d" sastraceloc=saslog;
```

To echo to the SAS log the location of the Hadoop configuration file used by SAS Foundation, include the following two lines in your submitted code:

```
%let CONFIG_PATH=%sysget(SAS_HADOOP_CONFIG_PATH);
%put &CONFIG_PATH;
```

To echo to the SAS log the location of the Hadoop JAR files used by SAS Foundation, include the following two lines in your submitted code:

```
%let JAR_PATH=%sysget(SAS_HADOOP_JAR_PATH);
%put &JAR_PATH;
```

## 3.5 PROC HADOOP with Secured Hadoop

The HADOOP procedure enables you to submit HDFS commands, MapReduce programs, and Pig language code against Hadoop data. As with the LIBNAME statement for accessing secure Hadoop environments, you should drop the USERNAME and PASSWORD options. The XML configuration file provides all the required options, so relatively simple code can be used:

```
filename cfg "/opt/sas/userContent/HadoopKerberosConfig.xml";

PROC HADOOP options=cfg verbose;
      hdfs mkdir='/user/sasdemo/hdfs_test';
run;
```

This code submits the HDFS make directory command and creates the directory specified.

## 3.6 The SAS High-Performance Analytics Infrastructure Installation Option

The installation of the SAS High-Performance Analytics Environment prompts you for additional options to mpirun. When you configure with Kerberos, the following option should be added at this prompt:

```
-genvlist `env | sed -e s/=.*/,/ | sed /KRB5CCNAME/d | tr -d
'\n'`TKPATH,LD_LIBRARY_PATH
```

This option is covered in the *SAS High-Performance Analytics Infrastructure – Installation and Configuration Guide*. (Access instructions are in the Instructions.html file of your deployment.) The option can be checked in an installed environment by reviewing the /opt/sas/TKGrid/tkmpirsh.sh script and examining this line:

```
export MPI_OPTIONS="$MPI_OPTIONS -genv DISPLAY=$DISPLAY -genvlist `env |
sed -e s/=.*/,/ | sed /KRB5CCNAME/d | tr -d '\n'`TKPATH,LD_LIBRARY_PATH"
```

## 3.7 GRID Options with Secured Hadoop

Code submitted to SAS In-Memory solutions requires you to specify a set of GRID options. These GRID options are used to initiate communications from SAS Foundation to the SAS High-Performance Analytics environment. The standard set of options used are as follows:

```
option set=GRIDHOST="gatesas01.gatehadoop.com";
option set=GRIDINSTALLLOC="/opt/sas/TKGrid_2.5/TKGrid_REP";
option set=GRIDDATASERVER="gatecdh01.gatehadoop.com";
option set=GRIDMODE="ASYM";
```

These options are required to run SAS In-Memory solutions alongside or asymmetrically against a Hadoop environment. GRIDHOST sets the host with the root node of the SAS High-Performance Analytics Environment. The GRIDDATASERVER sets the host of the Hadoop name node. GRIDINSTALLLOC specifies the install location of the SAS High-Performance Analytics Environment. And the GRIDMODE option tells the code to run in asymmetric mode. If the SAS In-Memory code is running while co-located with Hadoop, only the GRIDHOST and GRIDINSTALLLOC are required.

The option GRIDRSHCOMMAND can be used to set the SSH command used by SAS Foundation to initialize the connection to the SAS High-Performance Analytics Environment. By default, SAS Foundation uses a built-in SSH command to make this connection. This option enables you to use an alternative SSH command and allows debug options to be specified on the SSH command. To use Kerberos for the connection to the SAS High-Performance Analytics Environment via the GSSAPI, you must specify an alternative command. In addition, specific options can be passed to the SSH command to prevent authentication with password or public keys:

```
option set=GRIDRSHCOMMAND="/usr/bin/ssh -o StrictHostKeyChecking=no -o
PasswordAuthentication=no -o PubkeyAuthentication=no";
```

Adding –vvv after the SSH command enables verbose debugging for the SSH command, and this is returned to the SAS log:

```
option set=GRIDRSHCOMMAND="/usr/bin/ssh -vvv -o StrictHostKeyChecking=no -
o PasswordAuthentication=no -o PubkeyAuthentication=no";
```

# 4 References

Cloudera Inc. 2014. *Configuring Hadoop Security with Cloudera Manager*. Palo Alto, CA: Cloudera Inc.

Hortonworks, Inc. 2014. "Setting Up Kerberos for Hadoop 2.x." *Hortonworks Data Platform: Installing Hadoop Using Apache Ambari*. Palo Alto, CA: Hortonworks, Inc.

LWN.net. 2011. "SSSD: System Security Services Daemon." Eklektix Inc.

LWN.net. 2014. "How to configure sssd on SLES 11 to resolve names and authenticate to Windows 2008 and Active Directory." Novell, Inc.

Red Hat, Inc. 2013. "Chapter 12. Configuring Authentication." *Red Hat Linux 6: Deployment Guide, 5th ed*. Raleigh, NC: Red Hat Inc.

SAS Institute Inc. 2014. "LIBNAME Statement Specifics for Hadoop." *SAS/ACCESS 9.4 for Relational Databases: Reference, 4th ed.* Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2014. *Configuration Guide for SAS 9.4 Foundation for UNIX Environments*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2014. *SAS 9.4 In-Database Products: Administrator's Guide, 4th ed.* Cary, NC: SAS Institute Inc.

# 5 Recommended Reading


- SAS Institute Inc. Hadoop: What it is and why it matters. Cary, NC: SAS Institute, Inc.
- SAS Institute Inc. SAS 9.4 Support for Hadoop. Cary, NC: SAS Institute, Inc.
- SAS Institute Inc. SAS In-Memory Statistics for Hadoop. Cary, NC: SAS Institute, Inc.
- SAS Institute Inc. 2014. "Hadoop Procedure." *Base SAS 9.4 Procedures Guide, Third Edition.* Cary, NC: SAS Institute, Inc.

# 6 Credits and Acknowledgements

It would have been impossible to create this paper without the invaluable input of the following people:

- Evan Kinney, SAS R&D
- Larry Noe, SAS R&D
- Doug Haig, SAS R&D

SAS INSTITUTE INC. WORLD HEADQUARTERS SAS CAMPUS DRIVE CARY, NC 27513
TEL: 919 677 8000 FAX: 919 677 4444 U.S. SALES: 800 727 0025 **WWW.SAS.COM**