Global Architecture and Technology Enablement Practice

Hadoop with Kerberos – Architecture 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

Table of Contents

1 Introduction

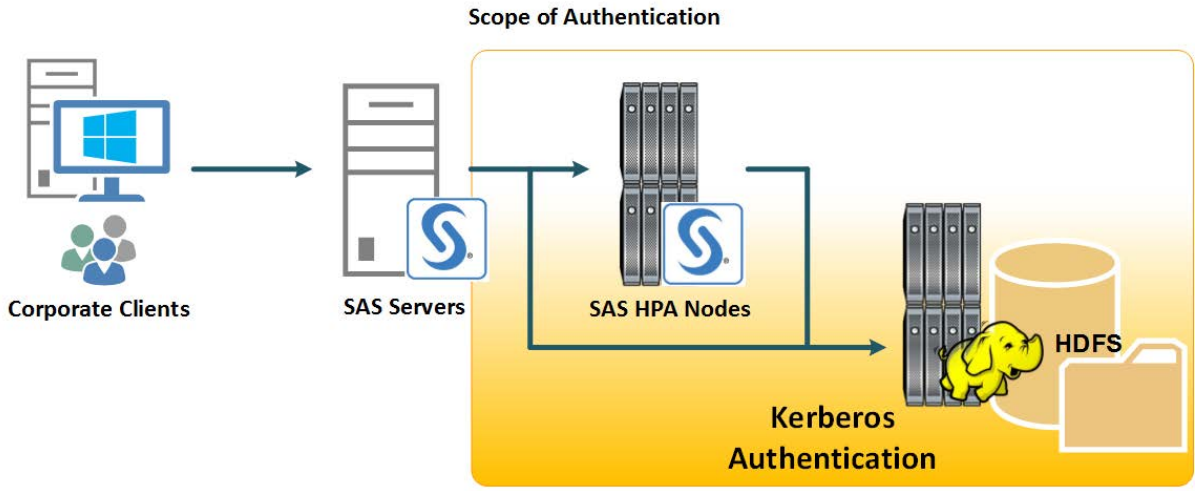
Note: The content of this paper refers exclusively to the second maintenance release (M2) of SAS 9.4.

1.1 Purpose of the Paper

This paper addresses the architecture considerations for setting up secure Hadoop environments with SAS products and solutions. Secure Hadoop refers to a deployment of Hadoop in environments where Kerberos has been enabled to provide strong authentication.

This paper includes the questions that you must address early in the design of your target environment. Responses to these questions will direct the deployment and configuration of the SAS products and solutions. The details of SAS deployment are outside the scope of this document and are covered in the Deployment Considerations document.

Using Kerberos with Hadoop does not necessarily mean that Kerberos will be used to authenticate users into the SAS part of the environment. The Kerberos authentication takes place between SAS and Hadoop. (You can use Kerberos between the client and SAS to provide end-to-end Kerberos authentication. But this, too, is outside the scope of this document.)



In the secure Hadoop environment, SAS interacts in a number of ways. First, SAS code can be written to use SAS/ACCESS to Hadoop. This can make use of the LIBNAME statement or PROC Hadoop statement. The LIBNAME statement can connect directly to HDFS, to HIVE, or to HIVE Server 2. This SAS code can be processed interactively or in batch, or it can be distributed with SAS Grid Manager.

SAS In-Memory solutions can leverage a SAS High-Performance Analytics Environment and connect to the secure Hadoop environment. The SAS High-Performance Analytics nodes can connect in parallel to the secure Hadoop environment to process data. This connection can again be directly to HDFS, via HIVE, or via HIVE Server2.

The first section of the paper provides a high-level overview of a secure Hadoop environment. The following sections address architecture considerations.

1.2 Architecture Overview

- SAS does not directly process Kerberos tickets. It relies on the underlying operating system and APIs.
- The operating system of SAS hosts must be integrated into the Kerberos realm structure of the secure Hadoop environment.
- A user repository that is valid across all SAS and Hadoop hosts is recommended rather than the use of local accounts.
- SAS does not directly interact with Kerberos. Microsoft Active Directory, MIT Kerberos, or Heimdal Kerberos can be used.
- The SAS process, either Java or C, must have access to the user's Ticket Granting Ticket (TGT) via the Kerberos credentials cache.
- The SAS Java process needs the addition of the Unlimited Strength Encryption Policy files to work with 256-bit AES encryption.

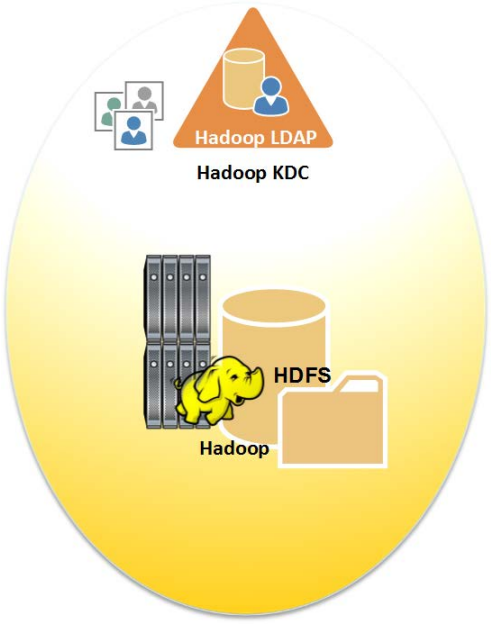
2 Hadoop Security

Hadoop Security is an evolving field with most major Hadoop distributors developing competing projects. Some examples of such projects are Cloudera Sentry and Hortonworks Knox Gateway. A common feature of these security projects is that they are based on having Kerberos enabled for the Hadoop environment.

The non-secure configuration relies on client-side libraries to send the client-side credentials as determined from the client-side operating system as part of the protocol. 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 the client must request a Service Ticket valid for the Hadoop environment and submit this Service Ticket as part of the client connection. Kerberos provides strong authentication in which tickets are exchanged between client and server. 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 results. See the following figure.

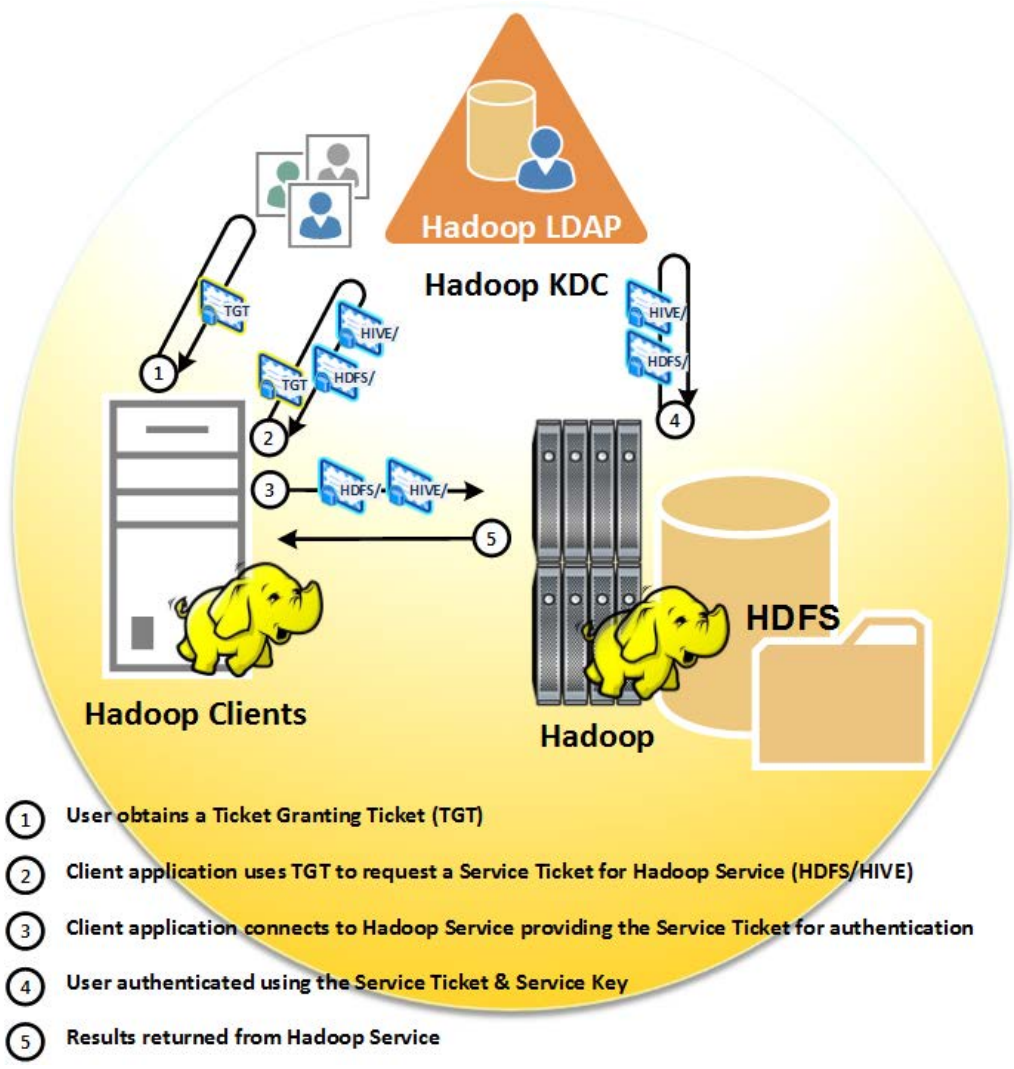


The Kerberos Key Distribution Center is used to authenticate both users and server processes. For example, the Cloudera 4.5 management tools include all the required scripts that are needed to configure Cloudera to use Kerberos. When you want Cloudera to use Kerberos, run these scripts after you register an administrator principal. This process can be completed in minutes after the Kerberos Key Distribution Center has been installed and configured.

2.1 Kerberos and Hadoop Authentication Flow

The process flow for Kerberos and Hadoop authentication is shown in the diagram 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 users run a kinit command after accessing the machine running the Hadoop clients. Others integrate the Kerberos configuration in the host operating system setup. In this case, the action of logging on to the machine that runs the Hadoop clients will generate the TGT.

After the user has a Ticket-Granting Ticket, the client application access to Hadoop Services initiates a request for the Service Ticket (ST) that corresponds to the Hadoop Service the user is accessing. The ST is then sent as part of the connection to the Hadoop Service. The corresponding Hadoop Service must then authenticate the user by decrypting the ST using the Service Key exchanged with the Kerberos Key Distribution Center. If this decryption is successful the end user is authenticated to the Hadoop Service.



3 Architecture Considerations

The architecture for a secure Hadoop environment will include various SAS software products and solutions. At the time of writing the products and solutions covered are as follows:

- SAS/ACCESS to Hadoop
- SAS High-Performance Analytics
- SAS Visual Analytics and SAS Visual Statistics

3.1 SAS and Kerberos

SAS does not manage Kerberos ticket caches, nor does it directly request Kerberos Tickets. This is an important factor when you are considering how SAS will interact with a secure Hadoop environment. Some software vendors maintain their own ticket cache and deal with requesting Kerberos tickets directly. SAS does not do this. It relies on the underlying operating system and APIs to manage the Kerberos ticket caches and requests. By definition, there can be a delay between the initial authentication process with the Kerberos Key Distribution Center (KDC) and any subsequent request for a Service Ticket (ST). The initial Ticket Granting Ticket (TGT) must be put somewhere, so it is put is the ticket cache. In Windows environments, this a memory location. On most UNIX operating systems, this will be a file. Alternative configurations are possible with Windows to switch to using a file-based ticket cache.

If the SAS process cannot access the ticket cache, then the process cannot use the TGT to request an ST. There are two types of SAS processes that need access to the ticket cache. The first is launched by SAS Foundation when processing a Hadoop LIBNAME statement. The second is launched by a SAS High-Performance Analytics Environment when an In-Memory Solution attempts to access Hadoop. Both of these processes must be able to access the ticket cache.

The following sections detail the architecture considerations for initializing these Kerberos ticket caches via the request for a TGT and then making them available to the SAS process.

3.2 User Repositories

In a secure Hadoop environment, the strong authentication provided by Kerberos means that processes will run as individual users across the Hadoop environment. Local user accounts can be used, but maintaining these accounts across a large number of hosts increases the chance for error. Therefore, it is recommended that you use a user repository to provide a central store for user details about the environment. This can either be an isolated user repository specifically for the Hadoop

environment or the general corporate user repository. Knowing what type of user repository is being used is important for the configuration of the operating system across the environment.

The user repository can be LDAP or Active Directory. The benefit of using Active Directory is that this includes all of the Kerberos Key Distribution Center infrastructure. If you use an LDAP repository, you will have to use a separate implementation of the Kerberos Key Distribution Center. One drawback to using Active Directory is that the domain database does not normally store the required POSIX user attributes. These attributes will be required for all users of the secure Hadoop environment. These POSIX user attributes are required because the users will running operating system processes on the secure Hadoop environment. Microsoft provides details of mechanisms to use to store the POSIX attributes in the Active Directory.

3.3 Kerberos Distribution

You have three main options when it comes to the distribution of Kerberos used in the environment. The first option, if Active Directory is used as the user repository, is to use the Microsoft implementation of Kerberos, which is fully integrated into Active Directory. Alternatively, if an LDAP repository is used, either the MIT or Heimdal distributions of Kerberos can be used. SAS is agnostic to the distribution of Kerberos.

3.4 Operating System Integration with Kerberos

As stated above SAS does not directly interact with the Kerberos Key Distribution Center (KDC) and initiate ticket requests. SAS operates through the standard GSSAPI and operating systems calls. Therefore, a key prerequisite is for the operating system to be correctly integrated with your chosen user repository and Kerberos distribution. There are many different ways this can be accomplished and SAS does not require any specific mechanism be used. The only requirements are that a Ticket-Granting Ticket (TGT) is generated as part of the user's session initialization and that this TGT is made available via the ticket cache.

All hosts that run SAS Foundation for SAS/ACCESS to Hadoop processing must be integrated with Kerberos. If you have SAS Grid Manager licensed, all grid nodes accessing the secure Hadoop environment must be integrated with Kerberos. For SAS High-Performance Analytics Environments all the nodes in the environment must be integrated with Kerberos and the SSH intercommunication must use Kerberos rather than SSH keys. In addition, in the SAS High-Performance Analytics Environment, the SAS Foundation hosts must also be integrated with Kerberos because they will initially run the Hadoop LIBNAME statement.

3.5 Kerberos Topology

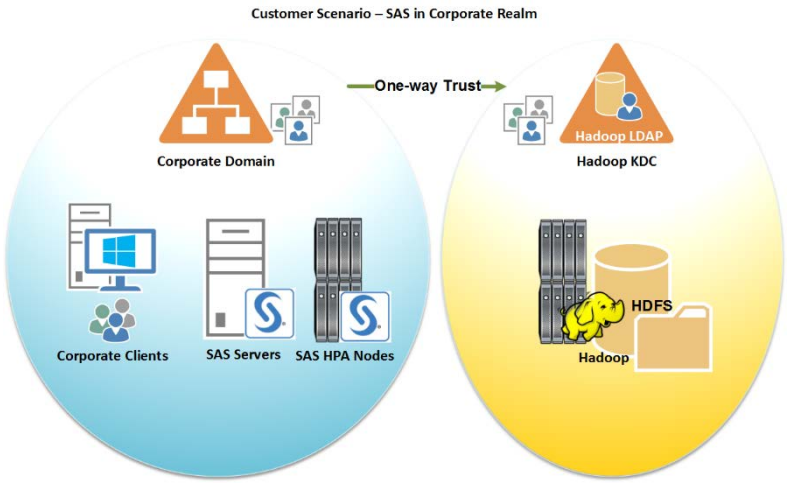
The key consideration for the integration of the operating systems with the Kerberos deployment for the secure Hadoop environment is where the different components are located. You can place the servers into different domains and those domains might or might not reflect the Kerberos realm setup. A domain is a group of computers, functioning and administered as a unit, that are identified by sharing the same common communications address. A domain does not have to be the same as a Kerberos realm and a domain qualified host name does not have to directly reflect the Kerberos realm a machine is a member of.

The Kerberos realm defines an instance of a Kerberos Key Distribution Center (KDC) and the database of principals associated with that. One realm can have one or more KDCs in the same way a domain can have one or more domain controllers. Because Active Directory tightly integrates Kerberos, each Active Directory domain will also be a Kerberos realm. If LDAP is used rather than Active Directory, there might not be close coupling between Kerberos realms and domains.

3.5.1 SAS in the Corporate Realm

In our first example, the SAS servers and the SAS High-Performance Analytics environment are part of the standard corporate domain. These SAS servers link their operating systems into the corporate domain structure. This enables the standard domain accounts to access the SAS servers and run SAS processes. However, the standard documentation for enabling Kerberos with Hadoop has been followed and an additional Kerberos realm is configured with the Hadoop environment located in this other realm.

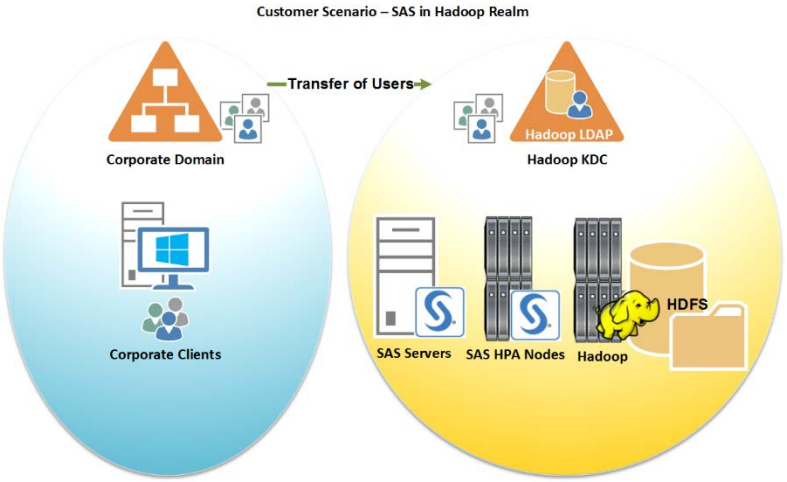
With separate realms for users, for one realm to access resources in another realm, a cross-realm trust must be configured. This is outside the scope of the SAS configuration and must be configured by the administrators of the two realms. After this cross-realm trust is in place, the users in the corporate realm can request a Ticket-Granting Ticket (TGT) for the Hadoop realm. Then they can obtain Service Tickets (ST) for the Hadoop environment.



The SAS servers are unable to access the Hadoop environment until this cross-realm trust is in place; in addition to the operating system of the servers being integrated with the corporate realm. This type of topology presents challenges for the initial configuration of the cross-realm trust. You need to work with your Kerberos administrators to ensure that everything is in place before the SAS configuration can succeed.

3.5.2 SAS in the Hadoop Realm

An alternative to placing the SAS Servers and High-Performance Analytics Environment in the corporate realm is to place them in the same realm as Hadoop. This greatly simplifies the initial configuration because after the operating system of the SAS hosts has been integrated, SAS can access the Hadoop environment.



The challenge with this topology is managing the user accounts within the Hadoop realm. Each user will have two sets of credentials: one that is valid in the corporate realm and the other that is valid in the Hadoop realm. To log on to the environment, users in the SAS environment need to provide a username and password that are valid in the Hadoop realm. Having a separate set of credential for the Hadoop realm could be ideal if you want the Kerberos authentication realm to be separate from the main corporate domain.

This topology, at the time of writing, is the most common topology chosen. The isolation of the Hadoop realm meets a number of security requirements and by including the SAS environments in this realm, the configuration is simplified.

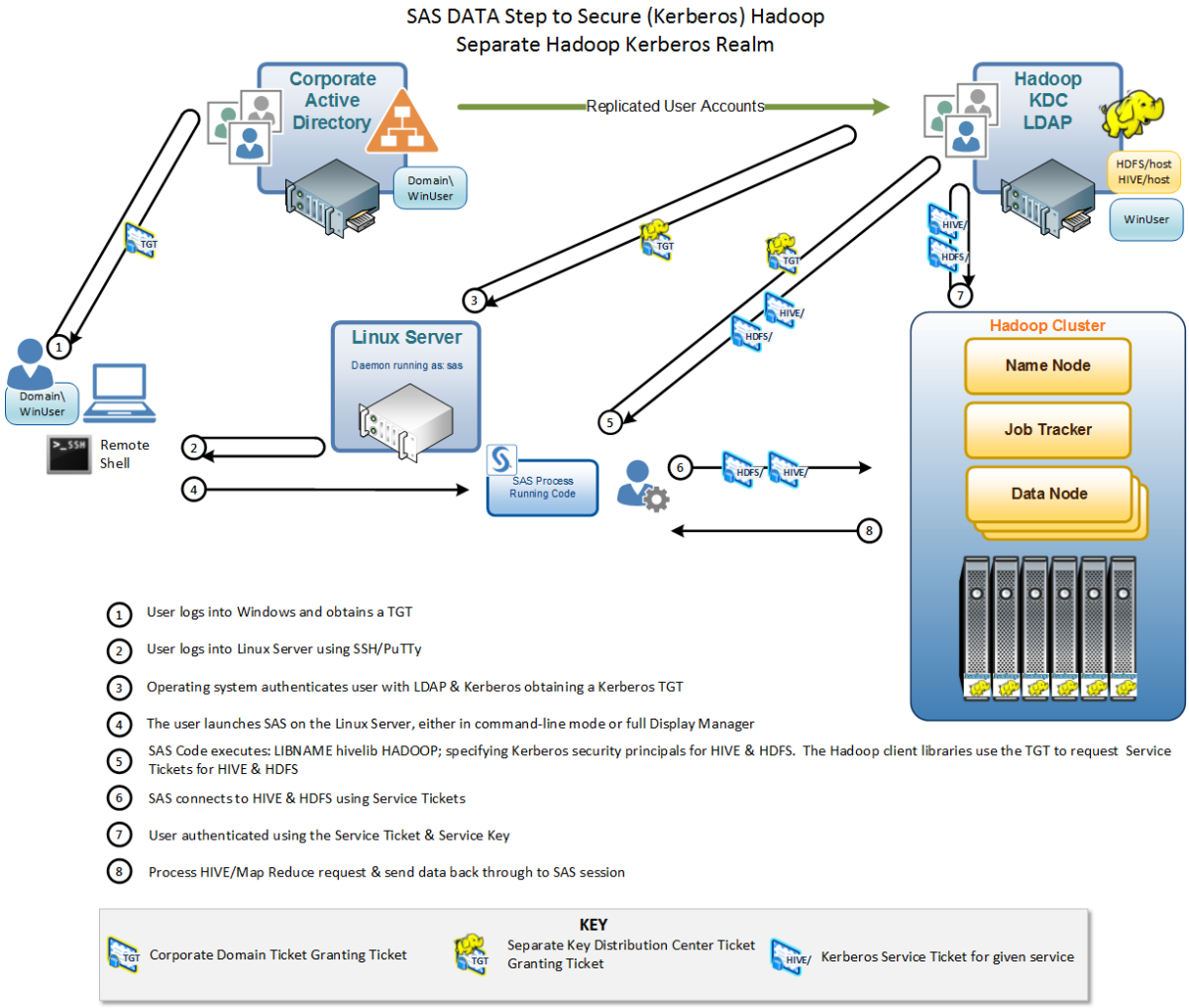
3.6 Encryption Strength and Java

The jproxy process started by SAS Foundation is one of the SAS processes which needs to interact with the secure Hadoop environment. The jproxy process is launched for example when a LIBNAME statement to Hadoop is submitted. Due to export limitations Java is unable to process the strongest

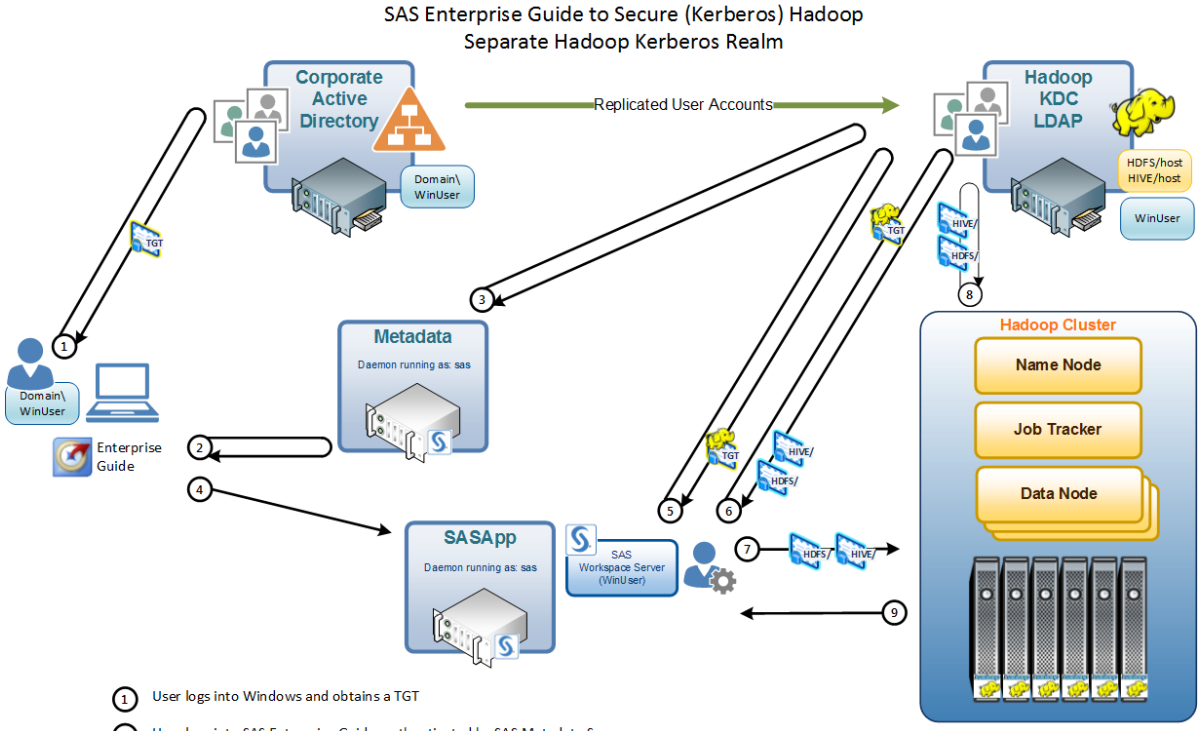
encryption available with Kerberos. Most Kerberos deployments will attempt to use the highest level of encryption possible-- AES 256-bit. By default Java is only able to work up to AES 128-bit. Therefore, the Unlimited Strength Encryption policy files must be added to the Java distribution for the AES 256-bit Kerberos tickets to be processed. For SAS systems running on AIX, these files are available from IBM. For all other operating systems these policy files are available from Oracle. 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.

4 Example Authentication Flows: Single Realm

4.1 SAS DATA Step to Secure Hadoop



4.2 SAS Enterprise Guide to Secure Hadoop



1. User logs into Windows and obtains a TGT
2. User logs into SAS Enterprise Guide, authenticated by SAS Metadata Server
3. SAS Metadata Server uses Operating System to validate user, which in turn uses PAM
4. SAS Enterprise Guide connects to the SAS Object Spawner and requests a SAS Workspace Server, passing credentials (WinUser)
5. SAS Object Spawner runs a shell script as the user and spawns a continuously running SAS Workspace Server. A Kerberos TGT is generated by PAM
6. SAS Code executes: LIBNAME hivelib HADOOP; specifying Kerberos security principals for HIVE & HDFS. The Hadoop client libraries use the TGT to request Service Tickets for HIVE & HDFS
7. SAS connects to HIVE & HDFS using Service Tickets
8. User authenticated using the Service Ticket & Service Key
9. Process HIVE/Map Reduce request & send data back through SAS Workspace Server to client

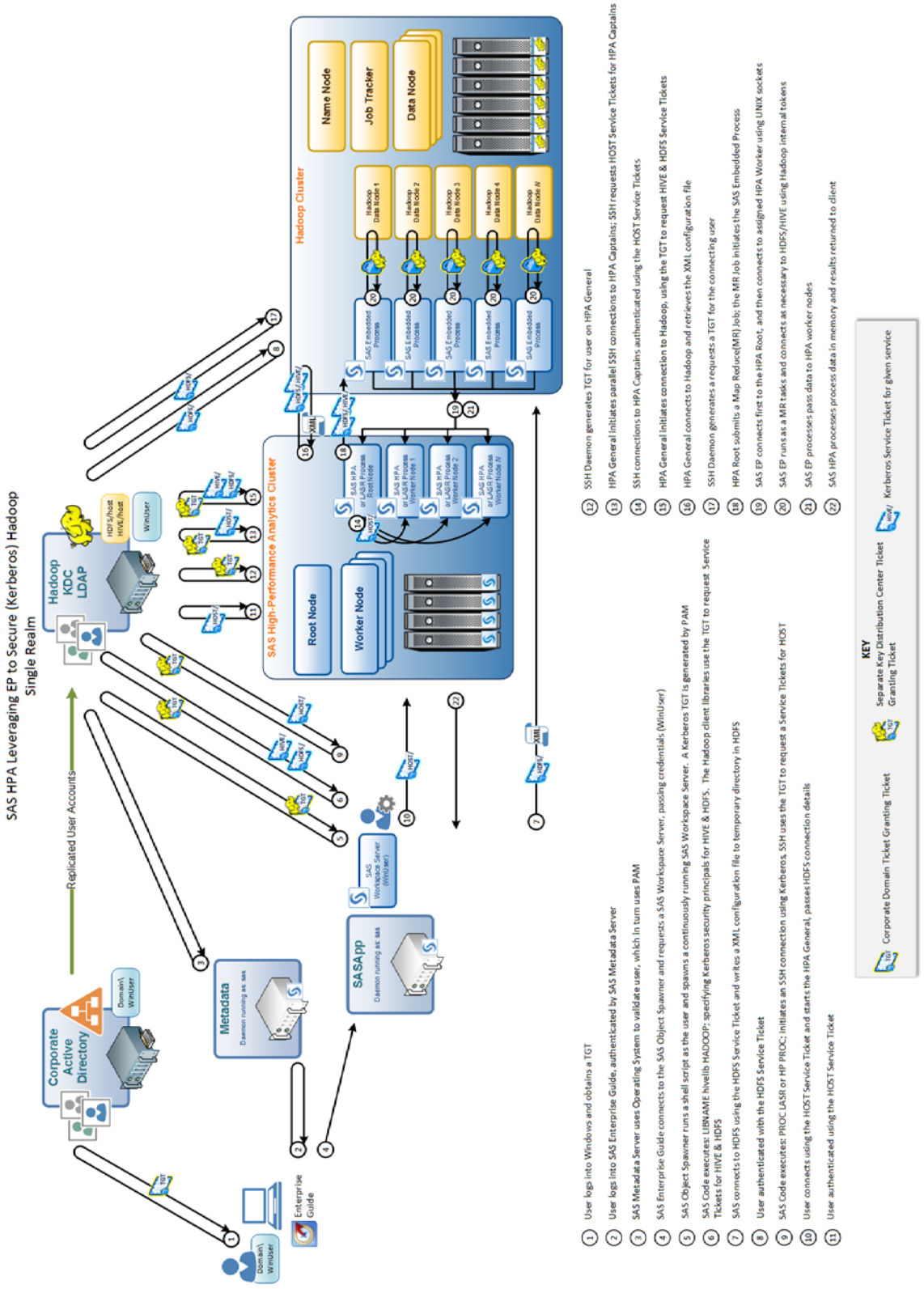
KEY

TGT Corporate Domain Ticket Granting Ticket

TGT Separate Key Distribution Center Ticket Granting Ticket

HIVE/ Kerberos Service Ticket for given service

4.3 SAS High-Performance Analytics



(1) User logs into Windows and obtains a TGT
(2) User logs into SAS Enterprise Guide, authenticated by SAS Metadata Server
(3) SAS Metadata Server uses Operating System to validate user, which in turn uses PAM
(4) SAS Enterprise Guide connects to the SAS Object Spawner and requests a SAS Workspace Server, passing credentials (WinUser)
(5) SAS Object Spawner runs a shell script as the user and spawns a continuously running SAS Workspace Server. A Kerberos TGT is generated by PAM
(6) SAS Code executes: LIBNAME hivelib HADOOP; specifying Kerberos security principals for HIVE & HDFS. The Hadoop client libraries use the TGT to request Service Tickets for HIVE & HDFS
(7) SAS connects to HDFS using the HDFS Service Ticket and writes a XML configuration file to temporary directory in HDFS
(8) User authenticated with the HDFS Service Ticket
(9) SAS Code executes: PROC LASR or HP PROC; initiates an SSH connection using Kerberos. SSH uses the TGT to request a Service Tickets for HOST
(10) User connects using the HOST Service Ticket and starts the HPA General, passes HDFS connection details
(11) User authenticated using the HOST Service Ticket
(12) SSH Daemon generates TGT for user on HPA General
(13) HPA General Initiates parallel SSH connections to HPA Captains; SSH requests HOST Service Tickets for HPA Captains
(14) SSH connections to HPA Captains authenticated using the HOST Service Tickets
(15) HPA General initiates connection to Hadoop, using the TGT to request HIVE & HDFS Service Tickets
(16) HPA General connects to Hadoop and retrieves the XML configuration file
(17) SSH Daemon generates a requests a TGT for the connecting user
(18) HPA Root submits a Map Reduce(MR) Job; the MR Job initiates the SAS Embedded Process
(19) SAS EP connects first to the HPA Root, and then connects to assigned HPA Worker using UNIX sockets
(20) SAS EP runs as a MR tasks and connects as necessary to HDFS/HIVE using Hadoop internal tokens
(21) SAS EP processes pass data to HPA worker nodes
(22) SAS HPA processes process data in memory and results returned to client

5 Questions That Must be Addressed during Pre-Installation

You should address the following questions as early as possible in the design of the SAS environment. The answers to these questions will impact directly the time required to implement the SAS environment.

5.1 SAS Software Components

1. Is SAS/ACCESS to Hadoop licensed?
2. Is SAS/ACCESS to Impala licensed?
3. Which SAS In-Memory products and solutions are licensed?
4. Is SAS Grid Manager licensed?

5.2 Users

1. Where are the user details stored for the Hadoop environment?
2. Is there a single repository used for the whole organization or is a separate repository used for the Hadoop environment?
3. What type of user repository is used? Active Directory, LDAP, or an alternative.
4. What implementation of Kerberos is used? A separate deployment of MIT Kerberos or Kerberos as part of an Active Directory domain.
5. If Active Directory is used as the user repository are the required UNIX/POSIX attributes already defined for all users?
6. If a separate repository is used on which hosts are the user accounts valid?

5.3 Hadoop Nodes and SAS Nodes

1. Is the operating system of each node already integrated with the user repository?
2. What mechanism(s) is used to integrate the operating system with the user repository?
3. What mechanism is used to request a TGT for new user sessions?
4. Are the Hadoop Nodes & SAS Nodes in the same Kerberos realm?
5. If different Kerberos realms are used what type(s) of trusts are configured between realms?

6 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.

SAS Institute Inc. 2014. "LIBNAME Statement Specifics for Hadoop." *SAS 9.4 for Relational Databases: Reference, 3rd ed.* Cary, NC: SAS Institute Inc.

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

7 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.

8 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

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**