Centrify Server Suite 2015

Securing Hadoop Clusters with Centrify

July 2015

Centrify Corporation

Legal notice

Contents

Overview of what Centrify provides for Hadoop clusters

This chapter provides an introduction to Hadoop, to the security issues presented by Hadoop clusters, and what the Centrify Server Suite solution provides to help you manage authentication for Hadoop clusters.

What Hadoop provides

Apache Hadoop is an open source framework for the distributed storage and processing of very large data sets commonly required by large organizations or web-based service providers. Hadoop enables you to store and retrieve data from a virtual file system that relies on multiple computers running as a cluster.

By distributing data and processing power, the cluster provides transparent load-balancing, optimized performance, and efficient network communication. However, configuring the Hadoop cluster and managing the services required to keep track of where data is stored and how administrative tasks are distributed often involves manual processes or difficult administrative challenges.

How implementations of Hadoop differ

In addition to the complexity of configuring Hadoop clusters, different implementations of Hadoop from different vendors rely on different services and subsystems to help you manage the cluster. For example, Hortonworks, Cloudera, and MapR are major vendors that provide implementations of Hadoop but use different services to manage operations for the cluster.

Most implementations of Hadoop, for example, include key services that are provided by the following core components:

- YARN is the resource management and distributed application framework for processing data for multiple applications.
- HDFS is the data storage layer for Hadoop.
- MapReduce is the underlying data-processing framework for handling data in parallel across a cluster.

In addition to core services, your Hadoop implementation might include several Hadoop-based applications, such as Hive, HBase, and Spark, for which you might want to manage service accounts and `keytab` files using Centrify. However, the specific applications and accounts you might need to secure will depend on the implementation of Hadoop you choose.

Although there is a common framework and many similar applications in different implementations of Hadoop, the specific implementation details about how the core framework and any additional services are delivered, managed, and secured vary from vendor to vendor. Before attempting to integrate a Hadoop cluster with Active Directory through the Centrify agent, you should be familiar with the architecture and implementation details that are specific to your vendor of choice.

Addressing security in a clustered environment

In a distributed environment, such as a Hadoop cluster, security is a major concern. For example, each implementation of Hadoop relies on a core set of service accounts that must be available on each computer in the cluster and must be able to communicate securely from one computer to another. You want to reduce the risk of these accounts being compromised and to be able to securely create and maintain the same set of credentials on every computer. Centrify enables you to securely create the service accounts in Active Directory, then leverage Active Directory to use Kerberos authentication for communication between nodes in the cluster. Through Centrify and Active Directory, you can enable silent authentication between computers without administrative intervention or exposing the password for the shared service accounts.

What Centrify provides

Centrify Server Suite is an enterprise-class solution that secures even the most complex Hadoop environments by leveraging an organization's existing Active Directory infrastructure to deliver access control, privilege management, and user-level auditing.

Centrify Server Suite secures the industry's broadest range of mission-critical servers from identity-related insider risks and outsider attacks, making security and regulatory compliance repeatable and sustainable. The solution leverages existing Active Directory infrastructure to centrally manage authentication, access controls, privileged identities, policy enforcement and compliance for on-premises and cloud resources.

Key features of the Centrify Server Suite solution

Centrify Server Suite provides identity, access control, and privilege management for Hadoop clusters by delivering the following:

- Full Kerberos integration through Active Directory so that the cluster runs securely.
- Automated service account creation and credential management.
- Single sign-on authentication for Active Directory users.
- Regulatory compliance through least privilege enforcement and auditing.
- Developer SDKs for secure client application access to Hadoop.

How you can take advantage of Centrify in a Hadoop cluster

By installing a Centrify agent on each node in the Hadoop cluster, you can provide identity and access management for the users who will log on to computers in the cluster with their Active Directory credentials. An agent on each node also ensures secure communication and access management for the service accounts that are required to be on multiple computers.

In addition, by installing the agent on the control node in the cluster, you can centrally create, secure, and distribute the service accounts and Kerberos key table (`keytab`) files that your cluster requires for distributed computing. The service accounts are stored securely in Active Directory with the domain controller acting as the Kerberos key distribution center (KDC). The password is managed automatically without human intervention, so there's no need to expose the account credentials or share account information.

How you benefit from using Centrify with Hadoop

In addition to the automated generation of service principals and `keytab` files, Centrify Server Suite provides several benefits for Hadoop and Big Data environments, including:

- Simple and secure access to Hadoop environments.

  Centrify makes it simple to run Hadoop in secure mode by using existing identity management infrastructure—Active Directory—without the drawbacks of introducing alternative solutions that do not scale and are not enterprise ready. Centrify Server Suite also saves money by letting organizations make use of existing skill sets within the enterprise.

- Single sign-on for IT administrators and big data users.

  By extending the power of Active Directory's Kerberos and LDAP capabilities to Hadoop clusters, Centrify Server Suite lets organizations use existing Active Directory-based authentication for Hadoop administrators and end users. New SSO functionality in Big Data environments makes users more productive and secure by allowing them to login as themselves, rather than sharing privileged accounts.

- Secure computer-to-computer communications.

  Centrify Server Suite automates Hadoop service account management within Active Directory. By automating machine-to-machine credential management, Centrify not only secures user identity, but also system and service account identity.

- Reduced identity-related risks and greater regulatory compliance.

  Hadoop environments store most if not all of an organization's most important data. Centrify Server Suite tracks user activity back to an individual in Active Directory, making data more secure. Centrify also reports on who did what across Hadoop clusters, nodes, and services. By enforcing access controls and least-privilege security across

Hadoop, Centrify delivers cost-effective compliance through combined access and activity reporting.

This guide describes how you can use Centrify to configure Hadoop clusters to be managed in an Active Directory environment. The specific steps to follow depend on the implementation of Hadoop you are using.

Preparing for integration with Centrify

Regardless of the implementation of Hadoop you deploy, there are some common steps to prepare for integration between Centrify and an Apache Hadoop cluster. This chapter describes the recommended steps that will enable Kerberos-based security for your Hadoop cluster through Centrify and Active Directory. After you complete the steps in this chapter, you should refer to the appropriate section for the implementation of Hadoop you use for vendor-specific steps.

Preparing to create unique principal names

The default Hadoop security architecture is based on Kerberos, which is also the core infrastructure for Active Directory authentication services. As a result, all principals are user or computer principals—based on the setting of the `hadoop.service.object.type` property—and there will be an Active Directory account for each service account that requires a Kerberos key table (`keytab`) file.

The key to managing Hadoop clusters in Active Directory is the addition of a cluster prefix to the associated Kerberos principal. The cluster prefix ensures that the user principal name (UPN) and service principal name (SPN) for the account each cluster depends upon are unique across the Active Directory forest. Unique user principal and service principal names are required.

To simplify the integration between Active Directory and the nodes in each cluster, you can install the Centrify agent on each node. You can then use Centrify to manage user and server principals and corresponding `keytab` files on those computer nodes or centrally from a Windows console on an administrator's workstation.

You should outline a naming convention for all Hadoop service principals that will reside in Active Directory. Ideally, you should be able to identify the service, cluster, and host by the naming convention you establish. However, you should keep in mind the limitations of the Active Directory `sAMAccountName` attribute. The `sAMAccountName` attribute has a maximum length of 20 characters and must be unique across the Active Directory forest.

Basic prerequisites

There are a few basic requirements you need to satisfy to get started. For example:

- You must have Active Directory installed and at least one domain controller available.
- You should have a Windows workstation joined to the domain where you can run administrative consoles.

- You should have access to at least two physical or virtual Linux computers to use as Hadoop nodes.
- You should have Centrify Server Suite software installed or available to be installed.

  You can request a free trial of Centrify Server Suite by filling out the http://www.centrify.com/free-trial/server-suite-form/ on the Centrify website and specifying `Hadoop` in the Comments field.
- You should have Centrify Server Suite documentation available for reference.

  You can download documentation from http://community.centrify.com/t5/custom/page/page-id/Centrify-Documentation after you register for a free trial and set up your Centrify account at https://www.centrify.com/account/register.asp.

Planning the organizational units to use

You should use an Active Directory organizational unit (OU) to manage all of your Hadoop clusters, such as `OU=Hadoop`. You might have to ask your Active Directory team to create this OU for you. The technical lead or Hadoop administrator should have full control of this Hadoop OU. Your Active Directory domain administrators might need to delegate administrative rights of this OU to you or your technical lead.

Each cluster should have its own OU to independently manage its nodes and service accounts. The OU name should reflect the name of the cluster, for example, `Cluster1`. This cluster-level OU is usually created within an OU that was created by the Active Directory administrator and delegated to you so that you can create an OU for each Hadoop cluster and manage the accounts and policies yourself.

Planning to use Centrify zones for Hadoop clusters

Centrify uses the Zones container to store the access and privilege permissions for the selected Active Directory users you authorize to access each Hadoop cluster. You will set up a unique zone for each Hadoop cluster that you deploy to ensure separation of duties and enable delegated administration. This Linux identity, access information, and privilege information is stored within the OU that was created for you in the steps above. Use the child zone name as the same name for the cluster prefix, for example, `HadoopTest`.

Creating Active Directory organizational units

You are now ready to create the Active Directory organizational unit for the Hadoop cluster.

To create the organizational units

1 Open Active Directory Users and Computers.

2 Select a location in the domain, right-click, then select **New > Organizational Unit**.

3 Type the name of the top-level Hadoop OU, then click **OK**.

For example, you might create the OU in a location similar to this:
`OU=Hadoop,DC=example,DC=com`

Note that you might need an Active Directory administrator to perform this step, then grant you delegated permission to manage this top level OU.

4 Select the Hadoop OU, right-click, then select **New > Organizational Unit**.

5 Create a new OU for each cluster.

For example, you might create the OU in a location similar to this:
`OU=Cluster1,OU=Hadoop,DC=example,DC=com`

If you want to manage nodes in the cluster separate from the service accounts in the cluster, you might want to create additional OUs for computer nodes (`OU=Nodes`) and service accounts (`OU=Services`)

6 Select the cluster-specific OU, right-click, then select **New > Organizational Unit**.

7 Create a separate OU for computer nodes, then repeat Step 6 and Step 7 to create an OU for service accounts.

For example, you might create the following additional OUs:
```
OU=Nodes, OU=Cluster1,OU=Hadoop,DC=example,DC=com
OU=Services, OU=Cluster1,OU=Hadoop,DC=example,DC=com
```

Installing Centrify DirectManage Access

You are now ready to install Centrify Server Suite on a Windows administrator's workstation. If you downloaded the documentation, you can use the *Centrify Server Suite Quick Start Guide* to guide you through the next steps.

To install the software

1 Open the Centrify Server Suite ISO or ZIP file for Windows 32-bit or Windows 64-bit on the Windows workstation.

2 Click **Access** on the Getting Started page or run the setup program in the DirectManage folder.

3 Follow the prompts displayed to select the suite edition and components to install.

Creating zones and defining a user profile

You are now ready to use Access Manager to finish setting up the Active Directory domain and create the zones for the Hadoop cluster.

To create zones and add a user profile to the zone

1. Open Access Manager to start the Setup Wizard.
2. Follow the prompts displayed to create the containers for Licenses and Zones.

   You can accept the default locations or create a Centrify organizational unit for the containers.
3. Create a zone for the cluster:
   a. Click **Create Zone** in Access Manager.
   b. Type a name for the zone, such as `HadoopTest`.
   c. Click **Next**, then click **Finish**.

   This step creates a new zone with the default options.
4. Select the new `HadoopTest` zone, right-click, then select **Add User** to search for and select an existing Active Directory user.
5. Select **Define user UNIX profile** and deselect **Assign roles**, then click **Next**.
6. Accept the defaults for all fields, click **Next**, then click **Finish**.
7. Create a child zone within the cluster zone:
   a. Select the `HadoopTest` zone.
   b. Right-click, then select **Create Child Zone**.
   c. Type a name for the zone, for example, `HadoopChild1` and an optional description.
   d. Click **Next** and **Finish** to create the new child zone.

Assigning roles

User profiles are inherited by child zones, so the users that you add to the `HadoopTest` zone automatically have a profile in the `HadoopChild1` zone. To log on to a computer, however, a user must have both a profile and a role assignment. Access Manager includes a default UNIX Login role that you can assign to enable users to log on.

To assign a role to a user in a zone

1. Expand the `HadoopTest` zone, **Child Zones**, the `HadoopChild1` zone, and **Authorization**.
2. Select **Role Assignments**, right-click, then click **Assign Role**.
3. Select the UNIX Login role from the list of roles and click **OK**.
4. Click **Add AD Account** to search for and select the Active Directory user you added to the `HadoopTest` zone, then click **OK**.

Next steps

You are now ready to configure a demonstration environment that uses two Linux virtual machines with the Centrify agent to illustrate integrating the Hadoop cluster with Centrify software.

For more information, see the following sections:

- Integrating Cloudera into Active Directory
- Integrating Hortonworks into Active Directory
- Integrating MapR into Active Directory

Integrating Cloudera into Active Directory

Centrify Server Suite is an enterprise-class solution that supports the Cloudera implementation of Apache Hadoop. Together, Centrify and Cloudera allow you to use your organization's existing Active Directory infrastructure to deliver access control, privilege management, and user-level auditing.

This chapter describes how to configure multiple computers in a Cloudera cluster to be managed through the Centrify agent in an Active Directory environment.

After you perform the procedures described in this chapter, Server Suite automatically creates and distributes the `hdfs` service principal and the `hdfs keytab` file when you enable Kerberos security through the Cloudera Manager interface.

Preparing for Cloudera integration

The instructions in this chapter assume you have access to Cloudera Manager administrative interface. Cloudera Manager manages all per-host service principals and their Kerberos `keytab` files. For instance, per-host service principals are generated automatically when Kerberos is enabled. The only service principal that needs to be generated is the `hdfs` service principal, which is copied to all cluster nodes.

Create a Cloudera cluster that uses Centrify

If you have performed the common steps described in "Preparing for integration with Centrify" on page 9, you are ready to configure a demonstration environment to illustrate integrating the Cloudera cluster with Centrify software.

The instructions that follow describe how to prepare two Linux virtual machines, install the Centrify agent, and install Cloudera to create a two-node cluster. If you have already installed Cloudera on two or more physical or virtual machines in an isolated environment for testing, you can install the Centrify agent as described in "Install the Centrify agent" on page 15, then continue on to "Enabling security for the cluster" on page 17.

Prepare virtual machines

For the sake of demonstration, perform the following steps to prepare the virtual environment for testing with two Linux computers:

1 Provision two new Linux virtual machines running a version of Linux that supports your Cloudera release.

For example, create new Red Hat Enterprise Linux 6.x 64-bit or CentOS 6.x 64-bit virtual machines using the following settings:

- m2n1.centrifyimage.vms

  IP address 192.168.1.46, with 2 processors, 4GB RAM, 2 HD (20/ 50 GB)

- m2n2.centrifyimage.vms

  IP address 192.168.1.47, with 2 processors, 4GB RAM, 2 HD (20/ 50 GB)

2 Create the corresponding DNS address (A) records in the appropriate DNS zone. In this example, the DNS zone is `centrifyimage.vms`.

3 Create the proper reverse DNS entries.

4 Perform a `yum` update.

Install the Centrify agent

You can now install the Centrify agent on each computer in the cluster and join the nodes to an Active Directory domain.

To install and join the domain

1 Download the appropriate Centrify agent for the operating system of the virtual machines.

  For example, copy the `centrify-suite-2015-rhel3-x86_64.tgz` from the ISO to the computers that make up the cluster.

2 Unzip and extract the agent package.

  For example:

```
gunzip centrify-suite-2015-rhel3-x86_64.tgz
tar -xvf centrify-suite-2015-rhel3-x86_64.tar
```

3 Run the `install.sh` installation script interactively.

  For example:

```
./install.sh
```

  You can automate the installation of the agent by creating a custom configuration file to use with the `install.sh` script after you are familiar with the settings to provide. For the sake of the demonstration, however, you should run the script the script interactively.

4 Follow the prompts displayed to install Standard Edition (S) or Enterprise Edition (E) and join a domain.

  You can press ENTER to accept the default for any prompt. For example:

```
How do you want to proceed? (E|S|X|C|Q) [E]: S
Do you want to run adcheck to verify your AD environment? (Q|Y|N) [Y]: Y
```

In this example, you would enter `centrifyimage.vms` as the *`domain-name`*. You can then accept the defaults for the next two prompts.

```
Please enter the Active Directory domain to check [company.com]: domain-name
Join an Active Directory domain? (Q|Y|N) [Y]:
Enter the Active Directory domain to join [centrifyimage.vms]:
```

Next, you must type the user name and password for an Active Directory user with permissions to update the top-level Hadoop organizational unit as described in "Creating Active Directory organizational units" on page 10.

```
Enter the Active Directory authorized user [administrator]:
Enter the password for the Active Directory user:
Enter the computer name [m2n1.centrifyimage.vms]:
```

If you have created the organizational unit structure as described in "Creating Active Directory organizational units" on page 10, you would specify that path here.

```
Enter the container DN [Computers]: OU=Cluster1, OU=Hadoop, DC=centrifyimage,
DC=vms
Enter the name of the zone: HadoopTest
Enter the name of the Domain Controller [auto detect]:
Reboot the computer after installation? (Q|Y|N) [Y]:
```

You should see confirmation that the computer has successfully joined Active Directory. After the computer restarts, you can log on using the Active Directory user name and password for the user you previously assigned the UNIX Login role.

5 Disable and stop the `iptables` service.

```
chkconfig iptables off && service iptables stop
```

6 Enable the `ntpd` service.

```
chkconfig ntpd on
```

7 Enable the EPEL repository on Red Hat:

```
wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-
8.noarch.rpm
rpm -ivh epel-release-6-8.noarch.rpm
yum install sshpass
```

Install Cloudera on each node in the cluster

You can now install Cloudera on each computer in the cluster. For details about installing Cloudera, see http://www.cloudera.com/content/cloudera/en/documentation/core/latest/PDF/cloudera-installation.pdf.

After you install Cloudera, you have a working environment with the Centrify agent controlling access to Linux hosts and coexisting with the Cloudera cluster. You can now view the dashboard of the cluster from Cloudera Manager.

Enabling security for the cluster

The procedures in this section assume that you will use Cloudera Manager to configure and manage clusters. Cloudera Manager provides an interface for enabling Kerberos security for Cloudera clusters. Centrify adds an automation script that works with Cloudera security to create the `hdfs` service principal and `keytab` files automatically when you enable Cloudera Kerberos security.

The following sections describe how to use Cloudera Manager together with the Centrify automation script to enable Kerberos security for the cluster.

Key tasks for enabling security

Configuring Kerberos authentication for a Cloudera cluster involves both manual and automated tasks. As a preview, you should plan to perform the following tasks:

- Create an input file with comma-separated values that will be used by the Centrify automation script.
- Modify the Centrify `hadoop.conf` configuration file used by the Centrify automation script.
- Run the Centrify automation script on one cluster node.
- Stop all Hadoop and Cloudera management services.
- Create symbolic links for Centrify LDAP CLIs.
- Verify and, if necessary, enable LDAP over SSL.
- Use Cloudera Manager to enable Kerberos security.
- Use Cloudera Manager to generate service principals, accounts, and `keytab` files.
- Restart all Hadoop and Cloudera management service.

Preparing the input file

Centrify provides an automation script—`kerberos_security_setup.pl`—to manage service principal names (SPN) and `keytab` files in the cluster. To use the `kerberos_security_setup.pl` script, you must first create the input file used by the automation script. The input file is a CSV-formatted file containing entries for the `hdfs` Kerberos service principals required to enable Kerberos security for a Hadoop cluster.

The input file consists of the following information on each line:

- Host name of the computer node that the `keytab` file is delivered to.
- Display name to use for the Cloudera service account.
- Service principal name for the Cloudera service account.
- Name of the `keytab` file for the service account.
- Path to the `keytab` file.

- User name of the owner of the `keytab` file.
- Group name of the owner of the `keytab` file.
- Permissions set on the `keytab` file.

You can find a sample of the input file format in `/usr/share/centrifydc/samples/hadoop/host-principal-keytab-list.csv`. You must manually create the input file to include the appropriate information for the `hdfs` service on each of the computers in the cluster.

To create the input file

1. Log on to the computer you are using as the control node for the cluster.

   The control node is typically the computer where you run the Cloudera Manager service (`cloudera-scm-server`), but could be another computer depending on your implementation.

2. Copy the sample input file to create a working copy.

   For example:

   ```
   cd /usr/share/centrifydc/samples/hadoop
   cp host-principal-keytab-list.csv myclouderainput.csv
   ```

3. Open the file in a text editor and replace the sample content with the `hdfs` service information for your cluster. Your file should have one line for each computer in the cluster. For example:

   ```
   m2n1.centrifyimage.vms,HDFS User,hdfs@EXAMPLE.COM,hdfs.keytab,/etc/security/keytabs,hdfs,hadoop,440

   m2n2.centrifyimage.vms,HDFS User,hdfs@EXAMPLE.COM,hdfs.keytab,/etc/security/keytabs,hdfs,hadoop,440
   ```

4. Save the text file.

Modifying the hadoop.conf file

Before running the `kerberos_security_setup.pl` automation script, you should modify the default `hadoop.conf` configuration file on the computer you are using as the control node for the cluster. The `hadoop.conf` configuration file controls how objects are created and named in Active Directory and how commands are executed on remote hosts when the automation script copies files and sets permissions.

At a minimum, you should uncomment and set the following properties in the `hadoop.conf` configuration file:

- hadoop.service.container

  Use this property to specify the appropriate Active Directory location you created for Hadoop objects. For example:

  ```
  OU=Cluster1,OU=Hadoop
  ```

- hadoop.cluster.shortname

  Use this property to specify a short name for the cluster to be used as a prefix to differentiate the same service running on computers in different clusters. For example:

  `cdh1`

By default, the automation script uses the `hadoop.conf` file located in `/usr/share/centrifydc/samples/hadoop`. If you move `hadoop.conf` to a different location, you must specify that location with the `--config` option when you execute the automation script as described in Running the automation script.

For more information about these properties, the default naming conventions, and other configuration properties you can set, see the comments in the `hadoop.conf` file.

Running the automation script

After you generate the input file and modify the `hadoop.conf` configuration file, you are ready to run the `kerberos_security_setup.pl` automation script that is located in `/usr/share/centrifydc/samples/hadoop`. You can run the automation script from that location, or move it to a directory of your choice. The `kerberos_security_setup.pl` script performs the following steps for you:

- Verify that the Centrify agent is installed and joined to Active Directory on all cluster nodes.
- Delete the default HTTP SPN from computer object in Active Directory, if necessary.
- Create the Kerberos `keytab` directory on all cluster nodes.
- Create the local directory to generate Kerberos `keytab` files.
- Create the service account principals in Active Directory.
- Create the `keytab` files in the local directory.
- Distribute the `keytab` files to the computers and locations specified in the input file.
- Set the correct ownership and permission for all Kerberos keytab files distributed to the computers in the cluster.
- Modify agent configuration file, if necessary.

To run the automation script

1 Ensure that these prerequisites are met:

- You have `root` access to the computer where you will run the automation script.
- You have secure shell (SSH) and secure copy (SCP) access to all cluster nodes without having to provide a password. One way to achieve this is to use public key authentication.
- You have the necessary permissions to generate Kerberos tickets with the `kinit` command.

2 Log on to the computer you are using as the control node for the cluster.

3 Run `kinit` to get a Kerberos ticket for the administrative account you are using to run the script.

4 Execute the `kerberos_security_setup.pl` automation script using the input file that you prepared to create service accounts and `keytab` files:
```
perl kerberos_security_setup.pl --input myinput.csv -c hadoop.conf --create
```

5 Execute the `kerberos_security_setup.pl` automation script with the `--deploy` option to distribute the `keytab` files to the appropriate nodes in the cluster, and set the file ownership and permission:
```
perl kerberos_security_setup.pl --input myinput.csv --deploy
```

6 Verify the service principal accounts have been created by opening Active Directory Users and Computers and checking the organizational unit you created for the Cloudera cluster.

Stopping Hadoop and Cloudera services

Log on to the computer where Cloudera Manager is running. In Cloudera Manager, stop all Hadoop services and Cloudera management services.

For details about stopping Hadoop services in Cloudera Manager, see http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cdh_ig_services_stop.html.

For details about stopping Cloudera services in Cloudera Manager, see http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_mc_start_stop_cluster.html.

Creating symbolic links for Centrify LDAP CLIs

Cloudera Manager uses the `ldapmodify` and `ldapsearch` CLIs to execute the **Import Kerberos Account Manager Credentials** and **Generate Credentials** operations from the **Administration > Kerberos** page. You must create symbolic links to the Centrify versions of the `ldapmodify` and `ldapsearch` CLIs so that the Centrify versions are used.

Perform the following steps create symbolic links to Centrify LDAP CLIs:

1 Log on to the computer where Cloudera Manager is running.

2 Create the following symbolic links:
```
/usr/bin/ldapmodify -> /usr/share/centrifydc/bin/ldapmodify
/usr/bin/ldapsearch -> /usr/share/centrifydc/bin/ldapsearch
```

Verifying and enabling LDAP over SSL

Cloudera Manager requires LDAP over SSL (also known as *LDAPS* and *LDAP over TLS*) to execute the Kerberos operations described in Creating symbolic links for Centrify LDAP CLIs.

The procedure described in this section enables auto-enrollment for certificates so that node computers automatically use LDAP over SSL when they are joined to the domain.

Note For more details and background information about this procedure, see the "Enabling encrypted communication" section in the "Using Centrify OpenLDAP proxy service" chapter of the *Centrify Server Suite Administrator's Guide for Linux and UNIX*.

On the domain controller:

1. Enable the Active Directory certificate service.
2. If the root certificate is not in an Enterprise PKI managed AD container, import the root certificate into the **Trusted Root Certification Authorities** group policy.
3. Enable auto-enrollment as described in the *Centrify Server Suite Administrator's Guide for Linux and UNIX*. When you enable auto-enrollment, make sure that the certificate template has KDC authentication as one of its application policies.
4. Verify whether LDAP over SSL is enabled on Active Directory. For details about this step, see http://social.technet.microsoft.com/wiki/contents/articles/2980.ldap-over-ssl-ldaps-certificate.aspx#Verify.

On the computer where Cloudera Manager is running:

1. Convert the auto-enrolled certificate to PEM format. You can optionally automate this step by using the **Specify commands to run** group policy as described in the *Centrify Server Suite Group Policy Guide*.
2. Modify the `/etc/centrifydc/openldap/ldap.conf` configuration file so that it contains the following line:
   `TLS_CACERT /var/centrify/net/certs/auto_enrolled_cert.pem`

   For example:
   `TLS_CACERT /var/centrify/net/certs/auto_LDAPS_CA.pem`
3. Verify whether LDAP over SSL is enabled by executing the `ldapmodify` or `ldapsearch` CLI on the cluster node where Cloudera Manager is running.

Enabling Kerberos security in Cloudera Manager

Use Cloudera Manager to enable Kerberos security for the cluster. For details about using Cloudera Manager to enable Kerberos security, see http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_sg_intro_kerb.html.

To enable Kerberos security

1. In Cloudera Manager cluster options, select **Enable Kerberos**.
2. Select all of the check boxes on the first screen of the wizard and click **Continue**.
3. On the KDC Information page, select **Active Directory for the KDC Type** and enter information about the KDC server, Kerberos configuration, and Active Directory configuration. For example:

   KDC Type: `Active Directory`

   Active Directory Suffix: `ou=Cluster1,ou=hadoop,DC=centrifyimage,DC=vms`

   Kerberos Security Realm: `CENTRIFYIMAGE.VMS`

   Active Directory Account Prefix: *`prefixname-`*
4. On the KRB5 Configuration screen, ensure that **Manage krb5.conf through Cloudera Manager** is not selected and click **Continue**.

   Centrify, and not Cloudera Manager, will manage krb5.conf.
5. After the wizard finishes, click the **Generate Credentials** button on the Cloudera Manager **Administration > Kerberos Credentials** page.

Restarting Hadoop and Cloudera services

Log on to the computer where Cloudera Manager is running. In Cloudera Manager, start all Hadoop services and Cloudera management services.

See the documents referred to in "Stopping Hadoop and Cloudera services" on page 20 for details about using Cloudera Manager to start services.

Validating Cloudera cluster security

Now that the Cloudera cluster is using Centrify for Active Directory based authentication, a user can log in using Active Directory credentials directly at the console prompt, or could use a Kerberized SSH client such as the Centrify version of PuTTY to get single sign-on access to the cluster.

Once logged in, the user has Kerberos credentials and will be able to run Hadoop jobs such as the example used below that computes the value of Pi. Because the cluster is now running in secure mode, users without Kerberos will not be able to successfully submit a job to the cluster.

If a user does not have Kerberos credentials and tries to run a Hadoop job, the attempt fails:

```
[sunny@cdh1-cent64-1 native]$ hadoop jar /opt/cloudera/parcels/CDH/lib/
hadoop-0.20-mapreduce/hadoop-examples.jar pi 10 1000
Number of Maps  = 10
```

```
Samples per Map = 1000
15/02/12 22:33:10 WARN ipc.Client: Exception encountered while connecting to
the server : javax.security.sasl.SaslException: GSS initiate failed [Caused by
GSSException: No valid credentials provided (Mechanism level: Failed to find
any Kerberos tgt)]
...
```

If a user has Kerberos credentials and tries to run a Hadoop job, the attempt succeeds:

```
[gpu1@cdh1-cent64-1 ~]$ hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-0.20-
mapreduce/hadoop-examples.jar pi 10 1000
Number of Maps  = 10
Samples per Map = 1000
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
Wrote input for Map #5
Wrote input for Map #6
Wrote input for Map #7
Wrote input for Map #8
Wrote input for Map #9
Starting Job
15/02/12 18:06:51 INFO client.RMProxy: Connecting to ResourceManager at cdh1-
cent64-2.voyager.test/192.168.233.172:8032
...
15/02/12 18:06:54 INFO impl.YarnClientImpl: Submitted application
application_1423735119104_0001
15/02/12 18:06:54 INFO mapreduce.Job: The url to track the job: http://cdh1-
cent64-2.voyager.test:8088/proxy/application_1423735119104_0001/
15/02/12 18:06:54 INFO mapreduce.Job: Running job: job_1423735119104_0001
15/02/12 18:07:13 INFO mapreduce.Job: Job job_1423735119104_0001 running in
uber mode : false
15/02/12 18:07:13 INFO mapreduce.Job:  map 0% reduce 0%
15/02/12 18:07:25 INFO mapreduce.Job:  map 10% reduce 0%
15/02/12 18:07:32 INFO mapreduce.Job:  map 20% reduce 0%
15/02/12 18:07:38 INFO mapreduce.Job:  map 30% reduce 0%
15/02/12 18:07:44 INFO mapreduce.Job:  map 40% reduce 0%
15/02/12 18:07:50 INFO mapreduce.Job:  map 50% reduce 0%
15/02/12 18:07:56 INFO mapreduce.Job:  map 60% reduce 0%
15/02/12 18:08:02 INFO mapreduce.Job:  map 70% reduce 0%
15/02/12 18:08:08 INFO mapreduce.Job:  map 80% reduce 0%
15/02/12 18:08:14 INFO mapreduce.Job:  map 90% reduce 0%
15/02/12 18:08:20 INFO mapreduce.Job:  map 100% reduce 0%
15/02/12 18:08:31 INFO mapreduce.Job:  map 100% reduce 100%
15/02/12 18:08:31 INFO mapreduce.Job: Job job_1423735119104_0001 completed
successfully
...
Job Finished in 100.085 seconds
Estimated value of Pi is 3.14080000000000000000
```

For additional information about validating Cloudera security, see http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_sg_s8_verify_kerb.html.

Troubleshooting

This section describes issues you might encounter when securing a Cloudera cluster using Centrify.

Automation script fails and reports adkeytab permission issues

If the automation script fails and reports issues with `adkeytab` permissions, the likely cause is that `adkeytab` is attempting to load the root-owned Kerberos credential cache (`/tmp/krb5cc_cm_agent`) instead of the default credential cache (`/tmp/krb5cc_0`).

This problem only occurs when the TGT in the Kerberos credential cache `/tmp/krb5cc_cm_agent` is valid and not expired. Other Centrify CLIs such as `adjoin` and `adleave` also exhibit this behavior.

To work around this issue, either set the environment variable `KRB5CCNAME` to the default credential cache (`/tmp/krb5cc_0`) or remove `/tmp/krb5cc_cm_agent` as follows:

1. Stop the `cloudera-scm-agent` service. In most situations, this step is optional. See the note following this procedure for more information.
2. Remove `/tmp/krb5cc_cm_agent`.
3. Run the automation script (`kerberos_security_setup.pl`) again.
4. If you stopped the `cloudera-scm-agent` service in Step 1, restart it now.

Note In most situations, the `/tmp/krb5cc_cm_agent` cache is not required by the `cloudera-scm-agent` service after the service has been started, and the Kerberos credential cache will be expired without renewal. Also, in most situations you do not need to stop the `cloudera-scm-agent` service before you remove `/tmp/krb5cc_cm_agent`. However, it is recommended that you confirm this with Cloudera before removing the `/tmp/krb5cc_cm_agent` cache without first stopping the `cloudera-scm-agent` service.

Disabling security on a cluster

If you run into issues, you might want to remove the service accounts, `keytab` files, and directories that were created by the Centrify automation script. You can re-run the automation script with the `--undeploy` and `--delete` options to clean up unused items.

To disable security on a cluster:

1. Execute `kerberos_security_setup.pl` with the `--undeploy` option to remove distributed `keytab` files from cluster nodes.
   ```
   perl kerberos_security_setup.pl --input myinput.csv --undeploy
   ```
2. Execute `kerberos_security_setup.pl` with the `--delete` option to delete the service accounts and their `keytab` files.
   ```
   perl kerberos_security_setup.pl --input myinput.csv --delete
   ```

Integrating Hortonworks into Active Directory

Centrify Server Suite is an enterprise-class solution that supports the Hortonworks implementation of Apache Hadoop. Together, Centrify and Hortonworks allow you to use your organization's existing Active Directory infrastructure to deliver access control, privilege management, and user-level auditing.

This chapter describes how to configure multiple computers in a Hortonworks cluster to be managed through the Centrify agent in an Active Directory environment. After you perform the procedures described in this chapter, Server Suite automates the creation of Hadoop service principals and `keytab` files when you enable Kerberos security through the Apache Ambari interface.

Preparing for Hortonworks integration

The key to managing Hortonworks clusters in Active Directory is the addition of a cluster prefix to the associated Hortonworks Kerberos principal for per-host principals. The cluster prefix ensures that the user principal name (UPN) and service principal name (SPN) for the account each cluster depends upon are unique across the Active Directory forest. Unique user principal and service principal names are required.

The instructions in this chapter assume you have access to Apache Ambari from each physical or virtual Linux computer on which you install Hortonworks.

Create a Hortonworks cluster that uses Centrify

If you have performed the common steps described in "Preparing for integration with Centrify" on page 9, you are ready to configure a demonstration environment to illustrate integrating the Hortonworks cluster with Centrify software.

The instructions that follow describe how to prepare two Linux virtual machines, install the Centrify agent, and install Hortonworks to create a two-node cluster. If you have already installed Hortonworks on two or more physical or virtual machines in an isolated environment for testing, you can install the Centrify agent as described in "Install the Centrify agent" on page 26, then continue on to "Enabling security for the cluster" on page 28.

Prepare virtual machines

For the sake of demonstration, perform the following steps to prepare the virtual environment:

1. Provision two new Linux virtual machines running a version of Linux that supports your Hortonworks release.

   For example, create new Red Hat Enterprise Linux 6.x 64-bit or CentOS 6.x 64-bit virtual machines using the following settings:

   - m2n1.centrifyimage.vms

     IP address 192.168.1.46, with 2 processors, 4GB RAM, 2 HD (20/ 50 GB)

   - m2n2.centrifyimage.vms

     IP address 192.168.1.47, with 2 processors, 4GB RAM, 2 HD (20/ 50 GB)

2. Create the corresponding DNS address (A) records in the appropriate DNS zone. In this example, the DNS zone is `centrifyimage.vms`.
3. Create the proper reverse DNS entries.
4. Perform a `yum` update.

Install the Centrify agent

You can now install the Centrify agent on each computer in the cluster and join the nodes to an Active Directory domain.

To install and join the domain

1. Download the appropriate Centrify agent for the operating system of the virtual machines.

   For example, copy the `centrify-suite-2015-rhel3-x86_64.tgz` from the ISO to the computers that make up the cluster.

2. Unzip and extract the agent package.

   For example:

   ```
   gunzip centrify-suite-2015-rhel3-x86_64.tgz
   tar -xvf centrify-suite-2015-rhel3-x86_64.tar
   ```

3. Run the `install.sh` installation script interactively.

   For example:

   ```
   ./install.sh
   ```

   You can automate the installation of the agent by creating a custom configuration file to use with the `install.sh` script after you are familiar with the settings to provide. For the sake of the demonstration, however, you should run the script the script interactively.

4 Follow the prompts displayed to install Standard Edition (S) or Enterprise Edition (E) and join a domain.

You can press ENTER to accept the default for any prompt. For example:

```
How do you want to proceed? (E|S|X|C|Q) [E]: S
Do you want to run adcheck to verify your AD environment? (Q|Y|N) [Y]:
```

In this example, you would enter `centrifyimage.vms` as the *`domain-name`*. You can then accept the defaults for the next two prompts.

```
Please enter the Active Directory domain to check [company.com]: domain-name
Join an Active Directory domain? (Q|Y|N) [Y]:
Enter the Active Directory domain to join [centrifyimage.vms]:
```

Next, you must type the user name and password for an Active Directory user with permissions to update the top-level Hadoop organizational unit as described in "Creating Active Directory organizational units" on page 10.

```
Enter the Active Directory authorized user [administrator]:
Enter the password for the Active Directory user:
Enter the computer name [m2n1.centrifyimage.vms]:
```

If you have created the organizational unit structure as described in "Creating Active Directory organizational units" on page 10, you would specify that path here.

```
Enter the container DN [Computers]: OU=Cluster1, OU=Hadoop, DC=centrifyimage,
DC=vms
Enter the name of the zone: HadoopTest
Enter the name of the Domain Controller [auto detect]: <ENTER>
Reboot the computer after installation? (Q|Y|N) [Y]: <ENTER>
```

You should see confirmation that the computer has successfully joined Active Directory. After the computer restarts, you can log on using the Active Directory user name and password for the user you previously assigned the UNIX Login role.

5 Disable and stop the `iptables` service.

```
chkconfig iptables off && service iptables stop
```

6 Enable the `ntpd` service.

```
chkconfig ntpd on
```

7 Enable the EPEL repository on Red Hat:

```
wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-
8.noarch.rpm
rpm -ivh epel-release-6-8.noarch.rpm
yum install sshpass
```

Install Hortonworks on each node in the cluster

You can now install Hortonworks on each computer in the cluster. For details about installing Hortonworks, see http://hortonworks.com/hdp/docs.

After you install Hortonworks, you have a working environment with the Centrify agent controlling access to Linux hosts and coexisting with the Hortonworks cluster. You can now view the dashboard of the cluster from Apache Ambari.

Enabling security for the cluster

In a default Hortonworks installation, Apache Ambari provides an administrative interface for enabling Kerberos security for Hortonworks clusters. Centrify adds an automation script that works with Hortonworks security to create service accounts and `keytab` files automatically when you enable Hortonworks Kerberos security.

The following sections describe how to use the Hortonworks Security UI together with the Centrify automation script to enable Kerberos security for the cluster.

Key tasks for enabling security

Configuring Kerberos authentication for a Hortonworks cluster involves both manual and automated tasks. As a preview, you should plan to perform the following tasks:

- Use the security page in Ambari to create an input file with comma-separated values. The input file is used by the Centrify automation script.
- Modify the Centrify `hadoop.conf` configuration file used by the automation script.
- Run the automation script on one cluster node.
- Complete the security setup and start Hadoop services from the Hortonworks security page in Ambari.

Generating the input file

Centrify provides an automation script—`kerberos_security_setup.pl`—to manage service principal names (SPN) and `keytab` files in the cluster. To use the `kerberos_security_setup.pl` script, you must first use the Hortonworks security page in Ambari to generate the input file for the script to use. The input file is a CSV-formatted file containing all Kerberos principals and keytabs required to enable Kerberos security for a Hadoop cluster.

To generate the input file:

1. Log on to the computer on which the Ambari server is running.

   To determine whether the Ambari server is installed and running on a computer, issue the following command:
   ```
   sudo service ambari-server status
   ```

   If the Ambari server is running, the message "`Ambari Server running`" displays.

2. In Ambari, open the **Admin** top-level tab, select the **Security** page, and click **Enable Security**.

3. Follow the steps in the Security dialogs until you have the option of saving the security configuration as a comma-separated (CSV) file. Save the CSV file to a location of your choice.

Modifying the hadoop.conf file

Before running the `kerberos_security_setup.pl` automation script, you should modify the default `hadoop.conf` configuration file on the computer you are using as the control node for the cluster. The control node is typically the computer where you run the Ambari service (`ambari-server`), but could be another computer depending on your implementation.

The `hadoop.conf` configuration file controls how objects are created and named in Active Directory and how commands are executed on remote hosts when the automation script copies files and sets permissions.

At a minimum, you should uncomment and set the following properties in the `hadoop.conf` configuration file:

- hadoop.service.container

  Use this property to specify the appropriate Active Directory location you created for Hadoop objects. For example:

  ```
  OU=Cluster1,OU=Hadoop
  ```

- hadoop.cluster.shortname

  Use this property to specify a short name for the cluster to be used as a prefix to differentiate the same service running on computers in different clusters.

By default, the automation script uses the `hadoop.conf` file located in `/usr/share/centrifydc/samples/hadoop`. If you move `hadoop.conf` to a different location, you must specify that location with the `--config` option when you execute the automation script as described in "Running the automation script" on page 29.

For more information about these properties, the default naming conventions, and other configuration properties you can set, see the comments in the `hadoop.conf` file.

Running the automation script

After you generate the input file and modify the `hadoop.conf` configuration file, you are ready to run the `kerberos_security_setup.pl` automation script that is located in `/usr/share/centrifydc/samples/hadoop`. You can run the automation script from that location, or move it to a directory of your choice. The `kerberos_security_setup.pl` script performs the following steps for you:

- Verify that the Centrify agent is installed and joined to Active Directory on all cluster nodes.
- Delete the default HTTP SPN from computer object in Active Directory, if necessary.
- Create the Kerberos `keytab` directory on all cluster nodes.
- Create the local directory to generate Kerberos `keytab` files.
- Create the service account principals in Active Directory.

- Create the `keytab` files in the local directory.
- Distribute the `keytab` files to the computers and locations specified in the input file.
- Set the correct ownership and permission for all `keytab` files distributed to the computers in the cluster.
- Modify agent configuration file, if necessary.

To run the automation script

1. Ensure that these prerequisites are met:
   - You have `root` access to the computer where you will run the automation script.
   - You have secure shell (SSH) and secure copy (SCP) access to all cluster nodes without having to provide a password. One way to achieve this is to use public key authentication.
   - You have the necessary permissions to generate Kerberos tickets with the `kinit` command.
2. Log on to the computer you are using as the name node for the cluster.
3. Run `kinit` to get a Kerberos ticket for the administrative account you are using to run the script.
4. Execute the `kerberos_security_setup.pl` automation script using the input file that you prepared to create service accounts and `keytab` files:
   ```
   perl kerberos_security_setup.pl --input myinput.csv -c hadoop.conf --create
   ```
5. Execute the `kerberos_security_setup.pl` automation script with the `--deploy` option to distribute the `keytab` files to the appropriate nodes in the cluster, and set the file ownership and permission:
   ```
   perl kerberos_security_setup.pl --input myinput.csv --deploy
   ```
6. Verify the service principals accounts have been created by opening Active Directory Users and Computers and checking the organizational unit you created for the Hortonworks cluster.

Default settings for service accounts and keytab files

Centrify automatically maintains the `keytab` entries that are part of the computer account when `adclient` joins a domain. By default, `adclient` randomly generates a new password for its computer account every 28 days.

Other service principals and `keytab` files—such as those created for Hadoop services—are not automatically refreshed. By default, however, the automation script creates the accounts for Hadoop services as user accounts with passwords that never expire. These default settings eliminate long term maintenance for the service accounts, but might raise regulatory compliance issues in some organizations. You can modify the `hadoop.conf` file to create the accounts as computer accounts with passwords that never expire. In most cases,

you should not modify the automation script to remove the "`--password-never-expire`" setting.

If using a computer account with the "`--password-never-expire`" option does not resolve your compliance issues, you could write a script that periodically run the `adkeytab` command with the `-C` option to change the password and update the `keytab` file for a specified account. For more information about the `adkeytab` program and command line options, see the `adkeytab` man page.

Completing security setup in Ambari

After the automation script executes, complete the remaining steps in the Ambari Security dialogs until you have finished enabling Kerberos for Hortonworks clusters. These steps include starting Hadoop services.

Validating Hortonworks cluster security

Now that the Hortonworks cluster is using Centrify for Active Directory based authentication, a user can log in using Active Directory credentials directly at the console prompt, or could use a Kerberized SSH client such as the Centrify version of PuTTY to get single sign-on access to the cluster.

Once logged in, the user has Kerberos credentials and will be able to run Hadoop jobs such as the example used below that computes the value of Pi. Because the cluster is now running in secure mode, users without Kerberos will not be able to successfully submit a job to the cluster.

If a user does not have Kerberos credentials and tries to run a Hadoop job, the attempt fails:

```
[sunny@leo-rhes64-h1 hadoop]$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-
mapreduce-examples.jar pi 10 1000
Number of Maps  = 10
Samples per Map = 1000
15/02/11 00:00:07 WARN ipc.Client: Exception encountered while connecting to
the server : javax.security.sasl.SaslException: GSS initiate failed [Caused by
GSSException: No valid credentials provided (Mechanism level: Failed to find
any Kerberos tgt)]
...
```

If a user has Kerberos credentials and tries to run a Hadoop job, the attempt succeeds:

```
[gpu1@leo-rhes64-h1 ~]$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-
mapreduce-examples.jar pi 10 1000
Number of Maps  = 10
Samples per Map = 1000
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
```

```
Wrote input for Map #5
Wrote input for Map #6
Wrote input for Map #7
Wrote input for Map #8
Wrote input for Map #9
Starting Job
15/02/10 23:58:21 INFO client.RMProxy: Connecting to ResourceManager at leo-
rhes64-h2.voyager.test/192.168.233.155:8050
...
15/02/10 23:58:25 INFO impl.YarnClientImpl: Submitted application
application_1421805403535_0007
15/02/10 23:58:25 INFO mapreduce.Job: The url to track the job: http://leo-
rhes64-h2.voyager.test:8088/proxy/application_1421805403535_0007/
15/02/10 23:58:25 INFO mapreduce.Job: Running job: job_1421805403535_0007
15/02/10 23:58:49 INFO mapreduce.Job: Job job_1421805403535_0007 running in
uber mode : false
15/02/10 23:58:49 INFO mapreduce.Job:  map 0% reduce 0%
15/02/10 23:59:08 INFO mapreduce.Job:  map 20% reduce 0%
15/02/10 23:59:17 INFO mapreduce.Job:  map 40% reduce 0%
15/02/10 23:59:25 INFO mapreduce.Job:  map 60% reduce 0%
15/02/10 23:59:32 INFO mapreduce.Job:  map 80% reduce 0%
15/02/10 23:59:38 INFO mapreduce.Job:  map 90% reduce 0%
15/02/10 23:59:44 INFO mapreduce.Job:  map 90% reduce 30%
15/02/10 23:59:45 INFO mapreduce.Job:  map 100% reduce 30%
15/02/10 23:59:47 INFO mapreduce.Job:  map 100% reduce 100%
15/02/10 23:59:47 INFO mapreduce.Job: Job job_1421805403535_0007 completed
successfully
...
Job Finished in 86.971 seconds
Estimated value of Pi is 3.14080000000000000000
```

Disabling security for the cluster

If you disable security through Ambari and no longer need the principals, files, and directories that were created by the Centrify automation script when you enabled security, you can re-run the automation script with the `--undeploy` and `--delete` options to clean up unused items.

To disable security on a cluster:

1. Execute `kerberos_security_setup.pl` with the `--undeploy` option to remove distributed `keytab` files from cluster nodes.
   ```
   perl kerberos_security_setup.pl --input myinput.csv --undeploy
   ```
2. Execute `kerberos_security_setup.pl` with the `--delete` option to delete the service accounts and their `keytab` files.
   ```
   perl kerberos_security_setup.pl --input myinput.csv --delete
   ```

Integrating MapR into Active Directory

Centrify Server Suite is an enterprise-class solution that supports the MapR Distribution implementation of Apache Hadoop. Together, Centrify and MapR allow you to use your organization's existing Active Directory infrastructure to deliver access control, privilege management, and user-level auditing.

This chapter describes how to configure multiple computers in a MapR cluster to be managed through the Centrify agent in an Active Directory environment.

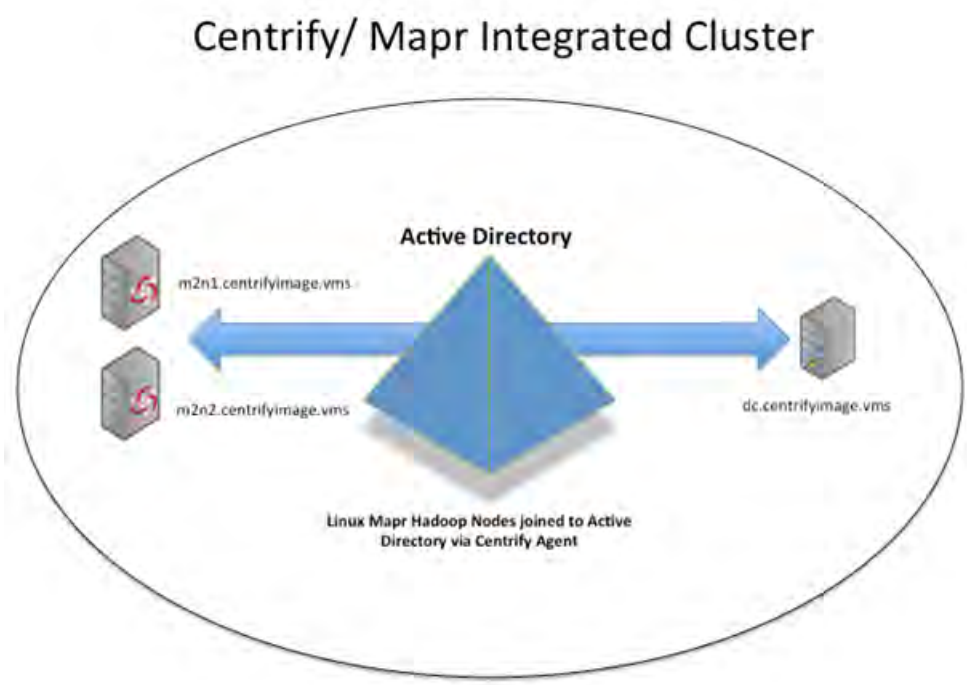
Preparing for MapR integration

The MapR implementation of Hadoop has its own proprietary security architecture that is similar to Kerberos. Because of this proprietary architecture, not all MapR services support Kerberos-based authentication. For example, most of the internal communication between core MapR services relies on MapR security and MapR tickets. Other services such as the web service and CLDB can be configured to use Kerberos instead of MapR security.

Therefore, the first step in preparing for integration is to identify the components in the MapR cluster that support Kerberos authentication, then deciding for each of those components whether you want to enable Kerberos security. This information will determine the number of user or computer principals and `keytab` files you need to create and distribute. In general, MapR clusters only support Kerberos authentication for communication between open source components. In most cases, each service that uses Kerberos requires at least one per-host `keytab` file.

To simplify the integration between Active Directory and the nodes in each cluster, you can install the Centrify agent on each node. You can then use Centrify to manage user and server principal and corresponding `keytab` files on those computer nodes or centrally from a Windows console on an administrator's workstation.

The following illustration provides a simplified view of the integration.



Create a MapR cluster that uses Centrify

If you have performed the common steps described in "Preparing for integration with Centrify" on page 9, you are ready to configure a demonstration environment to illustrate integrating the MapR cluster with Centrify software.

The instructions that follow describe how to prepare two Linux virtual machines, install the Centrify agent, and install MapR to create a two-node cluster. If you have already installed MapR on two or more physical or virtual machines in an isolated environment for testing, you can install the Centrify agent as described in "Install the Centrify agent" on page 35, then continue on to "Enabling security for the cluster" on page 41.

Prepare virtual machines

For the sake of demonstration, perform the following steps to prepare the virtual environment for testing with two Linux computers:

1 Provision two new Linux virtual machines running a version of Linux that supports your MapR release.

For example, create new Red Hat Enterprise Linux 6.x 64-bit or CentOS 6.x 64-bit virtual machines using the following settings:

- m2n1.centrifyimage.vms

  IP address 192.168.1.46, with 2 processors, 4GB RAM, 2 HD (20/ 50 GB)

- m2n2.centrifyimage.vms

  IP address 192.168.1.47, with 2 processors, 4GB RAM, 2 HD (20/ 50 GB)

2 Create the corresponding DNS address (A) records in the appropriate DNS zone. In this example, the DNS zone is `centrifyimage.vms`.

3 Create the proper reverse DNS entries.

4 Perform a `yum` update.

Install the Centrify agent

You can now install the Centrify agent on each computer in the cluster and join the nodes to an Active Directory domain.

To install and join the domain

1 Download the appropriate Centrify agent for the operating system of the virtual machines.

  For example, copy the `centrify-suite-2015-rhel3-x86_64.tgz` from the ISO to the computers that make up the cluster.

2 Unzip and extract the agent package.

  For example:

```
gunzip centrify-suite-2015-rhel3-x86_64.tgz
tar -xvf centrify-suite-2015-rhel3-x86_64.tar
```

3 Run the `install.sh` installation script interactively.

  For example:

```
./install.sh
```

  By default, installing the Centrify agent also installs the Centrify OpenSSH package. The Centrify OpenSSH package has a conflict with the MapR PAM library that prevents tickets from being generated automatically. You can choose not to install the OpenSSH package by using `-C` option on the command line and selecting the individual packages to install, or you can install the default package, then modify the `centrify-sshd` startup script to include the library required by the MapR PAM module.

  You can automate the installation of the agent by creating a custom configuration file to use with the install.sh script after you are familiar with the settings to provide. For the sake of the demonstration, however, you should run the script the script interactively.

4 Follow the prompts displayed to install Standard Edition (S) or Enterprise Edition (E) and join a domain.

You can press ENTER to accept the default for any prompt. For example:

```
How do you want to proceed? (E|S|X|C|Q) [E]: S
Do you want to run adcheck to verify your AD environment? (Q|Y|N) [Y]:
Please enter the Active Directory domain to check [company.com]: domain-name
```

In this example, you would enter `centrifyimage.vms` as the *domain-name*. You can then accept the defaults for the next two prompts.

```
Join an Active Directory domain? (Q|Y|N) [Y]:
Enter the Active Directory domain to join [centrifyimage.vms]:
```

Next, you must type the user name and password for an Active Directory user with permissions to update the top-level Hadoop organizational unit as described in "Creating Active Directory organizational units" on page 10.

```
Enter the Active Directory authorized user [administrator]:
Enter the password for the Active Directory user:
Enter the computer name [m2n1.centrifyimage.vms]:
```

If you have created the organizational unit structure as described in "Creating Active Directory organizational units" on page 10, you would specify that path here.

```
Enter the container DN [Computers]: OU=Cluster1, OU=Hadoop, DC=centrifyimage,
DC=vms
Enter the name of the zone: HadoopTest
Enter the name of the Domain Controller [auto detect]:
Reboot the computer after installation? (Q|Y|N) [Y]:
```

You should see confirmation that the computer has successfully joined Active Directory. After the computer restarts, you can log on using the Active Directory user name and password for the user you previously assigned the UNIX Login role.

5 Disable and stop the `iptables` service.

```
chkconfig iptables off && service iptables stop
```

6 Enable the `ntpd` service.

```
chkconfig ntpd on
```

7 Enable the EPEL repository on CentOS or Red Hat 6.x:

```
wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-
8.noarch.rpm
rpm -ivh epel-release-6-8.noarch.rpm
yum install sshpass
```

8 Repeat Step 2 through Step 7 on the other virtual machine in the cluster.

Install MapR on each node in the cluster

You can now install MapR on each computer in the cluster. You can find additional information about installing MapR in the *MapR Quick Installation Guide* http://doc.mapr.com/display/MapR/Quick+Installation+Guide.

To install MapR on a computer

1. Download the MapR setup package.
   ```
   http://package.mapr.com/releases/v4.0.1/redhat/mapr-setup
   ```
2. Reset permissions on the setup file to make it executable.
   ```
   [root@m2n1 Downloads]# chmod 700 mapr-setup
   ```
3. Run the setup program.
   ```
   [root@m2n1 Downloads]# ./mapr-setup
   ===============================================
   Self Extracting Installer for MapR Installation
   ===============================================

   Extracting installer.......
   Copying setup files to "/opt/mapr-installer"......
   Installed to "/opt/mapr-installer"
   ==================================

   Run "/opt/mapr-installer/bin/install" as super user, to begin install process
   ```
4. Run the MapR installation program.
   ```
   /opt/mapr-installer/bin/install
   ```
5. Follow the prompts displayed and select the default values on each node, except change cluster name to Mapr2.

   In most cases, you can install MapR by creating a custom configuration file to skip the interview process. For the sake of the demonstration, however, you should respond to the prompts interactively.
   ```
   Do you have a config file (y/n) [n]:
   Enter the hostnames of all the control nodes separated by spaces or commas []:
   m2n1.centrifyimage.vms
   Enter the hostnames of all the data nodes separated by spaces or commas []:
   m2n2.centrifyimage.vms
   Set MapR User Name [mapr]:
   Set MapR User Password [mapr]:  XXXXXX
   Is this cluster going to run YARN? (y/n) [y]:
   Is this cluster going to run MapReduce1? (y/n) [n]:
   Is this cluster going to run Apache HBase? (y/n) [n]:
   Is this cluster going to run MapR-DB? (y/n) [y]:
   Enter the full path of disks for hosts separated by spaces or commas []: /dev/sdb

           Current Information (Please verify if correct)
           ==============================================

           Accessibility settings:

               Cluster Name: "my.cluster.com"
               MapR User Name: "mapr"
               MapR Group Name: "mapr"
               MapR User UID: "2000"
               MapR User GID: "2000"
               MapR User Password (Default: mapr): "****"
   ```

```
        Functional settings:

            WireLevel Security: "n"
            MapReduce Services: "n"
            YARN: "y"
            MapR-DB: "y"

            HBase: "n"
            Disks to use: "/dev/sdb"
            Client Nodes: ""
            Control Nodes: "m2n1.centrifyimage.vms"
            Data Nodes: "m2n2.centrifyimage.vms"
            Repository (will download core software from here):
"http://package.mapr.com/releases"
           Ecosystem Repository (will download packages like Pig, Hive etc from
here): "http://package.mapr.com/releases/ecosystem"

            MapR Version to Install: "4.0.1"
            Java Version to Install: "OpenJDK7"
            Allow Control Nodes to function as Data Nodes (Not recommended for
large clusters): "n"
            Local Repository: "n"

        Metrics settings:

            Metrics DB Host and Port: ""
            Metrics DB User Name: ""
            Metrics DB User Password: ""
            Metrics DB Schema: ""

(c)ontinue with install, (m)odify options, or save current configuration and
(a)bort? (c/m/a) [c]: m

            Pick an option to modify
            ========================

            N] Cluster Name: "my.cluster.com"
            u] MapR User Name: "mapr"
            g] MapR Group Name: "mapr"
            U] MapR User UID: "2000"
            G] MapR User GID: "2000"
            p] MapR User Password: "*****"
            S] WireLevel Security: "n"
            d] Disk Settings: "/dev/sdb"
            sw] Disk Stripe Width: ""
            F] Force Format Disks: "n"
            c] Client Nodes: ""
            C] Control Nodes: "m2n1.centrifyimage.vms"
            D] Data Nodes: "m2n2.centrifyimage.vms"
            b] Control Nodes to function as Data Nodes: "n"
            v] Version: "4.0.1"
            L] Local Repository: "n"
            mr] MapReduce1: "n"
            db] MapR-DB: "y"
            hb] HBase: "n"
            y] YARN: "y"
```

```
                uc] Core Repo URL: "http://package.mapr.com/releases"
               ue] Ecosystem Repo URL: "http://package.mapr.com/releases/ecosystem"
                dbh] Metrics DB Host and Port: ""
                dbu] Metrics DB User: ""
                dbp] Metrics DB Password: ""
                dbs] Metrics DB Schema: ""
                cont] Continue
                : N
Enter the Cluster Name [my.cluster.com]: Mapr2

                Pick an option to modify
                ========================

                N] Cluster Name: "Mapr2"
                u] MapR User Name: "mapr"
                g] MapR Group Name: "mapr"
                U] MapR User UID: "2000"
                G] MapR User GID: "2000"
                p] MapR User Password: "****"
                S] WireLevel Security: "n"
                d] Disk Settings: "/dev/sdb"
                sw] Disk Stripe Width: ""
                F] Force Format Disks: "n"
                c] Client Nodes: ""
                C] Control Nodes: "m2n1.centrifyimage.vms"
                D] Data Nodes: "m2n2.centrifyimage.vms"
                b] Control Nodes to function as Data Nodes: "n"
                v] Version: "4.0.1"
                L] Local Repository: "n"
                mr] MapReduce1: "n"
                db] MapR-DB: "y"
                hb] HBase: "n"
                y] YARN: "y"
                uc] Core Repo URL: "http://package.mapr.com/releases"
               ue] Ecosystem Repo URL: "http://package.mapr.com/releases/ecosystem"
                dbh] Metrics DB Host and Port: ""
                dbu] Metrics DB User: ""
                dbp] Metrics DB Password: ""
                dbs] Metrics DB Schema: ""
                cont] Continue
                : cont

            Current Information (Please verify if correct)
            ==============================================

            Accessibility settings:

                Cluster Name: "Mapr2"
                MapR User Name: "mapr"
                MapR Group Name: "mapr"
                MapR User UID: "2000"
                MapR User GID: "2000"
                MapR User Password (Default: mapr): "****"

            Functional settings:
```

```
            WireLevel Security: "n"
            MapReduce Services: "n"
            YARN: "y"
            MapR-DB: "y"

            HBase: "n"
            Disks to use: "/dev/sdb"
            Client Nodes: ""
            Control Nodes: "m2n1.centrifyimage.vms"
            Data Nodes: "m2n2.centrifyimage.vms"
            Repository (will download core software from here):
"http://package.mapr.com/releases"
          Ecosystem Repository (will download packages like Pig, Hive etc from
here): "http://package.mapr.com/releases/ecosystem"

            MapR Version to Install: "4.0.1"
            Java Version to Install: "OpenJDK7"
            Allow Control Nodes to function as Data Nodes (Not recommended for
large clusters): "n"
            Local Repository: "n"

        Metrics settings:

            Metrics DB Host and Port: ""
            Metrics DB User Name: ""
            Metrics DB User Password: ""
            Metrics DB Schema: ""


(c)ontinue with install, (m)odify options, or save current configuration and
(a)bort? (c/m/a) [c]: c
SSH Username: root
SSH password:
Now running on Control Nodes: [m2n1.centrifyimage.vms]
* 12:34:33 Interrogating Node(s), Validating Prerequisites, and Starting Install
* 12:34:41 Installing Extra Package Repositories If Needed
* 12:34:43 Installing Extra Package Repositories If Needed for CentOS/RedHat
* 12:34:59 Detecting Operating System
* 12:35:01 Installing Prerequisite Packages for CentOS/RedHat
* 12:37:48 Detecting Operating System
* 12:37:52 Configuring Firewall for CentOS/RedHat
* 12:37:56 Creating MapR User
* 12:38:13 Installing and Configuring NTP Service
* 12:38:26 Installing OpenJDK Packages If Needed
* 12:39:06 Detecting Operating System
* 12:39:10 Initializing MapR Repository for CentOS/RedHat
* 12:40:49 Installing MapR Packages
* 12:44:54 Disabling MapR Services Until Configured
* 12:45:04 Configuring MapR Services
* 12:45:24 Configuring Disks for MapR File System
* 12:45:44 Starting MapR Services
* 12:46:01 Finalizing MapR Cluster Configuration
* 12:50:17 Configuring MapR Ecosystem
* 12:50:21 Configuring Hive
* 12:50:27 Configuring Spark
```

```
MapR Installation Successful on Control Nodes. Please login via the web console
at https://m2n1.centrifyimage.vms:8443 or manage the cluster using 'maprcli' or
'hadoop' commands

Now running on Data Nodes: [m2n2.centrifyimage.vms]
* 12:50:34 Interrogating Node(s), Validating Prerequisites, and Starting Install
* 12:50:38 Installing Extra Package Repositories If Needed
* 12:50:41 Installing Extra Package Repositories If Needed for CentOS/RedHat
* 12:51:01 Detecting Operating System
* 12:51:03 Installing Prerequisite Packages for CentOS/RedHat
* 12:53:43 Detecting Operating System
* 12:53:47 Configuring Firewall for CentOS/RedHat
* 12:53:51 Creating MapR User
* 12:54:06 Installing and Configuring NTP Service
* 12:54:23 Installing OpenJDK Packages If Needed
* 12:54:57 Detecting Operating System
* 12:55:01 Initializing MapR Repository for CentOS/RedHat
* 12:56:34 Installing MapR Packages
* 13:00:19 Disabling MapR Services Until Configured
* 13:00:26 Configuring MapR Services
* 13:00:45 Configuring Disks for MapR File System
* 13:01:05 Starting MapR Services
* 13:01:25 Finalizing MapR Cluster Configuration
* 13:02:48 Configuring MapR Ecosystem
* 13:02:56 Configuring Hive
* 13:03:03 Configuring Spark
```

You now have a working environment with the Centrify agent controlling access to Linux hosts and coexisting with the MapR cluster. You can now view the dashboard of the cluster from a browser at `https://node1-hostname:8443`.

Enabling security for the cluster

As described in "Preparing for MapR integration" on page 33, MapR has its own security architecture, so Kerberos-based security is only available for a subset of Hadoop services, such as CLDB and HBase. Because of this separate security architecture, you also must manually enable Kerberos security for the services that support it. Therefore, before you can run the Centrify script to create service accounts and keytab files, some configuration steps are required to prepare the cluster to operate securely using Kerberos authentication, the Centrify agent, and Active Directory.

Key tasks for enabling security

Configuring Kerberos authentication for a MapR cluster involves both manual and automated tasks. As a preview, you should plan to perform the following tasks:

- Decide which MapR services should use Kerberos authentication.
- Prepare an input file with comma-separated values.
- Modify configuration files for each service that supports Kerberos authentication.
- Modify the Centrify `hadoop.conf` configuration file used by the automation script.

- Run the automation script on one cluster node.
- Enable security using the `configure.sh` script, which should automatically restart the cluster upon completion.

Preparing the input file

Centrify provides an automation script—`kerberos_security_setup.pl`—to manage service principal names (SPN) and `keytab` files in the cluster. To use the `kerberos_security_setup.pl` script, you must first manually create the input file for the script to use. The input file consists of the following information on each line:

- Host name of the computer node that the `keytab` file is delivered to.
- Display name to use for the MapR service account.
- Service principal name for the MapR service account.
- Name of the `keytab` file for the service account.
- Path to the `keytab` file.
- User name of the owner of the `keytab` file.
- Group name of the owner of the `keytab` file.
- Permissions set on the `keytab` file.

You can find a sample of the input file format in `/usr/share/centrifydc/samples/hadoop/host-principal-keytab-list.csv`. You must manually create the input file to include the appropriate information for each service and each of the computers in the cluster. For example, if you have two nodes in the cluster, the input file might include the following lines:

```
mapr1.test.org,HDFS
User,hdfs@TEST.ORG,hdfs.keytab,/etc/security/keytabs,hdfs,hadoop,440
mapr2.test.org,HDFS
User,hdfs@TEST.ORG,hdfs.keytab,/etc/security/keytabs,hdfs,hadoop,440
```

To create the input file

1. Log on to the computer you are using as the control node for the cluster.

   The control node is the computer where you run the container location database (CLDB) service.

2. Copy the sample input file to create a working copy.

   For example:

   ```
   cd /usr/share/centrifydc/samples/hadoop
   cp host-principal-keytab-list.csv mymaprinput.csv
   ```

3. Open the file in a text editor and replace the sample content with the information for your cluster.

4. Save the text file.

Modifying configuration files

Because MapR has its own security architecture, you must manually configure the services that support Kerberos authentication to use it. The specific files you must modify and the changes required depend on the which services you have identified as requiring Kerberos security. For example, if you are enabling Kerberos for user authentication, you should modify the `/opt/mapr/conf/mapr.login.conf` file.

For the most common components—such as the container location database (CLDB), MapR Control System (MCS), and the Hadoop database (HBase)—you need to modify the following files on each node in the cluster:

```
/opt/mapr/conf/mapr.login.conf
/opt/mapr/conf/web.conf
/opt/mapr/hbase/hbase-0.94.21/conf/hbase-site.xml
/opt/mapr/conf/env.sh
```

For more detailed information about enabling Kerberos-based security for different components and services, see the MapR documentation and the references in the `INSTALL` file included with the Centrify agent.

Modifying the hadoop.conf file

Before running the `kerberos_security_setup.pl` automation script, you should modify the default `hadoop.conf` configuration file on the control node that runs the container location database (CLDB) service. The `hadoop.conf` configuration file controls how objects are created and named in Active Directory and how commands are executed on remote hosts when the automation script copies files and sets permissions.

At a minimum, you should uncomment and set the following properties in the `hadoop.conf` configuration file:

- hadoop.service.container

  Use this property to specify the appropriate Active Directory location you created for Hadoop objects. For example:

  ```
  OU=Cluster1,OU=Hadoop
  ```

- hadoop.cluster.shortname

  Use this property to specify a short name for the cluster to be used as a prefix to differentiate the same service running on computers in different clusters.

- hadoop.host.shortname.extracted.last.chars

  Use this property to specify the number of character to extract from the host name to create a short name for the computers in a cluster.

For more information about these properties, the default naming conventions, and other configuration properties you can set, see the comments in the `hadoop.conf` file.

Running the automation script

After you prepare the input file and modify the `hadoop.conf` configuration file, you are ready to get a Kerberos credential and run the `kerberos_security_setup.pl` automation script. The `kerberos_security_setup.pl` script performs the following steps for you:

- Verify the Centrify agent is installed and joined to Active Directory on all cluster nodes.
- Delete the default HTTP SPN from computer object in Active Directory, if necessary.
- Create the Kerberos `keytab` directory on all cluster nodes.
- Create the local directory to generate Kerberos `keytab` files.
- Create the service account principals n Active Directory.
- Create the `keytab` files in the local directory.
- Distribute the `keytab` files to the computers and locations specified in the input file.
- Set the correct ownership and permission for all `keytab` files distributed to the computers in the cluster.
- Modify agent configuration file, if necessary.

To run the automation script

1. Ensure that these prerequisites are met:
   - You have `root` access to the computer where you will run the automation script.
   - You have secure shell (SSH) and secure copy (SCP) access to all cluster nodes without having to provide a password. One way to achieve this is to use public key authentication.
   - You have the necessary permissions to generate Kerberos tickets with the `kinit` command.
2. Log on to the computer you are using as the control node for the cluster.
3. Run `kinit` to get a Kerberos ticket for the administrative account you are using to run the script.
4. Execute the `kerberos_security_setup.pl` automation script using the `--dry-run` option to review the commands that will be executed without running the script.
   ```
   perl kerberos_security_setup.pl --input mymaprinput.csv -c hadoop.conf --create
   --dry-run
   ```
5. Execute the `kerberos_security_setup.pl` automation script using the input file that you prepared to create service accounts and `keytab` files:
   ```
   perl kerberos_security_setup.pl --input myinput.csv -c hadoop.conf --create
   ```
6. Execute the `kerberos_security_setup.pl` automation script with the `--deploy` option to distribute the `keytab` files to the appropriate nodes in the cluster, and set the file ownership and permission:
   ```
   perl kerberos_security_setup.pl --input myinput.csv --deploy
   ```

7 Verify the service principals accounts have been created by opening Active Directory Users and Computers and checking the organizational unit you created for the MapR cluster.

Default settings for service accounts and keytab files

Centrify automatically maintains the `keytab` entries that are part of the computer account when `adclient` joins a domain. By default, `adclient` randomly generates a new password for its computer account every 28 days.

Other service principals and `keytab` files—such as those created for Hadoop services—are not automatically refreshed. By default, however, the automation script creates the accounts for Hadoop services as user accounts with passwords that never expire. These default settings eliminate long term maintenance for the service accounts, but might raise regulatory compliance issues in some organizations. You can modify the `hadoop.conf` file to create the accounts as computer accounts with passwords that never expire. In most cases, you should not modify the automation script to remove the "`--password-never-expire`" setting.

If using a computer account with the "`--password-never-expire`" option does not resolve your compliance issues, you could write a script that periodically run the `adkeytab` command with the `-C` option to change the password and update the `keytab` file for a specified account. For more information about the `adkeytab` program and command line options, see the `adkeytab` man page.

Using the configure.sh script to enable security

After running the automation script, you must enable security on each node in the cluster using the `configure.sh` script. Before running the `configure.sh` script, however, you should perform the following tasks:

- Verify the following files are not in the `/opt/mapr/conf` directory:

  ```
  cldb.key
  maprserviceticket
  ssl_keystore
  ssl_truststore
  ```

  If any of these files exist, delete them.

- Verify that the MapR security setting is set to false in the `/opt/mapr/conf/mapr-clusters.conf` file.

  ```
  Mapr2 secure=false m2n1.centrifyimage.vms:7222
  ```

To enable MapR security with configure.sh

1 Log on to the computer you are using as the control node for the cluster.

2 Shut down the cluster services on the control node.

```
service mapr-warden stop
service mapr-zookeeper stop
```

Note that to shut down the cluster completely, you must run `service mapr-warden stop` on every node and `service mapr-zookeeper stop` on all of the nodes where the zookeeper service runs.

3 Run the `configure.sh` script with the appropriate command-line options on the control node.

```
/opt/mapr/server/configure.sh –secure –genkeys –C CLDB-Node –Z Zookeeper-Node
–N cluster-name
```

4 Copy key files from the secured control node to all other nodes in the cluster

For example, copy the following files to the/opt/mapr/conf directory

```
cldb.key
maprserviceticket
ssl_keystore
ssl_truststore
```

Note that you should only copy the `cldb.key` file to any node that has the CLDB or Zookeeper service installed. You can use `chmod` to set the permissions on the `ssl_truststore` files to 444. You can set the permissions on the other files to 600 or 400.

5 Run the `configure.sh` script with the appropriate command-line options on all of the other nodes in the cluster.

```
/opt/mapr/server/configure.sh –secure –C CLDB-Node –Z Zookeeper-Node
-N cluster-name
```

Running the `configure.sh` script should restart the cluster automatically.

Validating MapR cluster security

Now that the MapR cluster is using Centrify for Active Directory based authentication, the user Diana Worth can log on using her Active Directory credentials directly at the console prompt or could use a Kerberized SSH client such as the Centrify version of PuTTY to get single sign-on access to the cluster.

Once logged in, she will have Kerberos credentials and will be able to run Hadoop jobs such as the example used below that computes the value of Pi. Since the cluster is now running in secure mode, users without Kerberos will not be able to successfully submit a job to the cluster.

```
[dwirth@m2n1 hadoop-0.20.2]$ kinit
Password for dwirth@CENTRIFYIMAGE.VMS:
[dwirth@m2n1 hadoop-0.20.2]$ maprlogin kerberos
MapR credentials of user 'dwirth' for cluster 'Centrify1' are written to
'/tmp/maprticket_1627391058'
[dwirth@m2n1 hadoop-0.20.2]$ hadoop fs –mkdir /user/dwirth
[dwirth@m2n1 hadoop-0.20.2]$ hadoop fs –chown dwirth:dwirth /user/dwirth
[dwirth@m2n1 hadoop-0.20.2]$ hadoop fs -ls /user
Found 3 items
drwxr-xr-x - dwirth     dwirth        0 2014-12-08 15:24 /user/dwirth
drwxr-xr-x - mapr       mapr          1 2014-11-17 06:39 /user/mapr
```

```
[dwirth@m2n1 hadoop-0.20.2]$ hadoop jar ./hadoop-0.20.2-dev-examples.jar pi 6
50
Number of Maps  = 6
Samples per Map = 50
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
Wrote input for Map #5
Starting Job
15/01/12 08:26:01 INFO client.RMProxy: Connecting to ResourceManager at
m2n1.centrifyimage.vms/192.168.1.45:8032
15/01/12 08:26:02 INFO input.FileInputFormat: Total input paths to process : 6
15/01/12 08:26:02 INFO mapreduce.JobSubmitter: number of splits:6
15/01/12 08:26:03 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1421067725962_0001
15/01/12 08:26:03 INFO security.ExternalTokenManagerFactory: Initialized
external token manager class -
com.mapr.hadoop.yarn.security.MapRTicketManager
15/01/12 08:26:04 INFO impl.YarnClientImpl: Submitted application
application_1421067725962_0001
15/01/12 08:26:04 INFO mapreduce.Job: The url to track the job:
https://m2n1.centrifyimage.vms:8090/proxy/application_1421067725962_0001/
15/01/12 08:26:04 INFO mapreduce.Job: Running job: job_1421067725962_0001
15/01/12 08:26:18 INFO mapreduce.Job: Job job_1421067725962_0001 running in
uber mode : false
15/01/12 08:26:18 INFO mapreduce.Job:  map 0% reduce 0%
15/01/12 08:26:26 INFO mapreduce.Job:  map 17% reduce 0%

15/01/12 08:26:33 INFO mapreduce.Job:  map 33% reduce 0%
15/01/12 08:26:37 INFO mapreduce.Job:  map 67% reduce 0%
15/01/12 08:26:39 INFO mapreduce.Job:  map 83% reduce 0%
15/01/12 08:26:46 INFO mapreduce.Job:  map 100% reduce 0%
15/01/12 08:26:52 INFO mapreduce.Job:  map 100% reduce 100%
15/01/12 08:26:54 INFO mapreduce.Job: Job job_1421067725962_0001 completed
successfully
15/01/12 08:26:54 INFO mapreduce.Job: Counters: 46

        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=556491
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                MAPRFS: Number of bytes read=1900
                MAPRFS: Number of bytes written=633
                MAPRFS: Number of read operations=233
                MAPRFS: Number of large read operations=0
                MAPRFS: Number of write operations=164
        Job Counters
                Launched map tasks=6
                Launched reduce tasks=1
                Data-local map tasks=6
                Total time spent by all maps in occupied slots (ms)=55290
                Total time spent by all reduces in occupied slots (ms)=10542
```

```
                Total time spent by all map tasks (ms)=55290
                Total time spent by all reduce tasks (ms)=3514
                Total vcore-seconds taken by all map tasks=55290
                Total vcore-seconds taken by all reduce tasks=3514
                Total megabyte-seconds taken by all map tasks=56616960
                Total megabyte-seconds taken by all reduce tasks=10795008
                DISK_MILLIS_MAPS=27646
                DISK_MILLIS_REDUCES=4674
        Map-Reduce Framework
                Map input records=6
                Map output records=12
                Map output bytes=108
                Map output materialized bytes=0
                Input split bytes=798
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=144
                Reduce input records=12
                Reduce output records=0

                Spilled Records=24
                Shuffled Maps =6
                Failed Shuffles=0
                Merged Map outputs=7
                GC time elapsed (ms)=1606
                CPU time spent (ms)=4990
                Physical memory (bytes) snapshot=1523752960
                Virtual memory (bytes) snapshot=14398550016
                Total committed heap usage (bytes)=1217921024
        Shuffle Errors
                IO_ERROR=0
        File Input Format Counters
                Bytes Read=708
        File Output Format Counters
                Bytes Written=97
Job Finished in 52.807 seconds
Estimated value of Pi is 3.16000000000000000000
```

Disabling security for the cluster

If you disable security through Ambari and no longer need the principals, files, and directories that were created by the Centrify automated script when you enabled security, you can run the automated script with the `--delete` option to clean up unused items. For example, to delete principals, files, and directories in the sample environment described earlier, you would run the following script:

```
perl kerberos_security_setup.pl --input mymaprinput.csv --delete
```

If you are only disabling Centrify management of service accounts and `keytab` files, but still using Kerberos authentication for some MapR services, no further steps are necessary. If you want to disable Kerberos-based security and revert to MapR security for all services,

you should review and update the configuration files you modified to enable Kerberos-based security.

Troubleshooting

This section describes issues you might encounter when securing a MapR cluster using Centrify.

MapR ticket is not generated when using Centrify-enabled OpenSSH

MapR provides its own Pluggable Authenticator Module (`libmapr_pam.so`) that generates MapR tickets during login. You can add this module to PAM configuration files. If MapR ticket generation fails after configuring the MapR cluster to use Centrify, the issue is most likely the result of the PAM module failing to run when logging in using Centrify-enabled OpenSSH.

To work around this issue:

1. In `/etc/init.d/centrify-sshd`, modify the `start()` function by changing this line:
   ```
   $SSHD $OPTIONS || failure
   ```
   To this:
   ```
   LD_PRELOAD=/lib64/libgcc_s.so.1 $SSHD $OPTIONS || failure
   ```
2. Restart the `centrify-sshd` service.

It is highly recommended that you test this workaround in a lab environment before deploying it to a production environment.

It is recommended that, if possible, you use a login script defined in the **Specify commands to run** group policy instead of the `libmapr_pam.so` MapR PAM module to initialize MapR tickets. For more information, see the *Centrify Server Suite Group Policy Guide* and http://doc.mapr.com/display/MapR/The+maprlogin+Utility#ThemaprloginUtility-MapRTicketsandthePAMAuthenticator.

Incompatible cryptography results in failed handshakes

The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST). If you are securing a MapR cluster with Centrify and using the open source Java Development Kit 7 (`jdk 1.7`), AES is not supported when using SPNEGO.

To workaround this issue, you can either of the following:

- Use the Oracle version of the Java Development Kit and an updated Java Cryptography Extension (JCE).

- Disable the use of AES encryption on the KDC and the Kerberos clients joining the domain. For this workaround, you need to modify the `centrifydc.conf` configuration file before joining the domain.

  Modify the following two parameters in the `centrifydc.conf` configuration file to disable AES encryption:

  ```
  adclient.krb5.tkt.encryption.types
  adclient.krb5.permitted.encryption.types
  ```

  For example:

  ```
  adclient.krb5.tkt.encryption.types: arcfour-hmac-md5 des-cbc-md5 des-cbc-crc
  ```

For more information about configuring Kerberos authentication and resolving encryption incompatibility, see the following topic:

http://doc.mapr.com/display/MapR/Configuring+Kerberos+User+Authentication

Disabling security on a cluster

If you run into issues, you might want to remove the service accounts, `keytab` files, and directories that were created by the Centrify automation script. You can re-run the automation script with the `--undeploy` and `--delete` options to clean up unused items.

To disable security on a cluster:

1. Execute `kerberos_security_setup.pl` with the `--undeploy` option to remove distributed `keytab` files from cluster nodes.
   ```
   perl kerberos_security_setup.pl --input myinput.csv --undeploy
   ```
2. Execute `kerberos_security_setup.pl` with the `--delete` option to delete the service accounts and their `keytab` files.
   ```
   perl kerberos_security_setup.pl --input myinput.csv --delete
   ```

Integrating with Hadoop manually

If you don't want to simplify the creation and distribution of service accounts and `keytab` files using the Centrify automation script, you can still use Centrify configuration parameters and command-line programs to manage the accounts and `keytab` files for common Hadoop services.

This chapter provides a quick reference to the parameters, command-line programs, and command-line options that are applicable for Hadoop integration. For more detailed information about setting configuration parameters or using Centrify command-line programs, see the following guides:

- *Configuration and Tuning Reference Guide*
- *Centrify Command Reference*

Key configuration parameters

You can use Centrify configuration parameters to automate Kerberos ticket renewal for service accounts, users, and groups on Hadoop. The following sections describe how to configure automatic renewal.

Configure service accounts for automatic renewal

Service accounts can be used to move files from one computer to another without providing a password. You can configure the Kerberos tickets for these service accounts to automatically renew without expiring.

If you want the `adclient` process to automatically renew service account credentials, you need to perform the following tasks:

- Configure the `adclient.krb5.cache.renewal.service.accounts` configuration parameter with a list of service accounts eligible for automatic renewal.
- Configure the *`service-account`*`.krb5.ccache.unixnames` configuration parameter with the user names where the service account credentials will be cached.

adclient.krb5.cache.renewal.service.accounts

Use the `adclient.krb5.cache.renewal.service.accounts` configuration parameter to specify the individual service accounts that should be renewed automatically or to specify a file name that contains a list of service accounts that should be renewed automatically.

For example, to specify a list of individual service accounts, you would the set `adclient.krb5.cache.renewal.service.accounts` parameter with values similar to this:

```
adclient.krb5.cache.renewal.service.accounts: ambari-qa,hdfs
```

Alternatively, if you have created a file with a list of service accounts, you would the set `adclient.krb5.cache.renewal.service.accounts` parameter value similar to this:
`adclient.krb5.cache.renewal.service.accounts: file:/tmp/hadoop-services.lst`

The default setting for the parameter is:
`adclient.krb5.cache.renewal.service.accounts:file:/etc/centrifydc/service_accts.lst`

service-account.krb5.ccache.unixnames

After you create the list of service accounts that can have Kerberos tickets renewed, you can use the *`service-account`*`.krb5.ccache.unixnames` configuration parameter to specify the user cache file where the service account credentials will be stored. For example, to have the service account credentials for the `hdfs` service stored in the `hdfsaccount` cache, you might set the configuration parameter like this:
`hdfs.krb5.ccache.unixnames: hdfsaccount`

You can specify multiple user names for any service, separated by commas. For example:
`hdfs.krb5.ccache.unixnames: hdfsm01,hdfsm02,hdfsm03`

Configure users for automatic renewal

You can specify users whose Kerberos credentials require infinite renewal even after the users have logged out. Specify users to automatically renew by listing them in the `krb5.cache.infinite.renewal.batch.users` configuration parameter in `centrifydc.conf`, or by listing them in the **Specify users to infinitely renew Kerberos credentials** group policy.

krb5.cache.infinite.renewal.batch.users

Use this configuration parameter in `centrifydc.conf` to specify a list of users whose Kerberos credentials require infinite renewal even after the users have logged out. These users must be zone enabled (that is, mapped users are not supported).

You can use any of the following formats to specify user names:
`unixName`
`userPrincipleName`
`SamAccountName`
`SamAccountName@domain`

For example:
```
krb5.cache.infinite.renewal.batch.users test_user, test_user@example.com,
test_user_sam, test_user_sam@example.com
```

By default, this parameter does not list any users.

Specify users to infinitely renew Kerberos credentials

Use this group policy to specify a list of users whose Kerberos credentials require infinite renewal even after the users have logged out. Users that you specify must be zone enabled (that is, mapped users are not supported).

If this group policy is enabled, user credentials are renewed automatically. To specify users when you enable this group policy, use any of the user name formats that are supported by the `krb5.cache.infinite.renewal.batch.users` parameter.

This group policy is located in **Computer Configuration > Centrify Settings > DirectControl Settings > Kerberos Settings**.

By default, this group policy is disabled.

This group policy modifies the `krb5.cache.infinite.renewal.batch.users` setting in the `centrifydc.conf` configuration file.

Configure groups for automatic renewal

You can specify groups whose members' Kerberos credentials require infinite renewal even after the group members have logged out.Specify Active Directory groups whose members' credentials are automatically renewed by listing the groups in the `krb5.cache.infinite.renewal.batch.groups` configuration parameter in `centrifydc.conf`, or by listing them in the **Specify groups to infinitely renew Kerberos credentials** group policy.

krb5.cache.infinite.renewal.batch.groups

Use this configuration parameter in `centrifydc.conf` to specify a list of Active Directory groups whose members' Kerberos credentials require infinite renewal even after the users have logged out. Groups that you specify must be Active Directory groups, but do not need to be zone enabled. However, only zone enabled users in a group will have their credentials automatically renewed.

You must use the following format to specify group names:
`SamAccountName@domain`

For example:
`krb5.cache.infinite.renewal.batch.groups test_group_sam@example.com`

By default, this parameter does not list any groups.

Specify groups to infinitely renew Kerberos credentials

Use this group policy to specify a list of Active Directory groups whose members' Kerberos credentials require infinite renewal even after the users have logged out. Groups that you specify must be Active Directory groups, but do not need to be zone enabled. However, only zone enabled users in a group will have their credentials automatically renewed.

If this group policy is enabled, group members' credentials are renewed automatically. To specify groups when you enable this group policy, you must use the *`SamAccountName@domain`* group name format. For example:
`test_group_sam@example.com`

This group policy is located in **Computer Configuration > Centrify Settings > DirectControl Settings > Kerberos Settings**.

By default, this group policy is disabled.

This group policy modifies the `krb5.cache.infinite.renewal.batch.groups` setting in the `centrifydc.conf` configuration file.

Key adkeytab parameters

You can use the Centrify adkeytab program to manually create, adopt, update, or delete service account principals and `keytab` files for Hadoop services. The key command-line parameters that are specifically for Hadoop clusters are the following:

`-M, --computer-object`
: This option allows you to create shared "headless" accounts for services such as `ambari-qa`, `hdfs`, `hbase` that are on all nodes in a cluster as computer accounts instead of user accounts. The primary reason to use this option is to avoid compliance issues that stem from having shared user accounts with shared passwords. If you create the service account as a computer object, you can have its password randomly generated so that it is never exposed to human users. You can use this option in conjunction with the `--password-never-expire` option to avoid compliance issues and service account maintenance.

`-W, --password-never-expire`
: This option sets the service account password to never expire when creating a new account.

`-L, --localconfig`
: This option adds service account and keytab information to the centrifydc.conf configuration file to indicate that a specified account is managed by the Centrify agent. This option must be used with --adopt option.

`-O, --ccache-unixusers`
: This option specifies a comma-separated list of UNIX user names to be used for the credential cache when storing service account tickets.

For more information about using adkeytab and the command-line options available to perform different tasks, see the `man` page for `adkeytab` or the Centrify Command Reference.

Enabling dzdo execution of the automation script

This chapter describes how to configure your environment so that the `kerberos_security_setup.pl` automation script can be executed with `dzdo`.

Overview

By default, the automation script `kerberos_security_setup.pl` can be run only by root. Some organizations might have policies that restrict privilege escalation and root access. In such cases, you can configure your environment to use `sudo` or `dzdo` to control who can run the `kerberos_security_setup.pl` automation script.

After you perform the procedures described in this chapter, an Active Directory user can run the automation script with root privilege (through `dzdo`) on a master node. The automation script then connects to cluster nodes through SSH using the Active Directory user, and executes commands with root privilege (through `dzdo`).

Configuring your environment for `dzdo` execution of the automation script requires several manual configuration tasks. You can also perform several optional tasks depending on your needs. As a preview, the required and optional tasks are as follows. See "Performing the configuration" on page 56 for details about completing these tasks.

- Edit the `hadoop.conf` configuration file to enable SSH for dzdo, specify the user who can run commands on nodes through SSH, and specify the user who can copy files to nodes through SCP.
- Configure an Active Directory user profile in the zone containing the cluster nodes.
- Configure roles and command rights for the Active Directory user.
- Optional: Manually expire the Centrify cache on each node so that the new roles and command rights are loaded immediately.
- Set or override the `KRB5CCNAME` environment variable.
- Generate a valid TGT for the Kerberos principal.
- Enable single sign-on or set up public key authentication so that a password prompt is not generated when the automation script runs SSH and SCP.
- Run the automation script as an Active Directory user.
- Delete the TGT for the Kerberos principal.

Performing the configuration

The following sections describe how to perform the required and optional tasks to configure your environment so that the `kerberos_security_setup.pl` automation script can be executed with `dzdo`.

Editing the hadoop.conf configuration file

1. Open the `hadoop.conf` file for editing.

   The file is located in `/usr/share/centrifydc/samples/hadoop`.

2. Configure parameters as shown here to enable `dzdo` when running commands through SSH on cluster nodes.
   ```
   hadoop.secure.shell.privilege.enable: true
   hadoop.secure.shell.privilege: dzdo
   ```

3. Configure parameters as shown here to specify the UNIX user account to run commands through SSH on cluster nodes. In this example, the Active Directory user with the UNIX name `aduser` is used. You can use a UNIX user name of your choice to fit your needs.
   ```
   hadoop.secure.shell.user.enable: true
   hadoop.secure.shell.user: aduser
   ```

4. Configure parameters as shown here to specify the Kerberos credentials cache for secure shell (SSH), which will override environment variable KRB5CCNAME.
   ```
   hadoop.secure.shell.krb5ccname.enable: true
   hadoop.secure.shell.krb5ccname: /tmp/krb5cc_hadoop_ssh
   ```

5. Configure parameters as shown here to specify the UNIX user account to copy files (for example, Kerberos keytab files) through SCP to cluster nodes. In this example, the local root user is used:
   ```
   hadoop.secure.copy.user.enable: true
   hadoop.secure.copy.user: root
   ```

6. Configure parameters as shown here to specify the Kerberos credentials cache for secure copy (SCP), which will override environment variable KRB5CCNAME.
   ```
   hadoop.secure.copy.krb5ccname.enable: true
   hadoop.secure.copy.krb5ccname: /tmp/krb5cc_hadoop_scp
   ```

Configuring an Active Directory user profile

In the Centrify zone containing the cluster nodes, create a profile for the Active Directory user with the UNIX name that you specified in Step 3 of the preceding section (for example, `aduser`). Configure the profile so that the Active Directory user can log in to all cluster nodes.

Configuring roles and command rights

Set up roles and command rights for the Active Directory user (`aduser` in the preceding example). For details about setting up roles and command rights, see the "Managing access rights and roles" chapter in the *Centrify Server Suite Administrator's Guide for Windows*.

When you set up roles and rights, grant the user command rights to run the sample script and to run commands through SSH on cluster nodes. You can grant command rights in a glob expression or a regular expression. The examples shown here create a very specific and granular set of command rights. The command rights that you create should be specific to your own environment.

Command rights in a glob expression

```
/usr/share/centrifydc/samples/hadoop/kerberos_security_setup.pl
/usr/bin/adinfo
test -d /etc/security/keytabs
chown *:* /etc/security/keytabs/*
perl -e sysopen(undef, "/var/centrify/tmp/centrifydc.conf.lock", 194, 0600) ||
die;
cp --force /etc/centrifydc/centrifydc.conf
/var/centrify/tmp/centrifydc.conf.tmp
echo adclient.krb5.service.principals: ftp cifs nfs
mv --force /var/centrify/tmp/centrifydc.conf.tmp
/etc/centrifydc/centrifydc.conf
rm /var/centrify/tmp/centrifydc.conf.lock
tee --append /var/centrify/tmp/centrifydc.conf.tmp
/usr/sbin/adkeytab --delspn -P http -m
/usr/sbin/adkeytab --new
/usr/sbin/adkeytab --delete
```

Command rights in a regular expression

```
mkdir -p /etc/security/keytabs/?.*
chmod [0-7]+ /etc/security/keytabs/?.*
rmdir /etc/security/keytabs/?.*
```

Note After you specify new command rights, you can verify them by executing the automation script with the `--dry-run` option, and then executing `dzinfo`. Output from the `--dry-run` option displays the new command rights that will be implemented when the automation script runs, and output from `dzinfo` displays the current command rights. You can compare the output from both commands to see if the new command rights will provide the new capabilities that you need for your environment.

Optional: Manually expire the DirectControl cache

To immediately load the newly created roles and command rights on a node, expire the DirectControl cache on the node manually. If you do not expire the cache manually, the roles and rights are loaded after the next scheduled cache expiration.

To expire the cache manually

1. Log into a node.
2. Execute the following command to expire the cache:
   `adflush --expire`
3. Execute the following command to verify that the new roles and rights were loaded:
   `dzinfo`
4. Log out of the node and execute the `adflush --expire` command on all other nodes.

Setting or overriding the KRB5CCNAME environment variable

The Centrify Agent resets the environment variable `KRB5CCNAME` for Active Directory users. After `KRB5CCNAME` is reset, some Kerberos utilities such as `klist` and `kinit` might not execute as expected when run as another user (for example, root) using `sudo` or `dzdo`. To correct this condition, you must set the environment variable `KRB5CCNAME` to the default credential cache (`/tmp/krb5cc_0`) or override the `KRB5CCNAME` setting in command rights.

Generating a TGT for the Kerberos principal

The automation script checks for a valid TGT before executing commands. Therefore, you must generate a valid TGT for the Kerberos principal before you run the automation script so that accounts can be created and deleted in Active Directory automatically. The Kerberos principal must have the administrative privilege to create and delete accounts in Active Directory. If it does not, commands and utilities run by the automation script (for example, `adkeytab`) might prompt for a password.

Because the sample script requires root privilege, you must generate the valid TGT as root.

Execute the following command to generate the TGT:

```
dzdo kinit administrator@EXAMPLE.COM -c /tmp/krb5cc_0
```

Preventing a password prompt when the script runs SSH and SCP

You must enable single sign-on (SSO) or set up public key authentication so that a password prompt is not generated when the automation script runs SSH and SCP.

SSH

You can prevent a password prompt when the automation script runs SSH in one of these ways:

- Create an Active Directory user as described in this chapter, and enable SSO. Then run the automation script as the Active Directory user.
- Create an Active Directory user as described in this chapter, and generate a public key. Then run the automation script as the Active Directory user.

SCP

You can prevent a password prompt when the automation script runs SCP in one of these ways:

- Map root to an Active Directory user *other than* the AD user that you configured in Step 3 on page 56 and "Configuring an Active Directory user profile" on page 56, and enable SSO. Then run the automation script as root or as another user with root privileges.

  Note Mapping root to a different AD user prevents two local users—root and `aduser` in this case—from being mapped to the same AD user.
- Generate a public key for root. Then run the automation script as root or as another user with root privileges.

To enable SSO

1. Generate a valid TGT for the Active Directory user as root using the non-default Kerberos ccache file. For example:
   ```
   dzdo kinit aduser@EXAMPLE.COM -c /tmp/krb5cc_hadoop_ssh.
   ```
2. Modify the subroutine `GetSSHCommand()` in the automation script to set `KRB5CCNAME` to the non-default Kerberos ccache file.

Executing the automation script

After you complete the configuration, execute the automation script as an Active Directory user.

1. On the master node, log in as the Active Directory user.
2. Execute the sample script as root using `dzdo`. For example:
   ```
   dzdo KRB5CCNAME=/tmp/krb5cc_0 ./kerberos\_security\_setup.pl --input host-
   principal-keytab-list.csv --create
   ```

Deleting the TGT

Perform the following procedure after you verify that the automation script runs correctly.

1. Delete the TGT by executing the following command:
   ```
   dzdo kdestroy -c /tmp/krb5cc_0
   ```
2. If you enabled SSO as described earlier, delete `/tmp/krb5cc_hadoop_ssh`.