Enabling Kerberos

Date of Publish: 2018-10-15

http://docs.hortonworks.com

Contents

Enable Kerberos

You can use Ambari to enable Kerberos for your Hortonworks Cybersecurity Platform (HCP) environment.

Checklist: Installing and Configuring the KDC

Ambari is able to configure Kerberos in the cluster to work with an existing MIT KDC, or existing Active Directory installation. This section describes the steps necessary to prepare for this integration.

You can choose to have Ambari connect to the KDC and automatically create the necessary Service and Ambari principals, generate and distribute the keytabs (“Automated Kerberos Setup”). Ambari also provides an advanced option to manually configure Kerberos. If you choose this option, you must create the principals, generate and distribute the keytabs. Ambari will not do this automatically (“Manual Kerberos Setup”).

Supported Key Distribution Center (KDC) Versions

- Microsoft Active Directory 2008 and above
- MIT Kerberos v5
- FreeIPA 4.x and above

There are four ways to install/configure the KDC:

- Using an existing MIT KDC
- Install a new MIT KDC (See "Optional: Install a new MIT KDC")
- Using an existing IPA
- Using an existing AD
- Using manual Kerberos setup

Option	Checklist
Using an existing MIT KDC	• Ambari Server and cluster hosts have network access to both the KDC and KDC admin hosts. • KDC administrative credentials are on-hand.
Install a new MIT KDC	See “Optional: Install a new MIT KDC”
Using an existing IPA	See “Optional: Use an Existing IPA”
Using an existing AD	• Ambari Server and cluster hosts have network access to, and be able to resolve the DNS names of, the Domain Controllers. • Active Directory secure LDAP (LDAPS) connectivity has been configured. • Active Directory User container for service principals has been created and is on-hand. For example, "OU=Hadoop,OU=People,dc=apache,dc=org" • Active Directory administrative credentials with delegated control of “Create, delete, and manage user accounts” on the previously mentioned User container are on-hand.
Using manual Kerberos setup	• Cluster hosts have network access to the KDC. • Kerberos client utilities (such as kinit) have been installed on every cluster host. • The Java Cryptography Extensions (JCE) have been setup on the Ambari Server host and all hosts in the cluster. • The Service and Ambari Principals will be manually created in the KDC before completing this wizard. • The keytabs for the Service and Ambari Principals will be manually created and distributed to cluster hosts before completing this wizard.

Optional: Install a new MIT KDC

The following gives a very high level description of the KDC installation process.

About this task

To get more information see specific Operating Systems documentation, such as RHEL documentation, CentOS documentation, or SLES documentation (links below).

Procedure

1. Install the KDC Server:
   a) Install a new version of the KDC server:

OS Flavor	Enter
RHEL/CentOS/Oracle Linux	yum install krb5-server krb5-libs krb5-workstation
SLES	zypper install krb5 krb5-server krb5-client
Ubuntu/Debian	apt-get install krb5-kdc krb5-admin-server

   b) Using a text editor, open the KDC server configuration file, located by default here: vi /etc/krb5.conf.
   c) Change the [realms] section of this file by replacing the default “kerberos.example.com” setting for the kdc and admin_server properties with the Fully Qualified Domain Name of the KDC server host. In the following example, “kerberos.example.com” has been replaced with “my.kdc.server”.

```
realms]
 EXAMPLE.COM = {
   kdc = my.kdc.server
   admin_server = my.kdc.server
}
```

2. Use the utility kdb5_util to create the Kerberos database:

OS Flavor	Enter
RHEL/CentOS/Oracle Linux	kdb5_util create -s
SLES	kdb5_util create -s
Ubuntu/Debian	krb5_newrealm

3. Start the KDC server and the KDC admin server:

OS Flavor	Enter
RHEL/CentOS/Oracle Linux 6	/etc/rc.d/init.d/krb5kdc start /etc/rc.d/init.d/kadmin start
RHEL/CentOS/Oracle Linux 7	systemctl start krb5kdc systemctl start kadmin
SLES	rckrb5kdc start rckadmind start
Ubuntu/Debian	service krb5-kdc restart service krb5-admin-server restart

4. Set up the KDC server to auto-start on boot:

OS Flavor	Enter
RHEL/CentOS/Oracle Linux 6	chkconfig krb5kdc on chkconfig kadmin on
RHEL/CentOS/Oracle Linux 7	systemctl enable krb5kdc systemctl enable kadmin
SLES	chkconfig rckrb5kdc on chkconfig rckadmind on
Ubuntu/Debian	update-rc.d krb5-kdc defaults update-rc.d krb5-admin-server defaults

5. Create a Kerberos Admin:

   Kerberos principals can be created either on the KDC machine itself or through the network, using an “admin” principal. The following instructions assume you are using the KDC machine and using the kadmin.local command line administration utility. Using kadmin.local on the KDC machine allows you to create principals without needing to create a separate "admin" principal before you start.

   a) Create a KDC admin by creating an admin principal: kadmin.local -q "addprinc admin/admin".

   b) Confirm that this admin principal has permissions in the KDC ACL. Using a text editor, open the KDC ACL file:

OS Flavor	Enter
RHEL/CentOS/Oracle Linux	vi /var/kerberos/krb5kdc/kadm5.acl
SLES	vi /var/lib/kerberos/krb5kdc/kadm5.acl
Ubuntu/Debian	vi /etc/krb5kdc/kadm5.acl

   c) Ensure that the KDC ACL file includes an entry so to allow the admin principal to administer the KDC for your specific realm. When using a realm that is different than EXAMPLE.COM, be sure there is an entry for the realm you are using. If not present, principal creation will fail. For example, for an admin/admin@HADOOP.COM principal, you should have an entry: */admin@HADOOP.COM *.

   d) After editing and saving the kadm5.acl file, you must restart the kadmin process:

OS Flavor	Enter
RHEL/CentOS/Oracle Linux 6	/etc/rc.d/init.d/kadmin restart
RHEL/CentOS/Oracle Linux 7	systemctl restart kadmin
SLES	rckadmind restart
Ubuntu/Debian	service krb5-admin-server restart

Optional: Use an Existing IPA

You can use an existing FreeIPA setup with Kerberos.

To use an existing IPA KDC with Automated Kerberos Setup, you must prepare the following:

- All cluster hosts should be joined to the IPA domain and registered in DNS- If IPA is not configured to authoritatively manage DNS, explicitly configuring the private IP and corresponding fully qualified domain names of all hosts, in the /etc/hosts file on all the hosts is recommended.
- If you do not plan on using Ambari to manage the krb5.conf file, ensure the following is set in each krb5.conf file in your cluster: default_ccache_name = /tmp/krb5cc_%{uid} - Redhat/Centos 7.x changed the default ticket cache to keyring, which is problematic for the hadoop components.
- The Java Cryptography Extensions (JCE) have been setup on the Ambari Server host and all hosts in the cluster- If during installation you chose to use the Ambari provided JDK, this has already been done for you. If

you configured a custom JDK, ensure the unlimited strength JCE policies are in place on all nodes. For more information, refer to “Install the JCE for Kerberos”.

Please also note:

- If you plan on leveraging this IPA to create trusts with other KDCs, please follow the FreeIPA “Considerations for Active Directory integration” to ensure your hosts use a non-overlapping DNS domain, with matching uppercase REALM.
- Kerberos authentication allows maximum 3 seconds time discrepancy. Use of IPA’s NTP server or an external time management service is highly recommended for all cluster hosts, including the FreeIPA host.
- To avoid exposing the IPA admin account, consider creating a dedicated hadoopadmin account that is a member of the admins group, or has been added to a role with User & Service Administratration privileges. Remember to reset the initial temporary password for the account before use in Ambari. For more details on this process see the section below.

Creating an IPA account for use with Ambari

Example creating hadoopadmin account with explicit privileges

```
# obtain valid ticket as IPA administrator
kinit admin

# create a new principal to be used for ambari kerberos administration
ipa user-add hadoopadmin --first=Hadoop --last=Admin --password

# create a role and and give it privilege to manage users and services
ipa role-add hadoopadminrole
ipa role-add-privilege hadoopadminrole --privileges="User Administrators"
ipa role-add-privilege hadoopadminrole --privileges="Service Administrators"

# add the hadoopadmin user to the role
ipa role-add-member hadoopadminrole --users=hadoopadmin

# login once, or kinit, to reset the initial temporary password for the
 hadoopadmin account
kinit hadoopadmin
```

Important: Do not install an Ambari Agent on the IPA host.

- IPA leverages the SPNEGO principal (HTTP/ipa.your.domain.com) for secure access to its Web UI component. Installing the Ambari Agent on the IPA host causes the kvno of SPNEGO principal to increase, which causes problems for IPA HTTP server. If you have already accidentally done this and IPA is not able to start, symlink IPA’s http keytab path (/var/lib/ipa/gssproxy/http.keytab) to /etc/security/keytabs/spnego.service.keytab and contact your IPA provider’s support.
- The /etc/krb5.conf file on the IPA host has some additional properties not captured in Ambari’s krb5.conf template. Since letting Ambari manage krb5.conf on the cluster hosts is recommended, making the IPA host a part of the cluster is problematic for the IPA services. If you had this option checked when the ambari agent was installed, and do not have a backup of the original krb5.conf, reference the “krb5.conf template” to restore immediate functionality.

Install the JCE for Kerberos

Before enabling Kerberos in the cluster, you must deploy the Java Cryptography Extension (JCE) security policy files on the Ambari Server and on all hosts in the cluster, including the Ambari Server. If you are using OpenJDK, some distributions of the OpenJDK (such as RHEL/CentOS and Ubuntu) come with unlimited strength JCE automatically and therefore, installation of JCE is not required.

Procedure

1. On the Ambari Server, obtain the JCE policy file appropriate for the JDK version in your cluster:

Option	
Oracle JDK 1.8	JCE Unlimited Strength Jurisdiction Policy Files 8 Download
Oracle JDK 1.7	JCE Unlimited Strength Jurisdiction Policy Files 7 Download

   wget --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jce/8/jce_policy-8.zip"
2. Save the policy file archive in a temporary location.
3. On Ambari Server and on each host in the cluster, add the unlimited security policy JCE jars to $JAVA_HOME/jre/lib/security/.

   For example, run the following to extract the policy jars into the JDK installed on your host:

```
unzip -o -j -q jce_policy-8.zip -d /usr/jdk64/jdk1.8.0_40/jre/lib/security/
```

4. Restart Ambari Server: sudo ambari-server restart.

What to do next

Proceed to “Running the Kerberos Security Wizard”.

Launch the Kerberos Wizard (Automated Setup)

Choose the Kerberos Wizard Automated Setup if you will use an existing MIT KDC or Active Directory, as opposed to managing Kerberos principals and keytabs manually.

Procedure

1. Be sure you have installed and configured your KDC and have prepared the JCE on each host in the cluster.
2. Log in to Ambari Web and Browse to Admin > Kerberos.
3. Click “Enable Kerberos” to launch the wizard.
4. Select the type of KDC you are using and confirm you have met the prerequisites.
5. Provide information about the KDC and admin account.
   a) In the KDC section, enter the following information:
      - In the KDC Host field, the IP address or FQDN for the KDC host. Optionally a port number may be included.
      - In the Realm name field, the default realm to use when creating service principals.
      - (Optional) In the Domains field, provide a list of patterns to use to map hosts in the cluster to the appropriate realm. For example, if your hosts have a common domain in their FQDN such as host1.hortonworks.local and host2.hortonworks.local, you would set this to: .hortonworks.local,hortonworks.local

   b) In the Kadmin section, enter the following information:
      - In the Kadmin Host field, the IP address or FQDN for the KDC administrative host. Optionally a port number may be included.
      - The Admin principal and password that will be used to create principals and keytabs.
      - (Optional) If you have configured Ambari for encrypted passwords, the Save Admin Credentials option will be enabled. With this option, you can have Ambari store the KDC Admin credentials to use when making cluster changes. Refer to “Managing Admin Credentials” for more information on this option.

6. Modify any advanced Kerberos settings based on your environment.
   a) (Optional) To manage your Kerberos client krb5.conf manually (and not have Ambari manage the krb5.conf), expand the Advanced krb5-conf section and uncheck the "Manage" option. You must have the krb5.conf configured on each host.

      When manually managing the krb5.conf it is recommended to ensure that DNS is not used for looking up KDC, and REALM entries. Relying on DNS can cause negative performance, and functional impact. To ensure that DNS is not used, ensure the following entries are set in the libdefaults section of your configuration.

      ```
      [libdefaults]
      dns_lookup_kdc = false
      dns_lookup_realm = false
      ```

   b) (Optional) to configure any additional KDC's to be used for this environment, add an entry for each additional KDC to the realms section of the Advanced krb5-conf's krb5-conf template.

      ```
      kdc = {{kdc_host}}
      kdc = otherkdc.example.com
      ```

   c) (Optional) To not have Ambari install the Kerberos client libraries on all hosts, expand the Advanced kerberos-env section and uncheck the “Install OS-specific Kerberos client package(s)” option. You must have the Kerberos client utilities installed on each host.
   d) (Optional) If your Kerberos client libraries are in non-standard path locations, expand the Advanced kerberos-env section and adjust the “Executable Search Paths” option.
   e) (Optional) If your KDC has a password policy, expand the Advanced kerberos-env section and adjust the Password options.
   f) (Optional) Ambari will test your Kerberos settings by generating a test principal and authenticating with that principal. To customize the test principal name that Ambari will use, expand the Advanced kerberos-env section and adjust the Test Kerberos Principal value. By default, the test principal name is a combination of cluster name and date (${cluster_name}-${short_date}). This test principal will be deleted after the test is complete.
   g) (Optional) If you need to customize the attributes for the principals Ambari will create, when using Active Directory, see “Customizing the Attribute Template” for more information. When using MIT KDC, you can pass Principal Attributes options in the Advanced kerberos-env section. For example, you can set options related to pre-auth or max. renew life by passing:
   -requires_preauth -maxrenewlife "7 days"
7. Proceed with the install.
8. Ambari will install Kerberos clients on the hosts and test access to the KDC by testing that Ambari can create a principal, generate a keytab and distribute that keytab.
9. Customize the Kerberos identities used by Hadoop and proceed to kerberize the cluster.

   On the Configure Identities step, be sure to review the principal names, particularly the Ambari Principals on the General tab. These principal names, by default, append the name of the cluster to each of the Ambari principals. You can leave this as default or adjust these by removing the "-${cluster-name}" from principal name string. For example, if your cluster is named HDP and your realm is EXAMPLE.COM, the hdfs principal will be created as hdfs-HDP@EXAMPLE.COM.
10. Confirm your configuration. You can optionally download a CSV file of the principals and keytabs that Ambari will automatically create.
11. Click Next to start the process.
12. After principals have been created and keytabs have been generated and distributed, Ambari updates the cluster configurations, then starts and tests the Services in the cluster.
13. Exit the wizard when complete.
14. Ambari Server communicates with components in the cluster, and now with Kerberos setup, you need to make sure Ambari Server is setup for Kerberos. As part of the automated Kerberos setup process, Ambari Server has

been given a keytab and setup is performed. All you need to do is restart Ambari Server for that to take effect. Therefore, restart Ambari Server at this time: ambari-server restart.

Set up TGT Renewal

Apache Storm does not handle automatic TGT renewal for running topologies. As a result, you must manage the TGT renewal process to ensure that your access does not expire. HCP includes a Python script you can use to manage the TGT renewal process. Run the script on an interval that is shorter than the renew_lifetime property configured for your TGT.

Procedure

1. On a node running Storm, install the following:

```
sudo yum install -y gcc krb5-devel python-devel
sudo yum install -y libffi libffi-devel
sudo yum install -y python-cffi
sudo yum install -y openssl-devel
```

*** Does Ubuntu run Kerberos? If so, do we need Ubuntu commands for this step?***

2. Set up Python with metron user:

```
su - metron
```

3. Export the following:

```
export PYTHON27_HOME=/opt/rh/python27/root
export LD_LIBRARY_PATH="/opt/rh/python27/root/usr/lib64"
```

4. Create a project_dir directory:

```
mkdir project_dir
```

5. Install ???

```
${PYTHON27_HOME}/usr/bin/virtualenv venv
source venv/bin/activate
pip install --upgrade setuptools==18.5
pip install requests-kerberos
```

6. Execute the tgt_renew.py script:

```
su - metron
python $METRON_HOME/bin/tgt_renew.py $HOST:PORT $TOPOLOGY_OWNER
```

Where:

HOST:PORT	The port for the Storm UI server.
TOPOLOGY_OWNER	The topology owner is typically "metron" for a Kerberized cluster with Metron topologies.

What to do next

Create a cron job to run the tgt_renew.py script at intervals shorter than the renew_lifetime property configured for your TGT.