

Configuring Ambari and Hadoop for Kerberos

This chapter describes how to configure Kerberos for strong authentication for Hadoop users and hosts in an Ambari-managed cluster.

Kerberos Overview

Strongly authenticating and establishing a user's identity is the basis for secure access in Hadoop. Users need to be able to reliably "identify" themselves and then have that identity propagated throughout the Hadoop cluster. Once this is done, those users can access resources (such as files or directories) or interact with the cluster (like running MapReduce jobs). Besides users, Hadoop cluster resources themselves (such as Hosts and Services) need to authenticate with each other to avoid potential malicious systems or daemon's "posing as" trusted components of the cluster to gain access to data.

Hadoop uses Kerberos as the basis for strong authentication and identity propagation for both user and services. Kerberos is a third party authentication mechanism, in which users and services rely on a third party - the Kerberos server - to authenticate each to the other. The Kerberos server itself is known as the **Key Distribution Center**, or **KDC**. At a high level, it has three parts:

- A database of the users and services (known as **principals**) that it knows about and their respective Kerberos passwords
- An **Authentication Server (AS)** which performs the initial authentication and issues a **Ticket Granting Ticket (TGT)**
- A **Ticket Granting Server (TGS)** that issues subsequent service tickets based on the initial **TGT**

A **user principal** requests authentication from the AS. The AS returns a TGT that is encrypted using the user principal's Kerberos password, which is known only to the user principal and the AS. The user principal decrypts the TGT locally using its Kerberos password, and from that point forward, until the ticket expires, the user principal can use the TGT to get service tickets from the TGS. Service tickets are what allow a principal to access various services.

Because cluster resources (hosts or services) cannot provide a password each time to decrypt the TGT, they use a special file, called a **keytab**, which contains the resource principal's authentication credentials. The set of hosts, users, and services over which the Kerberos server has control is called a **realm**.

Terminology

Term	Description
Key Distribution Center, or KDC	The trusted source for authentication in a Kerberos-enabled environment.
Kerberos KDC Server	The machine, or server, that serves as the Key Distribution Center (KDC).

Term	Description
Kerberos Client	Any machine in the cluster that authenticates against the KDC.
Principal	The unique name of a user or service that authenticates against the KDC.
Keytab	A file that includes one or more principals and their keys.
Realm	The Kerberos network that includes a KDC and a number of Clients.
KDC Admin Account	An administrative account used by Ambari to create principals and generate keytabs in the KDC.

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.

Note : If you do not have an existing KDC (MIT or Active Directory), [Install a new MIT KDC](#) . Please be aware that installing a KDC on a cluster host **after** installing the Kerberos client may overwrite the krb5.conf file generated by Ambari.

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”).

For convenience, use the instructions to [\(Optional\) Install a new MIT KDC](#) if you do not have an existing KDC available.

Install a new MIT KDC

The following gives a very high level description of the KDC installation process. To get more information see specific Operating Systems documentation, such as [RHEL documentation](#), [CentOS documentation](#), or [SLES documentation](#).

Note : Because Kerberos is a time-sensitive protocol, all hosts in the realm must be time-synchronized, for example, by using the Network Time Protocol (NTP). If the local system time of a client differs from that of the KDC by as little as 5 minutes (the default), the client will not be able to authenticate.

Install the KDC Server

1. Install a new version of the KDC server on Ubuntu/Debian. We select host impetus-i0163 as a KDC server as it has very less component installed as of now.

```
apt-get install krb5-kdc krb5-admin-server
```

Configuring krb5-admin-server

Setting up a Kerberos Realm

This package contains the administrative tools required to run the Kerberos master server.

However, installing this package does not automatically set up a Kerberos realm. This can be done later by running the "krb5_newrealm" command.

Please also read the /usr/share/doc/krb5-kdc/README.KDC file and the administration guide found in the krb5-doc package.

<Ok>

```
root@impetus-I0163:~# apt-get install krb5-kdc krb5-admin-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libverto-libevent1 libverto1
Suggested packages:
  openbsd-inetd inet-superserver krb5-kdc-ldap
The following NEW packages will be installed:
  krb5-admin-server krb5-kdc libverto-libevent1 libverto1
0 upgraded, 4 newly installed, 0 to remove and 680 not upgraded.
Need to get 269 kB of archives.
After this operation, 1,044 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu/ trusty/main libverto1 amd64 0.2.4-1ubuntu2
[9,178 B]
Get:2 http://in.archive.ubuntu.com/ubuntu/ trusty/main libverto-libevent1 amd64 0.2.4-
1ubuntu2 [5,354 B]
Get:3 http://in.archive.ubuntu.com/ubuntu/ trusty-updates/universe krb5-kdc amd64
1.12+dfsg-2ubuntu5.2 [171 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu/ trusty-updates/universe krb5-admin-server
amd64 1.12+dfsg-2ubuntu5.2 [82.8 kB]
Fetched 269 kB in 1s (162 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libverto1:amd64.
(Reading database ... 175540 files and directories currently installed.)
Preparing to unpack .../libverto1_0.2.4-1ubuntu2_amd64.deb ...
Unpacking libverto1:amd64 (0.2.4-1ubuntu2) ...
Selecting previously unselected package libverto-libevent1:amd64.
Preparing to unpack .../libverto-libevent1_0.2.4-1ubuntu2_amd64.deb ...
Unpacking libverto-libevent1:amd64 (0.2.4-1ubuntu2) ...
Selecting previously unselected package krb5-kdc.
Preparing to unpack .../krb5-kdc_1.12+dfsg-2ubuntu5.2_amd64.deb ...
Unpacking krb5-kdc (1.12+dfsg-2ubuntu5.2) ...
Selecting previously unselected package krb5-admin-server.
Preparing to unpack .../krb5-admin-server_1.12+dfsg-2ubuntu5.2_amd64.deb ...
Unpacking krb5-admin-server (1.12+dfsg-2ubuntu5.2) ...
Processing triggers for man-db (2.6.7.1-1) ...
Processing triggers for ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on next reboot
Setting up libverto1:amd64 (0.2.4-1ubuntu2) ...
Setting up libverto-libevent1:amd64 (0.2.4-1ubuntu2) ...
Setting up krb5-kdc (1.12+dfsg-2ubuntu5.2) ...
* Starting Kerberos KDC krb5kdc
krb5kdc: cannot initialize realm IMPETUS.CO.IN - see log file for details

[fail]
Processing triggers for ureadahead (0.100.0-16) ...
Setting up krb5-admin-server (1.12+dfsg-2ubuntu5.2) ...
```

```
* Starting Kerberos administrative servers kadmind
kadmind: No such file or directory while initializing, aborting
```

[fail]

```
Processing triggers for libc-bin (2.19-0ubuntu6) ...
Processing triggers for ureadahead (0.100.0-16) ...
```

2. Using a text editor, open the KDC server configuration file, located by default here:

```
vi /etc/krb5.conf
```

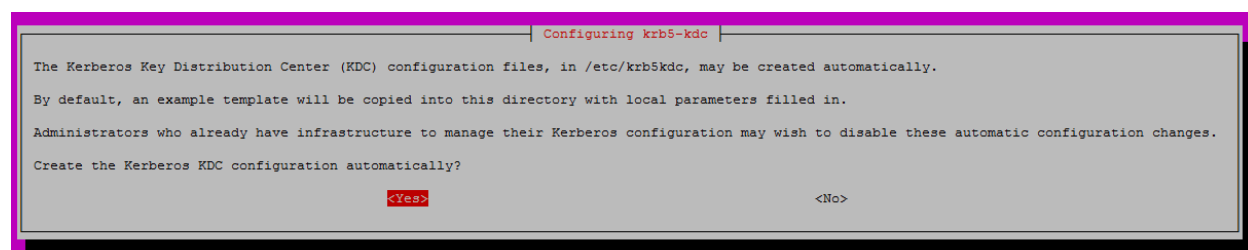
3. Change the [realms] section of this file by replacing the default “kerberos.example.com” setting for the kdc and admin_server properties with the Fully Qualified Domain Name of the KDC server host. In the following example, “kerberos.example.com” has been replaced with “my.kdc.server”.

```
[realms]
IMPETUS.CO.IN = {
kdc = impetus-I0163.impetus.co.in
kdc = impetus-dc11.impetus.co.in
admin_server = impetus-I0163.impetus.co.in impetus-dc11.impetus.co.in
}
```

4. Some components such as HUE require renewable tickets. To configure MIT KDC to support them, ensure the following settings are specified in the libdefaults section of the /etc/krb5.conf file.

```
[libdefaults]
default_realm = IMPETUS.CO.IN
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
```

Note : For Ubuntu/Debian, the setup of the default realm for the KDC and KDC Admin hostnames is performed during the KDC server install. You can re-run setup using `dpkg-reconfigure krb5-kdc`. Therefore, Steps 2 and 3 above are not needed for Ubuntu/Debian.



```
root@impetus-I0163:~# dpkg-reconfigure krb5-kdc
* Stopping Kerberos KDC krb5kdc
* Starting Kerberos KDC krb5kdc
```

```
krb5kdc: cannot initialize realm IMPETUS.CO.IN - see log file for details
```

[fail]

Create the Kerberos Database

- Use the below utility `kdb5_util` to create the Kerberos database for Ubuntu/Debian

krb5_newrealm

```
root@impetus-I0163:/etc# krb5_newrealm
This script should be run on the master KDC/admin server to initialize
a Kerberos realm. It will ask you to type in a master key password.
This password will be used to generate a key that is stored in
/etc/krb5kdc/stash. You should try to remember this password, but it
is much more important that it be a strong password than that it be
remembered. However, if you lose the password and /etc/krb5kdc/stash,
you cannot decrypt your Kerberos database.
Loading random data
Initializing database '/var/lib/krb5kdc/principal' for realm 'IMPETUS.CO.IN',
master key name 'K/M@IMPETUS.CO.IN'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
 * Starting Kerberos KDC krb5kdc
[ OK ]
 * Starting Kerberos administrative servers kadmind
[ OK ]
```

Now that your realm is set up you may wish to create an administrative principal using the `addprinc` subcommand of the `kadmin.local` program. Then, this principal can be added to `/etc/krb5kdc/kadm5.acl` so that you can use the `kadmin` program on other computers. Kerberos admin principals usually belong to a single user and end in `/admin`. For example, if `jruser` is a Kerberos administrator, then in addition to the normal `jruser` principal, a `jruser/admin` principal should be created.

Don't forget to set up DNS information so your clients can find your KDC and admin servers. Doing so is documented in the administration guide.

Start the KDC

Start the KDC server and the KDC admin server. For **Ubuntu** run the below command

- `service krb5-kdc restart`
- `service krb5-admin-server restart`

Important : When installing and managing your own MIT KDC, it is **very important** to **set up the KDC server to auto-start on boot**

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.

Note : You will need to provide these admin account credentials to Ambari when enabling Kerberos. This allows Ambari to connect to the KDC, create the cluster principals and generate the keytabs.

1. Create a KDC admin by creating an admin principal.

```
root@impetus-I0163:/etc# kadmin.local -q "addprinc admin/admin"
Authenticating as principal root/admin@IMPETUS.CO.IN with password.
WARNING: no policy specified for admin/admin@IMPETUS.CO.IN; defaulting to no policy
Enter password for principal "admin/admin@IMPETUS.CO.IN":
Re-enter password for principal "admin/admin@IMPETUS.CO.IN":
Principal "admin/admin@IMPETUS.CO.IN" created.
```

2. Confirm that this admin principal has permissions in the KDC ACL. Using a text editor, open the KDC ACL file and uncomment the following line `*/admin *`

For Ubuntu/Debian

```
vi /etc/krb5kdc/kadm5.acl
```

```
*/admin *
```

3. 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 *`

```
vi /etc/krb5kdc/kadm5.acl
```

```
*/admin@IMPETUS.CO.IN *
```

4. After editing and saving the `kadm5.acl` file, you must restart the `kadmin` process.

```
root@impetus-I0163:/etc# service krb5-admin-server restart
* Restarting Kerberos administrative servers kadmin [ OK ]
```

Enabling Kerberos Security

Whether you choose automated or manual Kerberos setup, Ambari provides a wizard to help with enabling Kerberos in the cluster. This section provides information on preparing Ambari before running the wizard, and the steps to run the wizard.

- [Installing the JCE](#)
- [Running the Kerberos Wizard](#)

Important : Prerequisites for enabling Kerberos are having the JCE installed on all hosts on the cluster (including the Ambari Server) and having the Ambari Server host as part of the cluster. This means the Ambari Server host should be running an Ambari Agent.

Note : Ambari Metrics will not be secured with Kerberos unless it is configured for distributed metrics storage. By default, it uses embedded metrics storage and will not be secured as part of the Kerberos Wizard. If you wish to have Ambari Metrics secured with Kerberos, please see [this topic](#) to enable distributed metrics storage prior to running the Kerberos Wizard.

Installing the JCE

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.

Important: If you are using Oracle JDK, **you must [distribute and install the JCE](#) on all hosts** in the cluster, including the Ambari Server. **Be sure to restart Ambari Server after installing the JCE.** If you are using OpenJDK, some distributions of the OpenJDK come with unlimited strength JCE automatically and therefore, installation of JCE is not required.

Install the JCE

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

- For Oracle JDK 1.8:

<http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html>

- For Oracle JDK 1.7:

<http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>

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/.

Download the file **jce_policy-8.zip** onto one server and then push to all

```
impetus-i0161:~$ scp jce_policy-8.zip
172.26.60.17:/home/IMPETUS/sudesh.shettigar
jce_policy-8.zip
100% 8409      8.2KB/s   00:00
```

```
impetus-i0161:~$ scp jce_policy-8.zip
172.26.60.18:/home/IMPETUS/sudesh.shettigar
jce_policy-8.zip
100% 8409      8.2KB/s   00:00
```

```
impetus-i0161:~$ scp jce_policy-8.zip
172.26.60.19:/home/IMPETUS/sudesh.shettigar
sudesh.shettigar@172.26.60.19's password:
jce_policy-8.zip
100% 8409      8.2KB/s   00:00
```

Run the following to extract the policy jars into the JDK installed on your host:

```
impetus-i0161:~$ sudo unzip -o -j -q jce_policy-8.zip -d  
/usr/jdk64/jdk1.8.0_60/jre/lib/security/
```

Restart Ambari Server.

```
root@impetus-i0161:~# ambari-server restart  
  
Using python /usr/bin/python  
Restarting ambari-server  
Using python /usr/bin/python  
Stopping ambari-server  
Ambari Server stopped  
Using python /usr/bin/python  
Starting ambari-server  
Ambari Server running with administrator privileges.  
About to start PostgreSQL  
Organizing resource files at /var/lib/ambari-server/resources...  
Server PID at: /var/run/ambari-server/ambari-server.pid  
Server out at: /var/log/ambari-server/ambari-server.out  
Server log at: /var/log/ambari-server/ambari-server.log  
Waiting for server start.....  
Ambari Server 'start' completed successfully.
```

Proceed to [Running the Security Wizard](#).

Running the Kerberos Security Wizard

Ambari provides three options for enabling Kerberos:

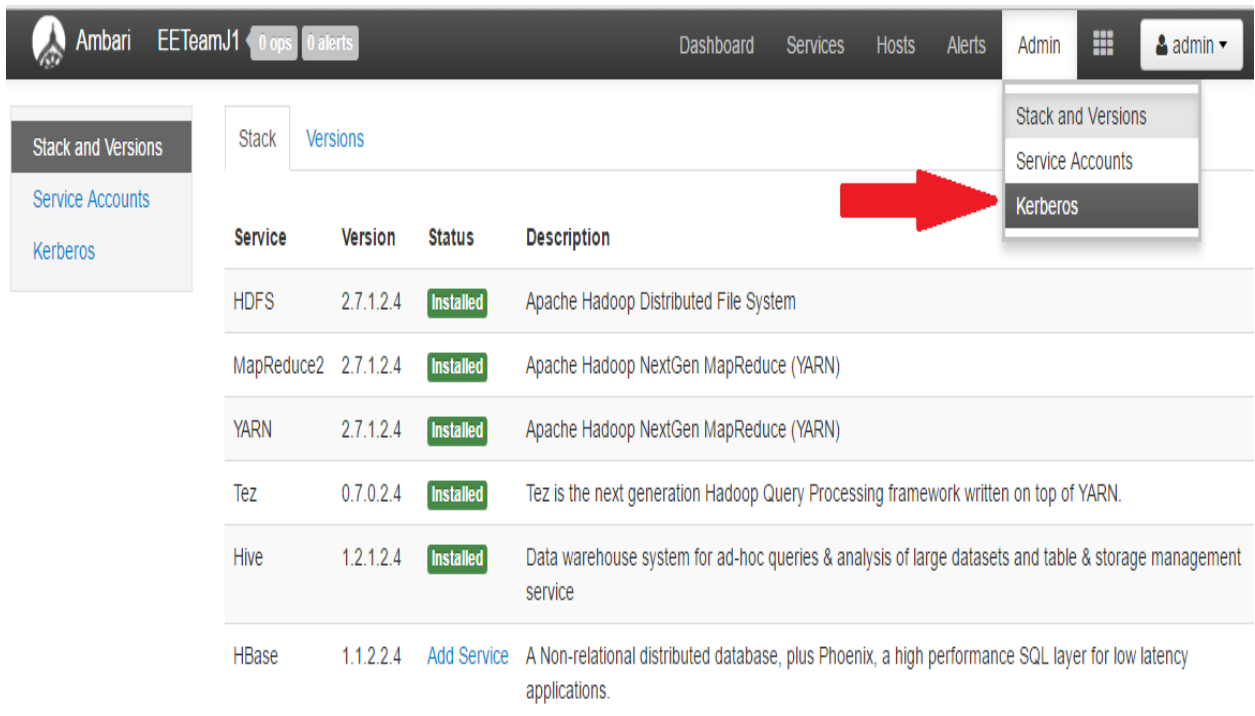
- Existing MIT KDC
- Existing Active Directory
- Manage Kerberos principals and keytabs manually

When choosing **Existing MIT KDC** or **Existing Active Directory**, the Kerberos Wizard prompts for information related to the KDC, the KDC Admin Account and the Service and Ambari principals. Once provided, Ambari will automatically create principals, generate keytabs and distribute keytabs to the hosts in the cluster. The services will be configured for Kerberos and the service components are restarted to authenticate against the KDC. We select the **Automated Setup** option.

When choosing **Manage Kerberos principals and keytabs manually**, you must create the principals, generate and distribute the keytabs. Ambari will not do this automatically. This is the **Manual Setup** option. See [Launching the Kerberos Wizard \(Manual Setup\)](#) for more details.

Launching the Kerberos Wizard (Automated Setup)

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.



Ambari EETeamJ1 0 ops 0 alerts

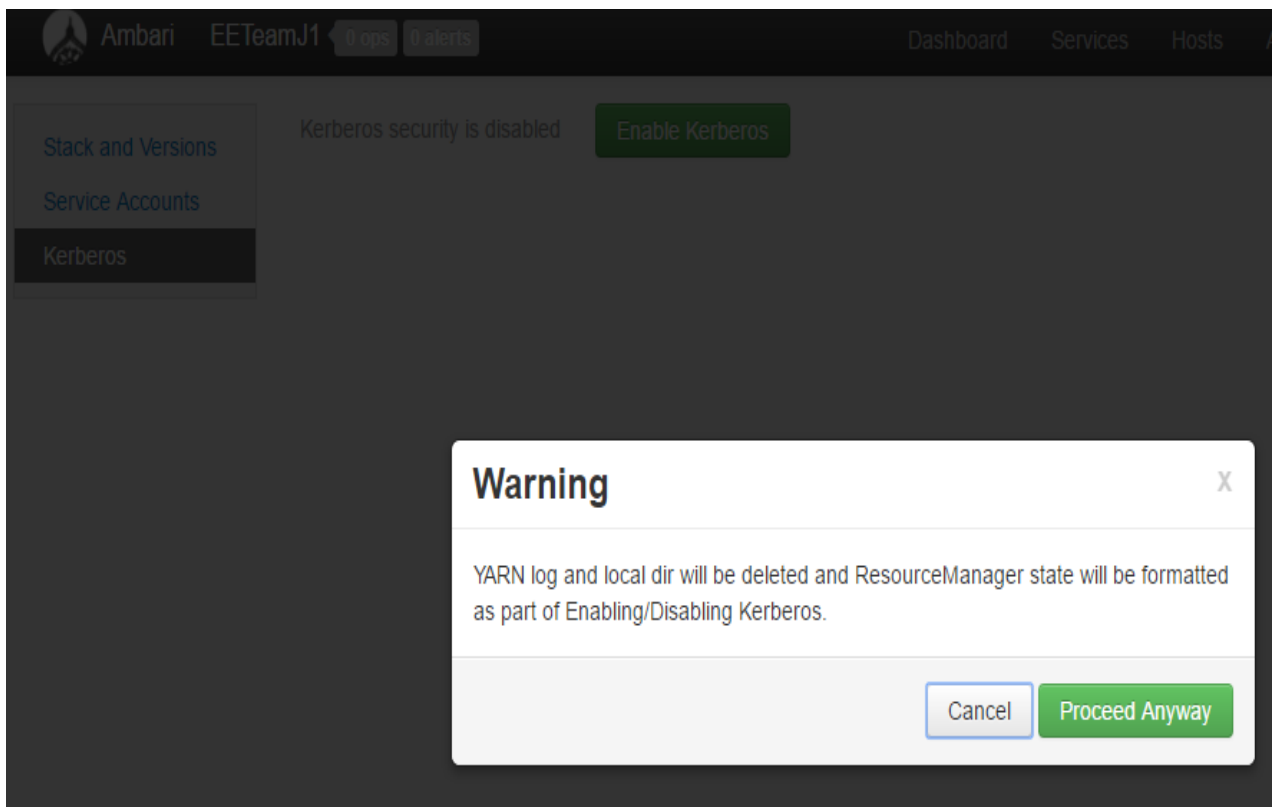
Dashboard Services Hosts Alerts Admin

Stack and Versions
Service Accounts
Kerberos

Stack Versions

Service	Version	Status	Description
HDFS	2.7.1.2.4	Installed	Apache Hadoop Distributed File System
MapReduce2	2.7.1.2.4	Installed	Apache Hadoop NextGen MapReduce (YARN)
YARN	2.7.1.2.4	Installed	Apache Hadoop NextGen MapReduce (YARN)
Tez	0.7.0.2.4	Installed	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
Hive	1.2.1.2.4	Installed	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
HBase	1.1.2.2.4	Add Service	A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications.

3. Click "Enable Kerberos" to launch the wizard.



Ambari EETeamJ1 0 ops 0 alerts

Dashboard Services Hosts

Stack and Versions
Service Accounts
Kerberos

Kerberos security is disabled [Enable Kerberos](#)

Warning

YARN log and local dir will be deleted and ResourceManager state will be formatted as part of Enabling/Disabling Kerberos.

[Cancel](#) [Proceed Anyway](#)

4. Select the type of KDC you are using and confirm you have met the prerequisites.

Get Started

Welcome to the Ambari Security Wizard. Use this wizard to enable kerberos security in your cluster.
Let's get started.

Note: This process requires services to be restarted and cluster downtime. As well, depending on the options you select, might require support from your Security administrators. Please plan accordingly.

What type of KDC do you plan on using?

- ☒ Existing MIT KDC
- ☐ Existing Active Directory
- ☐ Manage Kerberos principals and keytabs manually

Existing MIT KDC:
Following prerequisites needs to be checked to progress ahead in the wizard.

- ☒ Ambari Server and cluster hosts have network access to both the KDC and KDC admin hosts.
- ☒ KDC administrative credentials are on-hand.
- ☒ The Java Cryptography Extensions (JCE) have been setup on the Ambari Server host and all hosts in the cluster.

Next →

5. Provide information about the KDC and admin account. Please enter the details shown in picture.

In the **Kadmin** section, enter the **Admin principal** and **password** that was created in “Create a Kerberos Admin” section. 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.

Configure Kerberos

Please configure kerberos related properties.

Kerberos


▼ KDC

KDC type Existing MIT KDC

KDC host

Realm name

Domains


Connection OK 

▼ Kadmin

Kadmin host

Admin principal

Admin password

☐ Save Admin Credentials 

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.**

Note : 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 the 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.

Install and Test Kerberos Client

Kerberos service has been installed and tested successfully.

✓ [Install Kerberos Client](#)

✓ [Test Kerberos Client](#)

[← Back](#)

[Next →](#)

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.

Important : 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](#).

Configure Identities

Configure principal name and keytab location for service users and hadoop service components.

[General](#) [Advanced](#)

Global

Keytab Dir	<input type="text" value="/etc/security/keytabs"/>
Realm	<input type="text" value="IMPETUS.CO.IN"/>
Additional Realms	<input type="text" value="(Optional)"/>
Spnego Principal	<input type="text" value="HTTP/_HOST@\${realm}"/>
Spnego Keytab	<input type="text" value="\${keytab_dir}/spnego.service.keytab"/>

Ambari Principals

Smokeuser Principal Name	<input type="text" value="\${cluster-env/smokeuser}-\${cluster_name}@\${realm}"/>
Smokeuser Keytab	<input type="text" value="\${keytab_dir}/smokeuser.headless.keytab"/>
HDFS user principal	<input type="text" value="\${hadoop-env/hdfs_user}-\${cluster_name}@\${realm}"/>
Path to HDFS user keytab file	<input type="text" value="\${keytab_dir}/hdfs.headless.keytab"/>
spark.history.kerberos.principal	<input type="text" value="\${spark-env/spark_user}-\${cluster_name}@\${realm}"/>
spark.history.kerberos.keytab	<input type="text" value="\${keytab_dir}/spark.headless.keytab"/>

Leave all Defaults

Configure Identities

Configure principal name and keytab location for service users and hadoop service components.

General

Advanced

▸ Ambari Metrics

▸ HDFS

▸ Hive

▸ MapReduce2

▸ Oozie

▸ Spark

▸ Tez

▸ YARN

▸ ZooKeeper

✓ All configurations have been addressed.

10. Confirm your configuration. You can optionally download a CSV file of the principals and keytabs that Ambari will automatically create.

Confirm Configuration

Please review the configuration before continuing the setup process

Using the **Download CSV button**, you can download a csv file which contains a list of the principals and keytabs that will automatically be created by Ambari.

Executable path: /usr/bin, /usr/kerberos/bin, /usr/sbin, /usr/lib/mit/bin, /usr/lib/mit/sbin

KDC Host: impetus-I0163.impetus.co.in

KDC Type: Existing MIT KDC

Realm Name: IMPETUS.CO.IN

← Back

Exit Wizard

Download CSV

Next →

11. Click **Next** to start the process.

Enable Kerberos Wizard

ENABLE KERBEROS WIZARD

Get Started

Configure Kerberos

Install and Test Kerberos Client

Configure Identities

Confirm Configuration

Stop Services

Kerberize Cluster

Start and Test Services

Stop Services

Services have been successfully stopped.

✓ Stop Services

← Back

Next →

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.

Enable Kerberos Wizard

ENABLE KERBEROS WIZARD

- Get Started
- Configure Kerberos
- Install and Test Kerberos Client
- Configure Identities
- Confirm Configuration
- Stop Services
- Kerberize Cluster**
- Start and Test Services

Kerberize Cluster

Kerberos has successfully been enabled on the cluster.

- ✓ Preparing Operations
- ✓ Create Principals
- ✓ Create Keytabs
- ✓ Distribute Keytabs
- ✓ Update Configurations
- ✓ Finalize Operations

← Back

Note : If your cluster includes Storm, after enabling Kerberos, you must also [Set Up Ambari for Kerberos](#) for Storm Service Summary information to be displayed in Ambari Web. Otherwise, you will see n/a for Storm information such as Slots, Tasks, Executors and Topologies.

Enable Kerberos Wizard

ENABLE KERBEROS WIZARD

- Get Started
- Configure Kerberos
- Install and Test Kerberos Client
- Configure Identities
- Confirm Configuration
- Stop Services
- Kerberize Cluster
- Start and Test Services**

Start and Test Services

Services have been successfully tested with kerberos environment.

✓ Start and Test Services

Complete

Final status should like below.

Kerberos security is enabled

Disable Kerberos

Regenerate Keytabs

Edit

GeneralAdvanced

Global

Keytab Dir

/etc/security/keytabs

Realm

IMPETUS.CO.IN

Additional Realms

(Optional)

Spnego Principal

HTTP/_HOST@\${realm}

Spnego Keytab

\${keytab_dir}/spnego.service.keytab

Ambari Principals

Smokeuser Principal Name

\${cluster-env/smokeuser}-\${cluster_name}@\${realm}

Smokeuser Keytab

\${keytab_dir}/smokeuser.headless.keytab

HDFS user principal

\${hadoop-env/hdfs_user}-\${cluster_name}@\${realm}

Path to HDFS user keytab file

\${keytab_dir}/hdfs.headless.keytab

spark.history.kerberos.principal

\${spark-env/spark_user}-\${cluster_name}@\${realm}

spark.history.kerberos.keytab

\${keytab_dir}/spark.headless.keytab

✔ All configurations have been addressed.