Puneetha B M
Knowledge is an ocean…

Configure Hadoop Security with Cloudera Manager 5 or later – using Kerberos

By puneetha | August 23, 2014 4 Comments

If you are using Cloudera Manager version less than 5. Check out the other blog here

Kerberos is a network authentication protocol created by MIT, and uses symmetric-key cryptography to authenticate users to network services, which means passwords are never actually sent over the network.Rather than authenticating each user to each network service separately as with simple password authentication, Kerberos uses symmetric encryption and a trusted third party (a **key distribution center or KDC**) to authenticate users to a suite of network services. The computers managed by that KDC and any secondary KDCs constitute a realm.
When a user authenticates to the KDC, the KDC sends a set of credentials (a ticket) specific to that session back to the user's machine, and any Kerberos-aware services look for the ticket on the user's machine rather than requiring the user to authenticate using a password.To enable Security in Hadoop, we integrate Kerberos Authentication.

If you want to know more about Kerberos. Check out this link: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Managing_Smart_Cards/Using_Kerberos.html

For this example, lets say our cluster has 3 nodes, which is managed by cloudera Manager
host1.example.com (or) host1 –> Kerberos Server & Client (KDC) You can make a remote node as server
host2.example.com (or) host2 –> Kerberos Client
host3.example.com (or) host3 –> Kerberos Client
Our realm name > PUNEETHA.COM

Cloudera Manager version 5 or later

Pre-requisite:
You have a Hadoop Cluster managed by Cloudera Manager. If you dont have one, check this link to create a cluster managed by cloudera manager –> http://blog.puneethabm.in/hadoop-cluster-set-up-cloudera/

Note: Stop All services
Step 1:
To install packages for a Kerberos server:

```
# yum -y install krb5-server krb5-libs krb5-auth-dialog krb5-workstation
```

To install packages for a Kerberos client:

```
# yum -y install krb5-workstation krb5-libs krb5-auth-dialog
```

Step 2:

Server:

–> Change Realm Name > PUNEETHA.COM

–> Add parameters > max_life = 1d and max_renewable_life = 7d

```
# vim /var/kerberos/krb5kdc/kdc.conf
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
  PUNEETHA.COM = {
  #master_key_type = aes256-cts
  acl_file = /var/kerberos/krb5kdc/kadm5.acl
  dict_file = /usr/share/dict/words
  admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
  supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal arcfour-hmac:nor
  max_life = 1d
  max_renewable_life = 7d
 }
```

Step 3:

Add below properties in All Clients:

> udp_preference_limit = 1

> default_tgs_enctypes = arcfour-hmac

> default_tkt_enctypes = arcfour-hmac

```
# vim /etc/krb5.conf
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = PUNEETHA.COM
 dns_lookup_realm = false
 dns_lookup_kdc = false
```

```
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 udp_preference_limit = 1
 default_tgs_enctypes = arcfour-hmac
 default_tkt_enctypes = arcfour-hmac

[realms]
  PUNEETHA.COM = {
  kdc = host1.example.com
  admin_server = host1.example.com
 }

[domain_realm]
   .example.com = PUNEETHA.COM
   example.com = PUNEETHA.COM
```

Step 4:

Create the database using the kdb5_util utility. (Server)

```
# /usr/sbin/kdb5_util create -s
Loading random data
Initializing database '/var/kerberos/krb5kdc/principal' for realm 'PUNEETHA.COM',
master key name 'K/M@PUNEETHA.COM'
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:
```

Step 5:

In Server, add cloudera-scm principal, it will be used by Cloudera Manager later to manage Hadoop principals.

```
# kadmin.local
kadmin.local:  addprinc cloudera-scm@PUNEETHA.COM
WARNING: no policy specified for cloudera-scm@PUNEETHA.COM; defaulting to no policy
Enter password for principal "cloudera-scm@PUNEETHA.COM":
Re-enter password for principal "cloudera-scm@PUNEETHA.COM":
Principal "cloudera-scm@PUNEETHA.COM" created.
```

Step 6:

Add */admin and cloudera-scm to ACL(Access Control List), which gives privilege to add principals for admin and cloudera-scm principal

```
# vim /var/kerberos/krb5kdc/kadm5.acl
*/admin@PUNEETHA.COM *
cloudera-scm@PUNEETHA.COM admilc
```

Step 7:

Adds the password policy to the database.

```
# kadmin.local
kadmin.local:  addpol admin
kadmin.local:  addpol users
kadmin.local:  addpol hosts
kadmin.local:  exit
```

Step 8:

Start Kerberos using the following commands:

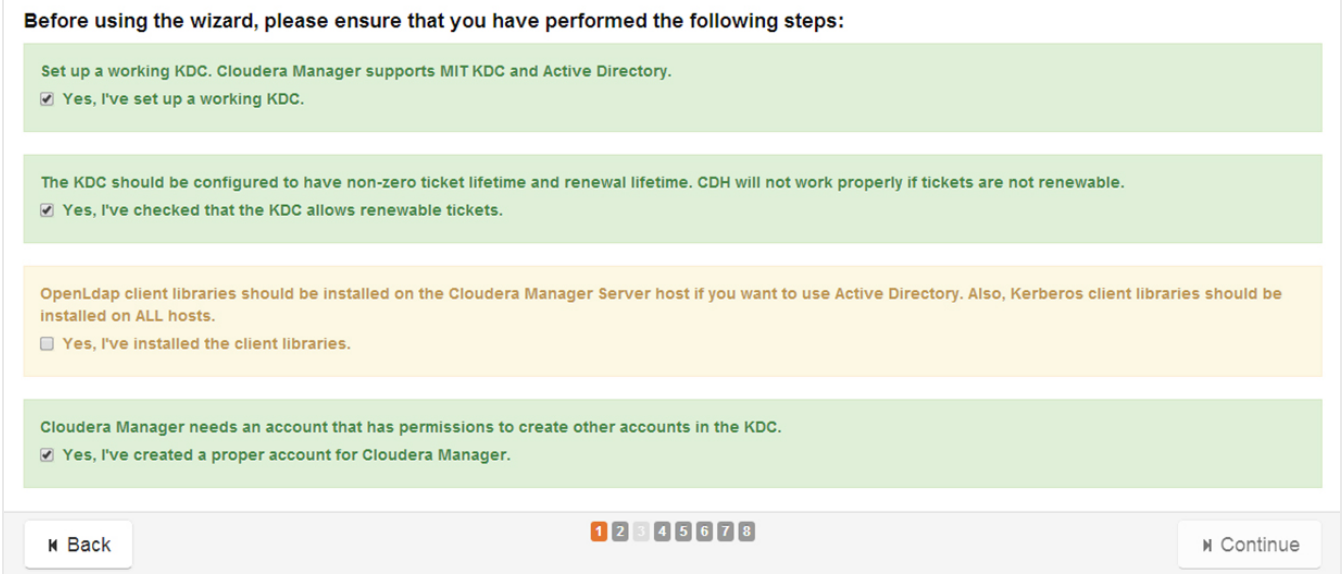
```
#service krb5kdc start
#service kadmin start
```

Step 9:

Now go to Cloudera Manager UI:

Administration > Kerberos > Enable Kerberos

Check all the boxes:



Choose the below parameters:

KDC Type: MIT KDC

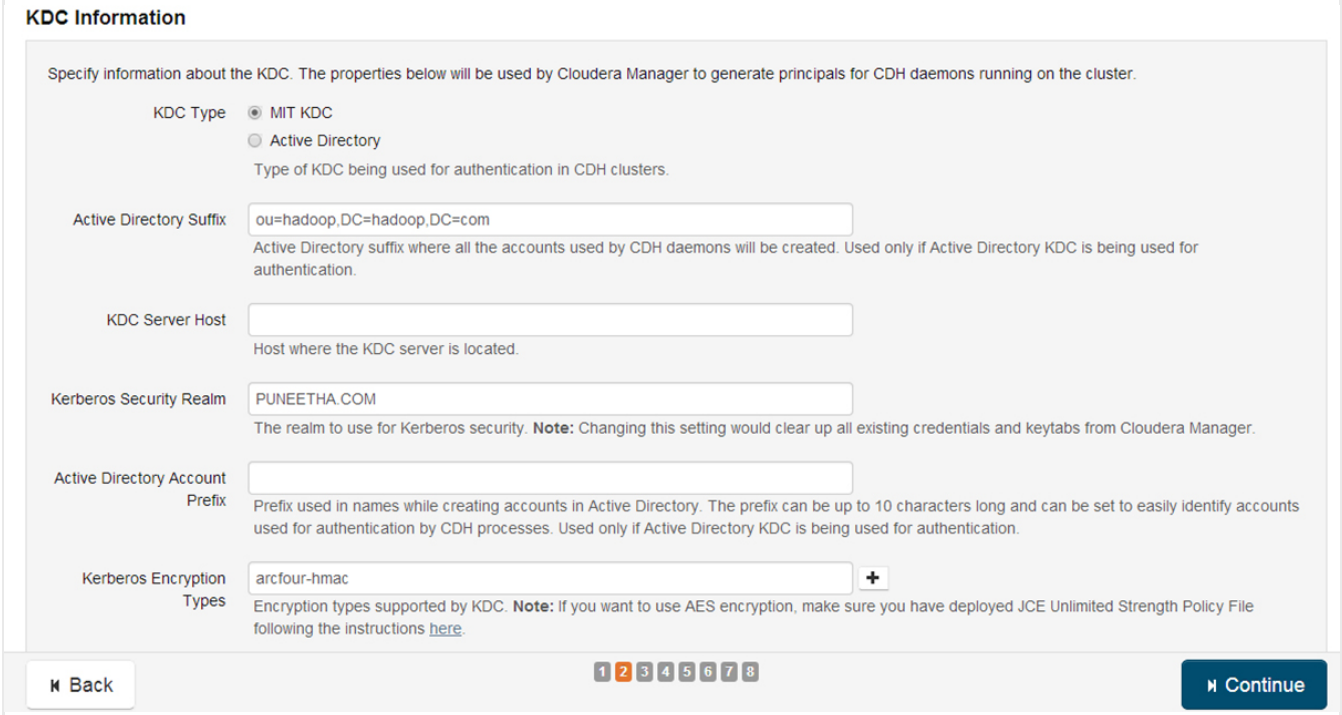
KDC Server Host: host1.example.com

Kerberos Security Realm: PUNEETHA.COM

Kerberos Encryption Types: **arcfour-hmac**

Manage krb5.conf through Cloudera Manager: Check the box

Maximum Renewable Life for Principals: 7 days

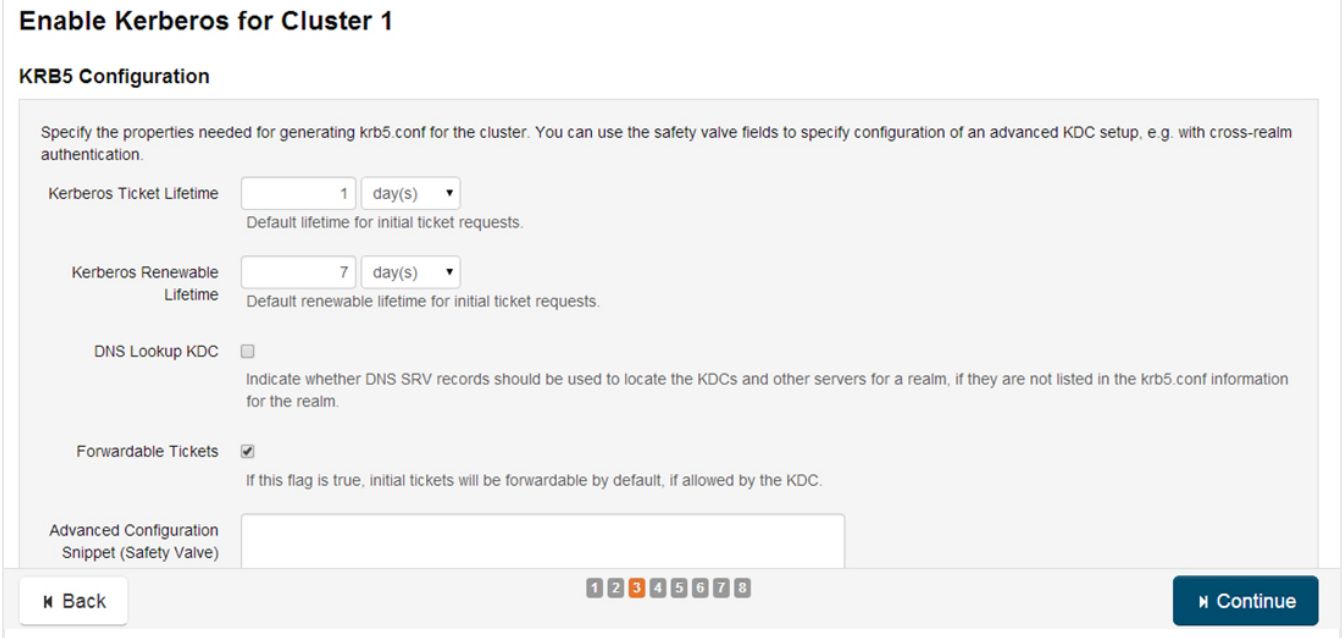


Choose the below parameters:

Kerberos Ticket Lifetime: 1 days

Kerberos Renewable Lifetime: 7 days

Forwardable Tickets: check the box



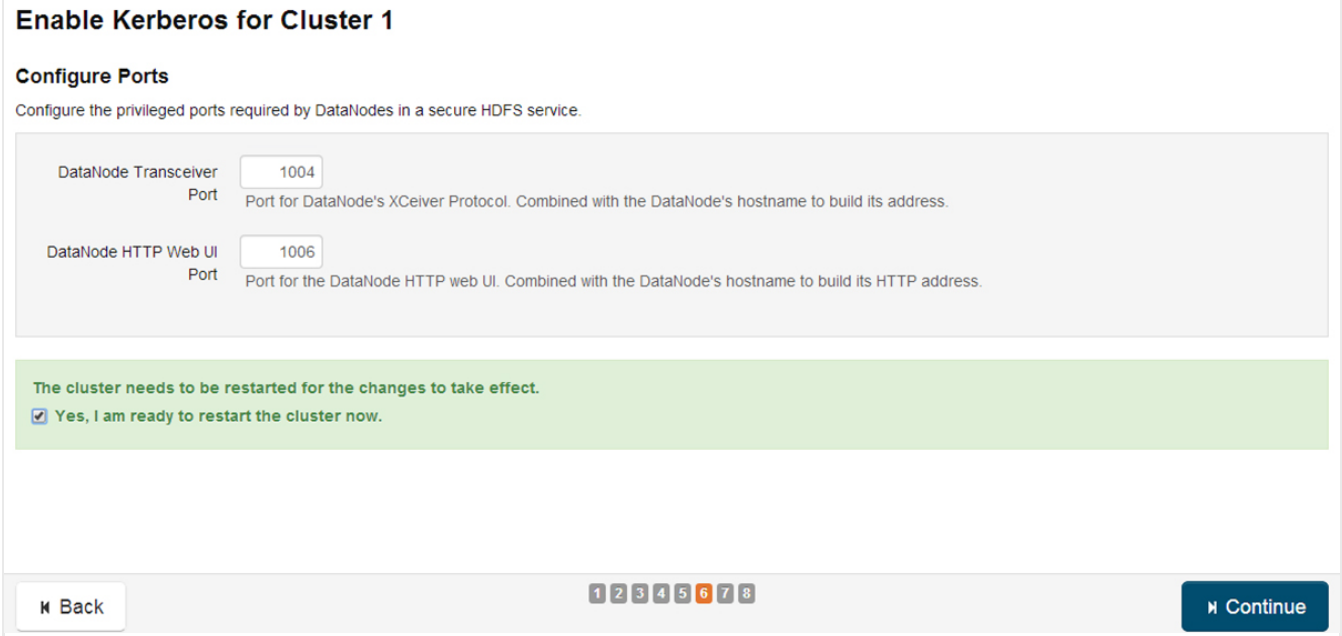
Provide credentials of cloudera-scm principal, which we created before

KDC Account Manager Credentials:

Username: cloudera-scm@PUNEETHA.COM

Password:

Check the box to restart the cluster, this will deploy kerberos configuration across cluster



Note: Configure Security for only those services which you have on your cluster as below:

Hue Security:

```
Hue Service -> Add -> Instances -> Assign the Kerberos Ticket Renewer role instance to the same h
```

Hive Security:

Hive Service -> Configuration -> Service-wide -> Advanced -> Hive Service Configuration Safety Valve for hive-site.xml

Add the below 3 property tags there:

```
<property>
  <name>hive.server2.authentication</name>
  <value>KERBEROS</value>
</property>
<property>
  <name>hive.server2.authentication.kerberos.principal</name>
  <value>hive/_HOST@PUNEETHA.COM</value>
</property>
<property>
 <name>hive.stats.ndv.error</name>
 <value>5.0</value>
</property>
```

You have a Kerberized Cluster now 🙂

Comment below if you find this blog useful.

Few more useful things. (FYI)

Lets go one step ahead, now that we have a kerberized cluster, users wont be able to access the cluster by the command 'hadoop fs -ls '

He has to be a kerberos user. Only hdfs user can add users to the cluster Ex: hadoop fs -mkdir /user/puneetha

Generate a keytab for hdfs principal

If we want to use the keytab from the node host2, then we generate hdfs keytab for the node2 principal as below:

```
#kadmin.local
kadmin.local: xst -norandkey -k hdfs.keytab hdfs/host2.example.com@PUNEETHA.COM HTTP/host2.exampl

(OR)
kadmin.local: addprinc hdfs@PUNEETHA.COM
kadmin.local: exit
```

```
If you have hdfs keytab file >> $kinit hdfs -k -t /unix-path/hdfs.keytab
If you are hdfs user >> $kinit hdfs
```

Create Kerberos user

Ex: I want to create a kerberos user called 'puneetha'

Add user 'puneetha' to all nodes (user puneetha should be present in hadoop nodes, I am talking about UNIX shell)

In all nodes of the cluster:

```
#useradd puneetha -u 1000
```

Generate UNIX password for the user

```
#passwd puneetha
```

Create hdfs user 'puneetha' using hdfs.keytab

```
$ kinit hdfs -k -t /unix-path/hdfs.keytab
$ hadoop fs -mkdir /user/puneetha
$ hadoop fs -chown puneetha:puneetha /user/puneetha
```

In Kerberos, add principal for the user 'puneetha'

```
#kadmin.local
kadmin.local: addprinc puneetha@PUNEETHA.COM
kadmin.local:  exit
```

To access the cluster, you need to issue kinit command and obtain a ticket.

```
$kinit  puneetha@PUNEETHA.COM
(OR)
$kinit  puneetha
```

and start accessing the hadoop cluster

Ex: `$ hadoop fs -ls /user/puneetha`

Other commands:

To list all principals:

```
#kadmin.local
kadmin.local: getprincs
kadmin.local: exit
```

To provide password for the principal while creating principal:

```
#kadmin.local
kadmin.local: addprinc -pw <Password> puneetha@PUNEETHA.COM
kadmin.local: exit
```

To add user from command line:

```
# kadmin.local -q "addprinc dummyuser"
```

To enter Impala shell

```
$impala-shell -k
```

To refresh metadata while entering Impala shell

```
$impala-shell -k -r
```

Category: Hadoop Security - Kerberos Tags: cloudear kerberos, hadoop kerberos, kerberos

4 thoughts on “Configure Hadoop Security with Cloudera Manager 5 or later – using Kerberos”

Guna
January 2, 2016

Mate it is very useful . Thanks

Sanjay Bhatnagar
January 13, 2016

Good Job Puneetha!! Very helpful and clear.
Sanjay.

Gregory Grubbs
April 26, 2016

Just wanted to let you know this post was immensely helpful as I set up my first KDC and got it working with Cloudera CDH.

Thank you!

Vaidya
June 7, 2016

Hi Puneetha,
Requirement: Setup Local KDC.

I performed all the steps sequentially till Step 9. I can login kadmin.local using cloudera-scm . But when I login as kinit test, it gives an error “KDC has no support for encryption type”.

And I configured using Cloudera Manager, the error is “kinit(v5): KDC has no support for encryption type while getting initial credentials”.

I configured as per your document and uisng rc4-hmac / arcfour-hmac as encryption type.

Please advise me