|  |
| --- |
|  |
| Enable Kerberos on Cloudera |
| Kerberos configuration on centos 7 |

|  |
| --- |
| Shalaj Shukla  5/3/2017 |

Contents

[Introduction 2](#_Toc481740716)

[Configuration Setup for Kerberos 3](#_Toc481740717)

[Set host name 3](#_Toc481740718)

[Install the JCE Policy File 3](#_Toc481740719)

[Install Kerberos 3](#_Toc481740720)

[Configure Kerberos 4](#_Toc481740721)

[Create kerberos database 5](#_Toc481740722)

[Add principal for Cloudera-scm 6](#_Toc481740723)

[Enable Kerberos in Cloudera 7](#_Toc481740724)

[Give permission to root user 18](#_Toc481740725)

[Add hdfs principal 19](#_Toc481740726)

[Create keytab file 19](#_Toc481740727)

[Add Ticket to hdfs User 20](#_Toc481740728)

[Switch to hdfs User 21](#_Toc481740729)

[Add new user to filesystem 22](#_Toc481740730)

[Check hive shell 25](#_Toc481740731)

[Check beeline 25](#_Toc481740732)

[Check mapreduce job 27](#_Toc481740733)

[Access hdfs using client machine 29](#_Toc481740734)

[Enable Kerberos Authentication for HTTP Web Console 31](#_Toc481740735)

[Issue Resolution 34](#_Toc481740736)

[Minimum User Id issue 34](#_Toc481740737)

# Introduction

Kerberos is a way of authenticating users that was developed at MIT and has grown to become the most widely used authentication approach. Hadoop requires kerberos to be secure because in the default authentication Hadoop and all machines in the cluster believe every user credentials presented. To overcome this vulnerability kerberos provides a way of verifying the identity of users. Kerberos identity verification is implemented through a client/server model.

There are several terminologies that are used when implementing kerberos identity verification. An identity that needs to be verified is referred to as a **principal**. Principals are divided into two categories vit user principals and service principals. User principal names (UPN) are used to refer to users, these users are similar to users in an operating system. Service principal names (SPN) refer to services accessed by a user such as a database. A **realm** in kerberos refers to an authentication administrative domain. Principals are assigned to specific realms in order to demarcate boundaries and simplify administration.

Information on principals and realms resides in a **key distribution center** (KDC). Therefore it is very important to put in place physical and network security measures to protect KDC because if it is compromised the entire realm is compromised. KDC can be logically considered divided into three parts the **kerberos database**, **Authentication Server** (AS) and **Ticket Granting server** (TGS). The kerberos database is the repository of all principals and realms.

The **Authentication Server** is the part of the KDC which replies to the initial authentication request from the client, when the user, not yet authenticated, must enter the password. In response to an authentication request, the AS issues a special ticket known as the **Ticket Granting Ticket** (TGT).

The **Ticket Granting Server** is the KDC component which distributes service tickets to clients with a valid TGT, guaranteeing the authenticity of the identity for obtaining the requested resource on the application servers

# Configuration Setup for Kerberos

## Set host name

|  |
| --- |
| [root@mac127 ~]# hostnamectl set-hostname mac127  [root@mac127 ~]# hostname  mac127 |

## Install the JCE Policy File

If you are using CentOS/Red Hat Enterprise Linux 5.6 or later, or Ubuntu, which use AES-256 encryption by default for tickets, you must install the **Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy File** on all cluster and Hadoop user machines

Download UnlimitedJCEPolicyJDK7.zip file from <http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>

This file is for jdk7 you can choose JCE file as per JDK release

Make copy of original files

|  |
| --- |
| cp /usr/java/jdk1.7.0\_67-cloudera/jre/lib/security/local\_policy.jar local\_policy.jar.orig  cp /usr/java/jdk1.7.0\_67-cloudera/jre/lib/security/US\_export\_policy.jar US\_export\_policy.jar.orig |

|  |
| --- |
| # copy the new jars into place  cp /root/kerberos/UnlimitedJCEPolicy/local\_policy.jar /usr/java/jdk1.7.0\_67-cloudera/jre/lib/security/local\_policy.jar  cp /root/kerberos/UnlimitedJCEPolicy/US\_export\_policy.jar /usr/java/jdk1.7.0\_67-cloudera/jre/lib/security/US\_export\_policy.jar |

## Install Kerberos

Install Kerberos server, work station and open LDAP client

|  |
| --- |
| **yum install -y krb5-server krb5-workstation openldap** |

## Configure Kerberos

Make the changes as shown below in different files

Here we are configuring CYBAGE.COM as realm

|  |
| --- |
| [root@mac127 ~]# **cd /var/kerberos/krb5kdc/**  [root@mac127 krb5kdc]# ls  kadm5.acl kdc.conf  [root@mac127 krb5kdc]# **vi kadm5.acl** |

Replace EXAMPLE.COM to CYBAGE.COM

|  |  |
| --- | --- |
| **Original file** | **Updated file** |
| \*/admin@EXAMPLE.COM \* | \*/admin@CYBAGE.COM \* |

|  |
| --- |
| [root@mac127 krb5kdc]# **vi kdc.conf** |

|  |  |
| --- | --- |
| **Original file** | **Updated file** |
| [kdcdefaults]  kdc\_ports = 88  kdc\_tcp\_ports = 88  [realms]  EXAMPLE.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:normal camellia256-cts:normal camellia128-cts:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal  } | [kdcdefaults]  kdc\_ports = 88  kdc\_tcp\_ports = 88  [realms]  CYBAGE.COM = {  #master\_key\_type = aes256-cts  acl\_file = /var/kerberos/krb5kdc/kadm5.acl  dict\_file = /usr/share/dict/words  max\_renewable\_life = 7d  max\_life = 1d  admin\_keytab = /var/kerberos/krb5kdc/kadm5.keytab  supported\_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal arcfour-hmac:normal camellia256-cts:normal camellia128-cts:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal  default\_principal\_flags = +renewable, +forwardable  } |

|  |
| --- |
| **vi /etc/krb5.conf** |

|  |  |
| --- | --- |
| **Default file** | **Updated file** |
| # Configuration snippets may be placed in this directory as well  includedir /etc/krb5.conf.d/  [logging]  default = FILE:/var/log/krb5libs.log  kdc = FILE:/var/log/krb5kdc.log  admin\_server = FILE:/var/log/kadmind.log  [libdefaults]  dns\_lookup\_realm = false  ticket\_lifetime = 24h  renew\_lifetime = 7d  forwardable = true  rdns = false  # default\_realm = EXAMPLE.COM  default\_ccache\_name = KEYRING:persistent:%{uid}  [realms]  # EXAMPLE.COM = {  # kdc = kerberos.example.com  # admin\_server = kerberos.example.com  # }  [domain\_realm]  # .example.com = EXAMPLE.COM  # example.com = EXAMPLE.COM | # Configuration snippets may be placed in this directory as well  includedir /etc/krb5.conf.d/  [logging]  default = FILE:/var/log/krb5libs.log  kdc = FILE:/var/log/krb5kdc.log  admin\_server = FILE:/var/log/kadmind.log  [libdefaults]  dns\_lookup\_realm = false  ticket\_lifetime = 24h  renew\_lifetime = 7d  forwardable = true  rdns = false  default\_realm = CYBAGE.COM  default\_ccache\_name = KEYRING:persistent:%{uid}  [realms]  CYBAGE.COM = {  kdc = mac127  admin\_server = mac127  }  [domain\_realm]  .cybage.com = CYBAGE.COM  cybagee.com = CYBAGE.COM |

## Create kerberos database

|  |
| --- |
| [root@mac127 krb5kdc]# **kdb5\_util create -s**  Loading random data  Initializing database '/var/kerberos/krb5kdc/principal' for realm 'CYBAGE.COM',  master key name 'K/M@CYBAGE.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: |

Password – cloudera

**Start and enable services**

|  |
| --- |
| [root@mac127 krb5kdc]# **systemctl start krb5kdc kadmin**  [root@mac127 krb5kdc]# **systemctl enable krb5kdc kadmin**  Created symlink from /etc/systemd/system/multi-user.target.wants/krb5kdc.service to /usr/lib/systemd/system/krb5kdc.service.  Created symlink from /etc/systemd/system/multi-user.target.wants/kadmin.service to /usr/lib/systemd/system/kadmin.service. |

|  |
| --- |
| [root@mac127 krb5kdc]# **kadmin.local**  Authenticating as principal root/admin@CYBAGE.COM with password.  kadmin.local: **listprincs**  K/M@CYBAGE.COM  kadmin/admin@CYBAGE.COM  kadmin/changepw@CYBAGE.COM  kadmin/mac127.cybage.com@CYBAGE.COM  kiprop/mac127.cybage.com@CYBAGE.COM  krbtgt/CYBAGE.COM@CYBAGE.COM |

## Add principal for Cloudera-scm

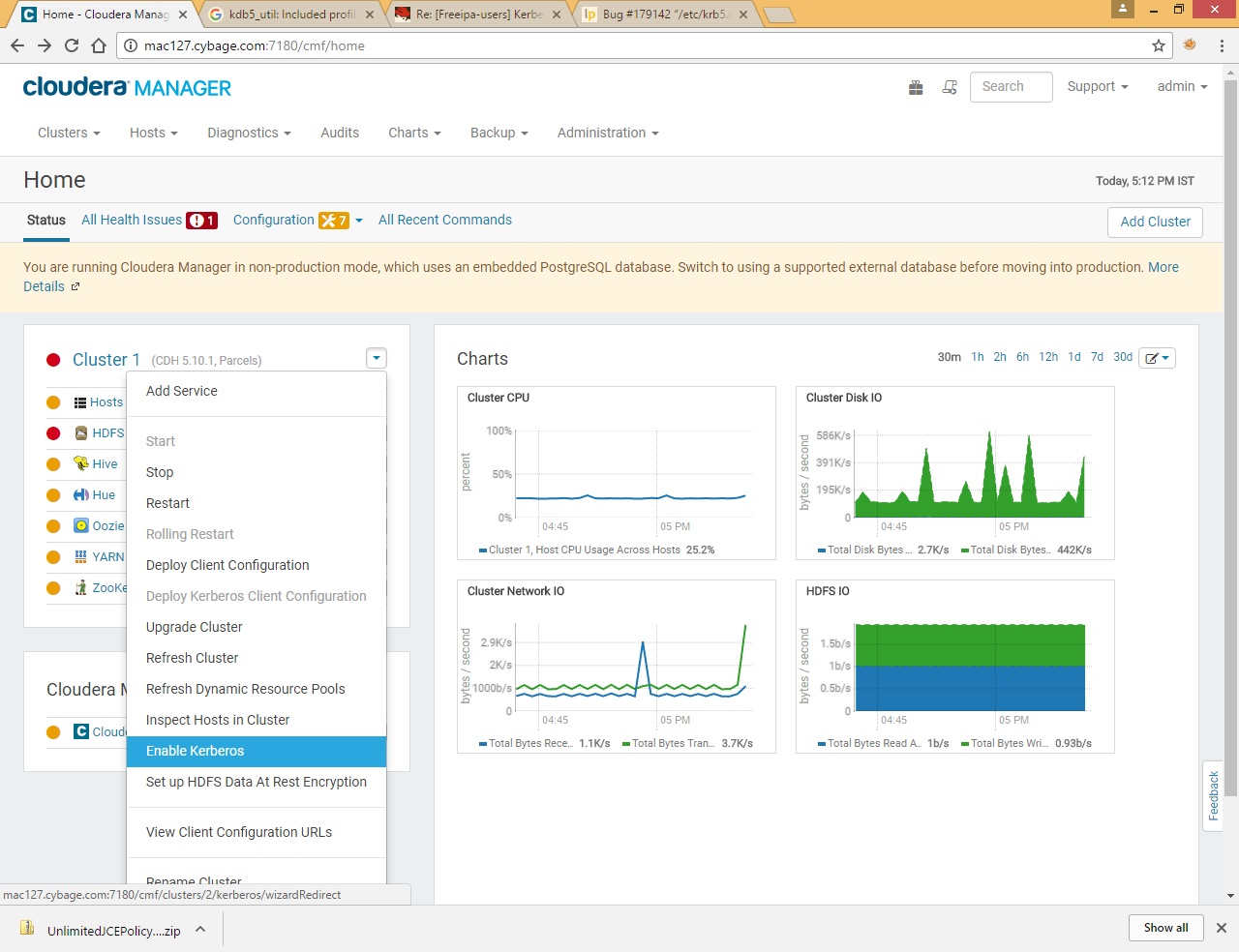
In Server, add cloudera-scm principal, it will be used by Cloudera Manager later to manage Hadoop principals.

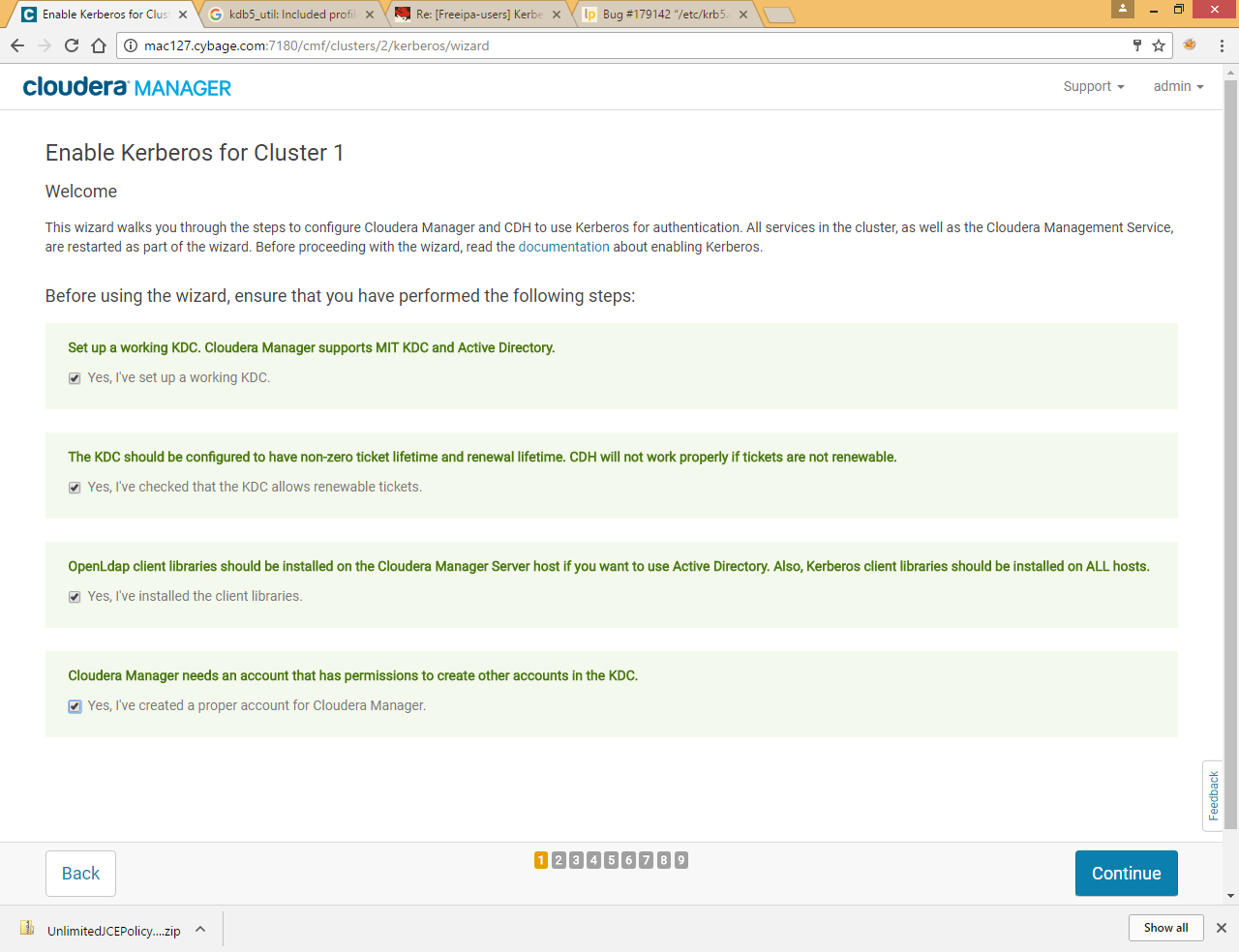
|  |
| --- |
| kadmin.local: **addprinc cloudera-scm/admin**  WARNING: no policy specified for cloudera-scm/admin@CYBAGE.COM; defaulting to no policy  Enter password for principal "cloudera-scm/admin@CYBAGE.COM":  Re-enter password for principal "cloudera-scm/admin@CYBAGE.COM":  Principal "cloudera-scm/admin@CYBAGE.COM" created.  kadmin.local: |

Password- cloudera

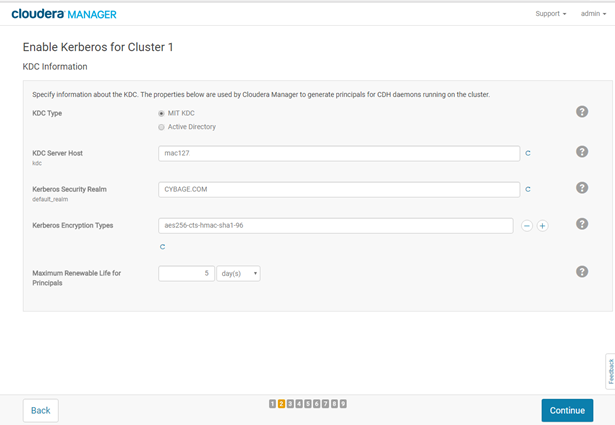
# Enable Kerberos in Cloudera

Go to Cluster and click on Enable Kerberos from drop down menu





Tick all check boxes as we did all configuration and click on continue

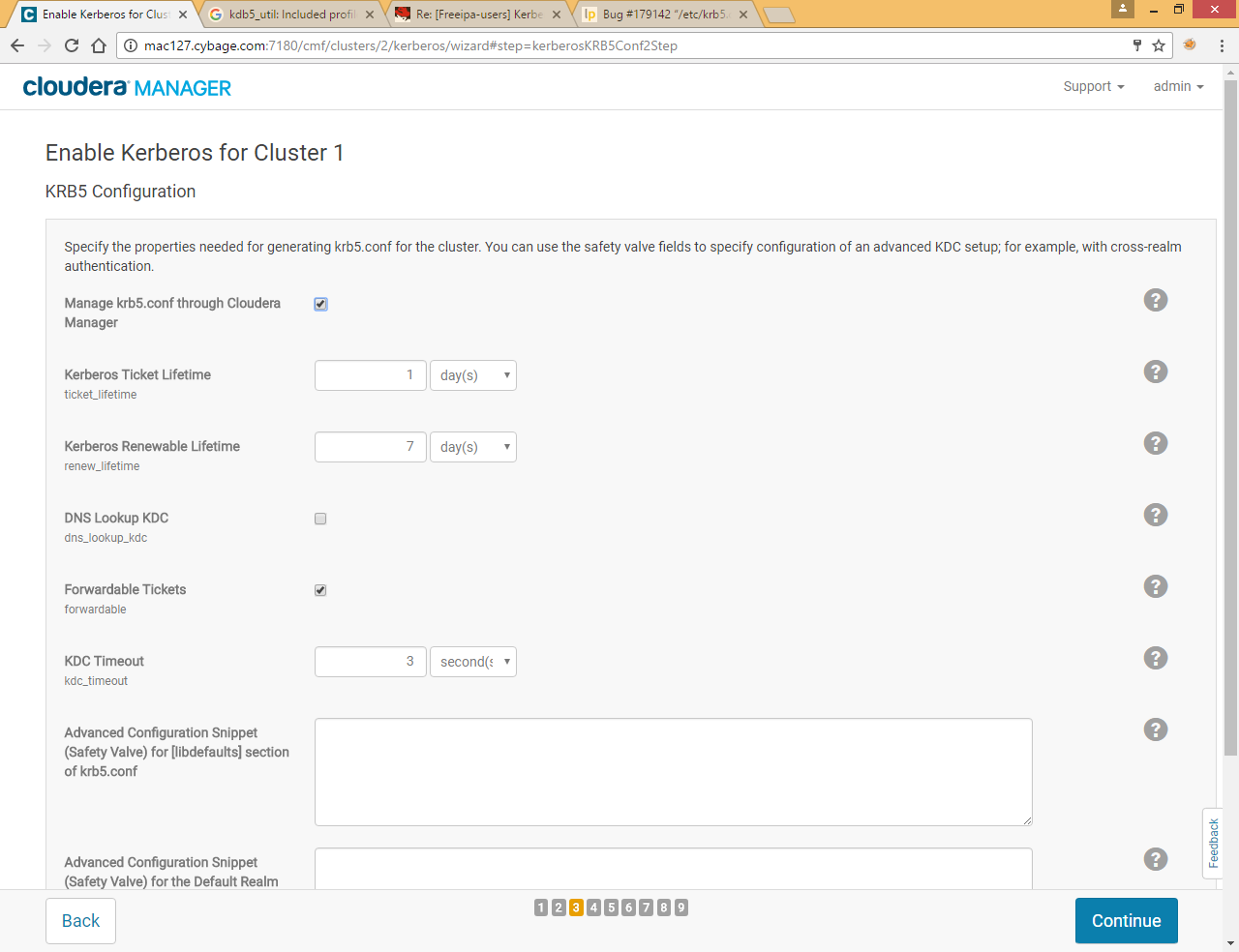


KDC Server Host – mac127

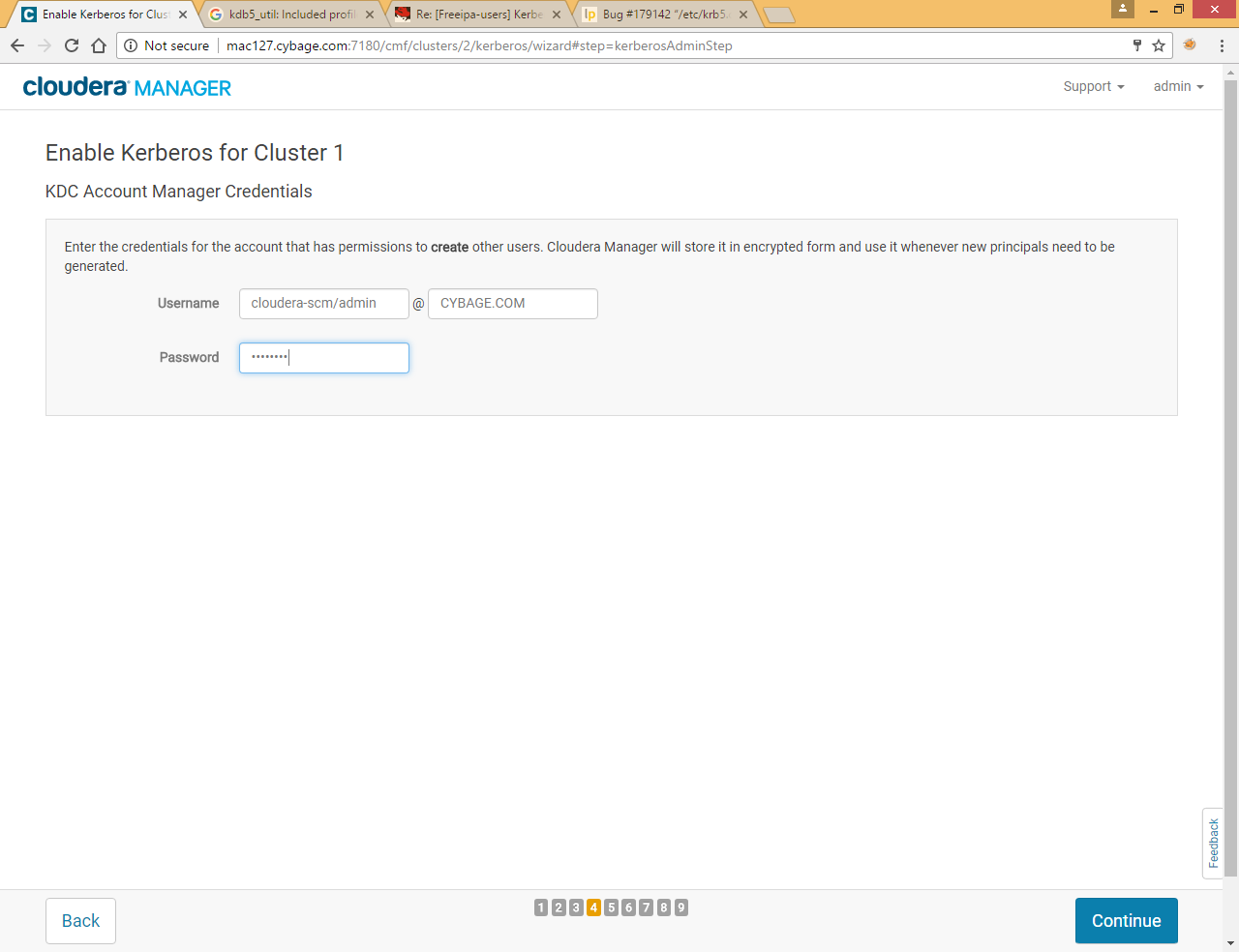
Kerberos Security Realm – CYBAGE.COM

Kerberos Encryption Types- aes256-cts-hmac-sha1-96

Click Continue



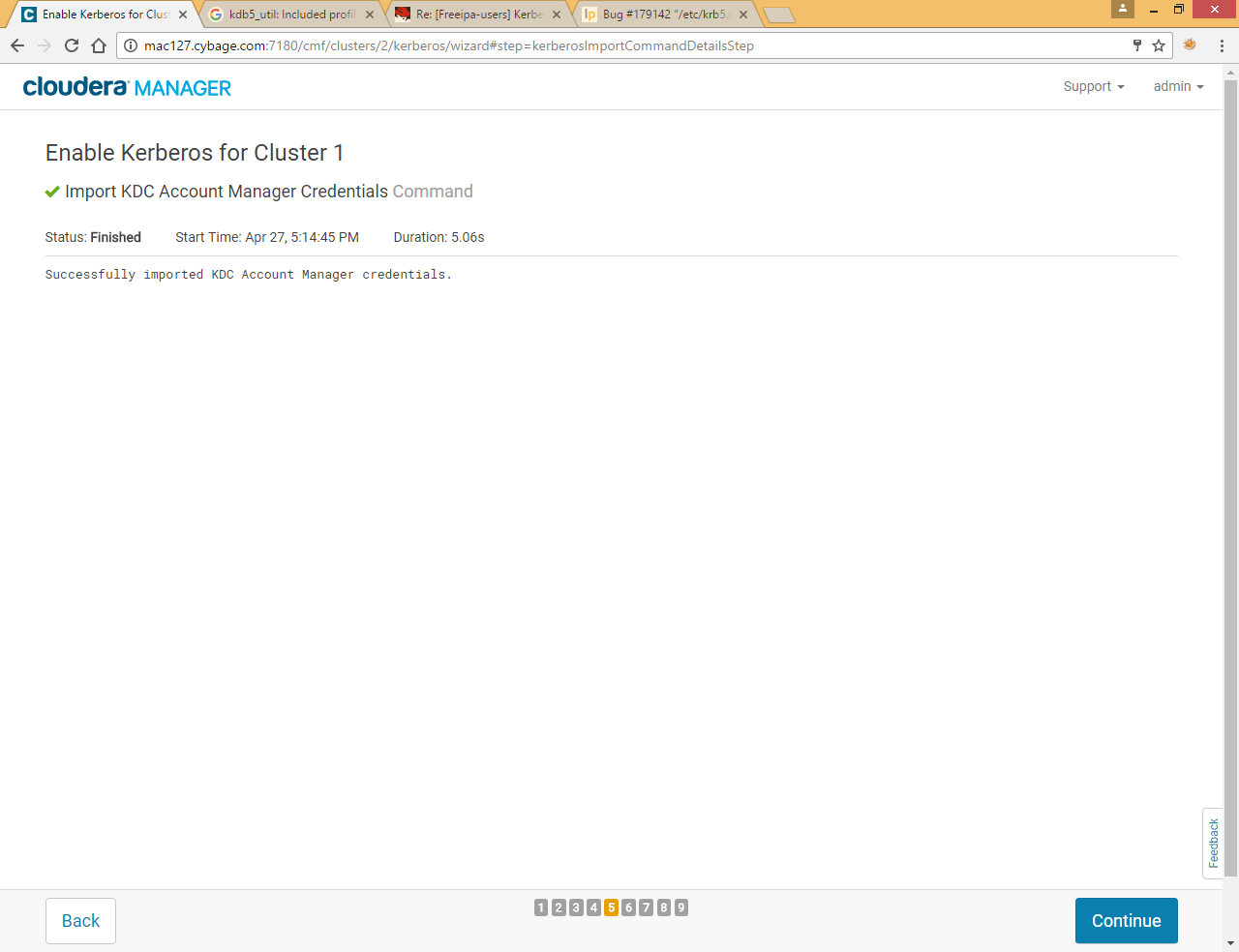
Select Manage Krb5.conf through Cloudera and Click Continue



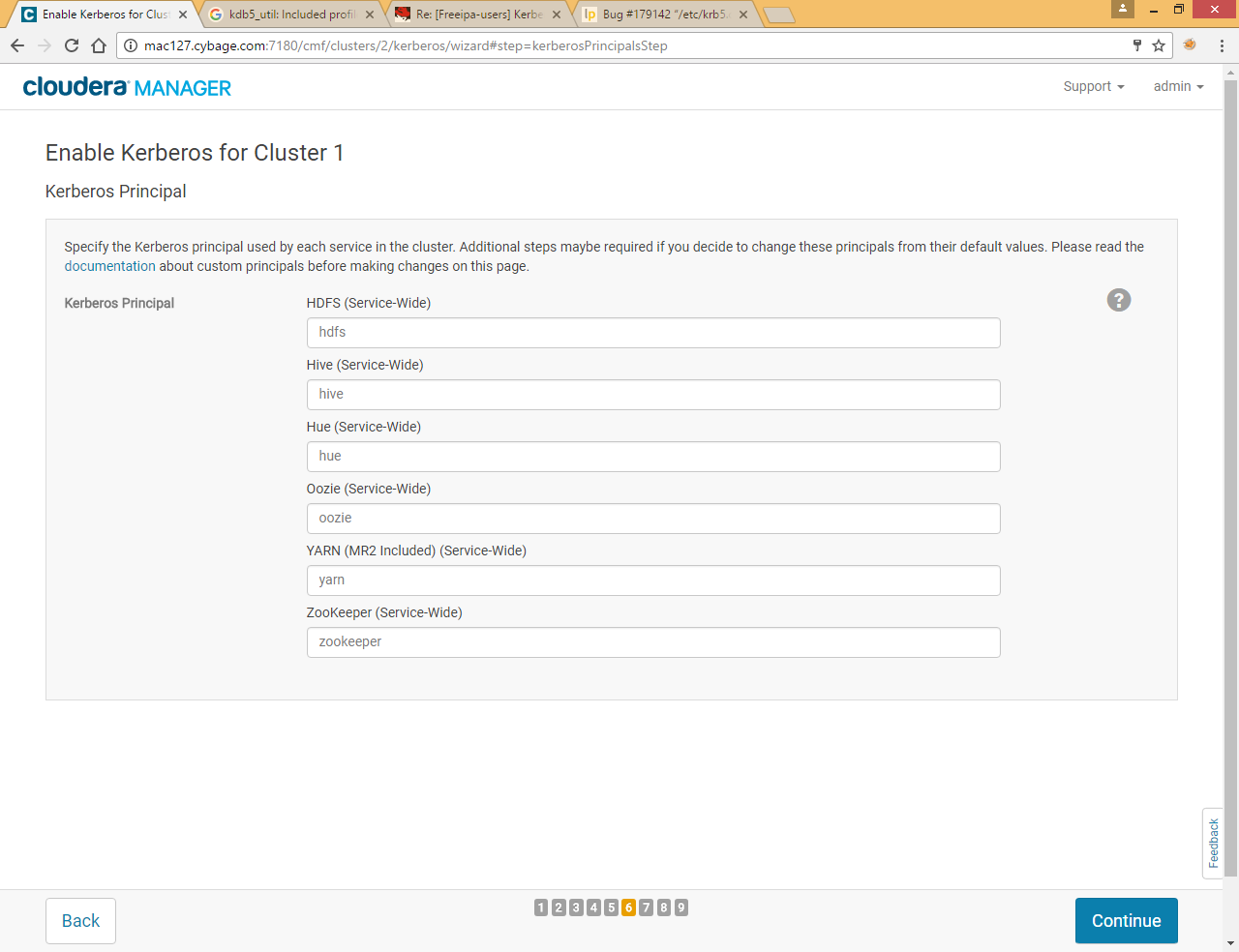
User name - cloudera-scm/admin,

Password – cloudera (as we set during configuration steps)

Click continue

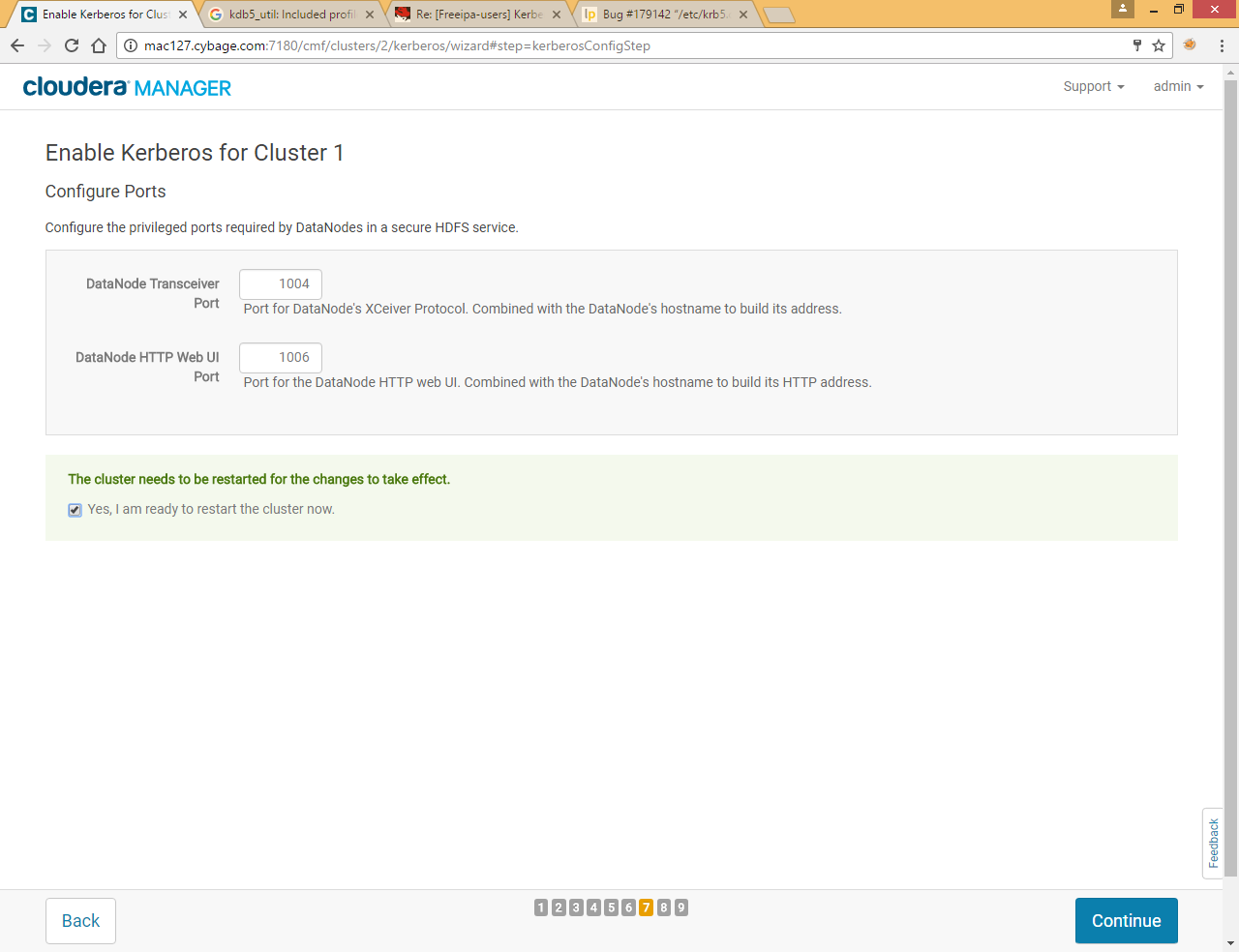


Click continue

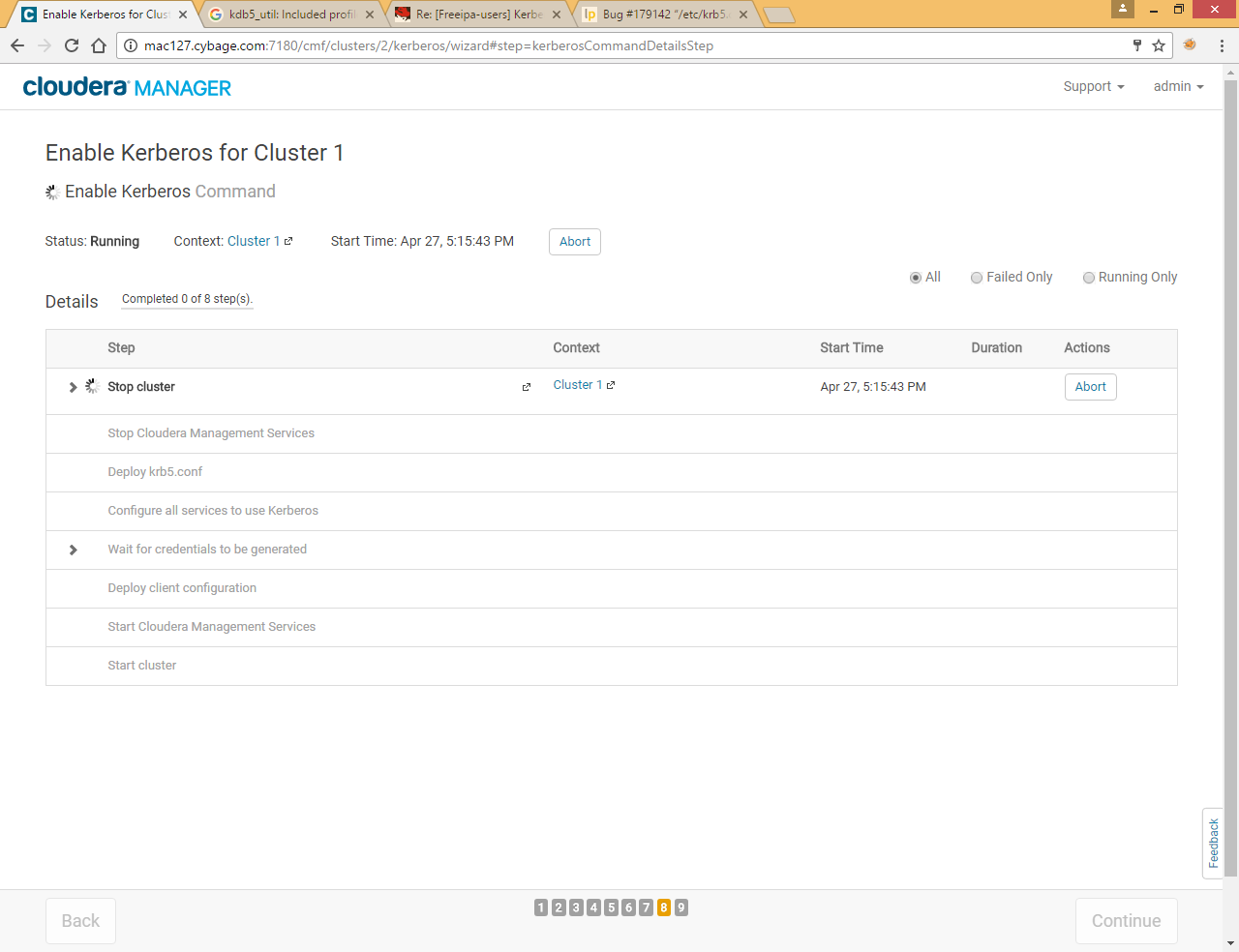


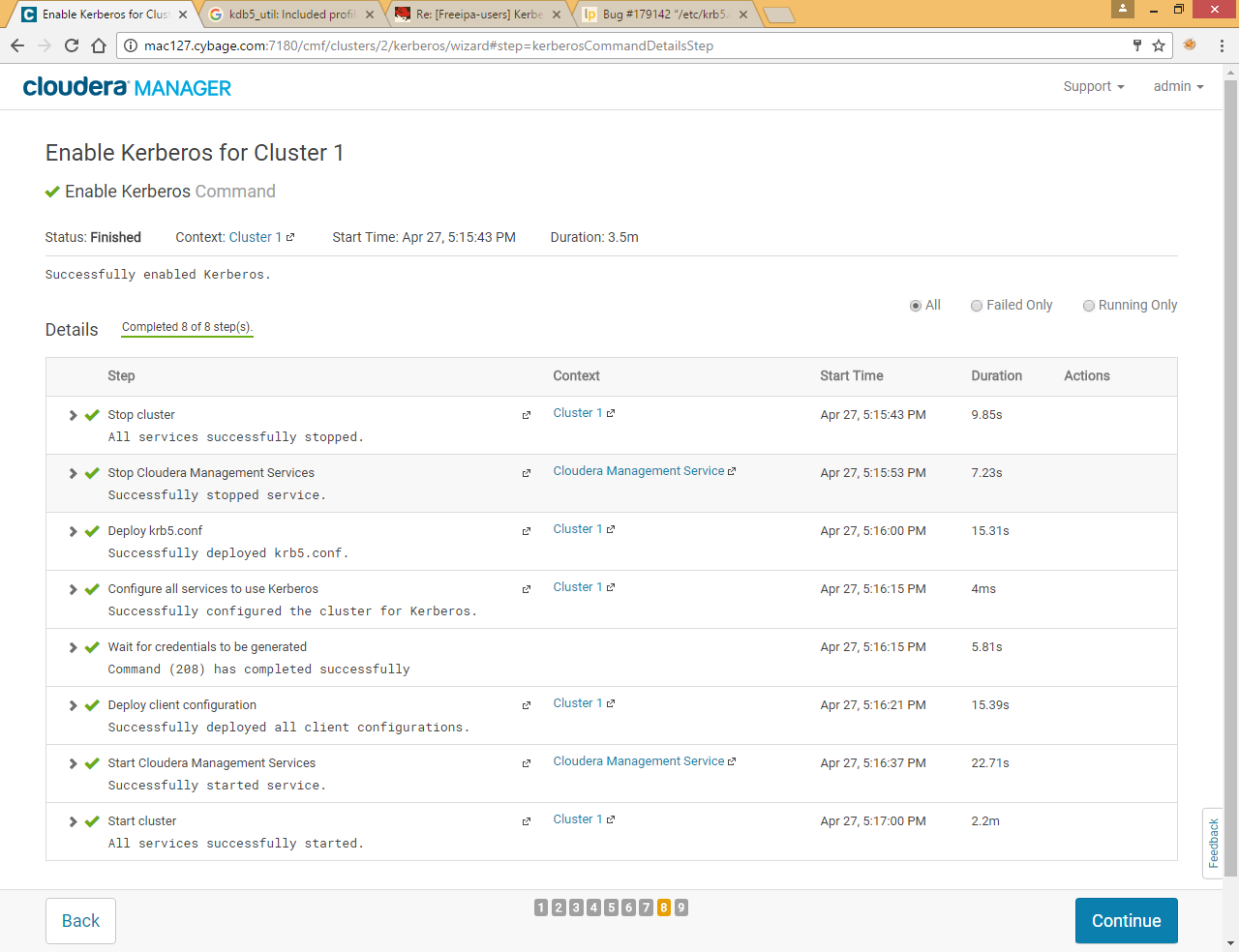
Cloudera will create principals for all deployed services

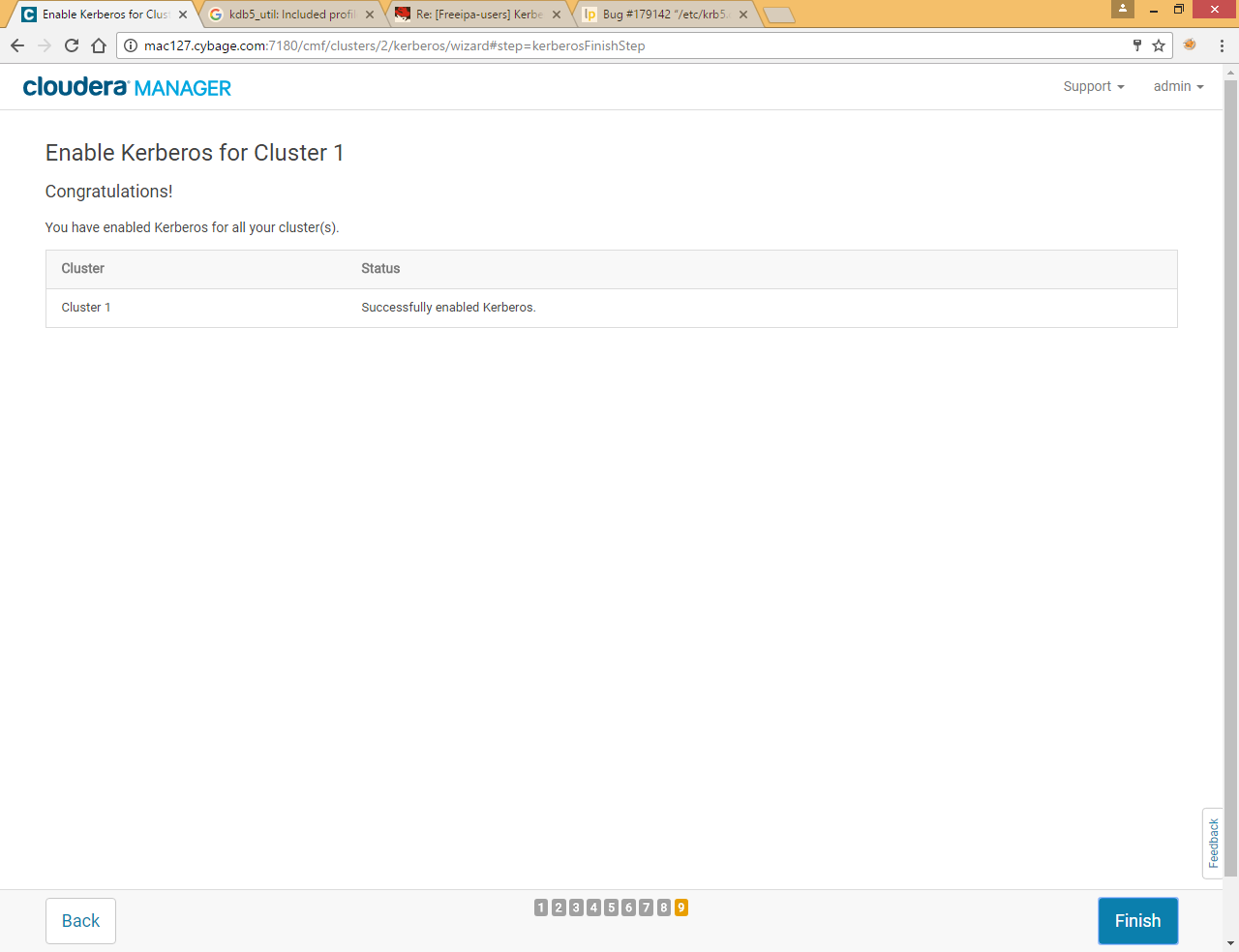
Click Continue



Select check box and click continue







After successful configuration

krb5.conf file turned into

|  |
| --- |
| [libdefaults]  default\_realm = CYBAGE.COM  dns\_lookup\_kdc = false  dns\_lookup\_realm = false  ticket\_lifetime = 86400  renew\_lifetime = 604800  forwardable = true  default\_tgs\_enctypes = aes256-cts-hmac-sha1-96  default\_tkt\_enctypes = aes256-cts-hmac-sha1-96  permitted\_enctypes = aes256-cts-hmac-sha1-96  udp\_preference\_limit = 1  kdc\_timeout = 3000  [realms]  CYBAGE.COM = {  kdc = mac127  admin\_server = mac127  } |

Try to use hdfs dfs command now

|  |
| --- |
| [root@mac127 keytabs]# **hdfs dfs -ls /**  17/04/28 11:04:26 WARN security.UserGroupInformation: PriviledgedActionException as:root (auth:KERBEROS) cause:javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)]  17/04/28 11:04:26 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)]  17/04/28 11:04:26 WARN security.UserGroupInformation: PriviledgedActionException as:root (auth:KERBEROS) cause:java.io.IOException: javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)]  ls: Failed on local exception: java.io.IOException: javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)]; Host Details : local host is: "mac127.cybage.com/172.27.155.127"; destination host is: "mac127.cybage.com":8020; |

# Give permission to root user

Check the latest hdfs.keytab file location under /**var/run/cloudera-scm-agent/process/<maxNumber>-hdfs-NAMNODE** folder

|  |
| --- |
| [root@mac127 etc]# **kinit -kt /var/run/cloudera-scm-agent/process/57-hdfs-NAMENODE/hdfs.keytab hdfs/mac127@CYBAGE.COM** |

Check Ticket

|  |
| --- |
| [root@mac127 etc]# **klist**  Ticket cache: FILE:/tmp/krb5cc\_0  Default principal: hdfs/mac127@CYBAGE.COM  Valid starting Expires Service principal  05/04/2017 10:31:32 05/05/2017 10:31:32 krbtgt/CYBAGE.COM@CYBAGE.COM  renew until 05/09/2017 10:31:32 |

Now you can access hdfs by root user

|  |
| --- |
| [root@mac127 etc]# **hdfs dfs -ls /**  Found 2 items  drwxrwxrwt - hdfs supergroup 0 2017-05-02 16:39 /tmp  drwxr-xr-x - hdfs supergroup 0 2017-05-02 15:00 /user |

# Add hdfs principal

|  |
| --- |
| root@mac127 security]# **mkdir -p /etc/security/keytabs/** |

Add principal for hdfs with random key

|  |
| --- |
| [root@mac127 ~]# **kadmin.local**  Authenticating as principal root/admin@CYBAGE.COM with password.  kadmin.local: **addprinc -randkey hdfs**  WARNING: no policy specified for hdfs@CYBAGE.COM; defaulting to no policy  Principal "hdfs@CYBAGE.COM" created.  kadmin.local: **listprincs**  HTTP/mac127@CYBAGE.COM  K/M@CYBAGE.COM  cloudera-scm/admin@CYBAGE.COM  hdfs/mac127@CYBAGE.COM  hdfs@CYBAGE.COM  hive/mac127@CYBAGE.COM  hue/mac127@CYBAGE.COM  kadmin/admin@CYBAGE.COM  kadmin/changepw@CYBAGE.COM  kadmin/mac127@CYBAGE.COM  kiprop/mac127@CYBAGE.COM  krbtgt/CYBAGE.COM@CYBAGE.COM  mapred/mac127@CYBAGE.COM  oozie/mac127@CYBAGE.COM  yarn/mac127@CYBAGE.COM  zookeeper/mac127@CYBAGE.COM |

# Create keytab file

A **keytab** is a file containing pairs of Kerberos principals and encrypted keys (which are derived from the Kerberos password). ... **Keytab files** are commonly used to allow scripts to automatically authenticate using Kerberos, without requiring human interaction or access to password stored in a plain-text file.

Create the hdfs.headless.keytab file that will contain the hdfs principal

|  |
| --- |
| kadmin.local: **ktadd -k /etc/security/keytabs/hdfs.headless.keytab -norandkey** [**hdfs@CYBAGE.COM**](mailto:hdfs@CYBAGE.COM)  Entry for principal hdfs@CYBAGE.COM with kvno 1, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/security/keytabs/hdfs.headless.keytab.  Entry for principal hdfs@CYBAGE.COM with kvno 1, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/security/keytabs/hdfs.headless.keytab.  Entry for principal hdfs@CYBAGE.COM with kvno 1, encryption type des3-cbc-sha1 added to keytab WRFILE:/etc/security/keytabs/hdfs.headless.keytab.  Entry for principal hdfs@CYBAGE.COM with kvno 1, encryption type arcfour-hmac added to keytab WRFILE:/etc/security/keytabs/hdfs.headless.keytab.  Entry for principal hdfs@CYBAGE.COM with kvno 1, encryption type camellia256-cts-cmac added to keytab WRFILE:/etc/security/keytabs/hdfs.headless.keytab.  Entry for principal hdfs@CYBAGE.COM with kvno 1, encryption type camellia128-cts-cmac added to keytab WRFILE:/etc/security/keytabs/hdfs.headless.keytab.  Entry for principal hdfs@CYBAGE.COM with kvno 1, encryption type des-hmac-sha1 added to keytab WRFILE:/etc/security/keytabs/hdfs.headless.keytab.  Entry for principal hdfs@CYBAGE.COM with kvno 1, encryption type des-cbc-md5 added to keytab WRFILE:/etc/security/keytabs/hdfs.headless.keytab.  kadmin.local: **quit** |

Or we can use following command instead of **ktadd**

|  |
| --- |
| **xst -norandkey -k /etc/security/keytabs/hdfs.headless.keytab hdfs@CYBAGE.COM** |

We can also use **ktadd** [**hdfs@CYBAGE.com**](mailto:hdfs@CYBAGE.com)but in this case it ask for create new password which we can use later while generating Kerberos ticket to the user using **kinit** [**hdfs@CYBAGE.COM**](mailto:hdfs@CYBAGE.COM)command

Generally for service we assign keytab file and for user we create password

Give permission and change ownership of keytab file

|  |
| --- |
| root@mac127 ~]# **chown hdfs:hadoop /etc/security/keytabs/hdfs.headless.keytab**  [root@mac127 ~]# **chmod 440 /etc/security/keytabs/hdfs.headless.keytab** |

# Add Ticket to hdfs User

|  |
| --- |
| [root@mac127 ~]# **su - hdfs**  Last login: Fri Apr 28 11:11:42 IST 2017 on pts/1  -bash-4.2$ **kinit -kt /etc/security/keytabs/hdfs.headless.keytab hdfs@CYBAGE.COM** |
| -bash-4.2$ **klist**  Ticket cache: FILE:/tmp/krb5cc\_985  Default principal: hdfs@CYBAGE.COM  Valid starting Expires Service principal  04/28/2017 11:14:51 04/29/2017 11:14:51 krbtgt/CYBAGE.COM@CYBAGE.COM  renew until 05/05/2017 11:14:51 |

By Default Ticket is valid for 1 day

## Switch to hdfs User

|  |
| --- |
| [root@mac127 keytabs]# **su - hdfs**  Last login: Tue May 2 10:19:11 IST 2017 on pts/1  -bash-4.2$  -bash-4.2$ **hdfs dfs -ls /**  Found 2 items  drwxrwxrwt - hdfs supergroup 0 2017-05-02 13:44 /tmp  drwxr-xr-x - hdfs supergroup 0 2017-05-02 13:44 /user  -bash-4.2$ |

Now we can access hdfs files

# Add new user to filesystem

To access hdfs file system form some different user we need to do following steps

|  |
| --- |
| [root@mac127 ~]# **useradd shalaj**  [root@mac127 ~]# |

Repeat all steps that we did for hdfs for user shalaj

|  |
| --- |
| [root@mac127 keytabs]# **kadmin.local**  Authenticating as principal hdfs/admin@CYBAGE.COM with password.  kadmin.local: **addprinc -randkey shalaj**  WARNING: no policy specified for shalaj@CYBAGE.COM; defaulting to no policy  Principal "shalaj@CYBAGE.COM" created.  kadmin.local: **listprincs**  HTTP/mac127@CYBAGE.COM  K/M@CYBAGE.COM  cloudera-scm/admin@CYBAGE.COM  hdfs/mac127@CYBAGE.COM  hdfs@CYBAGE.COM  hive/mac127@CYBAGE.COM  hue/mac127@CYBAGE.COM  kadmin/admin@CYBAGE.COM  kadmin/changepw@CYBAGE.COM  kadmin/mac127@CYBAGE.COM  kiprop/mac127@CYBAGE.COM  krbtgt/CYBAGE.COM@CYBAGE.COM  mapred/mac127@CYBAGE.COM  oozie/mac127@CYBAGE.COM  shalaj@CYBAGE.COM  yarn/mac127@CYBAGE.COM  zookeeper/mac127@CYBAGE.COM |

|  |
| --- |
| kadmin.local: **ktadd -k /etc/security/keytabs/shalaj.keytab -norandkey shalaj@CYBAGE.COM**  Entry for principal shalaj@CYBAGE.COM with kvno 1, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/security/keytabs/shalaj.keytab.  Entry for principal shalaj@CYBAGE.COM with kvno 1, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/security/keytabs/shalaj.keytab.  Entry for principal shalaj@CYBAGE.COM with kvno 1, encryption type des3-cbc-sha1 added to keytab WRFILE:/etc/security/keytabs/shalaj.keytab.  Entry for principal shalaj@CYBAGE.COM with kvno 1, encryption type arcfour-hmac added to keytab WRFILE:/etc/security/keytabs/shalaj.keytab.  Entry for principal shalaj@CYBAGE.COM with kvno 1, encryption type camellia256-cts-cmac added to keytab WRFILE:/etc/security/keytabs/shalaj.keytab.  Entry for principal shalaj@CYBAGE.COM with kvno 1, encryption type camellia128-cts-cmac added to keytab WRFILE:/etc/security/keytabs/shalaj.keytab.  Entry for principal shalaj@CYBAGE.COM with kvno 1, encryption type des-hmac-sha1 added to keytab WRFILE:/etc/security/keytabs/shalaj.keytab.  Entry for principal shalaj@CYBAGE.COM with kvno 1, encryption type des-cbc-md5 added to keytab WRFILE:/etc/security/keytabs/shalaj.keytab. |

|  |
| --- |
| root@mac127 ~]# **chown shalaj:hadoop /etc/security/keytabs/shalaj.keytab**  [root@mac127 ~]# **chmod 440 /etc/security/keytabs/shalaj.keytab** |
| [root@mac127 ~]# **su - shalaj**  Last login: Thu May 4 11:21:24 IST 2017 on pts/2  [shalaj@mac127 ~]$ **kinit -kt /etc/security/keytabs/shalaj.keytab shalaj@CYBAGE.COM** |
| Here kt denotes to keytab |
| -bash-4.2$ **klist**  Ticket cache: FILE:/tmp/krb5cc\_1001  Default principal: shalaj@CYBAGE.COM  Valid starting Expires Service principal  05/02/2017 14:33:41 05/03/2017 14:33:41 krbtgt/CYBAGE.COM@CYBAGE.COM  renew until 05/09/2017 14:33:41 |

|  |
| --- |
| [shalaj@mac127 ~]$ **hdfs dfs -ls /**  Found 2 items  drwxrwxrwt - hdfs supergroup 0 2017-05-02 13:44 /tmp  drwxr-xr-x - hdfs supergroup 0 2017-05-02 13:44 /user |

Now create directory shalaj and change ownership

|  |
| --- |
| [root@mac127 keytabs]# **sudo -u hdfs hadoop fs -mkdir /user/shalaj**  [root@mac127 keytabs]# **sudo -u hdfs hadoop fs -chown shalaj:shalaj /user/shalaj** |

Or you can do the same thing without using separate keytab file

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| First delete existing principal   |  | | --- | | kadmin.local: **delete\_principal shalaj**  Are you sure you want to delete the principal "shalaj@CYBAGE.COM"? (yes/no): yes  delete\_principal: Principal does not exist while deleting principal "shalaj@CYBAGE.COM" |   Add principal for user shalaj and specify password for the same   |  | | --- | | kadmin.local: **addprinc shalaj**  WARNING: no policy specified for shalaj@CYBAGE.COM; defaulting to no policy  Enter password for principal "shalaj@CYBAGE.COM":  Re-enter password for principal "shalaj@CYBAGE.COM":  Principal "shalaj@CYBAGE.COM" created.  kadmin.local:quit |   Now switch to shalaj user   |  | | --- | | [shalaj@mac127 ~]$ **kinit**  Password for shalaj@CYBAGE.COM:  [shalaj@mac127 ~]$ **klist**  Ticket cache: FILE:/tmp/krb5cc\_1001  Default principal: shalaj@CYBAGE.COM  Valid starting Expires Service principal  05/04/2017 11:22:24 05/05/2017 11:22:24 krbtgt/CYBAGE.COM@CYBAGE.COM  renew until 05/11/2017 11:22:24 |   Now you can access hdfs   |  | | --- | | [shalaj@mac127 ~]$ **hdfs dfs -ls /**  Found 2 items  drwxrwxrwt - hdfs supergroup 0 2017-05-02 16:39 /tmp  drwxr-xr-x - hdfs supergroup 0 2017-05-04 10:48 /user | |

## Check hive shell

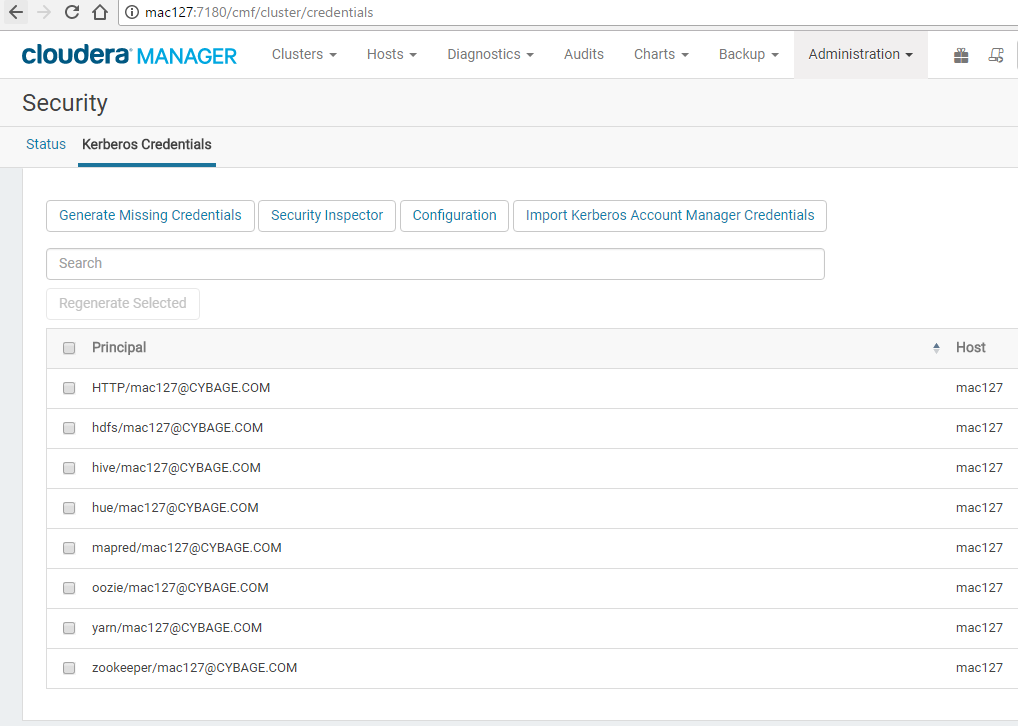
|  |
| --- |
| [shalaj@mac127 ~]$ hive  Logging initialized using configuration in jar:file:/opt/cloudera/parcels/CDH-5.10.1-1.cdh5.10.1.p0.10/jars/hive-common-1.1.0-cdh5.10.1.jar!/hive-log4j.properties  WARNING: Hive CLI is deprecated and migration to Beeline is recommended.  hive> |

## Check beeline

|  |
| --- |
| [shalaj@mac127 ~]$ **beeline**  Beeline version 1.1.0-cdh5.10.1 by Apache Hive  beeline> **!connect jdbc:hive2://mac127:10000**  scan complete in 3ms  Connecting to jdbc:hive2://mac127:10000  Enter username for jdbc:hive2://mac127:10000: root  Enter password for jdbc:hive2://mac127:10000: \*\*\*\*\*\*\*\*  Unknown HS2 problem when communicating with Thrift server.  Error: Could not open client transport with JDBC Uri: jdbc:hive2://mac127:10000: Peer indicated failure: Unsupported mechanism type PLAIN (state=08S01,code=0) |

Can’t connect without giving principal, we need to provide principal while connecting hive using beeline

You can get principal list from Cloudera manager web console—Administration>>Security



|  |
| --- |
| beeline> **!connect jdbc:hive2://mac127:10000/default;principal=hive/mac127@CYBAGE.COM**  Connecting to jdbc:hive2://mac127:10000/default;principal=hive/mac127@CYBAGE.COM  Connected to: Apache Hive (version 1.1.0-cdh5.10.1)  Driver: Hive JDBC (version 1.1.0-cdh5.10.1)  Transaction isolation: TRANSACTION\_REPEATABLE\_READ  0: jdbc:hive2://mac127:10000/default> |

## Check mapreduce job

|  |
| --- |
| **hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-0.20-mapreduce/hadoop-examples.jar pi 10 10000**  Number of Maps = 10  Samples per Map = 10000  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  17/05/02 15:01:32 INFO client.RMProxy: Connecting to ResourceManager at mac127/172.27.155.127:8032  17/05/02 15:01:32 INFO hdfs.DFSClient: Created token for shalaj: HDFS\_DELEGATION\_TOKEN owner=shalaj@CYBAGE.COM, renewer=yarn, realUser=, issueDate=1493717492392, maxDate=1494322292392, sequenceNumber=3, masterKeyId=2 on 172.27.155.127:8020  17/05/02 15:01:32 INFO security.TokenCache: Got dt for hdfs://mac127:8020; Kind: HDFS\_DELEGATION\_TOKEN, Service: 172.27.155.127:8020, Ident: (token for shalaj: HDFS\_DELEGATION\_TOKEN owner=shalaj@CYBAGE.COM, renewer=yarn, realUser=, issueDate=1493717492392, maxDate=1494322292392, sequenceNumber=3, masterKeyId=2)  17/05/02 15:01:32 INFO input.FileInputFormat: Total input paths to process : 10  17/05/02 15:01:32 INFO mapreduce.JobSubmitter: number of splits:10  17/05/02 15:01:33 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1493716904621\_0001  17/05/02 15:01:33 INFO mapreduce.JobSubmitter: Kind: HDFS\_DELEGATION\_TOKEN, Service: 172.27.155.127:8020, Ident: (token for shalaj: HDFS\_DELEGATION\_TOKEN owner=shalaj@CYBAGE.COM, renewer=yarn, realUser=, issueDate=1493717492392, maxDate=1494322292392, sequenceNumber=3, masterKeyId=2)  17/05/02 15:01:34 INFO impl.YarnClientImpl: Submitted application application\_1493716904621\_0001  17/05/02 15:01:34 INFO mapreduce.Job: The url to track the job: http://mac127:8088/proxy/application\_1493716904621\_0001/  17/05/02 15:01:34 INFO mapreduce.Job: Running job: job\_1493716904621\_0001  17/05/02 15:01:48 INFO mapreduce.Job: Job job\_1493716904621\_0001 running in uber mode : false  17/05/02 15:01:48 INFO mapreduce.Job: map 0% reduce 0%  17/05/02 15:01:57 INFO mapreduce.Job: map 20% reduce 0%  17/05/02 15:01:58 INFO mapreduce.Job: map 30% reduce 0%  17/05/02 15:02:05 INFO mapreduce.Job: map 50% reduce 0%  17/05/02 15:02:06 INFO mapreduce.Job: map 60% reduce 0%  17/05/02 15:02:13 INFO mapreduce.Job: map 80% reduce 0%  17/05/02 15:02:14 INFO mapreduce.Job: map 90% reduce 0%  17/05/02 15:02:19 INFO mapreduce.Job: map 100% reduce 0% |

# Access hdfs using client machine

Install Kerberos workstation on client machine

On client machine (mac92)

|  |
| --- |
| [root@a ~]# yum install krb5-workstation |

Copy krb5.conf file from Kerberos server to client machine

|  |
| --- |
| [root@mac127 krb5kdc]# **scp /etc/krb5.conf root@172.27.155.92:/etc/**  The authenticity of host '172.27.155.92 (172.27.155.92)' can't be established.  ECDSA key fingerprint is bf:90:22:74:9a:72:fa:7d:7c:bf:86:20:00:1c:19:03.  Are you sure you want to continue connecting (yes/no)? yes  Warning: Permanently added '172.27.155.92' (ECDSA) to the list of known hosts.  root@172.27.155.92's password:  krb5.conf |

Copy keytabs files from server to client

|  |
| --- |
| [root@mac127 ~]# **scp /etc/security/keytabs/\* root@172.27.155.92:/etc/security/keytabs**  root@172.27.155.92's password:  hdfs.headless.keytab 100% 482 0.5KB/s 00:00  shalaj.keytab 100% 498 0.5KB/s 00:00 |

Change permission and ownership of keytab file in client machine

|  |
| --- |
| [root@a keytabs]# **chown hdfs:hadoop /etc/security/keytabs/hdfs.headless.keytab**  [root@a keytabs]# **chmod 440 /etc/security/keytabs/hdfs.headless.keytab** |

On client machine get the tickets for user hdfs

|  |
| --- |
| [root@a keytabs]# **su - hdfs**  Last login: Wed May 3 01:28:17 EDT 2017 on pts/1  -bash-4.2$ **kinit -kt /etc/security/keytabs/hdfs.headless.keytab hdfs@CYBAGE.COM**  -bash-4.2$ **klist**  Ticket cache: FILE:/tmp/krb5cc\_984  Default principal: hdfs@CYBAGE.COM  Valid starting Expires Service principal  05/03/2017 01:29:03 05/04/2017 01:29:03 krbtgt/CYBAGE.COM@CYBAGE.COM  renew until 05/10/2017 01:29:03 |

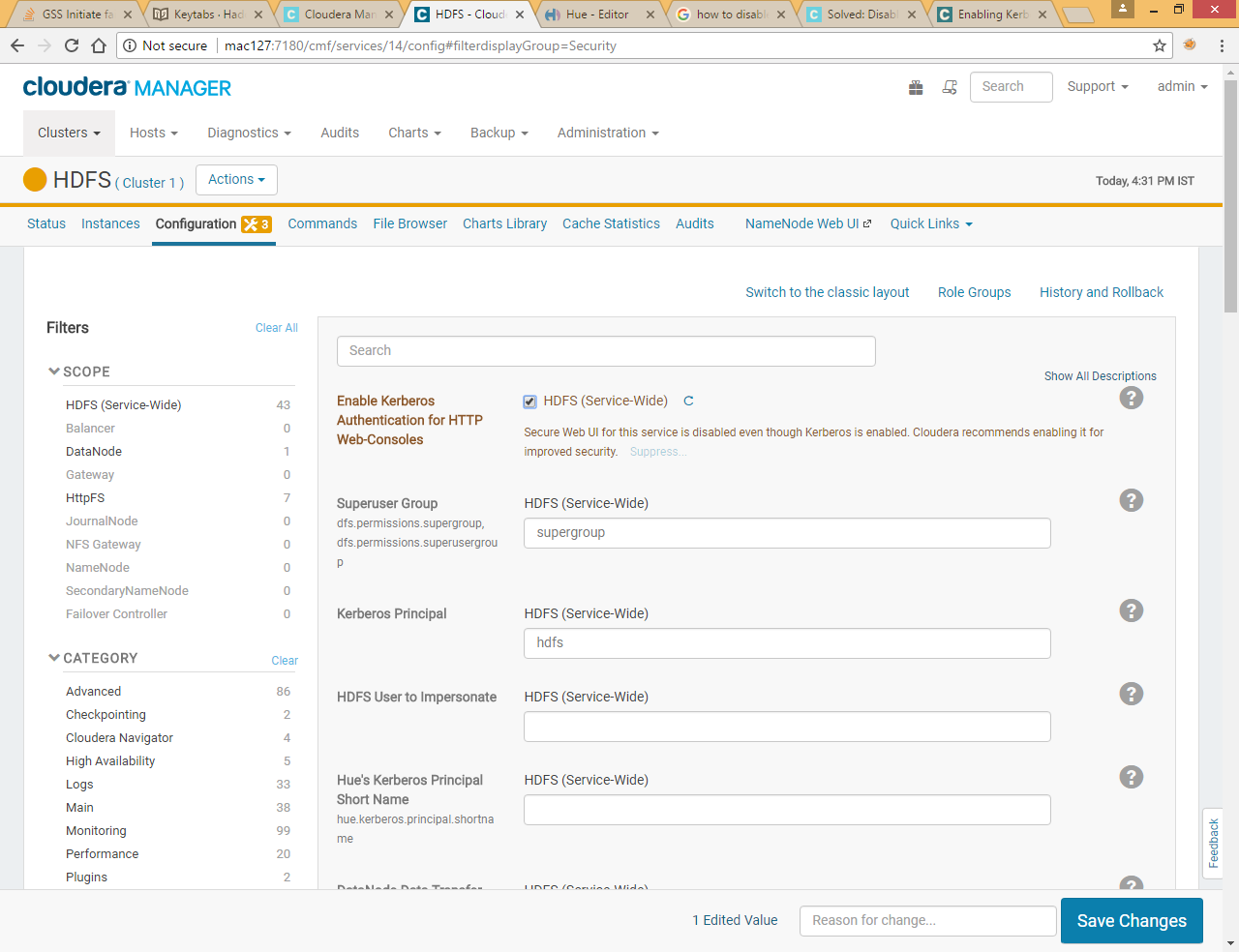
Now access hdfs system

|  |
| --- |
| -bash-4.2$ **hdfs dfs -ls /**  Found 2 items  drwxrwxrwt - hdfs supergroup 0 2017-05-02 07:09 /tmp  drwxr-xr-x - hdfs supergroup 0 2017-05-02 05:30 /user  -bash-4.2$ |

# Enable Kerberos Authentication for HTTP Web Console

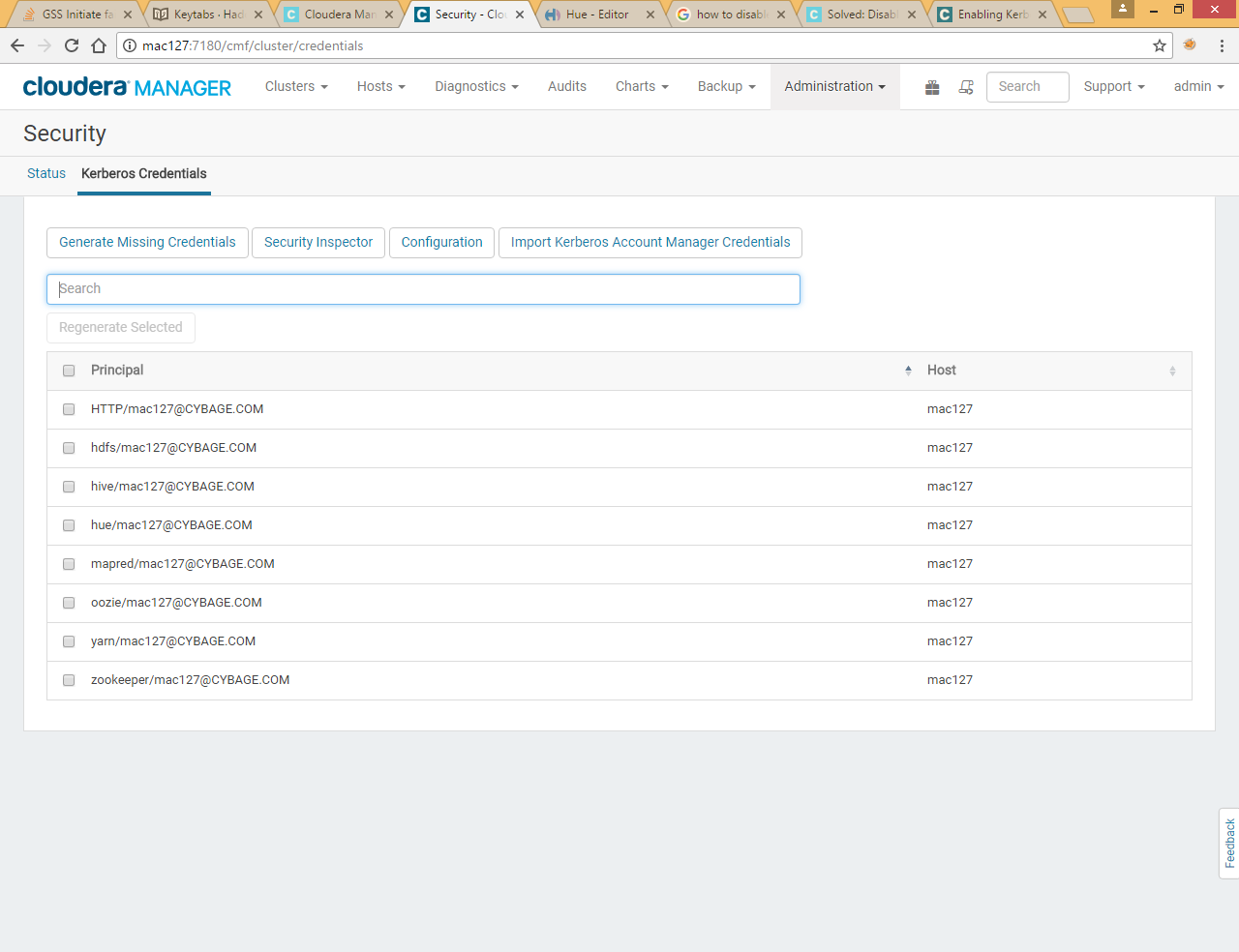
Security for WebHDFS is disabled by default. If you want use WebHDFS with a secure cluster, this is the time to enable and configure it. (this is an optional step)

Click on HDFS >>Configuration and Search Enable Kerberos Authetication

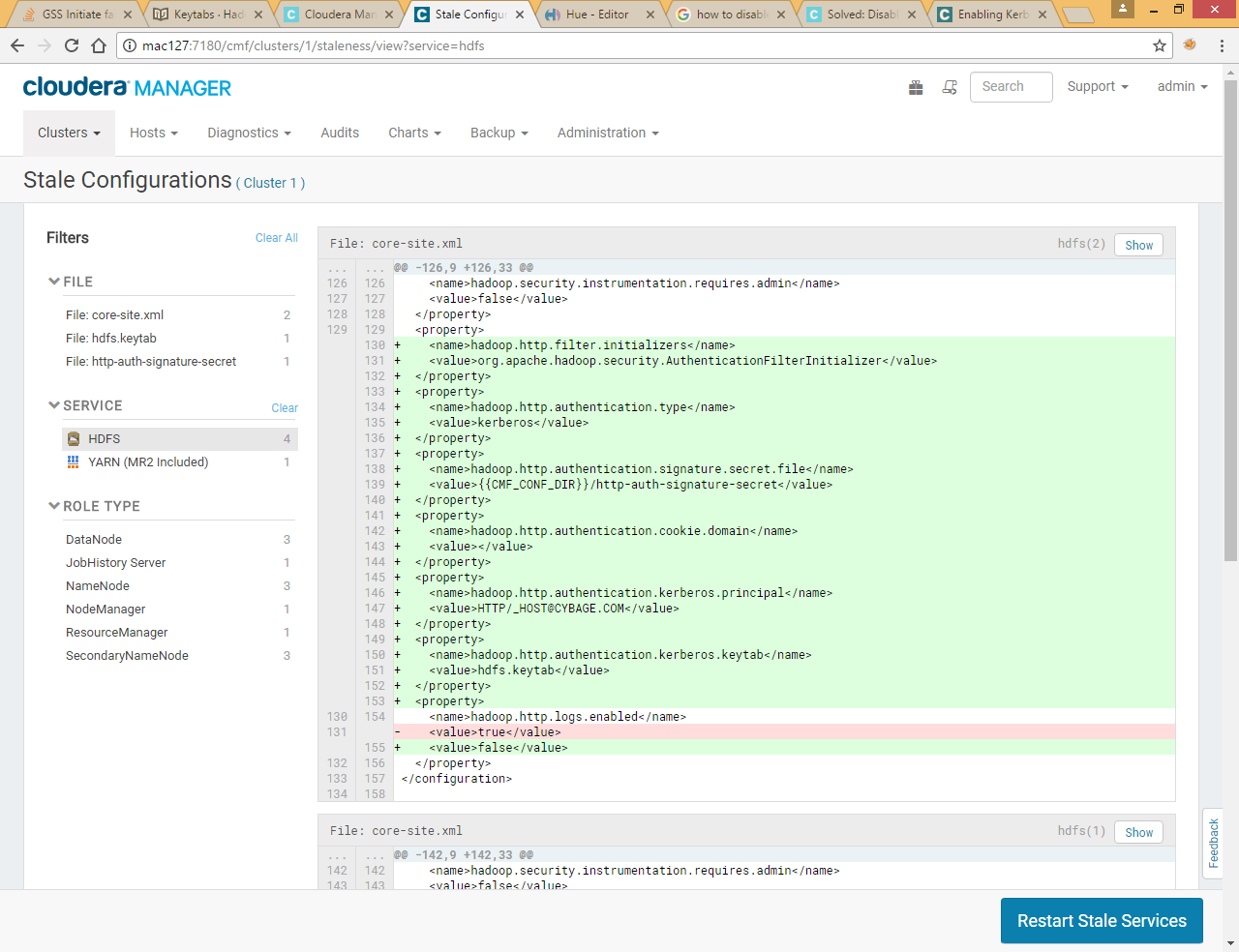


Select check box

Go to Administration>>Security



Click on Generate Missing Credentials



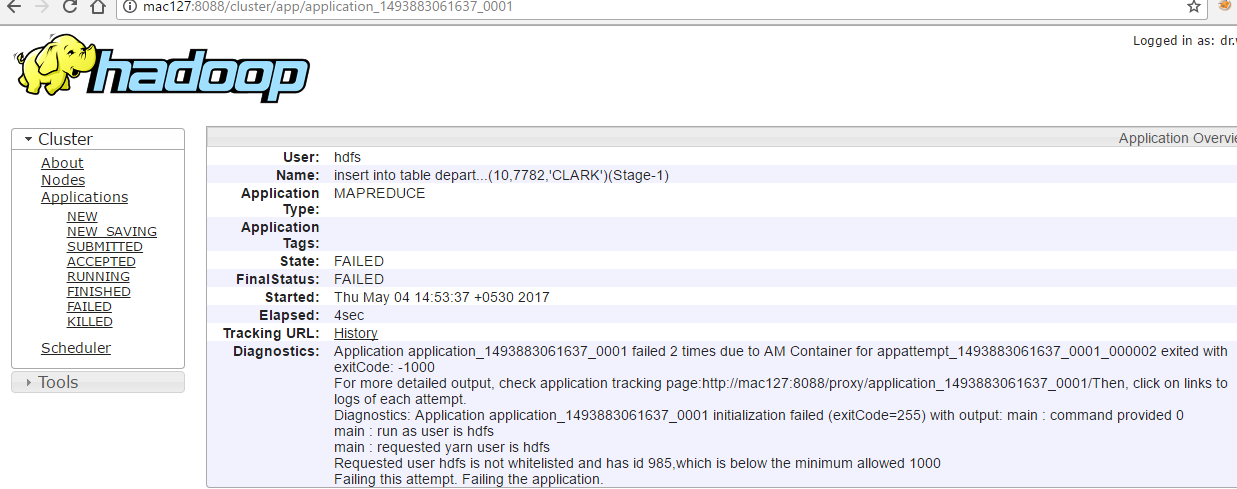
Restart Stale Services

# Issue Resolution

## Minimum User Id issue

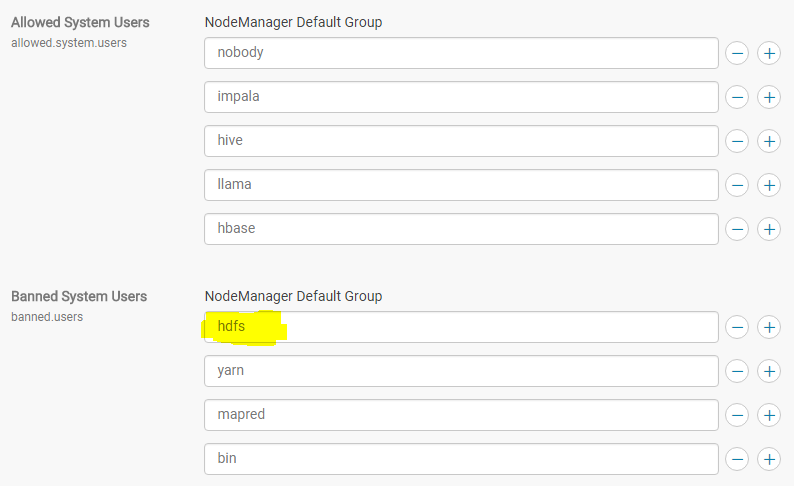
During execution of map reduce job get issue below

Requested user hdfs is not whitelisted and has id 985, which is below the minimum allowed 1000



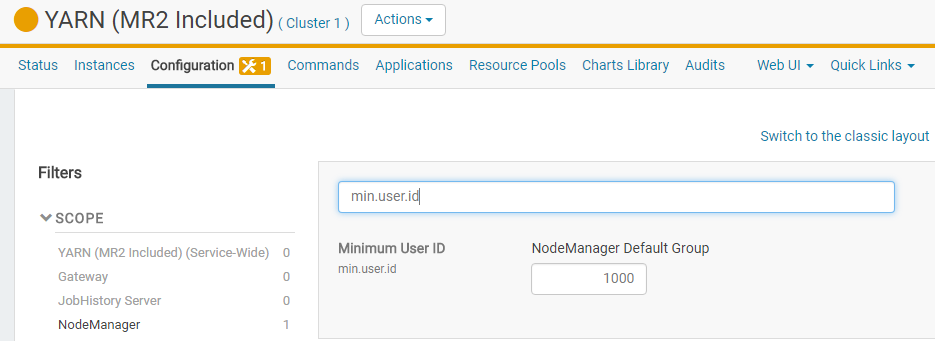
Here we have two issues first hdfs user is under banned user list and second minimum user id of hdfs user is 985 which is less than minimum allowed user id

Go to yarn configuration and check banned.users you can find hdfs user there, we can remove hdfs from banned user list and add to allowed system users but ideally hdfs user should not be allowed since it is the superuser and could circumvent the HDFS access permissions.



So we must not run the map reduce job through hdfs user, we should create new user and assign new ticket to this user.

Now check min user id



Now run id hdfs command on linux file sytem to check the hdfs user id

|  |
| --- |
| [root@mac127]# **id hdfs**  uid=985(hdfs) gid=979(hdfs) groups=979(hdfs),981(hadoop) |

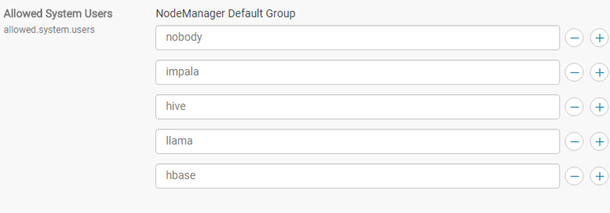
So we can clearly say that hdfs user id is less than minimum allowed user id which is 1000

Ideally we should not change the **allowed minimum user id**, instead we should run the mapreduce job from some other user

Only two type of user can run yarn container

1. Whose id is greater than specified minimum user id (default is 1000)

2. If user id is less than minimum user id then user should be included under **Allowed System User**



So ideally to run the yarn container We should create one new user and [assign valid ticket](#_Add_new_user)

We can create same user for hue also to run any map reduce job