**Configure a system to authenticate using Kerberos.**

**Prerequisites**

Before configuring a **Kerberos** client, you have to [configure a KDC](https://www.certdepot.net/rhel7-configure-kerberos-kdc/).  
Also, to get **Kerberos** running, [NTP synchronization](https://www.certdepot.net/rhel7-set-ntp-service/) and **hostname resolution** must be working.  
If no working **DNS**, add the following lines in the **/etc/hosts** file (replace the specified ip addresses with yours):

192.168.1.11 **kbserver.example.com**

192.168.1.12 **kbclient.example.com**

**Client Configuration**

Install the **Kerberos** client packages:

# **yum install -y krb5-workstation pam\_krb5**

Edit the **/etc/krb5.conf** file, uncomment all the lines, replace **EXAMPLE.COM** with your own realm, **example.com** with your own domain name, and **kerberos.example.com** with your own **KDC** server name (here **kbserver.example.com**).

**Alternatively**, you get the **/etc/krb5.conf** file from the **KDC** server (here **kbserver.example.com**):

# **scp root@kbserver.example.com:/etc/krb5.conf /etc/krb5.conf**

Create a user for test:

# **useradd user01**

Add the client machine name (here **kbclient.example.com**) to the principals:

# **kadmin**

Authenticating as principal root/admin@EXAMPLE.COM with password.

Password for root/admin@EXAMPLE.COM:

kadmin: **addprinc -randkey host/kbclient.example.com**

WARNING: no policy specified for host/kbclient.example.com@EXAMPLE.COM; defaulting to no policy

Principal "host/kbclient.example.com@EXAMPLE.COM" created.

Create a local copy stored by default in the **/etc/krb5.keytab** file:

kadmin: **ktadd host/kbclient.example.com**

Entry for principal host/kbclient.example.com with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/kbclient.example.com with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/kbclient.example.com with kvno 2, encryption type des3-cbc-sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/kbclient.example.com with kvno 2, encryption type arcfour-hmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/kbclient.example.com with kvno 2, encryption type camellia256-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/kbclient.example.com with kvno 2, encryption type camellia128-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/kbclient.example.com with kvno 2, encryption type des-hmac-sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/kbclient.example.com with kvno 2, encryption type des-cbc-md5 added to keytab WRFILE:/etc/krb5.keytab.

kadmin: **quit**

Edit the **/etc/ssh/ssh\_config** file and add/uncomment the following lines:

**GSSAPIAuthentication** yes

**GSSAPIDelegateCredentials** yes

Reload the **sshd** service configuration:

# **systemctl reload sshd**

Configure the **PAM** component at the command line:

# **authconfig --enablekrb5 --update**

Test your configuration (here **kbserver.example.com** is the **KDC** server name):

# **su - user01**

$ **kinit**

Password for user01@EXAMPLE.COM:

$ **klist**

Ticket cache: KEYRING:persistent:1000:1000

Default principal: user01@EXAMPLE.COM

Valid starting Expires Service principal

07/22/2014 17:20:15 07/23/2014 17:19:54 krbtgt/EXAMPLE.COM@EXAMPLE.COM

renew until 07/22/2014 17:19:54

$ **ssh kbserver.example.com**

Now, you should be able to quit and reconnect without giving any password.  
In addition, the first time you log in to a **Kerberos** client, you have to provide a login/password (see **kinit** above). Then, you get a ticket that allows you to log in to all the other **Kerberos** clients in the same realm and you don’t need to provide a password any more as long as your ticket is valid.  
Note: To delete a ticket, use the **kdestroy** command.