RHCSA 7

Advanced Topics Study Guide For RHCSA

Table of Contents

Introduction

This study guide is designed to be smaller and less cluttered and will only include advanced topics that require extra memorization or understanding for the RHCSA 7 Certification. The key to being successful in performance based exams like RHCSA is to know "how to find documentation". In the exam, you can use any documentation on the system.

Packages That Need to be Memorized and Installed

- virt-manager
- qemu-kvm qemu-img
- libvirt
- libvirt-python
- python-virtinst
- libvirt-client
- setroubleshoot-server
- firewalld
- firewall-config
- selinux-policy-devel (lot of SELinux man pages)
  - mandb -> rebuilt the man pages
  - man -k _selinux

Configure Key-based Authentication for SSH

- ssh-keygen -t rsa (if -t is not specified then it is by default RSA)
- ssh-copy-id server@server.com
- It is best practice to configure with a passphrase. However, you have to continuously enter the passphrase during SSHing to remote machines. You have the ability to add the passphrase to the shell "for your current session only".
- ssh-agent bash
- ssh-add
- Now you can easily perform remote commands on servers with advanced SSH authentication without having to enter the pass phrase over and over again.

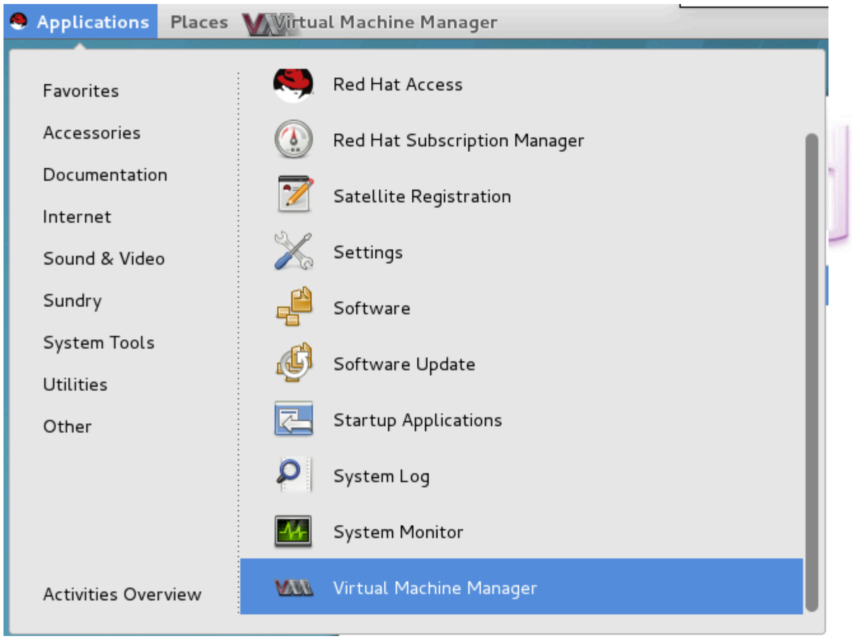
KVM

Once KVM is installed, this guide will focus on using the KVM manager inside of the server console GUI. For certification testing purposes, it is faster and easier to perform the given objectives.

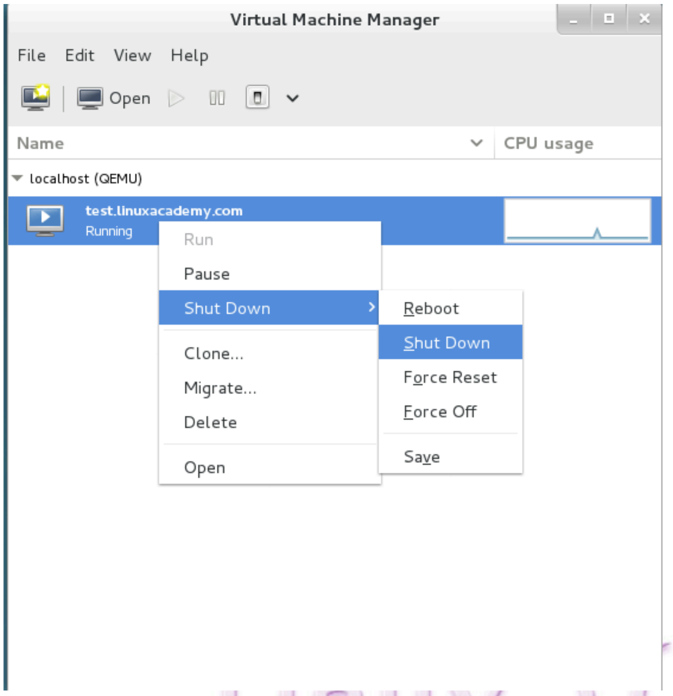
Task: Install KVM software packages

```
[root@localhost]# yum install virt-manager qemu-kvm qemu-img

[root@localhost]# yum install libvirt libvirt-python python-virtinst libvirt-client

[root@localhost]# systemctl enable libvrtd

[root@localhost]# virsh autostart vmname
```

Open The Virtual Machine Manager



Task: Shutdown a running virtual machine



Note: Starting a virtual machine is the same; after it is shut down right click on the virtual machine in the KVM manager and select “run”.

Task: Configure virtual machines to run at boot

Considerations:

- After installing the virtualization packages manually, the service is started but not enabled to start at boot time.
- A specific virtual machine that is asked to start at boot time must also be set with the autostart option from the virsh command line.

```
[root@localhost]# systemctl enable libvirtd

[root@localhost]# virsh

virsh> autostart vmname

[root@localhost]# reboot
```

list --all (show all running vms and stopped); list will show just running

FirewallD

Note: For the exam it might be easier to launch firewall-config to make quick firewall changes to your environment in a console environment.

Configuration files: /etc/firewalld/zones (show all rules)

Mask IP tables if you are going to use firewalld

```
[root@localhost]# systemctl mask iptables
```

Reload the firewall rules after making permanent changes

```
[root@localhost]# firewall-cmd --reload
```

Note: A reload is required to make changes persistent if a persistent change is made

```
[root@localhost]# firewall-cmd --get-zones
[root@localhost]# firewall-cmd --list-all-zones
[root@localhost]# firewall-cmd --get-default-zone
```

Add a rule to the public zone for port 80 that applies only to the current runtime environment

```
[root@localhost]#  firewall-cmd --zone=public --add-port=80/tcp
```

List all current active firewalld rules <u>for the default zone</u>

```
[root@localhost]# firewall-cmd --list-all
```

List all current active rules for just the public zone

```
[root@localhost]# firewall-cmd --zone=public --list-all
```

Add a persistent rule for port 80 TCP

```
[root@localhost]# firewall-cmd --permanent --zone=public --add-port=80/tcp
```

View all rules

```
[root@localhost]# firewall-cmd --list-all (view the new rule exists)
```

Remove port 80 from default zone (remember, if you do not specify a zone, then it uses the default)

```
[root@localhost]# firewall-cmd --remove-port=80/tcp
```

Specify the internal zone and remove port 80 for the internal zone persistently

```
[root@localhost]# firewall-cmd --zone=internal  --permanent --remove-port=80/tcp
```

Add a source to the internal zone

```
[root@localhost]#  firewall-cmd --permanent --zone=internal --add-source=10.0.0.0/24
```

Remove a source to the internal zone

```
[root@localhost]#  firewall-cmd --permanent --zone=internal --remove-source=10.0.0.0/24
```

Set the default zone

```
firewall-cmd --set-default-zone=zone
```

View available zones

```
firewall-cmd --list-all-zones
```

List all predefined services

```
firewall-cmd --get-services
```

List zones currently in use

```
firewall-cmd --get-active-zones

Systemctl enable firewalld

Systemctl start firewalld
```

Panic mode

```
[root@localhost]# firewall-cmd --panic-on

[root@localhost]# firewall-cmd --query-panic
```

Set Enforcing and Permissive Modes for SELinux

Permissive mode for SELinux means that it is monitoring and logging events but not enforcing.

- Set permissive mode that is not persistent
  - [root@localhost]# setenforce 0
- Set enforcing mode that is not persistent
  - [root@localhost]# setenforce 1
- /etc/selinux/config
  - SELINUX=enforcing|permissive|disabled
  - Note: In order to transition into disabled mode, you must change the configuration file and then reboot.

Introduction to SELinux

Enforcing mode

Permissive mode

Disabled mode

List and Identify SELinux File and Process Context

The SELinux contents of a parent directory determine the context for the newly created file or directory.

View contexts of directories, files, and processes

- ls -Z
- ps auxZ

Changing SElinux context: The best method for changing SELinux context on a file or directory is to update the context using semanage fcontext and then use “restorecon” command on the files or directories that need the context applied. The restorecon command will update the default SEContext rules.

semanage fcontext command is used to display or modify SELinux context rules.

- View all SELinux contexts currently configured
  - **semanage fcontext -l**

Reliable all files and directories on the operating system

- **touch /.autorelabel**
- **reboot**

Restore Default File Contexts

Default contexts should be added and then the directory or files should use restorecon to receive those default contexts.

Note: commands issued with semanage are permanent by default

- Add a new default context
  - **semnage fcontext -a -t httpd_sys_content_t ‘/content(/.*)?’**
  - **restorecon -Rv /content**
- Delete a default context
  - **semnage fcontext -d -t httpd_sys_content_t ‘/content(/.*)?’**
  - **restorecon -Rv /content**

Use Boolean Settings to Modify System SELinux Settings

Boolean values are used to enable or disable a specific feature set for a server/service. For example, vsftp can enable anonymous login, apache can enable or disable CGI, apache can enable or disable user home directories all through switching a Boolean value. It is important to remember that setting a Boolean value does not make it persistent unless you specify a persistent change. Non-persistent changes will not survive a reboot.

- List a Boolean value
  - **getsebool -a**
- Determine if a Boolean value is persistent
  - **semanage Boolean -l**
- Get a specific Boolean value
  - **getsebool httpd_can_sendmail**
- Make a non persistent change to the Boolean value
  - **setsebool httpd_can_sendmail on**
- Make a persistent change to enable the Boolean value
  - **setsebool -P httpd_can_sendmail on**

- View only the default changes, for example if you have changed a default view, all Booleans that have been changed
    - **semanage boolean -l -C**

Diagnose and Address Routine SELinux Policy Violations

install package setroubleshoot-server

sealert - Sealert is the user interface component to the setroubleshoot system used to diagnose SELinux denials. It attempts to provide user friendly explanations for SELinux denials and recommendations for how one might adjust the system to prevent the denial in the future.

Example:

```
sealert -a /var/log/audit/audit.log
```

Working with LVM (Logical Volume Manager)

Partition volumes (if they do not already exist) to make physical volumes

Ensure LVM Is installed

```
[root@localhost]# yum install lvm2
```

Create LVM physical volumes

```
[root@localhost]# pvcreate /dev/disk /dev/disk (however many disks you are adding)
```

Display all available physical volumes

```
[root@localhost]# pvdisplay
```

Create a volume group

```
[root@localhost]# vgcreate vg-name /dev/disk /dev/disk
```

vg-name is the name you assign the volume group and the disks are the physical disks that make up the volume group

Display all available volume groups

```
[root@localhost]# vgdisplay
```

Create a logical volume

```
[root@localhost]# lvcreate -n nameoflvmvolume -L 20G volumegrouptouse
```

-L size of the volume in bytes (also use 1M, 1G 20G etc)

-l size in physical extents; generally a physical extent is 4MiB in size

add the designated amount of space onto the already existing space (increases size)

Display view logical volumes

```
[root@localhost]# lvdispaly
```

Extending Volume Groups and Logical Volumes

Extend a volume group with a new disk

```
[root@localhost]#  pvcreate /dev/newdisk
```

```
[root@localhost]#  vgextend vg-name /dev/newdisk
```

```
[root@localhost]#  vgdisplay
```

Extend the LVM to be exactly 5G in size

[root@localhost]# lvextend -L 5G /dev/vg-name/lvmname this makes the volume 5G in size

Extend the LVM and add an additional 5G

```
[root@localhost]# lvextend -L +5G /dev/vg-name/lvmname
```

(Remember you can use the -l option and use physical extents as units of measurements as well)

Remove Logical Volumes, Volume Groups, and Physical Volumes

Remove a logical volume

```
[root@localhost]#  lvremove /dev/vg-name/volume
```

Remove a volume group

```
[root@localhost]#  vgremove vg-name
```

Remove a physical volume

```
[root@localhost]#  pvremove /dev/device
```

Resetting Root Password

1. Start or reboot a system to get into the boot menu.

2. Press any key to stop the auto selection of a menu item.

3. Ensure the kernel you intend to boot into is highlighted and press the "e" key to edit the entry.

4. Navigate to the linux16 kernel line and hit the "end' key to go to the end of the linux16 line.

5. Append **rd.break** to the **linux16** kernel line.

6. Hit crtl + x to continue.

7. The system will boot into an emergency mode that has the /sysroot directory mounted as read only.

8. Mount the /sysroot directory with read and write permissions.

```
mount -oremount, rw /sysroot
```

9. Switch into chroot jail and set the /sysroot as the root file system.

```
chroot /sysroot
```

10. Reset the root password.

```
passwd root
```

11. Clean up, -> Make sure that all unlabled files get relabeled during the boot process (for SELinux).

```
touch ./autorelabel
```

12. Exit chroot jail.

```
exit
```

13. Exit the initramfs debug shell.

```
exit
```

Troubleshooting notes:

Go through the process and reboot but password not changed?

- Chances are the touch ./autorelabel was missed or performed incorrectly.
- Chances are the file system was not mounted as read/write so the changes made were not persistent.

Kickstart

Know where to find kickstart documentation

```
[root@localhost]# rpm -qd kickstart.py
```

Install the kickstart template builder

```
[root@localhost]#
```

Configuration sections start with % and end with %end

```
%packages
```

Package groups can be @^groupname

%end

%pre

Configure the system before packages are installed

```
%end
%post
```

Configure the system after it has installed packages

Installation commands

url --url =“url to installation media”

repo –name=”Custome packages” --baseurl=”repourl”

Configuration options

clears the specified partitions before installation

```
clearpart --all -drives=sda,sdb --initlabel
```

part: Set root password

Generate an encrypted hash password

```
openssl passwd -1 “here”
```

part: specifies the size, format, and name of partition

```
part /home --fstype=ext4 --label=home --size=4096 --maxsize=8192 --grow
```

ignoredisk: ignores the specified disks when install

```
ignoredisk --drives=sdc
```

bootloader: defines where to install the bootloader

```
bootloadert --location=mbr --boot-drive=sda
```

volgroup, logvol: Creates LVM volume groups and logical volumes

ksvalidator verifies the syntax of a kickstart file

A kickstart bootable redhat file needs to be available in order to start kickstart.

Kickstart Installation

1. Create a kickstart configuration file
    a. yum install system-config-kickstart
    b. system-config-kickstart
    c. can also use /root/anaconda-ks.cfg which is generated from the kickstart installation from the current install
2. Publish the kickstart configuration file to the installer
3. Boot anaconda and point I to the kickstart configuration file

Configure a system to use an existing authentication service for user and group information

Networking

Exam Tip: Your key to success is learning how to use bash completion when completing the nmcli command options. For example, become familiar with all of the modify options available. Because of bash completion you do not need to memorize the command; rather you should know how to use the completion and be familiar with configuring networking.

IP command

- Display IP information for a device
  - **Ip addr show eth0**
- Display network statistics for a device
  - **ip -s link show eth0**
- Display routes for a device
  - **ip route**
- Display network devices
  - **ip link**

Troubleshooting network connectivity

- **tracepath** - show all hops a packet has to go through and the MTA for each router; not all routers support this and require it to be enabled on the router
- **traceroute** - older version of traceroute and works on all routers
- **ping**
- **ss -** utility used to investigate sockets
  - –a all, listening and established
  - -t display TCP sockets
  - ss -ta
  - -n show numbers instead of names for interfaces and ports
  - -u display udp sockets
  - -l show only listening sockets

Network Manager

/etc/sysconfig/network-scripts are the location of all configuration files for network cards etc.

ls /sys/class/net to view which network cards are attached to the system or you can use nmcli dev status to list all the devices.

- Display a list of connections
  - nmcli con show
  - nmcli con show --active to display only active connections
  - nmcli con show “con-name”
- Display “device” status

    - nmcli dev status
- Display help for adding connections
    - nmcli con add help
- Add a connection
    - nmcli con add con-name “nameofthecon” type Ethernet ifname eth0
        - ifname is the name of the Ethernet device; can find them by doing ls /sys/class/net
        - By not specifying a gateway or IP, the connection will attempt to go with DHCP
    - nmcli con add con-name “nameofthecon” ifname eth0 autoconnect no type Ethernet ip4 ipaddress gw4 gateway
        - autoconnect will automatically bring up the network connection when the system start
- Modify an existing network connection
    - nmcli con mod “nameofthecon”
        - Tab tab to auto complete and view potential options
        - After modifying a network connection, you should reload the configuration
            - nmcli con reload
- List all devices
    - **nmcli dev status**
- Turn off a network device
    - **nmcli dev disconnect “device”**
- Turn on a network device
    - **nmcli dev connect “device”**
- List all connections (connections are configurations that are attached to a device)
    - **nmcli con show**
- Activate or “bring up a connection”
    - **nmcli con up “connectionname”**
- Deactivate or “bring down a connection”
    - **nmcli con down “connectionname”**
- Delete a connection
    - **nmcli con del “connectionname”**

- Change the method from static ip to DHCP

    - **nmcli con mod "con-name ipv4.method manual**
- Example: Add static ip address to eno1 with ip of 192.168.122.5/24 and gateway of 192.168.1.1
    - **nmcli con mod eno1 ipv4.addresses "192.168.122.5/24 192.168.1.1"**

Tip: Anytime you change the IP address on a network connection manually, be sure to modify the connection and set the ipv4.method to manual.

Tip: You need DNS, IP Address, and Gateway at the minimum to connect to the internet.

Note: Network connections have their own DNS. If you set it in the DNS, it will look there before looking into the /etc/resolv.conf file. You can disable DHCP DNS through the modify options in the using the nmcli utility.

Start and stop network manager

- **systemctl start NetworkManager**
- **systemctl enable NetworkManager**
- **systemctl restart network**

Tip: If you get lost, use the following man page "nmcli-examples" to help when configuring nmcli.

nm-connection-manager -> GUI interface to manage connections

Managing Hostnames

- Display the systems fully qualified hostname
    - hostname
- /etc/hostname manages the "static" hostname for the system. Rather than modifying the file, you can use the hostnamectl command.
    - hostnamectl set-hostname system.domain.com
    - hostnamectl status
- DNS resolution is stored in /etc/resolv.conf; this file is not managed by nmcli so to perform persistent (if dhcp is enabled on a connection) changes using nmcli or updating the /etc/sysconfig/network-scripts connection configuration file is required.
- nmcli con mod "connection id" ipv4.dns IP
- nmcli con mod "connection id" +ipv4.dns ip
    - The above command will add instead of replace

- nmcli con mod “connection id” -ipv4.dns ip

Note: When DNS is added to a specific connection, that connection will first look at the DNS servers on the connection configuration file and then it will look at /etc/resolv.conf.

File Access Control Lists

File access control lists (ACLs) are intended to give finer grained control over specific file permissions. Named users and named groups that have a UID and GID can take advantage or be assigned to ACLs. ACLs are added in addition to the regular permissions already existent on a file, owner/group owner/other. One issue with ACLs is the file system has to support them and it has to be mounted with ACL option enabled.

Note: When ACLs are set on a file or directory using the chmod command, it only updates the masks and not the permissions.

When viewing a file’s permissions, if the permissions end with a +, this represents ACLs being set on the directory or file.

Extended ACL Entries

Extended ACL entries are those that contain named group or named users or “more than the minimum ACLs”.

Base/minimum ACL Entries

Base/minimum ACL entries are the original ACL entries on a file that contain ACL entries for the owner, group, and other.

File systems must be enabled and mounted with ACL support in order for ACLs to work. By default the XFS and EXT4 file systems ON RED HAT 7 have ACL support enabled.

- Set the named group permissions to rw for file file
    - **setfacl -m g:namedgroup:rw file**
- Set the named user permissions to rw for file file
    - **setfacl -m u:nameduser:rw file**
- Set the user owner permissions to rw for a file
    - **setfacl -m u::rw file**
- Set the group owner permission to rw for a file
    - **setfacl -m g::rw file**

- Set “other” permissions on a file
  - **setfacl -m o::rw file**
- In order to remove permissions on a file, you can denote - .
- Multiple ACL entries can be specified by separating the entries with a comma
  - **setfacl -m o::rw,u::rw file**

Removing ACLs

- Remove a named group entry from a file’s ACL
  - **setfacl -x g:groupname file**
  - **setfacl --remove-all <file/dir>**
  - **-R for recursive**

Copying ACLs from one file to another

- getfacl file1 | setfacl --set-file=- file2
  - Take the ACL from file1 as standard input for setfacl command. The - at the end of --set-file=- represents the use of standard input (stdin).

Setting Default ACLs

Directories can have default ACLs that files will inherit when new files are created inside of the directory. Default ACLs on a directory ONLY provide support for inheritance; they are not the ACLs that are enforced on a directory. Thus, you will also have to set regular base/extended ACLs on the directory.

- Set a default named user on directory “dir1”
  - **setfacl -m d:u:nameduser:rx dir1**
  - Notice the d for “default”; all else is the same when setting default permissions.
- Delete all default ACLs on a directory
  - **setfacl -x d:u:named directory**
  - **setfacl --remove-default directory**

Working with the ACL mask

An ACL mask sets the maximum level of ACL permissions allowed on a file or directory. If there are ACL entries on that file or directory that exceed the mask, they are “masked”. If permissions do not exceed the mask, no action is taken.

- Setting a mask on a file
  - setfacl -m m::rx file
  - This will remove write access from all groups and all named users
  - Essentially, you are setting the "maximum" permissions available
  - *Note: The mask should be set only after all ACL permissions are set. The mask will be reset when using chmod or after modifying the ACL.*
- *Note: the X (upper case x), when used for setting permissions, instructs execute permissions to be set on directories but not on files.*

Troubleshooting Permissions

- chmod will change the mask on ACLs when used
- Default ACLs are set on a directory but they don't "work" on the directory (defaults are only for inheritance)
- Directories need to have execute permissions in order to change into them or list contents. An X (uppercase x) will make sure execute permissions are applied ONLY to directories.
- Work as the proper user who owns a file
- set-uid or set-gid
- Masks should only be set after the ACL is set because the mask is recalculated
- cp command does not preserve ACL permissions
- mv command DOES preserve the ACL permissions
- ACLs use the GID or UID to set permissions. Even if the name is used, it still maps to the gid.uid. If the id changes for the group or user, the ACLs are not automatically updated.
- Execute permissions on a directory and not a file

Managing YUM Repositories

- **yum repolist**
  - List the repository ID, name, and number of packages that are available for each **enabled** repository on the system
- **yum repolist -v**
  - List more information about the repositories
- **yum repoinfo**
  - Show information about all enabled and disabled repositories
- **yum repolist all**
  - List all repositories active and not active

- /ec/yum.repos.d/Red Hat.repo

- **yum-config-manager**
  - **yum-config-manager** --enable reponame
  - **yum-config-manager** --disable reponame
  - **yum-config-manager** --add-repo repourl

CIFS/NFS

CIFS using samba

```
[root@localhost]# yum install sambaclient cifs-utils nfs-utils
```

```
[root@localhost]# mount -t cifs -o username=smbusername,password=smbpassword //serverip/share_name /mnt/mountlocation
```

[/etc/fstab persistent configuration]

```
//serverip/share_name /mnt/mountlocation cifs username=smbusername,password=smbpassword 0 0
```

NFS

```
[root@localhost]# yum install -y nfs-utils
```

```
[root@localhost]# mount -t nfs serverip:/mountlocation /mnt/mountlocation
```

[/etc/fstab persistent configuration]

```
serverip:/mountlocation /mnt/mountlocation nfs defaults 0 0
```

Connecting to a SSO LDAP/AD Server

Assuming FQDN is ad.linuxacademy.com

First, install the realmd package:

1. # **yum -y install realmd**
2. **# realm discover ad.linuxacademy.com**

```
ad.linuxacademy.com
type: kerberos
realm-name: AD.LINUXACADEMY.COM
domain-name: ad.linuxacademy.com
configured: no
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
```

```
  required-package: adcli
  required-package: samba-common
```

Now we will install the packages required from the output

1. # **yum –y install oddjob oddjob-mkhomedir sssd adcli samba-common**

Now join the domain

2. r**ealm join ad.linuxacademy.com**

   If the password is entered correctly, you will not have any return and you have joined the domain.

Now we will see the status of being joined to the domain:

```
# realm discover ad.linuxacademy.com
   ad.linuxacademy.com
    type: kerberos
     realm-name: AD.LINUXACADEMY.COM
   domain-name: ad.linuxacademy.com
   configured: kerberos-member
   server-software: active-directory
   client-software: sssd
   required-package: oddjob
   required-package: oddjob-mkhomedir
   required-package: sssd
   required-package: adcli
   required-package: samba-common
   login-formats: %U@ad.linuxacademy.com
  login-policy: allow-realm-logins
```

Before we can remotely login to our RHEL server using an existing AD credential through SSH, we need to make sure the following items in our SSH config file allow this:

1. # vi /etc/ssh/sshd_config

```
# Kerberos options
KerberosAuthentication yes
KerberosOrLocalPasswd yes
KerberosTicketCleanup yes
KerberosGetAFSToken yes
KerberosUseKuserok yes

# GSSAPI options
 GSSAPIAuthentication yes
```

```
GSSAPICleanupCredentials yes
```

Final Tip: The best thing you can do is practice hands-on and become familiar with the objectives. The second best thing you can do is to learn how to “find” documentation and “help” within the Linux system. Sometimes it requires searching and installing additional man pages and sometimes it just comes down to knowing how to look. However, there is A LOT of information and documentation but it’s not always right in front of you.

Use whatis, /usr/share/doc, rpm -qd (query documentation), rpm -ql, info, man, yum install man-pages (extra man pages) and of course install the extra SELinux man pages selinux-policy-devel.