Linux Academy

Study Guide

Linux Foundation Certified Engineer

Contents

Related Courses

Linux Foundation Certified Systems Engineer

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Configure Network Services to Start Automatically at Boot

- **Sysvinit (older service management, CentOS/RHEL 6.x, Ubuntu/Debian prior to latest)**
  - » Installing an example server service:

    ```
    yum install openssh-server
    ```

  - » Enabling the service to start on reboot:

    ```
    chkconfig openssh-server
    ```

  - » Starting the service in current session:

    ```
    service openssh-server start
    ```

- **Systemd (modern service management, all recent distributions)**
  - » Installing an example server service:

    ```
    yum install openssh-server
    ```

  - » Enabling the service to start on reboot:

    ```
    systemctl enable openssh-server
    ```

  - » Starting the service in current session:

    ```
    systemctl start openssh-server
    ```

  - » Querying the status of a service (running or otherwise):

    ```
    systemctl status openssh-server
    ```

Implement Packet Filtering

- IPTABLES firewall
- Listing the existing rules:

  ```
  sudo iptables -L
  ```

- Policies generally available:
  - » ACCEPT – lets the packet through
  - » DROP – drops the packet quietly
  - » REJECT – rejects the packet and returns a message to the requestor
- General format for IPTABLES rules:

```
iptables -A name_of_chain criteria_to_meet -j target_of_rule
```

» `-A` – append the rule to the end of the chain

» `Name_of_chain` – one of INPUT, OUTPUT or FORWARD

» `Criteria_to_meet` – the conditions against which all packets are inspected against to determine whether the rule will apply

» `Target` – the action or policy to apply (ACCEPT, REJECT, or DROP)

- **Example**: Drop ICMP Ping requests between servers:

```
iptables -A INPUT -protocol icmp -in-interface enp0s3 -j DROP
```

» This will cause ping commands remotely to drop with no returned response

- **Example**: Reject ICMP Ping requests between servers:

```
iptables -A INPUT -protocol icmp -in-interface enp0s3 -j REJECT
```

» This will return a message to the requestor "*Destination Port Unreachable*"

» This can inadvertently expose that your IP address is valid; this setting should be used only internally when you want the client systems to know the port is filtered

Monitor Network Performance

- Socket connections

» Use the utility `ss` (replacement for netstat)

- "socket statistics"

» Show all TCP ports open on a server:

```
ss -t -a
```

- `-t` – all tcp ports
- `-a` – all connections

» Show established connections with their timers:

```
ss -t -o
```

- `-t` – all tcp ports
- `-o` – time established

» Filtering by socket:

```
ss -tn sport = :22
```

- `sport = :22` – source port of the established connection

- Identify open ports and active hosts:
  - » Use the `nmap` utility (defensive scanning of your own network)
  - » Scan ports on the system or remote host:

```
nmap -A -sS [IP/Hostname]
```

- `-A` – deep scan for all discoverable ports and services
- `-sS` – use TCP SYN (prevents leaving a logged footprint on the remove system)

- Monitor all IP Traffic
  - » Utility to use: `iptraf`
    - Ncurses-based, showing all packets across all interfaces (local, physical, virtual)
  - » `dstat`
    - Shows second by second monitoring statistics on your system by process, including read/write/pids and system usage

Produce and Deliver Reports on System Use, Outages and User Requests

- CPU Utilization Statistics:

```
top
```

- Terminal-based listing of all user and system processes, resources allocated with each, by user – also provides overall memory utilization and resource load
- LOAD: simply defined as the number of "fully engaged" CPU/cores

  - » **Example**: Server1 has 4 CPUs (2 physical sockets, 2 cores each)
    - Load is reported as 2.78
    - Translates to 2.78 fully engaged processor cores, so a little more than one core out of the four remains for additional work
    - Reported in 1,5,15 minute increments at the top

```
htop
```

- Provides a cleaner ncurses based view of the system, contains the SAME information as

the `top` process

- Generally needs to be installed:

```
yum install htop
```

- Memory Utilization Statistics:

```
free -m
```

- `-m` – provides a human readable formatted listing of physical memory, swap memory and cache utilization (file and memory cache for paging)

- Disk Utilization:

```
df -h
```

- `-h` – disk space allocated and in use, human readable format, by filesystem/disk mount, in percent and mount location

```
df -hTi
```

- `-hTi` – display the inodes allocated and in use, human readable format, by filesystem/disk mount, in percent used and mount location

- **NOTE**: Fileshares (remote filesystems) inodes cannot be accurately read by local `df` command inode listing

» File system utilization, by file and/or directory:

```
du -sh [/directory/mount]
```

- `-sch` – human readable format, summarized by directory

- Omit the parameters to get a full file by file listing in every directory and filesystem starting with the passed in parameter

- Process Management and Reporting:

```
ps ef
```

- Process listing

```
ps aux
```

- Show ALL system processes

» Count processes related to HTTP server, not including the grep command match:

```
ps aux | grep http | grep -v grep | wc -l
```

- `aux` – everything for HTTP on the system

        - `grep http` – find only processes containing http
        - `grep -v grep` – remove the grep line match
        - `wc -l` – count the results

- Auditing Logs
    - » dmesg
        - Information logged during the boot process (boot order, drivers, IP addresses, kernel parameters, CPU information, last reboot, etc)
    - » httpd
        - Default location for access and error logs
    - » yum
        - Installation logs
    - » Messages
        - What dmesg reads from to display the boot log
    - » Xorg
        - X Windows logging
    - » Security
        - User logins/interactions

Configure Systems to Mount Standard, Encrypted and Network File Systems on Demand

- Standard File Systems
    - » Partitioning:

      `fdisk/gdisk /dev/drive`

        - Allows us to create a partition from the available indicated disk
        - Can set type of partition (GPT or MBT depending on use), list the partition types in the menu
    - » Format the filesystem partition created:

```
mkfs -t ext4 /dev/disk1
```

- Formats the disk partition *disk1* as an ext4 partition

» Mount the formatted partition:

```
mkdir /mnt/mount
mount -t ext4 /dev/disk1 /mnt/mount
```

- Will mount the formatted drive, as an EXT4 mount on the created directory

» Persistently mounting the disk above

- Obtain the UUID of the device:

```
ls -al /dev/disk/by-uuid
```

- Add entry to `/etc/fstab` like so:

```
UUID=UUDI_OBTAINED    /mnt/mount     ext4      defaults     0      0
```

- Mount the defined entry automatically if not mounted in current session:

```
mount -a
```

- This will scan all defined mount points, if not mounted, mount them using the definitions in `/etc/fstab`

- Encrypted File Systems

» System support encrypted file system query:

```
grep -I config_dm_crypt /boot/config-$(uname -r)
```

» Determine if module is loaded:

```
lsmod | grep dm_crypt
```

» Load module if needed:

```
modprobe dm_crypt
```

» Partitioning

- Handled the same way as a typical disk and drive as defined above in the *Standard File Systems* section

» Install the cryptsetup utility:

```
yum install cryptsetup
```

» Default encryption key setup

- Luks – Linux Unified Key Setup

» Set up the partitions with passphrase:

```
cryptsetup -y luksformat /dev/disk1
```

- Prompted for passphrase for unencrypting drive during mount/use
- Large drives will take a long time to encrypt

» Open partition for use:

```
cryptsetup luksOpen /dev/disk1 reference_name
```

» Format the filesystem partition created, using the mapper overlay created above

- Handled the same way as a typical disk and drive as defined above in the "Standard File Systems" section
- Example – `mkfs -t ext4 /dev/mapper/reference_name`

» Mount the drive:

```
mount /dev/mapper/reference_name /mnt/mount
```

» Close the partition once used and unmounted:

```
cryptsetup luksClose reference_name
```

Provide/Configure Network Shares via NFS (Client and Server Setup)

- Installing the server packages (Server Setup):

```
yum install nfs-utils
```

- Create a directory for the shared or exported filesystem:

```
mkdir /var/myshare
```

- Change the permissions to be sure everyone has access:

```
chmod -R 777 /var/myshare
```

- Enable and start the appropriate services:

```
rpcbind
nfs-server
nfs-lock
nfs-idmap
```

- Create/edit the `/etc/exports` file, add the filesystem with appropriate options

```
/var/myshare            192.168.1.0/24(rw,sync,no_root_squash,no_all_squash)
```

  - `rw` – read/write filesystem
  - `sync` – keep local and remote filesystem caches in sync
  - `no_root_squash, no_all_squash` – don't attempt to do root or user level remapping based on UIDs for security

- Start the NFS Server service:

```
systemctl restart nfs-server
```

- Installing the packages (Client Setup):

```
yum install nfs-utils
```

- Create a directory to mount the share:

```
mkdir /mnt/shareddrive
```

- Enable and start the appropriate services:
  - » rpcbind
  - » nfs-server
  - » nfs-lock
  - » nfs-idmap
- Command line mount example:

```
mount -t nfs SERVERIP:/var/myshare /mnt/shareddrive
```

- Persistent mounting in `/etc/fstab` example:

```
SERVERIP:/var/myshare       /mnt/shareddrive      nfs       default     0     0
```

- Can then be mounted automatically on boot or on demand with:

```
mount -a
```

Update Packages from the Network, a Repository or the Local File System

- Local upgrade of system and system packages:

```
yum update/upgrade
```

- Will update all packages installed on the system

```
yum update/upgrade package
```

- Will update just the package indicated on the command line

- Local Repository (mirror example):

» Install `createrepo`

```
yum install createrepo
```

» Create a directory to house the mirror (or local repository)

```
createrepo /var/www/html/repos/centos/7
```

» The initial repository created will be empty, you can then add whatever files you want and update the repo; in this case, we are going to mirror a full online repo and then update, synchronizing our repo is something like:

```
rsync -avz rsync://some.repo.url/centos/7/os/x86_64/ /var/www/html/repos/7/
```

» The download in this case will take some time and can be verified with `du -sch` on the `/var/www/html` directory

» Update the repo:

```
createrepo -update /var/www/html/repos/centos/7
```

- Subscribe client to the repo

» Create file *local.repo* in `/etc/yum.repos.d`, example below:

```
[localrepo]
name=Sample local repository for CentOS 7
baseurl=http://SERVERIP/repos/centos/7/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-7
```

» Update the client:

```
yum update
```

» Verify installation and subscription:

```
yum info package
```

- The package can be any package not installed, the info should clearly state the “Repo: localrepo” or whatever was indicated in the local.repo file as the source for install

Configure Apache Log Files

- Log format contained, by default, in the *httpd.conf* file and looks like (for example):

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

- Can always be overridden within a specific vhost, using the same directive and format
- Format options are:
  - `%a` – remote client IP
  - `%A` – local client IP
  - `%b` – number of bytes transmitted
  - `%B` – number of bytes transmitted
  - `%{var}e` – value of environment variable indicated
  - `%f` – requested file
  - `%h` – remote host name (or IP if reverse lookup fails)
  - `%H` – requested protocol
  - `%{var}I` – contents of the header line name (User-agent)i for example
  - `%l` – remote log name
  - `%m` – request method
  - `%{var}n` – contents of the note named
  - `%{var}o` – contents of the header named
  - `%p` – canonical port that serviced the request
  - `%P` – process ID that serviced the request
  - `%q` – query string or search argument used (prefaced with "?")
  - `%r` – first line of the request
  - `%s` – server response status (i.e. HTTP 200 OK)
  - `%t` – time in common log formatting
  - `%T` – time (seconds) taken to serve the content
  - `%u` – named of authenticated user (if any)
  - `%U` – requested URL path

  - » `%v` – server name servicing the request
  - » `%V` – server name according to UseCanonicalName setting
- Reference names are given to custom logging formats (like *common* in the example above)
- ErrorLog and AccessLog directives can then refer to the reference names (when configured in *httpd.conf* with one) rather than overriding the LogFormat directive, within a vhost:
  - Example:

```
ErrorLog logs/mysite-error.log reference_name
```

  - Would format the site error log file, using the 'reference_name' format as defined in httpd.conf

Configure the Firewall with IPTables

- IPTABLES firewall
- Listing the existing rules:

```
sudo iptables -L
```

- Policies generally available:
  - » ACCEPT – lets the packet through
  - » DROP – drops the packet quietly
  - » REJECT – rejects the packet and returns a message to the requestor
- General format for IPTABLES rules:

```
iptables -A name_of_chain criteria_to_meet -j target_of_rule
```

  - `-A` – append the rule to the end of the chain
  - `name_of_chain` – one of INPUT, OUTPUT or FORWARD
  - `criteria_to_meet` – the conditions against which all packets are inspected against to determine whether the rule will apply
  - `target` – the action or policy to apply (ACCEPT, REJECT, or DROP)
- **Example**: Drop ICMP Ping requests between servers:

```
iptables -A INPUT -protocol icmp -in-interface enp0s3 -j DROP
```

  - This will cause ping commands remotely to drop with no returned response
- **Example**: Reject ICMP Ping requests between servers:

```
iptables -A INPUT -protocol icmp -in-interface enp0s3 -j REJECT
```

- This will return a message to the requestor "*Destination Port Unreachable*"
- This can inadvertently expose that your IP address is valid, this setting should be used only internally when you want the client systems to know the port is filtered

Configuring SSH-Based Remote Access Using Public/Private Key Pairs

- Generate a public/private key pair for SSH key exchange utilization:

```
ssh-keygen
```

  - Passphrase prompt is optional, designed to add an additional security layer (i.e. both the key AND passphrase would then be required when connecting to a server in this manner)

- Copy the public key from a user to a remote host (note that the referenced account on the remote host must already exist):

```
ssh-copy-id user@[servername/serverip]
```

  - Will be prompted for remote user password the first time
  - Will not connect the session on the key copy, copies the key only

- Testing

```
ssh user@[servername/serverip]
```

  - If done correctly, will prompt for passphrase (if one was entered) or will simply connect you to the remote host as the indicated user if no passphrase was entered during key creation

Install and Configure SSL with Apache (Keys and Certificates)

- Install the `openssl` utility:

```
yum install openssl
apt-get install openssl
```

- Generate a certificate key file (standard example):

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /path/to/cert/dir/
site.key
```

  - Generates a key for a certificate authority to generate a valid certificate expiring one year from the date of creation with industry standard NoDES, x509 RSA 2048 bit encryption

- Generate a certificate key file and use it to generate a self signed certificate for local use

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /path/to/cert/dir/
site.key -out /path/to/cert/dir/site.crt
```

  - Takes the key created and immediately signs it locally with the information provided during certificate creation

Install and Configure SSL with Apache (Server Configuration and VHost Directives)

- Install `mod_ssl`:

```
yum install mod_ssl
```

- Activate `mod_ssl`:

  » CentOS/RHEL – restart Apache after `mod_ssl` installation:

```
systemctl restart httpd
```

  » Debian/Ubuntu – enable the module and restart Apache

```
a2enmod mod_ssl
systemctl restart apache2
```

- VHost directives needed:

```
<VirtualHost *:443>
```

  - Enables port 443 for current vhost

```
SSLEngine on
```

  - Enables SSL for the current vhost

```
SSLCertificateKeyFile /path/to/cert/dir/site.key
```

  - Whatever path that the created key file exists in

```
SSLCertificateFil /path/to/cert/dir/site.crt
```

  - Whatever path that the install certificate file exists in

- Restart Apache once VHosts are created to activate them on the server

Configure Email Aliases, Install and Configure SMTP/IMAP/IMAPS

- Packages to install:
  - » dovecot
  - » postfix
    - CentOS/RHEL – yum install dovecot postfix
    - Debian/Ubuntu – apt-get install dovecot postfix
- Add email aliases
  - » `/etc/postfix/aliases`

    ```
    email_account: alias_name1, alias_name2
    ```

    - More than one alias can be added for each account to alias
- Refresh the alias table:

  ```
  postalias /etc/postfix/aliases
  ```
- Configure postfix
  - » `/etc/postfix/main.cf`

    ```
    myorigin = /etc/mailname (or domain itself)
    ```

    - The file that contains the domain name of the server
    - Can contain the domain name directly, filename is older standard still in use
  - » mydestination = list of domains the mail server will deliver messages to locally instead of forwarding to another system/mail server

    ```
    mydestination = myserver.domain.com, localhost.domain.com, localhost (examples)

    mynetworks = subnet
    ```

    - Indicates that we are serving IPs in the local subnet the server exists on

    ```
    inet_interfaces = all
    ```

    - Accept connections and messages to/from all defined network interfaces (localhost, physical, virtual, other)

    ```
    mailbox_size_limit = #####
    message_size_limit = #####
    ```

- Self explanatory, can be set to whatever requirements needed, in bytes

» `/etc/postfix/transport`

- Contains definitions of relationships between domains and the next server that mail messages need to be forwarded to:

```
example.domain.com       local:

.example.comain.com       local:
```

» Process the transport file and create/update the values to the mail DB format:

```
postmap /etc/postfix/transport
```

- Restrict access to SMTP

» `/etc/postfix/main.cf`

```
smtpd_helo_required = yes
```

- Require mail client introduce the mail transaction with standard HELO identification

```
smtpd_helo_restsrictions = permit_mynetworks, reject_invalid_helo_hostname
```

- Permits only defined networks (see above) and hosts that identify with HELO appropriately

```
smtpd_sender_restrictions = permit_mynetworks, reject_unknown_sender_domain
```

- Only accepts email to be sent from defined networks and rejects all others from unknown domains

```
smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination
```

- Accept only locally defined user destinations (see above configuration) and reject any unauthorized

- Enable and restart the postfix service:

```
systemctl enable postfix
systemctl restart postfix
```

- Configure Dovecot

» Initial installation supports IMAP and POP3, need to add IMAPS:

- `/etc/dovecot/dovecot.conf`

```
protocols = pop3 imap imap3
```

- Test listening on available ports:

```
netstat -npltu | grep dovecot
```

- Should see entries for ports 110 (pop3), 143 (imap), 993 (imaps) and 995 (pop3 secure)

- Use local `mail` subsystem to test:

```
mail email_account
```

- This is our alias account from earlier in the configuration, should go to two users as defined in the /etc/postfix/aliases file

» Enter subject, message and then exit to send

» Log in as either of the alias users, email should be available for them to read using local 'mail' command