LFCE Study Guide

Study Guide

Rob Marti
rob@linuxacademy.com
Feb 24, 2020

Contents

Useful Commands

Configuring Network Services to Start Automatically at Boot

- `yum install httpd`
  Install an example server service

- `systemctl enable httpd`
  Enable the service to start on reboot

- `systemctl start httpd`
  Start the service in current session

- `systemctl status httpd`
  Query the status of a service (running or otherwise)

Parallel SSH (pssh)

- `pssh -h hosts.txt -A -i “hostname”`
    - `-h`: Indicates a file with a list of hosts in it
    - `-A`: Prompts for a password (or passphrase for a key)
    - `-i`: Displays standard output and standard error as the command completes on each host

Implement Packet Filtering – firewalld

- `firewall-cmd --list-all-zones`
  Displays the configuration of all zones

- `firewall-cmd --get-default-zone`
  Displays the currently configured default zone

- `firewall-cmd --list-services`
  Lists all services that are currently configured to be open

- `firewall-cmd --list-ports`
  Lists all ports that are currently configured to be open

- `firewall-cmd --zone=public --add-service=http --permanent`
  Adds http as a service to the permanent configuration

- `firewall-cmd --reload`
  Reloads the firewall to bring the permanent configuration to the live, in memory, configuration

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

1. Generate a public/private key pair for SSH key exchange utilization:
   `ssh-keygen`

2. The passphrase prompt is optional, and is designed to add an additional security layer:
   - Both the key *and* passphrase would then be required when connecting to a server in this manner

3. 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]`

4. You will be prompted for remote user password the first time
5. It will not connect the session on the key copy, but just copies the key
6. Test with: `ssh user@[servername/serverip]`
7. If it's done correctly, one of two things will happen:
    - There will be a passphrase prompt (if one was entered in the first place)
    - You'll simply connect to the remote host as the indicated user if no passphrase was entered during key creation

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

CentOS

- Local upgrade of system and system packages:
  - `yum update/upgrade`
    Updates all packages installed on the system
  - `yum update/upgrade package`
    Updates just the package indicated on the command line

Ubuntu

- Local upgrade of system and system packages:
  - `apt update && apt upgrade`
    Upgrades all packages installed on the system
  - `apt install --only-upgrade packagename`
    Upgrades that specific package

Storage Management

LVM – Logical Volume Manager

- `pvcreate /dev/disk`
  Labels a device as usable by LVM

- `vgcreate volumename /dev/disk`
  Creates a volume group out of physical devices
- `lvcreate -L 10G volumename`
  - `-L` specifies the size of the LV
  - Use extents instead with `-l`
  - Designate a percentage of free space using 100%FREE rather than a specific size:

    Create a logical volume that is a subset of the volume group, but can take up all space allocated to the volume group if needed:
    `lvcreate -L 100%FREE volumename`

Block devices

- `lsblk` Used for looking at all block devices on the system
- `blkid` Search by label or display information about installed filesystems

Remote block devices – iSCSI

iSCSI has some terminology that takes some getting used to. The iSCSI server is called the *target* while the server that mounts the iSCSI device is called the *initiator*.

Configuring the target:

- `yum install targetcli`
  Installs the required package
- `targetcli`
  Runs the iscsi target configuration tool
- `backstores/block/create newdevice /dev/devicename`
  Tells the iSCSI Target software that we're creating a new device with a physical device as the back store.
- `iscsi/create iqn.2018-11.com.mylabserver:t1`
  Create an IQN (iSCSI Qualified Name)
- `cd iqn.2018-11.com.mylabserver:t1`
  The configuration software can be navigated like a directory. We switch into the directory to continue configuring the iSCSI Target
- `luns/ create /backstores/block/newdevice`
  Create a LUN on this target backed by the backstory we create earlier
- `acls/ create iqn.2018-11.com.mylabserver:client`
  Set up what initiators are allowed to connect to this target.

We can then exit out of that and make sure that the 'target' service is set to start up:

- `systemctl enable target`
- `systemctl start target`

On the Initiator (Client):

- `yum install iscsi-initiator-utils -y`
  Installs the required software

- Edit `/etc/iscsi/initiatorname` and set an `InitiatorName`:
  `InitiatorName=iqn.2018-11.com.mylabserver:client`
  This is the IQN that was used when creating the ACL on the Target

- Start the iscsi service:
  `systemctl enable iscsi`
  `systemctl start iscsi`

Now we need to discover the target. For that we use the following command:
`iscsiadm -m discovered -t st -p IP.ADDR.OF.TARGET -D`

- `-m`: Set the mode to discovery database
- `-t`: Set the discovery type to sendtargets (st)
- `-p`: Portal address
- `-D`: discover shared storage LUNs

Once that runs successfully, we can run the following to set the `Target` up as a disk you can use as normal:
`iscsiadm -m node -T iqn.2018-11.com.mylabserver:client -l`

- `-m`: Set the mode to node
- `-T`: Target IQN
- `-l`: Attempt to log in to that IQN

The disk should be visible in `fdisk` now.

Useful Network Commands

Socket connections

Use the `ss` utility (a replacement for `netstat`), which stands for *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`
  - `-n`: Show numbers (ports or IP addresses) instead of trying to resolve hostnames or service names
  - `sport = :22`: Source port of the established connection

Identifying open ports and active hosts

Use the `nmap` utility for 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 remote system)

IPTraf: Monitor Network Traffic

`iptraf-ng`

- Can be used interactively or programmatically

  ```
  iptraf -i all -t 1 -B
  ```

  - `-i`: Start the IP Traffic Monitor on named interfaces (or all)
  - `-t`: How much time to run the scan
  - `-B`: Forks the command into the background

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

CPU Utilization Statistics:

- `top`
  - Terminal-based listing of all user and system processes, and resources allocated to each
  - Also provides overall memory utilization and resource load
  - `LOAD`: simply defined as the number of processes waiting on either CPU or I/O time. For example:

    Load is reported as 2.78

    A load of 2.78 means that over the last reporting period, an average of 2.78 processes were waiting for a resource.

    Reported in 1, 5, and 15 minute increments, above (in the upper right corner) all of the other information in the terminal
- `htop`
  - Provides a cleaner ncurses based view of the system
  - Contains the *same* information as `top`
  - 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`
  - `df`: Disk space allocated and in use by filesystem/disk mount, with space usage percentage, and mount location

    `-h`: In human-readable format (10M vs 10000B)
- `df -hTi`
  - `-hTi`: Display the inodes allocated and in use, in human readable format, by filesystem/disk mount, with space usage percentage, 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]`
  - `-sh`: 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 [h]ttp | wc -l`
  - `aux`: Every process on the system
  - `grep [h]ttp`: Find only processes containing http excluding the grep line
  - `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 (/var/log/httpd/)
- `yum`
  Information about package installations and removals

Installation logs

- `messages`
  What dmesg reads from to display the boot log
  Also contains messages logged form all processes

- `Xorg`
  X Windows logging

- `secure`
  User logins/interactions

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

- Configure Postfix
    - Edit `/etc/postfix/main.cf` and make sure these are set correctly:

      `myorigin = hostname`

      `mydestination`: A list of domains the mail server will deliver messages to locally, instead of forwarding to another system/mail server. Examples are:

        `mydestination = myserver.domain.com`

        `mydestination = localhost.domain.com`

        `mydestination = localhost`

      `mynetworks = subnet` Indicates that we are serving IPs in the local subnet that the server exists on

      `inet_interfaces = all` Accepts 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