LFCSA

- LFCS: Operations & Deployment (25%)
- LFCS: Networking (25%)
- LFCS: Storage (20%)
- LFCS: Essential Commands (20%)
- LFCS: Users and Groups (10%)
- LFCS Mock Exam
- LFCS Mock Exam 2

LFCS: Operations & Deployment (25%)

This section covers **system administration tasks** related to configuring the kernel, managing processes and services, scheduling jobs, managing software, recovering systems, and working with virtualization, containers, and SELinux.

Goals

Operations Deployment 25%

1. Configure kernel parameters, persistent and non-persistent
2. Diagnose, identify, manage, and troubleshoot processes and services
3. Manage or schedule jobs for executing commands
4. Search for, install, validate, and maintain software packages or repositories
5. Recover from hardware, operating system, or filesystem failures
6. Manage Virtual Machines (libvirt)
7. Configure container engines, create and manage containers
8. Create and enforce MAC using SELinux

---

1. Configure Kernel Parameters

Learn

- Difference between **temporary (runtime)** and **persistent** kernel parameters.
- Tools: `sysctl`, `/etc/sysctl.conf`, and `/etc/sysctl.d/*.conf`.

Commands

```
# Show all parameters
sysctl -a

# Set a runtime parameter (non-persistent)
sudo sysctl net.ipv4.ip_forward=1

# Persist parameter across reboots
echo "net.ipv4.ip_forward = 1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p

# Modern persistent configuration (preferred)
echo "net.ipv4.ip_forward = 1" | sudo tee /etc/sysctl.d/99-custom.conf
sudo sysctl --system
```

Common sysctl Parameters to Know

Networking

IP forwarding

- `net.ipv4.ip_forward`
- `net.ipv6.conf.all.forwarding`

Reverse Path Filtering (security)

- `net.ipv4.conf.all.rp_filter`

Redirects / Source Routing (disable for security)

- `net.ipv4.conf.all.accept_redirects = 0`
- `net.ipv6.conf.all.accept_redirects = 0`
- `net.ipv4.conf.all.send_redirects = 0`
- `net.ipv4.conf.all.accept_source_route = 0`

TCP stack tuning

- `net.ipv4.tcp_syncookies = 1`
- `net.core.somaxconn`
- `net.core.netdev_max_backlog`
- `net.ipv4.tcp_fin_timeout`
- `net.ipv4.tcp_keepalive_*`

- `net.ipv4.ip_local_port_range`

Memory & VM

Swappiness

- `vm.swappiness`

Cache pressure

- `vm.vfs_cache_pressure`

Memory overcommit

- `vm.overcommit_memory`
- `vm.overcommit_ratio`

Kernel Stability

File handles

- `fs.file-max`

Kernel panic options

- `kernel.panic`
- `kernel.panic_on_oops`

Networking Security

Disable IPv6

- `net.ipv6.conf.all.disable_ipv6 = 1`

ARP filtering

- `net.ipv4.conf.all.arp_filter`

Logging & Debugging

printk log levels

- `kernel.printk`

Essential LFCS Knowledge

You must be able to: - View kernel parameters

- Modify runtime parameters
- Apply persistent configurations
- Reload sysctl configs

Most common parameters for the exam: - `net.ipv4.ip_forward` - `vm.swappiness` - `fs.file-max` - `kernel.hostname`

Exercises

1. Enable IP forwarding temporarily and verify:

   ```
   sysctl net.ipv4.ip_forward
   ```

2. Make IP forwarding permanent using `/etc/sysctl.d/`.

---

2. Manage Processes and Services

Learn

- Identify, monitor, and troubleshoot processes.
- Manage services with **systemd**.

Commands

```
# Process management
ps aux | grep <name>
top
htop        # if installed
kill <PID>
kill -9 <PID>

# Systemd service management
systemctl status <service>
```

```
systemctl start <service>
systemctl stop <service>
systemctl enable <service>
systemctl disable <service>
systemctl restart <service>
systemctl reload <service>

# View logs for a service
journalctl -u <service>

# List active services
systemctl list-units --type=service
```

Exercises

1. Find the PID of a running process and stop it gracefully.
2. Enable and start a service at boot time.
3. View service logs with `journalctl`.

---

3. Manage Scheduled Jobs

Learn

- Use **cron** for recurring tasks (user or system-level).
- Use **at** for one-time scheduled jobs.

Commands

```
# List user cron jobs
crontab -l

# Edit user cron jobs
crontab -e

# System-wide cron directories
```

```
ls /etc/cron.daily/
ls /etc/cron.weekly/

# At job example
echo "touch /tmp/testfile" | at now + 1 minute
atq        # list scheduled at jobs
atrm <job_number>
```

Exercises

1. Schedule a script to run every day at 3 AM.
2. Schedule a one-time task using `at`.
3. Check `/etc/cron.daily/` for system maintenance tasks.

---

4. Software Management

Learn

- Install, update, and remove software using your system’s package manager.
- Manage repositories and dependencies.

Commands

```
# Debian/Ubuntu
sudo apt update
sudo apt install <package>
sudo apt remove <package>
dpkg -l | grep <package>
dpkg -s <package>

# Manage repositories
sudo add-apt-repository ppa:<repo>
cat /etc/apt/sources.list.d/*.list

# RHEL/CentOS
```

```
sudo yum install <package>
sudo yum remove <package>
rpm -qa | grep <package>
rpm -qi <package>

# Repository management
yum repolist
```

Exercises

1. Install `curl` and verify installation.
2. Remove a package and confirm removal.
3. List all enabled repositories.

---

5. Recover from Hardware, OS, or Filesystem Failures

Learn

- Boot into rescue or recovery mode.
- Use Live CD/USB for system repair.
- Use `fsck` to check and repair filesystems.

Commands

```
# Filesystem check (unmounted partition)
sudo fsck /dev/sdX1

# Mount for inspection or repair
sudo mount /dev/sdX1 /mnt

# Kernel/hardware diagnostics
dmesg | less
journalctl -xb
```

```
# Disk health (if smartmontools installed)
sudo smartctl -a /dev/sdX
```

Exercises

1. Simulate a corrupted filesystem on a test partition and repair it using `fsck`.
2. Boot into rescue mode and inspect partitions with `lsblk`.
3. Review logs with `journalctl -xb`.

---

6. Virtual Machines (libvirt)

Learn

- Manage VMs using **libvirt** tools (`virsh`, `virt-install`).
- Control VM lifecycle and connect via console.

Commands

```
virsh list --all
virsh start <vm>
virsh shutdown <vm>
virsh destroy <vm>        # Force shutdown
virsh console <vm>

# Virtual network management
virsh net-list
virsh net-start default
```

Exercises

1. Create a VM using `virt-install`.
2. Start, stop, and connect to the VM via console.
3. Verify virtual networks using `virsh net-list`.

---

7. Containers (Docker / Podman)

Learn

- Create, start, stop, and manage containers and images.
- Understand container **networking** and **volumes**.

Commands

```
# Docker examples
docker run -it --name mycontainer ubuntu bash
docker ps
docker stop mycontainer
docker rm mycontainer
docker images
docker rmi <image>

# Volumes and networks
docker volume ls
docker network ls

# Podman equivalents (rootless alternative)
podman run -it --name mycontainer ubuntu bash
podman ps
podman images
podman rm -a
```

Exercises

1. Create an Ubuntu container and install `curl` inside it.
2. Commit the container as a new image:

   ```
   docker commit mycontainer ubuntu-with-curl
   ```

3. Remove the container and image.
4. List container networks and volumes.

---

8. SELinux (Mandatory Access Control)

Learn

- SELinux modes: **enforcing**, **permissive**, **disabled**.
- Troubleshoot SELinux denials and manage contexts.

Commands

```
# Check SELinux status
sestatus

# Change mode temporarily
sudo setenforce 0    # permissive
sudo setenforce 1    # enforcing

# Check logs for denials
ausearch -m avc

# Check and restore file contexts
ls -Z /path
restorecon -Rv /path

# Manage custom contexts
sudo semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"
sudo restorecon -Rv /web
```

Exercises

1. Switch SELinux to permissive mode and verify.
2. Trigger an SELinux denial (e.g., mislabel file) and view logs with `ausearch`.
3. Restore a file’s correct SELinux context.

□□ Exam Tips

- Practice all commands **without** `sudo`, as exam environments often run as root.
- Understand the difference between **runtime vs persistent** configurations:
  - Kernel parameters (`sysctl`)
  - SELinux modes
  - Service enablement
- Use **lab VMs** to safely experiment with:
  - Containers and virtual machines
  - Cron jobs and recovery procedures
- Review logs regularly (`journalctl`, `dmesg`, `ausearch`).

LFCS: Networking (25%)

This section covers **Linux networking**, hostname resolution, time synchronization, OpenSSH configuration, packet filtering, routing, bridges, and load balancing.

Goals

Networking 25%

1. Configure IPv4 and IPv6 networking and hostname resolution
2. Set and synchronize system time using time servers
3. Monitor and troubleshoot networking
4. Configure the OpenSSH server and client
5. Configure packet filtering, port redirection, and NAT
6. Configure static routing
7. Configure bridge and bonding devices
8. Implement reverse proxies and load balancers

---

1. Configure IPv4/IPv6 Networking & Hostname Resolution

Learn

- Assign IP addresses (static & dynamic).
- Configure hostnames and DNS resolution.
- Understand `/etc/hosts`, `/etc/resolv.conf`, and `hostnamectl`.

Commands

```
# View network interfaces
ip addr show
ip link show

# Configure static IP (example)
sudo nmcli con mod eth0 ipv4.addresses 192.168.1.100/24
sudo nmcli con mod eth0 ipv4.gateway 192.168.1.1
sudo nmcli con mod eth0 ipv4.dns 8.8.8.8
sudo nmcli con mod eth0 ipv4.method manual
sudo nmcli con up eth0

# Set hostname
hostnamectl set-hostname myhost
hostnamectl status

# DNS check
cat /etc/resolv.conf
ping google.com
```

Exercises

1. Assign a static IP to an interface and verify connectivity.
2. Change the hostname and update `/etc/hosts` accordingly.

---

2. Time Synchronization

Learn

- Sync system time using `chrony` or `ntpd`.
- Check and update time zone.

Commands

```
# Check current time
timedatectl
```

```
# Set time zone
timedatectl set-timezone Europe/Paris

# Sync time using chrony
sudo systemctl start chronyd
sudo systemctl enable chronyd
chronyc tracking
chronyc sources
```

Exercises

1. Configure NTP synchronization with a public server.
2. Verify time sync status using `chronyc tracking`.

---

3. Monitor & Troubleshoot Networking

Learn

- Use tools to check connectivity and troubleshoot issues.

Commands

```
ping <host>
traceroute <host>
ss -tulnp       # check listening ports
netstat -rn     # routing table
curl -I http://example.com
dig example.com # DNS query test
```

Exercises

1. Ping a remote host and check for packet loss.
2. Check which service is listening on port 80.
3. Use `traceroute` to identify network path latency.

---

4. OpenSSH Configuration

Learn

- Configure SSH server and client.
- Manage SSH keys and permissions.

Commands

```
# Start and enable SSH server
sudo systemctl start sshd
sudo systemctl enable sshd

# Connect to remote server
ssh user@remote_host

# Generate SSH key pair
ssh-keygen -t rsa -b 4096

# Copy SSH public key to remote server
ssh-copy-id user@remote_host

# Change SSH port (example: 2222)
sudo nano /etc/ssh/sshd_config
# Port 2222
sudo systemctl reload sshd
```

Exercises

1. Configure SSH to listen on a non-default port.
2. Set up key-based authentication.
3. Disable root password login for security.

5. Packet Filtering, Port Redirection, and NAT

Learn

- Configure firewalls using `iptables` or `firewalld`.
- Perform NAT and port forwarding for network access.

Commands

```
# Check firewall status
sudo firewall-cmd --state

# Allow SSH port 22
sudo firewall-cmd --add-port=22/tcp --permanent
sudo firewall-cmd --reload

# Add NAT rule (masquerade)
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# View rules
sudo iptables -t nat -L -n -v
```

Exercises

1. Open a specific port (e.g., 8080) on the firewall.
2. Configure port forwarding for SSH (e.g., 2222 → 22).
3. Enable masquerading to share internet from one interface to another.

---

6. Static Routing

Learn

- Add static routes to control network traffic paths.

Commands

```
# Add a static route
sudo ip route add 192.168.2.0/24 via 192.168.1.1 dev eth0

# Delete a static route
sudo ip route del 192.168.2.0/24

# Show routes
ip route show
```

Exercises

1. Add a static route to reach a remote network.
2. Verify connectivity via the static route using `ping` or `traceroute`.

---

7. Bridge & Bonding Devices

Learn

- Create network **bridges** for virtual machines or containers.
- Create **bonded interfaces** for redundancy or link aggregation.

Commands

```
# Create a bridge
sudo nmcli con add type bridge con-name br0 ifname br0
sudo nmcli con add type bridge-slave con-name eth0-br0 ifname eth0 master br0

# Create bonded interface
```

```
sudo nmcli con add type bond con-name bond0 ifname bond0 mode active-backup
sudo nmcli con add type bond-slave con-name eth1-bond0 ifname eth1 master bond0
sudo nmcli con add type bond-slave con-name eth2-bond0 ifname eth2 master bond0

# View connection status
nmcli con show
```

Exercises

1. Create a bridge interface and attach an Ethernet device.
2. Configure a bonded interface using two NICs in active-backup mode.

---

8. Reverse Proxies & Load Balancers

Learn

- Understand reverse proxies and load balancing concepts.
- Use **Nginx** as a basic reverse proxy or load balancer.

Commands

```
# Install and configure Nginx
sudo apt install nginx -y

# Nginx reverse proxy configuration example
sudo nano /etc/nginx/conf.d/reverse.conf

# Example config:
# server {
#     listen 80;
#     location / {
#         proxy_pass http://backend:8080;
```

```
#     }
# }

# Enable and reload Nginx
sudo systemctl enable nginx
sudo systemctl reload nginx
```

Exercises

1. Configure Nginx to proxy requests to a backend server.
2. Implement round-robin load balancing for multiple backend servers.
3. Test the setup using `curl` or a web browser.

□□ Exam Tips

- Always verify **connectivity and DNS resolution** (`ping`, `dig`, `nslookup`).
- Understand **network configuration files**:
  - `/etc/NetworkManager/system-connections/`
  - `/etc/sysconfig/network-scripts/` (RHEL-based)
- Practice switching between **NetworkManager CLI (`nmcli`)** and `ip` commands.
- Familiarize yourself with **firewalld zones** and **NAT configurations**.
- Understand **basic Nginx directives** for proxying and load balancing.

LFCS: Storage (20%)

This section covers managing **LVM**, filesystems, remote storage, swap, automount, and monitoring storage performance.

Goals

Storage 20%

1. Configure and manage LVM storage
2. Manage and configure the virtual file system
3. Create, manage, and troubleshoot filesystems
4. Use remote filesystems and network block devices
5. Configure and manage swap space
6. Configure filesystem automounters
7. Monitor storage performance

---

1. Configure and Manage LVM Storage

Learn

- Logical Volume Manager (LVM) basics: **PV**, **VG**, **LV**.
- Create, extend, and reduce logical volumes.

Commands

```
# Create physical volume
sudo pvcreate /dev/sdb
```

```
# Create volume group
sudo vgcreate vg01 /dev/sdb

# Create logical volume
sudo lvcreate -L 10G -n lv01 vg01

# Extend logical volume
sudo lvextend -L +5G /dev/vg01/lv01
sudo resize2fs /dev/vg01/lv01

# Reduce logical volume (careful)
sudo umount /mnt
sudo e2fsck -f /dev/vg01/lv01
sudo resize2fs /dev/vg01/lv01 5G
sudo lvreduce -L 5G /dev/vg01/lv01
```

Exercises

1. Create a new LVM on a spare disk.
2. Extend the logical volume and verify the new size.

---

2. Manage and Configure the Virtual Filesystem

Learn

- Mount and unmount filesystems.
- Configure persistent mounts using `/etc/fstab`.

Commands

```
# Mount filesystem
sudo mount /dev/sdb1 /mnt

# Unmount filesystem
sudo umount /mnt

# Add to /etc/fstab for persistent mounting
/dev/sdb1 /mnt ext4 defaults 0 2
```

Exercises

1. Mount a new disk temporarily.
2. Add it to `/etc/fstab` for persistent mounting.

---

3. Create, Manage, and Troubleshoot Filesystems

Learn

- Create and repair filesystems (ext4, xfs, btrfs).
- Diagnose filesystem errors with `fsck`.

Commands

```
# Create filesystem
sudo mkfs.ext4 /dev/sdb1
sudo mkfs.xfs /dev/sdb2

# Check filesystem integrity
sudo fsck /dev/sdb1

# Repair filesystem
sudo fsck -y /dev/sdb1
```

Exercises

1. Format a partition as ext4.
2. Simulate and repair a filesystem error.

---

4. Use Remote Filesystems and Network Block Devices

Learn

- Mount **NFS**, **CIFS**, or **iSCSI** shares.
- Manage remote storage connections.

Commands

```
# Mount NFS
sudo mount -t nfs server:/share /mnt

# Mount CIFS (SMB)
sudo mount -t cifs //server/share /mnt -o username=user,password=pass

# iSCSI discovery and login
sudo iscsiadm -m discovery -t st -p 192.168.1.100
sudo iscsiadm -m node -l
```

Exercises

1. Mount an NFS share.
2. Connect to an iSCSI target.

---

5. Configure and Manage Swap Space

Learn

- Add, remove, and monitor swap space.

Commands

```
# Create swap file
sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile

# Make swap permanent
echo "/swapfile none swap sw 0 0" | sudo tee -a /etc/fstab

# Monitor swap usage
swapon -s
free -h
```

Exercises

1. Create a swap file and activate it.
2. Remove the swap file and update `/etc/fstab`.

---

6. Configure Filesystem Automounters

Learn

- Configure **autofs** to mount filesystems automatically when accessed.

Commands

```
# Install autofs
sudo apt install autofs

# Configure auto.master
sudo nano /etc/auto.master
# Example entry:
# /mnt /etc/auto.nfs

# Restart autofs
sudo systemctl restart autofs
```

Exercises

1. Configure automount for an NFS share.
2. Test automount by accessing the directory.

---

7. Monitor Storage Performance

Learn

- Use tools to monitor disk I/O and performance.

Commands

```
# Display I/O statistics
iostat -x 2 3
```

```
# Show disk usage
df -h

# Disk usage by directory
du -sh /var/log/*

# Monitor disk activity in real-time
iotop -o
```

Exercises

1. Check disk I/O using `iostat`.
2. Identify large directories consuming storage space.

□□ Exam Tips

- Understand **LVM creation, resizing, and removal**.
- Know how to **mount local and remote filesystems permanently**.
- Be able to **create and manage swap**.
- Use **autofs** for automount configurations.
- Monitor storage performance and identify I/O bottlenecks.

LFCS: Essential Commands (20%)

This document covers essential Linux commands for Git, services, system performance, application constraints, disk troubleshooting, and SSL management.

Goals

Essential Commands 20%

- Basic Git Operations
- Create, configure, and troubleshoot services
- Monitor and troubleshoot system performance and services
- Determine application and service specific constraints
- Troubleshoot diskspace issues
- Work with SSL certificates

---

1. Basic Git Operations

Learn

- Clone repositories, check status, commit, and push.

Commands

```
# Clone repository
git clone <url>
```

```
# Check status
git status

# Add files and commit
git add <file>
git commit -m "message"

# Push to remote
git push origin main

# Pull latest changes
git pull
```

Exercises

1. Clone a public repository.
2. Make a change, commit it, and push to a forked repo.

---

2. Create, Configure, Troubleshoot Services

Learn

- Use `systemctl` to manage services.
- Check logs for troubleshooting.

Commands

```
# Start, stop, restart service
sudo systemctl start <service>
sudo systemctl stop <service>
sudo systemctl restart <service>
```

```
# Enable/disable at boot
sudo systemctl enable <service>
sudo systemctl disable <service>

# Check logs
journalctl -u <service>
```

Exercises

1. Enable and start a service.
2. Check the last 20 lines of the service log.

---

3. Monitor & Troubleshoot System Performance

Learn

- Check CPU, memory, disk usage.
- Identify resource-consuming processes.

Commands

```
# CPU and memory
top
htop
vmstat 2 5

# Disk usage
df -h
du -sh /path/to/dir

# Check running processes
ps aux | sort -nk 3 | tail -10
```

Exercises

1. Identify top 5 processes consuming CPU.
2. Check disk usage of `/var/log`.

---

4. Application & Service Constraints

Learn

- Resource limits for users or services.
- Configure ulimit.

Commands

```
# Check limits
ulimit -a

# Set temporary limit
ulimit -n 4096

# Persistent limits
sudo nano /etc/security/limits.conf
# user soft nofile 4096
# user hard nofile 8192
```

Exercises

1. Increase the open file limit for a user temporarily.
2. Make the change permanent.

---

5. Troubleshoot Diskspace Issues

Learn

- Find large files or directories.
- Free space using cleanup tools.

Commands

```
# Find large directories
du -ah / | sort -rh | head -20


# Remove old logs
sudo journalctl --vacuum-time=7d
```

Exercises

1. Identify top 10 largest files.
2. Clear old logs to free space.

---

6. Work with SSL Certificates

Learn

- Generate, verify, and check certificates.
- Use OpenSSL.

Commands

```
# Generate self-signed certificate
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout server.key -out server.crt
```

```
# Verify certificate
openssl x509 -in server.crt -text -noout

# Check SSL connection
openssl s_client -connect example.com:443
```

Exercises

1. Generate a self-signed certificate.
2. Inspect and verify the certificate using OpenSSL.

Exam Tips

- Practice using **systemctl, top, df, du** frequently.
- Know basic Git workflows.
- Be comfortable troubleshooting disk space and service issues.
- Understand SSL certificate basics with OpenSSL.

LFCS: Users and Groups (10%)

This document covers user and group management, environment profiles, ACLs, resource limits, and LDAP integration.

Goals Coverage

Users and Groups 10%

1. Create and manage local user and group accounts
2. Manage personal and system-wide environment profiles
3. Configure user resource limits
4. Configure and manage ACLs
5. Configure the system to use LDAP user and group accounts

---

1. Create and Manage Local User Accounts

Learn

- Add, modify, and delete local users.
- Manage home directories and shells.

Commands

```
# Add user
sudo useradd -m -s /bin/bash username
sudo passwd username

# Modify user
sudo usermod -aG sudo username   # Add to group
sudo usermod -d /new/home username  # Change home directory

# Delete user
sudo userdel -r username   # Delete user and home directory
```

Exercises

1. Create a user with home directory and bash shell.
2. Add the user to the sudo group.
3. Remove a test user.

---

2. Create and Manage Groups

Learn

- Create, modify, and delete groups.
- Assign users to groups.

Commands

```
# Create group
sudo groupadd developers

# Modify group
sudo gpasswd -a username developers

# Delete group
sudo groupdel developers
```

```
# List groups
groups username
```

Exercises

1. Create a group `admins` and add a user.
2. Verify the user is part of the group.
3. Delete the group.

---

3. Manage Environment Profiles

Learn

- System-wide vs user-specific profiles.
- Configure PATH, variables, and shell startup files.

Commands

```
# Edit system-wide profile
sudo nano /etc/profile
sudo nano /etc/bash.bashrc

# Edit user profile
nano ~/.bashrc
source ~/.bashrc  # Reload changes
```

Exercises

1. Add a directory to PATH in `.bashrc` and reload.
2. Set a custom environment variable system-wide.

---

4. Configure User Resource Limits

Learn

- Use `ulimit` and `/etc/security/limits.conf` for limits on processes, open files, and memory.

Commands

```
# Temporary limit
ulimit -n 4096

# Persistent limit
sudo nano /etc/security/limits.conf
# user soft nofile 4096
# user hard nofile 8192
```

Exercises

1. Increase the maximum number of open files for a user.
2. Verify limits with `ulimit -a`.

---

5. Configure and Manage ACLs (Access Control Lists)

Learn

- Set fine-grained permissions beyond standard Unix mode bits.
- Commands: `getfacl` and `setfacl`.

Commands

```
# Check ACL
getfacl /path/to/file
```

```
# Set ACL for user
setfacl -m u:username:rwx /path/to/file

# Remove ACL
setfacl -x u:username /path/to/file
```

Exercises

1. Give read/write access to a specific file for a user without changing group ownership.
2. Remove the ACL and verify permissions.

---

6. Configure the System to Use LDAP for Users and Groups

Learn

- Integrate Linux system with LDAP server for centralized authentication.
- Configure NSS and PAM.

Commands

```
# Install LDAP client
sudo apt install libnss-ldap libpam-ldap ldap-utils

# Configure /etc/nsswitch.conf
sudo nano /etc/nsswitch.conf
# passwd: files ldap
group: files ldap
shadow: files ldap

# Test LDAP user
getent passwd ldapuser
```

```
id ldapuser
```

Exercises

1. Connect a Linux system to an LDAP server.
2. Verify that LDAP users can login.

Exam Tips

- Practice creating users and groups both locally and via LDAP.
- Understand persistent vs temporary environment variable settings.
- Familiarize with ACL commands for fine-grained permission control.

LFCS Mock Exam

Generated by ChatGPT

This mock exam is designed to test your knowledge across all LFCS topics: Operations, Networking, Storage, Essential Commands, and Users & Groups.

Instructions

- Complete the exercises without looking at the answers.
- The answers are provided at the end of the document.

Part 1: Operations & Deployment

1. Configure a system to automatically load a kernel module `dummy_module` at boot.
2. Identify a device connected via PCI and check if a kernel module is loaded for it.
3. Create a cron job to run `/usr/local/bin/backup.sh` every day at 2:30 AM.
4. Check if the system is vulnerable to the Meltdown CPU flaw.

Part 2: Networking

1. Assign a static IP address `192.168.50.10/24` with gateway `192.168.50.1` to interface `eth0`.
2. Configure the system to synchronize time with `pool.ntp.org`.
3. Allow SSH connections on port 2222 through the firewall.
4. Add a static route to reach network `10.10.0.0/16` via gateway `192.168.50.254`.

Part 3: Storage

1. Create a 10 GB LVM logical volume named `lv_data` in volume group `vg01`.

2. Format it with `ext4` and mount it on `/mnt/data`.
3. Create a 2 GB swap file and enable it.
4. Mount an NFS share `server:/share` on `/mnt/nfs` automatically at boot.

Part 4: Essential Commands

1. Check which process is consuming the most CPU and kill it.
2. Find the 10 largest files in `/var/log`.
3. Generate a self-signed SSL certificate valid for 365 days.
4. Clone a Git repository and push a file named `README.md` after adding a new line.

Part 5: Users & Groups

1. Create a new user `developer1` with bash shell and home directory.
2. Add this user to a group `developers`.
3. Set the maximum number of open files for `developer1` to 4096.
4. Set ACL so that `developer1` has read/write access to `/var/www/html/index.html` without changing the file's group ownership.

Answers (Do not read until done!)

Part 1: Operations & Deployment

1. `echo "dummy_module" | sudo tee /etc/modules-load.d/dummy_module.conf`
2. `lspci -v` and look for `kernel driver in use`
3. `crontab -e` → `30 2 * * * /usr/local/bin/backup.sh`
4. `grep bugs /proc/cpuinfo` or `cat /proc/cpuinfo | grep -i meltdown`

Part 2: Networking

1. `sudo nmcli con mod eth0 ipv4.addresses 192.168.50.10/24` `sudo nmcli con mod eth0 ipv4.gateway 192.168.50.1` `sudo nmcli con mod eth0 ipv4.method manual` `sudo nmcli con up eth0`
2. `sudo apt install chrony` `sudo nano /etc/chrony/chrony.conf` → add `server pool.ntp.org iburst` `sudo systemctl restart chrony`
3. `sudo firewall-cmd --permanent --add-port=2222/tcp` `sudo firewall-cmd --reload`
4. `sudo ip route add 10.10.0.0/16 via 192.168.50.254`

Part 3: Storage

1. `sudo lvcreate -L 10G -n lv_data vg01`
2. `sudo mkfs.ext4 /dev/vg01/lv_data` `sudo mkdir -p /mnt/data` `sudo mount /dev/vg01/lv_data /mnt/data`
3. `sudo fallocate -l 2G /swapfile` `sudo chmod 600 /swapfile` `sudo mkswap /swapfile` `sudo swapon /swapfile`
4. `/etc/fstab` entry: `server:/share /mnt/nfs nfs defaults 0 0`

Part 4: Essential Commands

1. `top` or `ps aux --sort=-%cpu | head` → `kill <PID>`
2. `du -ah /var/log | sort -rh | head -10`
3. `openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout server.key -out server.crt`
4. `git clone <url>` `echo "new line" >> README.md` `git add README.md` `git commit -m "Add new line"` `git push origin main`

Part 5: Users & Groups

1. `sudo useradd -m -s /bin/bash developer1` `sudo passwd developer1`
2. `sudo groupadd developers` `sudo usermod -aG developers developer1`
3. `ulimit -n 4096` (temporary) or add to `/etc/security/limits.conf`: `developer1 hard nofile 4096`
4. `setfacl -m u:developer1:rw /var/www/html/index.html`

LFCS Mock Exam 2

Generated by ChatGPT

This is a second mock exam for LFCS covering all major topics: Operations, Networking, Storage, Essential Commands, and Users & Groups.

Instructions

- Complete the exercises without looking at the answers.
- Answers are provided at the end of the document.

Part 1: Operations & Deployment

1. Temporarily disable a kernel module `usb_storage` and verify it is no longer in use.
2. List all currently loaded kernel modules and identify one used by the system audio.
3. Schedule a one-time job to run `/usr/local/bin/update_logs.sh` in 15 minutes.
4. Check system boot messages for any hardware errors.

Part 2: Networking

1. Configure interface `eth1` to use DHCP.
2. Verify DNS resolution for `example.com`.
3. Open port 8080 for TCP connections and make the change persistent.
4. Create a bonding device `bond0` using interfaces `eth0` and `eth1`.

Part 3: Storage

1. Create a new 5 GB logical volume `lv_backup` in volume group `vg01`.

2. Format it with XFS and mount it at `/mnt/backup`.
3. Add a new swap partition on `/dev/sdc1` and activate it.
4. Mount a CIFS share `//server/share` on `/mnt/cifs` with credentials stored securely.

Part 4: Essential Commands

1. Monitor memory usage in real-time and identify the process using the most memory.
2. Find all `.log` files larger than 50 MB in `/var/log`.
3. Generate a private key and CSR for a certificate signing request.
4. Clone a Git repository, create a new branch `feature1`, and push it to the remote.

Part 5: Users & Groups

1. Create a system user `service1` without login permissions.
2. Set a hard limit of 10 processes for user `developer1`.
3. Set ACL so that group `developers` has read/write access to `/opt/project/config.yml`.
4. Configure the system to authenticate users against an LDAP server and verify an LDAP user.

Answers (Do not read until done!)

Part 1: Operations & Deployment

1. `sudo modprobe -r usb_storage` `lsmod | grep usb_storage` (should show nothing)
2. `lsmod | grep snd` (example: `snd_hda_intel`)
3. `echo "/usr/local/bin/update_logs.sh" | at now + 15 minutes`
4. `dmesg | less` or `journalctl -b`

Part 2: Networking

1. `sudo nmcli con mod eth1 ipv4.method auto` `sudo nmcli con up eth1`
2. `dig example.com` or `nslookup example.com`
3. `sudo firewall-cmd --permanent --add-port=8080/tcp` `sudo firewall-cmd --reload`
4. `sudo nmcli con add type bond con-name bond0 ifname eth0,eth1 mode active-backup`

Part 3: Storage

1. `sudo lvcreate -L 5G -n lv_backup vg01`
2. `sudo mkfs.xfs /dev/vg01/lv_backup` `sudo mkdir -p /mnt/backup` `sudo mount /dev/vg01/lv_backup /mnt/backup`
3. `sudo mkswap /dev/sdc1` `sudo swapon /dev/sdc1`
4. `sudo mount -t cifs -o credentials=/root/.cifs_credentials //server/share /mnt/cifs`

Part 4: Essential Commands

1. `htop` or `top` and sort by memory usage
2. `find /var/log -type f -name "*.log" -size +50M`
3. `openssl req -new -newkey rsa:2048 -nodes -keyout server.key -out server.csr`
4. `git clone <url>` `cd <repo>` `git checkout -b feature1` `git push -u origin feature1`

Part 5: Users & Groups

1. `sudo useradd -r -s /usr/sbin/nologin service1`
2. Add to `/etc/security/limits.conf`: `developer1 hard nproc 10`
3. `setfacl -m g:developers:rw /opt/project/config.yml`
4. Install and configure LDAP client (`libnss-ldap`, `libpam-ldap`) Verify: `getent passwd ldapuser`