

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`).
-

Revision #6

Created 2025-11-07 10:32:40 UTC by Loïc

Updated 2025-11-19 16:19:08 UTC by Loïc