CompTIA Linux+ Certification Exam Objectives

EXAM NUMBER: XK0-006

About the Exam

The CompTIA Linux+ certification exam will certify the successful candidate has the knowledge and skills required to configure, manage, operate, and troubleshoot Linux server environments while using security best practices, scripting, containerization, virtualization, and automation.

This is equivalent to 12 months of hands-on experience working with Linux servers. Certifications in and/or knowledge about A+, Network+, Server+ are recommended.

These content examples are meant to clarify the test objectives and should not be construed as a comprehensive listing of all the content of this examination.

EXAM ACCREDITATION

The CompTIA Linux+ exam is accredited by ANSI to show compliance with the ISO 17024 standard and, as such, undergoes regular reviews and updates to the exam objectives.

EXAM DEVELOPMENT

CompTIA exams result from subject matter expert workshops and industry-wide survey results regarding the skills and knowledge required of an IT professional.

COMPTIA AUTHORIZED MATERIALS USE POLICY

CompTIA Certifications, LLC is not affiliated with and does not authorize, endorse, or condone utilizing any content provided by unauthorized third-party training sites (aka "brain dumps"). Individuals who utilize such materials in preparation for any CompTIA examination will have their certifications revoked and be suspended from future testing in accordance with the CompTIA Candidate Agreement. In an effort to more clearly communicate CompTIA's exam policies on use of unauthorized study materials, CompTIA directs all certification candidates to the CompTIA Certification Exam Policies. Please review all CompTIA policies before beginning the study process for any CompTIA exam. Candidates will be required to abide by the CompTIA Candidate Agreement. If a candidate has a question as to whether study materials are considered unauthorized (aka "brain dumps"), he/she should contact CompTIA at examsecurity@comptia.org to confirm.

PLEASE NOTE

The lists of examples provided in bulleted format are not exhaustive lists. Other examples of technologies, processes, or tasks pertaining to each objective may also be included on the exam, although not listed or covered in this objectives document. CompTIA is constantly reviewing the content of our exams and updating test questions to be sure our exams are current, and the security of the questions is protected. When necessary, we will publish updated exams based on existing exam objectives. Please know that all related exam preparation materials will still be valid.

TEST DETAILS

Required exam	XK0-006
Number of questions	
Types of questions	Multiple-choice and performance-based
Length of test	
Recommended experience	12 months of hands-on experience working with Linux servers; A+, Network+, Server+, or similar certifications and/or knowledge recommended

EXAM OBJECTIVES (DOMAINS)

The table below lists the domains measured by this examination and the extent to which they are represented.

DOMAIN		PERCENTAGE OF EXAMINATION
1.0	System Management	23%
2.0	Services and User Management	20%
3.0	Security	18%
4.0	Automation, Orchestration, and Scripting	17%
5.0	Troubleshooting	22%
Total		**100%**

1.0 System Management

1.1 Explain basic Linux concepts.

- Basic boot process
  - Bootloader
    - Configuration files
  - Kernel
    - Parameters
  - Initial RAM [random-access memory] disk (initrd)
  - Preboot Execution Environment (PXE)

- Filesystem Hierarchy Standard (FHS)
  - `/`
  - `/bin`
  - `/boot`
  - `/dev`
  - `/etc`
  - `/home`
  - `/lib`
  - `/proc`
  - `/sbin`
  - `/tmp`
  - `/usr`
  - `/var`
- Server architectures
  - AArch64
  - Reduced instruction set computer, version five (RISC-V)
  - x86
  - x86_64/AMD64
- Distributions
  - RPM Package Manager (RPM)-based
  - Debian packet manager (dpkg)-based
- Graphical User Interface (GUI)
  - Display managers
  - Window managers
  - X Server
  - Wayland
- Software licensing
  - Opensource software
  - Free software
  - Proprietary software
  - Copyleft

1.2 Summarize Linux device management concepts and tools.

- Kernel modules
  - depmod
  - insmod
  - lsmod
  - modinfo
  - modprobe
  - rmmod
- Device management
  - dmesg
  - dmidecode
  - ipmitool
  - lm_sensors
  - lscpu
  - lshw
  - lsmem
  - lspci
  - lsusb
- initrd management
  - dracut
  - mkinitrd
- Custom hardware
  - Embedded systems
  - Graphics Processing Unit (GPU) use cases
    - nvtop

1.3 Given a scenario, manage storage in a Linux system.

- Logical Volume Manager (LVM)
  - Logical volume
    - lvchange
    - lvcreate
    - lvdisplay
    - lvremove
    - lvresize/lvextend
    - lvs
- Volume group
  - vgchange
  - vgcreate
  - vgdisplay
  - vgexport
  - vgextend
  - vgimport
  - vgremove
  - vgs
  - vgscan
- Physical volume
  - pvcreate
  - pvdisplay
  - pvmove
  - pvremove
  - pvresize
  - pvs
  - pvscan
- Partitions
  - blkid
  - fdisk/gdisk
  - growpart
  - lsblk
  - parted
- Filesystems
  - Formats
    - xfs
    - ext4
    - btrfs
    - tmpfs
- Utilities
  - df
  - du
  - fio
  - fsck
  - mkfs
  - resize2fs
  - xfs_growfs
  - xfs_repair
- Redundant Array of Independent Disks (RAID)
  - /proc/mdstat
  - mdadm
- Mounted storage
  - Mounting
    - /etc/fstab
    - /etc/mtab
    - /proc/mounts
    - autofs
    - mount
    - umount
  - Mount options
    - noatime
    - nodev
    - nodiratime
    - noexec
    - nofail
    - nosuid
    - remount
    - ro
    - rw
  - Network mounts
    - Network file system (NFS)
    - Server Message Block (SMB) Samba
- Inodes

1.4 Given a scenario, manage network services and configurations on a Linux server.

- Network configuration
  - /etc/hosts
  - /etc/resolv.conf
  - /etc/nsswitch.conf
- NetworkManager
  - nmcli
  - nmconnect
- Netplan
  - netplan apply
  - netplan status
  - netplan try
    - Configuration files
      - (i) /etc/netplan
- Common network tools
  - arp
  - curl
  - dig
  - ethtool
  - hostname
  - ip
    - ip address
    - ip link
    - ip route
  - iperf3
  - mtr
  - nc
  - nmap
  - nslookup
  - ping/ping6
  - ss
  - tcpdump
  - tracepath
  - traceroute

1.5 Given a scenario, manage a Linux system using common shell operations.

- Common environmental variables
  - DISPLAY
  - HOME
  - PATH
  - PS1
  - SHELL
  - USER
- Paths
  - Absolute
    - ~
    - /
- Relative
  - `.`
  - `..`
  - `-`
- Shell environment configurations
  - .bashrc
  - .bash_profile
  - .profile
- Channel redirection
  - `<`
  - `>`
  - `<<`
  - `>>`
  - `|`
  - Standard output
  - Standard error
  - Standard input
  - Here docs
    - `<<<`
- Basic shell utilities
  - `!`
  - `!!`
  - alias
  - awk
  - bc
  - cat
  - cut
  - echo
  - grep
  - head
  - history
  - less
  - more
  - printf
  - sed
  - sort
  - source
  - tail
  - tee
  - tr
  - uname
  - uniq
  - wc
  - xargs
- Text editors
  - vi/vim
  - nano

1.6 Given a scenario, perform backup and restore operations for a Linux server.

- Archiving
  - `cpio`
  - `tar`
- Compression tools
  - 7-Zip
  - `bzip2`
  - `gzip`
  - `unzip`
  - `xz`
- Other tools
  - `dd`
  - `ddrescue`
  - `rsync`
  - `zcat`
  - `zgrep`
  - `zless`

1.7 Summarize virtualization on Linux systems.

- Linux hypervisors
  - Quick Emulator (QEMU)
  - Kernel-based Virtual Machine (KVM)
- Virtual machines (VMs)
  - Paravirtualized drivers
  - VirtIO
  - Disk image operations
    - Convert
    - Resize
    - Image properties
  - VM states
  - Nested virtualization
- VM operations
  - Resources
    - Storage
    - RAM
    - Central processing unit (CPU)
    - Network
  - Baseline image templates
  - Cloning
  - Migrations
  - Snapshots
- Bare metal vs. virtual machines
- Network types
  - Bridged
  - Network address translation (NAT)
  - Host-only/isolated
  - Routed
  - Open
- Virtual machine tools
  - `libvirt`
  - `virsh`
  - `virt-man`

2.0 Services and User Management

2.1 Given a scenario, manage files and directories on a Linux system.

- Utilities
  - `cd`
  - `cp`
  - `diff`
  - `file`
  - `find`
  - `ln`
  - `locate`
  - `ls`
  - `lsof`
  - `mkdir`
  - `mv`
  - `pwd`
  - `rm`
  - `rmdir`
  - `sdiff`
  - `stat`
  - `touch`
- Links
  - Symbolic link
  - Hard link
- Device types in /dev
  - Block devices
  - Character devices
  - Special character devices

2.2 Given a scenario, perform local account management in a Linux environment.

- Add
  - `adduser`
  - `groupadd`
  - `useradd`
- Delete
  - `deluser`
  - `groupdel`
  - `userdel`
- Modify
  - `chsh`
  - `groupmod`
  - `passwd`
  - `usermod`
- Lock
  - `chage`
  - `passwd`
  - `usermod`
- Expiration
  - Configuration files
  - `chage`
- List
  - `getent passwd`
  - `groups`
  - `id`
  - `last`
  - `lastlog`
  - `w`
  - `who`
  - `whoami`
- User profile templates
  - `/etc/profile`
  - `/etc/skel`
- Account files
  - `/etc/group`
  - `/etc/passwd`
  - `/etc/shadow`
- Attributes
  - Unique Identifier (UID)
  - Group Identifier (GID)
  - Effective User Identifier (EUID)
  - Effective Group Identifier (EGID)
- User accounts vs. system accounts vs. service accounts
  - UID range

2.3 Given a scenario, manage processes and jobs in a Linux environment.

- Process verification
  - `/proc/<PID>`
  - `atop`
  - `htop`
  - `lsof`
  - `mpstat`
  - `pidstat`
  - `ps`
  - `pstree`
  - `strace`
  - `top`
- Process ID
  - Parent Process Identification Number (PPID)
  - Process Identification Number (PID)
- Process states
  - Running
  - Blocked
  - Sleeping
  - Stopped
  - Zombie
- Priority
  - nice
  - renice
- Process limits
- Job and process management
  - `&`
  - `bg`
  - `Ctrl + c`
  - `Ctrl + d`
  - `Ctrl + z`
  - `exec`
  - `fg`
  - `jobs`
  - `kill`
  - `killall`
  - nohup
  - `pkill`
  - Signals
    - 1 HUP
    - 9 KILL
    - 15 TERM
- Scheduling
  - `anacron`
  - `at`
  - `crontab`

2.4 Given a scenario, configure and manage software in a Linux environment.

- Installation, update, and removal
  - Repository
  - Source
  - Package dependencies and conflicts
  - Package managers
  - Language-specific
    - pip
    - cargo
    - npm
- Repository management
  - Enabling/disabling
  - Third party
  - Gnu's Not Unix (GNU) Privacy Guard (GPG) signatures
- Package and repository exclusions
- Update alternatives
- Software configuration
- Sandboxed applications
- Basic configurations of common services
  - Domain Name System (DNS) protocol
  - Network Time Protocol (NTP)/ Precision Time Protocol (PTP)
  - Dynamic Host Configuration Protocol (DHCP)
  - HyperText Transfer Protocol (HTTP)
    - Apache HTTP Server (httpd)
    - Nginx
  - Simple Mail Transfer Protocol (SMTP)
  - Internet Messaging Access Protocol (IMAP4)

2.5 Given a scenario, manage Linux using systemd.

- Systemd units
  - Services
  - Timers
  - Mounts
  - Targets
- Utilities
  - `hostnamectl`
  - `resolvectl`
  - `sysctl`
  - `systemctl`
  - `systemd-analyze`
  - `systemd-blame`
  - `systemd-resolved`
  - `timedatectl`
- Managing unit states
  - `daemon-reload`
  - `disable`
  - `edit`
  - `enable`
  - `mask`
  - `reload`
  - `restart`
  - `start`
  - `status`
  - `stop`
  - `unmask`

2.6 Given a scenario, manage applications in a container on a Linux server.

- Runtimes
  - runC
  - Podman
  - containerd
  - Docker
- Image operations
  - Pulling images
  - Build an image
    - Dockerfile
      - (i) ENTRYPOINT
      - (ii) CMD
      - (iii) USER
      - (iv) FROM
  - Pruning
  - Tags
  - Layers
- Container operations
  - Read container logs
  - Map container volumes
  - Start/stop containers
  - Inspect containers
  - Delete a container
  - Run
  - Exec
  - Pruning
  - Tags
  - Environmental variables
- Volume operations
  - Create volume
  - Mapping volume
  - Pruning
  - SELinux context
  - Overlay
- Container networks
  - Create network
  - Port mapping
  - Pruning
  - Types
    - macvlan
    - ipvlan
    - Host
    - Bridge
    - Overlay
    - None
- Privileged vs. unprivileged

3.0 Security

3.1 Summarize authorization, authentication, and accounting methods.

- Polkit
- Pluggable Authentication Modules (PAM)
- System Security Services Daemon (SSSD)/Winbind
- realm
- Lightweight Directory Access Protocol (LDAP)
- Kerberos
- Samba
- Logging
  - `journalctl`
  - `rsyslog`
  - `logrotate`
  - `/var/log`
- System audit
  - `audit.rules`
  - `auditd`

3.2 Given a scenario, configure and implement firewalls on a Linux system.

- firewalld
  - `firewall-cmd`
  - Runtime vs. permanent
  - Rich rules
  - Zones
  - Ports vs. services
- Uncomplicated Firewall (ufw)
  - Ports vs. services
- `nftables`
- `iptables`
- `ipset`
- Netfilter module
- Address translation
  - NAT
  - Port Address Translation (PAT)
  - Destination Network Address Translation (DNAT)
  - Source Network Address Translation (SNAT)
- Stateful vs. stateless
- Internet rotocol (IP) forwarding
  - `net.ipv4.ip_forward`

3.3 Given a scenario, apply operating system (OS) hardening techniques on a Linux system.

- Privilege escalation
  - `sudo`
    - `/etc/sudoers`
      - (i) `NOEXEC`
      - (ii) `NOPASSWD implications`
    - `/etc/sudoers.d`
    - `visudo`
    - `sudo -i`
    - `wheel group`
    - `sudo group`
  - `su -`
- File attributes
  - `chattr`
  - `lsattr`
    - immutable
    - append only
- Permissions
  - File permissions
    - `chgrp`
    - `chmod`
      - (i) Octal
      - (ii) Symbolic
    - `chown`
  - Special permissions
    - Sticky bit
    - `setuid`
    - `setgid`
  - Default user file-creation mode mask (`umask`)
- Access control
  - Access control lists (ACLs)
    - `setfacl`
    - `getfacl`
  - SELinux
    - `restorecon`
    - `semanage`
    - `chcon`
    - `ls -Z`
    - `getenforce`
    - `setenforce`
    - `getsebool`
    - `setsebool`
    - `audit2allow`
    - `sealert`
    - States
      - (i) Enforcing
      - (ii) Permissive
      - (iii) Disabled
- Secure remote access
  - Solid-state hybrid drive (SSHD)
    - Key vs. password authentication
    - Secure Shell (SSH) tunneling
    - PermitRootLogin
    - Disabling X forwarding
    - AllowUsers
    - AllowGroups
  - SSH agent
  - Secure File Transfer Protocol (SFTP)
    - `chroot`
  - fail2ban
- Avoid the use of unsecure access services
  - Telnet
  - File Transfer Protocol (FTP)
  - Trivial File Transfer Protocol (TFTP)
- Disabling unused file systems
- Removal of unnecessary Set User ID (SUID) permissions
- Secure boot
  - Unified Extensible Firmware Interface (UEFI)

3.4 Explain account hardening techniques and best practices.

- Passwords
  - Complexity
  - Length
  - Expiration
  - Reuse
  - History
- Multifactor authentication (MFA)
- Checking existing breach lists
- Restricted shells
  - `/sbin/nologin`
  - `/bin/rbash`
- `pam_tally2`
- Avoid running as root

3.5 Explain cryptographic concepts and technologies in a Linux environment.

- Data at rest
  - File encryption
    - GPG
  - Filesystem encryption
    - Linux Unified Key Setup 2 (LUKS2)
    - Argon2
- Data in transit
  - Open Secure Sockets Layer (OpenSSL)
  - WireGuard
  - LibreSSL
  - Transport Layer Security (TLS) protocol versions
- Hashing
  - SHA-256
  - Hashed message authentication code (HMAC)
- Removal of weak algorithms
- Certificate management
  - Trusted root certificates
    - No-cost
    - Commercial
- Avoiding self-signed certificates

3.6 Explain the importance of compliance and audit procedures.

- Detection and response
  - Anti-malware
  - Indicators of compromise (IOC)
- Vulnerability scanning
  - Common Vulnerabilities and Exposures (CVEs)
  - Common Vulnerability Scoring System (CVSS)
  - Backporting patches
  - Service misconfigurations
  - Tools
    - Port scanners
    - Protocol analyzer
- Standards and audit
  - Open Security Content Automation Protocol (OpenSCAP)
  - Center for Internet Security (CIS) Benchmarks
- File integrity
  - Advanced Intrusion Detection Environment (AIDE)
  - Rootkit hunter (rkhunter)
  - Signed package verification
  - Installed file verification
- Secure data destruction
  - `shred`
  - `badblocks -w`
  - `dd if=/dev/urandom`
  - Cryptographic destruction
- Software supply chain
- Security banners
  - `/etc/issue`
  - `/etc/issue.net`
  - `/etc/motd`

4.0 Automation, Orchestration, and Scripting

4.1 Summarize the use cases and techniques of automation and orchestration in a Linux environment.

- Infrastructure as code
  - Ansible
    - Playbooks
    - Inventory
    - Modules
    - Ad hoc
    - Collections
    - Facts
    - Agentless
- Puppet
  - Classes
  - Certificates
  - Modules
  - Facts
  - Agent/Agentless
- OpenTofu
  - Provider
  - Resource
  - State
  - Application programming interface (API)
- Unattended deployment
  - Kickstart
  - Cloud-init
- Continuous integration/ Continuous deployment (CI/CD)
  - Version control
  - Shift left testing
  - GitOps
  - Pipelines
  - DevSecOps
- Deployment orchestration
  - Kubernetes
    - ConfigMaps
    - Secrets
    - Pods
    - Deployments
    - Volumes
    - Services
    - Variables
  - Docker Swarm
    - Service
    - Nodes
    - Tasks
    - Networks
    - Scale
  - Docker/Podman Compose
    - Compose file
    - Up/down
    - Logs

4.2 Given a scenario, perform automated tasks using shell scripting.

- Expansion
  - Parameter expansion
    - `${var}`
  - Command substitution
    - `$(foo)`
    - `` `foo` ``
  - Subshell
    - `(foo)`
- Functions
- Internal Field Separator/ Output Field Separator (IFS/OFS)
- Conditional statements
  - `if`
  - `case`
- Looping statements
  - `until`
  - `for`
  - `while`
- Interpreter directive
  - `#!`
- Comparisons
  - Numerical
    - `-eq`
    - `-ge`
    - `-gt`
    - `-le`
    - `-lt`
    - `-ne`
  - String
    - `>`
    - `<`
    - `==`
    - `=`
    - `= ~`
    - `! =`
    - `<=`
    - `>=`
- Regular expressions
  - `[[ $foo =~ regex ]]`
- Test
  - `!`
  - `-d`
  - `-f`
  - `-n`
  - `-z`
- Variables
  - Environmental
  - Arguments
  - Assignments
    - `alias`
    - `export`
    - `local`
    - `set`
    - `unalias`
    - `unset`
  - Return codes
    - `$?`

4.3 Summarize Python basics used for Linux system administration.

- **Setting up a virtual environment**
- **Built-in modules**
- **Installing dependencies**
- **Python fundamentals**
  - Indentations
  - Current versions
  - Data types and structures
    - Boolean
    - Dictionary
    - Floating point
    - Integer
    - List
    - String
  - Extensible using modules and packages
- **Python Enhancement Proposal (PEP) 8 best practices**

4.4 Given a scenario, implement version control using Git.

- `.gitignore`
- `add`
- `branch`
- `checkout`
- `clone`
- `commit`
- `config`
- `diff`
- `fetch`
- `init`
- `log`
- `merge`
  - `squash`
- `pull`
- `push`
- `rebase`
- `reset`
- `stash`
- `tag`

4.5 Summarize best practices and responsible uses of artificial intelligence (AI).

- **Common use cases**
  - Generation of code
  - Generation of regular expressions
  - Generation of infrastructure as code
  - Document code/create documentation
  - Recommendations for how to improve compliance
  - Security review
  - Code optimization
  - Code linting
- **Best practices**
  - Avoid copy/paste without review/quality assurance
  - Verify output
  - Data governance
    - Security of data
      - (i) Large language model (LLM) training
      - (ii) Human review
    - Local models
      - (i) Private vs. public
  - Adhere to corporate policy
- **Prompt engineering**

5.0 Troubleshooting

5.1 Summarize monitoring concepts and configurations in a Linux system.

- **Service monitoring**
  - Service-level agreement (SLA)
  - Service-level indicator (SLI)
  - Service-level objective (SLO)
- **Data acquisition methods**
  - Simple Network Management Protocol (SNMP)
    - Traps
    - Management information bases (MIBs)
  - Agent/agentless
  - Webhooks
  - Health checks
  - Log aggregation
- **Configurations**
  - Thresholds
  - Alerts
  - Events
  - Notifications
  - Logging

5.2 Given a scenario, analyze and troubleshoot hardware, storage, and Linux OS issues.

- **Common issues**
  - Kernel panic
  - Data corruption issues
  - Kernel corruption issues
  - Package dependency issues
  - Filesystem will not mount
  - Server not turning on
  - OS filesystem full
  - Server inaccessible
  - Device failure
  - Inode exhaustion
  - Partition not writable
  - Segmentation fault
  - Grand Unified Bootloader (GRUB) misconfiguration
  - Killed processes
  - PATH misconfiguration issues
  - Systemd unit failures
  - Missing or disabled drivers
  - Unresponsive process
  - Quota issues
  - Memory leaks

5.3 Given a scenario, analyze and troubleshoot networking issues on a Linux system.

- **Common issues**
  - Misconfigured firewalls
  - DHCP issues
  - DNS issues
  - Interface misconfiguration
    - Maximum transmission unit (MTU) mismatch
    - Bonding
    - Media access control (MAC) spoofing
    - Subnet
    - Cannot ping server
  - Routing issues
    - Gateway
  - Server unreachable
  - IP conflicts
  - Dual stack issues (IPv4 and IPv6)
  - Link down
  - Link negotiation issues

5.4 Given a scenario, analyze and troubleshoot security issues on a Linux system.

- Common issues
  - SELinux issues
    - Policy
    - Context
    - Booleans
  - File and directory permission issues
    - ACLs
    - Attributes
  - Account access
  - Unpatched vulnerable systems
  - Exposed or misconfigured services
  - Remote access issues
  - Certificate issues
  - Misconfigured package repository
  - Use of obsolete or insecure protocols and ciphers
  - Cipher negotiation issues

5.5 Given a scenario, analyze and troubleshoot performance issues.

- Common symptoms
  - Swapping
  - Out of memory
  - Slow application response
  - System unresponsiveness
  - High CPU usage
  - High load average
  - High context switching
  - High failed log-in attempts
  - Slow startup
  - High input/output (I/O) wait time
  - Packet drops
  - Jitter
  - Random disconnects
  - Random timeouts
  - High latency
  - Slow response times
  - High disk latency
  - Low throughput
  - Blocked processes
  - Hardware errors
  - Sluggish terminal behavior
  - Exceeding baselines
  - Slow remote storage response
  - CPU bottleneck

CompTIA Linux+ Acronym List

The following acronyms appear on the CompTIA Linux+ exam. Candidates are encouraged to review the complete list and attain a working knowledge of all listed acronyms as part of a comprehensive exam preparation program.

ACRONYM	SPELLED OUT
ACL	Access Control List
AI	Artificial Intelligence
AIDE	Advanced Intrusion Detection Environment
API	Application Programming Interface
ARM	Advanced Reduced Instruction Set Computer (RISC) Machine
BIOS	Basic Input/Output System
CI/CD	Continuous Integration/Continuous Deployment
CIFS	Common Internet File System
CIS	Center for Internet Security
CPU	Central Processing Unit
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DHCP	Dynamic Host Configuration Protocol
DNAT	Destination Network Address Translation
DNS	Domain Name System
EGID	Effective Group Identifier
EUID	Effective User Identifier
FHS	Filesystem Hierarchy Standard
FTP	File Transfer Protocol
FUSE	Filesystem in Userspace
GID	Group Identifier
GNU	Gnu's Not Unix
GPG	GNU Privacy Guard
GPT	GUID (Globally Unique Identifier) Partition Table
GPU	Graphics Processing Unit
GRUB	Grand Unified Bootloader
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HMAC	Hashed Message Authentication Code
HTTP	HyperText Transfer Protocol
HTTPD	HyperText Transfer Protocol Daemon
initrd	Initial RAM Disk
I/O	Input/Output
IFS/OFS	Internal Field Separator/Output Field Separator
IMAP4	Internet Messaging Access Protocol 4
IoC	Indicators of Compromise
IOPS	Input/Output Operations Per Second
IP	Internet Protocol
ISO	International Standards Organization
JSON	JavaScript Object Notation

ACRONYM	SPELLED OUT
KVM	Kernel-based Virtual Machine
LDAP	Lightweight Directory Access Protocol
LLM	Large Language Model
LUKS	Linux Unified Key Setup
LUKS2	Linux Unified Key Setup 2
LVM	Logical Volume Manager
MAC	Media Access Control
MBR	Master Boot Record
MFA	Multifactor Authentication
MIB	Management Information Base
MTU	Maximum Transmission Unit
NAS	Network-attached Storage
NAT	Network Address Translation
NFS	Network File System
NTP	Network Time Protocol
NVMe	Non-Volatile Memory Express
OOM	Out of Memory
OpenSCAP	Open Security Content Automation Protocol
OpenSSL	Open Secure Sockets Layer
OS	Operating System
PAM	Pluggable Authentication Modules
PAT	Port Addresss Translation
PEP	Python Enhancement Proposal
PID	Process Identification Number
PKI	Public Key Infrastructure
PPID	Parent Process Identification Number
PTP	Precision Time Protocol
PXE	Preboot Execution Environment
QEMU	Quick Emulator
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RPM	Red Hat Package Manager
SAN	Storage Area Network
SELinux	Security Enhanced Linux
SFTP	Secure File Transfer Protocol
SGID	Set Group ID
SLA	Service-level Agreement
SLES	SUSE Linux Enterprise Server
SLI	Service-level Indicator
SLO	Service-level Objective
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol

ACRONYM	SPELLED OUT
SNAT	Source Network Address Translation
SNMP	Simple Network Management Protocol
SSD	Solid-state Drive
SSH	Secure Shell
SSHD	Solid-state Hybrid Drive
SSL	Secure Sockets Layer
SSO	Single Sign-On
SSSD	System Security Services Daemon
SUID	Set User ID
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
UEFI	Unified Extensible Firmware Interface
UFW	Uncomplicated Firewall
UID	Unique Identifier
USB	Universal Serial Bus
VM	Virtual Machine
YAML	YAML Ain't Markup Language

CompTIA Linux+ Hardware and Software List

CompTIA has included this sample list of hardware and software to assist candidates as they prepare for the Linux+ certification exam. This list may also be helpful for training companies that wish to create a lab component for their training offering. The bulleted lists below each topic are sample lists and are not exhaustive.

EQUIPMENT

- Internet access
- Laptop or desktop that supports virtualization or access to a cloud service provider
- Network
- Router
- Spare parts/hardware
- Solid-state drive (SSD)
- Switch
- Universal Serial Bus (USB) media
- Wireless access point

SOFTWARE

- Automation tools
  - Ansible
  - Puppet
- Containerization software
  - Docker
  - Kubernetes
    - Minikube
  - Podman
- Git
- Git repository
- LLM access
- Package repository
- PuTTY or SSH client
- Python 3
- Virtualization software

RECOMMENDED DISTRIBUTIONS

- Alma Linux
- Debian
- Fedora Linux
- OpenSUSE/SUSE Linux Enterprise Server (SLES)
- Red Hat Enterprise Linux
- Rocky Linux
- Ubuntu