

UNIT 2

Workstation, Server and Services

(16 Hours) — Network Administration (CACS406)

⚠ Please Note

These notes were prepared with the help of AI from the TU BCA syllabus and general knowledge. They may contain errors, omissions or outdated details. Please verify important points against your textbook, class notes and the latest TU syllabus before relying on them in an exam.

2.1 Workstations

A **workstation** (desktop/laptop/client machine) is the computer a user works on every day. In an enterprise there may be hundreds or thousands of them, so the administrator's aim is to make them **uniform, easy to install, easy to fix and easy to replace**. Managing many machines one by one does not scale; automation and standardization are essential.

2.1.1 Architecture Design

Workstation architecture decides *how* user machines are provided and managed. The main models are:

Model	Description	Advantages	Disadvantages
Fat client (standard PC)	Full OS and applications installed locally	Works offline, powerful, familiar	Harder to manage and patch; more hardware cost
Thin client	Minimal device; applications run on a server and screen is sent to the client	Easy management, low cost, secure	Depends on server and network
Virtual Desktop Infrastructure (VDI)	Each user's desktop is a VM in the data center	Central control, access from anywhere	Needs strong servers/storage
Diskless / network boot	OS loaded over the network at start-up	No local OS to maintain	Depends on boot server and network
BYOD	Users bring personal devices under policy	Cost saving, user choice	Security and support challenges

Design Principles

- **Standardization:** a small number of hardware models and one standard OS image reduce support cost.
- **Separation of data from OS:** keep user data on servers (network home directories, redirected folders, cloud) so a machine can be reinstalled without data loss.
- **Central management:** use directory services (Active Directory / LDAP), Group Policy and configuration-management tools.
- **Disposability:** treat workstations as replaceable; a failed machine is re-imaged rather than repaired for hours.
- **Security by default:** full-disk encryption, host firewall, antivirus, least-privilege accounts, automatic updates.
- **Scalability and remote support:** remote desktop, remote install and inventory tools.

2.1.2 Hardware Strategies

Hardware strategy is the plan for buying, standardizing, supporting and retiring workstation hardware.

- **Fewer models:** buy the same model in bulk (per department needs: standard office, power user, developer/designer, laptop).
- **Vendor selection and warranty:** choose reliable vendors, 3–5 year on-site warranty, spare-part availability.
- **Lifecycle / refresh policy:** replace hardware on a regular schedule (for example, every 3–4 years) instead of waiting for failure.
- **Spare pool:** keep a small stock of ready machines and parts to swap quickly.
- **Compatibility:** check drivers and OS support before buying; prefer standard, well-supported components.
- **Power, ergonomics and green IT:** energy-efficient devices, proper monitors and peripherals, e-waste disposal.
- **Asset tracking:** tag and record each device (serial number, owner, location, warranty date).

Example 2.1 — Choosing workstation models

A college lab has 120 identical PCs, the administration office has 15 desktops and 8 teachers have laptops.

Question: how many hardware models should be purchased?

Solution

Three models are enough: (1) lab desktop, (2) office desktop, (3) teacher laptop. Each group is bought in bulk with one OS image, which reduces training, spare parts and support effort.

2.1.3 OS Installation

Installing an OS by hand (insert disc, click through screens) takes 30–60 minutes per machine and produces slightly different systems each time. Good practice is **automated, repeatable installation**.

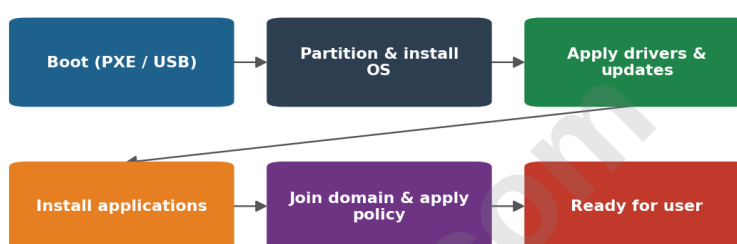


Figure 2.1 — Steps of an automated workstation installation

Method	How it works	Example tools
Manual install	Technician installs from media and configures by hand	OS DVD / USB
Automated install (unattended)	An answer file provides all choices, so no clicks are needed	Windows unattend.xml, Linux Kickstart, Debian Preseed
Disk imaging / cloning	A 'golden image' of a configured PC is copied to many machines	Clonezilla, Ghost, WDS images
Network (PXE) deployment	Machine boots from the network and pulls the image or installer	Windows Deployment Services, MDT, SCCM/Intune, FOG, PXE + TFTP
Package-based configuration	After base install, a configuration tool installs and enforces settings	Ansible, Puppet, Chef, Group Policy

Best Practices for OS Installation

- Create and version a **standard image** or answer file; test it before mass use.
- Use **separate partitions or storage** for OS and data where suitable.
- Automate **driver injection, updates and application installation**.
- Apply a consistent naming convention (e.g., LAB1-PC001) and set a local admin password securely (LAPS or similar).
- Join the machine to the **domain / directory** so users and policies are applied centrally.
- Keep the **update strategy** clear: test group first, then wide rollout (patch management).
- Document the process so anyone can rebuild a machine.

Note: Golden images become out-of-date. Rebuild them regularly, or use a base install plus automated configuration to avoid 'image sprawl'.

2.2 Servers

A **server** is a computer that provides services (files, e-mail, web, database, authentication) to many clients. Unlike workstations, servers must be **reliable, available, secure and manageable**, and a failure affects many users.

Aspect	Workstation	Server
Users	One user at a time	Many users/clients at once
Uptime need	Moderate	High (24×7, often 99.9%+)
Hardware	Consumer / business class	Enterprise class: redundant parts, ECC memory, remote management
Location	User's desk	Data center / server room
Cost of failure	One user affected	Whole department or organization affected

2.2.1 Hardware Strategies for Servers

- **Buy enterprise-class hardware:** designed for continuous operation with vendor support (Dell PowerEdge, HP ProLiant, Lenovo ThinkSystem, etc.).
- **Vendor and model consistency:** a few standard models make spares, firmware and training simpler.
- **Sizing and capacity:** estimate CPU, RAM, disk and network needs for the expected load plus growth (typically 2–3 years).

- **Scale-up vs. scale-out:** either a bigger single server (scale-up) or more servers sharing the load (scale-out).
- **Virtualization:** run many logical servers on fewer physical hosts to save cost, power and space and to simplify recovery.
- **Cloud alternative:** rent capacity when demand is uncertain or for short-term needs.
- **Service contracts:** next-business-day or 4-hour on-site support for critical servers.
- **Spare and standby:** keep spare parts or a standby server for critical services.
- **Lifecycle:** plan replacement (usually 4–5 years) before hardware becomes unsupported.

2.2.2 Hardware Features and Specifications

Feature	Purpose / what to look for
Form factor	Tower, rack-mount (1U, 2U, 4U) or blade. Rack servers save space and are standard in data centers.
CPU	Multi-core server processors (Intel Xeon, AMD EPYC); choose by core count and clock speed for the workload.
Memory (RAM)	ECC (Error-Correcting Code) RAM detects and fixes single-bit errors; ensure free slots for expansion.
Storage	SAS/SATA HDDs, SSDs, NVMe; hot-swappable drive bays; hardware or software RAID controller.
RAID	RAID 1 (mirror), RAID 5 (striping + single parity), RAID 6 (double parity), RAID 10 (mirror + stripe) for redundancy and speed.
Power supply	Redundant hot-swap power supplies connected to separate circuits and a UPS.
Cooling	Redundant hot-swap fans; suitable airflow in the rack/room.
Network interfaces	Multiple Gigabit/10 GbE ports; NIC teaming (bonding) for redundancy and bandwidth.
Remote management	Out-of-band management (iDRAC, iLO, IPMI) for remote console, power control and monitoring even if the OS is down.
Expansion	PCIe slots for HBAs, extra NICs, GPUs.
Reliability features	Hot-swap components, failure indicator lights, redundant boot disks, predictive failure alerts.
Security features	TPM, secure boot, BIOS/firmware passwords, lockable chassis.

Example 2.2 — RAID capacity

A server has four 2 TB disks. Find the usable capacity for RAID 0, RAID 1 (two mirrored pairs = RAID 10 layout), RAID 5 and RAID 6.

Solution

RAID 0: $4 \times 2 = 8$ TB (no redundancy).

RAID 10 (mirrored pairs striped): $(4 \div 2) \times 2 = 4$ TB; survives one failure per mirrored pair.

RAID 5: $(4 - 1) \times 2 = 6$ TB; survives one disk failure.

RAID 6: $(4 - 2) \times 2 = 4$ TB; survives two disk failures.

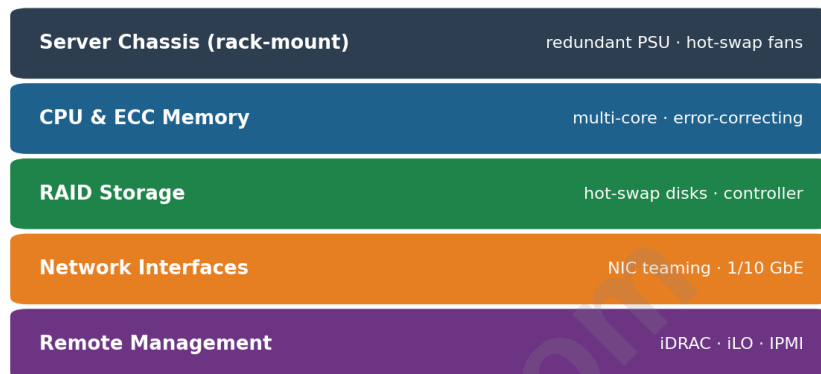


Figure 2.2 — Typical features of an enterprise server

2.3 Services

A **service** is a function the IT organization provides to users, such as e-mail, printing, web hosting, file storage, DNS or authentication. A service is more than a server: it includes hardware, software, people, processes, monitoring and support. A good service is **reliable**, **scalable**, **secure**, **easy to use** and **supportable**.



Figure 2.3 — Service life-cycle covered in this unit

2.3.1 Service Requirements

Before building a service, the administrator must understand what the customers really need. Requirements are gathered by talking to users, managers and other IT teams and are written down clearly.

- **Customer requirements:** what the users need the service to do (features, ease of use, access from where and when).
- **Functional requirements:** exactly what the service must perform (send mail, store 5 GB per user, authenticate through the directory).
- **Non-functional requirements:** performance, capacity, availability (e.g., 99.9%), reliability, security, scalability, response time.
- **Operational requirements:** monitoring, backup, support hours, maintenance windows, documentation.
- **Budget and constraints:** cost, staff skills, deadlines, existing technology, legal rules.
- **Service Level Agreement (SLA):** a written promise on availability, support response time and performance, agreed with customers.

Availability	Allowed downtime per year (approx.)
99%	3.65 days
99.9% (three nines)	8.76 hours
99.99% (four nines)	52.6 minutes
99.999% (five nines)	5.26 minutes

Example 2.3 — Requirements for a college e-mail service

Users: 1500 students and 120 staff. Needs: 5 GB mailbox each, web and mobile access, spam filtering, 99.9% availability, daily backup, help by the IT desk on working days.

Written as requirements

Functional: send/receive mail, webmail, mobile sync, spam/virus filtering. Capacity: 1620×5 GB ≈ 8.1 TB with growth margin. Availability: at most about 8.76 hours downtime per year. Security: encrypted connections (TLS), strong passwords/MFA. Operations: daily backup, monitoring, SLA response within 4 working hours.

Note: Poor requirements are the most common cause of failed IT projects. Always confirm the requirements with the customer in writing.

2.3.2 Planning and Engineering

With requirements known, the service is planned and designed so it can be built, run and grown without surprises.

- **Capacity planning:** estimate users, storage, CPU, memory and bandwidth now and for the next 2–3 years; add headroom.
- **Simplicity and standard components:** choose the simplest design that meets the requirements; reuse proven, supported products.
- **Redundancy and resilience:** remove single points of failure (dual power, RAID, clustering, load balancers, multiple sites).
- **Scalability:** design so more servers or resources can be added easily.
- **Security design:** authentication, authorization, encryption, network segmentation (DMZ), firewall rules, logging.
- **Dependencies:** list what the service needs (DNS, DHCP, directory, network, power) and what needs it.
- **Vendor and technology selection:** compare options (cost, support, skills, licensing, open source vs. commercial).
- **Monitoring, backup and maintainability:** plan them at design time, not after launch.
- **Documentation:** architecture diagrams, configuration, procedures and contact list.
- **Prototype / pilot:** test the design on a small scale first.

Example 2.4 — Removing single points of failure

A web service runs on one server, one disk and one switch port. What can be done to improve availability?

Solution

Use two servers behind a load balancer, RAID disks, redundant power supplies with UPS, dual network links connected to different switches, and a database replica. Now one failure does not stop the service.

2.3.3 Service Launch

Launching means moving the service from the design to real users safely. A careless launch can cause outages and lost trust.

- **Build and test environments:** use separate development, test/staging and production environments.
- **Testing:** functional, load/performance, security and failure tests; user acceptance testing (UAT).
- **Pilot / phased rollout:** launch first to a small group or one department, fix problems, then extend (avoid a single ‘big bang’).
- **Rollback plan:** a tested way to return to the old system if the launch fails.
- **Communication:** inform users about date, downtime, changes and how to get help; provide training and user guides.
- **Support readiness:** help desk trained, monitoring active, backups working, documentation complete, on-call staff assigned.
- **Change management:** obtain approval, choose a maintenance window and record the change.
- **Post-launch review:** check monitoring, collect feedback and fix issues; hand over to routine operations.

Rollout approach	Description	Risk
Big bang	Everyone switches at once	High
Phased / staged	Group by group or site by site	Low-medium
Parallel run	Old and new systems run together for a time	Low, but higher cost
Pilot	Small test group first	Low

2.3.4 Disaster Recovery

A **disaster** is any event that seriously interrupts services: fire, flood, earthquake (very relevant in Nepal), power failure, hardware failure, cyber attack (ransomware), human error or loss of a site. **Disaster recovery (DR)** is the set of plans, tools and processes to restore IT services and data after such an event. It is part of the wider **Business Continuity Plan (BCP)**.

Key Terms

Term	Meaning
RPO – Recovery Point Objective	How much data loss is acceptable, measured in time (e.g., 1 hour of data). Decides backup/replication frequency.
RTO – Recovery Time Objective	How quickly the service must be restored after failure (e.g., within 4 hours).
Backup	Copy of data used to restore after loss or corruption.
Replication	Continuous copy of data to another system or site.
Hot site	Fully equipped duplicate site ready almost immediately; highest cost.
Warm site	Site with hardware and network but data must be restored; medium cost/time.
Cold site	Empty space with power and connectivity; equipment must be brought in; low cost, slow recovery.
Failover	Automatic or manual switch to a standby system.

Steps in Disaster Recovery Planning

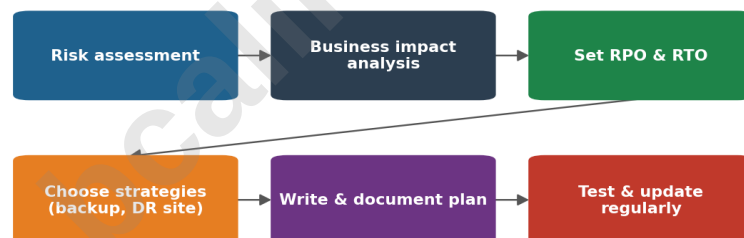


Figure 2.4 — Disaster-recovery planning process

- **Identify risks and critical services:** which services hurt most if lost (authentication, e-mail, finance system)?
- **Prioritize recovery order:** restore dependencies first (network, DNS, directory) then applications.
- **Backup strategy:** follow the 3-2-1 rule — 3 copies, 2 different media, 1 copy off-site (or offline/immutable against ransomware).
- **Off-site or cloud DR:** keep data and, if needed, standby systems in another location, away from the same earthquake or flood zone.
- **Redundant power and environment:** UPS, generator, fire suppression, water and temperature sensors.

- **Written runbooks and contact lists:** step-by-step recovery procedures, roles and responsibilities.
- **Regular testing:** restore drills and DR exercises; an untested backup is not a backup.
- **Post-incident review:** learn from real incidents and update the plan.

Example 2.5 — Choosing RPO/RTO

A college results-portal must not lose more than 15 minutes of data and must be back within 2 hours. Nightly tape backup is currently used. Is this enough?

Solution

No. A nightly backup gives an RPO of up to 24 hours. To meet RPO = 15 min, use database replication or log shipping to a standby server (or frequent incremental backups). To meet RTO = 2 hours, prepare a warm standby with documented failover steps and test it.

Note: Distinguish carefully: backup protects against data loss; high availability protects against downtime; disaster recovery restores service after a major failure. A complete design needs all three.

Key Takeaways — Unit II

- ✓ Workstation models: fat client, thin client, VDI, network boot, BYOD. Standardize hardware and OS images; keep data separate from the OS.
- ✓ Automate OS installation using answer files, imaging and PXE deployment; join the domain and apply policy centrally.
- ✓ Servers need enterprise features: ECC RAM, RAID, redundant PSU/fans, NIC teaming, remote management (iLO/iDRAC), rack form factor.
- ✓ RAID 5 usable = $(n - 1) \times \text{size}$; RAID 6 = $(n - 2) \times \text{size}$; RAID 10 = $n \div 2 \times \text{size}$.
- ✓ Service life-cycle: requirements → planning/engineering → launch → operation → disaster recovery.
- ✓ Requirements include functional, non-functional, operational needs and an SLA; availability 99.9% $\approx$ 8.76 hours downtime a year.
- ✓ Launch safely: test environments, pilot/phased rollout, rollback plan, user communication, support readiness.
- ✓ DR: define RPO/RTO, use the 3-2-1 backup rule, off-site copies, hot/warm/cold sites, written plans and regular tests.

— End of Unit 2 —