IT INFRASTRUCTURE

Lecture Notes (BS Level – HEC Pattern)

FAHAD NAEEM
GOVT. GRADUATE COLLEGE OF SCIENCE
Near PTCL Exchange Chungi No. 6 Bosan Road, Multan

Contents

Course Outline

Core Concepts: Definition of IT Infrastructure, Non-functional Attributes, Availability Concepts, Sources of Unavailability, Availability Patterns, Performance, Security Concepts, Data Centers
Servers: Availability, Performance, Security
Networking: Building Blocks, Availability, Performance, Security
Storage: Availability, Performance, Security
Virtualization: Availability, Performance, Security
Operating Systems: Building Blocks, Implementing Various OSs, Availability, Performance, Security
End User Devices: Building Blocks, Device Availability, Performance, Security
Management: IT Infrastructure Management
Processes: Service Delivery Processes, Service Support Processes
Broader Context: Ethics, Trends, organizational and technical issues related to IT infrastructure.

IT Infrastructure

Definition of IT Infrastructure

Think of IT infrastructure as the foundation beneath a city. Just as a city needs water pipes and electricity lines to function, IT applications need a foundation to run.

- **What it is:** It is the set of hardware, software, networks, and facilities required to develop, test, deliver, monitor, control, or support IT services.
- **Perspective matters:** It looks different depending on who you ask. To a business analyst, the IT systems are infrastructure. To a systems manager, the building and electricity are the infrastructure.

Non-functional Attributes

These are the "quality" attributes of a system. They describe *how* a system works, rather than *what* it does.

- **Function vs. Quality:** A car's function is to drive from A to B. Its non-functional attributes are safety, comfort, and reliability.
- **The Big Three:** In IT infrastructure, the three most critical attributes are **Availability** (is it working?), **Performance** (is it fast enough?), and **Security** (is it safe?)

Availability Concepts

Availability is the guarantee that a system is accessible when needed.

- **The "Nines":** Availability is measured in percentages. "Three nines" (99.9%) allows for about 8.8 hours of downtime a year. "Five nines" (99.999%) allows for only 5 minutes of downtime a year.
- **The Equation:** It is calculated using **MTBF** (Mean Time Between Failures—how long it runs before breaking) and **MTTR** (Mean Time to Repair—how long it takes to fix).
- **Goal:** You want high MTBF (reliable parts) and low MTTR (fast repairs).

Sources of Unavailability

Why do systems go down?

- **Human Error:** This is the #1 cause (approx. 80% of outages). Examples include pulling the wrong cable, typing a command incorrectly, or misconfiguring a router.
- **Software Bugs:** Errors in the code that crash the system.
- **Physical Defects:** Moving parts break. Fans, hard disks, and power supplies wear out or fail due to heat and vibration.
- **Environmental:** Power failures, cooling failures, fire, or floods.
- **Complexity:** Ironically, adding too many complex backup systems can sometimes cause *more* outages than a simple system.

Availability Patterns

These are design strategies to keep systems running.

- **Eliminate SPOFs:** A Single Point of Failure (SPOF) is one part that, if broken, stops the whole system. You must identify and remove them.
- **Redundancy:** having duplicates of critical parts (like two power supplies in one server).
- **Failover:** The automatic switching to a standby system if the primary one fails.

- **Fallback (Disaster Recovery):** Moving operations to a different location if the main datacenter is destroyed. This can be a **Hot site** (fully ready to go), a **Warm site** (needs data restoration), or a **Cold site** (empty room with power, no computers).

Performance

Performance is a "hygiene factor"—people only notice it when it is bad.

- **Perceived Performance:** This is how fast the system *feels* to the user. Predictability is key; users hate systems that are fast one minute and slow the next. Techniques like progress bars help users tolerate waiting.
- **Bottlenecks:** Every system has a limit. The slowest part of the system is the "bottleneck." If you fix one bottleneck, the limit just moves to the next slowest part.
- **Scalability:** The ability to handle more work. **Vertical scaling** means making a single computer bigger (more CPU/RAM). **Horizontal scaling** means adding more computers to share the work.

Security Concepts

Security manages risks to information. It is often defined by the **CIA Triad**:

- **Confidentiality:** Keeping data secret from unauthorized people.
- **Integrity:** Ensuring data hasn't been changed or corrupted.
- **Availability:** Ensuring data is accessible to authorized users.
- **Defense in Depth:** Using multiple layers of security (like a fence, a locked door, and a safe) so if one fails, the others still protect the asset.

Data Centers

This is the physical "home" for the infrastructure.

- **Purpose:** They provide power, cooling, fire suppression, and physical security for servers and storage.
- **Tiers:** Datacenters are rated by tiers (1 to 4). Tier 1 is basic (no redundancy), while Tier 4 is fault-tolerant (everything is duplicated, guaranteeing 99.995% availability.
- **Efficiency:** Measured by **PUE** (Power Usage Effectiveness). It compares total power used by the building vs. power used by the actual computers. A lower PUE is better.

Does that breakdown help clarify the landscape? We can dig deeper into any of these specific sections if you like.

Servers

A **server** is a powerful computer system or software application that provides services, resources, or data to other computers (called *clients*) over a network. A server is designed to **manage network resources**, store data, run applications, and respond to requests from users or other systems.

Availability

Availability for servers is about preventing crashes and minimizing downtime when hardware fails.

- **Hot Swappable Components:** Many modern servers allow you to replace parts like hard disks, power supplies, and fans while the server is still running. The operating system must be aware of this capability to use the new resources without rebooting.
- **Memory Protection (ECC):** Servers typically use Error Correction Code (ECC) memory. Unlike standard RAM, ECC memory can detect and fix single-bit errors (where a 0 accidentally flips to a 1) to prevent crashes.

- **Lock stepping:** For extremely critical systems, two servers can run in "lockstep," performing the exact same calculations simultaneously. If one fails or makes an error, the other continues without interruption.
- **Virtualization Failover:** If a physical server fails, the virtualization software (hypervisor) can automatically restart the virtual machines on a spare physical server. This protects against hardware failure but not against software crashes inside the virtual machine.

Performance

Server performance depends on the architecture, CPU speed, and memory.

- **Moore's Law:** Historically, the number of transistors on a CPU chip doubles approximately every two years, which generally doubled performance every 18 months. However, physical limits (like heat) have shifted the focus from raw speed (GHz) to adding more "cores" (processing units) to a single chip.
- **Multi-core & Hyper-threading:** To increase speed without melting the chip, modern CPUs use multiple cores to do parallel work. Hyper-threading tricks the system into thinking one physical core is actually two, keeping the processor busier and more efficient.
- **Caching:** Because main memory (RAM) is slower than the CPU, processors use "Cache" (super-fast memory located directly on the CPU) to store frequently used data. This drastically speeds up calculations.
- **Virtualization Overhead:** Running virtual machines adds a small performance penalty (usually less than 10%). A common bottleneck is "I/O" (Input/Output)—if too many virtual machines try to access the network or disk at once, the physical server can get overwhelmed.

Security

Server security involves protecting both the physical hardware and the virtual environments.

- **Physical Security:** Simple physical measures are critical. You should disable external USB ports (to prevent data theft or viruses) and password-protect the BIOS so unauthorized people cannot change boot settings.
- **Hypervisor Security:** In a virtualized environment, the hypervisor (the software managing the VMs) is a prime target. If a hacker compromises the hypervisor or the management console, they effectively control every server running on that machine. These management tools must be strictly isolated and monitored.
- **Hardening:** Whether physical or virtual, the operating system must be "hardened." This involves removing unnecessary software, closing unused network ports, and installing the latest security patches to reduce the "attack surface".

Networking

Networking refers to the interconnection of computers, servers, devices, and systems so they can communicate, share resources, and exchange data within an organization or over the internet.

Building Blocks

Networks are built using a standard model called the **OSI Model**, which splits communication into 7 layers. This allows different hardware and software to work together.

- **Physical Layer (Cables & Wireless):** This is the actual hardware connecting computers.

- **Cables:** We use **UTP** (copper cables, like standard Ethernet), **Coax** (like TV cables), and **Fiber Optics** (glass strands using light for long distances and high speed).
- **Wi-Fi:** Uses radio waves to connect devices without wires.
- **Data Link Layer (Switching):** This layer handles moving data between devices on the same network.
- **Ethernet:** The standard language for wired networks. It uses **MAC addresses** (unique IDs burned into network cards) to deliver data.
- **Switches:** Devices that connect computers. Unlike old "hubs" that shouted data to everyone, switches smartly send data only to the specific computer that needs it.
- **Network Layer (Routing):** This layer moves data between *different* networks (like from your house to the internet).
- **IP Protocol:** Uses **IP Addresses** (like 192.168.1.1) to find devices. **IPv4** is the old standard (running out of addresses), and **IPv6** is the new standard with unlimited addresses.
- **Routers:** The traffic cops that decide the best path for data to travel across the world.
- **Transport Layer (Reliability):**
- **TCP:** Guarantees data arrives correctly (like a registered letter). Used for web pages and emails.
- **UDP:** Sends data fast without checking if it arrived (like a postcard). Used for streaming video where speed matters more than perfection.
- **Application Services:** Helper services that make networks usable.
- **DNS:** The phonebook of the internet. It translates names (like www.google.com) into IP addresses computers understand.
- **DHCP:** Automatically assigns IP addresses to devices when they join a network.

Availability

Availability ensures the network stays online even when things break.

- **Redundancy:** The golden rule is "two of everything." If one cable or switch fails, another takes over 15.
- **Layered Topology:** Networks are built in layers (Core, Distribution, Access). This structure ensures that if a small switch in an office break, the main network keeps running.
- **Spanning Tree Protocol (STP):** If you connect multiple cables for redundancy, you might accidentally create a loop (data circling forever). STP automatically blocks redundant paths and only opens them if the main path breaks.
- **Teaming:** You can plug multiple network cables into one server. If one cable is unplugged, the server keeps running on the others.
- **Multihoming:** Connecting a network to two different Internet Service Providers (ISPs). If one ISP goes down, the other keeps you online.

Performance

Performance is about speed (Bandwidth) and delay (Latency).

- **Bandwidth vs. Throughput:** Bandwidth is the theoretical speed limit of the cable (e.g., 1 Gbps). Throughput is the actual speed you get after overhead and errors.
- **Latency:** The time it takes for a signal to travel from A to B. It depends on distance (speed of light) and how many devices it passes through. High latency makes connections feel "laggy".

- **Quality of Service (QoS):** This gives "VIP status" to important traffic. For example, voice calls (VoIP) are given priority over file downloads so your phone call doesn't stutter even if someone is downloading a huge file.
- **WAN Compression:** Compressing data before sending it over slow long-distance links (WAN) to make it arrive faster.

Security

Network security protects data from unauthorized access.

- **Firewalls:** The bouncers of the network. They sit between the safe internal network and the dangerous internet, blocking unauthorized traffic based on rules.
- **DMZ (De-Militarized Zone):** A buffer zone. Servers that must be accessed from the internet (like a web server) are placed here. If a hacker breaks into the DMZ, they still can't easily get into the internal network.
- **IDS/IPS:** Intrusion Detection/Prevention Systems are like security cameras and guards. They watch for suspicious patterns (like a hacker scanning for open doors) and alert administrators or block the attack.
- **VPN (Virtual Private Network):** Creates a secure, encrypted tunnel over the public internet, allowing remote users to safely connect to the office network.
- **Network Access Control (NAC):** Checks devices before letting them in. If a laptop doesn't have an updated virus scanner, NAC blocks it from the main network until it is fixed.

Storage

Storage refers to the hardware and systems used to save, manage, protect, and retrieve digital data within an organization. Storage allows data such as files, databases, applications, and backups to be securely kept and accessed when needed.

Availability

Availability ensures your data is there when you need it and don’t get lost if a disk brakes.

- **RAID (Redundant Array of Independent Disks):** This allows you to combine multiple physical disks into one "virtual" disk to protect against failure.
- **RAID 1 (Mirroring):** You save data on two disks at once. If one breaks, the other has a perfect copy. It is very safe but expensive because you need double the storage.
- **RAID 5 (Striping with Parity):** Data is spread across multiple disks along with "parity" (a calculated recovery code). If one disk fails, the system uses the parity to calculate the missing data. It is cheaper than mirroring but slower to fix if a disk brakes.
- **RAID 6:** Similar to RAID 5 but generates two parity blocks. It can survive **two** simultaneous disk failures.
- **RAID 10:** Combines speed and safety by mirroring data (RAID 1) and then spreading it across sets of disks (RAID 0). It is very fast and reliable but expensive.
- **Replication:** Copying data to a second storage system, often in a different location.
- **Synchronous:** Data is written to both locations at the exact same time. It guarantees zero data loss but can slow down the system if the locations are far apart.
- **Asynchronous:** Data is written to the main system first and copied to the backup later. It is faster but risks losing recent data if the main system crashes before copying.

- **Snapshots:** A "frozen" picture of your data at a specific moment. It allows you to instantly revert to a previous version if a file is corrupted or deleted, without needing a full restore from backup.

Performance

Performance is measured by how fast the storage can read and write data.

- **IOPS (Input/Output Operations Per Second):** This is the standard speed limit measurement. It tells you how many separate read/write actions a disk can handle in one second.
- **Mechanical Disks (HDD):** These use spinning platters. They are slow because a physical arm has to move to find data (seek time) and wait for the disk to spin (rotational delay). A fast 15,000 RPM drive might get ~160 IOPS.
- **Solid State Drives (SSD):** These use flash memory chips with no moving parts. They are incredibly fast, offering thousands of IOPS.
- **Caching:** A controller uses a small amount of super-fast RAM (Cache) to temporarily hold data before writing it to the slower disks. This makes the storage *feel* instant to the user, even if the disks are still working in the background.
- **Tiering:** This is an efficiency strategy. The system automatically moves "hot" data (used often) to fast SSDs (Tier 1) and moves "cold" data (rarely used) to slow, cheap disks (Tier 2). This gives you speed where you need it without paying for SSDs for everything.
- **RAID Penalty:** Some RAID types slow down writing. RAID 5 and 6 have a "write penalty" because the system has to read old data and calculate the new parity code every time you save a file.

Security

Security focuses on protecting data from theft or unauthorized access.

- **Data at Rest Encryption:** Encrypting the physical disks. If a hard drive is stolen from the datacenter, the thief cannot read it without the key. Self-Encrypting Drives (SED) handle this automatically.
- **Zoning (in SANs):** In a storage network, Zoning acts like a security fence. It groups servers and storage so that a server can only "see" and talk to the specific storage devices it is assigned to. This prevents a hacked server from attacking all the disks in the network.
- **LUN Masking:** This hides specific "virtual disks" (LUNs) from servers that shouldn't see them. Even if a server is in the right Zone, masking ensures it can't touch data belonging to another server.
- **Wiping:** When disks are retired, simply deleting files isn't enough. Professional wiping or physical destruction is required to ensure old data cannot be recovered by strangers.

Virtualization

Virtualization is a technology that allows multiple virtual systems (virtual machines) to run on a single physical hardware system by dividing its resources (CPU, RAM, storage, network). It enables better utilization of hardware, cost reduction, and flexible system management.

Availability

Virtualization uses a software layer (the **hypervisor**) to separate the operating system from the physical hardware. This creates powerful options to keep systems running.

- **Failover Clustering:** If a physical server fails, the virtualization software can automatically restart the affected virtual machines (VMs) on a spare physical server. This protects against hardware failure. It can also detect if a VM's operating system crashes (e.g., a "Blue Screen") and automatically reboot that specific VM.
- **Live Migration:** You can move a *running* virtual machine from one physical server to another with zero downtime. This is crucial for maintenance; you can empty a physical server, upgrade its hardware, and move the VMs back without users noticing.
- **Lock stepping (Fault Tolerance):** For critical systems, you can run two identical VMs on different physical servers in "lockstep." They perform the exact same tasks at the same time. If one physical server dies, the other VM continues instantly with no interruption.
- **Snapshots:** Before applying a risky software patch, you can take a "snapshot" of the VM. If the update breaks the system, you can revert to the snapshot state instantly.

Performance

Virtualization allows you to do more with less hardware, but it introduces specific performance challenges.

- **The "Tax":** Virtualization adds a small performance penalty (overhead), usually less than 10%, because the hypervisor has to translate instructions between the VMs and the hardware.
- **I/O Bottlenecks:** Consolidating many servers onto one physical machine saves CPU power, but all those VMs act like a "funnel" for disk and network traffic. If too many VMs try to read data at once, the Input/Output (I/O) becomes the bottleneck.
- **Memory Optimization:**
- **Memory Sharing:** If you run 10 Windows VMs, they all use the same system files in memory. The hypervisor smartly stores these common files only once in physical RAM and shares them, freeing up space.
- **Overcommit:** You can assign more RAM to your VMs than actually exists physically, betting that not everyone will need maximum RAM at the same time.
- **Raw Device Mapping (RDM):** For high-performance databases that need raw speed, you can bypass the hypervisor's storage layer and let the VM talk directly to the hard disk. This removes the virtualization overhead for storage.

Security

Virtualization changes the security landscape significantly. It adds new defenses but creates a major new target.

- **The Hypervisor Risk:** The hypervisor is the "brain" controlling everything. If a hacker breaks into the hypervisor, they effectively control *every* server running on that machine. Malware installed here (rootkits) is invisible to the virtual machines.
- **Management Console:** This is the tool used to create, delete, and move VMs. It is the "keys to the kingdom." If this is compromised, an attacker can copy sensitive hard drives or delete entire datacenters. It requires the highest level of security, such as two-factor authentication and strict access logging.
- **VM Sprawl:** Because it is so easy to create new VMs, admins often spin up test servers and forget about them. These "zombie" VMs sit unpatched and insecure, becoming easy entry points for attackers.

- **Physical Separation (DMZ):** For high-security zones (like the DMZ connected to the internet), it is often safer to use separate physical hardware. This ensures that if a web server is hacked, the attacker cannot simply "hop" through the hypervisor to access the internal database servers.

Operating System

An **Operating System (OS)** is system software that manages computer hardware, software resources, and provides services to applications and users. It acts as an interface between the user, applications, and the computer hardware.

Building Blocks

An operating system acts as a bridge between the physical hardware and the software applications users run. It hides the complexity of the hardware.

- **The Kernel:** This is the heart of the OS. It manages the most critical tasks: starting and stopping programs, managing the file system, and scheduling access to the hardware so two programs don't fight over the same resource.
- **Device Drivers:** These are small pieces of software that translate the OS's general commands into specific instructions for hardware, like a printer or a network card.
- **Utilities:** These are the "helper" applications built into the OS, such as the user interface (Shells/GUIs), configuration tools, text editors, and web browsers.
- **APIs (Application Programming Interfaces):** These allow software developers to ask the OS to do things (like "read a file" or "draw a window") without knowing the details of how the hardware actually does it.

Implementing Various Operating Systems

Different hardware platforms typically run different operating systems.

- **Mainframe OS (z/OS):** Used on IBM mainframes. It is designed for extreme backward compatibility (programs from 1974 still run today) and massive batch processing capabilities.
- **Midrange OS:**
- **IBM i (formerly OS/400):** Runs on IBM Power Systems. It is known for having a database and security built directly into the OS kernel, making it very complete and secure.
- **OpenVMS:** Originally from DEC, now mostly used on HP systems. It is legendary for its stability, often running for years without a reboot.
- **UNIX:** A powerful multi-user OS. Different vendors have their own "flavors" (e.g., IBM AIX, HP-UX, Oracle Solaris). Applications usually need to be rewritten to move between flavors.
- **x86 Server OS:**
- **Linux:** A UNIX-like OS that is open-source (free to use and modify). It is the engine behind most of the internet, supercomputers, and Android phones. Popular versions (distributions) include Red Hat, SuSe, and Ubuntu.
- **Windows:** The most popular OS for PCs and widely used for servers. Historically known for usability, it has evolved to become a stable platform for business applications like Exchange and SharePoint.

Availability

To keep operating systems running even when hardware or software fails, **Failover Clustering** is used.

- **Nodes & Resource Pools:** A cluster consists of multiple servers (nodes). An application runs on one node (active), while another node waits (passive).
- **Heartbeat:** The nodes continuously send "I'm alive" signals to each other over a dedicated network connection. If the heartbeat stops, the passive node assumes the active one died and takes over.
- **Shared Storage:** The data lives on a central disk system (SAN/NAS) accessible by all nodes. If the application moves to a new node, it just reconnects to the same disk.
- **Quorum:** To prevent "split-brain" (where two nodes lose contact and *both* think they should be the boss), a voting system is used. Often a shared "Quorum disk" acts as a tie-breaking vote.

Performance

Performance depends heavily on the hardware, but OS configuration plays a role.

- **Memory Management:** The OS decides which program gets to use the RAM.
- **Paging:** Moving data that hasn't been used in a while from fast RAM to the slow hard disk to free up space. This is normal.
- **Swapping:** When RAM is full, the OS is forced to constantly move active programs to the disk. This destroys performance and must be avoided by adding more RAM.
- **Disk Caching:** The OS uses any "free" RAM to store copies of frequently used disk data. This makes the system feel much faster because reading from RAM is instant compared to reading from a disk.
- **Kernel Tuning:** On systems like Linux, you can recompile the kernel to remove unused drivers and features. A smaller kernel starts faster and leaves more memory for applications.

Security

Securing the OS involves several layers of defense.

- **Patching:** Regularly installing software updates from the vendor. These include **Hot-fixes** (urgent security repairs) and **Service Packs** (bundled updates).
- **Hardening:** The process of stripping the OS down to the bare minimum. You switch off unnecessary services, remove unused accounts, and close open network ports to reduce the "attack surface".
- **Virus Scanning:** Essential for Windows and Linux servers to detect malware. Scanners should be configured to skip low-risk files (like massive database files) to avoid slowing down the system.
- **Host-based Firewalls:** A software firewall running on the OS itself. It acts as a second line of defense, blocking unwanted network traffic even if it made it past the main network firewall.
- **User Management:** Never use the "Administrator" or "Root" account for daily work. Limit user privileges so that if an account is hacked, the attacker cannot delete system files.

End User Device

End-User Devices are the hardware devices used by individuals (users) to access organizational systems, applications, data, and network resources. These devices act as the interface between users and the IT infrastructure.

Building Blocks

End user devices are the tools users touch daily. They fall into a few main categories:

- **Desktop PCs and Laptops:** The standard workhorses. PCs are powerful but fixed; laptops are portable but fragile. About 90% of x86 PCs run Microsoft Windows.
- **Mobile Devices:** Smartphones and tablets. They connect via public wireless networks (like 4G/5G) which are often less reliable than office networks. They have smaller screens and different keyboards, requiring applications to adapt.
- **Bring Your Own Device (BYOD):** A trend where employees use personal devices for work. This creates a conflict: users want freedom, but IT managers need control and security. Virtualization is often used to separate corporate data from personal data on the same device.

Printers:

- **Laser Printers:** Standard for high-quality text.
- **Multi-Functional Printers (MFPs):** Centralized machines that print, scan, copy, and fax. They often require users to identify themselves (with a badge) before printing to prevent sensitive papers from sitting in the tray.
- **Specialized Printers: Line printers** for high-speed massive printing (like invoices) and **Thermal printers** for receipts.

Availability

Availability for end-user devices is harder to guarantee than for servers because they move around and are treated roughly.

- **Physical Reliability:** End-user hardware is less reliable than datacenter hardware and has a lifespan of only 3-5 years. Laptops frequently break due to drops or spills.
- **Data Backup:** The biggest risk is data stored locally on the laptop's hard drive. If the laptop breaks or is lost, the data is gone. The solution is **Automated Synchronization**, where local files are instantly copied to a server whenever the device connects to the network 10.
- **The Human Factor:** Availability often fails because users simply forget to save their work before a crash. Users must be trained to save early and often.
- **Consumables:** A printer is "down" if it runs out of toner or paper. Service contracts and supply management are critical for printer availability.

Performance

Performance on these devices depends heavily on hardware specs and network connection.

- **RAM:** Adding more memory (RAM) is usually the most effective way to speed up a slow PC, often more effective than a faster CPU.
- **Storage Speed:** Replacing a mechanical hard disk with a Solid-State Drive (SSD) dramatically improves how fast the device feels (boot time, opening apps).
- **Network Bottlenecks:** For mobile devices, the performance bottleneck is usually the public wireless network (3G/4G). These connections fluctuate, causing apps to feel slow or unresponsive.

Security

Securing devices outside the locked datacenter is a major challenge.

- **Physical Security:** Laptops are easily stolen. Cable locks should be used to physically tie them to desks.
- **Disk Encryption:** Encrypting the entire hard drive is essential. If a laptop is stolen, the thief gets the hardware, but cannot read the business data.
- **Malware Protection:** Virus scanners are mandatory. They should automatically quarantine infected files and report back to a central system so administrators know which machines are infected.
- **Mobile Device Management (MDM):** Software used to manage phones and tablets. It allows administrators to remotely **wipe** (erase) a device if it is lost or stolen.
- **Limiting Permissions:** Users should generally **not** be Administrators on their own machines. This prevents them from accidentally (or intentionally) disabling antivirus software or installing dangerous programs.

The Infrastructure Lifecycle

Managing infrastructure isn't just about fixing things when they break. It follows a specific lifecycle:

1. **Deployment Option:** Deciding where and how to build it.
2. **Purchasing:** Buying the hardware and services.
3. **Build & Test:** Assembling the parts and making sure they work.
4. **Maintenance:** Keeping the system running (Operations).
5. **Application Deployment:** Putting software on the infrastructure.
6. **Decommissioning:** Safely removing the system at the end of its life.

Deployment Options (Where to put it)

Before buying anything, you must decide where the infrastructure will live:

- **On-Premises:** You build it in your own building. You control everything, but you also have to manage power, cooling, and space yourself.
- **Colocation:** You rent space in a professional datacenter. They provide power and cooling; you bring your own servers.
- **Outsourcing:** You pay a third party to handle everything. You don't own the hardware; you just pay for the service.

Purchasing

When buying complex infrastructure, two documents are critical:

- **Bill of Materials (BoM):** A detailed "shopping list" of every part needed, down to the specific cables and screws.
- **Statement of Work (SoW):** A document describing exactly who does what. For example, does the supplier just drop off the boxes, or do they also screw the servers into the racks and connect the power?

Deployment Models (How to build it)

You can build infrastructure in different ways:

- **Traditional:** You buy separate servers, storage, and network switches from different vendors and connect them yourself.

- **Converged:** You buy a pre-built rack where the computer, storage, and networking are already assembled and tested by one vendor.
- **Software-Defined Datacenter (SDDC):** All resources (CPU, storage, network) are virtualized and controlled by software. You can create a new "datacenter" with a few lines of code.
- **Cloud Computing (IaaS):** You rent virtual machines and storage from a provider (like Amazon or Azure) on-demand. You pay only for what you use.

Go-Live Scenarios

When the new system is ready, you need to switch from the old one to the new one. There are three main strategies:

- **Big Bang:** You switch off the old system and switch on the new one at the same time. It is fast but risky—if the new system fails, you are in trouble.
- **Parallel:** You run both systems at the same time for a while. It is safe because you can always go back, but it is expensive and requires double the work.
- **Phased:** You move users or functions to the new system step-by-step. It takes longer but reduces risk.

Maintenance & Operations

Once running, the system needs care. Several frameworks help organize this:

- **ITIL:** A set of standard processes for things like handling incidents (fixing broken things) and change management (safely updating things).
- **DevOps:** A modern approach where software developers and system operators work as one team. The philosophy is "If you build it, you run it".
- **Monitoring:** Using tools (like Nagios) to watch the system 24/7. It warns you if a disk is full or a server is too slow so you can fix it before users notice.
- **Logging:** Keeping a record of events (log files) to analyze problems after they happen or for security investigations.

Deploying Applications (DTAP)

You should never make changes directly on a production system. Instead, a **DTAP** street is used 15:

- **D**evelopment: Where software is written.
- **T**est: Where independent testers check if it works.
- **A**cceptance: Where the business users check if it does what they want.
- **P**roduction: The live system used by the company.

Decommissioning

When a system is too old, it must be removed safely. This involves:

- **Data Wiping:** Ensuring hard disks are erased or destroyed so no company secrets are found in the trash.
- **Cleanup:** Removing software licenses, stopping maintenance contracts, and updating the administration (CMDB) so you don't pay for things you no longer have.

Processes

In IT infrastructure management, processes are structured activities used to ensure IT services are delivered efficiently and supported properly within an organization. These processes are formally defined in frameworks like ITIL (Information Technology Infrastructure Library).

Service Support Processes (Day-to-Day Operations)

These processes focus on the daily support and maintenance of the IT services to ensure stability and user access.

Incident Management:

- **Goal:** To restore normal service operation as quickly as possible and minimize the adverse impact on business operations.
- **Context:** It is closely linked with **Business Continuity Management (BCM)** to handle critical incidents and crises.

Problem Management:

- **Goal:** To resolve the root cause of incidents to prevent them from recurring.
- **Context:** The text highlights that while incidents are about "fixing broken things" quickly, problem management investigates why they broke.

Change Management:

- **Goal:** To ensure that standardized methods and procedures are used for efficient and prompt handling of all changes to minimize the impact of change-related incidents.
- **Context:** This is critical because **human error** during changes (configuration, release) causes approximately **80%** of outages. Formal change processes (like **DTAP**: Development, Test, Acceptance, Production) are used to mitigate this risk.

Release Management:

- **Goal:** To plan, schedule, and control the build, test, and deployment of releases, and to deliver new functionality required by the business while protecting the integrity of existing services.
- **Context:** This includes activities like software updates and patching. In modern **DevOps** environments, this is often automated via **Continuous Delivery** pipelines.

Configuration Management:

- **Goal:** To maintain information about "Configuration Items" (CIs) required to deliver an IT service, including their relationships.
- **Context:** This information is stored in a **CMDB** (Configuration Management Database). It is vital during the "Assembling" and "Decommissioning" phases to track what assets exist and how they connect.

Service Delivery Processes (Long-Term Planning)

These processes focus on the long-term planning and improvement of IT service delivery.

Capacity Management:

- **Goal:** To ensure that the capacity of the IT infrastructure matches the evolving demands of the business in a cost-effective and timely manner.
- **Activities:** It involves monitoring resources (like disk space or CPU usage) to detect trends and purchasing hardware "just in time" to avoid waste. It requires input from business plans (e.g., upcoming marketing campaigns) to predict future load.

Availability Management:

- **Goal:** To ensure that the infrastructure services are available when the customer needs them.
- **Key Concepts:** This involves designing for **Redundancy** (eliminating Single Points of Failure), calculating **MTBF** (Mean Time Between Failures) and **MTTR** (Mean Time to Repair), and implementing failover mechanisms.

IT Service Continuity Management (ITSCM):

- **Goal:** To manage risks that could seriously impact IT services and ensure the business can recover from a disaster.
- **Key Concepts:** This includes **Business Continuity Management (BCM)** and **Disaster Recovery Planning (DRP)**. It defines the **RTO** (Recovery Time Objective - how fast you need to be back) and **RPO** (Recovery Point Objective - how much data you can lose).

Related Operational Processes

The text also highlights specific operational activities that support these processes:

- **Monitoring:** Continuously inspecting components (e.g., using Nagios or Zabbix) to detect errors or warning signs (like a full disk) before they cause downtime.
- **Logging:** Collecting and analyzing events from various components (servers, firewalls) to find trends, troubleshoot complex issues, or perform security forensics.

Broader Context

Trends in IT Infrastructure

The landscape is shifting from hardware-focused to software-focused.

- **Software-Defined Everything (SDx):** Traditional hardware (like switches and storage arrays) is being replaced or controlled by intelligent software. This includes **Software Defined Networking (SDN)**, **Software Defined Storage (SDS)**, and **Software Defined Compute (SDC)**. Together, they create a **Software Defined Datacenter (SDDC)**, where the entire facility is managed by code and automation.
- **Cloud Computing:** This is an outsourcing model where you pay for what you use. It includes **IaaS** (Infrastructure as a Service), which provides raw virtual machines, and **SaaS** (Software as a Service) like Office 365. It shifts costs from "Capital Expenses" (buying hardware) to "Operational Expenses" (monthly bills).
- **Hyperconverged Infrastructure:** Instead of buying separate servers, storage, and networks, organizations now buy "blocks" or "nodes" that contain all three. You just add more blocks to scale up. It simplifies management but can lead to "vendor lock-in" (you can't mix brands).
- **Containers:** technologies like **Docker** allow applications to run in isolated "boxes" that share the same operating system kernel. This uses far less memory and CPU than running full virtual machines for every application.

Organizational Issues

Managing infrastructure is often harder than building it.

- **The Human Factor:** People are the weakest link. **Human error** causes approximately **80%** of outages. Common mistakes include pulling the wrong cable, typing commands incorrectly, or misconfiguring routers.
- **Silos vs. DevOps:** Traditionally, separate teams managed servers, networks, and storage (Silos). This caused delays. The **DevOps** trend merges these roles ("If you build it, you run it"), requiring staff to have broader skills rather than just deep specialization.
- **Shadow IT & BYOD:** Users often bring their own devices (**BYOD**) or buy their own cloud services because IT is "too slow." This creates a conflict: users want freedom, but IT managers need control to ensure security and compliance.

- **Outsourcing:** Organizations must decide whether to keep knowledge in-house (On-Premises) or outsource it to the cloud. Outsourcing saves money but requires strong contract management skills rather than technical skills.

Technical Issues

- **Complexity:** Adding more backup systems to increase availability can ironically cause *more* downtime because complex systems are harder to manage and fix. Sometimes simple is better.
- **Legacy Systems:** Old systems (like **Mainframes**) are incredibly reliable but expensive. **Windows** systems historically struggled with stability because they tried to remain backward-compatible with very old software.
- **Bottlenecks:** Every system has a limit. As soon as you fix one bottleneck (e.g., slow hard drives), another one appears (e.g., network speed). Performance tuning is a never-ending game of finding the next bottleneck.
- **Data Growth: Kryder's Law** suggests storage density doubles every 13 months. Infrastructure must handle massive data growth (Big Data), requiring smart strategies like **Tiering** (moving old data to cheap tape/disk) and **Deduplication** (storing only unique data).

Ethics, Law, and Regulation

Infrastructure isn't just about machines; it's about responsibility.

- **Data Privacy & Retention:** Laws often dictate how long data must be kept. For example, medical records might need to be kept for **130 years** (lifetime of patient + 30 years). Infrastructure must guarantee data is readable decades later, requiring open file standards (like XML) rather than proprietary ones.
- **Power & Responsibility:** Systems administrators have "keys to the kingdom." They can read everyone's email and see all files. Ethical guidelines and "separation of duties" (requiring two people to approve critical actions) are essential to prevent abuse.
- **Security vs. Privacy:** Technologies like **Mobile Device Management (MDM)** allow companies to wipe a stolen phone. However, if the phone is a personal device (BYOD), wiping it destroys personal photos and data, raising ethical and legal issues.