DEVOPS

Training Material

TABLE OF CONTENTS

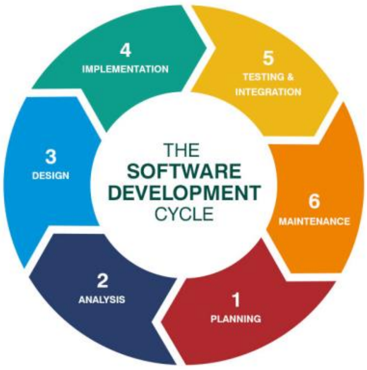
CHAPTER 1: SOFTWARE DEVELOPMENT

1.1 Software Development Overview

- **Software development** is the process of creating software applications that meet specific user or business needs. This process includes several stages, from conception to deployment and maintenance.
- The primary goal is to build functional, scalable, and maintainable software products that fulfill users' requirements and solve real-world problems.
- It involves several disciplines such as programming, project management, and user experience (UX) design.
- Software development can produce a wide range of applications, including web applications, mobile apps, desktop software, and enterprise systems.

1.2 Software Development Lifecycle (SDLC)

- **The Software Development Lifecycle (SDLC)** ensures that software development follows a systematic approach, minimizing risks, and improving efficiency. Each phase has specific objectives and deliverables:



  - **Planning**: The requirements and scope are gathered and analyzed. Teams define project goals, schedules, and necessary resources.
  - **Analysis**: In-depth examination of requirements to create detailed project specifications. This phase often includes feasibility studies and risk analysis.
  - **Design**: Architects and designers create a blueprint of the software's architecture, user interface (UI), and data flow. This step ensures that the system will meet performance and scalability requirements.
  - **Implementation (Development)**: Developers write code to build the system based on the design specifications.

    - **Testing**: The software is rigorously tested for bugs, performance issues, security vulnerabilities, and overall functionality. Testing ensures that the software meets quality standards before release.
    - **Deployment**: The final product is deployed to the production environment, making it available to users. Deployment includes setting up the infrastructure and ensuring that the system operates correctly.
    - **Maintenance**: After deployment, the software requires ongoing maintenance to fix bugs, add new features, or make improvements based on user feedback and evolving needs.

1.3 Frameworks

- **Frameworks** provide predefined structures and tools to streamline software development processes. They help developers focus on functionality by offering reusable components and best practices. Common frameworks include:
    - **Web Development**:
        - **Django (Python)**: A high-level Python web framework that encourages rapid development and clean, pragmatic design.
        - **React (JavaScript)**: A popular library for building user interfaces, primarily for single-page applications.
        - **Angular (JavaScript)**: A full-featured framework developed by Google for building dynamic web applications.
    - **Mobile Development**:
        - **React Native**: Allows developers to build native mobile apps using JavaScript and React.
        - **Flutter**: A Google-backed framework for building natively compiled applications for mobile, web, and desktop from a single codebase.
        - **Swift (iOS)**: A powerful and intuitive language for building apps for Apple devices.
        - **Kotlin (Android)**: A modern programming language used for Android app development, known for its concise syntax and interoperability with Java.
    - **Enterprise Applications**:
        - **Spring (Java)**: A comprehensive framework that simplifies enterprise Java development by providing a robust set of tools for handling infrastructure tasks.

        - **.NET (C#)**: A framework developed by Microsoft that supports the development of Windows applications, web apps, and cloud services.

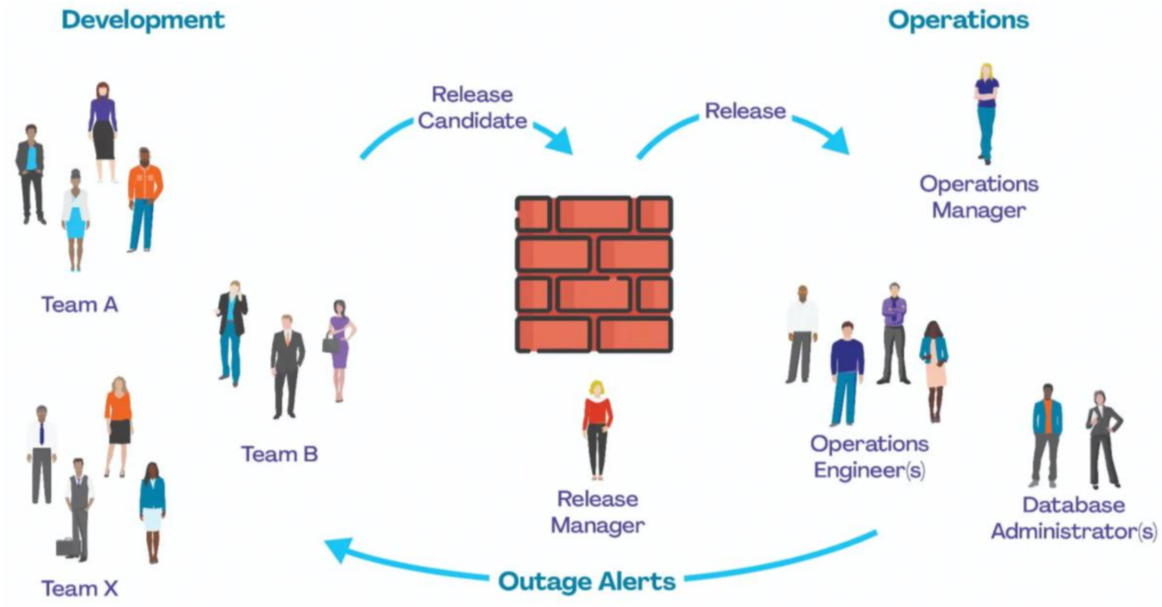
1.4 Software Development Methodologies

- **Software development methodologies** guide the workflow and project management strategies used to deliver software products. Different methodologies are chosen based on project requirements, timelines, and team structure:
    - **Waterfall Model**:
        - This is a **linear and sequential** development process where each phase must be completed before moving on to the next.
        - It is typically used in projects with well-defined requirements and minimal expected changes. The major drawback is its inflexibility in handling changes once the project has progressed beyond a phase.
    - **Agile Development**:
        - Agile promotes an **iterative and incremental** approach where projects are broken down into smaller units called sprints, typically 2-4 weeks long.
        - Teams deliver working software frequently, making it possible to adapt to changes in requirements. Agile methodologies prioritize customer collaboration, flexibility, and responsiveness.
    - **Lean Development**:
        - Lean focuses on **eliminating waste** by only implementing features that provide direct value to the customer. This methodology emphasizes maximizing efficiency, minimizing resources, and improving overall value.
        - It originated from lean manufacturing and applies principles like just-in-time development and continuous improvement (kaizen) to software engineering.
    - **Rapid Application Development (RAD)**:
        - RAD emphasizes **prototyping and user feedback** over strict planning and design.
        - It is ideal for projects that need to be developed quickly and where requirements might evolve during development. RAD often involves reusable components and focuses on speed and flexibility

CHAPTER 2: IT OPERATIONS

2.1 IT Operations Overview

- **IT Operations** encompass all activities involved in managing and maintaining an organization's technology infrastructure. This includes ensuring that systems, networks, applications, and data are available, reliable, and secure at all times.
- The main objective is to deliver consistent IT services that meet business requirements while optimizing performance and reducing downtime. IT operations play a critical role in supporting day-to-day business functions by providing necessary technical resources and ensuring that they are functioning effectively.



2.2 Core Components of IT Operations

IT operations revolve around several core components that form the foundation of an organization's technology landscape:

- **Infrastructure Management**:
  - **Servers (Physical and Virtual)**:
    - Servers, whether hosted on-premises or in the cloud, form the backbone of an IT infrastructure. They store and process data, run applications, and manage network resources. Virtual servers allow for better resource utilization and scalability, while physical servers provide more control over hardware.
  - **Networking Devices (Routers, Switches, Firewalls)**:
    - These devices facilitate communication between systems and users by transmitting data across networks. Routers connect different networks, switches manage data traffic within a network, and

        firewalls provide security by controlling incoming and outgoing traffic.

  - **Data Storage (NAS, SAN, Cloud Storage Solutions)**:
    - Data storage systems are essential for preserving critical data. Network-Attached Storage (NAS) and Storage Area Networks (SAN) offer centralized storage for large volumes of data, while cloud storage provides scalability, backup, and accessibility from any location.

- **Security Management**:
  - **Endpoint Protection**:
    - Endpoint protection includes securing all devices (servers, desktops, mobile devices) within a network. This is achieved through antivirus software, firewalls, and advanced solutions like Endpoint Detection and Response (EDR) to monitor and respond to security threats in real-time.
  - **Access Control (RBAC, MFA)**:
    - Role-Based Access Control (RBAC) ensures that users can only access the information and resources necessary for their role, thereby minimizing unauthorized access. Multi-Factor Authentication (MFA) adds an extra layer of security by requiring users to provide two or more verification factors before accessing sensitive systems.
  - **Security Audits**:
    - Regular security audits involve reviewing and assessing security policies, controls, and systems to ensure compliance with industry standards and regulations such as ISO 27001, GDPR, and HIPAA. These audits help identify vulnerabilities and areas for improvement.

2.3 Backup and Disaster Recovery

- **Backup and disaster recovery** are essential to ensuring data integrity and availability in the event of system failures, cyber-attacks, or natural disasters:
  - **Regular Backups**:
    - Regular, automated backups of critical systems and data are crucial to mitigate the risk of data loss. These backups can be stored on-premises or in the cloud, ensuring that organizations can restore data quickly.

    - **Tools**:
        - **Veeam**, **Acronis**, and **Bacula** are popular backup tools that offer robust features such as automated backup schedules, real-time replication, and recovery options for virtual and physical environments.
    - **Disaster Recovery Plans (DRP)**:
        - A DRP outlines the processes and procedures to restore IT systems and data in the event of a major outage or disaster. Regular testing and updating of DRPs are critical to ensuring that they remain effective in real-world scenarios.

2.4 Incident and Problem Management

- **Incident and problem management** focus on minimizing the impact of IT disruptions and preventing future issues:
    - **Incident Response**:
        - This process involves detecting and responding to IT incidents, such as system crashes, network failures, or security breaches. A well-defined incident response plan helps organizations resolve issues quickly, minimizing downtime and service interruptions.
    - **Root Cause Analysis**:
        - Root cause analysis identifies the underlying causes of recurring problems to implement long-term solutions. This process helps eliminate persistent issues and improves overall system reliability.

CHAPTER 3: SYSTEM ADMINISTRATION

3.1 System Administration Overview

- **System administration** involves managing and maintaining an organization’s computer systems, servers, and networks. The primary goal of system administration is to ensure that IT systems run smoothly and securely, minimizing downtime and optimizing performance.
- System administrators (SysAdmins) are responsible for configuring hardware and software, troubleshooting technical issues, and implementing security policies to protect the organization’s digital assets.

3.2 Common Tools Used in System Administration

System administrators use various tools to manage and monitor systems effectively:

- **Operating System Tools**:
  - **Event Viewer (Logs)**:
    - A Windows tool used to view detailed event logs, helping administrators troubleshoot and identify the root cause of system issues.
  - **Task Manager (Performance)**:
    - A tool used to monitor system performance, including CPU usage, memory consumption, and active processes. It helps identify resource bottlenecks.
  - **PowerShell (Automation)**:
    - PowerShell is a scripting language and automation tool for Windows that allows administrators to automate repetitive tasks, such as user account management and system configuration.

3.3 Monitoring and Alerting Tools

- Monitoring tools help system administrators ensure that systems are functioning optimally and alert them to any potential issues:
    - **Nagios**:
        - An open-source monitoring tool that provides real-time server monitoring, network device health, and system performance. It sends alerts when predefined thresholds are breached.
    - **Zabbix**:
        - Another open-source monitoring solution that tracks server performance, application metrics, and network traffic. It is scalable and supports automated alerts and reports.
    - **SolarWinds**:
        - A suite of IT management tools that provides detailed visibility into system performance, network activity, and application status, enabling real-time diagnostics and troubleshooting.

3.4 Configuration Management Tools

- Configuration management tools automate the process of managing, monitoring, and deploying system configurations across multiple environments:
    - **Ansible**:
        - An open-source automation tool that allows administrators to automate configuration management, application deployment, and server orchestration across large-scale infrastructures.
    - **Puppet**:
        - Puppet enables administrators to define system configurations in code and apply them consistently across multiple machines, ensuring that the infrastructure stays in the desired state.
    - **Chef**:
        - Chef automates the deployment and management of infrastructure by treating infrastructure as code. It simplifies complex tasks and ensures that systems remain consistent across environments.

3.5 Security Tools

- Security tools play a crucial role in protecting IT systems from unauthorized access, malware, and other security threats
    - **Firewall Configuration Tools**:
        - **iptables (Linux)**: A Linux-based firewall tool that helps administrators control inbound and outbound network traffic by defining filtering rules.
        - **Windows Firewall (Windows)**: A built-in firewall solution in Windows that blocks unauthorized access and prevents malware infections.
    - **Antivirus Solutions**:
        - **McAfee**, **Symantec**, and **Bitdefender** are popular antivirus solutions that detect and protect against malware, viruses, and other malicious software. They provide real-time protection and regular security updates to guard against evolving threats.

CHAPTER 4: NETWORKING

4.1 Networking Overview

- **Networking** refers to the process of connecting computers, servers, and other devices to share data, resources, and applications across local or wide geographic areas. Networking plays a critical role in both business environments and personal computing, allowing users to communicate, collaborate, and share information effectively.
- In a business context, networking ensures seamless access to shared files, databases, applications, and internet resources, enabling centralized management of systems and devices.

4.2 Types of Networks

Different types of networks exist depending on the geographic area they cover and the purpose they serve:

- **Local Area Network (LAN)**:
    - A **LAN** connects devices within a small geographic area, such as a home, office, or campus. LANs typically use Ethernet cables or Wi-Fi to enable devices like computers, printers, and servers to communicate and share data. LANs are widely used in businesses, schools, and homes for local file sharing, internet access, and peripheral device sharing.
    - LANs offer high-speed data transfer rates within limited distances, making them ideal for environments that need fast communication over short ranges.
- **Wide Area Network (WAN)**:
    - A **WAN** covers a much larger geographic area and connects multiple **LANs**. A WAN can span cities, countries, or even continents. The internet is the largest example of a WAN. WANs are commonly used by organizations with multiple branches to allow employees to access shared resources, applications, and data from different locations.
    - WANs typically use leased telecommunication lines, satellite links, or the internet for long-distance data transmission.
- **Metropolitan Area Network (MAN)**:
    - A **MAN** spans a city or large campus, larger than a **LAN** but smaller than a **WAN**. MANs are commonly used in large organizations or universities to connect multiple buildings within a city. They typically use high-speed connections like fiber-optic cables to provide fast data transfer across urban areas.
- **Wireless Networks**:
    - Wireless networks use radio waves to transmit data between devices, allowing users to connect to the network without physical cables. Common wireless network technologies include Wi-Fi (for local connectivity) and cellular networks (for wide area connectivity). Wireless networks provide flexibility and mobility, allowing users to access network resources from different locations.
- **Virtual Private Network (VPN)**:
    - A **VPN** is a secure connection that allows users to access a private network remotely over a public network, such as the internet. VPNs encrypt data to

ensure privacy and protect sensitive information from being intercepted by unauthorized users.

- **VPNs** are commonly used by remote workers and businesses to securely connect to internal networks and access resources from outside the office environment.

4.3 Network Devices

Network devices facilitate the transmission of data between connected devices and systems:

- **Router**:
  - A **router** is a networking device that forwards data packets between networks. It determines the most efficient route for data to travel across the network and connects different networks, such as **LANs** to **WANs**. Routers also handle IP addressing and manage the flow of data across networks.
  - In home or business environments, routers connect local devices to the internet and help distribute data among connected devices.
- **Switch**:
  - A **switch** connects devices within a **LAN** and forwards data between them. Unlike hubs, which broadcast data to all devices, switches intelligently direct data only to the intended recipient device based on their **MAC addresses**.
  - Switches play a critical role in managing traffic efficiently within a network, reducing the chance of data collisions and improving overall network performance.

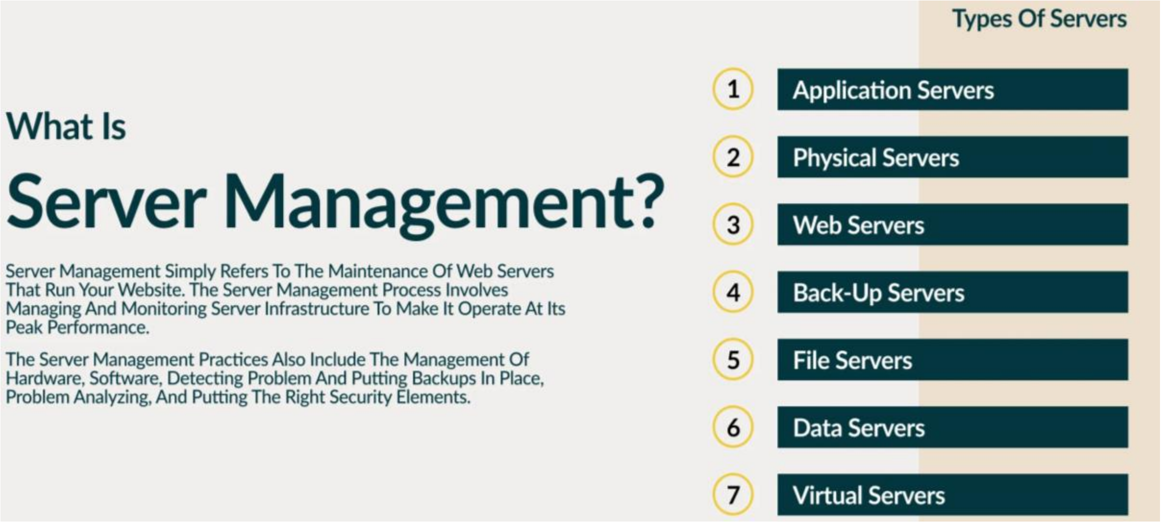
- **Firewall**:
    - A **firewall** is a security device that monitors and controls incoming and outgoing network traffic based on predefined security rules. Firewalls can be hardware- or software-based and are designed to protect internal networks from unauthorized access, malware, and other cyber threats.
    - Firewalls are used to block or allow data traffic based on security policies and can be configured to filter traffic at the perimeter of the network or internally to secure sensitive data.
- **Modem**:
    - A **modem** (modulator-demodulator) converts digital data from a computer into analog signals for transmission over phone lines and vice versa. **Modems** are used in **DSL** or cable internet connections to provide access to the internet.
    - Modems are commonly used in home and business environments to connect to an Internet Service Provider (ISP) and establish a connection to the internet.
- **Network Interface Card (NIC)**:
    - A **NIC** is a hardware component that allows a computer or device to connect to a network. **NICs** can be wired (Ethernet) or wireless (Wi-Fi) and are responsible for sending and receiving data over the network.
    - Most modern computers and laptops come with built-in **NICs**, allowing them to connect to local and wireless networks for internet access and file sharing.

4.4 Networking Protocols

Networking protocols define rules and standards for communication between devices on a network:

- **Transmission Control Protocol/Internet Protocol (TCP/IP)**:
    - **TCP/IP** is the core suite of protocols that enables communication on the internet and most other networks. **TCP** ensures reliable, ordered, and error-checked data transmission, while **IP** handles the addressing and routing of data packets to their destination.
    - Together, **TCP/IP** is used for transmitting data across networks and forms the foundation of the internet and many private networks.
- **Hypertext Transfer Protocol (HTTP/HTTPS)**:
    - **HTTP** is the protocol used for transferring web pages from web servers to browsers. **HTTPS** is the secure version of **HTTP**, which encrypts the data to ensure security and privacy during transmission.
    - **HTTPS** is commonly used on websites that handle sensitive information, such as online banking, e-commerce, and email services.
- **File Transfer Protocol (FTP)/Secure File Transfer Protocol (SFTP)**:
    - **FTP** is a standard network protocol used for transferring files between a client and a server over a **TCP/IP** network. **SFTP** is a secure version of **FTP** that uses **Secure Shell (SSH)** to provide encrypted, secure file transfer.
    - While **FTP** is suitable for basic file transfer needs, **SFTP** is preferred for transmitting sensitive data, as it provides encryption and authentication to ensure secure connections.
- **Domain Name System (DNS)**:
    - The **DNS** is a hierarchical and decentralized naming system used to translate human-readable domain names (like www.example.com) into IP addresses (like 192.0.2.1), which computers use to identify each other on the network.
    - **DNS** is essential for navigating the internet, as it allows users to access websites using easy-to-remember domain names instead of complex numeric IP addresses.

CHAPTER 5: SERVER MANAGEMENT



What Is

Server Management?

Server Management Simply Refers To The Maintenance Of Web Servers That Run Your Website. The Server Management Process Involves Managing And Monitoring Server Infrastructure To Make It Operate At Its Peak Performance.

The Server Management Practices Also Include The Management Of Hardware, Software, Detecting Problem And Putting Backups In Place, Problem Analyzing, And Putting The Right Security Elements.

Types Of Servers

1. Application Servers
2. Physical Servers
3. Web Servers
4. Back-Up Servers
5. File Servers
6. Data Servers
7. Virtual Servers

5.1 Server Configuration and Setup

- **Server configuration and setup** refer to the process of preparing physical or virtual servers to meet the specific requirements of an organization's applications, data, and users. Proper configuration ensures that servers perform optimally, are secure, and are able to handle the workloads assigned to them.
    - **Physical Servers**:
        - For on-premise setups, physical servers need to be assembled, installed, and configured according to hardware specifications. Tasks include configuring **CPU**, **RAM**, **storage**, and **network interfaces** to ensure they can handle the organization’s needs.
        - Physical servers also require the installation of an operating system (OS), typically **Linux** (such as **Ubuntu** or **CentOS**) or **Windows Server**, depending on the use case. Once the OS is installed, administrators configure security settings, network connectivity, and storage options.
    - **Virtual Servers**:
        - Virtual servers are created using **virtualization platforms** such as **VMware**, **Hyper-V**, or **KVM**. These platforms allow multiple virtual machines (VMs) to run on a single physical server, optimizing hardware utilization.
        - For virtual environments, administrators configure virtual CPUs, virtual memory, and virtual disk space to ensure each virtual server can handle its workload. They also assign appropriate network interfaces and install the required OS.

        - Virtual servers offer greater flexibility, as they can be easily created, cloned, and moved between physical hosts as needed, providing scalability and fault tolerance.

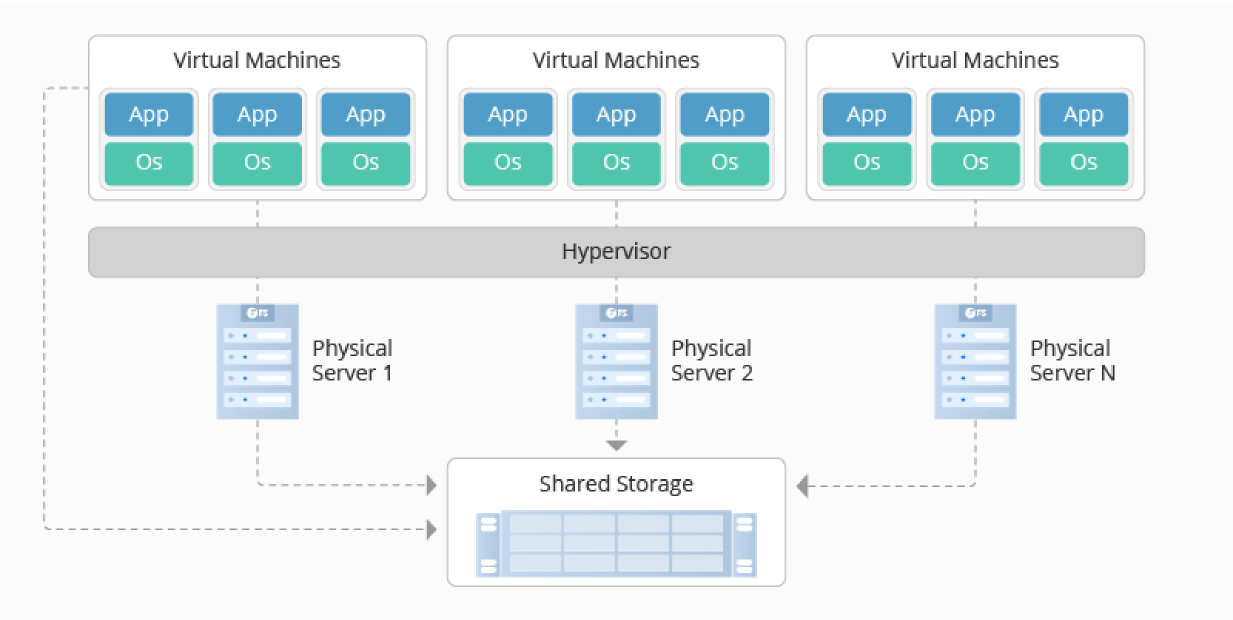
    - **Optimizing Server Performance**:

        - After the initial setup, servers must be fine-tuned for performance. This includes adjusting **I/O (Input/Output)** settings, **CPU affinity**, and **memory allocation** to ensure that applications run efficiently.

        - Server administrators also implement **load balancing** to distribute traffic and tasks across multiple servers, preventing any single server from becoming overwhelmed.

5.2 Virtualization and Cloud Management

- **Virtualization** involves creating virtual instances of computing resources, including servers, storage, and networks. **Cloud management** refers to overseeing and managing cloud computing services, such as virtualized infrastructure in public, private, or hybrid cloud environments. The most commonly used cloud platforms include **Amazon Web Services (AWS)**, **Microsoft Azure**, and **Google Cloud Platform (GCP)**.



    - **Managing Virtualized Environments**:

        - Virtualized environments enable organizations to maximize the use of physical resources by running multiple VMs on a single host. Virtualization technologies such as **VMware ESXi**, **Microsoft Hyper-V**, and **OpenStack** help manage virtual environments, providing tools for provisioning, monitoring, and scaling resources as needed.

    - Server administrators in virtualized environments are responsible for resource allocation, ensuring that VMs have sufficient CPU, memory, and storage. They also manage the underlying hypervisor, which handles the interaction between VMs and physical hardware.

  - **Cloud Infrastructure Management**:

    - **Cloud infrastructure** is managed through platforms like **AWS**, **Azure**, and **GCP**, which offer **Infrastructure as a Service (IaaS)**, allowing organizations to rent virtualized computing resources from cloud providers. Cloud infrastructure management involves tasks such as creating and scaling virtual machines (VMs), managing virtual networks, and configuring storage solutions.

    - **Microsoft Azure** provides similar services, including **Azure Virtual Machines**, **Azure Blob Storage**, and **Azure SQL Database**. **GCP** offers **Compute Engine**, **Cloud Storage**, and **Cloud SQL** for similar purposes.

  - **Key Virtualization and Cloud Management Tasks**:

    - **Provisioning**: Administrators allocate and configure VMs or cloud resources based on business requirements.

    - **Scaling**: Automatically or manually scaling resources up or down based on demand to optimize costs and performance.

    - **Backup and Recovery**: Managing backup solutions and ensuring disaster recovery plans are in place to recover data and services in case of failures or outages.

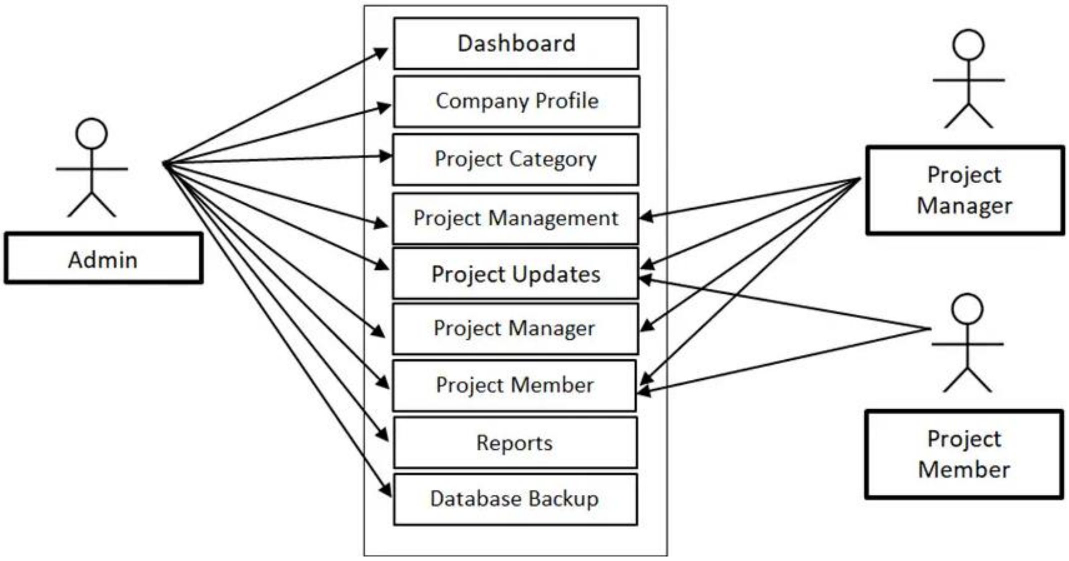
  - **Cost Management**:

    - Managing cloud costs involves monitoring resource usage to avoid over-provisioning, taking advantage of reserved instances or spot pricing, and optimizing cloud storage and bandwidth usage.

CHAPTER 6: BEST PRACTICES

6.1 System Administration Best Practices

- **System administration best practices** are essential for maintaining the stability, security, and efficiency of IT systems within an organization. These practices help ensure that systems run smoothly, downtime is minimized, and the risk of security breaches is reduced. Key best practices include:



  - **Regular Backups**:
    - **Data backups** are crucial to prevent data loss due to hardware failure, software corruption, or cyberattacks. Administrators should set up automated, regular backups of critical systems and data. These backups should be stored both **on-site** and **off-site** (e.g., in the cloud) to ensure availability during disasters.
  - **Security Monitoring and Updates**:
    - System administrators must continuously monitor systems for potential security threats. **Security monitoring tools** like **Nagios**, **Zabbix**, or **SolarWinds** help track server health, monitor for suspicious activity, and provide alerts in real-time.
  - **Task Automation**:
    - Automating repetitive tasks improves efficiency and reduces the chance of human error. System administrators can automate common tasks such as software updates, server provisioning, and configuration management using tools like **Ansible**, **Puppet**, or **Chef**.
  - **Enforcing Security Policies**:
    - Strong **security policies** should be enforced across all systems to protect against unauthorized access and data breaches. **Role-Based**

        Access Control (RBAC) should be implemented to ensure that users have only the permissions they need to perform their roles.
        - **Multi-Factor Authentication (MFA)** adds an extra layer of security, making it harder for attackers to gain unauthorized access to sensitive systems. Administrators should also enforce policies such as **password complexity**, **automatic account lockouts**, and **session timeouts**.
    - **System Monitoring and Incident Response**:
        - **System monitoring** ensures that administrators can detect and respond to potential issues before they escalate into major problems. Tools like **Nagios**, **Zabbix**, and **SolarWinds** allow for real-time monitoring of server performance, disk space, network traffic, and security incidents.
    - **Documentation**:
        - Proper documentation of system configurations, policies, and procedures is critical for effective system management. **Documentation** allows administrators to troubleshoot issues quickly, train new staff, and ensure consistency across the organization.