Practical 8 : Case study of Server OS: Windows Server 2022 operating System - Architecture, Components, Services, Configuration

(**Note** : Students should use appropriate images while presenting case study)

Windows Server 2022 is a server operating system developed by Microsoft. Let's explore its architecture, components, services, and configuration.

Architecture :

Windows Server 2022 follows a similar architecture to other Windows Server versions. It is based on the Windows NT kernel, which provides core functionality and services. It supports both 32-bit and 64-bit architectures.

Components :

1. **Kernel** : The Windows NT kernel forms the core of the operating system, managing processes, memory, and system resources.
2. **User Mode** : This layer provides the user interface and runs user applications, including graphical interfaces and command-line tools.
3. **Hardware Abstraction Layer (HAL)** : HAL abstracts hardware-specific details, allowing the operating system to work with different hardware configurations.
4. **File System** : Windows Server 2022 supports multiple file systems, including NTFS (New Technology File System), ReFS (Resilient File System), and FAT32 (File Allocation Table).

Services :

Windows Server 2022 offers various services to support server functionality. Some important services include:

1. **Active Directory Domain Services (AD DS):** Provides centralized user and computer management, authentication, and security policies.
2. **Domain Name System (DNS)**: Translates domain names into IP addresses, enabling network communication.
3. **Dynamic Host Configuration Protocol (DHCP)**: Assigns IP addresses and network configuration settings to devices on a network.
4. **File and Storage Services**: Offers file sharing, storage management, and data deduplication features.
5. **Hyper-V**: Provides virtualization capabilities, allowing multiple virtual machines to run on a single physical server.
6. **Internet Information Services (IIS)**: Supports hosting websites and web applications.
7. **Remote Desktop Services (RDS)**: Enables remote access to Windows-based applications and desktops.

Configuration : Windows Server 2022 can be configured using various tools, including:

1. **Server Manager** : Provides a graphical interface to manage server roles, features, and configuration settings.
2. **PowerShell**: A powerful command-line interface for system administration and automation.
3. **Group Policy**: Allows centralized management of security policies, user settings, and system configurations across multiple servers.
4. **Windows Admin Center**: A web-based management tool that simplifies server administration tasks.

These are just some of the key aspects of the Windows Server 2022 operating system. It offers a wide range of features and capabilities to support enterprise-level server deployments.

Practical 9. Case study of Android OS: Architecture, Components, Services, Configuration

(Note : Students should use appropriate images while presenting case study)

Android OS is a popular mobile operating system developed by Google. Let's delve into its architecture, components, services, and configuration.

Architecture :

Android follows a layered architecture that consists of the following components:

1. **Linux Kernel :** Android is built on top of the Linux kernel, which provides core operating system services such as process management, memory management, device drivers, and security.
2. **Libraries :** Android includes a set of libraries written in C/C++ that provide essential functionality to applications, such as graphics rendering, database access, and networking.
3. **Android Runtime (ART) :** It is the runtime environment in which Android applications run. ART utilizes ahead-of-time (AOT) and just-in-time (JIT) compilation techniques to optimize application performance.
4. **Application Framework :** The framework provides a set of APIs and tools for developers to build applications. It includes components such as activities, services, content providers, and broadcast receivers.
5. **Applications :** These are the end-user applications that run on the Android platform, such as web browsers, email clients, and social media apps.

Components :

1. **Activities:** Activities represent the user interface of an application and manage user interactions.
2. **Services:** Services perform background operations without a user interface, such as playing music or handling network requests.
3. **Broadcast Receivers:** They listen for system-wide or application-specific events and respond accordingly, enabling communication between different components.
4. **Content Providers:** Content providers manage shared data, allowing applications to access and share data across different apps securely.

Services :

Android OS provides various services that enhance the functionality of the platform, including:

1. **Activity Manager :** Manages the lifecycle of activities and their interactions.
2. **Package Manager :** Handles the installation, removal, and management of applications.
3. **Notification Manager :** Manages the display of notifications to the user.
4. **Location Manager :** Provides access to location-based services such as GPS and network location.
5. **Telephony Manager :** Offers access to telephony-related services such as making calls and sending messages.

Configuration :

Android OS can be configured using several methods, including:

1. **Settings App :** Android provides a settings app where users can configure various aspects of their device, such as network settings, display settings, and app permissions.
2. **Developer Options:** Advanced configuration options are available for developers, allowing them to enable debugging, control animation settings, and more.
3. **XML Configuration Files:** Developers can use XML files to configure various aspects of an application, such as layout design, resource values, and permissions.
4. Android offers a flexible and customizable platform for mobile devices, enabling a wide range of applications and services. Its architecture, components, services, and configuration options provide developers and users with a rich and adaptable experience.

(Note : Students should use appropriate images while presenting case study)

10 a) Case study of Cloud OS: AWS

AWS (Amazon Web Services) is a widely used cloud computing platform provided by Amazon. Let's explore the case study of AWS as a Cloud OS, including its architecture, components, services, and configuration.

Architecture :

AWS operates on a distributed architecture that spans multiple data centers worldwide. It is designed for high availability, scalability, and fault tolerance. The underlying infrastructure consists of a vast network of servers, storage systems, and networking components.

Components :

1. **Regions**: AWS is divided into multiple regions, each comprising several availability zones. Regions are geographically distinct and provide isolated infrastructure.
2. **Availability Zones**: Availability zones are data centers within a region that are physically separate from each other. They provide redundancy and fault tolerance.
3. **Virtual Private Cloud (VPC):** VPC enables users to create a private network within AWS. It allows customization of IP address ranges, subnets, and network gateways.
4. **Elastic Compute Cloud (EC2):** EC2 provides scalable virtual servers, known as instances, that can be configured with various operating systems and applications.
5. **Simple Storage Service (S3):** S3 offers object storage for storing and retrieving data. It provides durability, high availability, and scalability.
6. **Elastic Block Store (EBS)**: EBS provides persistent block-level storage volumes that can be attached to EC2 instances. It enables data persistence for applications.
7. **Identity and Access Management (IAM)**: IAM provides centralized access control and identity management for AWS services. It allows the creation of users, groups, and roles with specific permissions.
8. **Load Balancers :** AWS offers various load balancing services, such as Elastic Load Balancer (ELB), to distribute traffic across multiple instances for improved scalability and availability.

Services :

AWS provides a vast array of services catering to various computing needs, including:

1. **Compute Services**: EC2 for virtual servers, Elastic Container Service (ECS) for container management, Lambda for serverless computing.
2. **Storage Services**: S3 for object storage, EBS for block storage, Elastic File System (EFS) for shared file storage.
3. **Database Services**: Relational Database Service (RDS) for managed databases, DynamoDB for NoSQL databases, Aurora for high-performance databases.
4. **Networking Services**: Virtual Private Cloud (VPC), Direct Connect for dedicated network connection, Route 53 for DNS management.

5. **Analytics and Big Data**: Amazon Redshift for data warehousing, EMR for big data processing, Athena for interactive querying.
6. **AI and Machine Learning**: Amazon SageMaker for building and deploying ML models, Rekognition for image and video analysis, Polly for text-to-speech conversion.
7. **Security and Compliance**: AWS Identity and Access Management (IAM), AWS Key Management Service (KMS), AWS Shield for DDoS protection.

Configuration :

AWS services can be configured using various tools and APIs, such as:

1. **AWS Management Console**: A web-based interface for managing AWS resources, configuring services, and monitoring usage.
2. **AWS Command Line Interface (CLI)**: A command-line tool for managing AWS resources, automating tasks, and scripting.
3. **AWS CloudFormation**: A service for infrastructure as code, allowing the provisioning and management of AWS resources using templates.

AWS offers extensive documentation, guides, and support to assist users in configuring and utilizing its services effectively. Its comprehensive set of components, services, and configuration options make it a versatile and powerful Cloud OS for organizations of all sizes.

10 b) Case study of Cloud OS : Azure

Azure is a widely used cloud computing platform provided by Microsoft. Let's explore the case study of Azure as a Cloud OS, including its architecture, components, services, and configuration.

Architecture

Azure operates on a distributed architecture that spans multiple data centers worldwide. It is designed to provide high availability, scalability, and reliability. The underlying infrastructure consists of a global network of servers, storage systems, and networking components.

Components :

1. **Azure Regions**: Azure is divided into multiple regions globally. Each region consists of multiple data centers strategically located in different geographical locations to ensure data redundancy and availability.
2. **Availability Zones**: Availability zones are physically separate data centers within an Azure region. They provide redundancy and fault tolerance by hosting applications and data in different zones.
3. **Virtual Networks**: Azure Virtual Networks (VNet) enable users to create isolated networks in the cloud. VNets allow customization of IP address ranges, subnets, and network gateways.
4. **Virtual Machines**: Azure Virtual Machines (VMs) provide scalable and flexible compute resources. Users can create VMs with different operating systems and configurations to run their applications.

5. **Storage** : Azure offers various storage services, including Azure Blob Storage for object storage, Azure File Storage for shared file storage, and Azure Disk Storage for persistent block storage.
6. **Identity and Access Management**: Azure Active Directory (Azure AD) provides centralized identity management and access control for Azure services. It enables user authentication, single sign-on, and role-based access control.
7. **Load Balancers**: Azure Load Balancer distributes incoming network traffic across multiple VMs or instances for improved scalability and availability.
8. **Azure App Service**: Azure App Service is a fully managed platform for building, deploying, and scaling web and mobile applications. It supports multiple programming languages and frameworks.

Services

Azure offers a comprehensive range of services to meet various cloud computing needs, including:

1. **Compute Services**: Azure Virtual Machines (VMs), Azure Kubernetes Service (AKS) for container orchestration, Azure Functions for serverless computing.
2. **Networking Services**: Azure Virtual Network (VNet), Azure Load Balancer, Azure Application Gateway for load balancing and traffic management, Azure VPN Gateway for secure connectivity.
3. **Database Services**: Azure SQL Database for managed relational databases, Azure Cosmos DB for globally distributed NoSQL databases, Azure Database for MySQL and PostgreSQL.
4. **Storage Services**: Azure Blob Storage, Azure File Storage, Azure Queue Storage, Azure Table Storage for different storage requirements.
5. **Analytics and Big Data**: Azure Synapse Analytics for data warehousing and analytics, Azure HDInsight for big data processing, Azure Data Lake Storage for scalable data storage.
6. **AI and Machine Learning**: Azure Machine Learning for building and deploying ML models, Azure Cognitive Services for pre-built AI capabilities like vision, speech, and language recognition.
7. **Security and Compliance**: Azure Security Center for threat protection and security management, Azure Key Vault for managing cryptographic keys and secrets, Azure Active Directory for identity and access management.

Configuration

Azure services can be configured using various tools and interfaces, such as:

1. **Azure Portal**: A web-based interface for managing Azure resources, configuring services, and monitoring usage.
2. **Azure Command-Line Interface (CLI)**: A command-line tool for managing Azure resources, automating tasks, and scripting.
3. **Azure Resource Manager (ARM) Templates**: ARM templates enable infrastructure as code, allowing the provisioning and management of Azure resources through declarative templates.

Azure provides extensive documentation, tutorials, and support to assist users in configuring and leveraging its services effectively. Its robust architecture, wide range of components and services, and flexible configuration options make it a powerful Cloud OS for organizations across different industries.

10 c) Case study of Cloud OS: Google Cloud

Google Cloud is a popular cloud computing platform provided by Google. Let's explore the case study of Google Cloud as a Cloud OS, including its architecture, components, services, and configuration.

Architecture

Google Cloud operates on a distributed architecture that spans multiple data centers globally. It is designed for high scalability, availability, and reliability. The underlying infrastructure consists of a global network of servers, storage systems, and networking components.

Components :

1. **Regions** : Google Cloud is divided into multiple regions worldwide, each comprising several availability zones. Regions are geographically distinct and provide isolated infrastructure.
2. **Availability Zones** : Availability zones are data centers within a region that are physically separate from each other. They provide redundancy and fault tolerance.
3. **Virtual Private Cloud (VPC) :** VPC enables users to create isolated virtual networks in Google Cloud. It allows customization of IP address ranges, subnets, and network gateways.
4. **Compute Engine** : Compute Engine provides virtual machine instances in Google Cloud, allowing users to run applications on scalable and flexible virtual servers.
5. **Cloud Storage** : Google Cloud Storage offers object storage for storing and retrieving data. It provides durability, scalability, and high availability.
6. **Identity and Access Management (IAM) :** IAM provides centralized access control and identity management for Google Cloud services. It allows the creation of users, groups, and roles with specific permissions.
7. **Load Balancing :** Google Cloud offers load balancing services, including HTTP(S) Load Balancing and Network Load Balancing, to distribute traffic across multiple instances for improved scalability and availability.
8. **Kubernetes Engine**: Kubernetes Engine provides a managed environment for deploying, managing, and scaling containerized applications using Google Kubernetes Engine (GKE).

Services :

Google Cloud provides a wide range of services catering to various cloud computing needs, including:

1. **Compute Services**: Compute Engine for virtual machines, App Engine for scalable web applications, Cloud Functions for serverless computing.
2. **Networking Services**: Virtual Private Cloud (VPC), Cloud Load Balancing, Cloud DNS for domain name management, Cloud VPN for secure connectivity.