Slide 1: Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**IaaS** offers basic computer infrastructure; it allows for the rental of physical or virtual machines, such as servers, network equipment, and storage units, through the cloud. With this model, users have full control over the infrastructure, allowing them to install and manage operating systems, applications, and other software. Microsoft Azure Virtual Machines , Google Cloud Platform (GCP) Compute Engine. **PaaS** provides software developers with a platform and environment to develop and deploy their applications. In this model, infrastructure elements like the operating system, server, storage, and networking are managed by the service provider. Developers gain access to the tools and services necessary to build and run their applications. Big query , **Vertex AI**. **SaaS** offers users access to software over the internet. In this service model, both the software and the underlying infrastructure are fully managed by the cloud provider. Users can access the application through a web browser, often paying on a subscription basis. Microsoft Office 365

**Opportunities and challenges:**

Scalability , Flexibility : Sometimes limited control over infrastructure, Physical Area, Storage area for compute and network resources , Strict A/C Maintenance , Administrative , Software Updates , Component failures , Data reliability (redundancy) , Security Cost model , “pay as you go” vs “on going cost” , Can be complicated , Vendor lock-in problem

Public Cloud is a cloud computing model where computing resources (like servers, storage, applications, and more) are made available over the internet for public use. In this model, service providers offer the infrastructure and services to a broad user base. Users access these resources via the internet and typically pay based on their usage.

**Shared Resources:** Many customers share the same physical infrastructure, though each operates in a separate and secure virtual space.**Scalability:** Resources can be easily scaled up or down according to needs, allowing for quick response to demand fluctuations.**Cost-Effectiveness:** No capital expenditure is required. Users pay only for what they use.**Accessibility:** Users can access services from anywhere with an internet connection.

**Maintenance and Management:** The infrastructure is managed and maintained by the service provider, relieving users from IT management concerns

Private Cloud refers to a cloud computing model where the infrastructure and services are exclusively operated for a single organization. Unlike public clouds, which serve a broad range of customers, private clouds are dedicated to the needs and requirements of a single entity, providing a higher level of control and privacy.**Exclusive Use:** The infrastructure and resources are dedicated to one organization, ensuring that resources are not shared with others.**Control and Customization:** Organizations have greater control over the cloud environment and can customize it according to their specific needs.**Security and Privacy:** Because the infrastructure is not shared with other organizations, private clouds offer enhanced security and privacy, making them suitable for handling sensitive data and meeting regulatory compliance requirements.**Scalability:** While not as flexible as public clouds in terms of scalability, private clouds still offer the ability to scale resources up or down as needed, but within the confines of the organization’s own infrastructure.**Cost and Maintenance:** Private clouds can be more expensive than public clouds due to the cost of dedicated hardware, software, and maintenance. Organizations need to invest in their own infrastructure and manage the cloud environment.o**n-Premises:** The private cloud infrastructure is physically located within the organization’s own data center.

Hybrid Cloud is a cloud computing model that combines public and private cloud environments, allowing data and applications to be shared between them. This model enables businesses to benefit from the scalability and cost-efficiency of public clouds while maintaining the security and control of a private cloud.

**\*\*\*\*Now vs. Siri in terms of Computation Load:** Google Assistant heavily relies on cloud-based processing. When a query is made, the voice data is sent to Google's servers, where it's processed using Google's powerful data centers. This approach leverages Google's advanced AI and machine learning models, requiring significant server-side computation.Siri uses a combination of on-device and cloud-based processing. Initially, Siri processes the voice data on the device to understand the intent of the query. Depending on the complexity of the request, it might handle the task entirely on-device or send data to Apple's servers for further processingIn summary, Google Assistant typically has a higher computational load on servers due to its extensive cloud processing, while Siri may place more load on the device itself to maintain privacy and reduce cloud dependency.Amazon AppStream 2.0 is a fully managed application streaming service provided by Amazon Web Services (AWS). It allows users to stream desktop applications from AWS to any device with a web browser, without needing to rewrite them for the cloud. This service enables users to access the applications they need from anywhere, while the actual computing and storage happen on AWS servers.

**Containers Vs VMs:** Containers are often used as virtual environments to isolate and run applications in a lightweight, efficient, and portable manner. Unlike traditional virtual machines (VMs) that virtualize entire hardware stacks, containers virtualize the operating system, allowing multiple workloads to run on a single OS instance.

Containers are thinner and lighter weight than traditional VMs. A portable & rapid way to push application workloads from development, production • No dependencies on hardware or OS. Run several workloads on same platform: VM or bare metal. How it work: **Container Engine:** A container engine (like Docker) is used to run containers. It packages applications and dependencies into container images, which are lightweight, standalone, executable packages. **Images and Registries:** Container images are stored in registries (like Docker Hub) and can be pulled and run on any system that has a container engine installed. **Runtime:** When an image is run, the container engine creates a container instance from the image, which operates in isolation from other containers and the host system, though it may share some resources like libraries and the kernel. **Hardware-Level Virtualization** Also known as full virtualization, hardware-level virtualization involves creating a complete virtual machine (VM) that includes a full copy of an operating system along with the necessary hardware resources abstracted by a hypervisor.**Hypervisor:** A software layer called a hypervisor (or Virtual Machine Monitor - VMM) sits between the hardware and the VMs, managing the distribution of hardware resources to each VM. **Isolation:** Each VM is fully isolated from others, running its own OS, applications, and services as if it were on its own physical hardware.**Resource Overhead:** Because each VM includes a full OS instance, this approach can consume more resources compared to OS-level virtualization.

**OS-Level Virtualization**In OS-level virtualization, the virtualization layer allows for multiple isolated user-space instances, often referred to as containers or zones, within a single OS kernel. **Characteristics: Shared Kernel:** All containers share the same kernel but operate in isolated user spaces, making them lighter and faster than full VMs.**Efficiency:** Containers start up and shut down rapidly and use less memory and CPU resources than VMs because they don’t need to load a separate OS.**Application-centric:** OS-level virtualization is often used to deploy and manage applications rather than entire virtualized systems. **namespace: linux kernel feature that isolates and virtualizes system resources** Cgroups:control groups collect set of process tasks IDS together and apply limits, such as for resource utilization Layers file system: optimal way to make a copy of root filesystem for each container **\*\*\*\***Virtualization types**:Binary translation** is a technique used in computing to convert executable code from one binary format to another. This is often done to achieve compatibility between different operating systems or hardware architectures.**Para-virtualization** is a virtualization technique where the guest operating systems are modified to run on a virtual machine monitor (VMM) or hypervisor, which is aware of the presence of other virtual machines. This method requires the guest OS to be explicitly adapted or modified for the virtualized environment, which allows for more efficient system calls between the VM and the hypervisor.**Hardware-assisted virtualization** is a virtualization technique that relies on specific hardware capabilities provided by modern CPUs to enhance the performance and efficiency of virtual machines (VMs). This technology allows the hypervisor to delegate certain tasks to the hardware, minimizing the overhead traditionally associated with software-based virtualization.\*\*\***An Availability Zone (AZ)** is a concept used by cloud service providers to increase the reliability and availability of their services. Each Availability Zone is a distinct location within a region that is engineered to be isolated from failures in other Availability Zones. These zones are connected through low-latency links, enabling them to support high-availability configurations and resilient applications. An **SLA (Service Level Agreement**) is a formal contract between a service provider and a customer that defines the level of service expected from the provider. SLAs are critical components of any vendor contract for services, particularly in technology and telecommunications sectors. \*\*\*\*\*\* **Elasticity** in cloud computing refers to the ability of a cloud system to dynamically adjust and allocate computing resources as needed to maintain consistent performance levels. This feature is a cornerstone of cloud computing, enabling efficient resource management and cost-effectiveness.

**Scalability:Horizontal Scaling (Scaling Out/In):** Involves adding or removing resource instances (such as virtual machines) to match demand without interrupting services.**Vertical Scaling (Scaling Up/Down):** Entails adding or subtracting resources (such as CPU or memory) to an existing instance to handle increased or decreased load. **Dynamic Resource Allocation, Cost-Effectiveness, Automated Management**

**Scalability** refers to the ability of a system to handle increased loads by proportionally adding resources. It is about the capacity to grow or shrink in resource size or volume, usually in response to longer-term changes in demand rather than immediate fluctuations.

Elasticity is more about immediate, automatic resizing of resources based on real-time demands. Scalability is concerned more with planned, strategic resizing based on anticipated growth patterns.

metin, yazı tipi, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, yazı tipi, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, çizgi, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

**\*\*\*Serverless Computing** is a cloud computing model that allows developers to build and run applications and services without having to manage the underlying infrastructure typically associated with development. In this model, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code.**Event-driven**: Most serverless platforms run code in response to events or triggers, which can include HTTP requests, database events, queuing services, or file uploads. With serverless, you pay only for the execution time of your functions, not for idle server time. **Cold Starts**: When a function is invoked after sitting idle, there can be a delay (cold start) while the cloud provider initializes a runtime container for the function, which can impact performance.**Function as a Service (FaaS)** is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app. FaaS is a key component of serverless architectures, where developers write individual "functions" that are triggered by events.**Serverlerss is not good for**: long running, stateful, number crunching: databases, Deep learning training, video streaming, But good for short run, stateless, event driven: microservices, IOT, Bot, mobile backends. \*\*\*\* **Virtual Machine (VM) migration**, the process of moving a VM from one physical machine or environment to another, is a crucial capability in data center management and cloud computing. As challenges: **Downtime** **Impact**: Migrating VMs can lead to downtime, affecting application availability and user experience. **Data Integrity and Consistency, Network Configurations, Cost Considerations, scale and complexity.**

**metin, diyagram, ekran görüntüsü, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, diyagram, plan, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, diyagram, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**VM migration** refers to the process of moving a virtual machine (VM) from one physical host to another, either within the same data center or across different data centers. VM migration is essential in virtualized environments for load balancing, hardware maintenance, and fault tolerance.**Cold Migration** involves shutting down the VM before moving it to another host. This is suitable for planned maintenance when downtime is acceptable. The process includes shutting down the VM, transferring its files (disk images, configuration files) to the destination host, and then powering on the VM at the new host. Requires more downtime since the VM must be powered off and then restarted. Simpler and safer, with no risk of data corruption during migration since the VM is fully powered off..**Warm Migration** suspends the VM and then moves it, making it suitable for situations where brief downtime is acceptable. The process involves suspending the VM, transferring its memory state and files to the destination host, and then resuming the VM at the new host. Requires less downtime. **Live Migration** allows the VM to continue running with minimal downtime during the migration process, making it ideal for load balancing, hardware maintenance without downtime, and failover scenarios. The process includes a pre-copy phase where the VM's memory pages are copied while it is still running, followed by an iterative pre-copy phase where changed pages are re-copied until the number of changed pages is small, and finally, a stop-and-copy phase where the VM is briefly paused to copy the remaining changed pages and then resumed at the destination host.**Key Components in VM Migration** include the hypervisor, which manages the VMs and handles the migration process; the network, which must be reliable to ensure minimal downtime and data integrity; and storage, with shared or network-attached storage (NAS) simplifying the migration process by making the VM's disk files accessible from both source and destination hosts.**Benefits of VM Migration** are numerous: load balancing by distributing workloads across multiple hosts to optimize resource usage and performance, allowing hardware or software maintenance without shutting down VMs to minimize service disruptions, enabling fault tolerance by moving VMs away from failing or underperforming hosts, and resource optimization by consolidating VMs onto fewer hosts during low usage periods to save power and operational costs.**Challenges and Considerations** include the need for sufficient network bandwidth to transfer large amounts of data, especially during live migration; ensuring compatibility between source and destination hosts in terms of CPU features, memory, and network configurations; maintaining data security during migration, particularly over untrusted networks; and managing the performance impact on VMs during migration, especially for live migration where the VM continues to run.**Practical Applications** of VM migration are seen in cloud computing for dynamic resource management, in data centers for efficient load balancing and maintenance without service interruptions, and in disaster recovery strategies to move VMs to safe locations during hardware failures or other disasters.

**\*\*\*\*\*DNS-based load balancing** operates at the DNS level, where the DNS server resolves a domain name to an IP address. When a client requests a specific hostname, the DNS server can return different IP addresses according to predefined rules or policies. This method can direct traffic based on various factors such as server load, geographic location, and server health.**Round Robin:** Simplest form of DNS load balancing, where each server's IP address is returned in a rotating sequential order. It does not account for server load or geographic proximity. **Simple**: asy to implement as it does not require complex network infrastructure changes.**Cost-Effective:** Utilizes existing DNS infrastructure without additional costs for hardware load balancers. **Caching Issues:** DNS responses may be cached by ISPs or client-side resolvers, which can lead to outdated routing information and uneven distribution of traffic.**Limited Health Checks. Lack of Session Persistence:HTTP redirection load balancing:** When a client (like a web browser) makes a request to a server, the server can respond with a specific HTTP status code indicating that the requested resource has been moved to a different URL. The response will also include a **Location** header that provides the new URL where the resource can be found. The client then automatically makes a new request to the URL specified in the **Location** header.HTTP redirection is a powerful tool for managing website navigation, server load, and user experience, but it must be used wisely to avoid negative impacts on website performance and user satisfaction.

**Layer 4 load balancers** operate at the transport layer (Layer 4) of the OSI model. They make routing decisions based on transport layer information such as IP addresses, TCP ports, or UDP ports. **Traffic Direction:** Uses IP address and port number to direct traffic to the correct server.**Speed and Efficiency:** Typically faster than Layer 7 load balancers because they don't inspect packet content beyond the network layer. C**onnection Management:** Can perform NAT (Network Address Translation) to modify the destination IP address and port on the packets as they pass through. **Common use cases:** Simple load balancing where deep packet inspection is not necessary. Situations requiring high performance and low latency, such as video streaming or VOIP services.**Layer 7 load balancers** operate at the application layer (Layer 7) of the OSI model. They can inspect, modify, and route traffic based upon the content of the packets, such as URL paths, message syntax, or headers. **Content-Based Routing:** Analyzes the content within application messages to make routing decisions. This can include HTTP headers, URI, SSL session ID, and more.**Sophisticated Traffic Management:** Can manage traffic based on complex rules derived from application data. For example, it can distribute requests to different servers based on the requested URL or the type of HTTP request.**Application-Specific Features:** Offers application-specific features such as cookie persistence, SSL termination, and advanced traffic management capabilities.**Layer 4** is suitable for simple, fast traffic routing needs where high throughput and low latency are critical.**Layer 7** is ideal for complex applications requiring detailed traffic management based on the content of the messages, particularly in highly dynamic web environments.

Mobile gadget paradox: smaller H/W: less computation resource, fancier applicatiıns: increasing s/w complexity

**\*\*\*\*Edge computing** refers to the practice of processing data near the edge of the network, where the data is generated, rather than in a centralized data-processing warehouse. The primary goal of edge computing is to reduce latency and bandwidth use by minimizing the distance that data must travel, thereby providing faster response times and more efficient processing.**Key Concepts in Edge Computing:Edge Devices**: These are devices at the periphery of the network that generate and process data. Examples include IoT devices, sensors, and local servers.**Edge Nodes**: These are intermediary devices that aggregate and process data from edge devices before forwarding it to the cloud or data center. Examples include local servers and gateways.**Fog Computing**: Often used interchangeably with edge computing, fog computing extends the cloud closer to the edge devices, providing additional processing power and storage capabilities.**Cloudlet**: A small-scale data center located at the edge of the network. Cloudlets provide resources for computing, storage, and networking, allowing for low-latency processing close to the data source. They act as an intermediary between the edge devices and the central cloud.**Standalone Edge Computing**: This refers to edge computing systems that operate independently without the need for continuous connectivity to a central cloud. These systems process data locally and can function autonomously, which is crucial for applications requiring real-time processing and low latency. **Latency**:**Edge Computing**: Processes data closer to the source, reducing latency significantly. Ideal for real-time applications such as autonomous vehicles, industrial automation, and augmented reality.**Cloud Computing**: Centralized processing can introduce latency due to the distance data must travel. Suitable for applications where real-time processing is not critical.**Bandwidth Usage**:**Edge Computing**: Reduces bandwidth usage by processing data locally and only sending necessary data to the cloud. This is beneficial for IoT applications generating large volumes of data.**Cloud Computing**: Requires significant bandwidth to transmit data to and from the central servers, which can be costly and inefficient for large datasets. **Reliability and Availability**:**Edge Computing**: Provides higher reliability and availability by processing data locally. This is crucial for applications in remote or disconnected environments where continuous cloud access is not guaranteed.**Cloud Computing**: Highly reliable and available due to redundancy and failover mechanisms. However, it depends on stable internet connectivity. **Scalability**:**Edge Computing**: Limited by the local resources available at the edge nodes and devices. Scaling requires adding more edge nodes or upgrading existing ones. **Cloud Computing**: Highly scalable, leveraging vast resources of centralized data centers. Scaling can be done quickly and efficiently with minimal local resource constraints. **Security**:**Edge Computing**: Offers enhanced security by keeping sensitive data local, reducing the risk of data breaches during transmission. However, it requires robust security measures at multiple edge nodes. **Cloud Computing**: Centralized security measures can be more comprehensive but pose a risk if a breach occurs, as more data is stored and transmitted .**Which parts can NOT run on edge?:** Local UI, I/O operations on local H/W components, Code parts that interact too much with code segments on local side. **MAUI(Multi-platform App UI),** AIM: Code offloading for battery lifetime extension. It is primarily known as a client-side framework for building cross-platform applications that run on Android, iOS, macOS, and Windows. MAUI applications can use HTTP clients to send requests to RESTful APIs. MAUI: making smartphones last longer with code offload. **\*\*\*\*Software-Defined Networking (SDN)** is a modern approach to network management that enables dynamic, programmatically efficient network configuration to improve network performance and monitoring.**Legacy Networking Systems** refer to traditional network architectures where the control plane and data plane are tightly integrated within network devices such as routers, switches, and firewalls.**SDN**: Centralized control through a software-based controller allows for dynamic and automated network management.**Legacy Systems**: Distributed control within individual devices requires manual configuration and management, leading to higher complexity. **SDN**: Highly flexible and scalable **Legacy Systems**: Less flexible and scalable. **SDN**: Quickly adapts to changing network conditions.**SDN**: Potentially reduces operational costs by simplifying network management. **\*\*\*\*CPU (Central Processing Unit)** is the primary component of a computer that performs most of the processing. It is designed to handle a wide range of tasks and is optimized for sequential serial processing. CPUs have a few powerful cores with high clock speeds and are well-suited for general-purpose computing tasks.**GPU (Graphics Processing Unit)** is a specialized processor designed to accelerate graphics rendering. GPUs have a large number of smaller cores optimized for parallel processing, making them ideal for tasks that can be divided into multiple parallel operations, such as graphics rendering, machine learning, and scientific computations.**TPU (Tensor Processing Unit)** is an application-specific integrated circuit (ASIC) developed by Google specifically for accelerating machine learning tasks, particularly those involving TensorFlow. TPUs are highly optimized for the types of matrix operations and floating-point arithmetic used in neural network training and inference, offering significant performance improvements over CPUs and GPUs for these tasks.**ComparisonCPU**: Versatile, general-purpose processing, few powerful cores, sequential tasks.**GPU**: Specialized for parallel processing, many smaller cores, excellent for graphics and parallel computations.**TPU**: Specialized for machine learning, optimized for TensorFlow, highly efficient for neural network tasks.\*\*\* **Why AIaaS?Accessibility**:**Simplified AI Integration**: AIaaS makes advanced AI technologies accessible to organizations of all sizes, enabling them to leverage AI without requiring extensive in-house expertise. **Ease of Use**: Developers can easily integrate AI functionalities into their applications using APIs and pre-built models, reducing the complexity of AI development. **Cost Efficiency**: **Reduced Costs**: AIaaS eliminates the need for significant upfront investments in AI infrastructure and tools. Businesses can pay for what they use, converting capital expenses into operational expenses.. **Scalability**: AIaaS platforms offer scalable solutions that can handle varying workloads, allowing businesses to scale their AI capabilities according to demand without worrying about infrastructure limitations.**Speed to Market**: **Faster Deployment**: With AIaaS, organizations can quickly deploy AI solutions without the lengthy process of building and training models from scratch. This accelerates time-to-market for AI-driven applications. **Continuous Improvement**: AIaaS providers often update their services with the latest advancements in AI, ensuring that users benefit from cutting-edge technologies without needing to maintain or upgrade their own systems.**Focus on Core Competencies**: **Resource Allocation**: By outsourcing AI capabilities, businesses can focus their resources and efforts on their core competencies and strategic initiatives, rather than diverting them to develop and maintain AI systems. **Access to Advanced Tools**: **State-of-the-Art AI**: AIaaS providers typically offer access to advanced AI tools and frameworks, enabling businesses to utilize sophisticated AI technologies that would be challenging to develop internally.

metin, ekran görüntüsü, çizgi, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu