डॉ0 शकुन्तला मिश्रा राष्ट्रीय पुनर्वास विश्वविद्यालय, लखनऊ
Dr Shakuntala Misra National Rehabilitation University, Lucknow

Unit 2nd

[TCS 074 - Cloud Computing]

Cloud Computing Services and Technologies

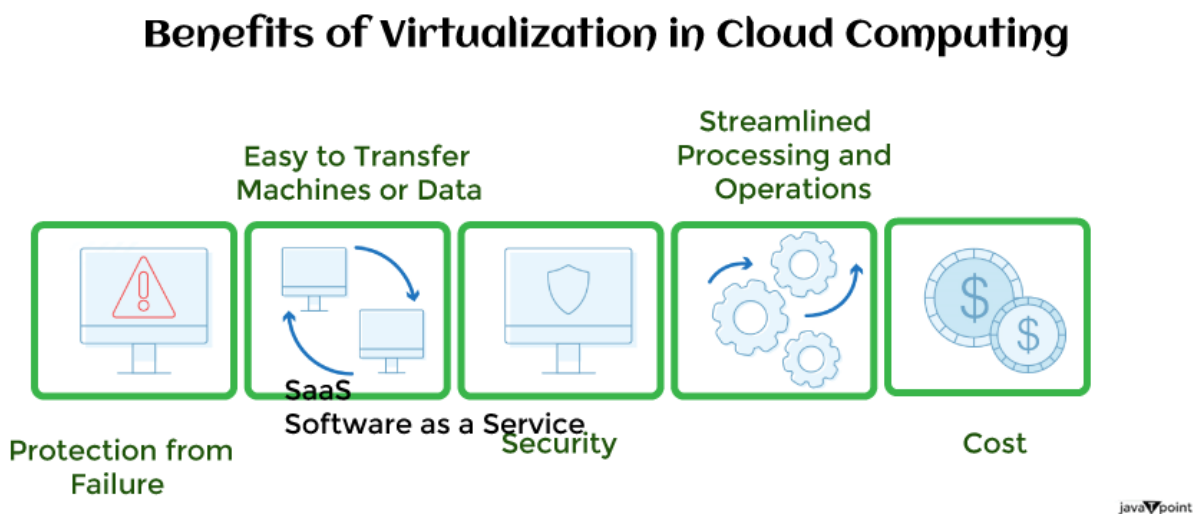
Virtualization

Virtualization is the process of creating a virtual environment to run multiple applications and operating systems on the same server. The virtual environment can be anything, such as a single instance or a combination of many operating systems, storage devices, network application servers, and other environments.
The concept of Virtualization in cloud computing increases the use of virtual machines. A virtual machine is a software computer or software program that not only works as a physical computer but can also function as a physical machine and perform tasks such as running applications or programs as per the user's demand.
Types of Virtualization
A list of types of Virtualization is given below -
Hardware Virtualization: Multiple virtual machines (VMs) can run on a single physical server thanks to hardware virtualization, which abstracts away physical hardware resources. This makes it possible to consolidate servers and use resources effectively.

Server Virtualization: A physical server is divided into several virtual servers, or VMs, each of which can run its own operating system and applications. This process is known as server virtualization. It increases server efficiency and streamlines administration.

Storage Virtualization: It creates a virtualized storage pool by abstracting physical storage resources. This improves scalability and flexibility, centralizes provisioning, and simplifies management.

Operating System Virtualization: This technology enables a host operating system to host a number of segregated user-space instances, often known as containers. With quicker startup times and less overhead, it offers virtualization that is both lightweight and effective.
Data Virtualization: It creates a cohesive virtual picture by abstracting data from diverse sources. Data integration is facilitated, and real-time data access is made possible by the ability of users and programs to access and alter data as if it were kept in a single location.
These virtualization techniques enhance IT environments' resource utilization, scalability, flexibility, and management effectiveness.



Service-Oriented Architecture (SOA)

Service-Oriented Architecture (SOA) allows organizations to access on-demand cloud-based computing solutions according to the change in business needs. It can work without or with cloud computing. The advantages of using SOA are that it is easy to maintain, platform-independent, and highly scalable.
Service Providers and Service consumers are the two major roles within SOA.
Applications of Service-Oriented Architecture:

There are the following applications of Service-Oriented Architecture –

Mobile Applications and Games: SOA offers a flexible and scalable architecture that enables smooth integration with back-end services and effective data management for the development of mobile applications and games.

Defense and the Air Force: The implementation of situational awareness systems for the air force makes use of SOA infrastructure, enabling real-time data integration from many sources, boosting decision-making abilities, and better mission planning and execution.

Healthcare: SOA is used in healthcare systems to provide secure patient information sharing, promote interoperability between various healthcare providers, and enhance the effectiveness of clinical procedures.

E-commerce: To enable scalable and adaptable e-commerce operations, SOA is utilized in online shopping platforms, payment gateways, and inventory management systems.

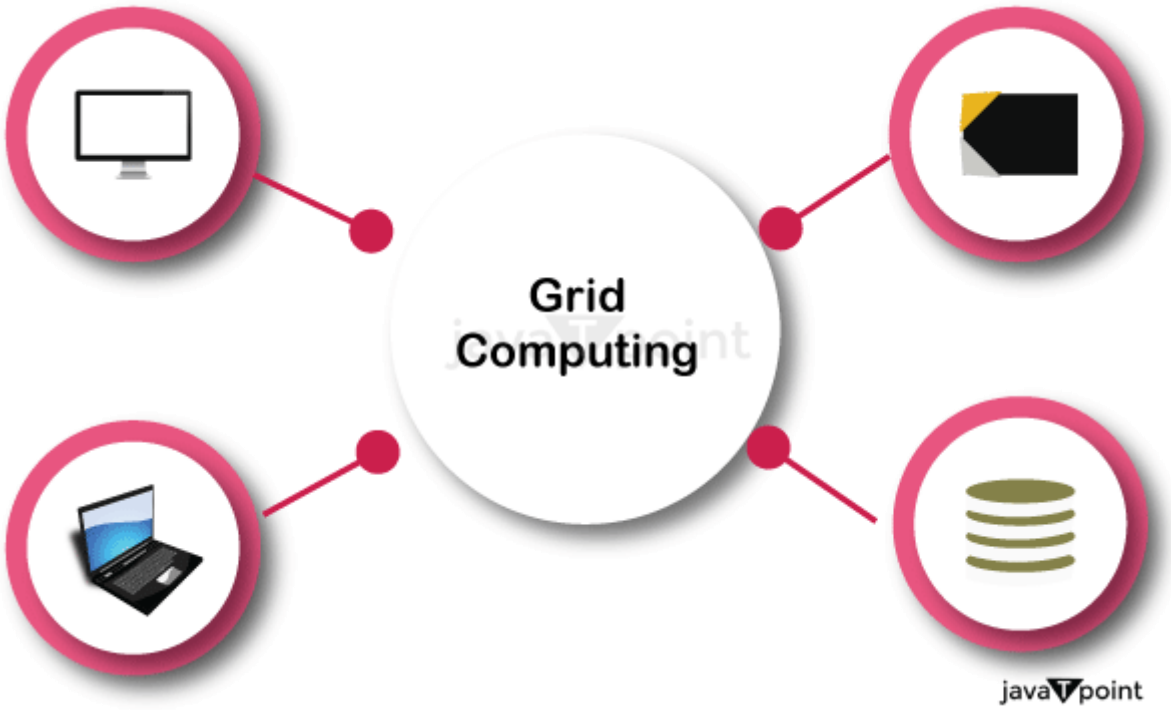
Financial Services: Integrating stock trading, banking, and insurance systems using SOA enables secure transactions and real-time data processing.
Government Systems: In government organizations, SOA promotes information sharing, cross-departmental cooperation, and citizen-centric services.
Supply Chain Management: SOA improves coordination and response by linking suppliers, manufacturers, distributors, and retailers. These streamlines supply chain processes.

Enterprise Resource Planning (ERP): SOA combines different corporate processes, including accounting, human resources, sales, and inventory control, in ERP systems.

Telecommunications: SOA combines phone, data, and video services across several networks to improve service delivery and client satisfaction.
Logistics and Transportation: SOA enhances efficiency and lowers costs in logistics and transportation systems by optimizing routing, monitoring shipments, and managing fleet operations.

Grid Computing

Grid computing is also known as distributed computing. It is a processor architecture that combines various different computing resources from multiple locations to achieve a common goal. In grid computing, the grid is connected by parallel nodes to form a computer cluster. These computer clusters are in different sizes and can run on any operating system.
Grid computing contains the following three types of machines -
Control Node: It is a group of servers that administrates the whole network.
Provider: It is a computer that contributes its resources to the network resource pool.
User: It is a computer that uses the resources on the network.
Mainly grid computing is used in ATMs, back-end infrastructures, and marketing research.



Utility Computing

Utility computing is the most trending IT service model. It provides on-demand computing resources (computation, storage, and programming services via API) and infrastructure based on the pay-per-use method. It minimizes the associated costs and maximizes the efficient use of resources. The advantage of utility computing is that it reduces IT costs, provides greater flexibility, and is easier to manage.
Large organizations such as Google and Amazon established their own utility services for computing storage and application.

Note: Grid computing, Cloud computing, as well as managed IT services follow the concept of utility computing.

Containers and Container Orchestration

Definition and Features of Containers:
Applications may be run reliably across many computing environments, including physical servers, virtual machines, and cloud platforms, thanks to containers, a lightweight form of virtualization. Without the overhead of a full operating system, containers offer an isolated and secure runtime environment for apps and their dependencies.
Features:

- Portability: Applications can be deployed and scaled rapidly and effectively since containers are easily transported between various computing environments.
- Resource Effectiveness: Containers provide larger densities and better utilization of computer resources because they are lightweight and use fewer resources than conventional virtual machines.
- Compatibility: Containers are an adaptable and flexible option for application deployment since they are compatible with a variety of programming languages, frameworks, and tools.
- Isolation: Containers give applications a safe and secure environment in which to execute, preventing them from interfering with one another or the host operating system.
- Scalability: Without complicated configuration or management, containers can be scaled up or down easily to suit shifting demands.

Container Orchestration Tools: Kubernetes and Docker

- Modern container-based application deployments require the use of container orchestration tools like Docker and Kubernetes. In a production setting, these tools offer a range of features and functionalities for managing, scaling, and deploying containers.
  A well-known container platform called Docker offers a runtime for containers as well as tools for creating, distributing, and executing them. Developers may simply deploy their programs and dependencies to any environment by packaging them using Docker into containers.
- A container orchestration system called Kubernetes offers cutting-edge features for controlling and growing containerized workloads. By automating container deployment, scaling, and management, Kubernetes makes it simple to deploy applications across a server cluster.

Additionally, Kubernetes offers sophisticated capabilities like rolling updates, load balancing, and self-healing to guarantee that applications are constantly accessible and functional.
Apache Mesos, Amazon ECS, and Google Container Engine (GKE) are further solutions for container orchestration. In terms of functionality, these solutions are comparable to Docker and Kubernetes, but based on the particular use case, they may have different feature sets or deployment approaches.
Benefits of Containerization in Cloud Computing

- Scalability: To adapt to changing demand, containers may be readily scaled up or down. Due to this, it is simple to manage unexpected traffic increases and guarantee that applications are constantly accessible.

- DevOps: Containers are a crucial part of DevOps workflows because they help developers create and test applications in a repeatable and standardized way. This enhances the effectiveness and speed of application distribution.
- Consistency: Regardless of where they are deployed, programs execute in a consistent environment that is provided by containers. This makes managing and upkeep of apps in a cloud environment simpler.
- Portability: Containers can run on any platform that supports containerization and are portable. This enables switching between various cloud providers, on-premises settings, and even various operating systems simple.
- Efficiency: Compared to conventional virtual computers, containers are lighter and use fewer resources. The consequence is better resource utilization and cost savings since more containers can run on a single host.

Use Cases of Containerization in Cloud Computing:

- Hybrid Cloud: Containers can be used to build a hybrid cloud environment that enables the deployment of applications across both public and private clouds. The ability to choose the best cloud for each workload while retaining consistency and portability is made possible by this.
- Modernization of Legacy Applications: By putting historical apps into containers and deploying them in the cloud, containerization can be used to modernize legacy applications. This can lower expenses and increase agility while also enhancing the performance and scalability of legacy applications.
- Cloud-Native Applications: Applications that are built particularly for the cloud are known as "cloud-native," and containerization is a crucial part of this process. Containerization and container orchestration tools like Kubernetes are frequently used in the development of cloud-native applications since they are meant to be highly scalable, resilient, and portable.
- Microservices: Building and deploying microservices, which are compact, independent components of a program, is a good fit for containerization. Microservices can easily be updated or replaced without affecting the system as a whole because they can be bundled into containers and deployed individually.

Big Data and Analytics

An important component of cloud computing is big data and analytics. Here are some details on it:

Big Data is the term used to describe big and complicated data collections that are difficult to process or analyze using conventional data processing techniques. Analytics describes the procedure of studying data and drawing conclusions from it.

Features:

- Volume: Big Data is characterized by its enormous volume, which is frequently measured in terabytes or petabytes.
- Veracity: Big Data is frequently characterized by its incompleteness and uncertainty.
- Velocity: Big Data must be processed in real-time or very close to real-time because it is produced quickly.
- Variety: There are many different types of big data, including structured, semi-structured, and unstructured data.

A scalable and affordable platform for Big Data processing and analytics is provided by cloud computing.

Benefits:

- Cost-Effectiveness: Using cloud computing, businesses only pay for the resources they really use, which lowers the cost of processing and analyzing Big Data.
- Scalability: The infrastructure offered by cloud computing is scalable and can meet the needs for processing and storing Big Data.
- Simple Integration: Big Data technologies and frameworks like Hadoop and Spark are simple to integrate with cloud computing systems.
- Real-time Analytics: Thanks to cloud computing, businesses can perform near-real-time analytics and act on data fast.

Use-Cases:

- Fraud Detection: By using big data analytics, fraud in financial transactions can be found and avoided.
- Customer Analytics: Big Data and analytics can assist businesses in better understanding consumer behavior, preferences, and needs.
- Optimization of Supply Chain: Big Data analytics can be used to optimize supply chain processes, bringing down costs and raising productivity.
- Healthcare: Big Data analytics can be used to optimize the delivery of healthcare and enhance patient outcomes.

Cloud-Based Big Data and Analytics Services: Spark, Hadoop, and Machine Learning Tools

- The term "cloud-based big data and analytics services" refers to the use of cloud computing infrastructure and resources for data storage, processing, and analysis on a massive scale.
- Hadoop, Spark, and machine learning tools are a few of the well-known big data and analytics services that are offered via the cloud.
- Large datasets are processed and stored using the open-source Hadoop architecture by distributed computer clusters. It is very scalable and made to handle massive amounts of data.
- Hadoop is made up of two primary parts: the MapReduce programming methodology for processing massive datasets and the Hadoop Distributed File System (HDFS).
- Another open-source large data processing framework with a focus on speed and usability is Spark.
- It utilizes memory more effectively than Hadoop and offers faster processing speeds thanks to the fact that it is built on top of the Hadoop Distributed File System.

- Businesses can also employ a range of cloud-based machine-learning tools in addition to these frameworks.
- These tools can be used for a variety of applications, including natural language processing, picture recognition, and predictive analytics, and can assist organizations in analyzing and extracting insights from massive datasets.

Benefits of Using Cloud for Big Data Processing and Analytics

- Cost-effectiveness: Big data and analytics services delivered through the cloud take the place of expensive on-premises infrastructure and software, saving businesses money.
- Flexibility: Organisations may test out various analytics techniques and technologies without making a long-term investment thanks to cloud-based big data and analytics services.
- Accessibility: Distributed teams can cooperate more successfully thanks to cloud-based big data and analytics platforms, which can be accessed from any location with an internet connection.
- Scalability: Because the cloud offers practically unlimited computing capabilities, businesses can scale up or down to handle massive amounts of data as needed.
- Innovation: Organisations can stay on the cutting edge of big data and analytics thanks to the ongoing evolution and incorporation of new technologies and capabilities in cloud-based big data and analytics services.
- Speed: Compared to conventional on-premises solutions, cloud-based big data and analytics services can analyze enormous volumes of data significantly more quickly.

Overall, cloud-based big data and analytics services give businesses a flexible, scalable, and affordable means to handle and analyze huge amounts of data, giving them the knowledge they need to improve their operations.

Serverless Computing:

A cloud computing architecture known as "serverless computing," or "Function-as-a-Service" (FaaS), relies on the cloud provider to manage the infrastructure required to run and scale applications while the user only needs to concentrate on creating and distributing code. In serverless computing, the user creates actions that are triggered by specific occurrences, such as HTTP requests or database changes, and the cloud provider automatically sets up and scales the necessary computing resources to carry out those actions.

Features:

- No Management of Infrastructure: Users may concentrate on writing and delivering code because the cloud provider manages the network, servers, and operating systems that make up the underlying infrastructure.
- Reduced Time-to-Market: Serverless computing enables developers to deploy and scale their applications quickly without worrying about infrastructure setup and management, which can reduce the time it takes to market new features and applications.
- Event-Driven Architecture: Serverless computing is made to react to particular occurrences like user requests, database changes, or predetermined events.
- Pay-per-use Pricing: Serverless computing allows customers to only pay for the resources they really use to carry out their duties, which can result in cost savings.

- Scalability and High Availability: Serverless computing platforms can dynamically assign computer resources based on the demand for processing, making them extremely scalable.

Event-Driven Computing Model and Pay-per-Invocation Pricing

- In a serverless computing architecture, the infrastructure is managed by the cloud provider, who then automatically allots resources to execute applications or services depending on real demand.
- In this model, the developer only needs to concentrate on writing the code for the particular function or task and does not need to manage or provision the servers or other infrastructure resources.
- The event-driven computing model of serverless computing is one of its key characteristics. This means that only when a specific event, such as a user request or a change in data, would the cloud provider allocate resources to execute the application or service.
- The resources are released and added back to the provider's pool once the task or function is finished.
- The pay-per-invocation price structure of serverless computing is another feature. Accordingly, the developer is only charged for the application's or service's actual usage, calculated as a function of how frequently it is called upon or triggered.
- When it comes to software or services, this pricing structure can result in cost savings when they are only sometimes utilized or receive little traffic.

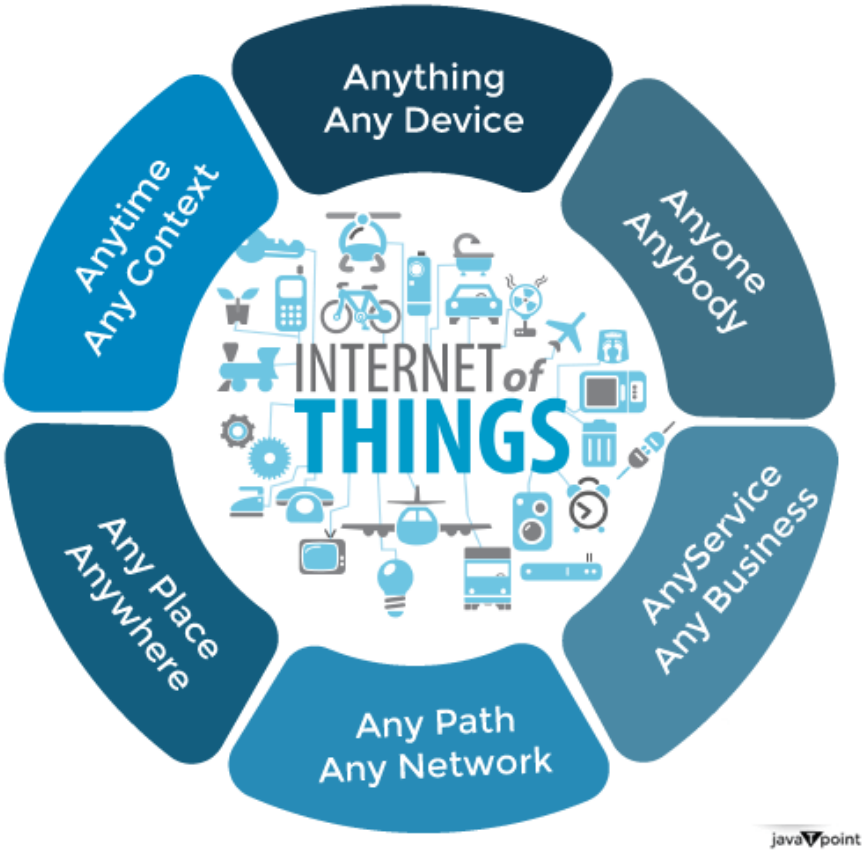
Benefits of Serverless Computing

- Faster Time to Market: Because serverless computing does not require infrastructure management, developers can concentrate on building code and releasing their apps more quickly.
- Better Scalability: Serverless computing automatically scales based on the volume of incoming requests, ensuring that your application can handle sudden increases in traffic without requiring any manual intervention.
- Reduced Operational Overhead: Since server provisioning, configuration, and maintenance are handled entirely by the cloud provider with serverless computing, developers don't have to worry about it. Developers can concentrate on writing code as a result of the decreased operational overhead.
- Savings: Because serverless computing uses a pay-per-invocation business model, you only pay for the resources your function really consumes. Especially for applications with intermittent or unexpected workloads, this might result in significant cost savings.

Serverless Computing has a Few Typical Use Cases, Such as:

- Data Processing and Analytics: Using tools like Apache Spark and AWS Lambda, serverless computing can be utilized to analyze enormous volumes of data and perform analytics on that data.
- Internet of Things (IoT): Serverless computing can be used to manage the data produced by IoT devices and enables real-time processing, storing, and analysis of that data.
- Web and Mobile Applications can employ serverless computing to power their back-ends, managing operations like content delivery, data processing, and authentication.
- Chatbots and Virtual Assistants can be powered by serverless computing and can handle activities like natural language processing and generating user query responses.

Internet Of Things (IoT)



Cloud-Based IoT Platforms and Services

The management and processing of data produced by connected devices in the Internet of Things (IoT) ecosystem is the goal of cloud-based IoT platforms and services. With the help of these services, IoT device data may be stored, processed, and analyzed in a secure and scalable architecture.

Features:

Processing and Analysing Data from IoT Devices: Cloud-based IoT systems and services offer strong analytics tools, such as real-time data stream processing, batch processing, and machine learning.

Security: Cloud-based IoT systems and services include strong security capabilities, such as encryption, access control, and threat detection, to shield data and devices from cyber threats.

Device Management: Tools for managing a lot of devices, including provisioning, monitoring, and firmware updates, are offered by cloud-based IoT platforms and services.

Data Intake and Storage: To handle the enormous amounts of data created by IoT devices, cloud-based IoT platforms, and services offer scalable and adaptable storage options.

The Following are Some Advantages of Employing Cloud-Based IoT Platforms and Services:

- Scalability: To accommodate the rising volume of data produced by IoT devices, cloud-based IoT systems, and services can easily scale up or down.
- Speed to Market: IoT platforms and services that are cloud-based provide a quick and simple way to develop and manage IoT applications, enabling businesses to swiftly launch new goods and services.
- Flexibility: To accommodate many use cases, including real-time analytics, predictive maintenance, and anomaly detection, cloud-based IoT platforms and services provide a wide range of storage and processing possibilities.

- Cost-Effectiveness: Pay-as-you-go pricing models are available on cloud-based IoT platforms and services, enabling businesses to only pay for the resources they actually use rather than making a large upfront investment in infrastructure and hardware.

The platforms and services offered by AWS IoT, Microsoft Azure IoT, Google Cloud IoT, and IBM Watson IoT are a few examples of cloud-based IoT solutions.

Integration of Cloud Computing with IoT Devices and Applications

- The process of connecting IoT devices to cloud-based platforms and services in order to store, process, and analyze the data produced by these devices is referred to as the Integration of Cloud Computing with IoT Devices and Applications.
- This integration enables IoT devices to utilize the scalability, agility, and cost-effectiveness of cloud computing to handle the enormous amounts of data created by these devices.
- The features and capabilities offered by cloud-based IoT platforms and services include data management, real-time data processing, analytics, and machine learning. Various IoT use cases, including smart homes, smart cities, industrial IoT, and healthcare IoT, are supported by these platforms and services.
- Additionally, various tools and frameworks for creating and deploying IoT applications, such as development kits, software development kits (SDKs), and application programming interfaces (APIs), are offered by cloud-based IoT platforms and services.
- The creation and deployment of IoT applications on cloud-based platforms and services are made simpler for developers by these tools and frameworks.
- In general, organizations may create and implement scalable, secure, and affordable IoT solutions that can spur innovation and enhance business outcomes thanks to the combination of cloud computing with IoT devices and apps.

Benefits:

Numerous Advantages of Cloud-Based IoT Solutions Include:

- Cost Savings: Organisations no longer need to invest in pricey infrastructure to support their IoT deployments thanks to cloud-based IoT solutions. Instead, companies can make use of the IoT platform provider's cloud infrastructure.
- Scalability: Depending on the amount of data provided by IoT devices, cloud-based IoT platforms can scale up or down. This enables businesses to manage massive amounts of data and scale up their IoT implementations as necessary.
- Enhanced Security Features, including encryption, multi-factor authentication, and access control, are available on cloud-based IoT solutions. This aids in shielding sensitive IoT data from online dangers.
- Real-Time Data Analysis: Organisations may swiftly make educated decisions with the help of real-time data analysis offered by cloud-based IoT technologies. Real-time insights can assist in preventing equipment failures and downtime in applications like predictive maintenance, where they are very helpful.
- Increased Adaptability: Organisations may tailor their IoT installations to suit their unique requirements thanks to the great degree of adaptability offered by cloud-based IoT solutions. This includes the option to customize the data processing and analytics tools as well as a range of IoT sensors and devices.

Use-Cases:
The Following are Some Use Cases for Cloud-Based IoT Solutions:

- Smart Houses and Buildings: Various features of a home or building, such as lighting, temperature, and security, can be automated and controlled via cloud-based IoT platforms.
- Industrial Automation: By using IoT devices to monitor and manage industrial equipment, businesses may improve production procedures and lower downtime.
- Healthcare: Real-time monitoring of patient vital signs and health data via IoT devices enables early intervention and better patient outcomes.
- Agriculture: Farmers can use IoT devices to monitor crop growth, weather conditions, and soil moisture levels to optimize irrigation and fertilization and boost agricultural yields.
- Smart Cities: By monitoring traffic, parking, and public transportation with IoT devices, city planners can optimize the city's infrastructure and ease congestion.

In conclusion, cloud computing technologies include virtualization, SOA, grid computing, containers, big data analytics, serverless computing, and IoT, whereas cloud computing services include utility computing. These products and services are essential for delivering effective and scalable cloud computing solutions.

What is cloud networking:

Cloud networking uses cloud-based technologies to connect users, applications, and other essential resources. All of the networking technologies — including physical hardware and management capabilities — are hosted in a public or private cloud environment.
To implement cloud networking, some organizations opt for network-as-a-service (NaaS) offerings from cloud providers. These offerings enable organizations to connect their on-premises data centers and cloud environments while reducing management complexity. Organizations do not need any of their own network infrastructure — only Internet connectivity. In other cases, organizations might choose to use cloud technologies to support private cloud or hybrid cloud environments.
Multi-cloud networking uses capabilities from more than one public cloud platform. With multi-cloud networking, organizations can optimize networking functions across those clouds for performance, cost, availability, and other criteria.

How does cloud networking work?

Cloud networking typically requires three primary components: cloud infrastructure, virtualization capabilities, and cloud-based management tools.

- **Cloud infrastructure:** When you use a public cloud platform for cloud networking, the cloud provider owns and operates the physical infrastructure. If that infrastructure is available across globally distributed data centers, you can give your users low-latency access to apps and resources from anywhere.
- **Virtualization:** Cloud networking virtualizes network components, abstracting their functions from underlying physical infrastructure. Virtualization enhances the flexibility of network configurations and increases resource utilization, which in turn helps reduce the costs for cloud networking.
- **Cloud-based management:** With cloud networking, you can manage, monitor, and secure the network through cloud-based tools. Like with software-defined networking, using cloud-based tools to manage networking enhances flexibility and increases efficiency.

What are the benefits of cloud networking?

When you use public cloud platforms for cloud networking, you can realize several key cloud computing benefits.

- **Security:** You can often tap into a wide range of security capabilities offered by cloud providers. In many cases, those providers offer more advanced capabilities than you could cost-effectively deploy on premises.
- **Performance:** By taking advantage of a cloud provider's distributed data centers, you can deliver robust, low-latency performance to users around the globe.
- **Scalability:** Unlike with traditional networking, you can easily scale services up — or down — as your needs change, without large infrastructure purchases.
- **Availability:** Cloud platforms' multiple, distributed data centers can offer the redundancy needed to maintain high networking availability.
- **Efficiency:** Administrators can manage all networking functions with straightforward, centralized software, eliminating the complexity of traditional network management.
- **Costs:** When using a public cloud platform, you can avoid upfront capital expenditures and choose subscriptions that let you pay for what you use. Still, efficient management is essential for controlling ongoing costs.
- **Agility:** Cloud networking lets you make changes quickly. Instead of taking weeks to support new enterprise locations or integrate new resources, you can make connections in a few hours.

What are the limitations of cloud networking:

Though there are numerous benefits from cloud networking, some organizations might be concerned with the potential for latency, vendor lock-in, and loss of control.

- **Latency:** Using a public cloud for cloud networking requires reliable, low-latency Internet connectivity. If you choose a cloud platform that does not have data centers close to all users, some users could experience latency when accessing key apps or other resources.
- **Vendor lock-in:** If you select a public cloud platform with proprietary technologies, it might be difficult to integrate additional clouds or move to other platforms in the future.
- **Limited control:** When you use a public cloud provider for cloud networking, you rely on that provider to manage the underlying physical infrastructure. You have less control and might have less configurability than if you owned and operated the infrastructure yourself.

Cloud networking vs. traditional networking:

In a traditional networking model, an organization owns and operates their own networking equipment, including switches, routers, gateways, load balancers, and firewalls. That network infrastructure is housed in on-premises data centers. When teams need to scale up their networking resources, they must buy and implement additional equipment.

If you use a public cloud for cloud networking, the physical infrastructure is owned and operated by the cloud provider. Because networking functions are abstracted from physical equipment, you have great configuration flexibility. And because cloud providers often have vast infrastructure resources, you can scale easily. Your team can configure and manage the network virtually, using cloud-based tools.

Cloud-based networking vs. cloud-enabled networking:

Cloud networking, sometimes called “cloud-based networking,” is different than “cloud-enabled” networking. With cloud-based networking, the physical infrastructure, networking capabilities, and management tools all come from a cloud provider.
With cloud-enabled networking, you have a traditional, on-premises network, but your administrators manage or secure that network using cloud-based resources. So, for example, you might use a cloud firewall or anti-virus solution to protect your on-premises network.

Cloud networking vs. cloud computing:

Cloud networking is just one of many types of services offered by cloud providers. These providers typically have additional types of infrastructure-as-a-service (IaaS) offerings, such as compute and storage services. Most providers also have multiple software-as-a-service(SaaS) offerings, including databases, analytics, security services, artificial intelligence (AI)/machine learning (ML) services, developer tools, and more.

Cloud Network Security:

Cloud network security is a critical aspect of safeguarding containerized applications and their data in the modern computing landscape. It involves securing network communication and configurations for these applications, regardless of the orchestration platform in use. Cloud network security addresses network segmentation, namespaces, overlay networks, traffic filtering, and encryption for containers. By implementing cloud network security technologies and best practices, organizations can effectively prevent network-based attacks like cryptojacking, ransomware, and BotNetC2 that can impact both public-facing networks and internal networks used by containers to exchange data.

Cloud Network Security Explained:

All workloads run on the same network stack and protocols, regardless of whether they run on bare-metal servers, virtual machines, or in containers. In other words, containerized workloads are subject to many of the same network-based attacks as legacy applications — cryptojacking, ransomware, BotNetC2, and more.
Network-based security threats, however, can impact containers in two ways — via public-facing networks that connect applications to the internet and via internal networks that Kubernetes containers use to exchange data within each other.
Cloud network security focuses on securing network communication and configurations for containerized applications in general, regardless of the orchestration platform. It addresses aspects such as network segmentation, namespaces, overlay networks, traffic filtering, and encryption for containers.
Kubernetes network security targets network security within a Kubernetes cluster and encompasses Kubernetes-specific features, such as network policies, ingress and egress controls, namespace isolation, role-based access control (RBAC), and service mesh implementation.
Detecting signs of malicious activity on both types of networks requires both container security and Kubernetes network security. As these are distinct domains, they warrant separate discussions to cover their unique aspects. In this section, we’ll delve into the

various aspects of cloud network security as it pertains to containers and discuss best practices for safeguarding your environment.
Once deployed, containers need to be protected from attempts to steal proprietary data and compute resources. Cloud network security proactively restricts unwanted communication and prevents threats from attacking your applications once deployed. Containerized next-generation firewalls, web application and API security (WAAS), and microsegmentation tools inspect and protect all traffic entering and exiting containers (north-south and east-west), granting full Layer 7 visibility, and control over the Kubernetes environment. Additionally, containerized firewalls dynamically scale with the rapidly changing size and demands on the container infrastructure to provide security and bandwidth for business operations.

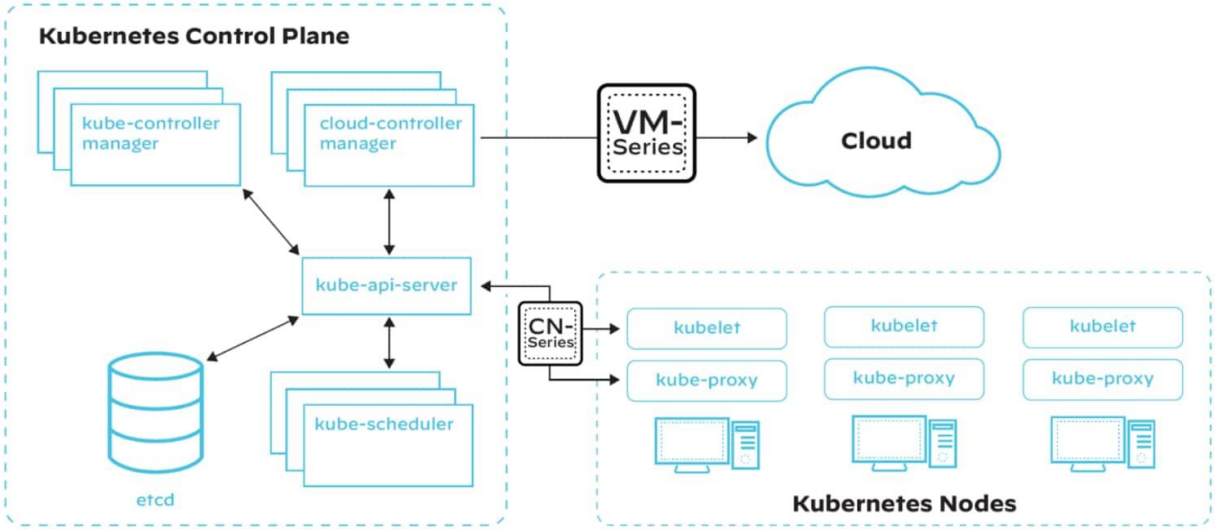


Figure 1: A simplified Kubernetes architecture with overlaid VM-Series (virtualized next-generation firewall) and CN-Series (containerized firewall tailored for securing Kubernetes-based containerized applications)

Network Segmentation

Network segmentation is the practice of dividing a network into smaller, isolated segments to limit unauthorized access, contain potential threats, and improve overall network performance. In containerized environments, security teams can achieve network segmentation through a variety of methods.

Network Namespaces

Network namespaces provide isolation between containers by creating a separate network stack for each, including their own network interfaces, routing tables, and firewall rules. By leveraging network namespaces, you can prevent containers from interfering with each other's network configurations and limit their visibility to only the required network resources.

Overlay Networks

Overlay networks create a virtual network layer on top of the existing physical network, which allows containers to communicate across different hosts as if they were on the same network. Popular overlay network solutions for containers include Docker's built-in overlay driver, Flannel, and Weave.

Network Partitions and Security Groups

Network partitions and security groups can further segment container networks by creating logical boundaries and applying specific firewall rules to restrict traffic between segments.

Traffic Filtering and Firewall Rules

Containerized next-generation firewalls stop malware from entering and spreading within the cluster, while also preventing malicious outbound connections used in data exfiltration and command and control (C2) attacks. Although shift-left security tools provide deploy-time protection against known vulnerabilities, containerized next-

generation firewalls provide protection against unknown and unpatched vulnerabilities.
Traffic filtering and firewall rules are essential for controlling the flow of traffic between containers, as well as between containers and the host.

Egress and Ingress Filtering

Egress filtering controls the outbound traffic from a container, while ingress filtering controls the inbound traffic to a container. By applying egress and ingress filtering, you can limit the exposure of your containers to external threats and restrict their communication to only the necessary services.

Applying Firewall Rules to Container Traffic

Firewall rules can be applied at various levels, including the host, container, and network level. You can use Linux iptables or firewalld, for instance, to create rules that govern container traffic and protect your infrastructure from unauthorized access and malicious activities.

Load Balancing and Traffic Routing

Load balancing and traffic routing are important for distributing traffic across multiple containers and ensuring high availability of your applications. Solutions like HAProxy, NGINX, or Kubernetes' built-in services can be used to route traffic to the appropriate container based on predefined rules and health checks.

Encryption and Secure Communication

Encrypting and securing communication between containers, and between containers and the host, is vital for protecting sensitive data and maintaining the integrity of your applications.

Transport Layer Security (TLS) for Container Traffic

TLS provides encryption and authentication for data transmitted over a network. By implementing TLS for container traffic, you can ensure that data transmitted between containers and between containers and the host is encrypted and secure from eavesdropping or tampering. You can achieve this by using tools like OpenSSL or Let's Encrypt to generate and manage TLS certificates for your containers.

Securing Container-to-Container Communication

To secure communication between containers, you can use container-native solutions like Docker's built-in encrypted networks or third-party tools like Cilium, which provides API-aware network security for containers. These solutions enable you to implement encryption, authentication, and authorization for container-to-container traffic.

Securing Container-to-Host Communication

Ensuring secure communication between containers and the host can be achieved by using host-level encryption and authentication mechanisms, such as SSH or TLS-protected APIs, to control access to container management interfaces and data storage systems.

Kubernetes Network Security

Network Policies

Network policies are a key feature of Kubernetes that allows you to control the flow of traffic within your cluster and between your cluster and external networks. Modern tools make it possible for security teams to define policies that essentially determine who and what are allowed to access any given microservice. Organizations need a framework for defining those policies and making sure that they’re consistently maintained across a highly distributed container application environment.

Defining and Enforcing Network Policies

Kubernetes network policies are defined using YAML files, which specify the allowed traffic between components like pods, services, and namespaces. Once defined, these policies can be enforced using network plugins that support Kubernetes network policy API, such as Calico or Cilium.

Whitelisting and Blacklisting Traffic

Network policies can be used to whitelist or blacklist traffic between components of

your cluster based on criteria that include pod labels, IP addresses, or namespaces. Establishing these will allow you to control which services can communicate with each other and prevent unauthorized access to sensitive data or resources.

Namespace Isolation and Segmentation

By applying network policies at the namespace level, you can isolate and segment applications or environments within your cluster, restricting traffic to only the necessary components and preventing potential security risks.

Ingress and Egress Controls

Controlling ingress and egress traffic is critical for managing the flow of data into and out of your Kubernetes cluster and protecting it from external threats.

Ingress Controllers and Load Balancing

Ingress controllers in Kubernetes manage the routing of external traffic to the appropriate services within your cluster based on predefined rules. Load balancing can be achieved through built-in Kubernetes services or third-party solutions like NGINX and HAProxy. These solutions allow you to route traffic based on criteria such as path, host, or headers. They also allow you to provide TLS termination and other security features.

Ingress Access Best Practices

- Correct the default "any-any-any allow" Kubernetes policy by applying a deny-all policy for every namespace.
- Prevent services from accepting incoming traffic directly from external IPs unless a load-balancer or ingress is attached. Only allow incoming traffic from load-balancers or ingresses.
- Limit traffic to specific protocols and ports according to the service's requirements (e.g., HTTP/HTTPS for web services, UDP 53 for DNS service).
- Accept traffic only from other services (pods) that consume them, whether in the same namespace or from another.
- To create an ingress policy from a pod in another namespace, add a label to the namespace.

Egress Traffic Management

Controlling egress traffic from your Kubernetes cluster is essential for preventing data leakage and ensuring that outbound connections are restricted to only the required destinations. You can achieve this by using egress network policies, which allow you to define rules for outbound traffic from your pods or namespaces. Additionally, egress gateways or proxy solutions like Squid can be used to control and monitor outbound traffic from your cluster.

Egress Access Best Practices

- Understand the need for each external service used by your microservices. Products like Prisma Cloud Compute Defender can help identify external flows engaged by microservices.
- If a pod needs to connect to a DNS (FQDN) name without a fixed IP address, use an external firewall or proxy, as Kubernetes network policies only support IP addresses.
- Prevent outbound traffic from pods that don't need external connections to reduce the risk of data exfiltration or downloading malicious binaries.
- Apply a block egress policy if you have no external dependencies but ensure essential services like Kubernetes DNS Service remain connected when enforcing egress policies.

Identity-based microsegmentation helps restrict the communication between applications at Layer 3 and Layer 4 while containerized next-gen firewalls perform Layer 7 deep packet inspection and scan all allowed traffic to identify and prevent known and unknown threats.

DNS Policies and Security

DNS is integral to Kubernetes networking, as it provides name resolution for services

and other components within your cluster. Ensure the security and integrity of your DNS infrastructure to prevent attacks like DNS spoofing or cache poisoning. Kubernetes provides built-in DNS policies for controlling the behavior of DNS resolution within your cluster, and you can also use external DNS providers or DNS security solutions like DNSSec to enhance the security of your DNS infrastructure.

Service Mesh and Network Encryption

Service mesh is a dedicated infrastructure layer that provides advanced networking features, such as traffic routing, load balancing, and security for your microservices and containerized applications.

Implementing Service Mesh

Service mesh solutions like Istio and Linkerd can be integrated with your Kubernetes cluster to provide advanced networking capabilities and enhance the security of your containerized applications. These solutions offer features like mutual TLS, access control, and traffic encryption, which can help protect your applications, particularly microservices, from various security threats.

Mutual TLS (mTLS) for Secure Communication

Mutual TLS (mTLS) is a security protocol where both the client and server authenticate each other's identities before establishing a secure connection. Unlike traditional TLS, where only the server is authenticated by the client, mTLS adds an additional layer of security by requiring the client to present a certificate. The added requirement verifies that both parties are who they claim, which can help prevent unauthorized access, data leakage, and man-in-the-middle attacks.

Observability and Control of Network Traffic

Service mesh solutions also provide observability and control over your network traffic, allowing you to monitor the performance and security of your applications in near-real time. Early identification of unauthorized access, unusual traffic patterns, and other potential security issues means you can take early corrective actions to mitigate the risks.

Encrypting Traffic and Sensitive Data

To ensure the confidentiality and integrity of data within your cluster, it’s essential to implement encryption techniques for both internal and external communications.

IPsec for Encrypting Communication Between Hosts

IPsec safeguards cluster traffic by encrypting communications between all master and node hosts. Remain mindful of the IPsec overhead and refer to your container orchestration documentation for enabling IPsec communications within the cluster. Import the necessary certificates into the relevant certificate database and create a policy to secure communication between hosts in your cluster.

Configuring Maximum Transmission Unit (MTU) for IPsec Overhead

Adjust the route or switching MTU to accommodate the IPsec header overhead. For instance, if the cluster operates on an Ethernet network with a maximum transmission unigt (MTU) of 1500 bytes, modify the SDN MTU value to account for IPsec and SDN encapsulation overhead.

Enabling TLS for API Communication in the Cluster

Kubernetes assumes API communication within the cluster is encrypted by default using TLS. Most installation methods create and distribute the required certificates to cluster components. Be aware, though, that some components and installation methods may enable local ports over HTTP. Administrators should stay informed about each component's settings to identify and address potentially insecure traffic.

Kubernetes Control Plane Security

Control planes, particularly in Kubernetes clusters, are prime targets for attacks. To enhance security, harden the following components through inspection and proper configuration:

- Nodes and their perimeters
- Master nodes

- Core components
- APIs
- Public-facing pods

While Kubernetes' default configuration provides a certain level of security, adopting best practices can strengthen the cluster for workloads and runtime communication.

Network Policies (firewall rules)

Kubernetes' flat network allows all deployments to reach other deployments by default, even across namespaces. This lack of isolation between pods means a compromised workload could initiate an attack on other network components. Implementing network policies can provide isolation and security.

Pod Security Policy

Kubernetes permits pods to run with various insecure configurations by default. For example, running privileged containers with root permissions on the host is high risk, as is using the host's namespaces and file system or sharing the host's networking. Pod security policies enable administrators to limit a pod's privileges and permissions before allowing deployment into the cluster. Isolating non-dependent pods from talking to one another using network policies will help prevent lateral movement across containers in the event of a breach.

Secrets Encryption

Base distributions of Kubernetes don’t encrypt secrets at rest by default (though managed services like GKE do). If an attacker gains access to the key-value store (typically Etcd), they can access everything in the cluster, including unencrypted secrets. Encrypting the cluster state store protects the cluster against data-at-rest exfiltration.

Role-Based Access Control

While RBAC is not exclusive to Kubernetes, it must be configured correctly to prevent cluster compromise. RBAC allows for granular control over the components in the cluster that a pod or user can access. By restricting what users and pods can view, update, delete, and create within the cluster, RBAC helps limit the potential damage of a compromise.

Addressing Control Plane Security Through Virtual Patching

Minimizing admin-level access to control planes and ensuring your API server isn’t publicly exposed are the most important security basics.

DevOps and SecOps teams can identify vulnerabilities in application packages, but mitigating these risks takes time. Vulnerable packages must be replaced or patched and tested before deployment, which leaves the environment exposed until the resolving the issue. Solutions like Prisma Cloud automate vulnerability mapping for each workload to provide virtual patching for known vulnerabilities. By utilizing its WAAS component, the solution adjusts traffic inspection policies to detect and block remote HTTP-based exploits.

Network Security Best Practices for Containers and Kubernetes

In summary of the areas discussed, the following best practices serve as a checklist to ensure your teams are equipped to safeguard your containerized applications and data from network-based threats.

Monitoring and Logging Network Traffic

Keeping a close eye on your network traffic is paramount for detecting and responding to security incidents and maintaining the overall health of your containerized environment.

Centralized Logging and Monitoring Solutions

Implementing a centralized logging and monitoring solution for your container and Kubernetes environment, such as ELK Stack, Prometheus, or Prisma Cloud, can help you collect, analyze, and visualize network traffic data from a variety of areas. Easy access to centralized data intel will enable you to identify trends, detect anomalies, and gain insights into the performance and security of your infrastructure.

Detecting and Responding to Security Incidents

Monitoring network traffic and establishing alerts for unusual or suspicious activities allows for quick detection and response to incidents involving unauthorized access, data exfiltration, and other malicious activities. Security teams are equipped to take appropriate measures such as isolating affected components, blocking malicious IPs, or updating firewall rules in a timely manner.

Network Traffic Visualization and Analysis

Visualizing and analyzing your network traffic data can help you identify patterns and trends that may indicate potential security risks. Tools like Kibana, Grafana, or custom-built dashboards can be used to create visual representations of your network traffic, allowing you to spot anomalies and investigate security incidents more effectively.

Secure Network Configurations

Hardening your network configurations and implementing strong access controls are vital for protecting your container and Kubernetes environment from security threats.

Hardening Host and Cloud Network Settings

Maintaining the security of your environment requires ensuring secure network configurations for both your container host and individual containers. Essential measures involve disabling unused network services, limiting network access to only the necessary components, and applying security patches and updates to your host operating system and container runtime.

Network Access Controls and Authentication

To prevent unauthorized access and maintain the integrity of your container and Kubernetes environment, it’s essential to implement strong access controls and authentication mechanisms. Key measures involve utilizing role-based access control (RBAC) for managing user permissions in Kubernetes, incorporating multifactor authentication (MFA), and employing network security solutions, such as VPNs or firewalls, to limit access to your environment.

Regular Network Security Assessments

Regular network security assessments — vulnerability scans, penetration tests, and security audits — are a must when it comes to identifying potential weaknesses in your container and Kubernetes environment. Key aspects of these assessments involve examining network configurations, firewall rules, and security policies to ensure adherence to industry best practices and compliance requirements.

By following these best practices and implementing effective network security measures, you can fortify your container and Kubernetes environment from potential network-based threats and ensure the safety and integrity of your applications and data.

What is a cloud server:

A cloud server is a virtualized server that runs in the cloud on infrastructure owned by a cloud service provider. Traditionally, organizations had to purchase and maintain their own physical servers. They used the servers to run and host applications and compute workloads required for data processing and analytics. The servers were located on-site or in nearby data centers. Today, your organization can spin up virtual cloud servers anywhere in the world. These virtual spaces run on physical servers that are purchased and maintained by third-party cloud providers. The virtual server replica, or cloud server, gives the same performance, configuration options, and usability as a physical server machine. You can access unlimited cloud servers in hundreds of different configuration types. With this kind of power, you can run and host all types of applications and workloads in the cloud.

What are the benefits of cloud servers:

Cloud servers are a critical part of cloud computing; they remove the need to buy, run, and manage physical servers. You can use them exclusively or in combination with existing server infrastructure. Launching servers in the cloud has never been easier or more configurable. There are now different types of cloud servers available for every business and personal use.

Flexible options
With cloud servers, you can spin up almost any type of server architecture—no matter the underlying hardware. This means you can choose cloud servers based on preferences like graphics capabilities, machine learning workloads, or networking functionality.
Achieving compliance objectives is easy, as you can also choose the geographic region the cloud server is located in. You can choose even its location zone in the cloud computing environment.

Cost-effective management
Investing in physical servers used to be costly and required significant long-term planning. Purchasing a physical server meant many years of investment. Now, you can rent a cloud server on demand for as little as per-second billing. It's possible to rent a number of cloud servers at any given time for different types of workloads, all without any billing lock-ins.
Cloud servers also require no ongoing maintenance costs. The cloud provider can take care of several management aspects like the operating system, configurations, and security updates. This removes the need for in-house management.
Moreover, cloud servers are defined in software, so they don't degrade over time. This also removes any decommissioning costs you'd have if you purchase then retire hardware-based servers.

Scalable provisioning
Cloud servers are often scalable. If you run out of space or power on a server, you can set the server type or number of servers to increase automatically. This adjusts for the bigger workload. You can also do the reverse and automatically downsize to accommodate for smaller workloads.
Cloud servers also come complete with mechanisms to help ensure high availability, such as advanced load balancing and in-built failover diversions.

What are some use cases for a cloud server:

You can use a cloud server to run all types of workloads. Here are some examples:

- Enterprise software, such as human resources (HR) and customer relationship management (CRM) systems
- Customer applications, such as mobile apps and document management
- High-end graphics processing, such as video streaming and games
- Scientific modeling applications
- Databases that are manipulated through incoming database queries
- Web applications and websites, through dedicated web servers running HTTP communications
- Machine learning (ML) workloads, for training ML models that require a large amount of compute power

A cloud server provides scalability and flexibility for all modern applications. You can

use one for artificial intelligence (AI) as well as microservices, analytics, and streaming.

How does a cloud server work:

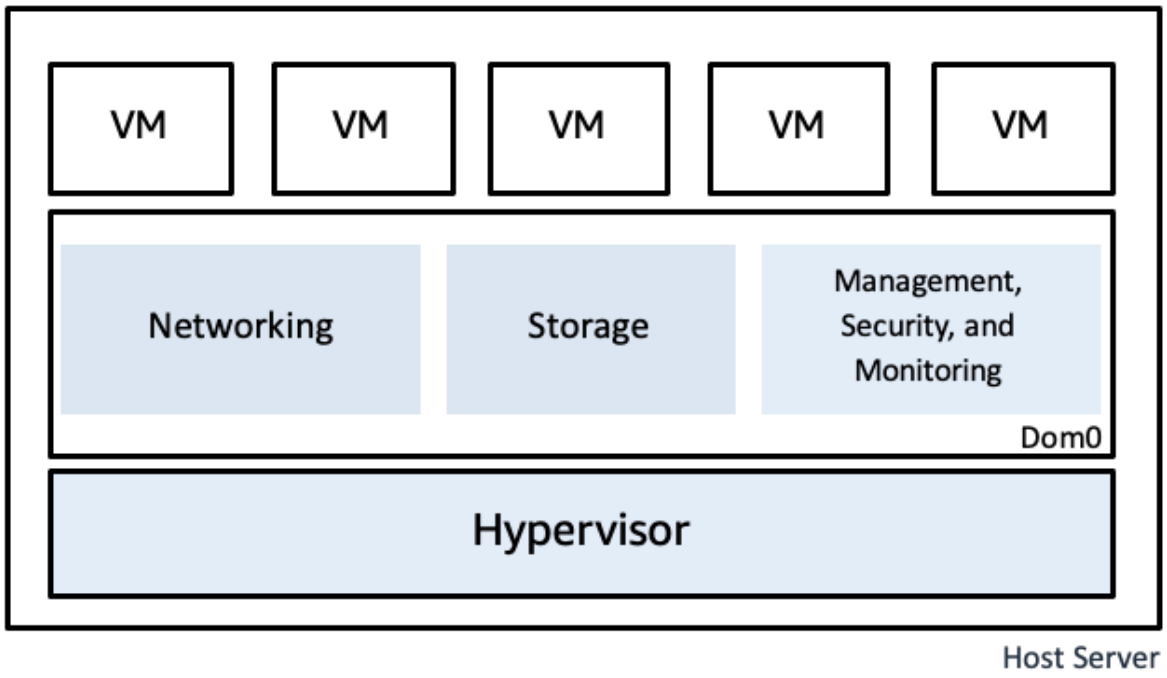
A bare-metal server (or physical server) is a box-like machine with circuits and chips, memory, storage, and CPU. It takes up physical space and requires electricity to run. In contrast, a cloud server, virtual server, cloud instance, or virtual machine (VM) is just software. But it behaves the same way as the physical machine. The cloud server also appears to any other device or connection as a physical server.
Organizations run VMs on their own physical servers. However, cloud servers are strictly virtual machines that are created and managed by a cloud provider. The cloud provider owns and manages the underlying hardware and infrastructure.
Much of cloud computing, including cloud servers and other services offered by cloud service providers, is built on virtualization.

Virtualization

Virtualization is the process of creating and running a virtual instance of a real-life IT resource. Multiple virtual servers can run on the same physical machine, sharing those underlying computing resources.
With virtualization, you don't have to lock the entire hardware to a single operating system and configuration environment. Instead you can run different operating systems, workloads, and apps in multiple fully isolated virtual environments. Isolated virtual servers help you with greater resource sharing. They're often more cost-effective for businesses.

Provisioning

With cloud server provisioning, you allocate and configure computing resources within a cloud environment to deploy VMs. You provision cloud servers using APIs. The APIs allow you to create, configure, delete, and manage your cloud servers remotely. This process typically starts with specifying the desired server attributes—such as CPU, memory, storage, and network capabilities. You also specify the operating system and any preinstalled software.
Once you define the parameters, automated tools within the cloud platform instantiate the cloud servers, associate them with the appropriate resources, and configure networking and security settings. This enables a quick and scalable deployment of computing power tailored to specific needs.
Typically, cloud servers come preloaded with a Linux-based OS. Choosing the right server type depends on the task at hand. Some server types and configurations are better suited to certain types of workloads.



What are the types of cloud servers:

We classify cloud servers by their configuration and how they map to the underlying physical server infrastructure.

Configuration

You can choose from a range of preconfigured cloud servers for different use cases. We give some examples next.

General purpose

These instances offer a balanced ratio of CPU, memory, and storage. This makes them suitable for a wide range of applications like web servers and small-to-medium databases.

Compute-optimized

These instances are designed for CPU-intensive workloads. They provide a high ratio of CPU cores to memory. So, they're ideal for compute-bound applications like batch processing and scientific modeling.

Memory-optimized

These instances offer a high amount of RAM relative to CPU cores. They're suitable for applications that require large datasets to be kept in memory, such as in-memory databases and big data analytics platforms.

Accelerated computing

These instances are equipped with hardware accelerators like graphics processing units (GPUs) or field-programmable gate arrays (FPGAs). They're optimized for specialized tasks such as MK, graphics rendering, and scientific simulation.

Storage-optimized

These instances offer high disk throughput and are optimized for workloads that require high-speed access to large volumes of data, such as big data analytics and data warehousing.

High-performance computing

High-performance computing (HPC) instances are customized for computationally intensive workloads that require high network performance and low latency. They're good for tasks like fluid dynamics simulations, seismic analyses, and other scientific computations.

Hosting type

Cloud servers can also be classified by their hosting type and plan. All cloud servers

are only used by one account. However, the underlying infrastructure differs between shared and dedicated hosting.

Shared hosting

In a shared hosting environment, multiple cloud servers share the resources of a single underlying physical server. High workloads on one cloud server may impact the performance of others.

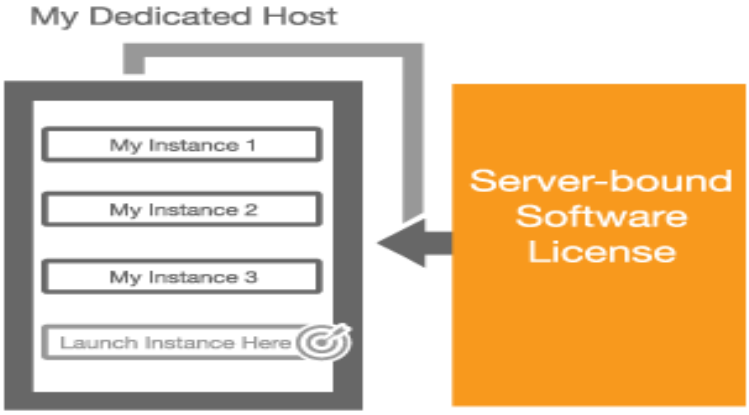
Virtual private server hosting

A virtual private server (VPS) is a cloud server that runs on the same physical server as other cloud servers. However, it's allocated its own dedicated portion of the server's resources. High workloads on other servers do not impact VPS performance.

Dedicated hosting

Dedicated hosting provides an entire physical server to a single organization. The organization can configure the physical machine as a single cloud server or multiple servers to completely control the environment they run their servers on.

In some places, shared hosting is also known as *public cloud servers* and dedicated hosting is known as *private cloud servers*. Despite that naming, all cloud servers are private. The difference is that dedicated hosting isolates cloud servers at the hardware level. On the other hand, public cloud servers isolate at the software level.



Client-Server Model

The Client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and delivers the data packets requested back to the client. Clients do not share any of their resources. Examples of the Client-Server Model are Email, World Wide Web, etc.

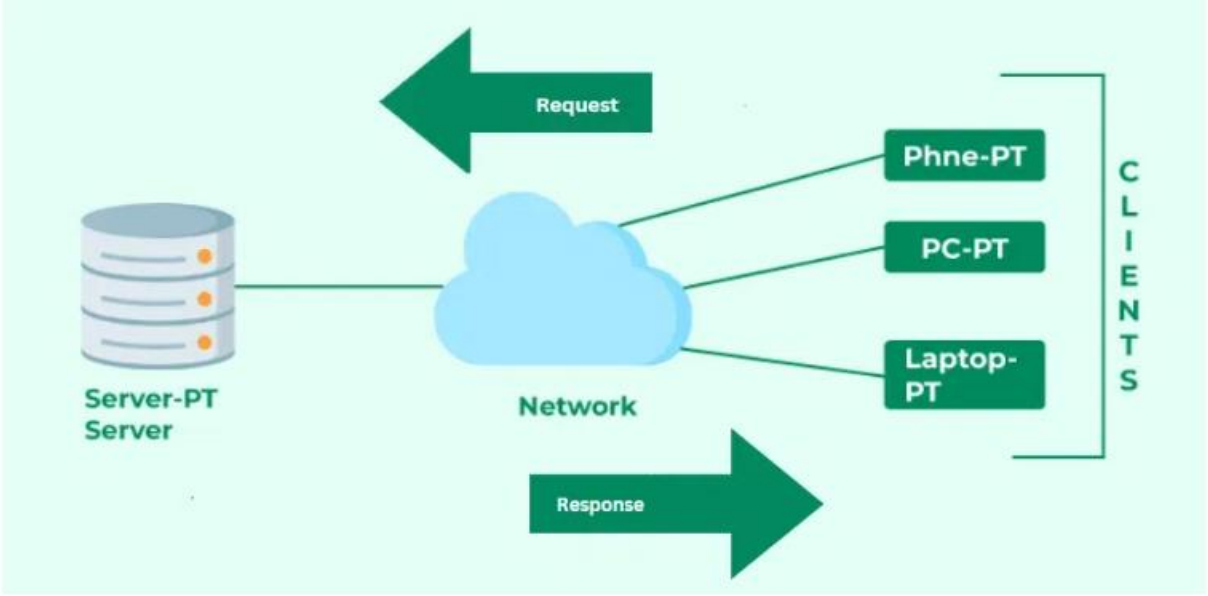
How Does the Client-Server Model Work:

In this article, we are going to take a dive into the **Client-Server** model and have a look at how the **Internet** works via, web browsers. This article will help us have a solid WEB foundation and help us easily work with WEB technologies.

- **Client:** When we say the word **Client**, it means to talk of a person or an organization using a particular service. Similarly in the digital world, a **Client** is a computer (**Host**) i.e. capable of receiving information or using a particular service from the service providers (**Servers**).
- **Servers:** Similarly, when we talk about the word **Servers**, It means a person or medium that serves something. Similarly in this digital world, a **Server** is a remote computer that provides information (data) or access to particular services.

So, it is the **Client** requesting something and the **Server** serving it as long as it is in

the database.
For those new to networking concepts, the System Design Course provides a comprehensive overview of the client-server model and its applications in modern computing.



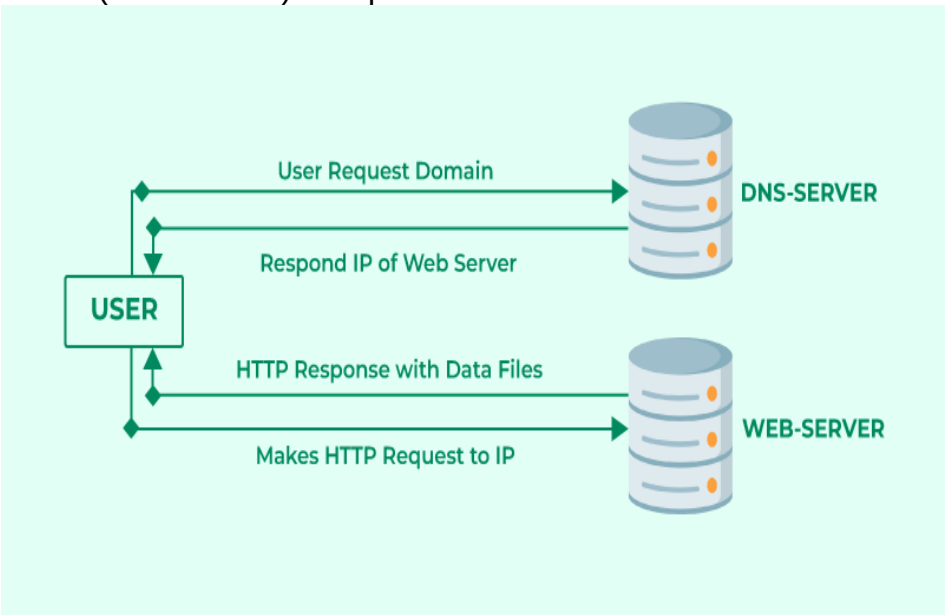
Client Server Model

Client Server Model

How the Browser Interacts With the Servers?

There are a few steps to follow to interacts with the servers of a client.

- User enters the **URL**(Uniform Resource Locator) of the website or file. The Browser then requests the **DNS**(DOMAIN NAME SYSTEM) Server.
- **DNS Server** lookup for the address of the **WEB Server**.
- The **DNS Server** responds with the **IP address** of the **WEB Server**.
- The Browser sends over an **HTTP/HTTPS** request to the **WEB Server's IP** (provided by the **DNS server**).
- The Server sends over the necessary files for the website.
- The Browser then renders the files and the website is displayed. This rendering is done with the help of **DOM** (Document Object Model) interpreter, **CSS** interpreter, and **JS Engine** collectively known as the **JIT** or (Just in Time) Compilers.



Client Server Request and Response

Advantages of Client-Server Model

- Centralized system with all data in a single place.
- Cost efficient requires less maintenance cost and Data recovery is possible.
- The capacity of the Client and Servers can be changed separately.

Disadvantages of Client-Server Model

- Clients are prone to viruses, Trojans, and worms if present in the Server or uploaded into the Server.
- Servers are prone to Denial of Service (DOS) attacks.
- Data packets may be spoofed or modified during transmission.
- Phishing or capturing login credentials or other useful information of the user are common and MITM(Man in the Middle) attacks are common.

Conclusion

The client-server architecture consolidates resources on servers for greater control and security, allows for flexible client options, and relies on a robust network for scalability and efficiency. While there are cost implications, the client-server model remains fundamental and has been shaped by trends such as cloud computing.

What is Cloud Storage:

Cloud Storage is a mode of computer data storage in which digital data is stored on servers in off-site locations. The servers are maintained by a third-party provider who is responsible for hosting, managing, and securing data stored on its infrastructure. The provider ensures that data on its servers is always accessible via public or private internet connections.

Cloud Storage enables organizations to store, access, and maintain data so that they do not need to own and operate their own data centers, moving expenses from a capital expenditure model to operational. Cloud Storage is scalable, allowing organizations to expand or reduce their data footprint depending on need.

How does Cloud Storage work

Cloud Storage uses remote servers to save data, such as files, business data, videos, or images. Users upload data to servers via an internet connection, where it is saved on a virtual machine on a physical server. To maintain availability and provide redundancy, cloud providers will often spread data to multiple virtual machines in data centers located across the world. If storage needs increase, the cloud provider will spin up more virtual machines to handle the load. Users can access data in Cloud Storage through an internet connection and software such as web portal, browser, or mobile app via an application programming interface (API).

Cloud Storage is available in four different models:

Public

Public Cloud Storage is a model where an organization stores data in a service provider’s data centers that are also utilized by other companies. Data in public Cloud Storage is spread across multiple regions and is often offered on a subscription or pay-as-you-go basis. Public Cloud Storage is considered to be “elastic” which means that the data stored can be scaled up or down depending on the needs of the organization. Public cloud providers typically make data available from any device such as a smartphone or web portal.

Private

Private Cloud Storage is a model where an organization utilizes its own servers and data centers to store data within their own network. Alternatively, organizations can deal with cloud service providers to provide dedicated servers and private connections that are not shared by any other organization. Private clouds are typically utilized by organizations that require more control over their data and have stringent compliance and security requirements.

Hybrid

A hybrid cloud model is a mix of private and public cloud storage models. A hybrid cloud storage model allows organizations to decide which data it wants to store in which cloud. Sensitive data and data that must meet strict compliance requirements may be stored in a private cloud while less sensitive data is stored in the public cloud. A hybrid cloud storage model typically has a layer of orchestration to integrate between the two clouds. A hybrid cloud offers flexibility and allows organizations to still scale up with the public cloud if need arises.

Multicloud

A multicloud storage model is when an organization sets up more than one cloud model from more than one cloud service provider (public or private). Organizations might choose a multicloud model if one cloud vendor offers certain proprietary apps, an organization requires data to be stored in a specific country, various teams are trained on different clouds, or the organization needs to serve different requirements that are not stated in the servicers' Service Level Agreements. A multicloud model offers organizations flexibility and redundancy.

Advantages of Cloud Storage

Total cost of ownership:
Cloud Storage enables organizations to move from a capital expenditure to an operational expenditure model, allowing them to adjust budgets and resources quickly.

Elasticity:
Cloud Storage is elastic and scalable, meaning that it can be scaled up (more storage added) or down (less storage needed) depending on the organization's needs.

Flexibility:
Cloud Storage offers organizations flexibility on how to store and access data, deploy and budget resources, and architect their IT infrastructure.

Security:
Most cloud providers offer robust security, including physical security at data centers and cutting edge security at the software and application levels. The best cloud providers offer zero trust architecture, identity and access management, and encryption.

Sustainability:
One of the greatest costs when operating on-premises data centers is the overhead of energy consumption. The best cloud providers operate on sustainable energy through renewable resources.

Redundancy:
Redundancy (replicating data on multiple servers in different locations) is an inherent trait in public clouds, allowing organizations to recover from disasters while maintaining business continuity.

Disadvantages of Cloud Storage

Compliance
Certain industries such as finance and healthcare have stringent requirements about how data is stored and accessed. Some public cloud providers offer tools to maintain compliance with applicable rules and regulations.

Latency
Traffic to and from the cloud can be delayed because of network traffic congestion or slow internet connections.

Control
Storing data in public clouds relinquishes some control over access and management of that data, entrusting that the cloud service provider will always be able to make that data available and maintain its systems and security.

Outages
While public cloud providers aim to ensure continuous availability, outages sometimes do occur, making stored data unavailable.

How to use Cloud Storage

Cloud Storage provides several use cases that can benefit individuals and organizations. Whether a person is storing their family budget on a spreadsheet, or a massive organization is saving years of financial data in a highly secure database, Cloud Storage can be used for saving digital data of all kinds for as long as needed.

Backup
Data backup is one of the simplest and most prominent uses of Cloud Storage. Production data can be separated from backup data, creating a gap between the two that protects organizations in the case of a cyber threat such as ransomware. Data backup through Cloud Storage can be as simple as saving files to a digital folder such as Google Drive or using block storage to maintain gigabytes or more of important business data.

Archiving
The ability to archive old data has become an important aspect of Cloud Storage, as organizations move to digitize decades of old records, as well as hold on to records for governance and compliance purposes. Google Cloud offers several tiers of storage for archiving data, including coldline storage and archival storage, that can be accessed whenever an organization needs them.

Disaster recovery
A disaster—natural or otherwise— that wipes out a data center or old physical records needs not be the business-crippling event that it was in the past. Cloud Storage allows for disaster recovery so that organizations can continue with their business, even when times are tough.

Data processing
As Cloud Storage makes digital data immediately available, data becomes much more useful on an ongoing basis. Data processing, such as analyzing data for business intelligence or applying machine learning and artificial intelligence to large datasets, is possible because of Cloud Storage.

Content delivery

With the ability to save copies of media data, such as large audio and video files, on servers dispersed across the globe, media and entertainment companies can serve their audience low-latency, always available content from wherever they reside.
Types of Cloud Storage
Cloud Storage comes in three different types: **object**, **file**, and **block**.

Object
Object storage is a data storage architecture for large stores of unstructured data. It designates each piece of data as an object, keeps it in a separate storehouse, and bundles it with metadata and a unique identifier for easy access and retrieval.

File
File storage organizes data in a hierarchical format of files and folders. File storage is common in personal computing where data is saved as files and those files are organized in folders. File storage makes it easy to locate and retrieve individual data items when they are needed. File storage is most often used in directories and data repositories.

Block
Block storage breaks data into blocks, each with an unique identifier, and then stores those blocks as separate pieces on the server. The cloud network stores those blocks wherever it is most efficient for the system. Block storage is best used for large volumes of data that require low latency such as workloads that require high performance or databases.

Subject Supervisor:
Mr. Adarsh Vardhan Srivastava
Assistant Professor
(Dept. of CSE)
IET, DSMNRU