Basic concept of networking

Internet Protocol **(IP):**

IP stands for Internet Protocol. It is a set of rules and protocols that govern the way data is sent and received over the internet or a computer network. IP provides a unique address for each device connected to a network, allowing them to communicate with each other.

There are two main versions of IP: IPv4 (Internet Protocol version 4) and IPv6 (Internet Protocol version 6). IPv4 is the most widely used version and is based on a 32-bit address scheme. It uses a series of four numbers separated by dots, such as 192.168.0.1, to identify devices on a network. IPv6, on the other hand, uses a 128-bit address scheme and can support a significantly larger number of unique addresses.

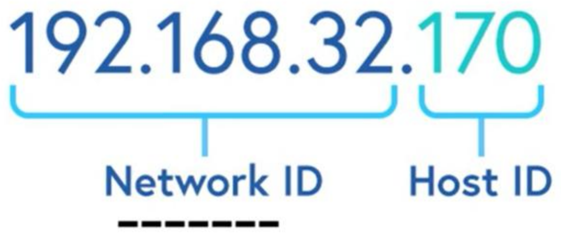
The IP address serves as the identifier for a device on a network. It enables devices to send and receive data packets over the internet or a local network. IP addresses are assigned to devices either statically (manually configured) or dynamically (automatically assigned by a DHCP server).

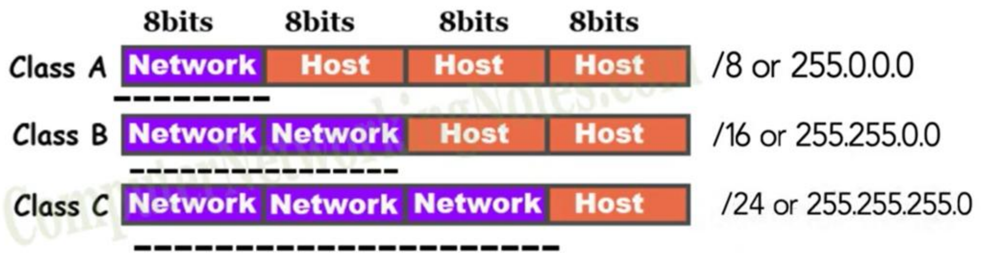
IP addresses are divided into two parts: the network portion and the host portion. The network portion identifies the specific network to which a device is connected, while the host portion identifies the individual device within that network.

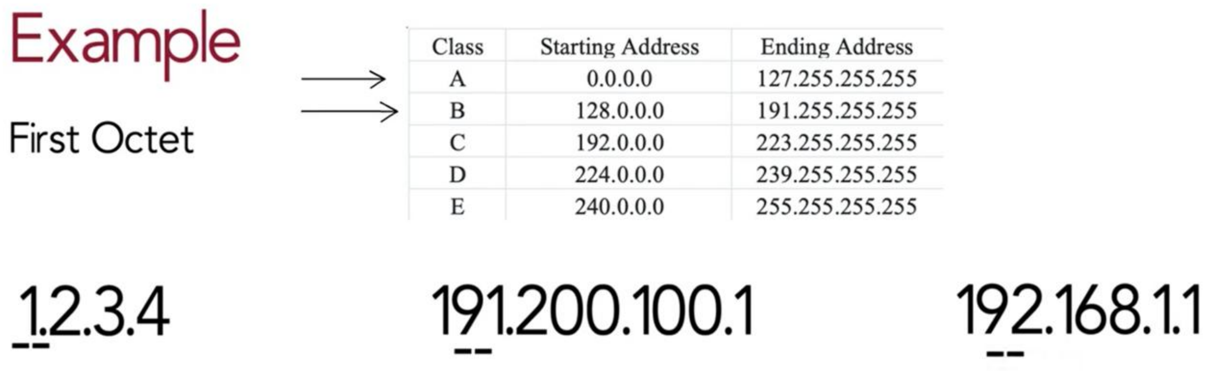
In addition to addressing, IP also defines protocols for packet encapsulation, routing, and fragmentation. It works in conjunction with other protocols, such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol), to provide reliable and efficient communication across networks.

Overall, IP is a crucial component of the internet and computer networks, allowing devices to communicate and exchange data in a standardized and efficient manner.

IP Classes:







Example

First Octet

Class	Starting Address	Ending Address
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.255

1.2.3.4   191.200.100.1   192.168.1.1

IP types:

Public IP: A public IP address is a unique address assigned to a device connected to a network that is accessible over the internet. It is obtained from an Internet Service Provider (ISP) and allows the device to communicate with other devices and services on the internet. Public IP addresses are globally unique and routable across the internet. They are used to identify devices and facilitate communication across different networks.

Private IP: A private IP address is an address used within a private network, such as a local area network (LAN) or a home network. Private IP addresses are assigned to devices by a network administrator and are not directly accessible from the internet. They are used for internal communication within the network and cannot be routed over the internet. Private IP addresses are reserved for use within private networks and are not unique globally. Multiple devices can have the same private IP address, as long as they are on different private networks.

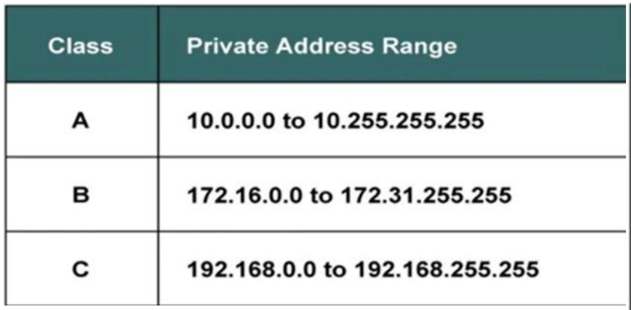
The most commonly used private IP address ranges are defined by the Internet Assigned Numbers Authority (IANA) as follows:

- Class A: 10.0.0.0 to 10.255.255.255 (10.0.0.0/8)
- Class B: 172.16.0.0 to 172.31.255.255 (172.16.0.0/12)
- Class C: 192.168.0.0 to 192.168.255.255 (192.168.0.0/16)

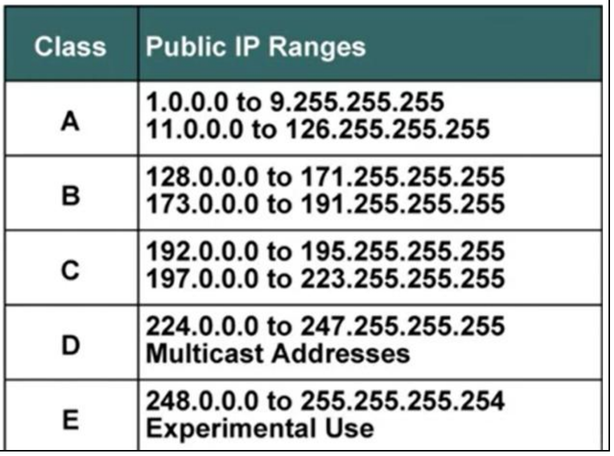
Private IP addresses are used for internal network addressing, allowing devices within the network to communicate with each other without requiring public internet access. They provide a way to conserve the limited number of available public IP addresses by allowing multiple devices to share a single public IP address through a process called network address translation (NAT).

It's important to note that private IP addresses are not directly accessible from the internet. To access resources within a private network from the internet, network address translation (NAT) or port forwarding techniques are commonly used to route traffic between public and private IP addresses.

Private IP	Public IP
LAN Not recognized over internet Assigned by LAN admin Unique only in LAN Free	Internet Recognized by internet Assigned by service provider/ Internet Assigned Number Authority (IANA) Unique globally Prince need



Class	Private Address Range
A	10.0.0.0 to 10.255.255.255
B	172.16.0.0 to 172.31.255.255
C	192.168.0.0 to 192.168.255.255



Class	Public IP Ranges
A	1.0.0.0 to 9.255.255.255 11.0.0.0 to 126.255.255.255
B	128.0.0.0 to 171.255.255.255 173.0.0.0 to 191.255.255.255
C	192.0.0.0 to 195.255.255.255 197.0.0.0 to 223.255.255.255
D	224.0.0.0 to 247.255.255.255 Multicast Addresses
E	248.0.0.0 to 255.255.255.254 Experimental Use

Subnet Mask

A subnet mask is a 32-bit number used in IPv4 networking to divide an IP address into network and host portions. It is used in conjunction with IP addresses to determine the network and host addresses within a given network.

The subnet mask consists of a series of 1s followed by a series of 0s. The 1s in the subnet mask indicate the network bits, while the 0s represent the host bits.

For example, consider the IP address 192.168.0.1 with a subnet mask of 255.255.255.0. In binary form, the IP address is 11000000.10101000.00000000.00000001, and the subnet mask is 11111111.11111111.11111111.00000000. In this case, the first 24 bits (the 1s in the subnet mask) represent the network portion, while the last 8 bits (the 0s in the subnet mask) represent the host portion.

By applying the subnet mask to an IP address, you can determine the network address. In the example above, the network address would be 192.168.0.0. The remaining host bits can be used to identify individual hosts within the network.

Subnet masks allow for efficient IP address allocation and routing by dividing the available IP address space into smaller subnets. They help determine which portion of the IP address represents the network and which portion represents the host. By using appropriate subnet masks, network administrators can create multiple smaller networks within a larger IP address range, enabling efficient utilization of IP addresses and facilitating network management.

IP/Subnetting

IP subnetting is a process used in networking to divide a larger IP network into smaller, more manageable subnetworks or subnets. Subnetting is crucial for efficient IP address allocation and helps control network traffic, reduce broadcast domains, and improve overall network management. Here are the key concepts related to IP subnetting:

1. **IP Addresses**:
    - IP addresses are numeric labels assigned to devices on an IP network to identify and communicate with them. In IPv4, addresses are represented as four octets (32 bits), while IPv6 uses longer hexadecimal addresses.
2. **Subnet Mask**:
    - A subnet mask is a 32-bit value (in IPv4) that is used to divide an IP address into network and host portions. It contains a series of contiguous '1' bits followed by '0' bits. The '1' bits represent the network part, and the '0' bits represent the host part.
3. **Network Address**:
    - The network address is the portion of an IP address that represents the specific subnet. It is obtained by performing a bitwise AND operation between the IP address and the subnet mask.
4. **Host Address**:
    - The host address is the portion of an IP address that identifies an individual device within a subnet. It can vary within the subnet based on the number of available host addresses.
5. **Subnet**:
    - A subnet is a logical division of an IP network created by applying a subnet mask to the IP address range. It contains a range of IP addresses that can be used within a specific network segment.
6. **CIDR Notation**:
    - Classless Inter-Domain Routing (CIDR) notation is a compact way to represent IP addresses and subnet masks. It combines the IP address and the subnet mask, such as "192.168.1.0/24," where "/24" indicates a 24-bit subnet mask.
7. **Subnet Size**:
    - The size of a subnet is determined by the number of host addresses it can accommodate. A subnet with a larger subnet mask (more '0' bits) provides fewer host addresses but more subnets, while a smaller subnet mask (fewer '0' bits) offers more host addresses but fewer subnets.
8. **Private and Public Addresses**:

- Subnetting is used to manage both private IP addresses (used within an organization) and public IP addresses (routed on the internet). Organizations often use private IP addresses internally and employ Network Address Translation (NAT) to map them to a smaller set of public addresses.

LAN/WAN

LAN (Local Area Network) and WAN (Wide Area Network) are two fundamental types of computer networks that differ in terms of their size, coverage, and the scope of the network. Here's an overview of each:

LAN (Local Area Network):

1. **Size and Coverage**:
    - LANs are relatively small networks that typically cover a limited geographic area, such as a single building, office, home, or campus.
    - The size of a LAN can vary, but it is generally confined to a few kilometers at most.
2. **Ownership**:
    - LANs are often privately owned and managed by an organization, individual, or institution for their specific needs.
3. **Connectivity**:
    - Devices within a LAN are connected using high-speed technologies like Ethernet, Wi-Fi, or other local networking protocols.
    - LANs can be wired (using Ethernet cables) or wireless (using Wi-Fi).
4. **Purpose**:
    - LANs are designed for local or internal use, providing connectivity between devices like computers, printers, servers, and other networked devices within a confined area.
5. **Examples**:
    - Common examples of LANs include the network within a home, a corporate office, a university campus, or a small business.
6. **Speed and Bandwidth**:
    - LANs often provide high data transfer speeds and ample bandwidth for local data exchange.
7. **Topologies**:
    - LANs can use various network topologies, such as star, bus, or ring, depending on the requirements of the network.

WAN (Wide Area Network):

1. **Size and Coverage**:
    - WANs are extensive networks that cover a wide geographic area, often spanning across cities, regions, countries, or even continents.
    - The size of a WAN can be immense, and it can span thousands of kilometers.
2. **Ownership**:
    - WANs are usually made up of various interconnected LANs and are often operated and maintained by telecommunications companies or service providers.
3. **Connectivity**:
    - Devices within a WAN are connected over long-distance communication technologies, such as leased lines, fiber optics, or satellite links.
    - The internet is the largest and most well-known example of a WAN, connecting networks worldwide.
4. **Purpose**:
    - WANs are designed to connect geographically dispersed LANs, allowing for long-distance communication and data transfer between different locations.
5. **Examples**:
    - The internet is the most extensive WAN, connecting people and organizations worldwide. Other examples of WANs include private networks that interconnect branch offices of a multinational corporation.
6. **Speed and Bandwidth**:
    - WANs can have varying data transfer speeds and bandwidth, depending on the underlying infrastructure. Speeds may be slower than LANs due to the long-distance nature of WANs.
7. **Topologies**:
    - WANs typically use point-to-point or mesh topologies for efficient long-distance communication.

In summary, LANs are local networks that cover a small, localized area, while WANs are large-scale networks that connect LANs across vast geographic distances. LANs are often privately owned, provide high-speed connections, and are used for local data exchange, while WANs, like the internet, connect LANs globally and use long-distance communication technologies.

SSH Server & SSH Client

SSH (Secure Shell) is a network protocol and a cryptographic tool used for secure remote access to devices and servers over a potentially unsecured network, such as the internet. It provides a secure channel for authentication and data communication. In SSH, there are two main components: the SSH server and the SSH client.

SSH Server:

1. **Definition**:
    - An SSH server is a software application or service that runs on a remote host, allowing users to establish secure connections to that host over a network. It listens for incoming SSH client connections on a specified port (commonly port 22).
2. **Functions**:
    - Authentication: The SSH server authenticates remote users based on their credentials (usually a username and password, or more securely, using key pairs).
    - Secure Communication: Once the client is authenticated, the server establishes an encrypted communication channel with the client for secure data exchange.
    - Execution of Commands: SSH servers can execute remote commands and provide terminal access to the host, enabling users to manage the remote system as if they were physically present.
3. **Examples**:
    - Common SSH server software includes OpenSSH (open-source and widely used), Microsoft Windows OpenSSH (for Windows servers), and various commercial SSH server solutions.

SSH Client:

1. **Definition**:
    - An SSH client is a software application or command-line tool used by a user to connect to a remote SSH server. It allows users to initiate secure connections to remote hosts and interact with them securely.
2. **Functions**:
    - Authentication: The SSH client initiates the connection and provides authentication information (username and password, or a private key) to the server.
    - Secure Communication: Once connected, the client and server establish an encrypted communication channel for secure data exchange.
    - Remote Access: Users can use the SSH client to access the remote host, run commands, transfer files, and perform various tasks securely.
3. **Examples**:

        - Common SSH client software includes OpenSSH (open-source and widely used, available on most Unix-like systems), PuTTY (a popular Windows SSH client), and various other SSH client applications for different platforms.

The SSH protocol and its client-server architecture are widely used for secure remote administration, file transfer, and tunneling encrypted network traffic. SSH is an essential tool for system administrators, developers, and anyone who needs to manage or access remote systems securely. It is known for its robust security features, including encryption, authentication, and strong access control.

Password Based Authentication:

There are several ways to connect to an SSH server, depending on your operating system and the tools available to you. Here are some common methods to connect to an SSH server:

1. Command Line (Terminal or Command Prompt):

- On Unix-like systems (Linux, macOS, etc.) and Windows (with the help of third-party tools), you can use the command line to connect to an SSH server using the **ssh** command. Here's the basic syntax:

```
ssh username@hostname
```

    - **username**: Your username on the remote server.
    - **hostname**: The IP address or domain name of the remote server.

You may also specify a custom port, if necessary:

```
ssh -p port_number username@hostname
```

2. SSH Clients:

- There are graphical SSH client applications available for various operating systems. These tools provide a user-friendly interface for connecting to SSH servers and managing your connections. Some popular SSH client applications include:
    - PuTTY (Windows)
    - WinSCP (Windows)
    - OpenSSH (Unix-like systems, includes the **ssh** command)
    - Bitvise SSH Client (Windows)
    - Cyberduck (macOS and Windows)

3. Integrated Development Environments (IDEs):

- Many integrated development environments, such as Visual Studio Code, offer extensions or built-in features for SSH connectivity. These can be convenient for developers who want to work on remote servers from within their development environment.

4. Web-Based SSH Clients:

- Some web-based SSH clients allow you to connect to SSH servers directly from your web browser. They are especially useful when you're on a computer where you can't install SSH client software. Examples include Shellinabox, GateOne, and WebSSH.

5. Mobile SSH Apps:

- There are SSH client apps available for mobile devices (smartphones and tablets). These apps allow you to connect to SSH servers from your mobile device and manage remote systems while on the go.

When connecting to an SSH server, you'll need the following information:

- Hostname or IP address of the remote server.
- Your username on the remote server.
- (Optionally) The port number (default is 22 for SSH).
- (Optionally) The path to your private key (if using key-based authentication).
- (Optionally) Any other relevant authentication information, such as a passphrase for your private key.

Additionally, make sure that the SSH server is running and that your username is valid on the remote system. You'll be prompted for your password or passphrase if you're using password-based or key-based authentication, respectively.

Key Based Authentication (Private key & public Key):

Key-based authentication, also known as public key authentication, is a more secure and convenient method for authenticating users to a server or system compared to traditional password-based authentication. It uses a pair of cryptographic keys: a private key and a public key. Here's how key-based authentication works:

Private Key:

- The private key is kept on the client (the user's computer). It is a closely guarded secret and should never be shared with anyone. It is protected by a passphrase (a password), making it even more secure.

Public Key:

- The public key, as the name suggests, is made public. It is stored on the server you want to access. This key is used to verify the authenticity of the user's private key.

The key-based authentication process typically works as follows:

1. **Key Pair Generation**:
    - The user generates a key pair on their client machine. This is typically done using tools like **ssh-keygen** on Unix-like systems or SSH key generation software on Windows.

2. **Copying the Public Key to the Server**:
   - The user copies the public key (usually found in a file named **id_rsa.pub** or **id_dsa.pub**) to the server they want to access. This is usually achieved by adding the public key to the **~/.ssh/authorized_keys** file on the server.
3. **Connecting to the Server**:
   - When the user tries to SSH into the server, the server checks if the user's public key matches any of the keys in the **authorized_keys** file.
4. **Key Pair Authentication**:
   - If the public key on the server matches the private key on the client, and the client provides the correct passphrase (if one is set for the private key), the user is granted access.

Key-based authentication offers several advantages:

- **Strong Security**: Private keys are difficult to compromise, making key-based authentication more secure than password-based authentication.
- **No Need for Passwords**: Once key-based authentication is set up, users don't need to remember or enter passwords each time they connect to the server.
- **Automation**: Key-based authentication is well-suited for automated tasks and scripts, as it doesn't require user interaction for password entry.
- **Multi-Factor Authentication**: Private keys can be used in combination with passwords or other authentication methods to create multi-factor authentication for enhanced security.
- **Logging**: Key-based authentication can be used to track user access more effectively by linking each user to a specific key.

Key-based authentication is a common practice in secure remote access to servers, especially in the context of SSH (Secure Shell) connections. It is a fundamental security measure in many organizations and is widely used to protect against unauthorized access.

Concept of Storage

Here are some of the most common types of storage:

- **Block base storage**: Block-based storage is a type of data storage where data is organized and accessed in fixed-size blocks or chunks. Each block typically has a unique address and can be read from or written to independently. Block-based storage is commonly used in various storage systems, including hard disk drives (HDDs), solid-state drives (SSDs), and network-attached storage (NAS) devices. we can partition and install OS.
- **File base storage**: File-based storage is a method of organizing and managing data in a hierarchical structure using files and directories. In file-based storage, data is grouped into files,

and these files are organized within directories or folders. This approach is the foundation of most traditional file systems and is commonly used on computers and networked storage systems. We can only store any file ex image, documents, media file etc.

- **Network-Attached Storage (NAS):** NAS devices are specialized storage systems connected to a network, providing shared storage accessible by multiple users or device

Concept of firewall

A firewall is a network security device or software that serves as a barrier between a trusted network and an untrusted network, typically the internet. Its primary purpose is to monitor and control incoming and outgoing network traffic, allowing or blocking data packets based on a set of predetermined security rules. Firewalls play a crucial role in protecting networked devices, systems, and data from unauthorized access, cyber threats, and malicious activities.
Firewalls are a critical component of network security and are used to establish security perimeters around networks, data centers, and individual devices. They help protect against various threats, including unauthorized access, malware, viruses, denial-of-service attacks, and other network-based security risks.

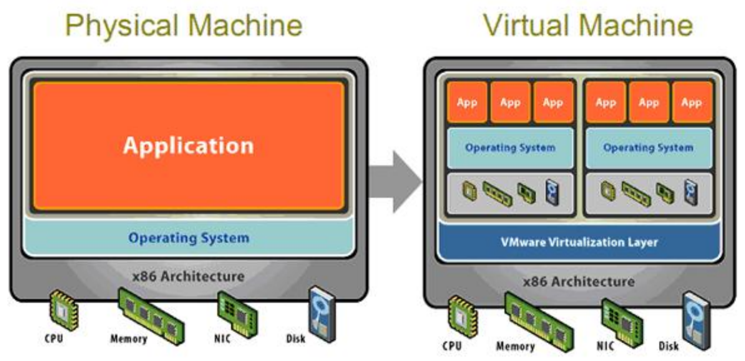
Concept of Virtualization

Virtualization is a technology that allows multiple operating systems or applications to run on a single physical server or computer. It allows for the creation of virtual versions of hardware, operating systems, storage devices, and other resources, which can be shared among multiple users or applications.

With virtualization, a physical server or computer can be partitioned into multiple virtual machines (VMs), each running its own operating system and applications. Each VM can be isolated from the others, and can have its own resources, such as CPU, memory, storage, and network connectivity.

Virtualization offers several benefits, including increased efficiency, flexibility, and cost savings. It allows multiple applications or operating systems to run on the same physical hardware, reducing the need for additional hardware and lowering costs. It also enables more efficient use of resources, as unused resources can be dynamically allocated to where they are needed most.

Virtualization is widely used in data centers, cloud computing environments, and desktop computing. Some of the most popular virtualization technologies include VMware, Hyper-V, KVM, and VirtualBox, among others.

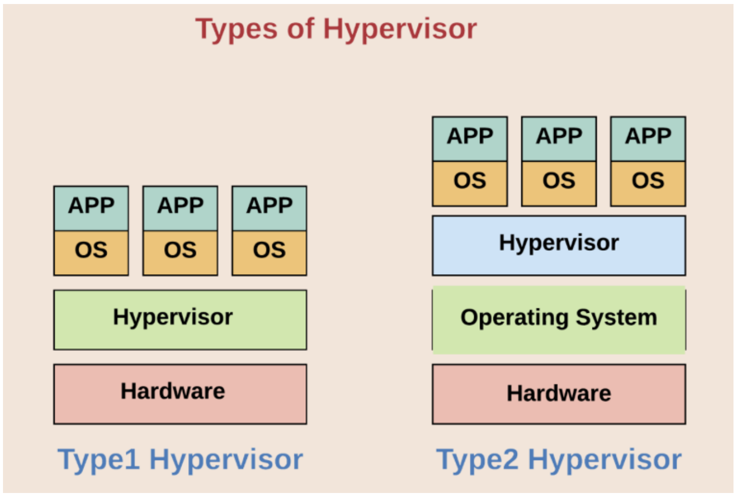


Two types of hypervisors:

Type 1 Hypervisor (Bare Metal Hypervisor): Also known as a "bare metal" or "native" hypervisor, a Type 1 hypervisor runs directly on the host server's hardware without the need for an underlying operating system. It has direct access to the hardware resources and manages the allocation of those resources to guest virtual machines. Type 1 hypervisors are typically used in server virtualization scenarios. Examples of Type 1 hypervisors include VMware ESXi, Microsoft Hyper-V, and Citrix XenServer.

Type 2 Hypervisor (Hosted Hypervisor): A Type 2 hypervisor, also known as a "hosted" hypervisor, runs on top of a host operating system. It requires a pre-installed operating system and then operates as an application within that OS. The Type 2 hypervisor provides a virtualization layer between the guest virtual machines and the host operating system. These hypervisors are commonly used for desktop virtualization and are suitable for personal or development use. Examples of Type 2 hypervisors include Oracle VirtualBox, VMware Workstation, and Microsoft Virtual PC.

Both Type 1 and Type 2 hypervisors offer similar virtualization capabilities, but they differ in terms of their architectural approach and use cases. Type 1 hypervisors are generally considered more efficient and secure because they have direct access to the hardware, while Type 2 hypervisors offer more flexibility and are easier to set up and use on a desktop or laptop environment.



Benefit of Virtualization

Virtualization is a technology that allows you to create multiple virtual instances of computer hardware, operating systems, storage devices, or network resources on a single physical server or system. This technology has several benefits, which is why it is widely used in data centers, cloud computing, and other IT environments. Some of the key benefits of virtualization include:

1. **Resource Efficiency**:

    - **Server Consolidation**: Virtualization allows you to run multiple virtual machines (VMs) on a single physical server, optimizing hardware utilization and reducing the number of physical servers required.

    - **Resource Allocation**: Resources such as CPU, memory, and storage can be dynamically allocated to VMs as needed, ensuring efficient use of hardware.

2. **Cost Savings**:
    - **Reduced Hardware Costs**: By consolidating servers and utilizing resources more effectively, organizations can reduce their hardware procurement and maintenance costs.
    - **Energy Savings**: Fewer physical servers mean lower energy consumption and reduced cooling costs.
3. **Improved Scalability**:
    - **Elasticity**: Virtualization allows for easy scaling of resources up or down based on changing workloads or demands. New VMs can be provisioned quickly.
    - **Resource Pooling**: Resources can be pooled and shared among VMs, providing flexibility and scalability.
4. **Isolation and Security**:
    - **Isolation**: VMs are isolated from each other, which helps prevent one VM from impacting the performance or security of others.
    - **Snapshot and Rollback**: VM snapshots enable the creation of backup points for VMs, simplifying recovery in case of issues or security breaches.
5. **Faster Provisioning and Disaster Recovery**:
    - **Rapid Deployment**: VMs can be provisioned much faster than physical servers, reducing time-to-market for new applications and services.
    - **Disaster Recovery**: VM snapshots and the ability to migrate VMs between physical servers make disaster recovery and backup processes more efficient.
6. **Simplified Management**:
    - **Centralized Control**: Virtualization platforms often provide centralized management tools for monitoring and managing VMs.
    - **Template-Based Deployment**: VM templates simplify the deployment of new VMs with predefined configurations.
7. **Testing and Development**:
    - **Testing Environments**: Virtualization allows for the creation of isolated testing environments, making it easier to test and develop new software and configurations.
    - **Development Snapshots**: Developers can take snapshots of VMs before making changes, enabling them to roll back to a previous state if needed.
8. **Legacy System Support**:
    - **Legacy Application Compatibility**: Virtualization can host older operating systems and applications that may not be compatible with newer hardware.

9. **Multi-tenancy**:
    - **Cloud and Service Providers**: Virtualization is a fundamental technology for cloud computing, enabling service providers to offer multi-tenant environments with resource isolation.
10. **High Availability**:
    - **Load Balancing**: VMs can be load-balanced across multiple physical servers, providing redundancy and high availability for applications and services.

Virtualization has revolutionized IT infrastructure and data center management, offering increased flexibility, efficiency, and cost savings. It is a foundational technology in modern computing, enabling the delivery of services and applications in a more agile and scalable manner.

What is cloud computing?

Cloud computing is a technology that allows users to access computing resources, such as servers, storage, databases, applications, and services, over the Internet or other network connections. Instead of having to maintain their own physical infrastructure, users can rent or use the resources they need on a pay-as-you-go basis, from a cloud service provider.

The cloud service provider is responsible for managing the underlying infrastructure, ensuring availability, reliability, and security of the resources. Users can access these resources using a variety of devices, such as desktops, laptops, smartphones, and tablets, from anywhere in the world.

Cloud computing offers several benefits, including:

Scalability: Cloud resources can be easily scaled up or down as needed, depending on the demand for resources.

Flexibility: Users can choose from a wide range of services and resources to meet their specific needs, without having to worry about the underlying infrastructure.

Cost-effectiveness: Cloud computing can be a more cost-effective option than maintaining your own physical infrastructure, as you only pay for the resources you use.

Reliability: Cloud providers typically offer high levels of availability and redundancy, ensuring that your data and applications are always available.

Security: Cloud providers offer a range of security measures to protect against unauthorized access, data breaches, and other security threats.

Cloud computing has become increasingly popular in recent years, with many businesses and organizations adopting it to improve their agility, reduce costs, and increase flexibility. Some of the most popular cloud service providers include Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform, among others.

Example for virtualization and cloud when deploying a webserver.

virtualization:
OS >> ISO >> Install OS >> Hostname >> Network >> Package IIS/Apache >> start >> Firewall >> users.

cloud: Take a cloud image for OS and do a few `steps to run the webserver for users.
OS >>Coolud-Image>>httpd>>users

Aws (Amazon web server)

AWS stands for Amazon Web Services, which is a cloud computing platform provided by Amazon. It offers a wide range of cloud computing services, including computing power, storage, and databases, as well as a variety of application services and tools for managing and deploying applications on the cloud.

AWS provides a highly scalable, reliable, and secure infrastructure that enables businesses of all sizes to build and deploy their applications in the cloud. It allows users to pay only for the resources they use, and to easily scale up or down their usage as needed.

Some of the most popular services provided by AWS include Amazon EC2 (Elastic Compute Cloud), Amazon S3 (Simple Storage Service), Amazon RDS (Relational Database Service), Amazon SQS (Simple Queue Service), and Amazon Lambda, among others. AWS also offers a wide range of tools and services for data analytics, machine learning, and Internet of Things (IoT) applications.

Why we use AWS instead of virtualization:

AWS provides a cloud computing platform that includes virtualization as one of its core technologies. So, in a sense, when you use AWS, you are using virtualization. However, AWS offers many additional benefits beyond basic virtualization that make it a compelling choice for businesses and organizations.

One of the main advantages of using AWS is that it provides a highly scalable, flexible, and cost-effective infrastructure. AWS offers a wide range of services that allow you to easily deploy and manage your applications and infrastructure on the cloud, without having to worry about the underlying hardware or infrastructure. You can easily scale your resources up or down as needed, and only pay for what you use.

AWS also provides a highly secure and reliable infrastructure, with multiple layers of security and redundancy to protect against data loss, downtime, and other potential issues. AWS has achieved numerous compliance certifications, including PCI DSS, HIPAA, and SOC 2, which make it a trusted choice for businesses that need to comply with regulatory requirements.

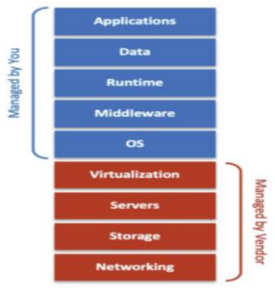
Another advantage of using AWS is that it provides a rich ecosystem of tools and services that can help you build, deploy, and manage your applications more easily and efficiently. AWS offers a wide range of services for data analytics, machine learning, Internet of Things (IoT), and other areas, which can help you innovate and stay ahead of the competition.

Overall, while virtualization is an important technology for deploying and managing applications, AWS offers many additional benefits that make it a compelling choice for businesses and organizations that want to leverage the power of the cloud.

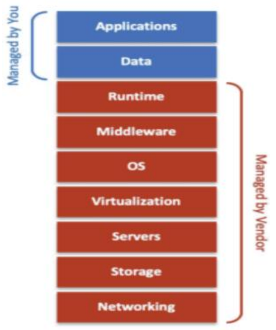
Types of clouds based on services:

There are three main types of cloud computing services based on service models, which are:

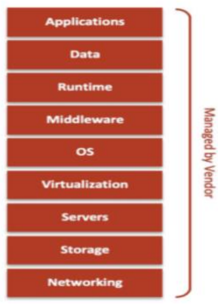
Infrastructure as a Service (IaaS): This is a type of cloud computing service that provides users with virtualized computing resources, such as servers, storage, and networking. Users can deploy and run their own applications and operating systems on these virtual machines and are responsible for managing their own software and applications. In an office Iaas service use system admin.
Examples of IaaS providers include AWS Ec2, Google Compute Engine, Digital Ocean and Microsoft Azure.



Platform as a Service (PaaS): This is a type of cloud computing service that provides users with a platform for developing, deploying, and managing their own applications. PaaS providers typically offer pre-configured software and development tools that allow users to quickly build and deploy applications, without having to worry about the underlying infrastructure. For example if you want in launch a php application for that purpose all dependencies will installed and provided by provider which is PaaS.
Examples of PaaS providers include Aws Elastic Beanstalk, Heroku, Google App Engine, and Microsoft Azure.



Software as a Service (SaaS): This is a type of cloud computing service that provides users with access to software applications over the Internet. SaaS providers typically host and manage the applications, and users can access them through a web browser or other client applications.
Examples of SaaS providers include Microsoft Office 365, and Dropbox.



Overall, each type of cloud computing service offers different levels of abstraction and control over the underlying infrastructure, allowing users to choose the service that best meets their needs.

Types of cloud based on deployment:

There are three main types of cloud computing services based on deployment models, which are:

Public Cloud: This is a type of cloud computing service that is provided by third-party cloud service providers over the internet. Public cloud providers make their services available to anyone who wants to use them, and users typically pay for the resources they use on a pay-as-you-go basis. Public clouds are generally scalable, flexible, and cost-effective, but may have security and privacy concerns.

Examples of public cloud providers include Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform.

Private Cloud: This is a type of cloud computing service that is operated and managed by a single organization, either on-premises or off-premises. Private clouds offer greater control over the infrastructure and data and can be more secure and reliable than public clouds, but require more

resources and expertise to set up and manage.
Examples of private cloud solutions include OpenStack, VMware, and Microsoft Azure Stack.

Hybrid Cloud: This is a type of cloud computing service that combines elements of both public and private clouds, allowing organizations to take advantage of the benefits of both. Hybrid clouds allow users to run their applications and workloads in both public and private environments, and can provide greater flexibility, scalability, and cost-effectiveness. For example, in balk Friday we have more user hit so our database server is in our private cloud and we use 10 AWS webserver this type of scenario is hybrid cloud.
Examples of hybrid cloud solutions include AWS Outposts, Azure Arc, and Google Anthos.

Overall, each type of cloud computing deployment model offers different levels of control, security, and scalability, allowing organizations to choose the deployment model that best meets their needs.

AWS Outposts:

AWS Outposts is a service provided by Amazon Web Services (AWS) that extends AWS's infrastructure and services to customer on-premises data centers or co-location facilities. It allows customers to run AWS services locally, ensuring low-latency access to those services and data processing, which is particularly important for applications with low-latency requirements or data residency and compliance needs.

Key features of AWS Outposts include:

1. **On-Premises Extension of AWS**: AWS Outposts brings AWS infrastructure and services to your on-premises data center, allowing you to use the same APIs, tools, and hardware that you would use in the AWS cloud.
2. **Fully Managed**: AWS takes care of installation, management, and maintenance of the Outpost infrastructure, so you can focus on building and running your applications.
3. **Local Data Processing**: Running AWS services locally on an Outpost enables low-latency access to data, making it suitable for applications that require real-time or low-latency processing.
4. **Hybrid Cloud**: AWS Outposts facilitates a hybrid cloud approach where you can seamlessly connect your on-premises environment to the AWS cloud, creating a unified infrastructure.
5. **Multiple Configurations**: AWS Outposts comes in two configurations: VMware Cloud on AWS Outposts and AWS Outposts Native. The former is a fully managed VMware environment on AWS hardware, and the latter runs AWS native services on Outposts hardware.
6. **Security and Compliance**: With Outposts, you can maintain data sovereignty and meet compliance requirements by keeping sensitive data within your on-premises environment while still benefiting from AWS's security practices.
7. **Local Storage**: You can use local storage on the Outposts to store and process data locally, reducing the need to transfer data back and forth between your on-premises location and the AWS cloud.

8. **High Availability**: Outposts are designed for high availability and are equipped with redundant power and networking to minimize downtime.

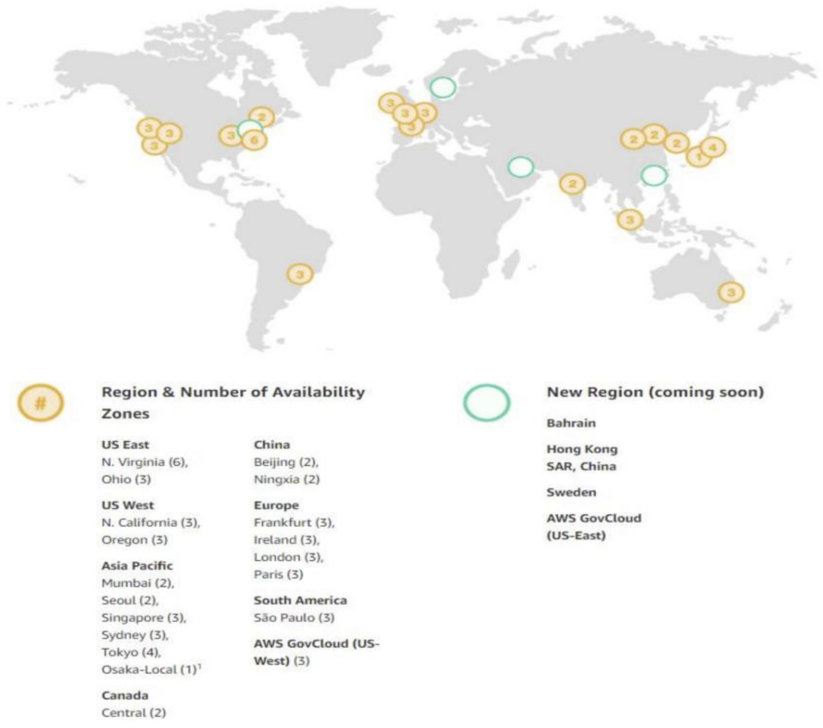
AWS Outposts is especially useful for applications that require low latency, like IoT applications, or in industries with data residency requirements, such as healthcare and finance. It offers a way to bridge the gap between on-premises and cloud environments, allowing you to leverage the benefits of both while maintaining data control and compliance.

Aws Global Infrastructure

Amazon Web Services (AWS) has a global infrastructure that consists of multiple regions and availability zones around the world.

As of my knowledge cutoff date, which is September 2021, AWS has 25 regions globally, including:

US East (N. Virginia)
US East (Ohio)
US West (N. California)
US West (Oregon)
Canada (Central)
South America (São Paulo)
Europe (Ireland)
Europe (London)
Europe (Paris)
Europe (Frankfurt)
Europe (Stockholm)
Middle East (Bahrain)
Asia Pacific (Tokyo)
Asia Pacific (Seoul)
Asia Pacific (Mumbai)
Asia Pacific (Singapore)
Asia Pacific (Sydney)
Asia Pacific (Hong Kong)
China (Beijing)
China (Ningxia)
AWS GovCloud (US-West)
AWS GovCloud (US-East)
Africa (Cape Town)
Europe (Milan)
Europe (Zurich)



Each region consists of multiple availability zones (AZs), which are physically separate data centers within a region.

The AWS global infrastructure enables customers to build highly available, fault-tolerant, and scalable applications that can run across multiple regions and availability zones.

Availability Zone (AZ)

An Availability Zone (AZ) in the context of Amazon Web Services (AWS) is a distinct data center within a specific AWS region. Each region is made up of multiple availability zones that are isolated from each other in terms of power, cooling, and network connectivity.

The purpose of having multiple availability zones within a region is to provide redundancy and fault tolerance. If one availability zone experiences an issue or outage, the other availability zones within the same region can continue to operate independently, ensuring high availability and resilience for AWS services and applications running in that region.

AWS customers can deploy their resources, such as virtual machines (EC2 instances), databases (RDS), and storage (S3), across multiple availability zones to achieve high availability and fault tolerance. This can be achieved through various techniques such as load balancing, replication, and automated failover.

It's important to note that while availability zones are designed to provide high availability within a region, they are not intended for cross-region redundancy. If you require data or application redundancy across different AWS regions, you would need to design and implement a multi-region architecture using AWS services such as AWS Global Accelerator, Amazon Route 53, or manually replicate data across regions using services like Amazon S3 cross-region replication or AWS Database Migration Service.

AWS Edge location

An AWS Edge Location is a site deployed by Amazon Web Services (AWS) to bring content closer to the end-users. These edge locations are part of AWS's Content Delivery Network (CDN) service, Amazon CloudFront, and are spread across the world to improve content delivery and reduce latency.

When a user requests content from a website or web application that is hosted on AWS, the request is routed to the nearest edge location, which then caches the content. This cached content is then served to the user from the edge location, reducing the time it takes to load the content and improving the user experience.

AWS edge locations are strategically located in cities and regions worldwide, providing low latency access to content to users from different geographical locations. These edge locations are also used to distribute large files, videos, and software updates, reducing the load on the origin servers and improving the download speeds for end-users.

In addition to caching content, AWS edge locations can also execute certain AWS services, such as AWS Lambda functions, providing a low latency computing platform for certain types of applications. This allows developers to build and deploy applications closer to their end-users and reduce the response time of their applications.

Overall, AWS Edge Locations are a critical component of the AWS infrastructure, enabling businesses to deliver content faster and more efficiently to their customers, regardless of their location.

AWS Elastic Compute Cloud (EC2)

AWS EC2 (Elastic Compute Cloud) is a web service provided by Amazon Web Services (AWS) that enables businesses and individuals to run applications on virtual machines (VMs) hosted in the cloud. EC2 provides scalable computing capacity in the cloud, which means users can quickly and easily provision and de-provision computing resources as their needs change.

With EC2, users can launch virtual servers called "instances" in a matter of minutes, and pay only for the capacity they actually use. EC2 offers a variety of instance types to suit different workloads, from general-purpose instances to compute-optimized, memory-optimized, and storage-optimized instances. EC2 also provides a number of tools for managing instances, including the EC2 console, the AWS CLI, and the EC2 API.

Overall, AWS EC2 is a flexible and powerful tool for running applications in the cloud, providing users with access to scalable compute resources without the need for upfront investment in hardware or infrastructure.

For connecting to ec2 instance

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is rajiv1.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

```
chmod 400 rajiv1.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-3-20-239-21.us-east-2.compute.amazonaws.com
```

Example:

```
ssh -i "private-key.pem" ec2-user@ec2-3-20-239-21.us-east-2.compute.amazonaws.com
```

Check any feature from AWS such as free tire or ec2 need to type your browser as follows
https://aws.amazon.com/free or https://aws.amazon.com/ec2