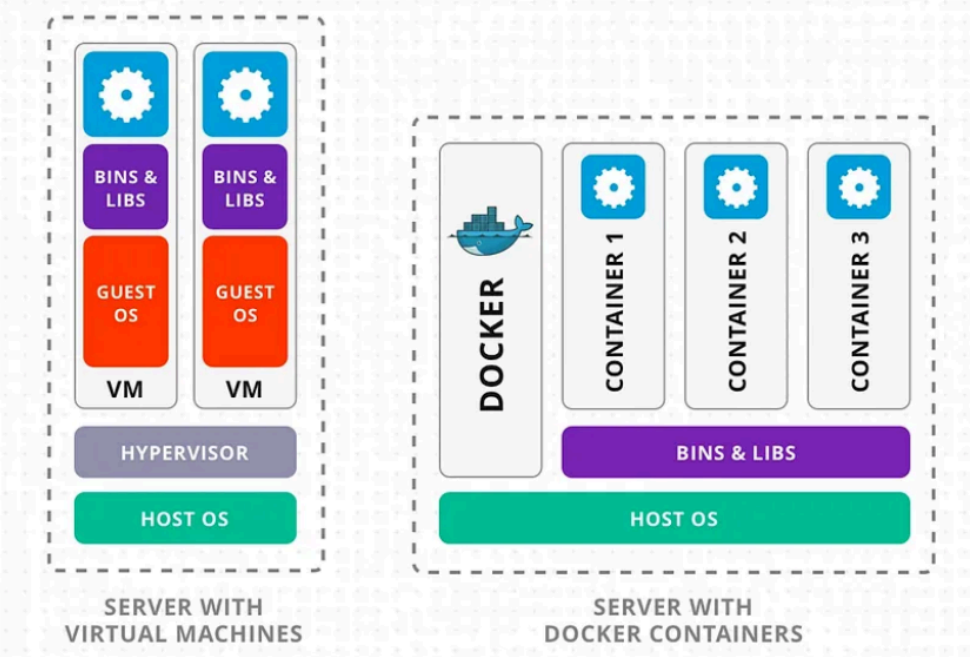
Networking

Virtual Machine vs Docker

Virtual machines (VMs) and Docker containers are both technologies used for virtualization and application deployment, but they have significant differences in terms of architecture and use cases.

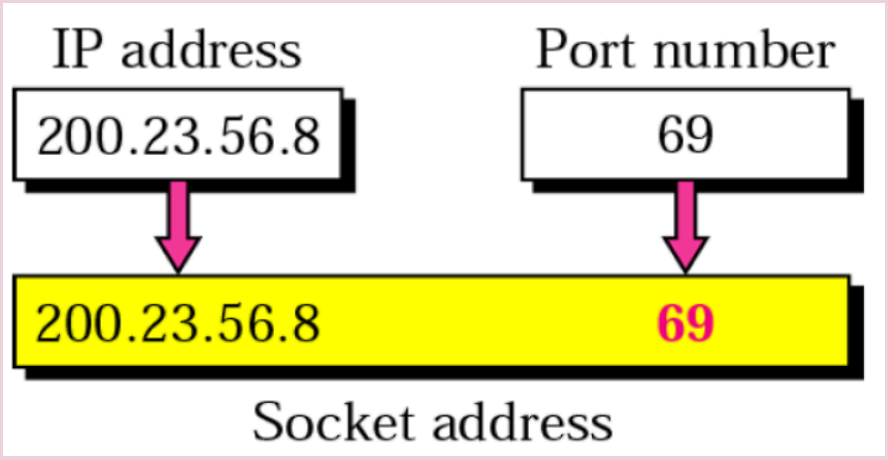


Virtual Machine	DOCKER
1. VMs emulate an entire physical computer, including the operating system (OS), on top of a hypervisor. Each VM runs its own OS and has its own resources, such as memory, storage, and CPU.	1. Containers share the host OS kernel and isolate the application processes. They package the application and its dependencies, along with a lightweight runtime, in a single container.
(VMs): VMs are resource-intensive because they include a full OS stack for each instance. They require more memory and storage compared to containers.	Containers are lightweight and share the host OS kernel, resulting in lower resource overhead. They start faster and use fewer resources compared to VMs.
VMs provide stronger isolation because each VM has its own OS, which means	Containers share the host OS kernel, offering a lighter form of isolation. While

that issues within one VM are less likely to affect others.	containers are isolated from each other, they are not as isolated as VMs.
VMs are less portable because they include the entire OS. Moving VMs between different environments can be more challenging.	Containers are highly portable. They encapsulate the application and its dependencies, making it easy to run the same container across different environments.
VMs can be slower to scale because they involve booting up an entire virtualized OS.	Containers are designed for quick scaling. They start and stop rapidly, making them well-suited for dynamic scaling in cloud and microservices architectures.
VMs are often used for running multiple applications with different OS requirements on a single physical server. They are common in traditional virtualization scenarios.	Containers are popular for microservices architectures, continuous integration/continuous deployment (CI/CD), and environments where rapid deployment and scalability are essential.

IP vs Port

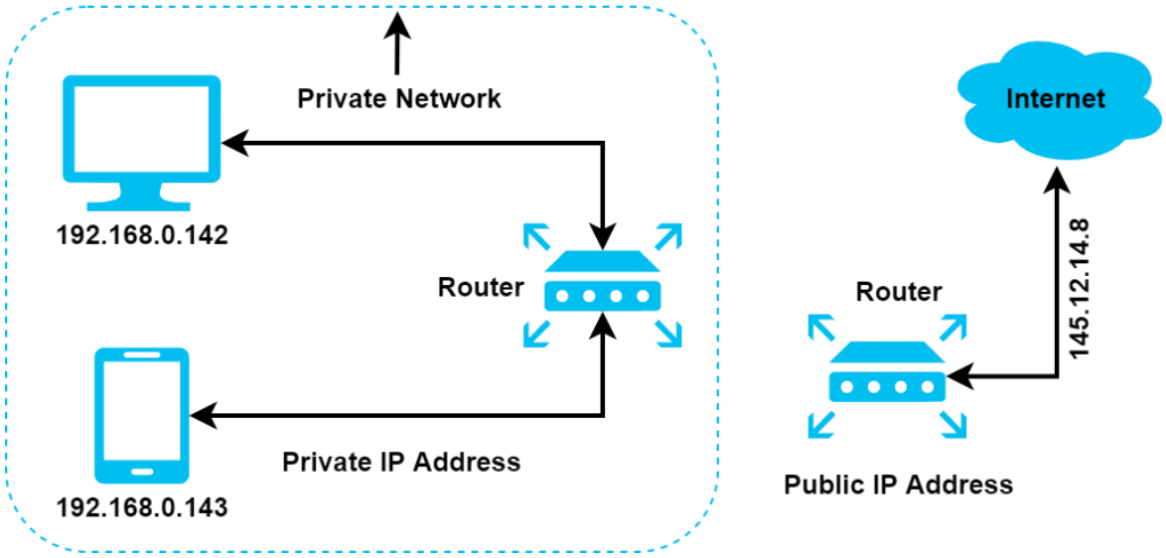
An IP address identifies a machine in an IP network and is used to determine the destination of a data packet. Port numbers identify a particular application or service on a system.



Uniqueness:

IP: Each device on a network has a unique IP address, allowing for the identification and communication with that specific device. Port: Ports are unique within the context of a specific device. Different devices can use the same port numbers without conflict because the combination of IP address and port creates a unique endpoint.

Public IP address routes over the Internet and provides remote access to the computer. The private IP addresses can't be routed over the Internet and don't allow traffic from it. These are uses as reserve IP addresses and only work within a local network:



The primary purpose of a port is to help a computer understand incoming traffic and send them. Suppose Sam wants to send an MP3 file to Mike. Sam uses the File Transfer Protocol (FTP) to transfer the MP3 file to Mike. Now let's assume that after receiving Sam's file, Mike's computer doesn't identify the MP3 file and sends it to an email application.

In such a case, the email application won't be able to open the MP3 file. But Sam uses a port allotted to FTP while transferring the MP3 file. Hence Mike's computer will identify the file using the port number used here and send it to the appropriate process. Also, parallel Mike can load HTTP webpages on his computer that uses port number .

MAC Address

Use: A MAC address (Media Access Control address) is like a unique ID for devices on a network, such as your computer, smartphone, or any device with network capabilities. It's used to identify and communicate with other devices on the same network.

Working: Unique Identifier: Just like how your home has a unique address to receive mail, devices on a network have unique MAC addresses assigned to them. This address is usually hard-coded into the device's network hardware, and it's unique worldwide.

Switching and Routing: Network devices, like switches and routers, use MAC addresses to figure out where to send data. It's like the postal service routing mail based on addresses.

Internal vs External Ips

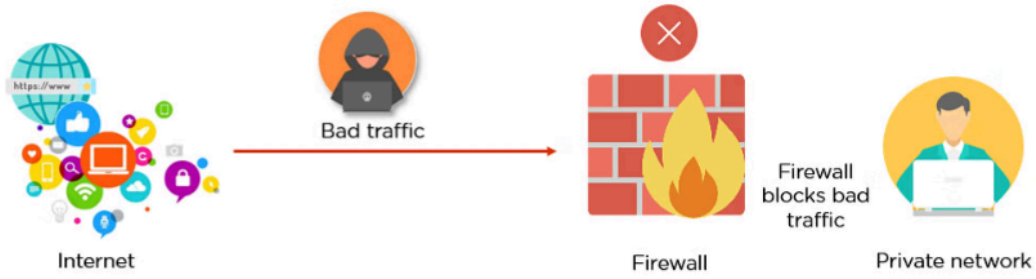
INTERNAL IP	EXTERNAL IP
Internal IP addresses are used to identify devices within a local network. These addresses are not routable over the internet and are meant for communication within the confines of the local network.	External IP addresses are used to identify a device on the internet. When your device communicates with servers or devices outside your local network, it uses its external IP address.
Internal IP addresses are not directly accessible from the internet.	External IP addresses are visible to the external world. Websites, servers, and other devices on the internet see your device's external IP address when you access their services.
Internal IP addresses often fall within certain private address ranges defined by standards such as RFC 1918. Common private address ranges include 192.168.x.x, 172.16.x.x–172.31.x.x, and 10.x.x.x.	External IP addresses are publicly routable, meaning they can be used to route traffic over the internet.

Ipconfig and ifconfig

ipconfig: Command used in Windows command prompt to display IP configuration information, including the IPv4 and IPv6 addresses, subnet mask, and gateway for network interfaces.

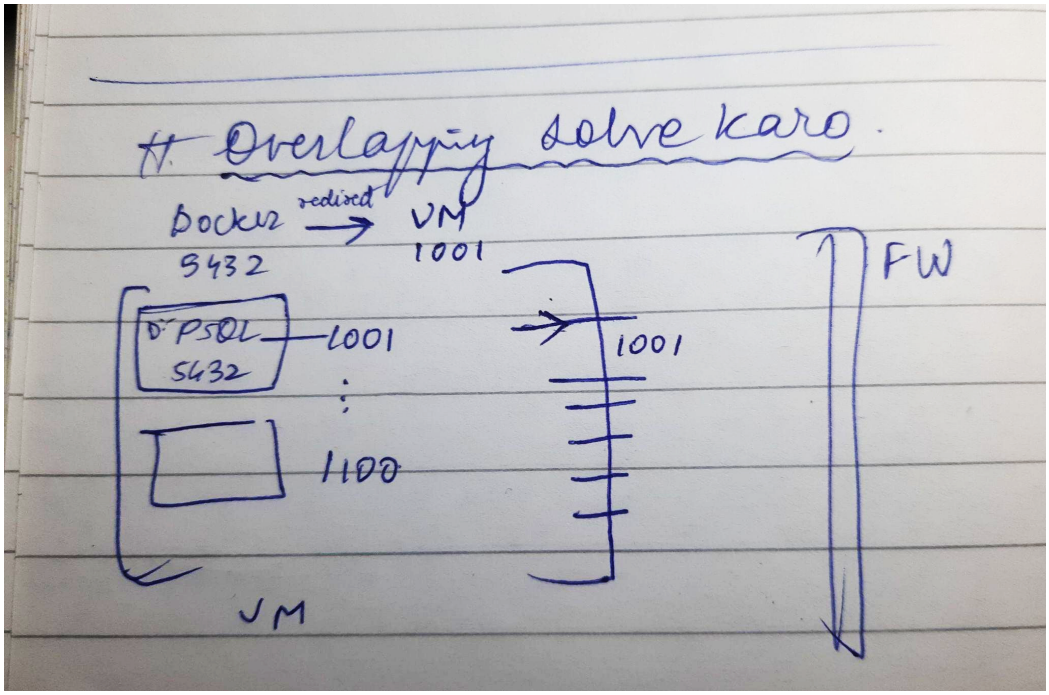
ifconfig: Command used in Unix-like operating systems (e.g., Linux) to display and configure network interfaces, providing information about IP addresses, netmask, and other network-related settings.

Firewall



A firewall is essential software or firmware in network security that is used to prevent unauthorized access to a network. It is used to inspect the incoming and outgoing traffic with the help of a set of rules to identify and block threats by implementing it in software or hardware form. Firewalls can be used in both personal and enterprise settings, and many devices come with one built-in, including Mac, Windows, and Linux computers.

Solving the overlapping laps



- Redirecting Ips from docker to VM inside a VM

If you want to redirect traffic from a virtual machine (VM) to a Docker container, you can achieve this using port forwarding. Port forwarding allows you to redirect traffic from a specific port on the VM to a port on the Docker container.

1. Identify VM Details: Find the VM's IP using ipconfig (Windows) or ifconfig (Unix).
2. Choose VM Port: Decide the VM port for forwarding.
3. Run Docker Container: Start the container with port forwarding:

   docker run -p <VM_port>:<container_port> <image_name>

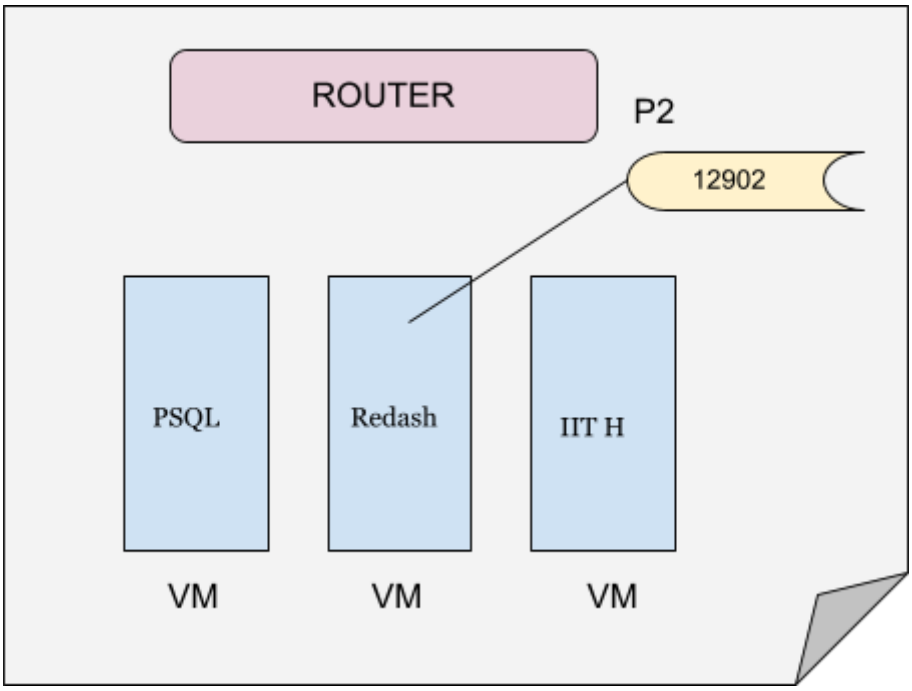
4. Access Service: Access the Docker service via VM's IP and chosen port. Example:

   VM IP: <VM_IP> VM

   Port: 8080 Docker

   Container Port: 80 This way, when someone tries to access the VM on port 8080, the traffic is automatically redirected to the Docker container running NGINX on port 80.

Router allocates public IPs to the server inside it.



Accessing server from inside= Check IP Address of that server's application

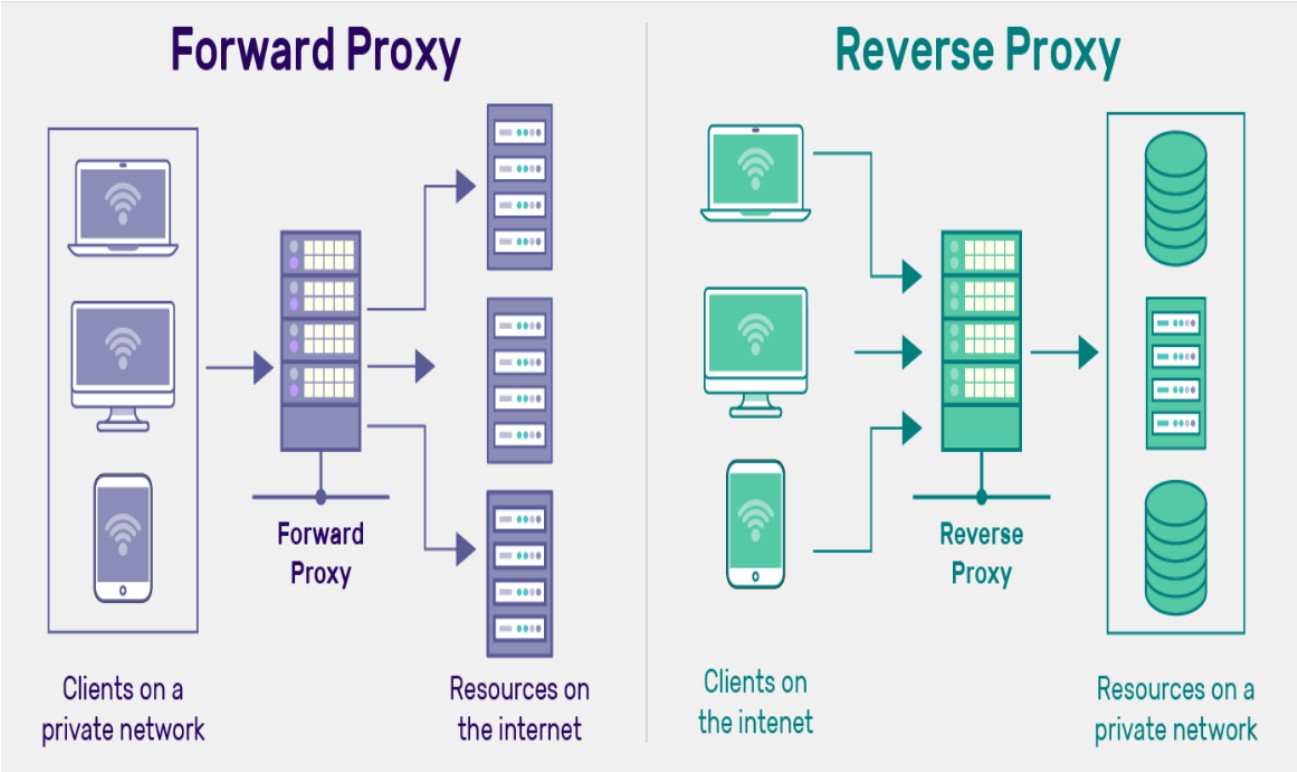
Use of reverse proxy to access ports

In simpler terms, it's like having a receptionist (reverse proxy) who takes messages at the main entrance (standard port) and directs them to the correct person or department (service on a specific port) inside the building (your server). This way, users only need to remember one entry point, and the reverse proxy handles the rest.

Imagine you have different services running on your server, each on its own port. Instead of giving out separate ports for each service, the reverse proxy allows you to use a single port (like 80 or 443) for all your services. It listens for requests and knows which service to send them to based on the web address or path.

Note: We never keep any port exposed.

Forward proxy

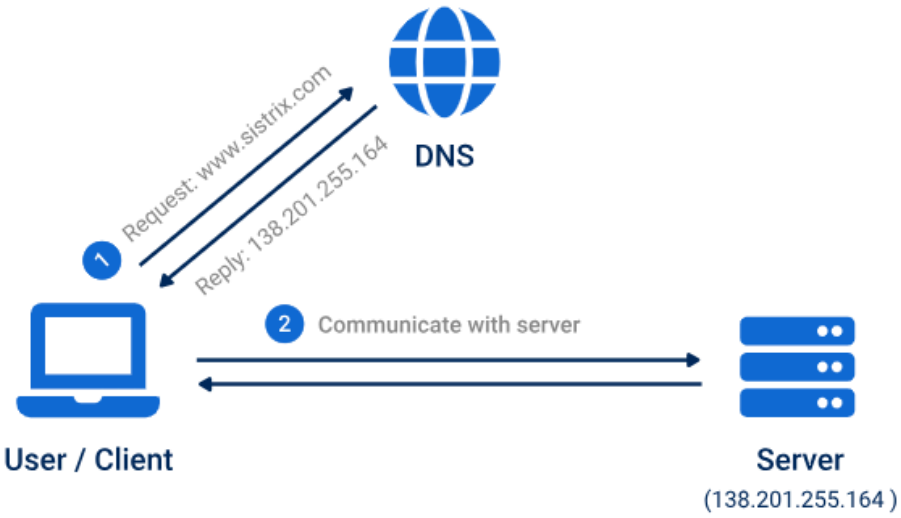


Reverse Proxy, tunneling / Port Forwarding (via SSH)

- DNS

Made by Sejal Randive

For continuously changing IP addresses, just type the website's name and whichever IP Is allotted will be there connected in the backend, Using GoDaddy



NGINX (same server pe alag alag routings) (like a receptionist)

Definition: NGINX is a powerful web server, load balancer, and reverse proxy server.