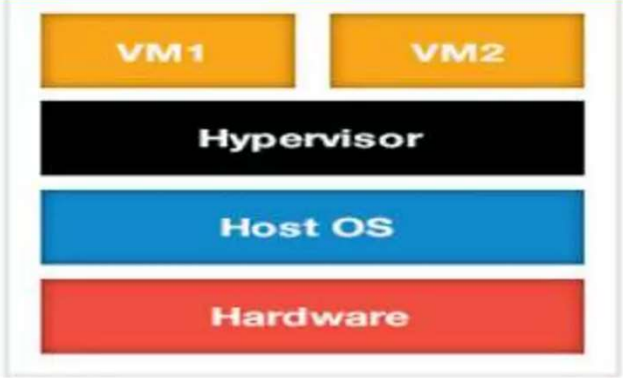
INTRODUCTION OF VIRTUALIZATION

- Virtualization is a technique, which allows to share single physical instance of an application or resource among multiple organizations or tenants (customers)..
- Virtualization is a proved technology that makes it possible to run multiple operating system and applications on the same server at same time.
- Virtualization is the process of creating a logical(virtual) version of a server operating system, a storage device, or network services.
- The technology that work behind virtualization is known as a virtual machine monitor(VM), or virtual manager which separates compute environments from the actual physical infrastructure.

What is the concept behind the Virtualization

- Creation of a virtual machine over existing operating system and hardware.
- **Host machine:** The machine on which the virtual machine is created.
- **Guest machine:** virtual machines referred as a guest machine.
- **Hypervisor:** Hypervisor is a firmware or low-level program that acts as a Virtual Machine Manager.

ARCHITECTURE OF VITUALIZATION



Hosted hypervisor

- Examples:

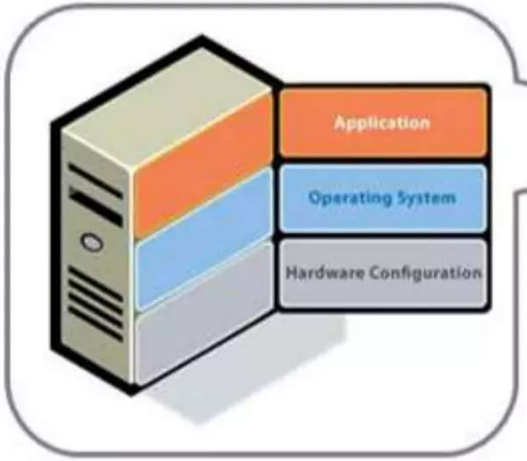
1. RTS Hypervisor
2. Oracle VM
3. Vmware
4. VirtualLogic VLX

Bare metal hypervisor

- Examples:

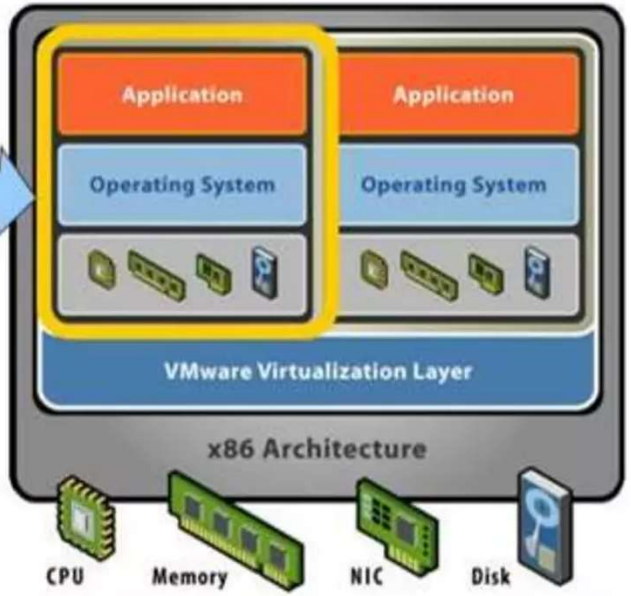
1. KVM
2. VMWare Fusion
3. Virtual Server 2005 R2
4. Windows Virtual PC
5. VMWare workstation 6.0

Before Virtualization



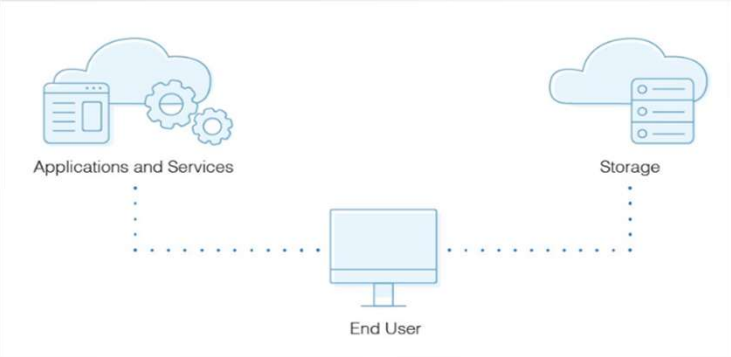
- Software tied to hardware
- Single OS image per machine
- One application workload per OS
- Inflexible and costly infra structure

After Virtualization



- Multiple workloads per machine
- Software independent of hardware
- System, data, apps are files

Virtualization opportunities:



Virtualization enables users to disjoint operating systems from the underlying hardware, i.e, users can run multiple operating systems such as Windows, Linux, on a single physical machine at the same time. Such operating systems are known as guest OS (operating systems).

•Virtualization deploys software that makes an abstraction layer across computer hardware, letting the hardware components such as processors, memory, storage etc of a particular computer to be segmented into several virtual elements (also known as virtual machines).

Some terminologies associated with Virtualization:

1.Hypervisor: It is an operating system, performing on the actual hardware, the virtual counterpart is a subpart of this operating system in the form of a running process. Hypervisors are observed as Domain 0 or Dom0.

2.Virtual Machine (VM): It is a virtual computer, executing underneath a hypervisor.

3.Container: Some light-weighted VMs that are subpart of the same operating system instance as its hypervisor are known as containers. They are a group of processes that runs along with their corresponding namespace for process identifiers.

4.Virtualization Software: Either be a piece of a software application package or an operating system or a specific version of that operating system, this is the software that assists in deploying the virtualization on any computer device.

5.Virtual Network: It is a logically separated network inside the servers that could be expanded across multiple servers.

Characteristics of Virtualization:

1. **Resource Distribution:** Either be a single computer or a network of connected servers, virtualization allows users to make a unique computer environment from one host machine that lets users to restrict the participants as active users, scale down power consumption and easy control.
2. **Isolation:** Virtualization software involves self-contained virtual machines, these VMs give guest users (not an individual but a number of instances as applications, operating systems, and devices) an isolated online, virtual environment. This online environment not only defends sensitive knowledge but also allows guest users to remain-connected.
3. **Availability:** Virtualization software provides various number of features that users won't obtain at physical servers, these features are beneficial in increasing uptime, availability, fault tolerance, and many more. These features help users to avoid downtime that subverts the users' efficiencies and productivities and also generates security threats and safety hazards.
4. **Aggregation:** Since virtualization allows several devices to split resources from a single machine, so it can be deployed to join multiple devices into a single potent host. In addition to that, aggregation also demands for cluster management software in order to connect a homogeneous group of computers or servers collectively for making a unified resource center.
5. **Authenticity and security:** At ease, virtualization platforms assure the continuous uptime by balancing load automatically that runs an excessive number of servers across multiple host machines in order to prevent interruption services.

Benefits of Virtualization:

There are following benefits of virtualization in cloud computing;

1. **Security:** :Security has been the advantageous concern for adopting virtualization. The security is served through firewalls that prevent from any unreliable access and preserve the data safe and confidential.
2. **Flexible Operations:** With the deployment of virtualization, users can work efficiently as the working process is very streamlined and agile. Presently, the employed network switch is easy to use, flexible and saves time.
3. **Economical:** This is the most prime reason to choose virtualization rapidly as with this technique companies can manage additional expenditure on physical devices and servers. Being active with a virtual environment, data can be gathered on virtual servers. It also reduces the rigorous use of electricity (that has been a concern if several physical devices and services are being used at the same time), lowering bills while executing the numerous components of an operating system and applications over the users and companies network.
4. **Flexible data transfer:** The data can be transferred to virtual servers anytime and also be retrieved due to this users or cloud providers need not to waste time in finding out hard drives to discover data. With the implementation of virtualization, it has become easy to allocate the required data and transfer them to the appropriate authorities. Moreover, there is no limitation of data transfer and can be transferred to a far distance with minimal charges.
5. **Remove system failure risks:** While performing any function, it often happens that the system might malfunction in critical timing such that this system failure could be adverse for a company's resources and also deteriorate its reputation. This system failure can be protected with virtualization as users could perform the same task simultaneously over multiple devices, and the accumulated data can also be retrieved anytime with any device.

Virtualization Technologies

28/03/24

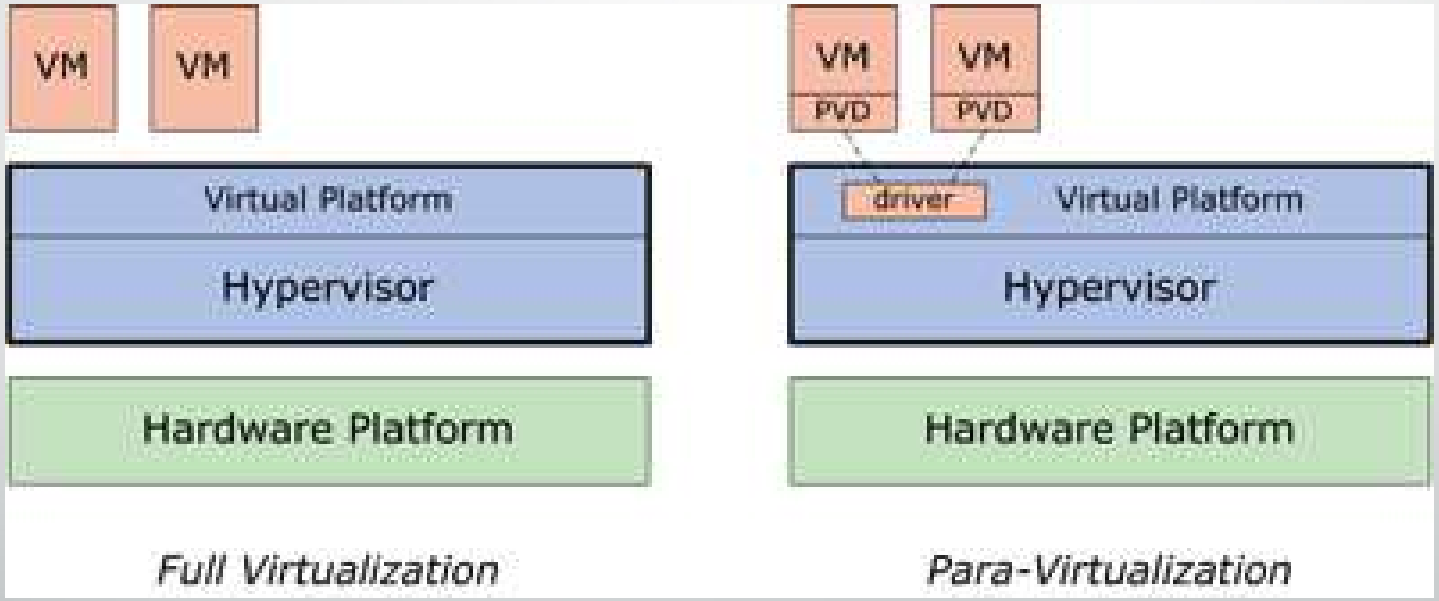
Full Virtualization

- Full Virtualization is a virtualization in which the guest operating system is unaware that it is in a virtualized environment, and therefore hardware is virtualized by the host operating system so that the guest can issue commands to what it thinks is actual hardware, but really are just simulated hardware devices created by the host.

Paravirtualization

- Paravirtualization is a virtualization technique in which the guest operating system is aware that it is a guest and accordingly has drivers that, instead of issuing hardware commands, simply issues commands directly to the host operating system.

Full Virtualization Vs. Para-Virtualization



Nested Virtualization

- Is simply a VM running inside another
- VMExample: A Windows 8 VM running a ubuntu VM.

Hybrid Virtualization

- This virtualization technique is a combination of Paravirtualization and Full Virtualization, where parts of the guest operating system use paravirtualization for certain hardware drivers, and the host uses full virtualization for other features. This often produces superior performance on the guest without the need for it to be completely paravirtualized.
- An example of this: The guest uses full virtualization for privileged instructions in the kernel but paravirtualization for IO requests using a special driver in the guest.

OS Virtualization

- Operating-system-level virtualization is a virtualization method where the kernel of an operating system allows for multiple isolated user space instances, instead of just one. Such instances (Ex. Containers) may look and feel like a real server from the point of view of its owners and users.
- Operating-system-level virtualization usually imposes little to no overhead, because programs in virtual partitions use the operating system's normal system call interface and do not need to be subjected to emulation or be run in an intermediate virtual machine, as is the case with whole-system virtualizers.
- This form of virtualization also does not require support in hardware to perform efficiently.

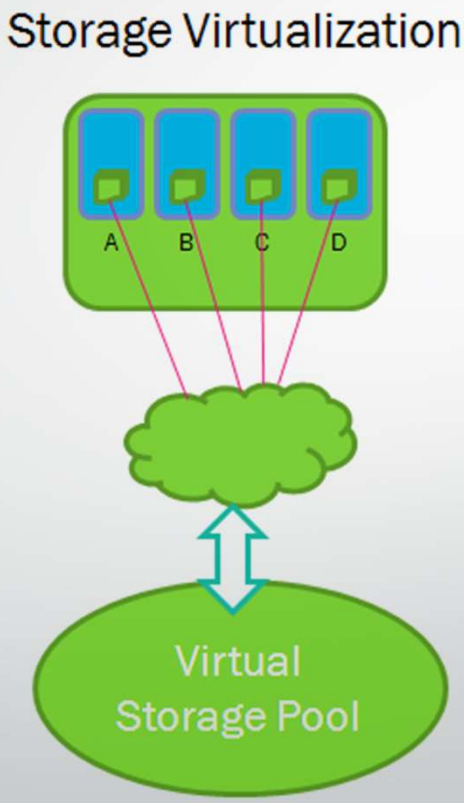
- This virtualization is not as flexible as other virtualization approaches since it cannot host a guest operating system different from the host one, or a different guest kernel. For example, with Linux, different distributions are fine, but other operating systems such as Windows cannot be hosted.

- Some operating-system-level virtualization implementations provide file-level copy-on-write mechanisms. This means that a standard file system is shared between partitions, and those partitions that change the files automatically create their own copies. This is easier to back up, more space-efficient and simpler to cache.

Storage Virtualization

- Storage virtualization is the fusion of multiple network storage devices into what appears to be a single storage unit. Storage virtualization is usually implemented via software applications and often used in SAN (storage area network), making tasks such as archiving, back-up, and recovery easier and faster.

Storage Virtualization (Ex.)

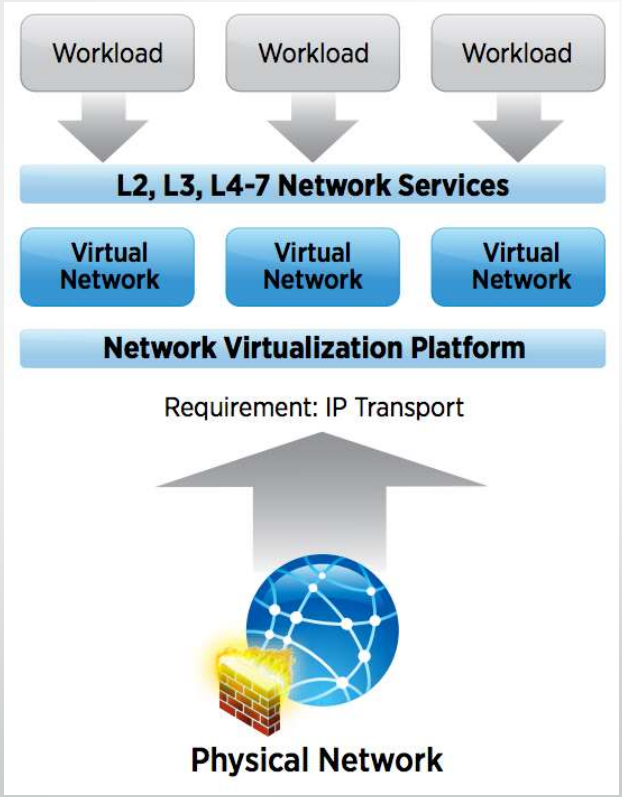


Network Virtualization

- When applied to a network, virtualization creates a logical software-based view of the hardware and software networking resources (switches, routers, etc.).

- The physical networking devices are simply responsible for the forwarding of packets, while the virtual network (software) provides an intelligent abstraction that makes it easy to deploy and manage network services and underlying network resources.

Network Virtualization(Ex.)

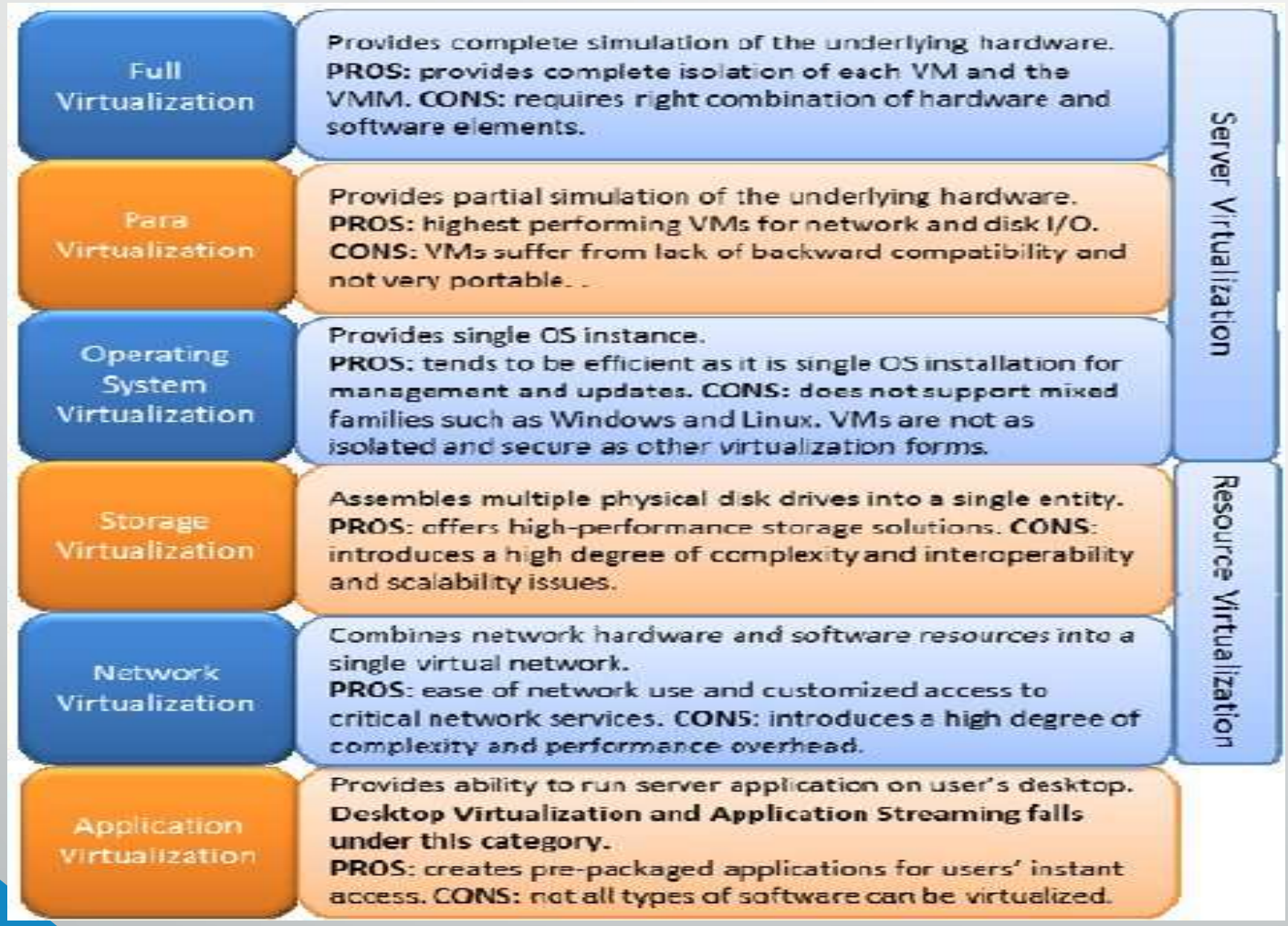


Desktop Virtualization

- Desktop virtualization is software technology that separates the desktop environment and associated application software from the physical client device that is used to access it.
- Desktop virtualization can be used in conjunction with application virtualization and user profile management systems, to provide a comprehensive desktop environment management system.
- In this mode, all the components of the desktop are virtualized, which allows for a highly flexible and much more secure desktop delivery model. In addition, this approach supports a more complete desktop disaster recovery strategy as all components are essentially saved in the data center and backed up through traditional redundant maintenance systems.

Desktop Virtualization

- If a user's device or hardware is lost, the restore is much more straightforward and simple, because basically all the components will be present at login from another device. In addition, because no data is saved to the user's device, if that device is lost, there is much less chance that any critical data can be retrieved and compromised.


Full Virtualization
Provides complete simulation of the underlying hardware. PROS: provides complete isolation of each VM and the VMM. CONS: requires right combination of hardware and software elements.
Para Virtualization
Provides partial simulation of the underlying hardware. PROS: highest performing VMs for network and disk I/O. CONS: VMs suffer from lack of backward compatibility and not very portable. .
Operating System Virtualization
Provides single OS instance. PROS: tends to be efficient as it is single OS installation for management and updates. CONS: does not support mixed families such as Windows and Linux. VMs are not as isolated and secure as other virtualization forms.
Storage Virtualization
Assembles multiple physical disk drives into a single entity. PROS: offers high-performance storage solutions. CONS: introduces a high degree of complexity and interoperability and scalability issues.
Network Virtualization
Combines network hardware and software resources into a single virtual network. PROS: ease of network use and customized access to critical network services. CONS: introduces a high degree of complexity and performance overhead.
Application Virtualization
Provides ability to run server application on user's desktop. Desktop Virtualization and Application Streaming falls under this category. PROS: creates pre-packaged applications for users' instant access. CONS: not all types of software can be virtualized.
Server Virtualization
Resource Virtualization

Hypervisor:

Hypervisors are run in supervisor mode. The division between privileged and nonprivileged instructions has posed challenges in designing virtual machine managers. It is expected that all the sensitive instructions will be executed in privileged mode. Without this assumption it is impossible to fully emulate and manage the status of the CPU for guest operating systems. It is expected that **in a hypervisor-managed environment, all the guest operating system code will be run in user mode in order to prevent it from directly accessing the status of the CPU**.

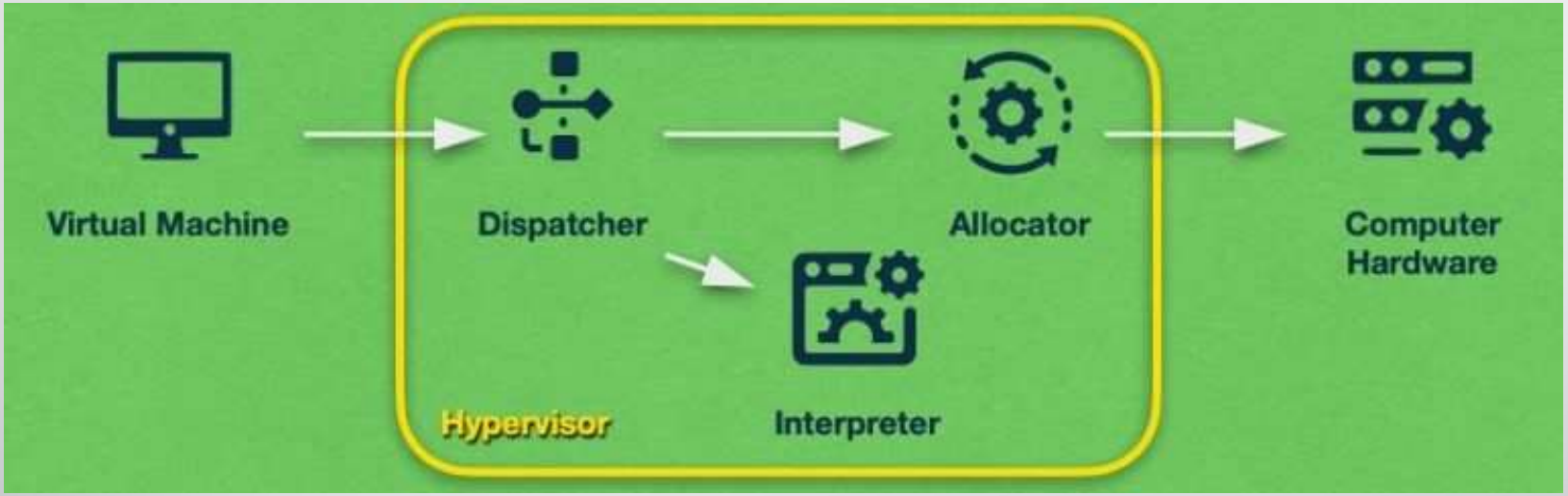
Hypervisors Or Virtual Machine Manager (VMM):

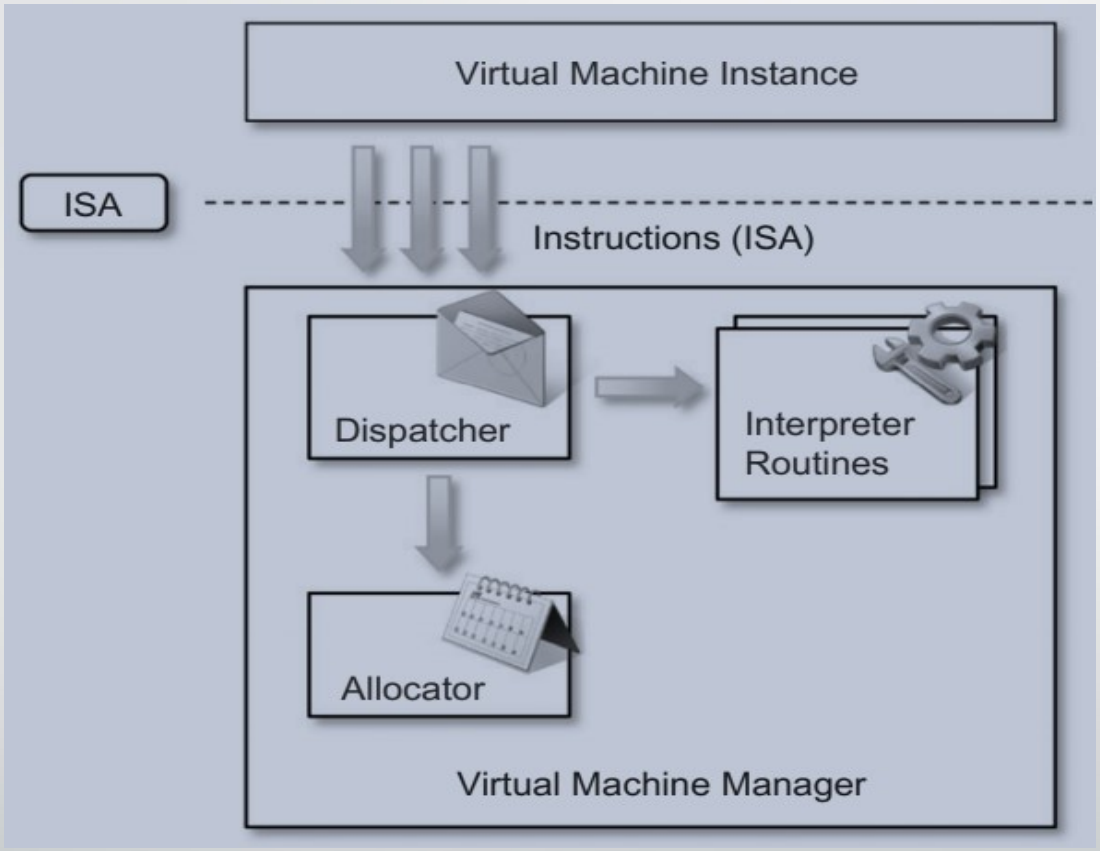
A fundamental element of hardware virtualization It recreates a hardware environment in which guest operating systems are installed.

Three main modules to coordinate their activity in order to emulate the underlying hardware.

1. Dispatcher
2. Allocator
3. Interpreter

- The ***dispatcher*** constitutes the entry point of the monitor and reroutes the instructions issued by the virtual machine instance to one of the two other modules.
- The *dispatcher* is the component that receives the instructions sent from the virtual machine. It does not carry out these instructions, but instead, it just forwards them to one of the other two modules.
- The ***allocator*** is responsible for deciding the system resources to be provided to the VM whenever a virtual machine tries to execute an instruction that results in changing the machine resources associated with that VM, the allocator is invoked by the dispatcher.
- The *allocator* manages the amount of system resources that are available to the virtual machine. If an instruction would require a change in the virtual machine's current resource allocation, then the allocator will be invoked to carry out that change.
- The ***interpreter*** module consists of interpreter routines.
- These are executed whenever a virtual machine executes a privileged instruction: a trap is triggered and the corresponding routine is executed.

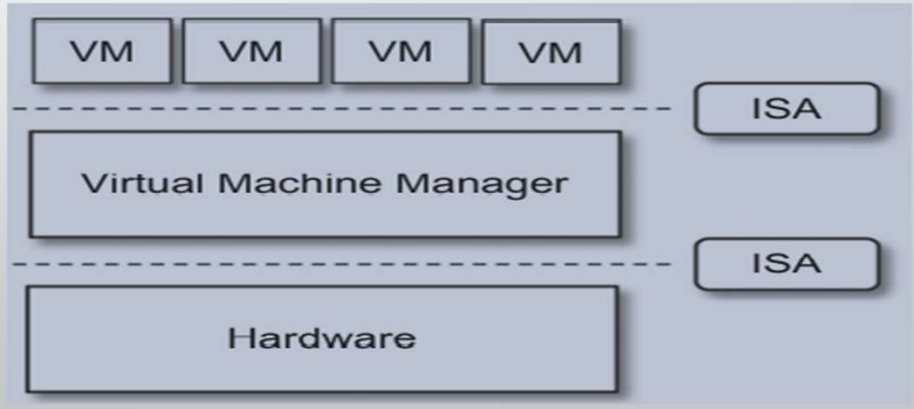

Virtual Machine
Dispatcher
Allocator
Computer Hardware
Hypervisor
Interpreter


Virtual Machine Instance
ISA
Instructions (ISA)
Dispatcher
Interpreter
Routines
Allocator
Virtual Machine Manager

Hypervisors Or Virtual Machine Manager (VMM):

Type I Hypervisor:

- **Run directly on top of the hardware**.
- Take the place of the operating systems and interact directly with the ISA interface
- They **emulate the ISA interface to allow the management of guest operating systems**
- This type of hypervisor is also called a native virtual machine since it runs natively on hardware.
- A bare-metal hypervisor (Type 1) is a layer of software installed directly on top of a physical server and its underlying hardware.
- There is no software or any operating system in between, hence the name ***bare-metal hypervisor***.
- A fundamental element of hardware virtualization. It recreates a hardware environment in which guest operating systems are installed.



- A Type 1 hypervisor is proven in providing excellent performance and stability since it does not run inside Windows or any other operating system.
- **Type 1 hypervisors are an OS themselves, a very basic one on top of which you can run virtual machines**.
- The physical machine the hypervisor is running on serves virtualization purposes only. You cannot use it for anything else.

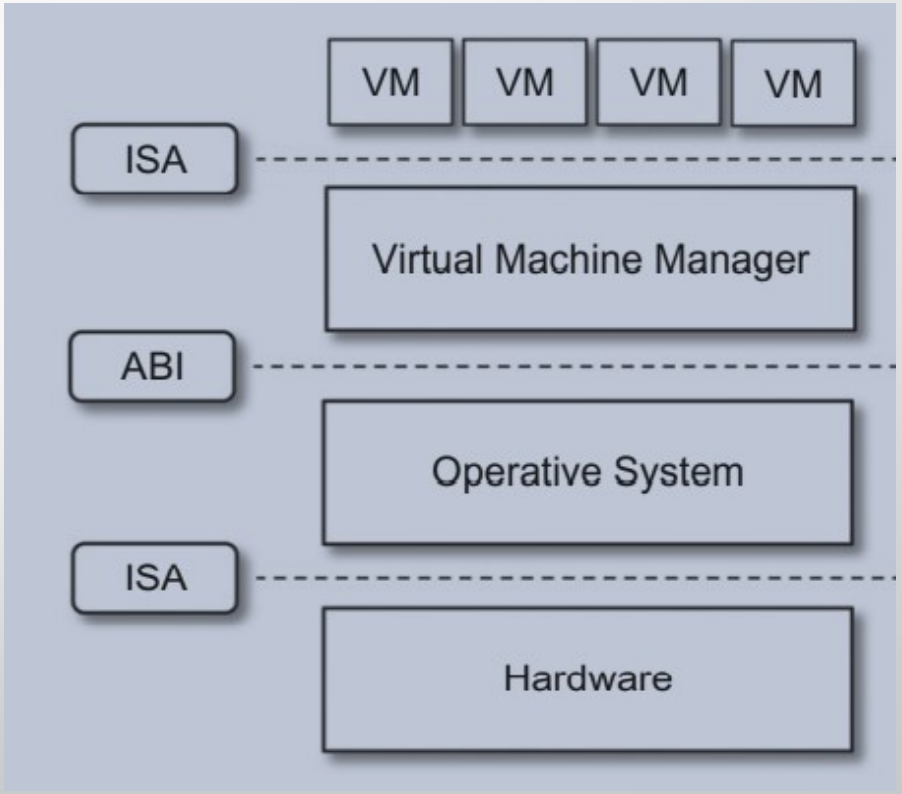
Advantages of Type-1 hypervisor

- **Great performance:** They are not constrained by the inherent limitations that come with OSes, and hence can provide great performance. This is one reason why we can find type-1 hypervisors in enterprise setups.
- **Highly secure:** Since they run directly on the physical hardware without any underlying OS, they are secure from the flaws and vulnerabilities that are often endemic to OSes. This ensures that every VM is isolated from any malicious software activity.

Type-II Hypervisor:

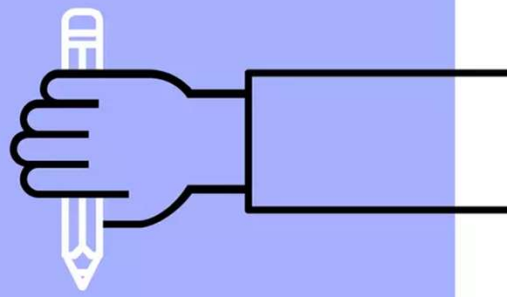
- Require the support of an operating system to provide virtualization services
- Typically **installed on top of an existing OS**
- Are programs managed by the operating system, which interact with it through the ABI and emulate the ISA of virtual hardware for guest operating systems.
- This type of hypervisor is **also called a hosted virtual machine** since it is hosted within an operating system.
- This type of hypervisor runs inside of an operating system of a physical host machine.
- This is why we call type 2 hypervisors – ***hosted hypervisors*.**

- As opposed to type 1 hypervisors that run directly on the hardware, hosted hypervisors have one software layer underneath.
- A physical machine: An operating system installed on the hardware (Windows, Linux, macOS).
- A type 2 hypervisor software within that operating system.
- The actual instances of guest virtual machines.

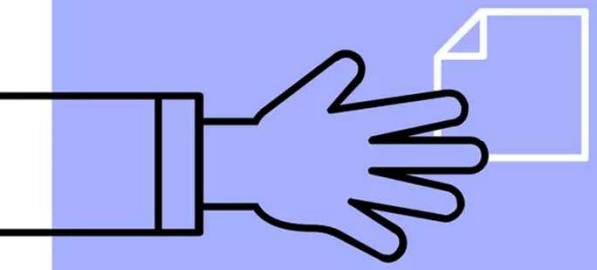


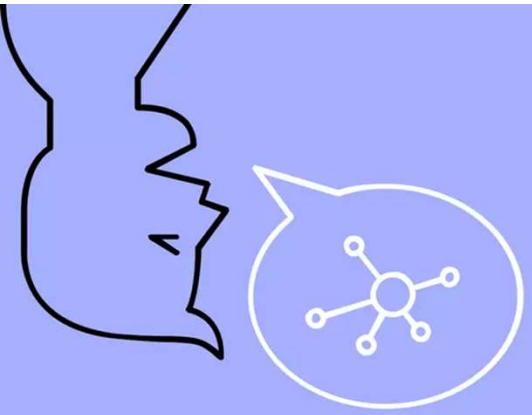
Suitability of Type II:

- Type 2 hypervisors are typically found in environments with a small number of servers.
- Type 2 hypervisors are generally not used for data center computing and are reserved for client or end-user systems sometimes called client hypervisors where performance and security are lesser concerns.
- They also come at a lower cost than Type 1 hypervisors and make an ideal test platform compared to production virtualized environments or the cloud.
- Type 2 hypervisors can support large and complex nested environments.

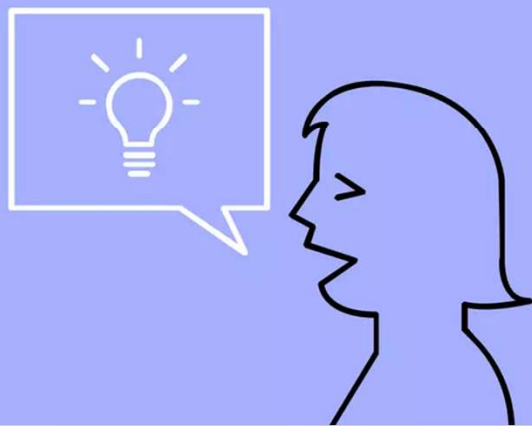


1. What is Docker?



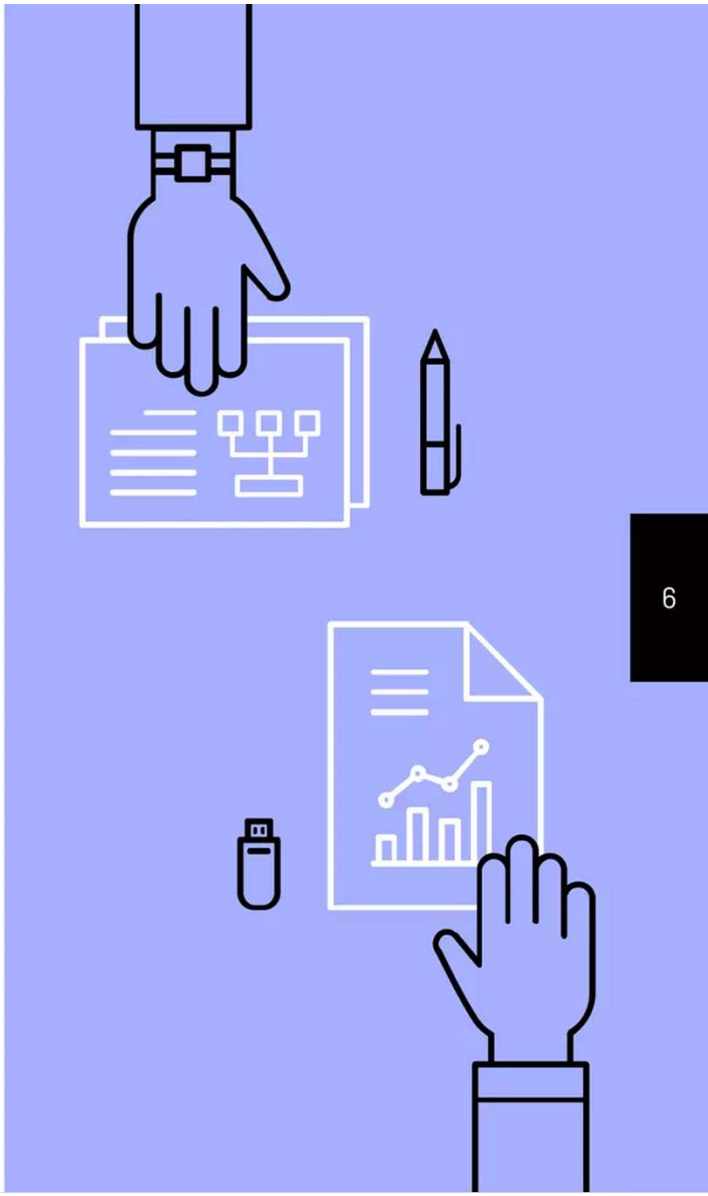


Docker is an open platform for **developing, shipping,** and **running** applications. **Docker** is designed to deliver your applications faster. With Docker you can separate your applications from your **infrastructure** and treat your infrastructure like a managed application.

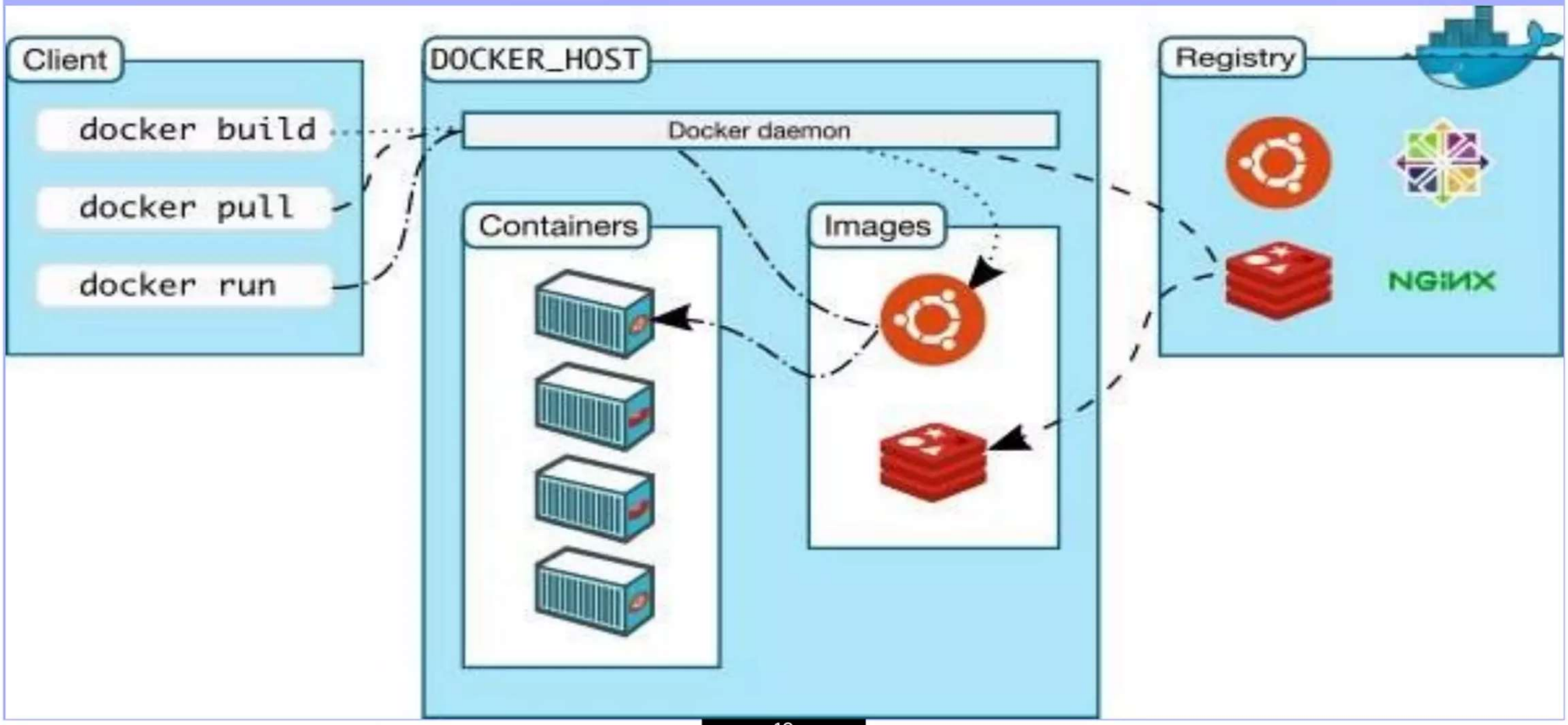


Features of Docker!

- Extremely fast and elegant isolation framework
- Inexpensive
- Low CPU/memory overhead
- Fast boot/shutdown
- Cross cloud infrastructure

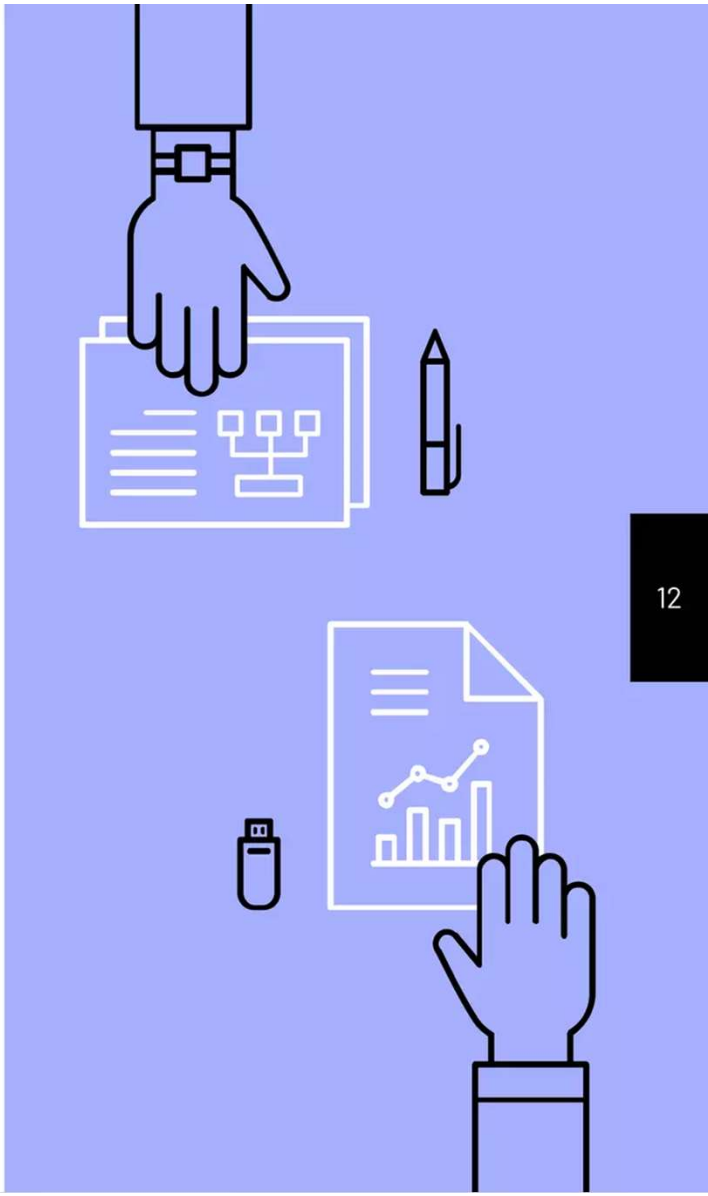


Docker Architecture



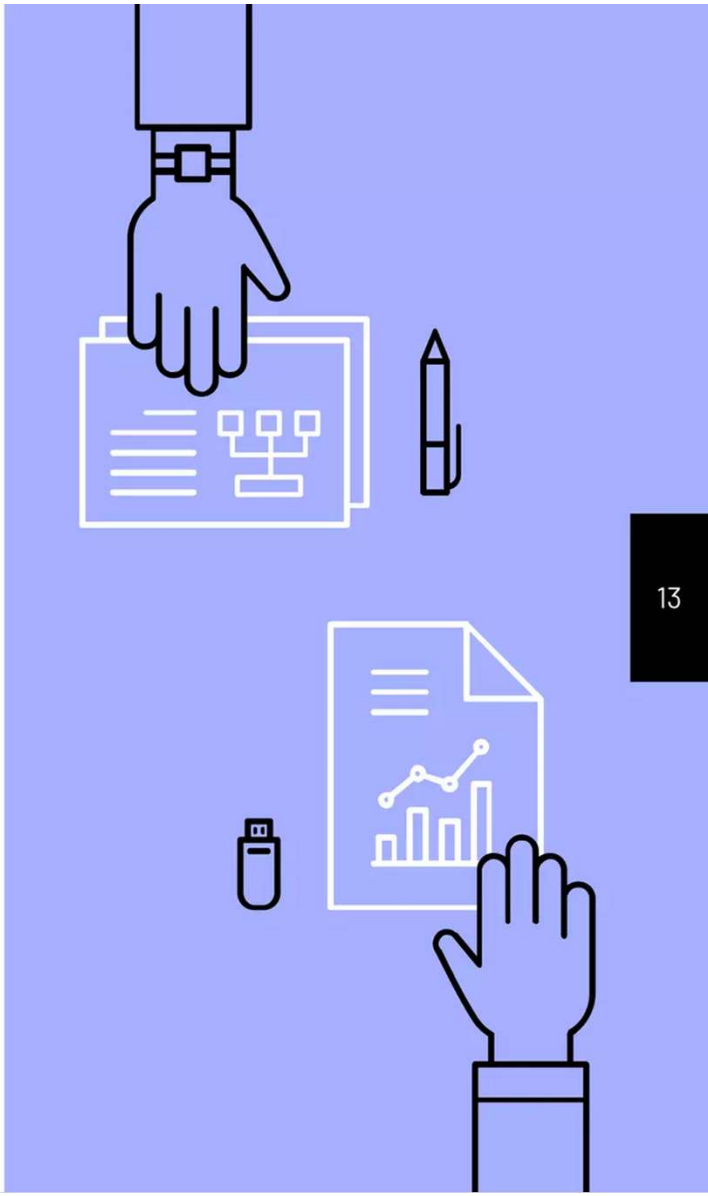
Docker Components!

- Docker Client
- Docker Daemon
- Docker Index
- Docker Containers
- Docker Images
- Dockerfile



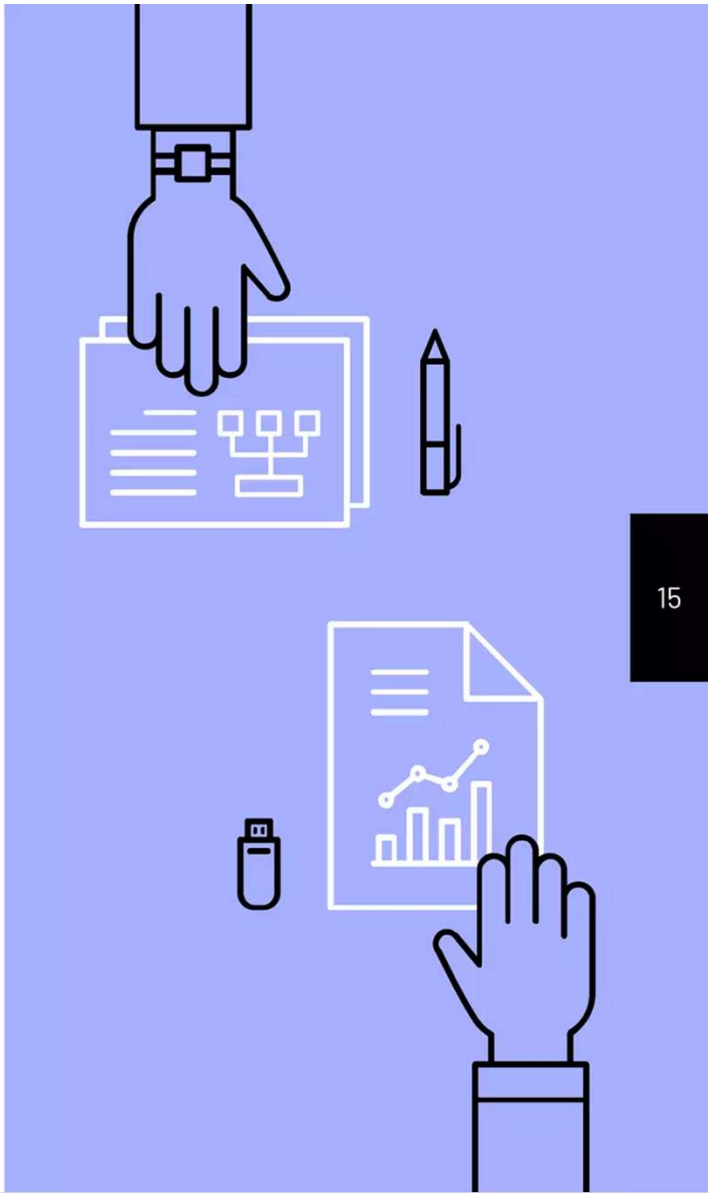
Docker Components!

- **Namespaces** serve as the first level of isolation. Makes sure a process running in a container cannot see or affect processes running outside the container.
- **Control Groups**, the key component of LXC, have resource accounting and limiting as their key functionality.
- **UnionFS** (FileSystem) serves as a building blocks of containers. It creates layers, and, thereby, accounts for Docker's lightweight and fast features.



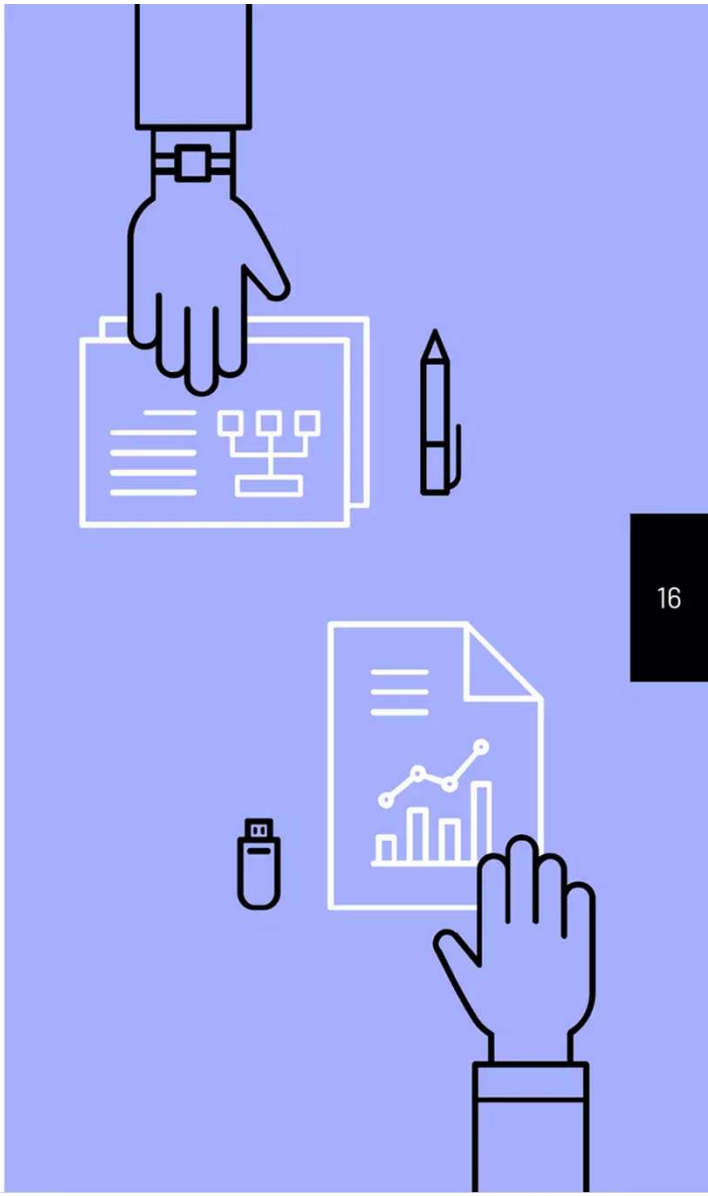
Run Applications

- **Step 1:** Build an image.
- **Step 2:** Run the container.



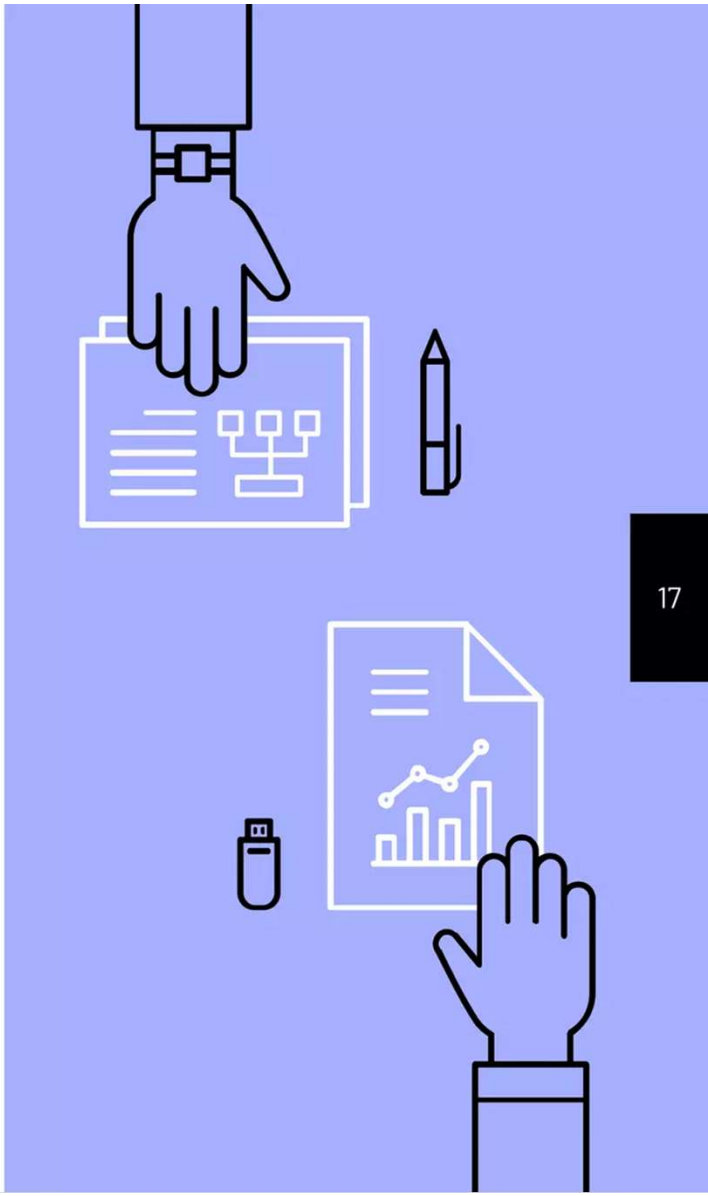
Build an Image

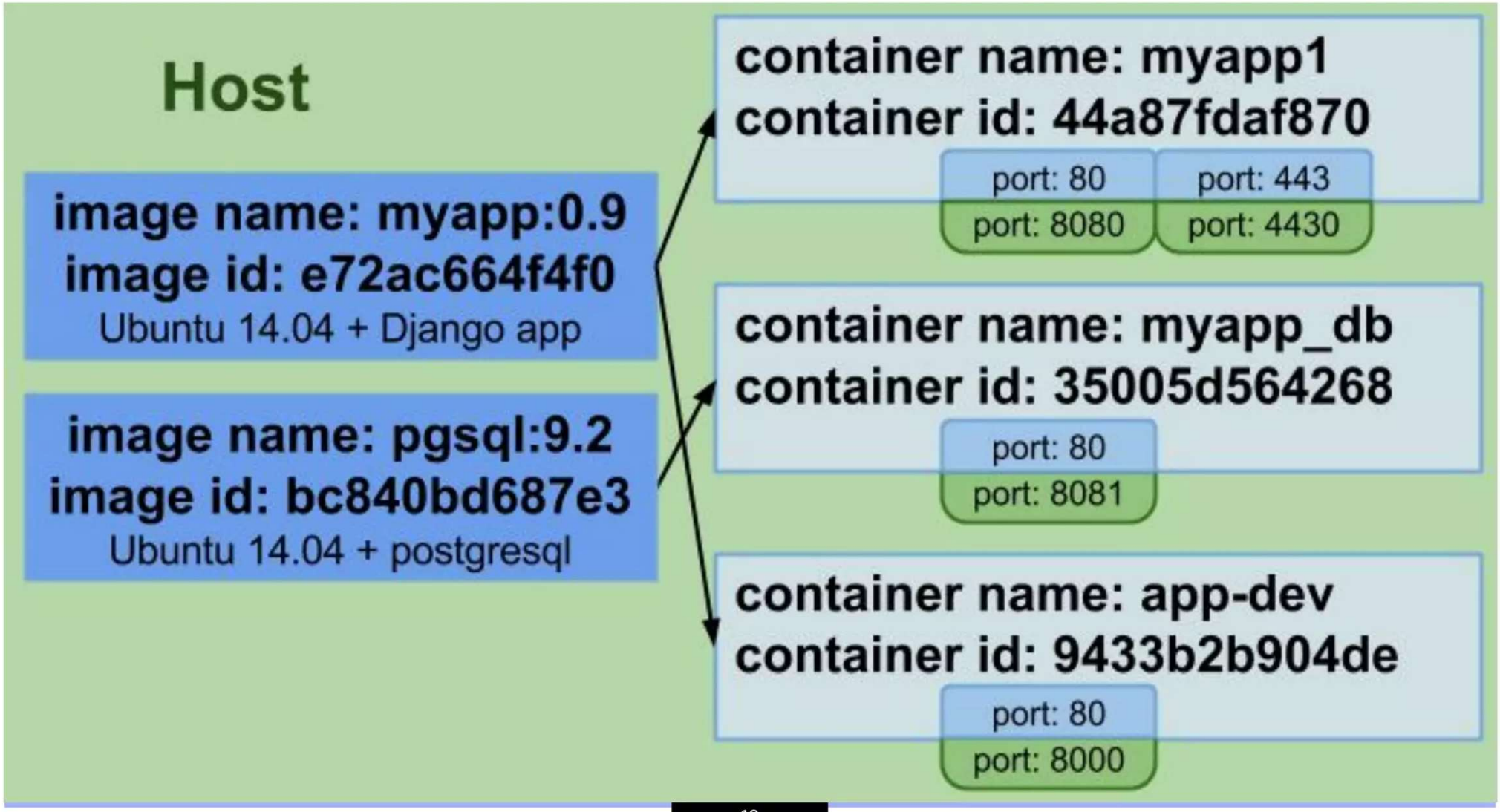
Docker Image is a read-only template to build containers. An image holds all the information needed to bootstrap a container, including what processes to run and the configuration data. Every image starts from a base image, and a template is created by using the instructions that are stored in the DockerFile. For each instruction, a new layer is created on the image.



Run The Container

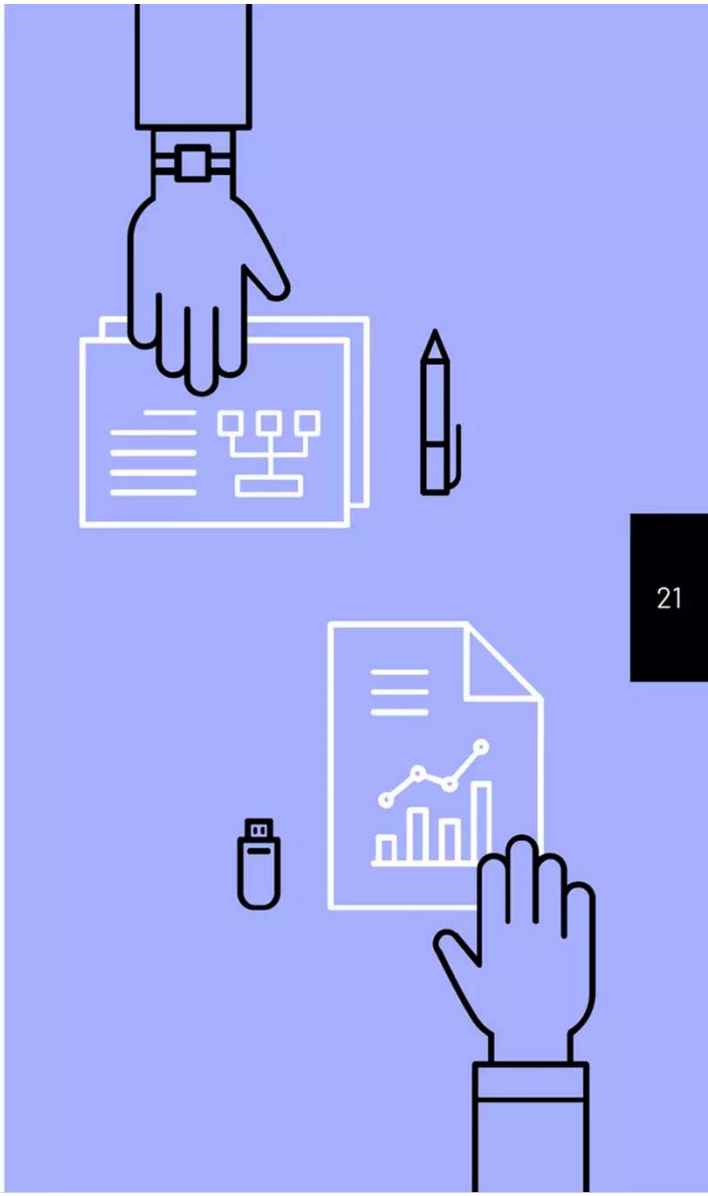
Running the container originates from the image we created in the previous step.
When a container is launched, a read-write layer is added to the top of the image.
After appropriate network and IP address allocation, the desired application can now be run inside the container.




Host
image name: myapp:0.9
image id: e72ac664f4f0
Ubuntu 14.04 + Django app
image name: pgsql:9.2
image id: bc840bd687e3
Ubuntu 14.04 + postgresql
container name: myapp1
container id: 44a87fdaf870
port: 80
port: 8080
port: 443
port: 4430
container name: myapp_db
container id: 35005d564268
port: 80
port: 8081
container name: app-dev
container id: 9433b2b904de
port: 80
port: 8000

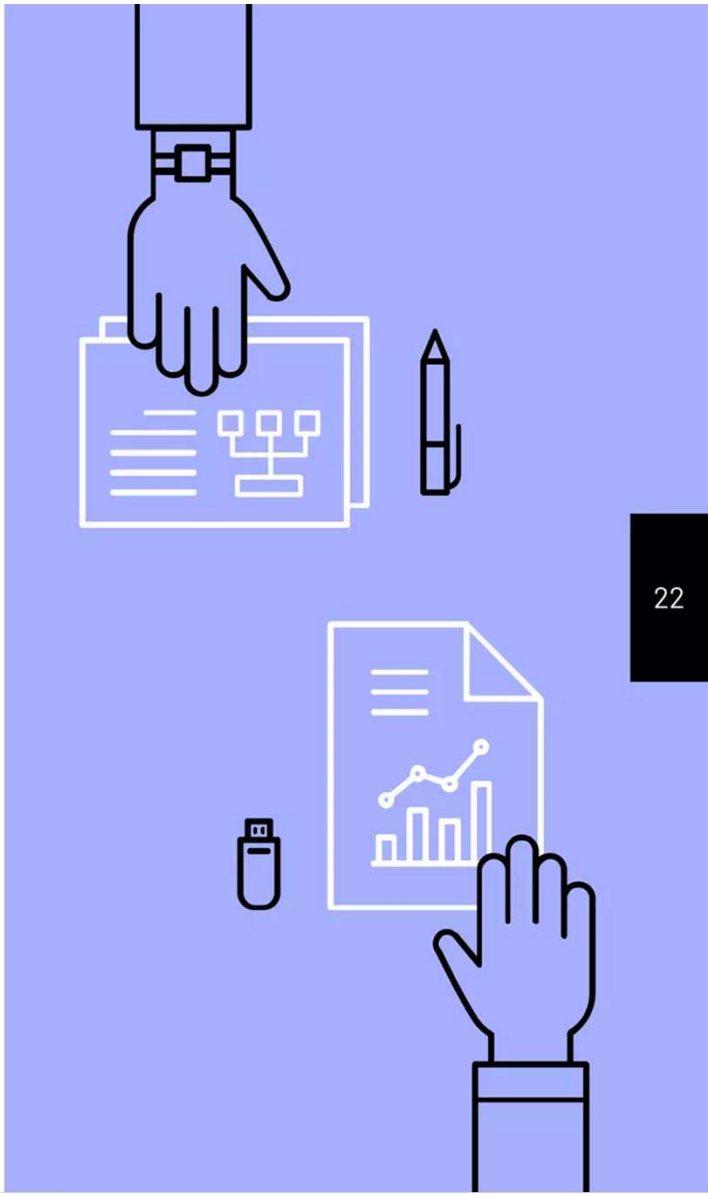
What's Kubernetes?

- Kubernetes(K8S) is an open source tool for managing containerised workloads.
- It operated at the container(not hardware) level to automate the deployment, scaling and management of applications.
- K8S works alongside a containerisation tool, like Docker. So if containers are the 'Ingredients' of an application, then K8S would be the 'Chef'.



What's Kubernetes?

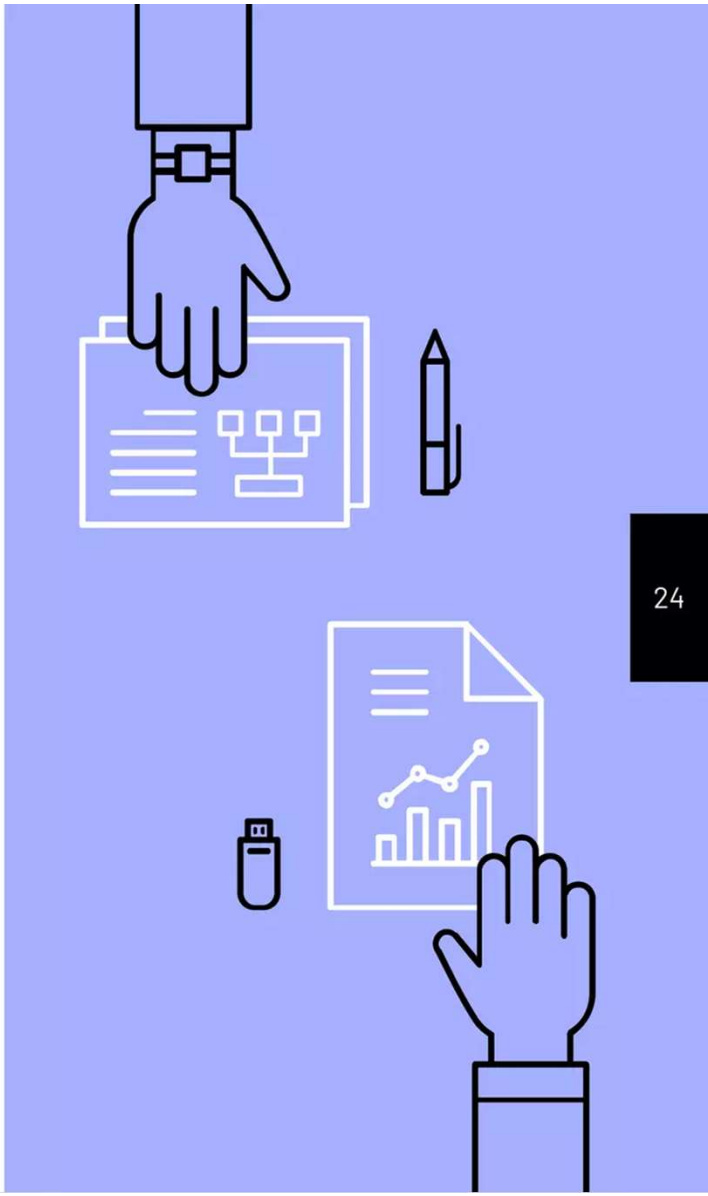
- As well as managing individual containers, K8S can also manage clusters:
  - A cluster is a series of servers connected to run containers.
  - K8S can scale upto 5000 server and 150,000 pods in a single cluster.
  - A pod is a group of containers that share resources, a network and can communicate with one another.



Why should I use it?

As an orchestration platform, K8S provides features to make the management, maintenance and life-cycle of containers easier than using a container-engine alone.

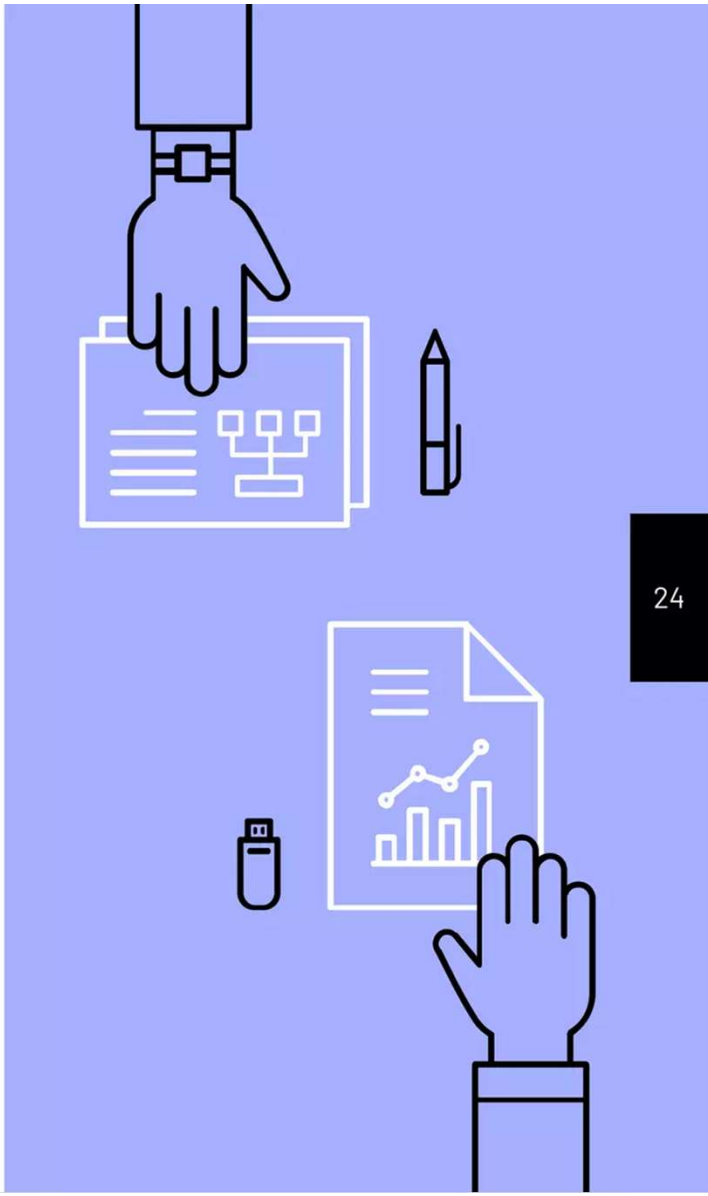
- Horizontal Scaling
- Self Healing
- Automated Rollouts
- Various other features like Service Discovery and load balancing etc.



Why should I use it?

As an orchestration platform, K8S provides features to make the management, maintenance and life-cycle of containers easier than using a container-engine alone.

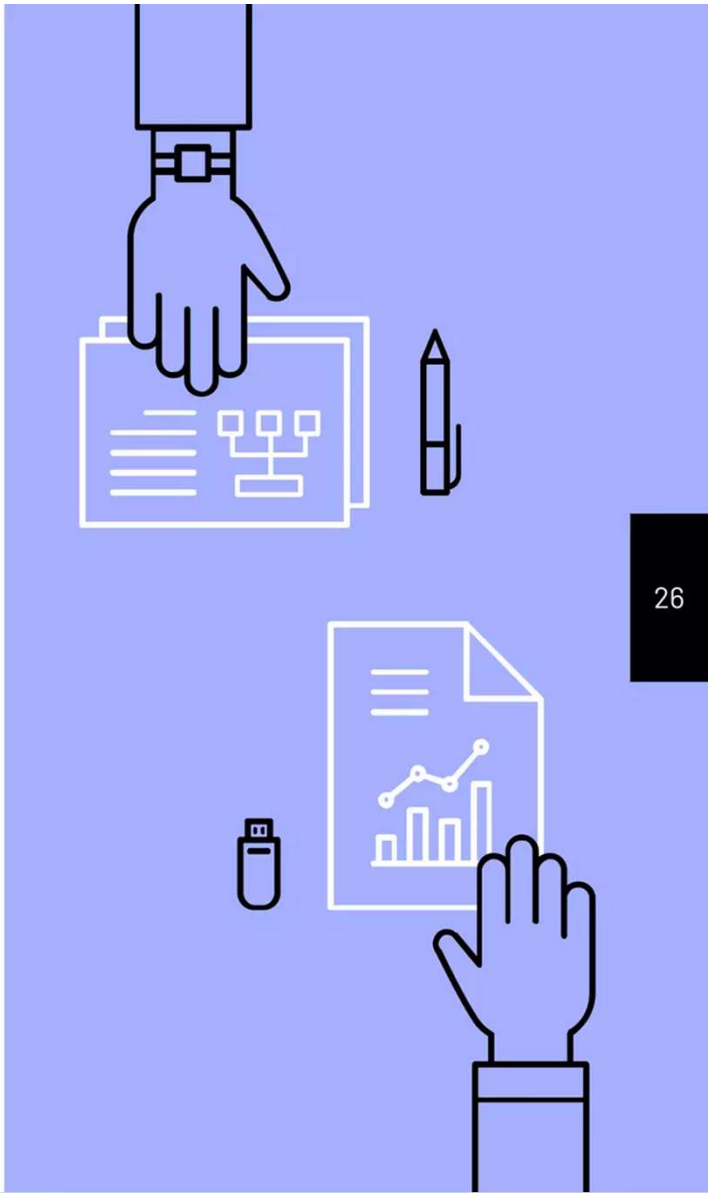
- Horizontal Scaling
- Self Healing
- Automated Rollouts
- Various other features like Service Discovery and load balancing etc.



Kubernetes Clusters

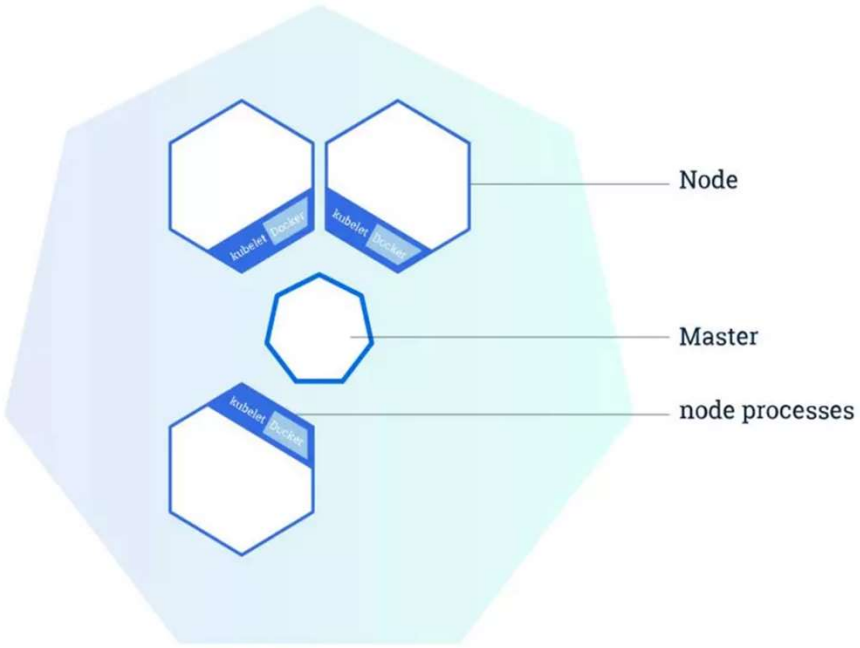
Containerised applications are deployed with K8S into highly available clusters

- Clusters run over several computers called Worker Nodes, that are connected to work as a single unit.
- Containerised apps are automatically distributed among the Worker Nodes at deploy time.
- A Master Node manages the cluster - coordinating scheduling, scaling and rolling updates.

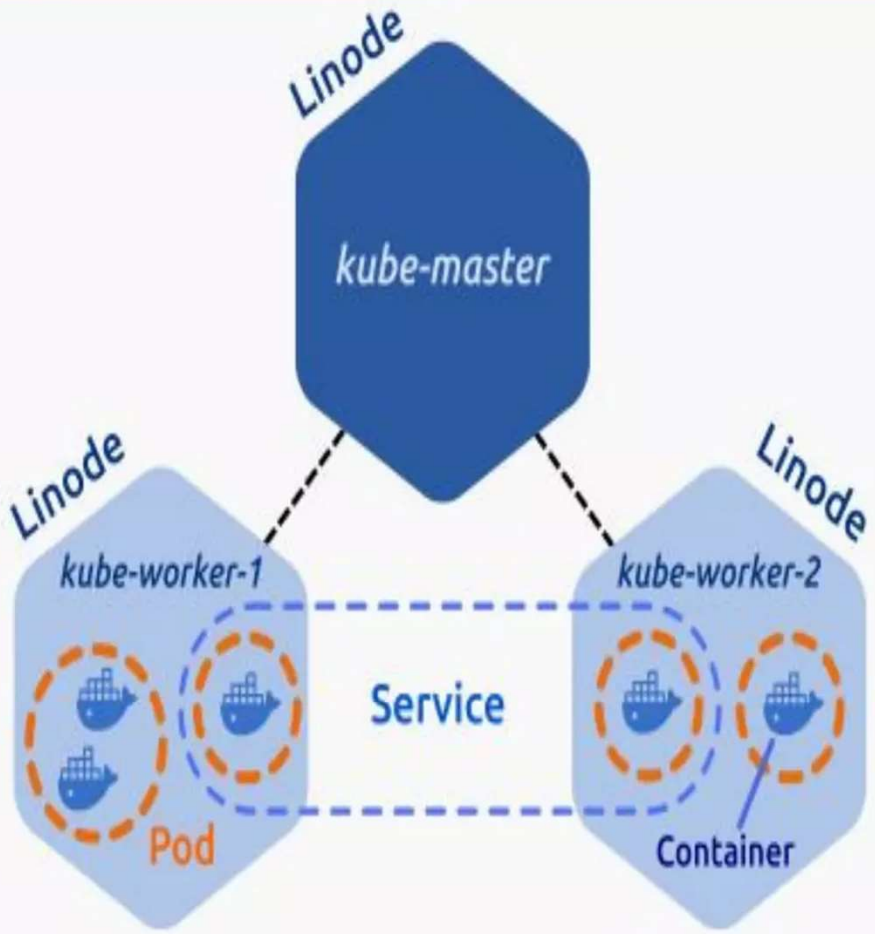


Kubernetes Clusters

Cluster Diagram



Kubernetes cluster



How does K8S Work?

Worker Node(Slave)

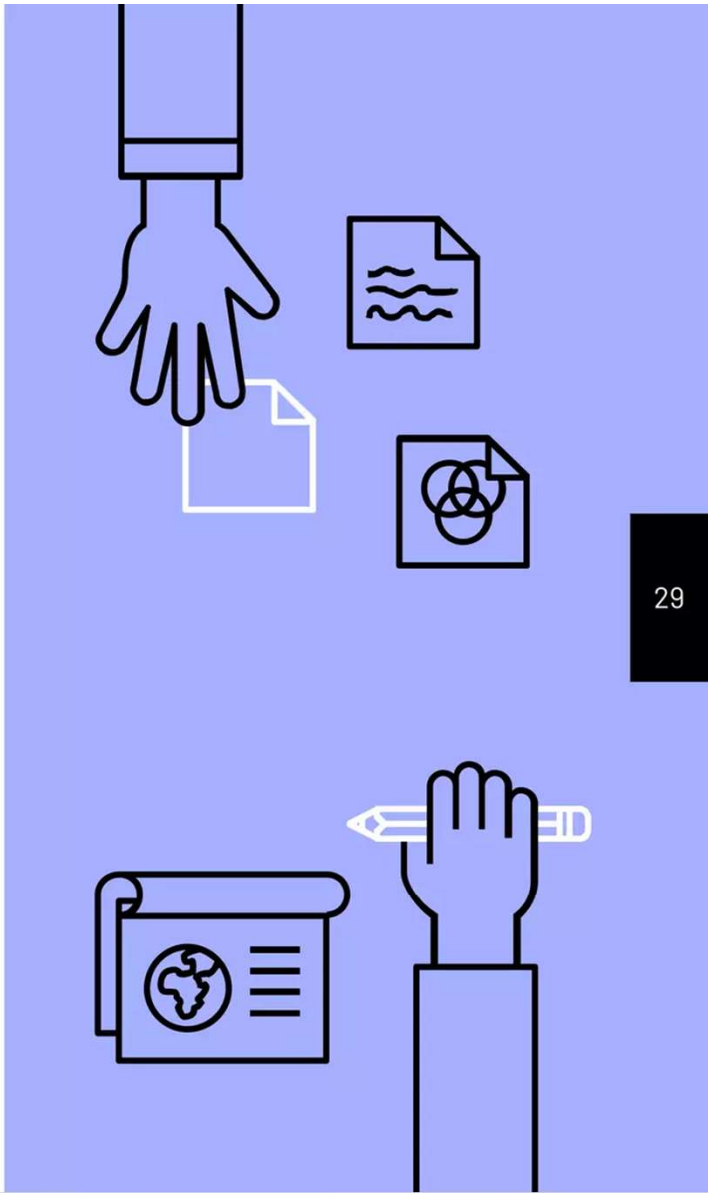
This is where containers are deployed. These nodes contains:

- Multiple Pods
- Docker Engine
- Any add-ons e.g. DNS
- Kubelet - This is the most important component as it carries out the instructions from the master node.

Master Node(Master)

This controls the deployment. This node contains:

- API Server
- Controller
- Scheduler
- Etcd - handles config management, service discovery etc



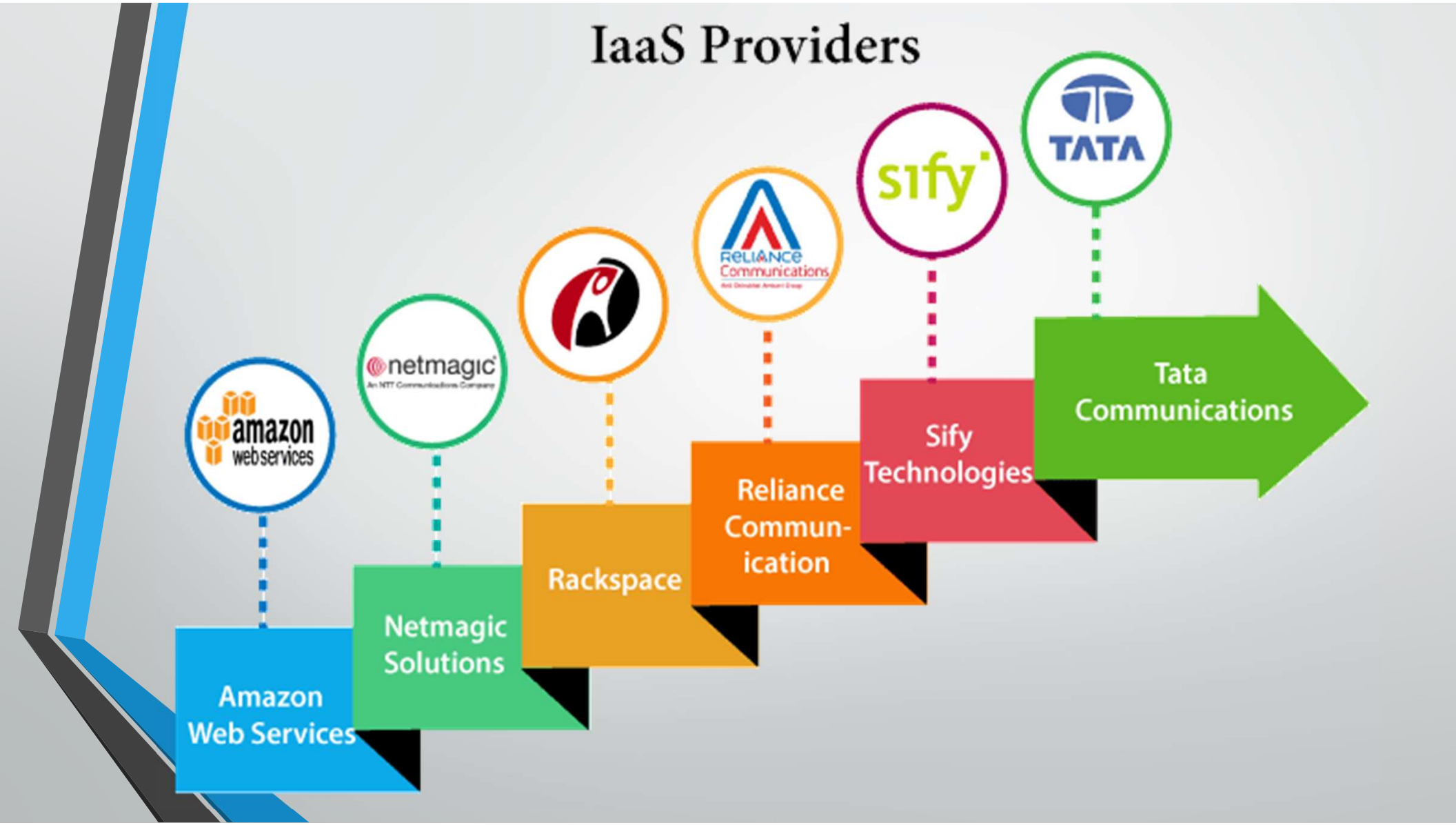
IAAS

IaaS stands for **Infrastructure as a Service**, which is a form of cloud computing that provides virtualized computing resources over the internet. Instead of buying and maintaining physical servers, storage, and networking equipment, businesses can rent these resources from a cloud service provider.

Some key features of IaaS include:

- **Scalability**: Easily scale resources up or down as needed.
- **Cost Efficiency**: Pay only for what you use, avoiding large upfront costs.
- **Flexibility**: Users have control over the operating systems, storage, and networking but rely on the provider for hardware management.

Examples of IaaS providers include Amazon Web Services (**AWS**), **Microsoft Azure**, and Google Cloud Platform (**GCP**).


IaaS Providers
amazon web services
netmagic
An NTT Communications Company
Reliance Communications
sify
TATA
Amazon Web Services
Netmagic Solutions
Rackspace
Reliance Commun-ication
Sify Technologies
Tata Communications

MANAGING VIRTUAL RESOURCES IN CLOUD

Managing virtual resources in the cloud involves handling the allocation, monitoring, scaling, and optimization of cloud-based infrastructure and services. Virtual resources refer to the computing, storage, and networking components that cloud providers offer through virtualization.

- Provisioning Resources
- Scaling and Auto-scaling
- Monitoring and Performance Management
- Load Balancing
- Backup and Disaster Recovery
- Cost Management and Optimization
- Security and Compliance

Tools for Managing Virtual Resources

Terraform: Infrastructure as Code (IaC) tool that allows you to define and manage cloud infrastructure using configuration files.

Ansible/Chef/Puppet: Configuration management tools that help automate and manage deployment and scaling.

Kubernetes: For containerized environments, Kubernetes automates deployment, scaling, and management of containerized applications.

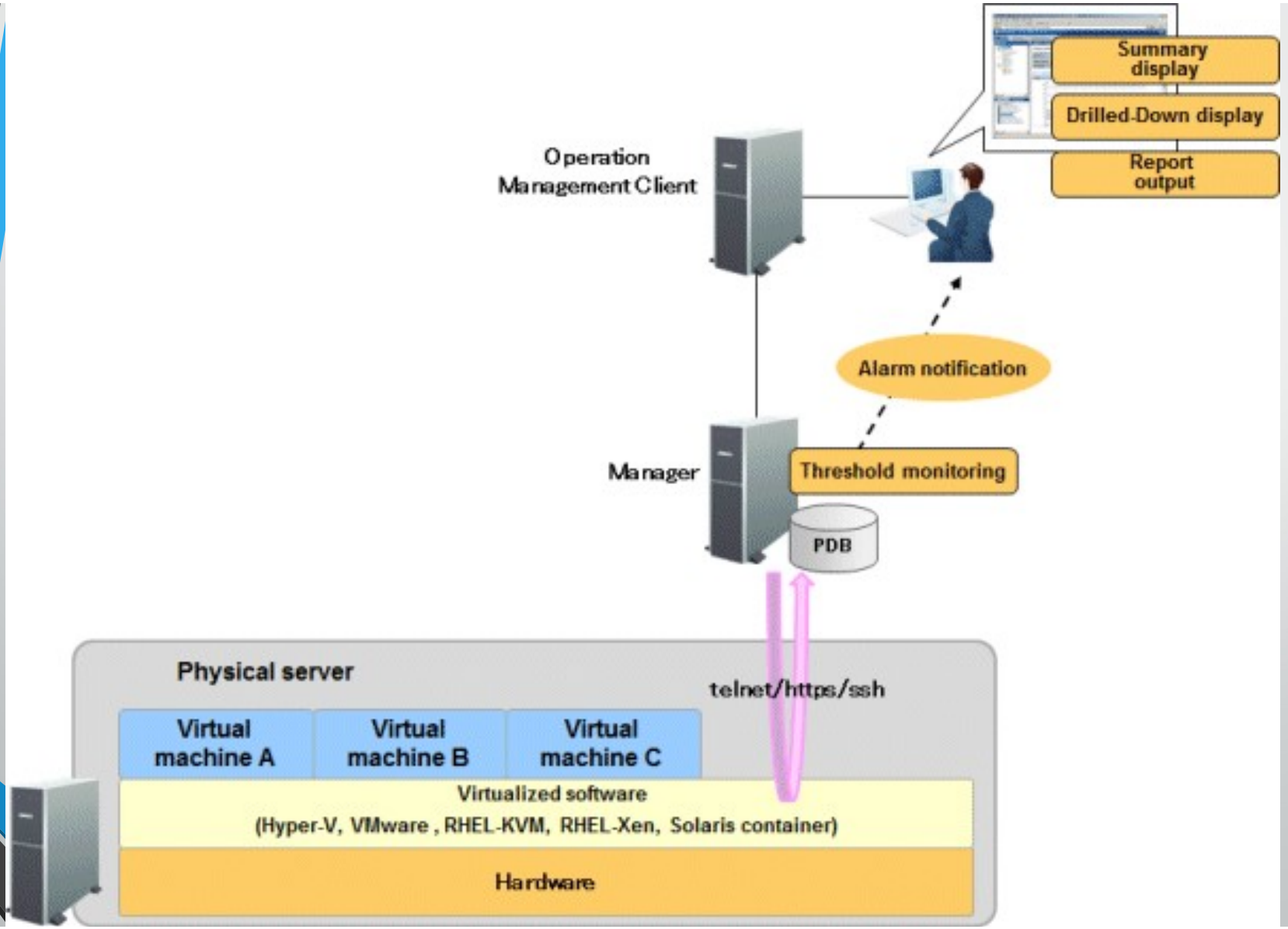
Virtual Resource Management

- Virtual resource management collects performance information **from physical servers** and **virtual machines** of operating systems and virtualized software and centrally manages it.

- Comparing the **virtual machine** performance information collected with this function with performance information about the **physical server**, overarching decisions can be made to optimize the resources in the server and improve use efficiency.

- Performance information for the **physical server** is displayed as a report. This allows usage of the physical server's **CPU, memory,** and **disk** to be seen.

- The virtual machine's performance information is stacked for display in reports for each guest. This allows usage of the CPU, memory, and disk to be seen for each guest.


Summary display
Drilled-Down display
Report output
Operation Management Client
Alarm notification
Manager
Threshold monitoring
PDB
Physical server
telnet/https/ssh
Virtual machine A
Virtual machine B
Virtual machine C
Virtualized software
(Hyper-V, VMware , RHEL-KVM, RHEL-Xen, Solaris container)
Hardware

Performance Monitoring

- Performance information for physical servers and virtual machines is collected by using the agent for Agentless Monitoring functions from virtualized software and connecting remotely through telnet, ssh or https.
- Performance information that can be collected depends on the virtualization software to be monitored.
- The methods for collecting the performance information of physical servers and virtual machines and the main types of performance information collected by the monitoring virtualized software are described in the figure.
- The virtual machine's resources are stacked for display in a report.
- Threshold monitoring can be performed on the different pieces of information and notifications can be sent as alarms when a monitored item exceeds a defined value.

Virtualized software	Physical Server	Virtual Machine
VMware ESX	Performance information for each host (CPU, memory, and disk) is collected from VMware.	Performance information for each guest (CPU, memory, and disk) is collected from VMware.
VMware ESXi		
VMware vCenter	Performance information for each cluster, resource pool, and data store is collected from VMware.	
Hyper-V	CPU performance information is collected from Hyper-V. Memory and disk performance information is collected from the host OS (Windows).	CPU performance information is collected from Hyper-V.
Red Hat virtualization function(KVM)	CPU, memory, and disk performance information is collected from the host OS (Linux).	CPU, memory, and disk performance information is collected from Red Hat virtualization function (KVM).
Red Hat virtualization function (Xen)	CPU, memory, and disk performance information is collected from the host OS (Linux).	CPU, memory, and disk performance information is collected from Red Hat virtualization function (Xen).
Solaris Containers	CPU, memory, and disk performance information is collected from the host OS (Solaris).	CPU and memory performance information is collected from Global Zone.

Cloud Elasticity	Cloud Scalability
Elasticity is used temporarily to handle sudden workload changes.	Scalability helps an application adapt to changing needs by adding or removing resources as needed to meet demands.
Elasticity changes when resource demand goes up or down.	Organizations use scalability to handle increased workloads.
Most small businesses use elasticity when they have sudden increases in demand and workload.	Big companies need to be able to grow and handle their operations well.
It is used temporarily to handle sudden increases in demand or seasonal demands.	Scalability is a technique used to handle a spike in demand when organizations needed.

THANK YOU