Virtualization consists, in computing, of running operating systems1 — we then speak of system virtualization — or applications2 — on a host machine in an isolated environment — we then speak of application virtualization. These virtual machines are called Virtual Private Server (VPS) or Virtual Environment (VE).

Data virtualization is an approach to unify data from multiple sources into a single layer so that applications, reporting tools, and end users can access data without the need for details about the original data source, location, and structures.

Virtualization is a technology that allows you to create multiple simulated environments or dedicated resources from a single physical system. Its software, called a hypervisor, is directly connected to the hardware and allows you to fragment this unique system into several separate secure environments. These are called virtual machines, or VMs. These exploit the hypervisor's ability to separate resources from hardware and distribute them properly. Virtualization helps you get the most out of your previous investments.

Virtualization is a technology that allows you to create useful IT services using resources that are typically hardware-related. It allows you to exploit the full capacity of a physical machine by distributing it among many different users or environments.

Physical hardware with a hypervisor is called a "host," while all virtual machines that use its resources are called "guests." These guests treat compute resources (such as CPU, memory, and storage) as a resource pool that can be moved without difficulty. Operators can control virtual instances of CPU, memory, storage, and other resources, so guests receive the resources they need, when they need them.

* The interests are: optimal use of the resources of a machine park (distribution of virtual machines on physical machines according to the respective loads); Easy installation, deployment and migration of virtual machines from one physical machine to another, especially in the context of going live from a qualification or pre-production environment, easier delivery;économie sur le matériel par mutualisation (consommation électrique, entretien physique, surveillance, support, compatibilité matérielle, etc.)
* installation, tests, développements, cassage et possibilité de recommencer sans casser le système d'exploitation hôte ;

Securing and/or isolating a network (breaking virtual operating systems, but not host operating systems that are invisible to the attacker, testing application and network architectures); isolation of different simultaneous users of the same machine (use of central site type); Dynamic allocation of computing power according to the needs of each application at a given time. Reduction of the risks related to the sizing of servers when defining the architecture of an application, the addition of power (new server, etc.) being then transparent.

Disadvantages Depends on the implementation of the virtual machine. In the case of a raw installation on a single server, the following problems can be observed: access to host server resources through the hardware abstraction layer impairs performance, and running any virtualized software will consume more resources than in native mode; in the event of a failure of a host server, all virtual machines hosted on it will be impacted. But virtualization is often implemented with redundancies, which it facilitates; implementation is complex and requires an initial investment; there are specific administrative constraints (deployment, backup, etc.).

**How does virtualization work? Software, called hypervisors, isolates physical resources from virtual environments, which require those resources. These hypervisors can be based on an operating system (laptop, for example) or be installed directly on a physical system (such as a server), which is the option most often chosen by companies that use virtualization. Hypervisors distribute your physical resources to allow virtual environments to use.Types de ressources virtualisées**

**Virtualisation des données**

**Data scattered across an environment can be aggregated into a single source. Data virtualization enables businesses to use data as a dynamic source. As a result, they benefit from processing capabilities that can bring together data from multiple sources, easily host new data sources, and transform data to meet user needs. Data virtualization tools span multiple data sources that can then be treated as a single source. They thus make available the necessary data, in the required form, at the appropriate time, for any application or any user.**

Desktop Virtualization

Often confused with operating system virtualization, which allows you to deploy multiple operating systems on a single machine, desktop virtualization allows a central administrator (or automated administration tool) to deploy simulated desktop environments to hundreds of physical machines at the same time. Unlikeenvironnements de postes de travail classiques que vous devez installer, configurer et mettre à jour physiquement sur each machine, virtualized desktops can be configured, updated and verified simultaneously by an administrator

**Server virtualization**

**Servers are computers designed to handle a high volume of specific tasks to allow other computers (laptops and desktops, for example) to perform various other tasks. Virtualizing a server optimizes the execution of these specific functions and involves partitioning it, so that the components can be used to perform different functions.Virtualisation des systèmes d'exploitation**

* Virtualization of operating systems occurs at the kernel level, which corresponds to the central task manager of the latter. This approach makes it possible to run Linux and Windows environments side by side. Enterprises can also transfer virtual operating systems to computers, which has the following advantages:

**Reduce hardware costs, as computers do not require a high number of out-of-the-box features Enhance security through the ability to monitor and isolate virtual instances Reduce time spent on IT services, such as software updates**

Virtualization of network functions

Network Functions Virtualization (NFV) separates key functions of a network (such as directory services, file sharing, and IP address configuration) to distribute them across environments. When software functions are independent of the physical machines on which they were hosted, it is possible to group specific functions into a new network and assign them to an environment. Network virtualization reduces the number of physical components required to create multiple independent networks, such as switches, routers, servers, cables, and hubs. It is particularly prevalent in the telecommunications sector.

What is a hypervisor?  
A hypervisor is software that allows you to create and run virtual machines. A hypervisor isolates its operating system and resources from virtual machines, and allows you to create and manage these virtual machines.

Physical hardware used as a hypervisor is called a "host," while all virtual machines that use its resources are called "guests."

**There are two types of hypervisors that can be used to virtualize resources: type 1 hypervisors and type 2 hypervisors.Type 1**

A Type 1

hypervisor, also known as a bare or native system hypervisor, runs directly on the host's hardware to manage guest operating systems. It takes the place of the host's operating system and directly schedules virtual machine resources on the hardware.

**This type of hypervisor is frequently used in enterprise data centers and other server-based environments. KVM, Microsoft Hyper-V, and VMware vSphere are examples of Type 1 hypervisors. The KVM solution was integrated into the Linux kernel in 2007. If you're using a recent version of Linux, you already have access to KVM.**

**Type 2**

A Type 2 hypervisor, also known as a hosted hypervisor, runs on a traditional operating system as a software layer or application.

It works by separating guest operating systems from the host operating system. Virtual machine resources are scheduled at the level of a host operating system, which in itself runs on the hardware.

Installing a Type 2 hypervisor is suitable for users who want to run multiple operating systems on a personal computer.

VMware Workstation and Oracle VirtualBox are examples of Type 2 hypervisors.

Containers and virtual machines Overall, containers and virtual machines look very similar. Both are packet computing environments that combine various components and isolate them from the rest of a system. Their main differences lie in the way they evolve and in their level of portability.

A container is a process or set of processes that are isolated from the rest of the system. The container allows the process to access only the volume of resources that has been specified. These resource limits ensure that the container can be run on a node that has sufficient capacity.

Virtual machines integrate their own operating system and can thus perform several resource-intensive functions simultaneously. With the large volumes of resources they have access to, virtual machines can unbundl, separate, duplicate, and emulate servers, operating systems, desktops, databases, and entire networks.Un hyperviseur vous permet également d'exécuter plusieurs systèmes d'exploitation dans une machine virtuelle, while containers can only run one type of operating system. For example, a container running on a Linux server will only be able to run a Linux operating system.

**Containers are sometimes seen as replacements for hypervisors. This is not entirely accurate, since** **containers and virtualization meet different needs.Virtualisation**

**Software called a "hypervisor" separates resources from physical machines in order to partition them and allocate them to virtual machines. When a user issues a virtual machine instruction that requires additional resources from the physical environment, the hypervisor forwards the request to the physical system and caches the changes. Virtual machines look and act as physical servers, which has disadvantages in terms of application dependencies and the size of the operating system footprint (most often oversized to run a single application or microservice).**

**containers**

Containers contain a microservice or application, as well as everything needed to run it. All the contents of a container are preserved in the form of an image. It is a code file that includes all libraries and dependencies. These files are comparable to the installation files of a Linux distribution, because the image comes with all its RPM packages and configuration files. Since containers are very small, they are usually several hundred, loosely coupled, and for this reason they are managed and provisioned by container orchestration platforms such as Red Hat OpenShift and Kubernetes.

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Proxmox VE is a platform to run virtual machines and containers. It is based on Debian Linux, and completely open source. For maximum flexibility, Proxmox implemented two virtualization technologies - Kernel-based Virtual Machine (KVM) and container-based virtualization (LXC). One main design goal was to make administration as easy as possible. You can use Proxmox VE on a single node, or assemble a cluster of many nodes. All management tasks can be done using our web-based management interface, and even a novice user can setup and install Proxmox VE within minutes.

Docker is a platform for managing the delivery of distributed applications in lightweight, portable, self sufficient containers ν Containers are an abstraction of capabilities built into the Linux kernel.

Container images become containers at runtime and in the case of Docker containers - images become containers when they run on [Docker Engine](https://www.docker.com/products/container-runtime). Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Kubernetes is an open source system for managing containerized applications across multiple hosts. ν It provides basic mechanisms for deployment, maintenance, and scaling of applications.

[Kubernetes](https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/), also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

The name Kubernetes originates from Greek, meaning helmsman or pilot. K8s as an abbreviation results from counting the eight letters between the "K" and the "s". Google open-sourced the Kubernetes project in 2014. Kubernetes combines [over 15 years of Google's experience](https://kubernetes.io/blog/2015/04/borg-predecessor-to-kubernetes/) running production workloads at scale with best-of-breed ideas and practices from the community.

A Kubernetes cluster consists of a set of worker machines, called [nodes](https://kubernetes.io/docs/concepts/architecture/nodes/), that run containerized applications. Every cluster has at least one worker node.

The worker node(s) host the [Pods](https://kubernetes.io/docs/concepts/workloads/pods/) that are the components of the application workload. The [control plane](https://kubernetes.io/docs/reference/glossary/?all=true#term-control-plane) manages the worker nodes and the Pods in the cluster. In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.

Control Groups: isolate processes within a group limiting resources they can access

Podman is a OCI compliant serverless container system developed in 2017.Podman is rootless containers, it means:

Being able to create and execute containers without the need of elevated privileges is a major improvement in Podman over Docker

Namespaces are a feature of the Linux kernel that partitions kernel resources such that one set of processes sees one set of resources and another set of processes sees a different set of resources. The feature works by having the same namespace for a group of resources and processes, but those namespaces refer to distinct resources

The physical complex is the same idea as a computer. Two namespaces (or more) can reside on the same physical computer, and much like the apartment building, namespaces can either share access to certain resources or have exclusive access

LXC is a userspace interface for the Linux kernel containment features. Through a powerful API and simple tools, it lets Linux users easily create and manage system or application containers

Current LXC uses the following kernel features to contain processes:

* Kernel namespaces (ipc, uts, mount, pid, network and user)
* Apparmor and SELinux profiles
* Seccomp policies
* Chroots (using pivot\_root)
* Kernel capabilities
* CGroups (control groups)

LXC containers are often considered as something in the middle between a chroot and a full fledged virtual machine. The goal of LXC is to create an environment as close as possible to a standard Linux installation but without the need for a separate kernel.

Vagrant is a tool for building and managing virtual machine environments in a single workflow. With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases production parity, and makes the "works on my machine" excuse a relic of the past.

Vagrant provides easy to configure, reproducible, and portable work environments built on top of industry-standard technology and controlled by a single consistent workflow to help maximize the productivity and flexibility of you and your team.

To achieve its magic, Vagrant stands on the shoulders of giants. Machines are provisioned on top of VirtualBox, VMware, AWS, or [any other provider](https://www.vagrantup.com/docs/providers/). Then, industry-standard [provisioning tools](https://www.vagrantup.com/docs/provisioning/) such as shell scripts, Chef, or Puppet, can automatically install and configure software on the virtual machine.