

A hypervisor, also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs). A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing.

Default containers list: <https://hub.docker.com/>

Docker concepts

Docker is a platform for developers and sysadmins to **build, run, and share** applications with containers. The use of containers to deploy applications is called *containerization*. Containers are not new, but their use for easily deploying applications is.

Containerization is increasingly popular because containers are:

* **Flexible**: Even the most complex applications can be containerized.
* **Lightweight**: Containers leverage and share the host kernel, making them much more efficient in terms of system resources than virtual machines.
* **Portable**: You can build locally, deploy to the cloud, and run anywhere.
* **Loosely coupled**: Containers are highly self sufficient and encapsulated, allowing you to replace or upgrade one without disrupting others.
* **Scalable**: You can increase and automatically distribute container replicas across a datacenter.
* **Secure**: Containers apply aggressive constraints and isolations to processes without any configuration required on the part of the user.

Images and containers

Fundamentally, a container is nothing but a running process, with some added encapsulation features applied to it in order to keep it isolated from the host and from other containers. One of the most important aspects of container isolation is that each container interacts with its own private filesystem; this filesystem is provided by a Docker **image**. An image includes everything needed to run an application - the code or binary, runtimes, dependencies, and any other filesystem objects required.

Containers and virtual machines

A container runs *natively* on Linux and shares the kernel of the host machine with other containers. It runs a discrete process, taking no more memory than any other executable, making it lightweight.

By contrast, a **virtual machine** (VM) runs a full-blown “guest” operating system with *virtual* access to host resources through a hypervisor. In general, VMs incur a lot of overhead beyond what is being consumed by your application logic.

|  |  |
| --- | --- |
| Container stack exampleVirtual machine stack example |  |