

**Vagrant** não é um Hypervisor, ele é uma ferramenta para controlar um Hypervisor.

Hypervisor

Um *Hypervisor*, também conhecido como monitor de máquina virtual, é um processo que cria e executa máquinas virtuais (VMs). Um *Hypervisor* permite que um computador *host* suporte múltiplas VMs, compartilhando virtualmente seus recursos, como memória e processamento.

Exemplos de *Hypervisors* são:

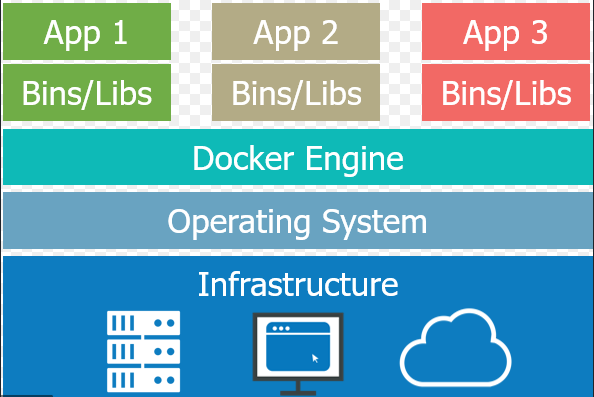
* Hyper-V
* vSphere
* Parallels
* VMware
* Virtualbox
* entre outros.

Existem dois *tipos de Hypervisors*: **Tipo 1** e **Tipo 2**.

Os *Hypervisors* do **Tipo 1** são chamados de "bare metal", pois são executados diretamente no hardware do *host*. Exemplos disso são **Hyper-V** e **vSphere** (entre vários outros).

Os *Hypervisors* do **Tipo 2** rodam como uma aplicação em cima do sistema operacional. Exemplos são o **VirtualBox** e **VMware**.

* **VirtualBox**, **VMware**, **Hyper-V**, entre outros, são ***Hypervisors***
* Um *Hypervisor* emula o hardware do computador para criar e executar máquinas virtuais
* O **Vagrant** é uma ferramenta que controla o *Hypervisor* a partir de um arquivo simples, o **Vagranfile**
* O **Vagrantfile** define detalhes da máquina virtual, como o sistema operacional, a rede, software utilizado, etc
* O comando vagrant init <box> cria um **Vagrantfile**
* A box é baixada da internet e possui a imagem do sistema operacional, entre outras configurações
* Para inicializar e rodar a VM com Vagrant, usa-se o comando: vagrant up
* O comando vagrant status mostra detalhes sobre o status da máquina virtual
* Para se conectar com a máquina virtual, usamos a ferramenta SSH



**= LINUX**

IINUX

Ao invés de trabalharmos com um **hipervisor**, utilizamos o recurso de **Container Engine** representado pelo **Docker**, o qual depende do Linux no SO host.

O Docker é uma tecnologia para criar, rodar e administrar containers, baseado no Linux

Containers virtualizam o sistema operacional

Máquinas virtuais virtualizam o hardware

Com este sistema não é preciso que os programas configurados possuam suas próprias bibliotecas, pois se conectam diretamente à fonte através dos containers, que já possuem uma biblioteca de dados carregada no sistema operacional, tornando este processo mais leve e rápido.

No caso do Windows, há poucas versões que comportam os containers automaticamente, tornando pouco acessível.