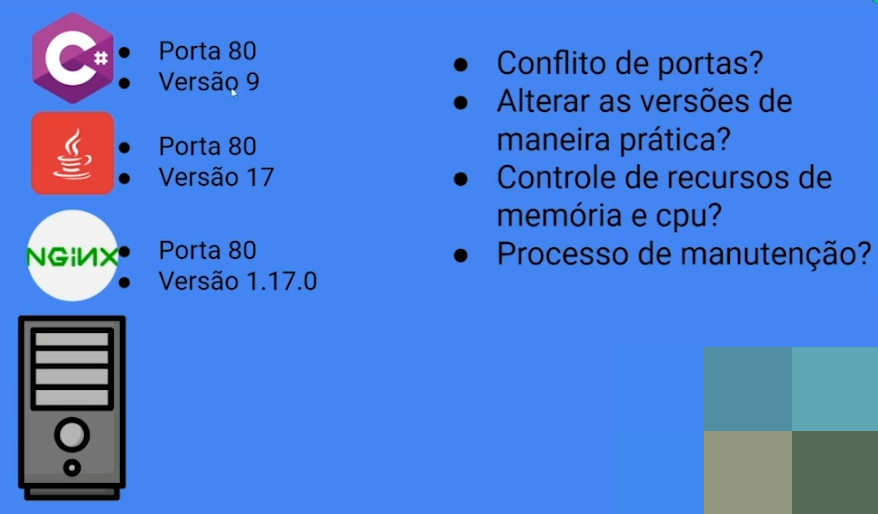
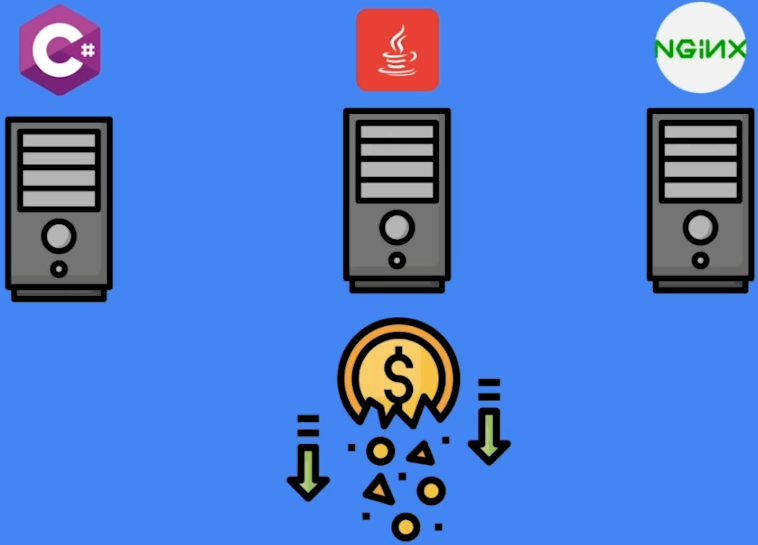
**Docker - criando e gerenciando containers Docker criando e gerenciando containers**

# 1- Conhecendo o problema

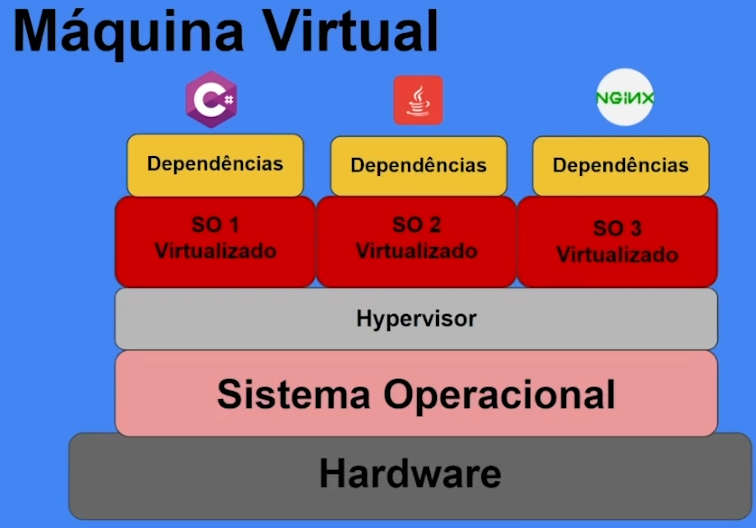
Supondo que em nosso sistema precisa ter diversas aplicações, as seguintes questões podem surgir



Uma possível solução seria comprar várias máquinas, uma para cada aplicação, porém acaba sendo inviável com o tempo por conta do alto custo.

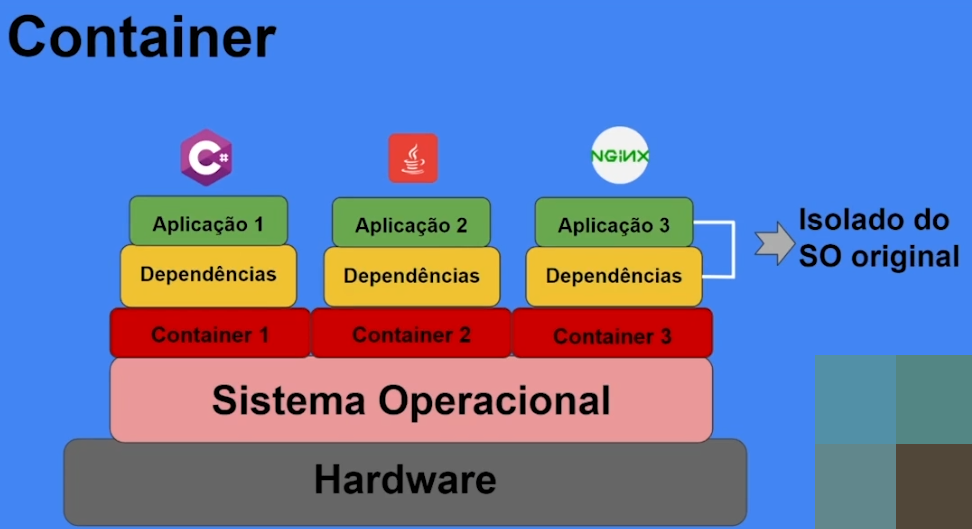


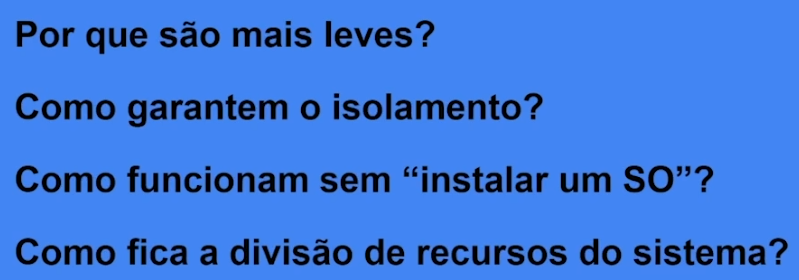
Outra solução é a utilização de máquinas virtuais. A ideia é utilizar um intermediador (Hypervisor) para virtualizar um SO com todas as configurações necessárias para a nossa aplicação. Com isso temos um isolamento dos SOs instalado com o SO principal.



Mas a dúvida é: realmente necessário tudo isso? A camada de Hypervisor é necessária pra o nosso sistema como um todo?

Utilizando Containers não temos mais a camada de Hypervisor e SO

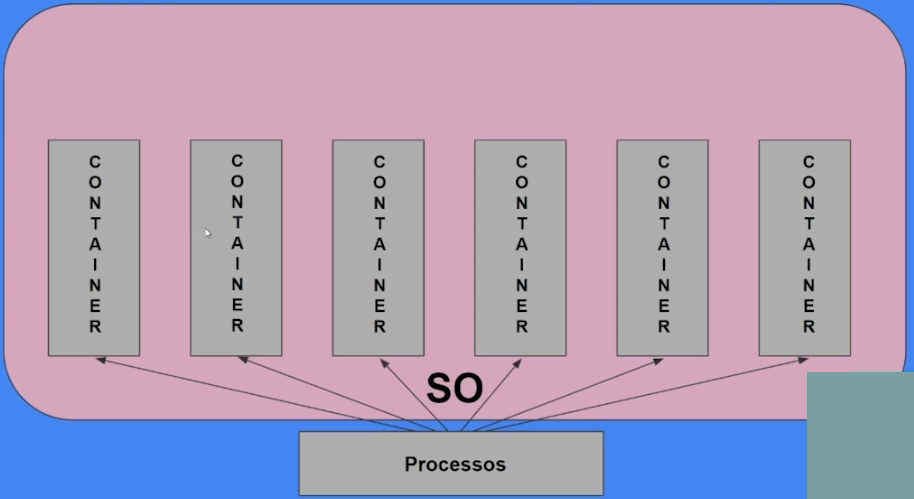




## Como containers funcionam

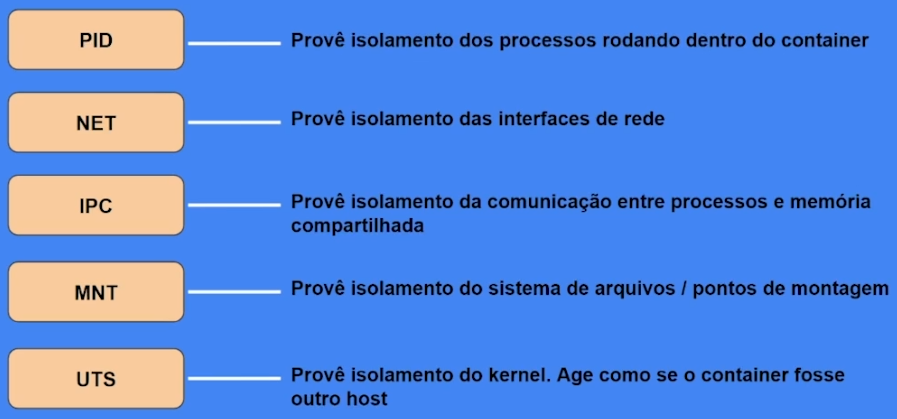
**Por que são mais leves?**

Os containers são executados diretamente como processos dentro de nosso SO, sem a necessidade de uma máquina virtual.



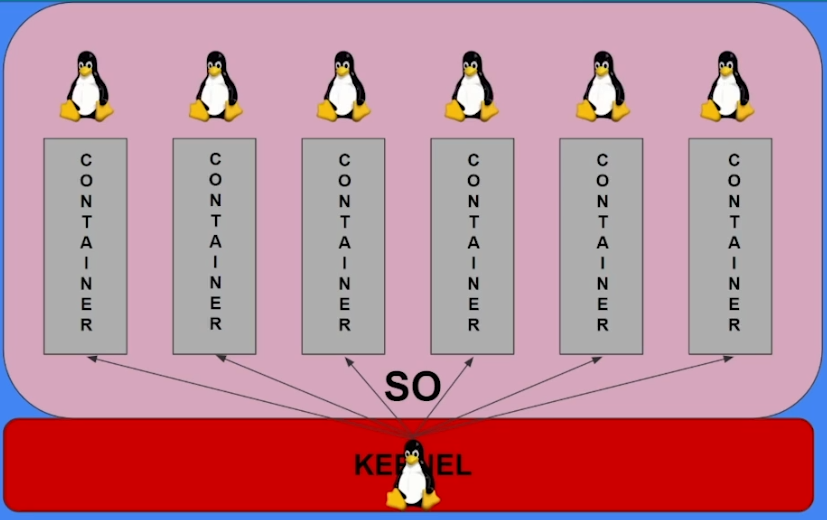
**Como garantem o isolamento?**

Quando os containers estiverem em execução dentro de nosso SO, a fim de garantir o isolamento, é utilizado o conceito de Namespaces, onde os containers são isolados em níveis.



**Como funcionam sem “instalar um SO”?**

Graças ao Namespace de UTS, caso tenhamos executando os nossos containers em uma máquina que tenha um Kernel Linux, cada um dos containers, a princípio, também terá um “pedaço” desse Kernel, porém, isolado.



**Como fica a divisão de recursos do sistema?**

Utilizando o conceito de Cgroups, podemos definir, de forma manual ou automática, como os consumos serão feitos para cada um dos processos (containers) dentro de nosso SO.

