

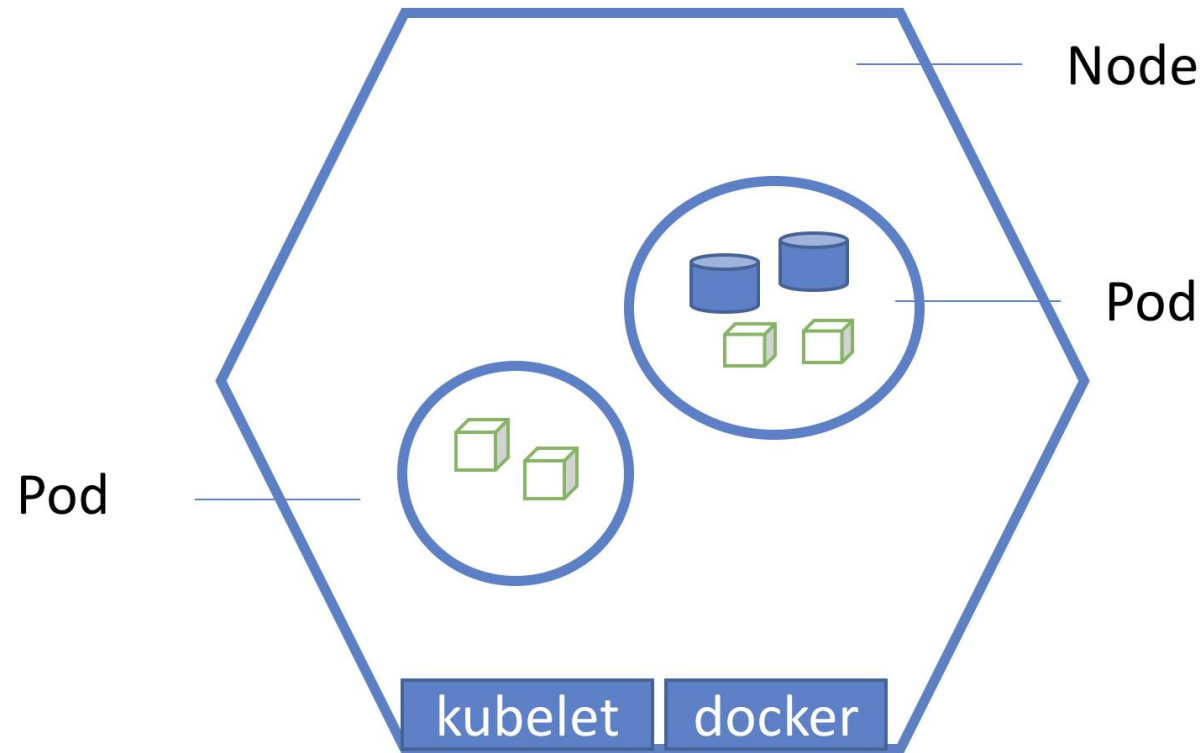


Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Conhecendo o POD em Kubernetes



Conhecendo o POD em Kubernetes



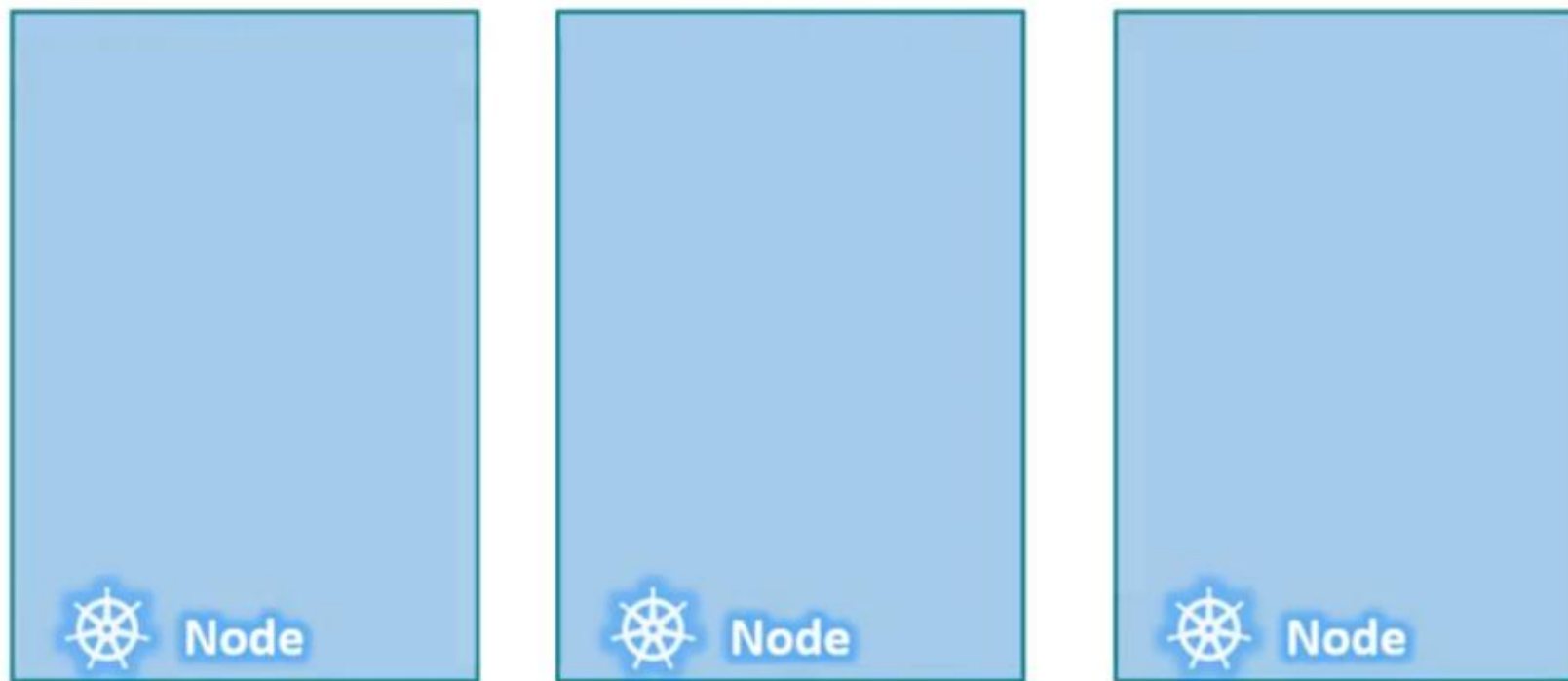
POD em Kubernetes é a menor unidade de computação publicável em um computador que podemos criar e gerenciar.



Conhecendo o POD em Kubernetes



Sabemos que quando trabalhamos com Kubernetes é criado um cluster onde temos o Master Node e o(s) Worker Node(s)

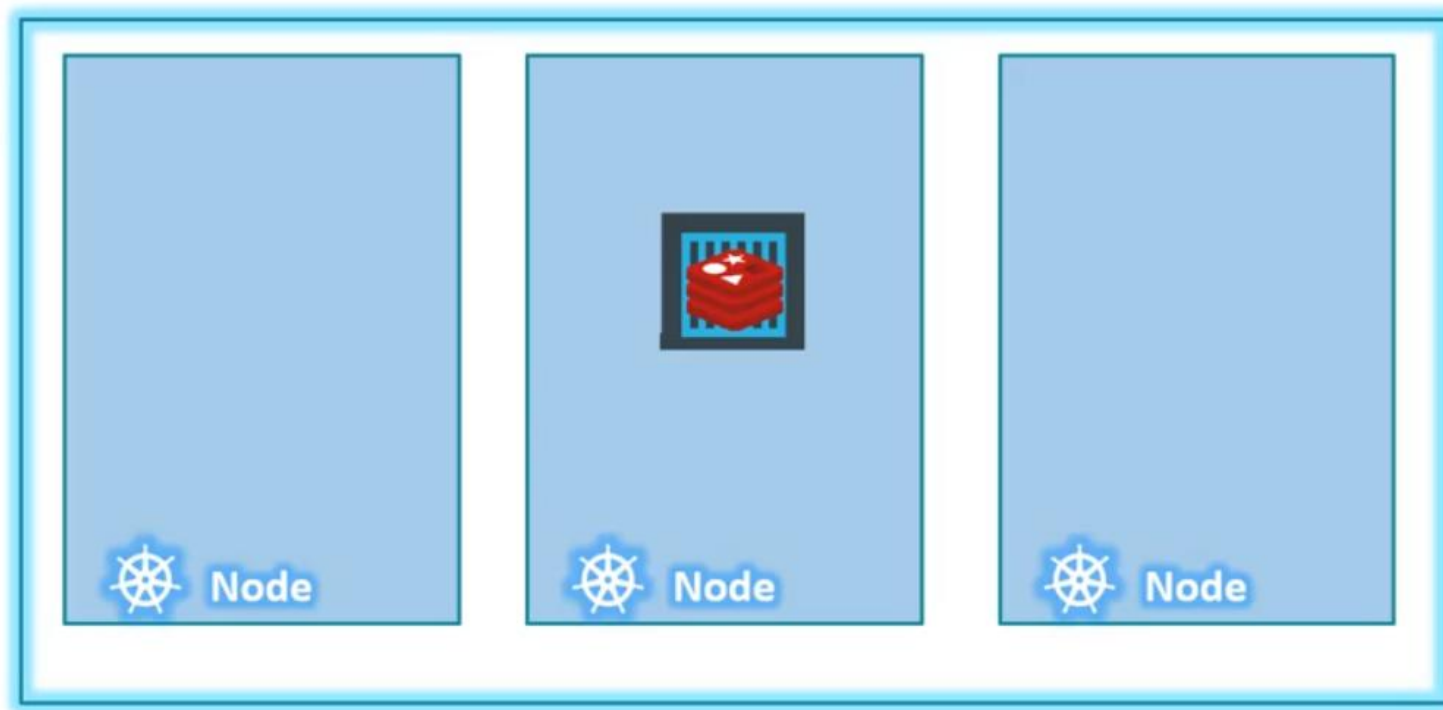


fonte: <https://kubernetes.io/docs/concepts/workloads/pods/>

Conhecendo o POD em Kubernetes



Quando criamos uma aplicação em um container o Kubernetes não faz uso/acesso direto ao container.

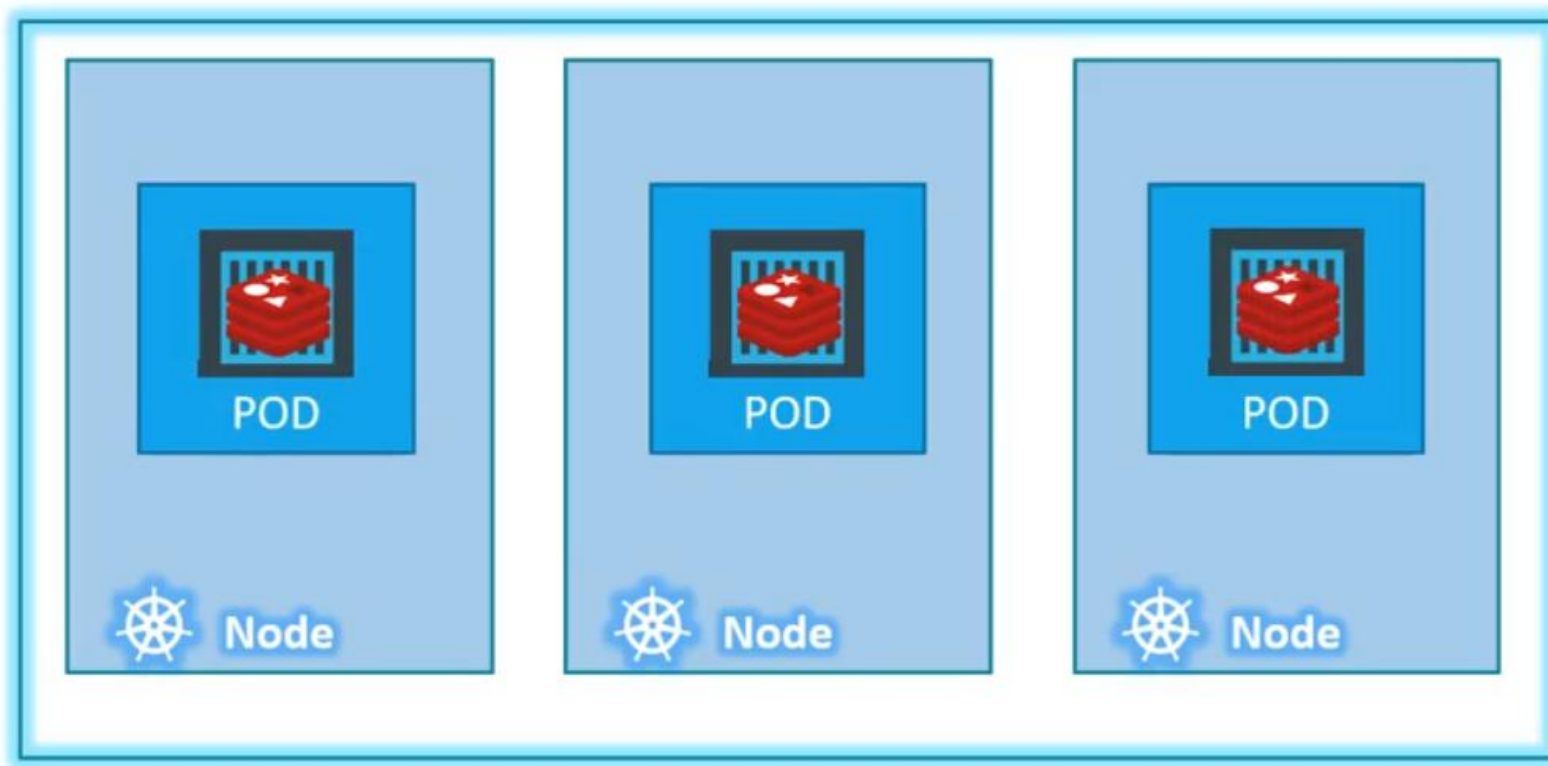


fonte: <https://kubernetes.io/docs/concepts/workloads/pods/>

Conhecendo o POD em Kubernetes



Os containers são “encapsulados” em um objeto do Kubernetes chamado POD.

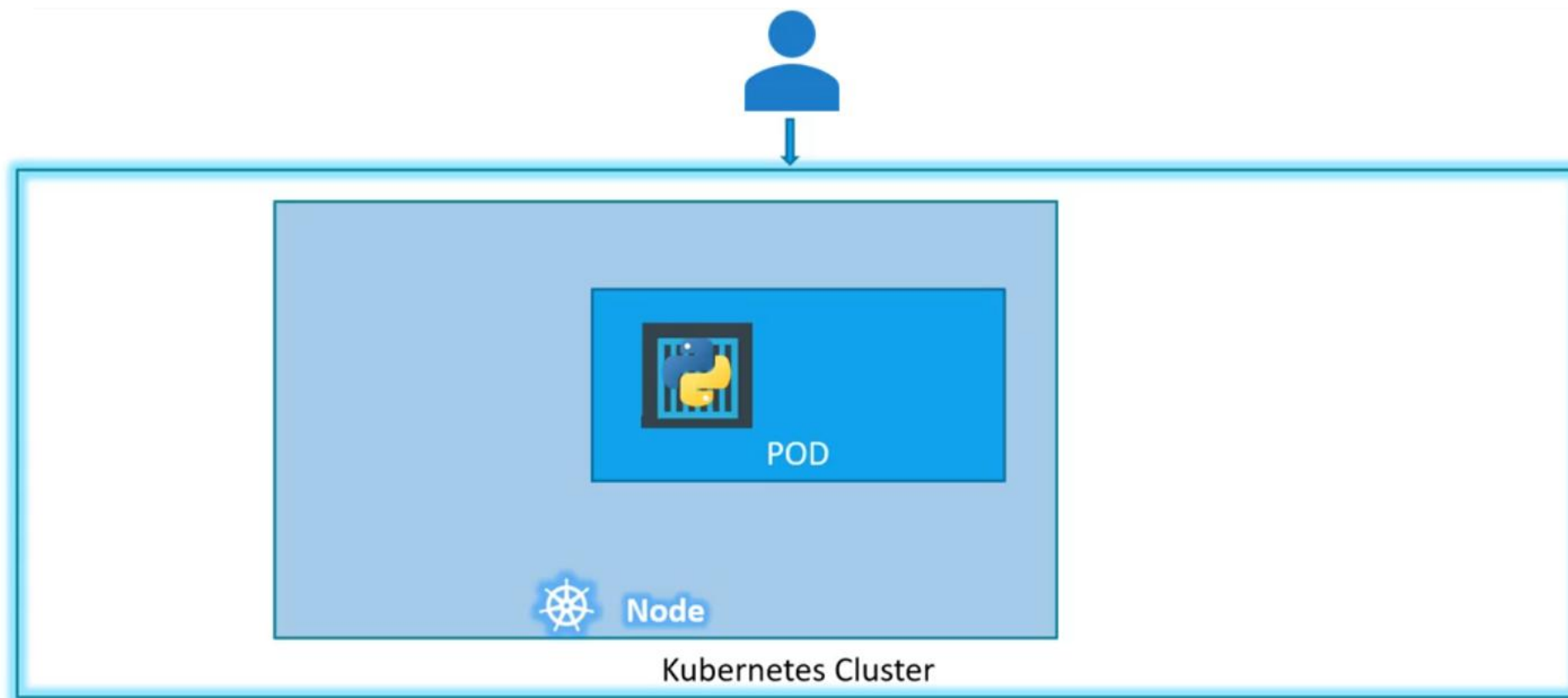


Desta forma, em Kubernetes, um pod é uma instância única de uma aplicação.

Conhecendo o POD em Kubernetes



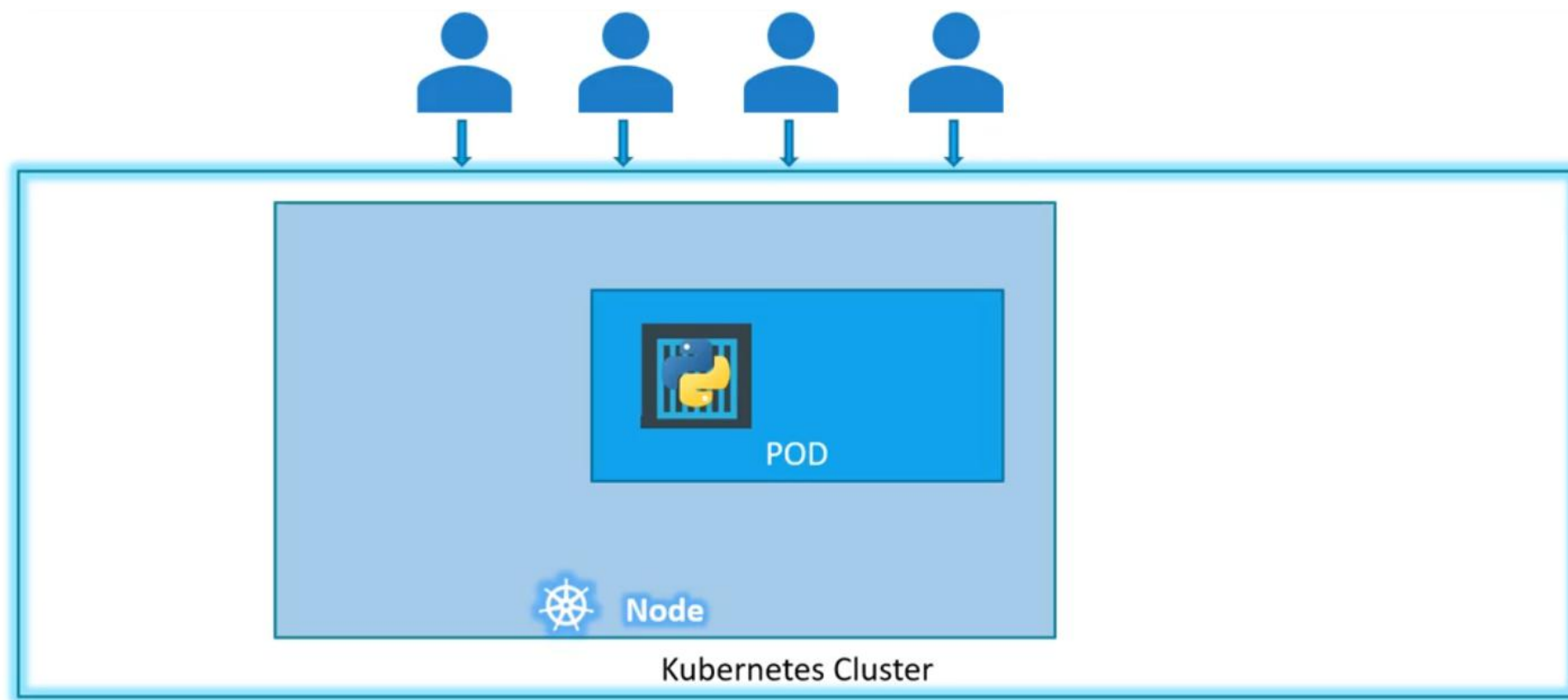
Exemplo de uma aplicação Python sendo executada em um único POD com o cluster Kubernetes.



Conhecendo o POD em Kubernetes



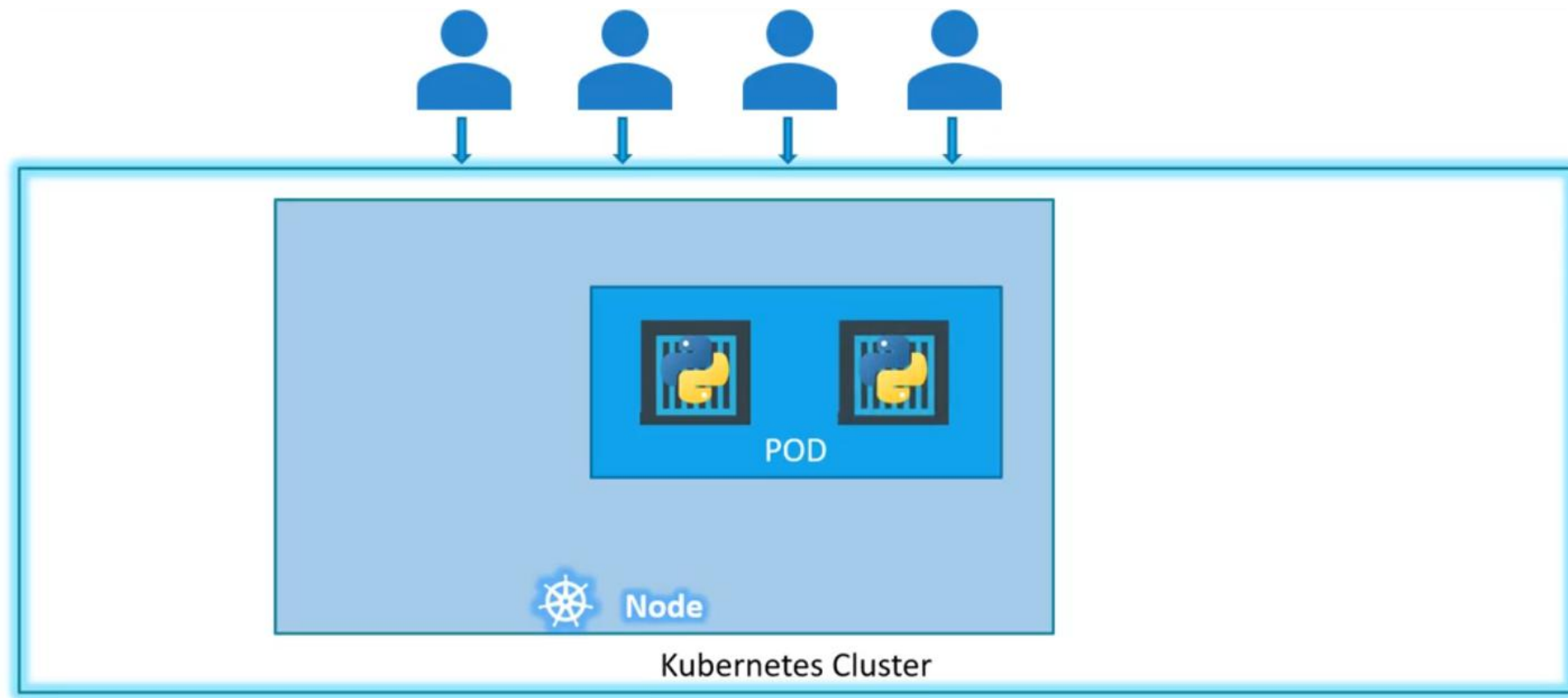
Mas o que ocorreria se sua aplicação fosse um sucesso e tivesse vários acessos?
O que o Kubernetes faria para “dividir” a carga?



Conhecendo o POD em Kubernetes



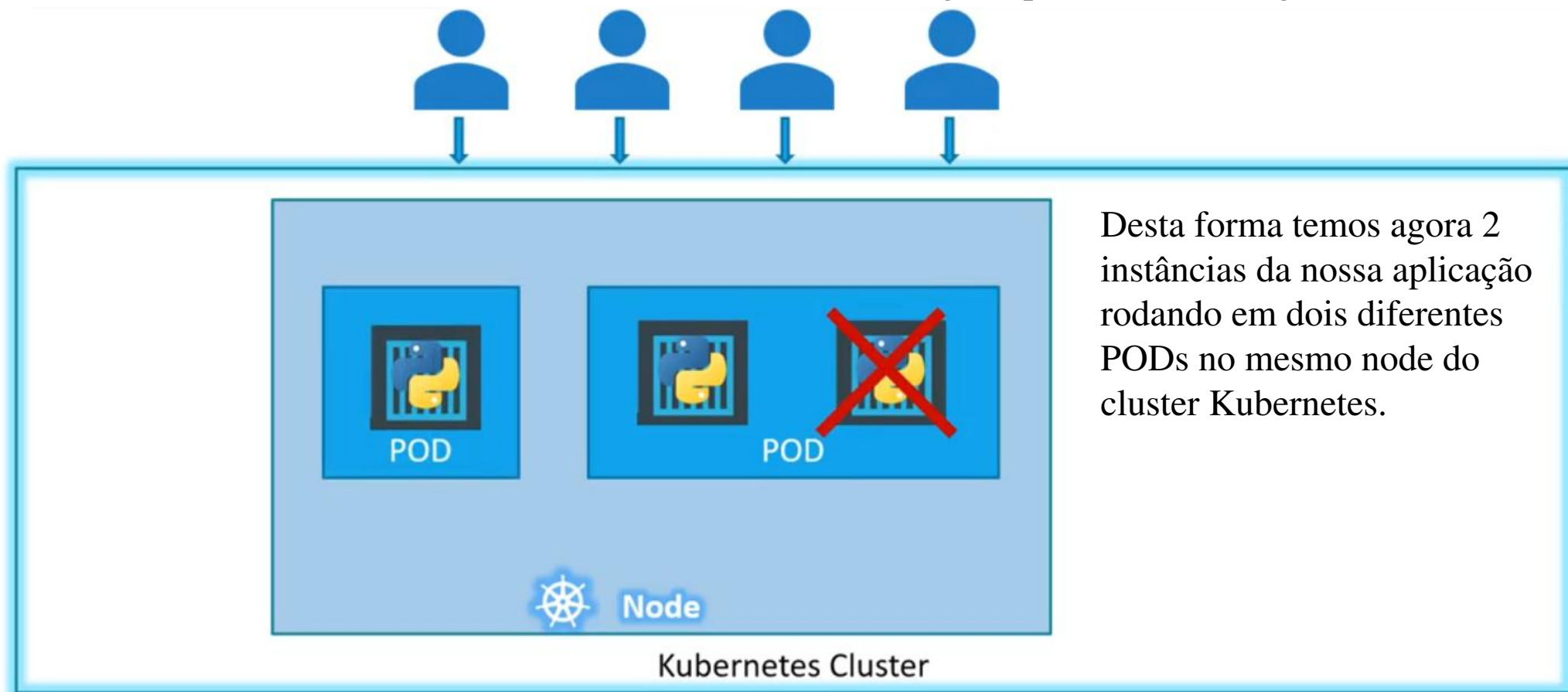
O Kubernetes iria instanciar um novo container no POD?



Conhecendo o POD em Kubernetes



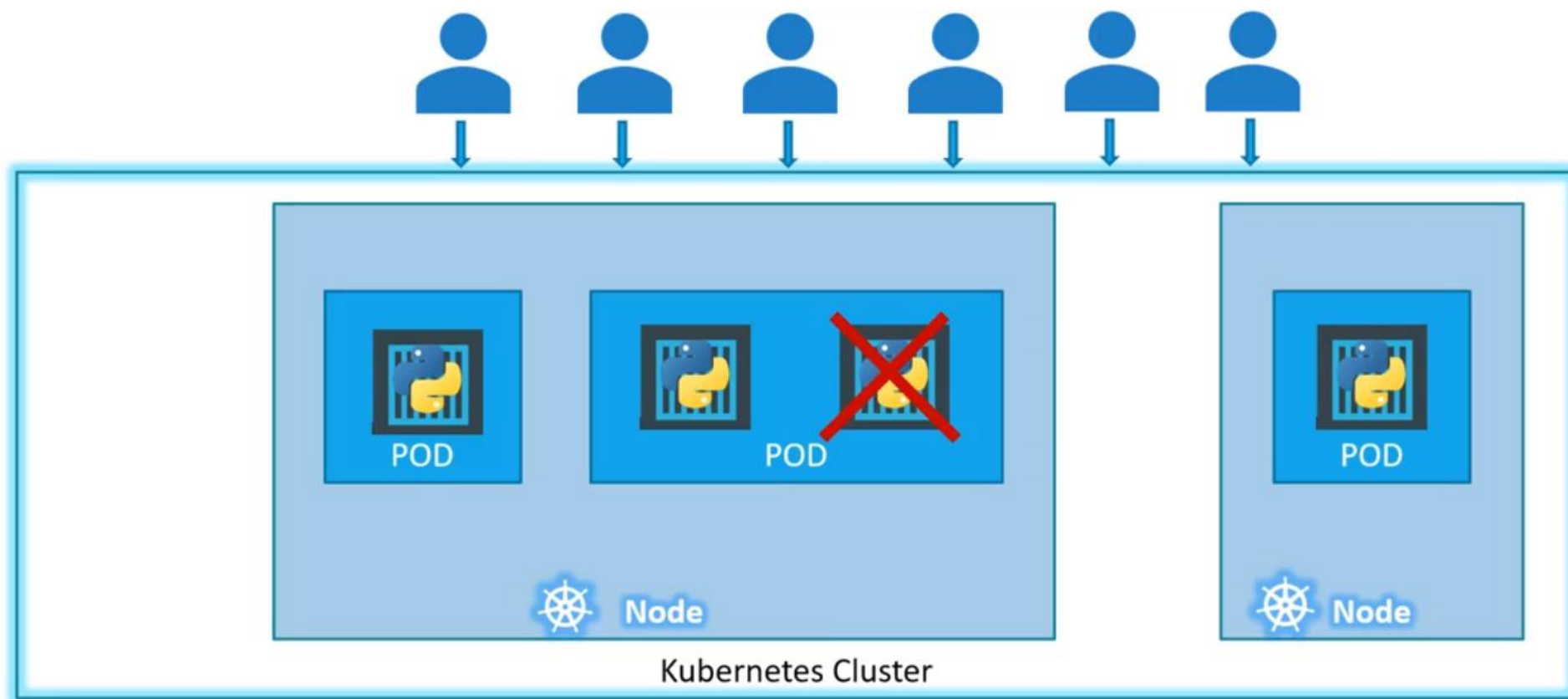
O Kubernetes vai criar um novo POD com um container idêntico ao original para dividir a carga do sistema.



Conhecendo o POD em Kubernetes



Se a quantidade de acessos à sua aplicação continuar a crescer e seu node não suportar novas instâncias da aplicação, ainda podemos adicionar quantos novos nodes forem necessários ao cluster para dividir a carga.



Conhecendo o POD em Kubernetes



Se eu pudesse escolher uma informação para que você pudesse sempre se lembrar do que aprendemos nesta aula seria:

Em Kubernetes, a relação POD/Container é 1:1

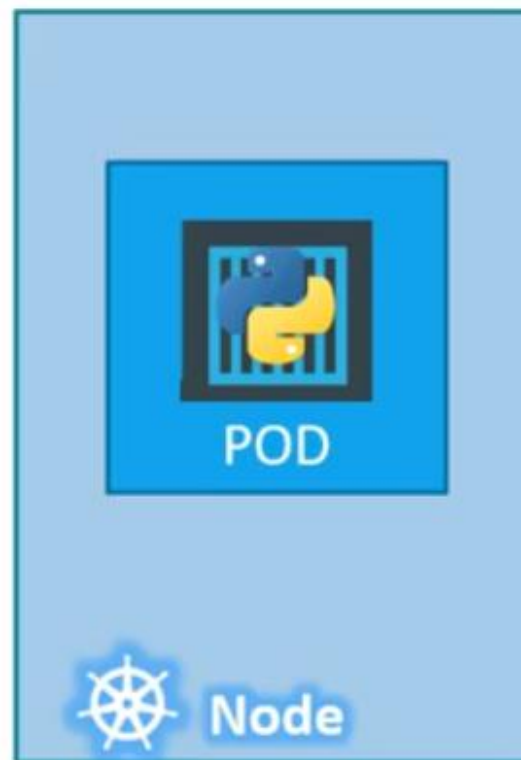
Ou seja, para 1 container temos 1 POD.

Para escalar up sua aplicação, você cria um novo POD, que irá instanciar um container idêntico ao já existente, e para escalar down você deleta o POD.

Ou seja, você não adiciona diretamente um novo container para escalar sua aplicação, o POD irá fazer isso.

OBS: Não se preocupe que iremos aprender a fazer isso tudo na prática em breve.

fonte: <https://kubernetes.io/docs/concepts/workloads/pods/>

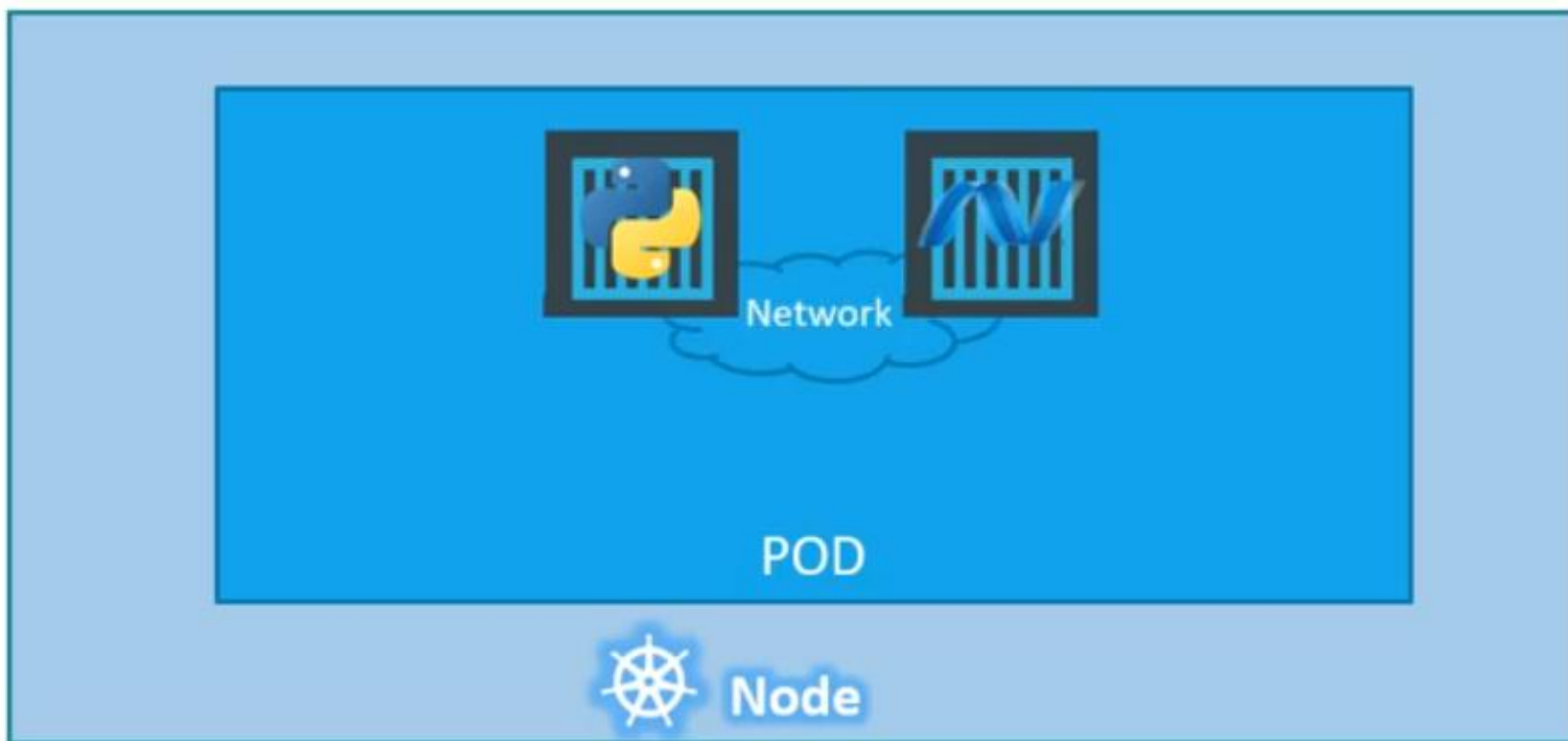


Conhecendo o POD em Kubernetes



PODs Multi-Container

Vimos a pouco que a relação POD/Container é 1:1, porém em alguns raros casos podemos ter a necessidade de uma aplicação fazer uso de um container helper/auxiliar.



Os containers se comunicam via “localhost” já que fazem parte da mesma rede e podem inclusive acessar dados de armazenamento em conjunto.

Conhecendo o POD em Kubernetes



Kubectl

Fazemos uso do kubectl para criar “pods” no nosso cluster, conforme o exemplo:

```
kubectl run nginx --image nginx
```

OBS: O comando acima irá criar um POD chamado ‘ngnx’ no nosso cluster* Kubernetes com um container através da imagem informada via parâmetro. Esta imagem funciona exatamente conforme aprendemos no curso de Docker, ou seja, o Docker irá procurar localmente e caso não encontre irá baixar do Docker Hub.

* *minikube start*



Conhecendo o POD em Kubernetes



Kubectl

Com o comando para criação do POD executado podemos fazer uso do kubectl para acessar informações dos PODs existentes conforme:

kubectl get pods

```
geek@university:~$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	0/1	ContainerCreating	0	8s

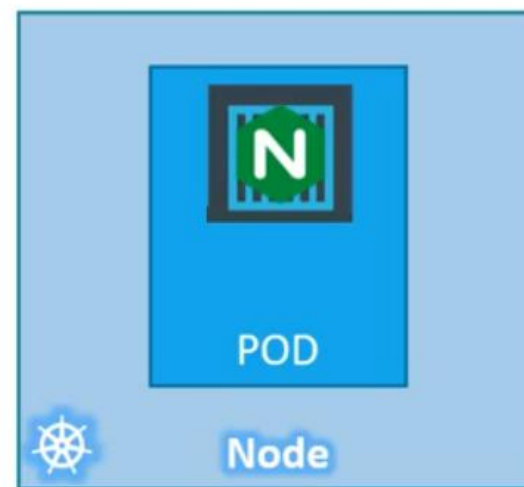
```
geek@university:~$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	1/1	Running	0	46s

OBS: Note que temos diferentes status para o pod.

Neste momento o nosso servidor Web NGINX não está acessível para usuários. Apenas internamente no container está acessível. Isso porque ainda não aprendemos os conceitos de redes, mas fique tranquilo que iremos aprender e praticar.

fonte: <https://kubernetes.io/docs/concepts/workloads/pods/>





Geek University

Evolua seu lado geek!

www.geekuniversity.com.br