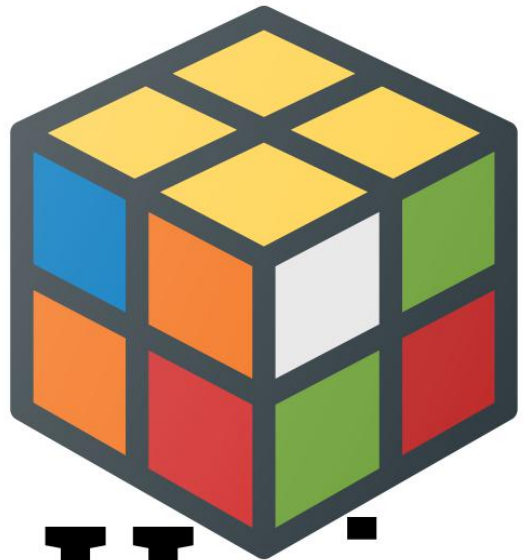


Geek University

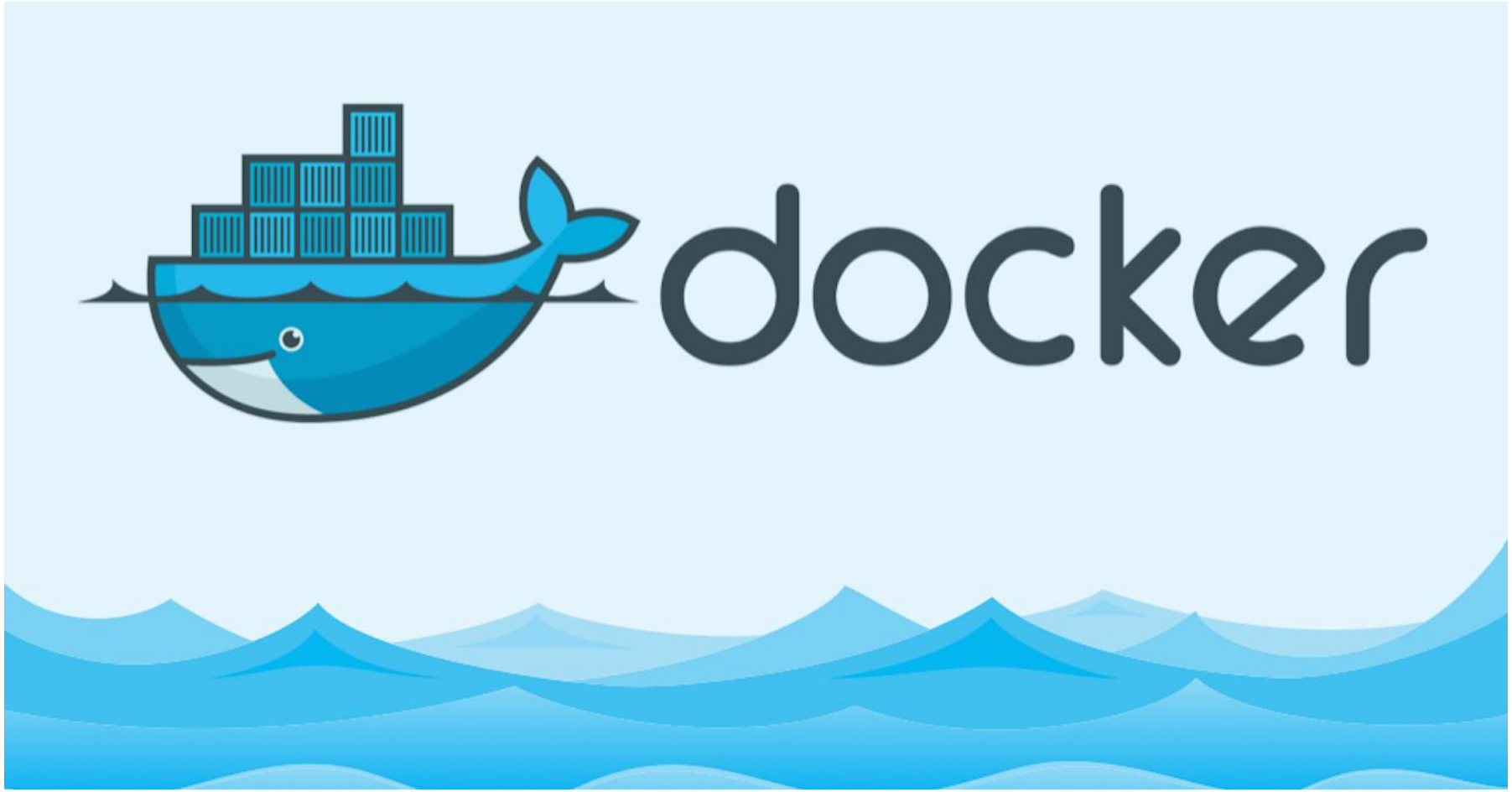
Evolua seu lado geek!

www.geekuniversity.com.br

Funcionamento básico de redes no Kubernetes

Funcionamento básico de redes no Kubernetes

Rápido review Redes no Docker



Funcionamento básico de redes no Kubernetes

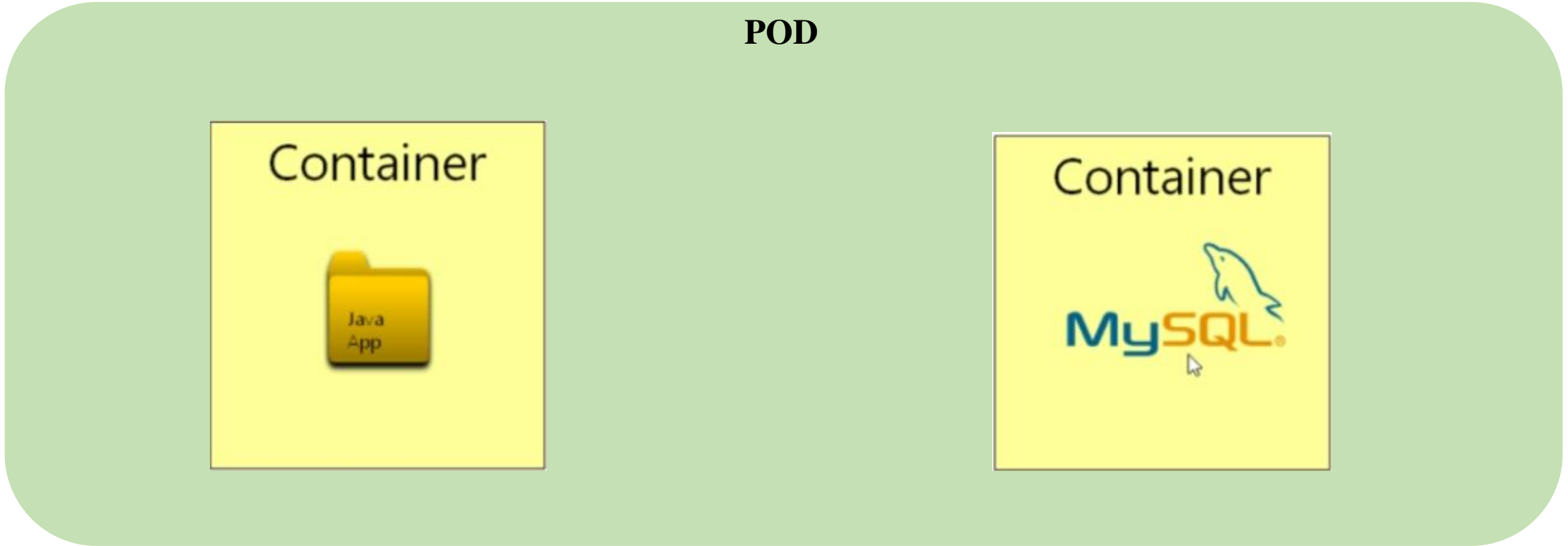
Rápido review Redes no Docker

Com Docker, aprendemos que idealmente cada serviço deve ser criado no seu próprio container.

Por exemplo, aqui temos 2 containers, sendo 1 contendo a aplicação Java e no outro o banco de dados MySQL.

Funcionamento básico de redes no Kubernetes

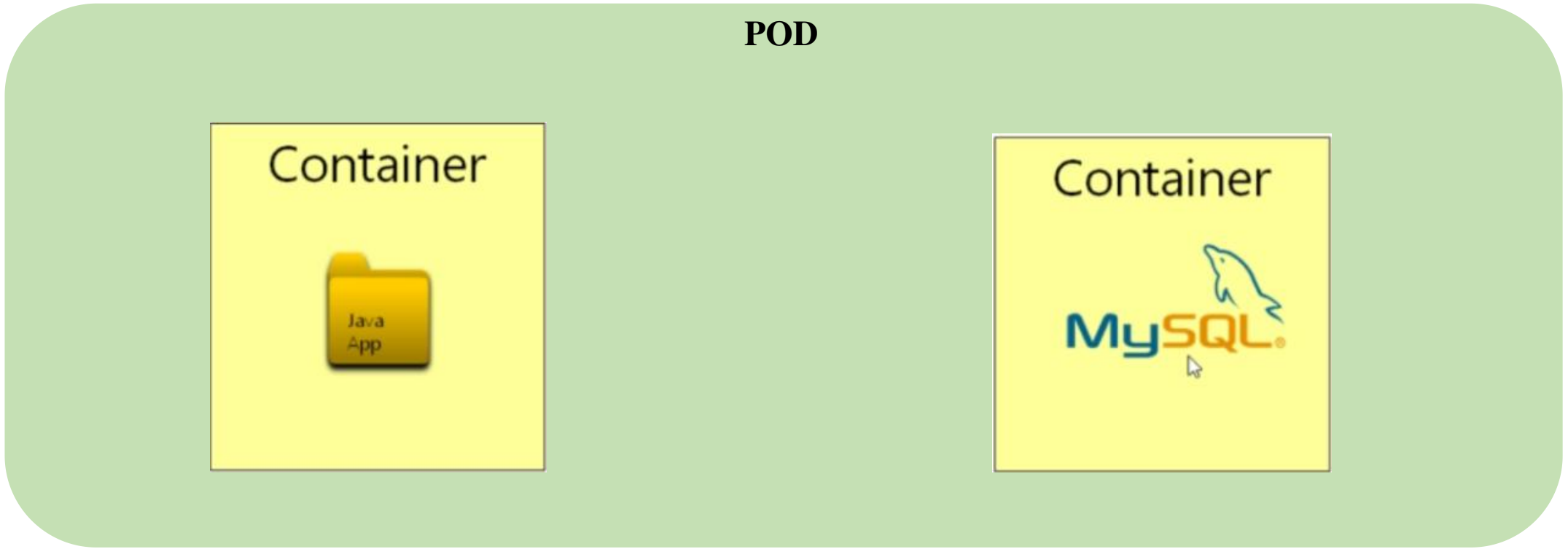
Trazendo a mesma ideia aqui para o Kubernetes, se tivermos um único pod no cluster, tudo fica fácil...



Funcionamento básico de redes no Kubernetes

ATENÇÃO: Esta é apenas uma explicação sobre conceitos básicos de redes no Kubernetes. Nunca crie um POD contendo dois serviços como neste exemplo, no caso um serviço web e um serviço banco de dados.

Caso este pod falhe, você terá mais trabalho para descobrir se o problema está em um serviço ou no outro.

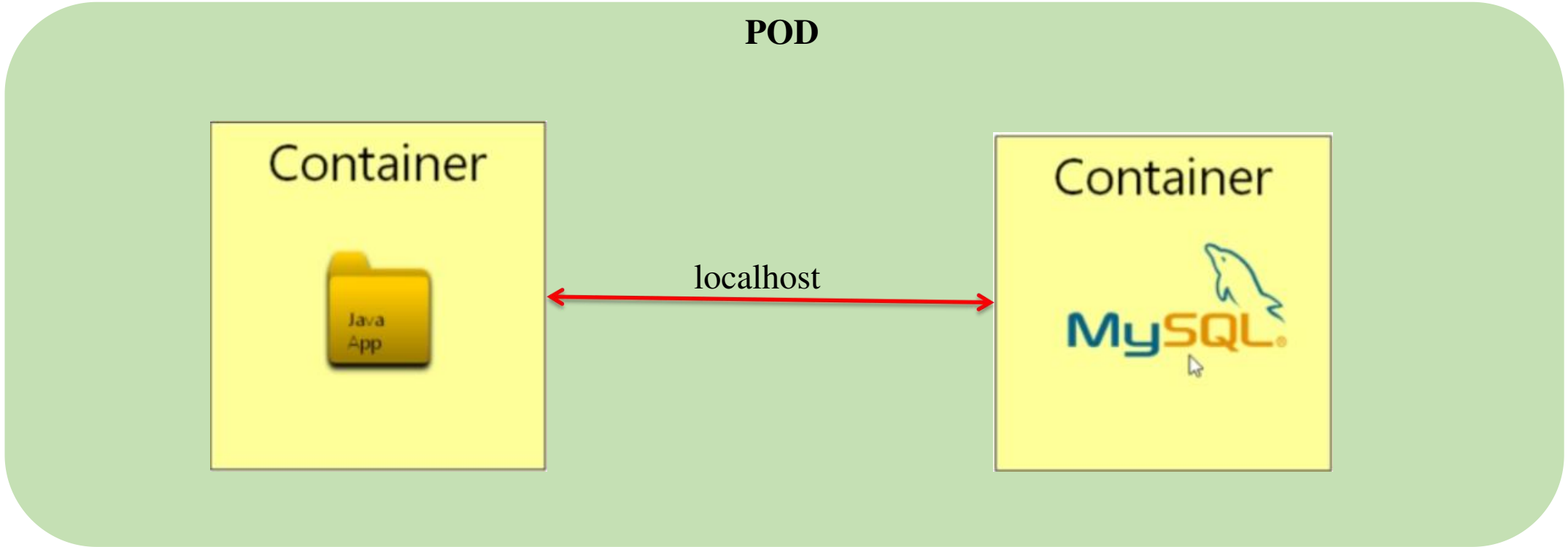


Funcionamento básico de redes no Kubernetes

Trazendo a mesma ideia aqui para o Kubernetes, se tivermos um único pod no cluster, tudo fica fácil...

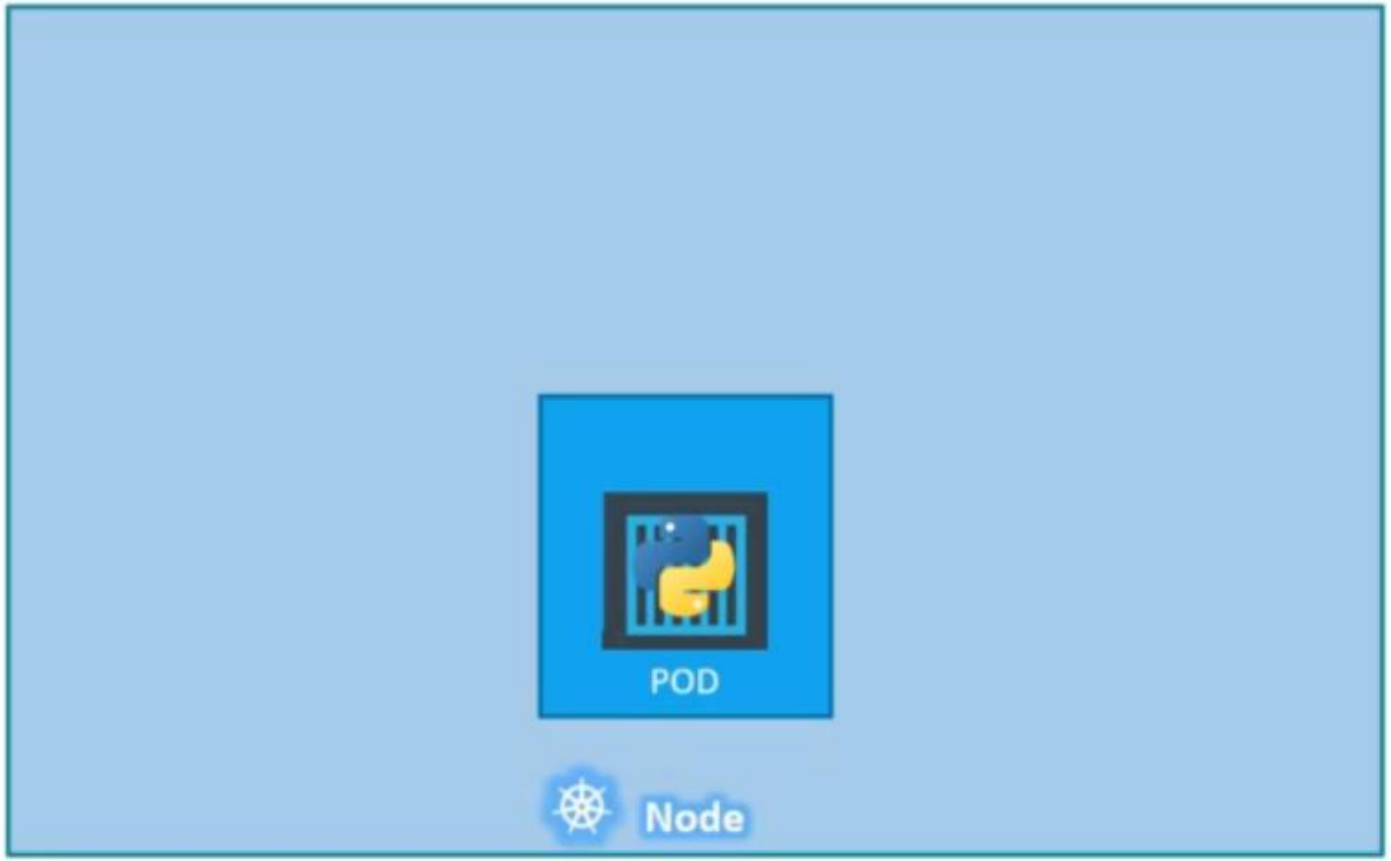
...pois os containers podem se comunicar entre eles através de localhost.

Vamos entender o por quê...



Funcionamento básico de redes no Kubernetes

Para um entendimento simples e direto do funcionamento de redes quando utilizamos Kubernetes vamos partir do princípio que temos um cluster (master node / node worker) e neste cluster temos 1 pod (o conceito vale para infinitos pods).

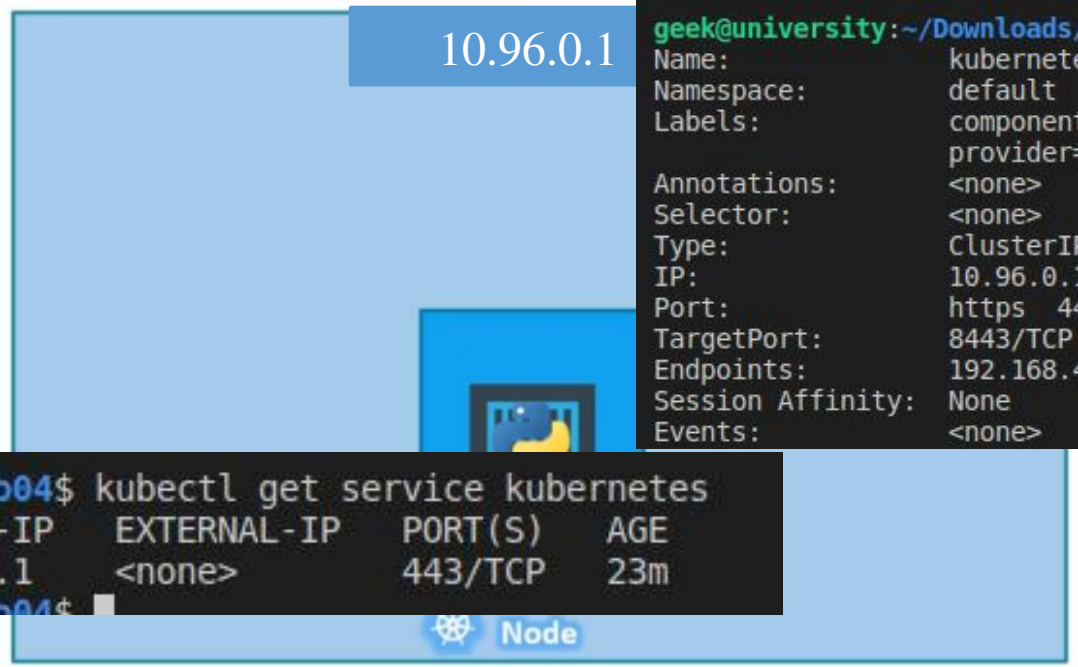


Funcionamento básico de redes no Kubernetes

O cluster “nasce” com dois endereços IP, sendo um do próprio cluster (10.96.0.1) e outro da apiServer (192.168.49.2)

```
geek@university:~/Downloads/secao04$ kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   11m
geek@university:~/Downloads/secao04$
```

OBS: É o endereço da apiServer que usamos para acessar o cluster via SSH e etc.



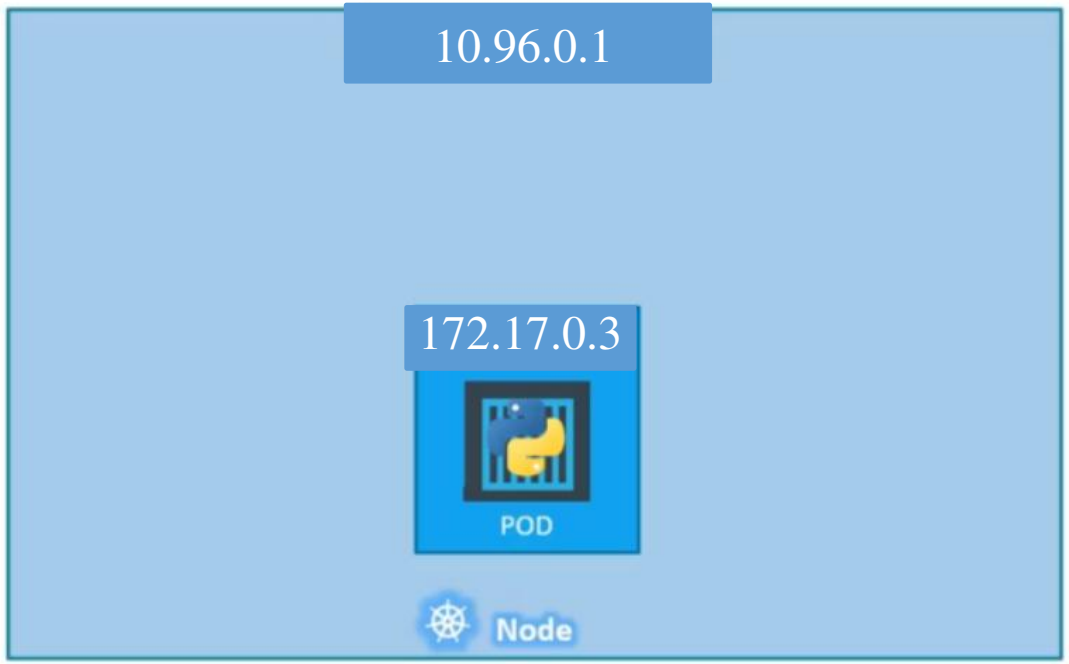

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

geek@university:~/Downloads/secao04$ kubectl describe service kubernetes
Name:              kubernetes
Namespace:         default
Labels:            component=apiserver
                   provider=kubernetes
Annotations:       <none>
Selector:          <none>
Type:              ClusterIP
IP:                10.96.0.1
Port:              https  443/TCP
TargetPort:        8443/TCP
Endpoints:         192.168.49.2:8443
Session Affinity:  None
Events:            <none>
```

```
geek@university:~/Downloads/secao04$ kubectl get service kubernetes
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   23m
```

Funcionamento básico de redes no Kubernetes

Aqui entra um dos pontos que difere o Kubernetes em relação ao Docker.

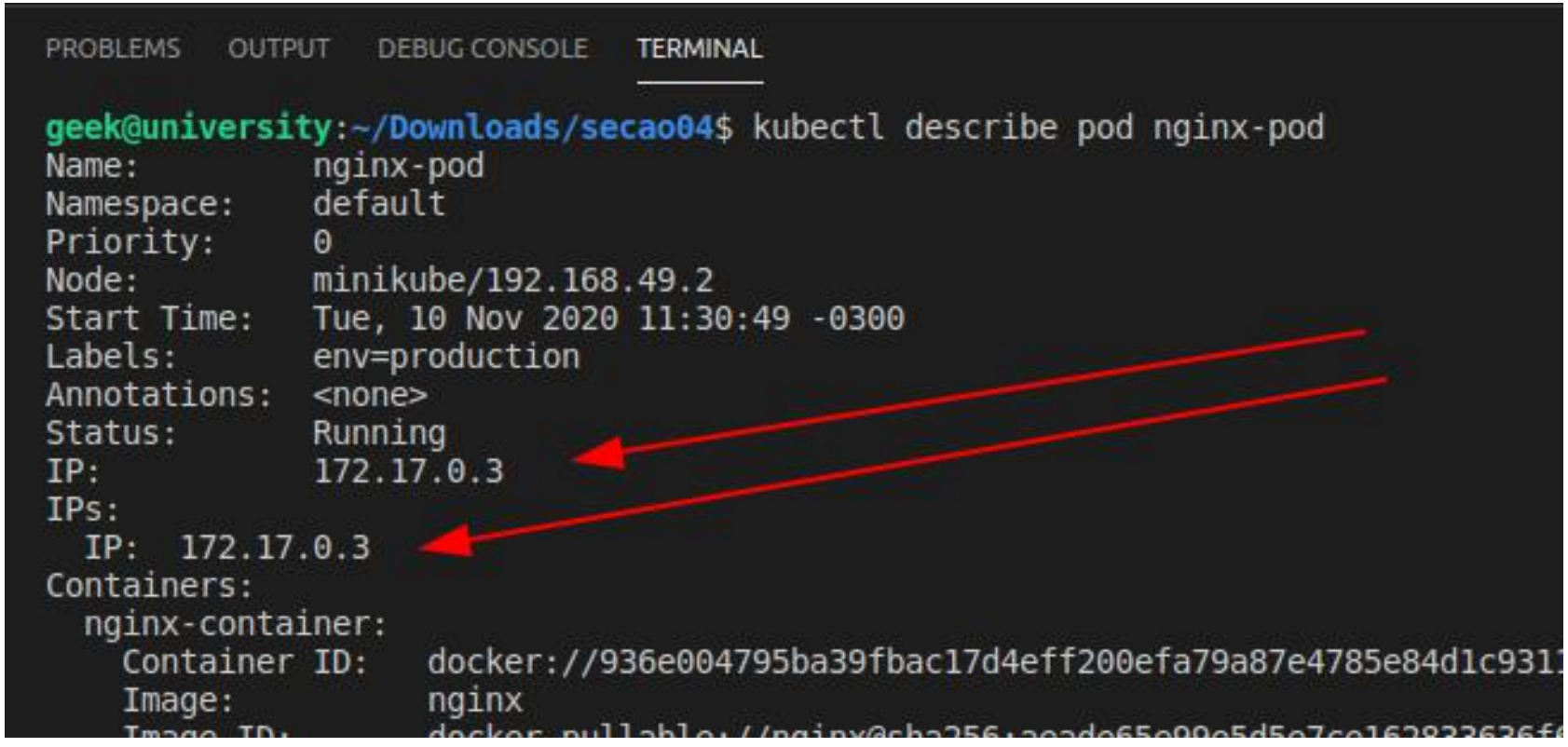
Enquanto no Docker aprendemos que o container recebe um endereço IP, fazendo uso do Kubernetes quem recebe o endereço IP é o POD.



Funcionamento básico de redes no Kubernetes

Aqui entra um dos pontos que difere o Kubernetes em relação ao Docker.

Enquanto no Docker aprendemos que o container recebe um endereço IP, fazendo uso do Kubernetes quem recebe o endereço IP é o POD.



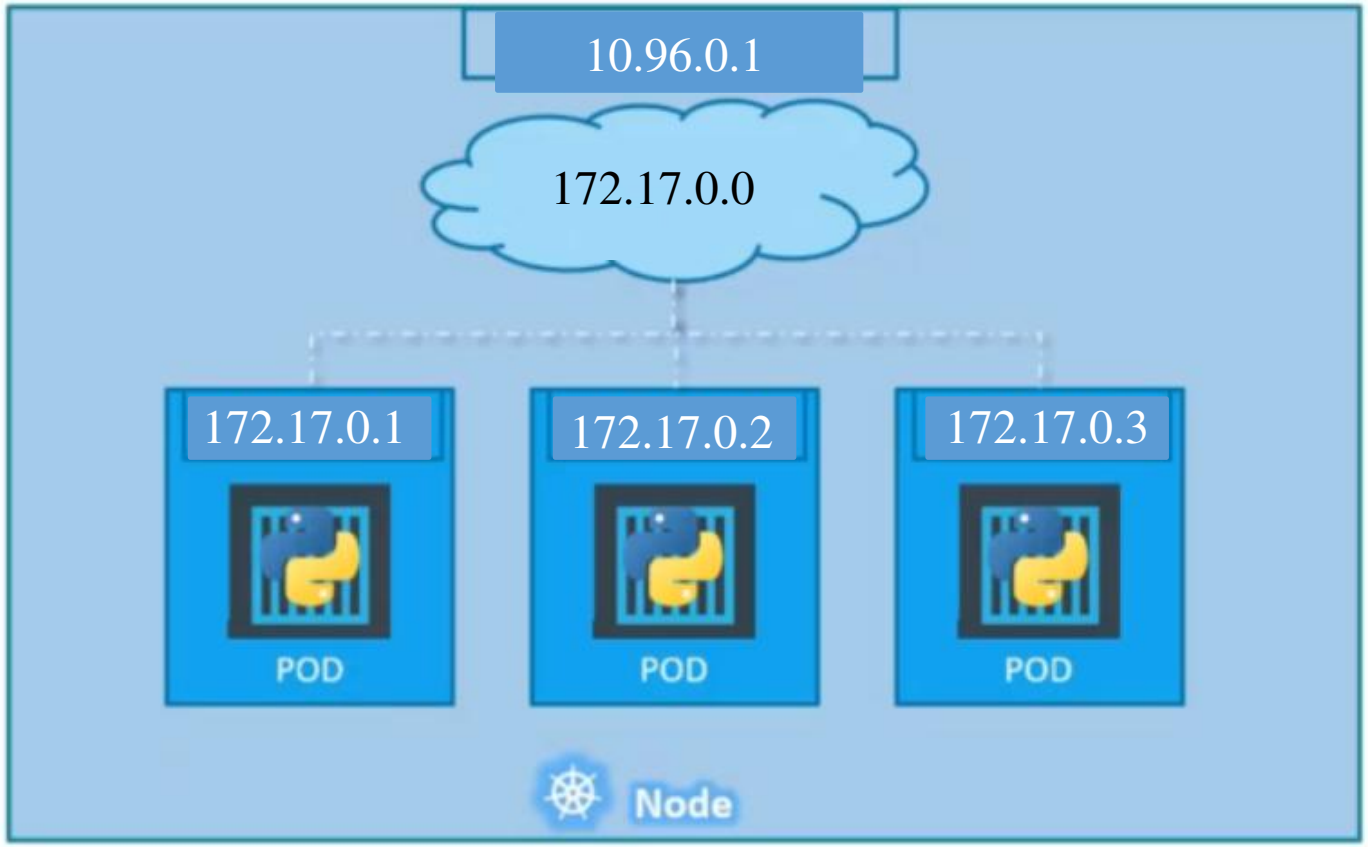
Funcionamento básico de redes no Kubernetes

Quando criamos um cluster, além do IP principal do cluster e do IP da apiServer, é criada uma rede que irá endereçar os PODs a medida que são criados.

OBS: Note que temos até então o uso das classes A, B e C de redes e isso pode variar de acordo com a implementação do Kubernetes a ser utilizada e também de acordo com o sistema operacional.

O importante aqui é você entender como ocorre a comunicação entre os objetos Kubernete

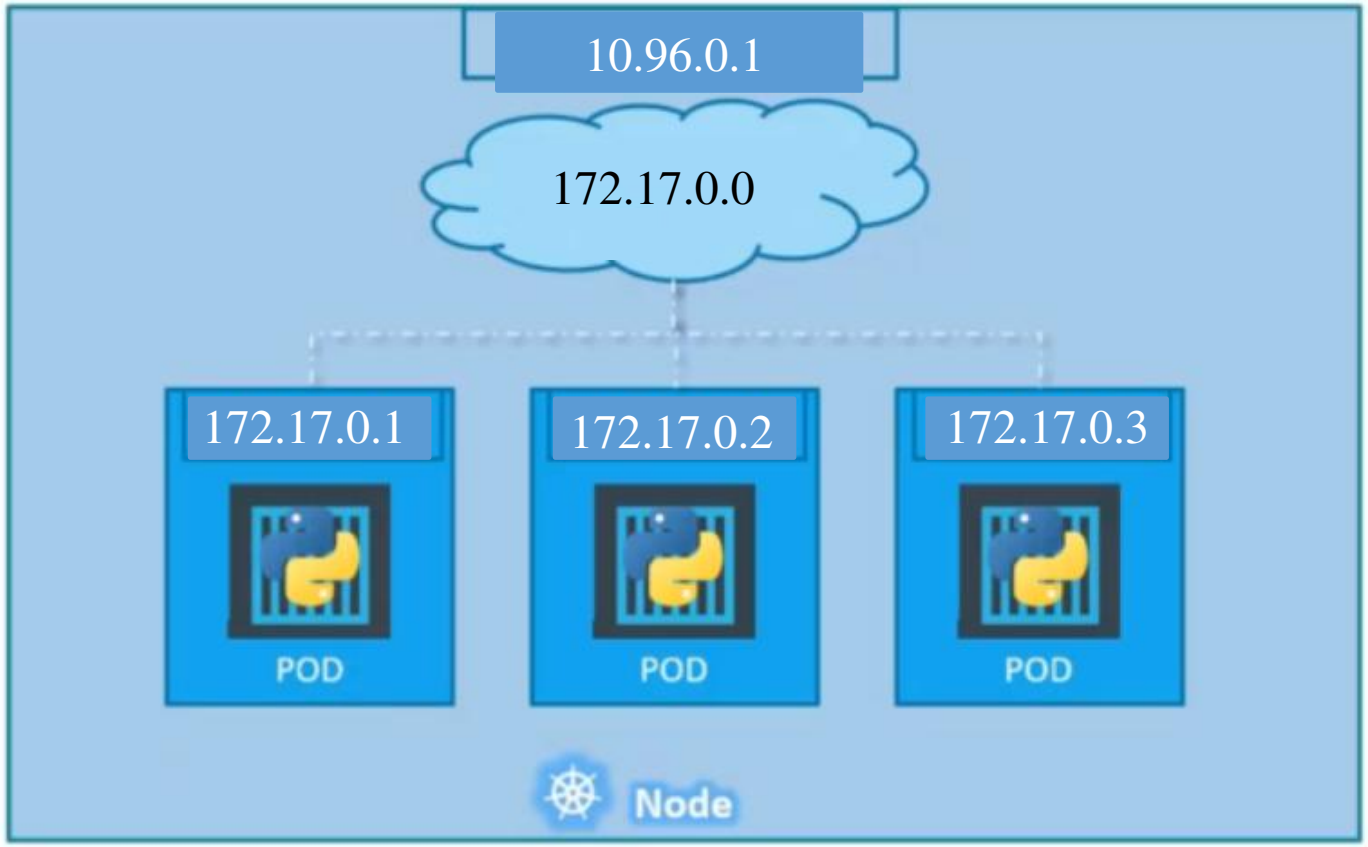
OBS: Os pods podem se comunicar entre eles através destes endereços de IP.



Funcionamento básico de redes no Kubernetes

Em todo caso não é recomendável fazer a comunicação/acesso de um pod com outro através do endereço IP, pois este endereço pode mudar com a recriação do pod, que ocorre por exemplo com uma atualização.

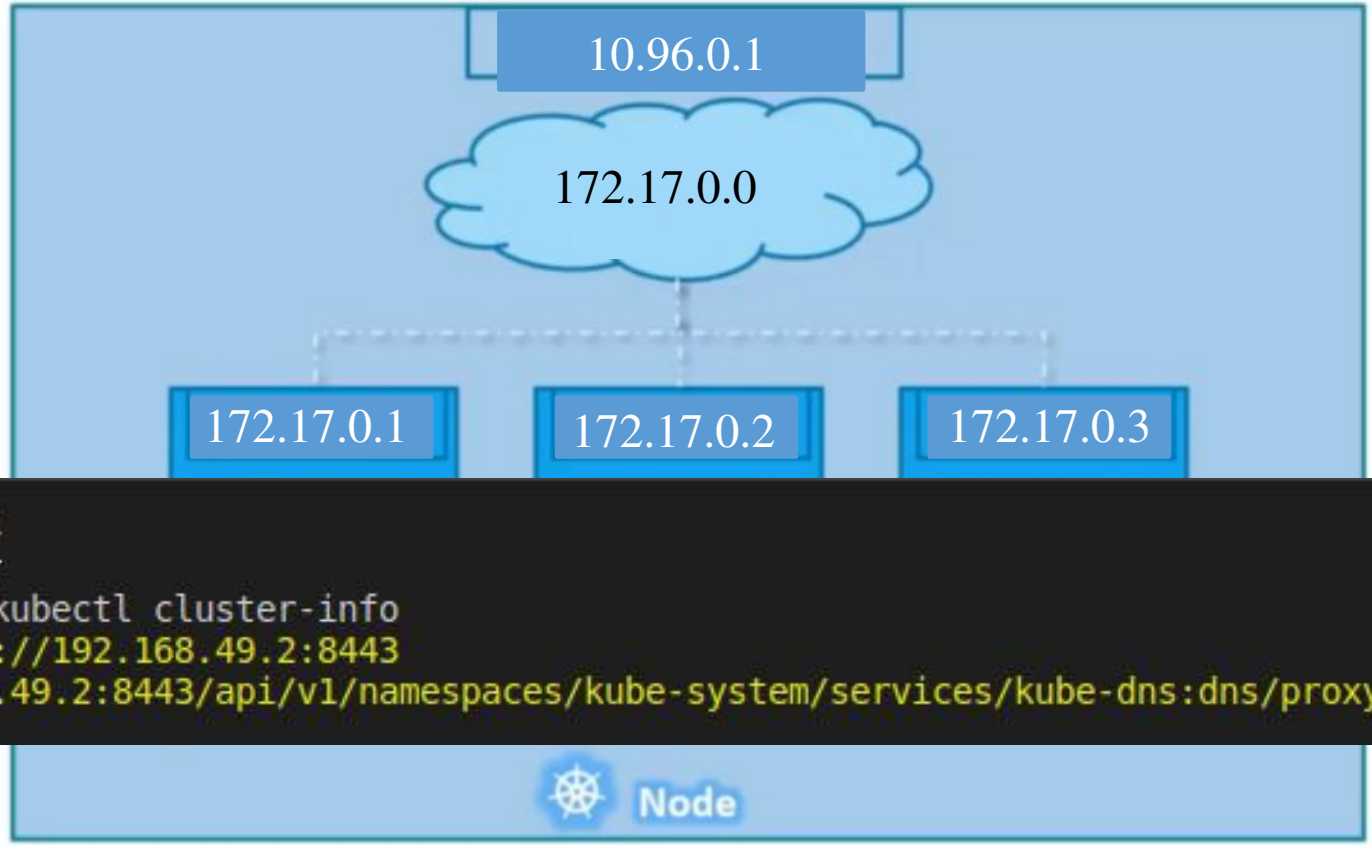
OBS: Iremos aprender na próxima seção deste curso, quando aprendermos sobre serviços, melhores formas de realizar a comunicação entre pods que não seja usando endereços IP mas sim os nomes dos serviços criados.



Funcionamento básico de redes no Kubernetes

KubeDNS

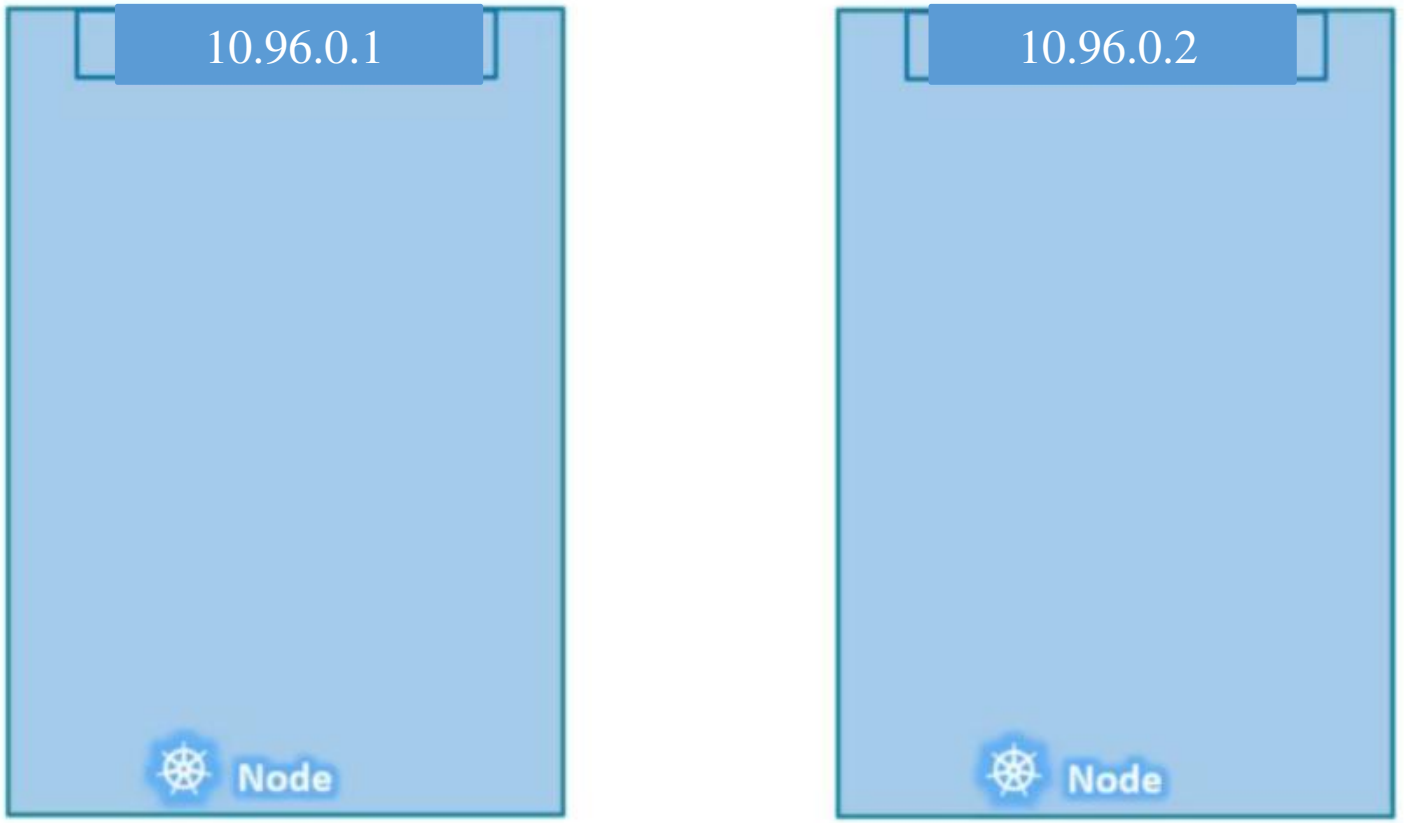
Dentre os serviços que são instalados junto com o Kubernetes está o KubeDNS que faz toda a parte de tradução de nomes para endereços IP assim como um servidor DNS normal.




```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

geek@university:~/Downloads/secao04$ kubectl cluster-info
Kubernetes master is running at https://192.168.49.2:8443
KubeDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

Funcionamento básico de redes no Kubernetes

O endereçamento IP em um cluster contendo apenas 1 node é simples, mas como funciona isso quando temos múltiplos nodes em um cluster?



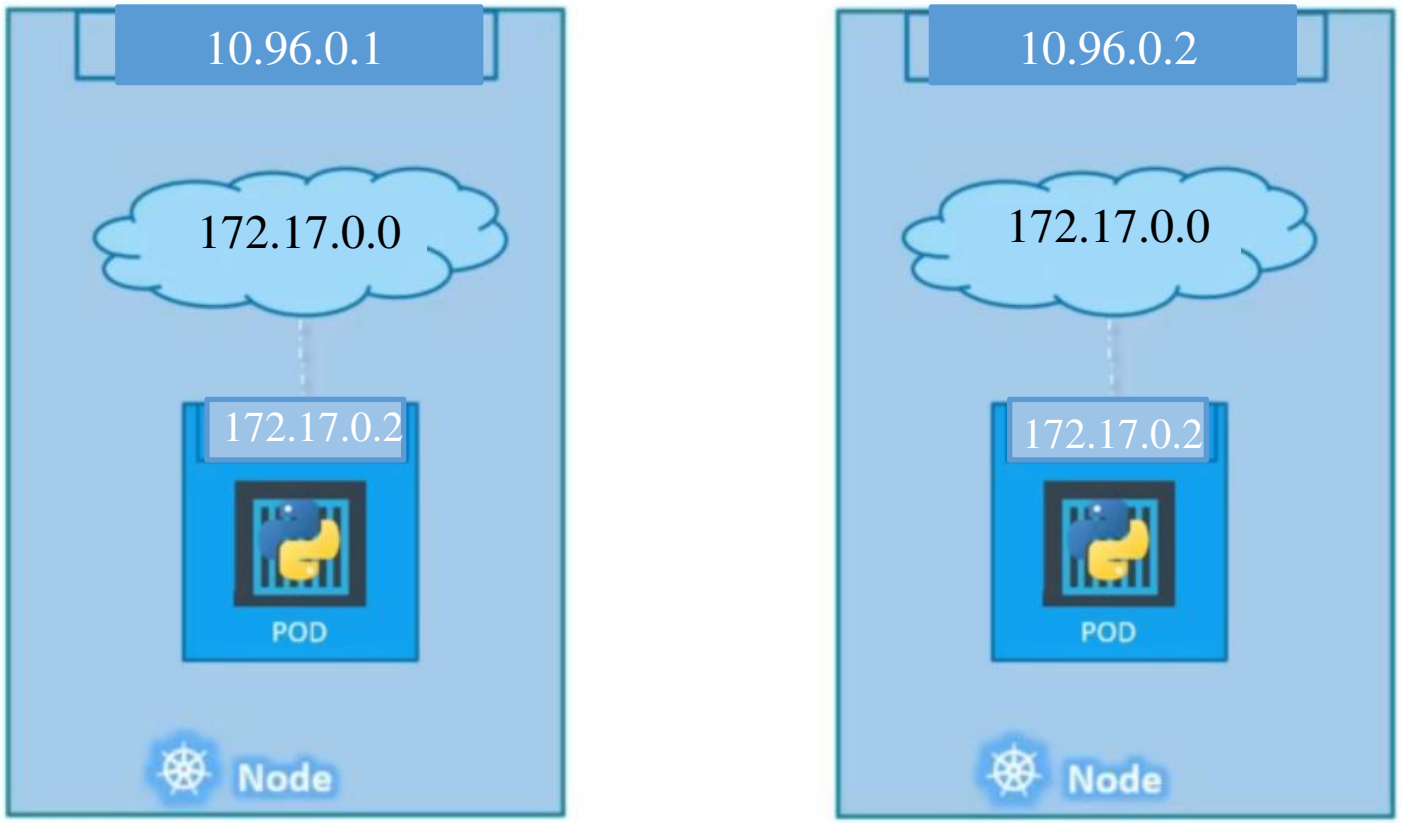
Funcionamento básico de redes no Kubernetes

O endereçamento IP em um cluster contendo apenas 1 node é simples, mas como funciona isso quando temos múltiplos nodes em um cluster?

Note que temos 2 clusters independentes, ou seja, eles não estão vinculados a um mesmo cluster. Cada cluster tem seu próprio endereço IP. Então eles podem se comunicar entre eles já que estão na mesma rede.

Note que internamente eles estão com o mesmo endereço IP no qual é usado para distribuir endereçamento IP para os pods.

Inclusive os pods estão com o mesmo endereço IP.



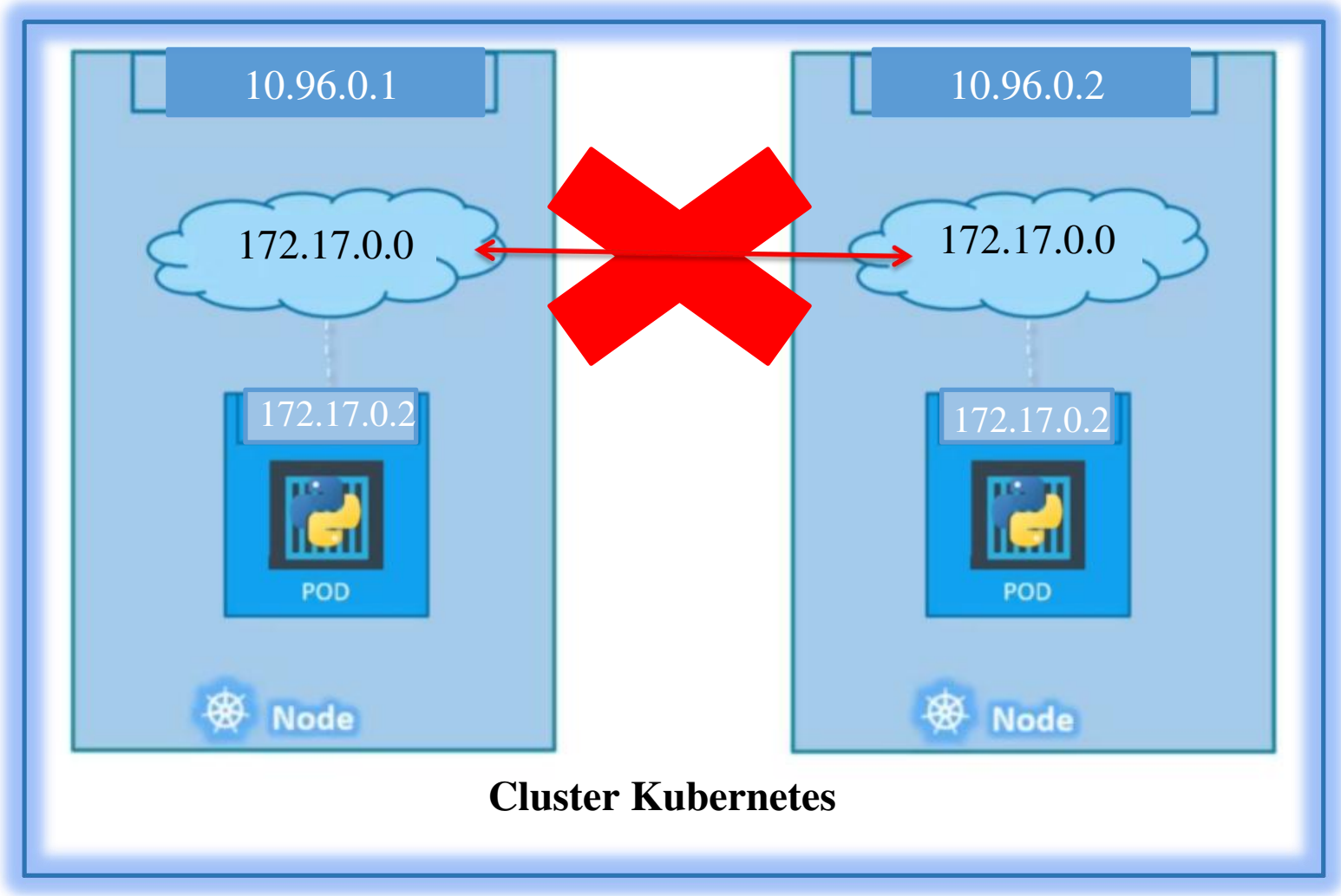
Funcionamento básico de redes no Kubernetes

Caso juntemos estes nodes em um único cluster Kubernetes teremos problemas.

Isso porque não podemos ter entidades em uma mesma rede com um mesmo endereço IP, já que o endereço IP é usado não apenas como endereçamento mas também como identificador em uma rede.

Desta forma ocorrerá um conflito de rede e o Kubernetes não tem nenhuma ferramenta para ajustar este problema de forma automática.

O Kubernetes espera que os operadores do cluster tenham conhecimentos mínimos de rede para que estes problemas não ocorram seguindo alguns critérios bem definidos.

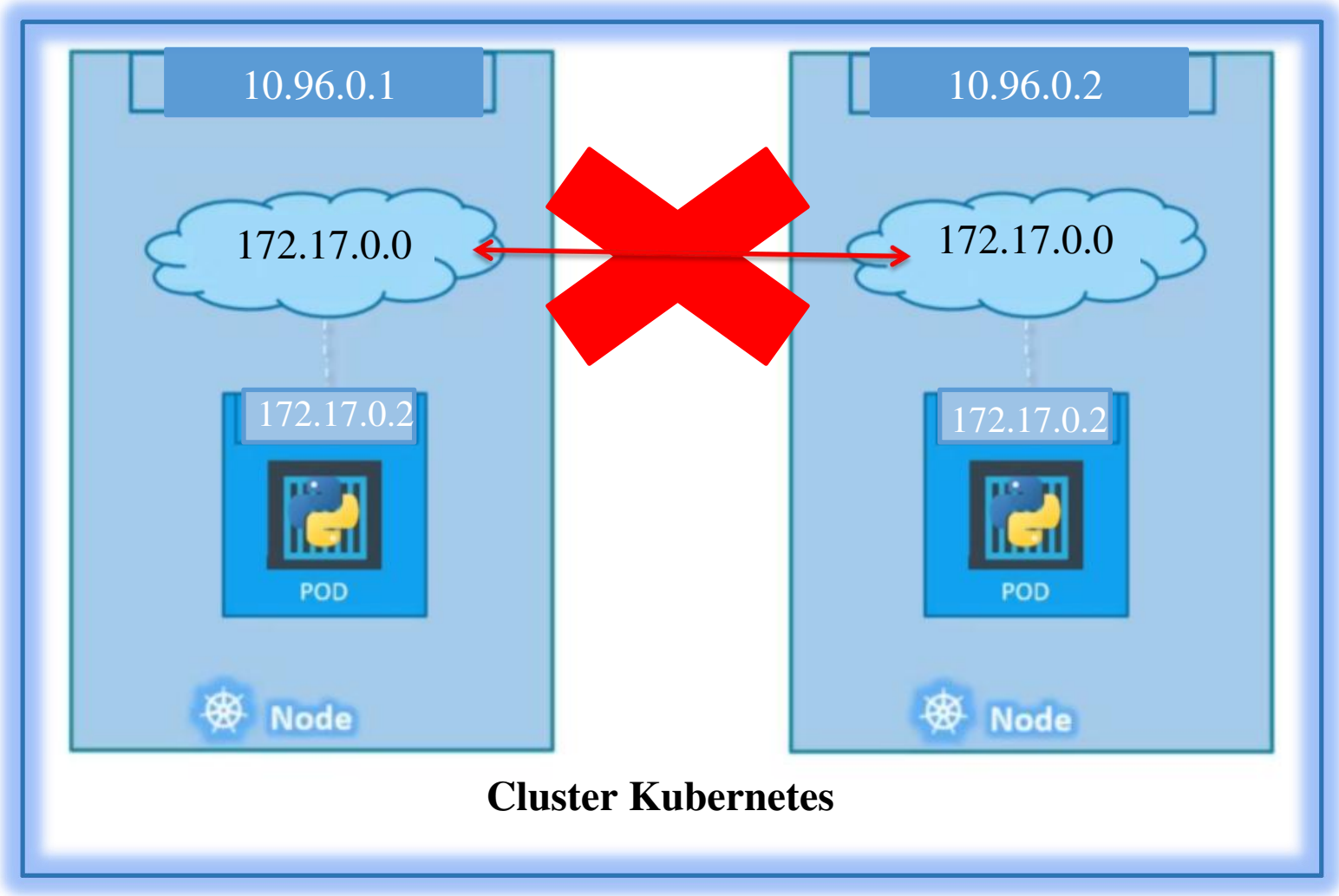


Funcionamento básico de redes no Kubernetes

Critérios para evitar problemas de rede com Kubernetes

- Todos containers/PODs podem se comunicar-se entre eles sem o uso de NAT*.

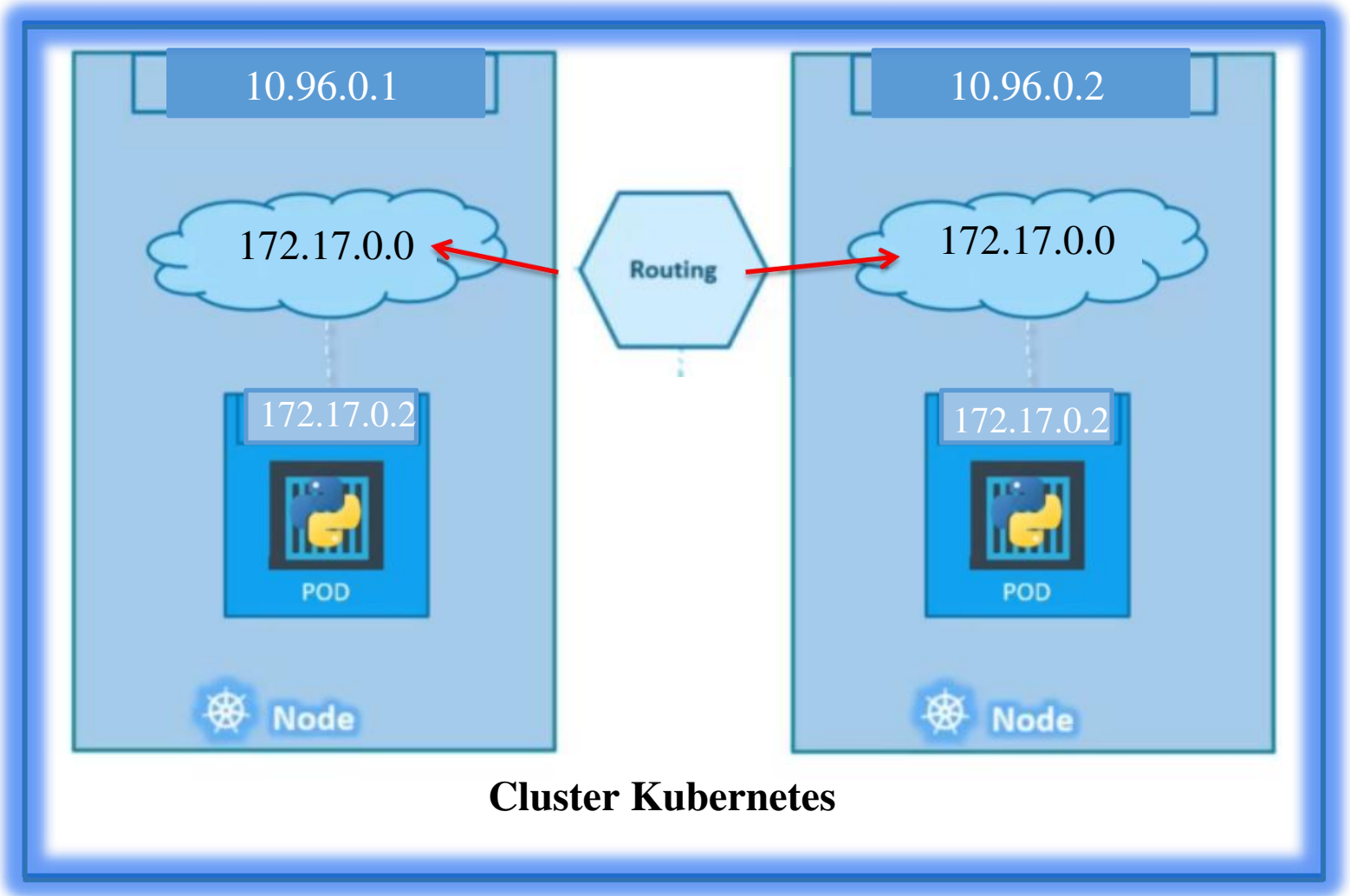
- Todos os nodes podem se comunicar com todos containers e vice-versa sem o uso de NAT*.

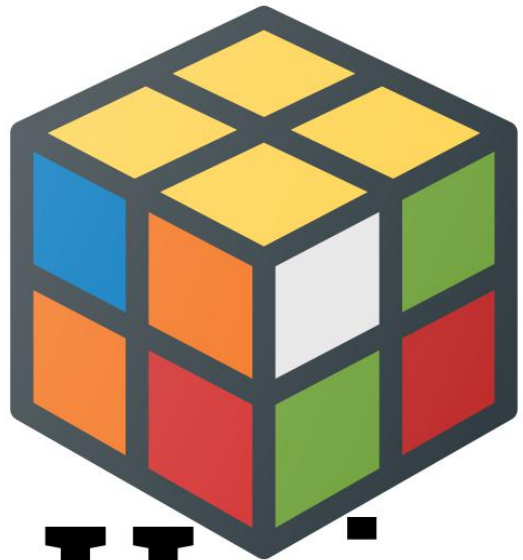


*https://purainfo.com.br/o-que-e-como-funciona-o-nat/

Funcionamento básico de redes no Kubernetes

Isso é resolvido facilmente utilizando técnicas de roteamento.





Geek University

Evolua seu lado geek!

www.geekuniversity.com.br