



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Conhecendo o NodePort no Kubernetes



Conhecendo o NodePort no Kubernetes



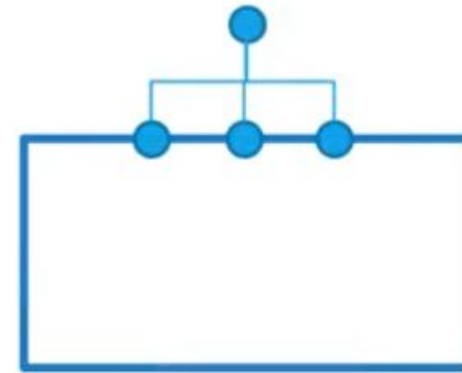
Tipos de Service



NodePort



ClusterIP



LoadBalancer

Conhecendo o NodePort no Kubernetes



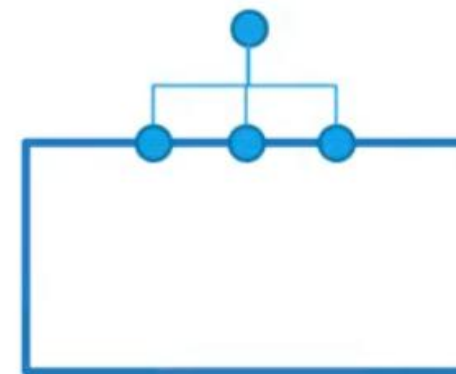
Tipos de Service



NodePort



ClusterIP



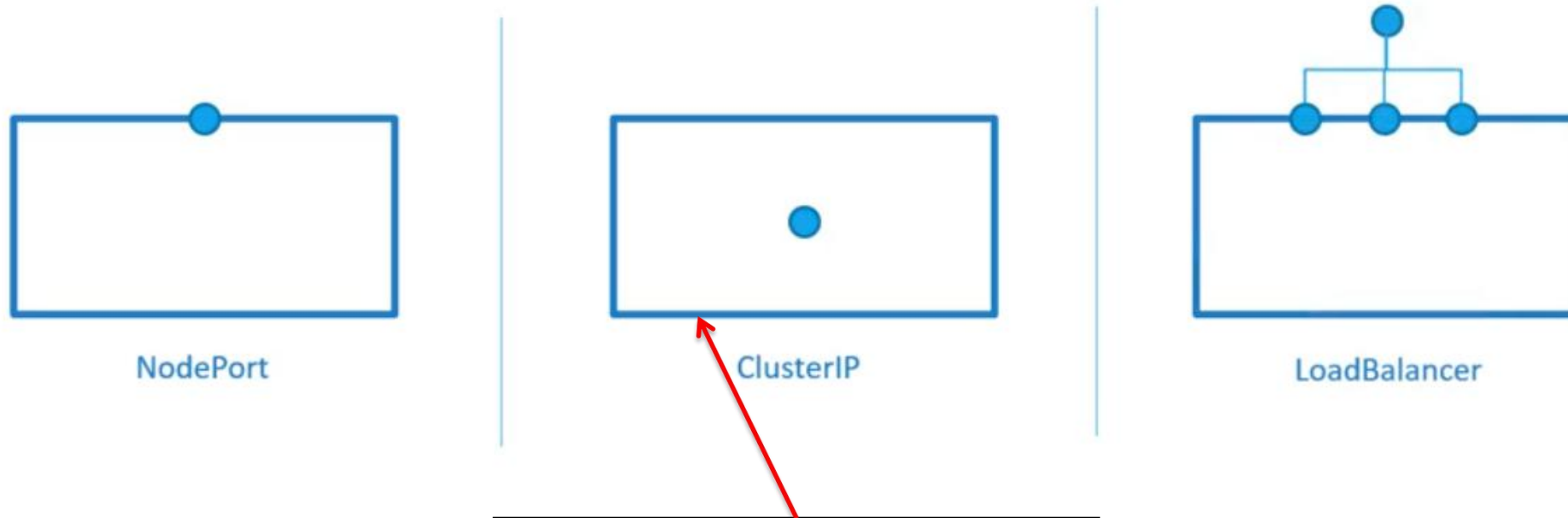
LoadBalancer

O NodePort abre uma comunicação entre o pod e agentes externos, como navegadores web se o serviço disponibilizado for web por exemplo.

Conhecendo o NodePort no Kubernetes



Tipos de Service

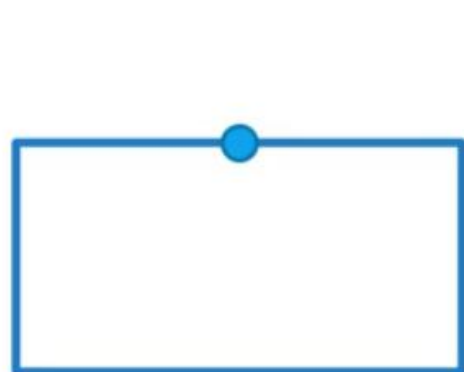


O ClusterIP é usado quando a comunicação entre os serviços é apenas interna (privada) ao cluster, sem acesso à agentes externos.

Conhecendo o NodePort no Kubernetes



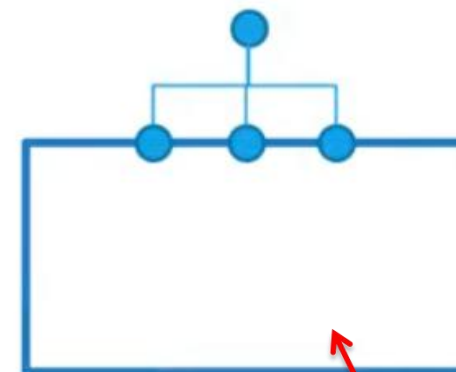
Tipos de Service



NodePort



ClusterIP



LoadBalancer

O LoadBalancer habilita a distribuição de carga entre os diversos pods que estão realizando o serviço. Este recurso não funciona localmente, mas sim em provedores cloud.



Conhecendo o NodePort no Kubernetes

NodePort

Note que temos 3 portas envolvidas neste serviço:

30008* (NodePort) -> Faz a comunicação externa com o Service.

80 (Service) -> Porta que faz comunicação com o pod (**Port**)

80 (Pod) -> Porta do servidor web (**Target Port**)

* Esta porta precisa estar no range 30000 - 32767





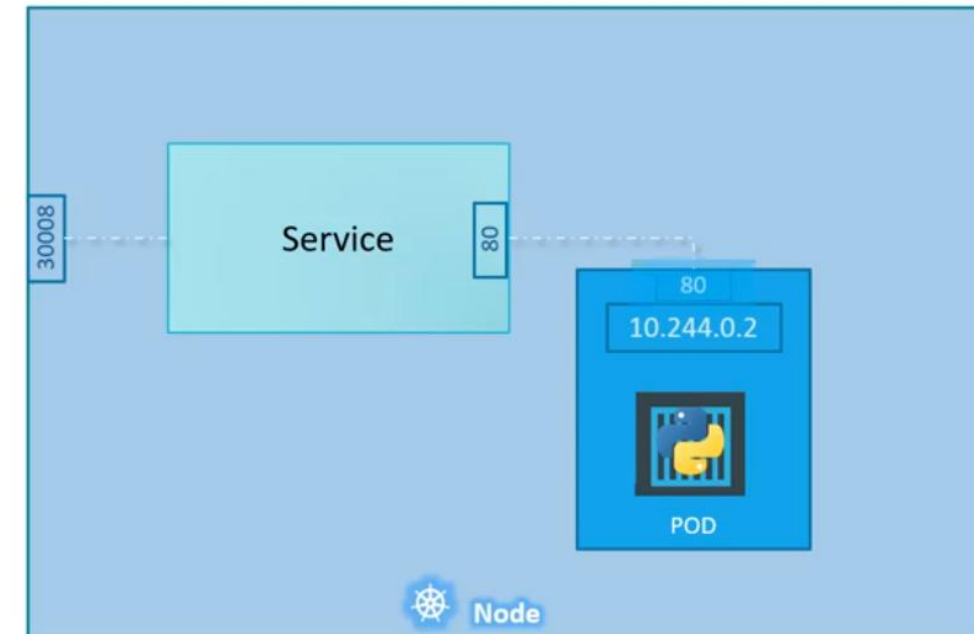
Conhecendo o NodePort no Kubernetes

Serviço “público”



NodePort

```
apiVersion: v1
kind: Service
metadata:
  name: frontend-svc
spec:
  selector:
    type: frontend
  ports:
    - name: http
      targetPort: 80
      port: 80
      nodePort: 30080
  type: NodePort
```



OBS: Apenas ‘port’ é um campo obrigatório.

Caso ‘targetPort’ não seja informado irá assumir o mesmo valor de ‘port’. E caso nodePort não seja informado irá assumir um valor aleatório dentro do range.



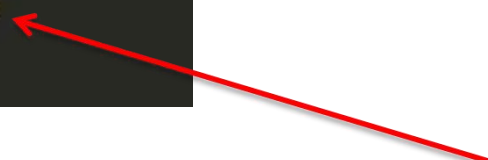
Conhecendo o NodePort no Kubernetes

Serviço “privado”

```
apiVersion: v1
kind: Service
metadata:
  name: frontend-svc
spec:
  selector:
    type: frontend
  ports:
    - name: http
      port: 80
  type: ClusterIP
```



ClusterIP

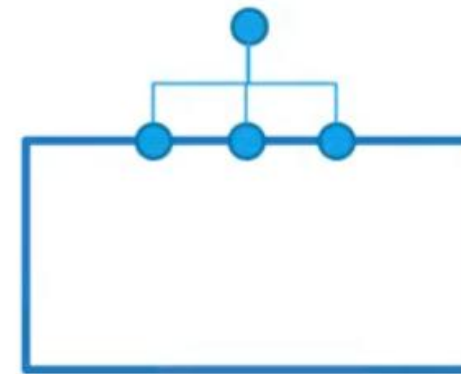




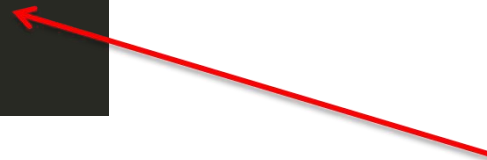
Conhecendo o NodePort no Kubernetes

Serviço “balanceador de carga”

```
apiVersion: v1
kind: Service
metadata:
  name: frontend-svc
spec:
  selector:
    type: frontend
  ports:
    - name: http
      targetPort: 80
      port: 80
  type: LoadBalancer
```



LoadBalancer





Conhecendo o NodePort no Kubernetes

Kubectl

Após a criação do arquivo de definição do objeto Kubernetes, a criação do objeto segue o mesmo padrão já conhecido:

```
> kubectl create -f service-definition.yml  
service "myapp-service" created
```

```
> kubectl get services
```

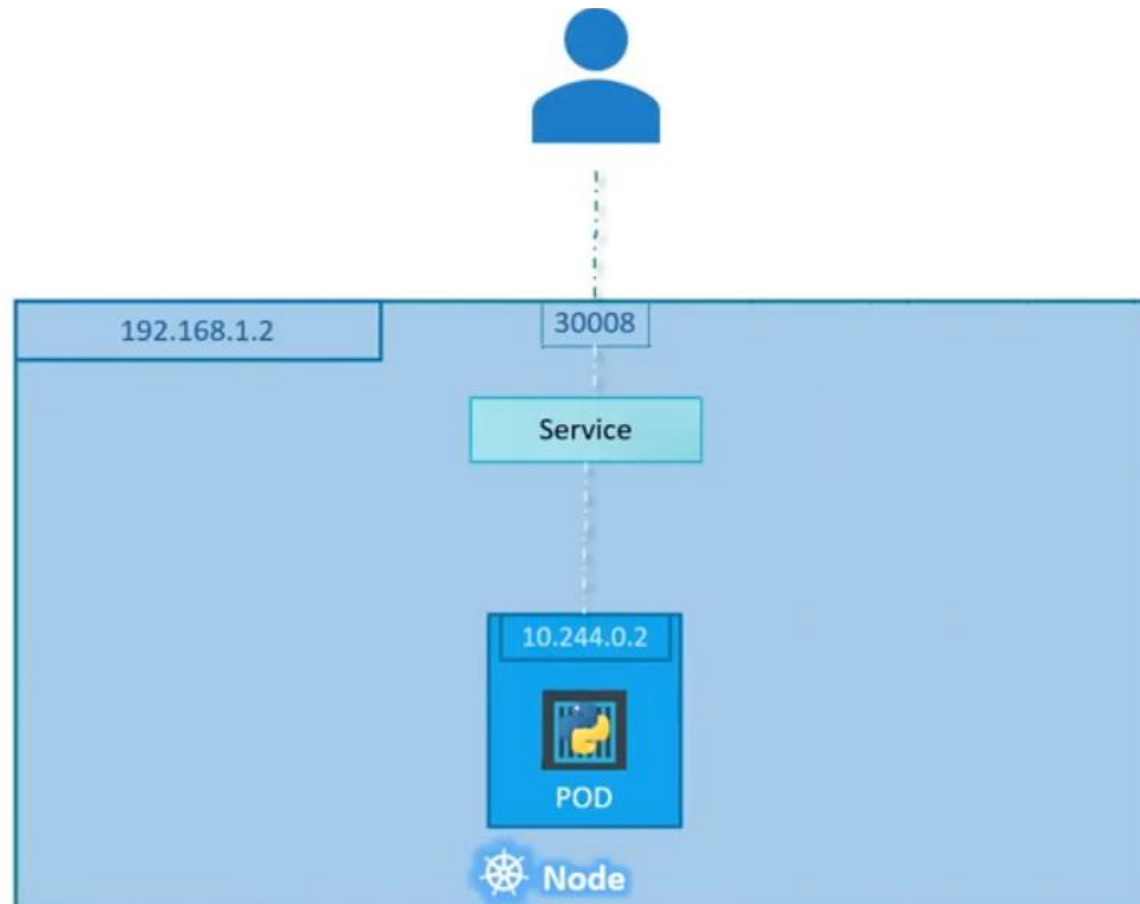
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16d
myapp-service	NodePort	10.106.127.123	<none>	80:30008/TCP	5m

Fazendo uso do serviço NodePort, a partir de então poderíamos acessar o servidor web no endereço do cluster, exemplo: <http://192.168.42.2:3008>



Conhecendo o NodePort no Kubernetes

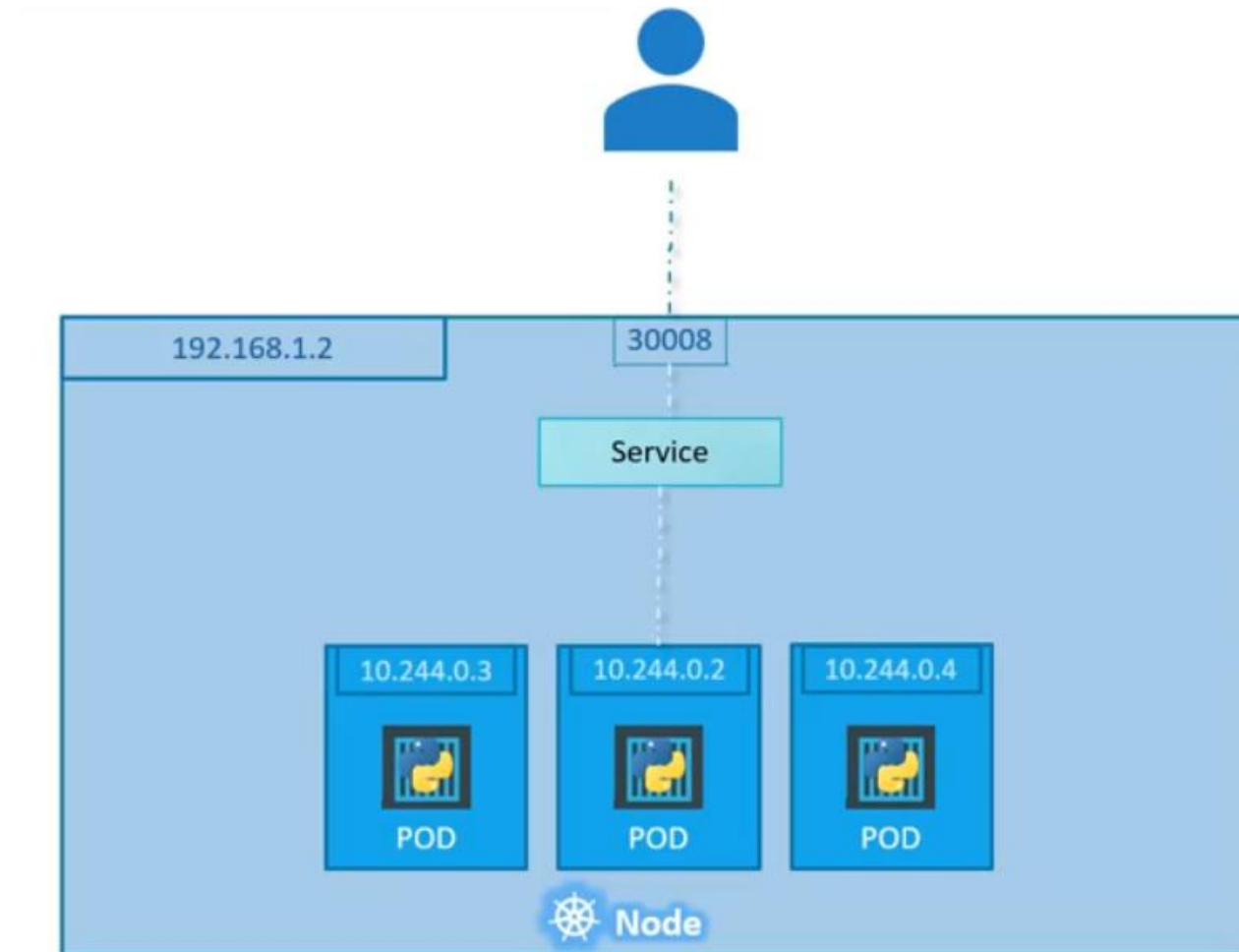
Tudo isso funciona de forma simples e objetiva quando temos 1 único POD com nosso serviço web....





Conhecendo o NodePort no Kubernetes

Mas o que ocorreria se tivéssemos múltiplos PODs dividindo a carga da aplicação web conforme é recomendado?

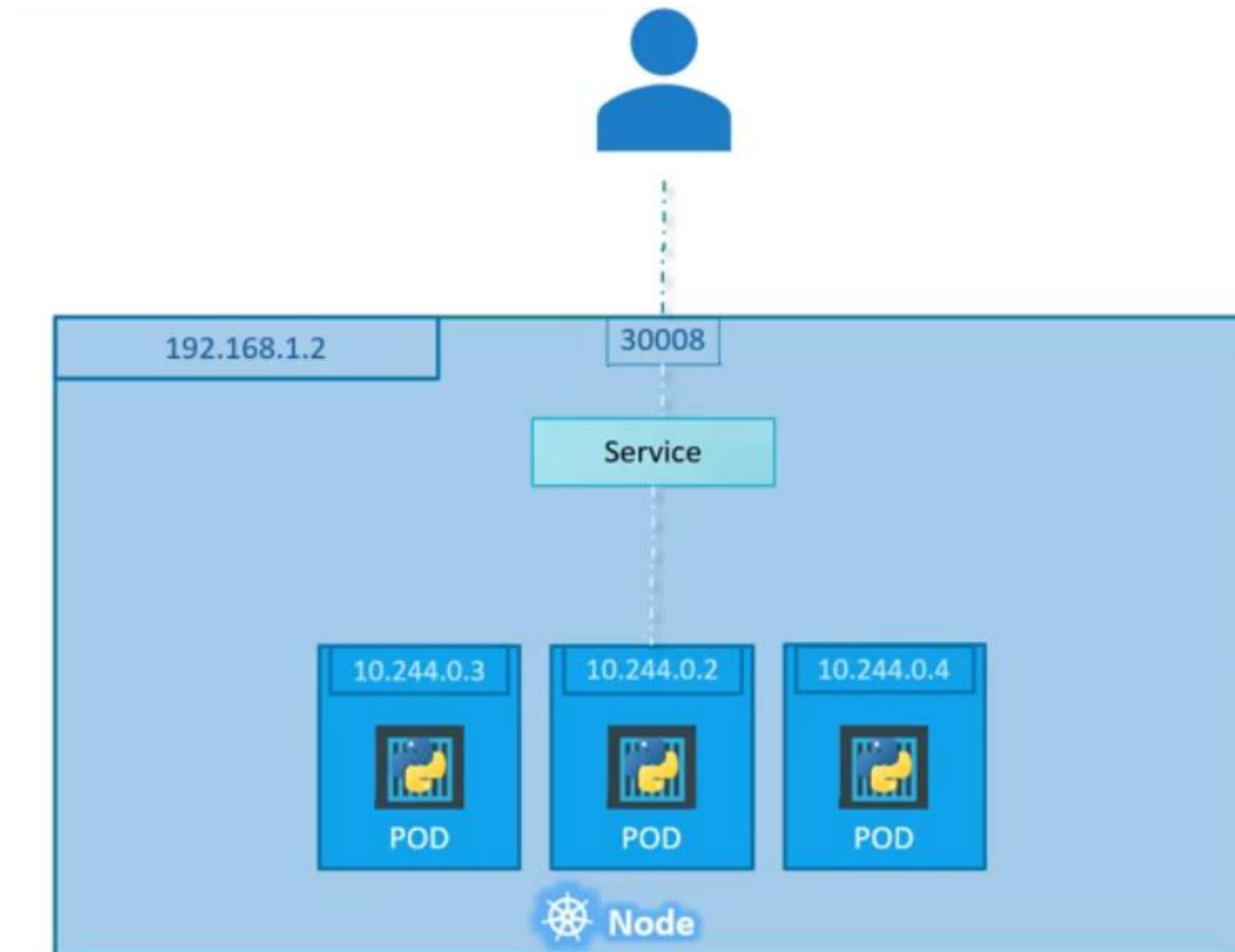




Conhecendo o NodePort no Kubernetes

Mas o que ocorreria se tivéssemos múltiplos PODs dividindo a carga da aplicação web conforme é recomendado?

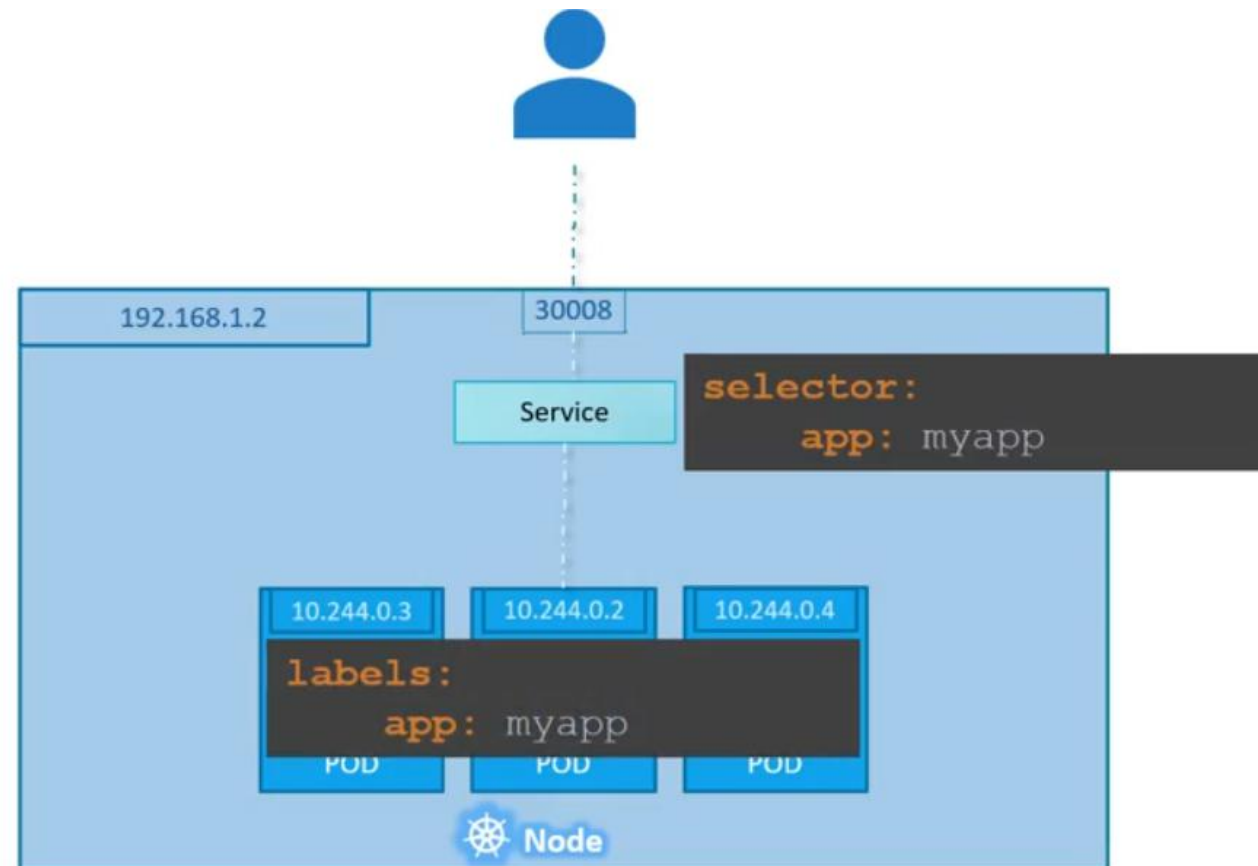
Nada mudaria!





Conhecendo o NodePort no Kubernetes

Mas o que ocorreria se tivéssemos múltiplos PODs dividindo a carga da aplicação web conforme é recomendado?



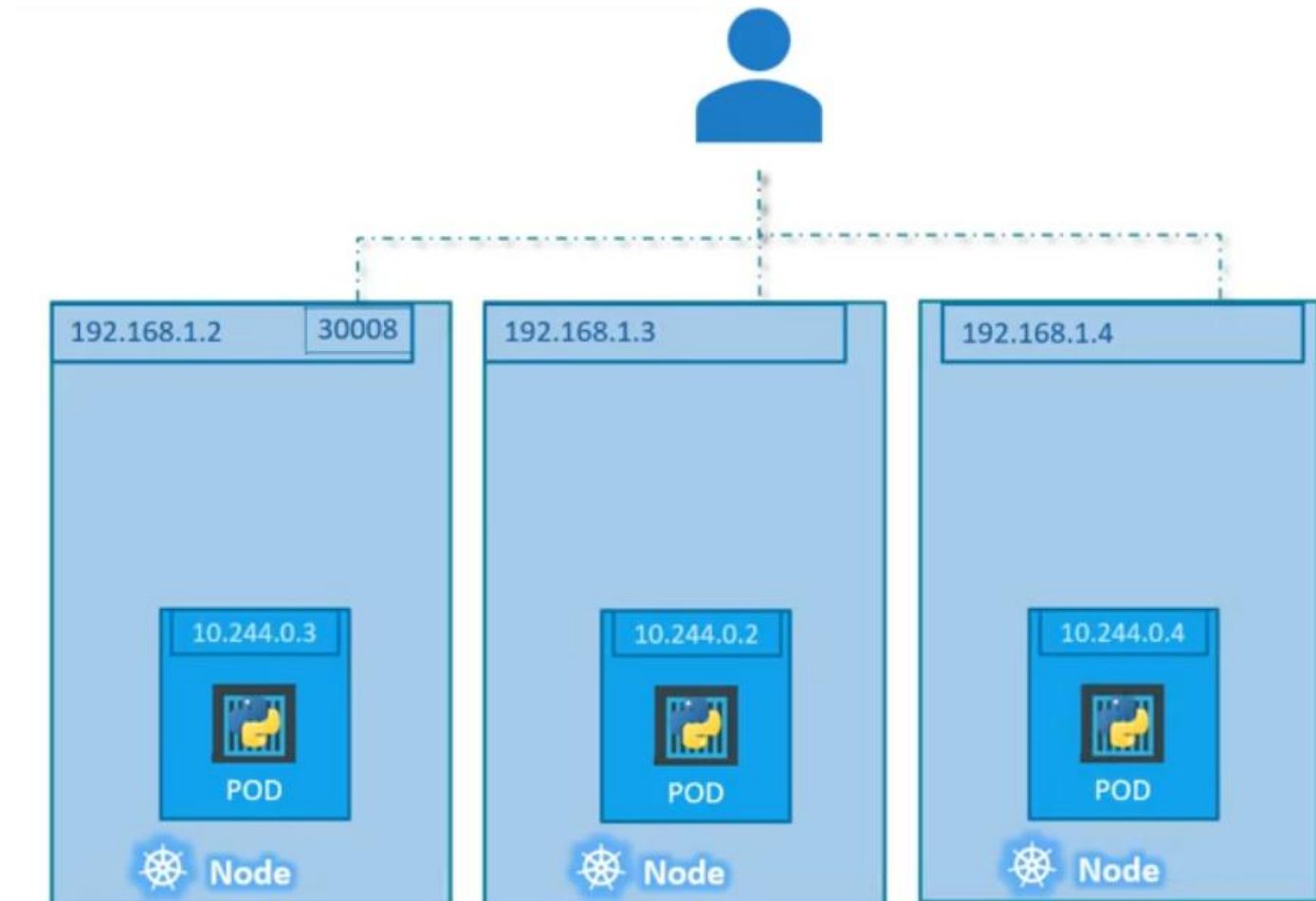
Neste exemplo, todos os pods possuem o mesmo labels “app” com valor “myapp” e o Service está utilizando no selector um label “app” com valor “myapp”.

Desta forma tudo irá funcionar sem nenhuma mudança!



Conhecendo o NodePort no Kubernetes

E se os PODs estiverem distribuídos entre múltiplos nodes?





Conhecendo o NodePort no Kubernetes

E se os PODs estiverem distribuídos entre múltiplos nodes?

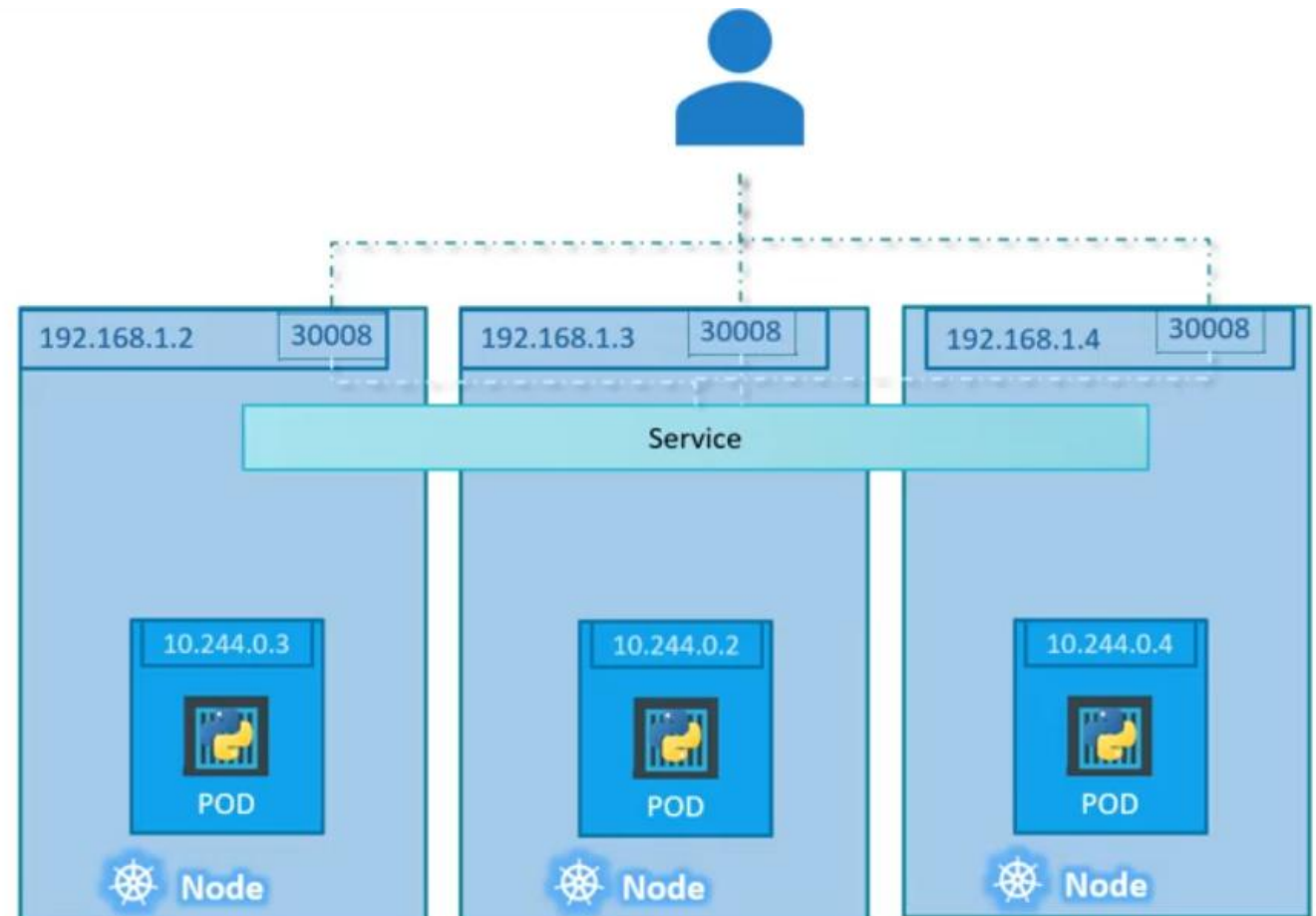
O Kubernetes cria um serviço englobando todos os nodes, fazendo com que o acesso ocorresse normalmente.

Neste caso podemos fazer acesso usando qualquer dos endereços dos diretores clusters:

`http://192.168.1.2:30008`

`http://192.168.1.3:30008`

`http://192.168.1.4:30008`





Geek University

Evolua seu lado geek!

www.geekuniversity.com.br