



# Geek University

**Evolua seu lado geek!**

[www.geekuniversity.com.br](http://www.geekuniversity.com.br)

# Funcionamento de namespaces no Kubernetes





# Funcionamento de namespaces no Kubernetes

Esta seção de redes serve de preparação para a próxima seção onde estudaremos serviços.

É importante que, além de entendermos os conceitos básicos de redes no Kubernetes possamos entender também o funcionamento de um conceito chamado de namespaces.



# Funcionamento de namespaces no Kubernetes

Ao executar o comando *kubectl get all* temos acesso a todos os objetos do nosso cluster Kubernetes, certo?

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
geek@university:~/Downloads/secao04$ kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/nginx-pod  1/1     Running   1           4h28m

NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    5h
geek@university:~/Downloads/secao04$
```



# Funcionamento de namespaces no Kubernetes

Ao executar o comando *kubectl get all* temos acesso a todos os objetos do nosso cluster Kubernetes, certo?

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

geek@university:~/Downloads/secao04$ kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/nginx-pod  1/1     Running   1           4h28m

NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    5h
geek@university:~/Downloads/secao04$
```

Lembra que quando o Kubernetes é instalado, um serviço de DNS chamado KubeDNS é instalado junto com ele?

```
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
```

Por quê ele não está aparecendo no resultado do comando acima?

É aqui que entra o conceito de namespaces.

# Funcionamento de namespaces no Kubernetes



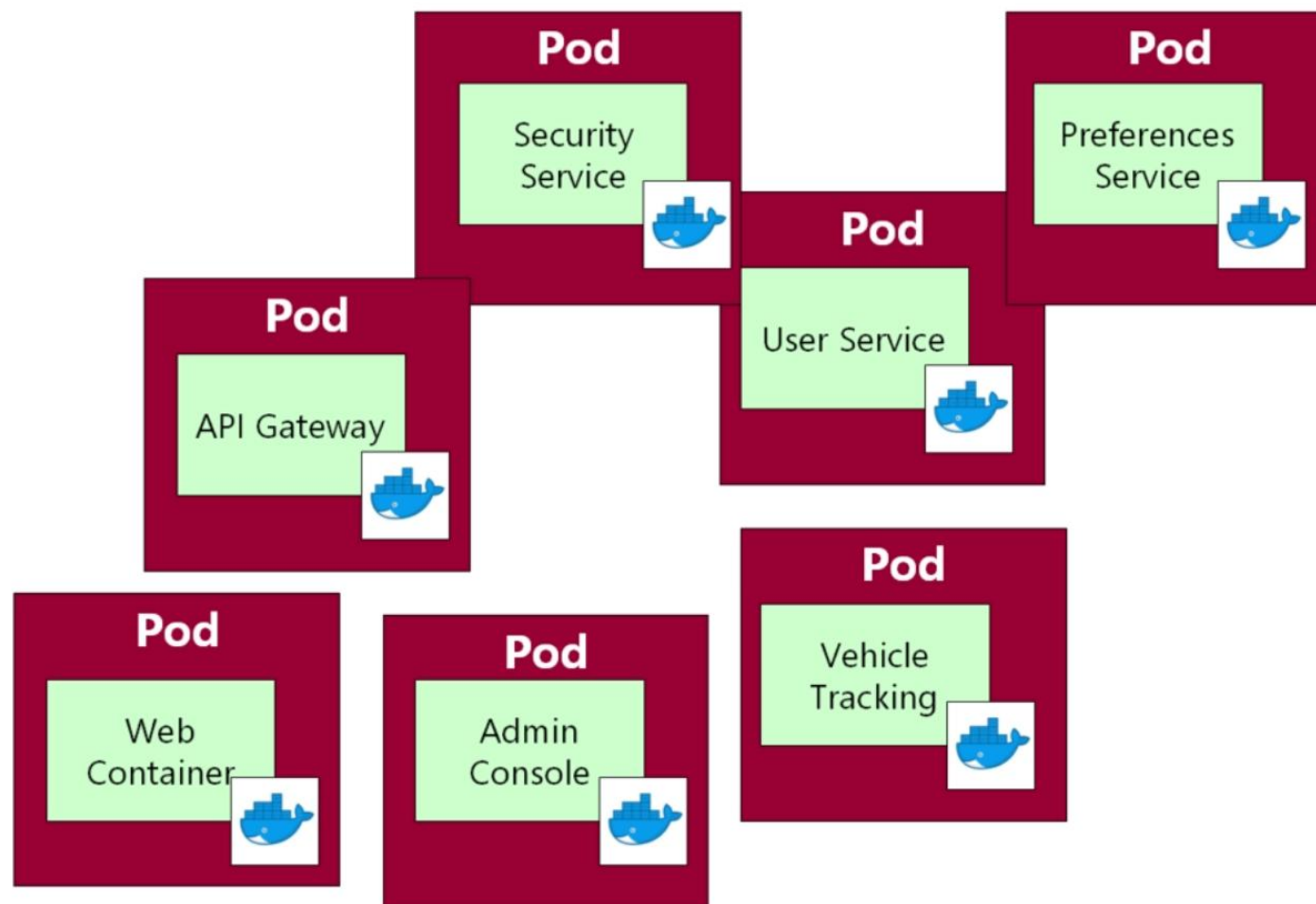
## Namespaces em Kubernetes

Namespace é um “espaço” na qual separamos nossos recursos dentro do Kubernetes em áreas separadas.

É aqui que inicia o conceito de arquitetura de micro-serviços que é tão falada ultimamente.

Na imagem ao lado temos vários pods que, se não forem organizados, pode ficar difícil manter.

Podemos então agrupar pods que fazem parte de uma mesma área em um namespace.





# Funcionamento de namespaces no Kubernetes

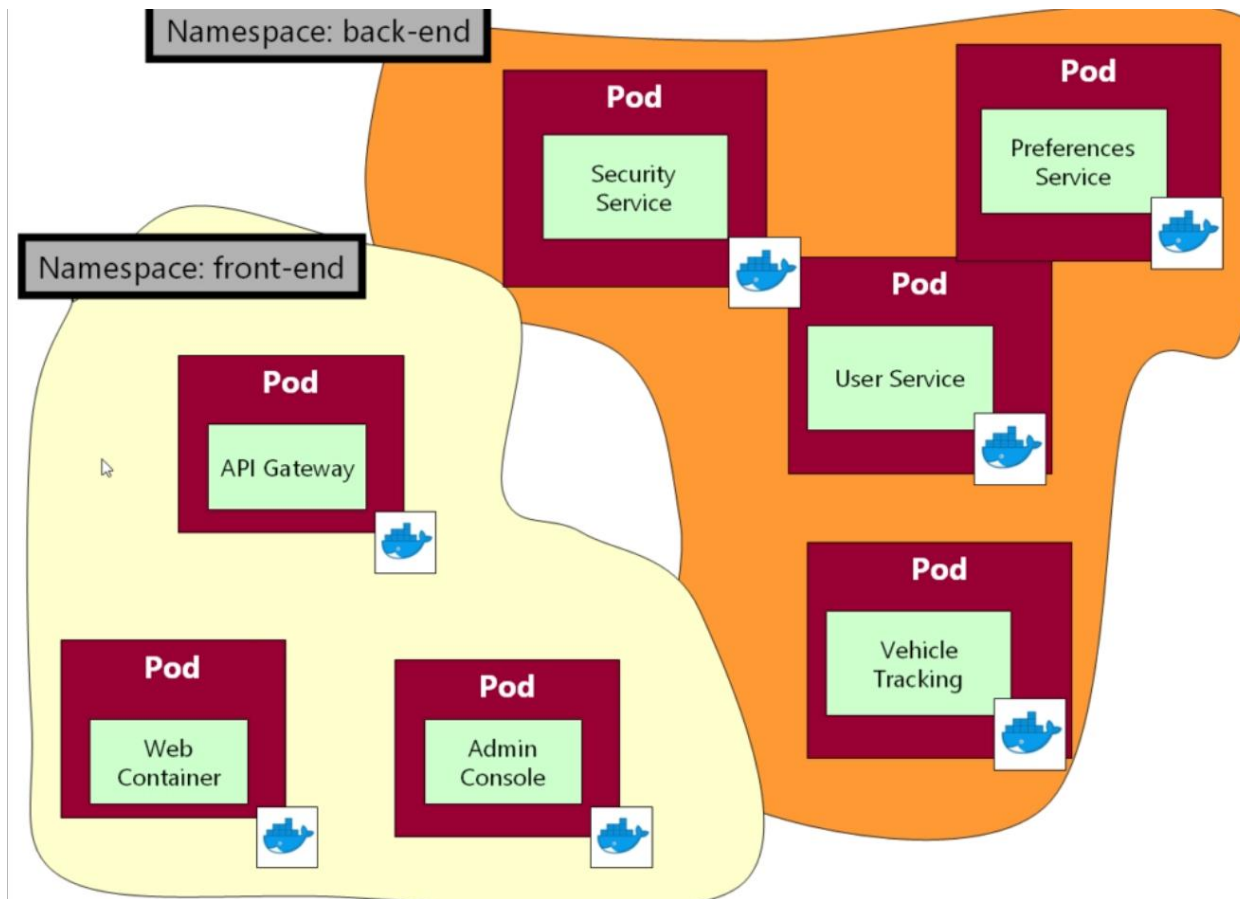
## Namespaces em Kubernetes

Neste exemplo agrupamos nossos pods em 2 namespaces, sendo um chamado de front-end e outro back-end.

O conceito de namespaces não é original no Kubernetes mas sim da ciência da computação.

Por exemplo no desenvolvimento de sistemas, costumamos criar pacotes para organizar nosso código. Fazemos isso em Java, Python e várias outras linguagens.

Linguagens como C# por exemplo fazem uso de namespaces diretamente.



# Funcionamento de namespaces no Kubernetes



## **Namespaces em Kubernetes**

Mas o que isso tudo tem a ver com o que vimos no início desta aula, com informações do KubeDNS não estar presente nas informações do cluster?

Note que até o momento não criamos nenhum namespace. Nós criamos pods, replicaset e deployments.

Na verdade a gente nem sabia da existência de namespaces aqui no Kubernetes.

O que ocorre é que se ao criamos um objeto no Kubernetes e não especificarmos um namespace para ele este será colocado no namespace default (padrão) do Kubernetes.





# Funcionamento de namespaces no Kubernetes

## Namespaces em Kubernetes

Desta forma, quando executamos o comando `kubectl get all`, estamos pegando todos os objetos existentes no namespace default (padrão) do Kubernetes.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

geek@university:~/Downloads/secao04$ kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/nginx-pod  1/1     Running   1           4h28m

NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    5h
geek@university:~/Downloads/secao04$
```



# Funcionamento de namespaces no Kubernetes

## Namespaces em Kubernetes

Podemos consultar por exemplo os namespaces existentes com o comando:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

geek@university:~/Downloads/secao04$ kubectl get namespaces
NAME                STATUS    AGE
default             Active    6h4m
kube-node-lease     Active    6h4m
kube-public         Active    6h4m
kube-system         Active    6h4m
geek@university:~/Downloads/secao04$ kubectl get ns
NAME                STATUS    AGE
default             Active    6h4m
kube-node-lease     Active    6h4m
kube-public         Active    6h4m
kube-system         Active    6h4m
geek@university:~/Downloads/secao04$
```



# Funcionamento de namespaces no Kubernetes

## Namespaces em Kubernetes

Podemos consultar por exemplo consultar objetos especificando o nome do namespace:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
geek@university:~/Downloads/secao04$ kubectl get pods -n default
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-pod	1/1	Running	1	5h34m

```
geek@university:~/Downloads/secao04$ kubectl get pods -n kube-node-lease
```

No resources found in kube-node-lease namespace.

```
geek@university:~/Downloads/secao04$ kubectl get pods -n kube-public
```

No resources found in kube-public namespace.

```
geek@university:~/Downloads/secao04$ kubectl get pods -n kube-system
```

NAME	READY	STATUS	RESTARTS	AGE
coredns-f9fd979d6-8v8g8	1/1	Running	1	6h6m
etcd-minikube	1/1	Running	1	6h6m
kube-apiserver-minikube	1/1	Running	1	6h6m
kube-controller-manager-minikube	1/1	Running	1	6h6m
kube-proxy-ngtjd	1/1	Running	1	6h6m
kube-scheduler-minikube	1/1	Running	1	6h6m
storage-provisioner	1/1	Running	3	6h6m

```
geek@university:~/Downloads/secao04$
```

Note quantos pods temos executando no namespace 'kube-system'



# Funcionamento de namespaces no Kubernetes

## Namespaces em Kubernetes

Podemos também consultar todos os objetos especificando o nome do namespace:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

geek@university:~/Downloads/secao04$ kubectl get all -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
pod/coredns-f9fd979d6-8v8g8        1/1     Running   1          6h10m
pod/etcd-minikube                   1/1     Running   1          6h10m
pod/kube-apiserver-minikube         1/1     Running   1          6h10m
pod/kube-controller-manager-minikube 1/1     Running   1          6h10m
pod/kube-proxy-ngtjd               1/1     Running   1          6h10m
pod/kube-scheduler-minikube         1/1     Running   1          6h10m
pod/storage-provisioner             1/1     Running   3          6h10m

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/kube-dns                    ClusterIP      10.96.0.10   <none>        53/UDP,53/TCP,9153/TCP 6h10m

NAME                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/kube-proxy           1         1         1       1             1           kubernetes.io/os=linux 6h10m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/coredns              1/1     1             1           6h10m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/coredns-f9fd979d6    1         1         1       6h10m
geek@university:~/Downloads/secao04$
```



# Funcionamento de namespaces no Kubernetes

## Namespaces em Kubernetes

**OBS:** Não é recomendado criar objetos kubernetes nos namespaces internos do Kubernetes

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
geek@university:~/Downloads/secao04$ kubectl get ns
NAME                STATUS   AGE
default             Active   6h13m
kube-node-lease     Active   6h13m
kube-public         Active   6h13m
kube-system         Active   6h13m
geek@university:~/Downloads/secao04$
```

Podemos criar objetos sem especificar nenhum namespace, que irá criar os objetos no namespace default, ou mesmo especificar um namespace próprio para o objeto.

O problema de criar namespaces próprios ao invés de usar o namespace default do Kubernetes é que você deverá lembrar e informar os namespaces próprios existentes sempre que for criar um objeto.

Eu recomendo que inicialmente você use o namespace padrão do Kubernetes. ;)





# Funcionamento de namespaces no Kubernetes

## Criando Namespaces em Kubernetes

Podemos criar facilmente um namespace com o comando:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
geek@university:~/Downloads/secao04$ kubectl create namespace frontend --save-config
namespace/frontend created
geek@university:~/Downloads/secao04$
```

E se agora consultarmos os namespaces existentes:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
geek@university:~/Downloads/secao04$ kubectl get ns
NAME                STATUS    AGE
default             Active   6h23m
frontend            Active   49s
kube-node-lease     Active   6h23m
kube-public         Active   6h23m
kube-system         Active   6h24m
```



# Funcionamento de namespaces no Kubernetes

## Criando Objetos no Namespace próprio em Kubernetes

Podemos criar facilmente um objeto especificando o nome do namespace com o comando:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
geek@university:~/Downloads/secao04$ kubectl create -f pods/nginx.yaml --save-config --namespace=frontend
pod/nginx-pod created
geek@university:~/Downloads/seca
```

E se agora consultarmos os pods existentes:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
geek@university:~/Downloads/secao04$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   1           5h54m
geek@university:~/Downloads/secao04$ kubectl get pods -n frontend
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0           55s
```

**OBS:** Note que originalmente o comando só traz os objetos do namespace default. Para trazer os objetos de um namespace específico devemos informar o nome do namespace.



# Funcionamento de namespaces no Kubernetes

## Criando Objetos no Namespace próprio em Kubernetes

Vamos supor que você queira criar um namespace para ser o padrão do seu projeto, ao invés de usar por padrão o default.

Podemos definir isso com:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
geek@university:~/Downloads/secao04$ kubectl config set-context --current --namespace=frontend  
Context "minikube" modified.
```

A partir de então, qualquer objeto criado será criado no namespace 'frontend' que criamos anteriormente, a não ser que outro namespace seja informado durante a criação do objeto.





# Funcionamento de namespaces no Kubernetes

## Criando namespaces com YAML

*backend.yaml*

```
apiVersion: v1
kind: Namespace
metadata:
  name: backend
```

Então basta criar o objeto conforme já aprendemos antes:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
geek@university:~/Downloads/secao04$ kubectl create -f namespaces/backend.yaml --save-config
namespace/backend created
geek@university:~/Downloads/secao04$
```



# Geek University

**Evolua seu lado geek!**

[www.geekuniversity.com.br](http://www.geekuniversity.com.br)