

## Faça o que eu fiz na aula

Ao Tentar acessar nosso pod via navegador, percebemos que não é possível. Por que isso acontece?

Acontece porque o endereço ip que nosso pod ganhou é um endereço interno dentro do `minikube`, ou seja, não está disponível para acesso externo. O ideal dentro da estrutura do kubernetes, é que nunca o usuário final acesse diretamente um Pod, pois futuramente queremos escalar esse pod; ou seja, criar réplicas. Cada réplica que criarmos vai ter um endereço ip diferente, além disso, precisamos que o kubernetes monitore a situação dos pods dentro do nosso cluster, para que se aconteça algum problema com esse pod, o kubernetes imediatamente crie um novo para substituí-lo e assim garantir que a aplicação fique de pé. Para que isso seja possível, temos que colocar um controlador que cuide para que nosso pod sempre esteja funcionando, e em seguida garantirmos o acesso ao pod através de um serviço.

Para isso vamos criar mais um arquivo dentro do diretório kubernetes que criamos na aula 1. O nome do arquivo será `deployment-aplicacao.yml` e terá o seguinte conteúdo:

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: aplicacao-deployment
spec:
  template:
    metadata:
      labels:
        name: aplicacao-noticia-pod
    spec:
      containers:
        - name: container-kubernetes
          image: jnlucas/image-php:latest
          ports:
            - containerPort: 80
      volumeMounts:
        - name: hostvol
          mountPath: /var/www/html/
    volumes:
      - name: hostvol
        hostPath:
          path: /Users/joaoneves/docker/pedidos/kubernetes/noticias
```

Notem que no final do arquivo no parâmetro `path`, estou fazendo referência a pasta onde o projeto se encontra em meu computador, então para que funcione corretamente, utilize o diretório onde o projeto está em sua máquina. A principal função desse arquivo é garantir o estado desejado dos nossos pods, ou seja, é ele que fica supervisionando o funcionamento do pod, e caso algo aconteça outro pod é criado imediatamente. Vamos então colocar o Deployment para rodar com o comando

```
kubectl create -f deployment-aplicacao.yml
```

Para verificar se o deployment está funcionando, vamos digitar no terminal:

```
kubectl get pods
```

Uma outra forma de verificar a situação do nosso cluster é acessando o dashboard que o minikube nos fornece com o comando:

```
minikube dashboard
```

com isso seremos direcionados a uma página web, onde podemos verificar a situação de cada pod que criamos.