

## Mãos à obra: Objeto Deployment

O objeto Pod seria o mais básico existente no Kubernetes e não adiciona uma camada do estado desejado de nossa aplicação para o Kubernetes gerenciar, dessa forma, quando trabalhamos com o Kubernetes, nós abstraímos o objeto Pod dentro de um objeto que oferece tais recursos, como por exemplo o objeto Deployment.

Utilize o editor de texto de sua preferência e crie um novo arquivo chamado **deployment.yaml**. Feito isso, vamos colocar na primeira linha a versão da API do Kubernetes responsável pela esquematização do objeto que iremos criar, o objeto Deployment está definido na versão **apps/v1**.

```
apiVersion: apps/v1
```

Feito isso, vamos especificar que queremos criar o objeto **Deployment** e daremos o nome de **aplicacao-deployment** para esse objeto:

```
kind: Deployment
metadata:
  name: aplicacao-deployment
```

Na sequência, iremos declarar o padrão de objeto que está sendo abstraído pelo objeto **Deployment**:

```
spec:
  template:
```

O padrão de objeto que deve ser abstraído pelo objeto Deployment é o **Pod**, que nós já configuramos na etapa anterior, volte ao arquivo **aplicacao.yaml** e copie a configuração que fizemos, copiando a chave **metadata** em diante e coloque logo abaixo do da chave **template**. Agora como estamos abstraindo o objeto Pod dentro do **Deployment**, não se esqueça de colocar o nome de identificação do Pod dentro das chaves **labels**:

```
  metadata:
    labels:
      name: aplicacao-pod
```

Uma vez feitas essas alterações, salve o arquivo. Agora vamos passar essa configuração para ser implementado no cluster:

```
kubectl create -f deployment.yaml
```

Feito isso, verifique se o pod foi de fato implementado no cluster com o comando:

```
kubectl get pods
```

Ao analisar que o pod de fato está em execução, tente removê-lo do cluster com o comando:

```
kubect1 delete pods [nome do pod]
```

Tente verificar agora quantos pods estão em execução no cluster. Qual o resultado?