04

Faça o que eu fiz na aula

Com o ambiente replicado, precisamos monitorar a quantidade de réplicas que temos para suportar a aplicação do portal de notícias. Para isso vamos alterar o arquivo `statefulset-sistema.yml` e adicionar os parâmetros de monitoração:

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: aplicacao-sistema-statefulset
spec:
  serviceName: aplicacao-sistema-statefulset
  selector:
    matchLabels:
      name: aplicacao-sistema-pod-statefulset
  template:
    metadata:
      labels:
        name: aplicacao-sistema-pod-statefulset
    spec:
      containers:
        - name: container-aplicacao-sistema-statefulset-v4
          image: jnlucas/noticia-alura:v4
          ports:
            - containerPort: 80
          lifecycle:
            postStart:
              exec:
                command: ["sh","enviarMensagens.sh"]

          volumeMounts:
            - name: imagens
              mountPath: /var/www/html/uploads
            - name: sessoes
              mountPath: /tmp
      volumes:
        - name: imagens
          persistentVolumeClaim:
            claimName: permissao-imagens
        - name: sessoes
          persistentVolumeClaim:
            claimName: permissao-sessao
```

Notem que criamos no atributo `lifecycle` , a chamada de um arquivo " `enviarMensagens.sh` ". Nesse arquivo iremos colocar o comando de envio de mensagem via slack. Para isso, vamos entrar no diretório sistema e criar um arquivo com o nome " `enviarMensagens.sh` " com o seguinte conteúdo:

```
#!/bin/bash

curl -X POST -H 'Content-type: alication/json' --data '{"text":"Olá, um novo pod acabou de ser criac
```

Com o arquivo criado, precisamos enviar esse arquivo para para a imagem docker que estamos utilizando dentro do arquivo do controle de réplicas. Vamos então fazer um novo build da imagem que estamos utilizando:

```
docker build -t minha-imagem
docker tag minha-imagem:v4 jnlucas/noticia-alura:v4
docker push jnlucas/noticia-alura:v4
```

Com a imagem atualizada no docker hub, vamos atualizar o arquivo `statefulset-sistema.yml` dentro do cluster gerenciado pelo minikube. Para isso vamos voltar ao terminal e digitar:

```
kubectl delete -f statefulset-sistema.yml
kubectl delete -f servico-statefulset.yml

kubectl create -f statefulset-sistema.yml
kubectl create -f servico-statefulset.yml
```

Pronto! O cluster está atualizado, para verificarmos se está tudo ok com a aplicação, vamos solicitar a url ao minikube com o comando:

```
minikube service servico-aplicacao-sistema --url
```

Agora toda vez que um pod for criado, automaticamente iremos notificar via slack os envolvidos.