**ATIVIDADE PRÁTICA**

**Nome do aluno: Clayton de Almeida Campos**

Entrega por e-mail: [patrick.cardoso@o2b.com.br](mailto:patrick.cardoso@o2b.com.br)

Assunto: Atividade prática final

**Prazo para entrega: 25/03/2022 até as 23:59 Horas.**

**Para realizar as atividades abaixo, você pode escolher a melhor estratégia para utilizar o Kubernetes.**

**Dica:** Faça a leitura da atividade antes de começar a implantar.

**Situação Problema 01**: Você recentemente foi contratado(a) pela empresa O2B e realizou um treinamento que abordou algumas das principais tecnologias utilizadas no mercado.

Após sua primeira semana de trabalho você está atuando em um novo projeto e nele precisa realizar o *deploy* de uma aplicação criada pelo time de desenvolvimento de uma empresa atendida pela O2B.

A atividade consiste em três etapas, conforme abaixo:

1. **Desenvolvimento da aplicação.** Essa tarefa já foi realizada pela equipe de desenvolvimento.
2. **Criação/hospedagem da imagem Docker.** Esta tarefa estará parcialmente preenchida e você deve finalizar e realizá-la.
3. **Implantação da aplicação no Kubernetes.** Esta tarefa estará parcialmente preenchida e você deve finalizar e realizá-la.

**Etapa 1** – **Desenvolvimento da Aplicação.**

Você recebeu o código fonte da aplicação e precisa criar um arquivo chamado **app.rb** e colar o conteúdo abaixo e acrescentar o seu nome completo no a frente do Hello world :

**loop** **do**

puts 'Hello world! - Clayton de Almeida Campos'

STDOUT.**flush**

sleep 1

**end**

**Etapa 2 -** **Criação/hospedagem da imagem Docker.**

Essa etapa está dividida em duas partes: Criar o Dockerfile e fazer o build da imagem e enviá-la para o Docker Hub.

**Parte 01:** Crie o arquivo Dockerfile e cole o conteúdo abaixo:

**FROM** ruby:2.6-alpine

**COPY** . /app

**WORKDIR** /app

**CMD** ["ruby", "app.rb"]

**Parte 02:** Fazer o build da imagem e enviá-la para o docker hub.

É sua responsabilidade definir o nome para a imagem a ser gerada, seguindo a regra que a imagem deve ter como Tag 1.0, conforme exemplo abaixo:

<username>/<imagename>:1.0

Preencha no campo abaixo o comando que deve ser utilizado para fazer o pull da sua imagem:

docker pull claytoncampos/apprb:1.0

Verifique se a nova imagem está disponível no seu Docker Hub, em um repositório público.

**Etapa 03 - Implantação da aplicação no Kubernetes.**

Nesta etapa você precisa criar um arquivo de manifesto capaz de fazer o deploy da aplicação. Cole abaixo uma cópia do seu arquivo.yaml que realizou o deploy da aplicação no cluster Kubernetes.

apiVersion: apps/v1

1 apiVersion: apps/v1

2 kind: Deployment

3 metadata:

4 name: apprb-deployment

5 labels:

6 app: apprb

7 spec:

8 replicas: 3

9 selector:

10 matchLabels:

11 app: apprb

12 template:

13 metadata:

14 labels:

15 app: apprb

16 spec:

17 containers:

18 - name: apprb-container

19 image: claytoncampos/apprb:1.0

20 ports:

21 - containerPort: 80

Em seguida, preencha as informações abaixo:

# Escreva o comando utilizado para aplicar o arquivo de manifesto:

kubectl apply -f Deployment.yaml

# Qual comando pode ser utilizado para listar os pods que foram criados:

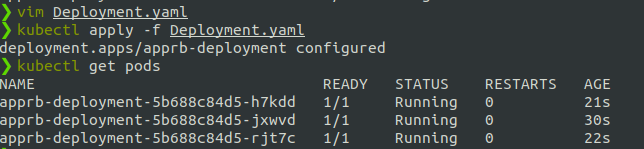
kubectl get pods

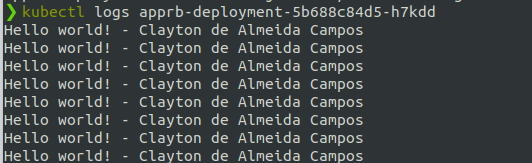
# Consulte o log de um dos pods criados para verificar se a mensagem “Hello World” está sendo exibida. Escreva o comando utilizado para essa tarefa.

kubectl logs apprb-deployment-5b688c84d5-h7kdd

**Se você receber a mensagem nos logs, você concluiu uma parte da nossa atividade.**

* Tire alguns prints da sua aplicação rodando e cole aqui nesta parte do arquivo.

****

****

**Situação Problema 02**: Um novo colega de trabalho solicitou a sua ajuda pois está com dificuldade em criar um Service no Kubernetes para expor uma aplicação que eles estão trabalhando.

Ele te apresentou o arquivo de manifesto com o deploy da aplicação e você precisa ajudá-lo a criar o arquivo de manifesto do Service do tipo adequado, expor a aplicação e testar.

Dado o seguinte arquivo de manifesto:

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deployment-50000

spec:

selector:

matchLabels:

app: metrics

department: engineering

replicas: 3

template:

metadata:

labels:

app: metrics

department: engineering

spec:

containers:

- name: hello

image: "us-docker.pkg.dev/google-samples/containers/gke/hello-app:1.0"

env:

- name: "PORT"

value: "50000"

**Etapa 1 -** **Deploy do arquivo de manifesto e Service.**

**Parte 01:** Faça o deploy do arquivo de manifesto e verifique se os pods foram criados corretamente.

**Parte 02:** Escreva o arquivo de manifesto que expõe a aplicação. Verifique qual é o tipo adequado para expor a aplicação para a rede interna da empresa. Importante, não faça alterações no arquivo de deployment.

Cole no campo abaixo o seu arquivo de manifesto do Service.

apiVersion:

1 apiVersion: v1

2 kind: Service

3 metadata:

4 name: metrics-svc

5 spec:

6 type: NodePort

7 selector:

8 app: metrics

9 ports:

10 - protocol: TCP

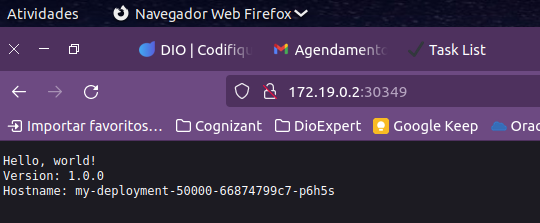
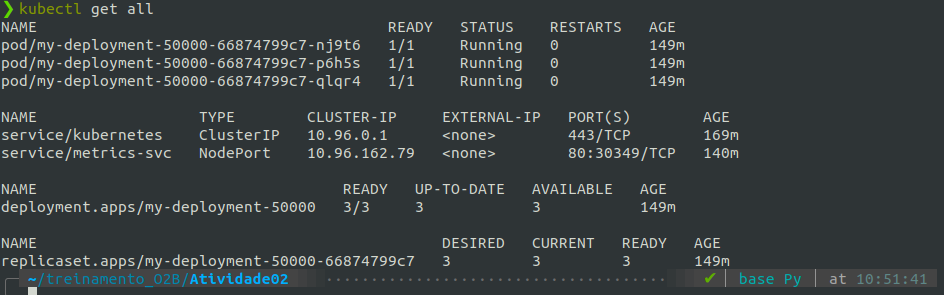
11 port: 80

12 targetPort: 50000

Verifique se o Service foi criado corretamente e faça o acesso à aplicação. Preencha no campo abaixo o comando que deve ser utilizado para acessar a aplicação na sua infraestrutura.

curl 172.19.0.2:30349

Cole um print do seu navegador, acessando a aplicação.



**Etapa 03 - Nova versão da Aplicação**

Após alguns dias, uma nova versão da aplicação foi lançada e o colega solicitou novamente sua ajuda para colocar em produção essa nova aplicação. Faça os ajustes necessários para que a nova aplicação fique disponível.

us-docker.pkg.dev/google-samples/containers/gke/hello-app:2.0

Em seguida, preencha as informações abaixo:

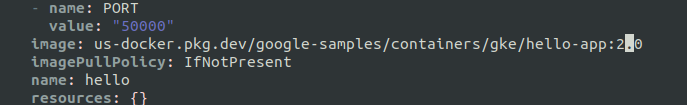
# O que você precisou fazer para que a nova versão ficasse disponível?

#Editar o Deployment

Kubectl edit deployment.apps/my-deployment-50000

Alterar a image

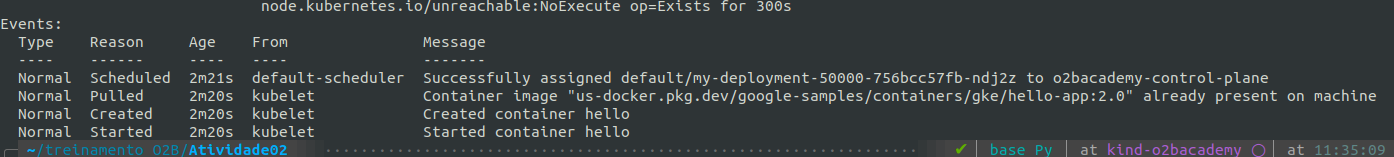
image: us-docker.pkg.dev/google-samples/containers/gke/hello-app:2.0



# Qual comando pode ser utilizado para verificar que os pods estão rodando a nova versão?

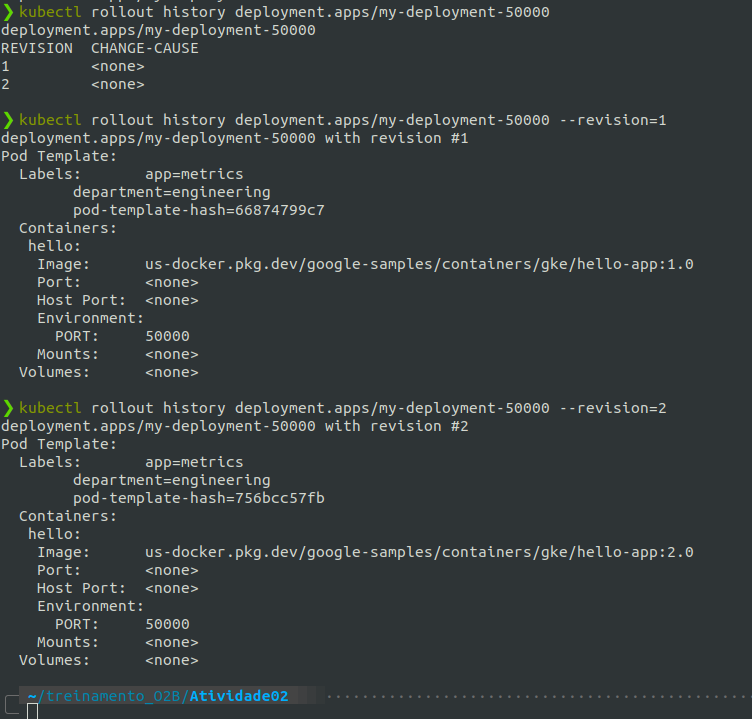
# Execute o comando e cole print da saída, destacando a parte onde mostra a nova versão em execução.

kubectl describe pod/my-deployment-50000-756bcc57fb-ndj2z

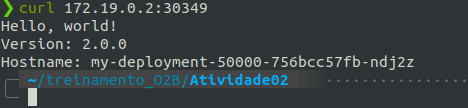


# Qual comando pode ser utilizado para exibir o histórico de lançamentos dessa aplicação?

# Execute o comando e cole print da saída, destacando a parte onde mostra o histórico.



* Tire alguns prints da sua aplicação rodando e cole aqui nesta parte do arquivo.

****

