

Arquitetura-de-sistema-distribuido-com-Kubernetes

aqui está o tutorial passo a passo numerado:

1. Crie um arquivo Dockerfile para construir a imagem do contêiner:

```
FROM nginx:alpine
COPY index.html /usr/share/nginx/html
```

Este Dockerfile usa a imagem oficial do Nginx como base e copia um arquivo index.html para o diretório padrão de documentos do Nginx.

1. Crie um arquivo index.html simples e salve-o na mesma pasta do Dockerfile.

```
<!DOCTYPE html>
<html>
<head>
  <title>Meu Site HTML</title>
</head>
<body>
  <h1>Olá, mundo!</h1>
  <p>Este é um site HTML hospedado em Kubernetes.</p>
</body>
</html>
```

1. Construa a imagem do Docker usando o Dockerfile e envie-a para um registro de contêineres público, como o Docker Hub.

```
$ docker build -t seu-usuario/nome-do-site:latest .
$ docker push seu-usuario/nome-do-site:latest
```

1. Crie um arquivo YAML do Kubernetes para implantar o contêiner.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nome-do-site
```

```

labels:
  app: nome-do-site
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nome-do-site
  template:
    metadata:
      labels:
        app: nome-do-site
    spec:
      containers:
        - name: nome-do-site
          image: seu-usuario/nome-do-site:latest
          ports:
            - containerPort: 80

```

Este arquivo YAML cria um objeto de implantação que define três réplicas do contêiner. Ele também define um seletor de rótulo para garantir que as réplicas corretas sejam implantadas e especifica o contêiner a ser usado e a porta a ser exposta.

1. Aplique o arquivo YAML do Kubernetes usando o comando `kubectl apply`.

```
$ kubectl apply -f nome-do-arquivo.yaml
```

1. Verifique o status do objeto de implantação usando o comando `kubectl get deployment`.

```
$ kubectl get deployment
```

1. Verifique se os pods foram criados usando o comando `kubectl get pods`.

```
$ kubectl get pods
```

1. Crie um objeto de serviço para expor o contêiner na rede.

```

apiVersion: v1
kind: Service
metadata:
  name: nome-do-site

```

```
spec:
  selector:
    app: nome-do-site
  ports:
  - name: http
    protocol: TCP
    port: 80
    targetPort: 80
  type: LoadBalancer
```

Este arquivo YAML cria um objeto de serviço que expõe a porta 80 do contêiner para a rede. Ele também define o tipo de serviço como `LoadBalancer`, o que permite que o Kubernetes crie um balanceador de carga para distribuir o tráfego para as réplicas do contêiner.

1. Aplique o arquivo YAML do serviço usando o comando `kubectl apply`.

```
$ kubectl apply -f nome-do-arquivo.yaml
```

1. Verifique o status do objeto de serviço usando o comando `kubectl get service`.

```
$ kubectl get service
```

Agora você deve ter um site HTML simples hospedado em um cluster do Kubernetes! Você pode acessá-lo usando o endereço IP do serviço ou o nome.