



Configuração do Cluster Kubernetes gerenciado pelo MicroK8s

Luiza Ignacio Gomes

<https://github.com/LuizaGomes03/Cp2-cluster>

RM97797

CP 2 Cluster(2TIAF)

Neste documento, detalharemos o processo de execução de um script em Python em um cluster Kubernetes. A infraestrutura consiste em um cluster Kubernetes gerenciado pelo MicroK8s. O cluster possui as seguintes características:

```
luizagnacio13@master-01:~$ kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
master-01     Ready     <none>   53m   v1.29.2
luizagnacio13@master-01:~$ kubectl get all
NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/kubernetes                 ClusterIP     10.152.183.1   <none>       443/TCP    53m
luizagnacio13@master-01:~$
```

O deploy será realizado no namespace `prod`. Todas as imagens incluídas neste documento apresentarão o nome de usuário `luizagnacio13` para identificação.

Passo a Passo

1. Preparação do Namespace

Primeiro, criamos o namespace onde faremos o deploy:

```
luizagnacio13@master-01:~$ touch 00-namespace.yaml
luizagnacio13@master-01:~$ nano 00-namespace.yaml
```

No arquivo `00-namespace.yaml`, inserimos o seguinte conteúdo:

```
GNU nano 4.8                                00-namespace.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: prod
```

Após salvar o arquivo, aplicamos as alterações:

```
luizagnacio13@master-01:~$ nano 00-namespace.yaml
luizagnacio13@master-01:~$ kubectl apply -f 00-namespace.yaml
namespace/prod unchanged
luizagnacio13@master-01:~$
```

<https://github.com/LuizaGomes03/Cp2-cluster>

2. Verificação do Namespace

Podemos verificar se o namespace foi criado com sucesso:

```
namespace/prod unchanged
luizaignacio13@master-01:~$ kubectl get namespaces
NAME                STATUS    AGE
default             Active    5d3h
kube-node-lease     Active    5d3h
kube-public         Active    5d3h
kube-system         Active    5d3h
prod                Active    96m
staging             Active    112m
```

3. Implantação do Script Python

```
luizaignacio13@master-01:~$ ls -l python.yaml
-rw-rw-r-- 1 luizaignacio13 luizaignacio13 354 Apr 28 23:23 python.ya
luizaignacio13@master-01:~$ python.yaml
python.yaml: command not found
luizaignacio13@master-01:~$ ls -l python.yaml
-rw-rw-r-- 1 luizaignacio13 luizaignacio13 354 Apr 28 23:23 python.ya
luizaignacio13@master-01:~$ cat python.yaml
scripts:
- name: "hello_world.py"
  description: "Um simples script que imprime 'Olá, mundo!'"
  code: |
    print("Olá, mundo!")

- name: "addition.py"
  description: "Um script que calcula a soma de dois números"
  code: |
    def soma(a, b):
        return a + b

    resultado = soma(5, 3)
    print("A soma é:", resultado)
```

<https://github.com/LuizaGomes03/Cp2-cluster>

4. Verificação do Deployment

Podemos verificar se o deployment foi bem-sucedido usando o seguinte comando:

```
Last login: Mon Apr 29 01:55:34 2024 from 35.235.241.145
luizaignacio13@master-01:~$ kubectl get deployments -n prod
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
meu-deployment                      0/1      1              0            167m
```

5. Verificação dos Pods

Podemos verificar os pods em execução no namespace `prod`:

NAME	STATUS	AGE
container-registry	Active	49m
default	Active	5d7h
kube-node-lease	Active	5d7h
kube-public	Active	5d7h
kube-system	Active	5d7h
prod	Active	4h53m
staging	Active	5h9m

6. Logs dos Pods

```
luizaignacio13@master-01:~$ kubectl get pods -n prod
NAME                                READY    STATUS              RESTARTS    AGE
meu-deployment-7c496f9867-5b9rw    0/1      InvalidImageName    0            168m
```

7. Informações Adicionais

Além disso, podemos obter informações detalhadas sobre todos os recursos implantados no namespace `prod` usando o seguinte comando:

```
luizaignacio13@master-01:~$ kubectl get all -n prod
NAME                                READY    STATUS              RESTARTS    AGE
pod/meu-deployment-7c496f9867-5b9rw 0/1      InvalidImageName    0            169m

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/meu-deployment       0/1      1              0            169m

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/meu-deployment-7c496f9867 1          1          0        169m
```

Isso fornecerá detalhes sobre os pods, serviços, deployments, entre outros recursos implantados no namespace `prod`.

8. Informações Adicionais

Além disso, podemos obter informações detalhadas sobre todos os recursos implantados no namespace `prod` usando o seguinte comando:

```
luizaignacio13@master-01:~$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/microbot-66f488557d-d44ww	1/1	Running	0	96m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.152.183.1	<none>	443/TCP	5d3h

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/microbot	1/1	1	1	96m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/microbot-66f488557d	1	1	1	96m

<https://github.com/LuizaGomes03/Cp2-cluster>