11

Mãos à obra: Deploy na Cloud

Nós fizémos na etapa anterior a configuração para acessarmos o cluster no **Google Cloud**, precisamos agora enviar os arquivos para serem implementados, vamos precisar fazer um ajuste na imagem do **MySQL** para poder integrar com o Google Cloud, para isso, vamos criar um outro diretório chamado **prod** e copiar todos os arquivos que fizémos até o momento para esse novo diretório[1].

Feito isso, no diretório **prod** abra o arquivo **statefulset.yaml** (ou statefulset_google_cloud.yaml se estiver no Windows) e altere a versão do MySQL para a 5.5:

```
spec:
     containers:
       - name: container-mysql
         image: mysql:5.5
```

Essa é basicamente a única alteração que nós precisamos realizar, feito isso, vamos criar esses objetos no cluster do Google Cloud, vá até o diretório onde os arquivos do banco estão salvos e comece a enviá-los para o cluster, digite no terminal:

```
kubectl create -f statefulset.yaml
```

Obs: Se estiver no Windows, seria o arquivo **statefulset_google_cloud.yaml**

Na sequência, vamos levar o objeto **Service** que irá abstrair o acesso a esse Pod com o banco de dados:

```
kubectl create -f servico-banco.yaml
```

Posteriormente, vamos passar o arquivo referente a configuração do **PersistentVolumeClaim**, colocamos no terminal:

```
kubectl create -f permissoes.yaml
```

Por fim, vá até o local onde os arquivos da aplicação web estão salvas e envie para o cluster no Google Cloud, vamos enviar o objeto deployment:

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

E na sequência, vamos enviar o objeto **Service** para abstrair o acesso ao objeto Pod que irá possuir o container da aplicação web.

```
kubectl create -f servico-aplicacao.yaml
```

Vá até o **Google Cloud** na aba **Container Engine** e depois em **Workloads** e certifique-se de que os objetos de Deployment e StatefulSet de fato foram criados. Vamos escalonar nossa aplicação web para termos 3 Pods, obtendo assim o mesmo

cenário que testamos com o minikube, digite no terminal:

```
kubectl scale deployment aplicacao-deployment --replicas=3
```

Agora a parte final será criar as tabelas no banco de dados, para isso precisamos saber o nome do Pod que está abstraindo o container com o banco de dados, digitamos no terminal:

```
kubectl get pods
```

Devemos ter ao todo 4 Pods, sendo 3 Pods referentes a aplicação web e 1 Pod do banco de dados, o Pod do banco de dados deverá estar com um nome **statefulset-mysql**

Uma vez identificado o nome do Pod, vamos acessá-lo para criar as tabelas, digite o comando:

```
kubectl exec -it [nome do Pod com o banco] sh
```

Posteriormente, vamos acessar o MySQL com usuário **root**:

```
mysql -u root
```

E depois iremos utilizar o banco **loja**:

```
use loja
```

Crie agora as tabelas no banco com os comandos SQL abaixo:

```
create table produtos (id integer auto_increment primary key, nome varchar(255), preco decimal(
alter table produtos add column usado boolean default false;
alter table produtos add column descricao varchar(255);
create table categorias (id integer auto_increment primary key, nome varchar(255));
insert into categorias (nome) values ("Futebol"), ("Volei"), ("Tenis");
alter table produtos add column categoria_id integer;
update produtos set categoria_id = 1;
```

Vamos agora testar nossa aplicação, vamos acessar o endereço IP do serviço que irá abstrair o acesso a aplicação web. Para isso digite no terminal:

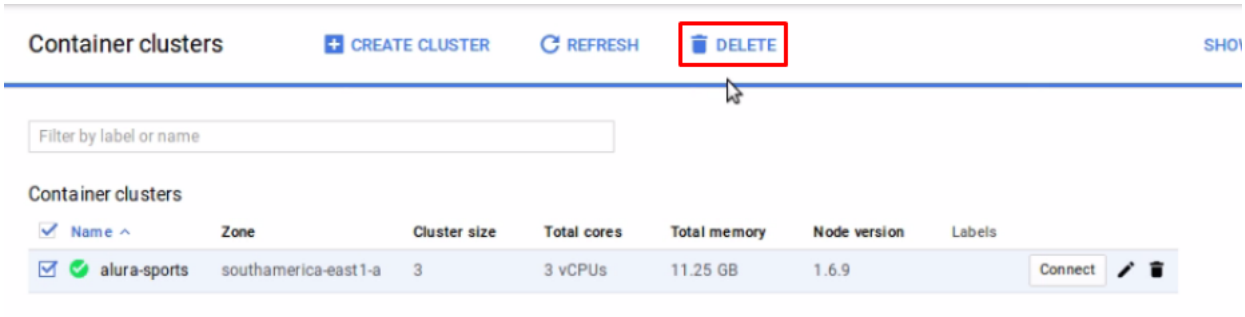
```
kubectl get service
```

Deverá aparecer uma lista com os serviços em execução, copie o endereço IP na coluna **External-IP** que seria referente ao **servico-aplicacao**. Segue exemplo abaixo, o endereço IP é ilustrativo, utilize o endereço IP que aparece para seu serviço:

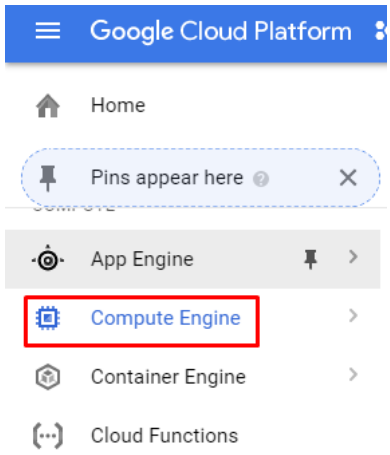
```
alura@alura-preto:~/kubernetes/prod/app$ kubectl get service
NAME                CLUSTER-IP      EXTERNAL-IP     PORT(S)          AGE
db                  10.35.242.96    <none>          3306/TCP         5m
kubernetes          10.35.240.1     <none>          443/TCP          32m
servico-aplicacao   10.35.243.116   35.198.57.152   80:30816/TCP     3m
```

Cole o endereço IP no browser, qual é o resultado? Você consegue acessar a aplicação? O cadastro dos produtos está sendo realizado?

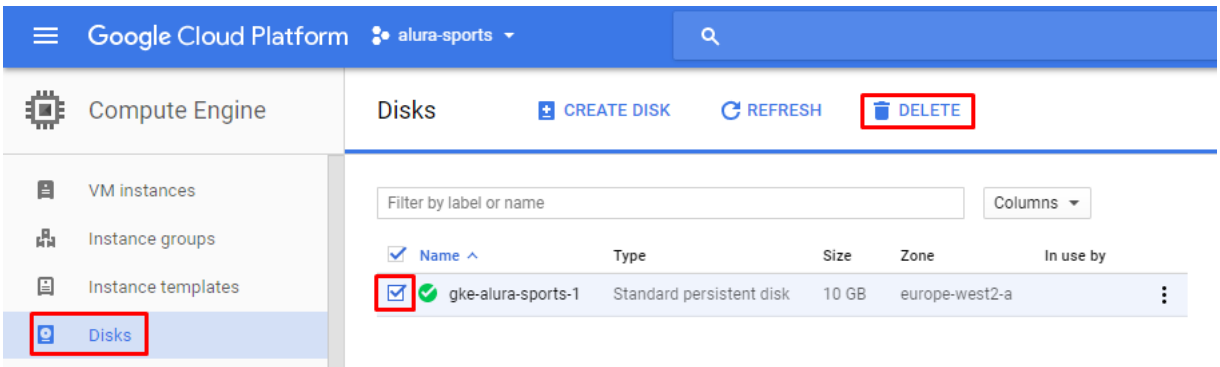
IMPORTANTE: Ao terminar o exercício, vamos remover o que fizémos no Google Cloud para evitar consumir os créditos que foram fornecidos pelo Google Cloud, para isso, volte a aba **Container Engine** e delete o cluster que criamos:



Feito isso, vamos remover também o volume persistente que criamos, vá na aba lateral esquerda e clique em **Compute Engine**:



Na sequência, clique na opção **Disks**, escolha os volumes e delete-os:



[1]**Obs:** No Windows, continuar mantendo tudo no mesmo diretório onde o minikube, o kubectl e os demais arquivos yaml estão salvos, para não perder o arquivo anterior do **statefulset.yaml** sugiro copiar o conteúdo desse arquivo para um novo arquivo e chamá-lo **statefulset_google_cloud.yaml** e fazer o ajuste nesse novo arquivo