

05

## Deploy da aplicação

### Transcrição

O comando para clonar o projeto **python-docs-samples** é o seguinte:

```
git clone https://github.com/GoogleCloudPlatform/python-docs-samples.git
```

[0:01] Endpoint está resolvido, vamos para o back end.

[0:04] Então eu volto aqui no terminal, e a gente faz o seguinte: a gente vai ter que criar aqui uma credencial, uma autorização, para que a gente possa mandar contêineres lá dentro do nosso cluster.

[0:17] Como é que faz isso no ambiente do GCloud? Primeiro, é legal fazer isso aqui: gcloud containers clusters list. Para que? Para ele listar os nomes lá do, o que eu já digitei? Toda vez eu inverto. Container é no singular, não tem problema. Container, clusters list.

[0:45] O que ele faz? Nisso ele vai listar os clusters que a gente tem.

[0:50] Se assim como eu você não guarda os nomes, eu prefiro olhar do que enlouquecer, está aqui.

[0:56] Por que esse comando todo? Por que essa saída aqui? Para a gente digitar o seguinte comando, agora sim de autorização: gcloud container clusters get-credentials. O nome do cluster, a vida assim é mais fácil, e a zona. Zone, qual é a zona que a gente utilizou? É essa daqui. Honestamente eu já nem lembrava disso.

[1:39] O que mais? Ainda tem que passar o nome do projeto. Project, e o nome dele. Esse eu decorei.

[1:57] Então ficou um comando bonito aqui, de quase duas linhas, vamos dar um enter para ver se ele grita, se ele faz.

[2:09] Já está lá autorizado. Então ele vai criar as configs, está resolvido.

[2:13] Agora sim, vamos trabalhar aqui no back end.

[2:19] Então com o endpoint resolvido, voltei para o browser, porque eu quero te mostrar o seguinte: google cloud python samples. Eu acho que é isso aqui.

[2:33] O link está também na plataforma, mas só para você entender da onde eu peguei a informação, e que informação.

[2:43] Back end agora, Kubernetes, já falei isso diversas vezes. Você tem que preparar a tua aplicação para rodar num contêiner, e aí?

[2:53] Eu já peguei a aplicação, aquela airport name, já dockerizei, esse termo é muito engraçado, dockerizei a aplicação, ela já está no container, vou te mostrar o link da imagem.

[3:05] Mas independente da imagem, lá nos exemplos do google, e isso aqui é legal para baixar, porque tem vários exemplos para você estudar as plataformas todas, e os serviços, tem aqui o endpoints, getting started, deixa eu ver se é esse aqui.

[3:33] Exatamente. Então vamos analisar juntos isso aqui, com calma.

[3:37] O que é isso daqui? Isso daqui é a preparação da aplicação para rodar no Kubernetes.

[3:44] Mas e daí, Ricardo? Ele dividiu, isso aqui é um docker compose, ele dividiu isso aqui de duas maneiras, então vamos analisar.

[3:54] Ele vai subir o contêiner chamado, cadê aqui embaixo? Ele sobe dois contêiners, um contêiner esp e outro contêiner aqui, echo.

[4:05] Esse echo é aplicação de exemplo que ele utilizou, o esp é aquele camarada da figura que eu te mostrei. É o proxy que vai tratar lá no endpoint para direcionar o tráfego para a nossa aplicação.

[4:20] Então o que a gente tem que fazer? A gente vai utilizar esse exemplo aqui, que eu vou passar já com você, e vamos utilizar a imagem do Google, lá de esp, que já é uma imagem pronta para trabalhar com proxy, e aqui no segundo contêiner que a gente subir, na segunda imagem, a gente aponta para a imagem que eu criei lá do airport.

[4:47] Então a minha ideia também, e sugestão para você é não ficar amarrado à aplicação que a gente utilizou.

[4:54] Pega a tua aplicação, depois que você concluir aqui os teus estudos, pega a tua aplicação, transforma ela num contêiner, usa essa estrutura aqui, e faz o deploy com a tua própria aplicação.

[5:06] Então baseado nesse exemplo aqui, é o que a gente vai ver aqui dentro dos nossos arquivos, que eu já deixei o link para baixar.

[5:19] Quando você pegou o arquivo, já tinha esse deploy aqui.

[5:23] Então vamos passar rapidinho, para a gente verificar o que é interessante.

[5:28] Eu troquei alguns nomes aqui, esp airportnames, isso daqui está tudo igual o default. Eu só troquei aqui na verdade o airport pelo echo, só para acertar.

[5:45] Contêiners, aqui. Esse contêiner é o esp do Google, ele sobe o contêiner na porta oito oito um, backend está na 80 80, e por aí vai.

[6:00] Serviço, aqui a gente tem que apontar para o nosso servidor. Então você vai editar isso aqui, e vai ajustar para o seu projeto.

[6:10] Qual foi o endpoint que a gente criou? Está lá no browser.

[6:19] Lembrando aqui do domínio que mudou, trouxe ele para cá, ajustou o seu código.

[6:32] Airportname, endpoints, está lá, beleza.

[6:35] E aí? Isso aqui tudo certinho, o que mais? Qual foi a mudança? Aqui, contêiner esp, e esse comentário é do arquivo original, eu dei, e tem o segundo contêiner aqui, que é o airport names, e ele está onde? É essa imagem que a gente está utilizando.

[6:56] Essa imagem nada mais é do que a imagem containerizada.

[7:01] Então peguei, gerei a imagem, coloquei lá no hub do docker para quando você rodar isso daqui ele já vai lá e já pega a imagem, já faz a instalação.

[7:13] Então esse é o arquivinho que a gente vai utilizar.

[7:16] Salvamos o arquivo, e a gente pode fazer deploy dele.

[7:20] Com o deploy, o que ele vai fazer? Vamos lá. Ele vai criar todo o contêiner com a nossa aplicação. Isso tudo dentro do cluster, e o cluster é o Kubernetes lá do Google.

[7:34] Então o arquivo está aqui, cd endpoints, está aqui deploy, então vamos lá, vamos fazer o deploy.

[7:43] Kubectl apply menos F, então vamos fazer o deploy, ver se ele ai funcionar direitinho.

[7:56] O serviço foi criado, tudo aqui.

[8:00] O que a gente precisa fazer? Testar esse pacotão todo.

[8:05] Próximo vídeo a gente faz o teste.