

02

## Kubernetes Engine e Cluster

### Transcrição

Observação: A *Google Container Engine API* se chama *Kubernetes Engine API* e a *Container Engine* se chama agora *Kubernetes Engine*.

No vídeo anterior testamos a aplicação na máquina local, e vimos que ela está funcionando como deveria. Agora, daremos um passo adiante e levaremos a aplicação para um ambiente de produção utilizando o **Google Cloud**.

Por ser um projeto desenvolvido pelo Google, o Kubernetes possui melhor integração com este projeto. Sendo assim, acessaremos [este link](http://cloud.google.com) (<http://cloud.google.com>), que possui como opção uma avaliação gratuita.

Na página a que somos redirecionados, existem maiores informações, como o fato do Google nos oferecer U\$300 de crédito para que esta versão seja usada por até 12 meses. Marcaremos então que estamos no Brasil e aceitamos os termos de uso e condições.

Na parte de cadastro, precisaremos informar se somos pessoa física ("Individual") ou jurídica ("Business") e, a partir disto, preencher os campos referentes a CPF ou CNPJ, endereço completo e dados sobre o cartão de crédito - não para cobranças automáticas após a expiração do período de avaliação, e sim para evitar a criação de contas por robôs, ou algo do tipo.

Feito o cadastro, somos redirecionados à plataforma do Google Cloud, após o qual precisaremos criar o projeto em que trabalharemos. Para isto, selecionaremos nosso nome de usuário na parte superior da plataforma e, na janela que se abre, clicaremos em +, que indica "Create project".

Podemos colocar qualquer nome de nossa preferência; usaremos "alura-sports" e clicaremos em "Create". Isso nos leva ao painel geral, em que faremos algumas configurações iniciais.

Para trabalharmos em nosso projeto, que se baseia em containers e é gerenciado pelo Kubernetes, focaremos na aba "APIs" e selecionaremos "Go to APIs overview". Depois, na nova página, clicaremos em "Enable APIs and services".

Em "Library", que se abre em seguida, precisamos justamente da API para habilitarmos a aplicação, que está trabalhando com containers. No campo de pesquisa, digitaremos "container engine", ao que aparecerá a opção "Google Container Engine API".

Isto será utilizado para habilitar aplicações que trabalham com containers gerenciados pelo Kubernetes. Clicaremos nela e em "Enable", localizado ao lado do nome da API.

Feita esta configuração, precisaremos criar o *cluster*, sendo que para isto poderemos usar linhas de comando ou o próprio painel, o *dashboard* do Google Cloud, disponível na plataforma.

Ao lado esquerdo de "Google Cloud Platform", no topo da página, clicaremos no ícone que representa "Products & services". Uma aba lateral se abre, e selecionaremos "Container Engine", que nos oferece a opção "Create Cluster".

Poderíamos colocar um nome qualquer, no caso escolheremos "alura-sports" também. Se quisermos, podemos incluir uma descrição, e a zona configurada neste *cluster* será "southamerica-east1-a".

Abaixo, existem as configurações das máquinas a serem utilizadas pelo *cluster*, o tamanho, a quantidade de máquinas com que queremos trabalhar - que vamos manter 3. Por fim, clicaremos em "Create".

Seremos redirecionados à tela com "*Container clusters*", e este processo pode demorar alguns minutos, já que o Google está instanciando as máquinas para formar o *cluster*.

Agora precisaremos nos comunicar com o *cluster*. Ainda em "*Container clusters*", clicaremos no botão "Connect" e copiaremos a primeira linha de comando que o Google nos fornece na nova janela que se abre.

Assim, conseguiremos utilizar o `kubectl` que estava apontando ao `minikube` para este levá-lo ao Google Cloud. Clicaremos em "OK" e abriremos o terminal para colar a linha de comando copiada. Para confirmar que isto foi feito, usaremos `kubectl config get-contexts`, com que perguntaremos sobre as configurações que existem na lista.

Então, temos o `minikube`, com que estávamos trabalhando, e agora também o asterisco, que informa que estamos apontando ao *cluster* do Google Cloud. Ou seja, já estamos nos comunicando com ele, e todos os comandos que fizermos com `kubectl` significa que estamos rodando-o no *cluster* localizado no Google Cloud.

Estávamos trabalhando com o `minikube`, bem como com os *pods*. Para nos certificarmos de que estamos de fato no Google Cloud, colocaremos `kubectl get pods`, ao que retorna-se a mensagem `No resources found.` Maravilha, não configuramos nada no *cluster* do Google Cloud.

Realizamos assim a configuração inicial, na sequência levaremos nossos arquivos YAML ao *cluster* que acabamos de criar. Vamos lá?