

Criando objeto serviços

Transcrição

No vídeo anterior, criamos um escalonamento da nossa aplicação. Saímos do ambiente em que tínhamos apenas um `Pod` e passamos a ter 3 deles, cada qual contendo o container web.

No entanto, como vimos, os objetos `Pod` estão em constante alteração: é possível aumentar ou diminuir sua quantidade, ou que sejam encerrados por algum problema e, no caso, o Kubernetes precisará criar um novo objeto para substituí-los, o que ocasionará em um novo endereço IP.

Por conta disto, não conseguimos acessar os novos objetos `Pod` diretamente, sendo necessário abstrair o acesso a eles em uma camada, ou objeto mais estável, de serviço, em inglês, "*Service*".

A ideia é que ele faça a abstração do acesso para os objetos `Pod`. Podemos configurá-lo para que ele atue como se fosse um **balanceador de cargas**, dividindo a quantidade de acessos e requisições que nossa aplicação terá, por estes *pods* que criamos.

Pelo fato do objeto `Service` ser mais estabilizado, nós, usuários, iremos utilizá-lo para podermos acessar a aplicação.

Como já sabemos, para criar um novo objeto no Kubernetes, precisaremos de outro arquivo de configuração YAML. No diretório `kubernetes` no terminal, digitaremos `atom servico-aplicacao.yaml` para que o Atom crie este arquivo.

Similar às demais etapas, teremos a estrutura a seguir:

```
apiVersion: v1
kind: Service
metadata:
  name: servico-aplicacao
spec:
  type: LoadBalancer
  ports:
    - port: 80
  selector:
    name: aplicacao-pod
```

Acima, é especificado que `Service` precisará acessar uma porta de comunicação da nossa aplicação web, a `80`, e que é necessário estabelecer um vínculo entre o objeto `Service` e o `Pod`, informando-se que o acesso deve ser abstraido a partir da seleção do critério utilizado por ele para encontrar os *pods*.

Atenção com a chave `labels`, pois estes valores precisam ser iguais! Neste caso em específico, por exemplo, o objeto `Service` selecionará todos os *pods* cujo nome seja "aplicacao-pod".

Tendo criado `Service`, queremos que ele seja implementado no nosso *cluster*. Vamos salvar o arquivo e voltar ao terminal para digitar o comando `kubectl create -f servico-aplicacao.yaml`. Obteremos a informação de que ele foi criado com sucesso.

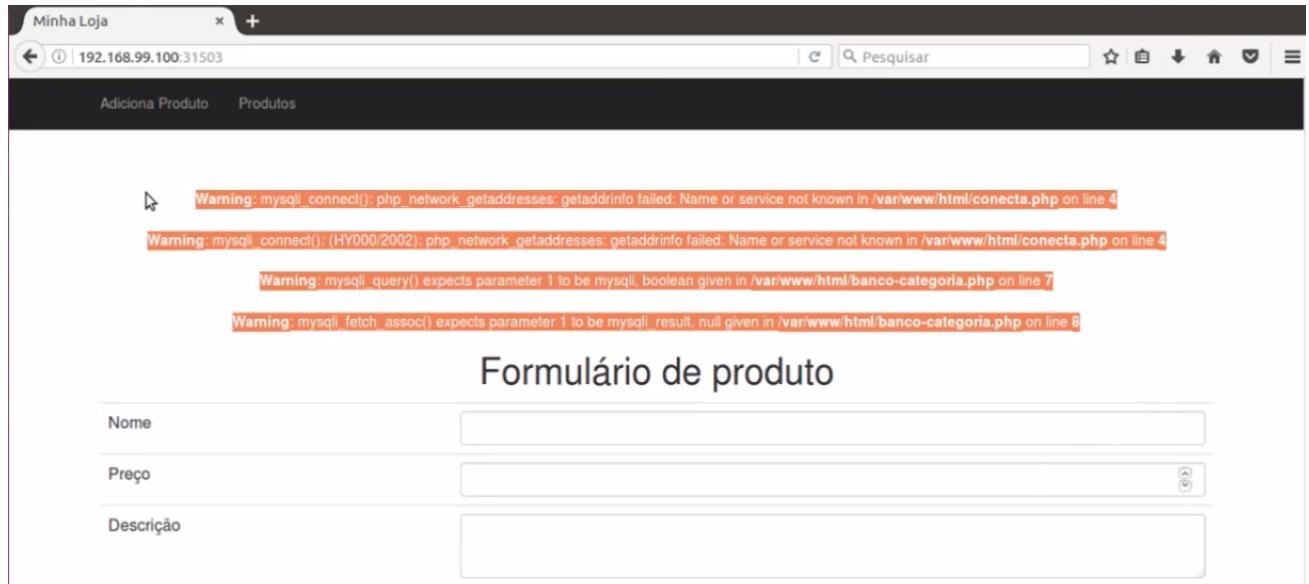
Agora, vamos pedir ao `minikube` para que seja passada a URL que devemos acessar, a qual está sendo vinculada ao objeto `Service` que acabamos de configurar:

```
minikube service servico-aplicacao --url
```

Com isto obteremos esta URL de acesso:

<http://192.168.99.100:31503>

Vamos copiá-la e colar na barra de endereços do *browser*. Lembrando que agora nossos *pods* estão sendo abstraídos pelo objeto `Service`, que estamos usando para acessar a aplicação.



A aplicação foi exibida! No entanto, com mensagens de erro, pois nossa aplicação web está se comunicando com o banco de dados que ainda não configuramos. Precisaremos fazê-lo na sequência!