



Introdução ao Kubernetes

1.

Panorama Geral do Kubernetes

Kubernetes é uma plataforma de código aberto projetada para automatizar a implantação, escalonamento e operação de aplicações em contêineres. Ele gerencia clusters de máquinas físicas ou virtuais e executa aplicações em contêineres, permitindo que as aplicações sejam escaladas de maneira eficiente.

No Google Cloud, o Kubernetes é oferecido como um serviço gerenciado chamado Google Kubernetes Engine (GKE). O GKE simplifica a configuração, gerenciamento e operação de clusters Kubernetes, aproveitando a infraestrutura global e escalável do Google.

Contêineres: Unidades leves e portáteis que agrupam aplicações e suas dependências, garantindo que elas funcionem de maneira consistente em qualquer ambiente.

Pods: A menor unidade de implantação no Kubernetes, que pode conter um ou mais contêineres.

Clusters: Um conjunto de máquinas (nós) que executam os aplicativos em contêineres gerenciados pelo Kubernetes.

Exemplo Contextual: Imagine uma aplicação de e-commerce que precisa lidar com flutuações de tráfego. Com Kubernetes, você pode escalar automaticamente as instâncias de sua aplicação durante períodos de alta demanda e reduzi-las quando o tráfego diminuir, otimizando custos e recursos.

2. Configuração de um Cluster no Kubernetes Engine

Para começar a trabalhar com Kubernetes no Google Cloud, você primeiro precisa configurar um cluster no Google Kubernetes Engine (GKE).

Passos para Configurar um Cluster no GKE:

Criação do Projeto no Google Cloud: Primeiro, é necessário ter um projeto ativo no Google Cloud.

Ativação do Kubernetes Engine API: A API do Kubernetes Engine deve estar ativada para poder criar e gerenciar clusters.

Criação do Cluster:

Acesse o Google Cloud Console.

Vá para o Kubernetes Engine > Clusters.

Clique em “Criar cluster”.

Configure o cluster, selecionando o número de nós, zona, tipo de máquina e outras configurações específicas.

Clique em “Criar”.

Exemplo Prático: Suponha que você precise testar uma nova aplicação web. Com o GKE, você pode rapidamente criar um cluster de três nós para hospedar a aplicação, definindo o tamanho e a localização do cluster conforme as necessidades do teste.

3. Implementação Inicial - Hello-Server

Depois que o cluster estiver configurado, o próximo passo é implementar uma aplicação básica para testar o ambiente.

Passos para Implementar o Hello-Server:

Criação de um Pod:

Crie um arquivo de manifesto YAML para o pod, que define a configuração do contêiner (por exemplo, uma aplicação simples como nginx).

Use o comando `kubectl apply -f hello-server.yaml` para implantar o pod no cluster.

Expondo o Pod:

Para acessar a aplicação fora do cluster, crie um serviço usando `kubectl expose pod hello-server --type=LoadBalancer --port=80`.

O Kubernetes atribuirá automaticamente um endereço IP externo ao serviço, tornando-o acessível.

Hands-on: Este exemplo ilustra a implantação de uma aplicação básica chamada Hello-Server, que responde com "Hello, World!" a cada solicitação HTTP. É uma maneira eficaz de validar que o ambiente Kubernetes está funcionando corretamente.

4. Criação de um Serviço Kubernetes

Um Serviço no Kubernetes é uma abstração que define um conjunto lógico de Pods e uma política para acessá-los.

Tipos de Serviços:

ClusterIP: Exponha o serviço em um IP interno do cluster, acessível apenas dentro do cluster.

NodePort: Exponha o serviço em uma porta específica de cada nó no cluster.

LoadBalancer: Exponha o serviço externamente usando um balanceador de carga de um provedor de nuvem.

Passos para Criar um Serviço:

Crie um arquivo YAML para definir o serviço.

Especifique o tipo de serviço (ClusterIP, NodePort, ou LoadBalancer) e a porta.

Implante o serviço com o comando `kubectl apply -f service.yaml`.

Exemplo Contextual: Considere uma aplicação de back-end que precisa ser acessada por várias aplicações front-end. Você pode criar um Serviço Kubernetes do tipo ClusterIP para permitir que apenas os componentes internos do cluster acessem essa aplicação, garantindo segurança e isolamento.

5. Limpeza e Remoção do Cluster

Depois de concluir os testes e a implantação, é importante limpar os recursos para evitar custos desnecessários.

Passos para Limpeza e Remoção:

Remoção dos Pods e Serviços:

Use `kubectl delete pod hello-server` e `kubectl delete service hello-server` para remover os recursos criados.

Deleção do Cluster:

No Google Cloud Console, vá para Kubernetes Engine > Clusters.

Selecione o cluster que deseja excluir e clique em “Excluir”.

Exemplo Prático: Se você implantou vários recursos para um projeto temporário, a remoção completa do cluster e seus recursos após a conclusão garantirá que você não incorrerá em custos adicionais de infraestrutura.

Arquivos yaml necessários:

hello-server.yaml:

Define um Pod chamado hello-server.

Utiliza a imagem Docker do nginx como o contêiner da aplicação.

O contêiner expõe a porta 80 para servir o tráfego HTTP.

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: hello-server
5    labels:
6      app: hello-server
7  spec:
8    containers:
9      - name: hello-container
10       image: nginx:latest
11       ports:
12         - containerPort: 80
```

service.yaml:

Cria um Serviço Kubernetes chamado hello-service.

O Serviço seleciona o Pod com a label app: hello-server.

Expõe a porta 80 do Pod para tráfego externo.

O tipo LoadBalancer faz com que o serviço seja acessível

externamente através de um IP público provisionado

automaticamente pelo Google Cloud.

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: hello-service
5  spec:
6    selector:
7      app: hello-server
8    ports:
9      - protocol: TCP
10       port: 80
11       targetPort: 80
12     type: LoadBalancer
```

Você pode aplicar esses arquivos no seu cluster usando os comandos:

kubectl apply -f hello-server.yaml

kubectl apply -f service.yaml

Esses comandos criarão o Pod e o Serviço no seu cluster Kubernetes.

Conteúdo Bônus

Para complementar seus estudos, indico o seguinte artigo:

Artigo: “Uma Introdução ao Kubernetes”

Este tutorial da DigitalOcean escrito por Justin Ellingwood and Fernando Pimenta fornece uma visão geral sobre o Kubernetes, seus componentes principais e casos de uso.

Referências Bibliográficas

BASSO, D. E. **Administração de Redes de Computadores**. Contentus, 2020.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma Abordagem Top-Down**. 8. ed. Pearson, 2021.

MARINHO, A. L.; CRUZ, J. L. da. (Orgs.). **Desenvolvimento de Aplicações para Internet**. 2. ed. Pearson, 2020.

PUGA, S.; RISSETTI, G. **Lógica de Programação e Estruturas de Dados, com Aplicações em Java**. 3. ed. Pearson, 2016.

ROHLING, L. J. **Segurança de Redes de Computadores**. Contentus, 2020.

SILVA, C. F. da. **Projeto Estruturado e Gerência de Redes**. Contentus, 2020.

TANENBAUM, A. S.; FEAMSTER, N.; WETHERALL, D. J. **Redes de Computadores**. 6. ed. Pearson, 2021.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas Digitais: Princípios e Aplicações**. 12. ed. Pearson, 2018.

Ir para exercício