THIAGO ADRIANO

SOFTWARE ARCHITECTURE

KUBERNETES

AULA 08

SUMÁRIO

O QUE VEM POR AÍ?

Nesta aula, você vai aprender sobre o Horizontal Pod Autoscaler (HPA) no Kubernetes. Por meio de alguns exemplos, conhecerá esta ferramenta poderosa, que permite que os clusters do Kubernetes ajustem automaticamente a escala dos Pods com base na demanda de tráfego.

HANDS ON

Nesta aula, os professores apresentarão o Horizontal Pod Autoscaler (HPA) no Kubernetes por meio de exemplos práticos. Eles mostrarão como o HPA pode ser usado para ajustar automaticamente a escala dos Pods com base na demanda de tráfego em um aplicativo.

SAIBA MAIS

HPA

O Horizontal Pod Autoscaler (HPA) é um recurso do Kubernetes que permite ajustar automaticamente a escala de Pods com base nas métricas de utilização de recursos. Isso significa que o HPA pode aumentar ou diminuir o número de réplicas de um Pod em um cluster Kubernetes com base na demanda do aplicativo, ajudando a manter o desempenho e a disponibilidade do aplicativo.

O HPA funciona monitorando as métricas de utilização de recursos dos Pods, como CPU e memória, comparando essas métricas com limites predefinidos. Quando a utilização de recursos ultrapassa esses limites, o HPA aumenta o número de réplicas de um Pod para lidar com a demanda do aplicativo. Quando a utilização de recursos diminui, o HPA diminui o número de réplicas do Pod para economizar recursos.

Ele pode ser configurado para usar diferentes métricas de recursos, incluindo CPU, memória e uso personalizado de métricas. Além disso, é possível definir limites de escala, que especificam o número mínimo e máximo de réplicas que o HPA pode ajustar.

O uso do HPA pode trazer diversos benefícios para o gerenciamento de aplicativos em um cluster Kubernetes. Em primeiro lugar, ele permite que os aplicativos sejam dimensionados automaticamente de acordo com a demanda do usuário, ajudando a garantir que o desempenho e a disponibilidade do aplicativo sejam mantidos em níveis aceitáveis.

Além disso, o HPA pode ajudar a reduzir os custos operacionais, permitindo que os recursos sejam alocados de maneira mais eficiente. Por exemplo, se um aplicativo estiver sendo executado com um número excessivo de réplicas, ele pode diminuir o número de réplicas para economizar recursos.

A seguir você tem uma visão geral das métricas suportadas pelo HPA:

- CPU: a métrica de CPU é a mais comum e é baseada na utilização da CPU pelos Pods. O HPA monitora a utilização de CPU dos Pods e ajusta a escala de réplicas com base em limites predefinidos.

- Memória: a métrica de memória é baseada na utilização de memória pelos Pods. O HPA monitora a utilização de memória dos Pods e ajusta a escala de réplicas com base em limites predefinidos.
- Uso personalizado de métricas: o HPA também suporta o uso de métricas personalizadas, que podem ser definidas pelo usuário para monitorar a utilização de recursos específicos para um aplicativo. Essas métricas personalizadas podem ser baseadas em estatísticas de aplicativos ou em indicadores de desempenho.
- Medidas externas: ele também pode usar métricas externas, como as fornecidas por sistemas de monitoramento de terceiros, como o Prometheus, para ajustar a escala de réplicas com base nas métricas de utilização de recursos.
- E/S de disco: em alguns casos, a E/S de disco pode ser uma métrica útil para ajustar a escala de réplicas. O HPA pode monitorar a E/S de disco de um Pod e ajustar a escala de réplicas com base em limites predefinidos.

Cada métrica tem suas próprias vantagens e desvantagens e, por isso, a escolha da métrica certa dependerá do aplicativo e das necessidades específicas do ambiente. Por exemplo, a métrica de CPU pode ser útil para aplicativos que requerem muita CPU, enquanto a métrica de memória pode ser mais importante para aplicativos que usam muita memória.

Para ficar mais evidente, vejamos alguns exemplos práticos demonstrando como usar o Horizontal Pod Autoscaler (HPA) no Kubernetes para ajustar a escala de um aplicativo com base na utilização de recursos.

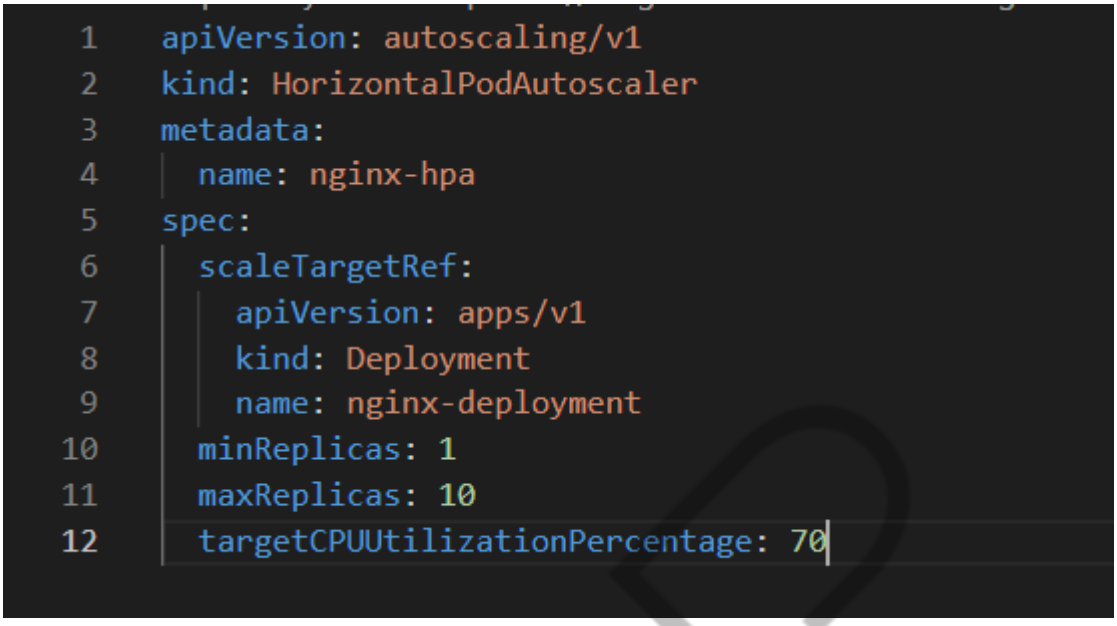
Primeiro, crie um Deployment no Kubernetes para o aplicativo que você deseja escalar. Na figura 1 – “Exemplo de arquivo de Deployment”, há um exemplo de um arquivo YAML para criar um Deployment simples que executa um container nginx.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

Figura 1 – Exemplo de arquivo de Deployment
Fonte: Elaborado pelo autor (2023)

Este Deployment define um único Pod com um container nginx. Note que o número de réplicas está definido como 1.

Em seguida, crie um objeto HPA para o Deployment. Na figura 2 – “Exemplo de arquivo HPA para deployment”, há um exemplo de um arquivo YAML para definir um HPA que monitora a utilização de CPU e ajusta a escala de réplicas entre 1 e 10.



```yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: nginx-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx-deployment
  minReplicas: 1
  maxReplicas: 10
  targetCPUUtilizationPercentage: 70
```

Figura 2 – Exemplo de arquivo HPA para deployment
Fonte: Elaborado pelo autor (2023)

Conforme mencionado acima, este HPA monitora a utilização de CPU e ajusta a escala de réplicas entre 1 e 10 com um objetivo de utilização de CPU de 70%.

Para testar, você pode executar teste de carga no aplicativo para aumentar a utilização de CPU. Existem algumas ferramentas que podem ajudar neste teste, como o K6 ou o Jmeter.

Depois que a carga iniciar, verifique se o HPA aumentou o número de réplicas do Pod. Você pode fazer isso usando o comando ***kubectl get hpa.***

Depois de verificar o HPA, reduza o teste de carga no aplicativo e verifique se o HPA reduziu o número de réplicas do Pod.

Como você pôde ver nos passos anteriores, o HPA é uma ferramenta poderosa para ajustar automaticamente a escala de aplicativos em um cluster Kubernetes com base nas métricas de utilização de recursos.

Ao definir um HPA e executar um teste de carga, é possível ver como o HPA ajusta automaticamente o número de réplicas do Pod para lidar com a demanda do aplicativo.

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, você aprendeu sobre o Horizontal Pod Autoscaler (HPA) na teoria e na prática. Os professores apresentaram os diferentes tipos de métricas suportadas pelo HPA, como CPU, memória e métricas personalizadas, e como configurar as métricas para o seu próprio aplicativo.

O que achou do conteúdo? Conte-nos no Discord! Estamos disponíveis na comunidade para tirar dúvidas, fazer networking, enviar avisos e muito mais.

REFERÊNCIA

DOBIES, J. **Operadores do Kubernetes: Automatizando a Plataforma de Orquestração de Contêineres.** [s.l.]: Novatec, 2020.

PALAVRAS-CHAVE

Auto Scaling. HPA. Kubernetes.