

THIAGO ADRIANO

POSTECH

SOFTWARE ARCHITECTURE

KUBERNETES

AULA 06

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON.....	4
SAIBA MAIS	5
O QUE VOCÊ VIU NESTA AULA?	10
REFERÊNCIAS	11

EMENDAS

O QUE VEM POR AÍ?

Nesta aula, os(as) alunos(as) serão apresentados aos volumes no Kubernetes. Através de alguns exemplos práticos, eles aprenderão como os volumes permitem que os dados sejam armazenados fora do container, facilitando o armazenamento e o gerenciamento de dados de maneira independente do ciclo de vida dos containers.

EMANIP

HANDS ON

Nesta aula prática, os docentes apresentarão os tipos de volumes que podem ser utilizados quando trabalhamos com o Kubernetes.



SAIBA MAIS

Conforme mencionamos nas aulas anteriores, o Kubernetes é uma plataforma de orquestração de containers que facilita a gestão de aplicações e serviços escaláveis. No entanto, o armazenamento de dados é um desafio comum em ambientes de containers.

Para resolver este problema, o Kubernetes oferece algumas soluções através dos conceitos de volumes (VOL), persistent volumes (PV), persistent volume claims (PVC) e storage classes (SC).

Nesta aula, vamos explorar o que são e como usar cada um desses conceitos no Kubernetes.

VOLUMES

Iniciando por volumes (VOL), eles são uma parte essencial do Kubernetes, pois fornecem uma maneira consistente e portátil de armazenar dados persistentes em um cluster. O uso de volumes também torna o gerenciamento de dados mais fácil e escalável. Os volumes são particularmente úteis em cenários em que os containers precisam armazenar dados que sobrevivem a reinicializações ou redimensionamentos de Pods, como em um cenário onde temos um container de banco de dados e, caso o container crashe e suba novamente, precisemos das informações que foram perdidas.

O Kubernetes suporta vários tipos de volumes, cada um com suas próprias propriedades e funcionalidades. Alguns dos tipos mais comuns de volumes são:

- emptyDir: um volume vazio que é criado quando um Pod é iniciado e é excluído quando o Pod é encerrado. Ele é geralmente usado para compartilhar arquivos temporários entre containers em um Pod.
- hostPath: um volume que monta um diretório ou arquivo do nó do cluster no container. É útil quando os containers precisam acessar arquivos ou diretórios no nó do cluster.

- `persistentVolumeClaim`: um volume que fornece armazenamento persistente para os containers em um Pod. É usado quando os dados precisam sobreviver à reinicializações ou redimensionamentos de Pods.

Para evidenciarmos mais a situação, veja a figura 1 – Exemplo de criação de volume, um exemplo que demonstra como criar um volume do tipo `emptyDir`.

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: test-pd-vol
5  spec:
6    containers:
7      - image: registry.k8s.io/test-webserver
8        name: container-fiap
9        volumeMounts:
10       - mountPath: /cache
11         name: cache-volume
12    volumes:
13      - name: cache-volume
14        emptyDir: {}
```

Figura 1 – Exemplo de criação de volume
Fonte: Elaborado pelo autor (2023)

Neste exemplo, o Pod é criado com um container que usa um volume do tipo `emptyDir`. O volume é montado em `/cache` dentro do container, e qualquer arquivo armazenado neste diretório será compartilhado entre os containers no Pod.

O `persistent volume (PV)` é um volume que sobrevive à reinicialização ou exclusão do Pod em que foi criado. Esses volumes podem ser reutilizados em vários Pods e têm um ciclo de vida separado do Pod. O Kubernetes permite que os

administradores do cluster provisionem persistent volumes e forneçam acesso a esses volumes a partir de um Pod.

O Kubernetes suporta vários tipos de persistent volumes. Temos uma lista dos tipos mais comuns:

- NFS: um protocolo de compartilhamento de arquivos em rede que permite que vários nós acessem o mesmo armazenamento.
- iSCSI: um protocolo de armazenamento em rede que permite que um sistema de armazenamento remoto seja montado em um nó do cluster.
- hostPath: um volume que monta um diretório ou arquivo do nó do cluster no container. É útil quando os containers precisam acessar arquivos ou diretórios no nó do cluster.
- EBS (Elastic Block Store): um serviço de armazenamento em nuvem oferecido pela AWS que permite criar volumes de armazenamento persistente.

Conforme mencionado, os administradores do cluster precisam criar um objeto de persistent volume e, em seguida, definir como ele será usado pelos pods. O exemplo da figura 2 – Exemplo de criação PV ilustra como criar um persistent volume:

```
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: meu-pv-fiap
5  spec:
6    capacity:
7      storage: 5Gi
8    accessModes:
9      - ReadWriteOnce
10   persistentVolumeReclaimPolicy: Retain
11   storageClassName: storage-classe-1
12   nfs:
13     server: 192.168.1.100
14     path: "/dados"
```

Figura 2 – Exemplo de criação PV

Fonte: Elaborado pelo autor (2023)

Neste exemplo, um objeto de persistent volume é criado com um tipo de NFS, o servidor NFS e o caminho do volume são especificados, bem como a capacidade de armazenamento do volume e as políticas de acesso.

Avançando para persistent volume claim (PVC), ele é uma maneira de solicitar um espaço de armazenamento persistente em um cluster Kubernetes. Os PVCs solicitam acesso a um volume persistente e especificam as reivindicações de armazenamento necessárias, como o tamanho do volume e o tipo de armazenamento necessário.

O Kubernetes utiliza essas reivindicações para associar um PV existente ou provisionar um novo PV.

Temos alguns dos tipos mais comuns de persistent volume claims:

- ReadWriteOnce: o PVC pode ser montado por um único nó do cluster para leitura e gravação.
- ReadOnlyMany: o PVC pode ser montado por vários nós do cluster para leitura.
- ReadWriteMany: o PVC pode ser montado por vários nós do cluster para leitura e gravação.

Por fim, nós temos o storage class (SC), ele é uma maneira de definir diferentes tipos de armazenamento para atender às necessidades específicas de uma aplicação ou serviço. As classes de armazenamento podem ser usadas para definir políticas de armazenamento, como o tipo de armazenamento, o nível de desempenho e o tamanho do armazenamento.

As storage classes são particularmente úteis em ambientes de container, onde diferentes aplicações e serviços têm necessidades de armazenamento diferentes.

Ao utilizar storage classes, os usuários do cluster podem definir diferentes políticas de armazenamento para cada aplicação ou serviço, o que torna o gerenciamento de armazenamento mais fácil e eficiente.

Temos uma lista de alguns dos tipos mais comuns de storage classes:

- Standard: um tipo de armazenamento padrão que é oferecido pela maioria dos provedores de nuvem. Ele é adequado para uso geral, mas pode não oferecer o melhor desempenho ou a melhor escalabilidade.
- SSD: um tipo de armazenamento que é mais rápido e mais escalável do que o armazenamento padrão. É adequado para aplicações que exigem alto desempenho de E/S, como bancos de dados.
- Storage area network (SAN): um tipo de armazenamento que é oferecido por alguns provedores de nuvem e permite que os usuários do cluster provisionem volumes em um pool compartilhado de armazenamento. É adequado para ambientes de missão crítica que exigem alta disponibilidade e desempenho.

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, você aprendeu sobre volumes no Kubernetes e como eles podem ser usados para persistência de dados em ambientes de containers. Através de exemplos práticos, vimos como os volumes permitem que os dados sejam armazenados fora do container, tornando mais fácil o armazenamento e o gerenciamento de dados de maneira independente do ciclo de vida dos containers.

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

REFERÊNCIAS

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

EMEND

PALAVRAS-CHAVE

Kubernetes, Volumes, VPC.

EMANIP



POSTECH