

THIAGO ADRIANO

POSTECH

SOFTWARE ARCHITECTURE

KUBERNETES

AULA 03

SUMÁRIO

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

EMSE

O QUE VEM POR AÍ?

Nesta aula, os docentes apresentaram o que seria um Pod, como criá-lo e como dar os seus primeiros passos utilizando os rótulos e anotações de um Pod.

EMSE

HANDS ON

Na primeira videoaula, os docentes demonstraram como criar um Pod por meio de alguns exemplos práticos.

Já na videoaula seguinte, nossos professores demonstraram como utilizar os rótulos e as anotações nos Pods.



SAIBA MAIS

PODS, RÓTULOS E ANOTAÇÕES

Nesta aula, aprenderemos sobre os Pods, rótulos e anotações, que são conceitos fundamentais no ecossistema do Kubernetes, que é uma plataforma de orquestração, conforme aprendemos na aula anterior. Apresentaremos um pouco mais da parte teórica sobre os Pods e avançaremos para a parte prática, criando os nossos primeiros Pods de maneira interativa.

Mas antes de falarmos sobre o que são os Pods, vamos nivelar o nosso conhecimento sobre um cluster no Kubernetes.

Um cluster no Kubernetes é um grupo de nodes que executam os Pods e outros recursos no ambiente do Kubernetes. Cada node, em um cluster, pode hospedar vários Pods, e esses Pods podem se comunicar com outros no mesmo cluster usando seus endereços IP internos.

Na figura 1 – “Exemplo de Cluster com nodes”, temos ilustrado como seria esta divisão:

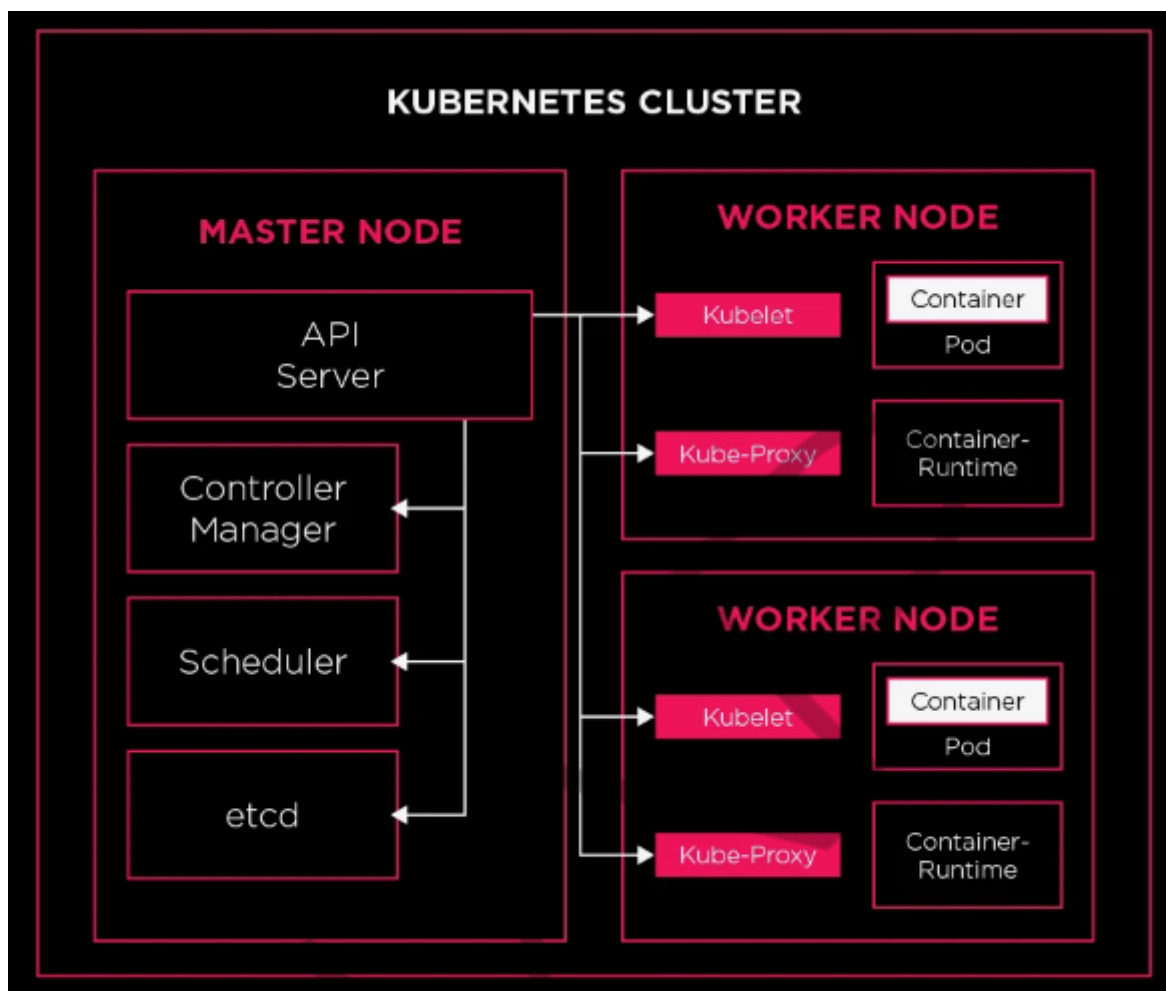


Figura 1 – Exemplo de Cluster com nodes
 Fonte: <https://5i4i.short.gy/EbMR1R> (2021), adaptado por FIAP (2023)

Na figura apresentada, nós temos um cluster com 3 nodes, um para o Master e outros dois que trabalham como Workers, hospedando os nossos Pods.

Os clusters no Kubernetes são escaláveis, o que significa que eles podem aumentar ou diminuir o número de nodes e Pods em resposta às mudanças na carga de trabalho. Isso permite que o Kubernetes gerencie, de forma inteligente, a distribuição de carga de trabalho em um cluster, garantindo que cada Pod tenha recursos suficientes para executar seu trabalho.

Como os clusters no Kubernetes funcionam?

Os clusters no Kubernetes são compostos por vários componentes, cada um desempenhando um papel importante no gerenciamento dos recursos no cluster.

Alguns dos principais componentes de um cluster do Kubernetes incluem:

- Master node: o Master node é o componente central do Kubernetes, responsável pelo gerenciamento do cluster e pela tomada de decisões sobre onde e como executar os Pods.
- Worker node: o Worker node é um node que executa os Pods e outros recursos no ambiente do Kubernetes.
- etcd: etcd é um banco de dados distribuído que armazena a configuração do Kubernetes e o estado do cluster.
- kubelet: o kubelet é um agente que executa em cada node do cluster e gerencia os Pods que são executados nesse node.
- kube-proxy: o kube-proxy é responsável por encaminhar o tráfego de rede para os Pods no cluster.

Os clusters, no Kubernetes, são altamente configuráveis e personalizáveis, permitindo que os desenvolvedores ajustem o cluster para atender às necessidades específicas de seus aplicativos.

Eles são importantes porque permitem que os desenvolvedores criem e gerenciem aplicativos escaláveis e resilientes. Ao dividir os aplicativos em unidades menores, como os Pods, e distribuí-los em vários nodes, os desenvolvedores podem se concentrar em escrever código para uma única responsabilidade, mantendo a coesão e a clareza do código.

Caso tenha ficado com curiosidade sobre como escrever um código coeso e com clareza, temos uma disciplina, na fase 1, sobre Arquitetura de Software, onde abordamos a importância deste assunto.

Voltando aos clusters no Kubernetes, eles também são importantes porque permitem que os aplicativos sejam facilmente escalonados. Como o Kubernetes gerencia a distribuição de carga de trabalho entre os nodes, ele pode criar ou destruir nodes e Pods conforme necessário para lidar com a carga de trabalho, eliminando a necessidade de gerenciar manualmente a infraestrutura.

Bom, agora que conhecemos a casa dos Pods (local onde eles são publicados), o que seria um Pod?

Um Pod é uma abstração do Kubernetes que representa um único processo em execução em um cluster. Ele pode conter um ou mais containers, compartilhando

o mesmo espaço de rede e armazenamento. Por exemplo, um Pod pode conter um container web e um banco de dados, ambos se comunicando por meio da mesma interface de rede.

Os Pods são efêmeros, o que significa que eles são criados e destruídos conforme necessário. Isso permite que o Kubernetes gerencie de forma inteligente a distribuição de carga de trabalho em um cluster, garantindo que cada Pod tenha recursos suficientes para executar seu trabalho.

Como os Pods funcionam?

Os Pods são criados a partir de definições YAML ou JSON, que especificam quais containers devem ser executados dentro do Pod, suas imagens e as portas em que cada container deve ouvir. Quando ele é criado, recebe um endereço IP interno dentro do cluster, permitindo que outros Pods se comuniquem com ele.

Na figura 2 – “Exemplo de criação de um Pod utilizando o arquivo YAML”, temos uma imagem demonstrando um arquivo YAML para criação de um Pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: meu-pod
spec:
  containers:
  - name: meu-container
    image: nginx
    ports:
    - containerPort: 80
```

Figura 2 – Exemplo de criação de um POD utilizando o arquivo YAML
Fonte: Elaborado pelo autor 2023

Os Pods também podem ser escalonados horizontalmente, o que significa que vários Pods idênticos podem ser criados para lidar com picos de tráfego ou aumentar a capacidade de processamento.

O Kubernetes gerencia a distribuição de carga de trabalho entre esses Pods de forma transparente para o usuário, garantindo que a carga seja distribuída de forma equilibrada.

Por que os Pods são importantes?

Os Pods são importantes, no Kubernetes, porque permitem que os desenvolvedores criem aplicativos escaláveis e resilientes. Ao dividir os aplicativos em unidades menores, como os Pods, os desenvolvedores podem se concentrar em escrever código para uma única responsabilidade. Além disso, os Pods permitem que os aplicativos sejam facilmente escalonados, permitindo que eles lidem com variações de tráfego sem comprometer o desempenho.

Agora, para que possamos criar um Pod no Kubernetes, nós precisamos entender o que são os rótulos e anotações na definição de um Pod.

Os rótulos e as anotações são recursos importantes no Kubernetes, para gerenciar e organizar os objetos, incluindo os Pods.

Os rótulos são pares de chave-valor que são atribuídos aos objetos no Kubernetes, como Pods, Serviços, Application Controllers, entre outros. Eles são usados para identificar, categorizar e selecionar objetos para operações de gerenciamento, como escalonamento e exclusão. Por exemplo, você pode adicionar um rótulo "app: myapp" ao seu Pod para indicar que ele faz parte da aplicação "myapp". Você pode usar esses rótulos para selecionar e executar operações em todos os Pods que fazem parte dessa aplicação.

As anotações, por outro lado, são pares de chave-valor opcionais que podem ser adicionadas a objetos no Kubernetes. Eles são usados para adicionar metadados adicionais aos objetos, como informações de depuração, documentação e políticas de segurança. Por exemplo, você pode adicionar uma anotação "author: Thiago S Adriano" ao seu Pod para indicar que ele foi criado por Thiago S Adriano. Essas anotações não são usadas para seleção de objetos, mas podem ser úteis para fins de documentação e auditoria.

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, você aprendeu os principais conceitos relacionados aos Pods, como eles funcionam e a sua importância no mundo dos Kubernetes.

EMANIP

REFERÊNCIAS

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

EMEND

PALAVRAS-CHAVE

Anotações e Rótulos. Kubernetes. Pods.

EMSE



POSTECH