Treinamento
Kubernetes

Francisco Rafael - francisco.rafaell.pereira@gmail.com - CPF: 068.778.653-32

Bruno Masuda
Instrutor

Formado em MBA Gerenciamento e Projetos de Data Centers pela Faculdade Impacta Tecnologia.
Admirador e praticante da cultura DevOps/SRE com quatro anos de experiência na área.
Especializado em Kubernetes e sustentação de micro serviços.

LUIS DEV

Ementa do treinamento

O curso faz uma introdução ao mundo Kubernetes com os principais tópicos de como preparar e implantar uma aplicação dentro de um ambiente em execução e quais recursos utilizar em diferentes tipos de aplicações.

- Conceitos básicos Kubernetes
- Comandos básicos (kubectl)
- Manifestos e recursos
- Atributos e configurações
- Services e Ingress
- Diagnosticando problemas básicos

LUIS DEV

Introdução ao Kubernetes

LUIS DEV

Francisco Rafael - francisco.rafaell.pereira@gmail.com - CPF: 068.778.653-32

O que é Kubernetes?

Kubernetes é um sistema de código aberto (open-source) para automação de deployments com escalabilidade e gerenciamento de aplicações em container.

Atualmente faz parte do catálogo do Cloud Native Computing Foundation (CNCF), iniciativa responsável por incubar diversos projetos open-source com características *cloud native*, assim como o Kubernetes.

Composição de um cluster Kubernetes

O Kubernetes é um conjunto de servidores que são responsáveis por gerenciar containers em execução. Esse conjunto é chamado de "cluster".

Um cluster é composto basicamente por:

- Master nodes: Máquinas que gerenciam o cluster (API Server, Scheduler, Controller Manager)
- Worker nodes: Máquinas que executam aplicações em containers
- ETCD: Banco de dados do cluster.

Interagindo com o cluster

LUIS DEV

kubectl

Ao contrário de um arquivo YAML (linguagem declarativa), o kubectl é considerado uma ferramenta de linha de comando que interage com o cluster para ler, criar, editar e excluir objetos (comando imperativo).

Também com o kubectl podemos aplicar manifestos YAML para a criação, edição ou exclusão de objetos.

Manifestos YAML

Uma das formas de interação com um cluster (não administrativa) é usando arquivos com sintaxe YAML.

Os arquivos possuem as características e atributos que definem como um objeto deve ser criado pelo Kubernetes (linguagem declarativa).

Tipos de deployment

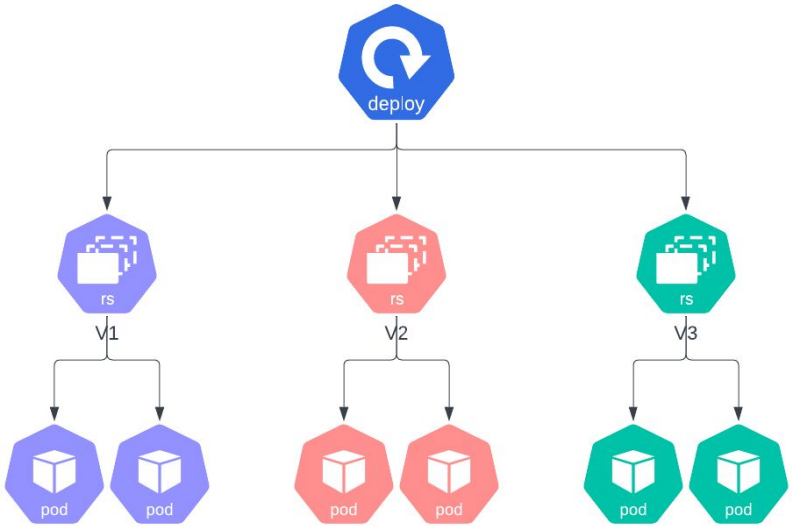
LUIS DEV

Como funciona um deployment

Um **objeto** do tipo "Deployment" provê definições e características para a criação de um "ReplicaSet" e consequentemente um "Pod"

Por sua vez, um "ReplicaSet" é responsável por manter as réplicas de um Pod em execução.

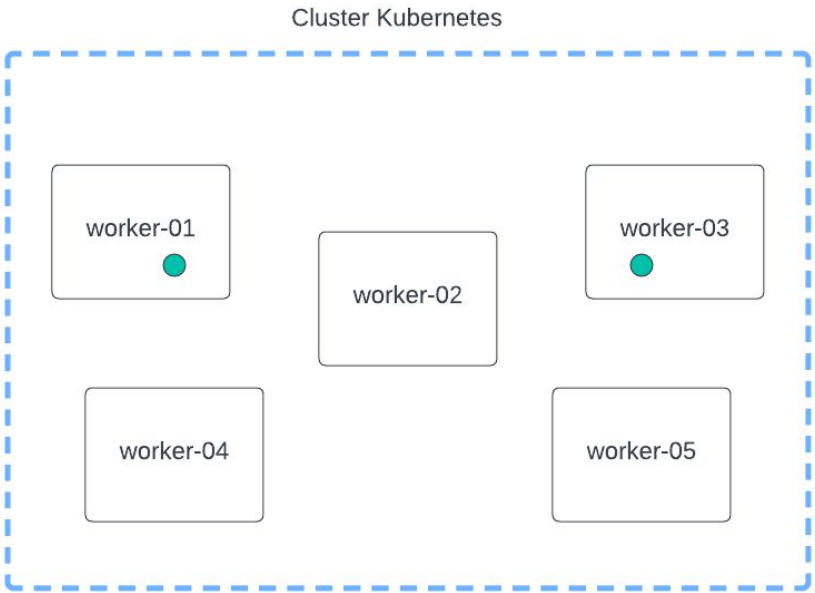
O Pod é o recurso mais baixo dessa hierarquia.



LUIS DEV

Deployment

Objetos do tipo "Deployment" criam "ReplicaSets" que por sua vez é responsável por manter réplicas de "Pods" dentro do cluster.
Os "Pods" são distribuídos entre os nós do cluster dependendo dos recursos livres de cada nó.

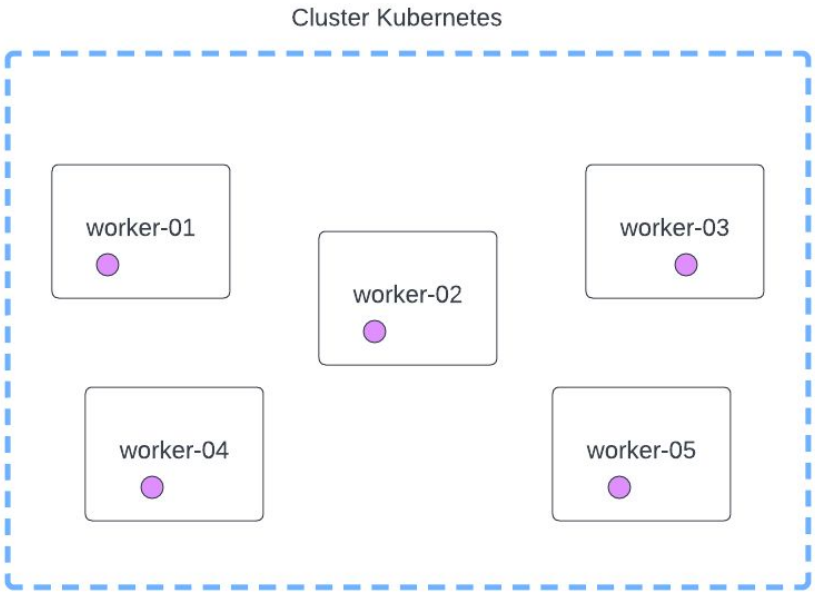


LUIS DEV

Francisco Rafael - francisco.rafaell.pereira@gmail.com - CPF: 068.778.653-32

Daemonset

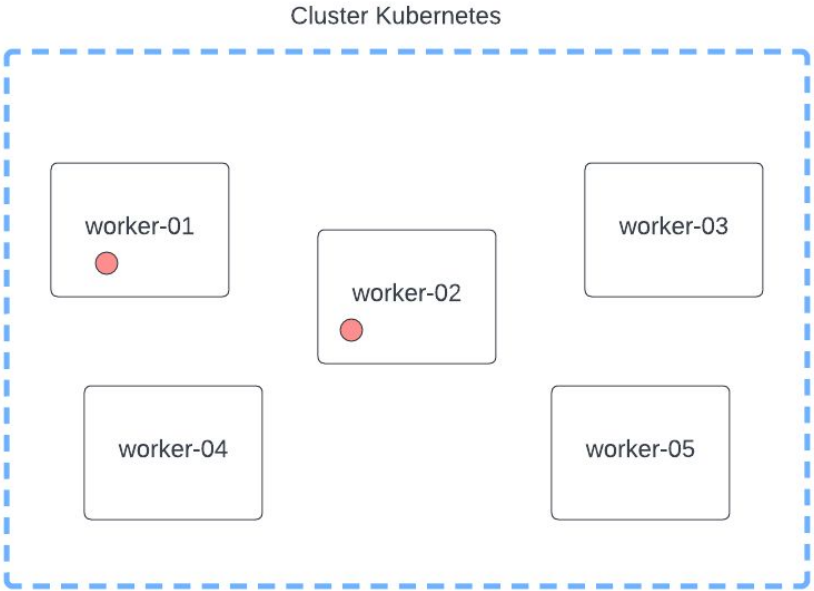
"Daemonset" é um tipo de deployment que cria uma réplica do seu "Pod" em cada um dos nós do cluster.
Essa estratégia é comum em ferramentas operacionais de infraestrutura, como coleta de métricas para monitoramento ou logs.



LUIS DEV

Statefulset

"Statefulset" normalmente é utilizado para criar aplicações que precisam manter estado (dados/arquivos), principalmente para o cenário de banco de dados, ou clusters de bancos de dados.



LUIS DEV

Configmaps e Secrets

LUIS DEV

Configmaps

Um "ConfigMap" é um objeto utilizado para armazenar dados que não contenham informações sensíveis (ex: senha de um banco de dados). Isso permite o desacoplamento de configurações específicas dos containers, deixando as aplicações mais portáteis entre ambientes.
ConfigMaps podem ser consumidos por Pods durante a sua execução de de formas:

- Como variáveis de ambiente
- Como arquivos montados dentro do sistema de arquivos do Pod

Secrets

Em casos onde há a necessidade de consumir dados sensíveis, a recomendação é utilizar um objeto Secrets com os devidos dados. Assim como ConfigMaps, Secrets também podem ser passadas como variáveis de ambientes, montados em diretórios ou como arquivos e até mesmo ser consumido por outros objetos dentro do cluster.

Um outro caso de uso de Secret é a criação de certificados TLS que são utilizados para conexões HTTPS válidas.

Principais configurações de um Deployment

LUIS DEV

Requests e Limits

Os campos "Requests" e "Limits" dentro de um manifesto de Deployment ditam a quantidade mínima e máxima de recursos computacionais (cpu e memória) que um container pode consumir dentro de um Pod.

- Requests: A quantidade de recursos requisitada para a criação de um Pod
- Limits: O limite de consumo permitida para o container

Caso não haja recursos mínimos para a criação de um container, o Pod não é inicializado no ambiente.
Caso o limite de recursos seja ultrapassado, essa aplicação pode ser reiniciada pelo controlador de forma inesperada, causando interrupções na aplicação.

Liveness e Readiness

Os campos "Liveness" e "Readiness" são responsáveis pela checagem da saúde do container que está dentro do Pod. Essa checagem pode ser feita através de um comando script, uma checagem de porta TCP ou a requisição num *endpoint* da aplicação que devolva um *status code* entre 200 e 399 (qualquer outro *status code* será considerado como falha).

- livenessProbe: Responsável por fazer a checagem de tempos em tempos para garantir que a aplicação está responsiva. Usado durante a execução da aplicação.
- readinessProbe: Responsável por fazer a checagem que garante que a aplicação esteja pronta para receber requisições. Usado até o fim da inicialização da aplicação.

Service e ingress

LUIS DEV

Service

O objeto service é utilizado para distribuir chamadas entre réplicas de um mesmo deployment.
Por exemplo uma aplicação que responde um front-end com três réplicas, as chamadas devem ser direcionadas ao *service* daquele front-end, e o mesmo é responsável por repassar as chamadas para um dos três pods, distribuindo de forma aleatória entre os eles.
Existem três tipos de service:

- NodePort
- ClusterIP
- LoadBalancer

LUIS DEV

Ingress

O Ingress atua como uma porta de entrada do cluster para aplicações http/https expondo apenas um único ponto de conectividade externa e resolvendo encaminhamentos de requisições entre *services* através do cabeçalho http.

LUIS DEV