

THIAGO ADRIANO

POSTECH

SOFTWARE ARCHITECTURE  
KUBERNETES

# AULA 02

---

## SUMÁRIO

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

## O QUE VEM POR AÍ?

Nesta aula, aprenderemos sobre a API do Kubernetes, como acessá-la utilizando um projeto que desenvolvemos em aula e o Kubectl.

EMAND

## HANDS ON

Nesta primeira videoaula, veremos os comandos básicos do Kubectl por meio de alguns exemplos práticos.

Na aula seguinte, aprenderemos como acessar a API RESFful do Kubernetes utilizando o C#.

EMANIP

## SAIBA MAIS

### CONHECENDO O KUBECTL

O Kubectl é um CLI (comando de linha de controle) que permite interagir com um cluster do Kubernetes. É a ferramenta principal usada pelos administradores de sistemas, desenvolvedores e outros usuários para gerenciar, implantar e monitorar aplicativos em um cluster do Kubernetes.

Ele é uma plataforma de código aberto para orquestração de containers, projetada para facilitar a implantação, o dimensionamento e o gerenciamento de aplicativos em containers. Com o Kubernetes, você pode implantar e gerenciar aplicativos em qualquer ambiente - local, nuvem pública ou privada - de forma consistente e escalável.

O Kubectl fornece uma interface de linha de comando intuitiva para gerenciar objetos Kubernetes, como pods, serviços, implantações, ingressos e muito mais. Com ele, você pode implantar aplicativos em um cluster, atualizar aplicativos existentes, visualizar o status de um aplicativo e solucionar problemas de um aplicativo em execução. Mas como ele funciona?

Ele é executado em uma máquina local e se conecta a um cluster do Kubernetes por meio de uma API do Kubernetes, por meio de solicitações RESTful para criar, atualizar ou excluir objetos Kubernetes.

O Kubectl usa um arquivo de configuração do Kubernetes chamado kubeconfig para determinar como se conectar ao cluster do Kubernetes. O kubeconfig contém informações como o endereço do servidor API do Kubernetes, as credenciais de autenticação e o contexto do cluster atual.

O Kubectl tem muitos recursos úteis para gerenciar objetos Kubernetes, como:

- Implantação de aplicativos: com kubectl, você pode implantar aplicativos em um cluster Kubernetes. Isso envolve criar pods e outros objetos relacionados, como serviços, ingressos e volumes persistentes.

- Atualização de aplicativos: você pode usar o kubectl para atualizar aplicativos existentes, alterar a imagem de container ou atualizar outras configurações.
- Gerenciamento de objetos Kubernetes: com Kubectl, você pode criar, excluir e modificar objetos Kubernetes, como pods, serviços, implantações, ingressos e muito mais.
- Visualização de status do aplicativo: o Kubectl pode mostrar o status de um aplicativo em execução, incluindo informações sobre os pods em execução, os serviços em uso e muito mais.
- Solução de problemas de aplicativos: o Kubectl permite depurar aplicativos em execução em um cluster, exibindo logs de container, conectando-se a um container em execução e muito mais.

No Kubectl temos algumas opções de comandos e é sempre difícil lembrar de todas as opções que temos. Ao executar o comando “**kucetl**”, assim mesmo, sem nenhum parâmetro, veremos todas as opções de comandos categorizados em: básicos, intermediários, comandos de implantação e de gerenciamento.

```
D:\FIAP_POS\Kubernetes>kubectl
kubectl controls the Kubernetes cluster manager.

Find more information at: https://kubernetes.io/docs/reference/kubectl/

Basic Commands (Beginner):
  create      Create a resource from a file or from stdin
  expose      Take a replication controller, service, deployment or pod and expose it as a new Kubernetes servi
  run         Run a particular image on the cluster
  set         Set specific features on objects

Basic Commands (Intermediate):
  explain     Get documentation for a resource
  get         Display one or many resources
  edit        Edit a resource on the server
  delete      Delete resources by file names, stdin, resources and names, or by resources and label selector

Deploy Commands:
  rollout     Manage the rollout of a resource
  scale       Set a new size for a deployment, replica set, or replication controller
  autoscale   Auto-scale a deployment, replica set, stateful set, or replication controller

Cluster Management Commands:
  certificate  Modify certificate resources.
  cluster-info Display cluster information
  top          Display resource (CPU/memory) usage
  cordon       Mark node as unschedulable
  uncordon     Mark node as schedulable
  drain        Drain node in preparation for maintenance
  taint        Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe    Show details of a specific resource or group of resources
  logs        Print the logs for a container in a pod
```

Figura 1 – Executando o comando “kubectl”

Fonte: Elaborado pelo autor (2023)

Os principais comandos Kubectl estão associados aos comandos get, delete e create. O get para obtermos informações a despeito da saúde e das configurações do nosso cluster, create para criação e delete para remoção de recursos no nosso K8's.

Por exemplo, vamos executar o comando:

```
"kubectl run nginx --image=nginx:1.14.2 --port:80"
```

O que ele faz? Cria um pod com nome "nginx" (e aqui poderemos dar o nome que quisermos) a partir da imagem "nginx:1.14.2" (que está no repositório central).

Após este comando, vamos verificar se de fato o pod foi criado e qual é o seu status. Use o comando "`kubectl get pods`".

Vamos apagar este pod para não "poluirmos" o nosso ambiente. Execute o comando "`kubectl delete pod nginx`" (onde nginx é o nome do pod. Lembre-se: ele pode ter o nome que quisermos).

Após a execução bem-sucedida do comando "`kubectl delete pod nginx`", o pod será removido e não será mais possível acessá-lo ou usá-lo no cluster. É importante lembrar que a remoção de um pod pode afetar a disponibilidade de um serviço ou aplicação em execução no cluster. Portanto, é importante tomar cuidado ao executar o comando de exclusão de um pod e verificar se essa ação não afetará outras partes do sistema.

Verifique a figura 2 – "Kubectl para entender como funciona o Kubectl".

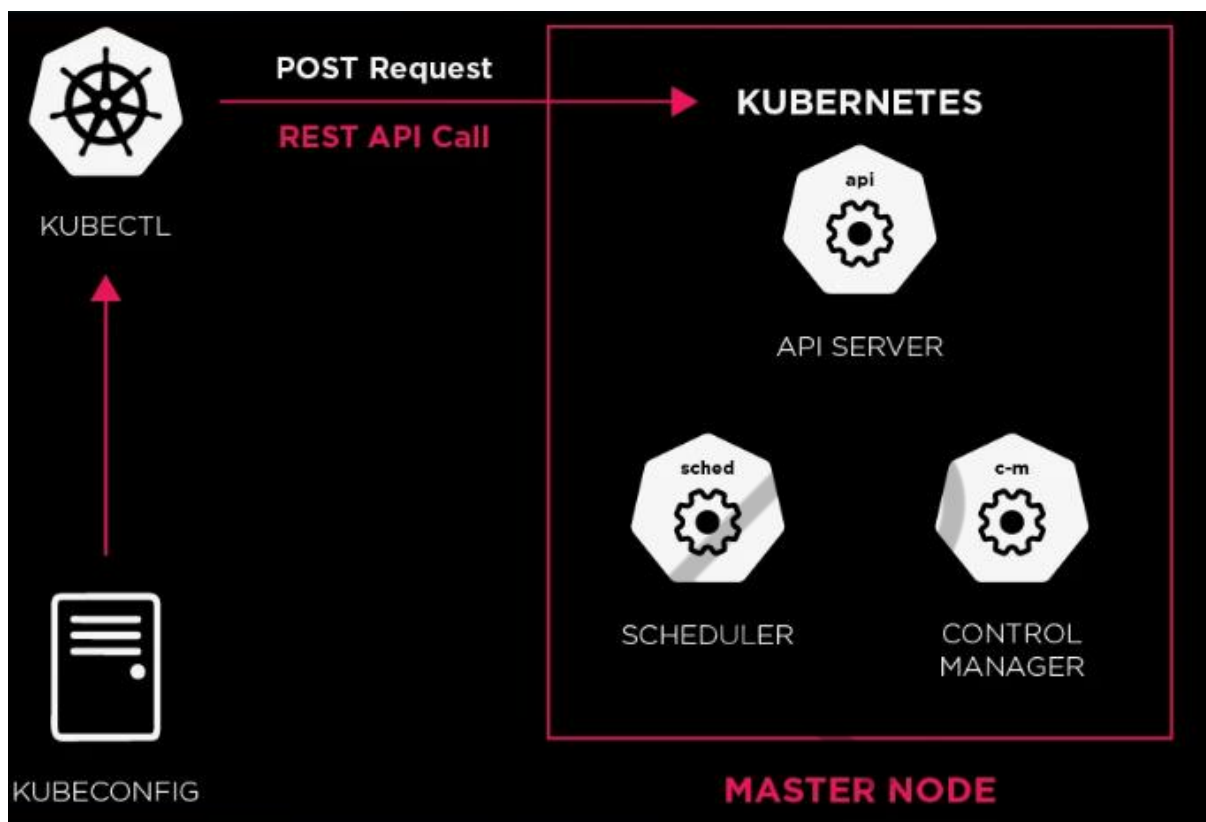


Figura 1 – Kubectl  
 Fonte: DZone (2021), adaptado por FIAP (2023)

## API do Kubernetes

A API do Kubernetes é a principal interface para interagir com um cluster Kubernetes. É por meio da API que as ferramentas e aplicativos se comunicam com o Kubernetes para gerenciar os recursos do cluster, como pods, serviços, deployments e outros objetos.

Esta API é projetada para ser extensível e flexível, permitindo que os desenvolvedores criem e adicionem novos recursos e funcionalidades. Além disso, a API é baseada na arquitetura RESTful, o que significa que pode ser acessada usando os métodos HTTP padrão, como GET, POST, PUT e DELETE.

A API do Kubernetes é dividida em vários grupos, cada um contendo recursos relacionados. Por exemplo, o grupo "core" contém recursos fundamentais como pods, serviços e replicaset, enquanto outros grupos contêm recursos para monitoramento, segurança, armazenamento e muito mais.



Para acessar a API do Kubernetes, nós podemos utilizar o Kubectl ou criar os nossos próprios clientes personalizados usando bibliotecas de linguagem de programação como Python, Go, Java, Ruby e outras. O Kubernetes também oferece uma API de serviço que permite que aplicativos rodando dentro do cluster acessem a API.

Ao criar aplicativos para o Kubernetes, é importante entender como a API funciona e como usá-la para gerenciar os recursos do cluster. A documentação da API do Kubernetes é abrangente e fornece informações detalhadas sobre cada recurso e como usá-lo. Com a API do Kubernetes, os desenvolvedores podem criar aplicativos escaláveis, confiáveis e fáceis de gerenciar em um ambiente de cluster.

## O QUE VOCÊ VIU NESTA AULA?

Nesta aula, aprendemos a como utilizar o comando de linha de controle Kubectl, que permite interagir com os nossos clusters Kubernetes, e quais as principais funções que ele nos disponibiliza.

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

EMANUELO

## REFERÊNCIAS

BASS, L.; CLEMENTS, P.; KAZMAN, R. Software Architecture in Practice. 3<sup>rd</sup> ed. [s.l.]: Addison-Wesley Professional, 2012.

EMAP

## **PALAVRAS-CHAVE**

Arquitetura de software. Kubernetes. Kubeclt.

EMENDAS



POSTECH