



# Dos contêineres à orquestração do mundo



Prof. Ramon Fontes  
IMD/UFRN

Entusiasta de Software Livre

Contribuidor de Materiais e Códigos Livres

Contribuidor do Kernel Linux

Membro do Leading Advanced Technology  
Center of Excellence - *LANCE*

# Agenda

- Contextualização
- Containers
- Orquestração
- Demonstração
- Segurança

# Contextualização



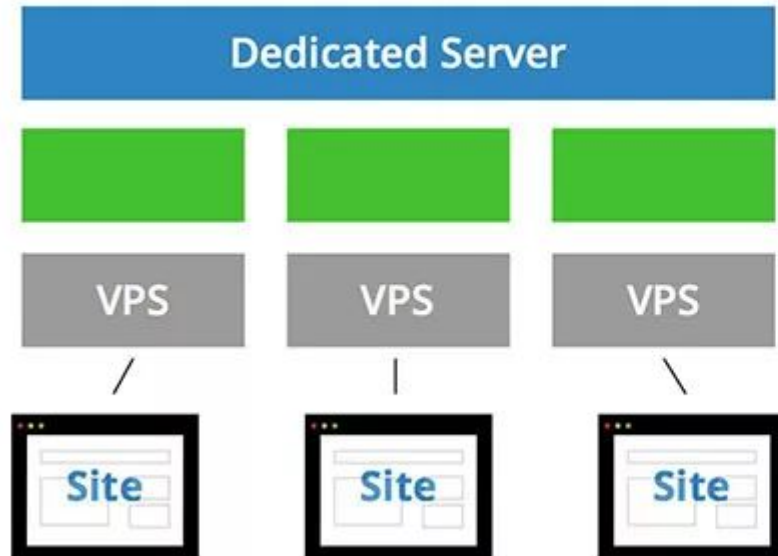
# Virtual Private Server



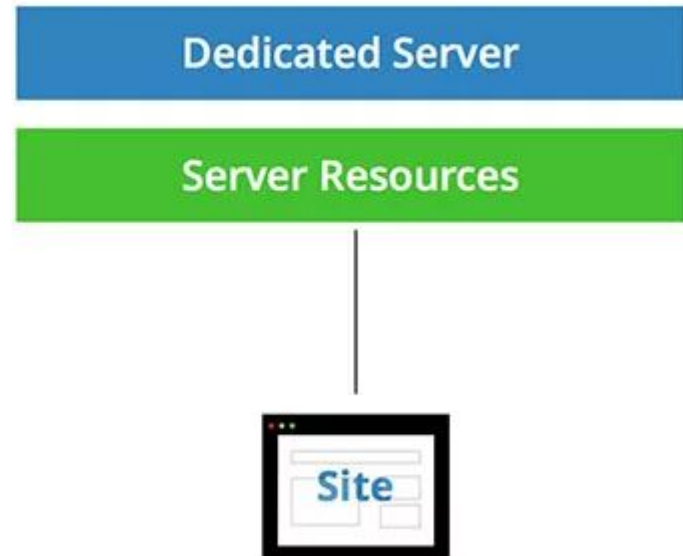
**VPS** - máquina virtual que fornece recursos de servidor virtualizados em um servidor físico compartilhado com outros

# Servidores Dedicados

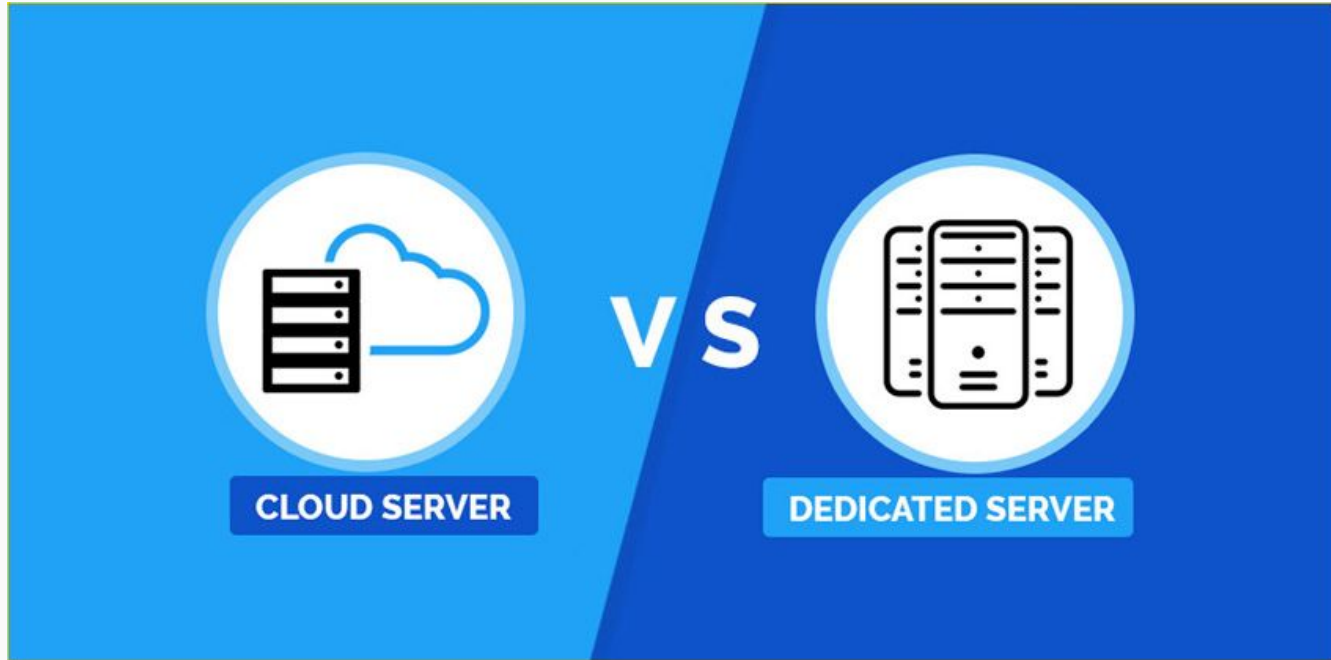
VPS Hosting



Dedicated Server Hosting



# Servidores de Cloud



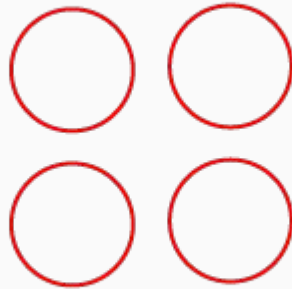
# Modelo

## Monolithic Vs SOA Vs Microservices



**Monolithic**

Single Unit



**SOA**

Coarse-grained



**Microservices**

Fine-grained

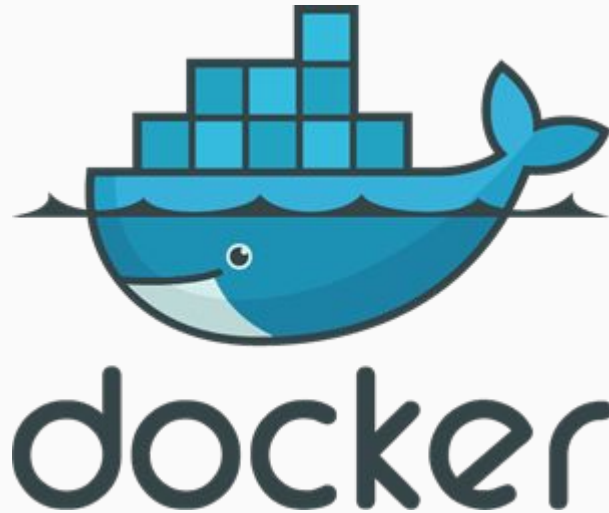


# Containers



# Container – O que há em um nome?

Vindo da indústria naval



# Contêineres de transporte

Portabilidade **pode ser usado** em qualquer um dos tipos de navios suportados

Grande variedade de carga **pode ser embalada**

Tamanhos padrão **acessórios padrão** em navios

Muitos Containers **em um navio**

Isola **a carga** uma da outra

# Traduzido para software

Portabilidade **pode ser usado em qualquer sistema suportado (sistema com ambiente de execução de contêiner)**

Grande variedade de software **pode ser embalado**

Formato Padrão

Muitos containers **para um nó físico**

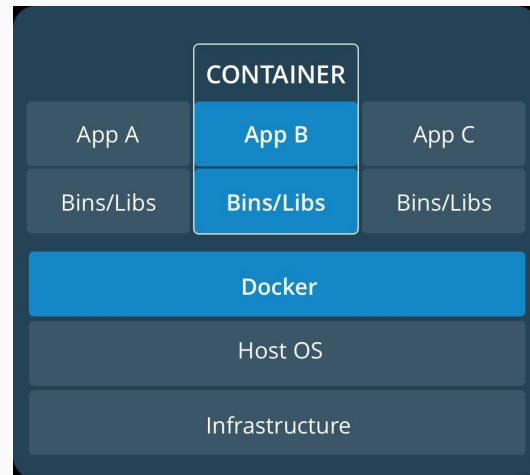
Isola **a execução de um contêiner de outro**

# O que é um contêiner?

Alternativa de empacotar código e dependências juntos

Pode rodar em qualquer lugar

Executa vários contêineres em uma máquina física



# Soa familiar?

Mesmo conceito das **máquinas virtuais**

Agrupa sistema operacional e software para execução em **instâncias isoladas**

Pode ser **executado em qualquer lugar** em que o hipervisor específico seja executado

Múltiplas VMs para uma máquina física

# Como funcionam as VMs?

Hipervisor = camada entre VM e Kernel

Emula chamadas do sistema






Permite vários tipos de sistemas operacionais em uma máquina (*Windows no Linux*)



Sobrecarga para hipervisor

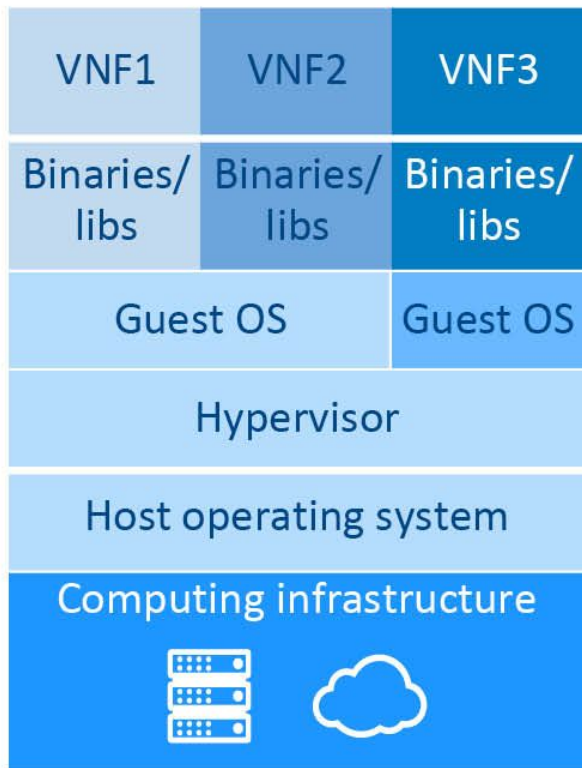
# Os contêineres, por outro lado...

-  **Contém apenas bibliotecas e estruturas** relacionadas a aplicativos que são executadas no kernel da máquina host
-  **Menor**
-  **Menor sobrecarga**

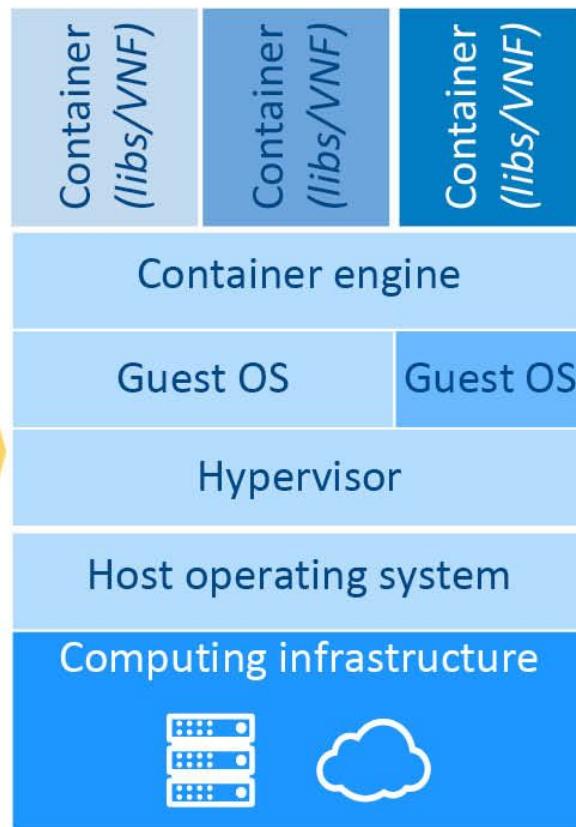
Diferenças nas distribuições e dependências do sistema operacional são abstraídas - mesmo kernel



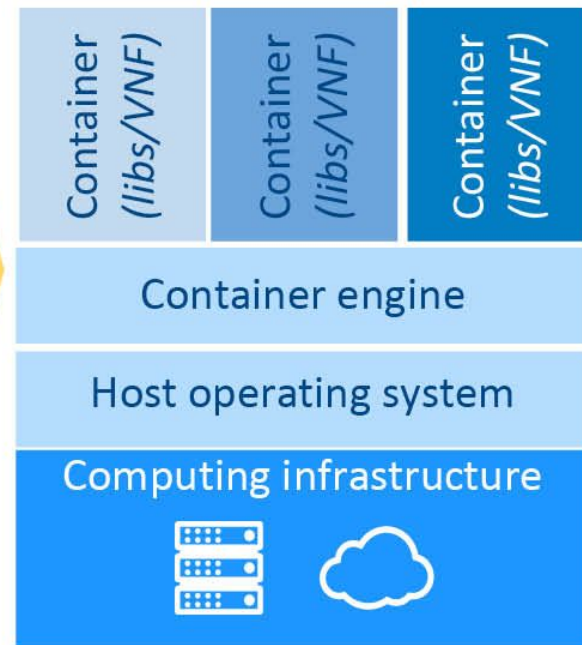
## Virtual machines



## Container on VM



## Container on bare metal



# Trabalhando juntos, não uns contra os outros

**Windows no Linux** possível apenas com VMs

**Software mais antigo** precisa ser adaptado para ser executado como contêineres

Uso de VMs como meio para contêineres (melhor isolamento e escalonamento mais fácil)

# Contêineres capacitando microsserviços

**Tempos de início mais rápidos** -> fácil de prototipar ou dimensionar

**Permite que o trabalho seja feito de forma independente em módulos** -> lançamentos independentes de componentes (cuidar das interfaces)

Ambientes de tempo de execução **isolados e abstraídos**, que podem ser **adaptados para cada módulo**

Ambiente de tempo de execução **compartilhado**, para **aplicações heterogêneas**

# Precisamos de algo mais?

Docker começou com uma ferramenta CLI em cima do lxc, que construía, criava, iniciava, parava e executava contêineres

Faz o gerenciamento em nível de nó, mediante solicitações específicas

Fácil de gerenciar manualmente com até centenas de contêineres e dezenas de nós,  
**mas e depois?**

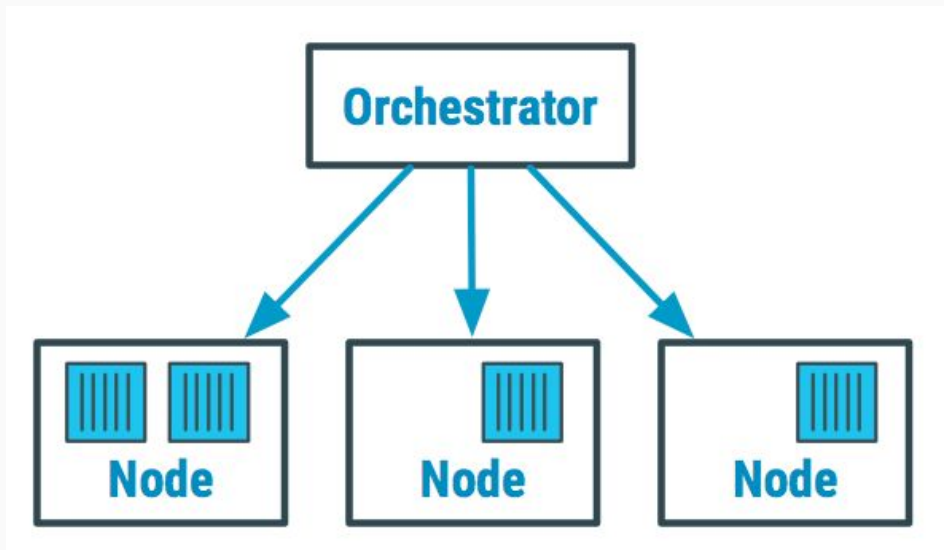
# Orquestração



# Orquestrador

Gerenciar e organizar hosts e contêineres em execução em um cluster

**Questão principal** - alocação de recursos - onde um contêiner pode ser alocado para atender aos seus requisitos (CPU/RAM/disco) + como acompanhar os nós e a escala



# Algumas tarefas do orquestrador

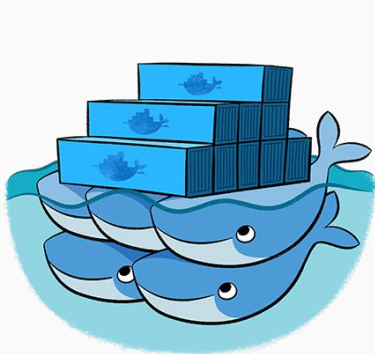
- \* Gerenciar rede e acesso
- \* Rastrear o estado dos contêineres
- \* Serviços de escala
- \* Fazer balanceamento de carga
- \* Realocação em caso de host que não responde
- \* Descoberta de serviço
- \* Atribuir armazenamento a contêineres
- ...

# Opções de Orquestradores

**Docker Swarm** – integrado na plataforma docker container

**Apache Mesos** – ferramenta de gerenciamento de cluster, sendo a orquestração de contêineres apenas uma das coisas que pode fazer, originalmente por meio de um plugin chamado Marathon

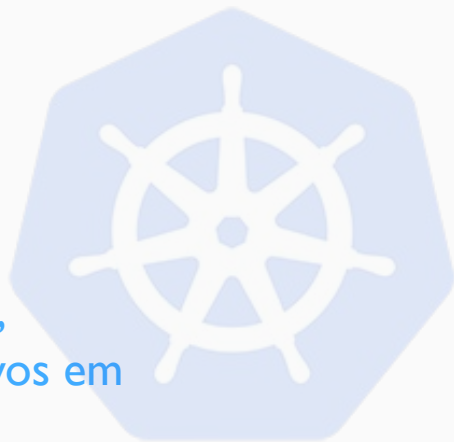
**Kubernetes** – código aberto, produto da CNCF - Cloud Native Computing Foundation





# O que é Kubernetes?

- “Kubernetes” = grego para governador, capitão
- Também pode ser chamado de **k8s** porque é uma letra **K** com oito letras no meio e depois **S**
- Sistema de orquestração de contêineres de código aberto
- Originalmente projetado pelo Google, mantido pela CNCF
- Visa fornecer "plataforma para **automatizar implantação, escalonamento e operações** de **contêineres** de aplicativos em **clusters** de hosts"



“Contêineres são pacotes de software que contém todos os elementos necessários para rodar em qualquer ambiente”,

Google

“Kubernetes [...] é um sistema de código aberto para **automatizar** a implantação, **escalonamento** e **gerenciamento** de aplicativos em contêineres”,

autores do Kubernetes

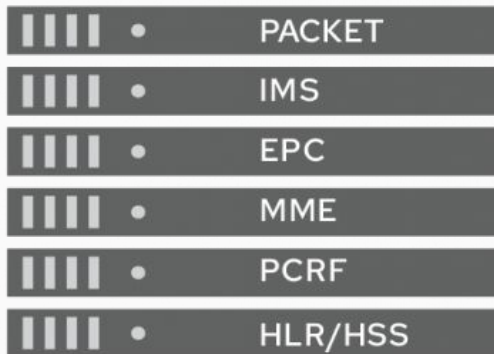
# Evolução das VNFs

Figure: The journey of network functions in telecommunications

## Traditional

### Classic network appliance approach

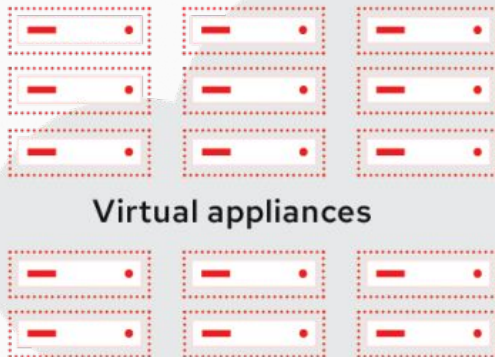
Powered by proprietary hardware and software



## Virtual

### Virtual network functions (VNFs)

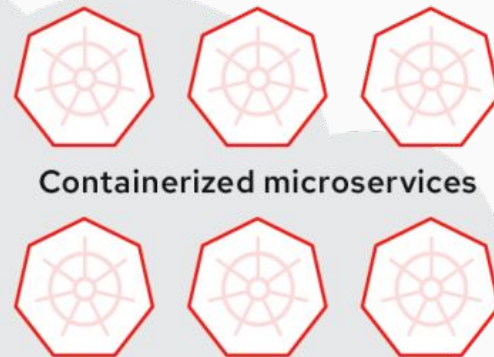
Powered by function application software



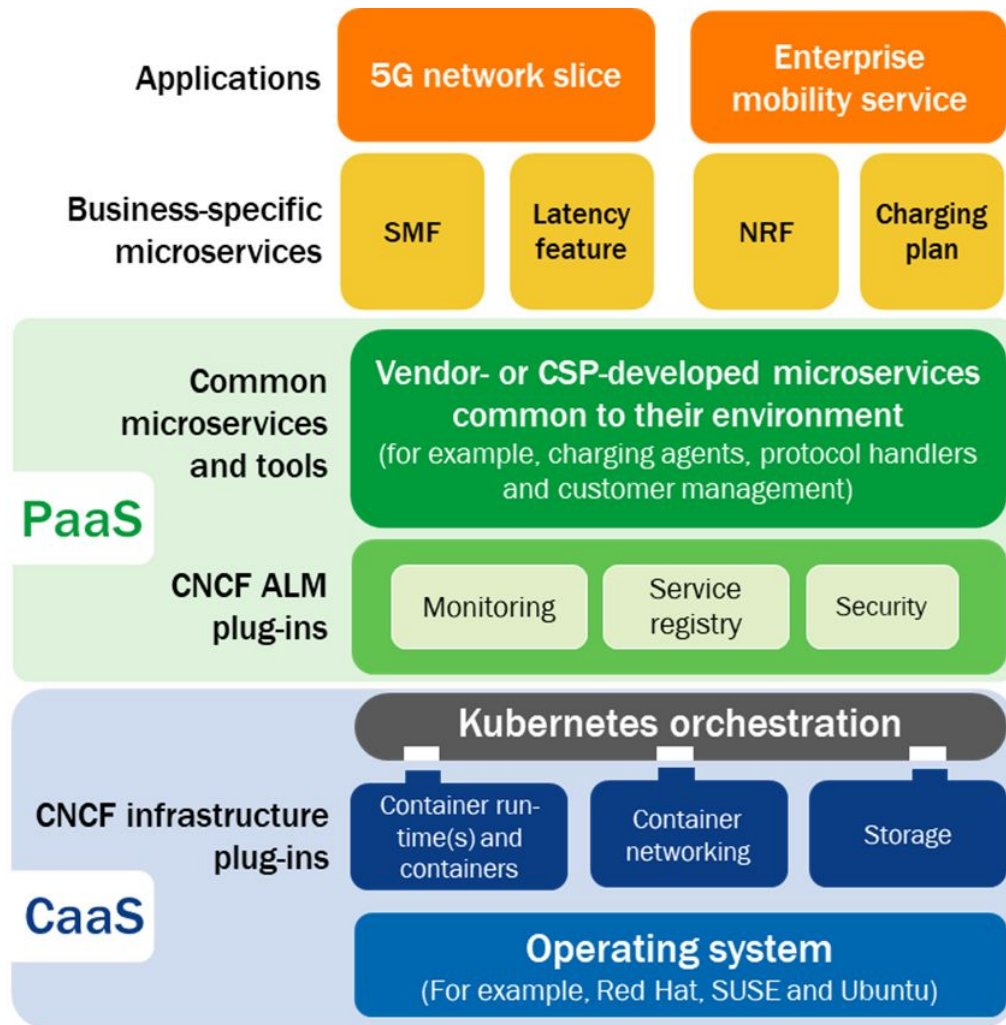
## Cloud-native

### Container network functions (CNFs)

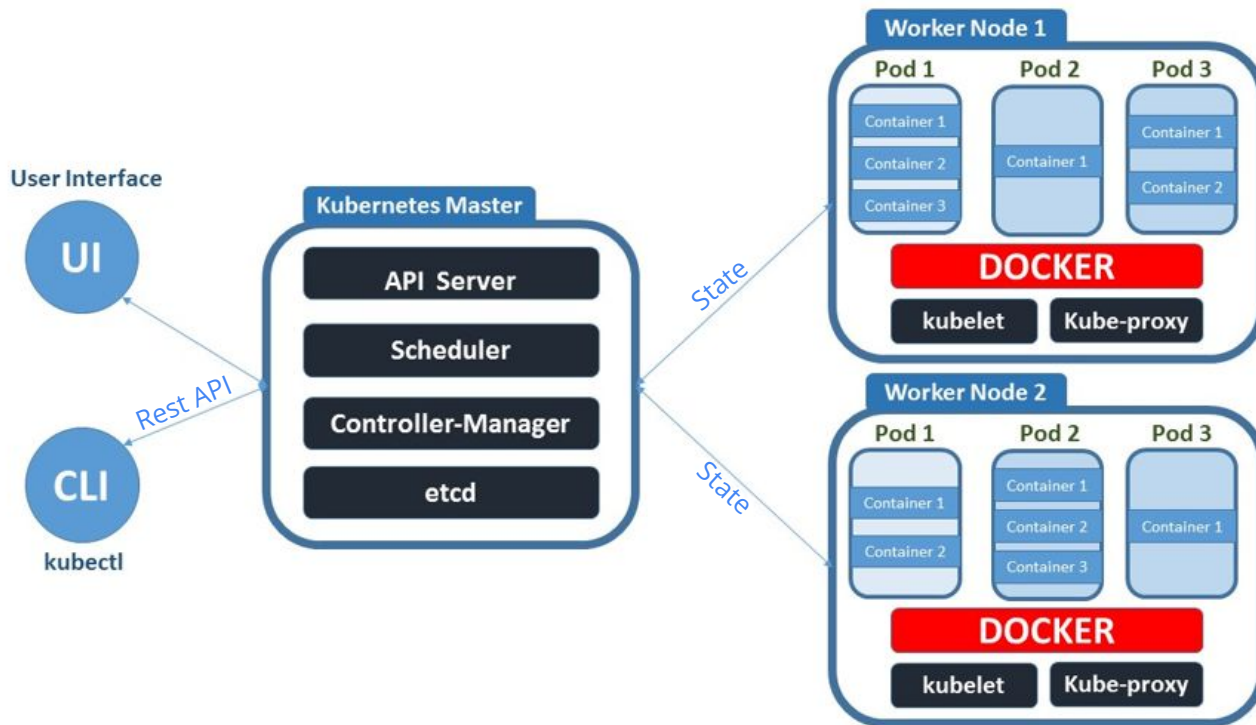
Powered by multiple disaggregated microservices



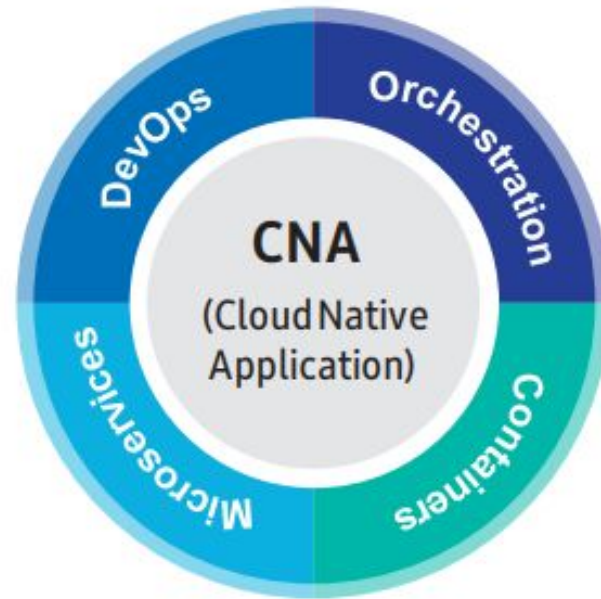
# Cloud-native technology stack architecture



# Topologia simplificada

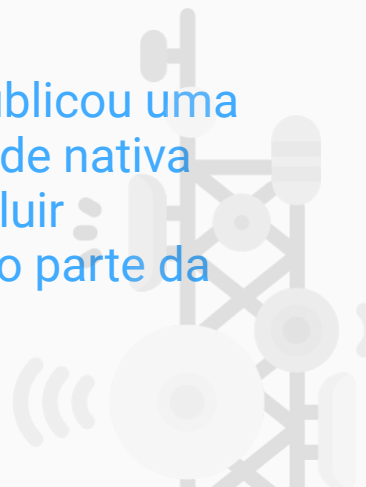


# Princípios do Cloud-native



# Onde entra o 5G?

- **Service Based Architecture** (SBA) proposta pelo 3GPP na Release 15 e do aprimoramento da Service Based Architecture (eSBA) na Release 16 (que estende o conceito de serviço de o plano de controle 5GC para a função do plano do usuário)
- O Instituto Europeu de Padrões de Telecomunicações (ETSI) publicou uma **arquitetura referenciada de NFV** para acomodar a função de rede nativa da nuvem (CNF) e aprimoramento da estrutura de NFV para incluir Zero-Touch, contêineres, balanceadores de carga e outros como parte da arquitetura de referência







# 5G



BROADBAND AND MEDIA EVERYWHERE



SMART VEHICLES. TRANSPORT



CRITICAL SERVICE AND INFRASTRUCTURE CONTROL



CRITICAL CONTROL OF REMOTE DEVICES



HUMAN MACHINE INTERACTION



SENSOR NETWORKS

A virtualização permite que as redes 5G suportem uma ampla gama de casos de uso, desde a Internet das Coisas (IoT) até aplicações de baixa latência, como realidade aumentada, veículos autônomos e vários outros casos

- Network Functions Virtualization
- Software Defined Networking
- *Edge Computing*
- Multi-Access Edge Computing (MEC)

# 5G como uma aplicação Cloud-native

- A velocidade com que a funcionalidade baseada em software da rede 5G pode ser desenvolvida, atualizada e substituída (teoricamente várias vezes ao dia)
- A agilidade com que pode ser implantado (em minutos, em vez de dias)
- A eficiência com que pode ser dimensionado e migrado para aproveitar as vantagens da economia da nuvem (os prestadores de serviços de comunicações (CSPs) estimam uma melhoria de 10 vezes)
- A resiliência (tempo de inatividade zero) resultante do seu design e automação incorporada

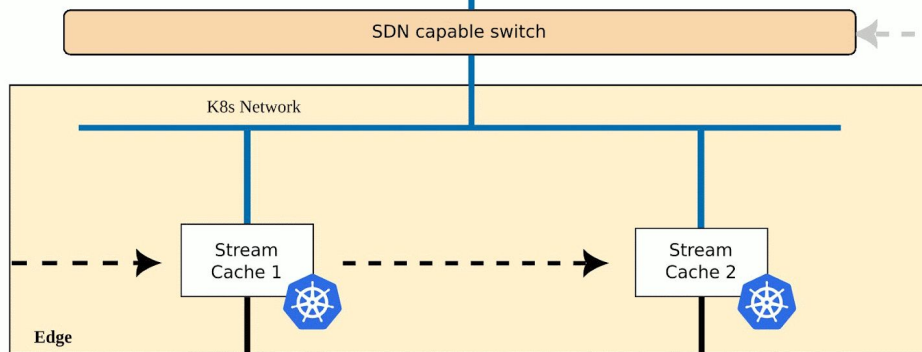
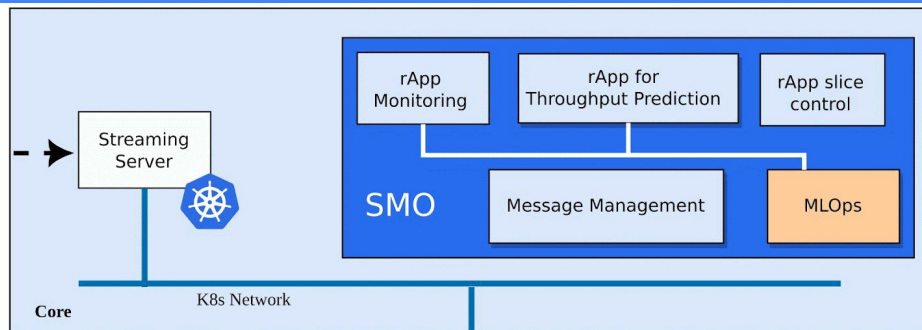
# Demonstração

# Intelligent QoE Control in Network Slice ORAN Systems





Instantiate  
KNFs

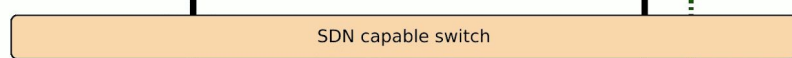
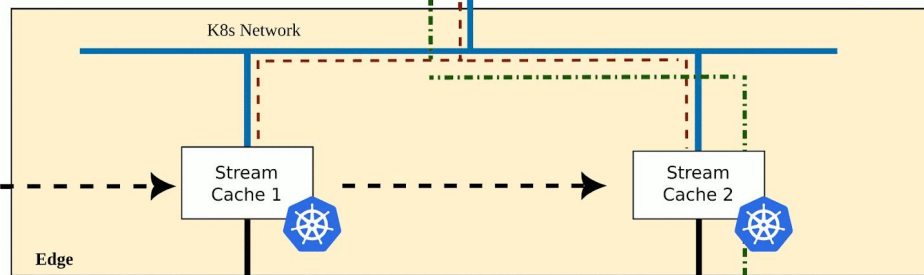
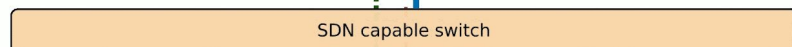
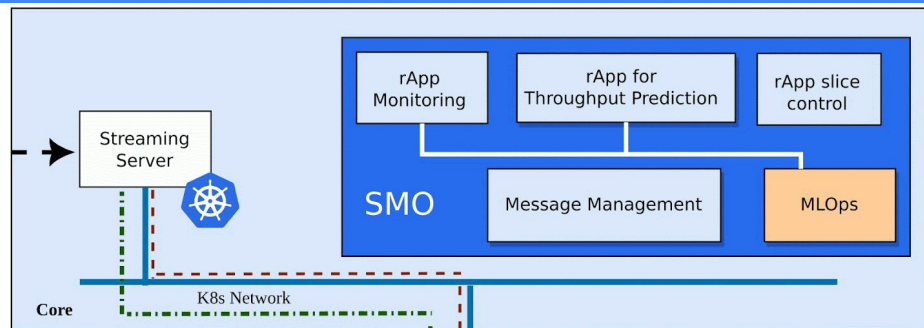


Flow Rules





Instantiate  
KNFS

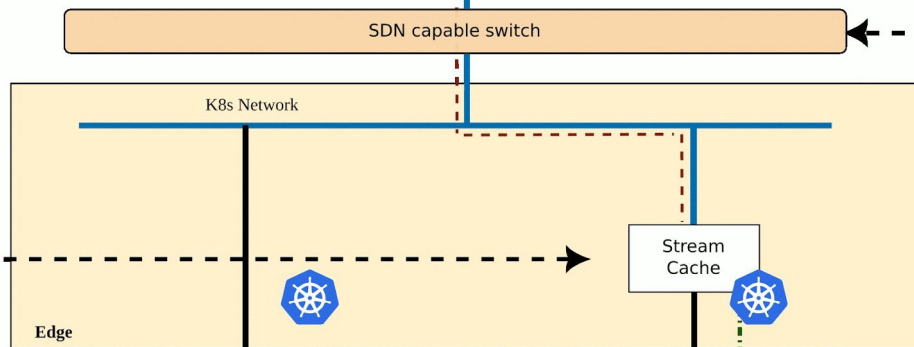
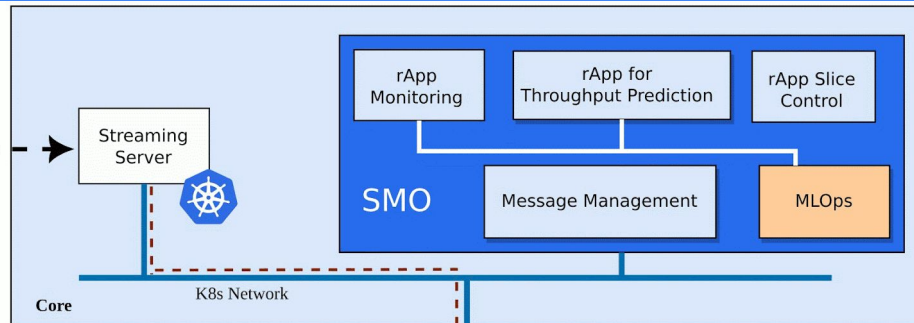


Flow Rules





Instantiate  
KNFs

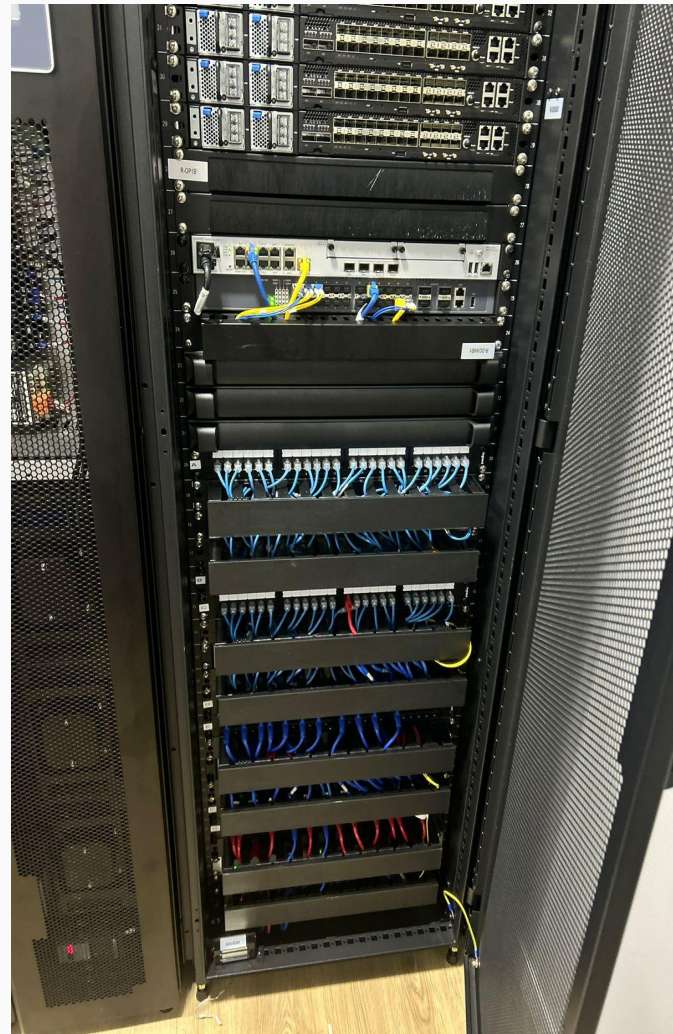


Flow Rules





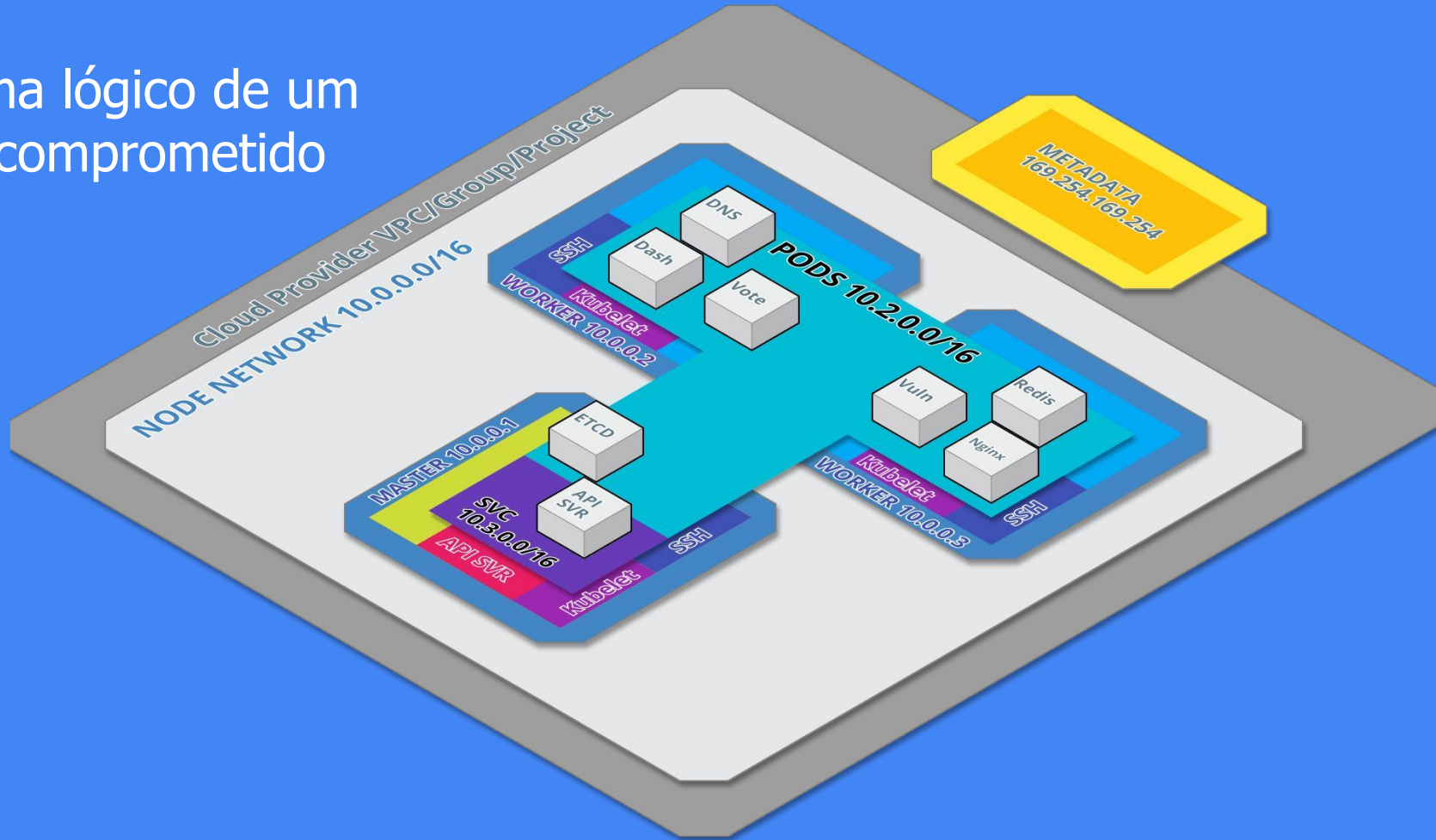




# Segurança

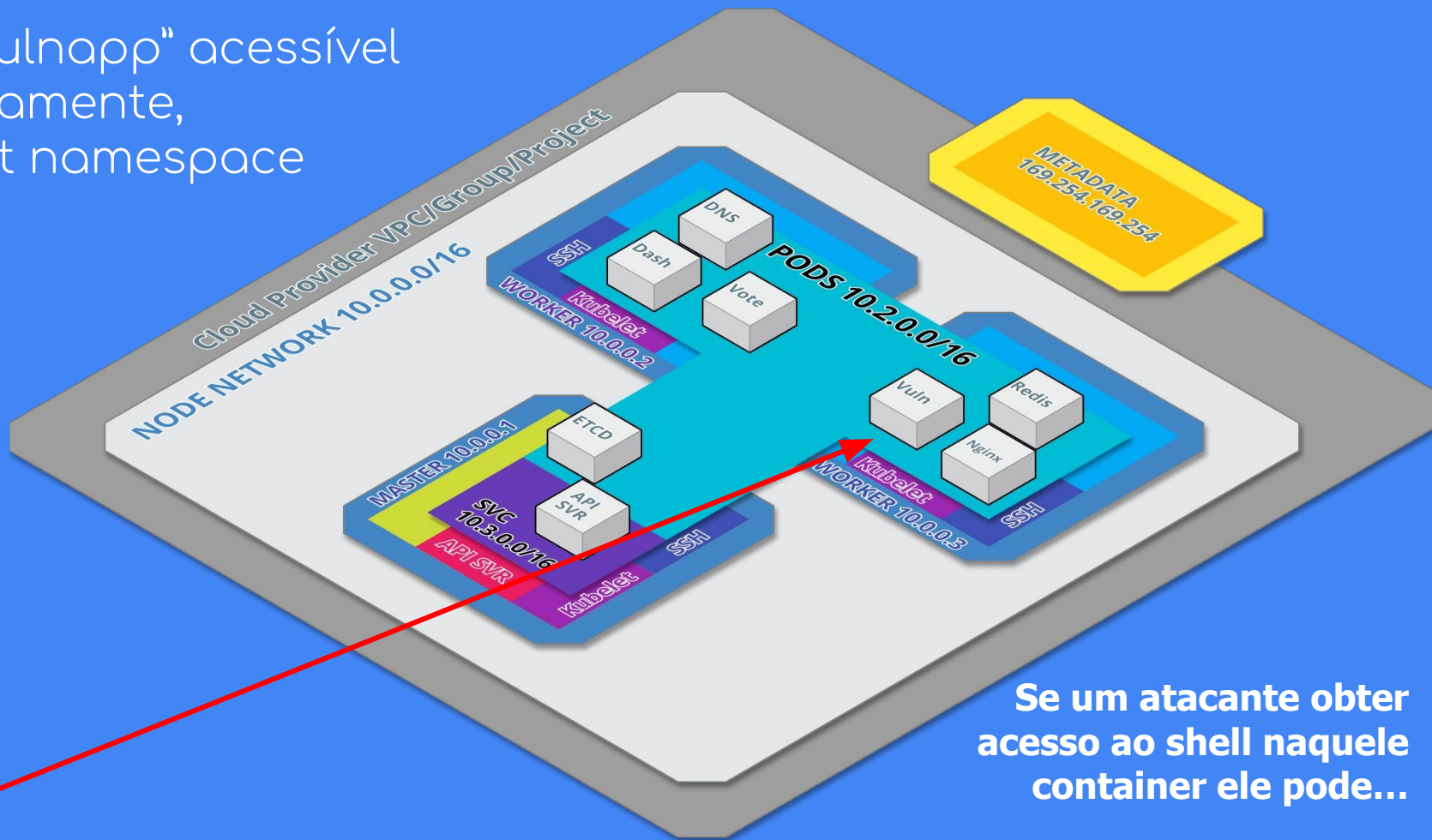


# Diagrama lógico de um cluster comprometido



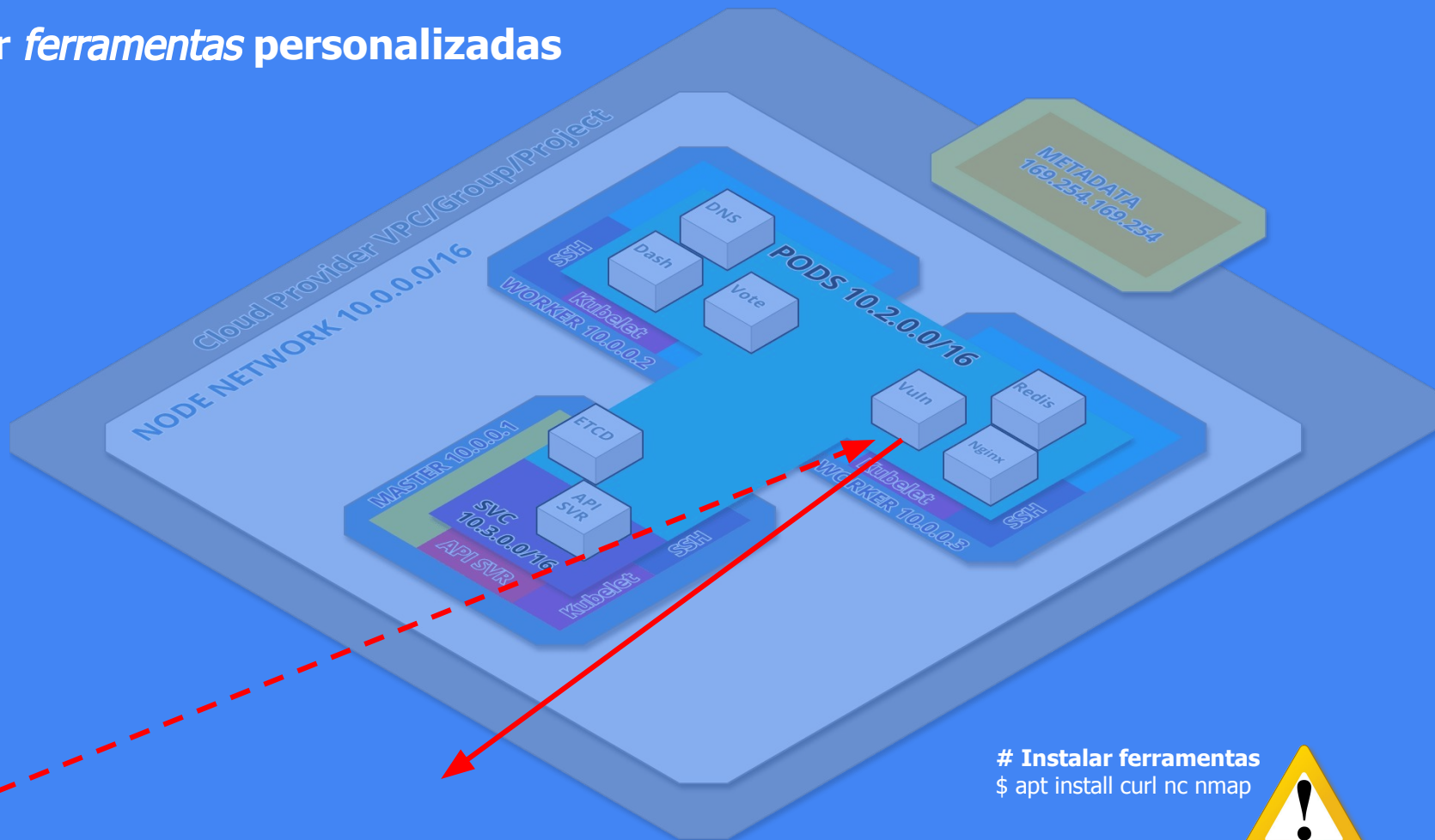


Pod "Vulnapp" acessível externamente, default namespace



Se um atacante obter acesso ao shell naquele container ele pode...

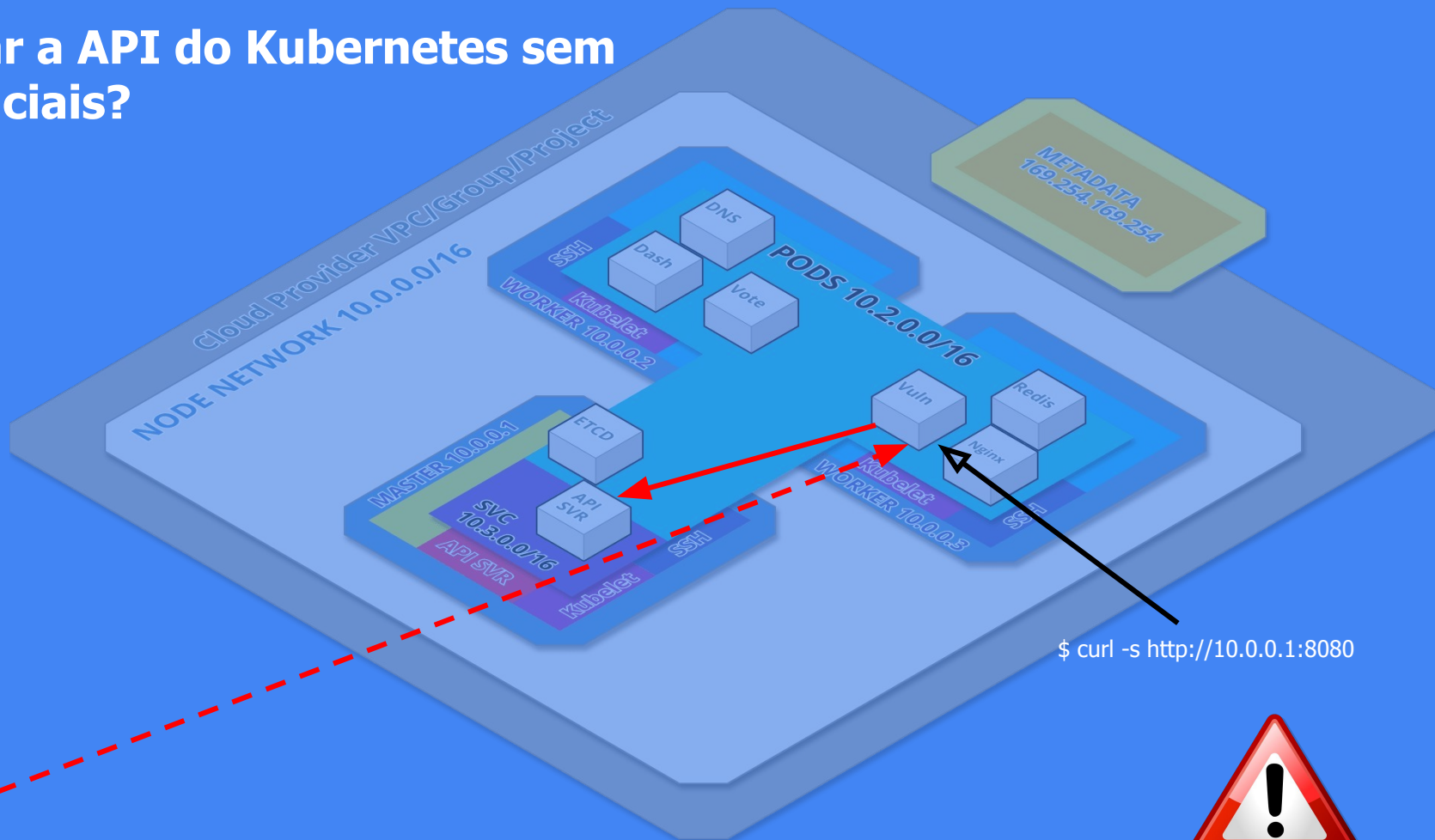
# Instalar *ferramentas* personalizadas



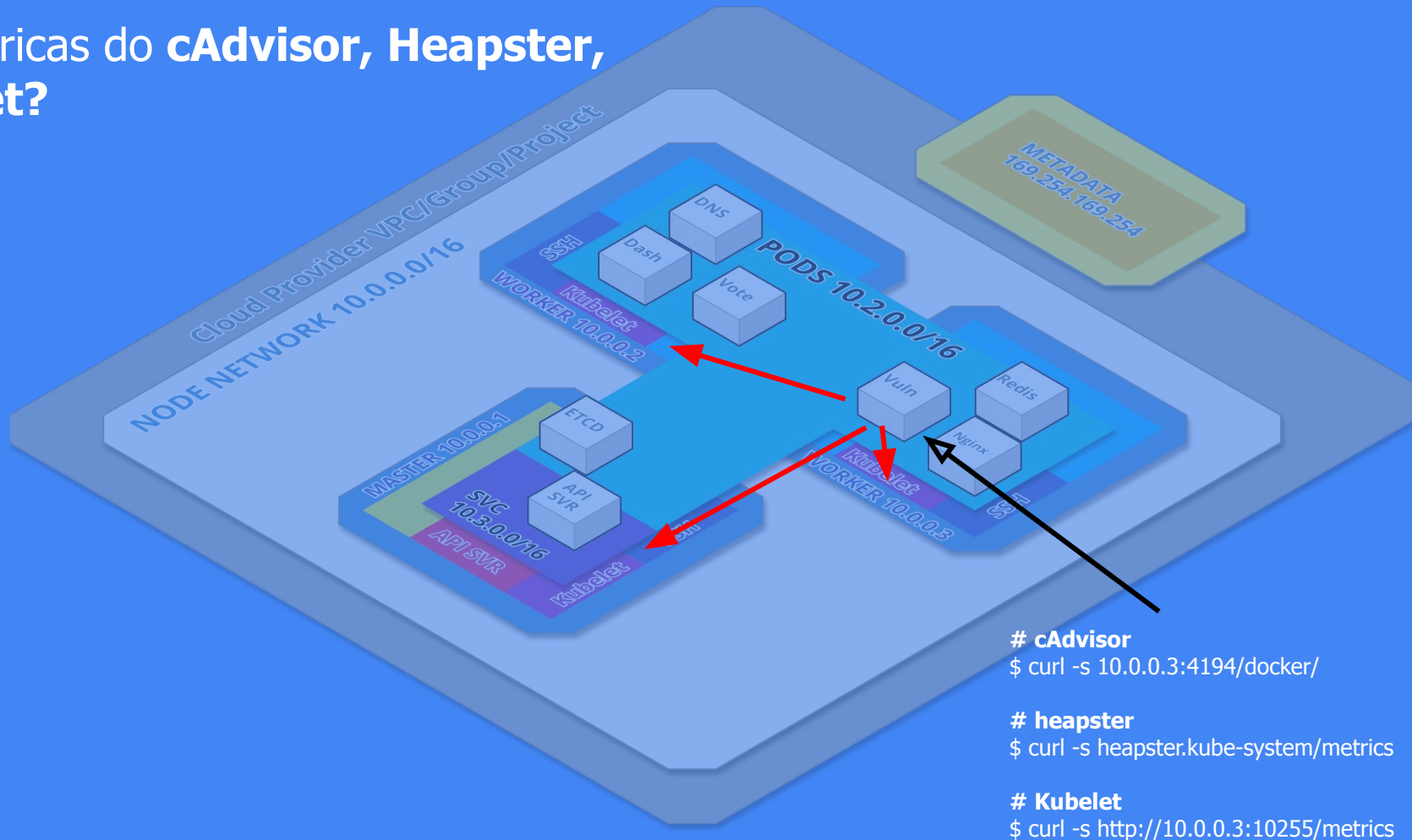
# Instalar ferramentas  
\$ apt install curl nc nmap



# Acessar a API do Kubernetes sem credenciais?

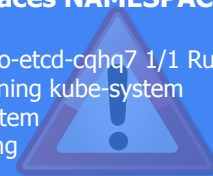
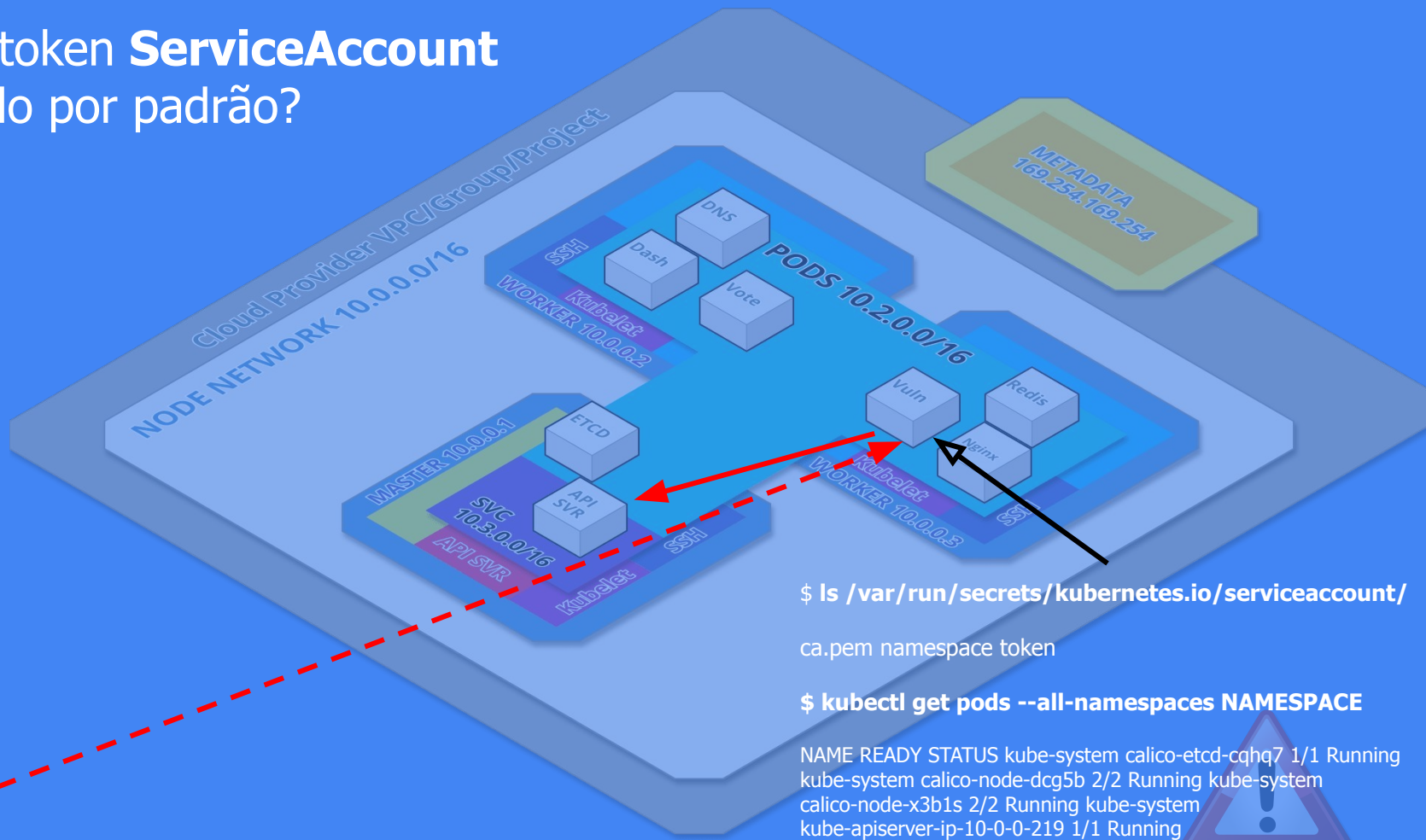


# Ler Métricas do **cAdvisor**, **Heapster**, **Kubelet**?

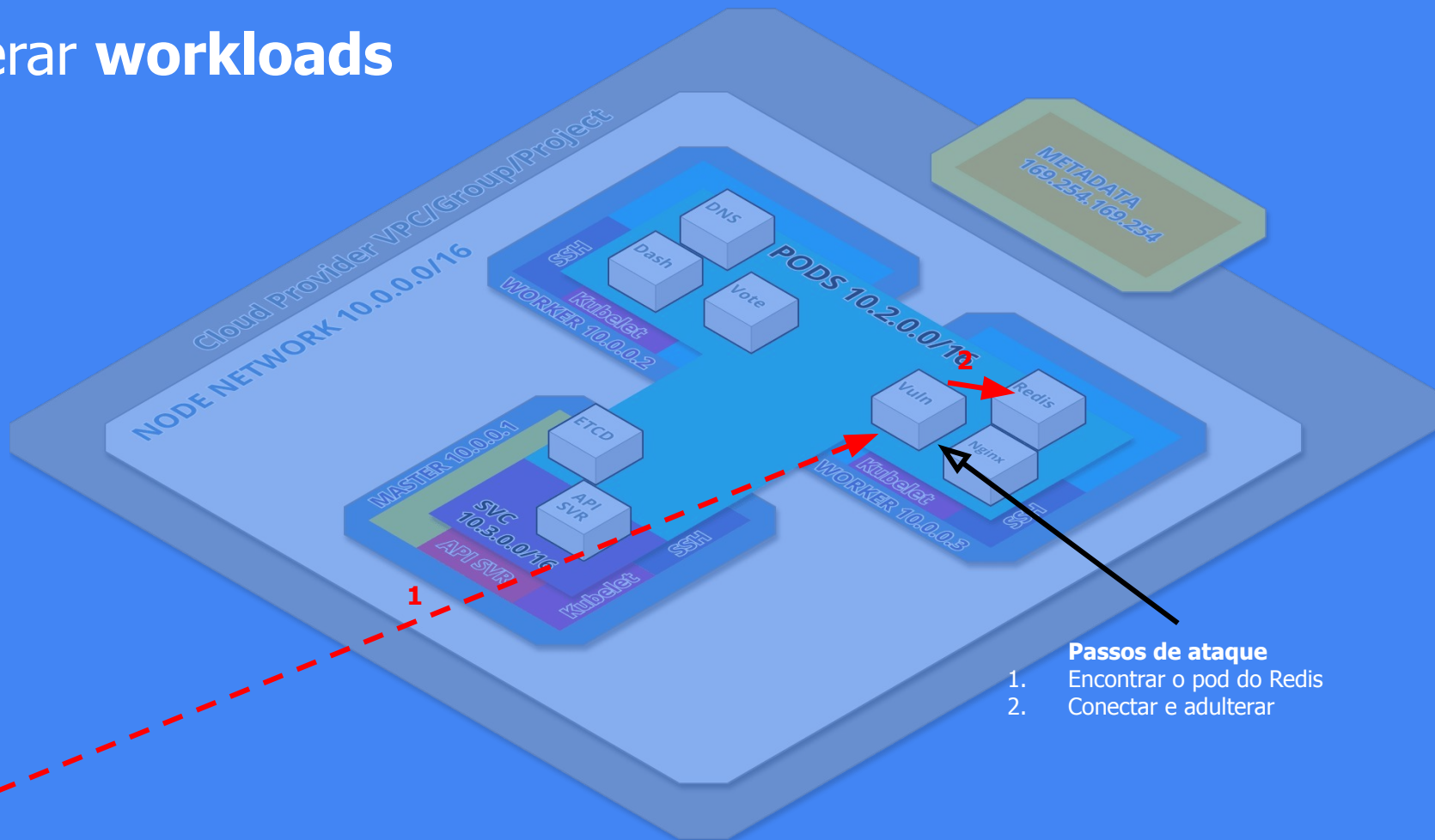




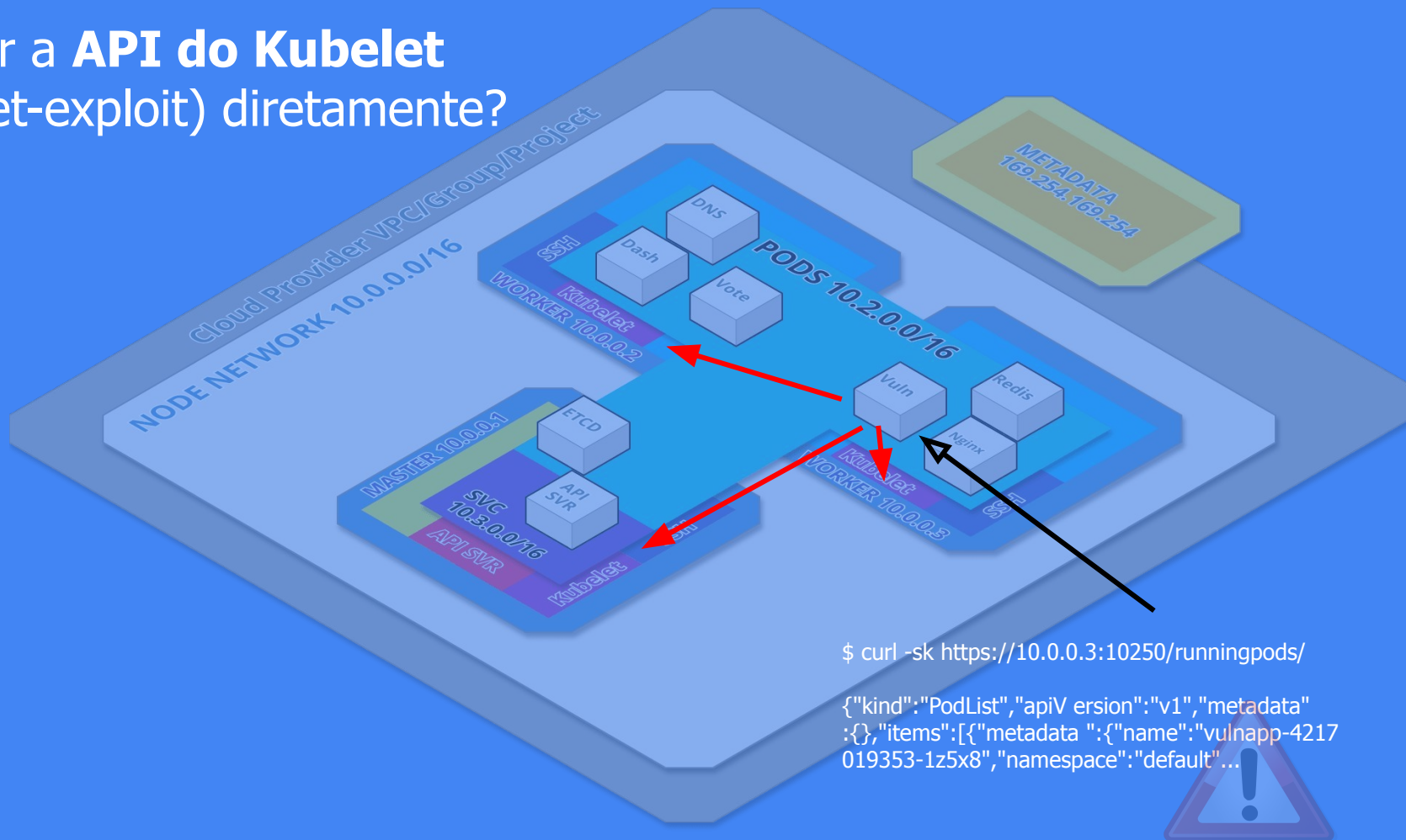
# Usar o token **ServiceAccount** montado por padrão?



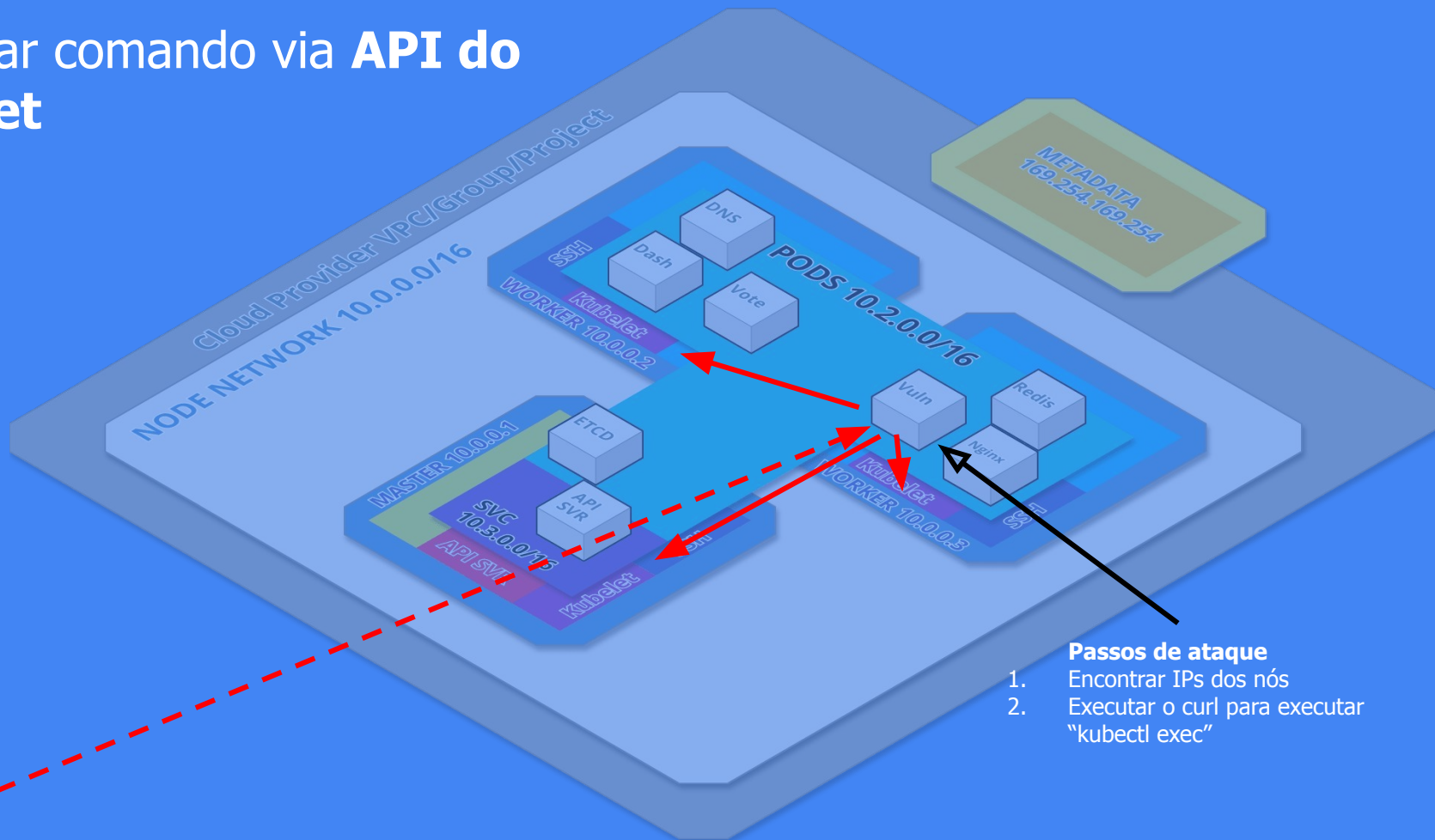
# Adulterar workloads



# Acessar a **API do Kubelet** (kubelet-exploit) diretamente?



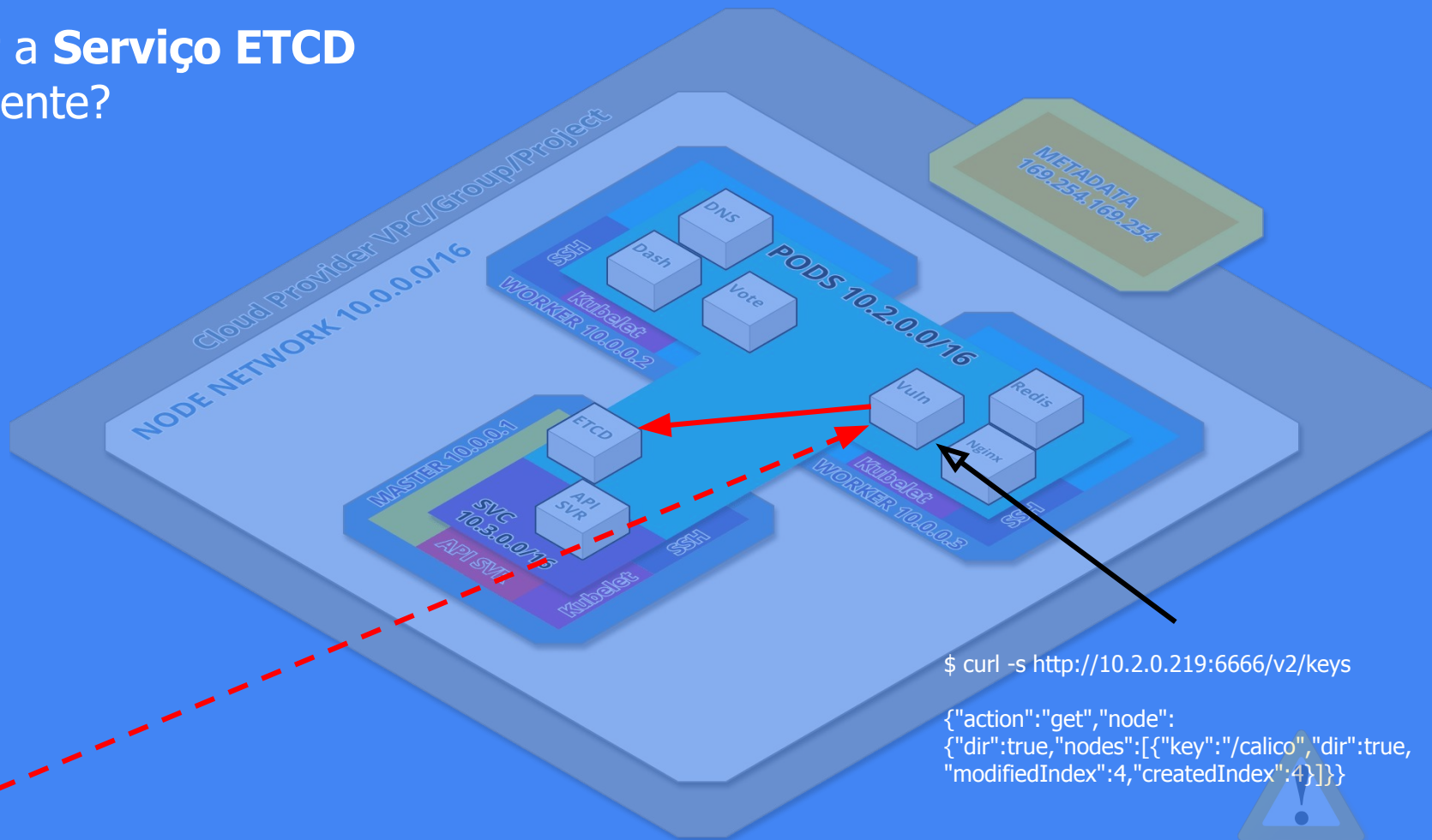
# Executar comando via **API do Kubelet**



## Passos de ataque

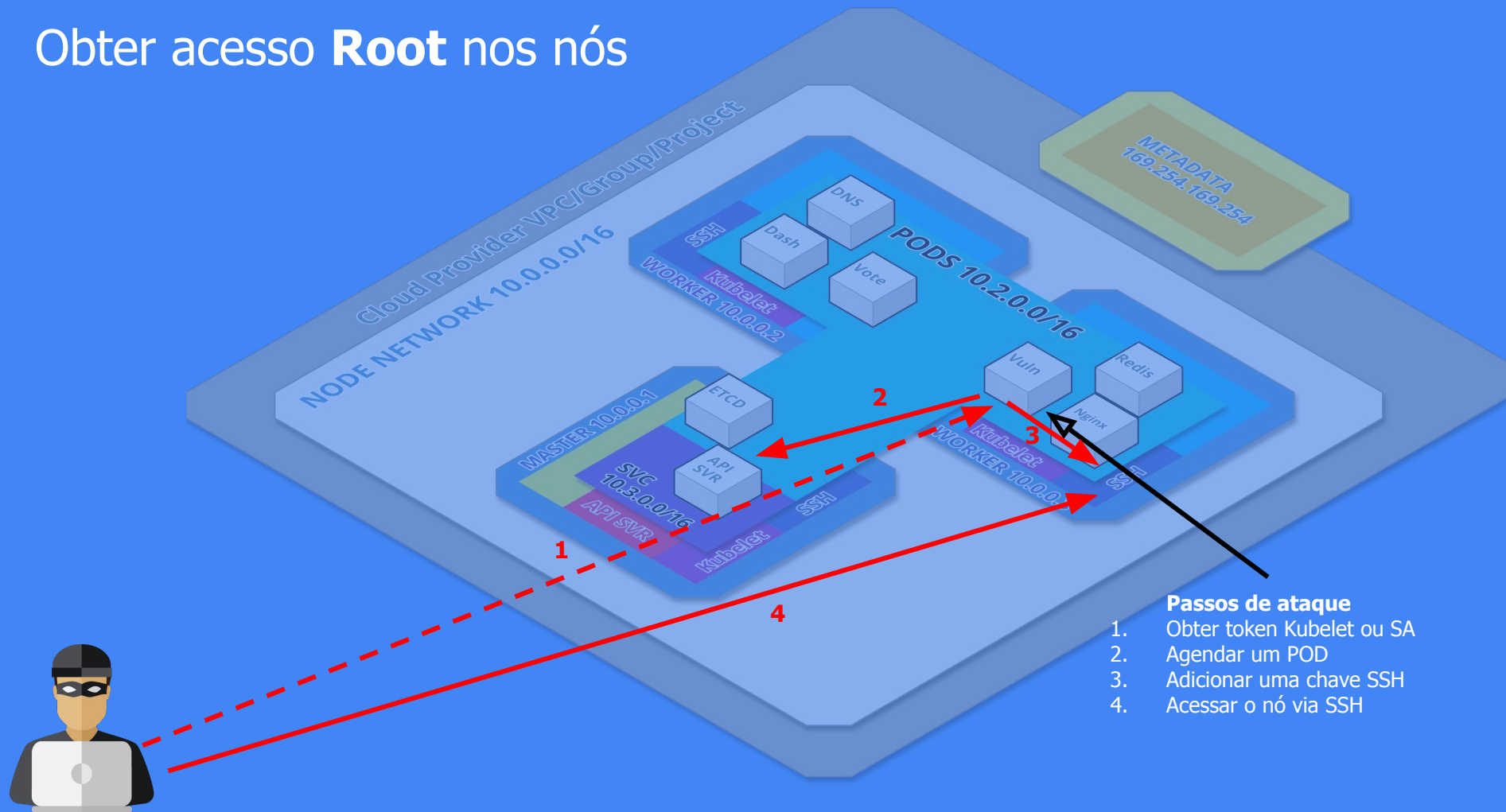
1. Encontrar IPs dos nós
2. Executar o curl para executar "kubectl exec"

# Acessar a **Serviço ETCD** diretamente?





# Obter acesso **Root** nos nós





**CLOUD NATIVE**  
— COMMUNITY GROUPS —  
RN

# Obrigado!



Experimente  
Kubernetes!

# Referências

- [1] <https://kubernetes.io/>
- [2] <https://docs.docker.com/get-started/>
- [3] <https://rominirani.com/learning-docker-move-to-the-cloud-3326369300ad>
- [4] <https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business/>
- [5] <https://thenewstack.io/happens-use-java-1960-ibm-mainframe/>
- [6] <https://blog.docker.com/2017/10/least-privilege-container-orchestration/>
- [7] <https://rancher.com/comparing-rancher-orchestration-engine-options/>
- [8] <https://medium.com/@jessgreb01/digging-into-docker-layers-c22f948ed612>
- [9] <https://www.bluedata.com/blog/2018/07/operation-stateful-bluek8s-and-kubernetes-director/kubernetes-reconciliation-loop/>
- [10] <https://kubernetes.io/blog/2018/04/30/zero-downtime-deployment-kubernetes-jenkins/>
- [11] <https://blog.openshift.com/make-a-kubernetes-operator-in-15-minutes-with-helm/>