

Começando com Kubernetes

Kubernetes é um sistema de orquestração de contêineres open-source que automatiza a implantação, o dimensionamento e a gestão de aplicações em contêineres. Ele foi originalmente projetado pelo Google e agora é mantido pela Cloud Native Computing Foundation. (WIKIPEDIA). O Kubernetes é escrito em Go e também é chamado de K8s.

Neste tutorial iremos aprender a instalar o Minikube, uma instalação do cluster do Kubernetes em apenas uma única máquina. O Minikube é usado apenas para aprendizado e não deve ser utilizado em ambiente de produção.

Também iremos aprender a fazer a instalação e configuração de um cluster kubernetes na segunda parte.

É necessário estar utilizando Linux e ter um mecanismo de virtualização instalado (VirtualBox, VMWare, etc). Assim como, para instalação do cluster, ter 3 máquinas (ou VMs) para serem utilizadas como nós.

1. Instalação do Minikube

*Primeiro iremos precisar realizar a instalação do kubectl, para isso iremos baixar o kubectl com curl, depois mudar as permissões do arquivo e movê-lo para o /usr/local/bin que está no path. Siga os comandos a seguir:

```
# curl -LO https://storage.googleapis.com/kubernetes-release/release/\$\(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt\)/bin/linux/amd64/kubectl  
# chmod +x kubectl && mv kubectl /usr/local/bin/
```

*Após ter realizado a instalação do kubectl iremos instalar o minikube, para isso realizaremos operações semelhantes as do minikube com curl, chmod e mv. Siga os comandos a seguir:

```
# curl -Lo minikube https://github.com/kubernetes/minikube/releases/download/v0.28.0/minikube-linux-amd64  
# chmod +x minikube && mv minikube /usr/local/bin/
```

*Pronto o Minikube está instalado!! Para inicializá-lo utilize o comando:

```
# minikube start
```

*Para ver todos os nós do cluster (apenas um) digite:

kubectl get nodes

1.1 Deployment de uma imagem do nginx para Teste

*Para realizar o deployment de uma imagem do nginx para teste do minikube execute o comando:

kubectl run nginx -- image nginx

*Para confirmar o deployment digite:

kubectl get deployment

*Você deverá ver o nginx rodando.

2. Instalando um cluster Kubernetes

*Para iniciar a instalação devemos ter as 3 máquinas a serem utilizadas no cluster abertas. Antes de iniciar a instalação atualizaremos a listagem de pacotes e instalaremos o apt-transport-https com os comandos:

apt-get update && apt install -y apt-transport-https

*Para utilizar o kubernetes devemos estar certos de que a swap está desligada e impedir que ela ligue novamente quando reiniciarmos as máquinas, comentando a entrada de swap em /etc/fstab , para isso siga os comandos:

swap off -a

vim /etc/fstab

*Depois faremos a instalação do Docker com um script na página do Docker:

curl -fsSL <https://get.docker.com> | bash

* Vamos realizar a instalação do kubectl, kubeadm e kubelet. Para isso iremos precisar adicionar o repositório do kubernetes. Então iremos importar a chave do repositório com o seguinte comando:

curl -s <https://packages.cloud.google.com/apt/doc/apt-key.gpg> | apt-key add -

* Iremos então adicionar o repositório a source list do Linux com o comando:

echo "deb <http://apt.kubernetes.io/> kubernetes-xenial main" > /etc/apt/sources.list.d/kubernetes.list

*Atualizando a listagem de pacotes e instalando o kubectl, kubeadm e kubelet:

apt-get update

apt-get install -y kubelet kubeadm kubectl

*O kubeadm é responsável pela montagem do cluster, o kubectl realiza as operações do cluster, e o kubelet é o agente do kubernetes que deve rodar em todos os nós e é responsável por se comunicar com a API do kubernetes no master. Também é importante lembrar de desativar qualquer firewall presente nas máquinas.

*Iremos iniciar o nosso cluster a partir da máquina escolhida como master, usando o kubeadm e passando o IP da nossa máquina como master. Digite o seguinte comando no master:

```
# kubeadm init --apiserver-advertise-address $(hostname -i)
```

*Na saída do comando serão mostrados os próximos passos para acrescentar nós ao cluster. O primeiro comando cria um diretório oculto na home do usuário para guardar as chaves do kubernetes, depois iremos fazer a copia do arquivo já criado em /etc/kubernetes e ajustar as permissões:

```
# mkdir -p $HOME/.kube
```

```
# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

```
# sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

*Devemos realizar a instalação do podnetwork do weavenet, que irá fornecer toda a comunicação entre os nós. Utilize o seguinte comando para isso:

```
# kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=\$\(kubectl version | base64 | tr -d '\n'\)"
```

*O comando kubeadm join deve ser copiado para as outras máquinas para que elas possam se juntar ao cluster, o comando será semelhante à:

```
# kubeadm join --token 39c341.a3bc3c4dd49758d5 --discovery-token-ca-cert-hash sha256:37092
```

*Iremos usar o kubectl para verificar os nós:

```
# kubectl get nodes
```

*Os nós devem mostrar Ready como status. Sua saída deve ser algo como:

```
NAME STATUS ROLES AGE VERSION
linux-debian0 Ready <none> 14s v1.11.0
linux-debian1 Ready master 14m v1.11.0
linux-debian2 Ready master 13m v1.11.0
```

O cluster está instalado e pronto para uso!

FONTES: <https://www.linuxtips.io/post/descomplicando-o-kubernetes-01>
<https://www.linuxtips.io/post/descomplicando-o-kubernetes-02>

AUTOR: WILLIAM GUILHERME E DE SOUZA