

Adicionando os demais nodes ao cluster

Agora que já temos o nosso cluster inicializado e já estamos entendendo muito bem o que é o arquivo `admin.conf`, chegou o momento de adicionar os demais nodes ao nosso cluster.

Para isso, vamos novamente utilizar o comando `kubeadm`, porém ao invés de executar o comando no node do control plane, nesse momento precisamos rodar o comando diretamente no node que queremos adicionar ao cluster.

Quando inicializamos o nosso cluster, o `kubeadm` nos mostrou o comando que precisamos executar no novos nodes, para que eles possam ser adicionados ao cluster como workers.

```
sudo kubeadm join 172.31.57.89:6443 --token if9hn9.xhxo6s89byj9rsmd \
    --discovery-token-ca-cert-hash
sha256:ad583497a4171d1fc7d21e2ca2ea7b32bdc8450a1a4ca4cfa2022748a99fa477
```

O comando `kubeadm join` é usado para adicionar um novo node ao cluster Kubernetes existente. Ele é executado nos worker nodes para que eles possam se juntar ao cluster e receber instruções do control plane. Vamos analisar as partes do comando fornecido:

- `kubeadm join`: O comando base para adicionar um novo nó ao cluster.
- `172.31.57.89:6443`: Endereço IP e porta do servidor de API do nó mestre (control plane). Neste exemplo, o nó mestre está localizado no IP `172.31.57.89` e a porta é `6443`.
- `--token if9hn9.xhxo6s89byj9rsmd`: O token é usado para autenticar o nó trabalhador no nó mestre durante o processo de adesão. Os tokens são gerados pelo nó mestre e têm uma validade limitada (por padrão, 24 horas). Neste exemplo, o token é `if9hn9.xhxo6s89byj9rsmd`.
- `--discovery-token-ca-cert-hash sha256:ad583497a4171d1fc7d21e2ca2ea7b32bdc8450a1a4ca4cfa2022748a99fa477`: Este é um hash criptográfico do certificado da autoridade de certificação (CA) do control plane. Ele é usado para garantir que o nó worker esteja se comunicando com o nó do control plane correto e autêntico. O valor após `sha256:` é o hash do certificado CA.

Ao executar este comando no worker, ele iniciará o processo de adesão ao cluster. Se o token for válido e o hash do certificado CA corresponder ao certificado CA do nó do control plane, o nó worker será autenticado e adicionado ao cluster. Após a adesão

bem-sucedida, o novo node começará a executar os Pods e a receber instruções do control plane, conforme necessário.

Após executar o join em cada worker node, vá até o node que criamos para ser o control plane, e execute:

```
kubect! get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
k8s-01	NotReady	control-plane	4m	v1.26.3
k8s-02	NotReady	<none>	3m	v1.26.3
k8s-03	NotReady	<none>	3m	v1.26.3

Agora você já consegue ver que os dois novos nodes foram adicionados ao cluster, porém ainda estão com o status Not Ready, pois ainda não instalamos o nosso plugin de rede para que seja possível a comunicação entre os pods. Vamos resolver isso agora. :)

Instalando o Weave Net

Agora que o cluster está inicializado, vamos instalar o Weave Net:

```
$ kubect! apply -f https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml
```

Aguarde alguns minutos até que todos os componentes do cluster estejam em funcionamento. Você pode verificar o status dos componentes do cluster com o seguinte comando:

```
kubect! get pods -n kube-system
```