

Visualizando detalhes dos nodes

Agora que já temos o nosso clustes com 03 nodes, nós podemos visualizar os detalhes de cada um deles, e assim entender cada detalhe.

Para ver a descrição do node, basta executar o comando abaixo:

```
kubectl describe node k8s-01
```

```
Name:          k8s-01
Roles:         control-plane
Labels:        beta.kubernetes.io/arch=amd64
               beta.kubernetes.io/os=linux
               kubernetes.io/arch=amd64
               kubernetes.io/hostname=k8s-01
               kubernetes.io/os=linux
               node-role.kubernetes.io/control-plane=
               node.kubernetes.io/exclude-from-external-load-balancers=
Annotations:   kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/containerd/containerd.sock
               node.alpha.kubernetes.io/ttl: 0
               volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp: Fri, 07 Apr 2023 11:52:46 +0000
Taints:        node-role.kubernetes.io/control-plane:NoSchedule
Unschedulable: false
Lease:
  HolderIdentity: k8s-01
  AcquireTime:    <unset>
  RenewTime:      Fri, 07 Apr 2023 12:49:09 +0000
Conditions:
  Type          Status LastHeartbeatTime          LastTransitionTime          Reason
  Message
  ----          -
  -----
```

NetworkUnavailable False Fri, 07 Apr 2023 11:57:03 +0000 Fri, 07 Apr 2023 11:57:03 +0000
WeaveIsUp Weave pod has set this
MemoryPressure False Fri, 07 Apr 2023 12:48:25 +0000 Fri, 07 Apr 2023 11:52:45 +0000
KubeletHasSufficientMemory kubelet has sufficient memory available
DiskPressure False Fri, 07 Apr 2023 12:48:25 +0000 Fri, 07 Apr 2023 11:52:45 +0000
KubeletHasNoDiskPressure kubelet has no disk pressure
PIDPressure False Fri, 07 Apr 2023 12:48:25 +0000 Fri, 07 Apr 2023 11:52:45 +0000
KubeletHasSufficientPID kubelet has sufficient PID available
Ready True Fri, 07 Apr 2023 12:48:25 +0000 Fri, 07 Apr 2023 11:57:05 +0000
KubeletReady kubelet is posting ready status. AppArmor enabled

Addresses:

InternalIP: 172.31.57.89

Hostname: k8s-01

Capacity:

cpu: 2

ephemeral-storage: 7941576Ki

hugepages-2Mi: 0

memory: 4015088Ki

pods: 110

Allocatable:

cpu: 2

ephemeral-storage: 7318956430

hugepages-2Mi: 0

memory: 3912688Ki

pods: 110

System Info:

Machine ID: c8a6ad1dd24342c48ba303688d3ada1f

System UUID: ec2b271b-8df3-f164-b01c-3b5078a2d15b

Boot ID: 93ae6b0c-13fa-432d-b15a-d3725b6c0e72

Kernel Version: 5.15.0-1031-aws

OS Image: Ubuntu 22.04.2 LTS

Operating System: linux

Architecture: amd64

Container Runtime Version: containerd://1.6.20

Kubelet Version: v1.26.3

Kube-Proxy Version: v1.26.3

PodCIDR: 10.10.0.0/24

PodCIDRs: 10.10.0.0/24

Non-terminated Pods: (6 in total)

Namespace	Name	CPU Requests	CPU Limits	Memory Requests	Memory Limits	Age
kube-system	etcd-k8s-01	100m (5%)	0 (0%)	100Mi (2%)	0 (0%)	56m
kube-system	kube-apiserver-k8s-01	250m (12%)	0 (0%)	0 (0%)	0 (0%)	56m
kube-system	kube-controller-manager-k8s-01	200m (10%)	0 (0%)	0 (0%)	0 (0%)	56m
kube-system	kube-proxy-skpfc	0 (0%)	0 (0%)	0 (0%)	0 (0%)	56m
kube-system	kube-scheduler-k8s-01	100m (5%)	0 (0%)	0 (0%)	0 (0%)	56m
kube-system	weave-net-hks8s	100m (5%)	0 (0%)	0 (0%)	0 (0%)	52m

Allocated resources:

(Total limits may be over 100 percent, i.e., overcommitted.)

Resource	Requests	Limits
cpu	750m (37%)	0 (0%)
memory	100Mi (2%)	0 (0%)
ephemeral-storage	0 (0%)	0 (0%)
hugepages-2Mi	0 (0%)	0 (0%)

Events:

Type	Reason	Age	From	Message
Normal	Starting	56m	kube-proxy	
Normal	Starting	56m	kubelet	Starting kubelet.

```

Warning InvalidDiskCapacity 56m kubelet invalid capacity 0 on image filesystem
Normal NodeHasSufficientMemory 56m kubelet Node k8s-01 status is now:
NodeHasSufficientMemory
Normal NodeHasNoDiskPressure 56m kubelet Node k8s-01 status is now:
NodeHasNoDiskPressure
Normal NodeHasSufficientPID 56m kubelet Node k8s-01 status is now:
NodeHasSufficientPID
Normal NodeAllocatableEnforced 56m kubelet Updated Node Allocatable limit across pods
Normal RegisteredNode 56m node-controller Node k8s-01 event: Registered Node k8s-01 in
Controller
Normal NodeReady 52m kubelet Node k8s-01 status is now: NodeReady

```

Na saída do comando acima é possível ver detalhes como o nome do node, o IP interno, o hostname, a capacidade de CPU, memória, armazenamento, pods, etc. Também é possível ver os pods que estão rodando no node, os recursos alocados e os eventos que ocorreram no node.

Caso você queira visualizar detalhes dos outros dois nodes, basta utilizar o comando abaixo:

```
kubectl get nodes k8s-02 -o wide
```

```
kubectl get nodes k8s-03 -o wide
```

```

NAME      STATUS  ROLES    AGE  VERSION  INTERNAL-IP  EXTERNAL-IP  OS-IMAGE
KERNEL-VERSION  CONTAINER-RUNTIME
k8s-02    Ready   <none>   59m  v1.26.3  172.31.59.34 <none>       Ubuntu 22.04.2 LTS  5.15.0-1031-
aws       containerd://1.6.20

```

Estou utilizando o parâmetro `-o wide` para que o comando retorne mais detalhes sobre o node, como o IP externo e o IP interno.

E claro, você ainda pode utilizar o comando `kubectl describe node` para visualizar mais detalhes dos demais nodes, como fizemos para o node `k8s-01`.