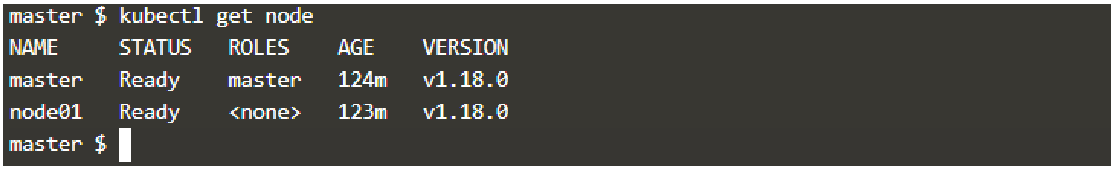
检查 K8S 预装环境



```
master $ kubectl get node
NAME      STATUS    ROLES     AGE     VERSION
master    Ready     master    124m    v1.18.0
node01    Ready     <none>    123m    v1.18.0
master $
```

下载 MULTUS 插件到目录下

```
$ git clone https://github.com/intel/multus-cni.git && cd multus-cni
```

安装 multus CNI

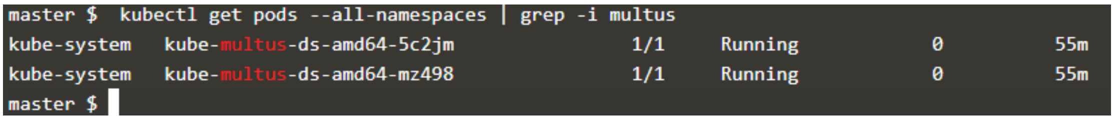
https://github.com/intel/multus-cni/blob/master/doc/quickstart.md

https://github.com/intel/multus-cni/blob/master/doc/how-to-use.md

```
$ cat ./images/multus-daemonset.yml | kubectl apply -f -
```

检查 MULTUS CNI 安装状态

```
$ kubectl get pods --all-namespaces | grep -i multus
```



```
master $  kubectl get pods --all-namespaces | grep -i multus
kube-system    kube-multus-ds-amd64-5c2jm              1/1       Running            0           55m
kube-system    kube-multus-ds-amd64-mz498              1/1       Running            0           55m
master $
```

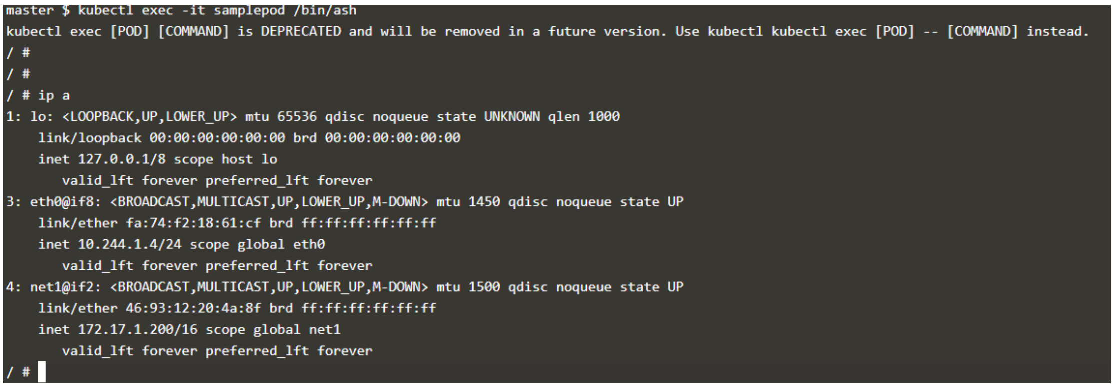
配置 MACVLAN 配置文件，按需更换宿主接口和 IP 地址范围（黄色标注部分为宿主接口）WORD 直接刷配置有问题，请在官方配置中截取配置并修改，以下部分仅仅为备注说明，仅供参考。

```
cat <<EOF | kubectl create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: macvlan-conf
spec:
  config: '{
      "cniVersion": "0.3.0",
      "type": "macvlan",
      "master": "eth0",
      "mode": "bridge",
      "ipam": {
        "type": "host-local",
        "subnet": "172.17.1.0/16",
        "rangeStart": "172.17.1.200",
        "rangeEnd": "172.17.1.240",
        "routes": [
          { "dst": "0.0.0.0/0" }
        ],
        "gateway": "172.17.1.1"
      }
    }'
EOF
```

运行测试 POD，名字为 samplepod，黄色部分调用 Macvlan-conf 配置

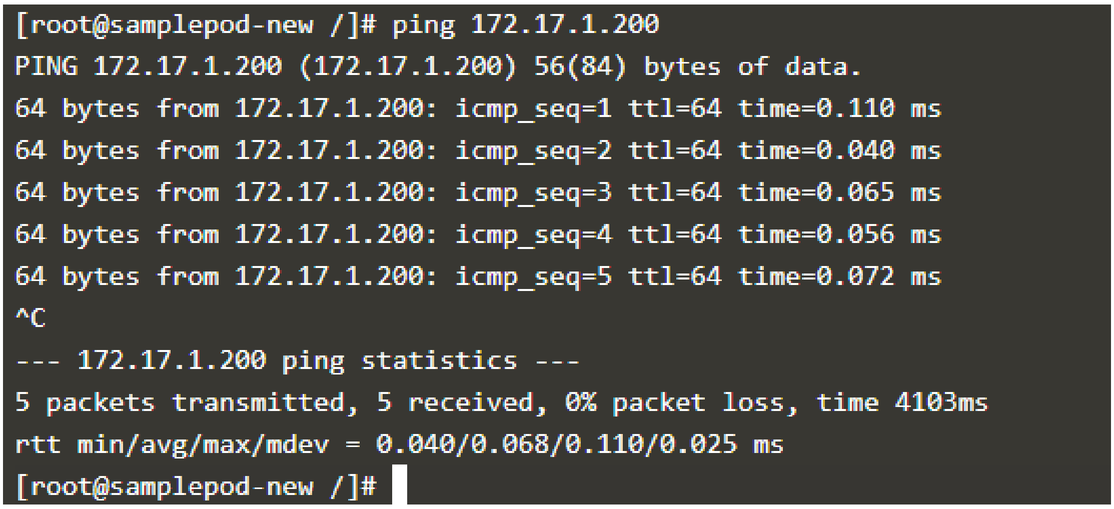
```
cat <<EOF | kubectl create -f -
apiVersion: v1
kind: Pod
metadata:
  name: samplepod
  annotations:
    k8s.v1.cni.cncf.io/networks: macvlan-conf
spec:
  containers:
  - name: samplepod
    command: ["/bin/ash", "-c", "trap : TERM INT; sleep infinity & wait"]
    image: alpine
EOF
```

登录进 POD 检查 IP 地址段为 macvlan 配置文件中调用的地址段，net1@if2 为 macvlan 创建的接口，eth0 忽略不计



```
master $ kubectl exec -it samplepod /bin/ash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl kubectl exec [POD] -- [COMMAND] instead.
/ #
/ #
/ # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: eth0@if8: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1450 qdisc noqueue state UP
    link/ether fa:74:f2:18:61:cf brd ff:ff:ff:ff:ff:ff
    inet 10.244.1.4/24 scope global eth0
       valid_lft forever preferred_lft forever
4: net1@if2: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether 46:93:12:20:4a:8f brd ff:ff:ff:ff:ff:ff
    inet 172.17.1.200/16 scope global net1
       valid_lft forever preferred_lft forever
/ #
```

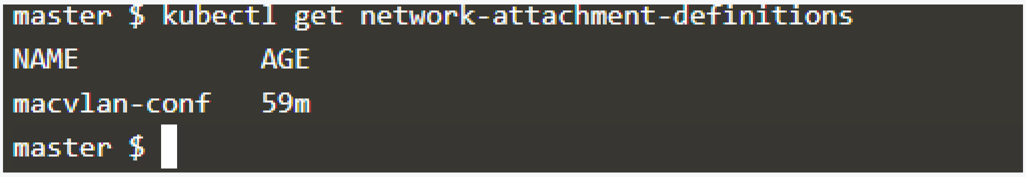
检查 Pod 间连通性



```
[root@samplepod-new /]# ping 172.17.1.200
PING 172.17.1.200 (172.17.1.200) 56(84) bytes of data.
64 bytes from 172.17.1.200: icmp_seq=1 ttl=64 time=0.110 ms
64 bytes from 172.17.1.200: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 172.17.1.200: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 172.17.1.200: icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from 172.17.1.200: icmp_seq=5 ttl=64 time=0.072 ms
^C
--- 172.17.1.200 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4103ms
rtt min/avg/max/mdev = 0.040/0.068/0.110/0.025 ms
[root@samplepod-new /]#
```

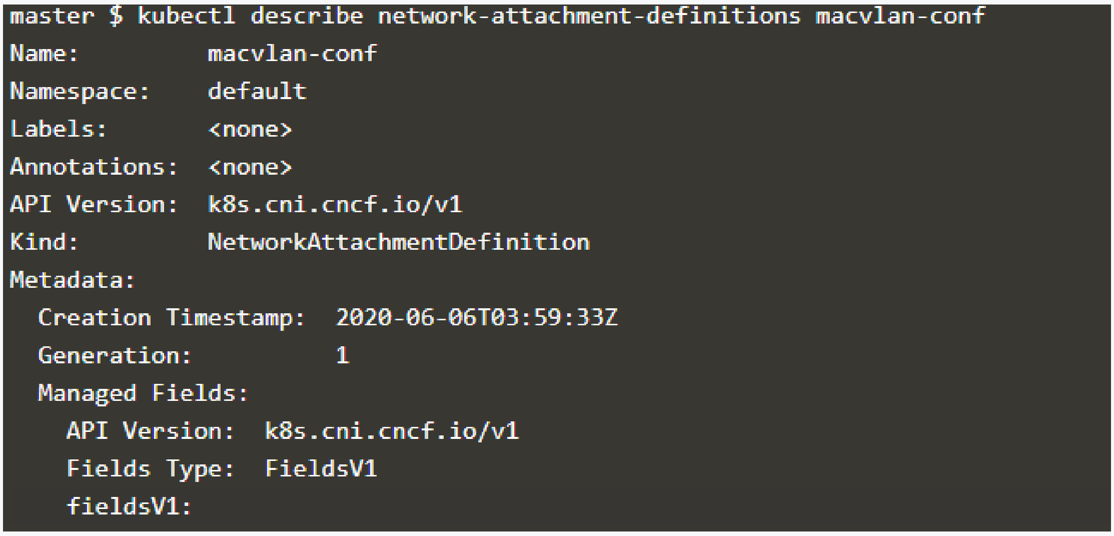
检查创建的 network attachement defi

```
kubectl get network-attachment-definitions
```



```
master $ kubectl get network-attachment-definitions
NAME           AGE
macvlan-conf   59m
master $
```

```
kubectl describe network-attachment-definitions macvlan-conf
```



```
master $ kubectl describe network-attachment-definitions macvlan-conf
Name:         macvlan-conf
Namespace:    default
Labels:       <none>
Annotations:  <none>
API Version:  k8s.cni.cncf.io/v1
Kind:         NetworkAttachmentDefinition
Metadata:
  Creation Timestamp:  2020-06-06T03:59:33Z
  Generation:          1
  Managed Fields:
    API Version:  k8s.cni.cncf.io/v1
    Fields Type:  FieldsV1
    fieldsV1:
```

创建 POD 指定缺省路由从指定接口出局，非 eth0 出局
本示例为创建的 centos 镜像，登录方式为

```
Kubectl exec -it samplepod-centos /bin/bash
Ip a
Ip route show
Route -n
```

```
cat <<EOF | kubectl create -f -
apiVersion: v1
kind: Pod
metadata:
  name: samplepod-centos
  annotations:
    k8s.v1.cni.cncf.io/networks: '[{
      "name": "macvlan-conf",
      "default-route": ["172.17.0.1"]
    }]'
spec:
  containers:
  - name: samplepod
    command: ["/bin/bash", "-c", "trap : TERM INT; sleep infinity & wait"]
    image: dougbtv/centos-network
EOF
```

/etc/kubernetes/kubelet

```
KUBELET_ARGS="--network-plugin=cni --cni-conf-dir=/etc/cni/net.d --cni-bin-dir=/opt/cni/bin"
```

```
systemctl daemon-reload && systemctl restart kubelet
```