作者：李晓辉

联系方式：

1. 微信：Lxh_Chat

2. 邮箱：939958092@qq.com

Kubernetes 是一个强大的容器管理平台，它可以帮助我们管理容器集的网络和服务网络。容器集网络为每个容器集提供了一个网络接口，这样容器集就可以相互通信了。在某些情况下，我们可能需要将一些容器集连接到其他网络，这样可以带来一些好处，比如提高特定流量的性能，或者满足一些特定的安全性要求。

例如，假设我们有一个需要处理大量数据的容器集，我们希望这个容器集的网络流量能够得到优先处理，以提高数据处理的速度。在这种情况下，我们可以使用 Multus CNI 插件将这个容器集连接到一个专用的网络上，这个专用网络可以提供更高的带宽和更低的延迟，从而提高数据处理的效率。

再比如，假设我们有一个需要处理敏感数据的容器集，我们希望这个容器集的网络流量能够得到额外的安全保护。在这种情况下，我们可以使用 Multus CNI 插件将这个容器集连接到一个具有更高安全性的专用网络上，这个专用网络可以提供更严格的访问控制和加密措施，从而提高数据的安全性。

Multus CNI 插件是一个非常有用的工具，它可以帮助我们将容器集附加到自定义网络上。这些自定义网络可以是集群外部的现有网络，也可以是集群内部的自定义网络。通过使用 Multus CNI 插件，我们可以根据不同的需求将容器集连接到不同的网络上，从而提高容器集的性能和安全性。

Multus 辅助网络案例

确认网络环境现状

在我们的课程环境中，master01这个节点上既是控制面也是数据面，所有的工作负载都运行在此，我们看看它的网络接口

ens4 接口是额外的网络接口，可用于需要额外网络的练习。此接口连接到 192.168.51.0/24 网络，其 IP 地址为 192.168.51.10

```
[student@workstation ~]$ oc debug node/master01 -- chroot /host ip addr
Temporary namespace openshift-debug-d5mzc is created for debugging node...
Starting pod/master01-debug-s9fgk ...
To use host binaries, run `chroot /host`
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state
    link/ether 52:54:00:00:32:0a brd ff:ff:ff:ff:ff:ff
    altname enp0s3
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default q
    link/ether 52:54:00:01:33:0a brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 192.168.51.10/24 brd 192.168.51.255 scope global dynamic noprefixroute ens4
       valid_lft 412787654sec preferred_lft 412787654sec
    inet6 fe80::878:11eb:73df:8a1b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

和master01有路由的机器是utility这台，只有这台才能ping通ens4的ip，而workstation是不行的，我们来试试

```
[student@workstation ~]$ ping -c 1 192.168.51.10
PING 192.168.51.10 (192.168.51.10) 56(84) bytes of data.

--- 192.168.51.10 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

[student@workstation ~]$ ssh root@utility
[root@utility ~]# ping -c 1 192.168.51.10
PING 192.168.51.10 (192.168.51.10) 56(84) bytes of data.
64 bytes from 192.168.51.10: icmp_seq=1 ttl=64 time=0.683 ms

--- 192.168.51.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.683/0.683/0.683/0.000 ms
```

网络验证好之后，我们要知道，稍后我们在集群的pod中，添加的额外接口，只能在utility这台机器上才能访问和ping通

向集群发布业务

```yaml
cat > deployment-service.yml <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: multus-test
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: registry.ocp4.example.com:8443/redhattraining/hello-world-nginx:latest
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 8080
EOF
```

```
oc create -f deployment-service.yml
```

确认服务工作正常

```
[student@workstation ~]$ oc get pod -o wide
multus-test-6645d8bb58-mgrfn       1/1     Running   0          2m33s   10.8.0.155   master
```

向集群发布辅助网络

这里我们发布了一个名为custom的网络，这个网络和主机上的ens4接口关联，并对外提供192.168.51.10/24

```
cat > multus-network.yml <<-'EOF'
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  name: custom
spec:
  config: |-
    {
      "cniVersion": "0.3.1",
      "name": "custom",
      "type": "host-device",
      "device": "ens4",
      "ipam": {
        "type": "static",
        "addresses": [
          {"address": "192.168.51.10/24"}
        ]
      }
    }
EOF
```

```
[student@workstation ~]$ oc create -f multus-network.yml
networkattachmentdefinition.k8s.cni.cncf.io/custom created
[student@workstation ~]$ oc get -f multus-network.yml
NAME     AGE
custom   3s
[student@workstation ~]$ oc describe -f multus-network.yml
Name:         custom
Namespace:    laoli
Labels:       <none>
Annotations:  <none>
API Version:  k8s.cni.cncf.io/v1
Kind:         NetworkAttachmentDefinition
Metadata:
  Creation Timestamp:  2024-12-20T12:11:08Z
  Generation:          1
  Resource Version:    262311
  UID:                 16b47917-5729-4e2e-b5a6-ca98a0227969
Spec:
  Config:  {
  "cniVersion": "0.3.1",
  "name": "custom",
  "type": "host-device",
  "device": "ens4",
  "ipam": {
    "type": "static",
    "addresses": [
      {"address": "192.168.51.10/24"}
    ]
  }
}
Events:  <none>
```

更新业务pod添加辅助网络

写一个补丁，用于添加我们的辅助网络

```
cat > multus-patch.yaml <<-EOF
spec:
  template:
    metadata:
      annotations:
        k8s.v1.cni.cncf.io/networks: custom
EOF
```

更新我们的业务pod

```
oc patch deployment multus-test --patch-file multus-patch.yaml
```

确认业务pod已经拥有辅助网络

很好，我们看到pod已经有了net1这个网卡，并拥有192.168.51.10这个地址

```
[student@workstation ~]$ oc get pod multus-test-66858889c-9jvh7 -o yaml | grep -B 20 custom
apiVersion: v1
kind: Pod
metadata:
  annotations:
    k8s.ovn.org/pod-networks: '{"default":{"ip_addresses":["10.8.0.157/23"],"mac_address":"
    k8s.v1.cni.cncf.io/network-status: |-
      [{
          "name": "ovn-kubernetes",
          "interface": "eth0",
          "ips": [
              "10.8.0.157"
          ],
          "mac": "0a:58:0a:08:00:9d",
          "default": true,
          "dns": {}
      },{
          "name": "laoli/custom",
          "interface": "net1",
          "ips": [
              "192.168.51.10"
          ],
          "mac": "52:54:00:01:33:0a",
          "dns": {}
      }]
    k8s.v1.cni.cncf.io/networks: custom
```

通过辅助网络访问业务

```
[root@utility ~]# ping -c 1 192.168.51.10
PING 192.168.51.10 (192.168.51.10) 56(84) bytes of data.
64 bytes from 192.168.51.10: icmp_seq=1 ttl=64 time=0.872 ms

--- 192.168.51.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.872/0.872/0.872/0.000 ms
[root@utility ~]# curl 192.168.51.10:8080
<html>
  <body>
    <h1>Hello, world from nginx!</h1>
  </body>
</html>
```

本文档在线版本：https://www.linuxcenter.cn