Multiple Networks and Isolation in Kubernetes

Haibin Michael Xie / Principal Architect Huawei
Agenda

• CNI and network plug-ins
• Multiple network use cases, design and implementation
• Network multi-tenancy requirement and implementation
• Demo
CNI and Network Plug-ins

• **What is CNI**
  – Common container network interface specification and libraries for writing plugins to configure network interfaces in Linux containers.

• **Assign IP to pod**
  – Kubelet startup parameter `--network-plugin=cni`
  – `--pod-cidr` for pod IP addresses
  – Network plugin assigns one IP from the CIDR to each pod

• **Many third party network plugins**
  – [https://github.com/containernetworking/cni](https://github.com/containernetworking/cni)
Definition of Multiple Networks

- Multiple physical networks
- Multiple logical networks
  - Multiple network interfaces per container
  - Multiple network address spaces per cluster
  - Multiple network tenants per cluster
  - ...

Kubernetes Cluster

Physical Network 1
- Logical Network A
- Logical Network B
- Logical Network C
- Pod

Physical Network 2
- Logical Network D
- Pod
Multiple Networks

Node 1

- Load Balancer
  - Service 1
  - Service 2
  - eth0
  - flannel0
  - veth0
  - eth0

- SrcPod
  - eth1

Node 2

- Load Balancer
  - Service 1
  - Service 2
  - eth0
  - flannel0
  - veth0
  - eth0

- DestPod A
  - eth1

- DestPod B
  - eth1
Why Multiple Networks

- **Logical network abstraction**
  - IP space, quota/speed, network policies

- **Multiple network tenants**
  - Physical isolation and logical isolation

- **Use multiple network solutions**

- **User scenarios:**
  - NFV: access to control plane, data plane and monitor plane
  - Applications that want to separate different traffic such as video streaming application
  - IPV6 co-existing with IPV4
  - Applications have both internal and public access
  - Servers that want to isolate traffic from multiple clients
  - Utilizing multiple physical NICs on host
Changes to Kubernetes

- New physical network object
- New logical network object
- Pod object with multiple networks
- Service in specific logical network
- Network based scheduling
- Network tenancy – isolation, bandwidth, QPS limiting etc
Multiple Network Workflow

Physical network

Logical network

PhysicalNetworks.yaml
```yaml
"provider": [
  {
    name: phy_net0,
    description: "***"
  },
  {
    name: phy_net1,
    description: "***"
  }
]
```

Service.yaml
```yaml
kind: Service
apiVersion: v1
metadata:
  name: my-service
  annotations:
    network: management
spec:
  selector:
    app: MyApp
  ports:
```

HostPhysicalNetwork.yaml
```yaml
"provider": [
  {
    alias: phy_net0,
    ref_nic: eth1,
  },
  {
    alias: phy_net1,
    ref_nic: eth2,
  }
]
```

Register node

LogicalNetwork.yaml
```yaml
apiVersion: v1
kind: Network
metadata:
  name: management
  labels:
spec:
  physicalNet: phy_net0
  plugin: Flannel|iCan
  subnet: 10.10.0.0/16
```

pod.yaml
```yaml
spec:
  containers:
    - image: test-webserver
      name: test-container
      metadata:
        annotations:
          networks:
            management:eth0
data:eth1
```

Service.yaml
```yaml
kind: Service
apiVersion: v1
metadata:
  name: my-service
  annotations:
    network: management
spec:
  selector:
    app: MyApp
  ports:
```

CNS Master (iCan master, flannel master ...)

Network plugins

ETCD

ApiServer

Scheduler

kubelet

CNI Genie

iCan

flannel

...
Network Tenancy Requirements

• Network isolation among tenants
  – Limit access to other tenants’ containers/services
  – Limit access to host network
  – Limit access to other tenant’s network resources like load balancers and DNS records

• Network connectivity
  – Containers have internet access
  – Allow services to have external IP for ingress
  – Access other tenants’ containers/services
Network Tenancy How

- Logical network, Kubernetes namespace and tenant mappings
- Network isolation:
  - Physical isolation
  - IPTables
  - VLAN/VXLAN
- DNS isolation – access control, dedicated DNS
- Gateway for ingress/egress
- Misc:
  - NodePort?
  - Support multiple namespaces and/or multiple logical networks in one tenant
  - Network based scheduling
  - Network quota allocation
  - Tenancy in federated clusters, cross data center or region
CNI-Genie

- Multiple physical and logical networks
- Adaptor to any network plug-in
- Network isolation with policy
- Admission control: validation, access control, scheduling
- SLA monitoring and enforcement
Example Usage

List of slave nodes

```
$ sudo /opt/paas/kubernetes/kubectl --client-certificate=tls.crt --client-key=tls.key --certificate-authority=ca.crt -s https://100.106.74.140:5443 get nodes -n multinet
NAME  STATUS AGE
multinet-1 Ready 5d
multinet-2 Ready 6d
multinet-3 Ready 6d
```

Node description

```
$ sudo /opt/paas/kubernetes/kubectl --client-certificate=tls.crt --client-key=tls.key --certificate-authority=ca.crt -s https://100.106.74.140:5443 get node multinet-1 -n multinet -o yaml
apiVersion: v1
kind: Node
metadata:
  annotations:
    network.alpha.kubernetes.io/mappings: networkmapping1
  volumes.kubernetes.io/controller-managed-attach-detach: "true"
  creationTimestamp: 2017-08-08T08:18:15Z
  enable: true
  labels:
    beta.kubernetes.io/arch: amd64
    beta.kubernetes.io/os: linux
    kubernetes.io/hostname: multinet-1
    network.alpha.kubernetes.io/phynt1: eth1
  os.architecture: amd64
```

List of Physical Networks

```
$ sudo /opt/paas/kubernetes/kubectl --client-certificate=tls.crt --client-key=tls.key --certificate-authority=ca.crt -s https://100.106.74.140:5443 get pn
NAME   TYPE  PVID  AGE
phynt1 overlay_12 1 6d
```

List of Logical Networks

```
$ sudo /opt/paas/kubernetes/kubectl --client-certificate=tls.crt --client-key=tls.key --certificate-authority=ca.crt -s https://100.106.74.140:5443 get net
NAME   PHYNET TYPE  SUBNET  AGE
net1   phynt1 overlay_12 122.29.0.0/16  6d
```
Example Usage

Deploy pod

```bash
root@root1-ThinkPad-T440p:/home/root1/app-yaml/new-crd-yaml# cat app-weave-flannel-multi.yaml | grep -e "^" -e "networks" -e "eth"
apiversion: v1
kind: Pod
metadata:
    name: nginx-logicalnet-wv-fln1-multi5555
labels:
    app: web
annotations:
    cni: 
    networks: net1:eth0,net2:eth4
spec:
    containers:
    - name: key-value-store
      image: busybox
      command: ["top"]
      imagePullPolicy: IfNotPresent
```

```bash
root@root1-ThinkPad-T440p:/home/root1/app-yaml/new-crd-yaml# kubectl create -f app-weave-flannel-multi.yaml
pod "nginx-logicalnet-wv-fln1-multi5555" created
root@root1-ThinkPad-T440p:/home/root1/app-yaml/new-crd-yaml# kubectl get pods | grep -e "^" -e "nginx-logicalnet-wv-fln1-multi5555"
NAME                      READY STATUS    RESTARTS AGE
nginx-logicalnet-wv-fln1-multi5555   1/1   Running   0   17s
```
Query pod

```
# /home/test
$ kubectl --client-certificate-tls.crt --client-key-tls.key --certificate-authority-ca.crt get pods -n multinet -o wide

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
<th>IP</th>
<th>NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>fuxi-72bds</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>6d</td>
<td>100.106.122.158</td>
<td>multinet-3</td>
</tr>
<tr>
<td>fuxi-qltam</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>6d</td>
<td>100.106.122.215</td>
<td>multinet-2</td>
</tr>
<tr>
<td>fuxi-x08dz</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>5d</td>
<td>100.106.75.232</td>
<td>multinet-1</td>
</tr>
<tr>
<td>nginx</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
<td>172.16.0.99,122.20.0.99</td>
<td>multinet-1</td>
</tr>
</tbody>
</table>
```

root@karun-cni-dev:/yamls/demo# kubectl exec -ti nginx-multip-per-container ip a | highlight -w red+b '10.32.*' -w red+b 'eth1' -w green+b

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qdisc qeth0
   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

3: eth0@lf4337: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1400 qdisc noqueue state UP group default
   link/ether d6:76:ca:d5:46:7d brd ff:ff:ff:ff:ff:ff
   inet 10.244.0.167/32 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::d476:caff:fed5:467d/64 scope link
       valid_lft forever preferred_lft forever
   48338: eth1@lf4339: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1376 qdisc noqueue state UP group default
   link/ether ca:5b:2b:e4:be:50 brd ff:ff:ff:ff:ff:ff
   inet 10.32.0.3/12 scope global eth1
       valid_lft forever preferred_lft forever
   inet6 fe80::c85b:2bff:feee:be50/64 scope link
       valid_lft forever preferred_lft forever
```
More?

Code repository: https://github.com/Huawei-PaaS/CNI-Genie/

Watch demo videos:
- Physical network and logical network: https://asciinema.org/a/xU5JJEJwq11LS3yiqnlyJRCZh
- Multiple IPs per pod: https://asciinema.org/a/120338
- Co-existence of multiple plugins: https://asciinema.org/a/120279
- CNI-Genie admission control: https://asciinema.org/a/KLptT8j37JNjBTwkvZpgvkbui
- Network policy controller: https://asciinema.org/a/kn4J3PCDx0Hzj3Me7A19qrnsW
Thank you

Haibin Michael Xie
haibin.michael.xie@huawei.com
wechat: 153346957