

SRIOV device plugin

- [Goal](#)
- [Implement details](#)
- [Use SRIOV device plugin in ICN](#)

Goal

This Wiki describes the steps on how to deploy SRIOV device plugin in edge location node and how to use SRIOV device plugin in ICN.

Implement details

The SRIOV network device plugin is Kubernetes device plugin for discovering and advertising SRIOV network virtual functions (VFs) in a Kubernetes host. and it provides below features:

- Handles SRIOV capable/not-capable devices (NICs and Accelerators alike)
- Supports devices with both Kernel and userspace(uio and VFIO) drivers
- Allows resource grouping using "Selector"
- User configurable resourceName
- Detects Kubelet restarts and auto-re-register
- Detects Link status (for Linux network devices) and updates associated VFs health accordingly
- Extensible to support new device types with minimal effort if not already supported

To automatically deploy SRIOV network device plugin in ICN, KuD addon SRIOV device plugin includes:

Installation script:

(1) Install and probe Intel Nic kernel driver (e.g. iavf.ko for Intel Ethernet Controller 700 series etc.), it requires an input parameter for the interface (e.g. eno2) which the virtual functions will be created on.

(2) Enable virtual functions (VFs) on assigned interface.

Daemon set yaml:

(1) sriov-cni.yaml: deploy sriov cni plugin as daemon set in edge location node, the sriov cni plugin is responsible to create network interface based on allocated SRIOV VF when creating pod.

(2) sriovdp-daemonset.yaml: deploy sriov network device plugin as daemon set in edge location node, the sriov network device plugin is responsible to discovery and advertise SRIOV VFs to K8s.

Use SRIOV device plugin in ICN

(1) Create NetworkAttachmentDefinition for SRIOV interfaces

Create sriov-nad.yaml with below contents:

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: sriov-eno2
  annotations:
    k8s.v1.cni.cncf.io/resourceName: intel.com/intel_sriov_700
spec:
  config: '{
    "type": "sriov",
    "ipam": {
      "type": "host-local",
      "subnet": "10.56.206.0/24",
      "routes": [
        { "dst": "0.0.0.0/0" }
      ],
      "gateway": "10.56.206.1"
    }
  }'
```

Apply to K8s with sriov-eno2 NetworkAttachmentDefinition:

```
cat sriov-nad.yaml | kubectl apply -f -
```

(2) Create Pod to request SRIOV interface

Create sriov-eno2-pod.yaml with below contents:

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-case-01
  annotations:
    k8s.v1.cni.cncf.io/networks: sriov-eno2
spec:
  containers:
    - name: test-pod
      image: docker.io/centos/tools:latest
      command:
        - /sbin/init
      resources:
        requests:
          intel.com/intel_sriov_700: '1'
        limits:
          intel.com/intel_sriov_700: '1'
```

Create a pod:

```
cat sriov-eno2-pod.yaml | kubectl apply -f -
```

Verify the SRIOV interface in the created pod (e.g. net1 as below):

```
kubectl exec -i pod-case-01 ip addr

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
3: eth0@if108: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UP group default
    link/ether a2:e7:69:81:c7:3e brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.244.0.162/24 scope global eth0
        valid_lft forever preferred_lft forever
36: net1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether da:e9:13:20:f5:98 brd ff:ff:ff:ff:ff:ff
    inet 10.56.206.6/24 scope global net1
        valid_lft forever preferred_lft forever
```

References:

<https://github.com/intel/sriov-network-device-plugin>