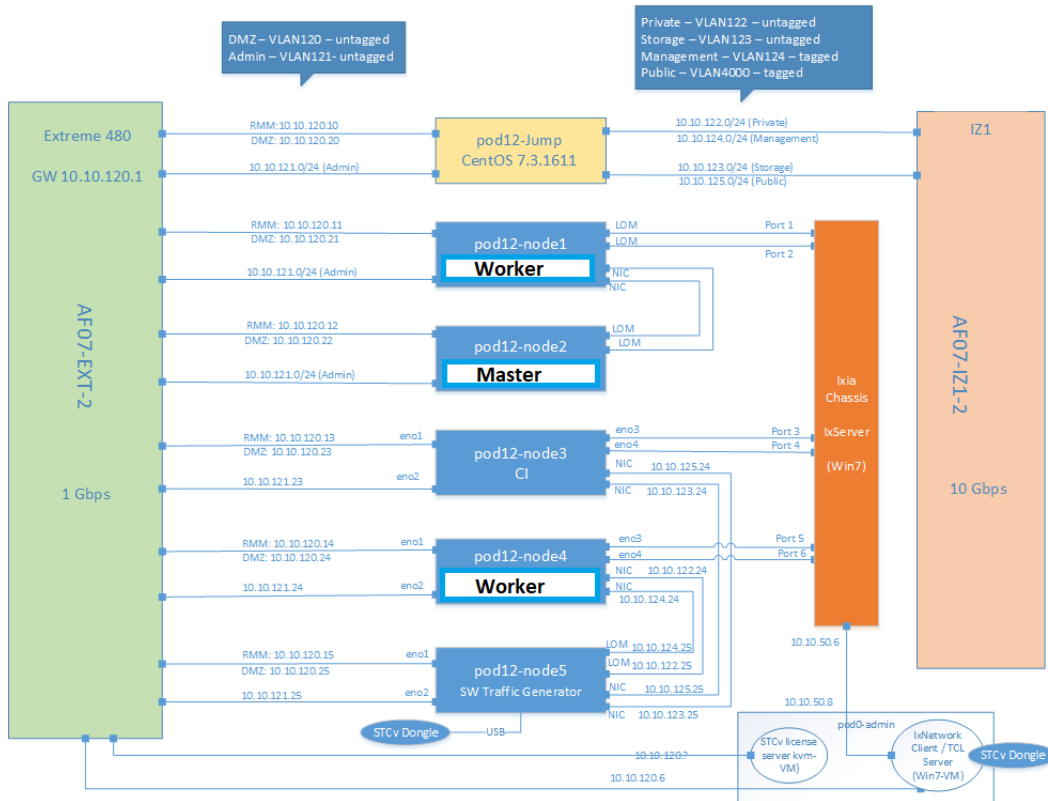


Kubernetes Cluster

- [Architecture](#)
- [Deployment Approach](#)
- [Deployment Manifests](#)
- [Access Details](#)
- [Features Supported](#)
- [Troubleshooting](#)

Architecture

We have 6 nodes in POD12. For Container Network Benchmarking we use node1 to run workloads for benchmarks and node2 to run virtual traffic generators. Configuration of datapath, running benchmarks, collecting results is automatically done by Vineperf tool running on node2.



Deployment Approach

Prerequisite before running Vineperf:

1. Kubernetes cluster
2. All required CNI plugins should be installed(Multus, Userspace, SRIOV, OVS)
3. Vineperf should be installed on node2(master node)

To meet 1, 2 we have ansible scripts to automatically deploy cluster and cni plugins on POD12, but they can work on any lab environment.

We have created Kubernetes cluster using pod12-node2(as master) and pod12-node1, pod12-node4(as worker) nodes. On this cluster, we have deployed few CNI plugins to meet our prerequisite. For Vineperf, we need k8s-pods with multiple interfaces to setup up separate data-path for benchmarking, but Kubernetes calls only 1 CNI plugin when it creates k8s-pods. So, to attach multiple interfaces to k8s-pod we use Multus CNI as our primary plugin and then we call other CNIs using Multus to attach extra interfaces to k8s-pod.

Some hosts network configurations are also needed to be done before running benchmarks. Vineperf automatically does all these configurations for us and sets up an end-to-end data-path, runs benchmarks, save results and then cleans changes made by it.

Currently, in the lab, we are using Flannel for default networking and benchmarking with the following CNIs:

1. Userspace-VPP
2. USerspace-OVS

3. Intel-SRIOV
4. Kubevirt-OVS

Deployment Manifests

location: <https://github.com/opnfv/vineperf/tree/master/tools/k8s/cluster-deployment/k8scluster>

step1:

```
git clone https://github.com/opnfv/vineperf
```

step2:

```
cd vineperf/tools/k8s/cluster-deployment/k8scluster/
```

step3: update host file with required connection details for ansible.

step4: to deploy cluster:

```
ansible-playbook k8sclustermanagement.yml -i hosts --tags "deploy"
```

step5: to deploy cni plugins on cluster:

```
ansible-playbook k8sclustermanagement.yml -i hosts --tags "cni"
```

For cleaning above setup:

```
ansible-playbook k8sclustermanagement.yml -i hosts --tags "clear"
```

Access Details

Login to Master Node, and run the following command

```
kubectl config view --raw
```

Features Supported

1. Multus
 - a. Userspace CNI
 - b. SRIOV
2. MetalLB Load Balancer
3. Kubevirt?

Troubleshooting