Copyright And Confidentiality

羅偉

牧云/罗伟
Network | IaaS | PaaS | ServiceMesh
交流 学习 沉淀 成长 分享
olaf.luo@foxmail.com
https://www.yuque.com/wei.luo
https://youdianzhishi.com

Rowan Luo

交流 学习 沉淀 成长 分享

Kubernetes Environment Preparation

Revision Date: 2023/02/23
Version：v1.5
DocID：CN00002023XLW

Kubernetes Environment Overview-VMWare Network Type

Whether you just bought yourself a copy of VMware Fusion (for your Mac), Workstation (for Windows or linux) or are using Player, Ace, or even the old VMware Server product, you'll be soon setting up your first virtual machine. Your new vm is going to want to talk to the outside world just like your physical machine, so it's a good idea to understand some basic options available to you to ensure your new vm works right out of the box. There are three types of networking available to virtual machines. Each type has its own uses, behaviors and features. They are as follows:

- ➢ Bridged networking
- ➢ Host-only networking
- ➢ Network Address Translation (NAT) networking

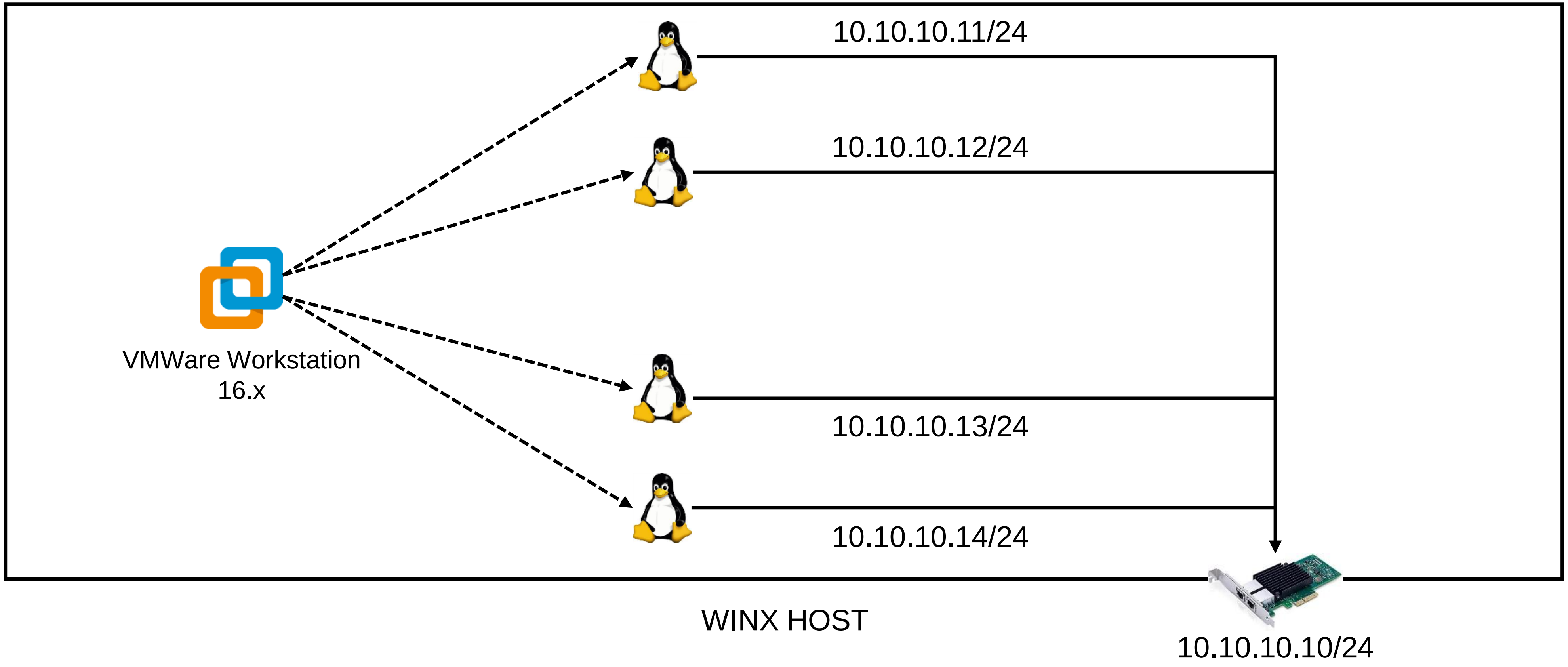
Using the wrong networking type or configuration settings may result in undesirable behavior, and frustration on your part, so lets understand these three variations and what they mean to you.

- ❑ Bridged networking connects a virtual machine to a network using the host computer's Ethernet adapter. If your host computer is on an Ethernet network ,this is often the easiest way to give your virtual machine access to that network. If you use bridged networking, your vm is a full participant in the network. It has access to other machines on the network and can be contacted by other machines on the network as if it were a physical computer on the network. This is a frequently used option.

- ❑ Host-only networking creates a network that is completely contained within the host computer. Host-only networking provides a network connection between the virtual machine and the host computer. In this setup, your vm will not have access to the outside world, only the physical machine your are running it on. This approach can be useful if you need to set up an isolated virtual network.

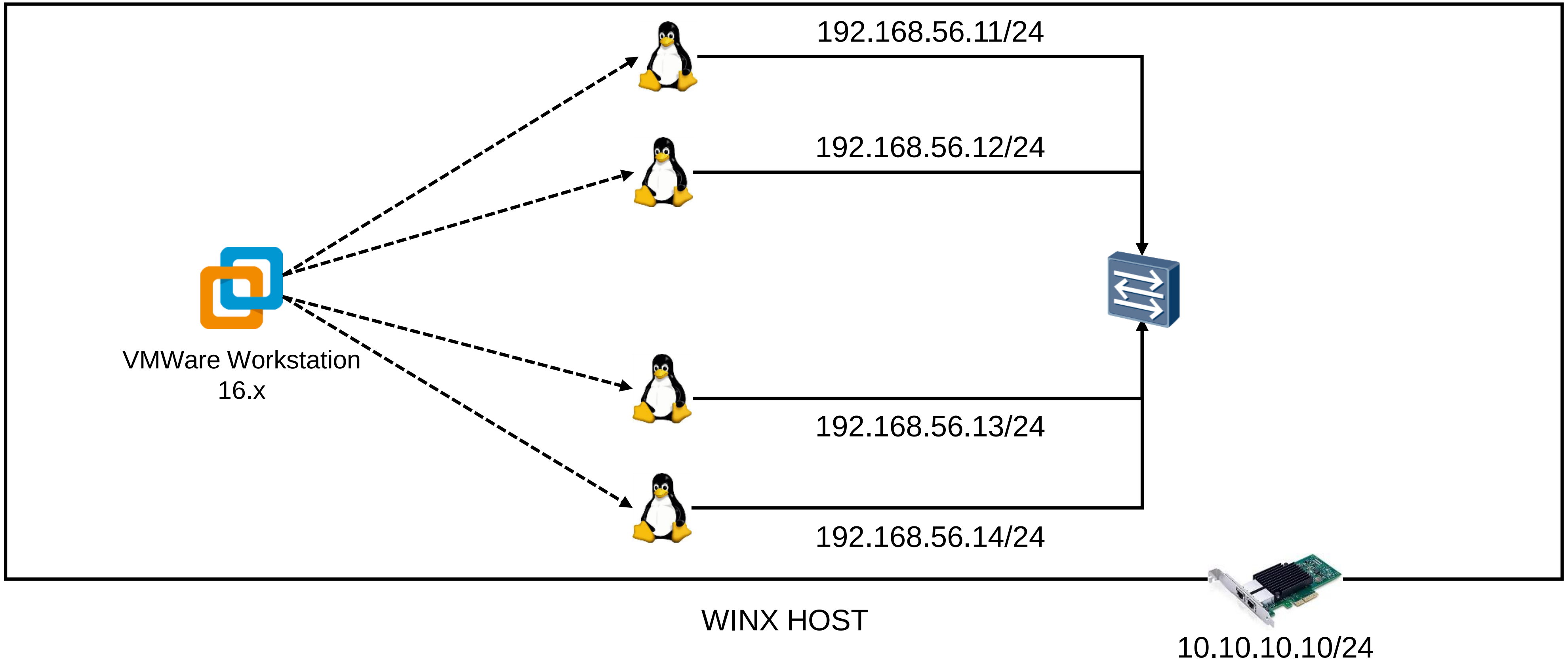
- ❑ Network Address Translation, or NAT for short, gives your virtual machine access to network resources using the host computer's IP address. How is this different than bridged you ask? If you use NAT, your virtual machine does not have its own IP address on the external network. Instead, a separate private network is set up on the host computer. This method might be best if your virtual machines do not provide services but still need to access a network.

Kubernetes Environment Overview-VMWare Network Type(Bridge Mode)



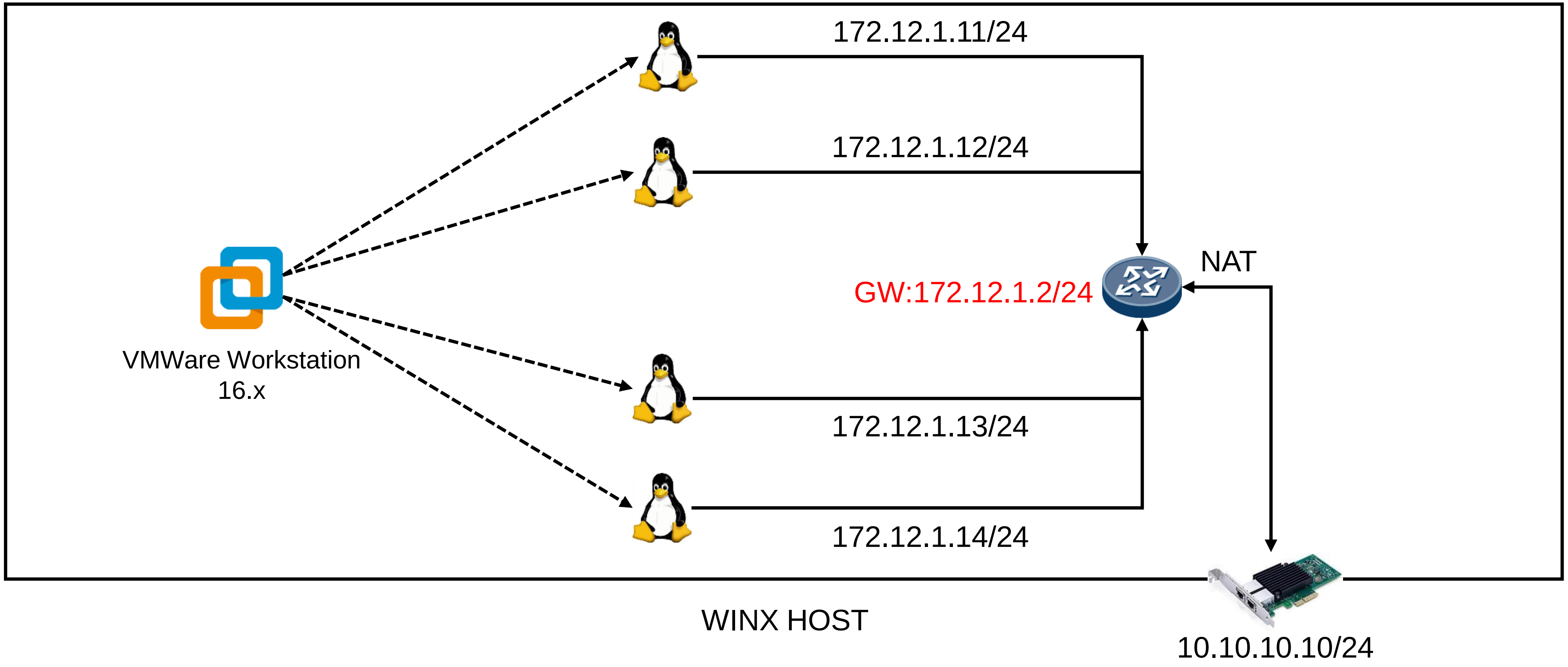
Bridged networking connects a virtual machine to a network using the host computer’s Ethernet adapter. If your host computer is on an Ethernet network ,this is often the easiest way to give your virtual machine access to that network. If you use bridged networking, your vm is a full participant in the network. It has access to other machines on the network and can be contacted by other machines on the network as if it were a physical computer on the network. This is a frequently used option.

Kubernetes Environment Overview-VMWare Network Type(HOST-Only Mode)



Host-only networking creates a network that is completely contained within the host computer. Host-only networking provides a network connection between the virtual machine and the host computer. In this setup, your vm will not have access to the outside world, only the physical machine your are running it on. This approach can be useful if you need to set up an isolated virtual network.

Kubernetes Environment Overview-VMWare Network Type(NAT Mode)



Network Address Translation, or NAT for short, gives your virtual machine access to network resources using the host computer’s IP address. How is this different than bridged you ask? If you use NAT, your virtual machine does not have its own IP address on the external network. Instead, a separate private network is set up on the host computer. This method might be best if your virtual machines do not provide services but still need to access a network.

Kubernetes Environment Overview-Ubuntu20.04 Kernel Upgrade

Upgrade to 5.11:

1. Access to the below URL:
https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.11/amd64/

2. Download *_all and *generic* file:
Like:

```
linux-headers-5.11.0-051100-generic_5.11.0-051100.202102142330_amd64.deb
linux-headers-5.11.0-051100_5.11.0-051100.202102142330_all.deb
linux-image-unsigned-5.11.0-051100-generic_5.11.0-051100.202102142330_amd64.deb
linux-modules-5.11.0-051100-generic_5.11.0-051100.202102142330_amd64.deb
```

3. Install the new kernel:

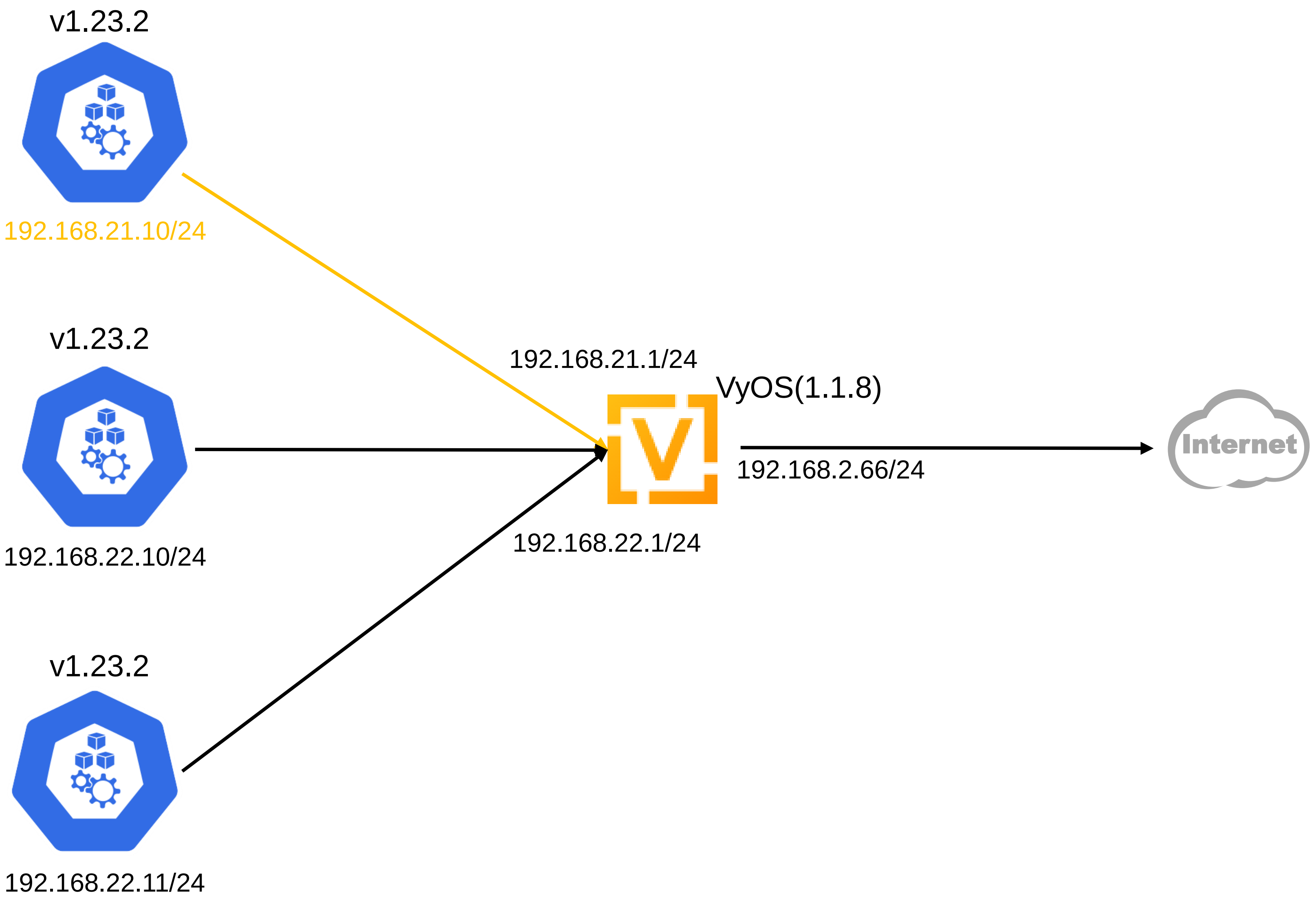
```
mkdir -p /root/new_kernel/ && cp -r *.deb /root/new_kernel/ && dpkg -i *.deb
```

4. Reboot the host and check the current kernel version:

```
uname -a
```

Cilium Feature	Minimum Kernel Version
IPv4 fragment handling	>= 4.10
Restrictions on unique prefix lengths for CIDR policy rules	>= 4.11
IPsec Transparent Encryption in tunneling mode	>= 4.19
WireGuard Transparent Encryption	>= 5.6
Host-Reachable Services	>= 4.19.57, >= 5.1.16, >= 5.2
Kubernetes Without kube-proxy	>= 4.19.57, >= 5.1.16, >= 5.2
Bandwidth Manager	>= 5.1
Local Redirect Policy (beta)	>= 4.19.57, >= 5.1.16, >= 5.2
Full support for Session Affinity	>= 5.7
BPF-based proxy redirection	>= 5.7
BPF-based host routing	>= 5.10
Socket-level LB bypass in pod netns	>= 5.7
Egress Gateway (beta)	>= 5.2
VXLAN Tunnel Endpoint (VTEP) Integration	>= 5.2

Kubernetes Environment Overview-Network TOPO



Kubernetes Environment Overview-Install Kubernetes(kubeadm)

```
cat <<EOF >> /etc/hosts
192.168.2.61 bpf1
192.168.2.62 bpf2
192.168.2.63 bpf3
EOF

apt install -y net-tools tcpdump chrony bridge-utils tree wget iftop ethtool curl

sed -ri 's/.*swap.*/#&/' /etc/fstab  &&  swapoff -a

apt-get update && apt-get install -y apt-transport-https
apt upgrade -y
curl https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | apt-key add -
tee /etc/apt/sources.list.d/kubernetes.list <<-'EOF'
deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes-xenial main
EOF
apt-get update -y

apt install docker-ce
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt install docker-ce docker-ce-cli containerd.io
mkdir -p /etc/docker
cat <<EOF > /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "registry-mirrors": ["https://cu2yw19m.mirror.aliyuncs.com"]
}
EOF
systemctl daemon-reload
systemctl restart docker
systemctl enable docker
```

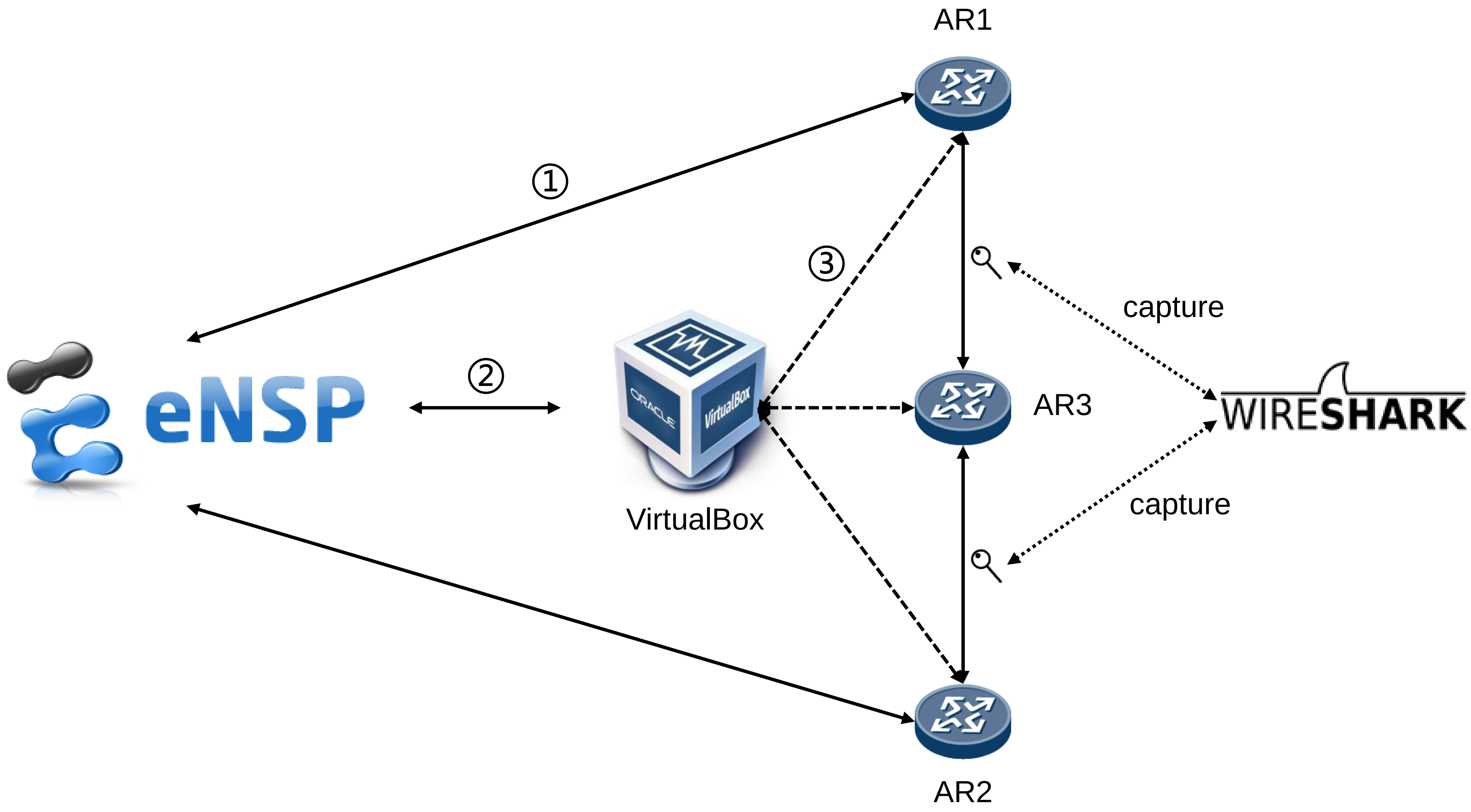
Kubernetes Environment Overview-Install Kubernetes

```
apt-get install -y kubelet kubeadm kubectl --allow-unauthenticated
systemctl enable kubelet && systemctl restart kubelet

cat > /var/lib/kubelet/config.yaml <<EOF      # [master node]
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
cgroupDriver: systemd
EOF
systemctl daemon-reload
systemctl restart docker
systemctl enable kubelet
systemctl restart kubelet

docker info|grep "Cgroup Driver"
 Cgroup Driver: systemd

# Master Node:
kubeadm config images pull --image-repository=registry.aliyuncs.com/google_containers
kubeadm init --kubernetes-version=v1.23.5 --image-repository registry.aliyuncs.com/google_containers --pod-network-cidr=10.244.0.0/16 --
service-cidr=10.96.0.0/12 --skip-phases=addon/kube-proxy --ignore-preflight-errors=Swap

# Worker Node:
kubeadm config images pull --image-repository=registry.aliyuncs.com/google_containers
kubeadm join 192.168.2.61:6443 --token ac4k64.e3i6j13sryj1twzt \
        --discovery-token-ca-cert-hash sha256:7feb5f701bbad147116daddda3e74e720738e61938eedccc7bfaa3d24aed23bf

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

kubectl taint nodes --all node-role.kubernetes.io/master-
kubectl apply -f https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml
kubectl run nettoolbox --image=burlyluo/nettoolbox --restart=Never
```

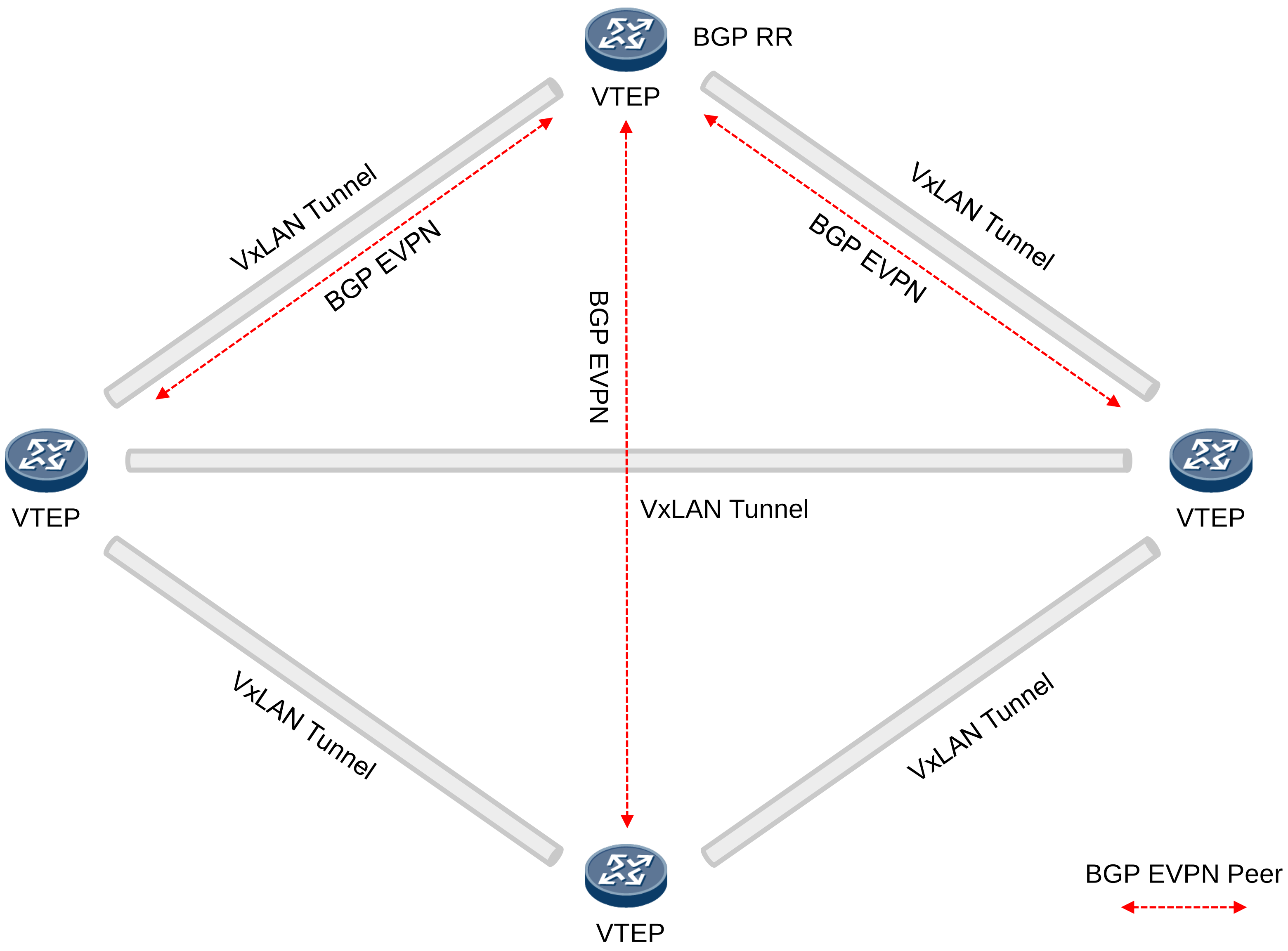
Kubernetes Environment Overview-Install Kubernetes(kind)

```
1.Download kind:
curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.14.0/kind-linux-amd64
chmod +x ./kind
mv ./kind /usr/bin/kind

2.Install Kubernetes cluster:
root@kind:~/kind/calico# kind create cluster --name=kind-k8s --
image=kindest/node:v1.23.6@sha256:b1fa224cc6c7ff32455e0b1fd9cbfd3d3bc87ecaa8fcb06961ed1afb3db0f9ae --config=kind-calico.yaml
Creating cluster "kind-k8s" ...
 ✓ Ensuring node image (kindest/node:v1.23.6)
 ✓ Preparing nodes
 ✓ Writing configuration
 ✓ Starting control-plane
 ✓ Installing StorageClass
 ✓ Joining worker nodes
Set kubectl context to "kind-kind-k8s"
You can now use your cluster with:
kubectl cluster-info --context kind-kind-k8s
Thanks for using kind!

root@kind:~/kind/calico# cat kind-calico.yaml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
networking:
        disableDefaultCNI: true
        kubeProxyMode: "ipvs"
nodes:
        - role: control-plane
        - role: worker
        - role: worker
root@kind:~/kind/calico#
root@kind:~/kind/calico# kubectl cluster-info --context kind-kind-k8s
root@kind:~/kind/calico# kubectl get pods -owide -A
```

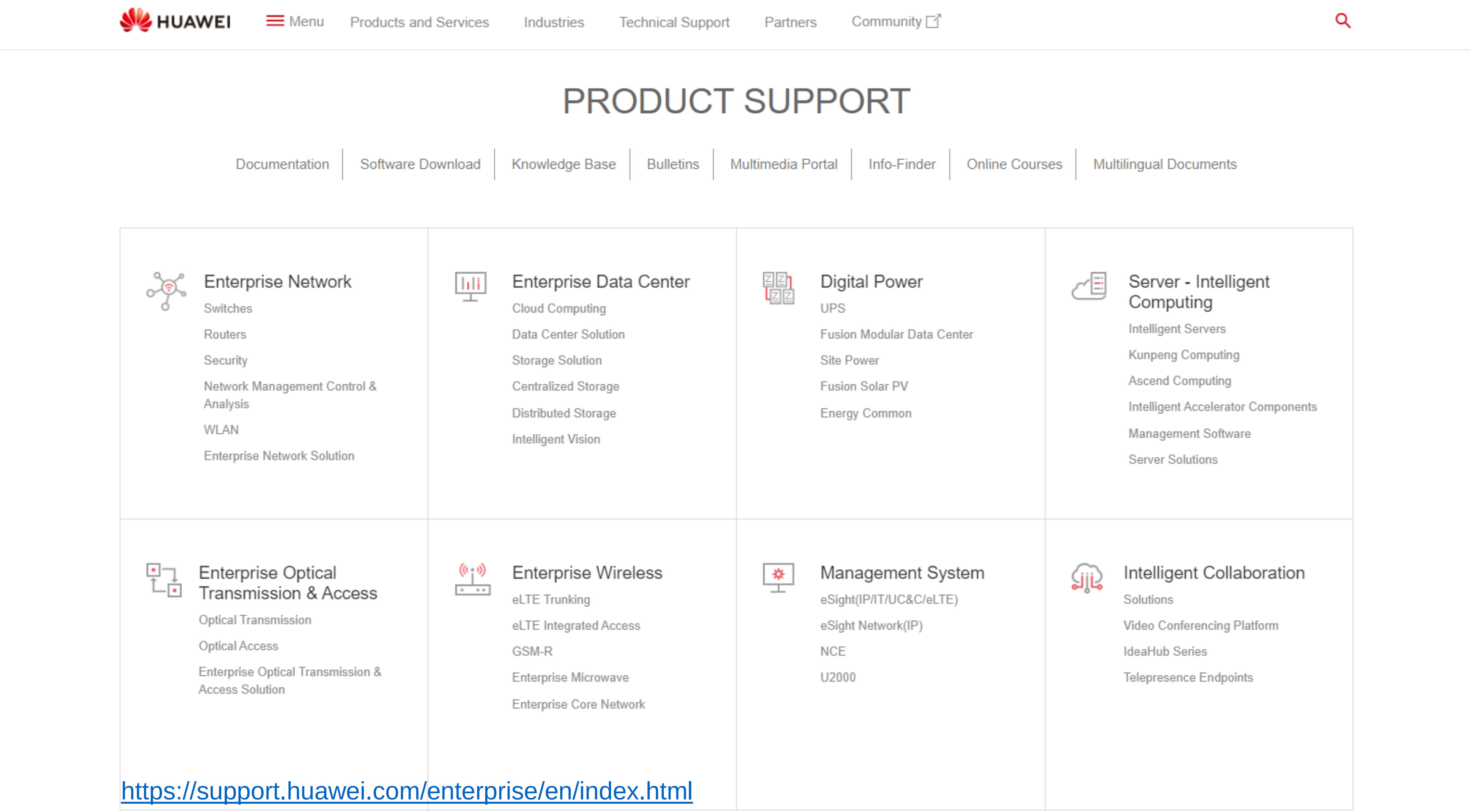
Kubernetes Environment Overview-eNSP



Kubernetes Environment Overview-eNSP Demo



Kubernetes Environment Overview-Docs Reference



HUAWEI ☰ Menu Products and Services Industries Technical Support Partners Community

PRODUCT SUPPORT

Documentation | Software Download | Knowledge Base | Bulletins | Multimedia Portal | Info-Finder | Online Courses | Multilingual Documents

Enterprise Network

- Switches
- Routers
- Security
- Network Management Control & Analysis
- WLAN
- Enterprise Network Solution

Enterprise Data Center

- Cloud Computing
- Data Center Solution
- Storage Solution
- Centralized Storage
- Distributed Storage
- Intelligent Vision

Digital Power

- UPS
- Fusion Modular Data Center
- Site Power
- Fusion Solar PV
- Energy Common

Server - Intelligent Computing

- Intelligent Servers
- Kunpeng Computing
- Ascend Computing
- Intelligent Accelerator Components
- Management Software
- Server Solutions

Enterprise Optical Transmission & Access

- Optical Transmission
- Optical Access
- Enterprise Optical Transmission & Access Solution

Enterprise Wireless

- eLTE Trunking
- eLTE Integrated Access
- GSM-R
- Enterprise Microwave
- Enterprise Core Network

Management System

- eSight(IP/IT/UC&C/eLTE)
- eSight Network(IP)
- NCE
- U2000

Intelligent Collaboration

- Solutions
- Video Conferencing Platform
- IdeaHub Series
- Telepresence Endpoints

https://support.huawei.com/enterprise/en/index.html

Copyright And Confidentiality

羅 偉

牧云/罗伟
Network | IaaS | PaaS | ServiceMesh
交流 学习 沉淀 成长 分享
olaf.luo@foxmail.com
https://www.yuque.com/wei.luo
https://youdianzhishi.com