F5 BIG-IP Next for Kubernetes on Nvidia BlueField-3 DPU

Lab Guide

Quang Nguyen, Wael Shahen

Table of contents

1. F5 Titan BIG-IP Next for Kubernetes Install Instructions on Nvidia BlueField-3

1.1 Overview

This guide will help you setup and install F5 BIG-IP Next for Kubernetes (BIG-IP Next for Kubernetes) on a platform with an Nvidia BlueField-3 DPU.

The NVIDIA DOCA Framework enables rapidly creating and managing applications and services on top of the BlueField networking platform, leveraging industry-standard APIs. For more information please refer to DOCA Documentation.

1.2 BIG-IP Next for Kubernetes Overview

BIG-IP Next for Kubernetes consists of two primary components:

1. **Data Plane**: Handling traffic processing and rules.
2. **Control Plane**: Monitors the Kubernetes cluster state and dynamically updates the Data Plane components.

1.2.1 Data Plane (TMM)

At the heart of Data Plane is the Traffic Management Microkernel (TMM). Which is responsible for processing network traffic entering and leaving the Kubernetes cluster, as well as integrating with the infrastructure beyond the cluster. The TMM and it's supporting components are deployed on the Nvidia BlueField-3 (BF3) DPU, fully utilizing its resources and offload engine, and freeing the CPU resources on the host for other tasks.

1.2.2 Control Plane

The Control Plane runs on the Host CPU worker node or generic workload worker nodes. It also acts as a controller for Kubernetes Gateway API

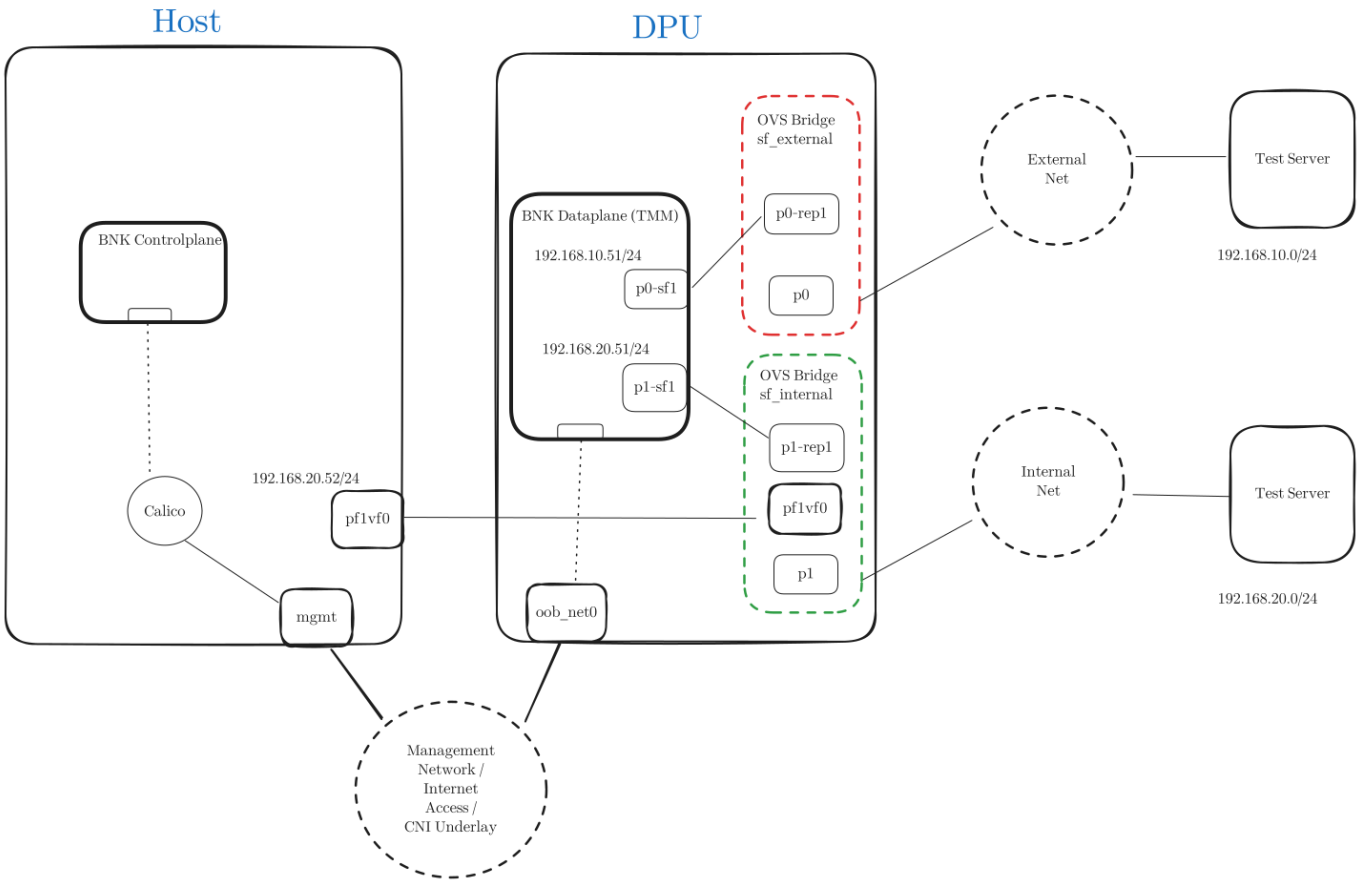
1.3 Lab Setup

see prerequisites The following section describes implementation details for a lab setup.

1.3.1 Deployment Strategy

For the purpose of this document, the diagram below illustrates a high-level deployment strategy for BIG-IP Next for Kubernetes on Nvidia BlueField-3 DPU. It assumes a specific Nvidia BlueField-3 networking configuration, utilizing Scalable Functions, Virtual Functions, and Open vSwitch (OVS) to connect the DPU, Host, and external uplink ports.

This lab guide configures a single Kubernetes cluster that includes Hosts and DPUs as worker nodes. It assumes that one of the hosts will act as a Kuberentes controller (and allows workload deployment) while other hosts and DPUs join the cluster as worker nodes.



There are three main networks in the diagram:

Management Network: The main underlay network for the Kubernetes cluster CNI and has the default gateway to reach internet. Both Host and the Nvidia BF-3 DPU are connected to this network and has addresses configured through DHCP.

Internal Network: Represents an internal network path between the host deployed services and the BNK Dataplane deployed in the DPU. This network will be utilized to route ingress and egress traffic for workload deployed on the host through BNK Dataplane.

External Network: The external network represents an "external-to-the-cluster" infrastructure network segment to reach external services/destinations.

The Test Servers represent clients and servers that are reachable on different segments of the network.

This could also be a single server connected to both Internal and External networks

2025-02-25

2. Prerequisites

2.1 Software

This lab guide will walk you through setup of Kubernetes cluster using kubeadm. The guide assumes that you have Ubuntu 22.04 installed on the host machine and the Nvidia BlueField-3 is running in the default DPU mode, and uplink port links set to `ETH`.

> **Note**
>
> The following table is provided as guidance if software installation is prefered outside of this guide.

Software	Version	Node/ Selector	Installed in this Guide	Reference
DOCA	2.8+	Host	**Yes**	NVIDIA DOCA Installation Guide for Linux
BF Bundle BFB	2.8+	DPU	**Yes**	Nvidia DOCA Downloads
Kubelet	1.29+	Host and DPU	**Yes**	Kubernetes Kubeadm guide
Kubeadm	1.29+	Host and DPU	**Yes**	
Kubectl	1.29+	Host and DPU	**Yes**	
Containerd	1.7.22+	Host and DPU	**Yes**	Containerd Getting Started
cert-manager	1.16.1+	Host and DPU	**Yes**	Cert-manager installation
SR-IOV Device Plugin	3.7.0+	DPU	**Yes**	SR-IOV Device Plugin
Multus	4.1.0+	Host and DPU	**Yes**	Multus quick install
Calico	3.28.1+	Host and DPU	**Yes**	Calico

2.2 Hardware

This lab guide was tested on the following hardware configurations:

> **Note**
>
> The hardware list below serves as example based on tested platforms. Only one of those or any other Nvidia DPU-3 compatible system is required for this guide.

Vendor	Model	CPU Architecture	# of Cores	RAM	Storage
Dell	Poweredge R750	x86_64	96	512 GB	21 TB
Supermicro	LB26-R16R12	aarch64	96	512 GB	20 TB
Supermicro	HGX AS-4125GS-TNRT	x86_64	128	768 GB	12 TB
Supermicro	MGX ARS-111GL-NHR	aarch64	72	512 GB	1.5 TB

2.2.1 AUX Cable

HGX : Part Numbers - CBL-PWEX-1040 and CBL-PWEX-1148-20

MGX : Part Number - CBL-PWEX-1040

Dell : Amazon.com: BestParts New 12Pin to 8+8 Pin GPU Power Cable Compatible with Dell PowerEdge R750 R750XS R7525 Server 16inches DPHJ8 : Electronics

2.2.2 Network Optics

The following network optics were tested on the DPU ports.

MGX & HGX : 200Gb SR4 Ethernet Only - NVIDIA Ethernet MMA1T00-VS Compatible QSFP56 200GBASE-SR4 850nm 100m DOM MPO12/UPC MMF Optical Transceiver Module, Support 4 x 50G-SR - FS.com

Dell R750 : F5 Networks F5-UPG-QSFP28-SR4 Compatible QSFP28 100GBASE-SR4 850nm \ 100m DOM MPO-12/UPC MMF Optical Transceiver Module, Support 4 x 25G-SR - FS.com

2.2.3 GPU (Optional)

HGX = Nvidia H100 (x86)

MGX = NVIDIA GH200 (arm64)

2.2.4 DPU

Model : B3220 Single-Slot FHHL w/ Crypto enabled

NVIDIA OPN : 900-9D3B6-00CV-AA0

PSID : MT_0000000884

2025-02-25

3. 2. Kubernetes Setup

Install Host Software

Create a directory for example `dpu-install` to prepare for installation.

Download the install-host.sh and modify the following default variables

Show content of install-host.sh

Bash

```
#!/bin/bash

set -euo pipefail

DEBUG=0

if [[ ${DEBUG} -eq 1 ]]; then
  set -x
fi

# defaults

# Change the MGMT_NET variable to the management network CIDR
# that will include both the host mgmt IP and DPU oob_net0 mgmt IP.
MGMT_NET="10.144.0.0/16"

# Change this variable to point to the correct PF1 interface
# name on the host.
PF_INT=enp83s0f1np1
VF_INT=${PF_INT/%np1/v0}

DOCA_VERSION=2.9.1
K8S_VERSION="1.29"
CONTAINERD_VERSION="1.7.23"
RUNC_VERSION="1.2.1"


arch=$(uname -m)
case "$arch" in
x86_64)
    ARCH="amd64"
    ;;
aarch64)
    ARCH="arm64"
    ;;
*)
    echo "Unsupported system architecture: $arch"
    exit 1
    ;;
esac


install_doca_all() {
  for f in $( dpkg --list | awk '/doca/ {print $2}' ); do
    echo "Uninstalling package $f"
    apt remove --purge "$f" -y || true
  done
  /usr/sbin/ofed_uninstall.sh --force || true
  apt-get -y autoremove
  DOCA_URL="https://linux.mellanox.com/public/repo/doca/2.9.1/ubuntu22.04/$arch/"
  curl https://linux.mellanox.com/public/repo/doca/GPG-KEY-Mellanox.pub | gpg --yes --dearmor > /etc/apt/trusted.gpg.d/GPG-KEY-Mellanox.pub
  echo "deb [signed-by=/etc/apt/trusted.gpg.d/GPG-KEY-Mellanox.pub] $DOCA_URL ./" > /etc/apt/sources.list.d/doca.list
  apt-get update
  apt-get -y install rshim
  systemctl enable rshim --now
  cat << EONETPLAN > /etc/netplan/50-tmfifo.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    tmfifo_net0:
      dhcp4: no
      addresses:
        - 192.168.100.1/30
EONETPLAN
  chmod 600 /etc/netplan/50-tmfifo.yaml
  netplan apply
  sleep 5
}

configure_virtual_function() {
  # TODO: add script to automatically discover PFs and adds a virtual
  # function to pf1.
  cat << EOFVFCONF > /etc/netplan/10-vf-config.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    $PF_INT:
      dhcp4: no
      virtual-function-count: 1
    $VF_INT:
      link: $PF_INT
      dhcp4: no
      addresses:
        - 192.168.20.41/24
EOFVFCONF
  chmod 600 /etc/netplan/10-vf-config.yaml
  netplan apply
  sleep 5
}

install_runc() {
```

```
    curl -LO https://github.com/opencontainers/runc/releases/download/v$RUNC_VERSION/runc.$ARCH
    install -m 755 runc.$ARCH /usr/local/sbin/runc

}

install_containerd() {
    mkdir -p /etc/containerd
    curl -LO https://github.com/containerd/containerd/releases/download/v$CONTAINERD_VERSION/containerd-$CONTAINERD_VERSION-linux-$ARCH.tar.gz

    tar Czxvf /usr/local/ containerd-$CONTAINERD_VERSION-linux-$ARCH.tar.gz

    /usr/local/bin/ctr oci spec > /etc/containerd/cri-base.json
    cat << EOL > /etc/containerd/config.toml
version = 2
root = "/var/lib/containerd"
state = "/run/containerd"
oom_score = 0
[grpc]
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
[debug]
  address = ""
  level = "info"
  format = ""
  uid = 0
  gid = 0
[plugins]
  [plugins."io.containerd.grpc.v1.cri"]
    sandbox_image = "registry.k8s.io/pause:3.10"
    max_container_log_line_size = 16384
    enable_unprivileged_ports = false
    enable_unprivileged_icmp = false
    enable_selinux = false
    disable_apparmor = false
    tolerate_missing_hugetlb_controller = true
    disable_hugetlb_controller = true
    image_pull_progress_timeout = "5m"
    [plugins."io.containerd.grpc.v1.cri".containerd]
      default_runtime_name = "runc"
      snapshotter = "overlayfs"
      discard_unpacked_layers = true
      [plugins."io.containerd.grpc.v1.cri".containerd.runtimes]
        [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc]
          runtime_type = "io.containerd.runc.v2"
          runtime_engine = ""
          runtime_root = ""
          base_runtime_spec = "/etc/containerd/cri-base.json"
          [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
            systemdCgroup = true
            binaryName = "/usr/local/sbin/runc"
EOL

    curl -L -o /etc/systemd/system/containerd.service https://raw.githubusercontent.com/containerd/containerd/main/containerd.service
    systemctl daemon-reload
    systemctl enable --now containerd
}

install_kubernetes_components() {
    apt-get update && apt-get install -y apt-transport-https ca-certificates curl gpg
    mkdir -p /etc/apt/keyrings
    curl -fsSL https://pkgs.k8s.io/core:/stable:/v$K8S_VERSION/deb/Release.key | gpg --yes --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
    echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v$K8S_VERSION/deb/ /" | tee /etc/apt/sources.list.d/
kubernetes.list
    cat << EOL > /etc/sysctl.d/kubernetes.conf
net.bridge.bridge-nf-call-ip6tables=1
net.bridge.bridge-nf-call-iptables=1
net.ipv4.ip_forward=1
net.ipv6.conf.default.forwarding=1
fs.inotify.max_user_watches=2099999999
fs.inotify.max_user_instances=2099999999
fs.inotify.max_queued_events=2099999999
EOL
    sysctl --system
    echo "br_netfilter" > /etc/modules-load.d/br_netfilter.conf
    modprobe br_netfilter
    swapoff -a
    sed -i.backup '/swap/d' /etc/fstab
    apt-get update
    apt-get install -y kubelet kubeadm kubectl
    apt-mark hold kubelet kubeadm kubectl
    systemctl enable --now kubelet
}


init_kubernetes() {
    kubeadm init --pod-network-cidr=10.244.0.0/16
    mkdir -p $HOME/.kube
    cp -f /etc/kubernetes/admin.conf $HOME/.kube/config
    kubectl get node
    echo "Installing Calico CNI ..."
    kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.29.1/manifests/tigera-operator.yaml
    cat << EOFCALICO | kubectl apply -f -
---
apiVersion: operator.tigera.io/v1
kind: Installation
metadata:
  name: default
spec:
  calicoNetwork:
    ipPools:
```

```
    - name: default-ipv4-ippool
      blockSize: 26
      cidr: 10.244.0.0/16
      encapsulation: VXLANCrossSubnet
      natOutgoing: Enabled
      nodeSelector: all()
    bgp: Disabled
    nodeAddressAutodetectionV4:
      cidrs:
      - "$MGMT_NET"
---
apiVersion: operator.tigera.io/v1
kind: APIServer
metadata:
  name: default
spec: {}
EOFCALICO
    # Wait for Calico system to start installation and create the calico-system namespace.
    sleep 30
    kubectl wait --for=condition=Ready pods --all --all-namespaces --timeout=300s
    kubectl taint nodes --all node-role.kubernetes.io/control-plane- || true
    kubectl get pod --all-namespaces
    echo "Adding node annotation for internal static route"
    for node in $(kubectl get node -o name); do
      kubectl annotate --overwrite $node 'k8s.ovn.org/node-primary-ifaddr={"ipv4":"192.168.20.41"}'
    done
    kubectl apply -f https://raw.githubusercontent.com/k8snetworkplumbingwg/multus-cni/master/deployments/multus-daemonset-thick.yml
    kubectl wait --for=condition=Ready pods --all --all-namespaces --timeout=300s
    cat << 'EOSRIOVCONF' | kubectl apply -f -
apiVersion: v1
kind: ConfigMap
metadata:
  name: sriovdp-config
  namespace: kube-system
data:
  config.json: |
    {
        "resourceList": [
            {
                 "resourceName": "bf3_p0_sf",
                  "resourcePrefix": "nvidia.com",
                  "deviceType": "auxNetDevice",
                  "selectors": [{
                      "vendors": ["15b3"],
                      "devices": ["a2dc"],
                      "pciAddresses": ["0000:03:00.0"],
                      "pfNames": ["p0#1"],
                      "auxTypes": ["sf"]
                  }]
              },
              {
                 "resourceName": "bf3_p1_sf",
                  "resourcePrefix": "nvidia.com",
                  "deviceType": "auxNetDevice",
                  "selectors": [{
                      "vendors": ["15b3"],
                      "devices": ["a2dc"],
                      "pciAddresses": ["0000:03:00.1"],
                      "pfNames": ["p1#1"],
                      "auxTypes": ["sf"]
                  }]
              }
        ]
    }
EOSRIOVCONF
    kubectl apply -f https://raw.github.com/k8snetworkplumbingwg/sriov-network-device-plugin/master/deployments/sriovdp-daemonset.yaml
    kubectl patch daemonset kube-sriov-device-plugin -n kube-system --type='json' -p='[{"op": "add", "path": "/spec/template/spec/tolerations", "value":
[{"effect": "NoSchedule", "operator": "Exists"}]}]'

    helm repo add jetstack https://charts.jetstack.io --force-update
    helm install cert-manager jetstack/cert-manager --namespace cert-manager --create-namespace --version v1.16.1 --set crds.enabled=true --set
featureGates=ServerSideApply=true
    cat << 'EOFCERTMGRCONF' | kubectl apply -f -
---
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
    name: selfsigned-cluster-issuer
spec:
    selfSigned: {}
---
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
    name: bnk-ca
    namespace: cert-manager
spec:
    isCA: true
    commonName: bnk-ca
    secretName: bnk-ca
    issuerRef:
        name: selfsigned-cluster-issuer
        kind: ClusterIssuer
        group: cert-manager.io
---
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
    name: bnk-ca-cluster-issuer
spec:
```

```
  ca:
    secretName: bnk-ca
EOFCERTMGRCONF

    kubectl apply -f https://github.com/kubernetes-sigs/gateway-api/releases/download/v1.2.0/experimental-install.yaml
    kubectl wait --for=condition=Ready pods --all --all-namespaces --timeout=300s

}


export DEBIAN_FRONTEND=noninteractive

trap 'unset DEBIAN_FRONTEND' ERR EXIT

# 1. Install DOCA software
install_doca_all

# 2. Install runc
install_runc

# 3. Install containerd
install_containerd

# 4. Install and init Kubernetes
install_kubernetes_components

# 5. Init Kubernetes Controller node and install required services.
init_kubernetes

# 6. Configure virtual function on PF1
configure_virtual_function

echo "======================"
echo "Installation complete."

unset DEBIAN_FRONTEND
```

Variable	Description	Default
`MGMT_NET`	Management Network CIDR for host and DPU	`10.144.0.0/16`
`PF_INT`	Host PF 1 netdev name. This is the port connected to **Internal** network	enp83s0f1np1

Note

Only use PF 1 for the variable `PF_INT`. Do not use `np0`.

Then run the script on the host machine.

Host Software Installation

```
host# chmod +x install-host.sh && ./install-host.sh
```

Note

The script initalizes Kubernetes cluster also using `kubeadm init` it should only run on Controller node.

2025-02-25

4. Install DPU Software

In the same directory `dpu-install` created in previous step.

4.1 1. Download BF Bundle

The BlueField bundle includes Operating System, Drivers, and DPU software tools. Nvidia DOCA download

4.2 2. Create bf config

The dpu-config.sh script will produce a BlueField install config file.

Show content of dpu-config.sh

Bash

```
#!/bin/bash -x

generate_bluefield_config() {
    bf_conf_template=$(cat << 'EOFBFTEMPLATE'
# UPDATE_DPU_OS - Update/Install BlueField Operating System (Default: yes)
UPDATE_DPU_OS="yes"

# ubuntu_PASSWORD - Hashed password to be set for "ubuntu" user during BFB installation process.
# Relevant for Ubuntu BFB only. (Default: is not set)
ubuntu_PASSWORD='{{PASSWORD}}'

################################################################################
# Other misc configuration
################################################################################

# MAC address of the rshim network interface (tmfifo_net0).
NET_RSHIM_MAC={{NET_RSHIM_MAC}}

# bfb_modify_os – SHELL function called after the file system is extracted on the target partitions.
# It can be used to modify files or create new files on the target file system mounted under
# /mnt. So the file path should look as follows: /mnt/<expected_path_on_target_OS>. This
# can be used to run a specific tool from the target OS (remember to add /mnt to the path for
# the tool).

bfb_modify_os()
{
    # Set hostname
    local hname="{{HOSTNAME}}"
    echo ${hname} > /mnt/etc/hostname
    echo "127.0.0.1 ${hname}" >> /mnt/etc/hosts

    # Overwrite the tmfifo_net0 interface to set correct IP address
    # This is relevant in case of multiple DPU system.
    cat << EOFNET > /mnt/var/lib/cloud/seed/nocloud-net/network-config
version: 2
renderer: NetworkManager
ethernets:
  tmfifo_net0:
    dhcp4: false
    addresses:
      - {{IP_ADDRESS}}/{{IP_MASK}}
  oob_net0:
    dhcp4: true
EOFNET

    # Modules for kubernetes and DPDK
    cat << EOFMODULES >> /mnt/etc/modules-load.d/custom.conf
overlay
br_netfilter
vfio_pci
EOFMODULES

    # sysctl settings for kubernets
    cat << EOFSYSCTL >> /mnt/etc/sysctl.d/kubernetes.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOFSYSCTL

    # Provision hugepages as part of grub boot
    # Default to 2M hugepage size and provision 24.5 GB of hugepages
    # TMM requires 1.5GB of hugepages per thread (CPU core) totaling
    # 24GB to run on all 16 threads of the DPU.
    local hpage_grub="default_hugepagesz=2MB hugepagesz=2M hugepages=12544"
    sed -i -E "s|^(GRUB_CMDLINE_LINUX_DEFAULT=\")(.*)\"|\1${hpage_grub}\"|" /mnt/etc/default/grub
    ilog "$(chroot /mnt env PATH=$PATH /usr/sbin/grub-mkconfig -o /boot/grub/grub.cfg)"

    # Provision SF to be used by the TMM on each PF
    # First clear out the current configurations for default SFs
    # These default SFs do not have trust mode set to.
    : > /mnt/etc/mellanox/mlnx-sf.conf
    # Then add new SFs with trust mode enabled.
    for pciid in $(lspci -nD 2> /dev/null | grep 15b3:a2d[26c] | awk '{print $1}')
        do
            cat << EOFSF >> /mnt/etc/mellanox/mlnx-sf.conf
/sbin/mlnx-sf --action create --enable-trust --device $pciid --sfnum 0 --hwaddr $(uuidgen | sed -e 's/-//;s/^\(..\)\(..\)\(..\)\(..\)\(..\).*$/02:\1:\2:\3:\4:\5/')
/sbin/mlnx-sf --action create --enable-trust --device $pciid --sfnum 1 --hwaddr $(uuidgen | sed -e 's/-//;s/^\(..\)\(..\)\(..\)\(..\)\(..\).*$/02:\1:\2:\3:\4:\5/')
EOFSF
        done
    # OVS changes
    # 1. Change bridge names to follow internal document as sf_external for pf0
    #    and sf_internal for pf1.
    sed -i -E "s|^(OVS_BRIDGE1=\")(.*)\"|\1sf_external\"|" /mnt/etc/mellanox/mlnx-ovs.conf
    sed -i -E "s|^(OVS_BRIDGE2=\")(.*)\"|\1sf_internal\"|" /mnt/etc/mellanox/mlnx-ovs.conf
    # 2. Add the new created SFs, "sfnum 1" to their corresponding bridges.
    #    Also include the virtual functions that are going to be created on host.
    #    These vfs may not exist yet.
    sed -i -E 's|^(OVS_BRIDGE1_PORTS=")[^"]*("|\1p0 en3f0pf0sf1\2|' /mnt/etc/mellanox/mlnx-ovs.conf
    sed -i -E 's|^(OVS_BRIDGE2_PORTS=")[^"]*("|\1p1 en3f1pf1sf1 pf1vf0\2|' /mnt/etc/mellanox/mlnx-ovs.conf

    # Cloud-init for upgrading containerd and runc
    cat << EOFCLOUDINIT >> /mnt/var/lib/cloud/seed/nocloud-net/user-data
write_files:
```

```
  - path: /etc/containerd/config.toml
    content: |
      version = 2
      root = "/var/lib/containerd"
      state = "/run/containerd"
      oom_score = 0
      [grpc]
        max_recv_message_size = 16777216
        max_send_message_size = 16777216
      [debug]
        address = ""
        level = "info"
        format = ""
        uid = 0
        gid = 0
      [plugins]
        [plugins."io.containerd.grpc.v1.cri"]
          sandbox_image = "registry.k8s.io/pause:3.10"
          max_container_log_line_size = 16384
          enable_unprivileged_ports = false
          enable_unprivileged_icmp = false
          enable_selinux = false
          disable_apparmor = false
          tolerate_missing_hugetlb_controller = true
          disable_hugetlb_controller = true
          image_pull_progress_timeout = "5m"
          [plugins."io.containerd.grpc.v1.cri".containerd]
            default_runtime_name = "runc"
            snapshotter = "overlayfs"
            discard_unpacked_layers = true
            [plugins."io.containerd.grpc.v1.cri".containerd.runtimes]
              [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc]
                runtime_type = "io.containerd.runc.v2"
                runtime_engine = ""
                runtime_root = ""
                base_runtime_spec = "/etc/containerd/cri-base.json"
                [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
                  systemdCgroup = true
                  binaryName = "/usr/local/sbin/runc"
  - path: /var/tmp/setup-script.sh
    permissions: '0755'
    encoding: base64
    content: |

IyEvYmluL2Jhc2gKClRNUERJUj0kKG1rdGVtcCAtZCkKL3Vzci9zYmluL250cHdhaXQgLXYKc3lzdGVtY3RsIHN0b3AgY29udGFpbmVyZCBrdWJlbGV0IGt1YmVwb2RzLnNsaWNlCnJtIC1yZiAvdmFyL2xpYi9jb250
0YWluZXJkLyoKcm0gLXJmIC9ydW4vY29udGFpbmVyZC8qCnJtIC1mIC91c3IvbGliL3N5c3RlbWQvc3lzdGVtL2t1YmVsZXQuc2VydmljZS5kLzkwLWt1YmVsZXQtYmx1ZWZpZWxkLmNvbmYKYXB0IC15IHB1cmdlIG
t1YmVsZXQga3ViZWFkbSB8fCB0cnVlCmN1cmwgLS1vdXRwdXQtZGlyICR7VE1QRElSfSAtTE8gaHR0cHM6Ly9naXRodWIuY29tL2NvbnRhaW5lcmQvY29udGFpbmVyZC8lMTyBodHRwczovL2dpdGh1Yi5jb20vY29udGFp
bmVyZC9jb250YWluZXJkL3JlbGVhc2VzL2Rvd25sb2FkL3YxLjcuMjMvY29udGFpbmVyZC0xLjcuMjMtbGludXgtYXJtNjQudGFyLmd6CnRhciBDenh2ZiAvdXNyL2xvY2FsLyAke1RNUERJUn0vY29udGFpbmVyZC0
xLjcuMjMtbGludXgtYXJtNjQudGFyLmd6Ci91c3IvbG9jYWwvYmluL2N0ciBvY2kgc3BlYyA+IC9ldGMvY29udGFpbmVyZC9jcmktYmFzZS5qc29uCmN1cmwgLUwgLW8gL2V0Yy9zeXN0ZW1kL3N5c3RlbS9jb250YW
luZXJkLnNlcnZpY2UgaHR0cHM6Ly9yYXcuZ2l0aHVidXNlcmNvbnRlbnQuY29tL2NvbnRhaW5lcmQvY29udGFpbmVyZC9tYWluL2NvbnRhaW5lcmQuc2VydmljZQpzeXN0ZW1jdGwgZGFlbW9uLXJlbG9hZpzeXN0ZW
W1jdGwgZW5hYmxlIC0tbm93IGNvbnRhaW5lcmQKbWtkaXIgLXAgL2V0Yy9hcHQva2V5cmluZ3MKY3VybCAtZnNTTCBodHRwczovL3BrZ3MuazhzLmlvL2NvcmU6L3N0YWJsZTovdjEuMjkvZGViL1JlbGVhc2Uua2V5
IHwgZ3BnIC0tZGVhcm1vciAtbyAvZXRjL2FwdC9rZXlyaW5ncy9rdWJlcm5ldGVzLWFwdC1rZXlyaW5nLmdwZwphcHQtZ2V0IGVwZGF0ZSAmJiBhcHQtZ2V0IGluc3RhbGwgLXkga3ViZWxldCBrdWJlYWRtIGt1YmV
jdGwKc3lzdGVtY3RsIGRhZW1vbi1yZWxvYWQKc3lzdGVtY3RsIGVuYWJsZSAtLW5vdyBrdWJlbGV0CnJtIC1yZiAke1RNUERJUn0K

runcmd:
  - [ /var/tmp/setup-script.sh ]
EOFCLOUDINIT
}

# bfb_post_install()
# {
#     log ====================== bfb_post_install ======================
#     mst start
#     mst_device=$(/bin/ls /dev/mst/mt*pciconf0 2> /dev/null)
#     # Setting SF enable per Nvidia documentation
#     # Ref: https://docs.nvidia.com/doca/sdk/nvidia+bluefield+dpu+scalable+function+user+guide/index.html
#     # and DPDK documentation
#     # Ref: https://doc.dpdk.org/guides-21.11/nics/mlx5.html
#     log "Setting SF enable and BAR size for $mst_device"
#     for mst_device in /dev/mst/mt*pciconf*
#     do
#       log "Disable port owner from ARM side for $mst_device"
#       mlxconfig -y -d $mst_device s PF_BAR2_ENABLE=0 PER_PF_NUM_SF=1 PF_TOTAL_SF=252 PF_SF_BAR_SIZE=12
#     done
# }
EOFBFTEMPLATE
)
    read -p "Enter the number of DPUs (default: 1): " num_dpus
    num_dpus=${num_dpus:-1}
    read -p "Enter the base hostname (default: dpu): " base_hostname
    base_hostname=${base_hostname:-dpu}
    echo "Enter the Ubuntu password minimum 12 characters (e.g. 'a123456AbCd!'): "
    # Password policy reference: https://docs.nvidia.com/networking/display/bluefielddpuosv490/
default+passwords+and+policies#src-3432095135_DefaultPasswordsandPolicies-UbuntuPasswordPolicy
    read -s clear_password
    ubuntu_password=$(openssl passwd -1 "${clear_password}")
    read -p "Enter tmfifo_net IP subnet mask. Useful if you have more than 1 DPU (default: 30): " ip_mask
    ip_mask=${ip_mask:-30}
    base_ip=${base_ip:-192.168.100}
    read -p "Do you want the DPU mgmt interface oob_net0 to use DHCP? (yes/no, default: yes): " use_dhcp
    use_dhcp=${use_dhcp:-yes}
    if [[ "$use_dhcp" =~ ^([nN][oO]|[nN])$ ]]; then
      read -p "Enter the static IP for oob_net0: " oob_ip
      read -p "Enter the subnet mask for oob_net0: " oob_mask
    fi

    for ((i=1; i<=num_dpus; i++)); do
        hostname="${base_hostname}-${i}"
        ip_address="${base_ip}.$(( i + 1 ))"
```

```
        net_rshim_mac=00:1a:ca:ff:ff:1${i}
        output_file="bfb_config_${hostname}.conf"

        echo "Generating configuration for ${hostname} with IP ${ip_address}..."
        echo "$bf_conf_template" | sed -e "s/{{HOSTNAME}}/${hostname}/g" \
            -e "s|{{PASSWORD}}|${ubuntu_password}|g" \
            -e "s/{{IP_ADDRESS}}/${ip_address}/g" \
            -e "s/{{IP_MASK}}/${ip_mask}/g" \
            -e "s/{{NET_RSHIM_MAC}}/${net_rshim_mac}/g" \
            > "${output_file}"
        cat << EOL
Configuration for ${hostname} is ${output_file}
To use the config run:
bfb-install --rshim rshim$(( i - 1 )) --config ${output_file} --bfb <bf-bundle-path>
EOL
done
}


generate_bluefield_config
```

Run the script to generate BlueField configuration.

Bash Session

```
host# chmod +x dpu-config.sh && ./dpu-config.sh
Enter the number of DPUs (default: 1): 1
Enter the base hostname (default: dpu): test-lab
Enter the Ubuntu password minimum 12 characters (e.g. 'a123456AbCd!'):
Enter tmfifo_net IP subnet mask. Useful if you have more than 1 DPU (default: 30):
Generating configuration for test-lab-1 with IP 192.168.100.2...
Configuration for test-lab-1 is bfb_config_test-lab-1.conf
To use the config run:
bfb-install --rshim rshim0 --config bfb_config_test-lab-1.conf --bfb <bf-bundle-path>
```

The script produced a file named `bfb_config_test-lab-1.conf` based on input.

4.3 3. Install BF Bundle

Use `bfb-install` tool to install the bf-bundle. The following example assumes bf-bundle `bf-bundle-2.9.0-83_24.10_ubuntu-22.04_dev.20241121.bfb`

Install bf-bundle on DPU

```
host# bfb-install --rshim rshim0 --config bfb_config_test-lab-1.conf --bfb bf-bundle-2.9.0-83_24.10_ubuntu-22.04_dev.20241121.bfb
```

Follow status of DPU installation on `/dev/rshim0/misc` until DPU is reported ready.

Bash Session

```
host# cat /dev/rshim0/misc
DISPLAY_LEVEL   2 (0:basic, 1:advanced, 2:log)
BF_MODE         Unknown
BOOT_MODE       1 (0:rshim, 1:emmc, 2:emmc-boot-swap)
BOOT_TIMEOUT    300 (seconds)
USB_TIMEOUT     40 (seconds)
DROP_MODE       0 (0:normal, 1:drop)
SW_RESET        0 (1: reset)
DEV_NAME        pcie-0000:53:00.2
DEV_INFO        BlueField-3(Rev 1)
OPN_STR         N/A
UP_TIME         9628(s)
SECURE_NIC_MODE 0 (0:no, 1:yes)
FORCE_CMD       0 (1: send Force command)
--------------------------------------
           Log Messages
--------------------------------------
INFO[PSC]: PSC BL1 START
INFO[BL2]: start
INFO[BL2]: boot mode (emmc)
INFO[BL2]: VDD_CPU: 870 mV
INFO[BL2]: VDDQ: 1120 mV
INFO[BL2]: DDR POST passed
INFO[BL2]: UEFI loaded
INFO[BL31]: start
INFO[BL31]: lifecycle GA Secured
INFO[BL31]: runtime
INFO[BL31]: MB ping success
INFO[UEFI]: eMMC init
INFO[UEFI]: eMMC probed
INFO[UEFI]: UPVS valid
INFO[UEFI]: PCIe enum start
INFO[UEFI]: PCIe enum end
```

```
INFO[UEFI]: UEFI Secure Boot (disabled)
INFO[UEFI]: PK configured
INFO[UEFI]: Redfish enabled
INFO[UEFI]: DPU-BMC RF credentials not found
INFO[UEFI]: exit Boot Service
INFO[MISC]: Linux up
INFO[MISC]: DPU is ready
```

4.4 4. Join the DPU to the Kubernetes cluster

4.4.1 4.1. Get the join token from controller node/host

Bash Session

```
host# kubeadm token create --print-join-command
kubeadm join 10.144.50.50:6443 --token ************************ --discovery-token-ca-cert-hash sha256:************************
```

4.4.2 4.2. Join the Kubernetes cluster on the DPU

Bash Session

```
dpu# kubeadm join 10.144.50.50:6443 --token ************************ --discovery-token-ca-cert-hash sha256:************************
```

2025-02-25

5. Setup F5 BIG-IP Next for Kubernetes

The Kubernetes cluster is now ready for BIG-IP Next for Kubernetes installation.

5.1 1. Taint and Label

This lab assumes that DPU is dedicated for BNK installation. In order to prevent other general workload from scheduling on DPU node add the following taint.

> **Note**
>
> Replace with DPU node name.

Bash Session

```
host# kubectl taint node <dpu-node-name> dpu=true:NoSchedule
```

In this lab, BNK Dataplane is going to be installed as a Kubernetes daemonset and scheduled on nodes with the label `app=f5-tmm`. Add the label to DPU node

Bash Session

```
host# kubectl label node <dpu-node-name> app=f5-tmm
```

5.2 2. Kubernetes Namespaces

The two main Kubernetes namespaces categories we use in this guide; Product, and Tenant namespaces.

5.2.1 Product Namespaces

Used to install core components of BNK. In this lab guide, the BIG-IP Next for Kubernetes product will use 2 namespaces

- **f5-utils:** All shared components for BIG-IP Next installation will use this namespace.
- **default:** Operator, BIG-IP Next control plane, and BIG-IP Next Dataplane components will use this namespace.

> **Note**
>
> `default` namespace is available by default after Kubernetes installation. We need to create only the `f5-utils` namespace.

Create Product Namespaces

```
host# kubectl create ns f5-utils
```

5.2.2 Tenant Namespaces

F5 BNK watches specific Kubernetes namespaces for tenant services onboarding and configuring ingress/egress paths for these services.

Note

As of the writing of this document BNK requires the namespaces to be created to product installation. This requirement may change in future.

In this guide we use two tenant namespaces, `red` and `blue`.

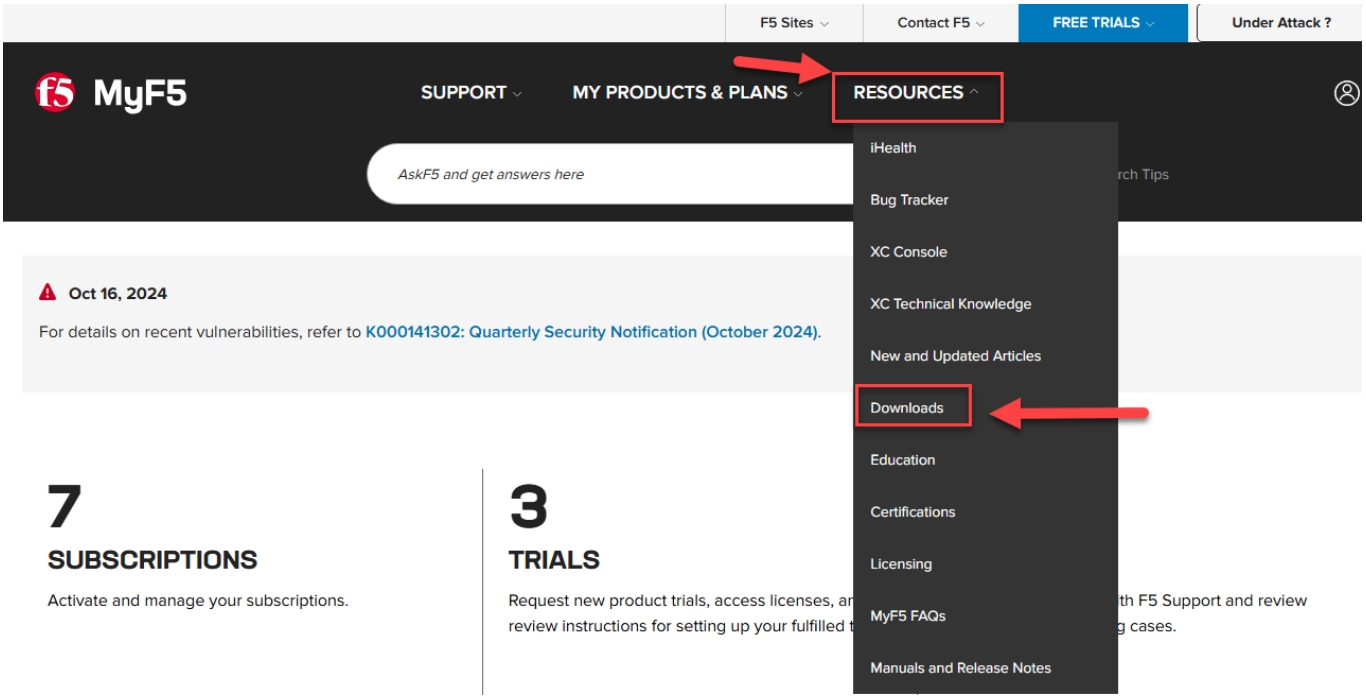
Create required namespaces:

Create Tenant Namespaces

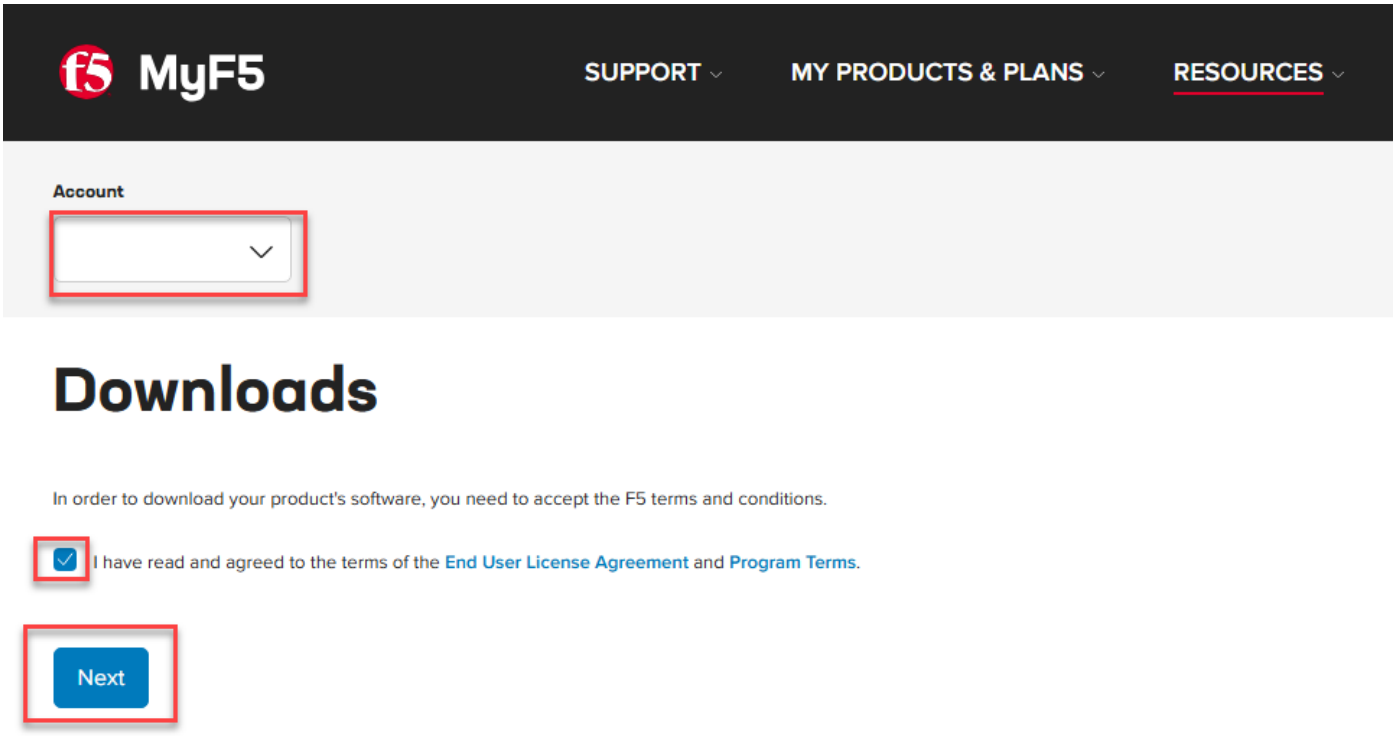
```
host# for ns in red blue; do kubectl create ns $ns; done
```

5.3 3. Authentication with F5 Artifact Registery (FAR)

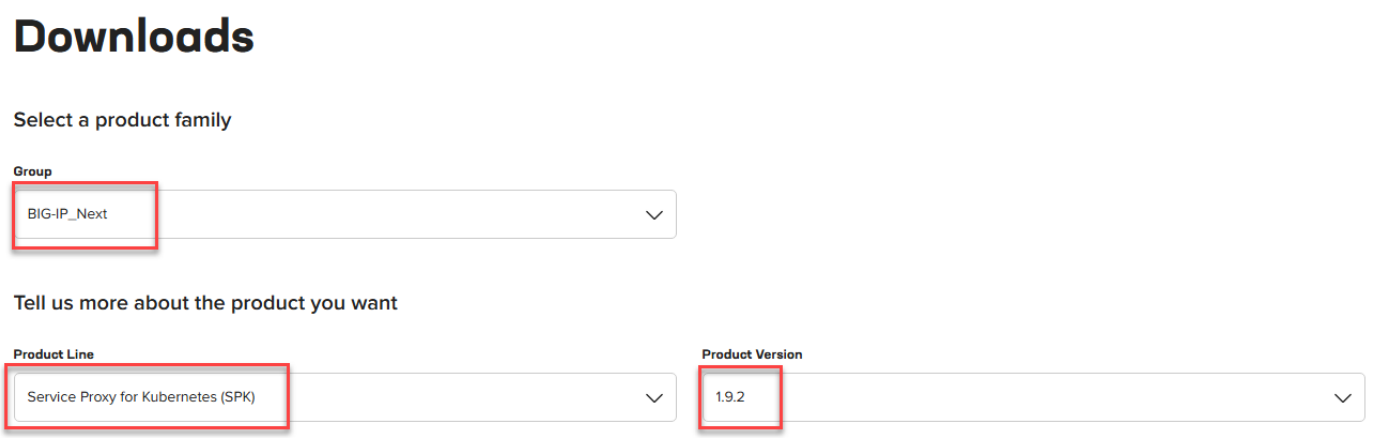
To access BNK product images, you must authenticate with the F5 Artifact Registry (FAR). In this section, we will go through obtaining the authentication key and creating Kubernetes pull secret.

1. Login to the MyF5.
2. Navigate to **Resources** and click **Downloads**.
3. 

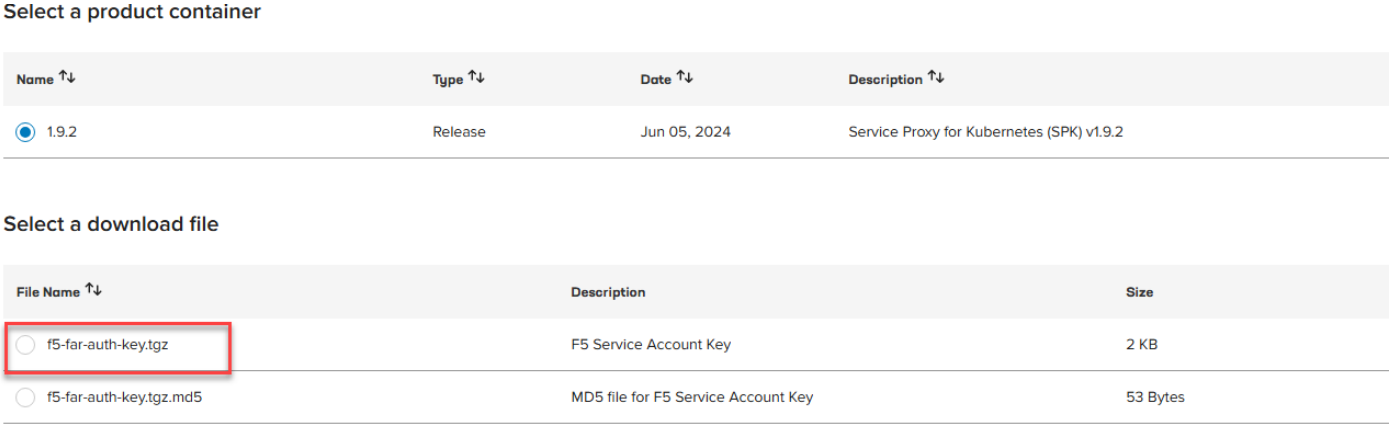
Ensure account is selected then review the End User License Agreement and the Program Terms and click to check the box for `I have read and agreed to the terms of the End User License Agreement and Program Terms.`



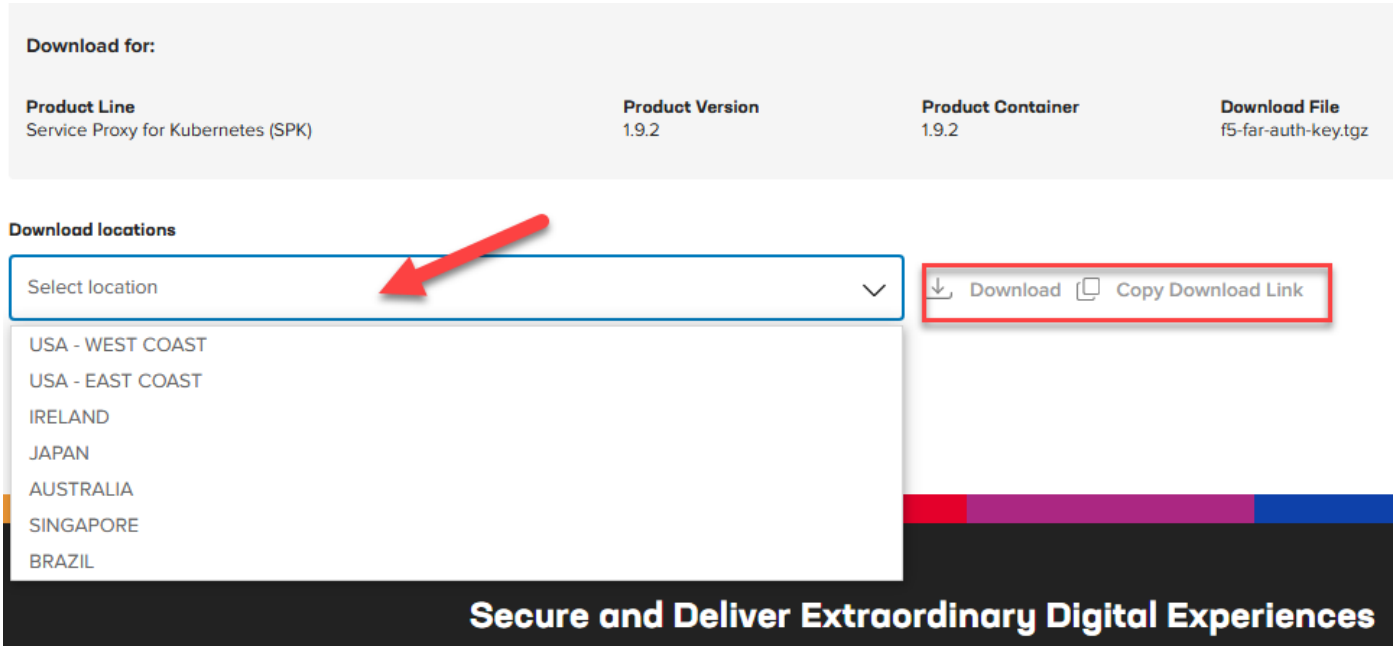
4. For Group select **BIT-IP_Next**, and **Service Proxy for Kubernetes (SPK)** in Product Line, and **1.9.2** for Product Version.



5. Select **f5-far-auth-key.tgz** to download.



6. Choose a location to download from and then download the file or copy link and download on the host linux.



7. Copy the downloaded file `zxvf f5-far-auth-key.tgz` to host dpu-install directory and expand to see a file named `cne_pull_64.json`. That is the file that contains FAR authentication key.

8. Use the far-kubernetes-secret.sh generate and install required Kubernetes pull secrets for FAR images.

Bash Session

```
host# ./far-kubernetes-secret.sh
```

9. Login to FAR helm registery from host terminal where kubectl and helm commands are available

Bash Session

```
host# cat cne_pull_64.json | helm registry login -u _json_key_base64 --password-stdin https://repo.f5.com
```

5.4 4. Cluster Wide Controller requirements

The Cluster Wide Controller (CWC) component manages license registeration and debug API. In this release there are some manual requirements that are needed. The steps also can be found in F5 guide to generate and install required certificates and ConfigMap.

Generate certificates that will be used to communicate with CWC component API, by pulling the script from F5 repo then generating certs for the f5-utils namespace service as follows.

- Pull and extract the chart containing cert generation scripts Install required package "make"

Bash Session

```
host# apt-get install -y make
```

Example Output

Bash Session

```
host# apt-get install -y make
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  make-doc
The following NEW packages will be installed:
  make
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 180 kB of archives.
After this operation, 426 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 make amd64 4.3-4.1build1 [180 kB]
Fetched 180 kB in 1s (218 kB/s)
Selecting previously unselected package make.
(Reading database ... 80515 files and directories currently installed.)
Preparing to unpack .../make_4.3-4.1build1_amd64.deb ...
Unpacking make (4.3-4.1build1) ...
Setting up make (4.3-4.1build1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
```

Bash Session

```
host# helm pull oci://repo.f5.com/utils/f5-cert-gen --version 0.9.1
host# tar zxvf f5-cert-gen-0.9.1.tgz
```

Example Output

Bash Session

```
host# helm pull oci://repo.f5.com/utils/f5-cert-gen --version 0.9.1
Pulled: repo.f5.com/utils/f5-cert-gen:0.9.1
Digest: sha256:89d283a7b2fef651a29baf1172c590d45fbd1e522fa90207ecd73d440708ad34
host# tar zxvf f5-cert-gen-0.9.1.tgz
cert-gen/
cert-gen/LICENSE
cert-gen/README.md
cert-gen/tls_gen/
cert-gen/tls_gen/tls-gen.md
cert-gen/tls_gen/__pycache__/
cert-gen/tls_gen/__pycache__/cli.cpython-39.pyc
cert-gen/tls_gen/__pycache__/info.cpython-39.pyc
cert-gen/tls_gen/__pycache__/__init__.cpython-39.pyc
cert-gen/tls_gen/__pycache__/verify.cpython-39.pyc
cert-gen/tls_gen/__pycache__/paths.cpython-39.pyc
cert-gen/tls_gen/__pycache__/extension_gen.cpython-39.pyc
cert-gen/tls_gen/__pycache__/gen.cpython-39.pyc
cert-gen/tls_gen/cli.py
cert-gen/tls_gen/extension_gen.py
cert-gen/tls_gen/__init__.py
cert-gen/tls_gen/paths.py
cert-gen/tls_gen/info.py
cert-gen/tls_gen/verify.py
cert-gen/tls_gen/gen.py
cert-gen/gen_cert.sh
cert-gen/Chart.yaml
cert-gen/openssl-cert-gen/
cert-gen/openssl-cert-gen/client-cert.conf
cert-gen/openssl-cert-gen/README.md
cert-gen/openssl-cert-gen/csr.conf
cert-gen/openssl-cert-gen/client-csr.conf
cert-gen/openssl-cert-gen/server-cert.conf
cert-gen/openssl-cert-gen/gen-yaml.sh
cert-gen/openssl-cert-gen/gen-certs.sh
cert-gen/basic/
cert-gen/basic/profile.py
cert-gen/basic/.DS_Store
cert-gen/basic/openssl.cnf
cert-gen/basic/grpc/
cert-gen/basic/grpc/grpc-service.ext
cert-gen/basic/grpc/validation-service.ext
cert-gen/basic/grpc/f5-fqdn-resolver.ext
cert-gen/basic/grpc/client.ext
cert-gen/basic/grpc/grpc.mk
cert-gen/basic/CertificateGenerator.md
cert-gen/basic/Makefile
cert-gen/common.mk
```

- Generate the API self-signed certificates. At the end of this step the script would have generated to main secret files Generating `cwc-license-certs.yaml` and `cwc-license-client-certs.yaml`

Bash Session

```
host# sh cert-gen/gen_cert.sh -s=api-server -a=f5-spk-cwc.f5-utils -n=1
```

Example Output

Bash Session

```
host# sh cert-gen/gen_cert.sh -s=api-server -a=f5-spk-cwc.f5-utils -n=1
-------------------------------------------------------------------
Service                   = api-server
Subject Alternate Name    = f5-spk-cwc.f5-utils
Working directory         = /root/bnk-dpu-install/api-server-secrets
-------------------------------------------------------------------
rm: cannot remove '/root/bnk-dpu-install/api-server-secrets': No such file or directory
Generating Secrets ...
python3 profile.py regenerate --password "" \
--common-name f5net \
--client-alt-name client \
--server-alt-name f5-spk-cwc.f5-utils \
--days-of-validity 3650 \
--client-certs 1 \
--key-bits 2048
Creating 1 client extensions...
Will generate a root CA and two certificate/key pairs (server and client)
=>  [openssl_req]
.+.......+.....+............+..+.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*..............+.+..+....+.............+......+...+.+..
+.......+.....+...+.......+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.............+...+.....+.+..+............+.......+..........
+.........+.+............+.................+.....+....+.........+.....+.+...+.....+.+.........+..........+......++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++
.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*..+.........+..........
+.+.....+.+.....+....+......+.....+....+...+......+..+.......+..+..........+........+...+...........+.......+.........+......++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++
-----
=>  [openssl_x509]
Will generate leaf certificate and key pair for server
Using f5net for Common Name (CN)
Using parent certificate path at /root/bnk-dpu-install/cert-gen/basic/testca/cacert.pem
Using parent key path at /root/bnk-dpu-install/cert-gen/basic/testca/private/cakey.pem
Will use RSA...
=>  [openssl_genpkey]
..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*....+.........+..+......+...........+.+..+....+........+.+.....+......+.........+.+......
+..+...+.......+........+...+.......+.....+.......+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.........+...+...+.................
+......+.+..............+..+.......+.+......+..+......+.+.....+.+.....+.........+....+.........+..+....+..+...+.........+...+................+...........
+....+......+.....+...+....+........+...+.......+...+..........+...+.+......+......+.........+.....+.+..+............+..+......+......+......+....+......+...
+..+.........+..+.........+.....+.+..+...+...+...+....+....+.........+.....+......+....+..........+...+.........+...+.....+.........+.+.........+........
+..........+...............+...........+.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+.........+......+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.
+..................+......+..+.+.....+.....................+...+.........+..+.........+....+...+..+...+...............+.......+........+...+.+.....+.........
+.+.....+.+.........+...+.....+......+....+...+........+.....................+...+...+.............+.....+....+..+...+.+.....+..............+......+.
+...........+...+........+......+...+.+.........................+..............+...............+.+..+.+......+........+...+....+.....+.+.............+...
+...+............+.....+.+...+...+........+....+...+...+.....+......+...+......+.+...+..+......+...+.+.....+.+.....+.......................+.........+.
+......+..+.+..+.......+...+..+.......+.........+..+...+..............+...+........+............+.......+...+.....+...+.......+........
+.......................+.+.........+.+.....+.+..+.......+......+..............+.+..+..........+...............+...+...........+..+...+.......+......+...+..
+.........+......+............+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
=>  [openssl_req]
-----
=>  [openssl_ca]
Using configuration from /tmp/tmpnso_b2s4
801B7E13897F0000:error:0700006C:configuration file routines:NCONF_get_string:no value:../crypto/conf/conf_lib.c:315:group=<NULL> name=unique_subject
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'f5net'
organizationName      :ASN.1 12:'server'
localityName          :ASN.1 12:'$$$$'
Certificate is to be certified until Jan  5 18:51:43 2035 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated
=>  [openssl_pkcs12]
Will generate leaf certificate and key pair for client
Using f5net for Common Name (CN)
Using parent certificate path at /root/bnk-dpu-install/cert-gen/basic/testca/cacert.pem
Using parent key path at /root/bnk-dpu-install/cert-gen/basic/testca/private/cakey.pem
Will use RSA...
=>  [openssl_genpkey]
......+.+......+...+..+....+...+..+...+...+...+..............+...........+...........+.+.........+...+.......+....+..........+++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++*.....+.+........+......+....+......+.....+.+.....+...+....+..........++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++*.....+.+..+.........+..+.......+.........+......+.....+....+.....+.........+..........+..+.+.......+.........+..+.........+....+..+...+.......
+........+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
.............+.........+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+.........+.+..........+.+...+.....+.......+++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++*......+.......+..+..........+.........+.....+.......+.........+....................+..+...+....+...+...+.....
+.........+....+............+...+..............+......+.......+.....+...+....................+.+..+......+.+..................+.+...+..+..............
+..............+...+.+...+..........+.+...+......+......+......+..+...........+...+......++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
=>  [openssl_req]
-----
=>  [openssl_ca]
Using configuration from /tmp/tmpnso_b2s4
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'f5net'
organizationName      :ASN.1 12:'client'
localityName          :ASN.1 12:'$$$$'
Certificate is to be certified until Jan  5 18:51:44 2035 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated
=>  [openssl_pkcs12]
Done! Find generated certificates and private keys under ./result!
```

```
python3 profile.py verify --client-certs 1
Will verify generated server certificate against the CA...
Will verify server certificate against root CA
/root/bnk-dpu-install/cert-gen/basic/result/server_certificate.pem: OK
Will verify generated client certificate against the CA...
Will verify client certificate against root CA
/root/bnk-dpu-install/cert-gen/basic/result/client_certificate.pem: OK
Copying secrets ...
Generating /root/bnk-dpu-install/cwc-license-certs.yaml
Generating /root/bnk-dpu-install/cwc-license-client-certs.yaml
```

- Install secrets.

Bash Session

```
host# kubectl apply -f cwc-license-certs.yaml -n f5-utils
host# kubectl apply -f cwc-license-client-certs.yaml -n f5-utils
```

- Install the cwc-qkview-cm.yaml qkview config map file.

YAML

```
apiVersion: v1
kind: ConfigMap
metadata:
   name: cwc-qkview-cm
   namespace: f5-utils
```

- Install the cpcl-non-prod.yaml file that contains Json Key Set for license activation

YAML

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cpcl-key-cm
  namespace: f5-utils
data:
  jwt.key: |
    {
        "keys": [
            {
                "kid": "v1",
                "alg": "RS512",
                "kty": "RSA",
                "n": "26FcA1269RC6WNgRghIB7X772zTTts02NsqqN-_keSz5FVq1Ekg151NFu_53Tgz1FGsUiX4OUj-fOmuhK9uzkQv0zYZgXY6zmRo_9P-
QgiycuFo7DWquDwEx4rZiMxXwlA9ER56s8PDdbXyfi3ceMV-aUQZFqMiU6gOTl5d7uMfskocPF4ja8ZRrLlXAzzRIR62VgbQa-3sT0_SZ4w1ME4eLzO1yb-
Ex9va4JnwToVLSKfsZp6jYs9nvAGjZ8aN2_lzBx8uiZ1HQozGqcf0AEjU-
FEY73Umvmyvzd4woQLQlbvyrRtL9_IkL2ySdQ9Znh2lXBdsmA9cLz4ZAYPdmvcjsyBaZmh15EOkczpVVan1_VVD4o28uLDpzQVDk_GNUYoZIRsuOzuKvzih0gkv-
StH29umHbdKXrUhlMWM1zyaxz8gkHatn-g5uh70WwVwqPtfHaNrQ0fFiWoyGVOA_-
XqsJWA9NLJorewp9HOVlyF8qzu5s9cFO4UGQas0fF2QR9QvhgCymK7iWbEFF3PXqUQTLfFsITgix3mmeXVYC3ODsPKvcFhNBqQxmeXM04N2XMLluz2qp581NUJygWAAfq7la0ylDJ1MtefyESD8SBs1at2a
8kSEBJCdCtAuNX2q33JjxQP3AiGvHcKEAjd1uaNeSgdHC93BzT3u0gbh2Ok",
                "e": "AQAB",
                "x5c": [

"MIIFqDCCBJCgAwIBAgIRAK+LbrS2gkaJSeoUQpMK0LswDQYJKoZIhvcNAQELBQAwgacxCzAJBgNVBAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRowGAYDVQQKDBFGNSBOZXR3b3JrcywgSW5jLjEeMB
wGA1UECwwVQ2VydGlmaWNhdGUgQXV0aG9yaXR5MTUwMwYDVQQDDCxGNSBTVEcgSXNzdWluZyBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkgVEVFTSBWMTEQMA4GA1UEBwwHU2VhdHRsZTAeFw0yMTEwMTEyMzI0N
TFaFw0yNjEwMTEwMDI0NTFaMIGBMQswCQYDVQQGEwJVUzETMBEGA1UECAwKV2FzaGluZ3RvbjEQMA4GA1UEBwwHU2VhdHRsZTEaMBgGA1UECgwRRjUgTmV0d29ya3MsIEluYy4xDTALBgNVBAsMBFRFRU0x
IDAeBgNVBAMMF0Y1IFNURyBURUVNIEpXVCBBdXRoIHYxMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA26FcA1269RC6WNgRghIB7X772zTTts02NsqqN+/keSz5FVq1Ekg151NFu/
53Tgz1FGsUiX4OUj+fOmuhK9uzkQv0zYZgXY6zmRo/9P+QgiycuFo7DWquDwEx4rZiMxXwlA9ER56s8PDdbXyfi3ceMV+aUQZFqMiU6gOTl5d7uMfskocPF4ja8ZRrLlXAzzRIR62VgbQa+3sT0/
SZ4w1ME4eLzO1yb+Ex9va4JnwToVLSKfsZp6jYs9nvAGjZ8aN2/lzBx8uiZ1HQozGqcf0AEjU+FEY73Umvmyvzd4woQLQlbvyrRtL9/
IkL2ySdQ9Znh2lXBdsmA9cLz4ZAYPdmvcjsyBaZmh15EOkczpVVan1/VVD4o28uLDpzQVDk/
GNUYoZIRsuOzuKvzih0gkv+StH29umHbdKXrUhlMWM1zyaxz8gkHatn+g5uh70WwVwqPtfHaNrQ0fFiWoyGVOA/
+XqsJWA9NLJorewp9HOVlyF8qzu5s9cFO4UGQas0fF2QR9QvhgCymK7iWbEFF3PXqUQTLfFsITgix3mmeXVYC3ODsPKvcFhNBqQxmeXM04N2XMLluz2qp581NUJygWAAfq7la0ylDJ1MtefyESD8SBs1at2
a8kSEBJCdCtAuNX2q33JjxQP3AiGvHcKEAjd1uaNeSgdHC93BzT3u0gbh2OkCAwEAAaOB8jCB7zAJBgNVHRMEAjAAMB8GA1UdIwQYMBaAFLDdK33QD9FdLnrVFw+ZAkQUayxCMB0GA1UdDgQWBBQw/
hNgf2AoJAF086NV7JGQj+B2NzAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMHMGA1UdHwRsMGowaKBmoGSGYmh0dHA6Ly9jcmwtdGVlbS1zdGctb3JlLWY1LnMzLnVz
LXdlc3QtMi5hbWF6b25hd3MuY29tL2NybC85ZGFmNGVlNy1iOGNkLTRiODEtOWE0MC00YjU3MGY0N2VhYWUuY3JsMA0GCSqGSIb3DQEBCwUAA4IBAQApzkSnsfuNSMHxVmL78pOQ+Rxkz1uYSVT0k1W45iu
fVmP0ixd8hFPcfb8u1RoHZ/58Gl52JPCudAB2sc4k/
lHNT9cKL4w5F8LybB8uNJXikAqzu4HFobRYMiPtVQ7M8cFz5SgvGclxzBAbZzK5u5xZuGSkI6tG9l+D5JhW2LesRuQSBQniBgRhmtAJB7SXuZ2sNKsq04h7DWcpdjCSferymeCOLQcgy5F3ragKML8zyuNe
KqtvZnUzJElKoU8G+Oo7MQXO7P5n6HX0NLfqqisv8CfSJUZTa1IRcfFUDJrcHtCgzingalzLLKzyelqR+YeY+j21jwVdnDVZIkFid2He",

"MIIFDjCCAvagAwIBAgIBBTANBgkqhkiG9w0BAQsFADCBhDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRgwFgYDVQQKEw9GNSBOZXR3b3JrcyBJbmMxHjAcBgNVBAsTFUNlcnRpZmljYXRlIEF1dGhvcm
l0eTEuMCwGA1UEAxMlRjUgSW50ZXJtZWRpYXRlIENlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0xODEyMTMxOTU3NDlaFw0yODEyMTAxOTU3NDlaMIGnMQswCQYDVQQGEwJVUzETMBEGA1UECAwKV2FzaGluZ
3RvbjEaMBgGA1UECgwRRjUgTmV0d29ya3MsIEluYy4xHjAcBgNVBAsMFUNlcnRpZmljYXRlIEF1dGhvcml0eTE1MDMGA1UEAwwsRjUgU1RHIElzc3VpbmcgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IFRFRU0g
VjExEDAOBgNVBAcMB1NlYXR0bGUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC1zQVzFAkzJdDMk+blcmQ506+GhWZOoe32rSWpkztdH7nYHtE6ITx44nHErRLOXXcYsf8nQZyV2RiE5Gyxwvd
g3ClC9XqfT702FxFWLPpD/
I53cRZasYH3dFfdvDztEOlTsUrMfo7Bzfh7ZMaMkBczHno0DdP61lp6pfzsQRLSfdaWxLKkCxc3P2xDUyx9F7uX3lXsT042OuiZpoCrMumO53hmvw6ZtP6mH7d6dM7nhYhTIGxRMYrzEAHKl+JM0Jnaabwx
w4UBMkxozxP+kLvDXrwLADjMslEuVeq1r7WwNa33y8aXfBUZpDCgJKfYvvPQIUD5d6ui0v7vwAQRgX/fAgMBAAGjZjBkMB0GA1UdDgQWBBSw3St90A/
RXS561RcPmQJEFGssQjAfBgNVHSMEGDAWgBRz1uVFvQMN0SWZ8zjGfQLZ+vrt6TASBgNVHRMBAf8ECDAGAQH/AgEBMA4GA1UdDwEB/
wQEAwIBhjANBgkqhkiG9w0BAQsFAAOCAgEABB8ygsfvpId2OPMh3jnTtEpfcJy80yu7vFVSMDQ/4xKTBSR0iFcCNmMJ8i4PL0E8RqFzcsUaG9Rq2uyiW71Y/+QiC0/
xN8pXTua9zH1aYPLKTa62IB5Dnfax+QccNCehCAoJ/W4yVeY9/nHbSlYt8+eOMSdUJf/hcaPuHbLs6rJI9GHo9CNeBtWH0q+Xw3rRAXSrNXMg+CRE55JOVaDdzRUOEdf962Pd/MRN7+Sypyj2dR9rCJ/
SKxf0HQr6NOGSAc3QbLun0bzew/0Nlww8UpCXV/
ABiBFUBDvIhapMQqoErMuPm0CvqBdVCWafOa8qylHHOCkEUxTlxtk3WTwEI4RcrHnHVO4eIkstLe8+4HvKvCwXwoDlcms44lIzQpoPvVclkYYKH0d9GjY1dDSXxYeIm7aPeA6VutQoTd8ozKqZFjueESJB7
JATC1q5PSiOhNIUr1d9Y7CbTpLWAl7ktJt5yZlcJBd3+5wztuCtwfQncjERRl8Sey3UuCjD836E7d4ZPldKUaJpDKpdzXIiiwDWCTL3G1iPBz+O1YyPAoQz6NFUsiHDnuIaGMfYLouf0ltuHTBwzcQbkFtH
7PeY5Qwts617AQBy5lCJ3HLdJ9Hg3CwTXlBqFR+T/8vF0n6+AuE0ZFjmbJYJs0m4EObk0IOcex4ft33fPDEFWRJpHIs=",
                    "MIIGFzCCA/
+gAwIBAgIBAjANBgkqhkiG9w0BAQsFADCBsDEYMBYGA1UEChMPRjUgTmV0d29ya3MgSW5jMSEwHwYDVQQLExhGNSBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkxHTAbBgkqhkiG9w0BCQEWDnJvb3RAbG9jYWxob
```

```
3N0MRAwDgYDVQQHEwdTZWF0dGxlMQswCQYDVQQIEwJXQTELMAkGA1UEBhMCVVMxJjAkBgNVBAMTHUY1IFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MB4XDTIyMDcyMTIxMTQxOVoXDTMyMDcxODIxMTQx
OVowgYQxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJXQTEYMBYGA1UEChMPRjUgTmV0d29ya3MgSW5jMR4wHAYDVQQLExVDZXJ0aWZpY2F0ZSBBdXRob3JpdHkxLjAsBgNVBAMTJUY1IEludGVybWVkaWF0ZSB
DZXJ0aWZpY2F0ZSBBdXRob3JpdHkwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCg/
uLiu6BbzIfZnehoaBepLVVQIeEgCUPxg6iHNANvlRC3FTElriLIDLJ3zAktdd3B1CSttG8sS4z3TKmN1+C1GXJRUZ2TlPRSaPBIhPs3xGFRgwBTJpTknY2oZITpfoDegAIuZcvfF5rwbcvndo2SiCbNHFyt
D/tIjWOJ+2a8k4T+8tBjaBxRygPSQycR6dciFo1ARJKfeWQV7Jkc0nzOd0T6JYh2Fjkyzx2EJca1zx88etIhfRg7XMo3SWLF6XJEoMBGoTeAAnE3oo+7KbBmcMYfcWGkwa/bOrBL4GCE/u3OS9z0wIoZ8/
ExdIvwvXfYCrHO7Q7mW/TL9VbnXQjqQiUu6KUaw7SnP2VnqOmWxZyeKGMPnp1CDNzljo97NUq+YBXWNUMrdG/
ahemcKoLQj6X9VNXrv5pE2u4HdsTHsXLE+bf4gvhWSPOoJR06d77C0eppMGmseYTIphvrFYbOkyUqJ3QPeh0alyRERPwZo7KWXbiWwwTs2Ya0IP4ndVxfnPJCAdyLs5dZcCPwaSZcqKS+ruGq/
NCdpv9c4qQlog4cgPJaLjdvyhgttHxKFb8gLwensE2R5j2EKk/eDVSMZH7DMxAMVCOwAXC7yU//
jzxbM79oLXJKtGUOqI5Lqo14oBQ9GN9jMadH7QIf98WUKxoI9jG2b7RVTaVI63xUuQIDAQABo2YwZDAdBgNVHQ4EFgQUc9blRb0DDdElmfM4xn0C2fr67ekwHwYDVR0jBBgwFoAUvt3/76pNZ1Iqkujy1aW
YEmZs4bowEgYDVR0TAQH/BAgwBgEB/wIBATAOBgNVHQ8BAf8EBAMCAYYwDQYJKoZIhvcNAQELBQADggIBAF8EmEr06Legji041di2NbG42oQ0Jgaa4du/V9jloUp/
N4Qo5t1upDrSQcEGdkLCgvGBDUKHaZWdJSRtoW4OxlNUfOeU0HEkt24TjwrW08eXDjmmDnqYjhPheeVJNMP2e0+Kj5l3ncTWPD/aS8HtZUdggpU8L9Y8vg6Tl143dZaePQEj+FHghmReIkRoJ2GT/
hXFp17p0lTTlcjdRv/zU/Yvtp7F0JL8tjkMqy1Al5xZYDWZznamKdMUT71ikMFVHOVRgK4L+mfLGjA3rHT/hTWQ6EentQmWwv1+wG8fvShBL48YwIFCW02VxD/
qdJjgcLGJ3KB9xxO2IBGo99bT2D1xLXgnu5odLWMIB3rUR2hKJWRJI+haODXgE8x+vzHjMpQjkv0Ud0TdL1/
ULnLkW0rIiksRQvtWNXfJe33MnMx+P+cQ8wvpCbtcLZVVlnpqRmfiN+YK9stvZS0RKOi10WdR2tJpYOUi/tJ4dQ/u7erYUrmu/
onVbe0M8P8w4CuDxhyEC3s1Gk4wppe9SqZgM2Op4FaRBzvm7oxMg27RngZ6BdK5JBlDY4SVXm5YxGkKMupTjXMo98pSn4mxlr5o4MU0YKsrUWENBM+MPUKb2rRRq0yyy0xsnvr33hr9OIlRjYYr06MiGHM5
YLKxsJm73YivOhZKAwhWEHSA6uZ7dCvA",
                    "MIIGFzCCA/
+gAwIBAgIBATANBgkqhkiG9w0BAQsFADCBsDEYMBYGA1UEChMPRjUgTmV0d29ya3MgSW5jMSEwHwYDVQQLExhGNSBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkxHTAbBgkqhkiG9w0BCQEWDnJvb3RAbG9jYWxob
3N0MRAwDgYDVQQHEwdTZWF0dGxlMQswCQYDVQQIEwJXQTELMAkGA1UEBhMCVVMxJjAkBgNVBAMTHUY1IFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MB4XDTE3MTAzMTIyMTQ1OVoXDTI3MTAyOTIyMTQ1
OVowgYQxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJXQTEYMBYGA1UEChMPRjUgTmV0d29ya3MgSW5jMR4wHAYDVQQLExVDZXJ0aWZpY2F0ZSBBdXRob3JpdHkxLjAsBgNVBAMTJUY1IEludGVybWVkaWF0ZSB
DZXJ0aWZpY2F0ZSBBdXRob3JpdHkwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCg/
uLiu6BbzIfZnehoaBepLVVQIeEgCUPxg6iHNANvlRC3FTElriLIDLJ3zAktdd3B1CSttG8sS4z3TKmN1+C1GXJRUZ2TlPRSaPBIhPs3xGFRgwBTJpTknY2oZITpfoDegAIuZcvfF5rwbcvndo2SiCbNHFyt
D/tIjWOJ+2a8k4T+8tBjaBxRygPSQycR6dciFo1ARJKfeWQV7Jkc0nzOd0T6JYh2Fjkyzx2EJca1zx88etIhfRg7XMo3SWLF6XJEoMBGoTeAAnE3oo+7KbBmcMYfcWGkwa/bOrBL4GCE/u3OS9z0wIoZ8/
ExdIvwvXfYCrHO7Q7mW/TL9VbnXQjqQiUu6KUaw7SnP2VnqOmWxZyeKGMPnp1CDNzljo97NUq+YBXWNUMrdG/
ahemcKoLQj6X9VNXrv5pE2u4HdsTHsXLE+bf4gvhWSPOoJR06d77C0eppMGmseYTIphvrFYbOkyUqJ3QPeh0alyRERPwZo7KWXbiWwwTs2Ya0IP4ndVxfnPJCAdyLs5dZcCPwaSZcqKS+ruGq/
NCdpv9c4qQlog4cgPJaLjdvyhgttHxKFb8gLwensE2R5j2EKk/eDVSMZH7DMxAMVCOwAXC7yU//
jzxbM79oLXJKtGUOqI5Lqo14oBQ9GN9jMadH7QIf98WUKxoI9jG2b7RVTaVI63xUuQIDAQABo2YwZDAdBgNVHQ4EFgQUc9blRb0DDdElmfM4xn0C2fr67ekwHwYDVR0jBBgwFoAUvt3/76pNZ1Iqkujy1aW
YEmZs4bowEgYDVR0TAQH/BAgwBgEB/wIBATAOBgNVHQ8BAf8EBAMCAYYwDQYJKoZIhvcNAQELBQADggIBAGgXhdFaLvqYyzBTsc2jrfJWvnwwQztwkk+
+R2vR5Skwhy1ke5+fycmaiwERtOuqqjq0pJpFJiO61T0wlm/
vF2HqsMMibvNgrSCvGurGyCdVTKahYNKqHWsevhhnqjoGWSlm7hgVz5wtGQoyImJMa3+qFvMtOZSFpHzSlteinLucPrA4EEuTNh1RjRNmq7J0oAl3+PG5bK5DpySOh4jX119G7P9VhX+aLVangYi9ZkBJgm
x4tmsg7Caqg7RF0tIsnTdad9uI+WKty/vsXDntb8zzonTg59BhW3zMcT1p6Xutz4WyC0BHeculq+8LtLO0G2Dxxzeik/
V9Z03mOW8bscjkPh5GcXtwTdSZiyh1ewGtyR0Jcj6vYqBLkXQtfX5JERuCuFcb15NE1Mr3V91kdJs1WPPY7fcwgPVEdBCa4Yo/FrwzoKuYqQIE8jnLEX+YOAcS8VS1eurPRl7v5ZZSMU2RnacvXL9TJ/
Wk32KgUCOLjy2O3MmaPZLnasgDVQGXOdP4Q2pp7TRwjvR3GJvLCFQtvKBOZO35EhvF0AwAxi5PmTwSL3k3zdYlYADIyyo1YMhiS/
FQueo06dtyShsoSPtmo7Jthus9xKxoyQVih11UdDieR9ZdikNRX805w1jc5O0DWFkq9AKDxLYKUkE/MxuvXzXls9RFHSwKMvzfxa0r"
                ],
                "use": "sig"
            }
        ]
    }
```


5.5 5. Scalable Function CNI Binary

F5 created a CNI binary used here to move Scalable Function netdevice and RDMA devices inside of the dataplane container. This CNI is invoked by Multus delegation when attaching the Dataplane component to defined networks.

Bash Session

```
host# helm pull oci://repo.f5.com/utils/f5-eowyn  --version 2.0.0-LA.1-0.0.11
host# tar zxvf f5-eowyn-2.0.0-LA.1-0.0.11.tgz
```

Example Output

Bash Session

```
host# tar zxvf f5-eowyn-2.0.0-LA.1-0.0.11.tgz
f5-eowyn/
f5-eowyn/sf
f5-eowyn/Chart.yaml
```

Note

The `sf` CNI must be copied to all DPU nodes in the `/opt/cni/bin/` directory. For example:

Bash Session

```
host# scp f5-eowyn/sf root@<dpu-ip>:/opt/cni/bin/
```

5.6 6. Configure Network Attachment Definitions

Now that the CNI binary is installed we can configure Multus Network Attachment Definitions based on the configuration used in SR-IOV Device Plugin ConfigMap and using the `sf` CNI.\ Apply the network-attachments.yaml configuration to the default namespace.

This step will create two network attachment definitions for internal and external scalable functions as described in the lab diagram.

5.7 7. (Optional) Install Grafana and Prometheus

Using Prometheus and Grafana to collect and visualize the metrics.

5.7.1 Install Prometheus

Prometheus example for this lab is defined in the prometheus.yaml file.

Show Prometheus deployment

YAML

```yaml
---
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: prometheus
spec:
  secretName: prometheus-secret
  issuerRef:
    group: cert-manager.io
    kind: ClusterIssuer
    name: bnk-ca-cluster-issuer
  duration: 8640h
  privateKey:
    rotationPolicy: Always
    encoding: PKCS1
    algorithm: RSA
    size: 4096
  revisionHistoryLimit: 10
  commonName: f5net.com
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: prometheus
  managedFields:
  - apiVersion: apps/v1
  name: prometheus
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: prometheus
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      annotations:
        prometheus.io/port: "9090"
        prometheus.io/scrape: "true"
      labels:
        app: prometheus
    spec:
      containers:
      - args:
        - '--storage.tsdb.retention.time=6h'
        - '--storage.tsdb.path=/prometheus'
        - '--config.file=/etc/prometheus/prometheus.yaml'
        image: prom/prometheus
        imagePullPolicy: Always
        name: prometheus
        ports:
        - containerPort: 9090
          name: web
          protocol: TCP
        volumeMounts:
        - mountPath: /etc/prometheus
          name: prometheus-config-volume
        - mountPath: /prometheus
          name: prometheus-storage-volume
        - name: prometheus-volume
          mountPath: /etc/ssl
          readOnly: true
      restartPolicy: Always
      schedulerName: default-scheduler
      volumes:
      - configMap:
          defaultMode: 420
          name: prometheus-config
        name: prometheus-config-volume
      - name: prometheus-volume
        secret:
          secretName: prometheus-secret
      - emptyDir: {}
        name: prometheus-storage-volume
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
  namespace: default
data:
  prometheus.yaml: |
    global:
      scrape_interval: 15s
      evaluation_interval: 15s
    scrape_configs:
      - job_name: 'k8s_pod'
        kubernetes_sd_configs:
```

```
        - role: pod
          namespaces:
            names:
            - default
        relabel_configs:
          - source_labels: [__meta_kubernetes_pod_label_metrics_prometheus, __meta_kubernetes_pod_container_port_number]
            regex: publish;9090
            action: keep
          - source_labels: [__meta_kubernetes_pod_name]
            action: replace
            target_label: pod_name
          - source_labels: [__meta_kubernetes_namespace]
            action: replace
            target_label: namespace
        scheme: https
        tls_config:
          ca_file: "/etc/ssl/ca.crt"
          cert_file: "/etc/ssl/tls.crt"
          key_file: "/etc/ssl/tls.key"
          insecure_skip_verify: true
---
apiVersion: v1
kind: Service
metadata:
    name: prometheus-service
    namespace: default
    annotations:
        prometheus.io/scrape: 'true'
        prometheus.io/port:   '9090'
spec:
    selector:
        app: prometheus
    type: NodePort
    ports:
    - port: 8080
      targetPort: 9090
      nodePort: 30000
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: prometheus-default
rules:
- apiGroups:
  - ""
  resources:
  - pods
  - services
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - extensions
  resources:
  - ingresses
  verbs:
  - get
  - list
  - watch
- nonResourceURLs:
  - /metrics
  verbs:
  - get
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: prometheus-default
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus-default
subjects:
- kind: ServiceAccount
  name: default
  namespace: default
```

Apply the file in default namespace

Bash

```
host# kubectl apply -f prometheus.yaml
```

5.7.2 Install Grafana

Grafana example for this lab is defined in the grafana.yaml file.

Show Grafana deployment

YAML

```yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grafana
spec:
  replicas: 1
  selector:
    matchLabels:
      app: grafana
  template:
    metadata:
      name: grafana
      labels:
        app: grafana
    spec:
      containers:
      - name: grafana
        image: grafana/grafana
        ports:
        - name: grafana
          containerPort: 3000
        resources:
          limits:
            memory: "1Gi"
            cpu: "1000m"
          requests:
            memory: 500M
            cpu: "500m"
        volumeMounts:
          - mountPath: /var/lib/grafana
            name: grafana-storage
          - mountPath: /etc/grafana/provisioning/datasources
            name: grafana-datasources
            readOnly: false
      volumes:
        - name: grafana-storage
          emptyDir: {}
        - name: grafana-datasources
          configMap:
              defaultMode: 420
              name: grafana-datasources
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: grafana-datasources
data:
  prometheus.yaml: |-
    {
        "apiVersion": 1,
        "datasources": [
            {
               "access":"proxy",
                "editable": true,
                "name": "prometheus",
                "orgId": 1,
                "type": "prometheus",
                "url": "http://prometheus-service.default.svc:8080",
                "version": 1
            }
        ]
    }
---
apiVersion: v1
kind: Service
metadata:
  name: grafana
  annotations:
      prometheus.io/scrape: 'true'
      prometheus.io/port:   '3000'
spec:
  selector:
    app: grafana
  type: NodePort
  ports:
    - port: 3000
      targetPort: 3000
      nodePort: 32000
```

Apply the file in default namespace

Bash

```
host# kubectl apply -f grafana.yaml
```

Grafana Dashboard

An example Grafana dashboard is provided in the grafana-dashboard.json file.

Show Grafana Dashboard

JSON


```
{
  "dashboard": {
    "annotations": {
      "list": [
        {
          "builtIn": 1,
          "datasource": {
            "type": "datasource",
            "uid": "grafana"
          },
          "enable": true,
          "hide": true,
          "iconColor": "rgba(0, 211, 255, 1)",
          "name": "Annotations & Alerts",
          "target": {
            "limit": 100,
            "matchAny": false,
            "tags": [],
            "type": "dashboard"
          },
          "type": "dashboard"
        }
      ]
    },
    "editable": true,
    "fiscalYearStartMonth": 0,
    "graphTooltip": 0,
    "id": 1,
    "links": [],
    "panels": [
      {
        "collapsed": true,
        "gridPos": {
          "h": 1,
          "w": 24,
          "x": 0,
          "y": 0
        },
        "id": 38,
        "panels": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "fieldConfig": {
              "defaults": {
                "color": {
                  "mode": "palette-classic"
                },
                "custom": {
                  "axisBorderShow": false,
                  "axisCenteredZero": false,
                  "axisColorMode": "text",
                  "axisLabel": "",
                  "axisPlacement": "auto",
                  "barAlignment": 0,
                  "barWidthFactor": 0.6,
                  "drawStyle": "line",
                  "fillOpacity": 0,
                  "gradientMode": "none",
                  "hideFrom": {
                    "legend": false,
                    "tooltip": false,
                    "viz": false
                  },
                  "insertNulls": false,
                  "lineInterpolation": "linear",
                  "lineWidth": 1,
                  "pointSize": 5,
                  "scaleDistribution": {
                    "type": "linear"
                  },
                  "showPoints": "auto",
                  "spanNulls": false,
                  "stacking": {
                    "group": "A",
                    "mode": "none"
                  },
                  "thresholdsStyle": {
                    "mode": "off"
                  }
                },
                "mappings": [],
                "thresholds": {
                  "mode": "absolute",
                  "steps": [
                    {
                      "color": "green",
                      "value": null
                    },
                    {
                      "color": "red",
```

```
                "value": 80
              }
            ]
          },
          "unit": "Bps"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 8,
        "w": 6,
        "x": 0,
        "y": 1
      },
      "id": 14,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "hideZeros": false,
          "mode": "single",
          "sort": "none"
        }
      },
      "pluginVersion": "11.5.2",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "prometheus"
          },
          "exemplar": true,
          "expr": "rate(f5_ingress_system_disk_io{}[$__rate_interval])",
          "interval": "",
          "legendFormat": "{{device}}-{{direction}}",
          "refId": "A"
        }
      ],
      "title": "F5 Ingress Disk IO",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "P1809F7CD0C75ACF3"
      },
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "barWidthFactor": 0.6,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "red",
                "value": 80
              }
```

```
                ]
              },
              "unit": "iops"
            },
            "overrides": []
          },
          "gridPos": {
            "h": 8,
            "w": 6,
            "x": 6,
            "y": 1
          },
          "id": 16,
          "options": {
            "legend": {
              "calcs": [],
              "displayMode": "list",
              "placement": "bottom",
              "showLegend": true
            },
            "tooltip": {
              "hideZeros": false,
              "mode": "single",
              "sort": "none"
            }
          },
          "pluginVersion": "11.5.2",
          "targets": [
            {
              "datasource": {
                "type": "prometheus",
                "uid": "prometheus"
              },
              "exemplar": true,
              "expr": "rate(f5_ingress_system_disk_operation_time{}[$__rate_interval])",
              "hide": false,
              "interval": "",
              "legendFormat": "{{device}}-{{direction}}",
              "refId": "A"
            }
          ],
          "title": "F5 Ingress Disk Operation Time",
          "type": "timeseries"
        },
        {
          "datasource": {
            "type": "prometheus",
            "uid": "prometheus"
          },
          "fieldConfig": {
            "defaults": {
              "color": {
                "mode": "palette-classic"
              },
              "custom": {
                "axisBorderShow": false,
                "axisCenteredZero": false,
                "axisColorMode": "text",
                "axisLabel": "",
                "axisPlacement": "auto",
                "barAlignment": 0,
                "barWidthFactor": 0.6,
                "drawStyle": "line",
                "fillOpacity": 0,
                "gradientMode": "none",
                "hideFrom": {
                  "legend": false,
                  "tooltip": false,
                  "viz": false
                },
                "insertNulls": false,
                "lineInterpolation": "linear",
                "lineWidth": 1,
                "pointSize": 5,
                "scaleDistribution": {
                  "type": "linear"
                },
                "showPoints": "auto",
                "spanNulls": false,
                "stacking": {
                  "group": "A",
                  "mode": "none"
                },
                "thresholdsStyle": {
                  "mode": "off"
                }
              },
              "mappings": [],
              "thresholds": {
                "mode": "absolute",
                "steps": [
                  {
                    "color": "green",
                    "value": null
                  },
                  {
                    "color": "red",
                    "value": 80
                  }
                ]
```

```
            }
          },
          "overrides": []
        },
        "gridPos": {
          "h": 8,
          "w": 5,
          "x": 12,
          "y": 1
        },
        "id": 12,
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "list",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "exemplar": true,
            "expr": "f5_ingress_system_cpu_time{}",
            "interval": "",
            "legendFormat": "{{cpu}}",
            "refId": "A"
          }
        ],
        "title": "F5 Ingress System CPU Time(Idle)",
        "type": "timeseries"
      },
      {
        "datasource": {
          "type": "prometheus",
          "uid": "prometheus"
        },
        "fieldConfig": {
          "defaults": {
            "color": {
              "mode": "palette-classic"
            },
            "custom": {
              "axisBorderShow": false,
              "axisCenteredZero": false,
              "axisColorMode": "text",
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
              "lineWidth": 1,
              "pointSize": 5,
              "scaleDistribution": {
                "type": "linear"
              },
              "showPoints": "auto",
              "spanNulls": false,
              "stacking": {
                "group": "A",
                "mode": "none"
              },
              "thresholdsStyle": {
                "mode": "off"
              }
            },
            "mappings": [],
            "thresholds": {
              "mode": "absolute",
              "steps": [
                {
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            },
            "unit": "decbits"
          },
```

```
            "overrides": []
          },
          "gridPos": {
            "h": 8,
            "w": 5,
            "x": 17,
            "y": 1
          },
          "id": 10,
          "options": {
            "legend": {
              "calcs": [],
              "displayMode": "list",
              "placement": "bottom",
              "showLegend": true
            },
            "tooltip": {
              "hideZeros": false,
              "mode": "single",
              "sort": "none"
            }
          },
          "pluginVersion": "11.5.2",
          "targets": [
            {
              "datasource": {
                "type": "prometheus",
                "uid": "prometheus"
              },
              "exemplar": true,
              "expr": "f5_ingress_system_memory_usage{}",
              "interval": "",
              "legendFormat": "",
              "refId": "A"
            }
          ],
          "title": "F5 Ingress System Memory Usage",
          "type": "timeseries"
        }
      ],
      "title": "Controller",
      "type": "row"
    },
    {
      "collapsed": true,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 1
      },
      "id": 37,
      "panels": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "fieldConfig": {
            "defaults": {
              "color": {
                "mode": "palette-classic"
              },
              "custom": {
                "axisBorderShow": false,
                "axisCenteredZero": false,
                "axisColorMode": "text",
                "axisLabel": "",
                "axisPlacement": "auto",
                "barAlignment": 0,
                "barWidthFactor": 0.6,
                "drawStyle": "line",
                "fillOpacity": 0,
                "gradientMode": "none",
                "hideFrom": {
                  "legend": false,
                  "tooltip": false,
                  "viz": false
                },
                "insertNulls": false,
                "lineInterpolation": "linear",
                "lineWidth": 1,
                "pointSize": 5,
                "scaleDistribution": {
                  "type": "linear"
                },
                "showPoints": "auto",
                "spanNulls": false,
                "stacking": {
                  "group": "A",
                  "mode": "none"
                },
                "thresholdsStyle": {
                  "mode": "off"
                }
              },
              "mappings": [],
              "thresholds": {
                "mode": "absolute",
                "steps": [
```

```
                {
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            }
          },
          "overrides": []
        },
        "gridPos": {
          "h": 9,
          "w": 5,
          "x": 0,
          "y": 2
        },
        "id": 4,
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "table",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_tmm_stat_0_cpu_usage_15mins{}",
            "interval": "",
            "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
            "range": true,
            "refId": "A"
          },
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_tmm_stat_2_cpu_usage_15mins{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
            "range": true,
            "refId": "B"
          }
        ],
        "title": "F5 TMM CPU Usage 15 mins",
        "type": "timeseries"
      },
      {
        "datasource": {
          "type": "prometheus",
          "uid": "P1809F7CD0C75ACF3"
        },
        "fieldConfig": {
          "defaults": {
            "color": {
              "mode": "palette-classic"
            },
            "custom": {
              "axisBorderShow": false,
              "axisCenteredZero": false,
              "axisColorMode": "text",
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
              "lineWidth": 1,
              "pointSize": 5,
              "scaleDistribution": {
                "type": "linear"
              },
              "showPoints": "auto",
```

```
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "decbits"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 9,
        "w": 5,
        "x": 5,
        "y": 2
      },
      "id": 2,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "table",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "hideZeros": false,
          "mode": "single",
          "sort": "none"
        }
      },
      "pluginVersion": "11.5.2",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "prometheus"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_tmm_stat_0_memory_used{}",
          "interval": "",
          "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
          "range": true,
          "refId": "A"
        },
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_tmm_stat_2_memory_total{}",
          "hide": false,
          "interval": "",
          "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
          "range": true,
          "refId": "B"
        }
      ],
      "title": "F5 TMM System Memory Usage",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "P1809F7CD0C75ACF3"
      },
      "description": "",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "barWidthFactor": 0.6,
            "drawStyle": "line",
```

```
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
              "lineWidth": 1,
              "pointSize": 5,
              "scaleDistribution": {
                "type": "linear"
              },
              "showPoints": "auto",
              "spanNulls": false,
              "stacking": {
                "group": "A",
                "mode": "none"
              },
              "thresholdsStyle": {
                "mode": "off"
              }
            },
            "mappings": [],
            "thresholds": {
              "mode": "absolute",
              "steps": [
                {
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            },
            "unit": "short"
          },
          "overrides": []
        },
        "gridPos": {
          "h": 9,
          "w": 5,
          "x": 10,
          "y": 2
        },
        "id": 33,
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "table",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_interface_stat_1_1_counters_pkts_in{}",
            "interval": "",
            "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
            "range": true,
            "refId": "A"
          },
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_interface_stat_1_1_counters_pkts_out{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
            "range": true,
            "refId": "B"
          }
        ],
        "title": "F5 TMM Interface 1.1 Packets In/Out",
        "type": "timeseries"
      },
      {
        "datasource": {
          "type": "prometheus",
          "uid": "P1809F7CD0C75ACF3"
        },
```

```
      "description": "",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "barWidthFactor": 0.6,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "decbits"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 9,
        "w": 5,
        "x": 15,
        "y": 2
      },
      "id": 32,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "table",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "hideZeros": false,
          "mode": "single",
          "sort": "none"
        }
      },
      "pluginVersion": "11.5.2",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_interface_stat_1_1_counters_bytes_in{}",
          "interval": "",
          "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
          "range": true,
          "refId": "A"
        },
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_interface_stat_1_1_counters_bytes_out{}",
```

```
            "hide": false,
            "interval": "",
            "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
            "range": true,
            "refId": "B"
          }
        ],
        "title": "F5 TMM Interface 1.1 Bytes In/Out",
        "type": "timeseries"
      },
      {
        "datasource": {
          "type": "prometheus",
          "uid": "P1809F7CD0C75ACF3"
        },
        "fieldConfig": {
          "defaults": {
            "color": {
              "mode": "palette-classic"
            },
            "custom": {
              "axisBorderShow": false,
              "axisCenteredZero": false,
              "axisColorMode": "text",
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
              "lineWidth": 1,
              "pointSize": 5,
              "scaleDistribution": {
                "type": "linear"
              },
              "showPoints": "auto",
              "spanNulls": false,
              "stacking": {
                "group": "A",
                "mode": "none"
              },
              "thresholdsStyle": {
                "mode": "off"
              }
            },
            "mappings": [],
            "thresholds": {
              "mode": "absolute",
              "steps": [
                {
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            },
            "unit": "decbits"
          },
          "overrides": []
        },
        "gridPos": {
          "h": 8,
          "w": 5,
          "x": 0,
          "y": 11
        },
        "id": 6,
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "list",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "editorMode": "code",
            "exemplar": true,
```

```
        "expr": "f5_tmm_tmm_stat_0_server_side_traffic_bytes_out{}",
        "interval": "",
        "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
        "range": true,
        "refId": "A"
      },
      {
        "datasource": {
          "type": "prometheus",
          "uid": "prometheus"
        },
        "editorMode": "code",
        "exemplar": true,
        "expr": "f5_tmm_tmm_stat_2_server_side_traffic_bytes_in{}",
        "hide": false,
        "interval": "",
        "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
        "range": true,
        "refId": "B"
      }
    ],
    "title": "F5 TMM Network Bytes In/Out",
    "type": "timeseries"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "P1809F7CD0C75ACF3"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "palette-classic"
        },
        "custom": {
          "axisBorderShow": false,
          "axisCenteredZero": false,
          "axisColorMode": "text",
          "axisLabel": "",
          "axisPlacement": "auto",
          "barAlignment": 0,
          "barWidthFactor": 0.6,
          "drawStyle": "line",
          "fillOpacity": 0,
          "gradientMode": "none",
          "hideFrom": {
            "legend": false,
            "tooltip": false,
            "viz": false
          },
          "insertNulls": false,
          "lineInterpolation": "linear",
          "lineWidth": 1,
          "pointSize": 5,
          "scaleDistribution": {
            "type": "linear"
          },
          "showPoints": "auto",
          "spanNulls": false,
          "stacking": {
            "group": "A",
            "mode": "none"
          },
          "thresholdsStyle": {
            "mode": "off"
          }
        },
        "mappings": [],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            },
            {
              "color": "red",
              "value": 80
            }
          ]
        },
        "unit": "short"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 8,
      "w": 5,
      "x": 5,
      "y": 11
    },
    "id": 8,
    "options": {
      "legend": {
        "calcs": [],
        "displayMode": "table",
        "placement": "bottom",
        "showLegend": true
      },
      "tooltip": {
```

```
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_tmm_stat_0_server_side_traffic_pkts_out{}",
            "interval": "",
            "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
            "range": true,
            "refId": "A"
          },
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_tmm_stat_2_server_side_traffic_pkts_in{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{tmmID}}-{{name}}-{{column}}",
            "range": true,
            "refId": "B"
          }
        ],
        "title": "F5 TMM Network Packets In/Out",
        "type": "timeseries"
      },
      {
        "datasource": {
          "type": "prometheus",
          "uid": "P1809F7CD0C75ACF3"
        },
        "fieldConfig": {
          "defaults": {
            "color": {
              "mode": "palette-classic"
            },
            "custom": {
              "axisBorderShow": false,
              "axisCenteredZero": false,
              "axisColorMode": "text",
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
              "lineWidth": 1,
              "pointSize": 5,
              "scaleDistribution": {
                "type": "linear"
              },
              "showPoints": "auto",
              "spanNulls": false,
              "stacking": {
                "group": "A",
                "mode": "none"
              },
              "thresholdsStyle": {
                "mode": "off"
              }
            },
            "mappings": [],
            "thresholds": {
              "mode": "absolute",
              "steps": [
                {
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            },
            "unit": "decbits"
          },
          "overrides": []
        },
        "gridPos": {
```

```
            "h": 8,
            "w": 5,
            "x": 10,
            "y": 11
          },
          "id": 26,
          "options": {
            "legend": {
              "calcs": [],
              "displayMode": "table",
              "placement": "bottom",
              "showLegend": true
            },
            "tooltip": {
              "hideZeros": false,
              "mode": "single",
              "sort": "none"
            }
          },
          "pluginVersion": "11.5.2",
          "targets": [
            {
              "datasource": {
                "type": "prometheus",
                "uid": "P1809F7CD0C75ACF3"
              },
              "editorMode": "code",
              "exemplar": true,
              "expr": "f5_tmm_interface_stat_1_2_counters_bytes_in{}",
              "interval": "",
              "legendFormat": "{{tmmID}}-{{column}}",
              "range": true,
              "refId": "A"
            },
            {
              "datasource": {
                "type": "prometheus",
                "uid": "P1809F7CD0C75ACF3"
              },
              "editorMode": "code",
              "exemplar": true,
              "expr": "f5_tmm_interface_stat_1_2_counters_bytes_out{}",
              "hide": false,
              "interval": "",
              "legendFormat": "{{tmmID}}-{{column}}",
              "range": true,
              "refId": "B"
            }
          ],
          "title": "F5 TMM Interface 1.2 Bytes In/Out",
          "type": "timeseries"
        },
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "fieldConfig": {
            "defaults": {
              "color": {
                "mode": "palette-classic"
              },
              "custom": {
                "axisBorderShow": false,
                "axisCenteredZero": false,
                "axisColorMode": "text",
                "axisLabel": "",
                "axisPlacement": "auto",
                "barAlignment": 0,
                "barWidthFactor": 0.6,
                "drawStyle": "line",
                "fillOpacity": 0,
                "gradientMode": "none",
                "hideFrom": {
                  "legend": false,
                  "tooltip": false,
                  "viz": false
                },
                "insertNulls": false,
                "lineInterpolation": "linear",
                "lineWidth": 1,
                "pointSize": 5,
                "scaleDistribution": {
                  "type": "linear"
                },
                "showPoints": "auto",
                "spanNulls": false,
                "stacking": {
                  "group": "A",
                  "mode": "none"
                },
                "thresholdsStyle": {
                  "mode": "off"
                }
              },
              "mappings": [],
              "thresholds": {
                "mode": "absolute",
                "steps": [
                  {
```

```
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            },
            "unit": "short"
          },
          "overrides": []
        },
        "gridPos": {
          "h": 8,
          "w": 5,
          "x": 15,
          "y": 11
        },
        "id": 28,
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "table",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_interface_stat_1_2_counters_pkts_in{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{tmmID}}-{{column}}",
            "range": true,
            "refId": "B"
          },
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_interface_stat_1_2_counters_pkts_out{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{tmmID}}-{{column}}",
            "range": true,
            "refId": "C"
          }
        ],
        "title": "F5 TMM Interface 1.2 Pkts In/Out",
        "type": "timeseries"
      }
    ],
    "title": "TMM",
    "type": "row"
  },
  {
    "collapsed": true,
    "gridPos": {
      "h": 1,
      "w": 24,
      "x": 0,
      "y": 2
    },
    "id": 35,
    "panels": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "P1809F7CD0C75ACF3"
        },
        "fieldConfig": {
          "defaults": {
            "color": {
              "mode": "palette-classic"
            },
            "custom": {
              "axisBorderShow": false,
              "axisCenteredZero": false,
              "axisColorMode": "text",
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
```

```
                "fillOpacity": 0,
                "gradientMode": "none",
                "hideFrom": {
                  "legend": false,
                  "tooltip": false,
                  "viz": false
                },
                "insertNulls": false,
                "lineInterpolation": "linear",
                "lineWidth": 1,
                "pointSize": 5,
                "scaleDistribution": {
                  "type": "linear"
                },
                "showPoints": "auto",
                "spanNulls": false,
                "stacking": {
                  "group": "A",
                  "mode": "none"
                },
                "thresholdsStyle": {
                  "mode": "off"
                }
              },
              "mappings": [],
              "thresholds": {
                "mode": "absolute",
                "steps": [
                  {
                    "color": "green",
                    "value": null
                  },
                  {
                    "color": "red",
                    "value": 80
                  }
                ]
              }
            },
            "overrides": []
          },
          "gridPos": {
            "h": 8,
            "w": 5,
            "x": 0,
            "y": 3
          },
          "id": 34,
          "options": {
            "legend": {
              "calcs": [],
              "displayMode": "table",
              "placement": "bottom",
              "showLegend": true
            },
            "tooltip": {
              "hideZeros": false,
              "mode": "single",
              "sort": "none"
            }
          },
          "pluginVersion": "11.5.2",
          "targets": [
            {
              "editorMode": "code",
              "expr": "f5_tmm_fw_context_stat_1_packets_reject{}",
              "legendFormat": "{{tmmID}}",
              "range": true,
              "refId": "A"
            }
          ],
          "title": "ACL Rejected",
          "type": "timeseries"
        }
      ],
      "title": "ACL",
      "type": "row"
    },
    {
      "collapsed": true,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 3
      },
      "id": 40,
      "panels": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "fieldConfig": {
            "defaults": {
              "color": {
                "mode": "palette-classic"
              },
              "custom": {
                "axisBorderShow": false,
```

```
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "barWidthFactor": 0.6,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 8,
        "w": 5,
        "x": 0,
        "y": 4
      },
      "id": 20,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "table",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "hideZeros": false,
          "mode": "single",
          "sort": "none"
        }
      },
      "pluginVersion": "11.5.2",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "prometheus"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_virtual_server_stat_blue_gatewayapi_l4_tcp_app_0_virtual_server_serverside_pkts_out{}",
          "interval": "",
          "legendFormat": "{{tmmID}}-{{column}}",
          "range": true,
          "refId": "A"
        },
        {
          "datasource": {
            "type": "prometheus",
            "uid": "prometheus"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_virtual_server_stat_blue_gatewayapi_l4_tcp_app_0_virtual_server_serverside_pkts_in{}",
          "hide": false,
          "interval": "",
          "legendFormat": "{{tmmID}}-{{column}}",
          "range": true,
          "refId": "B"
        }
      ],
      "title": "Virtual Server Packets In/Out",
```

```
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "P1809F7CD0C75ACF3"
      },
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "barWidthFactor": 0.6,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 8,
        "w": 6,
        "x": 5,
        "y": 4
      },
      "id": 24,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "table",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "hideZeros": false,
          "mode": "single",
          "sort": "none"
        }
      },
      "pluginVersion": "11.5.2",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_virtual_server_stat_blue_gatewayapi_l4_tcp_app_0_virtual_server_clientside_pkts_in{}",
          "interval": "",
          "legendFormat": "{{tmmID}}-{{column}}",
          "range": true,
          "refId": "A"
        },
        {
          "datasource": {
```

```
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_virtual_server_stat_blue_gatewayapi_l4_tcp_app_0_virtual_server_clientside_pkts_out{}",
          "hide": false,
          "interval": "",
          "legendFormat": "{{tmmID}}-{{column}}",
          "range": true,
          "refId": "B"
        }
      ],
      "title": "Virtual Server Client Packets In/Out",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "P1809F7CD0C75ACF3"
      },
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "barWidthFactor": 0.6,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 8,
        "w": 6,
        "x": 11,
        "y": 4
      },
      "id": 30,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "table",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "hideZeros": false,
          "mode": "single",
          "sort": "none"
        }
      },
      "pluginVersion": "11.5.2",
      "targets": [
        {
```

```
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_virtual_server_stat_blue_gatewayapi_l4_tcp_app_0_virtual_server_clientside_cur_conns{}",
            "interval": "",
            "legendFormat": "{{tmmID}}-{{column}}",
            "range": true,
            "refId": "A"
          },
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_virtual_server_stat_blue_gatewayapi_l4_tcp_app_0_virtual_server_clientside_tot_conns{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{namespace}}-{{tmmID}}-{{column}}",
            "range": true,
            "refId": "B"
          }
        ],
        "title": "Virtual Server Connections",
        "type": "timeseries"
      },
      {
        "datasource": {
          "type": "prometheus",
          "uid": "P1809F7CD0C75ACF3"
        },
        "fieldConfig": {
          "defaults": {
            "color": {
              "mode": "palette-classic"
            },
            "custom": {
              "axisBorderShow": false,
              "axisCenteredZero": false,
              "axisColorMode": "text",
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
              "lineWidth": 1,
              "pointSize": 5,
              "scaleDistribution": {
                "type": "linear"
              },
              "showPoints": "auto",
              "spanNulls": false,
              "stacking": {
                "group": "A",
                "mode": "none"
              },
              "thresholdsStyle": {
                "mode": "off"
              }
            },
            "mappings": [],
            "thresholds": {
              "mode": "absolute",
              "steps": [
                {
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            },
            "unit": "decbits"
          },
          "overrides": []
        },
        "gridPos": {
          "h": 8,
          "w": 5,
          "x": 17,
          "y": 4
        },
        "id": 18,
        "options": {
          "legend": {
```

```
            "calcs": [],
            "displayMode": "table",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_virtual_server_stat_blue_gatewayapi_l4_tcp_app_0_virtual_server_serverside_bytes_in{}",
            "interval": "",
            "legendFormat": "{{tmmID}}-{{column}}",
            "range": true,
            "refId": "A"
          },
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_virtual_server_stat_blue_gatewayapi_l4_tcp_app_0_virtual_server_serverside_bytes_out{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{tmmID}}-{{column}}",
            "range": true,
            "refId": "B"
          }
        ],
        "title": "Virtual Server Bytes In/Out",
        "type": "timeseries"
      }
    ],
    "title": "Blue Tenant",
    "type": "row"
  },
  {
    "collapsed": true,
    "gridPos": {
      "h": 1,
      "w": 24,
      "x": 0,
      "y": 4
    },
    "id": 39,
    "panels": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "P1809F7CD0C75ACF3"
        },
        "fieldConfig": {
          "defaults": {
            "color": {
              "mode": "palette-classic"
            },
            "custom": {
              "axisBorderShow": false,
              "axisCenteredZero": false,
              "axisColorMode": "text",
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
              "lineWidth": 1,
              "pointSize": 5,
              "scaleDistribution": {
                "type": "linear"
              },
              "showPoints": "auto",
              "spanNulls": false,
              "stacking": {
                "group": "A",
                "mode": "none"
              },
              "thresholdsStyle": {
                "mode": "off"
              }
```

```
            },
            "mappings": [],
            "thresholds": {
              "mode": "absolute",
              "steps": [
                {
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            },
            "unit": "short"
          },
          "overrides": []
        },
        "gridPos": {
          "h": 8,
          "w": 5,
          "x": 0,
          "y": 5
        },
        "id": 41,
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "table",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_virtual_server_stat_red_gatewayapi_l4_tcp_app_0_virtual_server_serverside_pkts_out{}",
            "interval": "",
            "legendFormat": "{{tmmID}}-{{column}}",
            "range": true,
            "refId": "A"
          },
          {
            "datasource": {
              "type": "prometheus",
              "uid": "prometheus"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_virtual_server_stat_red_gatewayapi_l4_tcp_app_0_virtual_server_serverside_pkts_in{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{tmmID}}-{{column}}",
            "range": true,
            "refId": "B"
          }
        ],
        "title": "Virtual Server Packets In/Out",
        "type": "timeseries"
      },
      {
        "datasource": {
          "type": "prometheus",
          "uid": "P1809F7CD0C75ACF3"
        },
        "fieldConfig": {
          "defaults": {
            "color": {
              "mode": "palette-classic"
            },
            "custom": {
              "axisBorderShow": false,
              "axisCenteredZero": false,
              "axisColorMode": "text",
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
```

```
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 8,
        "w": 6,
        "x": 5,
        "y": 5
      },
      "id": 42,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "table",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "hideZeros": false,
          "mode": "single",
          "sort": "none"
        }
      },
      "pluginVersion": "11.5.2",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_virtual_server_stat_red_gatewayapi_l4_tcp_app_0_virtual_server_clientside_pkts_in{}",
          "interval": "",
          "legendFormat": "{{tmmID}}-{{column}}",
          "range": true,
          "refId": "A"
        },
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "editorMode": "code",
          "exemplar": true,
          "expr": "f5_tmm_virtual_server_stat_red_gatewayapi_l4_tcp_app_0_virtual_server_clientside_pkts_out{}",
          "hide": false,
          "interval": "",
          "legendFormat": "{{tmmID}}-{{column}}",
          "range": true,
          "refId": "B"
        }
      ],
      "title": "Virtual Server Client Packets In/Out",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "P1809F7CD0C75ACF3"
      },
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
```

```
              "axisLabel": "",
              "axisPlacement": "auto",
              "barAlignment": 0,
              "barWidthFactor": 0.6,
              "drawStyle": "line",
              "fillOpacity": 0,
              "gradientMode": "none",
              "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
              },
              "insertNulls": false,
              "lineInterpolation": "linear",
              "lineWidth": 1,
              "pointSize": 5,
              "scaleDistribution": {
                "type": "linear"
              },
              "showPoints": "auto",
              "spanNulls": false,
              "stacking": {
                "group": "A",
                "mode": "none"
              },
              "thresholdsStyle": {
                "mode": "off"
              }
            },
            "mappings": [],
            "thresholds": {
              "mode": "absolute",
              "steps": [
                {
                  "color": "green",
                  "value": null
                },
                {
                  "color": "red",
                  "value": 80
                }
              ]
            },
            "unit": "short"
          },
          "overrides": []
        },
        "gridPos": {
          "h": 8,
          "w": 6,
          "x": 11,
          "y": 5
        },
        "id": 43,
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "table",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
          }
        },
        "pluginVersion": "11.5.2",
        "targets": [
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_virtual_server_stat_red_gatewayapi_l4_tcp_app_0_virtual_server_clientside_cur_conns{}",
            "interval": "",
            "legendFormat": "{{tmmID}}-{{column}}",
            "range": true,
            "refId": "A"
          },
          {
            "datasource": {
              "type": "prometheus",
              "uid": "P1809F7CD0C75ACF3"
            },
            "editorMode": "code",
            "exemplar": true,
            "expr": "f5_tmm_virtual_server_stat_red_gatewayapi_l4_tcp_app_0_virtual_server_clientside_tot_conns{}",
            "hide": false,
            "interval": "",
            "legendFormat": "{{namespace}}-{{tmmID}}-{{column}}",
            "range": true,
            "refId": "B"
          }
        ],
        "title": "Virtual Server Connections",
        "type": "timeseries"
      },
```

```
        {
          "datasource": {
            "type": "prometheus",
            "uid": "P1809F7CD0C75ACF3"
          },
          "fieldConfig": {
            "defaults": {
              "color": {
                "mode": "palette-classic"
              },
              "custom": {
                "axisBorderShow": false,
                "axisCenteredZero": false,
                "axisColorMode": "text",
                "axisLabel": "",
                "axisPlacement": "auto",
                "barAlignment": 0,
                "barWidthFactor": 0.6,
                "drawStyle": "line",
                "fillOpacity": 0,
                "gradientMode": "none",
                "hideFrom": {
                  "legend": false,
                  "tooltip": false,
                  "viz": false
                },
                "insertNulls": false,
                "lineInterpolation": "linear",
                "lineWidth": 1,
                "pointSize": 5,
                "scaleDistribution": {
                  "type": "linear"
                },
                "showPoints": "auto",
                "spanNulls": false,
                "stacking": {
                  "group": "A",
                  "mode": "none"
                },
                "thresholdsStyle": {
                  "mode": "off"
                }
              },
              "mappings": [],
              "thresholds": {
                "mode": "absolute",
                "steps": [
                  {
                    "color": "green",
                    "value": null
                  },
                  {
                    "color": "red",
                    "value": 80
                  }
                ]
              },
              "unit": "decbits"
            },
            "overrides": []
          },
          "gridPos": {
            "h": 8,
            "w": 5,
            "x": 17,
            "y": 5
          },
          "id": 44,
          "options": {
            "legend": {
              "calcs": [],
              "displayMode": "table",
              "placement": "bottom",
              "showLegend": true
            },
            "tooltip": {
              "hideZeros": false,
              "mode": "single",
              "sort": "none"
            }
          },
          "pluginVersion": "11.5.2",
          "targets": [
            {
              "datasource": {
                "type": "prometheus",
                "uid": "prometheus"
              },
              "editorMode": "code",
              "exemplar": true,
              "expr": "f5_tmm_virtual_server_stat_red_gatewayapi_l4_tcp_app_0_virtual_server_serverside_bytes_in{}",
              "interval": "",
              "legendFormat": "{{tmmID}}-{{column}}",
              "range": true,
              "refId": "A"
            },
            {
              "datasource": {
                "type": "prometheus",
                "uid": "prometheus"
```

```
                },
                "editorMode": "code",
                "exemplar": true,
                "expr": "f5_tmm_virtual_server_stat_red_gatewayapi_l4_tcp_app_0_virtual_server_serverside_bytes_out{}",
                "hide": false,
                "interval": "",
                "legendFormat": "{{tmmID}}-{{column}}",
                "range": true,
                "refId": "B"
              }
            ],
            "title": "Virtual Server Bytes In/Out",
            "type": "timeseries"
          }
        ],
        "title": "Red Tenant",
        "type": "row"
      }
    ],
    "preload": false,
    "refresh": "",
    "schemaVersion": 40,
    "tags": [],
    "templating": {
      "list": []
    },
    "time": {
      "from": "now-1h",
      "to": "now"
    },
    "timepicker": {},
    "timezone": "",
    "title": "F5 BNK Dashboard",
    "uid": null,
    "id": null,
    "version": 4,
    "weekStart": ""
  }
}
```


Import the dashboard into Grafana

Bash

```
host# kubectl -n grafana port-forward svc/grafana 3000:3000 &
host# curl -X POST -H 'Content-Type: application/json' -d @grafana-dashboard.json http://admin:admin@localhost:3000/api/dashboards/db
```

5.8 7. Install BIG-IP Next for Kubernetes Operator in default namespace

The operator helps in installing BIG-IP Next for Kubernetes software. It requires two Custom Resources to be defined for the installation. **`SPKInfrastructure`** to describe dataplane infrastructure connections, and **`SPKInstance`** which declares the state and configuration of the BNK product installation.

5.8.1 Install the Operator chart

Bash

```
host# helm install orchestrator oci://repo.f5.com/charts/orchestrator \
        --version v0.0.25-0.0.96 \
        --set global.imagePullSecrets[0].name=far-secret \
        --set image.repository=repo.f5.com/images \
        --set image.pullPolicy=Always
```

5.8.2 `SPKInfrastructure` Custom Resource

`SPKInfrastructure` resource includes refernces to the Network Attachment Definitions created earlier, and the resources provisioned for these networks as configured in the SR-IOV device plugin section.

The `SPKInfrastructure` resources is defined here infrastructure-cr.yaml.

Show SPKInfrastructure content

YAML

```yaml
apiVersion: charts.k8s.f5net.com/v1alpha1
kind: SPKInfrastructure
metadata:
  name: bnk-dpu-infra
spec:
  networkAttachment:
  - name: default/sf-external
  - name: default/sf-internal
  platformType: other
  hugepages: true
  sriovResources:
    nvidia.com/bf3_p0_sf: "1"
    nvidia.com/bf3_p1_sf: "1"
  wholeClusterMode: "enabled"
  calicoRouter: "default"
  egress:
    json:
      ipPoolCidrInfo:
        cidrList:
        - name: vlan_cidr
          value: "192.168.20.0/24"
        - name: vlan_ipv6_cidr
          value: "2001::192:168:20:0/112"
        ipPoolList:
        - name: default-ipv4-ippool
          value: "10.244.0.0/16"
```

5.8.3 Install required Otel Certificates

Otel service requires certificates to be installed with specific name. These certs will be used for TLS communication between Otel and Prometheus.

Certificate requests for this lab can be found at otel-certs.yaml.

Show Otel Certificates content

YAML

```yaml
---
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: external-otelsvr
spec:
  subject:
    countries:
      - US
    provinces:
      - Washington
    localities:
      - Seattle
    organizations:
      - F5 Networks
    organizationalUnits:
      - PD
  emailAddresses:
    - clientcert@f5net.com
  commonName: f5net.com
  # SecretName is the name of the secret resource that will be automatically created and managed by this Certificate resource.
  # It will be populated with a private key and certificate, signed by the denoted issuer.
  secretName: external-otelsvr-secret
  # IssuerRef is a reference to the issuer for this certificate.
  issuerRef:
    group: cert-manager.io
    kind: ClusterIssuer
    name: bnk-ca-cluster-issuer
  # Lifetime of the Certificate is 360 days.
  duration: 8640h
  privateKey:
    rotationPolicy: Always
    encoding: PKCS1
    algorithm: RSA
    size: 4096
  revisionHistoryLimit: 10
---
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: external-f5ingotelsvr
spec:
  subject:
    countries:
      - US
    provinces:
      - Washington
    localities:
      - Seattle
    organizations:
      - F5 Networks
    organizationalUnits:
      - PD
  emailAddresses:
    - clientcert@f5net.com
  commonName: f5net.com
  secretName: external-f5ingotelsvr-secret
  issuerRef:
    group: cert-manager.io
    kind: ClusterIssuer
    name: bnk-ca-cluster-issuer
  duration: 8640h
  privateKey:
    rotationPolicy: Always
    encoding: PKCS1
    algorithm: RSA
    size: 4096
  revisionHistoryLimit: 10
```

Apply the certificates to the default namespace.

Bash

```
host# kubectl apply -f otel-certs.yaml
```

5.8.4 `SPKInstance` Custom Resource

Download or copy the instance-cr-otel.yaml file and modify the `jwt:` with your license token obtained from MyF5.

Show SPKInstance content

YAML

```yaml
apiVersion: charts.k8s.f5net.com/v1alpha1
kind: SPKInstance
metadata:
  name: bnk-dpu
  namespace: default
spec:
  controller:
    watchNamespace: red,blue
  cwc:
    persistence:
      enabled: true
      size: 20Gi
    cpclConfig:
      jwt: <replace-with-jwt-token>
      operationMode: connected
  global:
    certmgr:
      issuerRef:
        group: cert-manager.io
        kind: ClusterIssuer
        name: bnk-ca-cluster-issuer
    imagePullSecrets:
    - name: far-secret
    imageRepository: repo.f5.com/images
    logging:
      fluentbitSidecar:
        enabled: true
        fluentd:
          host: f5-toda-fluentd.f5-utils.svc.cluster.local
          port: "54321"
  spkInfrastructure: bnk-dpu-infra
  spkManifest: unused
  afm:
    enabled: true
    pccd:
      enabled: true
      blob:
        maxFwBlobSizeMb: "512"
        maxNatBlobSizeMb: "512"
  tmm:
    replicaCount: 1
    nodeAssign:
      nodeSelector:
        app: f5-tmm
      tolerations:
        - key: "dpu"
          value: "true"
          operator: "Equal"
    palCPUSet: "8-15"
    usePhysMem: true
    tmmMapresHugepages: 6144
    resources:
      limits:
        cpu: "8"
        hugepages-2Mi: 13Gi
        memory: 2Gi
    debug:
      enabled: true
      resources:
        limits:
          cpu: 200m
          memory: 100Mi
        requests:
          cpu: 200m
          memory: 100Mi
    xnetDPDKAllow:
    - auxiliary:mlx5_core.sf.4,dv_flow_en=2
    - auxiliary:mlx5_core.sf.5,dv_flow_en=2
    blobd:
      enabled: true
      resources:
        limits:
          cpu: "1"
          memory: "1Gi"
        requests:
          cpu: "1"
          memory: "1Gi"
    dynamicRouting:
      enabled: false
      configMapName: spk-bgp
    tmrouted:
      resources:
        limits:
          cpu: "300m"
          memory: "512Mi"
        requests:
          cpu: "300m"
          memory: "512Mi"
    tmmRouting:
      resources:
        limits:
          cpu: "700m"
          memory: "512Mi"
```

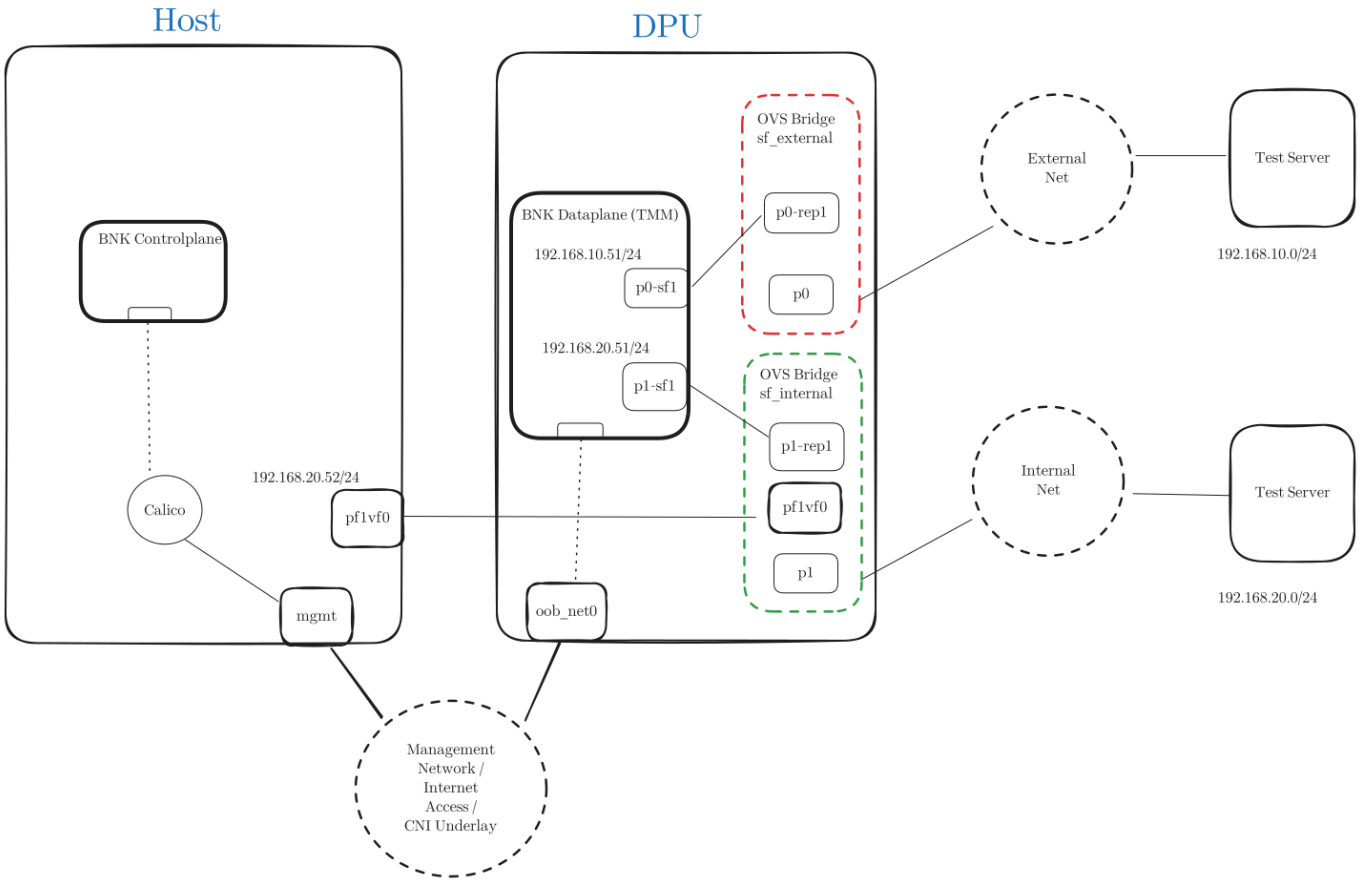
```
        requests:
          cpu: "700m"
          memory: "512Mi"
    sessiondb:
      useExternalStorage: "true"
```

Ensure that all pods in `default` and `f5-utils` namespaces are healthy. This can take up to 10 minutes.

2025-02-25

6. Lab Configuration

The BIG-IP Next for Kubernetes dataplane component (TMM) executes all networking stack operations entirely in user space and connects to Scalable Function (SF) interfaces using the DPDK driver. The TMM's networking configuration is managed through Custom Resources (CRs), and we will utilize these resources to set up the installation as outlined in the lab diagram.



6.1 Configure the Underlay Network

The underlay network consists of IP addresses directly connected to the physical network segments or infrastructure. These addresses are configured using the `F5SPKVlan` Custom Resource (CR).

The `F5SPKVlan` resources below configure two **untagged VLANs**: - **internal**: Connected to the internal network segment. - **external**: Connected to the external network segment.

The IPv4 and IPv6 address lists specify the **underlay IP addresses** reachable through these network segments. Each address from the list is assigned to one instance of TMM. The addresses list must include enough IP addresses enough for the number of TMM instances planned. For example if we have 3 DPU nodes, we require **at least three IP addresses in the list**.

Example

For example, if there are 3 DPU nodes in the deployment, you will need at least three IP addresses in the list.

To apply this configuration, download, modify if needed, and apply the VLAN configuration file: bnk-vlans.yaml.

Show content of bnk-vlans.yaml

YAML

```yaml
---
apiVersion: "k8s.f5net.com/v1"
kind: F5SPKVlan
metadata:
  name: internal
spec:
  name: internal
  interfaces:
    - "1.2"
  tag: 0
  selfip_v4s:
    - 192.168.20.201
    - 192.168.20.202
    - 192.168.20.203
  prefixlen_v4: 24
  selfip_v6s:
    - 2001::192:168:20:201
    - 2001::192:168:20:202
    - 2001::192:168:20:203
  prefixlen_v6: 112
  auto_lasthop: "AUTO_LASTHOP_ENABLED"
  internal: true
---
apiVersion: "k8s.f5net.com/v1"
kind: F5SPKVlan
metadata:
  name: external
spec:
  name:  external
  interfaces:
    - "1.1"
  tag: 0
  selfip_v4s:
    - 192.168.10.201
    - 192.168.10.202
    - 192.168.10.203
  prefixlen_v4: 24
  selfip_v6s:
    - 2001::192:168:10:201
    - 2001::192:168:10:202
    - 2001::192:168:10:203
  prefixlen_v6: 112
  auto_lasthop: "AUTO_LASTHOP_ENABLED"
```

Note

- A VLAN tag value of `0` (`tag: 0`) indicates an **untagged VLAN**. If tagging is required, replace `0` with the desired VLAN tag.
- Each interface can have **only one untagged VLAN**, while **multiple tagged VLANs** are allowed per interface.
- VLAN tags must be **unique across all interfaces**. The same VLAN tag cannot be assigned to more than one interface.

When network interfaces, such as **Scalable Functions (SFs)**, are connected to TMM (via the **Network Attachment Definition**), they are assigned **index numbers** based on the order in which they are configured. For example, interfaces are indexed as `1.1`, `1.2`, and so forth. In the `F5SPKVlan` configuration shown above, note the `interfaces` section referencing `1.1` and `1.2`.

Mapping Interfaces in This Lab Guide

This description provides a simplified overview of interface naming for clarity specific to this lab guide.

- **Network Attachments**:
- Configured with the names `sf-internal` and `sf-external`.
- **Interface Mapping in `SPKInfrastructure` CR**:
- The `SPKInfrastructure` Custom Resource connects these interfaces as follows:

YAML

```
networkAttachment:
- name: default/sf-external
- name: default/sf-internal
```

The order of the `networkAttachment` section determines the interface assignment:

- **`sf-external`**: Assigned `1.1`
- **`sf-internal`**: Assigned `1.2`

To verify the network configuration, check the status of the `F5SPKVlan` Custom Resources:

Bash Session

```
host# kubectl get f5-spk-vlan
NAME       READY   MESSAGE                                AGE
external   True    CR config sent to all grpc endpoints   30h
internal   True    CR config sent to all grpc endpoints   30h
```

6.2 Configure Calico CNI to allow VXLAN from BNK

In this lab we will build VXLAN networks between the host node and the TMM to segregate tenants based on namespaces. Calico CNI as installed and configured in this lab will create rules to deny VXLAN traffic from different external sources than node list, and thus we need to explicitly allow the TMM VXLAN traffic to pass through host node to workload.

In order to achieve that we need to patch calico's felixconfiguration to allow TMM's internal VLAN IP addresses.

Tip

Make sure the IP addresses match what is configured in the `F5SPKVlan` internal CR.

Bash Session

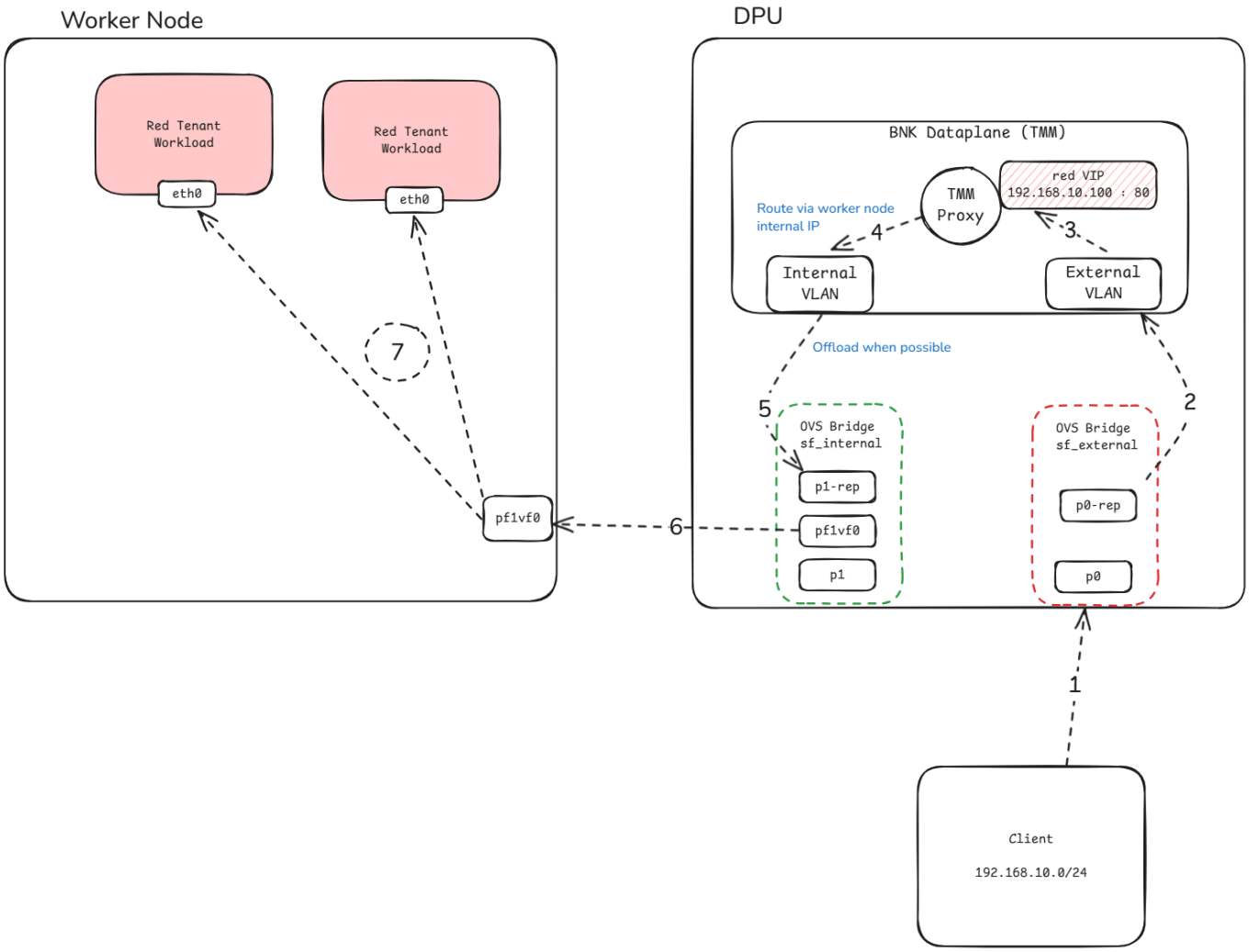
```
host# kubectl patch felixconfiguration default --type='merge' -p='{"spec": {"externalNodesList": ["192.168.20.201", "192.168.20.202", "192.168.20.203"]}}'
```

2025-02-25

7. 6. F5 BNK Ingress Configuration

BIG-IP Next for Kubernetes is also a controller for Kubernetes Gateway API. In the following example we will deploy a simple Nginx service in the **red** tenant namespace and advertise it's service to the infrastructure.

The following diagram represents the service ingress path.



Apply the nginx-deployment.yaml file to deploy Nginx service in the red namespace.

Show content of nginx-deployment.yaml

YAML

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: red
  labels:
    app: nginx-tcp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-tcp
  template:
    metadata:
      labels:
        app: nginx-tcp
    spec:
      containers:
      - name: nginx-tcp
        image: nginx:latest
        ports:
        - containerPort: 80
        imagePullPolicy: IfNotPresent
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-app-svc
  namespace: red
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx-tcp
```

Then expose the service to network by using Kubernetes Gateway API resources.

Apply the nginx-gw-api.yaml file to expose the Nginx service on virtual server IP `192.168.10.100` port `80` as the diagram suggested.

Show content of nginx-gw-api.yaml

YAML

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: f5-gateway-class
  namespace: red
spec:
  controllerName: "f5.com/f5-gateway-controller"
  description: "F5 BIG-IP Kubernetes Gateway"
---
apiVersion: gateway.k8s.f5net.com/v1
kind: Gateway
metadata:
  name: my-l4route-tcp-gateway
  namespace: red
spec:
  addresses:
  - type: "IPAddress"
    value: 192.168.10.100
  gatewayClassName: f5-gateway-class
  listeners:
  - name: nginx
    protocol: TCP
    port: 80
    allowedRoutes:
      kinds:
      - kind: L4Route
---
apiVersion: gateway.k8s.f5net.com/v1
kind: L4Route
metadata:
  name: l4-tcp-app
  namespace: red
spec:
  protocol: TCP
  parentRefs:
  - name: my-l4route-tcp-gateway
    sectionName: nginx
  rules:
  - backendRefs:
    - name: nginx-app-svc
      namespace: red
      port: 80
```

Note

For simplicity an IP address from the same subnet as the test server/client was used but this can be any IP address as long as the server/client is properly routed through one of the TMM's VLAN addresses.

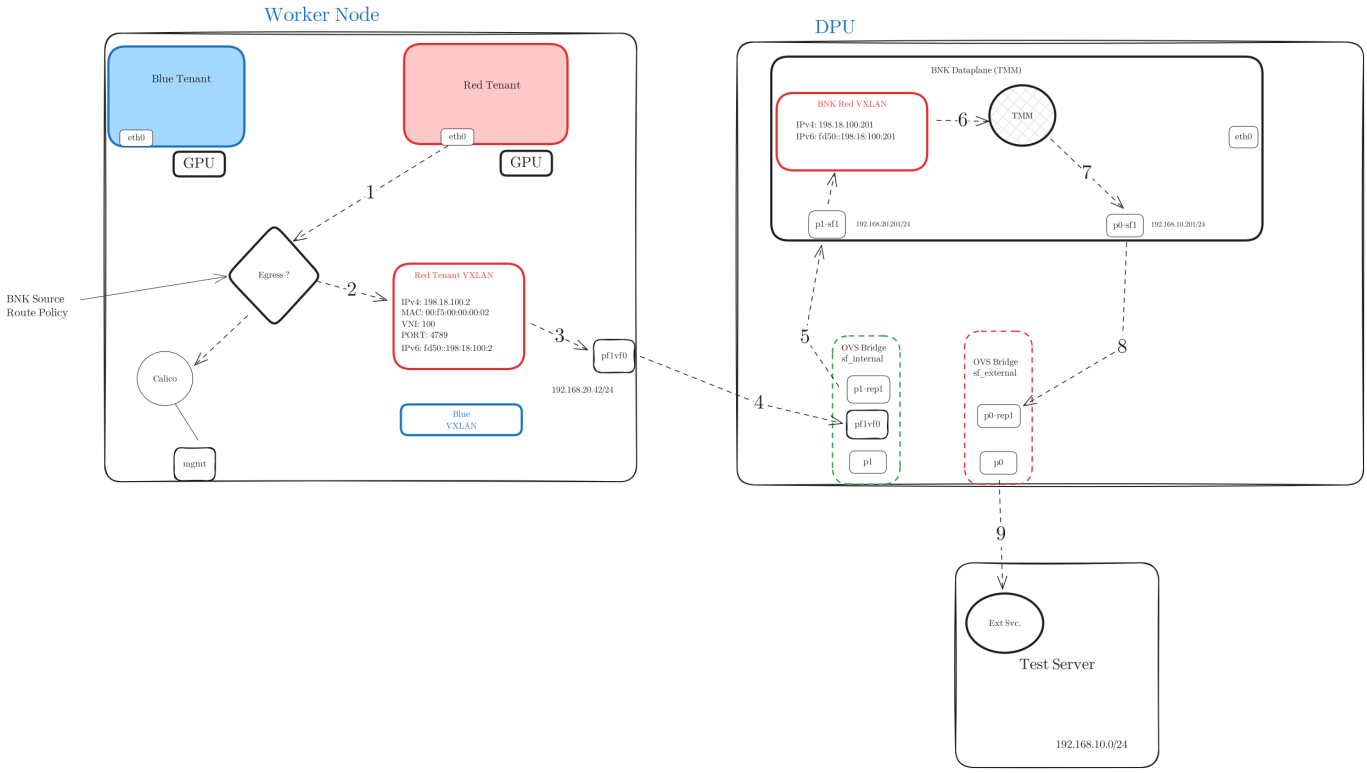
Note

In this configuration, the TMM will use SNAT-AUTOMAP feature which means it will SNAT external client IP addresses when communicating with backend endpoints with TMM's own IP address *not* an address from snatpool.

2025-02-25

8. BNK Egress

This lab guide assumes there will be two namespaces for tenant workload **red** and **blue** and that their egress/ingress is configured through VXLAN overlay. The following diagram shows tenant VXLAN config with focus on the **red** tenant knowing that blue tenant would be the same.



8.1 Configure VXLAN overlay

To configure this we use `F5SPKVxlan` CR which establishes the overlay configurations to the host, a `F5SPKSnatpool` CR to set IP addresses used for SNATing egress traffic towards the network infrastructure, and `F5SPKEgress` CR that assigns the egress rules for namespace to specific VXLAN.

The following `F5SPKVxlan` CRs configures two VXLANs **red** with VNI 100 and **blue** with VNI 200

NOTE: The virtual function created on host on PF1 is assumed to be `enp83s0f1v0` in this guide. Replace every instance of `enp83s0f1v0` with the actual configured host-side virtual function.

NOTE: The `remote_nodes` represent the host nodes only. Modify the list to properly reflect cluster node names and configured IP addresses as required. DPU nodes are not required here since workload is only expected on the host.

Apply bnk-vxlan.yaml to create VXLAN tunnels for red and blue namespaces.

Show bnk-vxlan.yaml content

YAML

```yaml
---
apiVersion: "k8s.f5net.com/v1"
kind: F5SPKVxlan
metadata:
  name: "red"
spec:
  name: "red"
  port: 4789
  key: 100
  # Interface name on host nodes that is used for underlay.
  # This is the previously configured Virtual Functionon PF1.
  remote_interface_name: "enp83s0f1v0"
  # Host nodes
  remote_nodes:
     # host node name in Kubernetes cluster.
   - node_name: "host-1"
     # Underlay IP address as configured on virtual function.
     # Change if different in your infrastructure.
     node_ip: "192.168.20.41"
     # Mac address and IP addresses that will be assigned to the
     # Host side VXLAN overlay interface.
     peer_mac: "00:f5:00:00:00:02"
     peerip_v4: "198.18.100.1"
     peerip_v6: "fd50::192:18:100:1"
   - node_name: "host-2"
     node_ip: "192.168.20.42"
     peer_mac: "00:f5:00:00:00:03"
     peerip_v4: "198.18.100.2"
     peerip_v6: "fd50::192:18:100:2"
   - node_name: "host-3"
     node_ip: "192.168.20.43"
     peer_mac: "00:f5:00:00:00:04"
     peerip_v4: "198.18.100.3"
     peerip_v6: "fd50::192:18:100:3"
  local_ips:
   - "192.168.20.201"
   - "192.168.20.202"
   - "192.168.20.203"
  selfip_v4s:
   - "198.18.100.201"
   - "198.18.100.202"
   - "198.18.100.203"
  prefixlen_v4: 24
  selfip_v6s:
   - "fd50::192:18:100:201"
   - "fd50::192:18:100:202"
   - "fd50::192:18:100:203"
  prefixlen_v6: 112
---
apiVersion: "k8s.f5net.com/v1"
kind: F5SPKVxlan
metadata:
  name: "blue"
spec:
  name: "blue"
  port: 4789
  key: 200
  remote_interface_name: "enp83s0f1v0"
  remote_nodes:
   - node_name: "host-1"
     node_ip: "192.168.10.41"
     peer_mac: "00:f5:01:00:00:02"
     peerip_v4: "198.18.200.1"
     peerip_v6: "fd50::192:18:200:1"
   - node_name: "host-2"
     node_ip: "192.168.10.42"
     peer_mac: "00:f5:01:00:00:03"
     peerip_v4: "198.18.200.2"
     peerip_v6: "fd50::192:18:200:2"
   - node_name: "host-3"
     node_ip: "192.168.10.43"
     peer_mac: "00:f5:01:00:00:04"
     peerip_v4: "198.18.200.3"
     peerip_v6: "fd50::192:18:200:3"
  local_ips:
   - "192.168.20.201"
   - "192.168.20.202"
   - "192.168.20.203"
  selfip_v4s:
   - "198.18.200.201"
   - "198.18.200.202"
   - "198.18.200.203"
  prefixlen_v4: 24
  selfip_v6s:
   - "fd50::192:18:200:201"
   - "fd50::192:18:200:202"
   - "fd50::192:18:200:203"
  prefixlen_v6: 112
```

8.2 Configure SNATPool

the `addressList` section is a list of lists of SNAT IP addresses that are assigned to each TMM. Since we have 3 TMMs here, we will create 3 lists one for each TMM.

The SNAT addresess are unique per TMM. And they are picked based on the closest IP address to the nexthop (gateway or direct network) for intended destination.

Apply bnk-snatpool.yaml to create SNAT addresses for workload in red and blue namespace.

Show bnk-snatpool.yaml content

YAML

```yaml
---
apiVersion: "k8s.f5net.com/v1"
kind: F5SPKSnatpool
metadata:
  name: "red-snat"
spec:
  name: "red-snat"
  addressList:
    - - 192.168.10.221
      - 2001::192:168:10:221
      - 192.168.20.221
      - 2001::192:168:20:221
    - - 192.168.10.222
      - 2001::192:168:10:222
      - 192.168.20.222
      - 2001::192:168:20:222
    - - 192.168.10.223
      - 2001::192:168:10:223
      - 192.168.20.223
      - 2001::192:168:20:223
---
apiVersion: "k8s.f5net.com/v1"
kind: F5SPKSnatpool
metadata:
  name: "blue-snat"
spec:
  name: "blue-snat"
  addressList:
    - - 192.168.10.231
      - 2001::192:168:10:231
      - 192.168.20.231
      - 2001::192:168:20:231
    - - 192.168.10.232
      - 2001::192:168:10:232
      - 192.168.20.232
      - 2001::192:168:20:232
    - - 192.168.10.233
      - 2001::192:168:10:233
      - 192.168.20.233
      - 2001::192:168:20:233
```

8.3 Configure `F5SPKEgress` to assign tenants egress to their prespective VXLAN

Now we can apply bnk-egress.yaml egress path in TMM with refernce to VXLAN and SNAT pools we just created.

Show bnk-egress.yaml content

YAML

```yaml
---
apiVersion: k8s.f5net.com/v3
kind: F5SPKEgress
metadata:
  name: red-egress
spec:
  dualStackEnabled: true
  snatType: SRC_TRANS_SNATPOOL
  egressSnatpool: red-snat
  pseudoCNIConfig:
    namespaces:
      - red
    # Routing default pod interface eth0.
    # Assumes pod does not have additional interfaces configured.
    appPodInterface: eth0
    # Name of VXLAN interface created on the host
    # This is basically the same name as the VXLAN name in CR.
    appNodeInterface: red
    # Name of VXLAN interface on TMM which is the VXLAN CR name.
    vlanName: red
---
apiVersion: k8s.f5net.com/v3
kind: F5SPKEgress
metadata:
  name: blue-egress
spec:
  dualStackEnabled: true
  snatType: SRC_TRANS_SNATPOOL
  egressSnatpool: blue-snat
  pseudoCNIConfig:
    namespaces:
      - blue
    appPodInterface: eth0
    appNodeInterface: blue
    vlanName: blue
```

2025-02-25

https://github.com/f5devcentral/f5-bnk-nvidia-bf3-installations