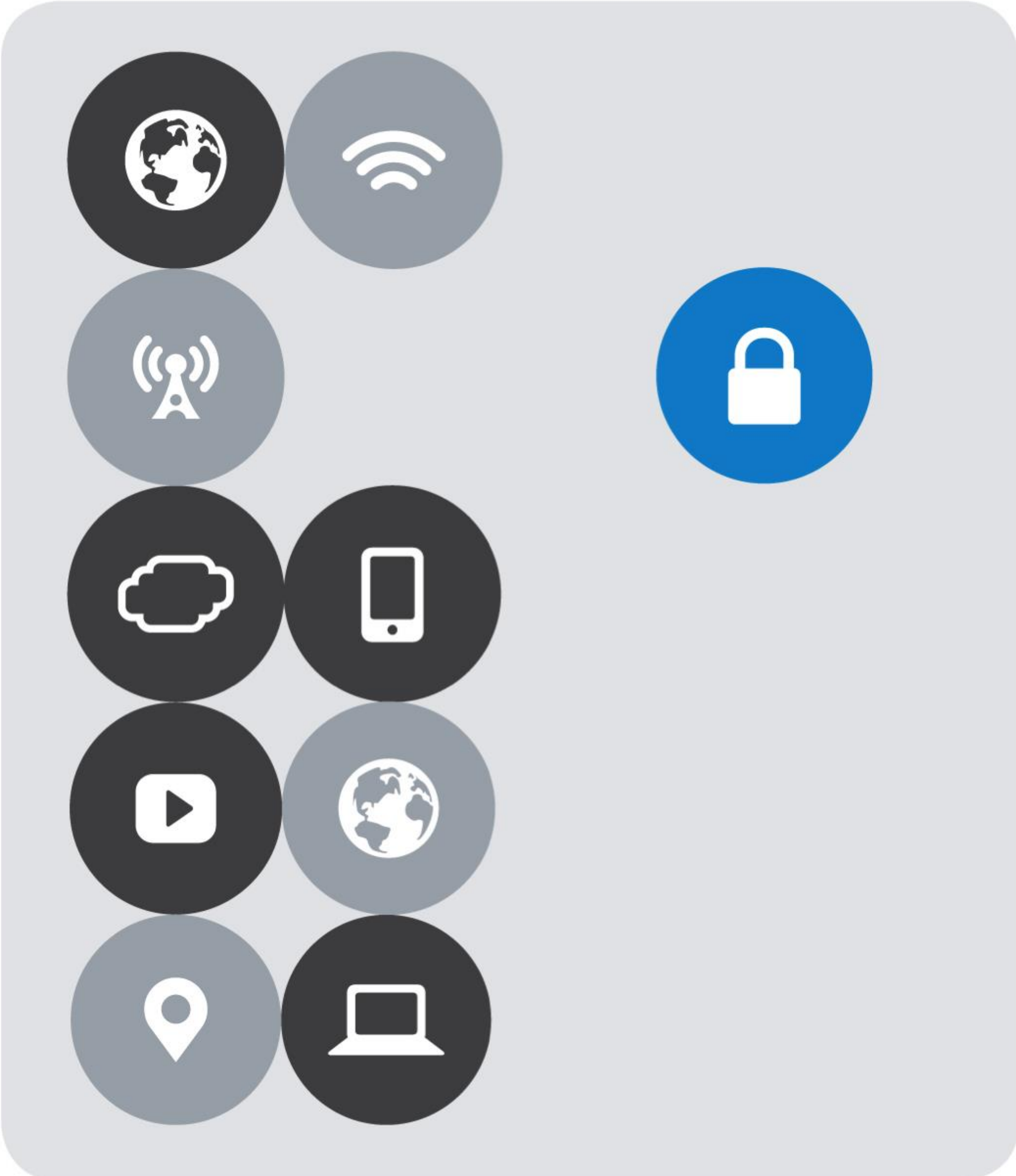
Migration Guide for VMware to Red Hat OpenShift

BIG-IP and vSphere networking



Version History

Date	Version	Author	Notes
Jul 2024	0.1	Ulises Alonso Camaró	Includes the sections: - Overview - Network considerations - Migration procedure - Additional information
Sep 2024	0.1r	Matt Mabis	Initial review
Sep 2024	0.2	Ulises Alonso Camaró	Changes: - Modifications for the initial review - Preparatory works section - Some additional information KB articles added - Added front page - Added MTU settings to NADs
Sep 2024	0.2r	Matt Mabis	Procedure review
Sep 2024	0.3	Ulises Alonso Camaró	Changes: - Modifications from the procedure review - Added Introduction section - Modifications to the overview section - Highlighted customizable text - Several minor edits thorough - Procedure review and eliminated device cluster re-creation
Sep 2024	1.0	Ulises Alonso Camaró	First version published
Oct 2024	1.1	Ulises Alonso Camaró	Removed duplication of NADs used Added 6.5 Testing the MTU of the interfaces Minor changes
Dec 2024	1.2	Ulises Alonso Camaró	Title change

1 Introduction

OpenShift® Virtualization [1] [2] is Red Hat´s® offering for KubeVirt virtualization, a solid virtualization platform for VM workloads.

This document covers the case of migrating workloads and BIG-IP Virtual Edition (VE) instances from VMware® with vSphere® networking to OpenShift Virtualization. The guide is aligned with the OpenShift Virtualization Reference Implementation Guide and the OpenShift Migration Toolkit for Virtualization. The latter facilitates the migrations of the application VMs. Also consider checking How to automate migrations with Red Hat Ansible Automation Platform.

The intended audience of this document are experienced BIG-IP admins and operators. It is highly recommended to be fluent with Kubernetes or be assisted by the OpenShift admin.

In this guide, it is provided a procedure and sample Kubernetes manifests to deploy the networking resources and the BIG-IP VEs in OpenShift. You will find in these samples highlighted text indicating it is likely that needs customization for your environment.

Migrating a BIG-IP VE from VMware to OpenShift Virtualization doesn´t require any additional licenses. The existing license keys can be re-used.

2 Overview of the migration process

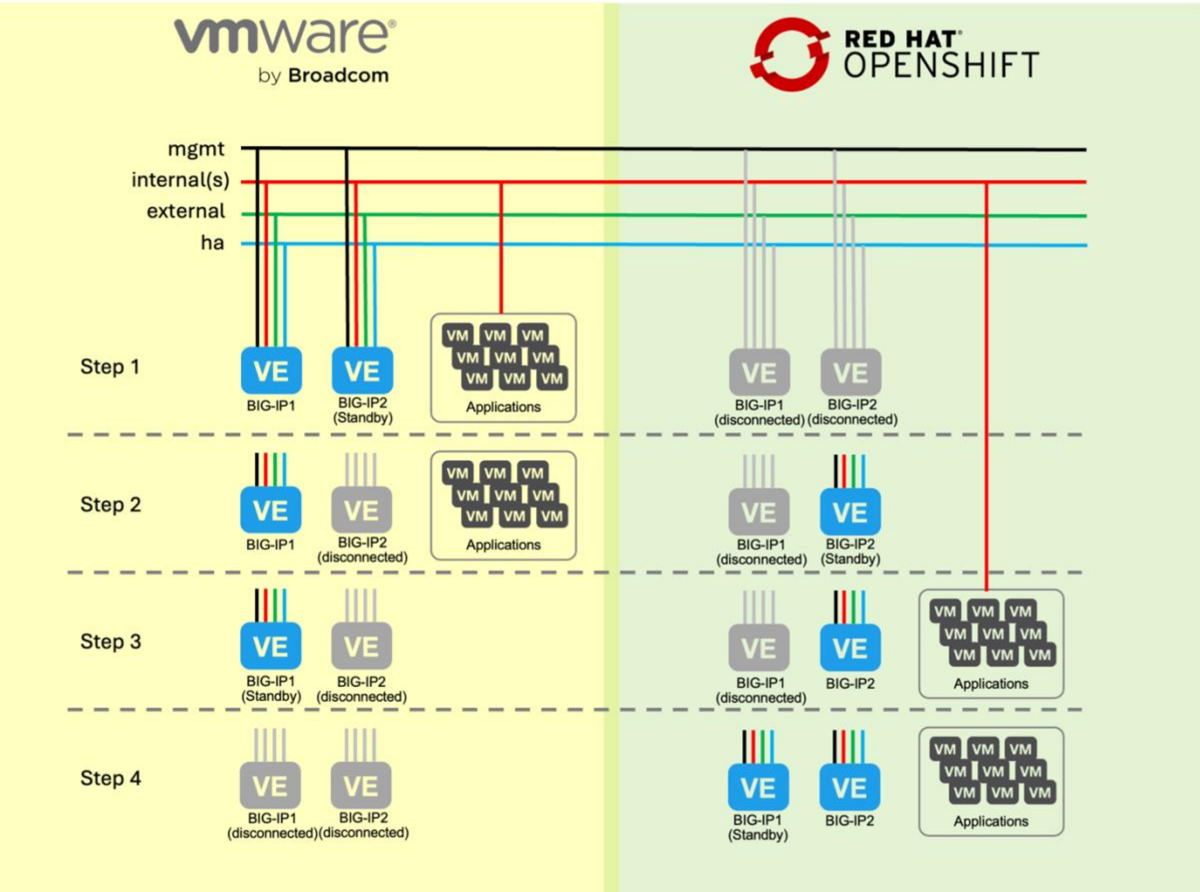
While performing application migrations, it is recommended to use Red Hat's Migration Toolkit for Virtualization. For migrating BIG-IP Virtual Edition (VE), this involves creating replicas of the existing VMware BIG-IP VE instances in the target OpenShift environment. Initially, these new VE instances deployed in OpenShift will be disconnected from the network. This also allows an easy and fast rollback, if necessary.

During the transition, the VMware BIG-IP VEs will save and export their current configuration as a UCS file (these include the installed iAppsLX packages), revoke their licenses, and then be disconnected from the network. Only after this will the new OpenShift BIG-IP VEs be connected to the network. This process is performed one BIG-IP unit at a time. Once the migration is complete, the VMware BIG-IP VEs can be safely deleted.

The procedure described in the section below is summarized in the figure below where the 4 steps are:

Step 1 – Deploy additional VEs in OpenShift
Step 2 – Migrate the Standby BIG-IP2 from VMware to OpenShift
Step 3 – Migrate the applications
Step 4 – Migrate the Standby BIG-IP1 from VMware to OpenShift

This procedure requires only a single BIG-IP failover event, minimizing disruption.



During the migration of BIG-IP units between hypervisors, the following functionalities will not be operational:

- Connection mirroring
- Persistence mirroring

The device group (High Availability) will need to be re-established because the new BIG-IP units will have different serial numbers and base MAC addresses. During the interval while the device group is being re-established, the standby unit will be in a forced offline state. This prevents the BIG-IP unit from entering an active state during the maintenance.

3 Network considerations

3.1 OpenShift L2 networking

BIG-IP VE can use either SR-IOV or VirtIO NICs; however, this guide focuses on VirtIO NICs. BIG-IP VE supports any VirtIO-based networking. This guide follows the recommendations of the OpenShift Virtualization Reference Implementation Guide, using NodeNetworkConfigurationPolicy with an OVS bridge and NetworkAttachmentDefinition with the ovn-k8s-cni-overlay type and localnet topology.

3.2 Features of ovn-k8s-cni-overlay type and localnet topology

This L2 networking provides access to the physical network and enables micro-segmentation using tags instead of IP addresses, while also offering general traffic visibility. This feature must be enabled on the Operator and configured using the MultiNetworkPolicy resource.

It's important to note that the physical network access is provided by ovn-k8s-cni-overlay with access ports to the VMs, meaning that VLAN trunking or guest level tagging cannot be applied to the VMs. If VLAN trunking or guest level tagging was previously used in VMware, you'll need to adjust the interfaces and tags configured for the VLANs to work with ovn-k8s-cni-overlay. BIG-IP VE supports up to 28 NICs. With one NIC allocated for management, this allows for connectivity to 27 application VLANs with this network configuration.

3.3 VLAN trunking

If VLAN trunking is required towards the VMs, a L2 configuration consisting of a `NodeNetworkConfigurationPolicy` linux-bridge and a `NetworkAttachmentDefinition` with cnv-bridge can be used.

3.4 Link Aggregation

Alike in VMware environments, link aggregation should be configured at the hypervisor level. This way link aggregations are transparent to the VMs.

3.5 MTU

It is recommended to set the mtu explicitly in the in the Network Attachment definitions. The provided sample files include this.

4 Preparatory works

Note: platform (OpenShift) configurations are typically done by the platform (OpenShift) admin and not the BIG-IP admin.

4.1 Platform compute configuration

The base installation of OpenShift Virtualization is done following these steps. After this, it is best practice to additionally setup CPU Manager to guarantee resource allocation and also setup Topology Manager to set a single `single-numa-node` policy which guarantees all VM resources are in the same NUMA domain following the Using CPU Manager and Topology Manager documentation:

- Use a `machineConfigPool` resource type apply the label `custom-kubelet: cpumanager-enabled` to nodes where it is desired where BIG-IP VE instances will be run.
- Use the options provided in the sample KubeletConfig file and add these to your existing KubeletConfig file.

Allow some time to allow these changes to be applied by OpenShift[1].

OpenShift Virtualization/KubeVirt uses QEMU/KVM under the hood and typically the QEMU Guest Agent is used to allow low level operations on the VMs. In the case of an F5 BIG-IP VE appliance these operations are not supported and therefore the agent is not needed.
Please check https://clouddocs.f5.com/cloud/public/v1/kvm/kvm_setup.html#prerequisites-for-big-ip-virtual-edition for up to date information.Please check https://clouddocs.f5.com/cloud/public/v1/kvm/kvm_setup.html#prerequisites-for-big-ip-virtual-edition for up to date information.

4.2 Platform storage configuration

The storage class used for BIG-IP VE can be either `ReadWriteOnce` or `ReadWriteMany`. KubeVirt only allows live migrations when using `ReadWriteMany`. Please note that a live migration should only be done while the BIG-IP unit is in Standby.

4.3 Platform and VirtualMachine network configuration

The network configuration of the platform is done using two resource types:

- `NodeNetworkConfigurationPolicy:` platform (not namespaced) configuration used create an OVS switch and VLANs.

- `NetworkAttachmentDefinition:` namespaced configuration that creates the VLANs where the VMs are connected.

Next are the sample files provided for these:

[1] You can monitor the change with `oc get mcp`

NodeNetworkConfigurationPolicy	NetworkAttachmentDefinition
nncp-bridge-vms.yaml	net-attach-mgmt.yaml net-attach-ha.yaml net-attach-ext.yaml net-attach-int.yaml

The highlighted text below indicates attributes that need to be customized of these resource types.

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: ovs-vms
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ''
  desiredState:
    interfaces:
    - name: ovs-vms
      type: ovs-bridge
      state: up
      bridge:
        allow-extra-patch-ports: true
        options:
          stp: true
        port:
          - name: eno1
    ovn:
      bridge-mappings:
      - localnet: net-mgmt
        bridge: ovs-vms
        state: present
      - localnet: net-ha
        bridge: ovs-vms
        state: present
      - localnet: net-ext
        bridge: ovs-vms
        state: present
      - localnet: net-int
        bridge: ovs-vms
        state: present
```

Where the switch port is suggested to be a link aggregation for redundancy.

```
kind: NetworkAttachmentDefinition
metadata:
  name: net-ext
  namespace: f5-bigip
spec:
  config: '{
    "cniVersion": "0.3.1",
    "name": "net-ext",
    "type": "ovn-k8s-cni-overlay",
    "topology": "localnet",
    "netAttachDefName": "f5-bigip/net-ext",
    "vlanID": 3011,
```

```
    "mtu": 1500,
    "ipam": {}
  }'
```

Where the network names must match between these two resource types and also the networks defined in the next section for the VirtualMachine resource types.

4.4 VirtualMachine configuration

4.4.1 BIG-IP VE images and storage provisioning

OpenShift virtualization requires using Q35 machine type. Please check this link to validate if your currently running BIG-IP version in VMware is also supports Q35. Make sure of using in OpenShift the same BIG-IP version used in VMware, upgrading if necessary.

There are 4 types of BIG-IP image: `LTM_1SLOT`, `LTM`, `ALL_1SLOT` and `ALL`. Check this link to select the appropriate type for your (typically the same as in the VMware environment). Take note of the required storage to be provisioned for the next sections.

KubeVirt is based on the KVM/QEMU projects thus it uses the QCOW2 format. When downloading these images from https://my.f5.com/ these will be found ZIP compressed and must be unzipped before being imported. Check the Creating virtual machines from custom images overview section to check the different options for importing the images.

4.4.2 Compute provisioning

In terms of CPU and memory, the BIG-IP VE doesn´t use more CPU or memory regardless of the hypervisor so it is safe to use the same provisioning previously used. For additional information you can check https://clouddocs.f5.com/cloud/public/v1/kvm/kvm_setup.html#prerequisites-for-big-ip-virtual-edition.

4.4.3 Cloud-init

If you are already using cloud-init in VMware this should not require any modification. If you are not using cloud-init, the sample VirtualMachine definitions provided have basic examples of cloud-init which just configure the management interface and sets the passwords for the admin and root users. This should be enough for most of the cases because the actual configuration will be set using a UCS backup file.

For additional information on cloud-init, see:

- For general information on KubeVirt and cloud-init https://kubevirt.io/user-guide/user_workloads/startup_scripts/#cloud-init.

- For BIG-IP VE specific information on cloud-init check https://clouddocs.f5.com/cloud/public/v1/shared/cloudinit.html#cloud-init-modules-supported-in-big-ip-ve

4.4.4 Kubernetes manifests to instantiate BIG-IP VE

Sample `VirtualMachine` and `DataVolume` manifests for a typical HA pair are provided in the repository with the typical external, internal, HA and management interfaces, and in which the QCOW2 images are downloaded from an HTTP server.

bigip1-vm.yaml bigip1-datavolume.yaml	bigip2-vm.yaml bigip2-datavolume.yaml

Next are described key sections in the `VirtualMachine` manifests (using the manifests of BIG-IP1 as sample):

- podAntiAffinity rules to avoid two BIG-IPs of the same HA-pair to run in the same node.

```
spec:
  affinity:
    podAntiAffinity:
      tequiredDuringSchedulingIgnoredDuringExecution:
      labelSelector:
        matchExpressions:
        - key: "bigip-unit"
          operator: In
          values:
          - unit-2
      topologyKey: "kubernetes.io/hostname"
```

- Resource allocation: CPU, memory and network queues

```
    spec:
[…]
      domain:
        cpu:
          sockets: 1
          # Adjust cores to the desired number of vCPUs
          cores: 4
          threads: 1
          dedicatedCpuPlacement: true
        resources:
          # memory must be 2Gi per core at least
          requests:
            memory: 8Gi
          limits:
            memory: 8Gi
        devices:
          networkInterfaceMultiqueue: true
```

- Annotations to hint the CPU scheduler to enable low latency to the BIG-IPs. See this link for details.

```
  annotations:
[…]
    cpu-quota.crio.io: "disable"
    cpu-load-balancing.crio.io: "disable"
    irq-load-balancing.crio.io: "disable"
```

- Network configuration (multus) to indicate the interfaces used. These are stretched across different sections.

```
  annotations:
    k8s.v1.cni.cncf.io/networks: "net-mgmt,net-ha,net-
ext,net-int"

[…]

            interfaces:
            - name: mgmt
              bridge: {}
            - name: int
              bridge: {}
            - name: ext
              bridge: {}
            - name: ha
              bridge: {}

[…]

        networks:
        - name: mgmt
          multus:
            networkName: net-mgmt
        - name: int
          multus:
            networkName: net-int
        - name: ext
          multus:
            networkName: net-ext
        - name: ha
          multus:
            networkName: net-ha
```

- Cloud-init passwords configuration

```
    chpasswd:
      list: |
        root:<existing password>
```

```
    admin:<existing password>
  expire: False
```

- Cloud-init network configuration

  In the example provided, static IP configuration of the management interface is used. Regardless of using DHCP or static configuration, the migration procedure expects that **the IP management address and the hostname remain the same in OpenShift Virtualization**. This latter is especially important for the UCS backup files to be installed fully.

```
tmsh modify sys global-settings mgmt-dhcp disabled
tmsh create sys management-ip 192.170.7.201/24
tmsh create sys management-route default gateway 192.170.7.1
tmsh modify sys global-settings hostname bigip1.ocp.f5-udf.com
```

- Cloud-init workaround for ID1492337.

  Depending the BIG-IP version, it might be affected by https://cdn.f5.com/product/bugtracker/ID1492337.html which requires to set the MTU of the data plane interfaces at boot time. The provided sample VirtualMachine definitions contain the required statement in cloud-init to set the MTU. Modify this statement with the appropriate MTU value or eliminate the statement depending if your version is affected.

```
content: |
  #!/bin/bash
  # Modify or eliminate the following statement according to
https://cdn.f5.com/product/bugtracker/ID1492337.html
  echo "ndal mtu 1500 1af4:1041" >> /config/tmm_init.tcl
```

  The example provided sets an MTU of 1500 bytes.

  Note that the MTU value still has to be set normally at VLAN level after the workaround above.

Next is described the key sections in the `DataVolume` manifests:

```
spec:
  source:
    http:
      url: "http://<http server address>/BIGIP-17.1.1.2-
0.0.10.qcow2"
  pvc:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: 80Gi
```

In the above example it is assumed that the default KubeVirt storageClass is used. This can be customized as well, please check https://github.com/kubevirt/containerized-data-importer/blob/main/doc/datavolumes.md

5 Migration procedure

The procedure below assumes the BIG-IP instances are named BIG-IP1 and BIG-IP2, with BIG-IP1 initially set as Active.

This procedure is based on a typical BIG-IP setup with four NICs: management, internal, external, and high availability. If your setup differs, you may need to adjust the procedure accordingly. If the VMware configuration included tagged or trunked ports, these settings will need to be modified after loading the UCS file of the BIG-IP VEs in OpenShift.

While the procedure outlined uses `oc` and `tmsh` commands, it can also be fully executed through the OpenShift and BIG-IP UIs, following the same steps.

The procedure assumes the platform configurations have been previously completed by the platform (OpenShift) admin. These include the following configurations previously indicated:

- Platform compute configuration (`KubeletConfig`)
- Platform storage configuration
- Platform network configuration (`NodeNetworkConfigurationPolicy`)

The procedure also assumes that the non-platform (namespaced) manifests have been customized using the guidelines in the previous sections.

The user running the procedure must be logged in with the necessary rights to create the following resource types in the designated namespace for the BIG-IP VEs:

```
NetworkAttachmentDefinition
VirtualMachine
DataVolume
```

5.1 Step 1 – Deploy additional VEs in OpenShift

5.1.1 Create a namespace for the namespaced resources

```
oc create ns f5-bigip
```

5.1.2 Create the NADs (Network Attachment Definitions) resources

The next must be applied for both units (example shown for BIG-IP1 only). The sample manifests provided assume 4 L2 networks: management, internal, external and high-availability.

```
$ oc apply -f net-attach-mgmt.yaml -f net-attach-int.yaml -f net-attach-ext.yaml -f net-attach-ha.yaml
networkattachmentdefinition.k8s.cni.cncf.io/net-mgmt configured
networkattachmentdefinition.k8s.cni.cncf.io/net-int configured
networkattachmentdefinition.k8s.cni.cncf.io/net-ext configured
networkattachmentdefinition.k8s.cni.cncf.io/net-ha configured
```

5.1.3 Create the VirtualMachine resources

The next must be applied for both units (examples shown for BIG-IP1 only).

- Apply the VM manifest and immediately pause the VM in order to disconnect the interfaces.

```
$ oc apply -f bigip1-vm.yaml -f bigip1-datavolume.yaml
virtualmachine.kubevirt.io/bigip1 created
datavolume.cdi.kubevirt.io/bigip1-datavolume created

$ while ! virtctl -n f5-bigip pause vm bigip1 ; do sleep 1 ; done
Error pausing VirutalMachineInstance bigip1: Operation cannot be
fulfilled on virtualmachineinstance.kubevirt.io "bigip1": VM is not
running
[…]
VMI bigip1 was scheduled to pause
```

- Monitor in a separate terminal the VM creation as this will likely take several minutes.

At present, virtctl or the KubeVirt manifests lacks the capability to modify the link status or start a VM with the links disconnected at boot time. For this reason, the step above uses a workaround to immediately pause the VM at boot time. See https://issues.redhat.com/browse/CNV-43347 for details.

- Disconnect the management interface from the network to avoid an IP address conflict.

```
$ LAUNCHER=$(oc -n f5-bigip get pods -o
jsonpath='{.items[?(@.metadata.ownerReferences[].name=="bigip1")].metadata.name}'
)
$ oc -n f5-bigip exec $LAUNCHER -- /bin/sh -c 'virsh domiflist 1 2>/dev/null |
grep "tap" | head -1 | awk "{print \$1}" | xargs -I {} virsh domif-setlink 1 {}
down'
Device updated successfully
```

5.1.4 Unpause the BIG-IP units

The next must be applied for both units (example shown for BIG-IP1 only).

After the interfaces are disconnected from the network is safe to run the VMs in OpenShift if their replicas in VMware are also running.

```
$ virtctl -n f5-bigip unpause vm bigip1
VMI bigip1 was scheduled to unpause
```

Once both BIG-IP units are created in OpenShift, with their links down, then unpaused, continue with the next step.

5.2 Step 2 – Migrate the Standby BIG-IP2 from VMware to OpenShift

5.2.1 Backup the configuration of the BIG-IP2 unit (Standby) in VMware

- Save a copy of the /config/bigip.license.

- Set the state of the unit to "forced offline" before saving a copy of a UCS archive so the unit will be in this state when loading the file later.

```
(tmos)# run sys failover offline
```

- Save a copy of a UCS archive.

```
(tmos)# save sys ucs bigip2.ocp.f5-udf.com
Saving active configuration...
/var/local/ucs/bigip2.ocp.f5-udf.com.ucs is saved.
```

- Keep the above files in a jumphost.

5.2.2 Disable the BIG-IP2 unit (Standby) in VMware

- Make sure the base and add-on registration keys are kept handy. See the section Additional information on how to get these from the license file.

- Revoke the license.

If license was not assigned using BIG-IQ, apply the next commands. Otherwise, please refer to BIG-IQ documentation.

```
root@(bigip2)(cfg-sync In Sync)(Standby)(/Common)(tmos)# revoke sys license
Revoking the license will return this BIG-IP to an unlicensed state.  It will
stop processing traffic. Are you sure? Y/N: y
License successfully revoked

Broadcast message from systemd-journald@bigip2.ocp.f5-udf.com (Tue 2024-06-25
00:36:48 PDT):

mcpd[8093]: 01070608:0: License is not operational (expired or digital signature
does not match contents).

2024 Jun 25 00:36:48 bigip2.ocp.f5-udf.com mcpd[8093]: 01070608:0: License is not
operational (expired or digital signature does not match contents).
```

```
root@(bigip2)(cfg-sync In Sync)(ModuleNotLicensed::LICENSE
INOPERATIVE)(/Common)(tmos)#
```

- Disconnect **all the interfaces of the BIG-IP2 in VMware**. This allows a fast rollback if needed.

NOTE: from now on, the BIG-IP2 unit used in the rest of this procedure will be the running in OpenShift.

5.2.3 Enable the BIG-IP2 unit (Standby) in OpenShift

- Enable the management interface of the VE from the hypervisor

```
$ LAUNCHER=$(oc -n f5-bigip get pods -o
jsonpath='{.items[?(@.metadata.ownerReferences[].name=="bigip2")].metadata.name}'
)
$ oc -n f5-bigip exec $LAUNCHER -- /bin/sh -c 'virsh domiflist 1 2>/dev/null |
grep "tap" | head -1 | awk "{print \$1}" | xargs -I {} virsh domif-setlink 1 {}
up'
Device updated successfully
```

- Make sure it is possible to access the BIG-IP through the management interface.

  Note that it will take some time cloud-init to complete the configuration. This is because cloud-init waits the control plane to be operational.

- License the unit.

```
root@(bigip2)(cfg-sync Standalone)(NO LICENSE)(/Common)(tmos)# install
sys license registration-key KJYYR-XXXXX-ALXSC-SNWAZ-ABCDEFG
Warning: the current license is not valid
New license installed
root@(bigip2)(cfg-sync Standalone)(Offline)(/Common)(tmos)#
```

Please note, if the VE is running in FIPS mode a slightly different procedure needs to be followed before continuing with the next steps.

See the Additional information section for details.

- Set the state of the unit to “forced offline”.

From the previous step, wait until the unit becomes Active. Once Active, run the following command:

```
root@(bigip2)(cfg-sync Standalone)(Active)(/Common)(tmos)# run sys failover
offline
root@(bigip2)(cfg-sync Standalone)(Active)(/Common)(tmos)#
root@(bigip2)(cfg-sync Standalone)(ForcedOffline)(/Common)(tmos)#
```

- Restore the UCS backup file

Note: please check K9420: Installing UCS files containing encrypted passwords or passphrases (11.5.x and later) if applies.

Upload the UCS backup file from the jumphost into the BIG-IP unit and apply the next tmsh commands.

```
root@(bigip2)(cfg-sync Standalone)(ForcedOffline)(/Common)(tmos)# load sys ucs
/var/local/ucs/bigip2.ocp.f5-udf.com.ucs no-license
Replace all configuation on the system? (y/n) y
Processing UCS file: /var/local/ucs/bigip2.ocp.f5-udf.com.ucs

Installing full UCS (17.1.1.2) data, excluding license file.
Extracting manifest: /var/local/ucs/bigip2.ocp.f5-udf.com.ucs
Product : BIG-IP
Platform: Z100
Version : 17.1.1.2
Edition : Point Release 2
Hostname: bigip2.ocp.f5-udf.com
Installing --full-- configuration on host bigip2.ocp.f5-udf.com
Installing configuration...
Post-processing...
usermod: no changes

Broadcast message from systemd-journald@bigip2.ocp.f5-udf.com (Mon 2024-06-24
01:41:08 PDT):

logger[14588]: Re-starting named

2024 Jun 24 01:41:08 bigip2.ocp.f5-udf.com logger[14588]: Re-starting named
Reloading License and configuration - this may take a few minutes...

Broadcast message from systemd-journald@bigip2.ocp.f5-udf.com (Mon 2024-06-24
01:41:20 PDT):

logger[15647]: Re-starting snmpd

2024 Jun 24 01:41:20 bigip2.ocp.f5-udf.com logger[15647]: Re-starting snmpd
/var/local/ucs/bigip2.ocp.f5-udf.com.ucs is loaded.
root@(bigip2)(cfg-sync Disconnected)(ForcedOffline)(/Common)(tmos)#
```

Run the following command from the bash shell and wait until “device ready online” is shown

```
[root@bigip2:ForcedOffline:Changes Pending] config # tail -100f
/var/log/ltm | grep "device ready for online"
Sep 13 03:41:49 bigip2.ocp.f5-udf.com warning sod[8913]: 010c007d:4:
Configuration load completed, device ready for online.
```

- Set BIG-IP2 Standby unit online again:

```
root@(bigip2)(cfg-sync In Sync)(ForcedOffline)(/Common)(tmos)# run sys failover
online
root@(bigip2)(cfg-sync In Sync)(ForcedOffline)(/Common)(tmos)#
root@(bigip2)(cfg-sync In Sync)(Standby)(/Common)(tmos)#
```

- Perform config sync to make sure device-group config works correctly and configuration is up to date

```
root@(bigip2)(cfg-sync Changes Pending)(Standby)(/Common)(tmos)# run cm config-
sync from-group device-group-failover
```

It will take few seconds until the unit is "In Sync" state.

5.3 Step 3 – Migrate the applications

5.3.1 Re-check applications' state

Before proceeding, verify the all the services are green in both the Active (BIG-IP1) and Standby (BIG-IP2) units and reachable

5.3.2 Perform a failover to the BIG-IP in OpenShift

Set the BIG-IP2 in the OpenShift as Active and verify that the services continue being reachable:

```
root@(bigip1)(cfg-sync In Sync)(Active)(/Common)(tmos)# run sys failover standby
root@(bigip1)(cfg-sync In Sync)(Active)(/Common)(tmos)#
root@(bigip1)(cfg-sync In Sync)(Standby)(/Common)(tmos)#
```

5.3.3 Migrate the applications

This is expected to be performed using the OpenShift migration toolkit for virtualization.

During the migration of the VMs from VMware to OpenShift the pool monitors will automatically detect when these are not available. Depending on the application it might be more desirable instead to drain the clients from the pools before the VMs (pool members) are migrated. This is achieved by using the **Disabled** or **Forced Offline** options. See K13310: Disable nodes or pool members for maintenance (11.x - 17.x) for more details.

In any case, it is not needed to remove the VMs from the pools during the migration process.

NOTE: during the migration, persistence mirroring or connection mirroring will not be operational, this is due to the VEs running in different in different hypervisors.

If persistence mirroring is enabled, the following might be seen in the TMM logs:

```
/var/log/tmm3:<13> Jun 24 03:56:34 bigip1.ocp.f5-udf.com notice DAG hash
mismatch; discarding mirrored state
```

No analogous message is shown if connection mirroring is left enabled.

5.4 Step 4 – Migrate the Standby BIG-IP1 from VMware to OpenShift

5.4.1 Backup the configuration of BIG-IP1 (Standby) in VMware

- Save a copy of the /config/bigip.license.
- Set the state of the unit to “forced offline” before saving a copy of a UCS archive so the unit will be in this state when loading the file later on.

```
(tmos)# run sys failover offline
```

- Save a copy of a UCS archive.

```
(tmos)# save sys ucs bigip1.ocp.f5-udf.com
Saving active configuration...
/var/local/ucs/bigip1.ocp.f5-udf.com.ucs is saved.
```

- Keep these files in a jumphost.

5.4.2 Disable the BIG-IP1 (Standby) in VMware

- Make sure the base and add-on registration keys are kept handy. See the section Additional information on how to get these from the license file.
- Revoke the license.

If license was not assigned using BIG-IQ, apply the next commands. Otherwise, please refer to BIG-IQ documentation.

```
root@(bigip1)(cfg-sync In Sync)(Standby)(/Common)(tmos)# revoke sys license
Revoking the license will return this BIG-IP to an unlicensed state.  It will
stop processing traffic. Are you sure? Y/N: y
License successfully revoked

Broadcast message from systemd-journald@bigip1.ocp.f5-udf.com (Tue 2024-06-25
00:36:48 PDT):

mcpd[8093]: 01070608:0: License is not operational (expired or digital signature
does not match contents).
```

```
2024 Jun 25 00:36:48 bigip1.ocp.f5-udf.com mcpd[8093]: 01070608:0: License is not
operational (expired or digital signature does not match contents).
root@(bigip1)(cfg-sync In Sync)(ModuleNotLicensed::LICENSE
INOPERATIVE)(/Common)(tmos)#
```

- Disconnect **all the interfaces of the BIG-IP1 in VMware**. This allows a fast rollback if needed.

5.4.3 Enable the Standby BIG-IP in OpenShift

- Enable the management interface

```
$ LAUNCHER=$(oc -n f5-bigip get pods -o
jsonpath='{.items[?(@.metadata.ownerReferences[].name=="bigip1")].metadata.name}'
)
$ oc -n f5-bigip exec $LAUNCHER -- /bin/sh -c 'virsh domiflist 1 2>/dev/null |
grep "tap" | head -1 | awk "{print \$1}" | xargs -I {} virsh domif-setlink 1 {}
up'
Device updated successfully
```

- Make sure it is possible to access the BIG-IP through the management interface.

  Note that it will take some time cloud-init to complete the configuration. This is because cloud-init waits the control plane to be operational.

- License the unit.

```
root@(bigip1)(cfg-sync Standalone)(NO LICENSE)(/Common)(tmos)# install
sys license registration-key XYZYR-CWPWW-ALXSC-SNWAZ-ABCDEFG
Warning: the current license is not valid
New license installed
root@(bigip1)(cfg-sync Standalone)(Offline)(/Common)(tmos)#
```

Please note, if the VE is running in FIPS mode a slightly different procedure needs to be followed before continuing with the next steps.

See the Additional information section for details.

- Set the state of the unit to “forced offline”

```
root@(bigip1)(cfg-sync Standalone)(Active)(/Common)(tmos)# run sys failover
offline
root@(bigip1)(cfg-sync Standalone)(Active)(/Common)(tmos)#
root@(bigip1)(cfg-sync Standalone)(ForcedOffline)(/Common)(tmos)#
```

- Restore the UCS backup file

```
root@(bigip1)(cfg-sync Standalone)(ForcedOffline)(/Common)(tmos)# load sys ucs
/var/local/ucs/bigip1.ocp.f5-udf.com.ucs no-license
Replace all configuration on the system? (y/n) y
Processing UCS file: /var/local/ucs/bigip1.ocp.f5-udf.com.ucs

Installing full UCS (17.1.1.2) data, excluding license file.
Extracting manifest: /var/local/ucs/bigip1.ocp.f5-udf.com.ucs
Product : BIG-IP
Platform: Z100
Version : 17.1.1.2
Edition : Point Release 2
Hostname: bigip1.ocp.f5-udf.com
Installing --full-- configuration on host bigip1.ocp.f5-udf.com
Installing configuration...
Post-processing...
usermod: no changes

Broadcast message from systemd-journald@bigip1.ocp.f5-udf.com (Mon 2024-06-24
06:05:46 PDT):

logger[6884]: Re-starting named

2024 Jun 24 06:05:46 bigip1.ocp.f5-udf.com logger[6884]: Re-starting named
Reloading License and configuration - this may take a few minutes...

Broadcast message from systemd-journald@bigip1.ocp.f5-udf.com (Mon 2024-06-24
06:06:00 PDT):

logger[7997]: Re-starting snmpd

2024 Jun 24 06:06:00 bigip1.ocp.f5-udf.com logger[7997]: Re-starting snmpd
/var/local/ucs/bigip1.ocp.f5-udf.com.ucs is loaded.
```

Run the following command from the bash shell and wait until “device ready online” is shown

```
[root@bigip1:ForcedOffline:Awaiting Initial Sync] config # tail -100f
/var/log/ltm | grep "device ready for online"
Sep 13 04:24:30 bigip1.ocp.f5-udf.com warning sod[8902]: 010c007d:4:
Configuration load completed, device ready for online.
```

- Set BIG-IP1 Standby unit online again:

```
root@(bigip1)(cfg-sync Awaiting Initial Sync)(ForcedOffline)(/Common)(tmos)# run
sys failover online
root@(bigip1)(cfg-sync Awaiting Initial Sync)(ForcedOffline)(/Common)(tmos)#
root@(bigip1)(cfg-sync Awaiting Initial Sync)(Standby)(/Common)(tmos)#
root@(bigip1)(cfg-sync Awaiting Initial Sync)(Standby)(/Common)(tmos)#
```

- Perform config sync from BIG-IP2:

```
root@(bigip1)(cfg-sync Awaiting Initial Sync)(Standby)(/Common)(tmos)# run cm
config-sync from-group device-group-failover
root@(bigip1)(cfg-sync Awaiting Initial Sync)(Standby)(/Common)(tmos)#
[…]
root@(bigip1)(cfg-sync In Sync)(Standby)(/Common)(tmos)#
```

It will take few seconds until the unit is "In Sync" state.

- Verify that the pool members in this unit just migrated (BIG-IP2) have the same state as the peer unit (BIG-IP1).

- At this point the migration is completed.

6 Additional information

6.1 Possible modifications to this migration

In step 1, both BIG-IP units are created in OpenShift ahead of the actual migration. This step could be created in a previous maintenance window to shorten the actual migration time.

6.2 Useful BIG-IP Knowledge Base articles

6.2.1 Licensing the BIG-IP system

- K7752: Licensing the BIG-IP system
- K2595: Activating and installing a license file from the command line
- K41458656: Reusing a BIG-IP VE license on a different BIG-IP VE system
- BIG-IQ documentation - Manage Software Licenses for Devices
- K91841023: Overview of the FIPS 140 Level 1 Compliant Mode license for BIG-IP VE

6.2.2 UCS backup files

- K13132: Backing up and restoring BIG-IP configuration files with a UCS archive
- K9420: Installing UCS files containing encrypted passwords or passphrases (11.5.x and later)

6.2.3 HA Trust domain configuration

- K42161405: Rebuilding device trust using the Configuration utility (11.x - 16.x)
- K63243467: Creating a device group using the Configuration utility (13.x - 17.x)
- K13639: Configuring a device group using tmsh

6.2.4 Changing the MTU of the management interface

- K000133081: Configure eth0 mgmt MTU

6.3 Getting the keys from the license file

To get the base registration key use the following command:

```
$ egrep "^Registration Key :" bigip1.license
Registration Key :             WRCKG-IJFMB-OISFV-OXEVJ-ABCDEFG
```

To get any possible Add-on key use the following command:

```
$ grep " key " bigip1.license | sed -e 's/.* key \(.*\)/\1/' | sort -u
HCADKSW-ABCDEFG
QYYXMUR-ABCDEFG
```

6.4 Re-creating a device-group and HA from the command line

In the procedure of this guide, it is not expected to need the re-creation of the device-group/HA. If for whatever reason It is needed to recreate the HA configuration, the following can be applied (example run from BIG-IP2):

- Reset device trust from BIG-IP2:

```
root@(bigip2)(cfg-sync Disconnected)(ForcedOffline)(/Common)(tmos)# delete cm
trust-domain Root
root@(bigip2)(cfg-sync Standalone)(ForcedOffline)(/Common)(tmos)#
```

- Recreate the device trust from BIG-IP2:

```
root@(bigip2)(cfg-sync Standalone)(ForcedOffline)(/Common)(tmos)# modify /cm
trust-domain /Common/Root add-device { device-ip 192.170.7.201 device-name
bigip1.ocp.f5-udf.com username admin password '<admin password>' ca-device true }
root@(bigip2)(cfg-sync Disconnected (Trust Domain
Only))(ForcedOffline)(/Common)(tmos)#
root@(bigip2)(cfg-sync In Sync (Trust Domain
Only))(ForcedOffline)(/Common)(tmos)#
```

- Recreate the failover device-group from BIG-IP2:

```
root@(bigip2)(cfg-sync In Sync (Trust Domain Only))(Active)(/Common)(tmos)#
modify cm device-group device-group-failover devices replace-all-with {
bigip1.ocp.f5-udf.com bigip2.ocp.f5-udf.com }
root@(bigip2)(cfg-sync Awaiting Initial Sync (Sync
Only))(ForcedOffline)(/Common)(tmos)#
```

- Recreate the traffic-groups from BIG-IP2

```
root@(bigip2)(cfg-sync Awaiting Initial Sync)(Active)(/Common)(tmos)# modify cm
traffic-group traffic-group-1 ha-order {  bigip1.ocp.f5-udf.com bigip2.ocp.f5-
udf.com }
```

6.5 Testing the MTU of the interfaces

Once the VM is up, it is recommended to verify that the MTU of the interfaces are setup properly.

The bundled ping utility comes from the iputils Linux package. To test the interface's MTU specify the the following value to the `-s <size>` parameter in the command line, where <size> must be calculated as:

```
Interface´s MTU - IP standard header (20) - ICMP standard header (8)
```

For example, if the MTU of the interface is 1400, then -s should be set to 1372:

```
[root@bigip2:Active:Awaiting Initial Sync] config # ping -M do -s 1372
192.170.7.201
PING 192.170.7.201 (192.170.7.201) 1372(1400) bytes of data.
1380 bytes from 192.170.7.201: icmp_seq=1 ttl=64 time=0.500 ms
1380 bytes from 192.170.7.201: icmp_seq=2 ttl=64 time=0.507 ms
1380 bytes from 192.170.7.201: icmp_seq=3 ttl=64 time=0.458 ms
1380 bytes from 192.170.7.201: icmp_seq=4 ttl=64 time=0.450 ms
1380 bytes from 192.170.7.201: icmp_seq=5 ttl=64 time=0.646 ms
^C
--- 192.170.7.201 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.450/0.512/0.646/0.071 ms
```

Note that it is required to use the `-M do` option to enable the Don´t Fragment bit.

Note also that the reported size of the reply packet is the whole ICMP packet.