# VM Image Packaging

VM image packaging can be done in two ways:

- VM Image Packaging Utility: This is an enhanced packaging process that allows the VM owner to run the **nfvpt.py** utility as a command with a combination of parameters to package the VM.
- Standard Image Packaging: This is a manual process in which a raw disk image (qcow2) is packaged along with the image properties file and bootstrap files (if needed) into a TAR archive file.



# VM Image Packaging Utility

A VM image package is a TAR archive file with the root disk image and other descriptor files. This packaging method simplifies the process of a VM image registration and deployment. The attributes specified for the image enable resource requirement specification, creation of VM profiles, and a host of other properties for the VM.

The Cisco Enterprise NFVIS VM image packaging tool, nfvpt.py, helps VM owners package their VMs. The tool takes one or more qcow2 images (raw disk file) as the input file along with VM specific properties, bootstrap configuration files (if any), and generates a compressed TAR file.

## Contents

The VM image packaging utility contains the following:

- nfvpt.py—It is a python based packaging tool that bundles the VM raw disk image/s along with VM specific properties.
- image_properties_template.xml—This is the template file for the VM image properties file, and has the parameters with default values. If the user provides new values to these parameters while creating the VM package, the default values get replaced with the user-defined values.
- nfvis_vm_packaging_utility_examples.txt—This file contains examples on how to use the image packaging utility to package a VM image.

- nfvis_vm_packaging_utility_3.3.1_userguide.pdf—This document provides information on how to use the tool.

# Usage

To get the list of parameters that can be included in the command, and to get an explanation of each of the parameters, run the **help** command for the tool.

```
nfvpt.py --help
Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -s, --simple          simple packaging with minimal options
  -o PACKAGE_FILE_NAME, --package_file_name=PACKAGE_FILE_NAME
                        [REQUIRED] file name for the target VNF package name-
                        default is root disk image name with extension .tar.gz
  -i DISK_IMG_NAMES, --root_disk_image=DISK_IMG_NAMES
                        [REQUIRED] List of root disk images to be bundled
                        example: --root_disk_image isrv.qcow2;
                        --root_disk_image isrv1.qcow2,isrv2.qcow2
  -n IMG_NAME, --image_name=IMG_NAME
                        [REQUIRED] Name of the VNF image
  -t VNF_TYPE, --vnf_type=VNF_TYPE
                        [REQUIRED] VNF type, e.g. router, firewall.. other
  -r VNF_VERSION, --vnf_version=VNF_VERSION
                        [REQUIRED] VNF version, e.g. --vnf_version 1.0 or
                        --vnf_version 0.9
  --monitored=MONITORED
    [REQUIRED] Monitored VNF: --monitored=true/false;
 --optimize=OPTIMIZE
   [REQUIRED] optimized VM: --optimize=true/false;
 --console_type_serial=CONSOLE_TYPE_SERIAL Attach the console serial to the VM;
   default is false; --console_type_serial=true/false;
 --root_file_disk_bus=ROOT_FILE_DISK_BUS
    root disk file type: --root_file_disk_bus=virtio/ide;
    default is virtio
 --virtual_interface_model=VIRTUAL_INTERFACE_MODEL
    --virtual_interface_model=rtl8139; default is none
 --thick_disk_provisioning=THICK_DISK_PROVISIONING
    --thick_disk_provisioning=true; default is false
 --bootstrap_cloud_init_bus_type=BOOTSTRAP_CLOUD_INIT_BUS_TYPE
    --bootstrap_cloud_init_bus_type=virtio; default is ide
 --bootstrap_cloud_init_drive_type=BOOTSTRAP_CLOUD_INIT_DRIVE_TYPE
    --bootstrap_cloud_init_drive_type=disk; default is cdrom
 --bootstrap=BOOTSTRAP
    bootstrap file/s for the VM (two params required in the format of dst:src; dst filename
 including path
    has to match exactly to what the VM expects; up to 20 bootstrap files accepted.)
   Examples: --bootstrap ovf-env.xml:file1,ios-xe.txt:file2 for ISRv; both files get mounted
 at the root level on the VM
    --bootstrap day0-config:filename1 for ASAv --bootstrap
 /:bootstrap.xml,/license/lic.txt:license.txt
    bootstrap.xml get mounted as bootstrap.xml at root and license.txt get mounted as
/license/lic.txt
 -v, --verbose  verbose
 -q, --quiet    quiet
 --no_compress  creates tar file without compressing the input files
 --cleanup      deletes all the input and configuration files upon tar file creation

 resources: min and max - vCPU, memory and disk :
```

```
 --min_vcpu=MIN_VCPU
    min #vCPU : min number of vCPU supported by VM
    example:--min_vcpu 2
 --max_vcpu=MAX_VCPU
    max #vCPU : max number if vCPU required for VM
   example:--max_vcpu 4
 --min_mem=MIN_MEM
   min mem : min mem in MB required for VM
   example:--min_mem 1024
 --max_mem=MAX_MEM
   max mem : max mem in MB required for VM
   example:--max_mem 4196
 --min_disk=MIN_DISK
    min disk : min disk in GB required for VM
    example:--min_disk 8
 --max_disk=MAX_DISK
    max disk : max disk in GB required for VM
    example:--max_disk 8
 --vnic_max=VNIC_MAX
    max number of Vnics allowed for VM
   example:--vnic_max 8

 Profile options :
 --profile=PROFILE
   enter the profile name, profile description, number of vCPUs required,
    min memory required in MB,
    min disk space required in MB,
    example:
    --profile profile1,"This is profile 1",2,2048,4096
    --profile profile2,"This is profile 2",4,4096,4096
 --default_profile=DEFAULT_PROFILE
    default profile

 Driver support options :
 --sriov=SRIOV Enable/Disable SRIOV support: --sriov=true/false;
    default is false
 --sriov_list=SRIOV_LIST
    list of SRIOV drivers
    example: --sriov_list igb,igbvf,i40evf
 --pcie=PCIE
   Not supported
 --pcie_list=PCIE_LIST
   Not supported

 Privilege/priority related options :
 --privileged=PRIVILEGED
    Not supported

 Custom properties :
 --custom=CUSTOM
   custom properties to be supported and/or passed to bootstrap config with tokenized
variables
    comma separated (key, val) pair to be passed for a list of values, use the same key and
 different value
    NOTE: mandatory for deployment through local portal if bootstrap config has tokenized
variables
    for local portal to prompt for custom property options
    example1: --custom tech_package,ax --custom tech_package, sec
    example2: --custom ip_address
```

The table lists the parameters that can be passed to the **nfvpt.py** command.

| Parameter | Mandatory/Optional | Description |
|---|---|---|
| version | Not applicable | Show program's version number and exit. |
| help | Not applicable | Show this help message and exit. |
| PACKAGE_FILE_NAME | Mandatory | File name for the target VNF package. The default is the root disk image name with extension *.tar.gz*. |
| DISK_IMG_NAMES | Mandatory | List of root disk images to be bundled. Only the qcow2 images are supported. |
| IMG_NAME | Mandatory | Name of the VNF image. |
| VNF_TYPE | Mandatory | VNF type<br>Supported types are: ROUTER, FIREWALL, vWAAS, vWLC, and OTHER. |
| VNF_VERSION | Mandatory | VNF version |
| MONITORED | Mandatory | VM health monitoring for those VMs that can be bootstrapped<br>Options are: true/false<br>Monitoring timeout period for a monitored VM is 600 seconds by default |
| OPTIMIZE | Mandatory | Optimized VM<br>Options are: true/false |
| VIRTUAL_INTERFACE_MODEL | Optional | Default is none. |
| THICK_DISK_PROVISIONING | Optional | Default is false. |
| BOOTSTRAP_CLOUD_INIT_BUS_TYPE | Optional | Default is IDE. |
| BOOTSTRAP_CLOUD_INIT_DRIVE_TYPE | Optional | Mounts the day0 configuration file as disk Default is CD-ROM. |
| BOOTSTRAP | Optional | Bootstrap files for VNF. Two parameters are required in the format of dst:src; dst filename including path has to match exactly to what the VM expects; up to 20 bootstrap files are accepted. For example: --bootstrap ovf-env.xml for ISRv and --bootstrap day0-config for ASAv. |
| MIN_VCPU | Optional | Minimum number of vCPUs supported by the VM.<br>The default is 1. |

| Parameter | Mandatory/Optional | Description |
|---|---|---|
| MAX_VCPU | Optional | Maximum number of vCPUs required for the VM.<br>The default is 8. |
| MIN_MEM | Optional | Minimum memory in MB required for the VM.<br>The default is 4 GB. |
| MAX_MEM | Optional | Maximum memory in MB required for the VM.<br>Physical memory: 2 GB<br>The default is 8 GB. |
| MIN_DISK | Optional | Minimum disk in GB required for the VM.<br>The default is 8 GB. |
| MAX_DISK | Optional | Maximum disk in GB required for the VM. Available disks are SSD and HDD: 15 GB<br>The default is 16 GB |
| VNIC_MAX | Optional | Maximum number of VNICs allowed for the VM.<br>The default is 8. |
| PROFILE | Optional | The profile name, profile description, number of vCPUs required, minimum memory required in MB and minimum disk space required in MB. |
| DEFAULT_PROFILE | Optional | The default profile. |
| SRIOV | Optional | Enable or disable SRIOV support. The default is false. |
| SRIOV_LIST | Optional | List of SRIOV drivers. |
| PCIE | Optional | Not supported. |
| PCIE_LIST | Optional | Not supported. |
| PRIVILEGED | Optional | Not supported. |

| Parameter | Mandatory/Optional | Description |
|---|---|---|
| CUSTOM | Optional | Custom properties to be supported and/or passed to the bootstrap configuration with tokenized variables. This is only used for the local portal to display options for the user to choose while deploying. |

# VM Packaging Utility Usage Examples

Given below are the contents of the file *nfvis_vm_packaging_utility_examples.txt*:

**Example 1: Usage for TinyLinux**

```
nfvpt.py -o TinyLinux -i TinyLinux.qcow2 -n TinyLinux -t linux -r 1.0 --monitored false
--min_vcpu 1 --max_vcpu 2 --min_mem 1024 --max_mem 1024 --min_disk 1 --max_disk 2
--vnic_max 1 --optimize false
```

**Example 2: Usage for ASAv**

**Note** The bootstrap filename has to be *day0-config*. This cannot be modified as ASAv looks for the exact filename.

```
nfvpt.py -o asav961-201 -i asav961-201.qcow2 -n ASAv -t firewall -r 961-201 --monitored
true --bootstrap day0-config:filename1
--min_vcpu 1 --max_vcpu 4 --min_mem 1024 --max_mem 8192 --min_disk 8 --max_disk 16 --vnic_max
 8 --optimize true
--profile ASAv5,"ASAv5 profile",1,1024,8192 --profile ASAv10,"ASAv10 profile",1,4096,8192
--profile ASAv30,"ASAv30 profile",4,8192,16384
--default_profile ASAv5
```

**Example 3: Usage for ISRv**

**Note** The bootstrap filename has to be *ovf-env.xml* . This cannot be modified as ISRv looks for the exact filename.

```
nfvpt.py -o isrv.16.03.01 -i isrv-universalk9.16.03.01.qcow2 -n ISRv.16.03.01 -t ROUTER -r
 16.03.01 --monitored true --privileged true
--bootstrap ovf-env.xml:file1,ios-xe.txt:file2 --min_vcpu 2 --max_vcpu 8 --min_mem 4096
--max_mem 8192 --min_disk 8 --max_disk 8
--vnic_max 8 --optimize true --profile ISRv-small,"ISRv small profile",2,4096,8192 --profile
 ISRv-medium,"ISRv medium profile",4,4096,8192
--default_profile ISRv-small --sriov_list igb,igbvf,i40evf --custom tech_package,ax
```

# Standard VM Image Packaging

The standard VM packaging is based on the Open Virtualization Format (OVF) packaging standard, in which a single file is distributed in open virtualization appliance (OVA) format. The VM image is shared using a TAR archive file with the root disk image and descriptor files.

**Note** Cisco Enterprise NFVIS supports VM packaging in *.tar.gz* (compressed form of OVA) format. Ensure that all supported third party VM images are available in the supported format.

## Generating a VM Package

Package files are provided for Cisco ISRv, Cisco ASAv, and tiny Linux and Windows server 2000. Vendors are responsible for packaging all third party VMs in the supported format.

1. Create a VM qcow2 image.
2. Create an *image_properties.xml* file with the VM properties. Ensure that you add all mandatory fields. Include the profiles supported for the VM in this file, and select one default profile. If you do not want to monitor the VM bootup, make the bootup time as -1.
3. Create *bootstrap-config* or *day0-config*, if any bootstrap configuration is required for the VM. If the bootstrap configuration requires inputs from the user, use the tokens in the xml or text file. These tokens are populated during the VM deployment with the provided data.

   **Note** A VM deployment may fail, if there are tokens in the configuration, and the user does not provide the token values in the deployment payload.

4. Create a *package.mf* file, which lists all the files to be bundled into the *.tar.gz* file along with checksums.
5. Generate the packaging file using "tar -cvf ova_file_name list_of_files_to_be_bundled".

   For example, *tar -cvzf isrv.tar.gz isrv-universalk9.03.16.02.S.155-3.S1a-ext-serial.qcow2 image_properties.xml isr_ovf_env.xml package.mf*.

# Appendix

## VM Image Package Files

The table lists the contents of the VM package that are generated using the packaging tool:

***Table 1: VM Image Package Files***

| File | Description | Mandatory/Optional |
|---|---|---|
| Package Manifest (package.mf) | Lists the files in the package and the expected checksum for the files. | Mandatory |
| VM image properties (vmname_properties.xml) | XML file with resources and features supported by the VM | Mandatory |

| | | |
|---|---|---|
| VM image (vmname.qcow2 ) | Image file of the VM. Multiple images are supported. One root_disk image file is mandatory. | Mandatory |
| BOOTSTRAP | Optional | Bootstrap files for VNF. Two parameters are required in the format of dst:src; dst filename including path has to match exactly to what the VM expects; up to 20 bootstrap files are accepted. For example: --bootstrap ovf-env.xml for ISRv and --bootstrap day0-config for ASAv. |

## Package Manifest File

The package manifest XML file provides a list of the files in the package with their names and their expected checksum. SHA1 algorithm (sha1sum) is used to calculate the checksum. This is a mandatory file to be bundled in the VM package. The manifest file must be named as *package.mf*.

***Table 2: Package Manifest File Details***

| Property Name | Description | Property Tag | Mandatory/Optional |
|---|---|---|---|
| File information | XML tree with details of file name, file type, and expected checksum. The root_image and image_properties files are required. | <file_info> | Mandatory |
| File name | Name of the file | <name> | Mandatory |
| File type | Describes the file type. Supported types:<br>• root_image<br>• image_properties<br>• bootstrap_config_file<br>• ephemeral_disk1_image<br>• ephemeral_disk2_image | <type> | Mandatory |
| Expected checksum | The calculated SHA1 checksum to be validated. | <sha1_checksum> | Mandatory |

## Bootstrap Configuration File

The bootstrap configuration file is an XML or a text file, and contains properties specific to a VM and the environment. Properties can have tokens, which can be populated during deployment time from the deployment payload.

## VM Image Properties File

This XML file provides information about the resources supported or required for the VM operation. All mandatory parameters have to be defined. It also supports custom attributes. This is a mandatory file to be bundled in the VM package. The VM package supports up to 10 disks to be bundled into the package.

***Table 3: VM Image Properties File Details***

| Property Name | Description | Property Tag | Possible Values | Mandatory/Optional |
|---|---|---|---|---|
| VNF Type | VM functionality provided. Router and firewall are predefined types. | <vnf_type> | Router, firewall, Windows, Linux, and custom_type | Mandatory |
| Name | Name associated with the VM packaging. This name is referenced for VM deployment. | <name> | Any | Mandatory |
| Version | Version of the package | <version> | Any | Mandatory |
| Boot-up time | Boot-up time (in seconds) of the VNF before it can be reachable via ping. | <bootup_time> | Any in seconds, (-1) to not monitor boot-up | Mandatory |
| Root Disk Image Bus | Root image disk bus | <root_file_disk_bus> | virtio, scsi, and ide | Mandatory |
| Disk-1 bus type | Additional disk1 image disk bus | <disk_1_file_disk_bus> | virtio, scsi, and ide | Optional |
| Disk-2 bus type | Disk2 image disk bus | <disk_2_file_disk_bus> | virtio, scsi, and ide | Optional |
| Disk-10 bus type | Disk10 image disk bus | <disk_10_file_disk_bus> | virtio, scsi, and ide | Optional |
| Root Disk Image format | Root image disk format | <root_image_disk_format> | qcow2 and raw | Mandatory |
| Disk-1 Image format | Additional disk 1 image format | <disk_1_image_format> | qcow2 and raw | Optional |

| | | | | |
|---|---|---|---|---|
| Disk-2 Image format | Disk 2 image format | <disk_2_image_format> | qcow2 and raw | Optional |
| Disk-10 Image format | Disk 10 image format | <disk_10_image_format> | qcow2 and raw | Optional |
| Serial Console | Serial console supported | <console_type_serial> | true, false | Optional |
| Minimum vCPU | Minimum vCPUs required for a VM operation | <vcpu_min> | | Mandatory |
| Maximum vCPU | Maximum vCPUs supported by a VM | <vcpu_max> | | Mandatory |
| Minimum memory | Minimum memory in MB required for VM operation | <memory_mb_min> | | Mandatory |
| Maximum memory | Maximum memory in MB supported by a VM | <memory_mb_max> | | Mandatory |
| Minimum root disk size | Minimum disk size in GB required for VM operation | <root_disk_gb_min> | | Optional |
| Maximum root disk size | Maximum disk size in GB supported by a VM | <root_disk_gb_max> | | Optional |
| Maximum vNICs | Maximum number of vNICs supported by a VM | <vnic_max> | | Mandatory |
| SRIOV support | SRIOV supported by VM interfaces. This should have a list of supported NIC device drivers. | <sriov_supported> | true, false | Optional |
| SRIOV driver list | List of drivers to enable SRIOV support | < sriov_driver_list> | | Optional |

| | | | | |
|---|---|---|---|---|
| PCI passthru support | PCI passthru support by VM interfaces | <pcie_supported> | true, false | Optional |
| PCIE driver list | List of VNICS to enable PCI passthru support | < pcie _driver_list> | | Optional |
| BOOTSTRAP | The bootstrap file for the VM | < bootstrap_file> | File name of the bootstrap file. | Optional |
| bootstrap_cloud_init_drive_type | Mounts day0 config file as disk (default is CD-ROM) | | | Optional |
| bootstrap_cloud_init_bus_type virtio | Default is IDE | | | Optional |
| BOOTSTRAP | Bootstrap files for VNF. Two parameters are required in the format of dst:src; dst filename including path has to match exactly to what the VM expects; up to 20 bootstrap files are accepted. For example: --bootstrap ovf-env.xml for ISRv and --bootstrap day0-config for ASAv | . | | Optional |

| | | | | |
|---|---|---|---|---|
| Custom properties | List of properties can be defined within the custom_property tree. (Example: For ISRv, the technology packages are listed in this block.)<br>If the Cisco Enterprise NFV portal is used to deploy the VM, the portal prompts you for inputs for custom properties fields, and can pass the values to the bootstrap configuration. | <custom_property> | | Optional |
| Profiles for VM deployment | List of VM deployment profiles. Minimum one profile is required | <profiles> | | Optional |
| Default profile | The default profile is used when no profile is specified during deployment. | <default_profile> | | Optional |
| Monitoring Support | A VM supports monitoring to detect failures. | <monitoring_supported> | true, false | Mandatory |
| Monitoring Method | A method to monitor a VM. Currently, only ICMP ping is supported. | <monitoring_methods> | ICMPPing | Mandatory if monitoring is true |
| Low latency | If a VM's low latency (for example, router and firewall) gets dedicated resource (CPU) allocation. Otherwise, shared resources are used. | <low_latency> | true, false | Mandatory |

| | | | | |
|---|---|---|---|---|
| Privileged-VM | Allows special features like promiscuous mode and snooping . By default, it is false. | <privileged_vm> | true, false | Optional |

## Profile Properties File

***Table 4: Profile Properties Details***

| Property Name | Description | Tag | Possible Values | Mandatory/Optional |
|---|---|---|---|---|
| Profile for VM deployment | A profile defines the resources required for VM deployment. This profile is referenced during VM deployment. | <profile> | | Optional |
| Name | Profile name | <name> | Any | Mandatory |
| Description | Description of the profile | <description> | Any | Mandatory |
| vCPU | vCPU number in a profile | <vcpus> | | Mandatory |
| Memory | Memory - MB in profile | <memory_mb> | | Mandatory |
| Root Disk Size | Disk size - MB in profile . | <root_disk_mb> | | Mandatory |

**Note** A virtual console is supported by default. Specify the root disk size as zero for multiple disks (for example, vWaas deployment) as the system does not support populating multiple disk sizes. Actual disk sizes are calculated from the root_disk files.

## Example: Package.mf

```
** sha1sum - for calculating checksum
<PackageContents>
  <File_Info>
    <name>ISRv_serial_3.16.02.qcow2</name>
    <type>root_image</type>
    <sha1_checksum>93de73ee3531f74fddf99377972357a8a0eac7b</sha1_checksum>
  </File_Info>
  <File_Info>
    <name>image_properties.xml</name>
    <type>image_properties</type>
    <sha1_checksum>c5bb6a9c5e8455b8698f49a489af3082c1d9e0a9</sha1_checksum>
</File_Info>
  <File_Info>
    <name>ISRv_ovf_env.xml</name>
    <type> bootstrap_file_1</type>
```

```
    <sha1_checksum>c5bb6a9c5e8455b8698f49a489af3082c1d9e0a9</sha1_checksum>
   </File_Info>
   <File_Info>
     <name>ISRv_disk1_image.qcow2</name>
     <type>ephemeral_disk1_image</type>
     <sha1_checksum>aac24513098ec6c2f0be5d595cd585f6a3bd9868</sha1_checksum>
   </File_Info>
</PackageContents>
```

## Example: Image Properties

```
<?xml version="1.0" encoding="UTF-8"?>
<image_properties>
    <vnf_type>ROUTER</vnf_type>
    <name>isrv-universalk9</name>
    <version>03.16.02</version>
    <bootup_time>600</ bootup_time >
    <root_file_disk_bus>virtio</root_file_disk_bus>
    <root_image_disk_format>qcow2</root_image_disk_format>
    <vcpu_min>1</vcpu_min>
    <vcpu_max>8</vcpu_max>
    <memory_mb_min>4096</memory_mb_min>
    <memory_mb_max>8192</memory_mb_max>
    <vnic_max>8</vnic_max>
    <root_disk_gb_min>8</root_disk_gb_min>
    <root_disk_gb_max>8</root_disk_gb_max>
    <console_type_serial>true</console_type_serial>
    <sriov_supported>true</sriov_supported>
    <sriov_driver_list>igb</sriov_driver_list>
    <sriov_driver_list>igbvf</sriov_driver_list>
    <sriov_driver_list>i40evf</sriov_driver_list>
    <pcie_supported>true</pcie_supported>
    <pcie _driver_list> igb </pcie_driver_list>
    <pcie _driver_list> igbvf</pcie_driver_list>
    <pcie _driver_list> i40evf</pcie_driver_list>
    <bootstrap_file_1> ovf-env.xml </bootstrap_file_1>
    <monitoring_supported>true</monitoring_supported>
    <monitoring_methods>ICMPPing</monitoring_methods>
    <low_latency>true</low_latency>
    <privileged_vm>true</privileged_vm>
    <cdrom>true</cdrom>
    <custom_property>
        <tech_package>ax</tech_package>
        <tech_package>sec</tech_package>
        <tech_package>ipbase</tech_package>
        <tech_package>appx</tech_package>
    </custom_property>
    <profiles>
         <profile>
            <name>ISRv1kv-small</name>
            <description>ISRv upto 50MBPS performance</description>
            <vcpus>1</vcpus>
            <memory_mb>4096</memory_mb>
            <root_disk_mb>8</root_disk_mb>
         </profile>
         <profile>
            <name>ISRv1kv-medium</name>
            <description>ISRv upto 250MBPS performance</description>
            <vcpus>2</vcpus>
            <memory_mb>4096</memory_mb>
            <root_disk_mb>8</root_disk_mb>
         </profile>
```

```
    </profiles>
    <default_profile>small</default_profile>
 </image_properties>
```

## Example: Bootstrap Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.ISRv.config-version.1" oe:value="1.0"/>
    <Property oe:key="com.cisco.isrv.enable-ssh-server.1" oe:value="True"/>
    <Property oe:key="com.cisco.isrv.login-password.1" oe:value="admin"/>
    <Property oe:key="com.cisco.isrv.login-username.1" oe:value="lab"/>
    <Property oe:key="com.cisco.isrv.mgmt-interface.1" oe:value="GigabitEthernet1"/>
   <Property oe:key="com.cisco.isrv.mgmt-ipv4-addr.1" oe:value="${NICID_0_IP_ADDRESS}/24"/>

    <Property oe:key="com.cisco.isrv.mgmt-ipv4-network.1" oe:value=""/>
    <Property oe:key="com.cisco.isrv.license.1" oe:value="${TECH_PACKAGE}"/>
   <Property oe:key="com.cisco.isrv.ios-config-0001" oe:value="vrf definition Mgmt-intf"/>

    <Property oe:key="com.cisco.isrv.ios-config-0002" oe:value="address-family ipv4"/>
    <Property oe:key="com.cisco.isrv.ios-config-0003" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.isrv.ios-config-0004" oe:value="address-family ipv6"/>
    <Property oe:key="com.cisco.isrv.ios-config-0005" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.isrv.ios-config-0006" oe:value="exit"/>
   <Property oe:key="com.cisco.isrv.ios-config-0007" oe:value="interface GigabitEthernet1"/>

   <Property oe:key="com.cisco.isrv.ios-config-0008" oe:value="vrf forwarding Mgmt-intf"/>

    <Property oe:key="com.cisco.isrv.ios-config-0009" oe:value="ip address
${NICID_0_IP_ADDRESS} ${NICID_0_NETMASK}"/>
    <Property oe:key="com.cisco.isrv.ios-config-0010" oe:value="no shut"/>
    <Property oe:key="com.cisco.isrv.ios-config-0011" oe:value="exit"/>
    <Property oe:key="com.cisco.isrv.ios-config-0012" oe:value="ip route vrf Mgmt-intf
0.0.0.0 0.0.0.0 ${NICID_0_GATEWAY}"/>
  </PropertySection>
</Environment>
```

# Image Properties Template File

The parameters that go into the image properties file are listed in the code extract below.

```
<?xml version="1.0" encoding="UTF-8"?>
<image_properties>
    <vnf_type>ROUTER</vnf_type>
    <name>TEMPLATE</name>
    <version>1.0</version>
    <bootup_time>600</bootup_time>
    <root_file_disk_bus>virtio</root_file_disk_bus>
    <root_image_disk_format>qcow2</root_image_disk_format>
    <vcpu_min>1</vcpu_min>
    <vcpu_max>8</vcpu_max>
    <memory_mb_min>4096</memory_mb_min>
    <memory_mb_max>8192</memory_mb_max>
```

```
    <vnic_max>8</vnic_max>
    <root_disk_gb_min>8</root_disk_gb_min>
    <root_disk_gb_max>16</root_disk_gb_max>
    <console_type_serial>false</console_type_serial>
    <sriov_supported>true</sriov_supported>
    <sriov_driver_list>s1</sriov_driver_list>
    <sriov_driver_list>s2</sriov_driver_list>
    <sriov_driver_list>s3</sriov_driver_list>
    <pcie_supported>false</pcie_supported>
    <monitoring_supported>true</monitoring_supported>
    <monitoring_methods>ICMPPing</monitoring_methods>
    <low_latency>true</low_latency>
    <privileged_vm>false</privileged_vm>
    <cdrom>true</cdrom>
    <bootstrap_file_1>b1.xml</bootstrap_file_1>
    <bootstrap_file_2>b2.txt</bootstrap_file_2>
    <custom_property>
        <key>val</key>
    </custom_property>
    <profiles>
        <profile>
            <name>small</name>
            <description>small</description>
            <vcpus>1</vcpus>
            <memory_mb>1024</memory_mb>
            <root_disk_mb>4096</root_disk_mb>
        </profile>
        <profile>
            <name>medium</name>
            <description>medium</description>
            <vcpus>2</vcpus>
            <memory_mb>4096</memory_mb>
            <root_disk_mb>8192</root_disk_mb>
        </profile>
    </profiles>
    <default_profile>small</default_profile>
</image_properties>
```