

Virtual machine creation and management

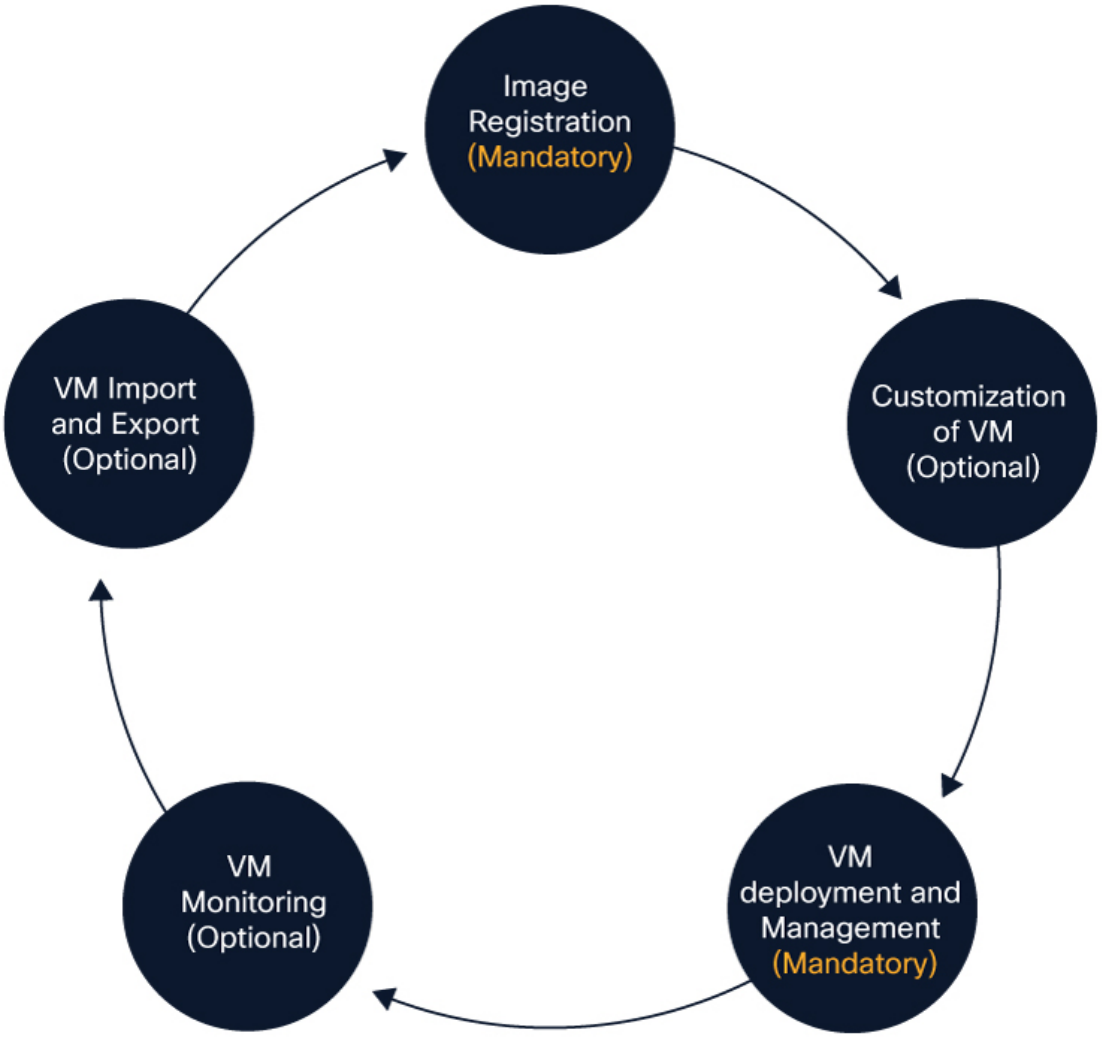
Objective

VM life cycle management is a process that

VM deployment workflow

- refers to the entire process of registering, deploying, updating, monitoring VMs
- gets VMs service chained as per your requirements, and
- can be performed using a set of REST APIs or NETCONF commands or the Cisco NFVIS portal.

VM Deployment Workflow Diagram:



- Uploading VM images to an NFVIS server, on page 2
- Register a virtual machine image, on page 3
- Customization of VM, on page 4
- Creating VM, on page 11
- Additional capabilities, on page 20

Uploading VM images to an NFVIS server

Uploading VM images to an NFVIS server is a file transfer process that copies VM images to the default location (/data/intdatastore/uploads) on the host server.

VM image upload methods

You can upload VM images to an NFVIS server in these ways:

- Copy the images from your local system to the NFVIS server—Use the Image Upload option from the Cisco NFVIS portal.
- Copy using the scp command (scp username@external_server:/path/image.tar.gz intdatastore:image.tar.gz).

Register a virtual machine image

Use this task to register a Virtual Machine (VM) image hosted on a remote server (HTTPS or SCP) for use in Cisco NFVIS. This process allows NFVIS to download the ISO image and OVA file, parse the metadata, and automatically create deployment profiles and flavors.

Table 1: Protocols

Protocol	Authentication	Notes
HTTPS	Not supported	No Username and password
SCP	Required	Username and password

Procedure

Step 1 From the Cisco NFVIS portal, choose **Configuration** > **Virtual Machine** > **Images** > **Image Repository**.

Step 2 Select the remote image registration option.

Register Image.

Select **Remote Image Registration**.

Step 3 Configure image properties.

Enter a descriptive name in the **Image name** field.

Step 4 Configure the remote server connection.

Select HTTPS or SCP from the **Protocol** drop-down list.

Enter the server hostname or IP address in the **IP Address** field.

Enter the directory path to the ISO file in the **Image File Path** field.

Step 5 Configure authentication.

If you selected FTP, SFTP, or SCP as the protocol, enter the username and password for the remote server.

Note Authentication is not supported for HTTP or HTTPS protocols.

Step 6 Upload the OVA metadata.

Select the **Metadata** check box.

Enter the path to the OVA file in the **OVA File Path** field.

Example: `/home/user/images/cucm_15.0_vmv17_v1.2.sha512.ova`

Note The OVA file must reside on the same remote server as the ISO file.

Step 7 Configure additional options.

Select the **Dedicated Cores** check box if dedicated CPU cores are required for the virtual machine.

Step 8 Submit the registration request.

Click **Submit** to submit the registration request.

Step 9 Monitor the registration progress.

Track the registration progress in the **Image Repository**. Downloading large files can take several minutes.

NFVIS downloads the ISO image and OVA file, parses the OVA metadata, registers the image, creates profiles and flavors, and makes the image available for deployment.

Customization of VM

Objective

This reference groups information about Customization of VM, including VM profiles or flavors, VM profile creation methods.

VM profiles or flavors

A VM profile or flavor is a configuration template that

- defines VMs in terms of number of parameters for how to run the VM
- specifies parameters such as number of vCPUs, RAM, disk size and so on, and
- provides standardized VM configurations based on requirements.

VM profile creation methods

Objective

VM profiles are created using different methods depending on the image package type:

VM profile creation methods

- Flavors are created as part of image registration if you use the tar.gz image packages for registering a VM.

- For other image packages such as .qcow2, iso, and raw, you must define custom flavors based on your requirements.
- See Install Guides of supported applications for what types of Images and Profiles are supported.



Note Unless specified otherwise in the deployment payload, the value assigned to the custom image property default_profile is used at the time of deploying the VM. Only applicable to tar.gz image packages.

Create VM profile using REST API

Procedure

Step 1 Create VM profile using REST API.

Create a VM profile to specify the resource allocation and configuration parameters for virtual machines in your environment.

VM profiles define the hardware specifications for virtual machines including memory allocation, disk space, and CPU configuration. Use REST API calls to programmatically create these profiles for consistent VM deployment.

Step 2 Execute the following cURL command to create a VM profile using the REST API.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/vm_lifecycle/flavors
 -d '<flavor>

    <name>windows</name>

    <ephemeral_disk_mb>0</ephemeral_disk_mb>

    <memory_mb>4096</memory_mb>

    <root_disk_mb>12288</root_disk_mb>

    <swap_disk_mb>0</swap_disk_mb>

    <vcpus>2</vcpus>

</flavor>'
```

The VM profile is successfully created with the specified configuration parameters and can be used for virtual machine deployment.

Internal management networks

An internal management network is a separate network within the Cisco NFVIS infrastructure that

- handles communication between the Cisco NFVIS and other management-related functions like port forwarding
- is used for VM monitoring and recovery, and

- is created by Cisco NFVIS as a network named int-mgmt-net by default.

How default configuration works

Summary

Here is an example of the default int-mgmt-net configuration:

Workflow

You can update the default configuration before deploying VMs/VNFs. When the default config is deleted, Cisco NFVIS recreates the internal management network upon rebooting the host.

Process details

```
vm_lifecycle networks network int-mgmt-net
 subnet int-mgmt-net-subnet
  address 10.20.0.0
  netmask 255.255.255.0
  gateway 10.20.0.1
 !
!
```

VNF deployment monitoring: Cisco NFVIS lets you configure the VNF deployment to be monitored. To monitor the VM deployment, an interface is attached to the internal management network (int-mgmt-net). The interface attached contains the NIC ID 0 by default.

Internal Management Network in the VM Deployment Payload	IP ADDRESS token/variable in Day-0 config	Behavior

Example configuration using IP address:
`<interface>`
`<nicid>0</nicid>`
`<network>int-mgmt-net</network>`
`<ip_address>10.20.0.21</ip_address>`
`</interface>`
	Token ${NICID_0_IP_ADDRESS}" is either present or absent.	If the IP payload is available, the payload is assigned to the interface.
		If there is a token/variable, the IP ADDRESS is replaced with the token.
		When the IP ADDRESS in the IP payload is unavailable or incorrect, the deployment fails.

Example configuration when no IP ADDRESS is configured:
`<interface>`
`<nicid>0</nicid>`
`<network>int-mgmt-net</network>`
`</interface>`
	${NICID_0_IP_ADDRESS} token is present in the day-0 config.	Cisco NFVIS auto assigns the IP ADDRESS and the token/variable is replaced in the day-0 config with the same IP address and the token/variable is replaced in the day-0 config with the same IP address.

Internal Management Network in the VM Deployment Payload	IP ADDRESS token/variable in Day-0 config	Behavior

Example configuration when no IP ADDRESS is configured with the NIC ID being absent:
`<interface>`
`<nicid>0</nicid>`
`<network>int-mgmt-net</network>`
`</interface>`
	Not present
	By default, the interface is configured as DHCP. The token is not replaced.
		Note
		Starting from Cisco NFVIS Release 4.12.1, configuring an interface as a DHCP server is supported.
		The IP ADDRESS is auto allocated using the internal DHCP server. Cisco NFVIS utilizes the IP address for monitoring the deployment.

Here is a sample day-0 config with the IP address and the token/variable:

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
    <Property oe:key="com.cisco.c8000v.enable-ssh-server.1" oe:value="True"/>
    <Property oe:key="com.cisco.c8000v.login-username.1" oe:value="${SSH_USERNAME}"/>
    <Property oe:key="com.cisco.c8000v.login-password.1" oe:value="${SSH_PASSWORD}"/>
    <Property oe:key="com.cisco.c8000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
!!!GigabitEthernet1-nicid(0)-int-mgmt-interface-don't change ip address or don't shutdown
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1"
oe:value="${NICID_0_IP_ADDRESS}/${NICID_0_CIDR_PREFIX}"/>         ===> IP address set via
token
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.license.1" oe:value="${TECH_PACKAGE}"/>
   <Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="vrf definition Mgmt-intf"/>

    <Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="address-family ipv4"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0004" oe:value="address-family ipv6"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="exit"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="interface
GigabitEthernet1"/>
   <Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="vrf forwarding Mgmt-intf"/>

    <Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip address
${NICID_0_IP_ADDRESS} ${NICID_0_NETMASK}"/>   ===> IP address set via token
    <Property oe:key="com.cisco.c8000v.ios-config-0010" oe:value="no shut"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0011" oe:value="exit"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0012" oe:value="ip route vrf Mgmt-intf
0.0.0.0 0.0.0.0 ${NICID_0_GATEWAY}"/>
  </PropertySection>
</Environment>
```

Here is a sample day-0 config using DHCP server:

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
    <Property oe:key="com.cisco.c8000v.enable-ssh-server.1" oe:value="True"/>
    <Property oe:key="com.cisco.c8000v.login-username.1" oe:value="${SSH_USERNAME}"/>
```

```
    <Property oe:key="com.cisco.c8000v.login-password.1" oe:value="${SSH_PASSWORD}"/>
    <Property oe:key="com.cisco.c8000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
!!!GigabitEthernet1-nicid(0)-int-mgmt-interface-don't change ip address or don't shutdown
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1" oe:value=""/>
                ===> IP address left blank here
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.license.1" oe:value="${TECH_PACKAGE}"/>
   <Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="vrf definition Mgmt-intf"/>

    <Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="address-family ipv4"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0004" oe:value="address-family ipv6"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="exit"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="interface
GigabitEthernet1"/>
   <Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="vrf forwarding Mgmt-intf"/>

   <Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip address dhcp setroute"/>
              ===> DHCP specified vs IP address
    <Property oe:key="com.cisco.c8000v.ios-config-0010" oe:value="no shut"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0011" oe:value="exit"/>
  </PropertySection>
</Environment>
```

VNF volumes

Objective

A VNF can be created and deployed with maximum two volumes. Currently, NFVIS supports a maximum of two volumes per VNF. The empty volume can be used as an extra storage by the VNF. Starting from NFVIS 4.1 release, storage volumes can now be deployed on external datastores and NFS.

Restrictions for VNF volumes

Objective

Example: payload for creating volumes

VNF volume restrictions and payload examples

- Volumes cannot be updated for a VNF after a VNF has already been deployed.
- NFVIS currently supports only two volumes per VNF.
- Starting from NFVIS 4.1 release, volumes can be stored in local storage or NFS, and other data stores like datastore1, datastore2 and datastore3.

Payload for creating volumes

```
…
<volumes>
  <volume>
    <name>Volume1</name>
    <volid>1</volid>
    <bus>ide</bus>
    <size>1</size>
    <sizeunit>GiB</sizeunit>
    <format>qcow2</format>
```

```
    <device_type>disk</device_type>
    <storage_location>local</storage_location>
  </volume>
</volumes>
…
```

Volume storage locations

The following are the accepted values for storage_location tags:

- <storage_location>local</storage_location>
- <storage_location>nfs:NFS_MOUNT_NAME</storage_location>
- <storage_location>datastore1</storage_location>
- <storage_location>datastore2</storage_location>
- <storage_location>datastore3</storage_location>

Volume deployment parameters

Objective

Cisco NFVIS supports the fields for volume deployment shown in this table.

Volume properties and allowed values

Property	Allowed Values	Description
NAME	string, { length 1..255 }	NAME of the volume
volid	uint16	Volumes will be presented to the VM sorted by volume ID.
bus	virtio, ide, scsi	The bus type
size	unit 16	Size of the Volume
sizeunit	MiB,GiB,TiB,PiB	Size units. MiB/GiB/TiB/PiB/EiB
format	qcow2, raw	Format of the disk to be created.
device_type	disk	Type of the device being attached to the VM.
storage_location	local, datastore1, datastore2, datastore3, nfs:NFS_MOUNT_NAME	Storage location NAME **Note** Starting from NFVIS 4.1 release, external datastore storage location is supported.

Port forwarding

Port forwarding is a network configuration mechanism that

- redirects incoming traffic from WAN to access the internal management network of the VM
- uses the WAN bridge interface (WAN-br) by default for traffic redirection, and
- allows modification of the bridge interface using the source_bridge tag in the deployment payload.

Port forwarding configuration

Objective

The bridge interface that is used to redirect traffic coming from the WAN side can be modified using the source_bridge tag in the deployment payload as shown in this example:

Port forwarding configuration examples

```
<port_forwarding>
  <port>
  <type>ssh</type>
  <protocol>tcp</protocol>
  <vnf_port>22</vnf_port>
  <source_bridge>MGMT</source_bridge>
  <external_port_range>
    <start>20122</start>
    <end>20122</end>
  </external_port_range>
  </port>
</port_forwarding>
```

With this payload, the traffic coming from the WAN side is redirected through the management interface (MGMT) instead of the default WAN bridge (WAN-br) interface.

vCPU pinning

vCPU pinning is a resource allocation mechanism that

- assigns VMs to specific virtual CPUs for dedicated processing
- enables low latency configuration through dedicated core allocation
- reuses the Node.js CPU allocation API to interface with NFVIS CPU allocation scripts.

vCPU pinning configuration

Objective

Similar to the NFVIS VM model, containers can also be pinned to a specific vCPU. If a container requires a dedicated core, a low latency flag can be set using the configuration options. By default, no low latency flag is set, this means that the core may be shared with other containers. Because the workflow for container vCPU pinning is the same as VM deployments, VimManager reuses the Node.js CPU allocation API to interface with the NFVIS CPU allocation scripts.

Container pinning and volume details

Volumes support for container deployments uses the same datamodel as VM deployments and provides external storage options to the containers. The `storage_location` tag in the payload refers to the file path within the container where the volume is mounted.



Note

- Volumes are equivalent to docker binds.
- All container volumes are stored on datastore1.

Example: Container volume configuration

To mount a container using a volume or a specific path, use this configuration:

```
<vm_group>
    <name>docker-nginx</name>
    <image>nginx</image>
    . . .
    <volumes>
      <volume>
        <name>test1</name>
        <volid>1</volid>
        <storage_location>/etc/datastore</storage_location> <!-- the path to mount inside
the container -->
        <size>120</size>
        <sizeunit>MiB</sizeunit>
      </volume>
      <!-- using default size allocations - 10 gibibytes (GiB) -->
      <volume>
        <name>test2</name>
        <volid>2</volid>
        <storage_location>/var/logs</storage_location>
      </volume>
```

Creating VM

Objective

This reference groups information about Creating VM, including VM deployment, VM Deployment using granular RBAC.

VM deployment

VM deployment is a virtualization process that

- creates and configures virtual machines through API or CLI methods
- requires prior registration of VM images before deployment can begin, and
- accepts mandatory and optional parameters to customize the deployment configuration.

VM name requirements

Objective

The VM name must meet these requirements:

VM name requirements

- Must contain an uppercase character and a lowercase character.
- Must contain a digit.
- Must contain one of the following special characters: dot (.), underscore (_) and hyphen (-).
- Must not have more than 256 characters.

Note Ensure that you have registered a VM image before attempting to deploy it. For more details, see Image Registration.

Example: deploy VMs using REST API

This example demonstrates how to deploy a VM using REST API with a sample payload and how to verify the deployment status.

This is a sample payload of deploying a VM.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST
https://209.165.201.1/api/config/vm_lifecycle/tenants /tenant/admin/deployments --data '
<deployment>
  <name>ISRdep</name>
  <vm_group>
    <name>ISRvmgrp</name>
    <image>ISR_IMAGE</image>
    <bootup_time>500</bootup_time>
    <recovery_wait_time>0</recovery_wait_time>
    <interfaces>
      <interface>
        <nicid>0</nicid>
        <network>int-mgmt-net</network>
        <ip_address>10.20.0.21</ip_address>
        <port_forwarding>
          <port>
            <type>ssh</type>
            <protocol>tcp</protocol>
            <vnf_port>22</vnf_port>
            <external_port_range>
              <start>20022</start>
              <end>20022</end>
            </external_port_range>
          </port>
        </port_forwarding>
      </interface>
    </interfaces>
    <kpi_data>
      <kpi>
        <event_name>VM_ALIVE</event_name>
        <metric_value>1</metric_value>
```

```
        <metric_cond>GT</metric_cond>
        <metric_type>UINT32</metric_type>
        <metric_collector>
          <type>ICMPPing</type>
          <nicid>0</nicid>
          <poll_frequency>3</poll_frequency>
          <polling_unit>seconds</polling_unit>
          <continuous_alarm>false</continuous_alarm>
        </metric_collector>
      </kpi>
    </kpi_data>
    <rules>
      <admin_rules>
        <rule>
          <event_name>VM_ALIVE</event_name>
          <action>ALWAYS log</action>
          <action>TRUE servicebooted.sh</action>
          <action>FALSE recover autohealing</action>
        </rule>
      </admin_rules>
    </rules>
    <config_data>
      <configuration>
        <dst>bootstrap_config</dst>
        <variable>
          <name>TECH_PACKAGE</name>
          <val>ax</val>
        </variable>
      </configuration>
    </config_data>
  </vm_group>
</deployment>'
```

VM deployment verification

Objective

This example shows how to get the operational data for a VM deployment using the command

VM deployment operational data

show vm_lifecycle opdata tenants tenant admin deployments @@VAR0@@/@@VAR1@@/@@VAR2@@

```
nfvis# show vm_lifecycle opdata tenants tenant admin deployments ROUTER

deployments ROUTER

 deployment_id SystemAdminTenantIdROUTER

 vm_group ROUTER

  bootup_time 600

  vm_instance d1c462e9-2706-4868-befd-d8f7806b9444

   name     ROUTER

   host_id  NFVIS

   hostname nfvis
```

```
   interfaces interface 0
    model       virtio
    type        virtual
    port_id     vnic0
    network     int-mgmt-net
    subnet      N/A
    ip_address  10.20.0.2
    mac_address 52:54:00:fe:34:53
    netmask     255.255.255.0
    gateway     10.20.0.1
   interfaces interface 1
    type        virtual
    port_id     vnic1
    network     GE0-1-SRIOV-1
    subnet      N/A
    mac_address 52:54:00:e4:13:67
 state_machine state SERVICE_ACTIVE_STATE
VM
NAME    STATE
------------------------
ROUTER  VM_ALIVE_STATE
```

Note If the deployment was accepted first but failed later, NFVIS sends a notification with error status and message. Also on VM deployment failure, the operational data may or may not show the complete data about the failed VM.

VM deployment parameters

Objective

VNFs can be deployed using multiple mandatory and option parameters.

VM deployment parameter definitions

Parameter	Notes	Example
Image	Mandatory. The image needs to be registered and active when it is being referred in deployment.	<IMAGE>ISR_IMAGE</IMAGE>
bootup_time	This parameter is no longer mandatory from 3.12 and later, provided that it is specified in IMAGE properties. Accepted Values: For unmonitored VMs: -1 For monitored VMs: Number of seconds	<bootup_time>500</bootup_time>
vim_vm_name	Optional. If a custom VM name is provided at the time of deployment, it may be used for all commands that accept VM name as an input.	
kpi_data	Mandatory for monitored VMs.	
rules	Mandatory for monitored VMs.	
config_data	Mandatory if the day-0 configuration has variables that have tokens assigned to them.	
Encrypted config Variable	Optional. Only one value for a Variable is allowed to be encrypted.	<VARIABLE> <name>TEST_VARIABLE</name> <encrypted_val>test_value</encrypted_val> </VARIABLE>
placement	Optional. The placement tag under VM group points to the location where the VNF would be deployed. This parameter supports deploying a VNF in a local data store (default-if not specified), external data store (datastore2), or NFS.	<placement> <type>zone_host</type> <host>nfs:nfs_storage</host> </placement>

Parameter	Notes	Example
volumes	Optional. Up to 2 volumes could be added to a deployment. Location of the volumes can be local or NFS (needs NFS mount name to be specified in case of NFS)	
port_forwarding	Optional. If port forwarding is included, all elements under it are mandatory.	<port_forwarding> <port> <type>ssh</type> <protocol>tcp</protocol> <vnf_port>22</vnf_port> <external_port_range> <start>20022</start> <end>20022</end> </external_port_range> </port> </port_forwarding>
Ngio interface	Optional. Used in config_data. To enable NIM support on a Cisco ISRv running on Cisco ENCS, you must use the variables in the ISRv deployment payload.	<VARIABLE> <name>ngio</name> <val>enable</val> </VARIABLE>
interface model	Optional. If the model is not specified for an interface, the default model is used. For Windows, the default model is rtl8139.	[illegible]
VNC	Optional. If the VNC password is not specified, there is no default password.	<VNC><password>vnc_password</password></VNC>

VM bootstrap configuration options with VM deployment

Objective

VM bootstrap configuration options with VM deployment are methods that

VM bootstrap configuration options

- enable inclusion of bootstrap configuration (day zero configuration) of a VM in the VM deployment payload
- provide flexibility in how configuration files are handled during deployment, and
- support different approaches based on deployment requirements and infrastructure setup.

Bootstrap configuration methods

Objective

You can include the bootstrap configuration (day zero configuration) of a VM in the VM deployment payload in these ways:

Bootstrap configuration methods

- Bundle bootstrap configuration files into the VM package: In this method, the bootstrap configuration variables can be assigned tokens. Token names must be in bold text. For each variable with a token assigned, key-value pairs must be provided during deployment in the deployment payload.
- Bootstrap configuration as part of the deployment payload: The entire bootstrap configuration is copied to the payload without tokens.
- Bootstrap configuration file in the NFVIS server: In this method, the configuration file is copied or downloaded to the NFVIS server, and referenced from the deployment payload with the filename, which includes the full path.

For examples on how to use bootstrap configuration options in the deployment payload, see the API Reference for Cisco Enterprise Network Function Virtualization Infrastructure Software.

VNF deployment placement

VNF deployment placement is a parameter mechanism that

- allows you to specify where a VNF should be deployed using the placement tag for parameter deployment
- supports various placement parameters with accepted values, and
- enables precise control over VNF location within the Cisco NFVIS environment.

Placement configuration requirements

Objective

If you are placing the VNF deployment on nfs, ensure that you have configured this storage option to be mounted on NFVIS using appropriate CLIs before deploying the VNF.

VNF placement configuration examples

Example: VM deployment using placement

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST
https://209.165.201.1/api/config/vm_lifecycle/tenants
/tenant/admin/deployments --data
'<deployment>
<name>WINIsodep</name>
<vm_group>
  <name>WINIsovmgrp</name>
  <image>WinServer2012R2.iso</image>
  <flavor>windows</flavor>
  <bootup_time>-1</bootup_time>
  <recovery_wait_time>0</recovery_wait_time>
  <kpi_data>
   <enabled>true</enabled>
  </kpi_data>
   <scaling>
   <min_active>1</min_active>
   <max_active>1</max_active>
   <elastic>true</elastic>
   </scaling>
   <placement>
   <type>zone_host</type>
   <enforcement>strict</enforcement>
   <host>datastore1</host>
   </placement>
   <recovery_policy>
   <recovery_type>AUTO</recovery_type>
   <action_on_recovery>REBOOT_ONLY</action_on_recovery>
   </recovery_policy>
</vm_group>
</deployment>'
```

Configure granular RBAC for VM deployment

The system administrator can define a set of groups and assign VNFs to these groups. When you create a user, you can assign that user to one of these groups to control VNF access.

Procedure

Step 1 Create a VM with a resource group.

```
nfvis(config)# vm_lifecycle tenants tenant admin deployments deployment sample resource_group
localgroup2 vm_group test image centos flavor small vim_vm_name sample bootup_time -1 recovery_wait_time
 0 recovery_policy recovery_type AUTO action_on_recovery REBOOT_ONLY monitor_on_error false
nfvis(config-vm_group-test)# commit
```

Step 2 Move the VM to another resource group.

```
nfvis(config)# vm_lifecycle tenants tenant admin deployments deployment sample

nfvis(config-deployment-sample)# no resource_group

nfvis(config-deployment-sample)# commit

Commit complete.

nfvis(config-deployment-sample)# resource_group localgroup

nfvis(config-deployment-sample)# commit

Commit complete.
```

Step 3 View the VMs and their resource group assignments.

```
nfvis# show vm_lifecycle deployments

 Name           Monitored  State           Internal-State           GroupInfo

-------------------------------------------------------------------------------------

centos1        No         ALIVE           VM_INERT_STATE           localgroup

cetos_global   No         ALIVE           VM_INERT_STATE                                Denotes global


sample         No         ALIVE           VM_INERT_STATE           localgroup
```

Step 4 View the resource groups, users, and VM mappings.

Example:

```
nfvis# show group-summary

 Group_Name             Policies                    Users                    Vms

-------------------------------------------------------------------------------------------------------


localgroup         resource-access-control        local_oper            centos1.centos1

                                                  sample.test

-------------------------------------------------------------------------------------------------------


localgroup2        resource-access-control         test_admin

-------------------------------------------------------------------------------------------------------


                           NA                       admin            cetos_global.cetos_global

                                                   localAdmin
```

The VNF is deployed with granular RBAC configuration, and you can view the resource group assignments and user mappings for proper access control.

Additional capabilities

Objective

This reference groups information about Additional Capabilities, including Deploy CUCM application, CDROM attachment and detachment, Recover the password for a Cisco CUCM application, Graceful VM stop.

Deploy CUCM application

This procedure follows successful registration of the CUCM application image in NFVIS. The deployment creates a functional CUCM virtual machine with automated initial configuration.

Before you begin, ensure that you have:

- A registered CUCM application image (ISO and OVA) in NFVIS.
- A Bootstrap ISO with Day0 configuration for the CUCM application.

For instructions on creating the Bootstrap ISO with Day0 configuration, see the CUCM documentation: Touchless Installation Task Flow.

Procedure

Step 1 Open the deployment page in the NFVIS portal.

From the NFVIS portal, click **Configuration** > **Deploy**.

Step 2 Select a VM type.

Select a VM type from the drop-down list.

If your application type is not listed, select **OTHER**.

Step 3 Select the CUCM image.

From the **Image** drop-down list, select the desired image. Images are populated based on the VM type selected.

For example, `Bootable_UCSInstall_UCOS_15.0.1.ISO`.

Step 4 Select a flavor.

From the **Flavor** drop-down list, select the appropriate flavor based on your sizing requirements. Flavors are automatically created from the OVA during image registration.

Step 5 Configure the network connections.

In the **Network Design** section, locate the **VM icon** on the canvas.

Add additional networks as required.

Drag a line from the VM icon to the desired network to connect the VM to the appropriate networks for your deployment.

Step 6 Select the Bootstrap ISO option.

Select the **Bootstrap ISO** check box.

Step 7 Upload the Day0 configuration ISO.

Click **Upload** and select your Day0 configuration ISO.

Step 8 Deploy the virtual machine.

Click **Deploy**.

The system validates the configuration. If validation errors occur, they are displayed for correction.

Step 9 Monitor the deployment and complete the installation.

Track the deployment progress and wait until the VM state changes to ACTIVE.

After the VM is ACTIVE, click the **Terminal** button in the VM row. Monitor the installation progress in the console and follow the application-specific installation prompts.

Table 2: Virtual machine deployment statuses

Status	Description	Action
DEPLOYING	VM is being created.	Wait for completion.
ACTIVE	VM is running.	Access the console and verify operation.
ERROR	Deployment failed.	Check logs and troubleshoot.
SHUTDOWN	VM is powered off.	Power on the VM if needed.

Note The deployment steps for Cisco Unity Connection and Cisco IM and Presence Service (IM&P) on NFVIS are the same as those for Cisco Unified Communications Manager (CUCM). These steps include:

Image registration: Register the ISO and OVA metadata using the same process as CUCM.

Virtual machine deployment: Deploy the VM with a Bootstrap ISO using the same process as CUCM.

Virtual machine management: Perform power operations, console access, and network updates as for CUCM.

For application-specific information, refer to the respective Cisco documentation for Cisco Unity Connection and Cisco IM and Presence Service.

The CUCM virtual machine is deployed on NFVIS with the VM status showing as ACTIVE, and the Day0 configuration from the Bootstrap ISO is applied automatically during installation.

CDROM attachment and detachment

CDROM attachment and detachment is a VM management capability that

- allows you to connect or disconnect ISO images to a VM

- supports various use cases, including system recovery, operating system installation, and software deployment, and
- enables the VM to access content from a virtual CDROM drive.

How CDROM attachment and detachment work

Summary

Attaching a CDROM to a VM is a two-step process:

Workflow

1. Image Registration: First, you must register the desired ISO image using the existing image registration workflows provided by NFVIS. This makes the ISO available for selection. For information about ISO image registration, see Image Registration.
2. CDROM Action: After registration, you can proceed to attach the registered ISO to the target VM by utilizing the dedicated CDROM actions available in the **Manage Deployments** page.

Process details

To detach an ISO image, you would use the corresponding detachment action in the **Manage Deployments** page. Both attachment and detachment operations require the VM to be in a shutdown state.

Requirement: prerequisites for CDROM attachment and detachment

To successfully perform CDROM attachment or detachment operations, the VM must be in a shutdown state. These operations cannot be initiated or completed on an active VM.

Manage CDROM attachment and detachment

Attach or detach ISO images to and from virtual machines for recovery and maintenance operations.

CDROM attachment and detachment operations allow you to mount ISO images to virtual machines for recovery purposes or maintenance tasks. These operations can only be performed when the VM is in a shutdown state.

- Ensure the ISO image you intend to attach is registered using the image registration workflows. For more information, see Image Registration.
- Verify that the VM is in a shutdown state. CDROM attach and detach operations cannot be performed on an active VM.

Follow these steps to manage CDROM attachment and detachment:

Procedure

Step 1 From the **Manage Deployments** page, click the **CD-ROM** icon.

Step 2 Choose the operation you want to perform.

To attach an ISO image, click **Attach**.

To detach an ISO image, click **Detach**.

Step 3 Select the appropriate image or disk.

Table 3: CDROM attachment and detachment options

If...	Then...
You are attaching a CDROM	From the **Images** drop-down list, choose the desired recovery ISO from the list of registered ISO images.
You are detaching a CDROM	From the **Disk Name** drop-down list, choose the disk name that corresponds to the recovery ISO you want to detach.

Step 4 Click **Submit**.

A confirmation message appears indicating the CDROM operation is complete.

After attaching a CDROM, to use the attached recovery ISO, you must start the VM and access its boot menu to select the ISO disk for booting. After detaching a CDROM, you can start the VM for normal operation. To confirm detachment, click **CD-ROM** > **Detach**. The ISO image should no longer be listed in the **Disk Name** drop-down list.

Recover a Cisco CUCM application using a recovery ISO

This task allows you to recover or repair a Cisco CUCM application when the virtual machine requires restoration from a corrupted or damaged state.

Use this procedure when a Cisco CUCM application needs recovery or repair. The process involves using recovery ISO media to restore the application to a functional state through the NFVIS portal interface.

Ensure you have access to the NFVIS portal and the appropriate recovery ISO file for your CUCM application version.

Follow these steps to recover a Cisco CUCM application using a recovery ISO:

Procedure

Step 1 From the NFVIS portal, click **Configuration** > **Virtual Machine** > **Images** > **Image Repository**.

Step 2 Upload the Recovery ISO.

a) If Recovery ISO registration is not allowed, upload the respective VM OVA and register it.
b) Name the image descriptively. For example, CUCM-15-recovery-ISO.

Step 3 Shut down the VM.

a) Click **Configuration** > **Virtual Machine** > **Manage**.
b) Select the VM.
c) Click **Switch Power** to shut down the VM.
d) Wait until the VM state changes to SHUTDOWN.

Step 4 Attach the Recovery ISO to the **CD-ROM**.

a) With the VM in the SHUTDOWN state, click the **CD-ROM** button.

b) Click **ATTACH**.
c) Select the **Recovery ISO** from the drop-down list.
d) Click **Submit**.

The Recovery ISO is attached to the VM.

Step 5 Power on and boot from the Recovery ISO.

a) Click **Switch Power** to start the VM.
b) Click **Terminal** to access the console.
c) Enter the Boot Menu by pressing the appropriate key during startup, typically F12 or ESC.
d) Select the **Recovery ISO device** from the boot menu.
e) Follow the application-specific recovery procedure in the console.

The VM boots from the Recovery ISO.

Step 6 Detach the **CD-ROM** after recovery.

a) Shut down the VM.
b) Click the **CD-ROM** button.
c) Select the **DETACH** action.
d) Select the **Recovery ISO disk**.
e) Click **Submit**.

Step 7 Power on the VM for normal operation.

The Cisco CUCM application is recovered and restored to a functional state. The VM boots normally and the application is ready for use.

Recover the password for a Cisco CUCM application

Use this procedure to reset the password for a Cisco CUCM application while the VM is in an active state.

Cisco NFVIS allows you to eject and insert ISO media, which is a required step in the password recovery process for CUCM applications. This procedure covers only the NFVIS media operations. For the complete application-specific password recovery procedure, refer to the Reset or Change CUCM OS Admin and Security Password guide in the official CUCM documentation.

Follow these steps to recover the password for a Cisco CUCM application:

Procedure

Step 1 From the NFVIS portal, choose **Configuration** > **Virtual Machine** > **Manage**.

Step 2 Verify the VM state.

Ensure the VM is running and in an ACTIVE state as password recovery is performed on a running VM.

Step 3 **Eject** the current ISO.

a) Select the VM.
b) Click the **CD-ROM** button.
c) Select **Eject**.

d) Select the ISO image to eject.
e) Click **Submit**.

Step 4 **Insert** the required ISO.

a) Click the **CD-ROM** button.
b) Select **Insert**.
c) Select the required ISO from the drop-down list.
d) Select the appropriate **Device ID** for the **CD-ROM**.
e) Click **Submit**.

Step 5 Complete the password recovery.

a) Access the VM console by clicking the **Terminal** button.
b) Follow the application-specific password recovery steps provided in the CUCM documentation.

The ISO media is successfully swapped, and the application-specific password recovery process is initiated.

Graceful VM stop

A graceful stop is a VM power management option that

- sends an ACPI signal to the operating system within the VM
- allows services and applications to terminate in an orderly manner before the VM powers off, and
- contrasts with forceful shutdown, which immediately cuts power to the VM.

Key features and behavior

- Default Behavior: When initiating a VM shutdown, the **Graceful stop** checkbox is selected by default for VMs deployed with type as OTHER.
- Timeout: The graceful stop operation has a timeout period of 20 minutes.
- Error State: If the VM does not shut down gracefully within the 20-minute timeout, it will transition to an error state.
- Forceful Fallback: If a graceful stop fails or the VM enters an error state, you can uncheck the **Graceful stop** option to perform a forceful shutdown.

To shut down a VM gracefully, see Perform a graceful stop.

Perform a graceful stop

Use a graceful stop for routine VM shutdowns to allow running processes to complete and help prevent data loss.

For VMs deployed with the type set to OTHER, the **Graceful stop** check box is selected by default. The graceful stop operation can take up to 20 minutes.

Procedure

Step 1 From the **Manage Deployments** page, click the **Switch Power** icon for the VM that you want to stop.

For VMs deployed with the type set to OTHER, the **Graceful stop** check box is selected by default.

Step 2 Click **OK** to confirm the VM shutdown.

The graceful shutdown process begins in the background.

Step 3 Monitor the VM status in the management interface.

If the VM does not shut down within 20 minutes, it moves to an error state. To perform a forceful shutdown, repeat this procedure and clear the **Graceful stop** check box.

The VM status changes from active to shutdown, indicating that the graceful stop succeeded.