CUWN 8.1 Virtual Wireless LAN Controller Deployment Guide Using Linux Kernel-based Virtual Machine (KVM)

First Published: 2015-06-16

Last Modified: 2020-12-01

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

Introduction

This chapter contains the following topics:

Introduction

Prior to release 7.3, Wireless LAN (WLAN) controller software ran on dedicated hardware you were expected to purchase. The Virtual Wireless LAN Controller (vWLC) runs on general hardware under an industry standard virtualization infrastructure. The vWLC is ideal for small and mid-size deployments with a virtual infrastructure and require an on-premises controller. Distributed branch environments can also benefit with a centralized virtual controller with fewer branches required (up to 200).

vWLCs are not a replacement of shipping hardware controllers. The function and features of the vWLC offer deployment advantages and benefits of controller services where data centers with virtualization infrastructure exist or are considered.

Advantages of the vWLC:

- Flexibility in hardware selection based on your requirements.
- Reduced cost, space requirements, and other overheads since multiple boxes can be replaced with single hardware running multiple instances of virtual appliances.

New Features in 8.1

The new features in 8.1 are:

- Support for Linux KVM
- Support for RTU licensing

Linux Kernel-based Virtual Machine (KVM)

This document is an update for vWLC based on the CUWN 8.1 software release and the support for Linux Kernel-based Virtual Machine (KVM). KVM is supported in Cisco Wireless Release 8.1.102.0 and later releases.

Note After KVM is deployed, it is recommended that you do not downgrade to a Cisco Wireless release that is older than Release 8.1.102.0.

VMware support is discussed in separate document and is not relevant in this document.

Features Not Supported on Cisco Virtual WLC

Following are some of the features not supported on Cisco vWLC:

- Internal DHCP server
- TrustSec SXP
- Access points in local mode
- Mobility/Guest Anchor
- Multicast

Note FlexConnect local-switched multicast traffic is bridged transparently for both wired and wireless on the same VLAN. FlexConnect access points do not limit traffic based on IGMP or MLD snooping.

- High Availability
- PMIPv6
- Workgroup Bridges
- Mesh (use Flex + Bridge mode for mesh-enabled FlexConnect deployments)

Note Outdoor APs in the FlexConnect mode are supported.

- Application Visibility and Control (AVC)
- Client downstream rate limiting for central switching

KVM Prerequisite for Hosting Virtual WLC (vWLC)

Following are the KVM prerequisites for hosting vWLC:

- Minimum of 2 G memory
- Minimum of 1 vCPU
- Minimum of 2 network interfaces
- Required storage of 8 G
- Network device model is "virtio"
- The physical devices connected to Open vswitch bridge should not have any IP addresses configured on it.

For more information, refer to KVM resource.

CHAPTER 2

Installing vWLC and KVM with Fedora

This chapter contains the following topics:

Installing Fedora OS

To install Fedora OS, perform the following steps:

Step 1 Install Fedora 21 or later. Click the following link to download Fedora.

https://getfedora.org/en/server/download/

Step 2 After installing Fedora, configure IP address to go to internet.

In this scenario, two dedicated Linux interfaces/ports are used for vWLC.

Step 3 Find out your interface using **ifconfig**.

Example

- 1st Interface —for uplink (service-port of WLC); no IP address is required to this interface but should be connected and up.
- 2nd Interface—for WLC Management interface; no IP address is required to this interface but should be connected and up.
- 3rd or 4th Interface—for Linux accessibility; provide IP address to this interface, so that there is a network connectivity to the Linux box.

Note By default, KVM uses first interface as service-port for vWLC.

Step 4 Configure IP address to the 3rd or 4th interface to access Linux and access internet to get update.

```
vi /etc/sysconfig/network-scripts/ifcfg-enp2s0f3
```

Note You will need to change BOOTPROTO from DHCP to static and add IPADDR, NETMASK, BROADCAST, and NETWORK variables. It is recommended to choose the static IP address.

Example

```
NM_CONTROLLED="yes"
BOOTPROTO=static
DEVICE=eth1
ONBOOT=yes
IPADDR=192.168.8.248
NETMASK=255.255.255.0
BROADCAST=192.168.8.255
NETWORK=192.168.8.0
GATEWAY=192.168.8.1
TYPE=Ethernet
PEERDNS=no
```

Step 5 Save the file.

OR

```
ifconfig <interface_name> <IP_address>
ifconfig <interface_name> netmask <netmask_address>
ifconfig <interface_name> broadcast <broadcast_address>
```

OR

```
ifconfig <interface_name> <IP_address> netmask <netmask_address> broadcast <broadcast_address>
```

Note Configure proxy and DNS information if required. Make sure internet is accessible after configuration.

Updating Fedora OS

To update Fedora OS after installation, perform the following steps:

Step 1 Update Fedora OS:

```
yum install update
```

Step 2 Install GUI:

```
yum install @gnome-desktop -y
```

Step 3 Install VNC server -- http://www.namhuy.net/3134/install-vnc-server-on-fedora-20.html:

```
yum install tigervnc-server -y
```

Step 4 Install x11:

```
yum groupinstall "X Software Development"
```

Installing KVM and openvswitch with supporting packages

```
yum install -y @standard @virtualization openvswitch
systemctl enable network.service
systemctl start network.service
systemctl enable openvswitch.service
systemctl start openvswitch.service
```

Verifying the Installation of KVM

```
lsmod | grep kvm
```

Example output on Intel processor:

```
[root@localhost system]# lsmod | grep kvm
kvm_intel             147785  0
kvm                   464964  1 kvm_intel
```

Network Configuration

Creating a Bridge and Mapping it to Port (Ethernet Interface)

```
ovs-vsctl add-br ov_10nw
ovs-vsctl add-port ov_10nw enp2s0f0
ovs-vsctl add-br ov_9nw
ovs-vsctl add-port ov_9nw en
```

The bridge name must be the same as created in the XML file.

Viewing the Bridge Mapping

```
ovs-vsctl show
```

Example

```
[root@localhost ~]# ovs-vsctl show
099e8b7e-bf00-4071-be62-ec55f9b543cc
    Bridge "ov_9nw"
        Port "ov_9nw"
            Interface "ov_9nw"
                type: internal
        Port "enp2s0f1"
            Interface "enp2s0f1"
    Bridge "ov_10nw"
        Port "ov_10nw"
            Interface "ov_10nw"
                type: internal
        Port "enp2s0f0"
            Interface "enp2s0f0"
    ovs_version: "2.3.1-git3282e51"
```

Creating XML Files

Create two XML files; one for service-nw (10nw) and the other for management (9nw).

Example

```
10nw_eth0_ov.xml
9nw_eth1_ov.xml
```

Both XML files contain VLAN information based on the network, or based on what you want to allow.

Example: To Allow All VLANs

```
<network>
  <name>10-nw</name>
  <forward mode='bridge'/>
  <bridge name='ov_10nw'/>
  <virtualport type='openvswitch'/>
  <portgroup name='vlan-any' default='yes'>
  </portgroup>
</network>
```

The bridge name must be the same as created during "ovs-vsctl" command.

If only specific VLANs need to be allowed, use the following format.

```
<network>
  <name>ov-nw</name>
  <forward mode='bridge'/>
  <bridge name='bridge_1'/>
  <virtualport type='openvswitch'/>
  <portgroup name='all_vlans' default='yes'>
  </portgroup>
  <portgroup name='vlan-152-untagged'>
    <vlan>
        <vlan mode='native-untagged'/>
   <tag id='152'/>
    </vlan>
  </portgroup>
  <portgroup name='vlan-153'>
    <vlan>
      <tag id='153'/>
    </vlan>
  </portgroup>
  <portgroup name='two-vlan'>
    <vlan trunk='yes'>
      <tag id='152'/>
      <tag id='153'/>
    </vlan>
  </portgroup>
</network>
```

In the above configuration:

- portgroup name='all_vlans' → allows all VLANs.
- portgroup name='vlan-152-untagged' → allows only untagged VLAN that is 152.
- portgroup name='vlan-153' → allows only 153 VLAN.
- portgroup name='two-vlan' → allows only two VLANs, that is, 152 and 153.

Allowing CDP Packets to Forward from Open vSwitch

```
ovs-vsctl set bridge ov_9nw other-config:forward-bpdu=true
```

Viewing the Virtual Network

```
virsh net-list --all
```

Deleting the Default Network

```
virsh net-undefine default
```

Creating Virtual Network

```
virsh net-define <xml_file_name>
```

Viewing the Virtual Network

```
virsh net-list --all
```

Starting the Virtual Network

```
virsh net-start <network_name_that is in the list>
```

Example

```
[root@localhost ~]# virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 default              inactive   no            yes

[root@localhost ~]# virsh net-undefine default
Network default has been undefined

[root@localhost ~]# virsh net-define 10nw_eth0_ov.xml
Network 10-nw defined from 10nw_eth0_ov.xml

[root@localhost ~]# virsh net-define 9nw_eth1_ov.xml
Network 9-nw defined from 9nw_eth1_ov.xml

[root@localhost ~]# virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 10-nw                inactive   no            yes
 9-nw                 inactive   no            yes

[root@localhost ~]# virsh net-start 10-nw
Network 10-nw started

[root@localhost ~]#
[root@localhost ~]# virsh net-start 9-nw
Network 9-nw started

[root@localhost ~]# virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 10-nw                active     no            yes
 9-nw                 active     no            yes
```

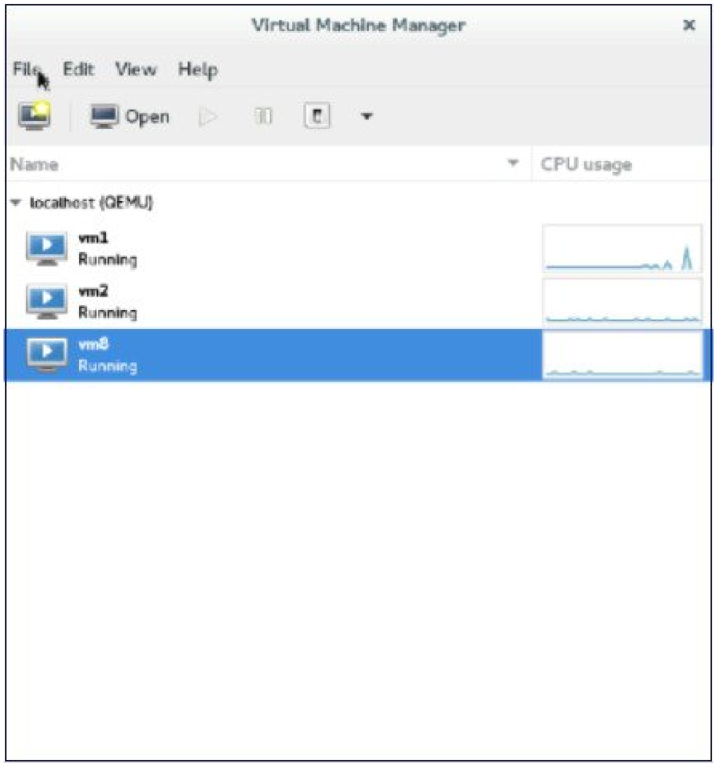
Installing vWLC Using Virtual Machine Manager (VMM)

To install vWLC using VMM in Fedora, perform the following steps:

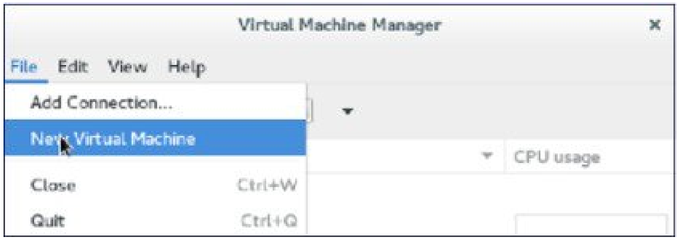
Note Console to Fedora. GUI is required for VMM.

Step 1 Open the terminal (command prompt).

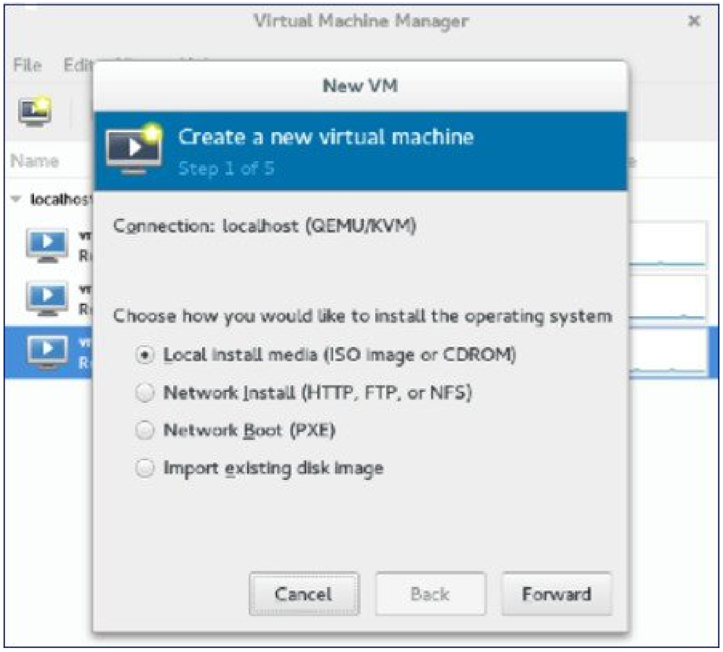
Step 2 Execute the command **virt-manager**.
The Virt Manager (VMM) pop-up window appears.



Step 3 Create a new virtual machine (VM).



Step 4 Select the path.



New VM

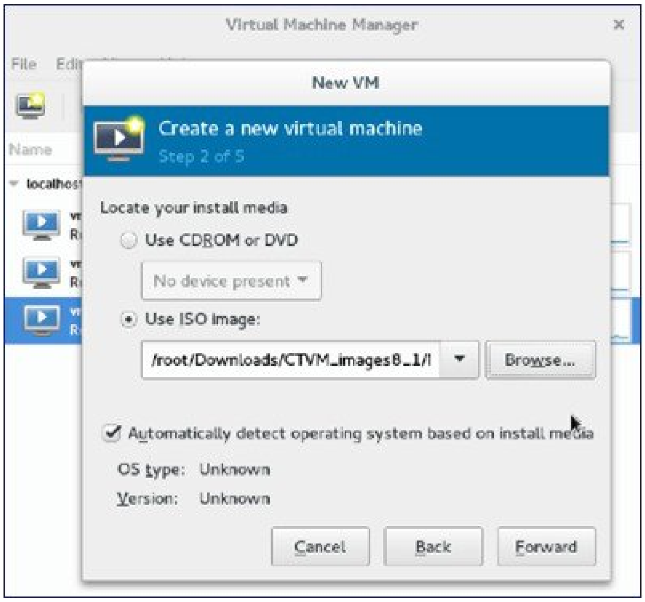
Create a new virtual machine
Step 1 of 5

Connection: localhost (QEMU/KVM)

Choose how you would like to install the operating system

- ● Local install media (ISO image or CDROM)
- ○ Network Install (HTTP, FTP, or NFS)
- ○ Network Boot (PXE)
- ○ Import existing disk image

Step 5 Select the ISO file of vWLC.



New VM

Create a new virtual machine
Step 2 of 5

Locate your install media

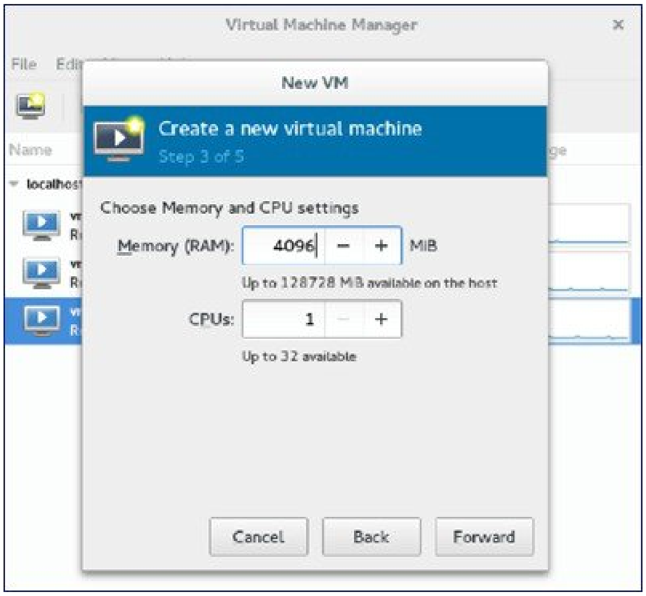
- ○ Use CDROM or DVD
  - No device present
- ● Use ISO image:
  - /root/Downloads/CTVM_images8_1/I

☑ Automatically detect operating system based on install media

OS type: Unknown

Version: Unknown

Step 6 Select the memory and CPU.



Create a new virtual machine
Step 3 of 5

Choose Memory and CPU settings

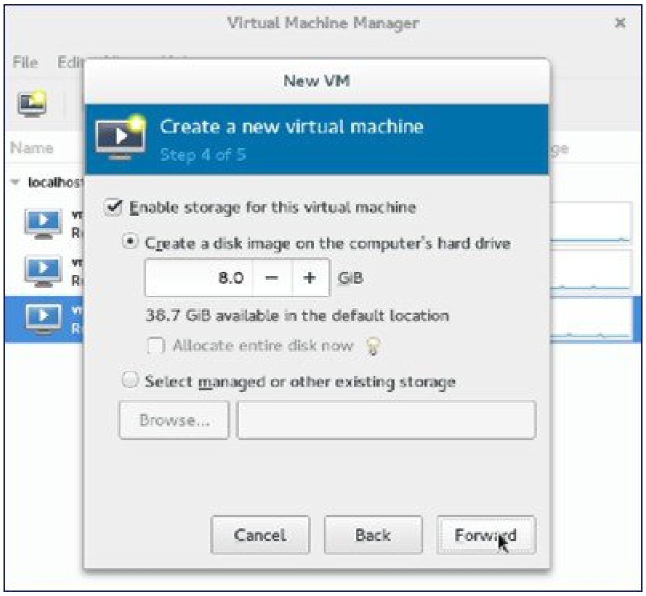
Memory (RAM): 4096 MiB

Up to 128728 MiB available on the host

CPUs: 1

Up to 32 available

Step 7 Select the disk space.



Create a new virtual machine
Step 4 of 5

☑ Enable storage for this virtual machine

● Create a disk image on the computer's hard drive

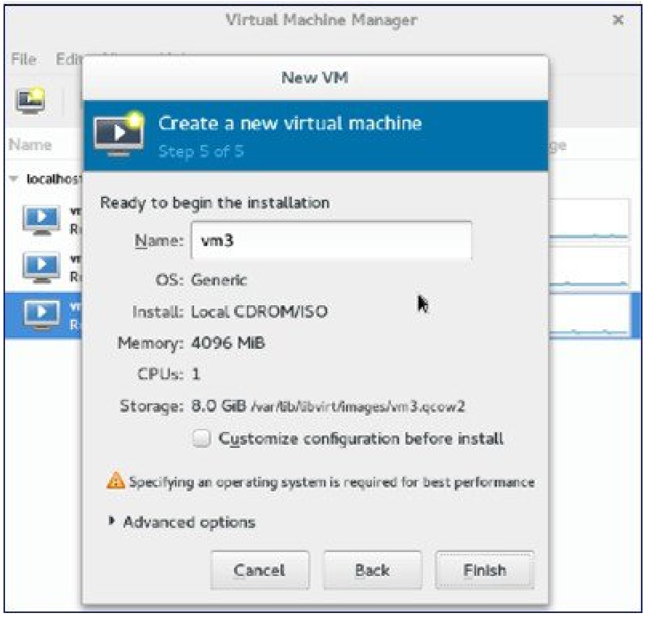
8.0 GiB

38.7 GiB available in the default location

□ Allocate entire disk now

○ Select managed or other existing storage

Step 8 Name the VM.



Virtual Machine Manager

New VM

Create a new virtual machine
Step 5 of 5

Ready to begin the installation

Name: vm3

OS: Generic

Install: Local CDROM/ISO

Memory: 4096 MiB

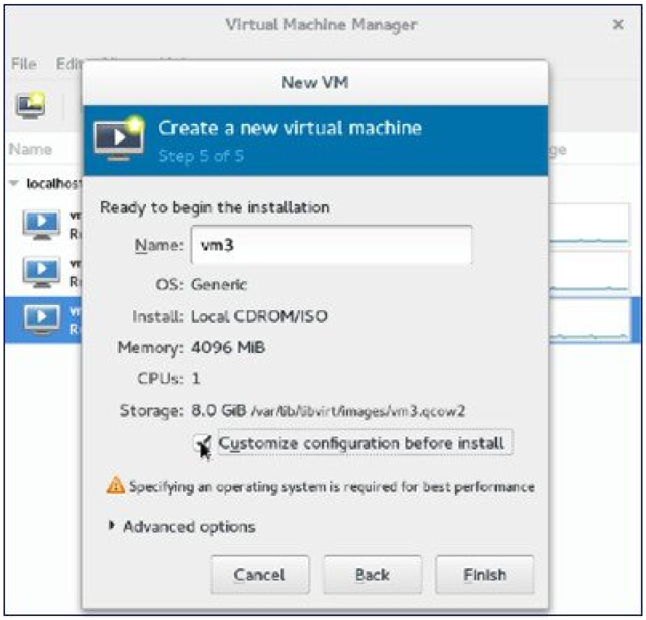
CPUs: 1

Storage: 8.0 GiB /var/lib/libvirt/images/vm3.qcow2

☐ Customize configuration before install

Specifying an operating system is required for best performance

Step 9 Check the **Customize configuration before install** check box and then click **Finish**. (This helps to configure other options)



Virtual Machine Manager

New VM

Create a new virtual machine
Step 5 of 5

Ready to begin the installation

Name: vm3

OS: Generic

Install: Local CDROM/ISO

Memory: 4096 MiB

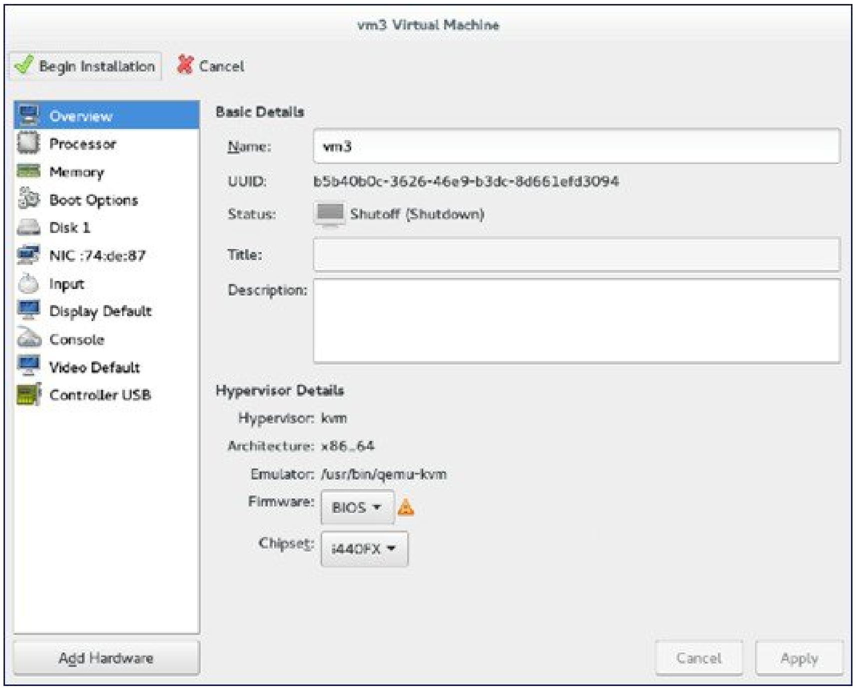
CPUs: 1

Storage: 8.0 GiB /var/lib/libvirt/images/vm3.qcow2

☑ Customize configuration before install

Specifying an operating system is required for best performance

Step 10 Click **Add Hardware**.



Basic Details

Name: vm3
UUID: b5b40b0c-3626-46e9-b3dc-8d661efd3094
Status: Shutoff (Shutdown)
Title:
Description:

Hypervisor Details

Hypervisor: kvm
Architecture: x86_64
Emulator: /usr/bin/qemu-kvm
Firmware: BIOS
Chipset: i440FX

The Add New Virtual Hardware window appears.

This window helps you to configure service port, management interface, and serial connection:

a. Click **Network** and do the following:

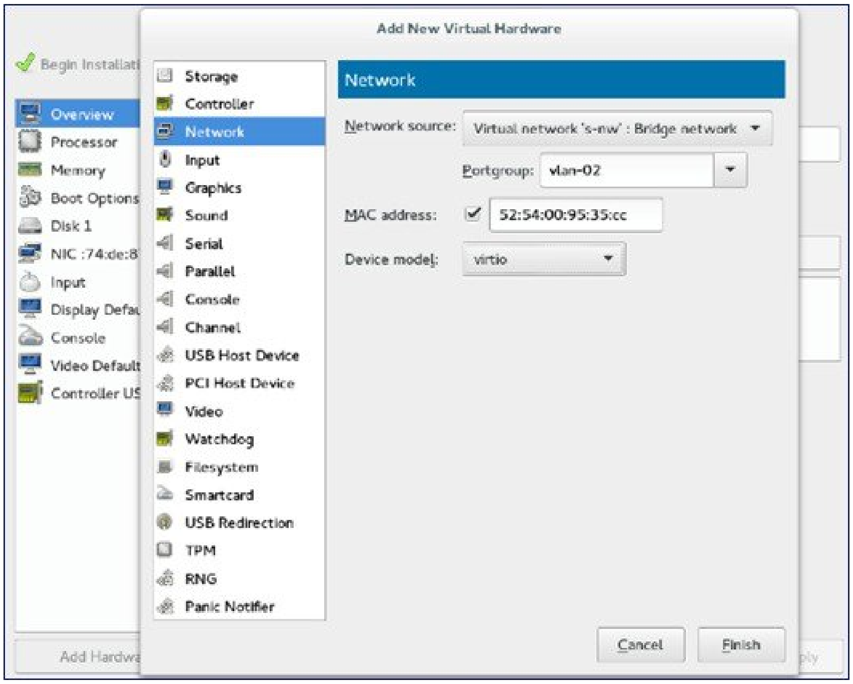
- From the **Network source** drop-down list, choose the virtual network. (It is recommended to select the virtual network of service port of vWLC)
- From the **Portgroup** drop-down list, choose the portgroup configured in xml if there are many.
- From the **Device model** drop-down list, choose **virtio** (only this is supported as of now) and then click **Finish**.

b. Repeat again by selecting **Add Hardware** > **Network** for virtual network of management interface.

Note vWLC supports only two physical ports; one for service port and the other for management/dynamic interface. The management interface is mapped to management/dynamic interface.

c. Click **Add Hardware** > **Serial** and then click **Finish**.

Note Fedora 21 has "Virt-Manager" version 1.1, which has the **portgroup** option. Older version does not have it.



Add New Virtual Hardware

Network

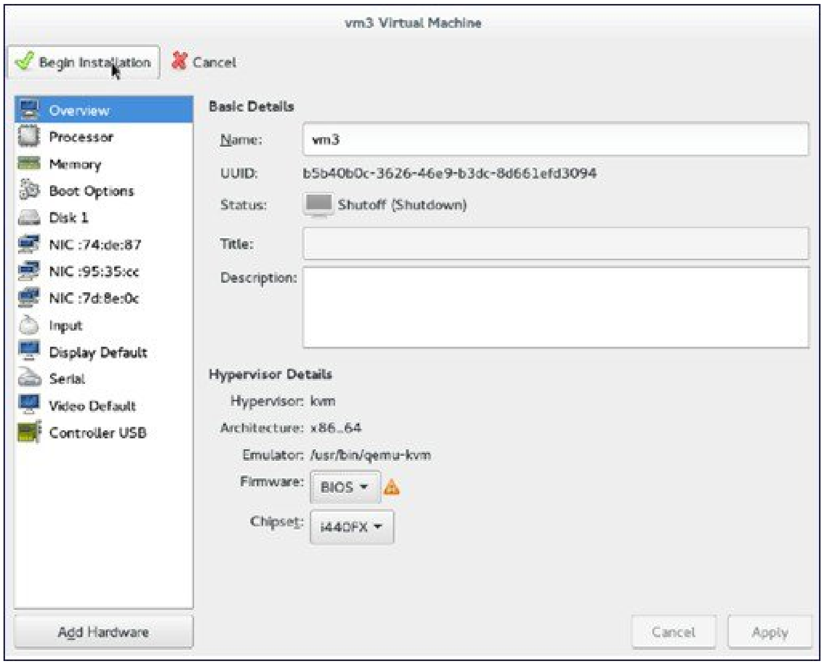
Network source: Virtual network 's-nw' : Bridge network

Portgroup: vlan-02

MAC address: ☑ 52:54:00:95:35:cc

Device model: virtio

Step 11 Click **Begin Installation**.



vm3 Virtual Machine

Basic Details

Name: vm3

UUID: b5b40b0c-3626-46e9-b3dc-8d661efd3094

Status: Shutoff (Shutdown)

Title:

Description:

Hypervisor Details

Hypervisor: kvm

Architecture: x86_64

Emulator: /usr/bin/qemu-kvm

Firmware: BIOS

Chipset: i440FX

Step 12 Wait for WLC prompt for initial configuration.

Accessing vWLC's Console

To access vWLC's console, perform the following steps:

Step 1 From the terminal, execute the following command:

```
virsh console <vm_name eg. vm1>
```

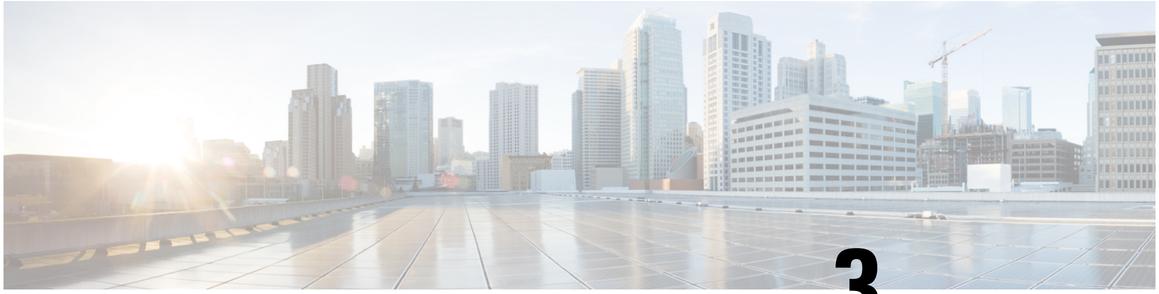
Step 2 Reboot vWLC through virt-manager.

Note To find out the vnet mapped to vWLC, execute the following command on vWLC:

```
show interface detail management
```

Match the last six octet with "ifconfig" output.

This is how, you get your targeted "vnet", if there are multiple vWLCs configured.



CHAPTER 3

Installing vWLC and KVM with Ubuntu

This chapter contains the following topics:

Installing Ubuntu and KVM

To install Ubuntu and KVM, perform the following steps:

Step 1 Install Ubuntu Server 13.10 or later, select virtualization module/package during installation.

Step 2 Install QEMU/KVM/Open vSwitch packages:

```
apt-get install qemu-kvm qemu-utils uml-utilities bridge-utils socat vnc4server vncviewer
apt-get install kvm libvirt-bin virtinst
apt-get install openvswitch-controller  openvswitch-switch openvswitch-datapath-source
```

Step 3 Start the open vswitch service.

```
service openvswitch-switch start
```

Step 4 Reboot the system.

One Time Network Configuration on Ubuntu

To execute one time network configuration on Host Linux, perform the following steps:

Step 1 Create two open vswitch bridges and map eth0, eth1 to the corresponding bridges:

```
ovs-vsctl add-br ovsbr0 [bridge name]
ovs-vsctl add-port ovsbr0 eth0

ovs-vsctl add-br ovsbr1 [bridge name]
ovs-vsctl add-port ovsbr1 eth1
```

```
ovs-vsctl set bridge ovsbr1 other-config:forward-bpdu=true [Required for CDP packets forwarding from
 Open Vswitch]
```

Step 2 To define a management network, create an XML file [mgmt.xml] as follows:

```
<network>
  <name>VM-Mgmt-Nw</name>
  <forward mode='bridge'/>
  <bridge name='ovsbr0'>
  <virtualport type='openvswitch'/>
    <!--
      If the linux host port[For eg, eth1] is connected in trunk mode to the
      downstream switch[which is also connected to the openvswitch bridge ovsbr],
      then by choosing the following portgroup,traffic from all vlan is passed up
      to the vWLC.The management interface should be in vlan tagged mode. And
      multiple interfaces can also be created with different vlans.
      If the linux host port is connected in untagged mode to the downstream switch,
      then on choosing this portgroup,untagged frames are passed up to the vWLC.
      Hence management interface has to be untagged.
    -->
    <portgroup name='default-portgroup' default='yes'>
    </portgroup>
   <!--
      If the linux host port is connected in trunk mode to the downstream switch
      [which is also connected to openvswitch bridge ovsbr],and if only certain
      vlans are to be allowed, choose this portgroup.
      Uncomment the following portgroup and edit the tag ids to the vlans allowed.
      You are free to add as many vlan ids as needed.
    -->
    <!--
    <portgroup name='Management-Portgroup'>
      <vlan trunk='yes'>
        <tag id='4092'/>
        <tag id='4093'/>
      </vlan>
    </portgroup>
    -->
</network>
```

Note Edit the vlan tags as per requirement.

Step 3 Run the following commands to create the management network:

```
 virsh net-define mgmt.xml
 virsh net-start  VM-Mgmt-Nw
```

Step 4 Repeat step 2 for creating a service port network. To define the service port network, create an XML file [service.xml] as follows:

```
<network>
  <name>VM-SP-Nw</name>
  <forward mode='bridge'/>
  <bridge name='ovsbr1'>
  <virtualport type='openvswitch'/>
    <!--
        If this portgroup is chosen, it is presumed that the linux host port
        [For eg :eth0, connected to the openvswitch bridge "ovsbr"] is connected in access
        mode to the neighboring switch.
    -->
    <portgroup name='default-portgroup' default='yes'>
    </portgroup>
   <!--
        If the same linux host port[connected to the openvswitch bridge "ovsbr"]
        as that of management interface is mapped to Service interface in vWLC
```

```
         and if the linux host port is in trunk mode ,then choose the following
         portgroup to have untagged packets for service port access.
         Uncomment the following portgroup and create the network.
         Also, edit the native-vlan as per your network settings.
   -->
   <!--
    <portgroup name='Service-portgroup'>
      <vlan>
        <vlan mode='native-untagged'/>
          <tag id='4094'/>
      </vlan>
    </portgroup>
    -->
</network>
```

Note Edit the vlan tags as per requirement.

Step 5 Run the following commands to create the service network:

```
virsh net-define service.xml
virsh net-start VM-SP-Nw
```

Step 6 Check the virtual network status by using the following command:

```
virsh net-list --all
```

All the created networks are listed as active.

Launching vWLC Using VMM

To launch vWLC using VMM, perform the following steps:

Note This is similar to installation using Fedora.

Step 1 Launch Virtual Machine Manager (VMM):

a) Launch VMM from GUI or type **virt-manager** from shell.
The GUI takes you through the following steps to create the vWLC instance easily.
b) Choose an ISO image.
c) Choose Memory = 4 GB.
d) Choose CPU = 1.
e) Provide a qcow2 image or raw image.
f) Click **Customize configuration before install**.
g) Click **NIC**, change device model to **virtio**, and change host device to **VM-Service-Network**.
h) Click **Add hardware**.
i) In the new window, click **Network**, and change host device to **VM-Mgmt-network** and device model to **virtio**.
j) Click **Begin installation**.

Step 2 From the command prompt, vWLC can be instantiated from shell as well with the following command (modify the filename and path as needed):

```
virt-install --connect=qemu:///system --network=network:VM-Service-network,model=virtio
--network=network:VM-Mgmt-network,model=virtio --name=vm1
--cdrom=/home/user/vWLC/images/<AS_CTVM_8_1_xx_xx.iso> --disk path=/var/lib/libvirt/images/4.img,size=8
--ram 2048 --vcpus=1 --vnc --vncport=5926
```

Accessing vWLC

Virtual WLC (vWLC) can be accessed in following ways:

Step 1 virsh console <Virtual Machine name>

Step 2 VNCviewer: For example, check the VNC details of vWLC through"virsh vncviewer <VirtualMachine name>" and then use that VNC connection details and access vWLC as "vncviewer 127.0.0.1:11".

Step 3 Console from VMM.

CHAPTER 4

Installing vWLC and Host Linux with SUSE Linux

This chapter contains the following topics:

Installing SUSE Linux

Download SLEs 12 - https://www.suse.com. (You must create a login)

- eth0—for uplink (service-port of WLC); no IP address is required to this interface but should be connected and up.
- eth1—for WLC Management interface; no IP address is required to this interface but should be connected and up.
- eth2 or 3— for Linux accessibility; provide IP address to this interface, so that there is a network connectivity for Linux box and internet from it.

Note Before working on any other package or KVM/vswitch, check the Linux kernel. Make sure the kernel version is 3.12.36-38 or above.

If the kernel version is not 3.12.36-38 or above, upgrade it by performing the following steps:

1. Install SLES 12 on the server.
2. Once the server comes up, copy the kernel rpm to the machine.
3. On a terminal, execute **rpm --ivh <kernel>.rpm**.

   The rpm is installed and would take some time to configure. You need not do anything else.
4. Reboot the machine once the installation is complete, and verify that the latest kernel is loaded using **uname --a**.

Install KVM and Supporting packages

Install KVM and supporting packages using the following commands:

```
zipper install openvswitch openvswitch-switch
zypper install kvm libvirt libvirt-python qemu virt-manager
```

Enabing SSH

Execute the following commands:

```
systemctl enable sshd.service → enabling sshd daemon
systemctl start sshd.service → starting ssh
netstat -an | grep :22 → to see if port# 22 is listening
```

Network Configuration

Creating a Bridge and Mapping it to Port (Ethernet Interface)

```
ovs-vsctl add-br ov_10nw
ovs-vsctl add-port ov_10nw eth0
ovs-vsctl add-br ov_9nw
ovs-vsctl add-port ov_9nw eth1
```

The bridge name must be the same as created in the XML file.

Viewing the Bridge Mapping

```
ovs-vsctl show
```

Example

```
linux-f8es:~ # ovs-vsctl show
51600b63-b508-45b0-9d0c-9f74036114c5
    Bridge "ov_9nw"
        Port "ov_9nw"
            Interface "ov_9nw"
                type: internal
        Port "eth1"
            Interface "eth1"
    Bridge "ov_10nw"
        Port "ov_10nw"
            Interface "ov_10nw"
                type: internal
        Port "eth0"
            Interface "eth0"
    ovs_version: "2.1.2"
```

Creating XML Files

Create two XML files; one for service-nw (10nw) and the other for management (9nw).

Example

```
10nw_eth0_ov.xml
9nw_eth1_ov.xml
```

Both XML files contain VLAN information based on the network, or based on what you want to allow.

Example: To Allow All VLANs

```
<network>
  <name>10-nw</name>
  <forward mode='bridge'/>
  <bridge name='ov_10nw'/>
  <virtualport type='openvswitch'/>
  <portgroup name='vlan-any' default='yes'>
  </portgroup>
</network>
```

The bridge name must be the same as created during "ovs-vsctl" command.

Starting Open vSwitch

```
service openvswitch-switch start
```

Configuring Open vSwitch to Start When the System Boots

```
chkconfig openvswitch-switch on
```

Note vSwitch must be started before creating the bridge using above command.

Starting libvirt

```
service libvirtd restart
```

Allowing CDP Packets to Forward from Open vSwitch

```
ovs-vsctl set bridge ov_9nw other-config:forward-bpdu=true
```

Viewing the Virtual Network

```
virsh net-list --all
```

Deleting the Default Network

```
virsh net-undefine default
```

Creating Virtual Network

```
virsh net-define <xml_file_name>
```

Viewing the Virtual Network

```
virsh net-list --all
```

Starting the Virtual Network

```
virsh net-start <network_name_that is in the list>
```

Example

```
linux-f8es:~ # virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 default              inactive   no            yes

linux-f8es:~ # virsh net-undefine default
Network default has been undefined

linux-f8es:~ # virsh net-define 10nw_eth0_ov.xml
Network 10-nw defined from 10nw_eth0_ov.xml

linux-f8es:~ # virsh net-define 9nw_eth1_ov.xml
Network 9-nw defined from 9nw_eth1_ov.xml

linux-f8es:~ # virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 10-nw                inactive   no            yes
 9-nw                 inactive   no            yes

linux-f8es:~ # virsh net-start 10-nw
Network 10-nw started

linux-f8es:~ #
linux-f8es:~ # virsh net-start 9-nw
Network 9-nw started

linux-f8es:~ # virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 10-nw                active     no            yes
 9-nw                 active     no            yes
```

Installing vWLC Using VMM

To install vWLC using VMM in SUSE Linux, perform the following steps:

Step 1 Similar to Fedora, go to the terminal and type **virt-manager**.

The Virt Manager (VMM) pop-up appears.

Step 2 Follow the steps covered in Installing vWLC Using Virtual Machine Manager (VMM) using VMM.

CHAPTER 5

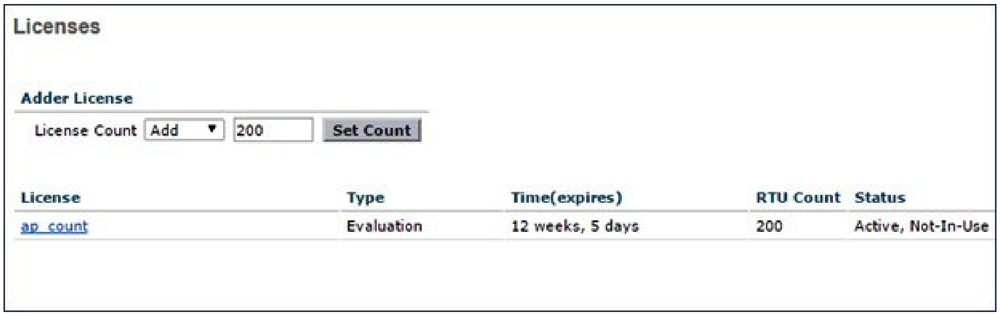
RTU Licensing

This chapter contains the following topics:

RTU Licensing Using Web GUI

Step 1 To install AP adder licenses, click **Management** > **Software Activation** > **Licenses**.

Step 2 In the **Adder License** area, in the **License Count** field, set the license task to **Add**, enter the number of AP licenses you have purchased for the vWLC, and then click **Set Count**.



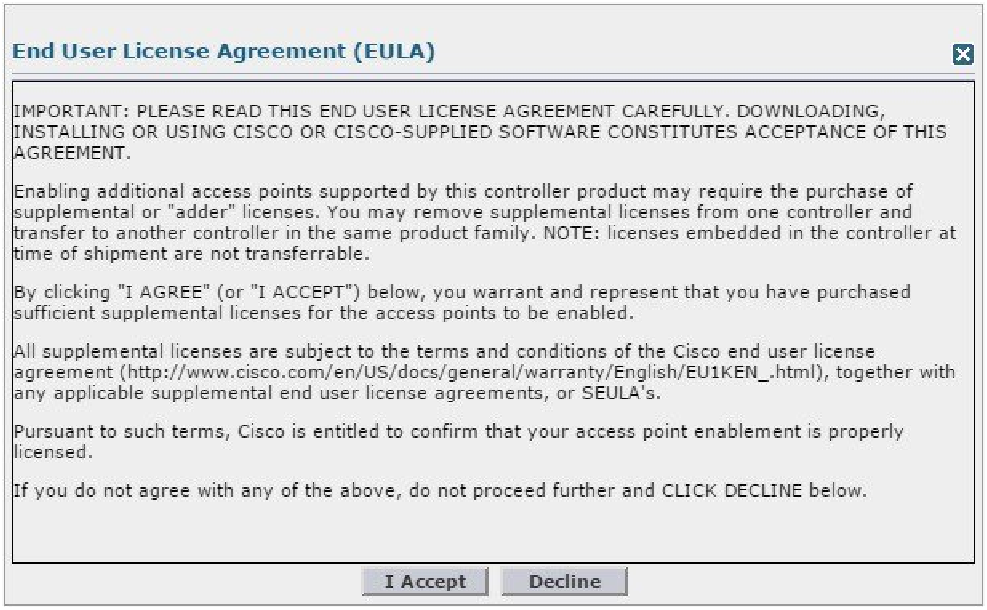
Licenses

Adder License

License Count Add ▼ 200 Set Count

License	Type	Time(expires)	RTU Count	Status
ap_count	Evaluation	12 weeks, 5 days	200	Active, Not-In-Use

Step 3 Read the **End User License Agreement (EULA)** and click **I Accept**.



End User License Agreement (EULA)

IMPORTANT: PLEASE READ THIS END USER LICENSE AGREEMENT CAREFULLY. DOWNLOADING, INSTALLING OR USING CISCO OR CISCO-SUPPLIED SOFTWARE CONSTITUTES ACCEPTANCE OF THIS AGREEMENT.

Enabling additional access points supported by this controller product may require the purchase of supplemental or "adder" licenses. You may remove supplemental licenses from one controller and transfer to another controller in the same product family. NOTE: licenses embedded in the controller at time of shipment are not transferrable.

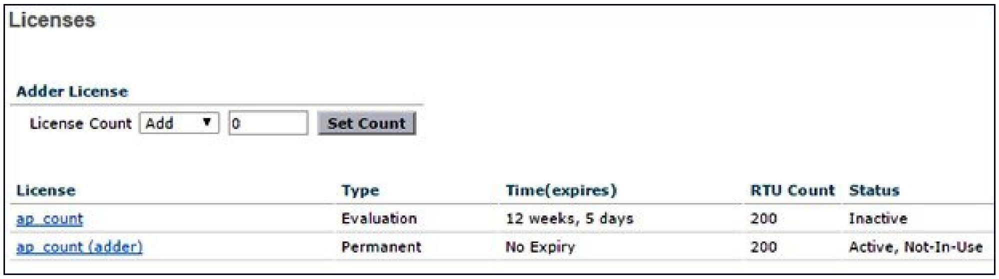
By clicking "I AGREE" (or "I ACCEPT") below, you warrant and represent that you have purchased sufficient supplemental licenses for the access points to be enabled.

All supplemental licenses are subject to the terms and conditions of the Cisco end user license agreement (http://www.cisco.com/en/US/docs/general/warranty/English/EU1KEN_.html), together with any applicable supplemental end user license agreements, or SEULA's.

Pursuant to such terms, Cisco is entitled to confirm that your access point enablement is properly licensed.

If you do not agree with any of the above, do not proceed further and CLICK DECLINE below.

The AP adder licenses are installed and activated on the vWLC.



Licenses

Adder License

License Count: Add 0

License	Type	Time(expires)	RTU Count	Status
ap_count	Evaluation	12 weeks, 5 days	200	Inactive
ap_count (adder)	Permanent	No Expiry	200	Active, Not-In-Use

RTU Licensing Using CLI

Step 1 To install AP adder licenses using the CLI, enter the following command:

(Cisco Controller) > **license add ap-count <1-200>**

Step 2 Read the **End User License Agreement (EULA)**, type **Y**, and press **Enter** to accept:

```
Feature Name: ap-count

Right to Use

Enabling additional access points supported by this controller product may require the purchase of
supplemental or "adder" licenses. You may remove supplemental licenses from one controller and transfer
to another controller in the same product family. NOTE: licenses embedded in the controller at time
of shipment are not transferrable.
```

```
By clicking "I AGREE" (or "I ACCEPT") below, you warrant and represent that you have purchased
sufficient supplemental licenses for the access points to be enabled.

All supplemental licenses are subject to the terms and conditions of the Cisco end user license
agreement (http://www.cisco.com/en/US/docs/general/warranty/English/EU1KEN_.html), together with any
applicable supplemental end user license agreements, or SEULA's.

Pursuant to such terms, Cisco is entitled to confirm that your access point enablement is properly
licensed.

If you do not agree with any of the above, do not proceed further and CLICK "DECLINE" below.

ACCEPT? [y/n]: Y

Successfully added the license.
```

Step 3 The AP adder licenses are installed and activated on the vWLC. You can view the installed licenses by typing the **show license summary** command:

(Cisco Controller) > **show license summary**

```
Feature name: ap_count

License type: Evaluation

License Eula: Not Accepted

Evaluation total period: 12 weeks 6 days

License state: Inactive, Not-In-Use

RTU License Count: 200

Feature name: ap_count (adder)

License type: Permanent

License state: Active, Not-In-Use

RTU License Count: 200
```

Step 4 To activate or deactivate a feature license, enter the following command:

license {activate | deactivate} feature *license_name*