# Install Cisco UCS VIC Drivers for Linux

## Installing Linux Drivers to the Local Drive Using a Driver Disk

### Before You Begin

Ensure that you adhere to the following best practice for installing the Linux drivers during the OS installation of Linux. Upgrade the infrastructure in the following order before upgrading the drivers.

- Upgrade the infrastructure firmware which includes the UCS Manager, the Fabric Interconnects, and the chassis I/O Modules.
- Upgrade the server and adapter firmware.
- Upgrade the OS VIC drivers.

**Caution** Failure to adhere to the proper upgrade sequence can cause the server to crash.

### Procedure

**Step 1** Enable CDN from the BIOS settings.

**Step 2** Insert the RHEL installation DVD and at the installation menu, enter `Linux dd`.
A prompt displays requesting the driver disk.

**Note** While performing an RHEL installation of drivers with dd.iso, press **Tab** at the initial launch of the DVD, edit the line and enter `dd`.

**Step 3** Remove the RHEL installation DVD and insert the DVD generated from the dd.iso image. The RHEL installer reads the new drivers and overrides the default drivers.

**Step 4** Reinsert the RHEL DVD to start the installation.

**Note** To configure Consistent Device Name (CDN) on RHEL, enter the biosdevname=1 option to the bootloader during the installation.

**Step 5** To install RHEL 6.X, choose the package at the time of install. For further details, refer to the RHEL 6.X Deployment Guide.

Minimal install option does not include RHEL 6.X.

**Step 6** RHEL 7.X is installed by default. For further details, refer to the RHEL 7.X Deployment Guide.

**Step 7** Complete the RHEL installation.

**Note** During installation using RHEL 7.X, unmap the OS ISO image and map the driver ISO image. Run the `refresh` command, and then select the driver with a numerical option. Run the `continue` command, and then after extraction, remap the dvd.iso.

**Step 8** Verify that the default RHEL driver was replaced by the driver in the dd.iso image.

For the eNIC driver, cat /sys/module/enic/version.

For the fNIC driver, cat /sys/module/enic/version.

To read the CDN label assigned to a given VNIC, use the command biosdevname -d in the host terminal.

# Installing Linux to SAN Storage Using the fNIC Driver and OS Driver Disk

### Procedure

**Step 1** Create a vHBA on the Cisco UCS M81KR Virtual Interface Card.

**Step 2** In Cisco UCS Manager, mark the vHBA bootable and add the WWPN of the SAN storage.

For more information on how to do this step, see the *Cisco UCS Manager Configuration Guide*.

**Step 3** Boot the server using the RHEL installation DVD through vMedia.

**Step 4** At the installation menu, enter `linux dd`.

The installer displays the available installation disks, including the local disk and the SAN disk discovered by the Cisco UCS M81KR Virtual Interface Card.

**Step 5** For the installation target, choose the SAN storage device.

The RHEL installer reads the new drivers and overrides the default drivers to install RHEL on the SAN disk.

**Step 6** Complete the RHEL installation and reboot the host, choosing SAN storage as the first boot option.

# Installing Linux Drivers using RPM

If the management connection is over the eNIC, Cisco recommends using the serial or KVM console to complete the driver installation. Completing an rmmod of the current driver results in a loss of eNIC network connectivity.

### Before You Begin

Remove existing drivers by entering one of the following commands:

- **rmmod** {**enic** | **fnic**}
- **modprobe** {**-r enic** | **-r fnic**}

**Note** If you are booting from SAN storage, you cannot remove the existing driver using the **rmmod** fNIC command because this driver is required to use the SAN LUNs. Instead, enter the **rpm --erase** *old-kmod-fnic-rpm-name* command to erase the old RPM. Then, enter the **rpm -ivh --nodeps** *new-kmod-fnic-rpm-name* command to update the driver and reboot the node. The **rpm -ivh --nodeps** *new-kmod-fnic-rpm-name* command replaces the older driver with the new driver in the system memory.

If an fNIC RPM is not installed, and the fNIC driver that is available with the OS kernel is used, do not remove any existing drivers.

To determine the package that the fNIC belongs to, perform the following steps:

1 Find the installed fNIC modules

```
$ find /lib/modules -name *fnic*
/lib/modules/2.6.18-194.el5/kernel/drivers/scsi/fnic/fnic.ko
```

2 Determine which package the fNIC modules belong to.

```
cd /lib/modules/2.6.18-194.el5/kernel/drivers/scsi/fnic/fnic.ko
rpm -qf ./fnic.ko
```

If this command displays the fNIC package name, uninstall the fNIC RPM. You do not need to remove the fNIC RPM if it belongs to the Linux kernel package.

If drivers were previously installed using the driver disk installation process, the driver disk's `enic/fnic.ko` file is in the /lib/modules/'uname -r'/updates/ directory. The default search order of depmod (as specified in the /etc/depmod.d/depmod.conf.dist file) places a higher priority on the updates/ directory. Because new RPM installations place the enic/fnic.ko file under `/lib/modules/'uname -r'/extra/`, you can rename, delete or move the driver in the `/lib/modules/'uname -r'/updates/` directory. Alternatively, you can modify the `/etc/depmod.d/depmod.conf.dist` to change the search order by placing extra/ before updates/. To ensure that the depmod picks up the RPM installation's driver and not the existing driver installed using the driver disk method.

### Procedure

**Step 1** Install the binary RPM by entering the corresponding command for your driver:

| Driver | Command |
|---|---|
| eNIC driver on RHEL | **rpm -ivh kmod-enic-*version*-0.x86_86.rpm** |

| Driver | Command |
|---|---|
| eNIC driver on RHEL with XEN support | **rpm -ivh kmod-enic-***version***-0.x86_86.rpm** |
| eNIC driver on SuSE | **rpm -ivh cisco-enic-kmp-default-***version***-0.x86_64.rpm** |
| eNIC driver on SuSE with XEN support | **rpm -ivh cisco-enic-kmp-default-***version***-0.x86_64.rpm** |
| fNIC driver on RHEL | **rpm -ivh --nodeps kmod-fnic-** |

The driver is installed but not loaded.

**Step 2** Load the driver in one of the following ways:

- Reboot the host.
- Load the driver manually while the host is running by entering the **modprobe**{**enic** | **fnic**} command.

**Note** If an error message displays that the module is in use, remove any modules that are using the driver, then remove the driver. Issuing the **lsmod | grep** {**enic** | **fnic**} command can help determine which modules are in use.

If there are many references to the eNIC or fNIC module and it is not possible to remove all of the dependencies, reboot the system.

**Step 3** Verify the driver installation by entering the **sbin/lsmod | grep** {**enic** | **fnic**} command.

# Installing Linux Drivers Using the Source Tarball

To install the Linux driver for eNIC or fNIC for the first time, complete the entire procedure. To upgrade an existing driver, remove the currently running eNIC or fNIC module, then complete steps 3-10.

**Note** You can use the source tarball for any Linux distribution.

### Before You Begin

Remove existing drivers by entering one of the following commands:

- **/sbin/rmmod** {**enic** | **fnic**}
- **modprobe** {**-r enic** | **-r fnic**}

**Note** If you are booting from SAN storage, you cannot remove the existing driver using the **rmmod** fNIC command because this driver is required to use the SAN LUNs. Instead, enter the **rpm --erase** *old-kmod-fnic-rpm-name* command to erase the old RPM. Then, enter the **rpm -ivh --nodeps** *new-kmod-fnic-rpm-name* command to update the driver and reboot the node. Entering the **rpm -ivh --nodeps** *new-kmod-fnic-rpm-name* command replaces the older driver with the new driver in the system memory.

If drivers were previously installed using the driver disk installation process, the driver disk's `enic/fnic.ko` file is in the `/lib/modules/'uname -r'/updates/` directory. The default search order of depmod (as specified in the `/etc/depmod.d/depmod.conf.dist` file) places a higher priority on the updates/ directory. Because new RPM installations place the `enic/fnic.ko` file under `/lib/modules/'uname -r'/extra/`, you can rename, delete or move the driver in the /lib/modules/'uname -r'/updates/ directory. Alternatively, you can also modify the `/etc/depmod.d/depmod.conf.dist` to change the search order by placing extra/ before updates/. To ensure that the depmod picks up the RPM installation's driver and not the existing driver installed using the driver disk method.

### Procedure

**Step 1** Copy the source tarball to the specified folder.
**cp** {**enic-** | **fnic**}*version-num***.tar.gz** *folder-name*

**Step 2** Change directories to the specified folder.
**cd** *folder-name*

**Step 3** Extract the source tarball.
**tar xvfz** {**enic-** | **fnic**}*version-num***.tar.gz**

**Step 4** Change directories to the eNIC or fNIC *version* folder.
**cd** {**enic-** | **fnic**}*version-num*

**Step 5** Create the driver by entering one of the following commands:

| Driver | Command |
|---|---|
| eNIC | **make CONFIG_ENIC=m** |
| fNIC | **make CONFIG_FNIC=m** |

Creating the driver builds the new .ko file and removes the existing driver. The new driver is copied to /lib/modules/'uname -r'/kernel/drivers/scsi/fnic/.

**Step 6** Install the driver by entering one of the following commands:

| Driver | Command |
|---|---|
| eNIC | **make CONFIG_ENIC=m install** |
| fNIC | **make CONFIG_FNIC=m install** |

If an enic or fnic.ko file already exists in that directory, it is renamed as enic or fnic.ko.orig during the make installation. The make file backs up the currently installed enic or fnic.ko module and replaces it with the newly build module. For the fNIC, if libfc.ko, fcoe.ko, and libfcoe.ko exist on the system, they are left unmodified.

**Step 7** (Optional) If you installed and are booting from SAN storage, rebuild the initrd file with the updated fNIC drivers.

**Example:**

```
# cp /boot/initrd-'uname -r'.img /boot/initrd-'uname -r'.img.orig
# mkinitrd /boot/initrd-'uname -r'.img 'uname -r'
```

**Step 8** Load the driver in one of the following ways:

- Reboot the host
- Load the driver manually while the host is running by entering the **modprobe**{**enic** | **fnic**} command.
- Load the installed module manually by entering the **/sbin/insmod /lib/modules/***uname***-r/extra/**{**enic** | **fnic**}/{**enic** | **fnic**}**.ko** command.

**Note** If an error message displays that the module is in use, remove any modules that are using the driver, then remove the driver. Entering the **lsmod | grep** {**enic** | **fnic**} command can determine which modules are in use.

If there are many references to the eNIC or fNIC module and it is not possible to remove all of the dependencies, reboot the system.

**Step 9** Verify the new driver is loaded.
**/sbin/lsmod | grep** {**enic**| **fnic**}

**Step 10** (Optional) Enter the **fcc** command with any of the following arguments to run the FCC tool.

| Argument | Description |
|---|---|
| No arguments specified | Lists all FC hosts, remote ports, and LUNs. |
| **fcc --help** | Lists all of the FC commands.<br>**Note** Only the list, luns, stats, info, and reset commands work at this time. The other options are supported in the software FCoE stack only. |
| **fcc list host***host-num* | Lists only the hosts, ports, and LUNs associated with a single host. |
| **fcc reset host***host-num* | Resets the specified host. |

The FCC tool that is packaged with the driver and can be used to list all of the associated Fibre Channel HBAs, remote ports, and LUNs. Entering the **make install** command copies the FCC to the /root/bin/ directory and creates a link to this file in the /bin/ directory.

The following example extracts version 11 of the tarball to a folder called tmp and installs the driver. The driver is loaded using the reboot method.

```
$ cp enic-11.tar.gz /tmp
$ cd /tmp
$ tar xvfz enic-11.tar.gz
$ cd enic-11
$ make CONFIG_ENIC=m
# make CONFIG_ENIC=m install
```

### What to Do Next

After the reboot, you can run the following commands to ensure that the correct driver is loaded:

```
$ ([root@linux-host]# dmesg | grep -i fnic
$ fnic: Cisco FCoE HBA Driver, ver 1.5.0.1
$ scsi0 : fnic
$ scsi1 : fnic
```

# Displaying sNIC Status Using the sNIC Admin Utility

### Procedure

# **snic_admin**
Displays the status of the sNIC device.

This example shows how to display the status of the sNIC device:

```
# snic_admin
SNIC HBAs:
host2
SCSI States:
HBA       Device      Mode             State           Busy [ DrVer ]
host2     snic2       Initiator        running            0      [ 0.0.1.2 ]

host2 Targets
snic_sas_tgt:2:0-0                       SNIC Target

host2 LUNs:
Path       Device     Size   Vendor          Model                State
2:0:0:0    sdb       32 GB   LSI             MR9271-8i            running
2:0:0:1    sdc       32 GB   LSI             MR9271-8i            running
```

# Displaying sNIC Statistics Using the sNIC Admin Utility

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | # **snic_admin stats** | Displays the statistics of the sNIC device. |

This example shows how to display the statistics of the sNIC device:

```
# snic_admin stats
host0 Statistics:
------------------------------------------
  IO Statistics
------------------------------------------
Active IOs                  : 28
Max Active IOs              : 50
Total IOs                   : 37751401
IOs Completed               : 37751373
IOs Failed                  : 0
IOs Not Found               : 0
Memory Alloc Failures       : 0
```

```
REQs Null                   : 0
SCSI Cmd Pointers Null      : 0
Max SGL for any IO          : 60
Max IO Size                 : 1024 Sectors
Max Queuing Time            : 1
Max Completion Time         : 1
Max IO Process Time(FW)     : 363 (363 msec)

SGL Counters
    117396       4038       5428     623191     590080       1787       1542   30045078
   6356933        578       2157        253        158         99         67       1918
        86         99         46         26         31         27         36         30
        19         29         27         45         20         25         31         56
        13          1          1          1          1          1          3          4
         2          1          2          0          1          0          2          1
         0          0          2          3          2          2          0          1
         1          1          2         12
-------------------------------------------
  Abort Statistics
---------------------------------------------
Aborts                      : 0
Aborts Fail                 : 0
Aborts Driver Timeout       : 0
Abort FW Timeout            : 0
Abort IO NOT Found          : 0

-------------------------------------------
  Reset Statistics
---------------------------------------------
HBA Resets                  : 0
HBA Reset Cmpls             : 0
HBA Reset Fail              : 0

-------------------------------------------
  Firmware Statistics
---------------------------------------------
Active FW Requests             : 28
Max FW Requests                : 50
FW Out Of Resource Errs        : 0
FW IO Errors                   : 0
FW SCSI Errors                 : 0

---------------------------------------------
  Other Statistics

---------------------------------------------
Last ISR Time               : 4367682369 ( 4367018.481279912)
Last Ack Time               : 4367682355 ( 4367018.467282040)
ISRs                        : 64909272
Max CQ Entries              : 9
Data Count Mismatch         : 0
IOs w/ Timeout Status       : 0
IOs w/ Aborted Status       : 0
IOs w/ SGL Invalid Stat     : 0
WQ Desc Alloc Fail          : 0
Queue Full                  : 0
Queue Ramp Up               : 0
Queue Ramp Down             : 0
Queue Last Queue Depth      : 0
Target Not Ready            : 0
IOs fw processing (<= 10ms) : 1042938
IOs fw processing (>10 && <= 100ms) : 34946117
IOs fw processing (>100 && <= 500ms): 1762320
IOs fw processing (>500ms)  : 0

-------------------------------------------
  IO Compl CQ Info
---------------------------------------------
CQ ring base        : 0x413854000
CQ ring size        : 192
CQ head             : 0
CQ tail             : 143
CQ tail color       : 0
CQ to clean idx      : 143
CQ last color        : 1
```

# sNIC Drivers for Linux

## Installing Linux to DAS Storage Using the sNIC Driver Disk

### Procedure

**Step 1** In the **Navigation** pane of Cisco UCS Manager, click the **Servers** tab.

**Step 2** On the **Servers** tab, expand **Servers** > **Service Profiles**.

**Step 3** Create a service profile with two LUNs and associate it with a server.
Detailed information about creating a service profile and associating it with a server is available in *Cisco UCS Server Manager CLI Configuration Guide*.

**Step 4** For the service profile that you created, configure a local disk as the first boot device.
Detailed information about configuring a local disk as the first boot device is available in *Cisco UCS Server Manager CLI Configuration Guide.* .

**Step 5** Boot the server using the OS installation DVD through vMedia.

**Step 6** At the installation menu, enter `linux dd`.
A message appears that asks you whether you have a driver disk.

**Step 7** If you have a driver disk, select **Yes**, and map the driver disk dd-snic-*version* to the vMedia.

**Step 8** Select the relevant vMedia.
The OS installer reads the new drivers and overrides the default drivers to install the OS on the DAS disk.

**Step 9** Ensure that the DAS storage is discovered.

**Step 10** Complete the OS installation, and reboot the host.

## Installing sNIC Linux Drivers using RPM

You can use RPM to install sNIC drivers only on RHEL and CentOS.

**Important** Use this procedure only to upgrade driver versions

### Procedure

**Step 1** Install the binary RPM on RHEL and CentOS by using the **rpm -ivh** *snic-rpm-package* command for the sNIC driver.

The driver is installed, but not loaded.

If the OS is already installed by using DD, then using this command displays an error message. You can use the RPM package only for upgrading the driver version by using the **rpm -Uvh** *snic-rpm-package* command. After this is done, you cannot unload the sNIC driver.

**Step 2** List the module information for the sNIC driver by entering the **modinfo snic** command.
The driver version in the kernel may not be upgraded before rebooting the host.

**Step 3** Reboot the host.
After reboot, the host boots successfully with the latest driver.

**Step 4** Verify that the driver version is the same when you run the **modinfo snic** command and the **cat /sys/module/snic/version** command.

# Installing sNIC Linux Drivers Using the Source Tarball

### Procedure

**Step 1** Copy the source tarball to the specified folder.
**cp** {**snic**}*version-num***.tar.gz** *folder-name*

**Example:**

```
cp snic-0.0.19.tar.gz
```

**Step 2** Change directories to the specified folder.
**cd** *folder-name*

**Example:**

```
cd /tmp
```

**Step 3** Extract the source tarball.
**tar xvfz** {**snic**}*version-num***.tar.gz**

**Example:**

```
# tar xvf snic-0.0.19.tar
```

**Step 4** Change directories to the sNIC *version* folder.
**cd** {**snic**}*version-num*

**Example:**

```
# cd snic-0.0.19
```

**Step 5** Make the driver by entering the following command:
**make CONFIG_SCSI_SNIC=m**

Making the driver builds the new `.ko` file and removes the existing driver. The new driver is copied to `/lib/modules/'uname -r'/kernel/drivers/scsi/snic/`.

**Example:**

```
]# make CONFIG_SCSI_SNIC=m
make -C /lib/modules/2.6.32-431.el6.x86_64/build M=/root/snic-0.0.1.19/drivers/scsi modules
make[1]: Entering directory `/usr/src/kernels/2.6.32-431.el6.x86_64'
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_attrs.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_main.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_res.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_isr.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_ctl.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_io.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_scsi.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_disc.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_debugfs.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_trc.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic_dbg.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/vnic_cq.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/vnic_intr.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/vnic_dev.o
  CC [M]  /root/snic-0.0.1.19/drivers/scsi/snic/vnic_wq.o
  LD [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /root/snic-0.0.1.19/drivers/scsi/snic/snic.mod.o
  LD [M]  /root/snic-0.0.1.19/drivers/scsi/snic/snic.ko.unsigned
  NO SIGN [M] /root/snic-0.0.1.19/drivers/scsi/snic/snic.ko
make[1]: Leaving directory `/usr/src/kernels/2.6.32-431.el6.x86_64'
```

**Step 6** Install the driver by entering the following command:
**make CONFIG_SCSI_sNIC=m install**

If the snic.ko file already exists in that directory, it is renamed as snic.ko.orig during the make installation. The make file backs up the currently installed snic.ko module and replaces it with the newly built module.

**Example:**

```
]# make CONFIG_SCSI_SNIC=m install
install: backing up old versions of modules
#
# Just find all .ko files in /lib/modules/2.6.32-431.el6.x86_64/kernel/drivers/scsi/
directory,
# and backup the file if it isn't a soft link.
#
find /lib/modules/2.6.32-431.el6.x86_64/kernel/drivers/scsi/ \
  \( -name snic.ko -o -false \) | \
  xargs -t -r -I {} -i sh -c '[ -h {} ] || cp {} {}.orig'
install: completed backing up original OS .ko files
install: backing up last built .ko files
find /lib/modules/2.6.32-431.el6.x86_64/extra/ \
  \( -name snic.ko -o -false \) | \
  xargs -t -r -I {} -i sh -c ' mv {} {}.prev '
sh -c  mv /lib/modules/2.6.32-431.el6.x86_64/extra/snic/snic.ko
/lib/modules/2.6.32-431.el6.x86_64/extra/snic/snic.ko.prev
make -C /lib/modules/2.6.32-431.el6.x86_64/build M=/root/snic-0.0.1.19/drivers/scsi
modules_install
make[1]: Entering directory `/usr/src/kernels/2.6.32-431.el6.x86_64'
  INSTALL /root/snic-0.0.1.19/drivers/scsi/snic/snic.ko
  DEPMOD  2.6.32-431.el6.x86_64
make[1]: Leaving directory `/usr/src/kernels/2.6.32-431.el6.x86_64'
if [ -d /lib/modules/2.6.32-431.el6.x86_64/kernel/drivers/scsi//snic ]; then \
 find /lib/modules/2.6.32-431.el6.x86_64/extra/ \
  \( -name snic.ko -o -false \) | \
  xargs -t -r -I {} -i sh -c 'rm -f `echo {} | sed -e \
  "s!extra!kernel/drivers/scsi!g"`; ln -s {} `echo {} | \
  sed -e "s!extra!kernel/drivers/scsi!g"`' ; \
 fi
#
# Finally, copy the snic_admin script to /bin/
```

```
#
rm -f /bin/snic_admin
cp /root/snic-0.0.1.19/tools/bin/snic_admin /bin/
/sbin/depmod -a > /dev/null
```

**Step 7** (Optional) If you installed and are booting from DAS storage, rebuild the `initrd` file with the updated sNIC drivers.

**Example:**

```
# cp /boot/initrd-'uname -r'.img /boot/initrd-'uname -r'.img.orig
# mkinitrd /boot/initrd-'uname -r'.img 'uname -r'
```

**Step 8** Verify that the new driver is loaded.
**/sbin/lsmod | grep** {**snic**}

**Example:**

```
# lsmod | grep snic

snic                  108564  2
```

### What to Do Next

After the reboot, you can run the following command to ensure that the correct driver is loaded:

```
$ ([root@linux-host]# dmesg | grep -i snic
snic:Cisco SCSI NIC Driver, ver 0.0.1.19
snic:Trace Facility Enabled.
snic:snic device 1137:  46:1137: 12a:
snic:snic device bus 5: slot 0: fn 0
scsi host0: snic0 = ffff880414a9a5e0 shost = ffff880414a9a000 device bus 5: slot 0: fn 0
snic 0000:05:00.0: PCI INT B -> GSI 17 (level, low) -> IRQ 17
snic 0000:05:00.0: setting latency timer to 64
snic:vNIC resources wq 64
snic:vNIC mtu 2048 intr timer 0
snic:vNIC flags 0x0 luns per tgt 256
snic:vNIC io throttle count 64
snic:vNIC port down timeout 0 port down io retries 30
snic:vNIC back end type = 1
snic:vNIC hid = 4
snic 0000:05:00.0: irq 33 for MSI/MSI-X
snic 0000:05:00.0: irq 34 for MSI/MSI-X
snic 0000:05:00.0: irq 35 for MSI/MSI-X
snic:vNIC interrupt mode: MSI-X
snic:wq 1 cq 2 intr 3
scsi0 : snic
scsi host0: snic state change from SNIC_INIT to SNIC_ONLINE
scsi host0: Retrieving snic params.
scsi host0: SNIC Device Probe Successful.
scsi host0: Scanning snic_das_tgt:0:0-5.
scsi host0: Scanning snic_das_tgt:0:0-4.
```

# Upgrading Ubuntu with sNIC Driver Disk Image

Upgrading Ubuntu drivers includes upgrading three packages in the following order:

1. snic-*image*-generic_*version*_-0ubuntu1_amd64.deb
2. snic_*version*_-0ubuntu1_amd64.deb
3. snic-common_*version*-0ubuntu1_amd64.deb

### Procedure

**Step 1** Upgrade the three packages.

**Example:**

```
# sudo dpkg -i snic-3.13.0-32-generic_0.0.1.14-0ubuntu1_amd64.deb
(Reading database ... 55192 files and directories currently installed.)
Preparing to unpack snic-3.13.0-32-generic_0.0.1.14-0ubuntu1_amd64.deb ...
Unpacking snic-3.13.0-32-generic (0.0.1.14-0ubuntu1) over (0.0.1.12-0ubuntu1) ...
Setting up snic-3.13.0-32-generic (0.0.1.14-0ubuntu1) ...

# sudo dpkg -i snic_0.0.1.14-0ubuntu1_amd64.deb
(Reading database ... 55192 files and directories currently installed.)
Preparing to unpack snic_0.0.1.14-0ubuntu1_amd64.deb ...
Unpacking snic (0.0.1.14-0ubuntu1) over (0.0.1.14-0ubuntu1) ...
Setting up snic (0.0.1.14-0ubuntu1) ...
Building module database ...
filename:       /lib/modules/3.13.0-32-generic/extra/snic/snic.ko
author:         abc <abc@email.com>
version:        0.0.1.14
description:    Cisco SCSI NIC Driver
license:        GPL v2
srcversion:     FE26EB9752C9F8C25FBCD95
alias:          pci:v00001137d00000046sv*sd*bc*sc*i*
depends:
vermagic:       3.13.0-32-generic SMP mod_unload modversions
parm:           snic_log_level:bitmask for snic logging levels (int)
parm:           snic_trace_max_pages:Total allocated memory pages for snic trace buffer
(uint)
parm:           snic_max_qdepth:Queue depth to report for each LUN (uint)
Updating initramfs ...
update-initramfs: Generating /boot/initrd.img-3.13.0-32-generic

# sudo dpkg -i snic-common_0.0.1.14-0ubuntu1_amd64.deb
(Reading database ... 55192 files and directories currently installed.)
Preparing to unpack snic-common_0.0.1.14-0ubuntu1_amd64.deb ...
Unpacking snic-common (0.0.1.14-0ubuntu1) over (0.0.1.12-0ubuntu1) ...
Setting up snic-common (0.0.1.14-0ubuntu1) ...
```

**Step 2** Verify that the upgrade has completed successfully.

**Example:**

```
$ sudo dpkg -s snic
Package: snic
Status: install ok installed
Priority: standard
Section: kernel
Installed-Size: 26
Maintainer: abc <abc@email.com>
Architecture: amd64
Version: 0.0.1.14-0ubuntu1
Provides: snic
Depends: snic-3.13.0-32-generic (= 0.0.1.14-0ubuntu1)
Description: Meta-package for installing the latest snic drivers.
 This is meta-package for Cisco SNIC driver (meta).
```

# Installing Ubuntu with sNIC Driver Disk Image

### Procedure

**Step 1** Download the disk image from the build server.

**Step 2** Map the driver disk image and OS ISO images to vMedia.

**Note** The driver disk image must be mapped as a removable disk under vMedia.

**Step 3** Boot from the **BIOS boot menu**, by using the mapped DVD.
The Ubuntu OS detects the virtual driver disk.

**Step 4** Select **Yes** to load drivers from the internal virtual driver disk.