

# Troubleshoot Nexus SmartNIC Hardware and Software Driver Installation

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Applicable Hardware](#)

[Troubleshoot Nexus SmartNIC Hardware Installation](#)

[Step 1. Verify Proper Hardware Installation.](#)

[Step 2. Verify PCI Express Slot Compatibility.](#)

[Step 3. Verify Installation in Active PCI Express Slot.](#)

[Step 4. Verify Nexus SmartNIC Receives Power.](#)

[Step 5. Verify Firmware Integrity.](#)

[Step 6. Verify Host Operating System PCI Express Detection.](#)

[Troubleshoot Nexus SmartNIC Software Driver Installation](#)

[Step 1. Verify Driver Usage.](#)

[Step 2. Verify Driver Installation.](#)

[Verify Driver Installation From RPM \(apt, yum, etc.\)](#)

[Verify Driver Installation From Source](#)

[Step 3. Attempt To Load Software Driver.](#)

["modprobe: FATAL: Module exanic not found" Error Message](#)

["modprobe: Required key not available" Error Message](#)

[Step 4. Confirm Nexus SmartNIC Utility Functionality.](#)

[Related Information](#)

## Introduction

This document describes the steps used to troubleshoot hardware and software driver installation for Nexus SmartNIC (formerly Exablaze ExaNIC) low latency network interface cards.

## Prerequisites

## Requirements

Cisco recommends that you understand the hardware installation procedure for the Nexus SmartNIC series of low latency network interface cards. Cisco also recommends you possess a basic understanding of the Linux command line interface.

## Components Used

The information in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Background Information

### Applicable Hardware

The procedures covered in this document are applicable to this hardware:

- Nexus SmartNIC X10
- Nexus SmartNIC X10-HPT
- Nexus SmartNIC X10-GM
- Nexus SmartNIC X25
- Nexus SmartNIC X40
- Nexus SmartNIC X100
- Nexus SmartNIC V5P
- Nexus SmartNIC V9P

## Troubleshoot Nexus SmartNIC Hardware Installation

This section of the document covers steps used to troubleshoot Nexus SmartNIC low latency network interface card hardware installation issues. Follow this section of the document when the host operating system (typically a Linux distribution or Windows Server) does not recognize a Nexus SmartNIC as a valid PCI Express peripheral device.

### Step 1. Verify Proper Hardware Installation.

Nexus SmartNIC network interface cards must be correctly installed in a PCI Express (PCIe) slot on the motherboard or riser/daughtercard of the host. For additional information about how to install a PCIe card, consult the manual for your host provided by the manufacturer.

For additional information about PCIe card installation on Cisco UCS servers, refer to this supporting documentation:

- [Cisco UCS C220 M5 Server Installation and Service Guide](#)
- [Cisco UCS C240 M5 Server Installation and Service Guide](#)

### Step 2. Verify PCI Express Slot Compatibility.

All Nexus SmartNIC network interface cards must be installed in a PCIe 3.0-compatible slot. Furthermore, a minimum of a PCIe x8 slot with 49 pins must be used to connect all Nexus SmartNIC network interface cards. For additional information about whether PCIe slots meet these specifications, consult the manual for your host provided by the manufacturer.

For additional information about PCIe slot specification on Cisco UCS servers, refer to this table and supporting documentation:

## Cisco UCS Server Model

## PCIe Slot Compatibility Supporting Documentation

Cisco UCS C220 M5	All PCIe slots compatible	<a href="#">Cisco UCS C220 M5 Server Installation and Service Guide</a>
Cisco UCS C240 M5	All PCIe slots compatible	<a href="#">Cisco UCS C240 M5 Server Installation and Service Guide</a>

### Step 3. Verify Installation in Active PCI Express Slot.

A Nexus SmartNIC must be properly inserted in a PCI Express slot connected to an active, installed CPU (Central Processing Unit). If a host is equipped with multiple CPU sockets where only a single socket has a CPU installed (also called a "single-CPU configuration"), not all PCIe slots may be active and functional. For additional information about which PCIe slots are activated in a single-CPU configuration, consult the manual for your host provided by the manufacturer.

For additional information about active PCIe slots on Cisco UCS servers in a single-CPU configuration, refer to this table and supporting documentation:

Cisco UCS Server Model	Active PCIe Slots	Supporting Documentation
Cisco UCS C220 M5	PCIe Riser 1, Slot 1	<a href="#">Cisco UCS C220 M5 Server Installation and Service Guide</a>
Cisco UCS C240 M5	PCIe Riser 1, Slot 1 PCIe Riser 1, Slot 2 PCIe Riser 1B, Slot 1 PCIe Riser 1B, Slot 2 PCIe Riser 1B, Slot 3	<a href="#">Cisco UCS C240 M5 Server Installation and Service Guide</a>

### Step 4. Verify Nexus SmartNIC Receives Power.

Each Nexus SmartNIC network interface card has a number of LEDs (Light Emitting Diodes) located on the PCIe bracket. These LEDs are typically visible from the exterior of the host. Each SFP+ and QSFP port on the network interface card has one larger LED associated with it. These LEDs are collectively referred to as **port LEDs**.

On network interface cards with only SFP+ ports, an additional smaller red LED is present that is associated with the PPS connector (with the exception of the Nexus SmartNIC X10-GM, where the small red LED is associated with the Grand Master clock state). This LED is referred to as the **PPS LED**.

**Note:** Nexus SmartNIC network interface cards equipped with QSFP ports do not have a PPS LED.

In summary, refer to the table here:

Nexus SmartNIC Model	Port LEDs	PPS LED
X10	Yes	Yes
X10-HPT	Yes	Yes
X10-GM	Yes	Yes (GPS)
X25	Yes	Yes
X40	Yes	No

X100	Yes	No
V5P	Yes	No
V9P	Yes	No

When a Nexus SmartNIC network interface card is first powered on alongside the host, all LEDs on the network interface card should flash momentarily. If no LEDs emit light at any time after the host is powered on, this indicates that the Nexus SmartNIC network interface card is not correctly receiving power from the PCIe bus. Troubleshoot this issue further with this procedure:

1. Verify whether the PCIe slot in use works with other peripheral devices that are known to be functional. Ideally, one would test with another network interface card.
2. Verify whether the Nexus SmartNIC network interface card operates without issue in another PCIe slot that is known to be functional.
3. Verify whether the Nexus SmartNIC network interface card operates without issue in a PCIe slot that is known to be functional on another host that is known to be functional.

If the Nexus SmartNIC network interface card fails to receive power regardless of the PCIe slot and host used, contact [Cisco TAC](#) for additional troubleshooting.

## Step 5. Verify Firmware Integrity.

As previously mentioned in Step 4, each Nexus SmartNIC network interface card potentially has two types of LEDs:

- Port LEDs
- PPS LED

When no SFP+/QSFP modules are inserted in the ports of the network interface card, these LEDs should remain off after the momentary flash described in Step 4 when the network interface card initially powers up.

There are three common scenarios where this is not the case as a result of corrupt or missing firmware:

- If both port and PPS LEDs remain illuminated after the momentary flash when no SFP+/QSFP modules are inserted and the host operating system does not recognize the network interface card as a valid PCIe peripheral device (such as through the `lspci` command), the firmware of the Nexus SmartNIC network interface card may need to be recovered. Follow the [Nexus SmartNIC Firmware Recovery Process](#) and load a new version of firmware onto the Nexus SmartNIC. If this does not resolve the issue, contact [Cisco TAC](#) for additional troubleshooting.
- If a Nexus SmartNIC network interface card is equipped with a PPS LED and that PPS LED is continuously lit, then the existing firmware loaded on the card is corrupt and the card has entered firmware recovery mode. You must follow the [Nexus SmartNIC Firmware Update Process](#) to use the network interface card normally. If this does not resolve the issue, contact [Cisco TAC](#) for additional troubleshooting.
- If a Nexus SmartNIC network interface card is not equipped with a PPS LED and the port LEDs are continuously alternating between off and amber, then the existing firmware loaded on the card is corrupt and the card has entered firmware recovery mode. You must follow the [Nexus SmartNIC Firmware Update Process](#) to use the network interface card normally. If this does not resolve the issue, contact [Cisco TAC](#) for additional troubleshooting.

## Step 6. Verify Host Operating System PCI Express Detection.

You can confirm that the host operating system can successfully detect a Nexus SmartNIC network interface card through the PCIe bus with the **lspci** command. Exablaze's 16-bit PCI Vendor ID (VID) is **0x1ce4**, which can be used to probe Nexus SmartNIC PCIe devices for information. This is demonstrated in the example here:

```
[root@host ~]# lspci -d 1ce4:
01:00.0 Ethernet controller: Exablaze ExaNIC X10
```

More detailed information about the PCIe card can be viewed by adding the **-v** verbose flag to the **lspci** command. This is demonstrated in the example here:

```
[root@host ~]# lspci -d 1ce4: -v
01:00.0 Ethernet controller: Exablaze ExaNIC X10
Subsystem: Exablaze ExaNIC X10
Flags: bus master, fast devsel, latency 0, IRQ 30
Memory at 92000000 (32-bit, non-prefetchable) [size=8M]
Memory at 92800000 (64-bit, non-prefetchable) [size=4M]
Capabilities: [80] Power Management version 3
Capabilities: [90] MSI: Enable+ Count=1/1 Maskable- 64bit+
Capabilities: [c0] Express Endpoint, MSI 00
Capabilities: [100] Advanced Error Reporting
Capabilities: [1b8] Latency Tolerance Reporting
Capabilities: [300] #19
Capabilities: [340] Vendor Specific Information: ID=0001 Rev=0 Len=02c <?>
Kernel modules: exanic
```

Even more detailed information can be viewed by adding the **-vv** verbose flag to the **lspci** command. This is demonstrated in the example here:

```
[root@host ~]# lspci -d 1ce4: -vv
01:00.0 Ethernet controller: Exablaze ExaNIC X10
Subsystem: Exablaze ExaNIC X10
Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B-
DisINTx+
Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR-
<PERR- INTx-
Latency: 0
Interrupt: pin A routed to IRQ 30
Region 0: Memory at 92000000 (32-bit, non-prefetchable) [size=8M]
Region 2: Memory at 92800000 (64-bit, non-prefetchable) [size=4M]
Capabilities: [80] Power Management version 3
Flags: PMEclk- DSI- D1- D2- AuxCurrent=0mA PME(D0-,D1-,D2-,D3hot-,D3cold-)
Status: D0 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
Capabilities: [90] MSI: Enable+ Count=1/1 Maskable- 64bit+
Address: 00000000fee003b8 Data: 0000
Capabilities: [c0] Express (v2) Endpoint, MSI 00
DevCap: MaxPayload 128 bytes, PhantFunc 0, Latency L0s <64ns, L1 <1us
ExtTag- AttnBtn- AttnInd- PwrInd- RBE+ FLReset- SlotPowerLimit 75.000W
DevCtl: Report errors: Correctable- Non-Fatal- Fatal- Unsupported-
RlxdOrd+ ExtTag- PhantFunc- AuxPwr- NoSnoop+
MaxPayload 128 bytes, MaxReadReq 512 bytes
DevSta: CorrErr- UncorrErr- FatalErr- UnsuppReq- AuxPwr- TransPend-
LnkCap: Port #0, Speed 8GT/s, Width x8, ASPM not supported, Exit Latency L0s
unlimited, L1 unlimited
ClockPM- Surprise- LLActRep- BwNot- ASPMOptComp+
LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- CommClk+
ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
```

```

LnkSta:      Speed 8GT/s, Width x8, TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
DevCap2: Completion Timeout: Range B, TimeoutDis+, LTR+, OBFF Not Supported
DevCtl2: Completion Timeout: 50us to 50ms, TimeoutDis-, LTR-, OBFF Disabled
LnkCtl2: Target Link Speed: 8GT/s, EnterCompliance- SpeedDis-
           Transmit Margin: Normal Operating Range, EnterModifiedCompliance- ComplianceSOS-
           Compliance De-emphasis: -6dB
LnkSta2: Current De-emphasis Level: -3.5dB, EqualizationComplete+, EqualizationPhase1+
           EqualizationPhase2-, EqualizationPhase3-, LinkEqualizationRequest-
Capabilities: [100 v2] Advanced Error Reporting
UESta:      DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC-
UnsupReq- ACSViol-
UEMsk:      DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC-
UnsupReq- ACSViol-
UESvrt:     DLP+ SDES+ TLP- FCP+ CmpltTO- CmpltAbrt- UnxCmplt- RxOF+ MalfTLP+ ECRC-
UnsupReq- ACSViol-
CESta:      RxErr- BadTLP- BadDLLP- Rollover- Timeout- NonFatalErr-
CEMsk:      RxErr- BadTLP- BadDLLP- Rollover- Timeout- NonFatalErr+
AERCap:     First Error Pointer: 00, GenCap- CGenEn- ChkCap- ChkEn-
Capabilities: [1b8 v1] Latency Tolerance Reporting
           Max snoop latency: 0ns
           Max no snoop latency: 0ns
Capabilities: [300 v1] #19
Capabilities: [340 v1] Vendor Specific Information: ID=0001 Rev=0 Len=02c <?>
Kernel modules: exanic

```

If the **lspci** command displays information about the Nexus SmartNIC network interface card, then this indicates that the host operating system has successfully detected the Nexus SmartNIC network interface card through the PCIe bus. Moving forward, one can install the Nexus SmartNIC software drivers and begin using the card.

## Troubleshoot Nexus SmartNIC Software Driver Installation

This section of the document covers steps used to troubleshoot Nexus SmartNIC low latency network interface card software driver installation issues. Follow this section of the document when the host operating system (typically a Linux distribution or Windows Server) recognizes a Nexus SmartNIC as a valid PCI Express peripheral device, but the host operating system does not recognize the Nexus SmartNIC's ports as a valid network interface. An example of this is shown in the output here:

```

[root@host ~]# ls /dev/exanic*
ls: cannot access /dev/exanic*: No such file or directory

```

This section of the document assumes that one has encountered an error while attempting to install the Nexus SmartNIC software drivers as described in the [Nexus SmartNIC Software Installation guide](#).

All commands in this procedure are executed from the root Linux account. If you are not utilizing the root Linux account to follow this procedure, you may need to use the **sudo** command to elevate your account's security privileges to that of a superuser.

### Step 1. Verify Driver Usage.

If the host operating system's kernel has loaded a software driver for a Nexus SmartNIC network interface card, the **lspci** command with the **-v** verbose flag will display the driver in use. One can

query for devices using Exablaze's PCI Vendor ID (0x1ce4) to display information specific to Nexus SmartNIC PCIe devices. An example of this is shown in the output here:

```
[root@host ~]# lspci -d 1ce4: -v | egrep Kernel.driver
Kernel driver in use: exanic
```

If the host operating system's kernel has not loaded the software driver, then this "Kernel driver in use" line will be omitted from the output of the `lspci -d 1ce4: -v` command.

## Step 2. Verify Driver Installation.

### Verify Driver Installation From RPM (apt, yum, etc.)

As described in the [Nexus SmartNIC Software Installation guide](#), the Nexus SmartNIC software drivers can be installed through a package manager (such as apt, yum, or directly through rpm). If you have installed ExaNIC software drivers with this method, you can verify that all files have been correctly installed as shown below.

Depending upon the CPU architecture of your host, the **libexanic.a** library file may be in **/usr/lib/** or in **/usr/lib64/**. Here is an example from an x86 (32-bit) CPU architecture:

```
[root@host ~]# ls /usr/lib/ | grep exanic
libexanic.a
```

Here is an example from an x86\_64 (64-bit) CPU architecture:

```
[root@host ~]# ls /usr/lib64/ | grep exanic
libexanic.a
```

Ensure that Nexus SmartNIC library header files are present in the **/usr/include/exanic/** directory. This is shown in the example here:

```
[root@host ~]# ls /usr/include/exanic/ -l
config.h
const.h
exanic.h
fifo_if.h
fifo_rx.h
fifo_tx.h
filter.h
firewall.h
hw_info.h
ioctl.h
pcie_if.h
port.h
register.h
time.h
util.h
```

Ensure that Nexus SmartNIC binary utilities are located in the **/usr/bin/** directory. This is shown in the example here:

```
[root@host ~]# ls /usr/bin/ -1 | grep exanic-  
exanic-capture  
exanic-clock-check  
exanic-clock-sync  
exanic-config  
exanic-fwupdate
```

Finally, ensure that the **exanic.ko.xz** module file is present in the **/lib/modules/`uname -r`/extra/** directory. Note that the inline **`uname -r`** command automatically inserts your current kernel release into the directory. This command is surrounded by grave accents (```), not single quotation marks. This is shown in the example here:

```
[root@host ~]# ls /lib/modules/`uname -r`/extra/ | grep exanic  
exanic.ko.xz
```

## Verify Driver Installation From Source

As described in the [Nexus SmartNIC Software Installation guide](#), the Nexus SmartNIC software drivers can be built and installed from the source code. If you have installed Nexus SmartNIC software drivers with this method, you can verify that all files have been correctly installed as shown in the examples here.

Ensure that the **libexanic.a** library file is present in the **/usr/local/lib/** directory. This is shown in the example here:

```
[root@host ~]# ls /usr/local/lib/ | grep exanic  
libexanic.a
```

Ensure that Nexus SmartNIC library header files are present in the **/usr/local/include/exanic/** directory. This is shown in the example here:

```
[root@host ~]# ls /usr/local/include/exanic/ -1  
config.h  
const.h  
exanic.h  
fifo_if.h  
fifo_rx.h  
fifo_tx.h  
filter.h  
firewall.h  
hw_info.h  
ioctl.h  
pcie_if.h  
port.h  
register.h  
time.h  
util.h
```

Ensure that Nexus SmartNIC binary utilities are located in the **/usr/local/bin/** directory. This is shown in the example here:

```
[root@host ~]# ls /usr/local/bin -1 | grep exanic-
```



```
exanic-capture
exanic-clock-check
exanic-clock-sync
exanic-config
exanic-fwupdate
```

Finally, ensure that the **exanic.ko** module file is present in the **/lib/modules/`uname -r`/extra/** directory. Note that the inline **`uname -r`** command automatically inserts your current kernel release into the directory. This command is surrounded by grave accents (**`**), not single quotation marks. This is shown in the example here:

```
[root@host ~]# ls /lib/modules/`uname -r`/extra | grep exanic
exanic.ko
```

### Step 3. Attempt To Load Software Driver.

The Nexus SmartNIC software driver can be manually loaded with the **modprobe exanic** command.

If Nexus SmartNIC drivers are loaded, then the Linux kernel will recognize it as a device. You can verify this with the **ls /dev/exanic\*** command, which will show all recognized Nexus SmartNIC devices. This is shown in the example here:

```
[root@host ~]# ls /dev/exanic*
/dev/exanic0
```

If the Nexus SmartNIC drivers are not successfully loaded, the **modprobe exanic** command may or may not return an error. The subsections below describes how to troubleshoot errors returned by this command.

#### "modprobe: FATAL: Module exanic not found" Error Message

This error message can potentially be caused by two different issues described in the below subsections.

##### Incorrectly Mapped Kernel Module Dependencies

The host operating system is unable to locate the module built for the current running kernel. As a result, the host operating system is unable to load the module into the system with the **modprobe exanic** command. This can be resolved with the **depmod -a** command, which will create a map of kernel module dependencies. The kernel module can then be loaded with the **modprobe exanic** command.

##### Kernel Module Not Built By DKMS

DKMS (Dynamic Kernel Module Support) allows for kernel modules to be automatically rebuilt whenever a new operating system kernel is installed. If DKMS has not built the Nexus SmartNIC kernel module, then it is possible that the Nexus SmartNIC kernel module was previously compiled for a different operating system kernel version.

In order for DKMS to work as expected, the kernel-devel and kernel-headers packages need to be installed using the host operating system's package manager. An example of how to confirm whether this package is installed with the **yum list** command is shown in the output here:

```
[root@host ~]$ yum list kernel-devel
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.internode.on.net
 * epel: ucmirror.canterbury.ac.nz
 * extras: mirror.internode.on.net
 * updates: centos.mirror.serversaustralia.com.au
Installed Packages
kernel-devel.x86_64                               3.10.0-1062.el7
@base
kernel-devel.x86_64                               3.10.0-1062.12.1.el7
@updates
```

```
[root@host ~]$ yum list kernel-headers
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.internode.on.net
 * epel: ucmirror.canterbury.ac.nz
 * extras: mirror.internode.on.net
 * updates: centos.mirror.serversaustralia.com.au
Installed Packages
kernel-headers.x86_64                             3.10.0-1062.12.1.el7
@updates
```

Refer to the manual of your host operating system's package manager to verify how to determine if a specific package is installed.

You can confirm whether the Nexus SmartNIC kernel module is successfully added, built, and installed by DKMS with the **dkms status** command. An example of the output of this command in a working, expected state is shown here:

```
[root@host ~]# dkms status
exanic, 2.4.1-1.el8, 3.10.0-957.27.2.el7.x86_64, x86_64: installed
```

If this output shows that the Nexus SmartNIC kernel module is in a state other than "installed", one will need to either build or install the Nexus SmartNIC kernel module using DKMS.

If the Nexus SmartNIC kernel module is added but not built, use the **dkms build -m exanic -v {version}** command to build the Nexus SmartNIC kernel module. An example of this command is shown here with the 2.4.1-1.el7 software version:

```
[root@host ~]$ dkms build -m exanic -v 2.4.1-1.el7

Kernel preparation unnecessary for this kernel.  Skipping...

Building module:
cleaning build area...
make -j16 KERNELRELEASE=3.10.0-1062.el7.x86_64 -C modules KDIR=/lib/modules/3.10.0-1062.el7.x86_64/build...
cleaning build area...

DKMS: build completed.
```

If the Nexus SmartNIC kernel module is built but not installed, use the **dkms install -m exanic -v**

**{version}** command to install the Nexus SmartNIC kernel module. An example of this command is shown here with the 2.4.1-1.el7 software version:

```
[root@host ~]$ dkms install -m exanic -v 2.4.1-1.el7

exanic.ko.xz:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
- Installation
  - Installing to /lib/modules/3.10.0-1062.el7.x86_64/extra/

exasock.ko.xz:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
- Installation
  - Installing to /lib/modules/3.10.0-1062.el7.x86_64/extra/
Adding any weak-modules

depmod...

DKMS: install completed.
```

After the Nexus SmartNIC kernel module is built and installed with DKMS, the Nexus SmartNIC kernel module can then be loaded with the **modprobe exanic** command.

### "modprobe: Required key not available" Error Message

This error message can be observed on hosts equipped with UEFI (Unified Extensible Firmware Interface) firmware that have the Secure Boot protocol enabled. Secure Boot prevents kernel drivers that are not signed with an acceptable digital signature from being loaded. As a result, when one attempts to load the Nexus SmartNIC kernel driver with the **modprobe exanic** command, the Secure Boot protocol prevents the kernel driver from being loaded.

To resolve this issue, the Secure Boot protocol needs to be disabled within the host's UEFI. For additional information about how to disable the Secure Boot protocol within the UEFI of your host, consult the manual for your host provided by the manufacturer.

### Step 4. Confirm Nexus SmartNIC Utility Functionality.

The **exanic-config** command with no arguments can be used to display basic information about Nexus SmartNIC network interface cards installed on the host after the Nexus SmartNIC kernel drivers are loaded into the operating system. An example of this output is here:

```
[root@host ~]$ exanic-config

Device exanic1:
Hardware type: ExaNIC X10
Temperature: 38.8 C VCCint: 0.95 V VCCaux: 1.83 V
Function: network interface
Firmware date: 20180409 (Mon Apr 9 23:27:40 2018)
PPS port: input, termination disabled
Port 0:
Interface: enp175s0
```

Port speed: 10000 Mbps  
Port status: enabled, SFP present, signal detected, link active  
MAC address: 64:3f:5f:xx:xx:xx  
RX packets: 11778 ignored: 0 error: 0 dropped: 0  
TX packets: 11836  
Port 1:  
Interface: enp175s0d1  
Port speed: 10000 Mbps  
Port status: enabled, SFP present, signal detected, link active  
MAC address: 64:3f:5f:xx:xx:xx  
RX packets: 11836 ignored: 0 error: 0 dropped: 0  
TX packets: 11778

If the **exanic-config** command/utility returns relevant information about installed Nexus SmartNIC network interface cards, then the Nexus SmartNIC hardware and software driver installation was successful.

## Related Information

- [Nexus SmartNIC Device Installation](#)
- [Cisco UCS C240 M5 Server Installation and Service Guide](#)
- [Cisco UCS C220 M5 Server Installation and Service Guide](#)
- [Nexus SmartNIC Benchmarking Guide](#)