Configuring NVMe Over Fabrics (NVMeoF) with RoCEv2 in Linux

Guidelines for using NVMe over Fabrics (NVMeoF) with RoCEv2 on Linux

General Guidelines and Limitations:

- Cisco recommends you check UCS Hardware and Software Compatibility specific to your UCS Manager release to determine support for NVMeoF. NVMeoF is supported on UCS M5 and later B-Series and C-Series servers.

- NVMe over RDMA with RoCEv2 is supported with the fourth generation Cisco UCS VIC 1400 Series UCS VIC 14000, and UCS VIC 15000 Series and UCS VIC 15000 Series adapters. NVMe over RDMA is not supported on UCS 6324 Fabric Interconnects or on UCS VIC 1200 Series and 1300 Series adapters.

- When creating RoCEv2 interfaces, use Cisco UCS Manager provided Linux-NVMe-RoCE adapter policy.

Note Do not use the default Linux Adapter policy with RoCEv2; RoCEv2 interfaces will not be created in the OS.

- When configuring RoCEv2 interfaces, use both the enic and enic_rdma binary drivers downloaded from Cisco.com and install the matched set of enic and enic_rdma drivers. Attempting to use the binary enic_rdma driver downloaded from Cisco.com with an inbox enic driver will not work.

- RoCEv2 supports maximum two RoCEv2 enabled interfaces per adapter.

- Booting from an NVMeoF namespace is not supported.

- Layer 3 routing is not supported.
- RoCEv2 does not support bonding.
- Saving a crashdump to an NVMeoF namespace during a system crash is not supported.
- NVMeoF cannot be used with usNIC, VMFEX, VxLAN, VMQ, VMMQ, NVGRE, GENEVE Offload, and DPDK features.
- Netflow monitoring is not supported on RoCEv2 interfaces.
- In the Linux-NVMe-RoCE policy, do not change values of Queue Pairs, Memory Regions, Resource Groups, and Priority settings other than to Cisco provided default values. NVMeoF functionality may not be guaranteed with different settings for Queue Pairs, Memory Regions, Resource Groups, and Priority.
- The QoS no drop class configuration must be properly configured on upstream switches such as Cisco Nexus 9000 series switches. QoS configurations will vary between different upstream switches.
- Set MTU size correctly on the VLANs and QoS policy on upstream switches.
- Spanning Tree Protocol (STP) may cause temporary loss of network connectivity when a failover or failback event occurs. To prevent this issue from occurring, disable STP on uplink switches.
- UCS Manager does not support fabric failover for vNICs with RoCEv2 enabled.

Interrupts

- Linux RoCEv2 interface supports only MSIx interrupt mode. Cisco recommends avoiding changing interrupt mode when the interface is configured with RoCEv2 properties.
- The minimum interrupt count for using RoCEv2 with Linux is 8.

Downgrade Limitations:

- Cisco recommends you remove the RoCEv2 configuration before downgrading to any non-supported RoCEv2 release.

Linux Requirements

Configuration and use of RoCEv2 in Linux requires the following:

- InfiniBand kernel API module ib_core
- UCS Manager release 4.1.1 or later
- Minimum VIC firmware 5.1(1x) for IPv4 support and 5.1(2x) for IPv6 support
- UCS M5 and later B or C-series servers with Cisco UCS VIC 1400 or 15000 Series adapters
- eNIC driver version 4.0.0.6-802-21 or later provided with the 4.1.1 release package
- enic_rdma driver version 1.0.0.6-802-21 or later provided with the 4.1.1 release package

Note Use eNIC driver version 4.0.0.10-802.34 or later and enic_rdma driver version 1.0.0.10-802.34 or later for IPv6 support.

- A storage array that supports NVMeoF connection

Configuring RoCEv2 for NVMeoF on UCS Manager

Use these steps to configure the RoCEv2 interface on UCS Manager.

Step 1 In the **Navigation** pane, click **Servers**.

Step 2 Expand **Servers** > **Service Profiles**.

Step 3 Expand the node for the organization where you want to create the policy.

If the system does not include multitenancy, expand the **root** node.

Step 4 Click on **vNICs** and go to the **Network** tab in the work area.

Modify the vNIC policy, according to the steps below.

a) On the **Network** tab, scroll down to the desired vNIC and click on it, then click **Modify**.
b) A popup dialog box will appear. Scroll down to the **Adapter Performance Profile** area, and click on the dropdown area for the Adapter Policy. Choose **Linux-NVMe-RoCE** from the drop-down list.
c) Click **OK**.

Step 5 Click **Save Changes**.

Step 6 Select **Reboot**.

Enabling an SRIOV BIOS Policy

Use these steps to configure the server's service profile with the RoCE v2 vNIC and enable the SRIOV BIOS policy before enabling the IOMMU driver in the Linux kernel.

Step 1 In the the **Navigation** pane, click **Servers**.

Step 2 Expand **Servers** > **Service Profiles**

Step 3 Expand the node for the organization where you want to create the policy.

If the system does not include multitenancy, expand the **root** node.

Step 4 Select the service profile node where you want to enable SRIOV.

Step 5 In the Work pane, select **Policies** tab.

Step 6 In the Policies Area, expand **BIOS Policy**.

Step 7 Choose the default SRIOV policy from the **BIOS Policy** drop-down list.

Step 8 Click **Save Changes**.

Configuring RoCEv2 for NVMeoF on the Host System

Before you begin

Configure the server's service profile with RoCEv2 vNIC and the SRIOV enabled BIOS policy.

Step 1 Open the `/etc/default/grub` file for editing.

Step 2 Add `intel_iommu=on` to the end of the line for `GRUB_CMDLINE_LINUX`as shown in the sample file below.

```
sample /etc/default/grub configuration file after adding intel_iommu=on:
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap biosdevname=1 rhgb quiet
 intel_iommu=on
GRUB_DISABLE_RECOVERY="true"
```

Step 3 After saving the file, run the following command to generate a new grub.cfg file

For Legacy boot:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

For UEFI boot:

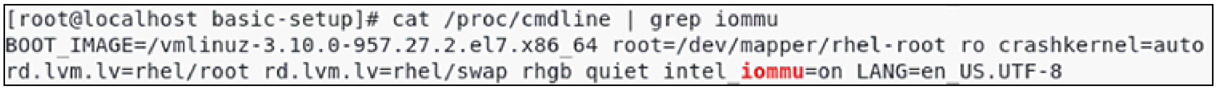
```
# grub2-mkconfig -o /boot/grub2/efi?EFI/redhat/grub.cfg
```

Step 4 Reboot the server. You must reboot your server for the changes to take after enabling IOMMU.

Step 5 Verify that the server booted with the `intel_iommu=on` option by checking the output file.

```
cat /proc/cmdline | grep iommu
```

Note its inclusion at the end of the output.



```
[root@localhost basic-setup]# cat /proc/cmdline | grep iommu
BOOT_IMAGE=/vmlinuz-3.10.0-957.27.2.el7.x86_64 root=/dev/mapper/rhel-root ro crashkernel=auto
rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet intel_iommu=on LANG=en_US.UTF-8
```

What to do next

Download the enic and enic_rdma drivers.

Installing Cisco enic and enic_rdma Drivers

The enic_rdma driver requires enic driver. When installing enic and enic_rdma drivers, download and use the matched set of enic and enic_rdma drivers on Cisco.com. Attempting to use the binary enic_rdma driver downloaded from Cisco.com with an inbox enic driver, will not work.

Step 1 Install the enic and enic_rdma rpm packages:

```
# rpm -ivh kmod-enic-<version>.x86_64.rpm kmod-enic rdma-<version>.x86_64.rpm
```

Note During enic_rdma installation, the enic_rdmalibnvdimm module may fail to install on RHEL 7.7 because the `nvdimm-security.conf` dracut module needs spaces in the `add_drivers` value. For workaround, please follow the instruction from the following links:

https://access.redhat.com/solutions/4386041

https://bugzilla.redhat.com/show_bug.cgi?id=1740383

Step 2 The enic_rdma driver is now installed but not loaded in the running kernel. Reboot the server to load enic_rdma driver into the running kernel.

Step 3 Verify the installation of enic_rdma driver and RoCE v2 interface:

```
# dmesg | grep enic_rdma
[    4.025979] enic_rdma: Cisco VIC Ethernet NIC RDMA Driver, ver 1.0.0.6-802.21 init
[    4.052792] enic 0000:62:00.1 eth1: enic_rdma: IPv4 RoCEv2 enabled
[    4.081032] enic 0000:62:00.2 eth2: enic_rdma: IPv4 RoCEv2 enabled
```

Step 4 Load the vme-rdma kernel module:

```
# modprobe nvme-rdma
```

After server reboot, nvme-rdma kernel module is unloaded. To load nvme-rdma kernel module every server reboot, create nvme_rdma.conf file using:

```
# echo nvme_rdma > /etc/modules-load.d/nvme_rdma.conf
```

Note For more information about enic_rdma after installation, use the **`rpm -q -l kmod-enic_rdma`** command to extract the README file.

What to do next

Discover targets and connect to NVMe namespaces. If your system needs multipath access to the storage, please go to the section for Setting Up Device Mapper Multipath.

Discovering the NVMe Target

Use this procedure to discover the NVMe target and connect NVMe namespaces.

Before you begin

Install **nvme-cli** version 1.6 or later if it is not installed already.

Note Skip to Step 2 below if nvme-cli version 1.7 or later is installed.

Configure the IP address on the RoCE v2 interface and make sure the interface can ping the target IP.

Step 1 Create an nvme folder in /etc, then manually generate host nqn.

```
# mkdir /etc/nvme
# nvme gen-hostnqn > /etc/nvme/hostnqn
```

Step 2 Create a settos.sh file and run the script to set priority flow control (PFC) in IB frames.

Note To avoid failure of sending NVMeoF traffic, you *must* create and run this script after *every* server reboot.

```
# cat settos.sh
#!/bin/bash
for f in `ls /sys/class/infiniband`;
do
        echo "setting TOS for IB interface:" $f
        mkdir -p /sys/kernel/config/rdma_cm/$f/ports/1
        echo 186 > /sys/kernel/config/rdma_cm/$f/ports/1/default_roce_tos
done
```

Step 3 Discover the NVMe target by entering the following command.

```
nvme discover --transport=rdma --traddr=<IP address of transport target port>
```

For example, to discover the target at 50.2.85.200:

```
# nvme discover --transport=rdma --traddr=50.2.85.200

Discovery Log Number of Records 1, Generation counter 2
=====Discovery Log Entry 0======
trtype:  rdma
adrfam:  ipv4
subtype: nvme subsystem
treq:    not required
portid:  3
trsvcid: 4420
subnqn:  nqn.2010-06.com.purestorage:flasharray.9a703295ee2954e
traddr:  50.2.85.200
rdma_prtype: roce-v2
rdma_qptype: connected
rdma_cms:    rdma-cm
rdma_pkey: 0x0000
```

Note To discover the NVMe target using IPv6, put the IPv6 target address next to the `traddr` option.

Step 4 Connect to the discovered NVMe target by entering the following command.

```
nvme connect --transport=rdma --traddr=<IP address of transport target port>> -n <subnqn value from
 nvme discover>
```

For example, to discover the target at 50.2.85.200 and the subnqn value found above:

```
# nvme connect --transport=rdma --traddr=50.2.85.200 -n
nqn.2010-06.com.purestorage:flasharray.9a703295ee2954e
```

Note To connect to the discovered NVMe target using IPv6, put the IPv6 target address next to the `traddr` option.

Step 5 Use the **nvme list** command to check mapped namespaces:

```
# nvme list
Node             SN                   Model                                    Namespace Usage
                 Format           FW Rev
---------------- -------------------- ---------------------------------------- ---------
-------------------------- ---------------- --------
/dev/nvme0n1     09A703295EE2954E     Pure Storage FlashArray                  72656       4.29  GB
 /   4.29  GB    512   B +  0 B   99.9.9
/dev/nvme0n2     09A703295EE2954E     Pure Storage FlashArray                  72657       5.37  GB
 /   5.37  GB    512   B +  0 B   99.9.9
```

Setting Up Device Mapper Multipath

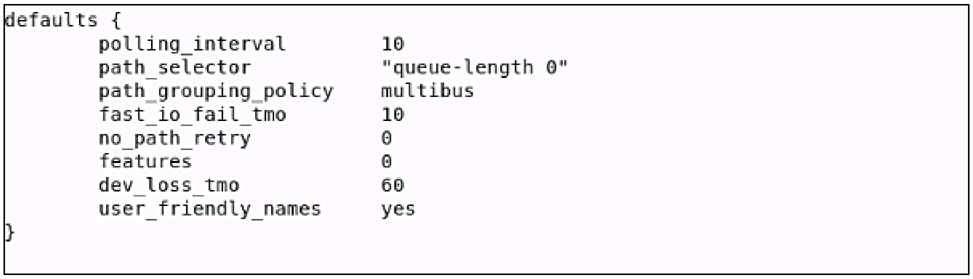
If your system is configured with configured with Device Mapper multipathing (DM Multipath), use the following steps to set up Device Mapper multipath.

Step 1 Install the `device-mapper-multipath` package if it is not installed already

Step 2 Enable and start multipathd:

```
# mpathconf --enable --with_multipathd y
```

Step 3 Edit the etc/multipath.conf file to use the following values :



```
defaults {
        polling_interval        10
        path_selector           "queue-length 0"
        path_grouping_policy    multibus
        fast_io_fail_tmo        10
        no_path_retry           0
        features                0
        dev_loss_tmo            60
        user_friendly_names     yes
}
```

Step 4 Flush with the updated multipath device maps.

```
# multipath -F
```

Step 5 Restart multipath service:

```
# systemctl restart multipathd.service
```

Step 6 Rescan multipath devices:

```
# multipath -v2
```

Step 7 Check the multipath status:

```
# multipath -ll
```

Deleting the RoCEv2 Interface Using UCS Manager

Use these steps to remove the RoCE v2 interface.

Step 1 In the **Navigation** pane, click **Servers**.

Step 2 Expand **Servers** > **Service Profiles**.

Step 3 Expand the node for the organization where you want to create the policy. If the system does not include multitenancy, expand the **root** node

Step 4 Click on **vNICs** and go to the **Network** tab in the work area.

Modify the vNIC policy, according to the steps below.

a) On the **Network** tab, scroll down to the desired vNIC and click on it, then click **Modify**.
b) A popup dialog box will appear. Scroll down to the **Adapter Performance Profile** area, and click on the dropdown area for the Adapter Policy. Choose **Linux** from the drop-down list.
c) Click **OK**.

Step 5 Click **Save Changes**.