

# How to Install the NVMe Host Module Patch on Compute Nodes under Ubuntu

This section provides the steps to installing the NVMe host module patch on compute nodes using Ubuntu.

To install the NVMe host module patch for Ubuntu complete the steps below on each compute node. All steps are applicable for Ubuntu-20.04.x (kernel 5.4.0-x-generic) and Ubuntu 18.04; this process has been tested on Ubuntu 18.04.5 LTS.

## Step 1. Compile the NVMe host kernel modules

To compile the NVMe host kernel modules, complete these following:

a) Install additional tools:

```
sudo apt-get install -y make gcc
```

b) Install the kernel source and development packages for the target kernel. For example, for version 5.13.0-30-generic, see <https://packages.ubuntu.com/bionic-updates/linux-image-5.4.0-137-generic>.

c) Apply the patch received from Kioxia:

```
SRC_DIR=/root/kernel_src/5.13.0-30-generic/drivers/nvme/host
PATH_TO_PATCH=<path-to-the-patch-dir>/nvme-host-fast-io-fail-patch-5.13.0-30-generic-1.0-0-all.patch
cd ${SRC_DIR}
patch --dry-run -p1 -i ${PATH_TO_PATCH}
patch -p1 -i ${PATH_TO_PATCH}
```

d) Define MODULE\_DESCRIPTION to identify the applied patch:

```
BUILD_TIME=$(date +%y%m%d%H%M)
DESCRIPTION="\nvme host Kioxia patch (${BUILD_TIME})\"
sed -i "/MODULE_DESCRIPTION.*/d" core.c
echo -e "\nMODULE_DESCRIPTION(${DESCRIPTION});" >> core.c
```

e) Build the patched kernel modules and sign them:

```
KDIR=/lib/modules/5.13.0-30-generic/build/
make -C ${KDIR} M=${SRC_DIR}
cd ${SRC_DIR}
KOS=$(ls *.ko)
for module in ${KOS[*]}; do echo "sign ${module}"; strip --strip-unneeded ${module}; ${KDIR}/scripts/sign-file sha256 ${KDIR}/certs/signing_key.pem ${KDIR}/certs/signing_key.x509 ${module}; done
```

NOTE: If the signing fails, generate keys to use with the kernel module you need to sign and copy them to /usr/src/kernels/\${uname -r}/certs/:

```
openssl req -new -x509 -newkey rsa:2048 -keyout signing_key.pem -outform DER -out signing_key.x509 -nodes -subj "/CN=Owner/"
cp signing_key.x509 signing_key.pem /usr/src/kernels/${uname -r}/certs/
```

f) Confirm the five kernel modules listed below are produced:

- nvme-core.ko
- nvme.ko
- nvme-fabrics.ko
- nvme-tcp.ko
- nvme-rdma.ko

## Step 2. Install the patched modules

To install the patched modules, complete these following:

a) Assuming the five patched modules listed in step 1(f) are placed in the current directory, verify that their version matches currently running kernel version:

```
modinfo *.ko | grep vermagic | awk '{print $2}'
uname -r
```

b) Define bash variables:

```
KVER=$(uname -r)
SAVED_DIR="/root/tmp/orig_${KVER}"
MODULES_DIR="/lib/modules/${KVER}/kernel/drivers/nvme/host"
```

c) Create a backup of the **initramfs** image:

```
cp /boot/initramfs-"$KVER".img /boot/saved-initramfs-"$KVER".img
```

d) Backup the original modules and replace them with the patched version:

```
mkdir -p ${SAVED_DIR}
cd ${MODULES_DIR}
tar -cvf ${SAVED_DIR}/${KVER}_kos.tar ./*.ko*
rm -f ${MODULES_DIR}/*.ko*
cp -f ${SRC_DIR}/*.ko ${MODULES_DIR}/
```

e) Regenerate dependencies and **initrd**:

```
/sbin/depmod -ae -F /boot/System.map-$(uname -r)
dracut --force -H
```

f) If the nvme (pci) module cannot be unloaded, reboot the machine:

```
reboot
```

Otherwise, unload the original modules and load the patched versions:

```
modprobe -r nvme-tcp
modprobe -r nvme-rdma
modprobe -r nvme-fc
modprobe -r nvme-fabrics
modprobe -r nvme
modprobe -r nvme-core
modprobe nvme-core
modprobe nvme
modprobe nvme-fabrics
modprobe nvme-tcp
modprobe nvme-rdma
```

**Step 3. Install the NVMe patch**

To install the NVMe patch, complete these following:

a) Confirm you have disabled Secure Boot from your firmware (BIOS). Note that you will of course lose the protection provided by UEFI Secure Boot.

b) Install the NVMe host patch:

```
sudo dpkg -i nvme-host-fast-io-fail-patch-5.4.0-91-generic-1.0-0-all.deb
```

c) Verify that patch has been installed by running:

```
modinfo nvme-core|grep description
```

It should return:

```
nvme host Kioxia patch (nvme-host-fast-io-fail-patch-5.4.0-91-generic-1.0-0-all)
```

You are now ready to use the compute nodes with KumoScale. If you need to uninstall the NVMe patch and restore the original kernel modules, see [How to uninstall the NVMe patch and restore original kernel modules.](#)

---

---

---