README

Precompiled Benefits

- Precompiled \rightarrow faster boot up after driver and/or kernel updates
- Pre-tested \rightarrow kernel and driver combination has been validated
- Removes gcc dependency \rightarrow no compiler installation required
- Removes dkms dependency \rightarrow enabling EPEL repository not required
- Removes kernel-devel and kernel-headers dependencies → no black screen if matching packages are missing

Below is a preview of packaging improvements for the NVIDIA driver on Red Hat Enterprise Linux (RHEL) 8

- 1. For supported RHEL kernel releases (see below), driver packages are provided that implement an alternative to DKMS. The new approach does not require gcc to be installed anymore, nor does the EPEL repository need to be enabled. The source files for these driver kmod packages are compiled in advance and then linked at installation time, hence these are called "precompiled drivers".
 - When a new driver update is released, precompiled driver packages are provided only for the most recently released kernel at the time of the driver update; likewise, if a new kernel update is released, precompiled driver packages will be provided for this kernel. Another way to phrase this is that at any point in time, precompiled drivers are enabled for the most recent kernel and the most recent driver at this point in time. Thus, this is the same constraint as for other kernel modules that are shipped as part of RHEL. If using an older kernel release, the matching older precompiled driver release can be used. Note that "most recent kernel" includes all RHEL z-stream updates.
 - In some cases such as embargoed kernel updates, precompiled driver packages for a kernel may not be publicly available before the availability of this kernel. When using precompiled drivers, a *dnf plugin* is enabled that prevents upgrading to a kernel for which no precompiled driver exists yet (a warning will be displayed by dnf during such an upgrade situation). This can delay the application of security fixes, but ensures that a tested kernel/driver combination is used at all times. We expect any potential delays to be short.
- 2. Using Modularity, the tech preview repository provides multiple update streams of the driver packages. Only updates on the selected stream will be considered. Update streams correspond to either all precompiled driver updates on specific driver branches (e.g., drivers with major version equal to "440").
- 3. Stale .ko files are cleaned up via a dnf plugin.

Support Matrix

- Currently these package improvements are supported for RHEL 8.1 (and newer) on x86_64 architecture only.
 - During this tech preview, we are providing precompiled driver packages only for kernel 4.18.0-147.8.1 and later.
 - If you use an earlier kernel, please update to get a precompiled driver package.
 - Precompiled drivers are not provided for RHEL EUS kernels.
- This tech preview will use Tesla Drivers from the 440 branch, 440.33.01 and later.
 - The repository for the tech preview has an end of life (EOL) of July 1, 2020.
 - After that date please remove the tech preview repo from your system and enable the CUDA repository.
- New kmod packages should be available within 24 hours of a new RHEL kernel update.

• To prevent system breakages, the dnf plugin will block kernel updates between a kernel going live and kmod package availability.

Pre-installation Actions

- · Remove any existing NVIDIA driver installation:
 - To uninstall a CUDA toolkit runfile installation

\$ sudo /usr/local/cuda-X.Y/bin/cuda-uninstall

• To uninstall a standalone NVIDIA driver runfile installation:

\$ sudo /usr/bin/nvidia-uninstall

· To uninstall an RPM installation:

```
$ sudo dnf remove nvidia-driver
$ sudo dnf module reset nvidia-driver
$ sudo dnf config-manager --set-disabled cuda
```

• On RHEL 8 Linux (x86_64 workstation), ensure AppStream, BaseOS and CRB repositories are enabled:

```
$ subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
$ subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
$ subscription-manager repos --enable=codeready-builder-for-rhel-8-x86_64-rpms
```

· Add the tech preview repository:

```
$ sudo dnf config-manager --add-repo=https://developer.download.nvidia.com/compute/cuda/preview/repos/rhel8
/x86_64/techpreview_nvidia_rh_drv.repo
```

Precompiled (default and recommended)

Verify that you have a supported kernel installed.

\$ uname -r 4.18.0-147.8.1.el8_1.x86_64

DKMS (non-supported kernels)

Note: This is only needed for special circumstances such as custom kernels.

• Install development packages for the installed kernel.

\$ sudo dnf install kernel-devel-\$(uname -r) kernel-headers-\$(uname -r)

• Make sure the EPEL repository is enabled.

\$ sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm

Package Manager Installation:

Select an installation stream

Choose only one stream from the three options below:

1. latest always updates to the highest versioned driver (precompiled)

\$ sudo dnf module install nvidia-driver:latest

2. <id> locks the driver updates to the specified driver branch (precompiled)

\$ sudo dnf module install nvidia-driver:<id>

note: replace <id> with the appropriate driver branch. For the tech preview, valid values are 440.

3. latest-dkms always updates to the highest versioned driver (non-precompiled)

\$ sudo dnf module install nvidia-driver:latest-dkms

note: this is only recommended for unsupported kernels.

Switch streams

To switch to another stream, first remove the driver packages

\$ sudo dnf remove nvidia-driver

Then reset the module stream

\$ sudo dnf module reset nvidia-driver

Now the driver can be installed from an appropriate stream. Refer to the "Select an installation stream" section above for the steps to do so.

Feedback

The purpose of launching a tech preview is to gauge interest and receive customer feedback.

- Comments: compute_installer@nvidia.com
- Report a bug: https://developer.nvidia.com/nvidia_bug/add

note: If you are not already a member, join the NVIDIA Developer Program: https://developer.nvidia.com/join

• GitHub: https://github.com/NVIDIA/yum-packaging-precompiled-kmod