TABLE OF CONTENTS

Chapter 1. Introduction ........................................................................................................ 1
  1.1. Pre-Installation Requirements ................................................................................. 1
  1.2. Installer Formats ....................................................................................................... 1
Chapter 2. Runfile Installers .............................................................................................. 2
Chapter 3. Package Managers ............................................................................................ 3
  3.1. Ubuntu LTS ............................................................................................................... 3
  3.2. Red Hat Enterprise Linux 8 .................................................................................... 4
  3.3. CentOS 8 .................................................................................................................. 6
  3.4. Red Hat Enterprise Linux 7 .................................................................................... 7
  3.5. CentOS 7 ................................................................................................................ 9
  3.6. SUSE Linux Enterprise Server 15 .......................................................................... 10
Chapter 1.
INTRODUCTION

The purpose of this document is to provide some quick start notes for installing NVIDIA drivers on Linux distributions for servers. For more comprehensive information on installing CUDA Toolkit and drivers on various Linux distributions (for both workstation and server variants), refer to the CUDA Installation Guide for Linux.

1.1. Pre-Installation Requirements

Before installing the NVIDIA driver on Linux, some pre-installation steps are recommended to:

1. Verify the system has a CUDA-capable GPU
2. Verify the system is running a supported version of Linux
3. Verify the system has build tools such as make, gcc installed
4. Verify the system has correct Linux kernel headers

For more detailed steps on completing each of these pre-installation steps, refer to the pre-installation actions in the CUDA Installation Guide for Linux.

Mixing installation methods (e.g. .run installers and .deb/.rpm packages) may result in driver conflicts and an unusable system. Refer to this section in the guide above for other pre-install actions to handle conflicting installation methods.

1.2. Installer Formats

NVIDIA drivers are available in three formats for use with Linux distributions:

- Runfile installers
- Package managers
- Containerized drivers
Chapter 2.
RUNFILE INSTALLERS

NVIDIA drivers are available as .run installer packages for use with Linux distributions from the NVIDIA driver downloads site. Select the .run package for your GPU product.

This page also includes links to all the current and previous driver releases: https://www.nvidia.com/en-us/drivers/unix.

The .run can also be downloaded using wget or curl as shown in the example below:

```bash
$ BASE_URL=https://us.download.nvidia.com/tesla
$ DRIVER_VERSION=450.80.02

$ curl -fSsl -O $BASE_URL/$DRIVER_VERSION/NVIDIA-Linux-x86_64-$DRIVER_VERSION.run
```

Once the .run installer has been downloaded, the NVIDIA driver can be installed:

```bash
$ sudo sh NVIDIA-Linux-x86_64-$DRIVER_VERSION.run
```

Follow the prompts on the screen during the installation. For more advanced options on using the .run installer, see the --help option or refer to the README.
Chapter 3.
PACKAGE MANAGERS

The Package Manager installation interfaces with your system's package management system. When using RPM or Deb, the downloaded package is a repository package. Such a package only informs the package manager where to find the actual installation packages, but will not install them.

The next few sections provide steps on installing the NVIDIA Tesla driver on these supported distributions:

‣ Ubuntu LTS
‣ Red Hat Enterprise Linux 8
‣ CentOS 8
‣ Red Hat Enterprise Linux 7
‣ CentOS 7
‣ SUSE Linux Enterprise Server 15

3.1. Ubuntu LTS

This section includes instructions for installing the NVIDIA driver on Ubuntu 16.04 LTS and Ubuntu 18.04 LTS distributions using the package manager.

1. The NVIDIA driver requires that the kernel headers and development packages for the running version of the kernel be installed at the time of the driver installation, as well whenever the driver is rebuilt. For example, if your system is running kernel version 4.4.0, the 4.4.0 kernel headers and development packages must also be installed.

The kernel headers and development packages for the currently running kernel can be installed with:

$ sudo apt-get install linux-headers-$\{(uname -r)\}

2. Ensure packages on the CUDA network repository have priority over the Canonical repository.

$ distribution=\$(. /etc/os-release; echo $ID$VERSION_ID | sed -e 's/\./\//g')

3. Install the CUDA repository public GPG key. Note that on Ubuntu 16.04, replace `https` with `http` in the command below.

   ```
   $ sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/$distribution/x86_64/7fa2af80.pub
   ```

4. Setup the CUDA network repository.

   ```
   $ echo "deb http://developer.download.nvidia.com/compute/cuda/repos/$distribution/x86_64 /" | sudo tee /etc/apt/sources.list.d/cuda.list
   ```

5. Update the APT repository cache and install the driver using the cuda-drivers meta-package. Use the `--no-install-recommends` option for a lean driver install without any dependencies on X packages. This is particularly useful for headless installations on cloud instances.

   ```
   $ sudo apt-get update
   $ sudo apt-get -y install cuda-drivers
   ```

6. Follow the post-installation steps in the CUDA Installation Guide for Linux to setup environment variables, NVIDIA persistence daemon (recommended) and to verify the successful installation of the driver.

### 3.2. Red Hat Enterprise Linux 8

This section includes instructions for installing the NVIDIA driver on RHEL 8 using the package manager.

1. If not done already (for example when using RHEL on cloud instances), register the system using `subscription-manager`. For more information, refer to this KB article on the Red Hat Customer Portal.

   ```
   $ sudo subscription-manager register --username <insert-username> --password <insert-password> --auto-attach
   ```

2. Satisfy the external dependency on EPEL for DKMS and enable any optional repos for RHEL 8 only.

   ```
   
   $ ARCH=${(/bin/arch)}
   
   $ sudo subscription-manager repos --enable codeready-builder-for-rhel-8-$ARCH-rpms --enable rhel-8-for-{$ARCH}-baseos-rpms --enable rhel-8-for-$ARCH-appstream-rpms
   ```
3. Install the CUDA repository public GPG key.

```bash
$ distribution=$(./etc/os-release;echo $ID`rpm -E "%{?rhel}%{?fedora}"")
```

4. Setup the CUDA network repository.

```bash
```

5. The NVIDIA driver requires that the kernel headers and development packages for the running version of the kernel be installed at the time of the driver installation, as well whenever the driver is rebuilt. For example, if your system is running kernel version 4.4.0, the 4.4.0 kernel headers and development packages must also be installed.

For RHEL 8, ensure that the system has the correct Linux kernel sources from the Red Hat repositories:

```bash
$ sudo dnf install -y kernel-devel-$(uname -r) kernel-headers-$(uname -r)
```

6. The choice of the driver package depends on the stream and profile to be used as shown below:

```bash
$ sudo dnf module install nvidia-driver:<stream>/<profile>
```

**stream** depends on the driver branch and type (e.g. precompiled). To install the latest driver, choose the **latest-dkms** driver stream. For more information on the supported streams, refer to the support matrix.

**profile** by default is "default" and does not need to be specified. Supported **profiles** can be chosen based on the use-case:

**Table 1 List of driver profiles available**

<table>
<thead>
<tr>
<th>Stream</th>
<th>Profile</th>
<th>Use-case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>/default</td>
<td>Installs all the driver packages in a stream.</td>
</tr>
<tr>
<td>Kickstart</td>
<td>/ks</td>
<td>Performs unattended Linux OS installation using a config file.</td>
</tr>
<tr>
<td>NVSwitch Fabric</td>
<td>/fm</td>
<td>Installs all the driver packages and components required for bootstrapping an NVSwitch system.</td>
</tr>
</tbody>
</table>

7. Update the repository cache and install the driver using the **nvidia-driver** meta-package. In this example, we use the default **profile**. For NVSwitch systems, use `nvidia-driver:latest-dkms/fm`.

```bash
$ sudo dnf clean all

$ sudo dnf -y module install nvidia-driver:latest-dkms
```
8. Follow the **post-installation steps** in the CUDA Installation Guide for Linux to setup environment variables, NVIDIA persistence daemon (recommended) and to verify the successful installation of the driver.

9. Optionally, if tearing down a cloud instance with RHEL, remove and unregister entitlements from Red Hat if the instance is no longer required.

   ```
   $ sudo subscription-manager remove --all
   $ sudo subscription-manager unregister
   ```

### 3.3. CentOS 8

This section includes instructions for installing the NVIDIA driver on CentOS 8 using the package manager.

1. In some cases, you may need to install some additional dependencies that are required for installing the NVIDIA drivers.

   ```
   $ sudo dnf install -y tar bzip2 make automake gcc gcc-c++ pciutils elfutils-libelf-devel libglvnd-devel
   ```

2. Enable the **PowerTools** repo and set up the external dependency on **EPEL** for DKMS.

   ```
   $ sudo dnf config-manager --set-enabled PowerTools
   $ sudo dnf install -y epel-release
   ```

3. Install the CUDA repository public GPG key.

   ```
   $ distribution=rhel8
   ```

4. Setup the CUDA network repository.

   ```
   $ ARCH=$( /bin/arch )
   ```

5. The NVIDIA driver requires that the kernel headers and development packages for the running version of the kernel be installed at the time of the driver installation, as well whenever the driver is rebuilt. For example, if your system is running kernel version 4.4.0, the 4.4.0 kernel headers and development packages must also be installed.

   For CentOS 8, ensure that the system has the correct Linux kernel sources from the CentOS repositories:

   ```
   $ sudo dnf install -y kernel-devel-$\{uname -r\} kernel-headers-$\{uname -r\}
   ```
Note that in some cases, you may have to update the running kernel as a prerequisite:

```bash
$ sudo dnf install -y kernel kernel-core kernel-modules
```

6. The choice of the driver package depends on the stream and profile to be used as shown below:

```bash
$ sudo dnf module install nvidia-driver:<stream>/<profile>
```

*stream* depends on the driver branch and type (e.g. precompiled). To install the latest driver, choose the *latest-dkms* driver stream. For more information on the supported streams, refer to the support matrix.

*profile* by default is "default” and does not need to be specified. Supported *profiles* can be chosen based on the use-case:

**Table 2 List of driver profiles available**

<table>
<thead>
<tr>
<th>Stream</th>
<th>Profile</th>
<th>Use-case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>/default</td>
<td>Installs all the driver packages in a stream.</td>
</tr>
<tr>
<td>Kickstart</td>
<td>/ks</td>
<td>Performs unattended Linux OS installation using a config file.</td>
</tr>
<tr>
<td>NVSwitch Fabric</td>
<td>/fm</td>
<td>Installs all the driver packages and components required for bootstrapping an NVSwitch system.</td>
</tr>
</tbody>
</table>

7. Update the repository cache and install the driver using the *nvidia-driver* metapackage. In this example, we use the default *profile*. For NVSwitch systems, use *nvidia-driver:latest-dkms/fm*.

```bash
$ sudo dnf clean all
$ sudo dnf -y module install nvidia-driver:latest-dkms
```

8. Follow the *post-installation steps* in the CUDA Installation Guide for Linux to setup environment variables, NVIDIA persistence daemon (recommended) and to verify the successful installation of the driver.

### 3.4. Red Hat Enterprise Linux 7

This section includes instructions for installing the NVIDIA driver on RHEL 7 using the package manager.
1. If not done already (for example when using RHEL on cloud instances), register the system using `subscription-manager`. For more information, refer to this KB article on the Red Hat Customer Portal.

   ```
   $ sudo subscription-manager register --username <insert-username> --password <insert-password> --auto-attach
   ```

2. Satisfy the external dependency on EPEL for DKMS and enable any optional repos for RHEL 7.

   ```
   $ sudo subscription-manager repos --enable="rhel-*-optional-rpms" --enable="rhel-*-extras-rpms" --enable="rhel-ha-for-rhel-*-server-rpms"
   ```

3. Install the CUDA repository public GPG key.

   ```
   $ distribution=$(./ /etc/os-release;echo $ID`rpm -E "%{?rhel}%{?fedora}"`)  
   ```

4. Setup the CUDA network repository.

   ```
   $ ARCH=$( /bin/arch )  
   ```

5. The NVIDIA driver requires that the kernel headers and development packages for the running version of the kernel be installed at the time of the driver installation, as well whenever the driver is rebuilt. For example, if your system is running kernel version 4.4.0, the 4.4.0 kernel headers and development packages must also be installed.

   For RHEL 7, ensure that the system has the correct Linux kernel sources from the Red Hat repositories:

   ```
   $ sudo yum install -y kernel-devel-$uname_r kernel-headers-$uname_r
   ```

6. Update the repository cache and install the driver using the `nvidia-driver-latest-dkms` meta-package.

   ```
   $ sudo yum clean expire-cache
   $ sudo yum install -y nvidia-driver-latest-dkms
   ```

7. Follow the post-installation steps in the CUDA Installation Guide for Linux to setup environment variables, NVIDIA persistence daemon (recommended) and to verify the successful installation of the driver.

8. Optionally, if tearing down a cloud instance with RHEL, remove and unregister entitlements from Red Hat if the instance is no longer required.

   ```
   $ sudo subscription-manager remove --all
   ```
3.5. CentOS 7

This section includes instructions for installing the NVIDIA driver on CentOS 7 using the package manager.

1. In some cases, you may need to install some additional dependencies that are required for installing the NVIDIA drivers.

   ```
   $ sudo dnf install -y tar bzip2 make automake gcc gcc-c++ pciutils elfutils-
   libelf-devel libglvnd-devel iptables firewalld vim bind-utils wget
   ```

2. Satisfy the external dependency on EPEL for DKMS.

   ```
   $ sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-
   latest-7.noarch.rpm
   ```

3. Install the CUDA repository public GPG key.

   ```
   $ distribution=rhel7
   ```

4. Setup the CUDA network repository.

   ```
   $ ARCH=$( /bin/arch )
   compute/cuda/repos/$distribution/${ARCH}/cuda-$distribution.repo
   ```

5. The NVIDIA driver requires that the kernel headers and development packages for the running version of the kernel be installed at the time of the driver installation, as well whenever the driver is rebuilt. For example, if your system is running kernel version 4.4.0, the 4.4.0 kernel headers and development packages must also be installed.

   For CentOS 7, ensure that the system has the correct Linux kernel sources from the CentOS repositories:

   ```
   $ sudo yum install -y kernel-devel-$(uname -r) kernel-headers-$(uname -r)
   ```

6. Update the repository cache and install the driver using the `nvidia-driver-

   ```
   latest-dkms` meta-package.

   ```
   $ sudo yum clean expire-cache
   $ sudo yum install -y nvidia-driver-latest-dkms
   ```

7. Follow the post-installation steps in the CUDA Installation Guide for Linux to setup environment variables, NVIDIA persistence daemon (recommended) and to verify the successful installation of the driver.
3.6. SUSE Linux Enterprise Server 15

This section includes instructions for installing the NVIDIA driver on SLES 15 using the package manager.

1. Install the CUDA repository public GPG key.

   $ distribution=$(./etc/os-release;echo $ID$VERSION_ID | sed -e 's/\.[0-9]//';)

2. Setup the CUDA network repository.


3. If not already done, activate the SUSE Package Hub with SUSEConnect. On OpenSUSE systems, this step can be skipped.

   $ sudo SUSEConnect -p PackageHub/15.1/x86_64

4. Update the repository cache.

   $ sudo zypper refresh

5. The NVIDIA driver requires that the kernel headers and development packages for the running version of the kernel be installed at the time of the driver installation, as well whenever the driver is rebuilt. For example, if your system is running kernel version 4.4.0, the 4.4.0 kernel headers and development packages must also be installed.

   For SUSE, ensure that the system has the correct Linux kernel sources from the SUSE repositories.

   Use the output of the `uname` command to determine the running kernel's version and variant:

   $ uname -r
   4.12.14-1p151.27-default

   In this example, the version is `4.12.14-1p151.27` and the variant is `default`. The kernel headers and development packages can then be installed with the following command, replacing `<variant>` and `<version>` with the variant and version discovered from the previous `uname` command:

   $ sudo zypper install -y kernel-<variant>-devel=<version>

6. Proceed to install the driver using the `cuda-drivers` meta-package.

   $ sudo zypper install -y cuda-drivers

7. On SUSE systems, add the user to the video group.
8. A reboot of the system may be required to verify that the NVIDIA driver modules are loaded and the devices visible under `/dev`.

```
$ sudo reboot
```

9. Follow the post-installation steps in the CUDA Installation Guide for Linux to setup environment variables, NVIDIA persistence daemon (recommended) and to verify the successful installation of the driver.
Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this guide. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the Unites States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2020 NVIDIA Corporation. All rights reserved.