**step-by-step guide to install and use the CUDA Toolkit on Red Hat (RHEL)**

**Objective**

Here’s a step-by-step guide to install and use the CUDA Toolkit on Red Hat (RHEL) Linux:

**Prerequisites:**

* **A supported NVIDIA GPU** installed in your machine.
* **Root (administrator) access** to the machine.
* **The correct NVIDIA drivers** installed for your GPU.

**Step 1: Install NVIDIA Driver**

Before installing CUDA, ensure the appropriate NVIDIA driver is installed for your GPU.

**1.1. Add the NVIDIA repository:**

Red Hat Linux doesn't come with NVIDIA drivers by default, so you will need to install them manually.

* First, download the repository RPM from NVIDIA’s site:
  + [NVIDIA Driver for RHEL](https://www.nvidia.com/Download/index.aspx?lang=en-us).
* Add the repository to your system:

lspci | grep -i nvidia

sudo dnf install -y epel-release

sudo dnf install -y https://download.nvidia.com/centos/7/x86\_64/nvidia-driver.repo

**1.2. Install the NVIDIA driver:**

* After adding the repository, install the NVIDIA driver package:

sudo dnf install -y nvidia-driver

**1.3. Reboot the system:**

After the driver installation, reboot the machine to ensure the driver is properly loaded:

sudo reboot

**1.4. Verify installation:**

After rebooting, check if the NVIDIA driver is properly installed:

nvidia-smi

This command should display information about the installed GPU.

**Step 2: Install CUDA Toolkit**

**2.1. Download CUDA Toolkit**

* Go to the [CUDA Toolkit Download Page](https://developer.nvidia.com/cuda-toolkit).
* Select the version of CUDA that is compatible with your system (you will want the version for RHEL or CentOS).
  + Select the .rpm package (this is typically for RHEL/CentOS).

**2.2. Install the CUDA Toolkit**

* Once you’ve downloaded the CUDA .rpm file (e.g., cuda-repo-rhel7-<version>.x86\_64.rpm), navigate to the directory where the file is located.
* Install the CUDA repository package:

sudo rpm -i cuda-repo-rhel7-<version>.x86\_64.rpm

**2.3. Install CUDA Toolkit:**

* Now, install the actual CUDA Toolkit using DNF (Red Hat’s package manager):

sudo dnf clean all

sudo dnf install cuda

**2.4. Update Environment Variables:**

You will need to update the PATH and LD\_LIBRARY\_PATH variables to ensure that CUDA is found by your system.

* Add the following lines to your .bashrc or .bash\_profile file:

export PATH=/usr/local/cuda-<version>/bin:$PATH

export LD\_LIBRARY\_PATH=/usr/local/cuda-<version>/lib64:$LD\_LIBRARY\_PATH

* After adding, reload your shell configuration file:

source ~/.bashrc

**2.5. Verify CUDA Installation:**

To verify that CUDA has been installed properly, check the version of the installed CUDA Toolkit:

nvcc --version

This should display the installed version of CUDA.

**Step 3: Install cuDNN (Optional)**

If you're planning to use CUDA for deep learning tasks, you may also need to install **cuDNN** (CUDA Deep Neural Network library).

**3.1. Download cuDNN:**

* Go to the [cuDNN Download Page](https://developer.nvidia.com/cudnn).
* Select the version of cuDNN compatible with your CUDA version.

**3.2. Install cuDNN:**

After downloading the cuDNN tar file, extract it and copy the files to the appropriate CUDA directories:

tar -xzvf cudnn-<version>-linux-x64-v<version>.tgz

sudo cp cuda/include/cudnn\*.h /usr/local/cuda/include

sudo cp cuda/lib64/libcudnn\* /usr/local/cuda/lib64

sudo chmod a+r /usr/local/cuda/include/cudnn\*.h /usr/local/cuda/lib64/libcudnn\*

**Step 4: Test CUDA with Sample Programs**

**4.1. Compile and Run Sample Programs:**

CUDA comes with some sample programs that can be compiled and tested to ensure everything is working properly.

* Navigate to the sample directory:

cd /usr/local/cuda-<version>/samples

* Compile the samples:

sudo make

* Run a sample program (e.g., deviceQuery):

cd /usr/local/cuda-<version>/samples/1\_Utilities/deviceQuery

./deviceQuery

If everything is set up correctly, this should output details about your GPU.

**Step 5: Using CUDA**

Now that you have CUDA installed, you can start writing and compiling CUDA programs.

* **Write a simple CUDA program** using your preferred text editor. For example:

#include <iostream>

\_\_global\_\_ void hello\_cuda() {

printf("Hello, CUDA!\n");

}

int main() {

hello\_cuda<<<1, 1>>>();

cudaDeviceSynchronize();

return 0;

}

* **Compile your program**:

nvcc -o hello\_cuda hello\_cuda.cu

* **Run the program**:

./hello\_cuda

This should print Hello, CUDA! to the console.