We're giving away 1,500 DJI Tello drones. **Enter to win ›**

TUTORIAL

Setting up an x86 system to build and package software for IBM POWER

Including applications using Mellanox OFED and NVIDIA CUDA

By Chris Ward, Paul Clarke | Published November 20, 2018 - Updated November 20, 2018

Linux Systems

Introduction

This tutorial explains how to set up an x86 system to build and package software to run on an IBM POWER processor-based system running the Linux operating system. This is useful if you want to develop and build software on your x86 notebook or desktop, but your customers want to use the software you develop on their IBM POWER hardware running Linux. It will take most of a working day for the x86 system to be set up for this; however most of the time is unattended and so you can get on with other work while the installation is in progress. All the software you need is available at no charge; much of it is open source.

This tutorial was originally written for a collaborator who develops an application which uses NVIDIA GPUs and Mellanox InfiniBand adapters, so the instructions include getting these software components setup as well.

Mellanox OFED is software for driving Mellanox InfiniBand adapters. It is a no-charge download from the Mellanox website.

NVIDIA CUDA is software for driving NVIDIA GPUs for the purpose of accelerated computation rather than graphical display. It is a no-charge download from the NVIDIA website.

Prerequisites

- An x86 system, 64-bit, with at least 2 GB of RAM, at least two processors, and at least 100 GB of disk space.
- An openSUSE DVD image for the x86 system. Available for download from: http://download.opensuse.org/tumbleweed/iso/openSUSE-Tumbleweed-DVD-x86_64-Current.iso
- An openSUSE DVD image for ppc64le. Available for download from: http://download.opensuse.org/ports/ppc/tumbleweed/iso/openSUSE-Tumbleweed-DVD-ppc64le-Current.iso
- Mellanox OFED software for SLES 15.0. Available from:http://www.mellanox.com/page/products_dyn?product_family=26

  On the Download tab, select the current version, SLES, SLES15 SP0, ppc64le, .tgz file.
- NVIDIA CUDA. Available from: https://developer.nvidia.com/cuda-downloads

  Select Linux, ppc64le, RHEL, 7, rpm (local).

Estimated time

It took my computer 8 hours to complete the installation; but I used an old and slow desktop computer. A modern computer would take about half this time.

Steps

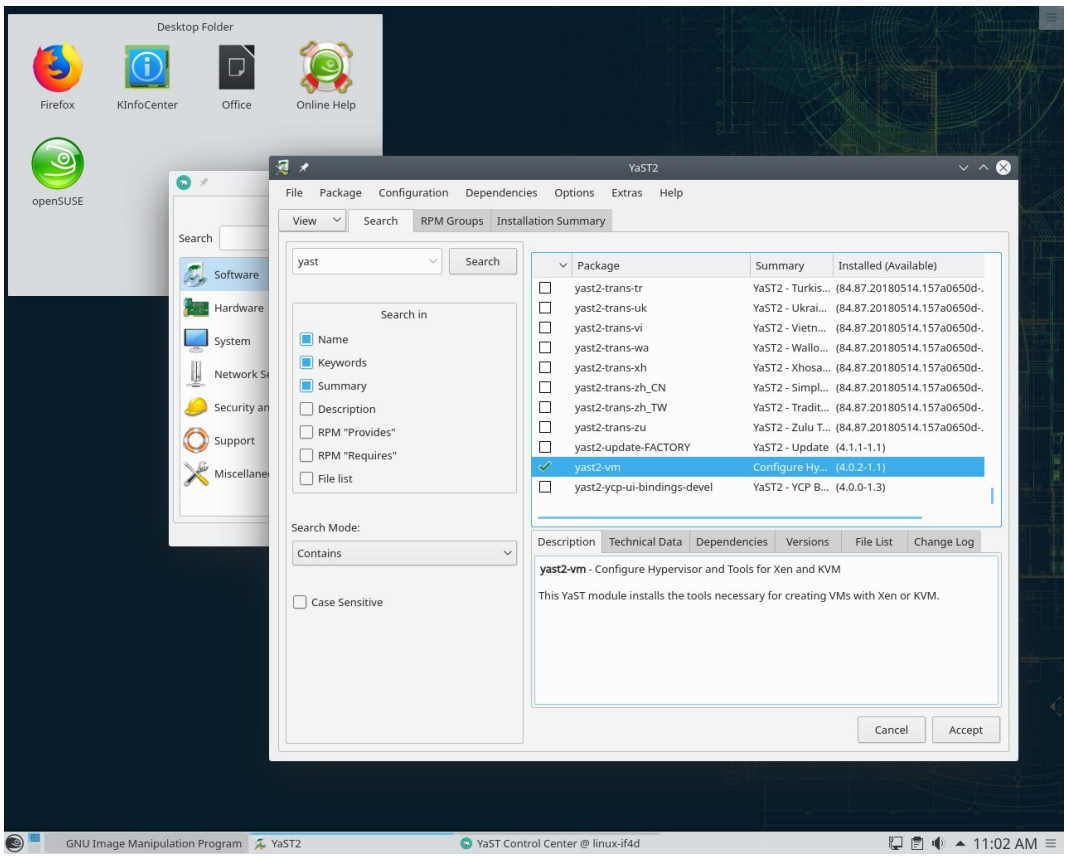
Here are the stages for installing and configuring your virtual POWER system.

1. Install your x86 system with the openSUSE Tumbleweed DVD. You can use a flash drive for this if you prefer. Tumbleweed has a graphical installer, which explains each step of the process.

   Allocate a root file system of at least 50 GB; I recommend **ext4** for root file system rather than the default **BtrFS**, because **BtrFS** needs space for snapshots that are unnecessary in this application. The root file system needs to be this large because it contains the virtual disk for the virtual POWER system that you will create.

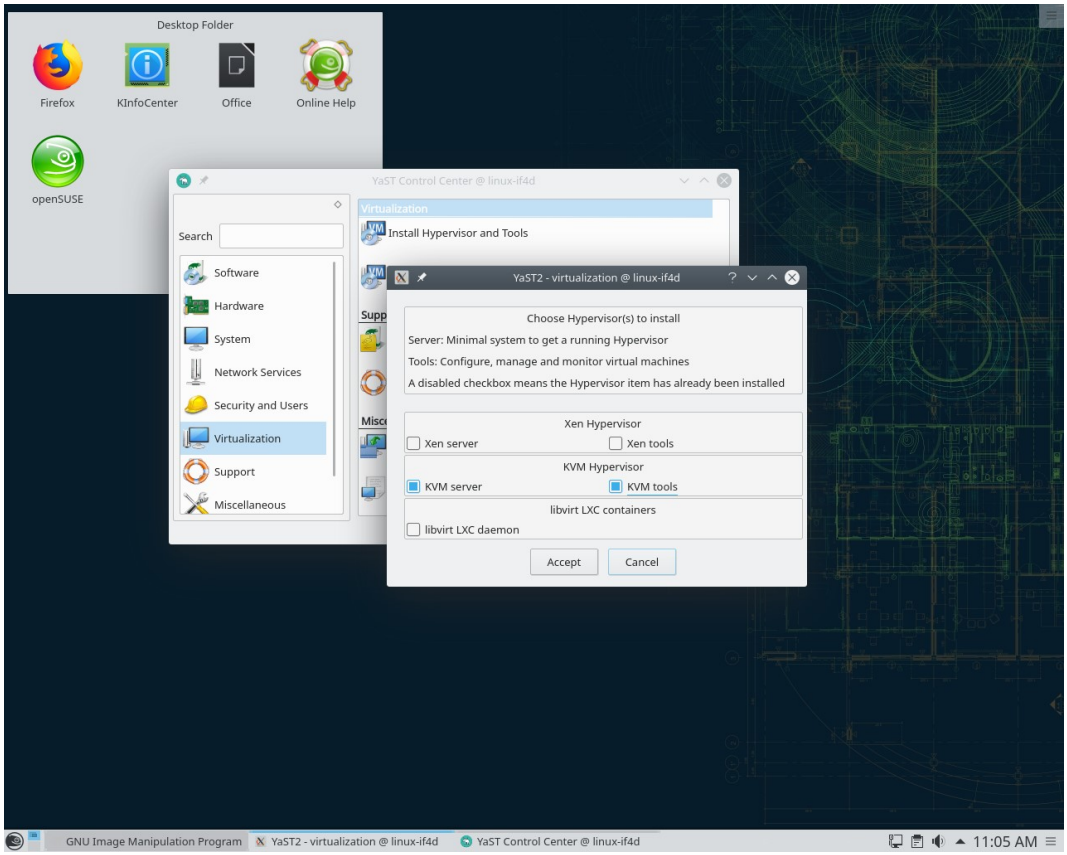
2. Install the **yast2-vm** software as shown in Figure 1 below.

   Figure 1. Installing yast2-vm



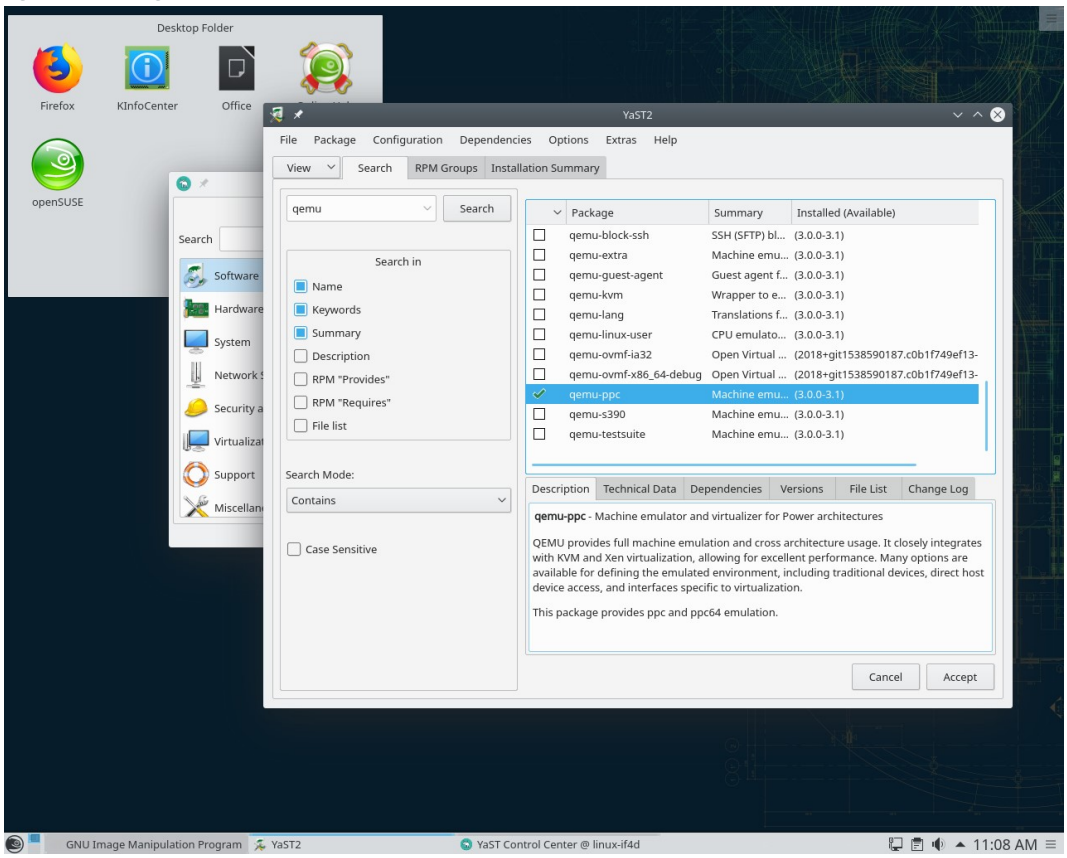
3. Exit YaST, open YaST again, and click **Virtualization**. Install the virtualization package for kernel-based virtual machine (KVM) and the KVM tools as shown in Figure 2 below.

Figure 2. Installing KVM server and tools



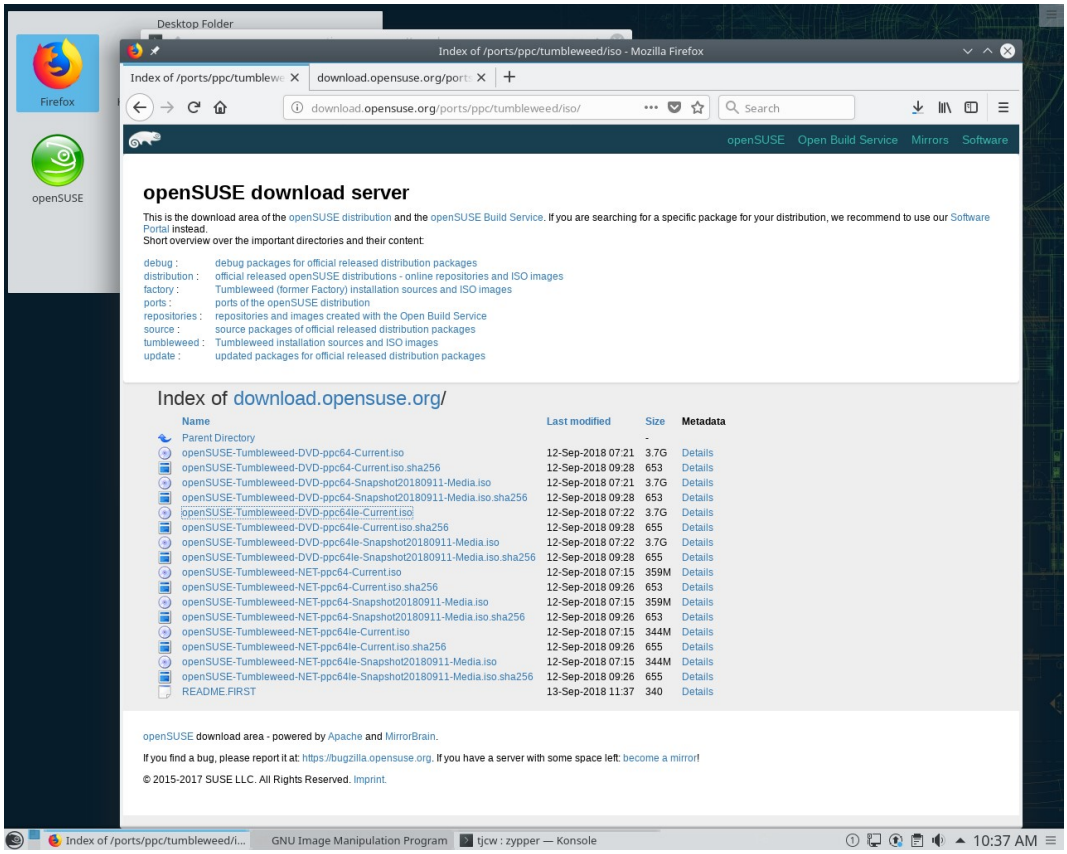
4. Click **Software** and install qemu-ppc as shown in Figure 3.

Figure 3. Installing qemu-ppc



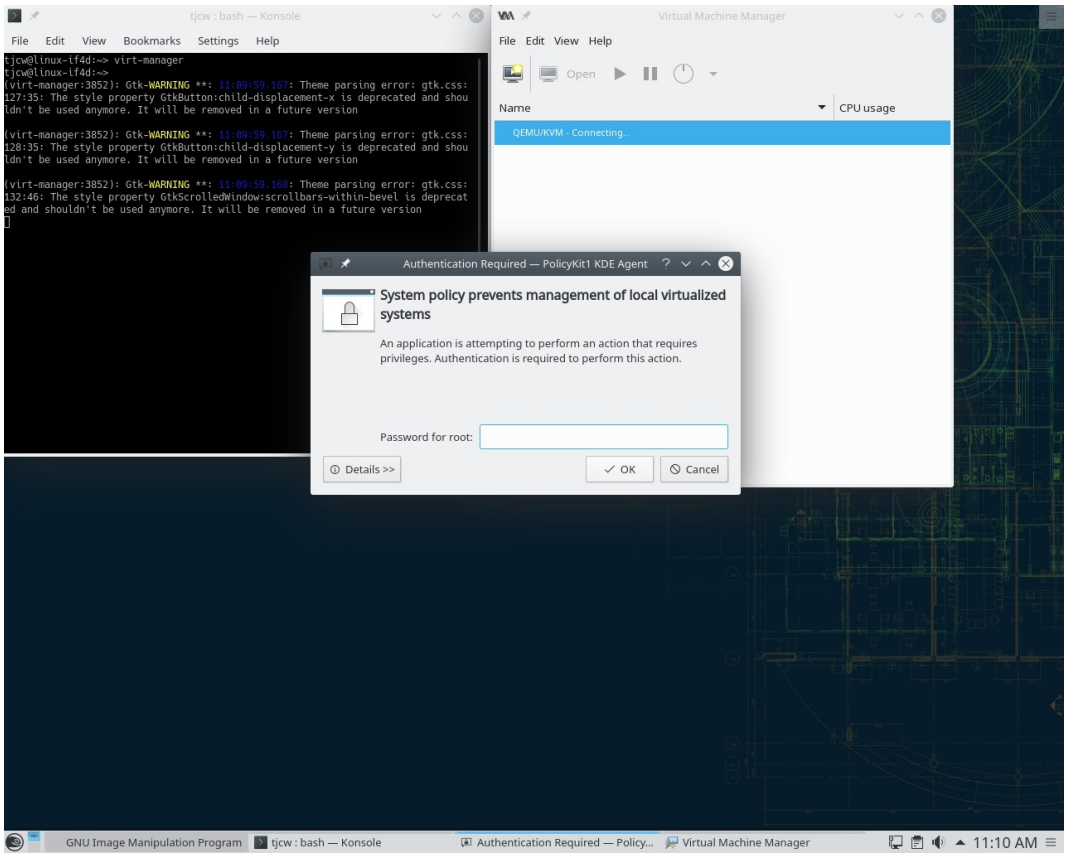
5. Download the POWER installation image described in the prerequisites as shown in Figure 4.

Figure 4. Downloading the POWER little endian installation image



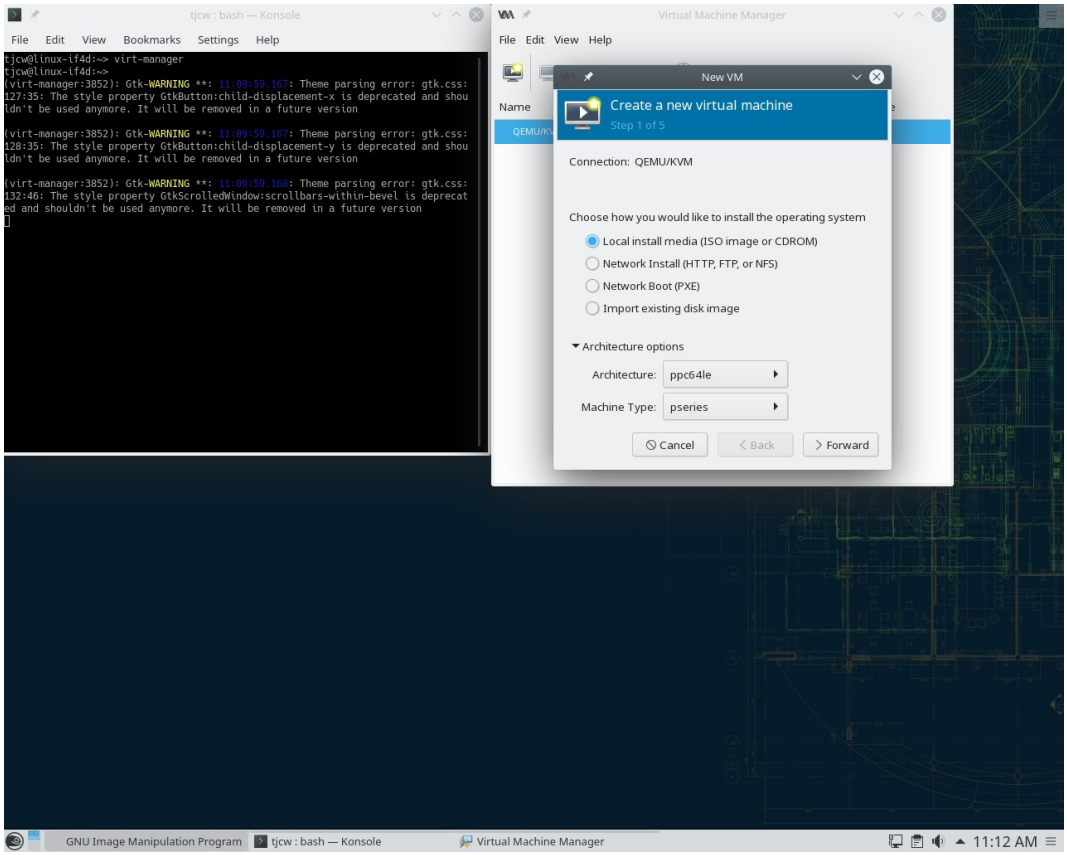
6. Open the console window and enter the `virt-manager` command. Then, connect to QEMU/KVM as shown in Figure 5.

Figure 5. Connecting to QEMU



7. Use the virtual machine manager to select a new POWER virtual machine. Set its virtual disk size to 30 GB and set its installation image to the POWER little endian ISO image you downloaded. Set the architecture option to **ppc64le**. I chose to allocate 1024 MB of memory and 1 CPU to the virtual machine. Indicate that you want a virtual network as shown in Figure 6.

Figure 6. Using the virtual machine (VM) manager to create a new VM



8. Install the virtual machine. Indicate that you want a **server** system role (this will set up the virtual machine without a graphical interface) and bring it up in runlevel 3. Use the expert partitioner based on the current proposal to select **Ext4** for the root file system; allow default sizes for all disk partitions. Indicate that you want a **ssh** server to run, and that the firewall should be opened for this.

Figure 7. Initial boot of the virtual POWER processor-based system

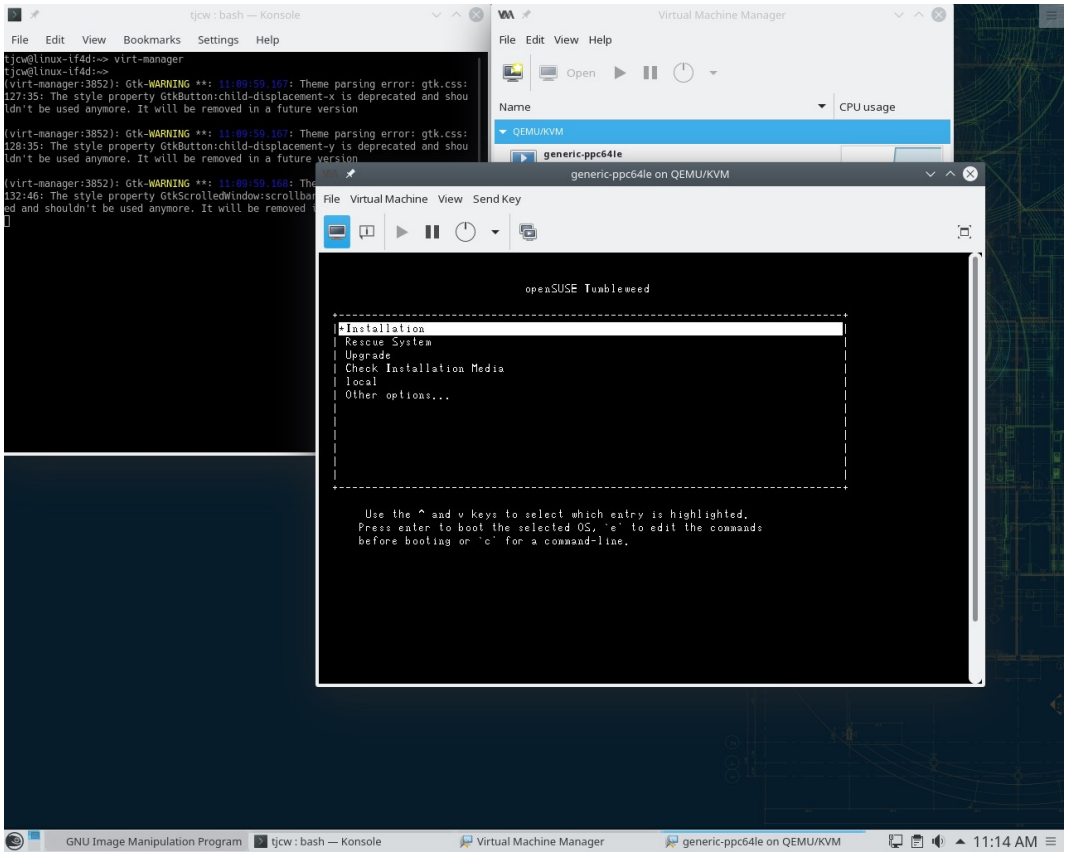


Figure 8. Starting configuration of the virtual POWER processor-based system

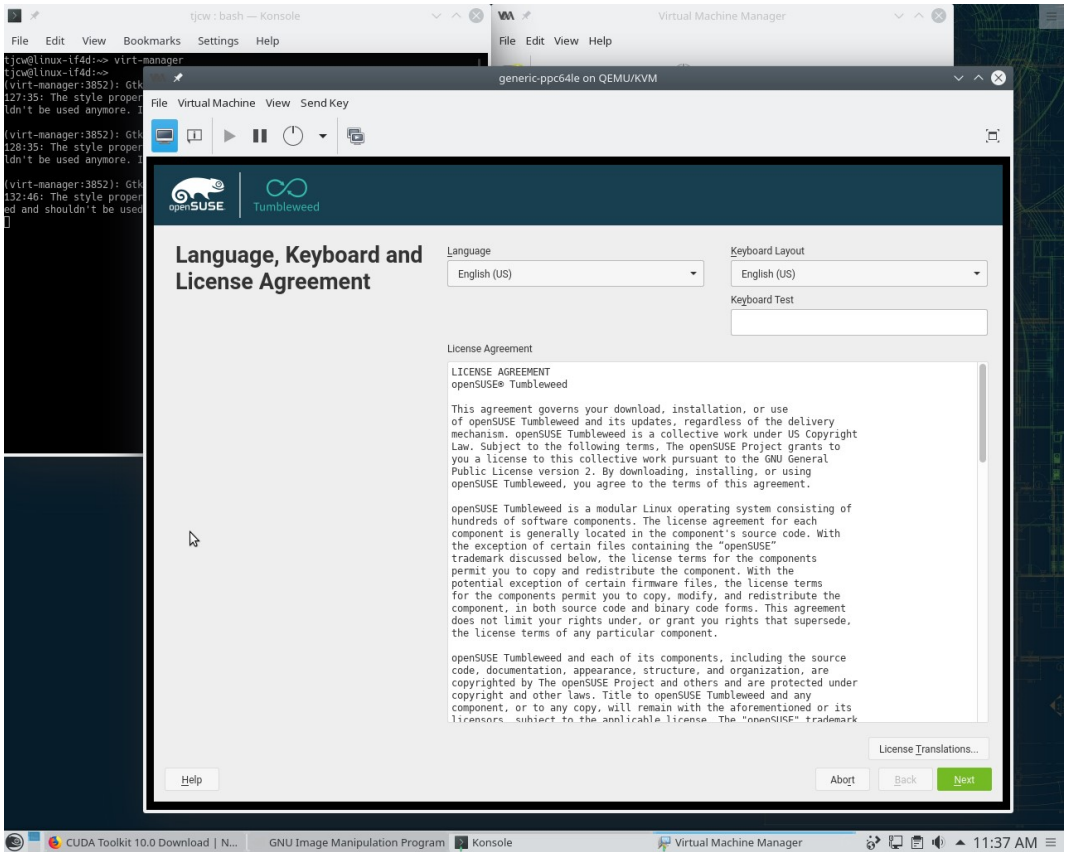
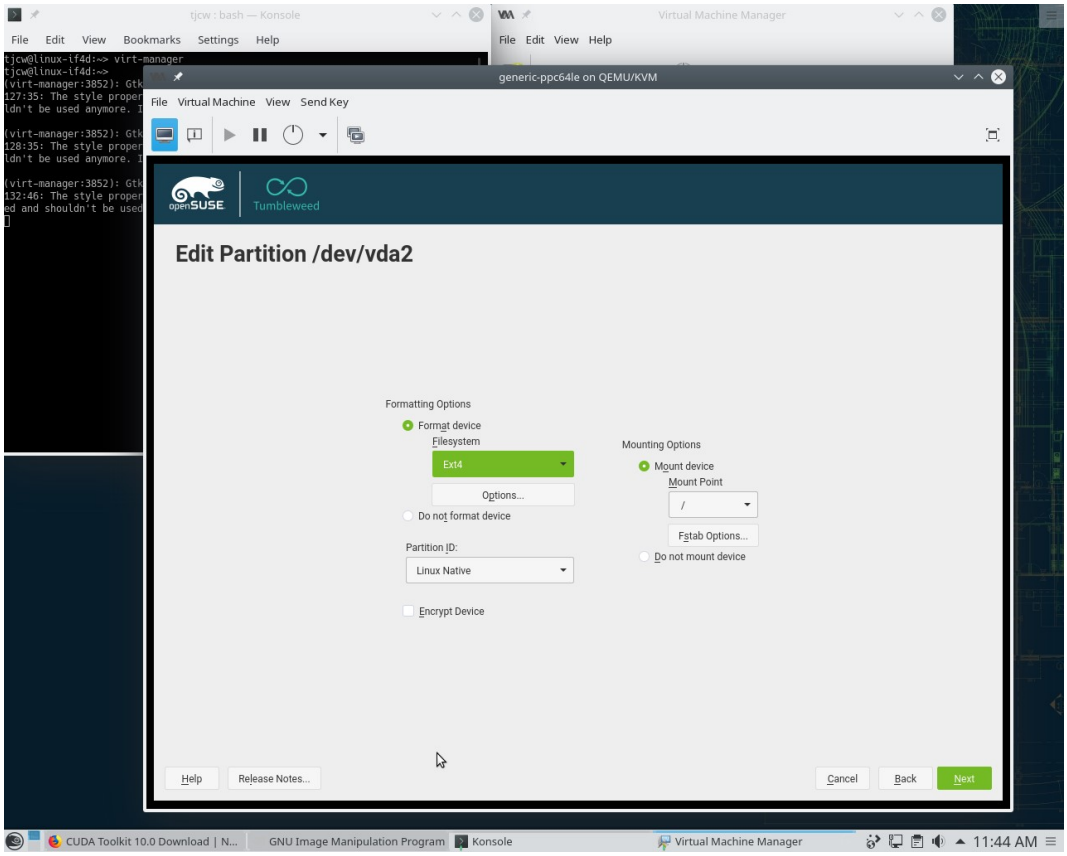


Figure 9. Selecting bext4' for the virtual root file system



Installing the virtual machine took about five hours on the x86 system that I was using; but it didn't require my attention, and so I got on with other work while the installation completed. When the installation using the ISO image is completed on the virtual machine, it will shut down. Start it again from the virt-manager window and let it boot. You now have a virtual POWER processor-based system running Linux.

9. Log on to the virtual machine console and use `ip addr` to find the IP address assigned to it as shown in Figure 10. From now on, you can use `ssh` on the host system to log on to the virtual machine instead of using its virtual console as shown in Figure 11; I prefer this as I can then resize the session window.

Figure 10. Finding the IP address of the virtual machine

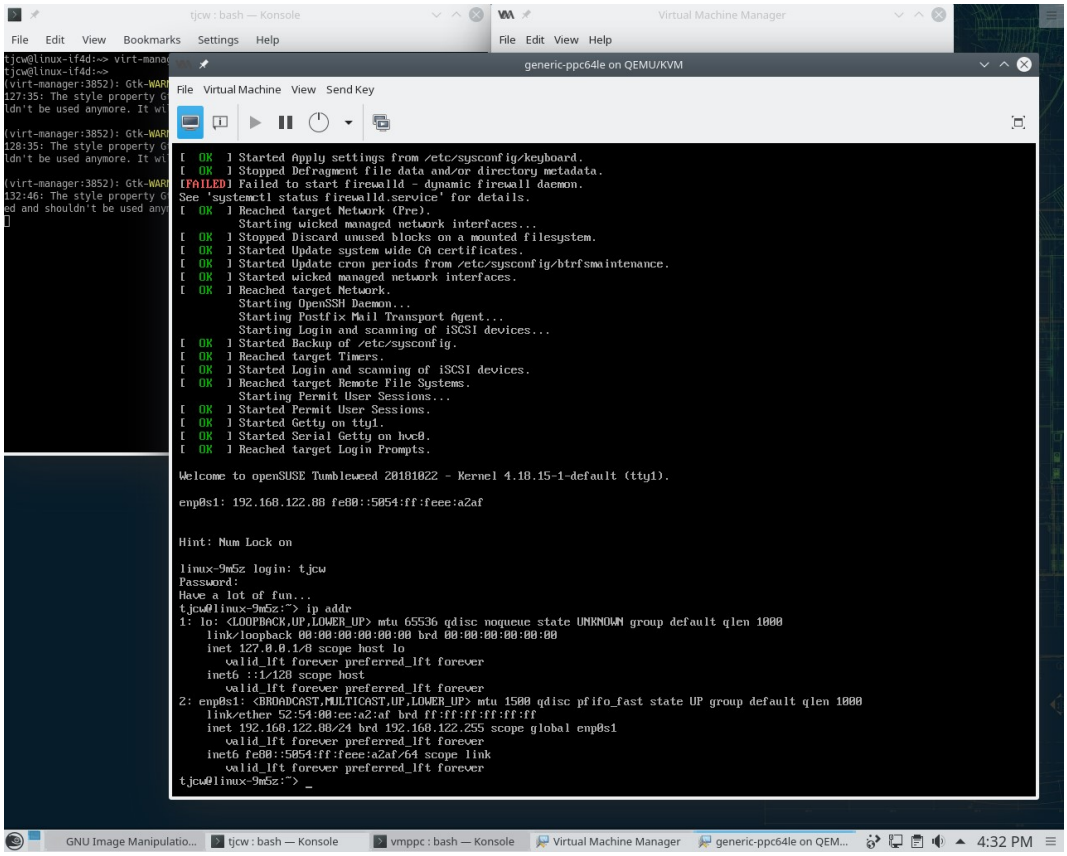
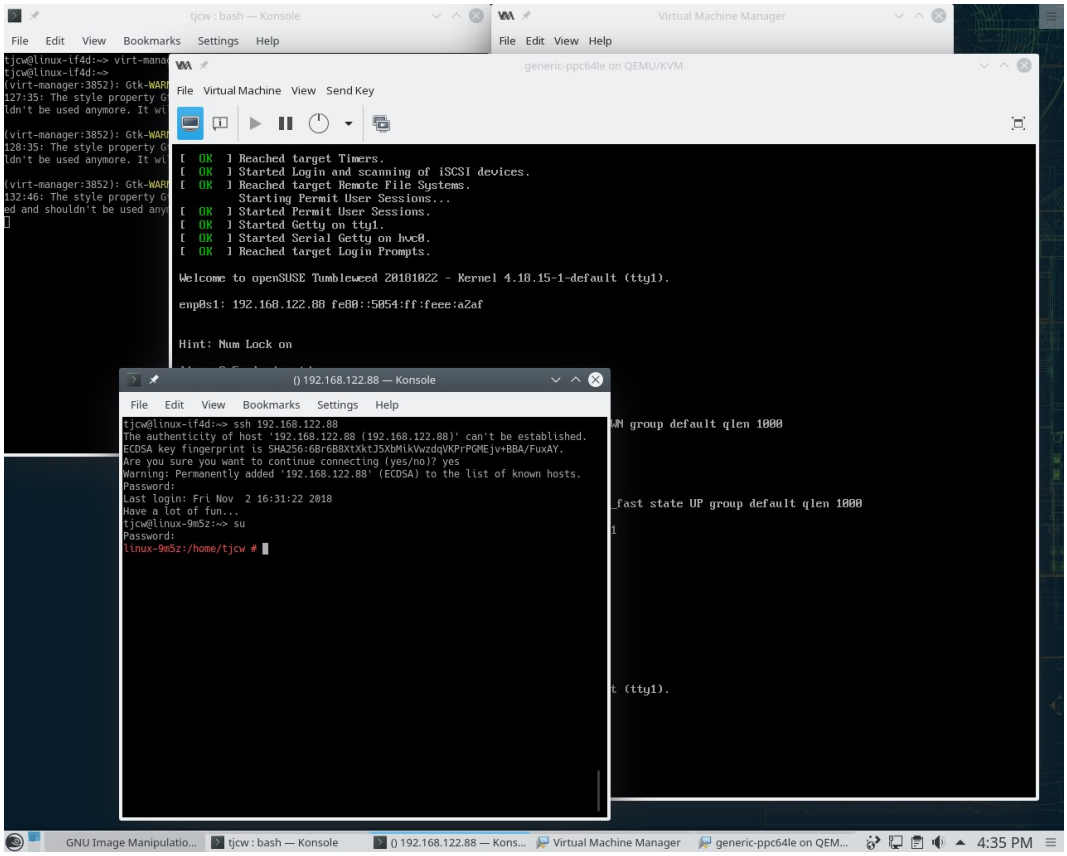
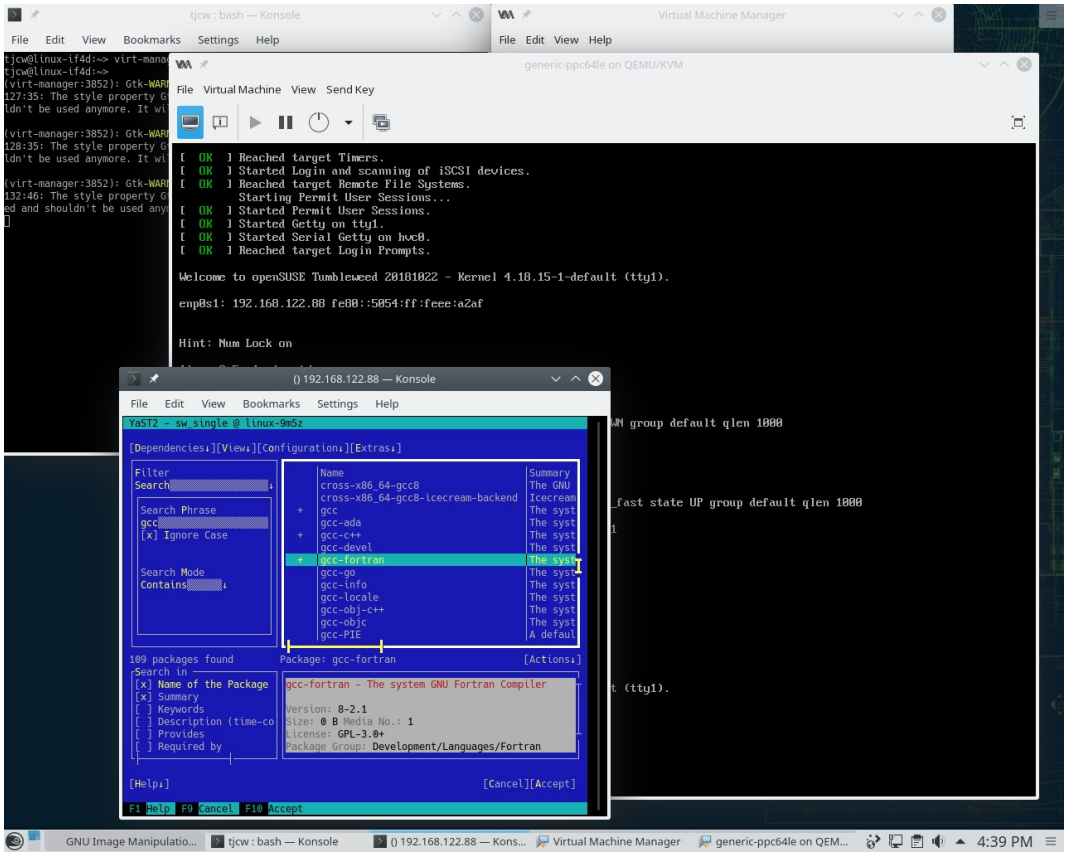


Figure 11. Logging in to the virtual machine with bssh'



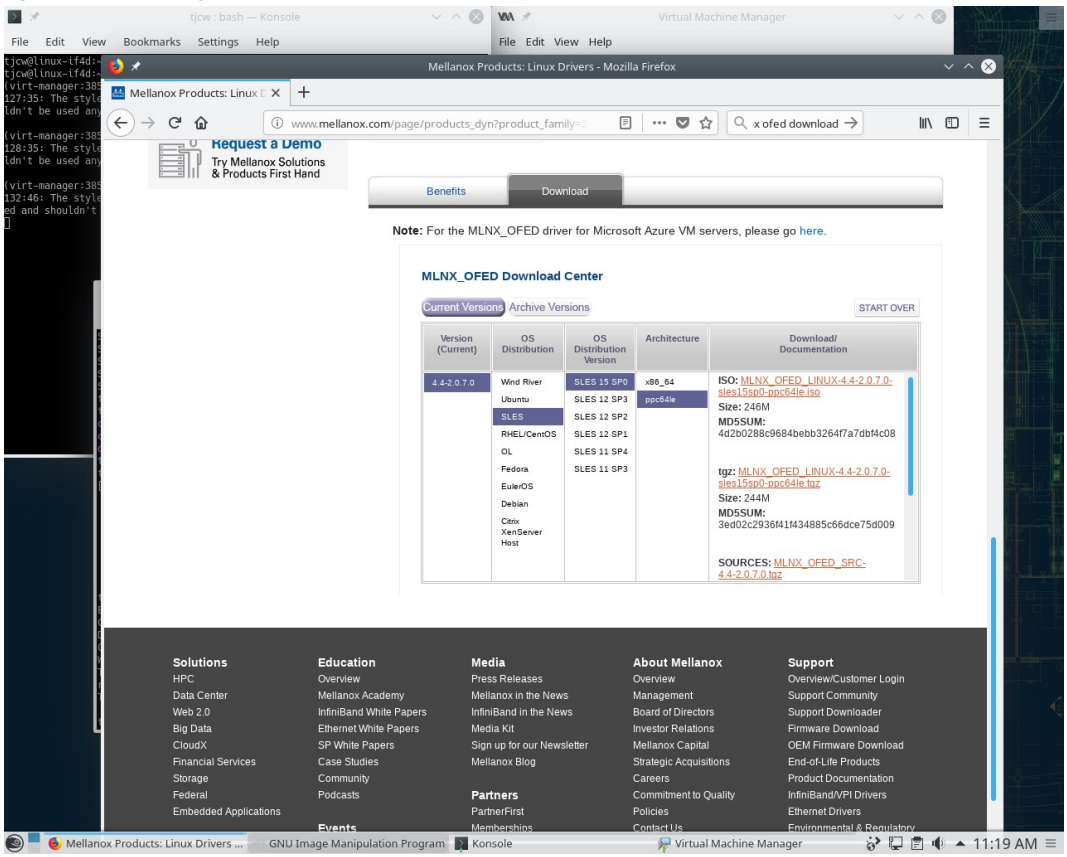
10. Log on to the virtual machine and start **YaST2**. Click **Software** and install gcc, gcc-c++, and gcc-fortran as shown in Figure 12. The virtual machine with server role doesn't have support for YaST2 to use X11, so YaST2 uses the **ncurses** interface; this uses the keyboard and function keys rather than the mouse to perform operations.

Figure 12. Installing gcc on the virtual machine



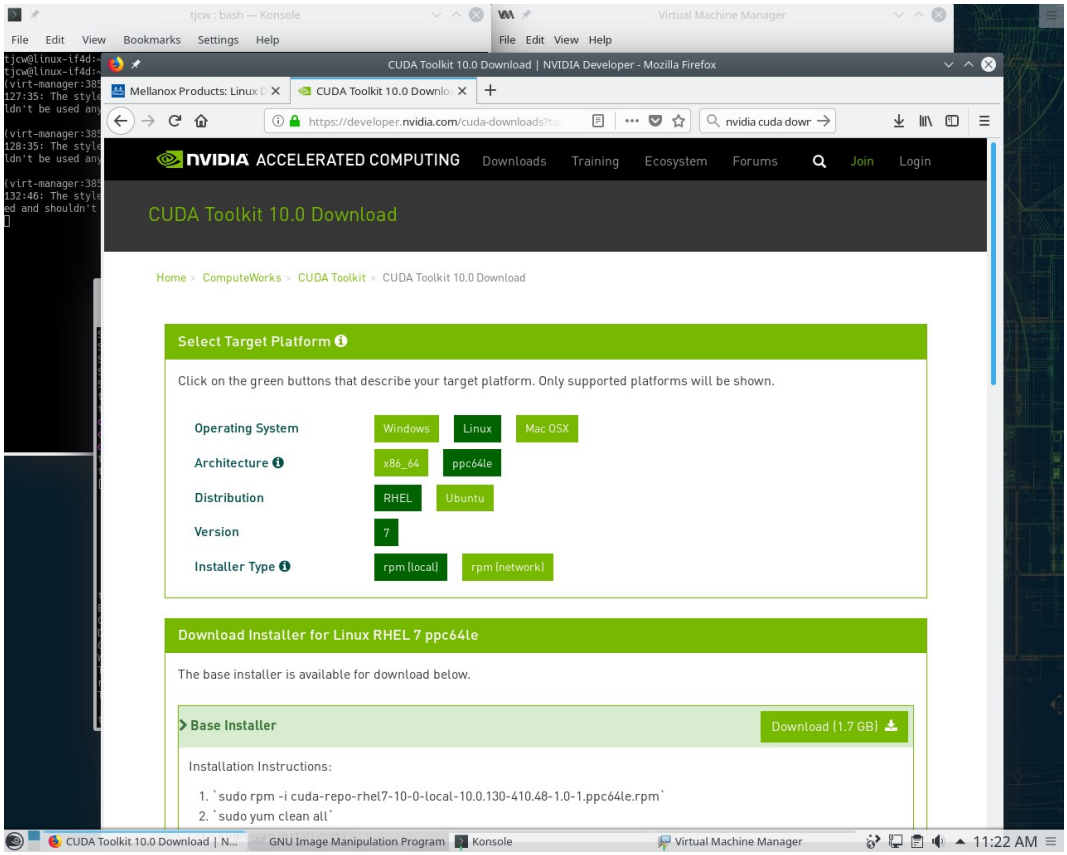
11. On the host system, download Mellanox OFED as listed in the prerequisites for POWER little endian SLES15 and shown in Figure 13 below.

Figure 13. Downloading Mellanox OFED for POWER



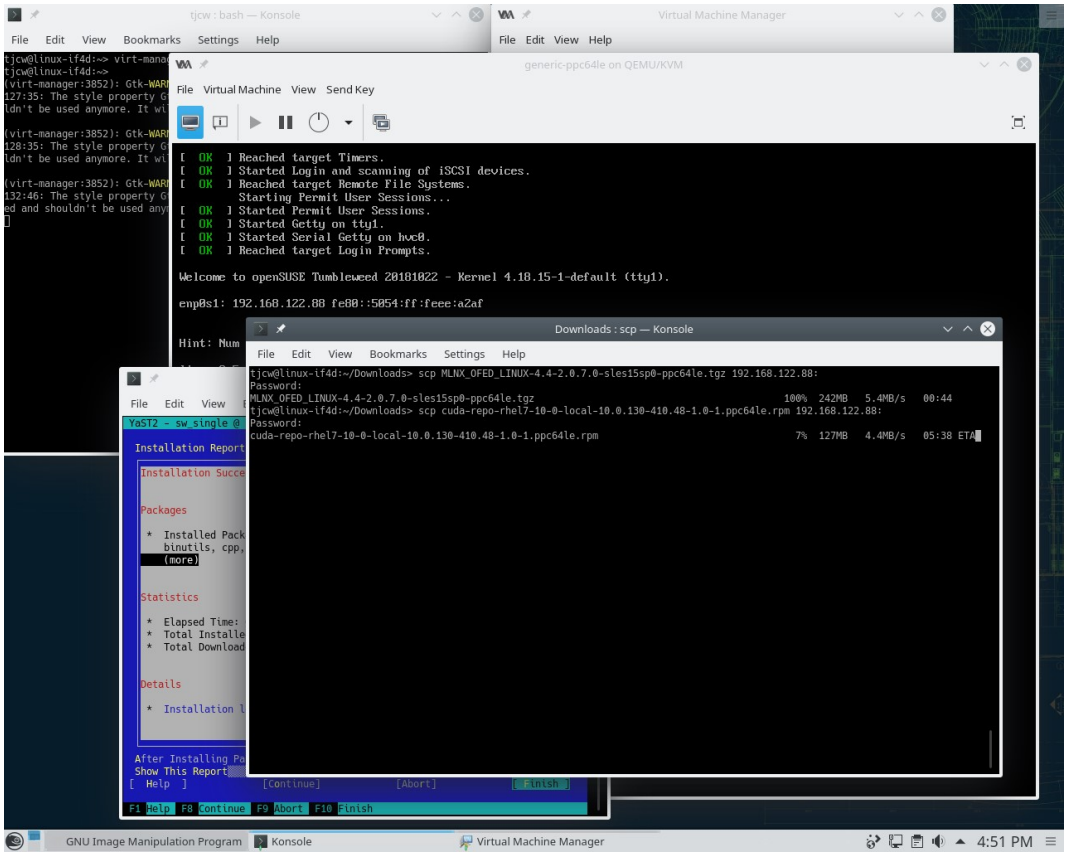
12. On the host system, download NVIDIA CUDA for POWER little endian RHEL7 as listed in the prerequisites and shown in Figure 14 below.

Figure 14. Downloading NVIDIA CUDA for POWER



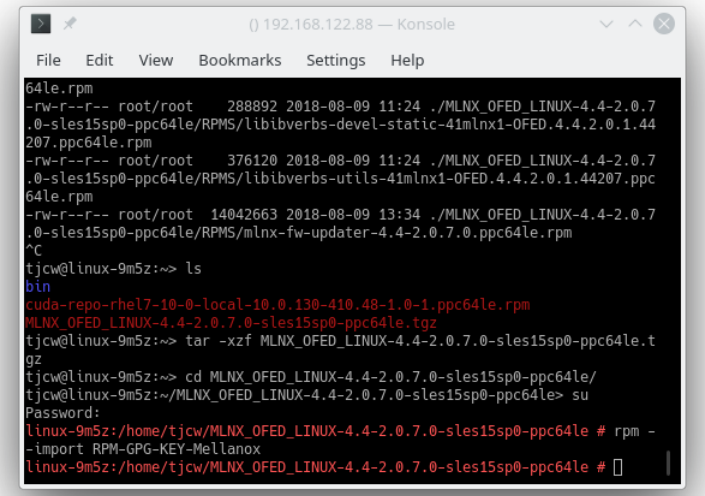
13. Use `scp` on the host system to copy the OFED and CUDA downloads to the virtual machine as shown in Figure 15.

Figure 15. Copying Mellanox OFED to the virtual machine



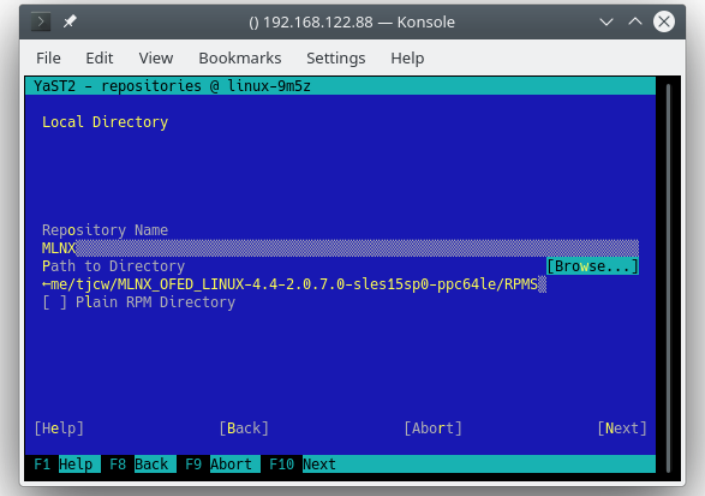
14. Unpack the OFED package on the virtual machine. Issue the `rpm -import ./RPM-GPG-KEY-Mellanox` command on the virtual machine to add the key for the OFED RPMs to the system and shown in Figure 16.

Figure 16. Importing the Mellanox OFED key on the virtual machine



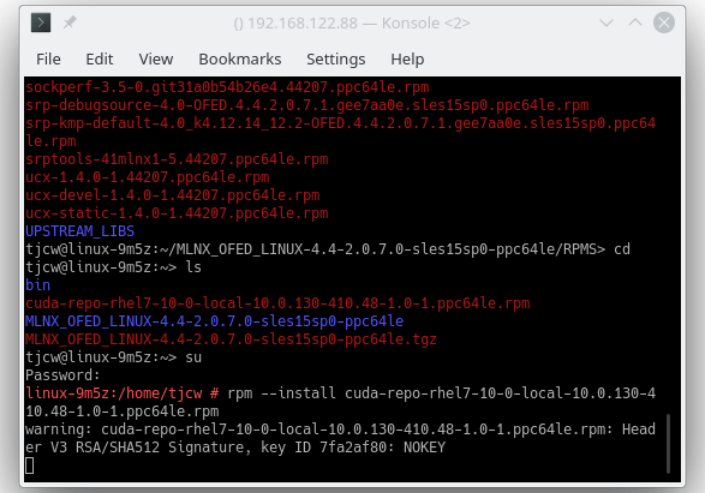
15. Go to YaST2 on the virtual machine, select **Software repositories**, and set up the unpacked OFED package as a repository as shown in Figure 17.

Figure 17. Setting up the Mellanox OFED repository



16. Install the CUDA RPM on the virtual machine using the `rpm -install ./cuda-repo-rhel7*` command as shown in Figure 18. This sets up files under /var/cuda-repo*.

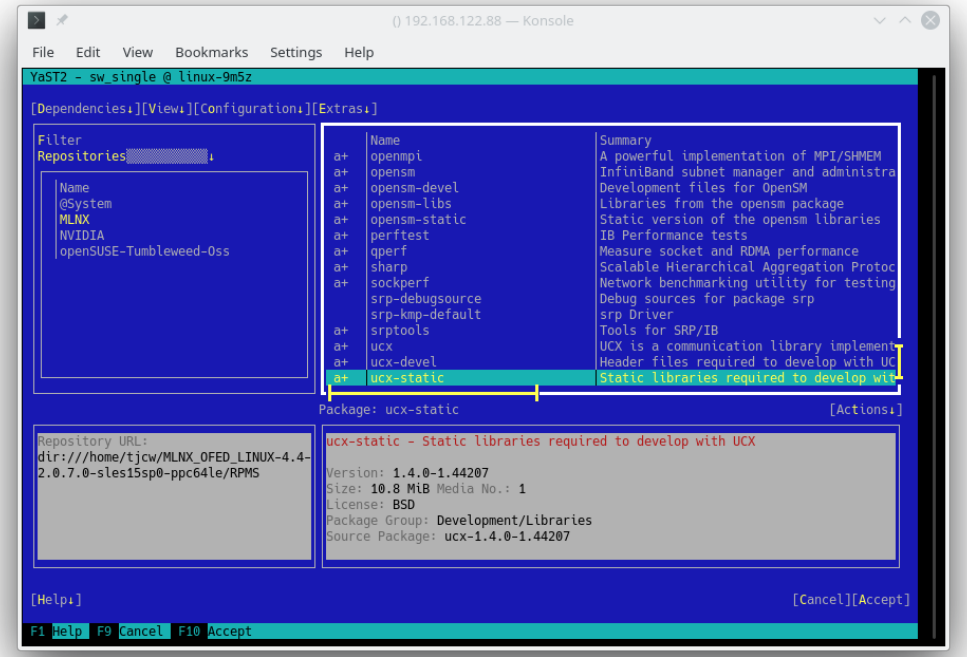
Figure 18. Installing the NVIDIA CUDA RPM on the virtual machine



17. Use YaST2 to set these up as a repository, similar to setting up the OFED repository.

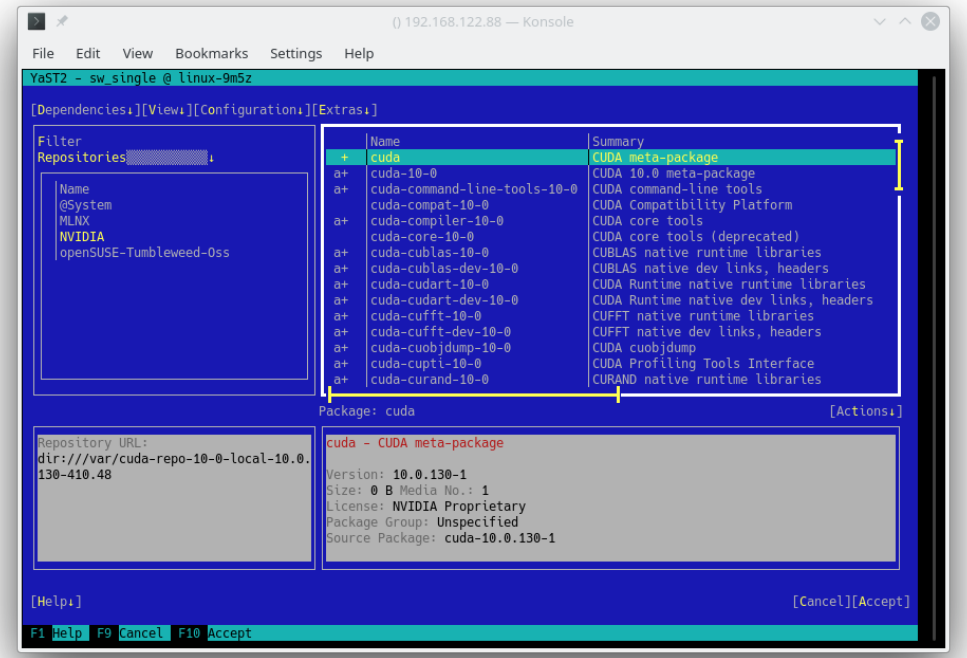
18. Install the OFED RPMs that you need from the OFED repository as shown in Figure 19.

Figure 19. Installing the Mellanox OFED RPMs on the virtual machine



19. Install the CUDA packages that you need from the CUDA repository to make a build machine as shown in Figure 20.

Figure 20. Installing the NVIDIA CUDA RPMs on the virtual machine



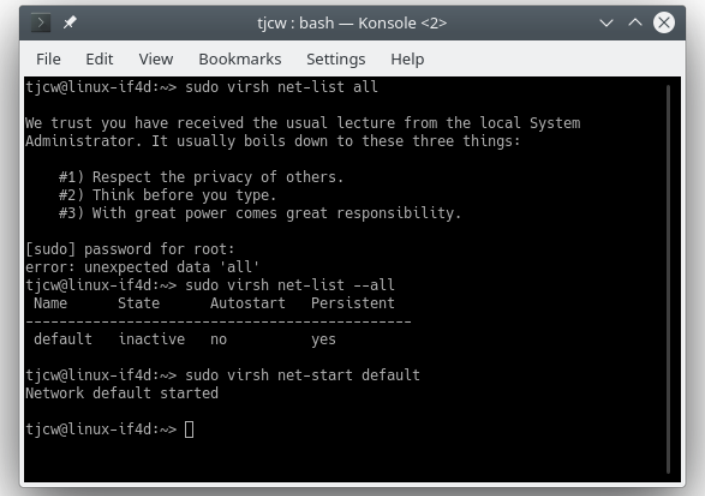
Notes

NVIDIA GPUs and Mellanox InfiniBand adapters are supported on little endian only. So, this tutorial shows how to set up a POWER little endian system. If you need a POWER big endian system, download the big endian DVD from

http://download.opensuse.org/ports/ppc/factory/iso/openSUSE-Tumbleweed-DVD-ppc64-Current.iso and select architecture ppc64 in step 7.

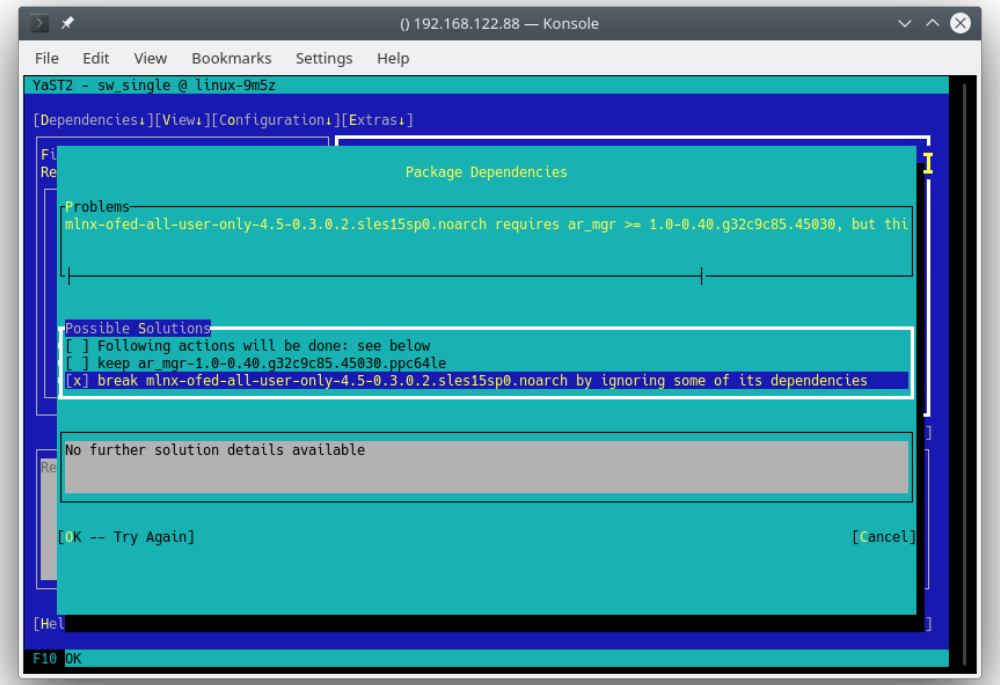
If you have to restart the virtual machine, you may get an error message about a virtual network not running. This can be fixed using the `sudo virsh net-start default` command.

Figure 21. Starting the network between the host system and the virtual machine



You may get warning messages about file mismatches during installation of OFED or CUDA. These are because you are using the free openSUSE Tumbleweed Linux distribution for POWER, rather than the paid-for SLES15 Linux distribution for POWER. The warnings look as shown in the Figure 22. They should all be answered with the **break** option.

Figure 22. Accepting that mismatches may break the installation



It is possible to use other Linux distributions such as Fedora Rawhide on the x86 system, or even to use Microsoft® Windows® if you install QEMU for Windows from https://qemu.weilnetz.de/w64/. I have written this tutorial using openSUSE Tumbleweed because it is conveniently packaged for the job.

Summary

Now you have a fully-installed POWER virtual machine, ready to build and package your software. Note that it will run much slower than a non-emulated system, and it has no InfiniBand adapter or GPU, but it is fully functional and should be adequate for building and packaging software.

There are many other ways of getting access to POWER servers; some are:

- Free cloud resources for developers: https://developer.ibm.com/linuxonpower/cloud-resources/
- Buy from: https://www.ibm.com/it-infrastructure/power
- Rent from an IBM Partner
- Access from IBM Cloud: https://www.ibm.com/cloud/bare-metal-servers/power

SOCIAL

CONTENTS

Introduction
Prerequisites
Estimated time
Steps
Notes
Summary

RESOURCES

OpenFabrics Alliance
OpenFabrics Alliance are the hardware vendors behind the OFED software package.

NVIDIA's developer website
More information about accelerated computing using GPUs.

rCUDA
rCUDA from the University of Valencia, Spain is the original motivation for this tutorial.

Related content

TUTORIAL | NOV 01, 2018

Optimized libraries for Linux on Power

Linux Systems

TUTORIAL | OCT 09, 2018

Live patching the Linux kernel

Linux Systems

ARTICLE | OCT 09, 2018

Enhancing QEMU virtio-scsi with Block Limits vital product data (VPD) emulation

Databases Infrastructure +