Welcome to Linux Foundation Virtual Training

by The Linux Foundation

09/19/2017

Version 6.30

Connection information for class

You will receive an email containing the specific login information for your screenshare and phone bridge, as well as exact dates and times, no later than the week before the class is scheduled.

Specific instructions for your course will be addressed in the Appendix.

After reading this document, please examine the FAQ and see if any remaining questions are answered by the assemblage of FAQs in that location:

`http://bit.ly/LF-FAQ`

or

http://training.linuxfoundation.org/linux-courses/general-information-and-faq

Contents

1 Hardware Requirements

Students are expected to **provide their own computers** for **Linux Foundation** courses. The following instructions should make clear the specifics that apply to you.

All courses have slightly different HW requirements. Specific HW requirements for your class can be found in the Appendix.

The Linux Foundation logistical staff may be consulted as required for further clarification.

1.1 Using a Virtual Machine Instead

Virtual Machines

If you elect to use a Virtual Machine (instead of native Linux) bear in mind that the hardware requirements double, since you now need enough CPU/RAM for the host operating system as well as the guest OS.

Using a VM for this course can make things faster/easier; if you make a fatal mistake, a simple reboot of the VM will restore things to normal.

More on what distro and software needs to be installed on the VM can be found in the Software Requirements chapter below.

If you want to build your own VM image

You can make sure your own Virtual Machine image is properly setup for the class using the ready-for.sh script which can be found as follows:
`http://bit.ly/LFprep`
or
https://training.linuxfoundation.org/cm/prep/

1.2 Pre-Built Virtual Machine Images

We provide pre-built **virtual machine images** that work with **VMware** products (e.g. **Workstation, VMplayer, VMFusion**) or **Oracle Virtual Box**. They can also be converted to work on **Linux** hosts using **KVM** as described in accompanying documentation.

Where are the prebuilt VMs?

These VMs can be found at: `http://bit.ly/LF-vm`
or
https://training.linuxfoundation.org/cm/VIRTUAL_MACHINE_IMAGES/
where you should log in with these credentials:

- **username:** `LFtraining`
- **password:** `Penguin2014`

The `000README` file in that directory contains deployment instructions and other considerations.

All the prebuilt Virtual Machine images have been setup for common classes using the aformentioned ready-for.sh script. **However, you may still want to run ready-for.sh again on the VM for your specific course to make sure your VM guest configuration is correct**.

2 Networking, Screenshare and Audio or Telephone Requirements

2.1 Internet

Reliable broadband connectivity is required

A minimum of 100 kb/s is required. This is used for the screensharing software.

2.2 Screenshare and Audio

Screenshare and Audio

We use **BlueJeans** for for both the screen share and the audio.

If you are joining the virtual class from a corporate network using proxy services, please see the Appendix for complete relevant instructions.

3 Software Requirements

You can use either a native **Linux** installation of any **recent** major distribution, or you can use a **virtual machine image** running under a **hypervisor**; either you can build your own or you can use one provided by the **Linux Foundation**.

- There are some courses that do not absolutely require a **Linux** installation, such as **LFS452**. You may want to consult the course-specific requirements first before doing a full **Linux** installation.

3.1 Developer Courses

A normal x86_64 installation of any major **recent** distribution (such as **Red Hat Enterprise Linux, Ubuntu, SUSE Enterprise Linux, openSUSE, CentOS, Fedora, Debian,** or **Mint**) will give you almost all necessary tools, and the instructor and course manual can provide guidance on missing ingredients if necessary. All courses require **root** access (administrator or superuser) either through a **root account** or **sudo** privilege.

Please beware that we cannot be responsible if your system winds up getting damaged. This warning is particularly important for kernel-level courses such as **LFD420**, **LFD430**, **LFD435**, **LFD440**, where you will be compiling and installing kernels and/or kernel modules.

Operating system damage, while rare, is possible. You may wish to do a fresh installation of a 64-bit **Linux** Distribution, perhaps on a fresh partition.

Linux Foundation provides pre-built virtual machine images (found above) which can be used for most courses, but not for all hardware intensive courses, such as **LFD435**, **LFD450** and **LFD460**. In this case memory and processor requirements tend to be more robust.

Please Note:

There are course-specific requirements that may supersede these general requirements; please see the course-specific section in the Appendix.

3.2 System Administration Courses

System Administration courses are written for enterprise distributions such as **RHEL/CentOS**, **Debian/Ubuntu** and **SLES/OpenSUSE**. A native or virtual installation of any of the two most recent releases of these **Linux** distributions is recommended. All courses require **root** (administrator or superuser) access either through a **root account** or **sudo** privilege.

Please Note:

There are course-specific requirements that may supersede these general requirements; please see the course-specific section in the Appendix.

3.3 Checking Your Hardware and Software Setup with ready-for.sh

Before you continue...

Get, and run, the online tool at the following URL which will automate checking the course-specific hardware and software requirements on your computer.
`http://bit.ly/LFprep`
or
https://training.linuxfoundation.org/cm/prep/

The Linux Foundation has provided a **bash** script which can be downloaded from the aformentioned webpage. This script is meant to be run on an installed computer to see if it is up to standards and has the necessary packaged installed and hardware for the course.

```
$ wget http://bit.ly/LFready -O ready-for.sh
```

Once you have downloaded the `ready-for.sh` script you can make it executable and run it as in:

```
$ chmod 755 ready-for.sh
$ ./ready-for.sh LFD420
$ ./ready-for.sh --install LFD420
```

(You should substitute the name of your course for LFD420.)

This script will check all course requirements, optionally install packages required for the course (the `--install` step above) and then optionally download `RESOURCES`, `SOLUTIONS`, and extra tarballs you will need for class. Please run those steps before class (somewhere with good Internet).

You probably want to download resources before class starts

For the **Linux Security** course (**LFS416**) there is a lot to download; doing this before class is essential to avoid delays due to possibly limited classroom bandwidth. The ready-for.sh script will download these files as a part of making sure your computer is ready for class.

Because **Linux** distributions are constantly being updated, the script is also always being updated and may not have all details filled in for all courses.

For More Information

For a more detailed explanation of all the possible methods of installation, please examine the Appendix or view it online at `http://bit.ly/LFinstall`
or
http://training.linuxfoundation.org/linux-courses/general-information-and-faq/on-site-linux-training-facility-requirements?id=780

Appendices

A Course-Specific Hardware and/or Software Requirements

A.1 LFD301: Introduction to Linux for Developers and GIT

Table 1: **Introduction to Linux for Developers and GIT**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	5 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-24+, LinuxMint-18+, openSUSE-42.1+, RHEL-7+, Ubuntu-16.04+, SLES-12+,

A.2 LFD401: Developing Applications for Linux

Table 2: **Developing Applications for Linux**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	5 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-24+, LinuxMint-18+, openSUSE-42.1+, RHEL-7+, Ubuntu-16.04+, SLES-12+,

A.3 LFD420: Linux Kernel Internals and Debugging

Table 3: **Linux Kernel Internals and Debugging**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	9 GiB
Free Disk Space in /boot	128 MiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-24+, LinuxMint-18+, openSUSE-42.1+, RHEL-7+, Ubuntu-16.04+, SLES-12+,

A.4 LFD430: Developing Linux Device Drivers

Table 4: **Developing Linux Device Drivers**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	9 GiB
Free Disk Space in /boot	128 MiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-24+, LinuxMint-18+, openSUSE-42.1+, RHEL-7+, Ubuntu-16.04+, SLES-12+,

A.5 LFD440: Linux Kernel Debugging and Security

Table 5: **Linux Kernel Debugging and Security**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	9 GiB
Free Disk Space in /boot	128 MiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-24+, LinuxMint-18+, openSUSE-42.1+, RHEL-7+, Ubuntu-16.04+, SLES-12+,

A.6 LFD461: KVM for Developers

Table 6: **KVM for Developers**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	5 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-24+, LinuxMint-18+, openSUSE-42.1+, RHEL-7+, Ubuntu-16.04+, SLES-12+,

A.7 LFS300: Fundamentals of Linux

Table 7: **Fundamentals of Linux**

OS required for class	Linux
Virtual Machine	Acceptable

Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	5 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-23+, LinuxMint-17+, openSUSE-42.1+, RHEL-7+, Ubuntu-14.04, Ubuntu-16.04+, SLES-12+,

A.8 LFS301: Linux System Administration

Table 8: **Linux System Administration**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	5 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-23+, LinuxMint-17+, openSUSE-42.1+, RHEL-7+, Ubuntu-14.04, Ubuntu-16.04+, SLES-12+,

A.9 LFS311: Advanced Linux System Administration and Networking

Table 9: **Advanced Linux System Administration and Networking**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	5 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-23+, LinuxMint-17+, openSUSE-42.1+, RHEL-7+, Ubuntu-14.04, Ubuntu-16.04+, SLES-12+,

A.10 LFS416: Linux Security

Table 10: **Linux Security**

Internet Access	Required
OS required for class	Linux
Virtual Machine	This course can't be run on a VM; you will be running VMs under a Linux host
Native Linux	Required
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 4)
Minimum CPU Performance	20000 bogomips
Required CPU features	vmx

Minimum Amount of RAM	8 GiB
Free Disk Space in $HOME	30 GiB
Kernel Configuration Options	HAVE_KVM KSM
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-23+, LinuxMint-17+, openSUSE-42.1+, RHEL-7+, Ubuntu-14.04, Ubuntu-16.04+, SLES-12+,

This course requires use of a hypervisor to run **Linux Foundation** supplied virtual machines. It is easiest to use any **VMWare** variant or **Oracle Virtual Box**. With some format translation other hypervisors can be used including **KVM** or **QEMU** or **AZURE**.

You Must Run Linux Natively

Due to use of hypervisor, one should run this on a native **Linux** machine rather than on a virtual machine. Nested virtualization is hard to set up and performance is much weaker.

A.11 LFS422: High Availability Linux Architecture

Table 11: **High Availability Linux Architecture**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Acceptable (with nested virtualization)
Required CPU Architecture	x86_64
Preferred Number of CPUs	4 (minimum 2)
Minimum CPU Performance	2000 bogomips
Required CPU features	vmx
Minimum Amount of RAM	4 GiB
Free Disk Space in $HOME	40 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS:amd64-7+, DebianUbuntu:amd64-14.04+LTS+,

Hardware virtualization capabilities support must be present and enabled; this is usually a **BIOS** setting that must be switched to On or Enabled. At the kernel level, the KVM module in use needs to support nested virtualization and have it enabled on module load.

```
$ modinfo kvm_intel | grep nested
parm:           nested:bool
$ cat /sys/module/kvm_intel/parameters/nested
Y
```

Typically the `kvm_intel` module has nesting support disabled by default. Try setting it to `kvm_intel.nested=1` on the kernel command line in the bootloader.

Be sure any firewalls or mandatory access control mechanisms like **AppArmor** and **SELinux** are disabled or in permissive mode.

The distributions packages for **KVM** and **Qemu** as well as **libvirt** and **virt-manager** must be installed.

A.12 LFS426: Linux Performance Tuning

Table 12: **Linux Performance Tuning**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	4 (minimum 2)
Minimum CPU Performance	20000 bogomips
Minimum Amount of RAM	2 GiB
Free Disk Space in $HOME	5 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-23+, LinuxMint-17+, openSUSE-42.1+, RHEL-7+, Ubuntu-14.04, Ubuntu-16.04+, SLES-12+,

This course runs best using a bare metal system with a fresh install of **Ubuntu 16.04**. The lab exercises include the appropriate commands and instructions for other recent **Linux** distributions, but **Ubuntu 16.04** has undergone the most testing.

While the use of a virtual machine (VM) is possible, it is not recommended as many performance measurements will lack meaning. Some labs will not be possible without a bare metal machine.

An AWS node will be made available for those without a bare metal or a usable VM. **PuTTY** and a web browser will be necessary to access the AWS VM.

A.13 LFS430: Linux Enterprise Automation

Table 13: **Linux Enterprise Automation**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	5 GiB
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-23+, LinuxMint-17+, openSUSE-42.1+, RHEL-7+, Ubuntu-14.04, Ubuntu-16.04+, SLES-12+,

A.14 LFS452: Essentials of OpenStack Administration

Table 14: **Essentials of OpenStack Administration**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser, terminal emulation program (ssh or putty)

Students must provide their own computers for this class capable of connecting to the online lab environment. Lab systems will be assigned during class. You will need a web browser and a terminal emulation program to access them.

A.15 LFS462: Introduction to Linux KVM Virtualization

Table 15: **Introduction to Linux KVM Virtualization**

OS required for class	Linux
Virtual Machine	This course can't be run on a VM
Native Linux	Required
Required CPU Architecture	x86_64
Preferred Number of CPUs	4 (minimum 2)
Minimum CPU Performance	10000 bogomips
Required CPU features	vmx
Minimum Amount of RAM	4 GiB
Free Disk Space in $HOME	40 GiB
Kernel Configuration Options	HAVE_KVM KSM
Distro Architecture	x86_64
Supported Linux Distros	CentOS-7+, Debian-8+, Fedora-23+, LinuxMint-17+, openSUSE-42.1+, RHEL-7+, Ubuntu-14.04, Ubuntu-16.04+, SLES-12+,

The distributions packages for **KVM** and **Qemu** as well as **libvirt** and **virt-manager** must be installed.

Hardware virtualization capabilities support must be present and enabled; this is usually a **BIOS** setting that must be switched to On or Enabled.

A.16 LFS465: Software Defined Networking with OpenDaylight

Table 16: **Software Defined Networking with OpenDaylight**

OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	4 (minimum 2)
Minimum CPU Performance	10000 bogomips
Minimum Amount of RAM	4 GiB
Free Disk Space in $HOME	20 GiB
Kernel Configuration Options	OPENVSWITCH
Distro Architecture	x86_64
Supported Linux Distros	Ubuntu:amd64-16.04+,

The kernel version must be at least 3.4 or more recent. Anything other than running **Ubuntu 16.04** will present software packaging problems and is not recommended or supported, even though things will work in other situations.

B More Details on Installing Linux

B.1 Installing Virtual Machine Images run under a Hypervisor

We can provide pre-built virtual machine images that work with **VMware** hypervisors, **Oracle Virtual Box**, or **KVM**. The host machine can be running any operating system with an available hypervisor, including all flavors of **Windows**, **Linux** and **Mac OS**.

Once you have the hypervisor installed, the actual installation time for a virtual machine is basically zero since all you have to do is attach our image file to it. These pre-built images already contain all the needed software and for the kernel-level courses, also conveniently contain a copy of the **Linux** kernel source git repository. The virtual machine images are updated with each new kernel release, which occurs every three months or so.

An advantage of using the virtual machine images is that you can't fundamentally destroy your system while running them, and they run as an unprivileged application and will get you into less trouble with IT staff if that is an issue. A further advantage, especially with on-line classes, is that a system failure does not take you off-line from the virtual class.

The disadvantages have mostly to do with performance and requiring somewhat more memory and CPU power. However, in most (but not all) courses this is not a disqualifying aspect.

Upon enrollment in a class we can make these virtual machine images available to you. (We do not make them available to the general public as they are quite large (2+ GB even in compressed form) and we do not have the dedicated bandwidth to support widespread downloading.)

B.2 Performing a Native Linux Installation

Virtually all popular **Linux** distributions have straightforward installation instructions these days, and most provide a **live CD** or **USB** stick which can also be used to do an install. One first boots off the Live media; a successful boot verifies that the **Linux** distribution is out-of-the-box compatible with your hardware, and you can then click on install to place the Linux distribution on your hard disk. (Using **Wubi** to install **Ubuntu** from within **Windows** does not count as a native installation. Performance is worse than using a virtual machine as discussed above and we do not support this option.

In order to proceed with installation, you generally need enough available space on the hard disk. Furthermore, free disk space may not be sufficient, as it has to be in either unallocated free space outside of any existing partition, or partitions must be available for reformatting.

This is non-trivial for most systems that have not already had multi-boot configurations setup before, and this step, which must be taken care of first, can easily be more time-consuming than the actual installation. We have seen systems which can take hours to prepare as far as the partitioning goes, but once done, installation can be performed in 20 minutes or so.

Most LiveCD/USB media contain system software to resize, move, create and delete disk partitions; most use a program called **gparted**. If you are lucky you can simply use **gparted** to shrink an already existing partition and free up 20-30 GB or so, then do your normal installation. Be careful during the procedure to properly answer any questions about your hard disk layout so you do not destroy previously existing in-use partitions.

However, many OEM-installed systems have already used four **primary** disk partitions; if this is the case you cannot create any new partitions. (You can have no more than four primary partitions, or up to three primary partitions plus an **extended** partition in which you can create a number of **logical** partitions.) On these brain-dead systems one usually finds two partitions reserved for **Windows** (a boot partition and the `C:` drive), one partition reserved for the recovery disk and one partition for manufacturer diagnostics. If you are stuck with this situation, you have to delete a partition to get your primaries down to three or do more complicated things such as converting one of the primary partitions to a logical one, and you will still have to do some steps of shrinking and moving partitions.

It is impossible for us at the **Linux Foundation** to give detailed instructions on how to do this. Each system varies as to its pre-existing layout, and the potential for turning your system into a doorstop is quite high. We do not have the technical support bandwidth to take care of things like this. Therefore, we will simply refer you to your favored distribution and its install pages for technical assistance.

Please note that very recent hardware may contain **UEFI Secure Boot** mechanisms on the motherboard. If this is enabled in the **BIOS**, the situation is more complicated and there is not a universally accepted method of making Linux co-exist with it for now. It is beyond our current ability to give technical support in this situation.

The bottom line is that unless you feel comfortable messing with your partitioning setup, have the time to deal with any potential problems, and have an available lifeline if disaster strikes, you will probably be better off doing a virtual machine installation.

As mentioned under **Installing Virtual Machine Images**, once you have the hypervisor installed, the actual installation time for a virtual machine is basically zero since all you have to do is attach our image file to it.

C Using a Proxy Server with BlueJeans

The information below contains the recommended firewall configuration for optimum quality when joining a **BlueJeans** hosted virtual class. For proxy configuration, we would recommend allowing the following exceptions:

```
*.bjn.vc
*.bluejeans.com
```

Connections made to the **BlueJeans** cloud server use the following TCP and UDP ports. Make sure to open these firewall ports against **BlueJeans**' entire ip range here:

```
199.48.152.0/22
31.171.208.0/21
103.20.59.0/24
103.255.54.0/24
8.10.12.0/24
165.254.117.0/24
```

Note: **BlueJeans** has several POPs distributed globally. The call will be automatically redirected to the closest/ native POP to the end point or media egress point. Audio/video traffic will be routed to any of above ip range, based on geo location. Hence it's important that firewall ports are opened against all the ip ranges shown above.

Browsers, BlueJeans App and Mobile App (iOS and Android):

Outbound TCP Port 443 or 5061 or 5000 - Call Setup Signaling and Media
Outbound UDP Ports 5000-5999 - RTP Media

Please Note:

a) If your web traffic routes via an HTTP Proxy, please create an exception to allow network traffic to route to `*.bjn.vc`

b) Please make sure that Akamai and AWS are not being blocked by your firewall. Blocking them might cause instability with certain features in our services]

H.323 based Room System:

Outbound TCP Port 1720 - H.225 Signaling for H.323
Outbound TCP Ports 5000-5999 - H.245 Call Control for H.323
Outbound UDP Ports 5000-5999 - RTP Media

SIP based Room System:

Outbound TCP Port 5060 - SIP Signaling
Outbound TCP Port 5061 - SIPS (TLS) Signaling
Outbound UDP Ports 5000-5999 - RTP Media

Microsoft Lync/ Skype For Business client:

Outbound and Inbound TCP Port 5061 - Lync Federation and SIP/TLS connection.
Outbound and Inbound UDP Ports 50000-59999 - RTP Media
Outbound and Inbound TCP Ports 50000-59999 - RTP Media

Please Note: Some firewalls, such as Palo Alto Networks, prefer to filter network traffic based on the Fully Qualified Domain Name (FQDN). If this applies to your firewall configuration please use the following FQDN in order to connect to **BlueJeans**:

```
bjn.vc
```