Welcome to Linux Foundation E-Learning Training

2023/09/21 Version 9.8

If you have any further questions after reading this document, the answer may be found in the FAQ on the website: `http://bit.ly/LF-FAQ` or `https://training.linuxfoundation.org/about/faqs/general-faq/`

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

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://prep.lf.training/`

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://cm.lf.training/VIRTUAL_MACHINE_IMAGES/`
where you should log in with these credentials:
Username: `LFtraining`
Password: `Penguin2014`
The `000README` file in that directory contains deployment instructions.

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**.

1.3 Using AWS

Amazon Web Services (AWS) offers a wide range of virtual machine products (instances) that can be accessed by remote users in the cloud.

In particular, you can use the **AWS Free Tier** account level for up to a year and the simulated hardware and software choices available may be all you need to perform the exercises for **Linux Foundation training courses** and gain experience with open source software. Or, they may furnish a very educational supplement to working on local hardware, and offer opportunities to easily study more than one Linux distribution.

How can I get a AWS free tier account?

You can download a guide we have prepared to help you experiment with the AWS free tier: `https://prep.lf.training/docs/aws.pdf`

2 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 **LFS252**. You may want to consult the course-specific requirements first before doing a full **Linux** installation.

2.1 Basic Requirements and Linux Distributions

Linux Foundation courses are primarily 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.

2.2 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://prep.lf.training/`

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 LFS211
$ ./ready-for.sh --install LFS211
```

(You should substitute the name of your course for LFS211.)

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).

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 `https://training.linuxfoundation.org/about/policies/on-site-training-facility-requirements/` []

Appendices

A Course-Specific Hardware and/or Software Requirements

A.1 Coursera: Open Source Software Development, Linux and Git

Table 1: **Open Source Software Development, Linux 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
Free Disk Space in /boot	128 MiB

A.2 LFD110: Introduction to RISC-V

Table 2: **Introduction to RISC-V**

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	2000 bogomips
Minimum Amount of RAM	2 GiB
Free Disk Space in $HOME	5 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.3 LFD111: Building a RISC-V CPU Core

Table 3: **Building a RISC-V CPU Core**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser

A.4 LFD112: Building Applications with RISC-V and FreeRTOS

Table 4: **Building Applications with RISC-V and FreeRTOS**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable

A.5 LFD132: Introduction to Cloud Foundry

Table 5: **Introduction to Cloud Foundry**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser

A.6 LFD133: Introduction to WebAssembly

Table 6: **Introduction to WebAssembly**

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)

A.7 LFD134: WebAssembly Actors: From Cloud to Edge

Table 7: **WebAssembly Actors: From Cloud to Edge**

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)

A.8 LFD201: Intro to Open Source Development, Git, and Linux

Table 8: **Intro to Open Source Development, Git, and Linux**

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
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.9 LFD232: Cloud Foundry for Developers

Table 9: **Cloud Foundry for Developers**

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), Cloud Foundry CLI

A 64-bit operating system is required (**Linux**, **OSX**, **Windows**) with a bare minimum of 1 GB RAM.

Git must be installed, a text editor must be available, and the ability to install software (such as the **Cloud Foundry CLI** is essential.

A.10 LFD254: Containers for Developers and Quality Assurance

Table 10: **Containers for Developers and Quality Assurance**

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), Google Cloud Account, Docker Desktop

A.11 LFD259: Kubernetes for Developers

Table 11: **Kubernetes for Developers**

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)

A.12 LFD272: Hyperledger Fabric Fundamentals for Developers

Table 12: **Hyperledger Fabric Fundamentals for Developers**

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	2 GiB
Free Disk Space in $HOME	5 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.13 LFS101: Introduction to Linux

Table 13: **Introduction to Linux**

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
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.14 LFS143: Introduction to Service Mesh with Linkerd

Table 14: **Introduction to Service Mesh with Linkerd**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	This course cannot be run from a VM. You will be running VMs as a host machine
Required SW for class	docker, docker-engine, kubectl, kind/k3s

A.15 LFS151: Introduction to Cloud Infrastructure Technologies

Table 15: **Introduction to Cloud Infrastructure Technologies**

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
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.16 LFS153: Building Microservice Platforms with TARS

Table 16: **Building Microservice Platforms with TARS**

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	2 GiB
Free Disk Space in $HOME	5 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

- At least a bare metal or virtual machine with 2G+ memory.
- OS can be Linux (recommended), MacOSX, or Windows.
- You will need the following Software installed:
  - bison >= 2.5
  - cmake >= 2.8.8
  - docker
  - flex >= 2.5
  - gcc >= 4.8
  - git
  - glibc-devel

- golang >= 1.9.X
- JDK >= 1.8
- Maven >= 2.2.1
- mysql >= 4.1.17
- Node.js >= 12.13.0
- nvm >= 0.35.1
- php >= 7
- swoole >= 2

A.17 LFS156: Introduction to Kubernetes on Edge with K3s

Table 17: **Introduction to Kubernetes on Edge with K3s**

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	3000 bogomips
Required CPU features	svm|vmx
Minimum Amount of RAM	8 GiB
Free Disk Space in $HOME	50 GiB
Free Disk Space in /boot	128 MiB
Distro Architecture	x86_64
Supported Linux Distros	

A.18 LFS157: Introduction to Serverless on Kubernetes

Table 18: **Introduction to Serverless on Kubernetes**

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	3000 bogomips
Required CPU features	svm|vmx
Minimum Amount of RAM	8 GiB
Free Disk Space in $HOME	20 GiB
Free Disk Space in /boot	128 MiB
Distro Architecture	x86_64
Supported Linux Distros	

- This course can be run on MacOS, Ubuntu Linux, Windows 10 with Git Bash.
- The instructions for this course have been tested with an Intel computer with 8GB RAM and 2 CPUs, using Ubuntu 18.04, MacOS, and Windows 10 with Git Bash. The software is hosted in the cloud, however, you may need up to 10GB of free space on your computer for building container images.

A.19 LFS158: Introduction to Kubernetes

Table 19: **Introduction to Kubernetes**

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)

A.20 LFS163: Introduction to ONAP: Complete Network Automation

Table 20: **Introduction to ONAP: Complete Network Automation**

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)

A.21 LFS164: Introduction to Open Source Networking Technologies

Table 21: **Introduction to Open Source Networking Technologies**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser

A.22 LFS165: Introduction to Open Source Networking Technologies

Table 22: **Introduction to Open Source Networking Technologies**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser

This course requires use of a web browser and basic knowledge of networking and virtualization.

A.23 LFS166: Introduction to Magma: Cloud Native Wireless Networking

Table 23: **Introduction to Magma: Cloud Native Wireless Networking**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser

A.24 LFS170: Blockchain: Understanding Its Uses and Implications

Table 24: **Blockchain: Understanding Its Uses and Implications**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser

A.25 LFS171: Introduction to Hyperledger Blockchain Technologies

Table 25: **Introduction to Hyperledger Blockchain Technologies**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser

A.26 LFS176: Hyperledger Besu Essentials: Creating a Private Blockchain Network

Table 26: **Hyperledger Besu Essentials: Creating a Private Blockchain Network**

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	8 GiB
Free Disk Space in $HOME	750 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.27 LFS182: Securing Your Software Supply Chain With Sigstore

Table 27: **Securing Your Software Supply Chain With Sigstore**

Internet Access	Required
OS required for class	Linux, MacOS
Virtual Machine	Acceptable

A.28 LFS200: Fundamentals of Open Source IT and Cloud Computing

Table 28: **Fundamentals of Open Source IT and Cloud Computing**

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
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.29 LFS203: Linux for Cloud Technicians Essentials

Table 29: **Linux for Cloud Technicians Essentials**

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	1 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.30 LFS207: Linux System Administration Essentials

Table 30: **Linux System Administration Essentials**

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
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.31 LFS211: Linux Networking and Administration

Table 31: **Linux Networking and 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
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.32 LFS216: Linux Security Fundamentals

Table 32: **Linux Security Fundamentals**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Highly Recommended
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 8)
Minimum CPU Performance	20000 bogomips
Required CPU features	svm|vmx
Minimum Amount of RAM	8 GiB
Free Disk Space in $HOME	40 GiB
Free Disk Space in /boot	128 MiB
Kernel Configuration Options	HAVE_KVM KSM
Supported Linux Distros	

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.33 LFS241: Monitoring Systems and Services with Prometheus

Table 33: **Monitoring Systems and Services with Prometheus**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 2)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	4 GiB
Free Disk Space in $HOME	50 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A single machine with Ubuntu 18.04 or similar Linux distribution installed. I expect all of this to work on any popular modern Linux distribution (like Red Hat, Arch, or Debian), but minor details may be different.

The machine should have at least:

- 4GB of RAM
- 2 CPU cores
- 50GB of free disk space.

Docker should be installed, with a non-root user in the docker group so that users can start Docker containers without becoming root. (Tutorial on this: `https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docke`)

The user should have sudo access to be able to execute commands as root. (Tutorial on this: `https://www.digitalocean.com/community/tutorials/how-to-create-a-sudo-user-on-ubuntu-quickstart`)

The following basic system utilities need to be installed (most of them should be pre-installed by default): tar, unzip, wget, curl

Note: When starting long-running processes (like the Prometheus server) throughout this course, we assume that you keep them running for the entire course duration unless noted otherwise. In production setups, you would typically use a supervisor software like systemd or a cluster manager like Kubernetes to keep server processes running in the background. In this course we will not assume a particular deployment system and run components manually from the command-line instead. To facilitate running multiple server processes over a single SSH session when working on a remote machine (even across logouts and reconnects), you can use terminal multiplexer tools like screen, tmux, or byobu, that allow you to create and manage multiple virtual terminals over the same connection. If you are new to terminal multiplexers, we recommend byobu, as it is the most modern and easiest to use.

A.34 LFS242: Cloud Native Logging with Fluentd

Table 34: **Cloud Native Logging with Fluentd**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Highly Recommended
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 2)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	2 GiB
Free Disk Space in $HOME	30 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.35 LFS243: Service Mesh Fundamentals

Table 35: **Service Mesh Fundamentals**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Highly Recommended
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 4)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	15 GiB
Free Disk Space in $HOME	30 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

Labs use a virtual machine (VM) to host a Kubernetes cluster with at least 2 CPUs and 8GB of RAM. This can be done using a cloud provider or desktop hypervisor. Ubuntu 20.04 LTS will be installed on the VM.

A.36 LFS244: Managing Kubernetes Applications with Helm

Table 36: **Managing Kubernetes Applications with Helm**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 2)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	4 GiB
Free Disk Space in $HOME	50 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

This course requires access to a Linux server or Linux desktop/laptop as well as VirtualBox on your machine (to run a VM) or access to the cloud environment (in this course, GCP will be used as an infrastructure platform, but no GCP-specific tools).

A.37 LFS250: Kubernetes and Cloud Native Essentials

Table 37: **Kubernetes and Cloud Native Essentials**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	modern web browser

A.38 LFS253: Containers Fundamentals

Table 38: **Containers Fundamentals**

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)

Student computers must be 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.39 LFS258: Kubernetes Fundamentals

Table 39: **Kubernetes Fundamentals**

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)

The labs were written using Ubuntu instances running on **Google Cloud Platform (GCP)**. They have been written to be vendor-agnostic so could run on AWS, local hardware, or inside of virtualization to give you the most flexibility and options. Each platform will have different access methods and considerations. As of v1.14.1 the minimum (as in barely works) size for VirtualBox is 3vCPU/8G memory/5G minimal OS disk for master and 1vCPU/1G memory/5G minimal OS disk for worker node. This would be space given entirely to the guest VMs, not shared with the host. On GCP we suggest 2vCPU/7.5G nodes. More details can be found in the lab exercises.

If using your own equipment you will have to disable swap on every node. There may be other requirements which will be shown as warnings or errors when using the kubeadm command. While most commands are run as a regular user, there are some which require root privilege. Please configure sudo access as shown in a previous lab.

You would also require a .pem or .ppk file to access the nodes. Each cloud provider will have a process to download or create this file. If attending in-person instructor led training the file will be made available during class.

Very Important

Please disable any firewalls while learning Kubernetes. While there is a list of required ports for communication between components, the list may not be as complete as necessary. If using GCP you can add a rule to the project which allows all traffic to all ports. Should you be using VirtualBox be aware that inter-VM networking will need to be set to promiscuous mode.

A.40 LFS260: Kubernetes Security Essentials

Table 40: **Kubernetes Security Essentials**

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)

A.41 LFS261: Implementing Continuous Delivery

Table 41: **Implementing Continuous Delivery**

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), Google Cloud account (free tier)

A.42 LFS263: ONAP Fundamentals

Table 42: **ONAP Fundamentals**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	Firefox browser and a Google Cloud Platform (GCP) account

A.43 LFS264: Anuket Fundamentals

Table 43: **Anuket Fundamentals**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	Firefox browser and a Google Cloud Platform (GCP) account

A.44 LFS266: DevOps for Network Engineers

Table 44: **DevOps for Network Engineers**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Required
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
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.45 LFS267: Jenkins Essentials

Table 45: **Jenkins Essentials**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	This course cannot be run from a VM. You will be running VMs as a host machine
Required SW for class	Vagrant and VirtualBox

This course requires only a recent version of a **Linux** distribution MacOS, or Windows with administrator privileges.

- A computer running a **baremetal** version of one of the following Operating Systems (which will be used as the host operating system to run this class' VM guest image):
  - A recent 64-bit Linux distribution
  - A recent 64-bit version of MacOS
  - A recent 64-bit version Windows 10
- At least 20 GB of free disk space
- 16GB RAM (lab VM will allocate 4GB)
- Vagrant version 2.2.10 (or later)
- VirtualBox version 6.1.14 (or later)
- Open access to the Internet and Chrome or Safari browsers
- Open ports 5000, 2222

A.46 LFS268: CI/CD with Jenkins X

Table 46: **CI/CD with Jenkins X**

Internet Access	Required
OS required for class	Linux, MacOS, Windows
Virtual Machine	Acceptable
Required SW for class	Google Cloud Platform (GCP) free tier account

A.47 LFS269: GitOps: Continuous Delivery on Kubernetes with Flux

Table 47: **GitOps: Continuous Delivery on Kubernetes with Flux**

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	2000 bogomips
Minimum Amount of RAM	2 GiB
Free Disk Space in $HOME	5 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.48 LFS272: Hyperledger Fabric Administration

Table 48: **Hyperledger Fabric Administration**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Highly Recommended
Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Required CPU features	svm|vmx
Minimum Amount of RAM	4 GiB
Free Disk Space in $HOME	30 GiB
Free Disk Space in /boot	128 MiB
Supported Linux Distros	

A.49 LFW111: Introduction to Node.js

Table 49: **Introduction to Node.js**

Internet Access	Required
OS required for class	Linux
Virtual Machine	Acceptable
Required CPU Architecture	x86_64
Preferred Number of CPUs	4 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	2 GiB
Free Disk Space in $HOME	5 GiB
Free Disk Space in /boot	128 MiB

A.50 LFW211: Node.js Application Development

Table 50: **Node.js Application Development**

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)

A.51 LFW212: Node.js Services Development

Table 51: **Node.js Services Development**

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)

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.