1.2. Course Introduction

1.3. Course Learning Objectives

By the end of this course, you should be able to:

- Containerize and deploy a new Python script.
- Configure the deployment with ConfigMaps, Secrets, and SecurityContexts.
- Understand multi-container Pod design.
- Configure probes for pod health.
- Update and roll back an application.
- Implement services and NetworkPolicies.
- Use PersistentVolumeClaims for state persistence.

1.4. Course Formatting

In order to make it easier to distinguish the various types of content in the course, we use the color coding and formats below:

- **Bold: names of programs or services (or used for emphasis)**
- Light blue: designates hyperlinks
- `Dark blue: text typed at the command line, system output at the command line.`

1.5. Grading and Certificate of Completion

Once you reach the end of the course, you will notice the following two buttons: **Completed** and **Not Completed**.

If you feel like you mastered the course, click on the **Completed** button to mark the course as complete. Once you completed the course, you can download your certificate of completion from **The Linux Foundation** website > Training > My Portal > Training Classes > Previous Classes > **Kubernetes for Developers** > Download Certificate.

Clicking on the **Not Completed** button will automatically take you to the beginning of this course.

1.6. Course Timing

This course is entirely self-paced; there is no fixed schedule for going through the material. You can go through the course at your own pace, and you will always be returned to exactly where you left off when you come back to start a new session. However, we still suggest you avoid long breaks in between periods of work, as learning will be faster and content retention improved.

You have unlimited access to this course for 12 months from the date you registered, even after you have completed the course.

The chapters in the course have been designed to build on one another. It is probably best to work through them in sequence; if you skip or only skim some chapters quickly, you may find there are topics being discussed you have not been exposed to yet. But this is all self-paced, and you can always go back, so you can thread your own path through the material.

1.7.a. Exercises-Lab Environment

The lab exercises were written using Google Compute Engine (GCE) nodes. They have been written to be vendor-agnostic, so they could run on AWS, local hardware, or inside of virtual machines, to give you the most flexibility and options.

Each node has 2 vCPUs and 7.5G of memory, running Ubuntu 16.04. Smaller nodes should work, but you should expect a slow response. Other operating system images are also possible, but there may be a slight difference in some command outputs.

Using GCE requires setting up an account, and will incur expenses if using nodes of the size suggested. The *Getting Started pages* can be viewed online.

Amazon Web Service (AWS) is another provider of cloud-based nodes, and requires an account; you will incur expenses for nodes of the suggested size. You can find videos and information about how to launch a Linux virtual machine on the AWS website.
Virtual machines such as KVM, VirtualBox, or VMWare can also be used for the lab systems. Putting the VMs on a private network can make troubleshooting easier.
Finally, using bare-metal nodes, with access to the Internet, will also work for the lab exercises.

1.7.b. Exercises - Labs

In all **The Linux Foundation** courses (in any format) we put a heavy emphasis on learning by doing. In live, instructor-led classes, we almost always aim for a 50/50 balance between lecture and discussion, and working on laboratory exercises, or homeworks, that either perform the tasks just described in the class, or try more ambitious variations. Instructors help students figure out how to do things, debug their code and scripts, etc. during these lab sessions. Because this course is self-paced, without a live instructor, it will be up to you to control your time budget and make sure you take enough time to do labs.

For each exercise we present the description in a file you can download.

For convenience, you can also download a single file containing all the lab exercises by clicking on the **Document** button below, or pressing the **D** keystroke.

DOCUMENT

1.7.c. Exercises - Knowledge Check

At the end of each chapter, you will also find a series of knowledge check questions. These questions, just like the labs, were designed with one main goal in mind: to help you better comprehend the course content and reinforce what you have learned. It is important to point out that the **labs and knowledge check questions are not graded**. We would like to emphasize as well that **you will not be required to take a final exam to complete this course**.

1.8. Course Resources

Resources for this course can be found online. Making updates to this course takes time. Therefore, if there are any changes in between updates, you can always access course updates, as well as the course resources online:

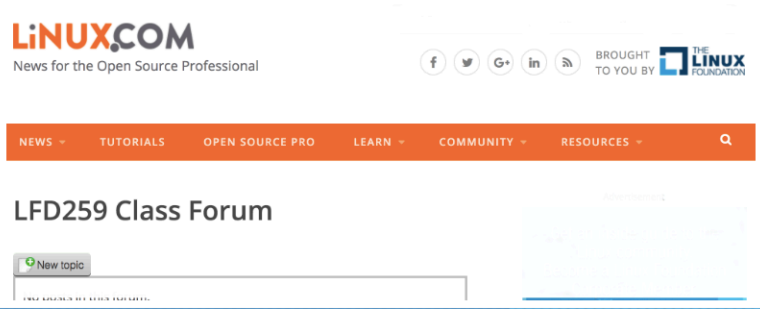
- Go to The Linux Foundation training website to obtain Course Resources
- The user ID is **LFtraining** and the password is **Penguin2014**.

1.9. Class Forum Guidelines

One great way to interact with peers taking this course is via the **Class Forum** that you can find at the linux.com website. The forum can be used in the following ways:

- To introduce yourself to other peers taking this course.
- To discuss concepts, tools and technologies presented in this course, or related to the topics discussed in the course materials.
- To ask questions about course content.
- To share resources and ideas related to Kubernetes.

The **Class Forum** will be reviewed periodically by **The Linux Foundation** staff, but it is primarily a community resource, not an 'ask the instructor' service.



1.10. Course Support

Important Note: We use a single sign-on service to launch the course once users are on their 'myportal' page. Do not attempt to change your password on the **QuickStart** course player engine, as this will break your single sign-on.

For any issues with your username or password, visit The Linux Foundation ID website.

For any course content-related questions, please use the course forum on Linux.com (see the details in the *Class Forum Guidelines* section).

If you need assistance beyond what is available above, please email us at: training@linuxfoundation.org and provide your Linux Foundation ID username and a description of your concern.

You can download a list of most frequently asked support questions by clicking on the Document button below or by using the D keystroke.

DOCUMENT

1.11. Course Audience and Requirements

This course is for developers looking to learn how to deploy, configure, and test their containerized applications on a multi-node Kuber

For a successful learning experience, basic Linux command line and file editing skills are required. Familiarity with using a programmir (such as Python, Node.js, Go) and Cloud Native application concepts and architectures is helpful.

Our free *LFS158x - Introduction to Kubernetes* MOOC on edX is a useful preparation for this course.

1.11. Course Audience and Requirements

This course is for developers looking to learn how to deploy, configure, and test their containerized applications on a multi-node Kubernetes cluster.

For a successful learning experience, basic Linux command line and file editing skills are required. Familiarity with using a programming language (such as Python, Node.js, Go) and Cloud Native application concepts and architectures is helpful.

Our free *LFS158x - Introduction to Kubernetes* MOOC on edX is a useful preparation for this course.

1.12. Software Environment

The material produced by The Linux Foundation is distribution-flexible. This means that technical explanations, labs and procedures should work on most modern distributions, and we do not promote products sold by any specific vendor (although we may mention them for specific scenarios).

In practice, most of our material is written with the three main Linux distribution families in mind:

- Debian/Ubuntu
- Red Hat/Fedora
- openSUSE/SUSE.

Distributions used by our students tend to be one of these three alternatives, or a product that is derived from them.

1.13. Which Distribution to Choose?

You should ask yourself several questions when choosing a new distribution:

- Has your employer already standardized?
- Do you want to learn more?
- Do you want to certify?

While there are many reasons that may force you to focus on one Linux distribution versus another, we encourage you to gain experience on all of them. You will quickly notice that technical differences are mainly about package management systems, software versions and file locations. Once you get a grasp of those differences, it becomes relatively painless to switch from one Linux distribution to another.

Some tools and utilities have vendor-supplied front-ends, especially for more particular or complex reporting. The steps included in the text may need to be modified to run on a different platform.

1.14. Debian Family

The **Debian** distribution is the upstream for several other distributions, including **Ubuntu**, **Linux Mint** and others. Debian is a pure open source project, and focuses on a key aspect: **stability**. It also provides the largest and most complete software repository to its users.

Ubuntu aims at providing a good compromise between long term stability and ease of use. Since Ubuntu gets most of its packages from Debian's unstable branch, Ubuntu also has access to a very large software repository. For those reasons, we decided to use Ubuntu as the reference Debian-based distribution for our lab exercises:

- Commonly used on both servers and desktops
- DPKG-based, uses apt-get and front-ends for installing and updating
- Upstream for Ubuntu, Linux Mint and others
- Current material based upon the latest release of Ubuntu and should work well with later versions
- x86 and x86-64
  - Long Term Release (LTS)
- **Note**: Ubuntu is used for demos and labs because it is available at no cost, as is Debian, but has a more relevant user base.

1.15. Red Hat/Fedora Family

Fedora is the community distribution that forms the basis of **Red Hat Enterprise Linux**, **CentOS**, **Scientific Linux** and **Oracle Linux**. Fedora contains significantly more software than Red Hat's enterprise version. One reason for this is that a diverse community is involved in building Fedora; it is not just one company.

The Fedora community produces new versions every six months or so. For this reason, we decided to standardize the Red Hat/Fedora part of the course material on the latest version of CentOS 7, which provides much longer release cycles. Once installed, CentOS is also virtually identical to Red Hat Enterprise Linux (RHEL), which is the most popular Linux distribution in enterprise environments:

- Current material is based upon the latest release of Red Hat Enterprise Linux (RHEL) - 7.x at the time of publication, and should work well with later versions
- Supports x86, x86-64, Itanium, PowerPC and IBM System Z
- RPM-based, uses yum (or dnf) to install and update
- Long release cycle; targets enterprise server environments
- Upstream for CentOS, Scientific Linux and Oracle Linux
- **Note**: CentOS is used for demos and labs because it is available at no cost.

1.16. openSUSE Family

The relationship between **openSUSE** and **SUSE Linux Enterprise Server** is similar to the one we just described between Fedora and Red Hat Enterprise Linux. In this case, however, we decided to use openSUSE as the reference distribution for the openSUSE family, due to the difficulty of obtaining a free version of SUSE Linux Enterprise Server. The two products are extremely similar and material that covers openSUSE can typically be applied to SUSE Linux Enterprise Server with no problem:

- Current material is based upon the latest release of openSUSE, and should work well with later versions.
- RPM-based, uses zypper to install and update
- YaST available for administration purposes
- x86 and x86-64
- Upstream for SUSE Linux Enterprise Server (SLES)
- **Note**: openSUSE is used for demos and labs because it is available at no cost.

1.17. New Distribution Similarities

Current trends and changes to the distributions have reduced some of the differences between the distributions.

- **systemd** (system startup and service management)
  systemd is used by the most common distributions, replacing the **SysVinit** and **Upstart** packages. Replaces **service** and **chkconfig** commands.
- **journald** (manages system logs)
  journald is a **systemd** service that collects and stores logging data. It creates and maintains structured, indexed journals based on logging information that is received from a variety of sources. Depending on the distribution, text-based system logs may be replaced.
- **firewalld** (firewall management daemon)
  firewalld provides a dynamically managed firewall with support for network/firewall zones to define the trust level of network connections or interfaces. It has support for IPv4, IPv6 firewall settings and for Ethernet bridges. This replaces the **iptables** configurations.
- **ip** (network display and configuration tool)
  The **ip** program is part of the **net-tools** package, and is designed to be a replacement for the **ifconfig** command. The **ip** command will show or manipulate routing, network devices, routing information and tunnels.

Since these utilities are common across distributions, the course content and lab information will use these utilities.

If your choice of distribution or release does not support these commands, please translate accordingly.

The following documents may be of some assistance translating older commands to their **systemd** counterparts:

- SysVinit Cheat Sheet
- Debian Cheat Sheet
- openSUSE Cheat Sheet

1.18. Amazon Web Services Free Tier

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 **The 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.

You can download a guide we have prepared to help you experiment with the AWS Free Tier.

1.19. Meet Your Instructor: Tim Serewicz

Timothy Serewicz is a Senior Instructor and Curriculum Developer at The Linux Foundation. With almost 20 years of experience working with the latest technologies including Cloud, NoSQL and Hadoop, he has trained countless professionals in their use. He currently maintains the Kubernetes Administration and OpenStack Administration courses, among others. In addition to The Linux Foundation, Tim is a certified trainer for Red Hat, Couchbase, MapR, Oracle University, and other startups.