Course Introduction

Course Learning Objectives

By the end of this course, you will learn the following:

- The history and evolution of Kubernetes.
- Its high-level architecture and components.
- The API, the most important resources that make the API, and how to use them.
- How to deploy and manage an application.
- Some upcoming features that will boost your productivity.

Course Audience and Requirements

Audience

This course is addressed to Linux administrators or software developers starting to work with containers and wondering how to manage them in production. In this course, you will learn the key principles that will put you on the journey to managing containerized applications in production.

Knowledge/Skills

To make the most of this course, you will need the following:

- A good understanding of Linux
- Familiarity with the command line
- Familiarity with package managers
- Familiarity with Git and GitHub
- Access to a Linux server or Linux desktop/laptop
- VirtualBox on your machine, or access to a public cloud.

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 Linux 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.

Lab Environment

The lab exercises were written and tested using Ubuntu instances running on Google Cloud Platform. 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 3 vCPUs and 7.5G of memory, running Ubuntu 18.04. 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 GCP requires setting up an account, and will incur expenses if using nodes of the size suggested. For more information review *Quickstart Using a Linux VM*.

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. As of Kubernetes v1.16.1, the minimum (as in barely
works) size for VirtualBox is 3vCPU/4G memory/5G minimal OS for the master, and 1vCPU/2G memory/5G minimal OS
for the worker node.

Finally, using bare-metal nodes, with access to the Internet, will also work for the lab exercises.

Course Resources

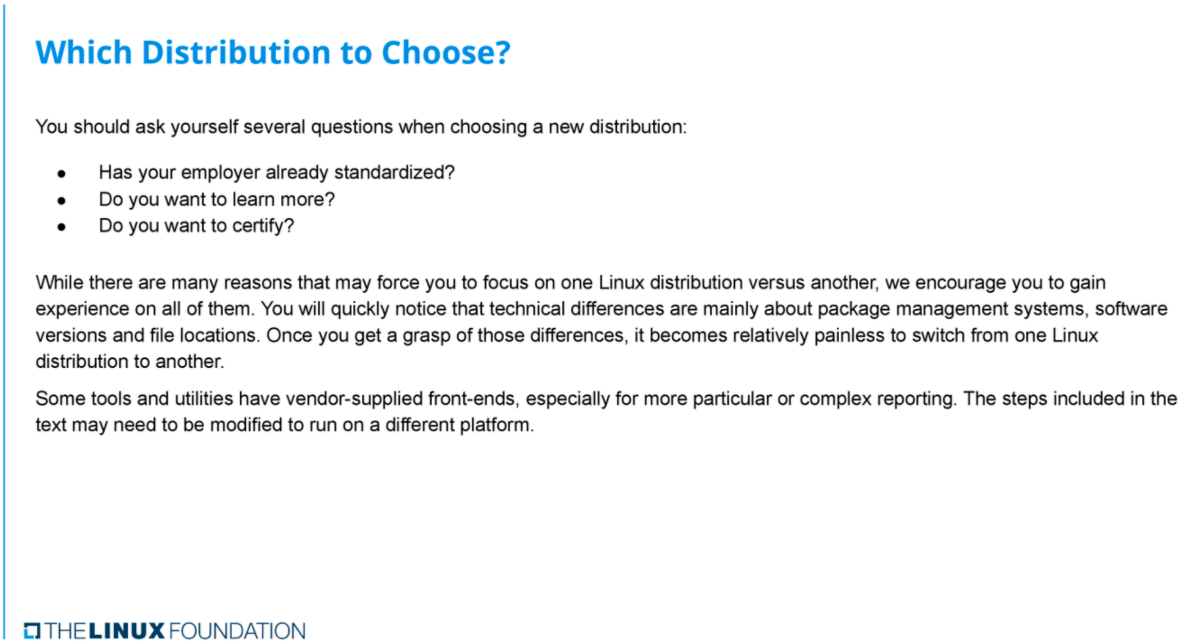
Resources for this course can be found online. Making updates to this course takes time. Therefore, if there are any changes in between updates, you can always access course updates, as well as the course resources online:

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

Which Distribution to Choose?

Which Distribution to Choose?

THE LINUX FOUNDATION



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.

THE LINUX FOUNDATION

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:

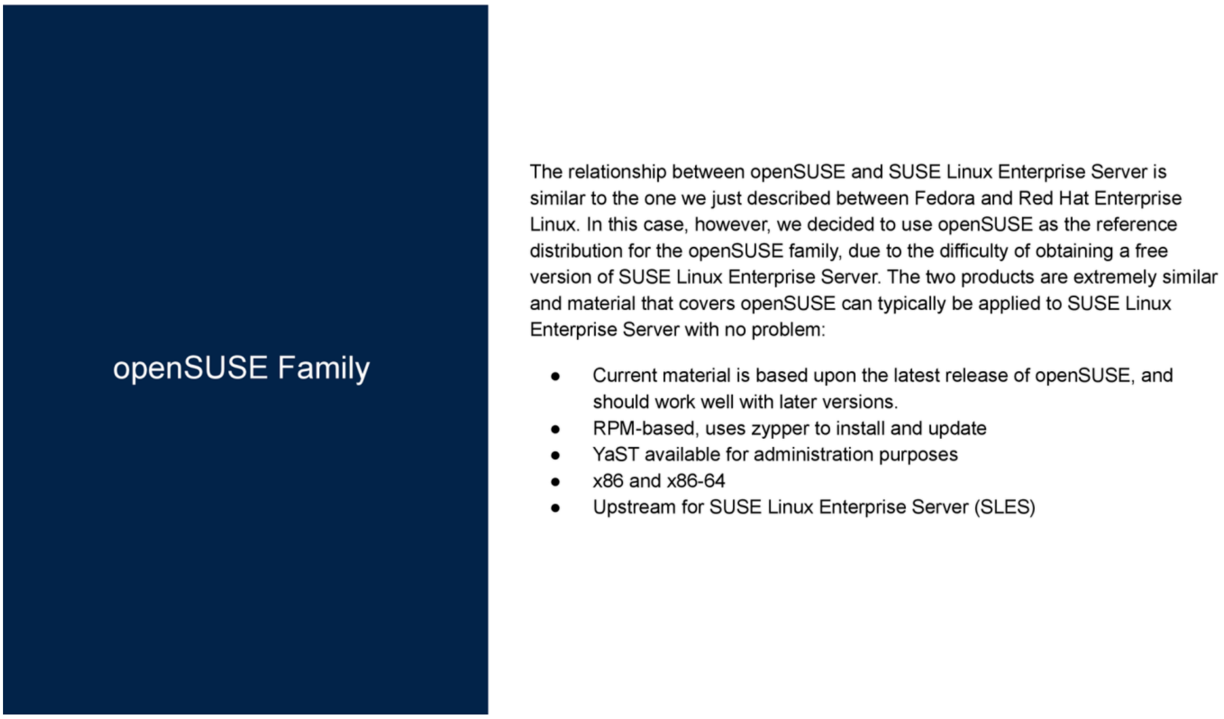
- Commonly used on both servers and desktops
- DPKG-based, uses apt-get and front-ends for installing and updating
- Upstream for Ubuntu, Linux Mint and others
- Current material based upon the latest release of Ubuntu and should work well with later versions
- x86 and x86-64
  - Long Term Release (LTS)

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:

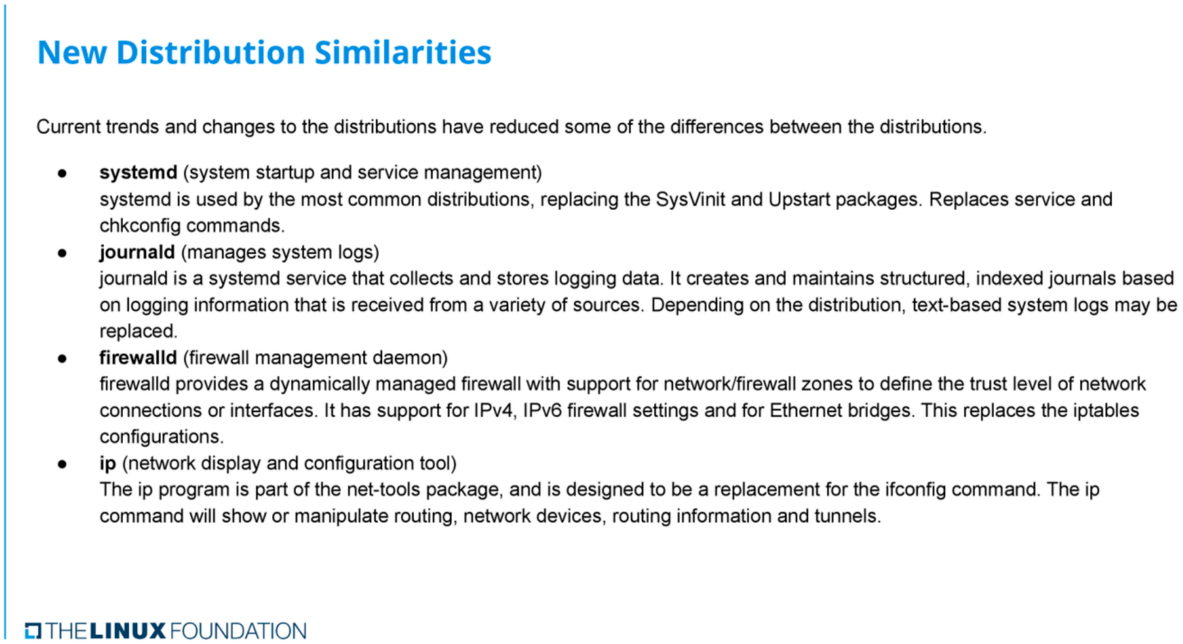
- Current material is based upon the latest release of Red Hat Enterprise Linux (RHEL) - 7.x at the time of publication, and should work well with later versions
- Supports x86, x86-64, Itanium, PowerPC and IBM System Z
- RPM-based, uses yum (or dnf) to install and update
- Long release cycle; targets enterprise server environments
- Upstream for CentOS, Scientific Linux and Oracle Linux



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)



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.

Using AWS to Set Up Labs

The Linux Foundation

The Linux Foundation

The Linux Foundation is dedicated to building sustainable ecosystems around open source projects to accelerate technology development and industry adoption. Founded in 2000, the Linux Foundation provided unparalleled support for open source communities through financial and intellectual resources, infrastructure, services, events, and training. Working together, the Linux Foundation and its projects form the most ambitious and successful investment in the creation of shared technology.

Linux is the world's largest and most pervasive open source software project in history. The Linux Foundation is home to the Linux creator Linus Torvalds and lead maintainer Greg Kroah-Hartman, and provides a neutral home where Linux kernel development can be protected and accelerated for years to come. The success of Linux has catalyzed growth in the open source community, demonstrating the commercial efficacy of open source and inspiring countless new projects across all industries and levels of the technology stack.

The Linux Foundation is the umbrella for many critical open source projects that power corporations today, spanning all industry sectors:

- Big data and analytics: ODPi, R Consortium
- Networking: OpenDaylight, OPNFV, ONAP
- Embedded: Dronecode, Zephyr
- Web tools: OpenJS Foundation
- Cloud computing: Cloud Foundry, Cloud Native Computing Foundation, Open Container Initiative
- Automotive: Automotive Grade Linux
- Security: The Core Infrastructure Initiative
- Blockchain: Hyperledger
- And many more.

Cloud Native Computing Foundation (CNCF)

Cloud Native Computing Foundation (CNCF) is an open source software foundation under the Linux Foundation umbrella dedicated to making cloud native computing universal and sustainable. Cloud native computing uses an open source software stack to deploy applications as microservices, packaging each part into its own container, and dynamically orchestrating those containers to optimize resource utilization. Cloud native technologies enable software developers to build great products faster.

CNCF serves as a vendor-neutral home for many of the fastest-growing projects on GitHub, including Kubernetes, Prometheus, and Envoy, fostering collaboration between the industry's top developers, end users and vendors.

The Linux Foundation Events

The Linux Foundation produces technical events around the world. Whether it is to provide an open forum for development of the next kernel release, to bring together developers to solve problems in a real-time environment, to host workgroups and community groups for active discussions, to connect end users and kernel developers in order to grow Linux and open source software use in the enterprise or to encourage collaboration among the entire community, we know that our conferences provide an atmosphere that is unmatched in their ability to further the platform.

The Linux Foundation hosts an increasing number of events each year, including:

- Open Source Summit North America, Europe, Japan, and China
- Embedded Linux Conference + OpenIoT Summit North America and Europe
- Open Source Leadership Summit

- Open Networking Summit North America and Europe
- KubeCon + CloudNativeCon North America, Europe, and China
- Automotive Linux Summit
- KVM Forum
- Linux Storage Filesystem and Memory Management Summit
- Linux Security Summit North America and Europe
- Cloud Foundry Summit
- Hyperledger Global Forum, etc.

You can learn more about the Linux Foundation events online.