O'REILLY®

Networking & Kubernetes

A Layered Approach

James Strong
& Vallery Lancey

Networking and Kubernetes

A Layered Approach

James Strong and Vallery Lancey

O'REILLY®

Beijing • Boston • Farnham • Sebastopol • Tokyo

Networking and Kubernetes

by James Strong and Vallery Lancey

Printed in the United States of America.

Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O’Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (http://oreilly.com). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: John Devins

Development Editor: Melissa Potter

Production Editor: Beth Kelly

Copyeditor: Kim Wimpsett

Proofreader: Piper Editorial Consulting, LLC

Indexer: Sam Arnold-Boyd

Interior Designer: David Futato

Cover Designer: Karen Montgomery

Illustrator: Kate Dullea

September 2021: First Edition

Revision History for the First Edition

- 2021-09-07: First Release

See http://oreilly.com/catalog/errata.csp?isbn=9781492081654 for release details.

The views expressed in this work are those of the authors, and do not represent the publisher's views. While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-492-08165-4

[LSI]

Preface

Just Another Packet

Since the first two computers were joined together over a cable, networking has been a crucial part of our infrastructure. Networks now have layers and layers of complexity to support a multitude of use cases, and the advent of containers and projects like Mesosphere and Kubernetes have not changed that. While the contributors of Kubernetes have attempted to abstract away this networking complexity for developers, computer science is just that, abstraction upon abstraction. Kubernetes, and its networking API, is another abstraction that makes it easier and faster to deploy applications for consumption. What about the administrator who has to manage Kubernetes? This book intends to dispel the mysticism around the abstractions Kubernetes puts in place, guide administrators through the layers of complexity, and help you realize Kubernetes is not just another packet.

Who This Book Is For

According to 451 Research, the global application container market is expected to grow from USD 2.1 billion in 2019 to USD 4.2 billion by 2022 . This explosive growth in the container market underscores the need for IT professionals to be knowledgeable in deploying, managing, and troubleshooting containers.

This book is intended to be read from beginning to end by new network, Linux, or cluster administrators, and it can be used by more experienced DevOps engineers to jump to specific topics for which they find themselves needing to be upskilled. Network, Linux, and cluster administrators need to be familiar with how to operate Kubernetes at scale.

In this book, readers will find the information required to navigate the layers of complexity that come with running a Kubernetes network. This book will peel back the abstractions that Kubernetes puts in place so that developers have a similar experience across deployments on-premises, in the cloud, and with managed services. Engineers responsible for production cluster operations and network uptime can use this book to bridge the gap in their knowledge of those abstractions.

What You Will Learn

By the end of this book, the reader will understand the following:

- The Kubernetes networking model
- The Container Network Interface (CNI) project and how to choose a CNI project for their clusters
- Networking and Linux primitives that power Kubernetes

- The relationship between the abstractions powering the Kubernetes network

Also, the reader will be able to do the following:

- Deploy and manage a production-scale network for Kubernetes clusters

- Troubleshoot underlying network-related application issues inside a Kubernetes cluster

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`

Shows commands or other text that should be typed literally by the user.

`Constant width italic`

Shows text that should be replaced with user-supplied values or by values determined by context.

> **TIP**
>
> This element signifies a tip or suggestion.

> **NOTE**
>
> This element signifies a general note.

> **WARNING**
>
> This element indicates a warning or caution.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at *https://github.com/strongjz/Networking-and-Kubernetes*.

If you have a technical question, or a problem using the code examples, please send email to *bookquestions@oreilly.com*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but generally do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Networking and Kubernetes* by James Strong and Vallery Lancey (O'Reilly). Copyright 2021 Strongjz tech and Vallery Lancey, 978-1-492-08165-4."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

O'Reilly Online Learning

NOTE

For more than 40 years, *O'Reilly Media* has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O’Reilly and 200+ other publishers. For more information, visit *http://oreilly.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O’Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

800-998-9938 (in the United States or Canada)

707-829-0515 (international or local)

707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *https://oreil.ly/NetKubernetes*.

Email *bookquestions@oreilly.com* to comment or ask technical questions about this book.

For news and information about our books and courses, visit *http://oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*.

Follow us on Twitter: *http://twitter.com/oreillymedia*.

Watch us on YouTube: *http://youtube.com/oreillymedia*.

Acknowledgments

The authors would like to thank the team at O'Reilly Media for helping them through the process of writing their first book. Melissa Potter was instrumental in getting this across the finish line. We would also like to recognize Thomas Behnken for aiding us with his Azure expertise.

James: Karen, thank you for all your faith in me and for helping him believe in himself even when he didn't. Wink, you are the reason I started working in this field, and I am forever grateful. Ann, I have come a long way since learning English is supposed to be capitalized. James would also like to thank all the other teachers and coaches in his life who supported him.

Vallery: I'd like to thank the friendly faces in SIG-Network for helping me get started in upstream Kubernetes.

Finally, the authors would like to thank the Kubernetes community; this book wouldn't exist without them. We hope it helps further the knowledge

for all engineers looking to adopt Kubernetes.

Chapter 1. Networking Introduction

“Guilty until proven innocent.” That’s the mantra of networks and the engineers who supervise them. In this opening chapter, we will wade through the development of networking technologies and standards, give a brief overview of the dominant theory of networking, and introduce our Golang web server that will be the basis of the networking examples in Kubernetes and the cloud throughout the book.

Let’s begin…at the beginning.

Networking History

The internet we know today is vast, with cables spanning oceans and mountains and connecting cities with lower latency than ever before. Barrett Lyon’s “Mapping the Internet,” shown in Figure 1-1, shows just how vast it truly is. That image illustrates all the connections between the networks of networks that make up the internet. The purpose of a network is to exchange information from one system to another system. That is an enormous ask of a distributed global system, but the internet was not always global; it started as a conceptual model and slowly was built up over time, to the behemoth in Lyon’s visually stunning artwork. There are many factors to consider when learning about networking, such as the last mile, the connectivity between a customer’s home and their internet service provider’s network—all the way to scaling up to the geopolitical landscape of the internet. The internet is integrated into the fabric of our society. In this book, we will discuss how networks operate and how Kubernetes abstracts them for us.

Figure 1-1. Barrett Lyon, “Mapping the Internet,” 2003

Table 1-1 briefly outlines the history of networking before we dive into a few of the important details.

Table 1-1. A brief history of networking

Year	Event
1969	ARPANET’s first connection test
1969	Telnet 1969 Request for Comments (RFC) 15 drafted
1971	FTP RFC 114 drafted
1973	FTP RFC 354 drafted
1974	TCP RFC 675 by Vint Cerf, Yogen Dalal, and Carl Sunshine drafted
1980	Development of Open Systems Interconnection model begins
1981	IP RFC 760 drafted

Year	Event
1982	NORSAR and University College London left the ARPANET and began using TCP/IP over SATNET
1984	ISO 7498 Open Systems Interconnection (OSI) model published
1991	National Information Infrastructure (NII) Bill passed with Al Gore's help
1991	First version of Linux released
2015	First version of Kubernetes released

In its earliest forms, networking was government run or sponsored; in the United States, the Department of Defense (DOD) sponsored the Advanced Research Projects Agency Network (ARPANET), well before Al Gore's time in politics, which will be relevant in a moment. In 1969, ARPANET was deployed at the University of California–Los Angeles, the Augmentation Research Center at Stanford Research Institute, the University of California–Santa Barbara, and the University of Utah School of Computing. Communication between these nodes was not completed until 1970, when they began using the Network Control Protocol (NCP). NCP led to the development and use of the first computer-to-computer protocols like Telnet and File Transfer Protocol (FTP).

The success of ARPANET and NCP, the first protocol to power ARPANET, led to NCP's downfall. It could not keep up with the demands of the network and the variety of networks connected. In 1974, Vint Cerf, Yogen Dalal, and Carl Sunshine began drafting RFC 675 for Transmission Control Protocol (TCP). (You'll learn more about RFCs in a few paragraphs.) TCP would go on to become the standard for network connectivity. TCP allowed for exchanging packets across different types of networks. In 1981, the Internet Protocol (IP), defined in RFC 791, helped break out the responsibilities of TCP into a separate protocol, increasing the modularity of the network. In the following years, many organizations, including the DOD, adopted TCP as the standard. By January 1983, TCP/IP had become the only approved protocol on ARPANET, replacing the earlier NCP because of its versatility and modularity.

A competing standards organization, the International Organization for Standardization (ISO), developed and published ISO 7498, "Open Systems Interconnection Reference Model," which detailed the OSI model. With its publication also came the protocols to support it. Unfortunately, the OSI model protocols never gained traction and lost out to the popularity of TCP/IP. The OSI model is still an excellent learning tool for understanding the layered approach to networking, however.

In 1991, Al Gore invented the internet (well, really he helped pass the National Information Infrastructure [NII] Bill), which helped lead to the creation of the Internet Engineering Task Force (IETF). Nowadays standards for the internet are under the management of the IETF, an open consortium of leading experts and companies in the field of networking, like Cisco and Juniper. RFCs are published by the Internet Society and the Internet Engineering Task Force. RFCs are prominently authored by individuals or groups of engineers and computer scientists, and they detail their processes, operations, and applications for the internet's functioning.

An IETF RFC has two states:

Proposed Standard

> A protocol specification has reached enough community support to be considered a standard. The designs are stable and well understood. A proposed standard can be deployed, implemented, and tested. It may be withdrawn from further consideration, however.

Internet Standard

> Per RFC 2026: “In general, an internet standard is a stable specification and well understood, technically competent, has multiple, independent, and interoperable implementations with substantial operational experience, enjoys significant public support, and is recognizably useful in some parts of the internet.”

NOTE

Draft standard is a third classification that was discontinued in 2011.

There are thousands of internet standards defining how to implement protocols for all facets of networking, including wireless, encryption, and data formats, among others. Each one is implemented by contributors of open source projects and privately by large organizations like Cisco.

A lot has happened in the nearly 50 years since those first connectivity tests. Networks have grown in complexity and abstractions, so let’s start with the OSI model.

OSI Model

The OSI model is a conceptual framework for describing how two systems communicate over a network. The OSI model breaks down the responsibility of sending data across networks into layers. This works well for educational purposes to describe the relationships between each layer's responsibility and how data gets sent over networks. Interestingly enough, it was meant to be a protocol suite to power networks but lost to TCP/IP.

Here are the ISO standards that outline the OSI model and protocols:

- ISO/IEC 7498-1, "The Basic Model"
- ISO/IEC 7498-2, "Security Architecture"
- ISO/IEC 7498-3, "Naming and Addressing"
- ISO/IEC 7498-4, "Management Framework"

The ISO/IEC 7498-1 describes what the OSI model attempts to convey:

> *5.2.2.1 The basic structuring technique in the Reference Model of Open Systems Interconnection is layering. According to this technique, each open system is viewed as logically composed of an ordered set of (N)-subsystems…Adjacent (N)-subsystems communicate through their common boundary. (N)-subsystems of the same rank (N) collectively form the (N)-layer of the Reference Model of Open Systems Interconnection. There is one and only one (N)-subsystem in an open system for layer N. An (N)-subsystem consists of one or several (N)-entities. Entities exist in each (N)-layer. Entities in the same (N)-layer are termed peer-(N)-entities. Note that the highest layer does not have an (N+l)-layer above it, and the lowest layer does not have an (N-1)-layer below it.*

The OSI model description is a complex and exact way of saying networks have layers like cakes or onions. The OSI model breaks the responsibilities of