# Build Serverless Apps on Kubernetes with Knative

*Build, deploy, and manage serverless applications on Kubernetes*

**Amit Deshpande**

**Anuj Gupta**

**Ashish Saxena**

www.bpbonline.com

First published: 2024

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

**UK | UAE | INDIA | SINGAPORE**

ISBN 978-93-55515-797

www.bpbonline.com

# Forewords

In recent years, serverless computing has emerged as a revolutionary paradigm, transforming the way we build and deploy applications. With its promise of scalability, reduced operational overhead, and pay-per-use pricing, serverless architecture has captured the imagination of developers and organizations alike. Among the many frameworks and platforms available for serverless development, Knative stands out as a powerful and flexible open-source solution. The ability to build and deploy applications without worrying about infrastructure management has revolutionized the way developers approach software development.

This book, "Serverless Apps with Knative" is a comprehensive resource that delves into the depths of Knative, empowering readers to harness its capabilities and unlock the full potential of serverless computing. With a blend of theoretical concepts, practical examples, and hands-on exercises, this book equips both beginners and experienced practitioners with the knowledge and skills needed to develop, deploy, and manage serverless applications using Knative.

The journey begins in **Chapter 1**, where readers are introduced to the fundamental concepts of serverless computing and provided with an overview of Knative's build components. This chapter sets the stage for the exploration of Knative's capabilities in the subsequent chapters. By the end of Chapter 1, readers will have a solid understanding of serving, eventing, event sourcing, and event consumption in a serverless context.

**Chapters 2 and 3** focus on the installation and configuration of Knative. In Chapter 2, readers will learn various methods to install and configure Knative, gaining insights into the background of installation, different installation approaches, and essential configuration concepts. The chapter also provides step-by-step recipes to guide readers through the installation process. Chapter 3 builds upon this foundation, introducing peripheral tools and frameworks for implementing DevSecOps and observability within the Knative serverless architecture. Through detailed recipes, readers will learn how to provision GitHub repositories for GitOps, install tools like ArgoCD, Prometheus, Grafana, Loki, and Jaeger, and verify the installation of these essential components.

The subsequent chapters dive deeper into Knative's core functionalities. **Chapter 4** introduces the Knative CLI, guiding readers through its installation, customization,

and plugin concepts. **Chapter 5** provides an overview of Knative Functions, explaining the simple programming model they offer. Through practical recipes, readers will learn to implement and deploy use cases using Knative CLI, kubectl, and YAML configurations.

**Chapter 6** explores Knative Eventing, focusing on understanding event-driven architectures and how Knative enables event-driven communication. Readers will gain insights into creating custom resources for Knative Eventing CRDs and deploying Apache Kafka clusters for event sourcing. This chapter also covers autoscaling with Apache Kafka and Knative Eventing, enabling readers to leverage these powerful features for handling event-driven workloads effectively.

**Chapter 7** introduces routers and autoscaling in the context of Knative. Readers will discover how to deploy multiple versions of a service and distribute traffic between them. The chapter also covers blue-green deployments and canary release patterns, empowering readers to adopt advanced deployment strategies with confidence.

The book culminates in **Chapter 8**, where patterns and best practices for utilizing Knative are explored. This chapter offers invaluable insights into using Istio as the default networking layer for Knative, implementing GitOps with GitHub actions and Argo CD for CI/CD, and achieving observability with log aggregation using Loki, as well as utilizing Jaeger, Prometheus, and Grafana.

"Serverless Apps with Knative" is a comprehensive and practical resource that equips readers with the necessary skills to leverage the power of Knative for building scalable and efficient serverless applications. Whether you are a developer, architect, or IT professional, this book will empower you to embrace serverless computing and harness the full potential of Knative.

I commend the authors for their meticulous attention to detail, comprehensive coverage of the subject matter, and their ability to make complex concepts accessible to readers of all levels. I am confident that this book will serve as an invaluable guide in your journey to mastering Knative and unlocking the true potential of serverless architecture.

Happy reading and serverless coding!

**— A B Vijay Kumar**
IBM Fellow
CTO Hybrid Cloud Services
IBM

# Second Foreword

As a Practice leader & Thought leader in the Hybrid Multi Cloud team at IBM, I am responsible for partnering with multiple clients across industries in their Business modernization strategies; and developing skills, capabilities, tools & assets to facilitate & accelerate their Cloud transformation journeys. Through my 26 years of career, I have been a constant innovator, embracing new technologies that have changed & disrupted the IT industry in a positive way. As an IBM Master Inventor with 15+ issued & 6+ filed patents, and 30+ IP.com technical publications, I have been an advocate of exploring new technologies to drive efficiencies, productivity, environment sustainability and quality.

Serverless computing has numerous advantages, as it allows application developers to focus only on the application code delivering business value without worrying about underlying infrastructure, capacity planning, scalability, and so on. Knative is one such novel upcoming framework in Serverless technology that is changing the way applications are built and / or modernized. Knative is a Cloud Native Computing Foundation (CNCF) incubation open-source project, supported by major companies like IBM, Google, VMWare, RedHat. It provides a Kubernetes based serverless container platform to build, deploy & manage serverless workloads. It abstracts away many of the complexities associated with infrastructure management. By providing a cloud agnostic framework, it provides for developers to leverage innovation from across cloud providers, shift from one provider to another easily eliminating vendor lock in, and cater to a multi / Hybrid Cloud environment.

Knative Serving, one of the two key components, provides a way to deploy & run containerized serverless workloads on Kubernetes cluster without worrying about infrastructure management. It handles the tasks related to managing the lifecycle of the containerized application, routing of traffic and revision control. It also scales the number of instances based on the traffic automatically, and can even scale down to zero when there is no traffic.

Knative Eventing, the other key component, provides the users a set of APIs to use event-driven architecture for Serverless applications. These APIs can be used to create components that route events from producers to consumers that receive events. Creation, parsing, sending & receiving of events is done in a cloud agnostic way.

This book introduces the reader to the concept of Serverless computing from a beginner's point of view, clarifying the concepts, detailing out the serverless offerings from the different cloud providers and breaking the myths related to the technology. It then introduces Knative and how it helps implement Serverless containers. The book talks in detail about its two key components (Serving & Eventing), Knative functions with hands-on code recipes, installing & configuring Knative including setting up GitOps, Observability and concepts like Autoscaling, Scale to zero, Revisions, Traffic Splitting between revisions and so on. The book provides detailed examples which will provide a deep understanding of the subject and equip the readers with all the information needed to develop production ready applications with Knative. In summary, whether you are an Architect, Developer, Tester or a Business leader, this book has something for you.

I am thrilled to see a book like this which starts from the basics, covering the fundamentals of Serverless & Knative in particular, and then deep dives into the various aspects of Knative. It also outlines the architectural perspectives of Knative to me, as it traverses a learner's journey from Novice to an Experienced professional.

As someone who has led several Cloud modernization projects, I am confident that Serverless & Knative provide tremendous benefits to developers & leaders in Cloud development. This book provides a complete reference to anyone who wishes to leverage Knative, one of the finest technologies to have evolved recently.

Cheers !

**— Deepak Gupta**
IBM Master Inventor
IBM
Linkedin:**/in/bankush/**

## Dedicated to

*My wonderful kids:*

***Ansh***

*&*

***Anisha***

# About the Authors

- **Amit Deshpande** is an Associate Partner & Executive Architect with IBM and has 21+ years of experience in the IT industry. He is a Thought Leader in the area of Digital Transformation & Hybrid Multi-Cloud Technologies. He has extensive experience in solutions & delivering multiple complex projects in Microservices/API, Integration, Event-Driven, Hybrid Cloud Architectures for various Banking, Manufacturing, and Automotive customers. He has been involved in the conceptualization and development of multiple Cloud Assets. Amit is an IBM Senior Certified Architect, Open Group Certified Distinguished Architect, Google Certified Professional Cloud Architect, and AWS Certified Solution Architect – Associate. Amit is a passionate mentor to several budding Architects in their journey in Architect Profession. He is an active member of the Architect Certification Review Board (CRB). He has also filed three patents in Kubernetes / Containerization area.

- **Anuj Gupta** is an Executive Architect in Hybrid Cloud Manage with a specialization in Redhat-OpenShift Microservices, which is part of IBM Cloud Application Services. Prior to this role, he was part of IBM Application Innovation Lab and Export Blue, working as a Senior Architect across multiple LOBs and Portfolios. During his 17+ years of experience, he has been involved in defining technical architecture and end-to-end delivery for multiple projects across Open Subsurface Data Lake (OSDU Opensource with the Open Group) for IBM Data & AI, B2B payments, Card Authorization System, Enterprise Digital Analytics, AP and AR systems of large Financial Services clients. He has extensive experience in developing cloud-native applications based on OpenShift for various workloads (stateful and stateless) and also migration of various monolith applications from VSS to IaaS and then to OpenShift using microservice architecture.

- **Ashish Saxena** is currently working as an Application Architect at IBM and has 15+ years of experience in the IT industry. He is an SME for Digital Transformation & Hybrid Cloud Technologies. He has extensive experience in solutions and designing applications using microservices and Event Driven architecture for various Telecommunication, Retail, and Banking customers. Ashish is a certified Azure Solution Architect Expert.

# About the Reviewer

**Gaurav** is a passionate technology leader and hands-on technologist, with a track record of driving technical innovation. Gaurav has delivered solutions for enterprises and start-ups operating in leadership, management, architectural, and development capacities. Gaurav has over 26 years of experience, collaborating with some of the most well-known technologies like Java, Microsoft, Angular, React, Python, PHP etc. pertaining to the domains Medical, Media, Construction, Gaming, Finance, ATM, Supply-chain, and so on. Gaurav has a doctorate in computer science (Machine Learning) from California Public University. Gaurav is a Microsoft MVP award recipient. He is a Mentor of Change with AIM NITI Aayog, Govt. of India, Business Coach with Business Blaster, Govt of NCT of Delhi. He is a lifetime member of the Computer Society of India (CSI), an advisory member and senior mentor at IndiaMentor. He has authored books across-the-technologies. Recently, Gaurav has recognized as a world record holder for writing books in exceptional technologies.

# Acknowledgement

I would like to take this opportunity to express my profound gratitude to my family – my parents, my wonderful wife Vidyuta, and my brother Amogh. Their unwavering support and encouragement have been instrumental in my journey.

I extend my heartfelt thanks to my esteemed friends and colleagues, Anuj Gupta and Ashish Saxena, for their invaluable contributions as co-authors of this book. Their extensive industry experience and technical prowess have added immense value to the book.

My sincere appreciation goes to the exceptional team at BPB Publications for their guidance and expertise throughout the book's development process. The contributions from reviewers, technical experts, and editors have been indispensable.

I am deeply grateful to my mentors, notably A B Vijay Kumar and Deepak Gupta, who have served as a constant source of inspiration and guided me towards the path of becoming an author. I would also like to express my gratitude to IBM and my management for their encouragement and support in writing this book.

Finally, I would like to extend my appreciation to all the readers who have taken an interest in this book and for their support in bringing it to fruition.

# Preface

In the ever-evolving landscape of software development, there has been a continuous search for tools and methodologies that streamline the process of creating and deploying applications. As cloud computing has become an integral part of modern development workflows, the focus has shifted towards harnessing its full potential, leading to the emergence of serverless architecture. Among the various technologies that support this paradigm, Knative has emerged as a powerful and versatile solution for deploying and managing serverless containers.

This book aims to provide a comprehensive guide to understanding, implementing, and optimizing Knative serverless containers for developers, DevOps engineers, and IT professionals. We will start by introducing the core concepts of serverless architecture and its benefits, followed by an in-depth exploration of Knative, its components, and how it fits within the broader Kubernetes ecosystem.

Through a series of practical examples and case studies, we will demonstrate how to build, deploy, and manage serverless containers using Knative. We will cover topics such as setting up a development environment, creating custom serverless applications, integrating with other cloud-native tools and services, and best practices for monitoring, logging, and troubleshooting.

In addition, we will delve into advanced topics such as scaling, and performance optimization, helping you gain a solid understanding of how to deploy and maintain high-performing, resilient serverless applications using Knative.

Whether you are new to serverless computing, a seasoned professional looking to expand your knowledge, or an organization considering adopting Knative, this book will serve as a valuable resource to guide you through the intricacies of serverless containers and their effective management.

As the field of serverless computing continues to evolve, it is our hope that this book will equip you with the knowledge and confidence to navigate and harness the power of Knative serverless containers, unlocking new possibilities and efficiencies in your software development journey.

Happy reading, and here's to a future of seamless, scalable, and efficient serverless applications!

**Chapter 1: Serverless and Knative in a Nutshell –** provides a detailed overview of the evolution of serverless, how it works, and its key advantages. The chapter also provides the reader with a point of view on when serverless should be considered and its pitfalls. Furthermore, the chapter covers serverless offerings, Function as a Service (FaaS), and Serverless Containers, along with an introduction to the Knative project.

**Chapter 2: Installation and Configuration of Knative - Part I -** presents detailed steps to install and configure various Knative pre-requisite software like Kubernetes (K8S) Cluster, Helm as a package manager, and Istio as a services mesh. The chapter explains the various ways of installing Knative and provides detailed installation and validation steps of Knative by deploying and running a pre-built app.

**Chapter 3: Installation and Configuration – Part II -** covers the installation of various monitoring and observability tools. The chapter explains the installation and validation steps for Kafka eventing and ArgoCD for GitOps. This chapter also covers the installation of various observability tools like Loki, Prometheus, Jaeger, and Grafana.

**Chapter 4: Knative Functions – An Overview -** allows the reader to learn fundamental concepts of Knative Functions by demonstrating how to create, build, deploy, and run Knative Functions on local and remote K8S Cluster using hands-on code recipes.

**Chapter 5: Knative Serving -** gives special attention to the Knative Serving component, demonstrating how to build, deploy and run serverless containers using Knative Serving on the local K8S Cluster. The chapter explains the Knative Serving concept based on a Case Study and hands-on code recipes.

**Chapter 6: Knative Eventing -** gives special attention to the Knative Eventing component, demonstrating how to build, deploy and run services using Knative Eventing on the local K8S Cluster. The chapter explains various Knative Eventing patterns with a Case Study and hands-on code recipes. This chapter also covers how Kafka can be used as a messaging service in Knative Eventing.

**Chapter 7: Scaling and Routing -** explains in detail how to scale the serverless application based on demand using a Case Study by deploying it on a remote K8S Cluster. This chapter also allows the reader to learn about various autoscaling configurations in Knative. Furthermore, the chapter covers the concepts of routing along with traffic splitting and switching using hands-on code recipes with detailed

implementation of various Deployment Strategies like Blue-Green, Canary, and A/B Testing.

**Chapter 8: Knative Best Practices –** explains how to leverage the power of Knative effectively, the problems associated with services deployment, and their solution using GitOps strategy with a Case Study. This chapter covers managing the deployments using GitOps strategy and continuous deployment using ArgoCD. This chapter also covers Observability with the configuration of various tools like Loki, Prometheus, Jaeger, and Grafana using hands-on code recipes.

## Code Bundle and Coloured Images

Please follow the link to download the
***Code Bundle*** and the ***Coloured Images*** of the book:

## https://rebrand.ly/yfiq8gl

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/Build-Serverless-Apps-on-Kubernetes-with-Knative**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

## Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

**errata@bpbonline.com**

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

## Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

## If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

## Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

## Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

**https://discord.bpbonline.com**

# Table of Contents

# Chapter 1
# Serverless and Knative in a Nutshell

## Introduction

Serverless Computing is commonly known as Serverless. The name Serverless Computing or Serverless itself sounds like a paradox, is it not? We all know for a fact that, for any software to run, it needs basic infrastructure - compute, storage, and network. So, does the Serverless claims the ability to run software without any server infrastructure? Certainly not.

This chapter will walk you through the evolution of **Serverless**. We will learn about the two main concepts of Serverless - **Function-as-a-Service** (**FaaS**) and **Serverless Containers**. We will also learn about the Knative project and how it helps to implement Serverless Containers.

## Structure

In this chapter, we will discuss the following topics:

- Introduction to Serverless
- Function-as-a-Service
- Serverless Containers
- Introduction to the Knative project

# Objectives

Upon completing this chapter, you will acquire a comprehensive grasp of Serverless Concepts and the distinctions between FaaS and Serverless Containers. Furthermore, you will be introduced to the Knative project.

# Introduction to Serverless

Over the last three decades, tremendous innovation in server infrastructure has been seen. With the advent of the Internet, enterprises were scrambling to make information available round the clock from anywhere in the world during the last decade of the 20th century. It all started with Physical Servers, also known as bare-metal. As depicted in the following figure, getting a bare-metal and deploying the application on it required a high lead time, a lot of effort, and a huge upfront **Capital Expenditure (CAPEX)**.


***Figure 1.1:*** *Installing application on the physical server*

Almost a decade later, such applications have grown significantly in numbers, and enterprises had to own/manage hundreds of bare-metal servers. The enterprises were looking for a solution that could improve the resource utilization of bare-metal servers. Then came the **Virtual Machine (VM)** technology which allows partitioning of the bare-metal, making it easy to adapt to dynamic workloads by reallocating resources. It also provided opportunities for public cloud providers to offer compute platforms on rent, an offering now popularly known as **Infrastructure-as-a-Service (IaaS)**. As depicted in the following figure, VM technology reduced the lead time to provision server infrastructure significantly and made deploying applications on VM a bit easier, but it still needed a lot of configuration and setup to be done.


***Figure 1.2:*** *Installing application on Virtual Machine*

A few years later, Docker splashed on the scene with the ability to package and run containers. It enabled multiple applications with different OS requirements to run on the same OS kernel as containers and provided an opportunity for further simplification and resource savings. Unlike VMs, containers need significantly less resource footprint, are blazingly fast to spin up or down, and need less overhead to manage. Shortly after, Kubernetes, an open-source container scheduler and orchestration tool, was launched, which now has become a default standard. As shown in *figure 1.3*, with containers, it was made possible to significantly reduce the lead time and scope to be able to deploy an application:


***Figure 1.3:*** *Installing the application as a container*

Then came the concept of Serverless, which challenged the notion of the need for continuously running servers to serve the applications. Serverless provided an architectural approach for ephemeral infrastructure to serve an application that can come into existence on an incoming request and disappears immediately after serving it. Serverless allowed application developers to focus only on the application code delivering business value without worrying about underlying infrastructure, capacity planning, scalability, and so on. As depicted in *figure 1.4*, Serverless approach improved lead time drastically to build and deploy an application:


***Figure 1.4:*** *Installing application on serverless platform*

As we know, software applications seldom have constant load, and it keeps changing with time. So, we tend to either over or under provision the infrastructure. As shown in *figure 1.5*, physical servers and virtual servers provide very less elasticity, the provisioned infrastructure remains constant. This leads to wastage of idle resources when load is less (overprovisioning) and poor quality of service/lost business

revenue in case of excess load (under provisioning). Please refer to the following figure:

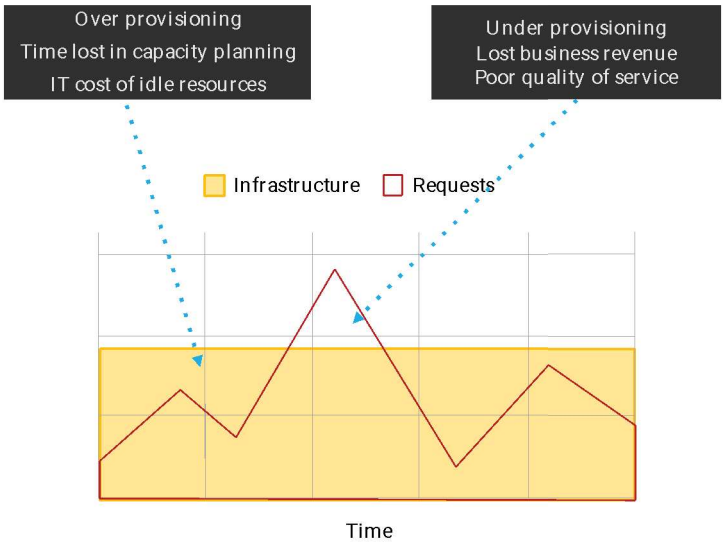


***Figure 1.5:*** *With bare metals and VMs*

As shown in the following *figure 1.6*, Cloud computing, with its autoscaling feature, tries to solve this problem to some extent. Since scaling of VM infrastructure (up or down) takes few minutes, the minimum server capacity must be kept active all the time, which leads to the idle resources even when load is less. Please refer to the following figure:

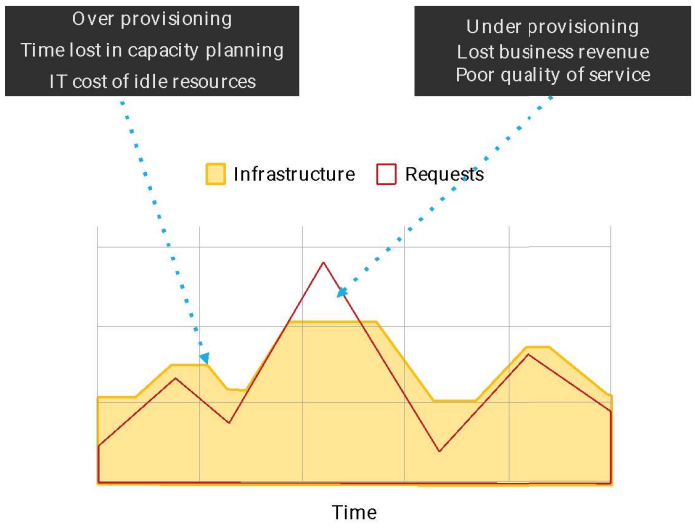


***Figure 1.6:*** *With auto-scaling on cloud*