Chapter 6
Containerization:
In the Context of Serverless Computing

Sai Samin Varma Pusapati
Chaitanya Bharathi Institute of Technology, India

ABSTRACT

This chapter delves into the synergy between containerization and serverless computing, pivotal for advancing cloud-native application deployment. It outlines the architectural foundations and benefits of each paradigm, emphasizing their combined impact on scalability, efficiency, and agility. The discussion progresses to technical integrations, focusing on container orchestration and serverless platforms, enhancing management and deployment. Addressing challenges like security and operational complexity, it highlights strategies for navigating these issues. Real-world examples illustrate the practical application across sectors, showcasing the integration's capacity to meet diverse computational needs. This convergence is posited as a significant driver for future cloud-native innovations, offering a glimpse into evolving trends and the potential reshaping of software development landscapes. The exploration underscores the critical role of this amalgamation in optimizing resource utilization and simplifying cloud infrastructure complexities.

1. INTRODUCTION

The landscape of software development and deployment is continuously evolving, with containerization and serverless computing emerging as key drivers of change. These technologies have significantly influenced how applications are developed, deployed, and managed, offering novel paradigms that enhance efficiency, scalability, and flexibility. Containerization is a technology that has transformed the landscape of software deployment by encapsulating applications and their dependencies into isolated units called containers. Unlike traditional virtualization, which virtualizes an entire operating system, containers virtualize at the application level, providing a lightweight and efficient solution for packaging, distributing, and running software. At its core, containerization involves bundling an application along with its dependencies, libraries, and runtime into a single container. By definition, a container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly

DOI: 10.4018/979-8-3693-1682-5.ch006

and reliably from one computing environment to another. Docker, one of the most popular containerization platforms, played a pivotal role in popularizing this technology. Serverless computing, in contrast, abstracts the infrastructure layer, enabling developers to focus on code rather than server management. This model optimizes resource utilization and operational costs, representing a shift towards more dynamic and cost-effective cloud computing services.

This chapter aims to explore the contemporary relevance and application of containerization and serverless computing within the software development ecosystem. It will delve into how these technologies synergize to streamline deployment processes and facilitate the development of scalable, resilient applications. By examining the current state of these technologies, including their benefits, challenges, and best practices, the chapter seeks to provide a comprehensive understanding of their role in modern software development.

Moreover, the discussion will extend to the integration of containerization with serverless computing, highlighting how this combination leverages the strengths of both paradigms to offer an unparalleled level of efficiency and agility in application deployment. The chapter will also address the challenges that accompany the adoption of these technologies, such as security concerns and complexity in management, and will propose strategies and solutions to navigate these obstacles effectively. The objective of this chapter is not only to elucidate the foundational concepts of containerization and serverless computing but also to showcase their practical implications in fostering a more agile, cost-effective, and scalable approach to software development. Through this exploration, readers will gain insights into leveraging these technologies to their full potential, thereby contributing to the advancement of cloud-native applications and the broader software development landscape.

In summary, the chapter will serve as a detailed guide to understanding the intricacies of containerization and serverless computing, providing valuable knowledge for developers, IT professionals, and organizations aiming to navigate the complexities of modern software development and deployment. Through a blend of theoretical exploration and practical insights, it will illuminate the path toward harnessing the transformative potential of these technologies in the digital age.

2. FUNDAMENTALS OF CONTAINERIZATION

2.1 Definition and Concepts

In the rapidly evolving domain of software development and deployment, container technology has emerged as a pivotal innovation, significantly altering how applications are created, deployed, and managed across diverse environments. This technology, by encapsulating an application and its dependencies into a single container, offers a streamlined and efficient approach to software delivery, distinguishing itself from traditional virtualization techniques through its lightweight nature and agility.

Container technology provides a standardized unit of software, ensuring that applications run reliably and consistently regardless of the deployment environment. This consistency addresses a common challenge in software development known as the “it works on my machine” syndrome, where applications behave differently across various environments due to discrepancies in operating systems, libraries, and dependencies. Containers encapsulate the application along with its runtime environment, making it possible to achieve uniformity across development, testing, and production stages. Docker, introduced in 2013, has been instrumental in popularizing container technology, providing a platform that simplifies