



White paper

Moving from Cloud Foundry

A smooth transition strategy

Table of contents

| | |
|--------------------------------------------------------|---|
| Abstract | 3 |
| What makes Cloud Foundry popular | 3 |
| Why move from Cloud Foundry then | 3 |
| Where to go | 3 |
| Kubernetes market trends | 4 |
| Developers still see Cloud Foundry | 5 |
| Understanding KF architecture | 6 |
| Beyond KF - preparing for a full Kubernetes transition | 6 |
| CF to KF migration challenges | 6 |
| Final thoughts | 7 |
| Reference | 7 |

Abstract

Organizations are always looking for ways to improve the efficiency, scalability and manageability of their application deployment strategies in the quickly changing cloud-native ecosystem. The transition from Cloud Foundry (CF) to Kubernetes for Enterprise (KF) signifies a significant change in favor of a platform that is more adaptable, durable, and expandable. This white paper explores the strategic factors, obstacles and detailed instructions required for a seamless and effective transition. It attempts to provide IT specialists, DevOps teams and decision makers with the knowledge and useful guidance required to manage the challenges of moving to KF, guaranteeing little downtime and optimizing the advantages of the cloud-native paradigm.

What makes Cloud Factory popular?

- Cloud Foundry is a platform-as-a-service (**PaaS**).
- It simplifies the approach to deploying and operating **stateless applications**.
- Operators can configure and secure platform capabilities in a **single location** for all users.
- Cloud Foundry uses “**service brokers**” to provide applications.
- It takes an **application—first approach**, hiding complex implementation details.
- Developers love the streamlined workflow. CF uploads app binaries, compiles them into container images via buildpacks, adds HTTP routes, connects services and deploys the app in under 30 seconds with a single command.

Why move from Cloud Foundry then?

- Open source Cloud Foundry **lacks dedicated customer support**, relying on community forums and documentation, which might not be as responsive or tailored to specific organizational needs as commercial support.
- While Cloud Foundry has a strong community and ecosystem, it's not as large or active as the other platform communities. This can mean **fewer third-party integrations**, less community support, and slower development of new features.
- Pivotal licensing **costs increase** significantly as deployment footprint increases
- It is **less flexible** than other platforms. It prescribes specific ways of doing things, which can be limiting if you have unique requirements that don't fit within its model.

Where to go

The most popular alternative to Cloud Foundry is “Kubernetes”

Here's why:

- Kubernetes is not as opinionated as Cloud Foundry. This means you have **more flexibility** to define your application's deployment and operation according to your specific needs.
- Kubernetes has a very active community and a **large ecosystem** of related projects. This means you have access to a wealth of knowledge and resources, as well as a **wide range of tools and integrations**.

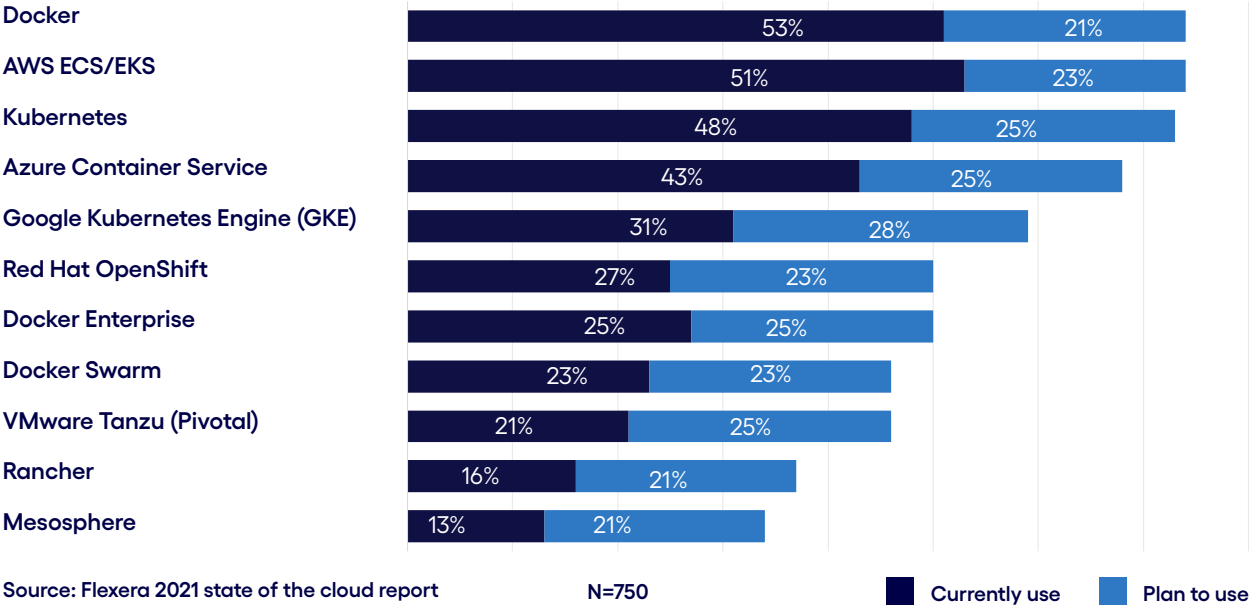
- Kubernetes itself is free and open source. While there are costs associated with running Kubernetes (such as the cost of the infrastructure it runs on), these can often be **less than the licensing costs** for commercial Cloud Foundry distributions.
- Kubernetes can run on almost **any infrastructure**. Whether you're using public cloud, private cloud or on-premises servers, you can run Kubernetes.
- Many cloud providers offer managed Kubernetes services, which can **reduce the operational burden** of running Kubernetes. These include Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (EKS) and Azure Kubernetes Service (AKS).

Kubernetes market trends

- Kubernetes has rapidly become a cornerstone in the world of container orchestration, with its adoption soaring among enterprises. As of recent reports, over 60% of organizations have adopted Kubernetes, with the Cloud Native Computing Foundation (CNCF) noting a rise in **adoption rates to 96%**. The platform's market size was valued at USD 1.46 billion in 2019 and is projected to grow at a compound annual growth rate (**CAGR**) of **23.4% by 2031**. This growth is driven by its scalability, flexibility and automation capabilities, making it essential for modern business operations. The United States leads in adoption, accounting for over 50% of users, with significant usage also seen in the UK, India, Germany, France, Canada, and Brazil. Kubernetes' value is expected to reach **USD 9.69 billion by 2031**, reflecting its critical role in managing and deploying containerized applications across various industries.

Container tools used

%of all respondents



The **2024 Kubernetes Benchmark** Report by Fairwinds analyzes over 330,000 Kubernetes workloads from hundreds of organizations. It highlights significant improvements in workload efficiency and reliability, though areas for improvement remain.

Start migration with KF

For enterprises embracing Kubernetes, migration from Cloud Foundry is the biggest challenge. KF built by Google acts as a front end for a Kubernetes cluster and it offers the Cloud Foundry experience while empowering operators to adopt declarative Kubernetes practices. It makes migrating Cloud Foundry workloads to Kubernetes straightforward, and most importantly, avoids major changes to developer workflows.

Using KF has several advantages:

- KF provides a **Cloud Foundry-like developer experience** on top of Kubernetes. This means developers can continue to use familiar CF commands and workflows, reducing the learning curve.
- KF **simplifies the process** of deploying applications to Kubernetes, making it more accessible to developers who may not be familiar with all the intricacies of Kubernetes.
- KF is built on top of Kubernetes, so you can take **full advantage** of the scalability, reliability, and ecosystem of Kubernetes.
- KF is designed to be cloud-agnostic. It's built on top of Kubernetes, which is a platform-agnostic container orchestration system. This means that KF, like Kubernetes, can run on any infrastructure where Kubernetes can run. This includes various cloud platforms like Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure, and even on-premises servers.

“Using KF, little to no changes to the existing source code and workflow.”

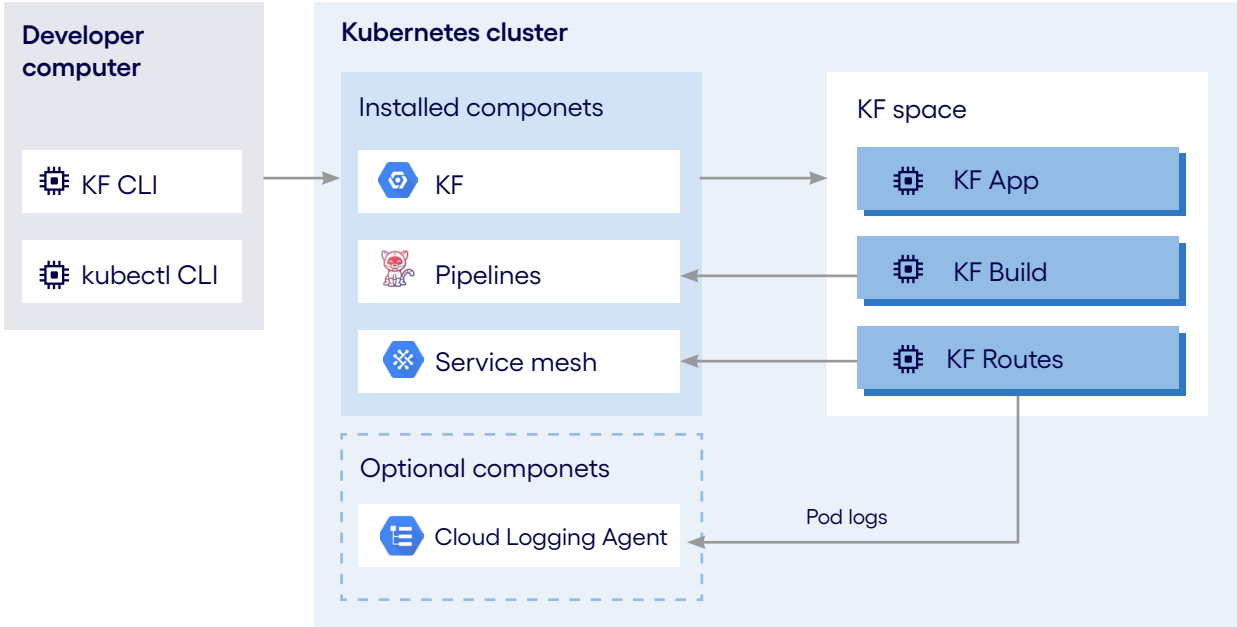
Developers still see Cloud Foundry

but underneath it's **Kubernetes**.

| | CF cli command | KF cli command |
|---------------------------------------------------------------------------|-------------------|-------------------|
| Deploy an app (no code changes or Kubernetes YAML required). | CF push | KF push |
| Creates a new service instance | CF create-service | KF create-service |
| Tail or show logs for an app (stream from Kubernetes to user's terminal). | CF logs | KF logs |
| List spaces on this Kubernetes cluster. | CF spaces | KF spaces |
| List available offerings in the marketplace. | CF marketplace | KF marketplace |
| List services in the targeted space. | CF services | KF services |
| List pushed apps (extract and display Kubernetes deployment metadata). | CF apps | KF apps |
| Binds a service instance to an application | CF bind-service | KF bind-service |
| Deletes an application | CF delete | KF delete |

Understanding KF architecture

KF is constructed upon a microservices architecture utilizing multiple open-source projects and Kubernetes. It has its own unique set of Kubernetes custom resources and controllers and serves as the KF API. This collection of resources includes apps, builds, routes, service brokers, service instance bindings, service instances and spaces. Different tasks, like managing the application's configuration, state and source code, are handled by each component. These components serve to harmonize and coordinate other parts of the system.



source: <https://kf.dev/docs/v2.11/operator/kf-dependencies/kf-components-diagram.svg>

Beyond KF—Preparing for a full Kubernetes transition

The goal of KF is to act as a transitional step before a full switch to Kubernetes or another container runtime. It enables the platform team to become more accustomed to and at ease with Kubernetes management. Your application developers can improve their Kubernetes environment abilities in the interim. Changing your software build process is the first step in switching from KF to another container runtime. You will be responsible for managing this process yourself, rather than depending on KF to build your containers. It's crucial to remember that Cloud Foundry best practices may not always align with container runtime best practices. This means that you might have to modify both your organizational and technical processes.

CF to KF migration challenges

- Since applications made for CF may need to be significantly modified to fit the container-based architecture of KF, the complexity of application rearchitecture poses a significant challenge and could cause downtime during the transition.
- Another significant obstacle is the learning curve and skill set shift. Teams used to CF's abstracted environment will need to learn the intricacies of Kubernetes, including its orchestration, networking and security models. This will require training new hires or investments in talent and may temporarily lower productivity.

- It could be difficult to redesign and modify CI/CD pipelines for KF because DevOps and CI/CD practices developed for CF might not translate directly to KF. This could cause delays in deployment cycles.
- When switching from CF's route-based networking to Kubernetes' load balancing and service discovery, networking and service discovery must be adjusted, which could cause service interruptions.

Final thoughts

In conclusion, the shift from Cloud Foundry to Kubernetes, facilitated by KF, represents a significant transformation that requires careful planning and execution. While Cloud Foundry has been popular due to its simplicity and opinionated workflow, it has certain limitations that can be addressed by moving to a more flexible and vibrant platform like Kubernetes. The migration process, although complex, can be made smoother and more manageable by using KF as an intermediate step. This approach allows organizations to leverage the strengths of both platforms, ensuring a seamless transition that caters to their specific needs. As we navigate this journey, it's important to remember that technology migration, much like upgrading a home, is essential for comfort and growth.

About the author



Chandrahasan Subbaiyan

Sr. Architect

Reference

- <https://kf.dev/docs/v2.11/getting-started/>
- <https://cloud.google.com/migrate/kf/docs/2.11>
- <https://www.cncf.io/blog/2024/01/26/2024-kubernetes-benchmark-report-the-latest-analysis-of-kubernetes-workloads/>
- <https://github.com/google/kf>
- <https://kf.dev/docs/v2.11/examples/>
- <https://www.youtube.com/watch?v=Mmvjg2Dtexc>
- <https://cloud.google.com/anthos?hl=en>
- https://www.youtube.com/watch?v=gWv7_JT-yD4
- <https://edgedelta.com/company/blog/kubernetes-adoption-statistics>



Cognizant helps engineer modern businesses by helping to modernize technology, reimagine processes and transform experiences so they can stay ahead in our fast-changing world. To see how Cognizant is improving everyday life, visit them at www.cognizant.com or across their socials @cognizant.

World Headquarters

300 Frank W. Burr Blvd.
Suite 36, 6th Floor
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277

European Headquarters

280 Bishopsgate
London
EC2M 4RB
England
Tel: +44 (0)1 020 7297 7600

India Operations Headquarters

5/535, Okkiam Thoraipakkam,
Old Mahabalipuram Road,
Chennai 600 096
Tel: 1-800-208-6999
Fax: +91 (0) 44 4209 6060

APAC Headquarters

1 Fusionopolis Link, Level 5
NEXUS@One-North, North Tower
Singapore 138542
Tel: +65 6812 4000

© Copyright 2024, Cognizant. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the express written permission of Cognizant. The information contained herein is subject to change without notice. All other trademarks mentioned here in are the property of their respective owners.