

[CLOUD NATIVE ECOSYSTEM](#) / [DEVOPS TOOLS](#) / [KUBERNETES](#) / [SPONSORED](#) / [CONTRIBUTED](#)

When to Choose Cloud Foundry over Kubernetes

2 Dec 2021 8:28am, by [Julian Fischer](#)



[anynines](#) sponsored this post.

Kubernetes (K8s) is a fascinating piece of technology. It can run a variety of different workloads. However, Kubernetes is not the only option.

A closer look at Cloud Foundry (CF) will reveal that for several use cases, a Cloud Foundry environment may be preferred over Kubernetes. This article highlights Cloud Foundry for VMs (BOSH), in contrast to CF-4-K8s which is worth a dedicated comparison.

Cloud Foundry

Julian Fischer



Julian is CEO of anynines and has dedicated his career to the automation of software operations. Within the field of platform automation, Julian has a



While the idea of [Kubernetes was to be a platform to build platforms](#), the concept of Cloud Foundry is to provide a turnkey application developer platform solution.

With Cloud Foundry, it is possible to set up a platform environment on any major infrastructure, whether public or on-premises. Especially when its life cycle is managed using BOSH, Cloud Foundry shows its strengths: running thousands of applications for hundreds of tenants in a single, large environment.

In such a Cloud Foundry environment, the update of operating systems, as it has been necessary after the [Shellshock exploit](#), is as simple as changing a few lines of YAML code. The update will then be executed as a wave of reconstructing virtual machines across the environment when using [BOSH-based data service automation](#). Often these environments comprise thousands of virtual machines hosting apps from different tenants. Cloud Foundry tenants, also called “organizations,” are regularly used to represent organizational units such as internal business units. Because of Cloud Foundry’s ability to isolate tenants, large environments are possible while maintaining a centralized platform, unfolding a significant economy-of-scale effect.

A Cloud Foundry platform environment initially comes with a relatively large infrastructure footprint. Generally speaking, the overhead of a Cloud Foundry environment depends on the level of redundancy and thus the desired availability of the resulting platform.

Once a Cloud Foundry environment has been set up, it provides several key services required by any application development platform, including user management and authentication services comprising the ability to create organizations and “spaces,” a container orchestrator to schedule workloads across the cluster, the ability to deploy code using buildpacks or container images as well as a central service marketplace.



fashion.

Assuming there is a list of service brokers available, application developers can instantly deploy applications from local code, create database instances such as a [PostgreSQL streaming cluster](#) and bind apps to the database(s). The developer's user experience is based on well-defined APIs, and workloads can be described using the CLI or by applying YAML specifications.

Simply speaking, Cloud Foundry provides enterprise-grade hosting, enabling a high degree of automation and the ability to enforce organizational policies.

The usage of buildpacks is an excellent example of how an organization can decide whether to allow local decisions in engineering teams or instead enforce the use of quality-assured provisioning automation by either allowing arbitrary buildpacks or restricting the choice to a list of well tested, security checked buildpacks.

The CF marketplace is another example of central governance. The selection of service brokers for data service automation and offering other services provides fine-grained control over how data is treated. Engineers do not need to select, install and life-cycle manage operators. All they have to do is select a data service, choose a service plan and create a service instance. It is the job of the service broker to allow comprehensive life-cycle management, including configuration changes, version upgrades, scalability, handling of logs and metrics, to name a few tasks.

Kubernetes

Vanilla Kubernetes provides namespaces, a concept used to separate workloads within a Kubernetes cluster. However, namespaces do not isolate as well as Cloud Foundry organizations, nor does Kubernetes provide a user-management and authentication service out of the box.

If used as an application developer platform, Kubernetes environments tend to have a different structure than Cloud Foundry environments. In particular, the number of Kubernetes clusters is much higher than the number of Cloud Foundry instances. While

operate than a single Cloud Foundry, managing 100 Kubernetes clusters is several orders of magnitude harder than managing a single Cloud Foundry. This scenario might appear extreme, but it shows the underlying problem.

Managing a large number of Kubernetes clusters creates complexity that is not necessary with Cloud Foundry. Admittedly, there is room for a dialectical discussion. Workloads must be compatible with Cloud Foundry, which is more demanding toward [12-factor compliance](#) than Kubernetes. Kubernetes can be customized much more than Cloud Foundry and therefore accept workloads that Cloud Foundry couldn't. Though this argument also makes a point for Cloud Foundry, which is a crucial statement to be pointed out: If an organization runs large swaths of Cloud Foundry-compatible workloads, Cloud Foundry might represent a better operational economy.

If workloads are more heterogeneous, Kubernetes wins, and increased operational complexity is to be tolerated.

Sponsor Note

anynines is focused on planning, building, and operating modern application developer platforms, as such anynines has years of experience in automating the delivery of enterprise-grade platforms based on various technologies. This comprises platform technologies such as Cloud Foundry and Kubernetes.



Cloud Foundry vs. Kubernetes

Although Kubernetes can be configured, extended and tweaked to approximate the behavior of Cloud Foundry, Cloud Foundry remains unique in its ability to manage large global workloads. The economy of large-scale Cloud Foundry environments is excellent. It remains a valid choice for organizations that primarily run 12-factor-compliant applications aiming to provide their developers a maximum degree of automation.

As an opinionated technology, Cloud Foundry is best to build centralized application development platforms. In its classic variant, using BOSH and virtual machine
nation, it is a rock-solid choice to create large platform environments with

be forgotten when the economy-of-scale effect kicks in, allowing a small operations team to handle huge platform environments across multiple public and on-premises infrastructures.

Especially if large organizations need to develop and operate many applications across infrastructures with a homogenous user experience, Cloud Foundry is a technology that should be considered.

Kubernetes, on the other hand, has become the de facto standard for containerizing workloads. Whenever a high degree of flexibility is required, Kubernetes is the way to go.

The New Stack is a wholly owned subsidiary of Insight Partners, an investor in the following companies mentioned in this article: Cloud Foundry.

Featured image via [Pixabay](#).

CONTRIBUTED

SPONSORED

THE NEW STACK UPDATE

A newsletter digest of the week's most important stories & analyses.

We don't sell or share your email. By continuing, you agree to our [Terms of Use](#) and [Privacy Policy](#).

RELATED STORIES