

About Stark & Wayne

Stark & Wayne is the leading cloud agnostic technology consulting firm, offering: architecture design, implementation, and operations enablement. These services empower enterprises to operate highly available and extensible platforms on their chosen infrastructure.

Stark & Wayne engineers bring years of experience and a strong passion for cloud development and automation. We have the expertise to help your organization maximize the benefits of Cloud Foundry or any type of multi-cloud you might need, on premises or in the public cloud.

Contact

Brett Kennedy
VP Professional Services

M: (716) 998-8632
E: bkennedy@starkandwayne.com

Cloud Foundry is for Small Business Too

Jared Placek is Chief Engineer at DemandBridge and leads the software development effort for DemandBridge Enterprises as well as DB Alliance and DB Response Platform. This is his own story about how DemandBridge opted to use Open Source Cloud Foundry along with developing a partnership with Stark & Wayne to enable his small team to build incredible software.

Brief history

DemandBridge is a very small technology company with a large, niche customer base in the print distributor industry. Our team builds and maintains a B2B e-commerce platform focused on marketing and brand management that has a 20-year code base.

Problems at hand

The problem we faced a few years ago starts the same way as most of the Cloud Foundry adoption stories we hear at these conferences:

- Large, outdated, monolithic application
- Long, painful deployments
- Need to scale, evolve, move faster, risk less, deploy more

Here's where we deviate - and this is not something we tout, but rather something that had historically made us feel like cutting edge development technology and practices were plainly out of reach for us:

- 4 application developers
- 0 dedicated ops (responsibility shared among the 4 application developers)

We were, for a long time, the software development arm of a large sales-driven print distributorship and the most we could hope for from a technical infrastructure standpoint was that our application server and database VMs were powered on and connected to the internet. The rest was up to us.

Early evaluation of Cloud Foundry

Due to database and app server licensing, our monolith was stuck on Java 1.5. While our team size and resources were fairly limited, we knew how important it was to keep up with the industry and at least understand what was going on, even if we couldn't embrace it.

We attended the annual Spring Conference and would hear about the growth of Spring Boot.. it sounded like a breath of fresh air, but we couldn't find a use for it in our environment.

Pivotal, as host of the Spring Conference for many years, started talking about Cloud Foundry and again we were impressed, but our WebLogic server loomed large and unforgiving and we sat there like a bunch of Charlies watching everyone else open Wonka bars.

After a while, we opened our eyes to the fact that we were sitting stuck on a rapidly moving timeline, with the latest features and frameworks moving further and further out of reach and a maintenance nightmare of outdated software and infrastructure piling up behind us. We reached out to Pivotal and they were kind enough to send a few people over for a technical discussion that answered a few of our questions and left us with a lot more.

Here's where we meet up again with the stories we've heard over and over for the past few years:

Problem: The monolith

Solution: Containers? Microservices?

We started to see the light and, particularly with the introduction of the Spring Cloud suite, we could wrap our heads around breaking pieces of our code out from the core application and embrace, as much as possible, the idea of 12-factor and a more distributed architecture.

One of our developers, Kim, built a proof-of-concept Spring Boot app and we were excited about its potential but we were, once again, at a loss on where to put it. For a while it ran as a lone Java process on an otherwise empty VM, but that was not sustainable and the idea of running and maintaining even a second app like that, let alone a whole network of microservices, made us panic.

Price tags were a factor

Pivotal, the most prominent Cloud Foundry vendor at the time, seemed to be positioning PCF for large enterprises. The number of pricing tiers for smaller companies was limited and, more importantly, each tier encompassed a much wider range than we were prepared to commit to. We worried about having to make tough architectural decisions when nearing app-instance thresholds rather than the more steady linear scaling we were hoping for.

We spent a few months toying with plain old Docker and the general container methodology was a no-brainer, but that only abstracted our problem of where to host these applications.

We knew we were living on borrowed time without an operations team; but, given that "operations team," in our case, meant a single resource, we wanted to make sure to find the most effective use of their time before we even began to look for a human being to fill the role.

Docker seemed great - we procured a large VM to host our containers and threw a bunch at it. We quickly realized that we were just heading for another cliff without a clustered solution. We weighed the pros and cons - resilience vs. ease of use, complexity vs. cost, orchestration vs. being able to actually setup the environment in our lifetime.

Then we got an invitation to the CF summit

We saw all the big platforms represented, not just Pivotal, and really felt the undertone of the open source community.

We decided if we were going to end up having to try tackling something like Mesos or Kubernetes, we might as well see if open source Cloud Foundry was another option - so we hopped on a plane to do our due diligence.

The first CF Summit we attended was eye opening - the stories of organizational transformations and the idea of continuous delivery enabled by Cloud Foundry was inspiring. The seamless integration with Spring and the whole concept of 'CF Push' was music to our ears. Toward the end of the keynote there was a joint announcement by the Cloud Foundry Foundation and Microsoft about support for the Azure hypervisor. Perfect - Azure is our cloud provider and Cloud Foundry will be our salvation. Everything was adding up and halfway through the first day we decided, confidently, that open source Cloud Foundry was our future. We just weren't sure how to get there.

We walked around the vendor hub and asked as many questions as possible. We made a beeline for the Azure booth to talk about how best to get started. The rep was excited at the prospect:

Vendor: "With Azure it's simple to deploy Cloud Foundry! Just click the 'install PCF' button."

DemandBridge: "Oh, no, sorry. We're interested in an open source deployment."

Vendor: "Oh... yikes."

I doubt she actually said "yikes," but I swear I heard something like that at least once that day. The general feedback we got from the numerous booths we attended was "good luck" and "that's really hard !"

We spent the rest of the day trying to absorb as much as possible and in the evening, back at the hotel, we banged away at BOSH-lite hoping for something to click. The documentation was thorough but we were obviously jumping head first into the deep end.

Engaging with S&W to build our Cloud Foundry

Dr. Nic was the emcee of CF Summit that year and the most common sight across the venue were the red Stark & Wayne shirts. They seemed like an authority and who better to ask the dozens of questions we'd come up with the night before? We peppered them at the vendor hub and the evangelist booth and their patience was much appreciated.

A lot of light bulbs went on, but the biggest realization we came to was that we needed a partner with Cloud Foundry expertise if we were going to really do this.

Still suffering from the feeling that we were too small to relate to the big players in the room - the types of organizations who would have their own personal Pivotal dojos - we decided that we wanted to find "a Stark & Wayne for the little guys" who could cater to an organization as small as ours. Months later, re-telling this story, this would turn into a joke, and a very brief moment of panic on their part - "What are we doing wrong with our marketing that made you want to look for someone 'like Stark & Wayne' and not just reach out to us?"

Long story short, after a bit of research and a few weeks of walking in circles, we did reach out to Stark & Wayne and they could not have been more accommodating. They helped clarify cost (open source != free), gave us a wide array of options and showed off not only the power of the Stark & Wayne Collective, but the open source CF community as a whole. They helped us confirm that for our team and our own product vision an open source Cloud Foundry installation was ideal - with their help, of course. Two of their guys flew in and immersed our whole team in BOSH and CF for a week, sharing best practices while building our Cloud Foundry environments (among other things - Concourse, Safe/Vault, various community plugins, etc.) and working out a plan for the ongoing maintenance and evolution of our platform. When they handed over the keys at the end of the week it was not "goodbye and good luck", but, rather "here we go, together."

And they kept a key too! Actually, I think the one they gave us was just for show. Just kidding - we use it all the time.

Daily use

So now we had Cloud Foundry.

We threw our proof of concept apps at it - no sweat.

We built out our Spring Cloud baseline infrastructure - simple, up and running in minutes.

The anecdotes from the keynote speeches about the magic of 'cf push', the ease of managing routes, binding services, scaling app instances in several directions and not having to worry about keeping them running once they're in Cloud Foundry's hands (which we have to remind ourselves not to take for granted) - they were all real and worked exactly as advertised.

There was a wave of emotion that hit more than just me when we realized that the heart of our application infrastructure was the same as these big organizations who were out there doing amazing things and, with Stark & Wayne helping to shore up our backbone, the difference between us and them now was, essentially, the number of Diego Cells.

The most important thing that Cloud Foundry offered us was the ability to have our small development team run full-speed ahead, building tons of new features into our product while tearing out older chunks from our monolith and rebuilding them. We were no longer limited by our infrastructure - the latest features of Java, Spring, and many other frameworks were at our fingertips and we didn't hesitate to put them to use. We were strangling our monolith before we ever heard the term used.



With so many years of code to rewrite and restructure, and with so few people to do it, we have a long way to go before we're able to completely shut down our WebLogic server, but we're no longer tossing buckets of water out of a slowly sinking ship - every sprint we throw new applications into production and our code base and application architecture is vibrant and exciting to work with. That's a complete 180 from 4 years ago.

Final thoughts

So we're a few years in now and we have a much different appreciation for the keynotes and user stories we hear at CF Summits and Spring Conferences. We're users now, not spectators. We've been through similar transformations, albeit at a smaller scale.

Here's how we stack up against our former selves:

Then:

Large, outdated, monolithic application
Long, painful deployments
Need to scale, evolve, move faster, risk less, deploy more

Now:

Smaller (still large), outdated, monolithic shell of an application
One (occasional), long but much less painful deployment when necessary
83 cf app instances (53 unique applications)
Many rapid, painless deployments to CF throughout our sprint cycle
We are scaling, evolving, moving faster, risking less, and deploying more

Then:

4 developers
0 (dedicated) ops

Now:

4 developers
10k+ ops
1 primary resource at Stark & Wayne
35 members of the Stark & Wayne Collective
10k+ Cloud Foundry community members

We work with Stark & Wayne daily, reach out to the Collective on a regular basis, and through Stark & Wayne, have contributed to, brainstormed with, and found answers from the community on many occasions.



What's next ?

We're running full-speed now, but still have a lot of work to do.

There's still a lot of monolith to rip apart and rebuild and our network of CF applications will continue to grow rapidly. We also want to encourage the rest of the company to use Cloud Foundry wherever possible, whether it's rewriting apps to be part of our Spring Cloud ecosystem, building additional ecosystems, or even embracing CFCR for non-standard applications and to continue the road to full containerization.