The future is easy, naming things is hard.

A brief look back at the beginning of Cloud Foundry
Derek Collison
Founder and CEO at Apcera

- CTO, Chief Architect at VMware
- Created and Architected Cloud Foundry
- Created NATS (nats.io)
- Technical Director at Google
- SVP and Chief Architect at TIBCO

A bit about me
Motivation
Project B29

- Started October 2009
- Two day design
- Originally Vadim Spivak and I coding
- 3 months to first prototype shown to Paul and Steve
- Name came from Mark Lucovsky, mimicking MS
First pass architecture
Early Days

https://www.youtube.com/watch?v=Tis1HuvIT4A
Some Fun Facts

• Was not part of SpringSource

• Project B29 - Cloud Foundry was not the first name choice

• DEA was named by me at 2a, by joking about droplets in a cloud - Sorry

• Vadim started coding in Java but grew frustrated at speed of development so switched to Ruby

• First open source project from VMW

• Original CLI was named vmc, written by Oleg (he did Bosh too)
Some More Fun Facts

• First customer was Salesforce with VMforce
• RabbitMQ was selected first as the messaging backplane
• First app was not Java or Spring, but a Sinatra App
• Added in Node.js and MongoDB support weekend before launch
• First project to embrace AWS at VMW
• Monitoring app built in last week before launch
Monitoring - 2010
What we got right

- Apps and Services, not VMs, machines, etc.
- Reduce opex spend, speed deployment
- Distributed systems architecture
- Run on any infrastructure
- Stem-cells
- Opensource
What we got **wrong**

- Ruby as implementation language
- All services need to be programmed
- Layer 7 ingress only
- Did not think hard enough about security and trust
- Too opinionated?
- Open-sourced BOSH
What you all got right

• Great ecosystem
• Well run, independent and powerful foundation
• Developer tooling
• Spring Boot and the Netflix model
• Micro-services and Cloud Native mantra
That’s all great

BUT..
Where do we go from here?
Making sense of the chaos!
Innovation is Everywhere
BUT...

• Inconsistent Interfaces
• Inconsistent Boundaries
• Lacking Trust
The Buckets of Technologies
Three Buckets

The “Chaos”

Infrastructure Provisioning

Workload Orchestration

Artifact to Workload
Apply the 80/20 Rule
Standardization of interfaces is crucial
Verticalization will fail

- Need specialization
- More options are better than few
- Vendors should not have to implement the complete stack
- Interoperability
The Call for an Open Cloud Ecosystem
Open Cloud Ecosystem

- Intent descriptions
- Container runtimes
- Workload Image Formats
- Orchestration and Deployment
- Storage and Network
- Policy and Governance
Time to work together..
Thanks
Q&A