



# Misadventures in Replatforming

Chris Lauer

Former Air Force Kessel Run Engineering Practice/Technical Product/Project Lead\*

\* Opinions are my own, and do not represent the above organization or my current employer

# Bottom Line Up Front (BLUF)

Building, scaling, and operating an orchestration platform for accelerating DevOps in the DoD is...

Difficult.

Challenging.

HARD!!!



# The Big Picture

- \* Kessel Run is the Air Force's attempt to adopt modern methods of software delivery
- \* Early scaling with deep enablement from Pivotal Labs, on Pivotal Cloud Foundry (TAS)
- \* HUGE scaling with HUGE challenges
- \* Air Gapped Production: DoD Level Process, Compliance, Security.

APRIL 29, 2021

## Kessel Run: The US Air Force's Digital Journey in 12 Parsecs or Less

By IT Revolution





# What is a DevSecOps Platform?

A platform:

- ⌘ Abstracts away infrastructure from the software it hosts
- ⌘ Accelerates software delivery by solving cross-cutting concerns (helps teams DevOps!)

In user story format: As a **developer**, I want to **deploy my application** so that **the warfighter can use it**

Our Key Differentiating Features: Security, Compliance, Access, Disaster Recovery

# Year 1: Accelerated Chaos



# Early Efforts

Just do it!

Two Airmen (military!) deployed Rancher Kubernetes Engine RKE in our air gapped environment to:

- \* Prove it could be done
- \* Give them a platform for the tools they needed

This was huge!



# Early Efforts

Home brewed bespoke vanilla Kubernetes: Platform Container Orchestration (PCO).

- ⌘ We've got commercial-off-the-shelf (COTS) products at scale
- ⌘ ... without vendor or community support.
- ⌘ ... or governance/compliance

But it works!



# September 2020: My Two Jobs

Or: The best seat in the house

Keep VMWare Tanzu Application Services (TAS) alive with a small team



**CLOUD FOUNDRY**

Retire TAS with a large blended team



**kubernetes**

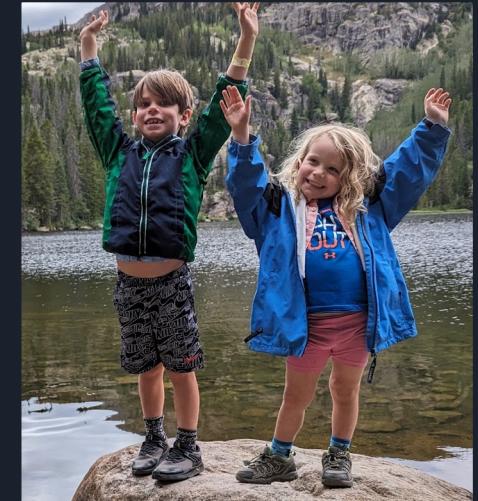
# Let's Partner!

VMWare Tanzu Kubernetes Grid (TKG)! Sounds good.

But... no Architectural Decision Record (ADR) existed.

I had to solve the mystery: Why TKG?

How do I know if it's time to change this decision?



# TKG Struggles

TKG and KR deployment opinions clashed. Not much buy-in to TKG from the rest of team.

We did have cloud clusters: Dev, staging, and prod serving application developers.

Struggling to get on-prem, and the clock is ticking.

## Time to Pivot!



# A New Hope: RKE2

Rancher Kubernetes Engine 2, AKA Rancher Federal

Much better match for Kessel Run's governance and compliance

Better buy-in and alignment from leadership and engineers





# Four Already?!?!





# Some Lessons About Decision Making

## For the Top:

- ⌘ Decide in the open
- ⌘ Document the process and the outcome
- ⌘ Be clear about what might change and what definitely will not

## For the Bottom:

- ⌘ Keep asking questions until you understand why
- ⌘ Write that why down for everyone
- ⌘ If you build something useful, it'll stick (be prepared for that)



# Year 2: Slow Order





# Migration deadline (Mistakes!!!)

PCF (TAS) and TKG licensing expire in 10 months, what do we do?

- 1. Freeze old PCF TAS platform: reduce load on teams, cease adding new capability.  
Clearly communicate the deprecation deadline.
- 2. Keep the two platforms separate so they don't get all tangled up with  
interdependencies
- 3. Upskill applications teams, partner with early adopters
- 4. Roadmap the death of TKG, PCO, and RKE1
- 5. Rely on all the teams to do their part in building the rest of the platform
- 6. My part: mature RKE2 clusters



# Persistent Government Staffing Problems

Slow Spring Hiring Wave: **7+ months** from interview to onboarding

Surprise Late Summer Failed Contract: whole teams disappeared **overnight**. **2+ months delay.**

**2 Months until deadline**



# Our First Ever Deprecation!

Successful (cutting it close) replatforming of dev, staging, and production (cloud)

- ⌘ Months of slow, steady progress for RKE2 maturity
- ⌘ Metrics: stability, compliance, manual effort, “fear” factor
- ⌘ No way we’re getting off of PCF (TAS) - start buying licenses



# And then there were three

But...

- ⌘ Clusters were barely stable
- ⌘ It was often impossible to get to a root cause: delete and rebuild
- ⌘ Our storage solution meant it could take days to upgrade a cluster





# Invest in: Observability, Recoverability, Cycle Time

- \* Off-cluster logging/metrics for anything infrastructure related
- \* Alerting for cluster down
- \* Breaking our parity for storage solution, massively reducing cluster upgrade times

These things all took surprisingly little time to do! Why did we wait so long?





# DevOps!

My team could do these things because we had:

- ⌘ Autonomy
- ⌘ Communication channels
- ⌘ Skill
- ⌘ Clusters in production with customers!

Had we given that to our customers?



# Air Gap Challenges

No easy-access parity Infrastructure for Production.

COVID, remote/distributed workforce, arduous rules and process..

We should have fixed this first!





# Turning the Corner: Product-led Maturation

Tremendous Progress:

- ⌘ A new round of leaders: Director of Product, Deputy of Product, Technical Product Managers, Design
- ⌘ Growing alignment with a roadmap of hierarchical epics, Value Stream Mapping, Dependency Mapping
- ⌘ Clusters are Stable

Roadmap:

- ⌘ Challenges across the air gap will require a big adjustment (refactoring)
- ⌘ Documentation and Developer Experience is terrible.





And we all lived happily ever after!!!

What could possibly go wrong?



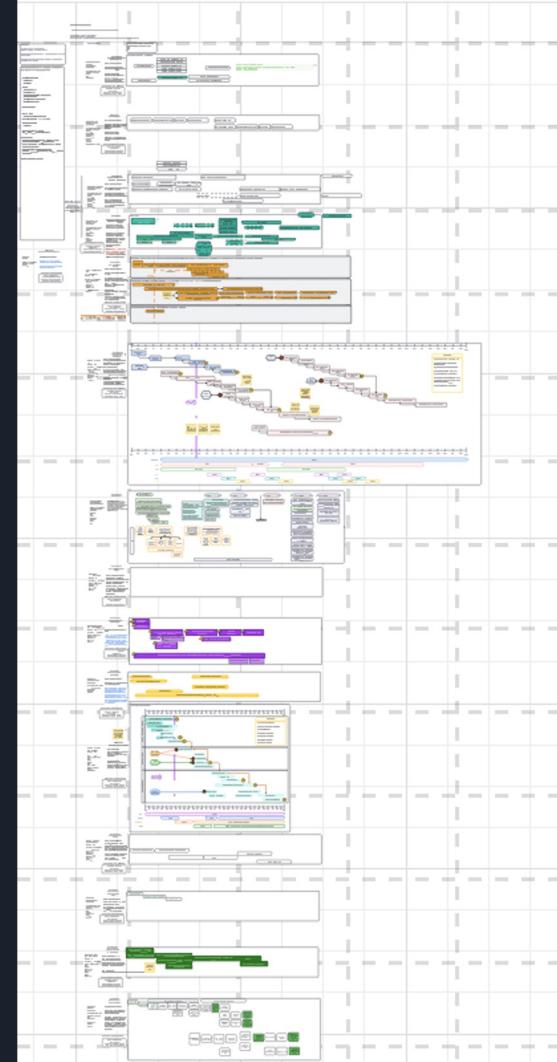
# Year 3: Murderous Plot Twist



# Major Change in Leadership = Major Change in Strategy

- \* Buy, not build
- \* Cloud only
- \* Project, not product
- \* No need for design on platform
- \* Gantt charts

Time to Pivot!



# Handling differences with empathy

Leaders, like everyone, want to feel safe

Owning the contrast in styles: build adapters!

- ✿ Seek to understand what your leadership needs to feel safe
- ✿ Continue to work the best way you know how
- ✿ Start delivering in a way that looks like what leadership expects

If your leadership gets what they expect, you can work the way you need



# We almost made it!

What went wrong:

- ⌘ Delays from pivots
- ⌘ Delays from staffing
- ⌘ Trying to do something very hard
- ⌘ Not enough emphasis on production parity
- ⌘ Big Bang Migration Plan

Meant we hadn't delivered anything across the air gap.



# Enter: EKS

Theoretically: an easy swap.

In practice: too much changed.

And so:

- \* Waves of platform leadership departures
- \* More reorgs



# RKE2 Progress: Frozen

RKE2 was poised to deprecate RKE1 and PCO in **months**.

**\*\*cancelled\*\***



# The death of RKE1

RKE1 caught the blame for a few outages

\*\*\*\*\*DELETE\*\*\*\*\*



# PCO Lives on!

RKE2 is frozen, EKS isn't ready.

PCO workloads have nowhere to go.





# The Winner!

After three years of trying, all of air gapped production is TAS!



## CLOUD FOUNDRY





# Here's the Help I'm Looking For

Ideas for helping government hire qualified senior technical leadership for CTO, Architect, CIO, CISO roles. How can non-technical people choose the right technical leaders?

Who is transforming government HR?

Mentorship as I grow into the above roles

Validation or dissent on baking observability in at the very beginning of any project (shifting observability left)