Kubernetes

As the control plane for the hybrid cloud

Clayton Coleman
RH Architect

Red Hat

Some problems can only be solved by adding another layer to the stack.

Linux
Virtualization
Config management
Public cloud
Private cloud
Kubernetes

Linux
Virtualization
Config management
Public cloud
Private cloud
Kubernetes

Improved
the state of the art of
building and sustaining
applications

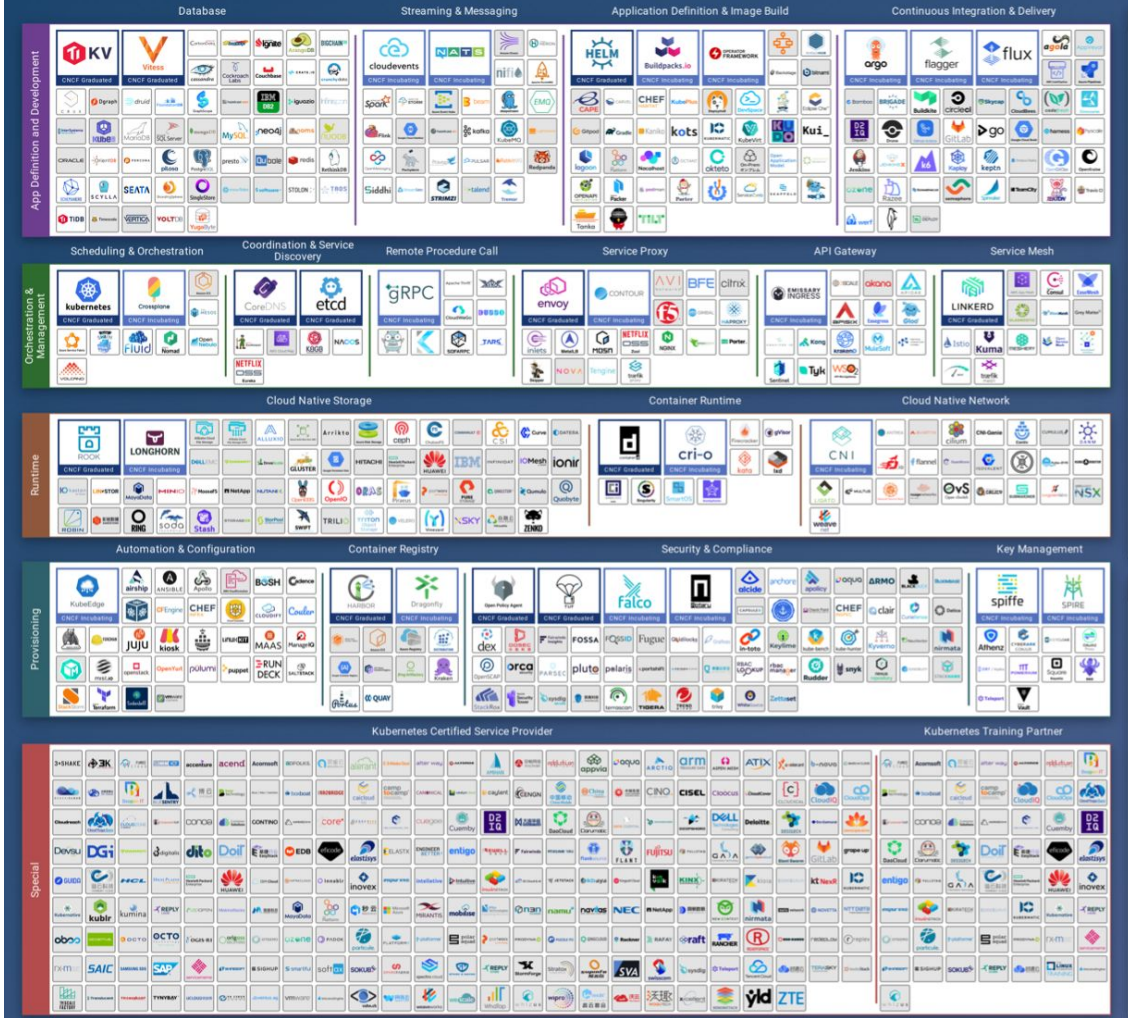
Linux
Virtualization
Config management
Public cloud
Private cloud
Kubernetes
???????

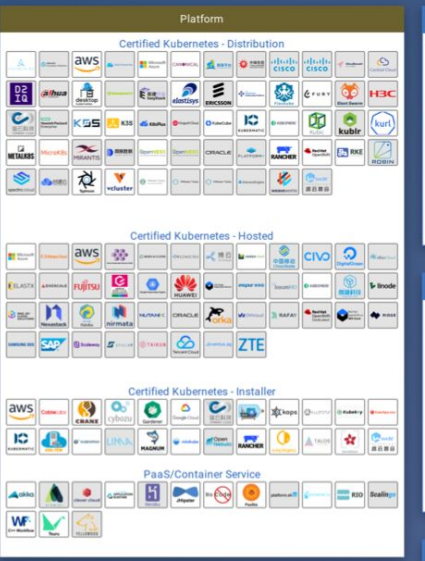
Improved
the state of the art of
building and sustaining
applications

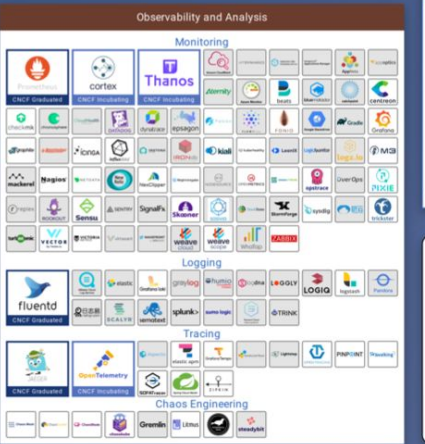
CNCF Cloud Native Landscape
2021-10-11T16:44:17Z c9a683a4

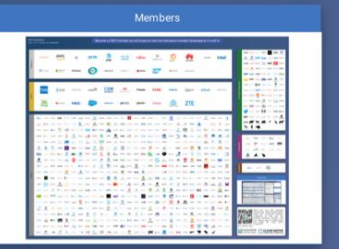
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

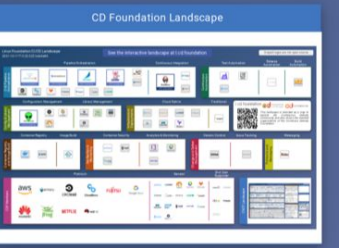
Greyed logos are not open source











CLOUD NATIVE Landscape

CLOUD NATIVE COMPUTING FOUNDATION
Redpoint Amplify

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

l.cncf.io

Yes please,
more of everything.

Yes please,
more of everything.

^- what hybrid cloud really means

Three questions that require a higher level answer:

How do I ...

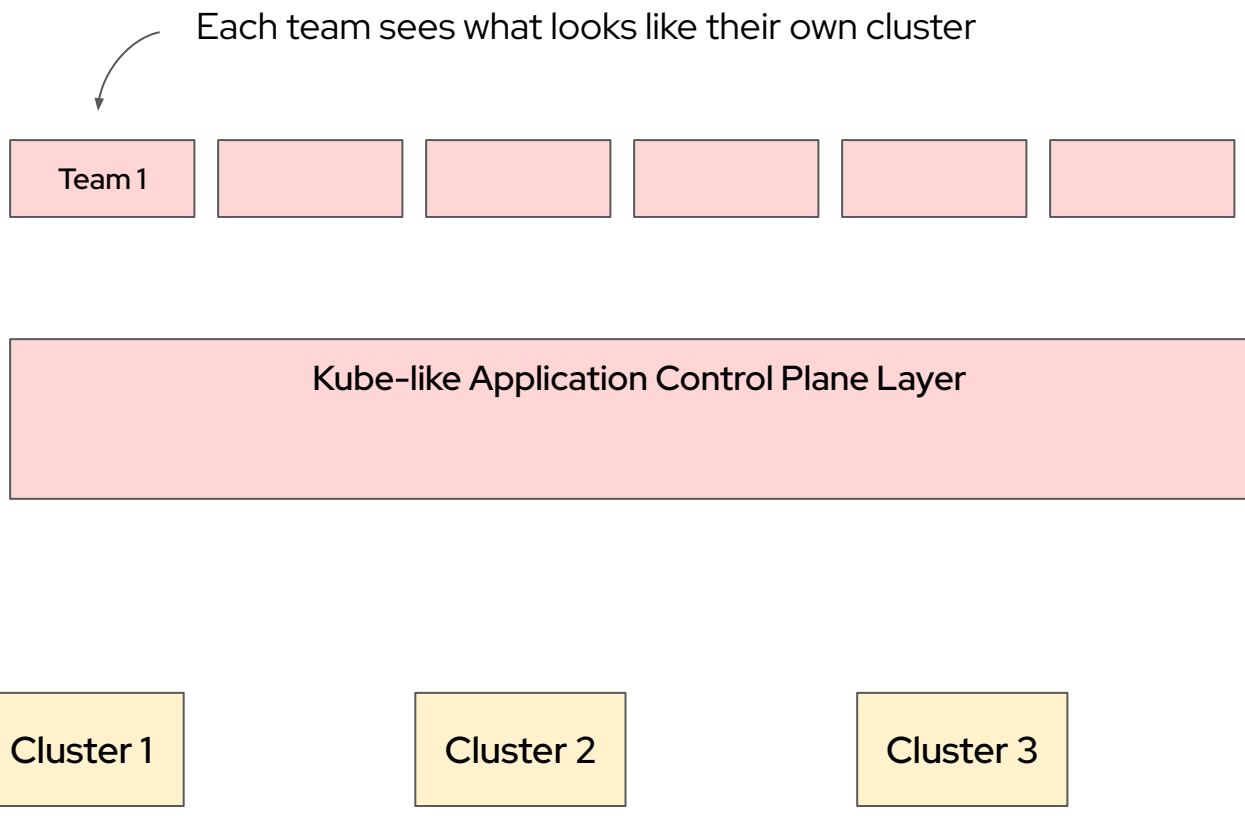
Grow beyond the limits of:	Easily integrate services from:	Keep my teams happy and safe:
one cluster	partners	from bugs
one cloud	vendors	from accidents
one region	clouds	from hackers
one vendor	my own teams	from overspending

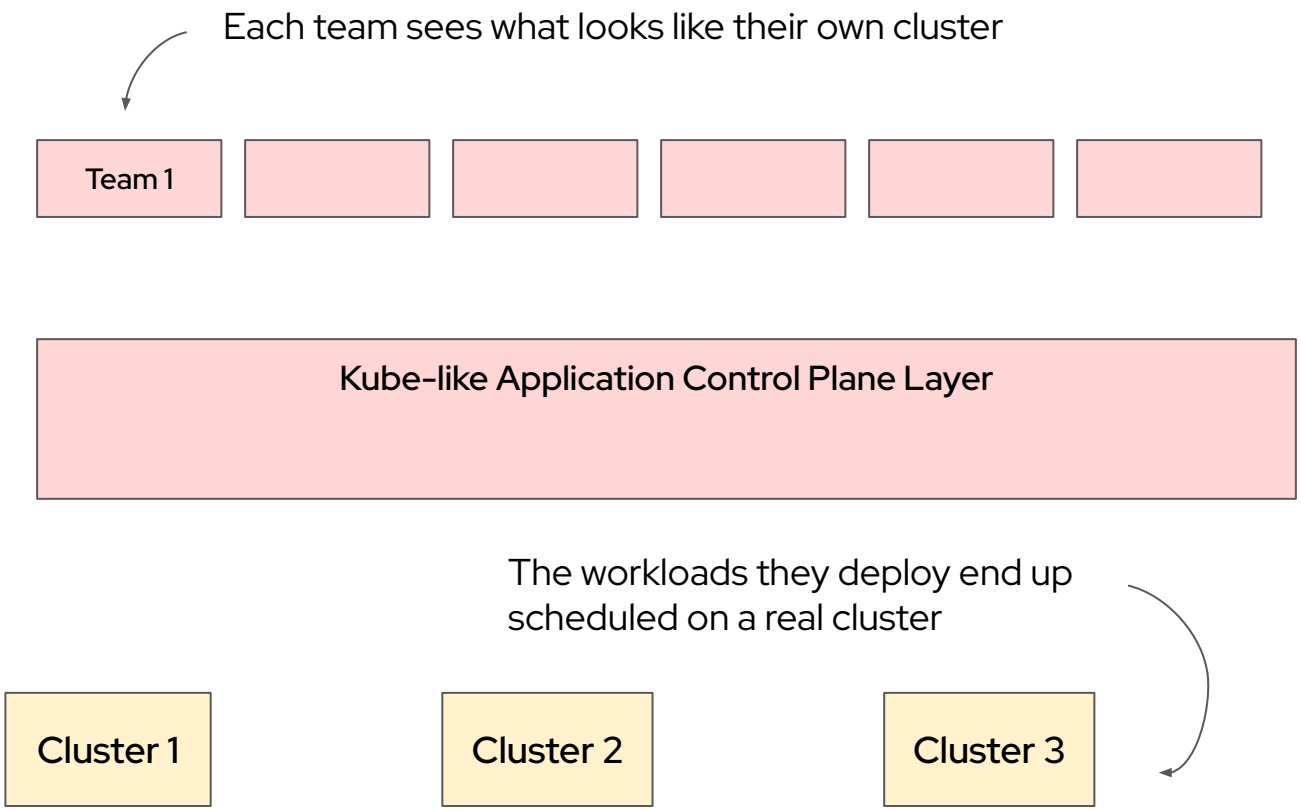
?

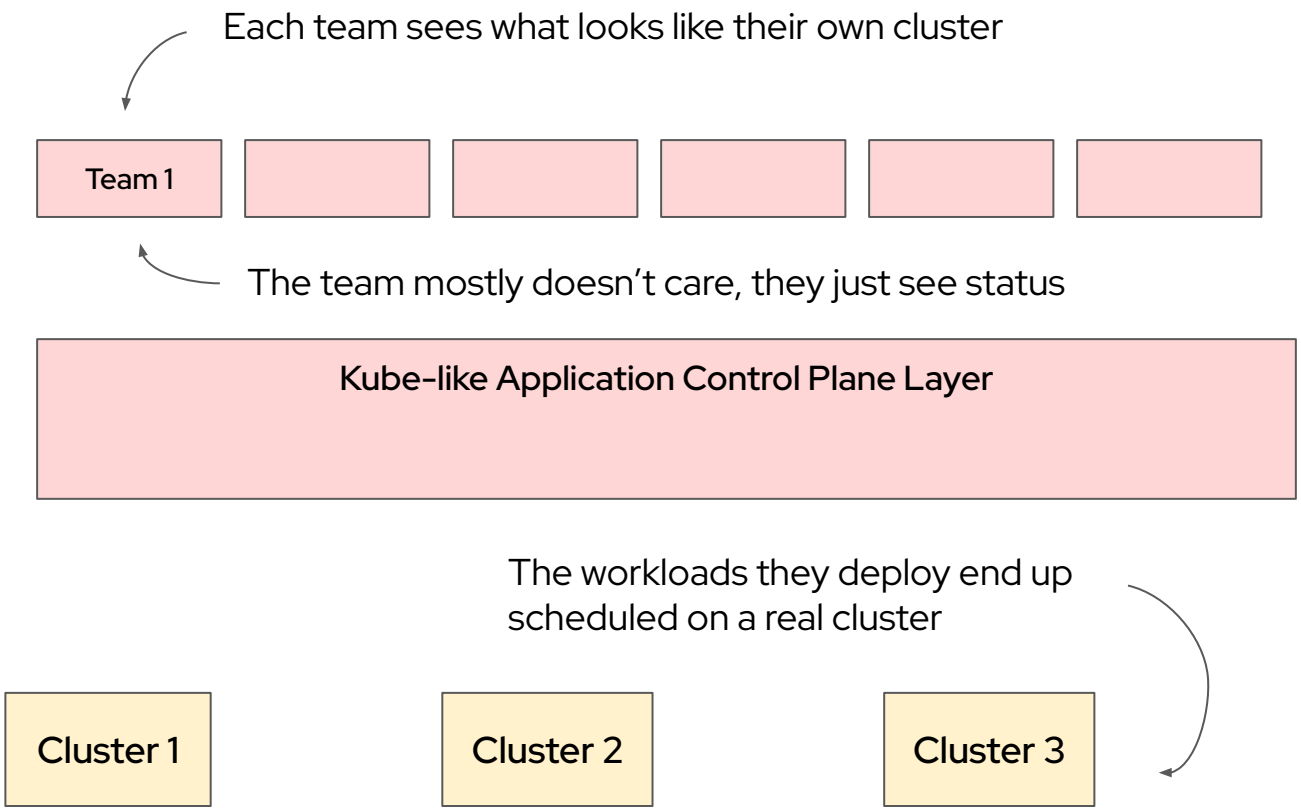
Can we find a common set of patterns we agree on in that higher layer?

What could an application-centric layer look like?

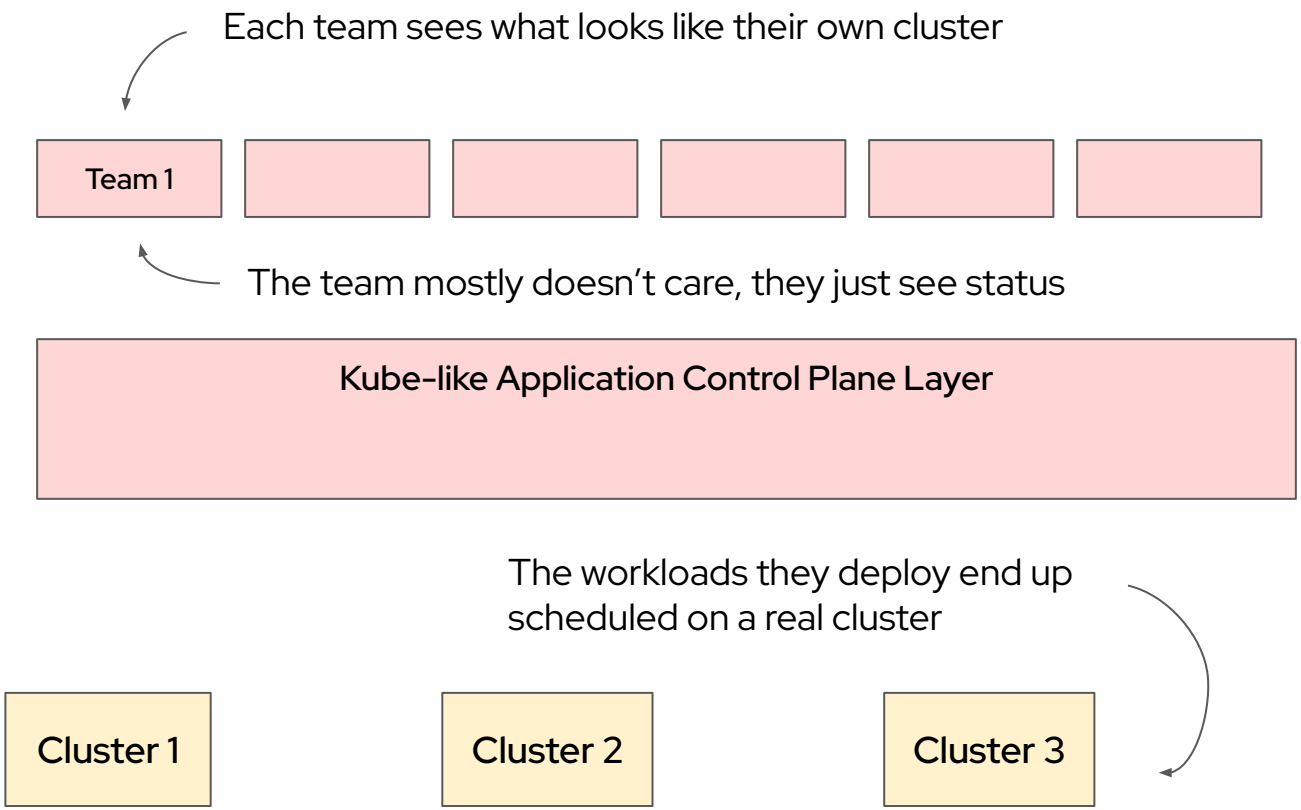
An application author can ‘kubectl apply’ most existing applications.

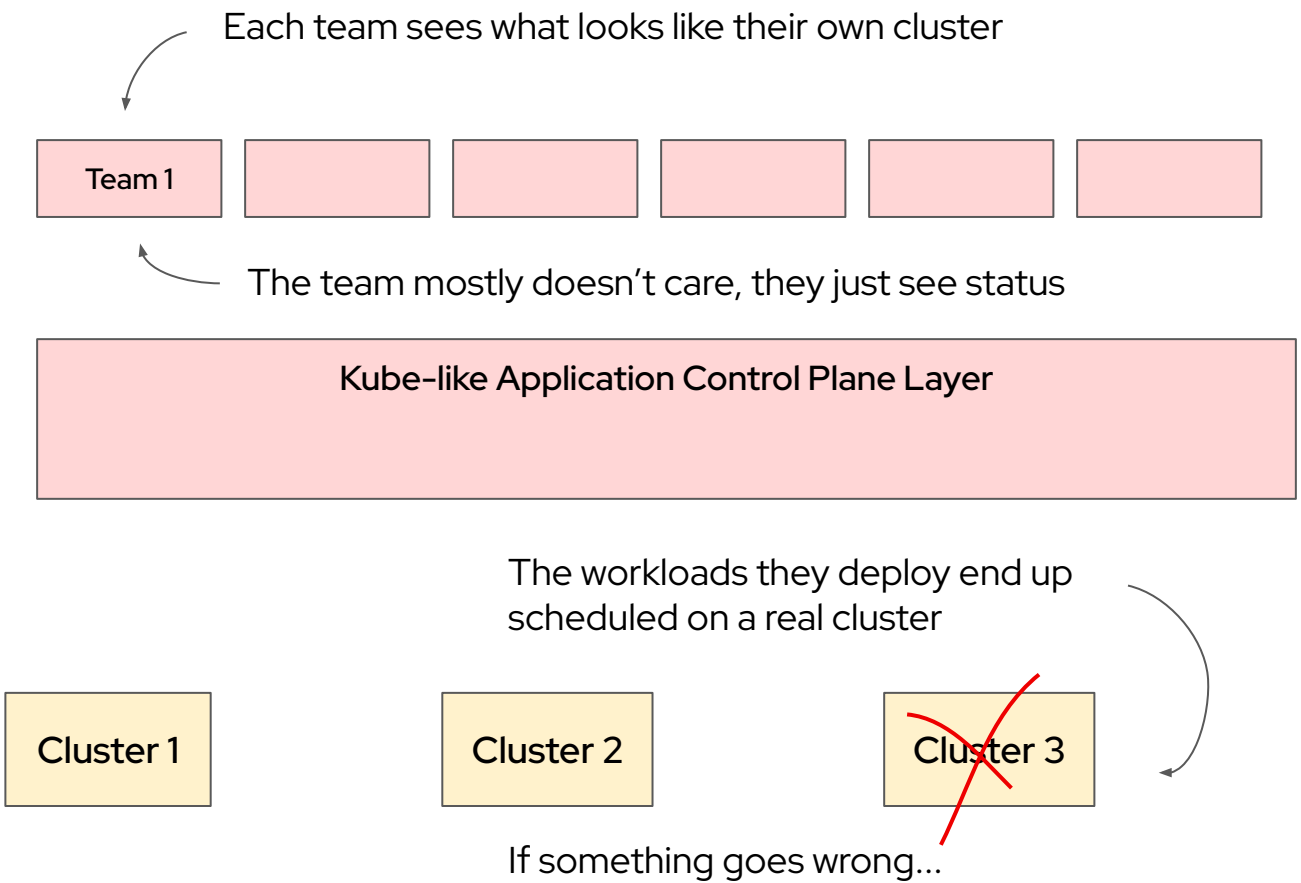

Each team sees what looks like their own cluster
Team 1
Kube-like Application Control Plane Layer
Cluster 1
Cluster 2
Cluster 3


Each team sees what looks like their own cluster
Team 1
Kube-like Application Control Plane Layer
The workloads they deploy end up scheduled on a real cluster
Cluster 1
Cluster 2
Cluster 3

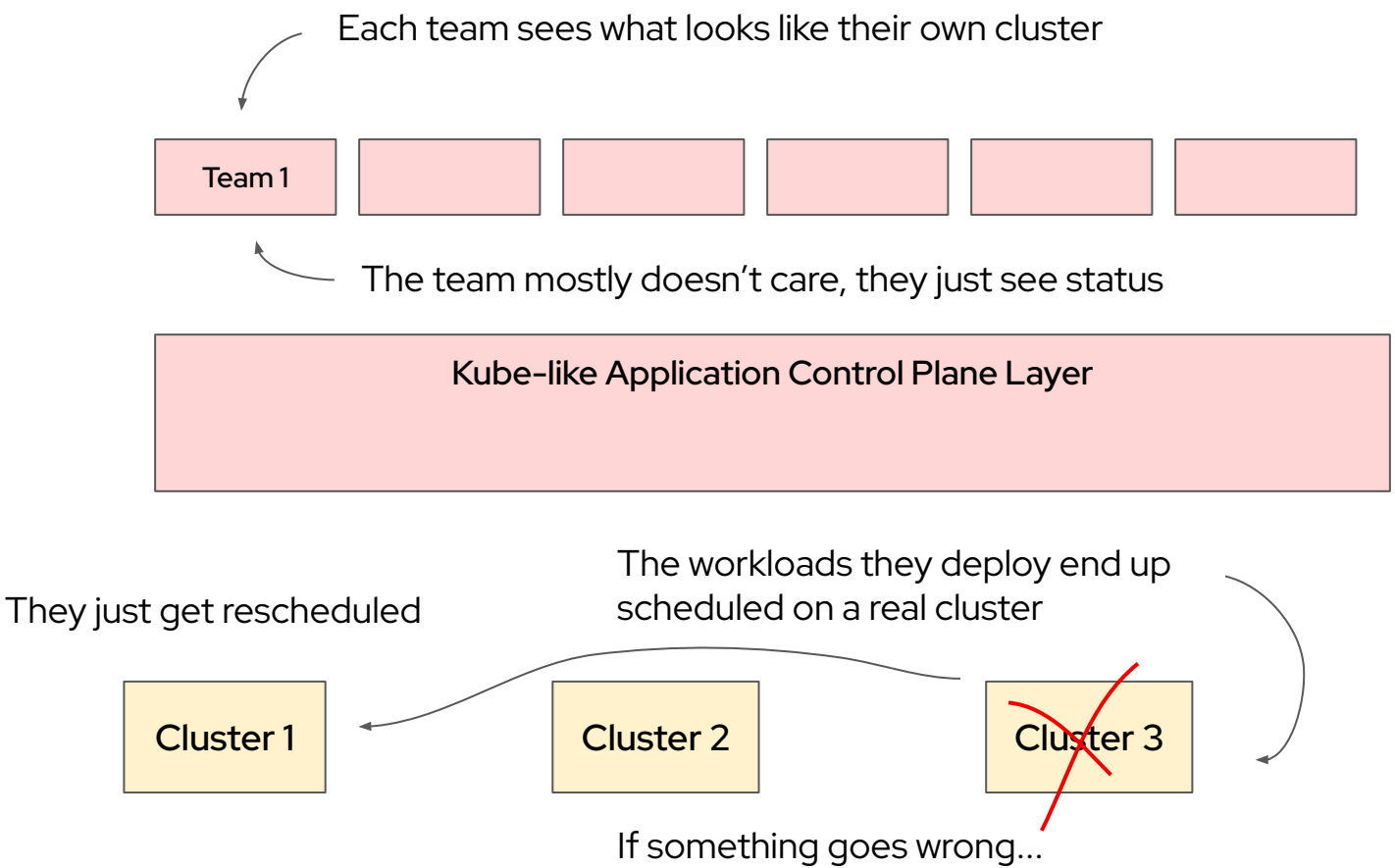

Each team sees what looks like their own cluster
Team 1
The team mostly doesn't care, they just see status
Kube-like Application Control Plane Layer
The workloads they deploy end up scheduled on a real cluster
Cluster 1
Cluster 2
Cluster 3

An operations team can transparently move applications between clusters with no disruption.


Each team sees what looks like their own cluster
Team 1
The team mostly doesn't care, they just see status
Kube-like Application Control Plane Layer
The workloads they deploy end up scheduled on a real cluster
Cluster 1
Cluster 2
Cluster 3

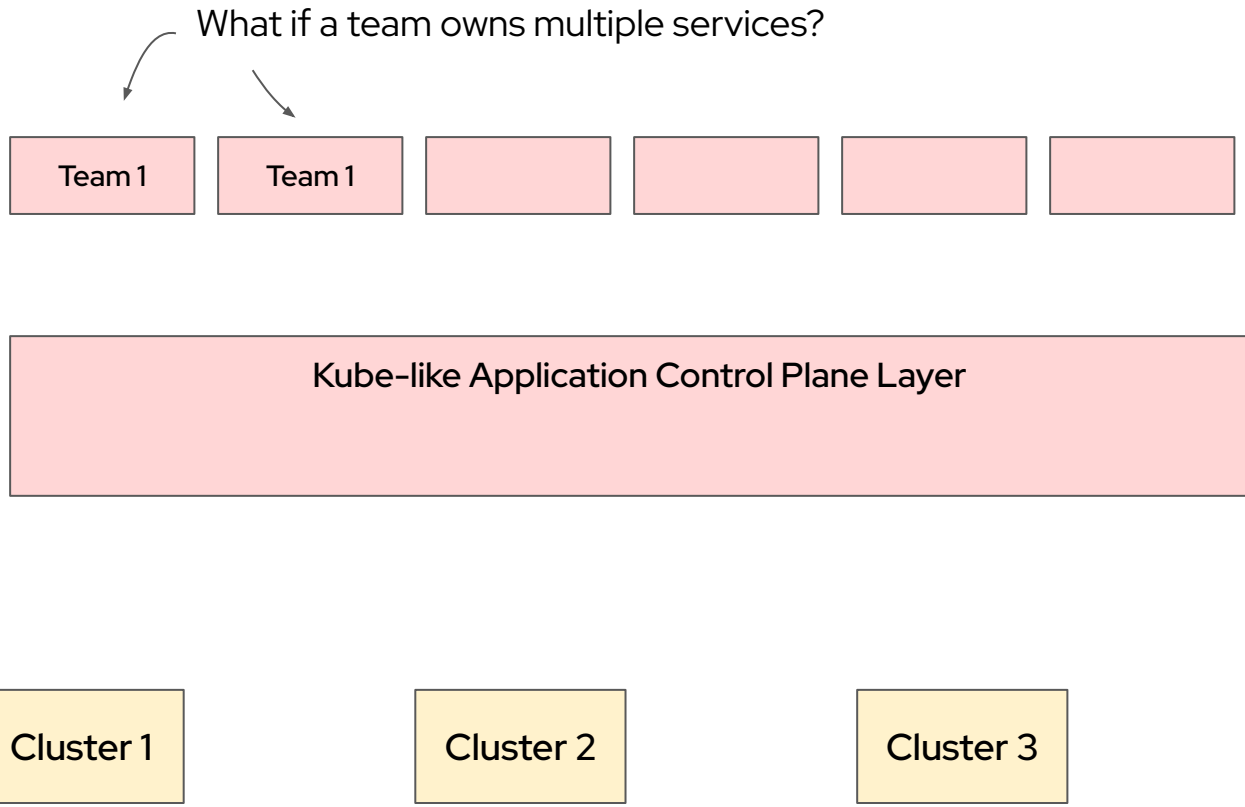

Each team sees what looks like their own cluster
Team 1
The team mostly doesn't care, they just see status
Kube-like Application Control Plane Layer
The workloads they deploy end up scheduled on a real cluster
Cluster 1
Cluster 2
Cluster 3
If something goes wrong...

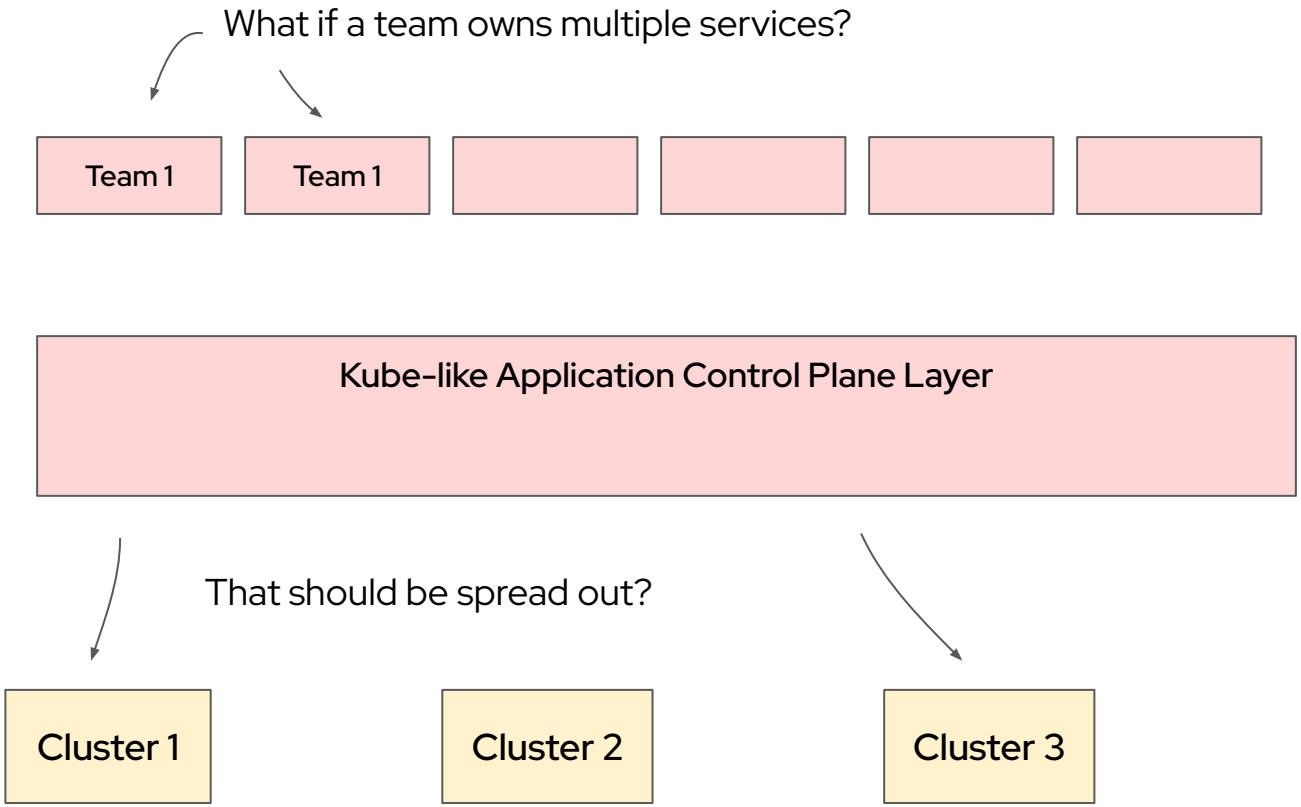
Red Hat

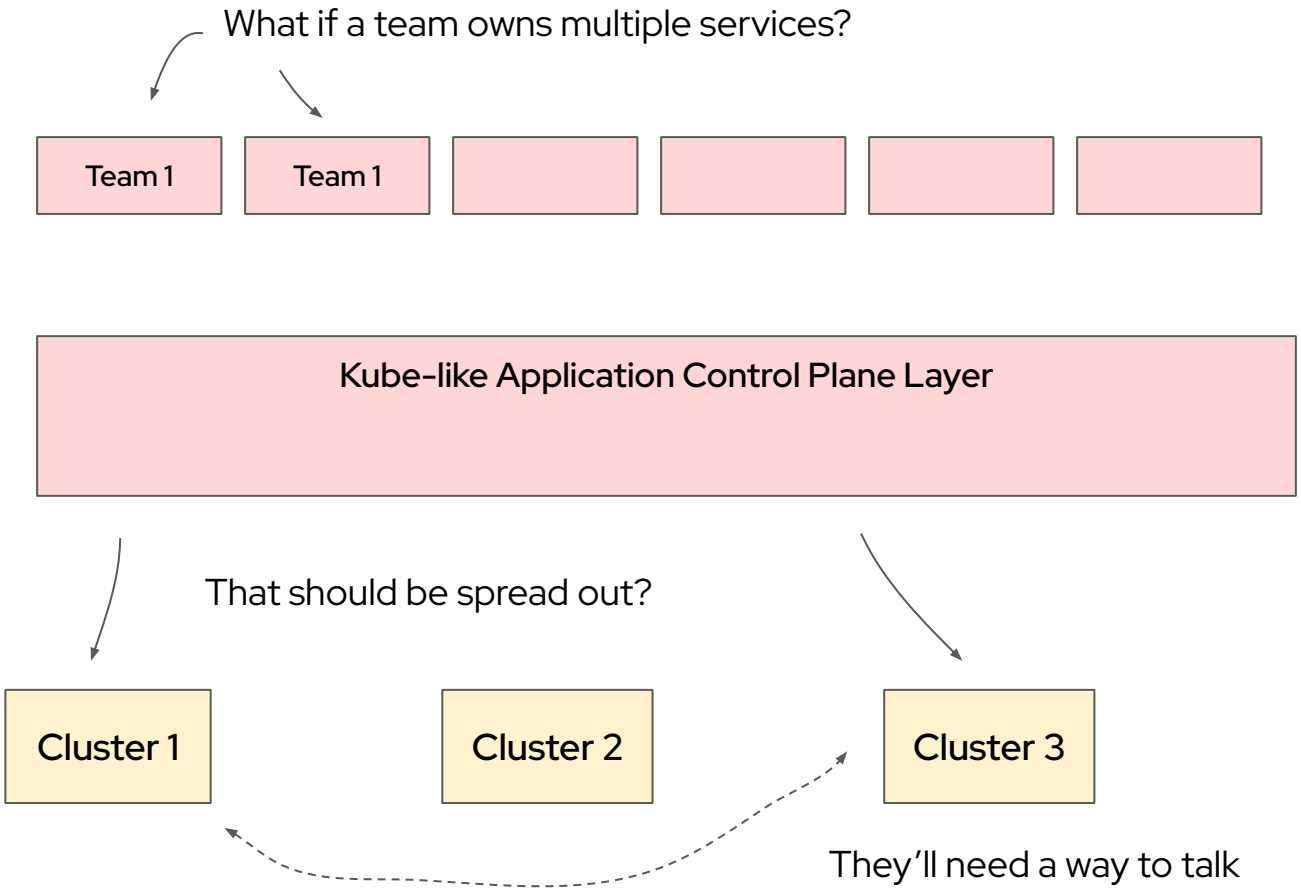

Each team sees what looks like their own cluster
Team 1
The team mostly doesn't care, they just see status
Kube-like Application Control Plane Layer
The workloads they deploy end up scheduled on a real cluster
They just get rescheduled
Cluster 1
Cluster 2
Cluster 3
If something goes wrong...

Red Hat

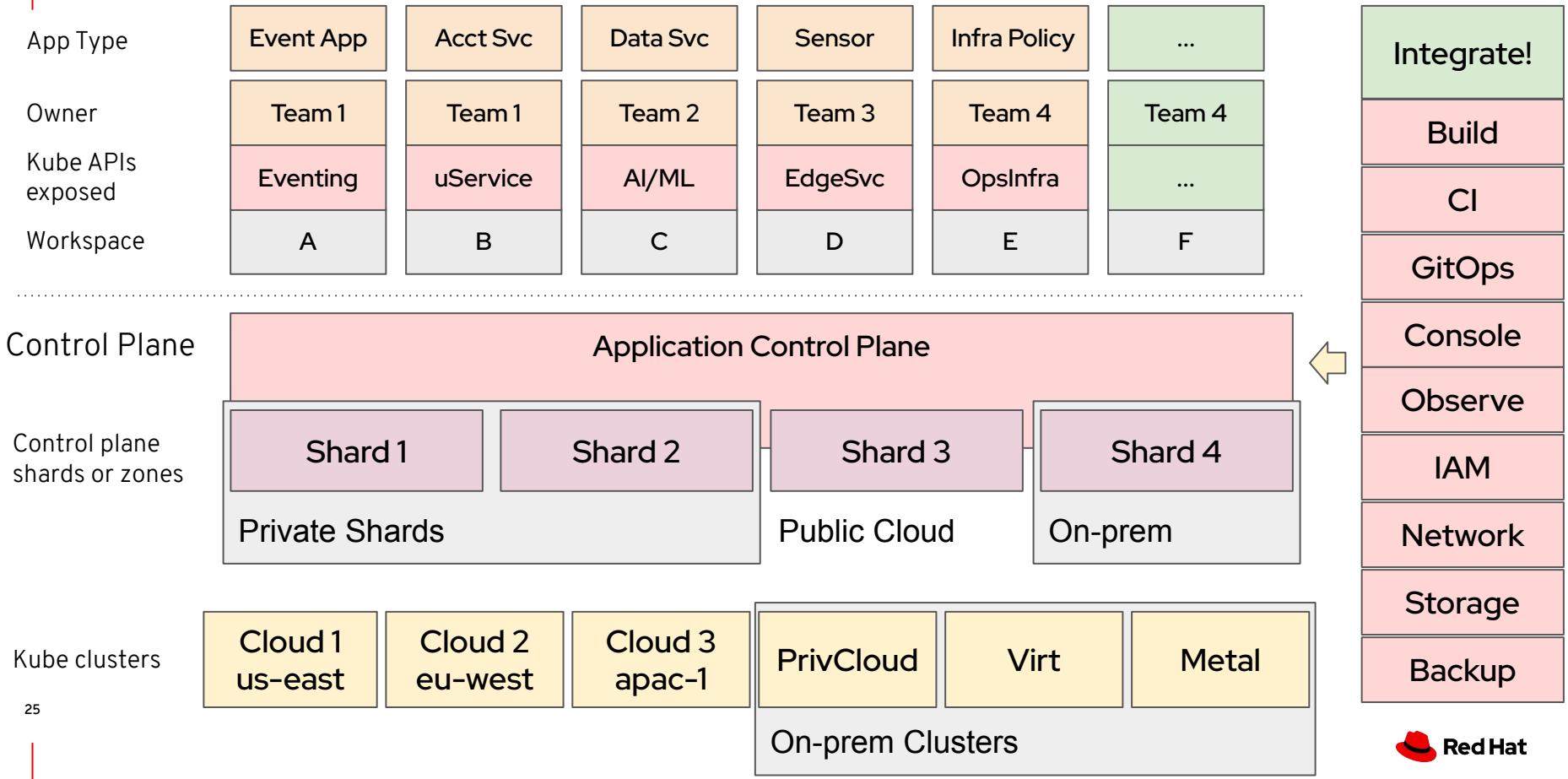
An application author can easily reference other services for use, for dev, test, or production.


What if a team owns multiple services?
Team 1
Team 1
Kube-like Application Control Plane Layer
Cluster 1
Cluster 2
Cluster 3


What if a team owns multiple services?
Team 1
Team 1
Kube-like Application Control Plane Layer
That should be spread out?
Cluster 1
Cluster 2
Cluster 3


What if a team owns multiple services?
Team 1
Team 1
Kube-like Application Control Plane Layer
That should be spread out?
Cluster 1
Cluster 2
Cluster 3
They'll need a way to talk

Q: Is this the future of Kubernetes?


App Type
Event App
Acct Svc
Data Svc
Sensor
Infra Policy
...
Integrate!
Owner
Team 1
Team 1
Team 2
Team 3
Team 4
Team 4
Build
Kube APIs exposed
Eventing
uService
AI/ML
EdgeSvc
OpsInfra
...
CI
Workspace
A
B
C
D
E
F
GitOps
Console
Control Plane
Application Control Plane
Observe
Control plane shards or zones
Shard 1
Shard 2
Shard 3
Shard 4
IAM
Private Shards
Public Cloud
On-prem
Network
Storage
Kube clusters
Cloud 1 us-east
Cloud 2 eu-west
Cloud 3 apac-1
PrivCloud
Virt
Metal
Backup
On-prem Clusters

Red Hat

Q: Is this the future of Kubernetes?

A: We don’t know yet

Community and Product Execution

(very, very, very early)

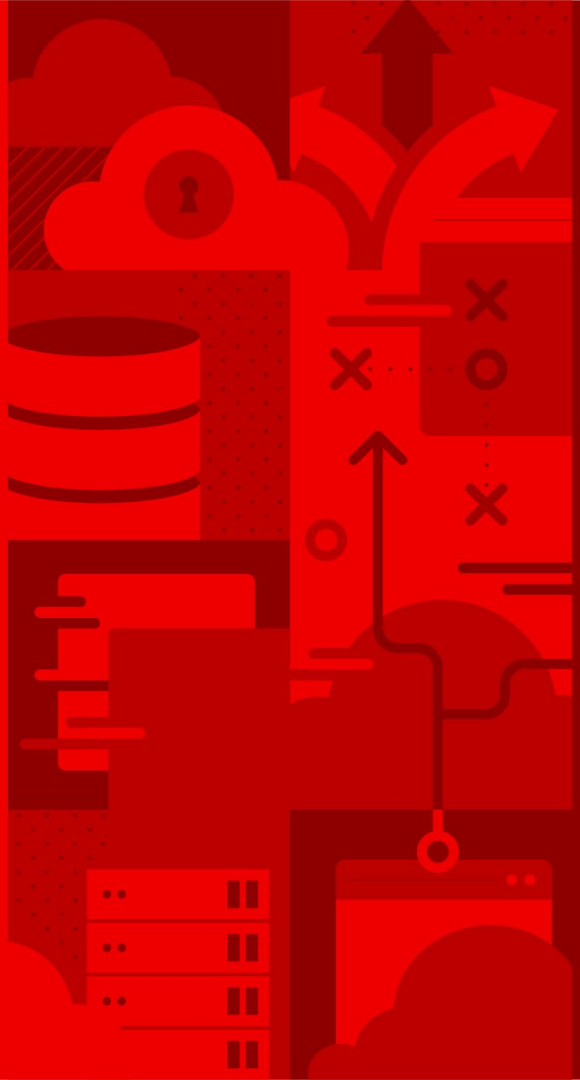
Phase 1 (3-6mo):

- Prototype of these ideas as “KCP” (github.com/kcp-dev) in Kube community
- Gather feedback on multi-cluster inside existing product roadmaps (OCP/ACM)
- Direct engagement with customers and partners on use cases and requirements

Phase 2 (6mo+)

- Build community consensus around one or more existing upstream projects
- Introduce specific capabilities via RH cloud services
- Integration into OCP/ACM roadmaps? Still TBD, this is just the beginning!

TOO EARLY



Thank you!

To get involved in shaping the direction of hybrid cloud as a customer or partner, contact:

Rob Szumksi rszumski@redhat.com
Clayton Coleman ccoleman@redhat.com

Join our community at github.com/kcp-dev

How are we iterating?

In the open, of course.

An application author can easily choose between functions, containers, and VMs to run their code.

Some Possible Design Constraints

1. Bring most existing apps along unchanged
2. Be incremental to the ecosystem
3. Orchestrate more than just containers
4. Scale from small (laptop) to large (global service)
5. First class tenancy and security isolation

Sounds like Kubernetes but more?

Some Possible Design Constraints

1. Bring most existing apps along unchanged
2. Be incremental to the ecosystem
3. Orchestrate more than just containers
4. Scale from small (laptop) to large (global service)
5. First class tenancy and security isolation

Sounds like Kubernetes but more?

Desirable Outcomes

1. Teams can self-service, agnostic of cluster or cloud
2. “Where” is a constraint, not an upfront choice
3. Consistent tools across diverse workloads
4. Encourage new APIs to streamline old problems
5. Reduce the cost of integrating of new ideas

Kubernetes without the cluster

An operations team can describe automatic multi-cluster failover without touching the application.

An application team can iterate in isolation from others without owning their own cluster or cloud accounts.

An operations team can roll out a new or updated integration to teams in a controlled manner.