
KubeCon
CloudNativeCon
Europe 2025
LONDON BUS

KubeCon | CloudNativeCon
Europe 2025

Exploring Multi-Tenant Kubernetes APIs and Controllers with kcp

https://docs.kcp.io/contrib/learning/

https://github.com/kcp-dev/contrib

https://link.excalidraw.com/l/2pL0lvjcXfb/1VUehl9hJha

Description

While Kubernetes transformed container orchestration, creating multi-tenant platforms remains a significant challenge. kcp goes beyond DevOps and workload management, to reimagine how we deliver true SaaS experiences for platform engineers. Think workspaces and multi-tenancy, not namespaces in a singular cluster. Think sharding and horizontal scaling, not overly large and hard to maintain deployments. With novel approaches to well-established building blocks in Kubernetes API-Machinery, this CNCF sandbox project gives engineers a framework to host and consume any kind of API they need to support their platforms.

In this hands-on workshop, participants will learn how to extend Kubernetes with KCP, build APIs, and design controllers to tackle multi-tenancy challenges. By exploring real-world scenarios like DBaaS across clusters, attendees will gain practical skills to create scalable, multi-tenant platforms for their Kubernetes environments.

While presenting this topic in the previous couple KubeCons we got full-room attendance. However, we discovered that a 35-minute presentation to present quite complicated kcp as a framework is a challenge. One of the feedbacks we received from participants is that a workshop, covering these things in detail would be very much desired. This is a follow-up from previous sessions to deliver on the promise.

We gathered a diverse group of KCP adopters and maintainers from multiple parties involved in the KCP project for this workshop to ensure wide and diverse representation.

Session outline: (1) Introduction to SaaS-in-Kubernetes topic and how kcp is key element in enabling platform engineers and developers to build such SaaS platforms, (2) familiarize attendees with writing kcp-aware code, (3) describe a practical example we'll help attendees create during the session, (4) work together and implement the demo. The example would touch on hosting and consuming SaaS-like APIs we'll create during the session: self-servicing databases to be used in a web-app platform.

Prerequisites

4 x shell terminal (Linux, Mac, **or GitHub Codespaces**) on the same box

- git
- Ability to run **kind** clusters
- Everything else we gonna setup on the go

https://docs.kcp.io/contrib/learning/

Intros

Mangirdas Judeikis - Staff Engineer, kcp maintainer, Cast AI

Robert Vasek - Software Engineer, Clyso GmbH

Marko Mudrinić - Senior Software Engineer, Kubermatic

Nabarun Pal - Principal Software Engineer, Kubernetes Maintainer

Varsha Prasad Narsing - Software Engineer, Red Hat

https://docs.kcp.io/contrib/learning/

High level plan

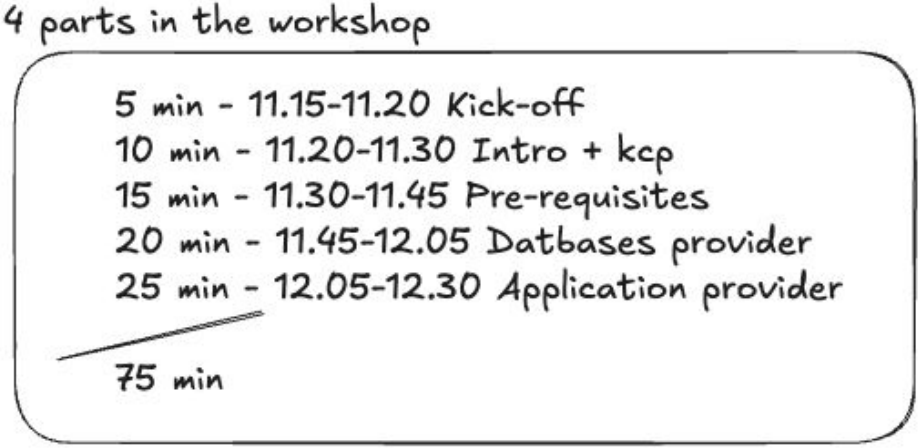
1. Setup environment - Kind & friends
2. Explore concepts - Workspaces & API{Bindings,Exports}
3. Lift & shift postgres operator to be kcp aware & multitenant
4. https://github.com/kubernetes-sigs/multicluster-runtime example controller

Ask questions any time!!!!

https://docs.kcp.io/contrib/learning/

Agenda

1. Intro to what we gonna gonna build
2. Setup the environment
3. Explore concepts
4. Postgres operator as provider setup
5. Controller-runtime example setup



https://docs.kcp.io/contrib/learning/

CONTENT WARNING

This is workshop, where you need to pay attention to details, follow the instructions and understand the concepts. Our goal is for participants to learn as much as possible in as short a time as possible.

It's not marketing material, **we are here not to sell you anything**. It's all about technology!

Brace yourselves!

https://docs.kcp.io/contrib/learning/

What is kcp?

KubeCon CloudNativeCon Europe 2025

1. Kubernetes-like control plane

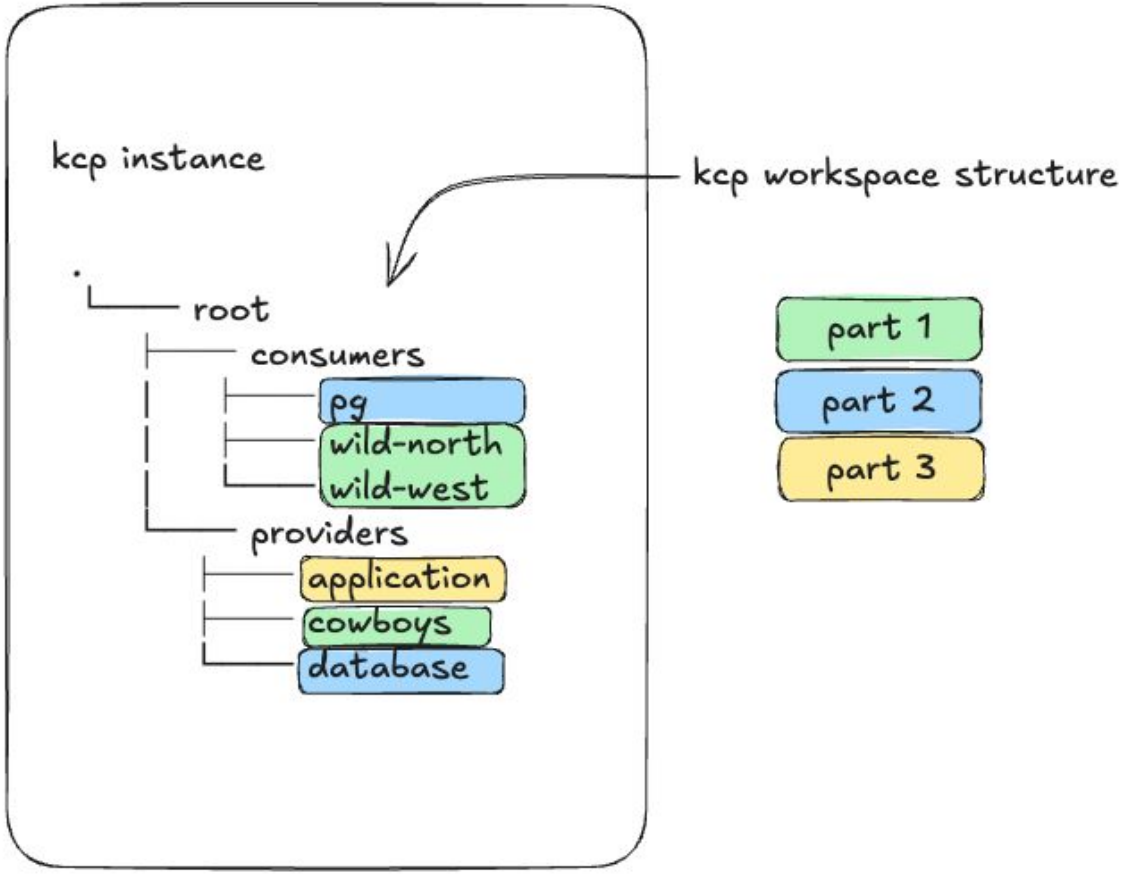
2. An open source horizontally scalable control plane for Kubernetes-like APIs.

3. A single kcp control plane hosts multiple isolated workspaces (lightweight virtual clusters), each with its own API endpoint, rbac and full isolation

4. It adds additional API management capabilities, not existing in K8S.

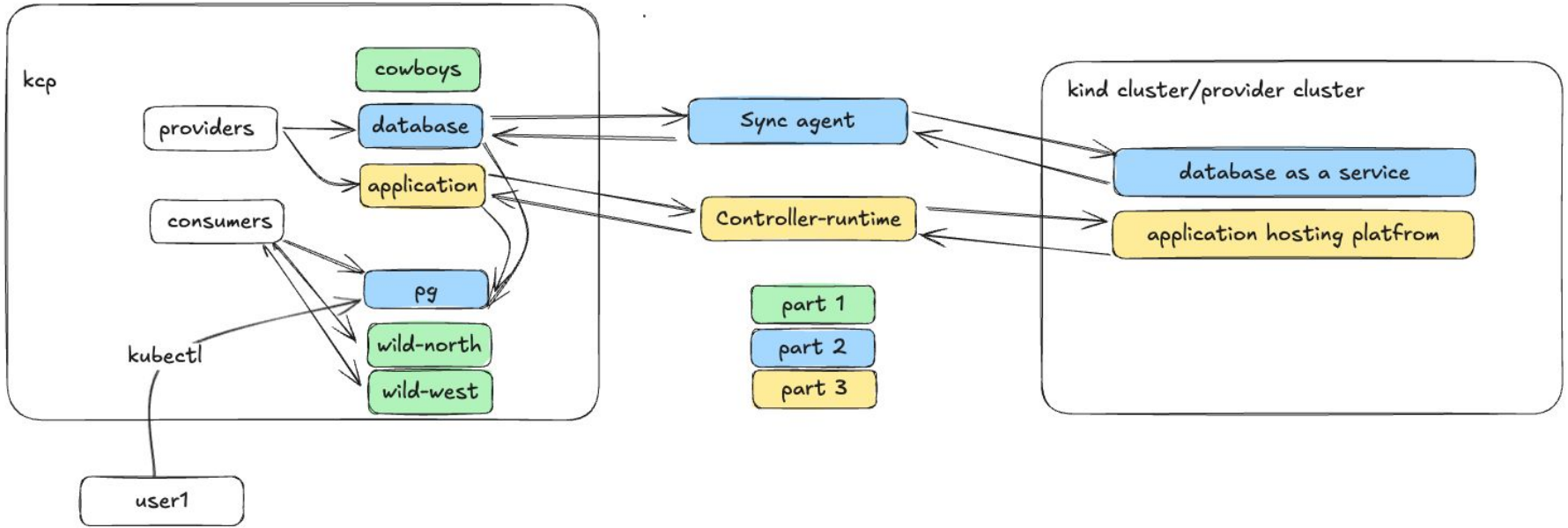
https://docs.kcp.io/contrib/learning/

Workshop structure



https://docs.kcp.io/contrib/learning/

Workshop structure



https://docs.kcp.io/contrib/learning/

https://docs.kcp.io/contrib/learning/

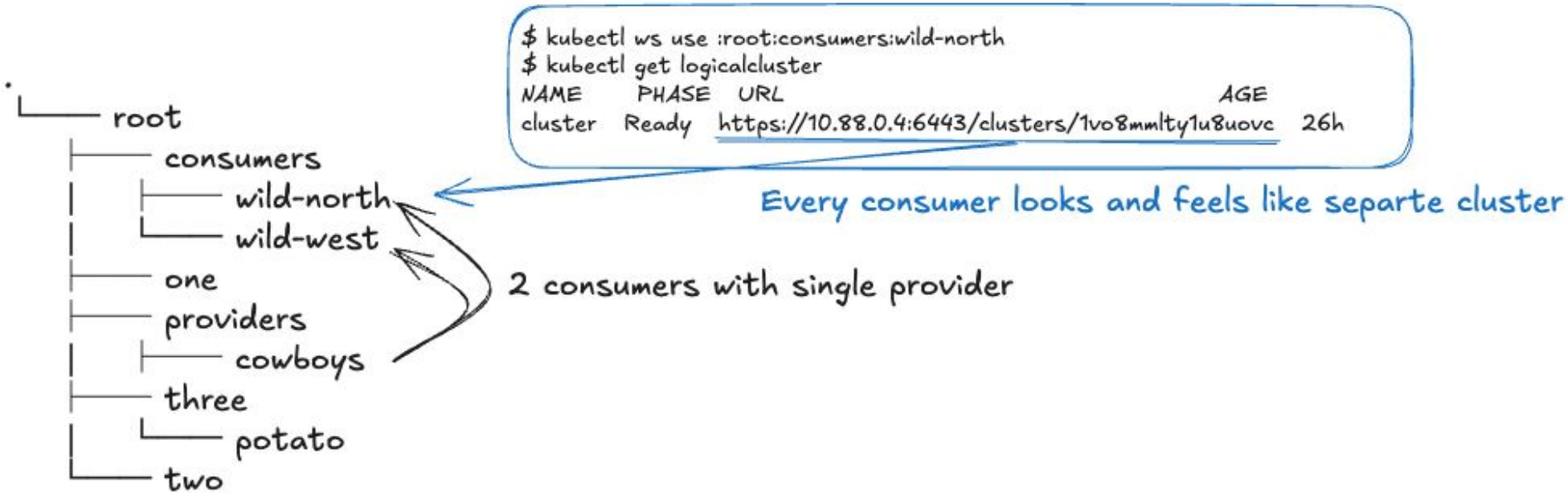
Part 1 - Prerequisites

https://docs.kcp.io/contrib/learning/

Part 1 - Prerequisites

https://docs.kcp.io/contrib/learning/

Part 2 - Explore the workspaces

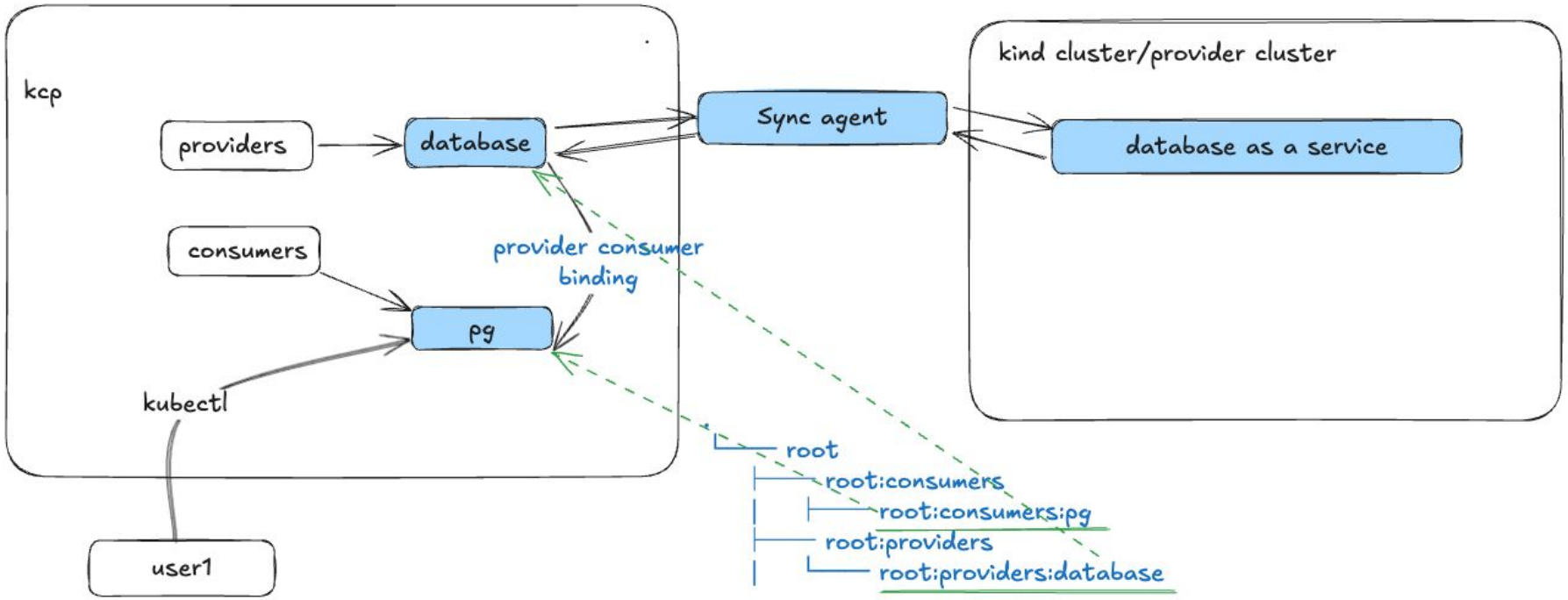


https://docs.kcp.io/contrib/learning/

Things will get complex soon. Get ready!

https://docs.kcp.io/contrib/learning/

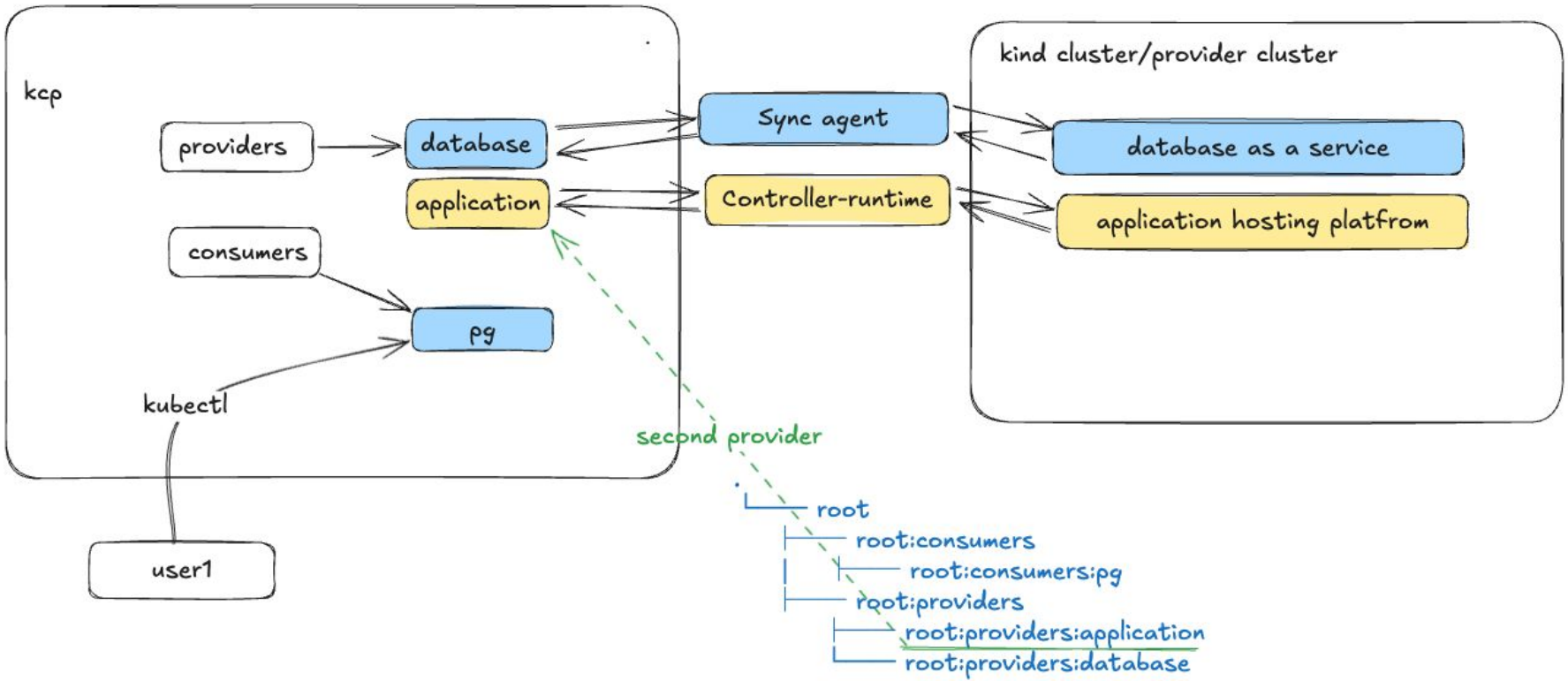
Part 3 - Postgres as a service



https://docs.kcp.io/contrib/learning/

MR

Part 4 - Custom application as a service



https://docs.kcp.io/contrib/learning/

Why?

Allow 3rd party to manage single CRD and few named resources in your cluster/workspace without compromising security

That is it!
Thanks for all who sticked with us to the end!

Other talks

Wednesday, April 2

11:15 BST

Tutorial: Exploring Multi-Tenant Kubernetes APIs and Controllers With Kcp - Robert Vasek, Clyso GmbH; Nabarun Pal, Independent; Varsha Narsing, Red Hat; Marko Mudrinic, Kubermatic GmbH; Mangirdas Judeikis, Cast AI

You are here!

Thursday, April 3

11:00 BST

Extending Kubernetes Resource Model (KRM) Beyond Kubernetes Workloads - Mangirdas Judeikis, Cast AI & Nabarun Pal, Independent (Description: kcp)

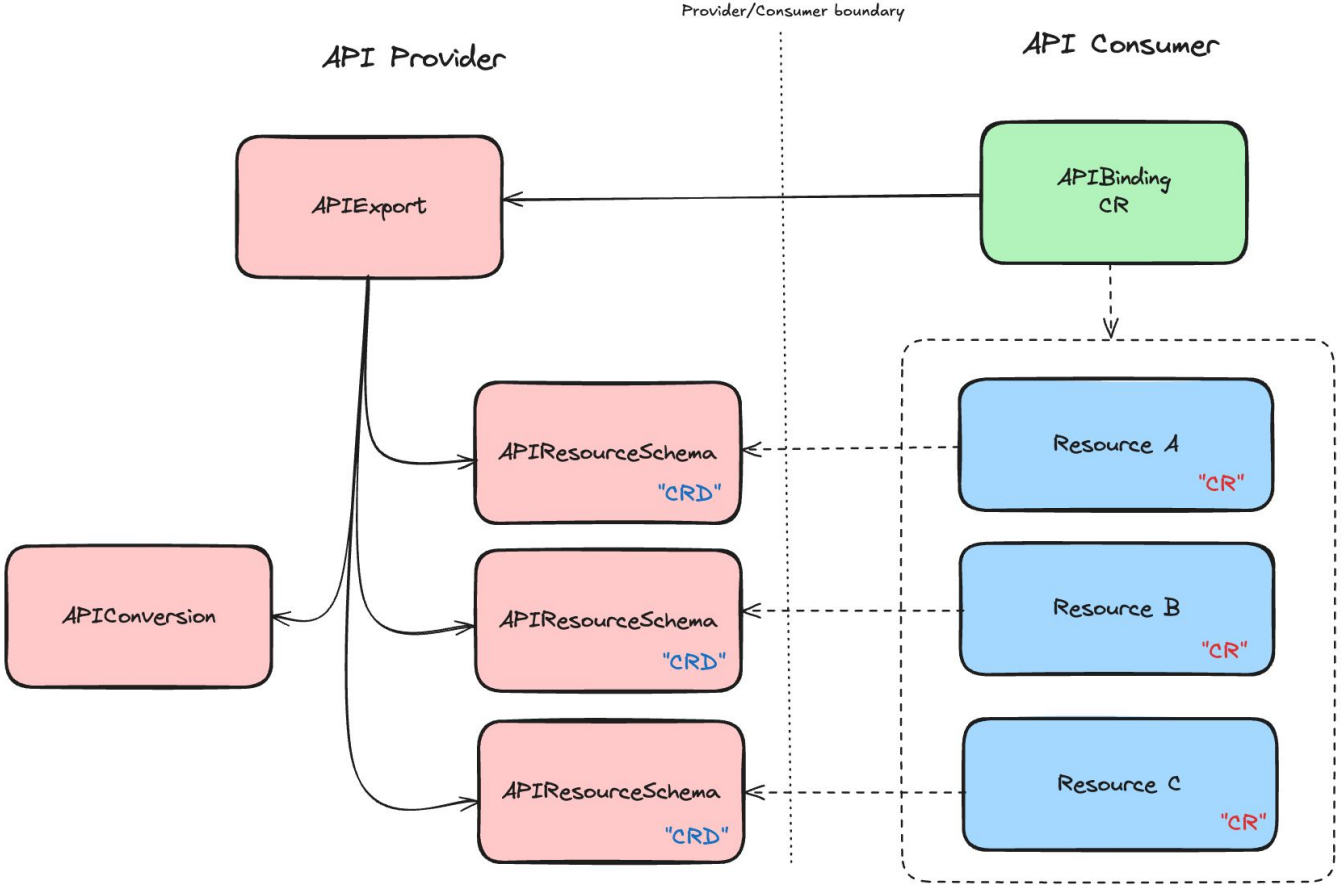
15:00 BST

Dynamic Multi-Cluster Controllers With Controller-runtime - Marvin Beckers, Kubermatic & Stefan Schimanski, Upbound (Description: kcp)

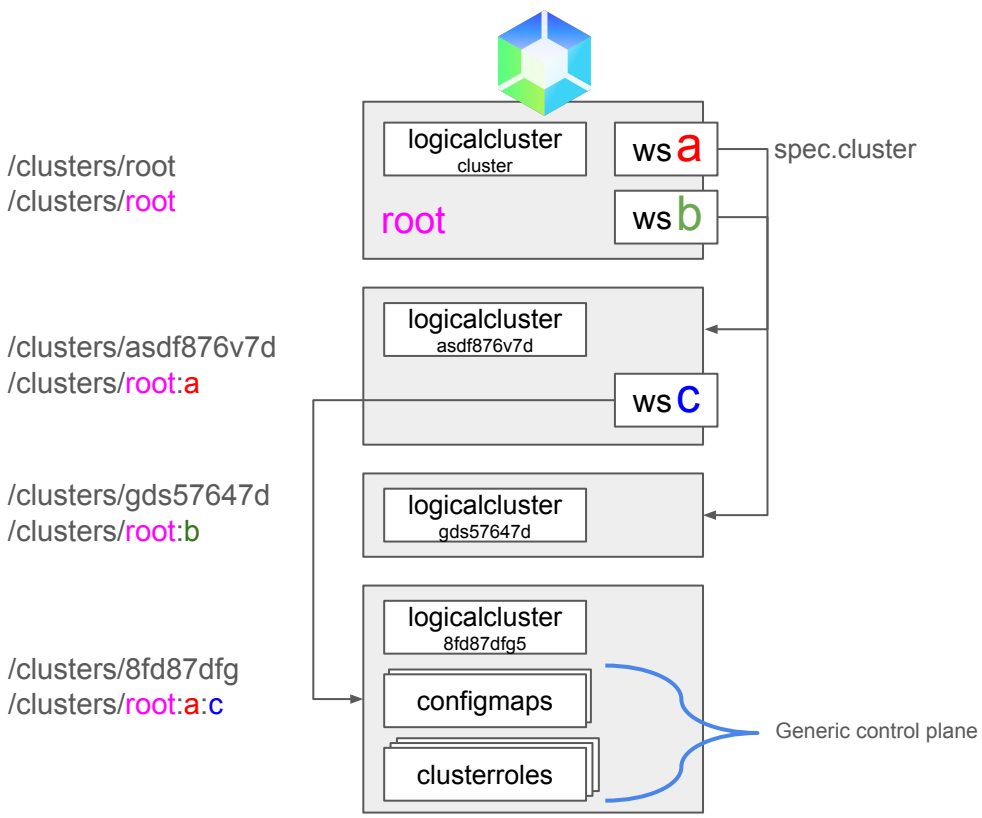
https://docs.kcp.io/contrib/learning/

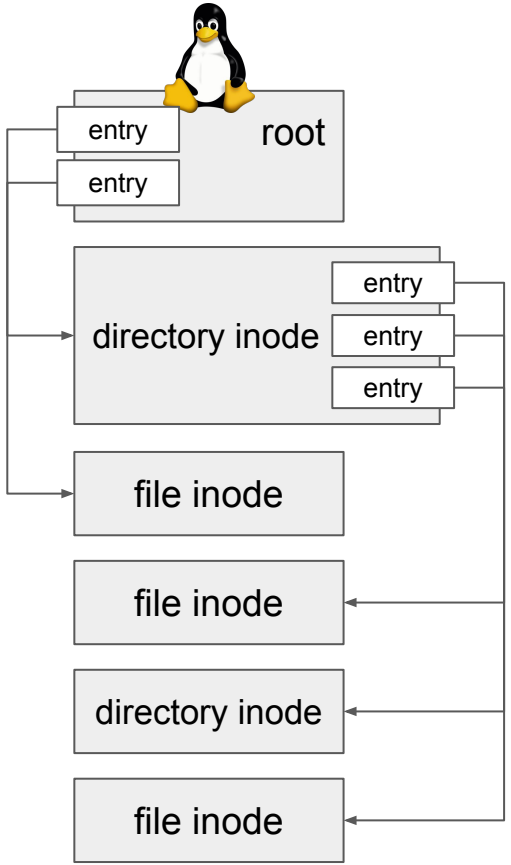
Helper slides

APIs - CRD's decoupling from CR's

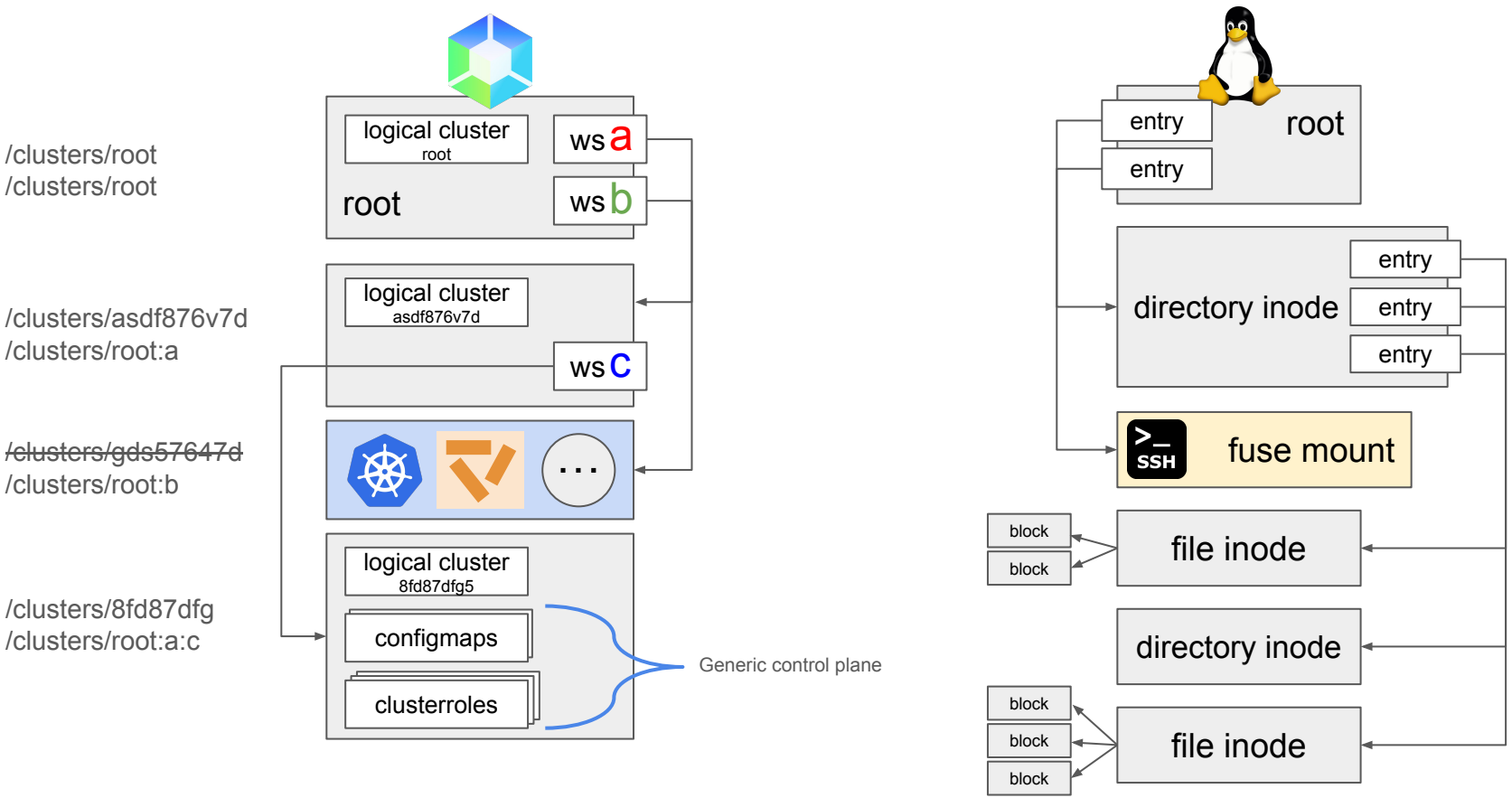


From Logical Cluster to Workspaces



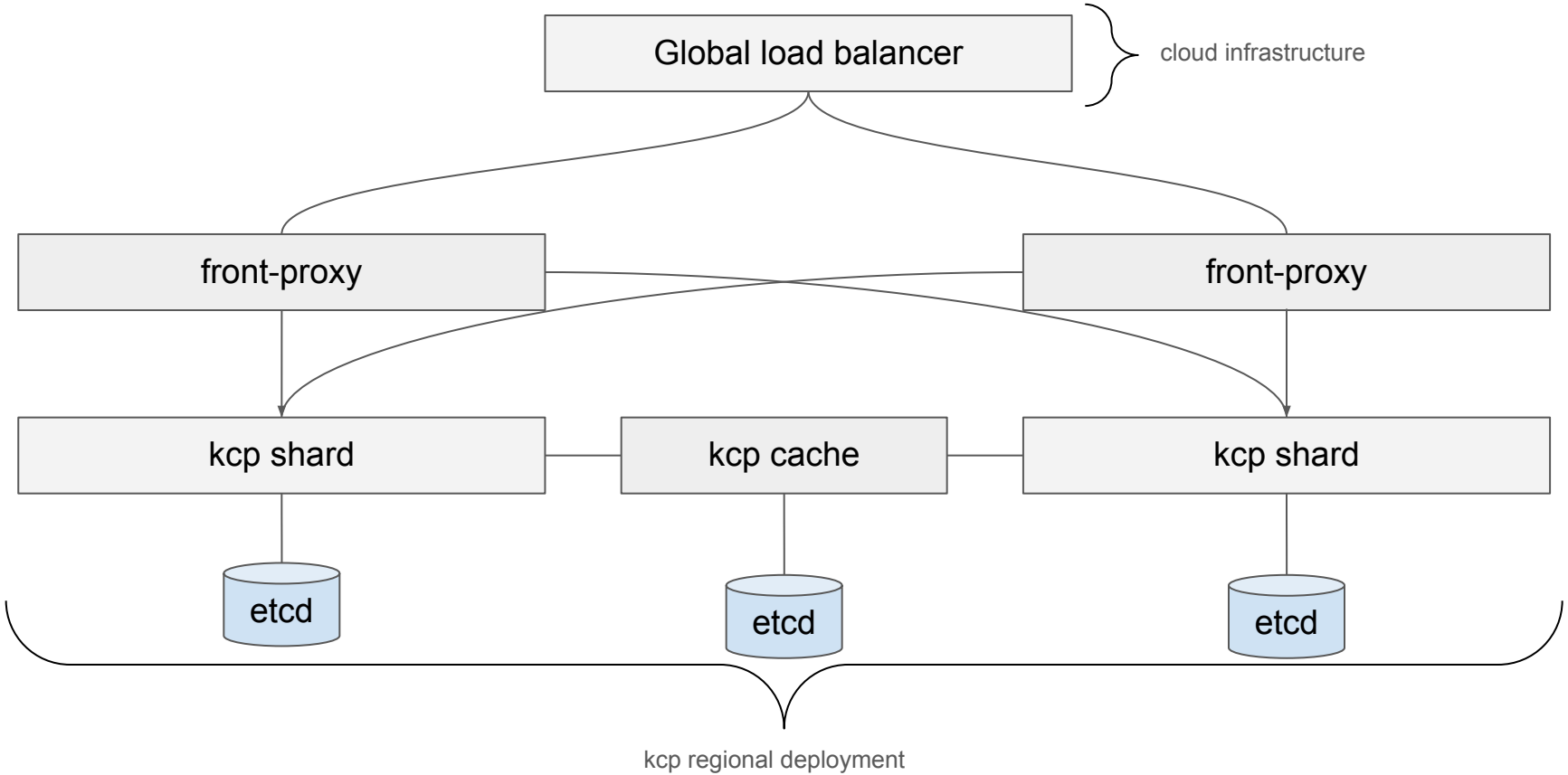


S

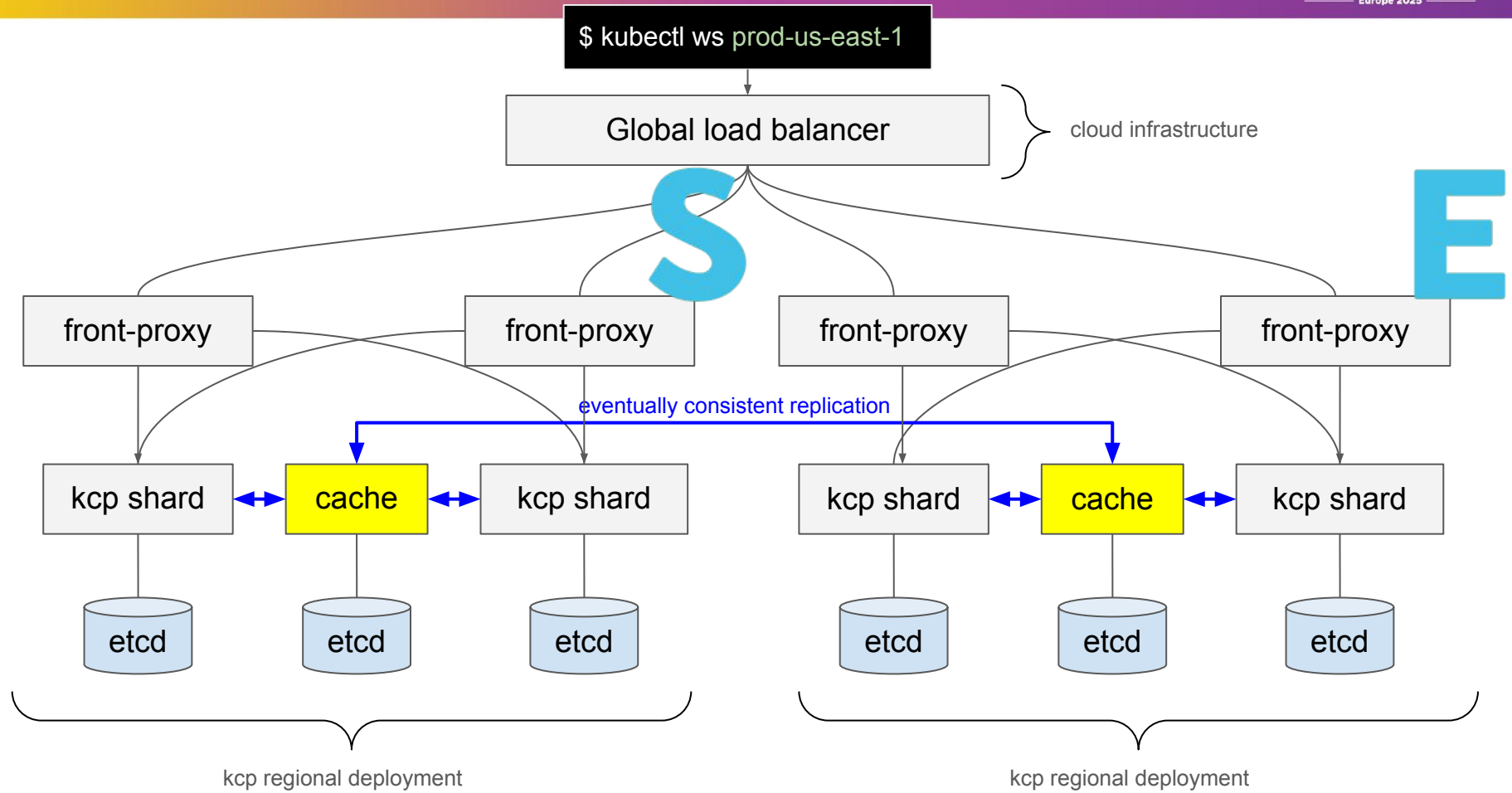

/clusters/root
/clusters/root
/clusters/asdf876v7d
/clusters/root:a
/clusters/gds57647d
/clusters/root:b
/clusters/8fd87dfg
/clusters/root:a:c
logical cluster
root
root
ws a
ws b
logical cluster
asdf876v7d
ws c
...
logical cluster
8fd87dfg5
configmaps
clusterroles
Generic control plane
entry
entry
root
directory inode
entry
entry
entry
>_
SSH
fuse mount
block
block
file inode
directory inode
block
block
block
file inode

M

Global architecture



Deep dive into components: outlook



Draft things from this slide