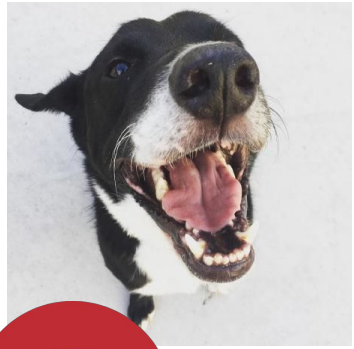
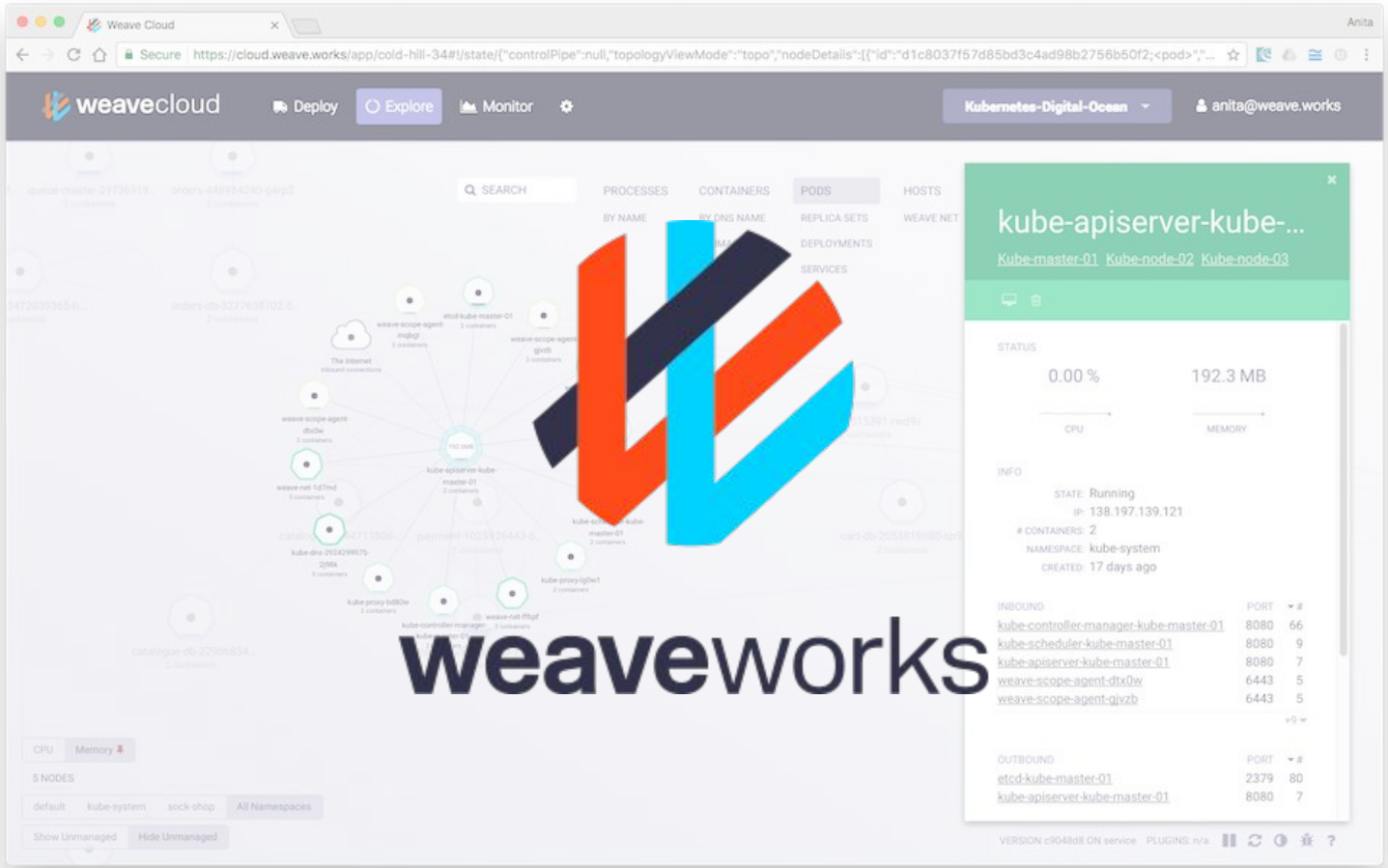
leigh capili



@capileigh stealthybox



@capileigh stealthybox

https://**fluxcd.io**

https://**gitops.community**

@capileigh stealthybox

Strategies for Multi-Cluster Routing

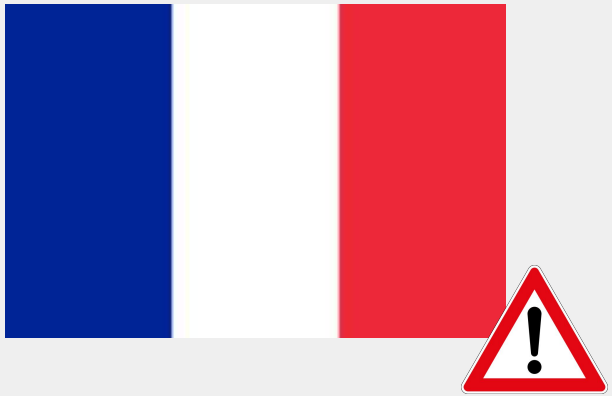
Why Multiple Clusters?

Why Multiple Clusters?

Workload Proximity

Why Multiple Clusters?

Failure Domains

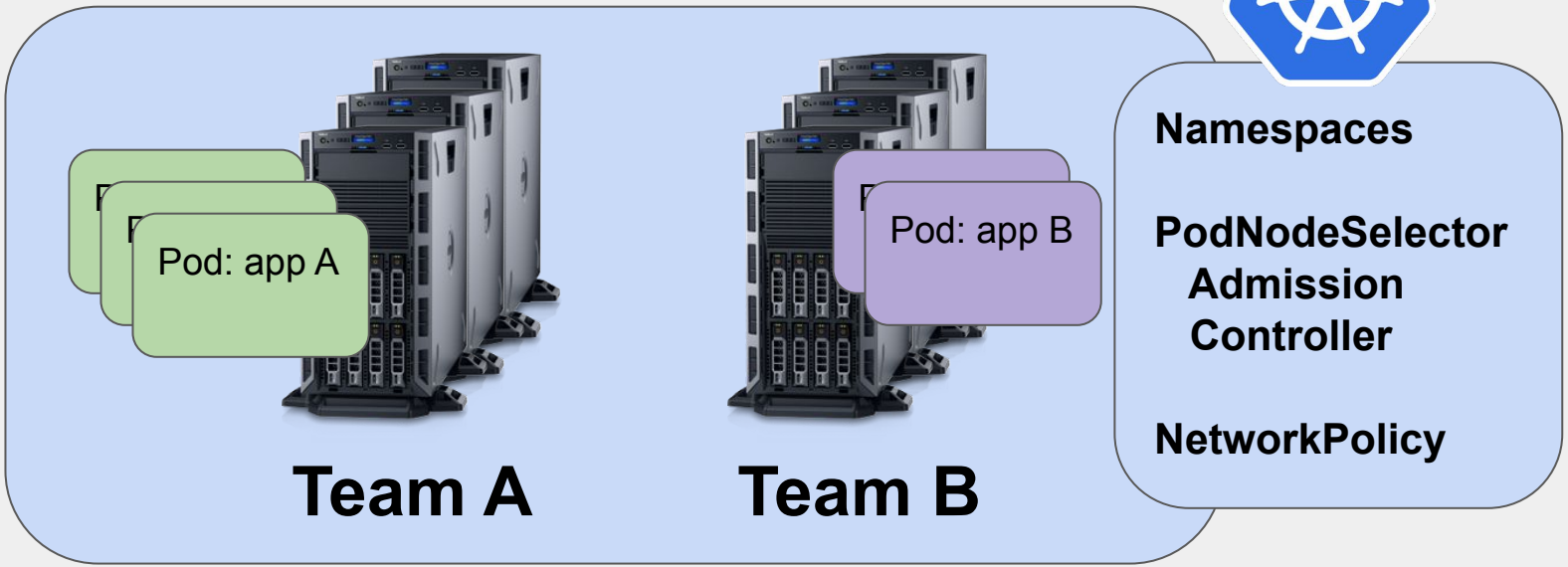


Why Multiple Clusters?

Tenancy / k8s API Complexity

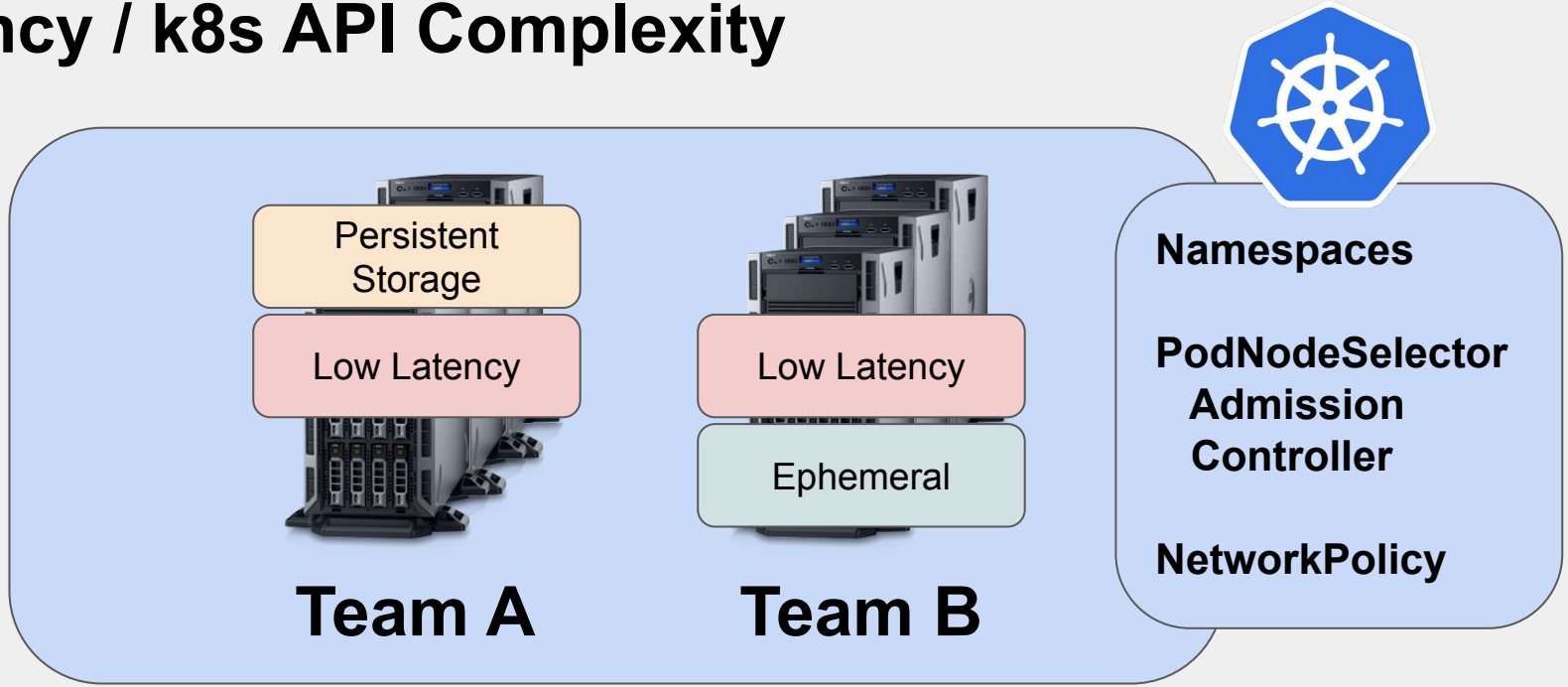
Why Multiple Clusters?

Tenancy / k8s API Complexity



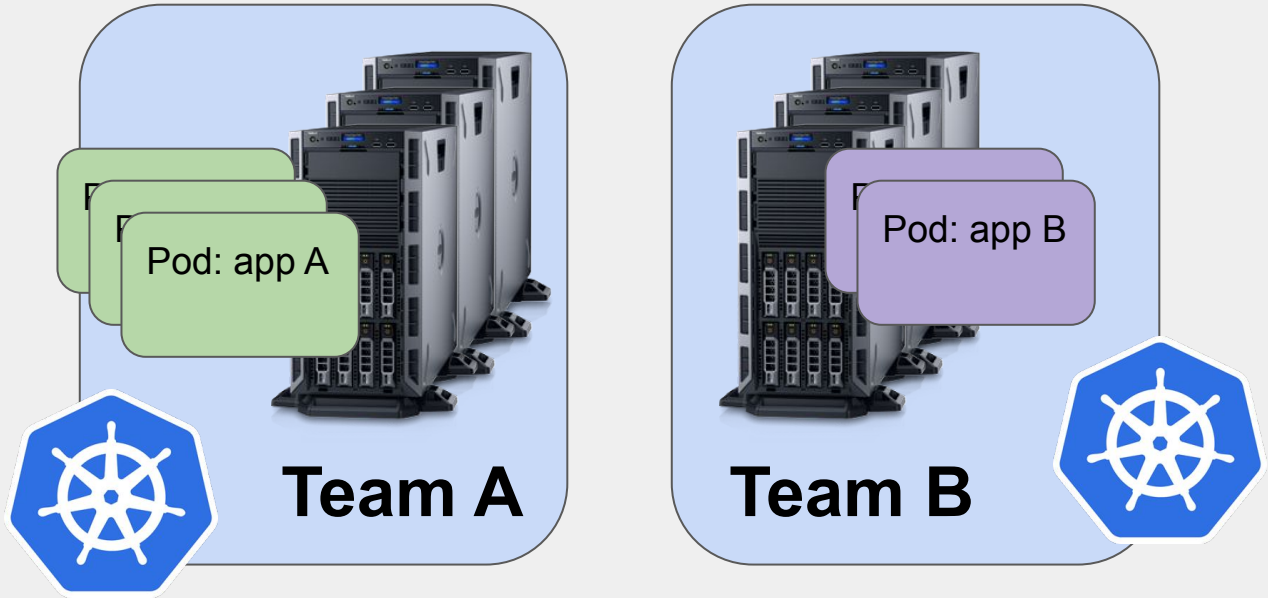
Why Multiple Clusters?

Tenancy / k8s API Complexity



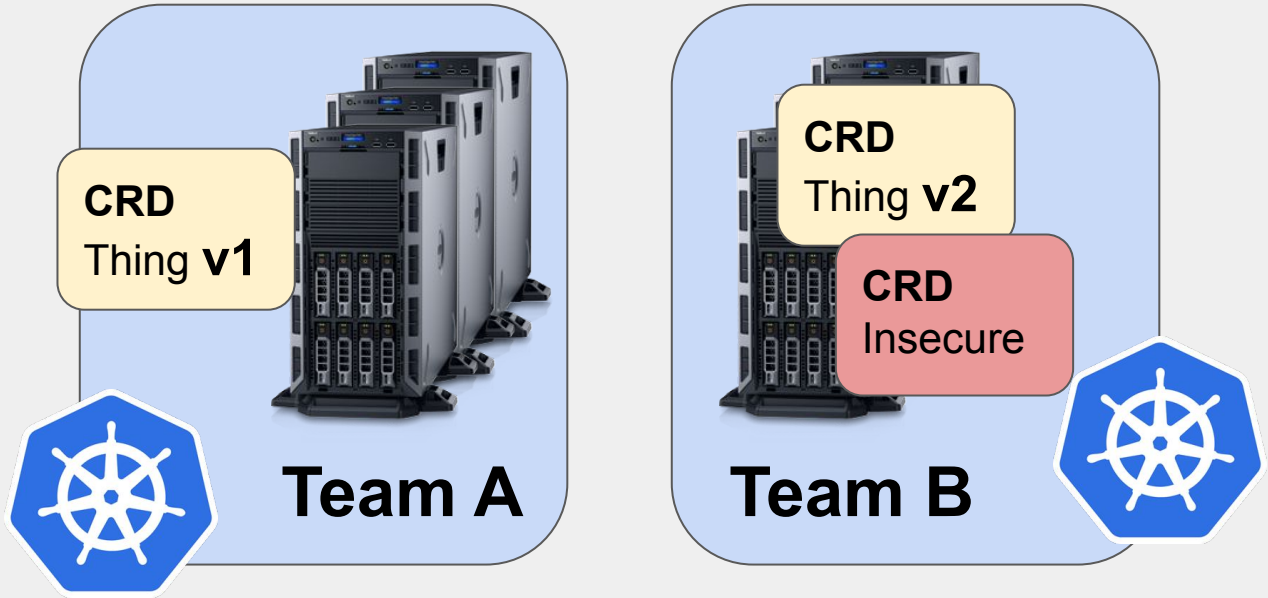
Why Multiple Clusters?

Tenancy / k8s API Complexity



Why Multiple Clusters?

Tenancy / k8s API Complexity



Why Multiple Clusters?

Reporting, Billing

Why Multiple Clusters?

Reporting, Billing

Why Multiple Clusters?

Misc.

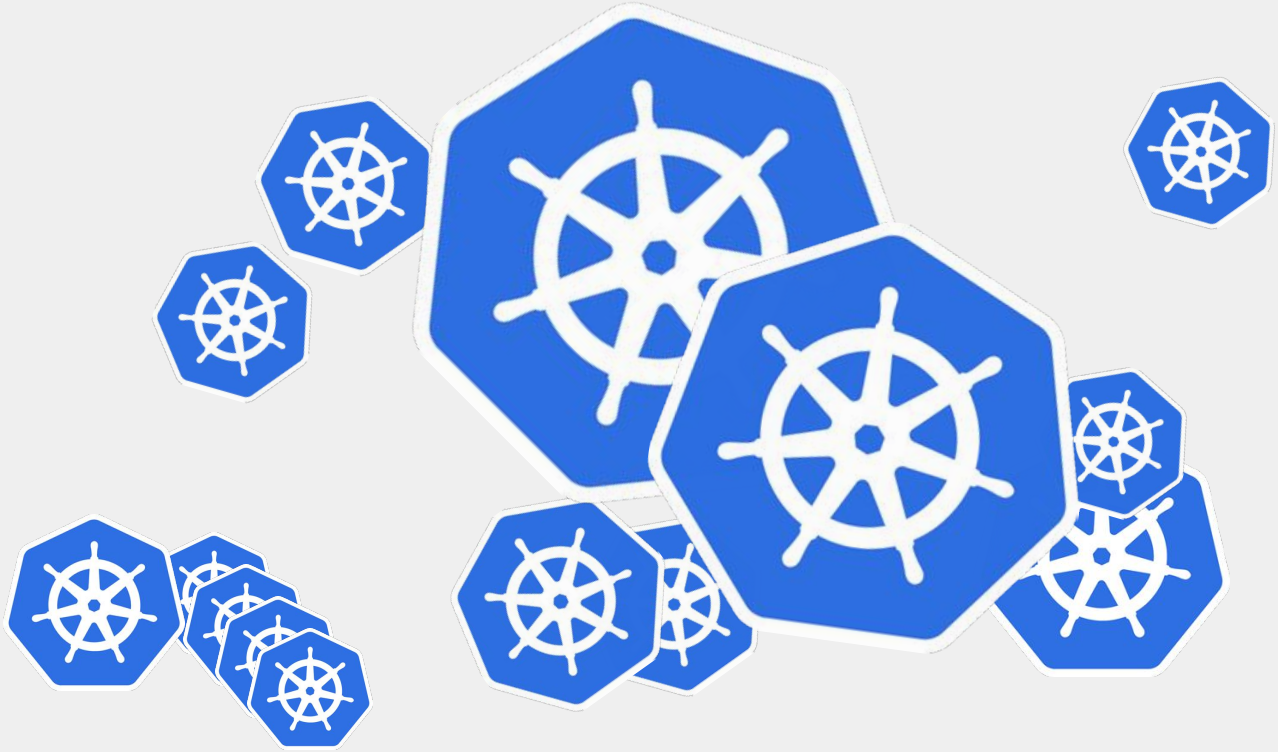
Multiple KaaS providers

Hybrid-cloud

Migrations

B2B Networks

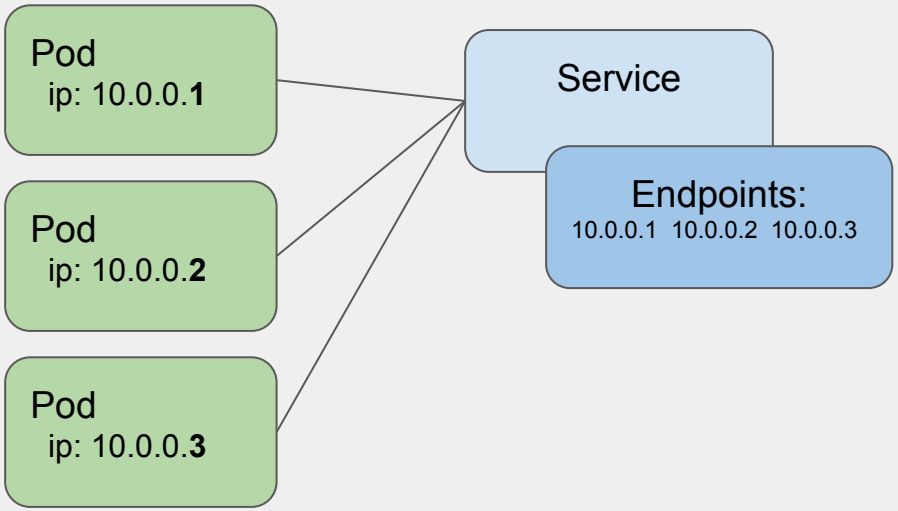
Mergers / Acquisitions

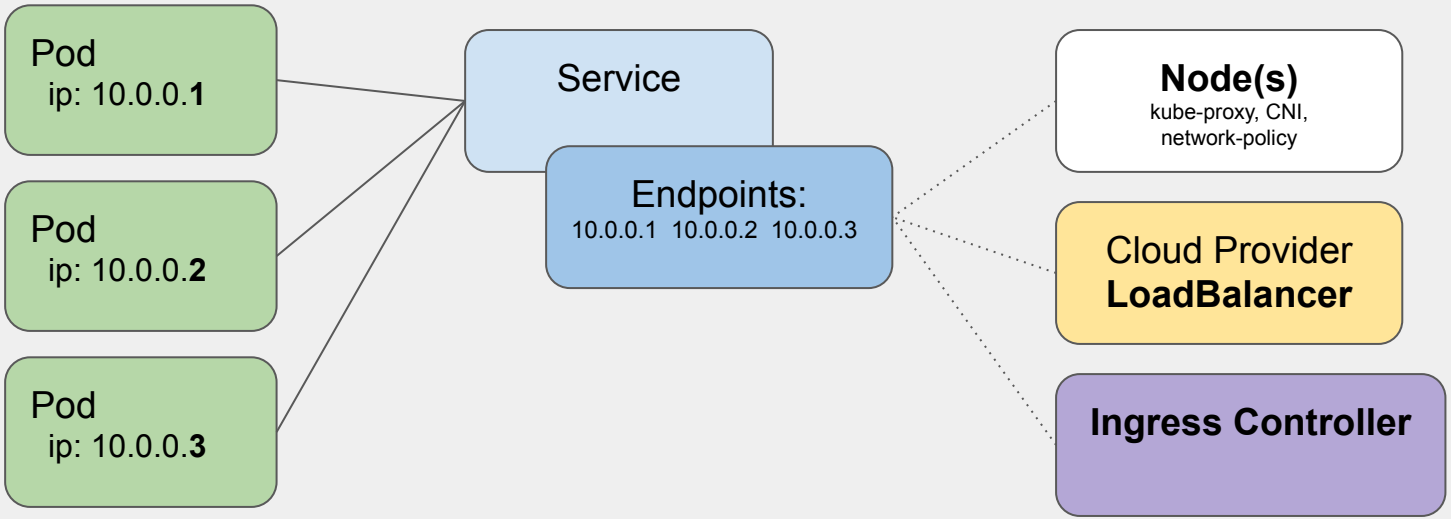


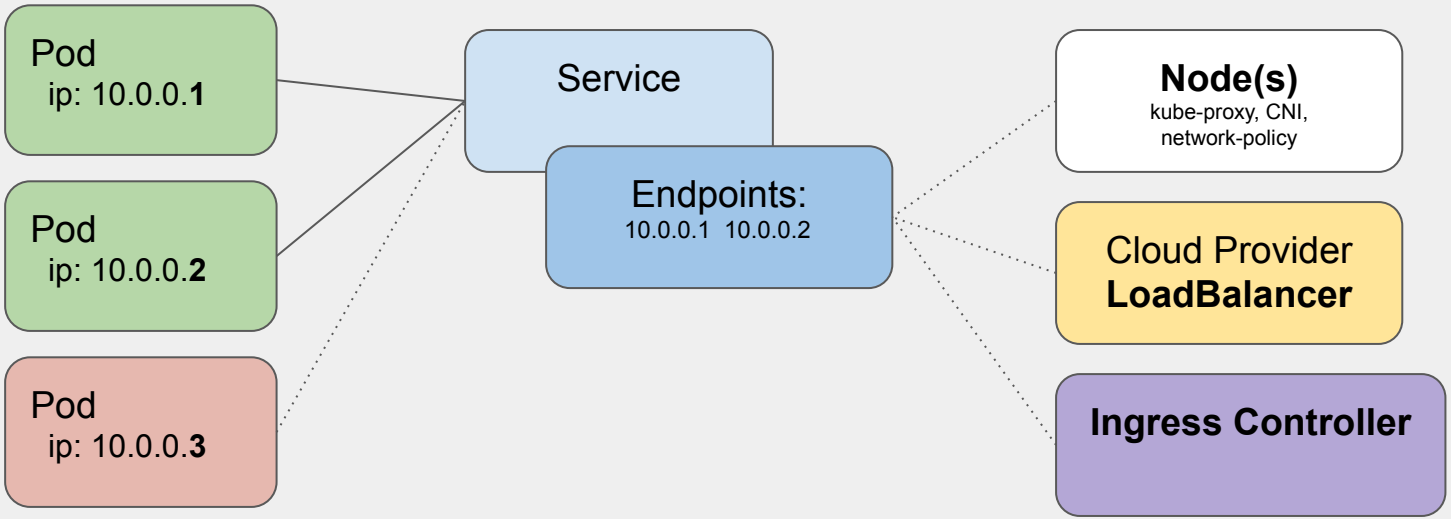
@capileigh stealthybox

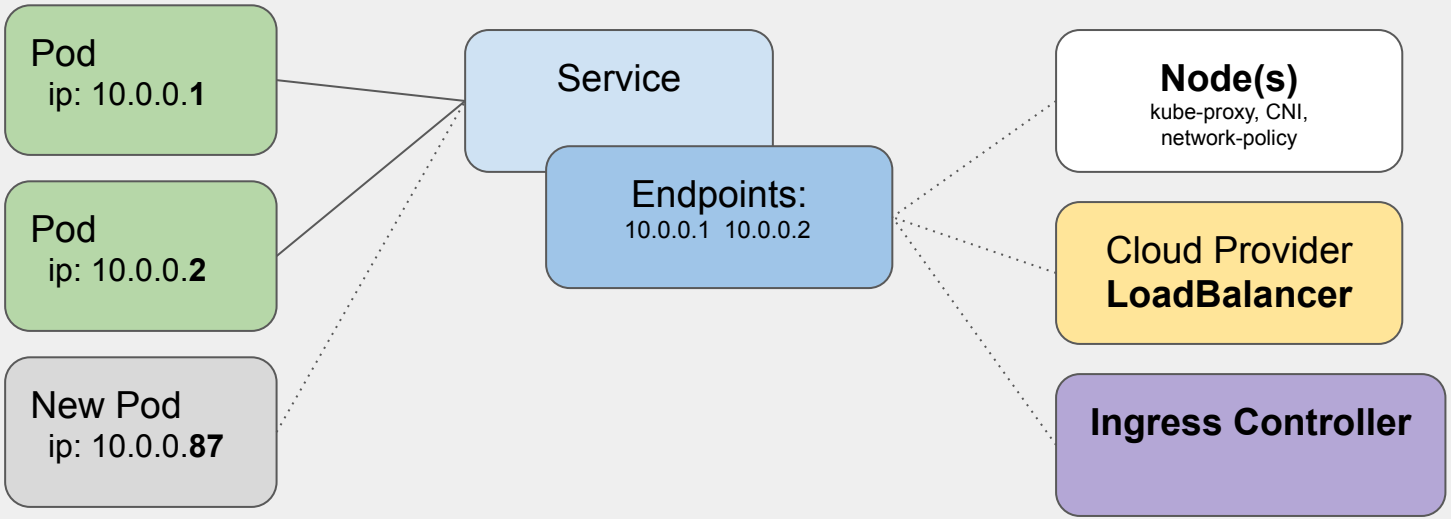
So you wanna route some **packetz ...**

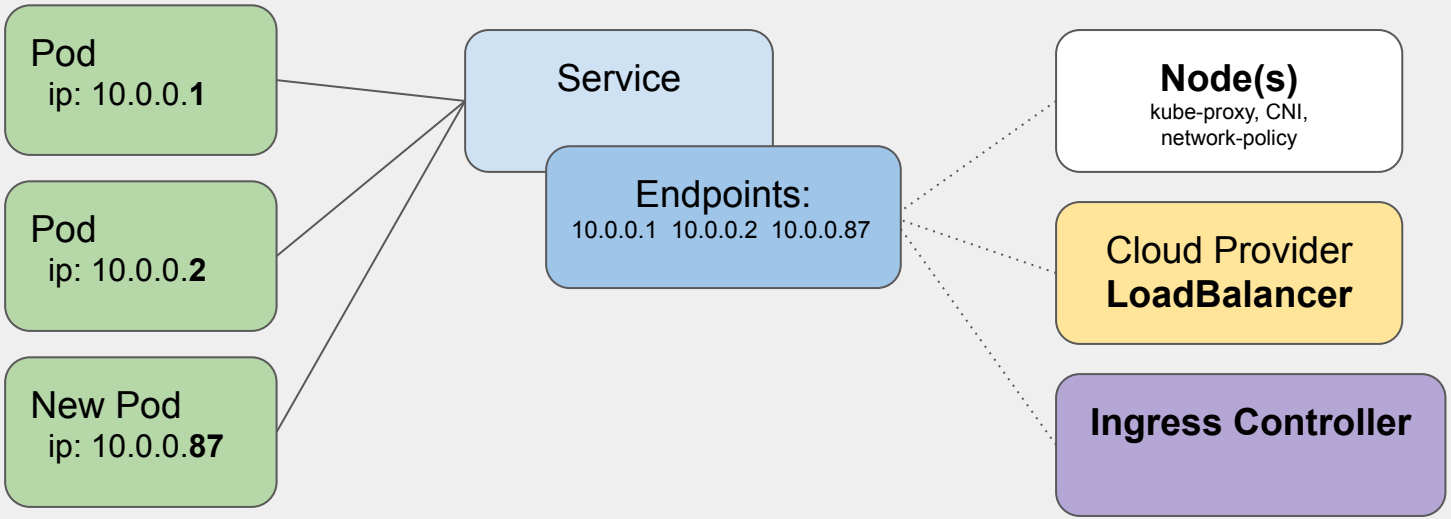

Pod
ip: 10.0.0.1
Pod
ip: 10.0.0.2
Pod
ip: 10.0.0.3

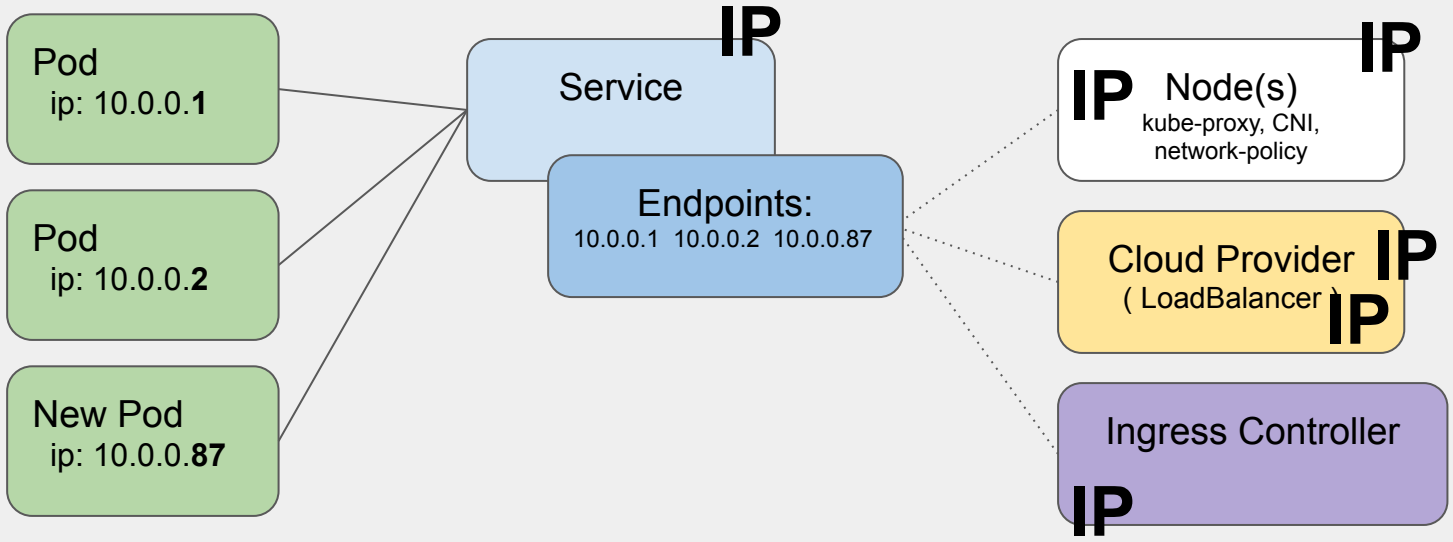

Pod
ip: 10.0.0.1
Pod
ip: 10.0.0.2
Pod
ip: 10.0.0.3
Service
Endpoints:
10.0.0.1 10.0.0.2 10.0.0.3


Pod
ip: 10.0.0.1
Pod
ip: 10.0.0.2
Pod
ip: 10.0.0.3
Service
Endpoints:
10.0.0.1 10.0.0.2 10.0.0.3
Node(s)
kube-proxy, CNI,
network-policy
Cloud Provider
LoadBalancer
Ingress Controller

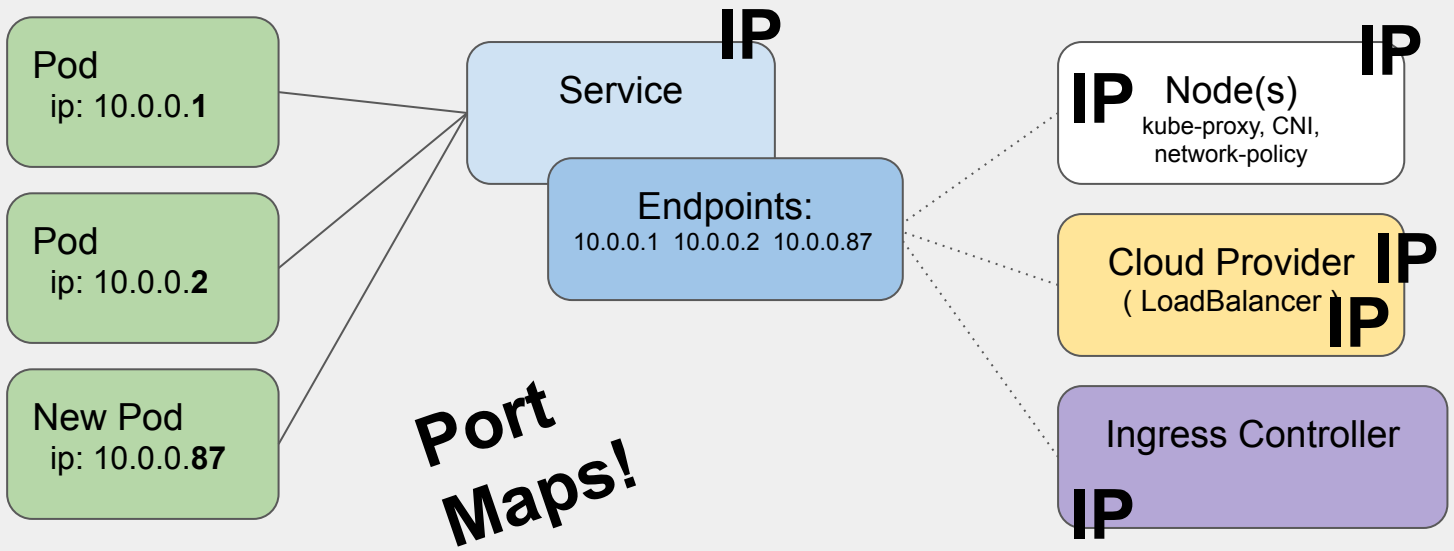

Pod
ip: 10.0.0.1
Pod
ip: 10.0.0.2
Pod
ip: 10.0.0.3
Service
Endpoints:
10.0.0.1 10.0.0.2
Node(s)
kube-proxy, CNI,
network-policy
Cloud Provider
LoadBalancer
Ingress Controller


Pod
ip: 10.0.0.1
Pod
ip: 10.0.0.2
New Pod
ip: 10.0.0.87
Service
Endpoints:
10.0.0.1 10.0.0.2
Node(s)
kube-proxy, CNI,
network-policy
Cloud Provider
LoadBalancer
Ingress Controller

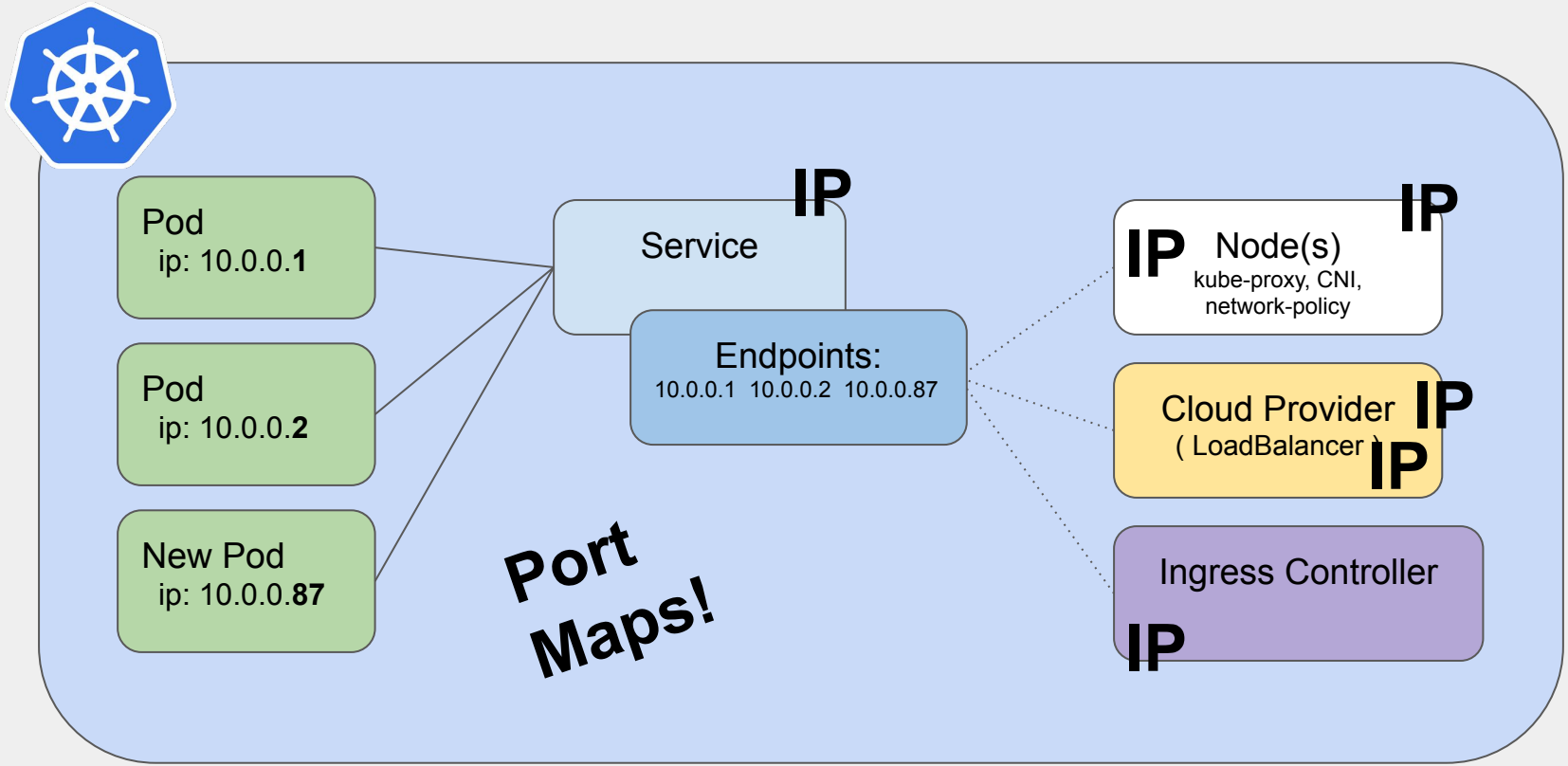

Pod
ip: 10.0.0.1
Pod
ip: 10.0.0.2
New Pod
ip: 10.0.0.87
Service
Endpoints:
10.0.0.1 10.0.0.2 10.0.0.87
Node(s)
kube-proxy, CNI,
network-policy
Cloud Provider
LoadBalancer
Ingress Controller



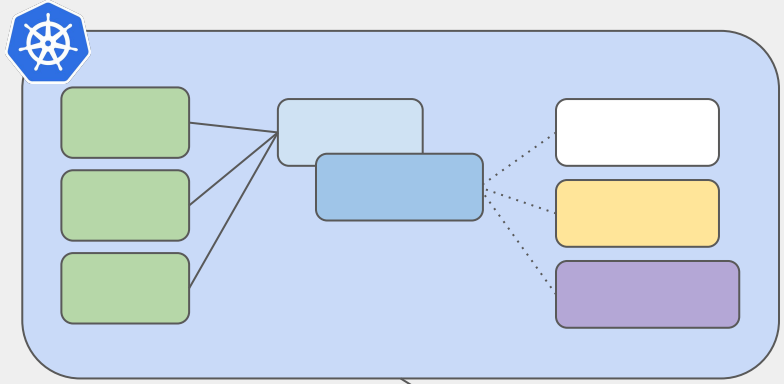
@capileigh stealthybox

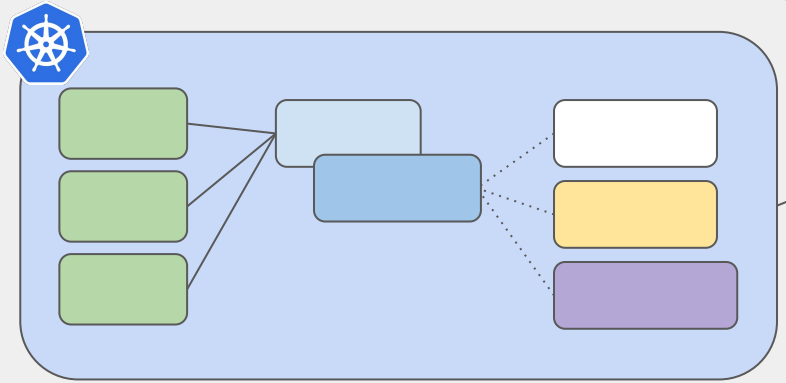


@capileigh stealthybox



@capileigh stealthybox





Δ?

Legacy
DB's & Apps

@capileigh stealthybox

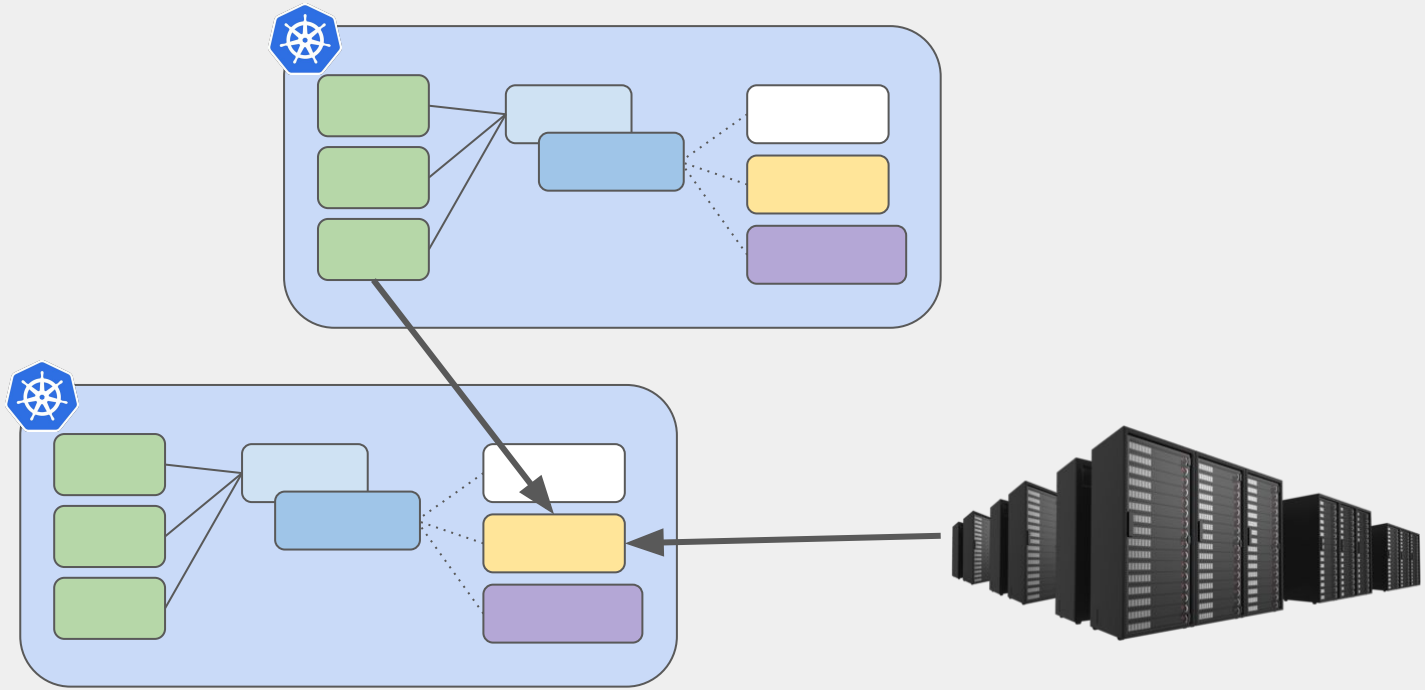
Service Type: **NodePort** / **LoadBalancer**

Layer 4 abstraction

Expose Service vIP's to outside traffic via
- Node IP's
- Cloud Provider Public or Private IP

Declarative Config:
external-dns for stable DNS
cert-manager for TLS within app Pods

Service Type: NodePort / LoadBalancer



Ingress controllers

Reverse Proxy Services based on protocol specifics.
Allows **1:1** and **1:many** setups.

Primarily a Layer 7 abstraction,
some support L4 protocols

1:many setups
reduce external network churn

Cloud Provider Ingress controllers

Cloud Providers sometimes expose
an internet or VPC accessible IP
and Load Balancer **per-Ingress (1:1)**

Still possible to route multiple Services
within a Namespace

Powerful, Often **$$$**

Self-managed Ingress controllers

Third-party / Self-managed solutions are often deployed in-cluster for **(1:many)** controllers for a number of Ingress resources.

- Composes well behind an L4 NodePort/LoadBalancer
- In-cluster Ingresses can be used to form a mesh

Ingress controllers

Declarative Config:

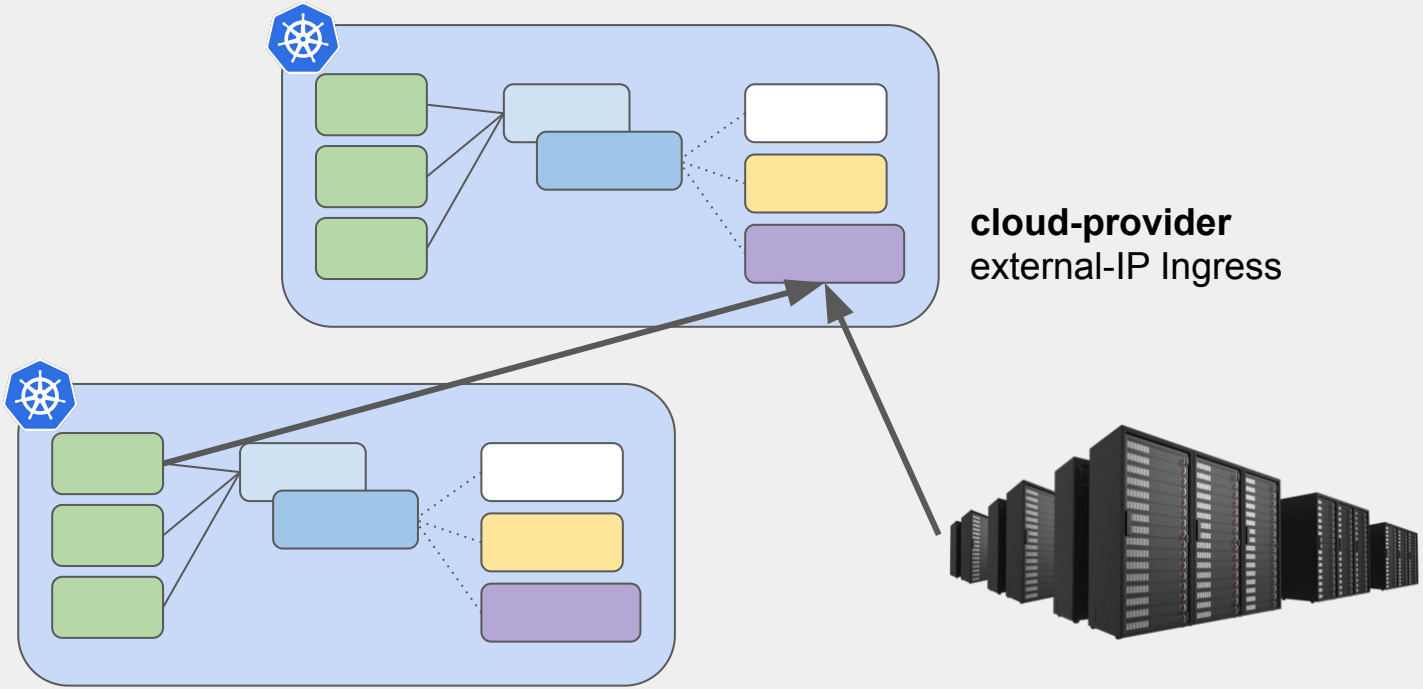
external-dns for stable DNS

cert-manager for terminated TLS

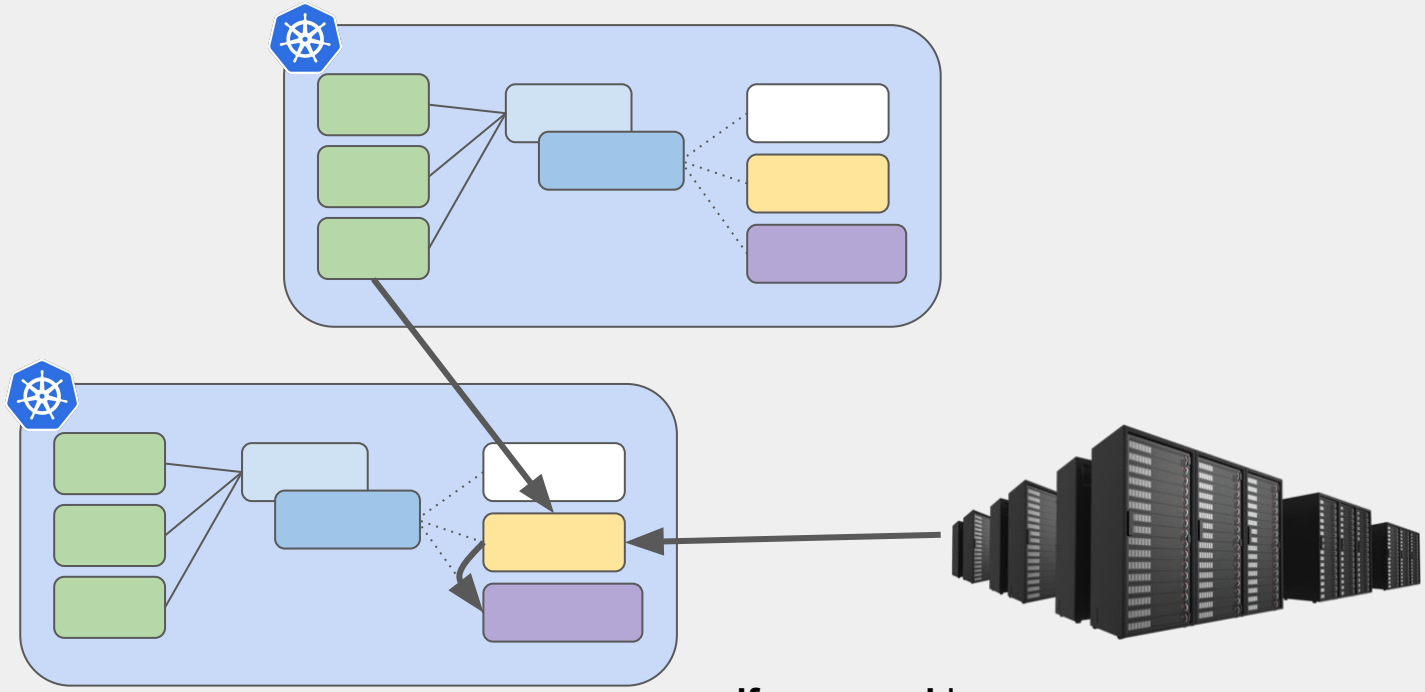
Use wildcard DNS/TLS for lower deploy-latency

* requires API-enabled DNS zone for ACMEv2

Ingress

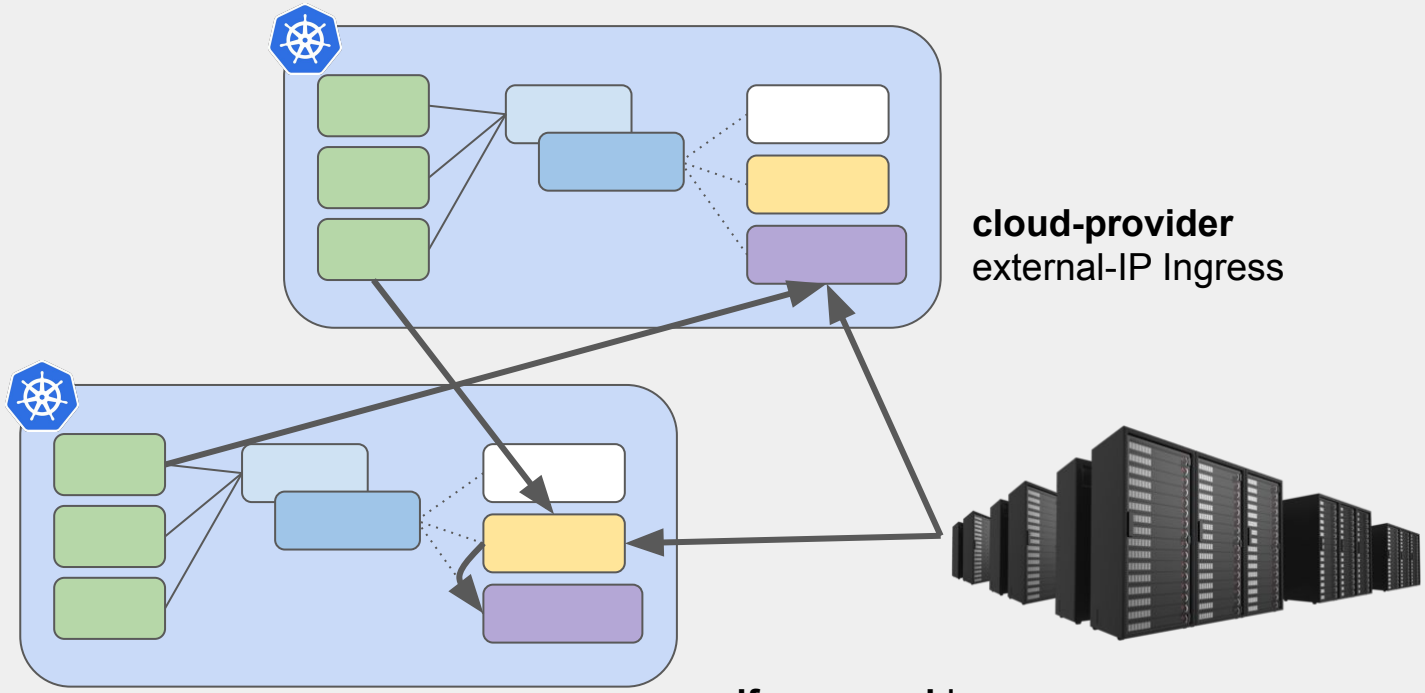


Ingress



self-managed Ingress
behind cloud LoadBalancer

Ingress



cloud-provider
external-IP Ingress

self-managed Ingress
behind cloud LoadBalancer

Route Sharing

Make Pod IP's natively routable
beyond kubernetes Nodes

Use-case:
Run an Ingress Controller outside the Cluster

Route Sharing

Make Pod IP's natively routable
beyond kubernetes Nodes

Use-cases:
Run an Ingress Controller outside the Cluster
*Run an Ingress Controller **inside another** Cluster ;)*

Route Sharing

It’s possible using Endpoints to determine & share which **Nodes** a **Service** has **Ready Pods** on.

You can advertise all of these Nodes’ IP’s as available routes for the Service vIP’s!

ECMP (equal-cost multi-path) routing:
load-balance / failover to those Nodes

Route Sharing

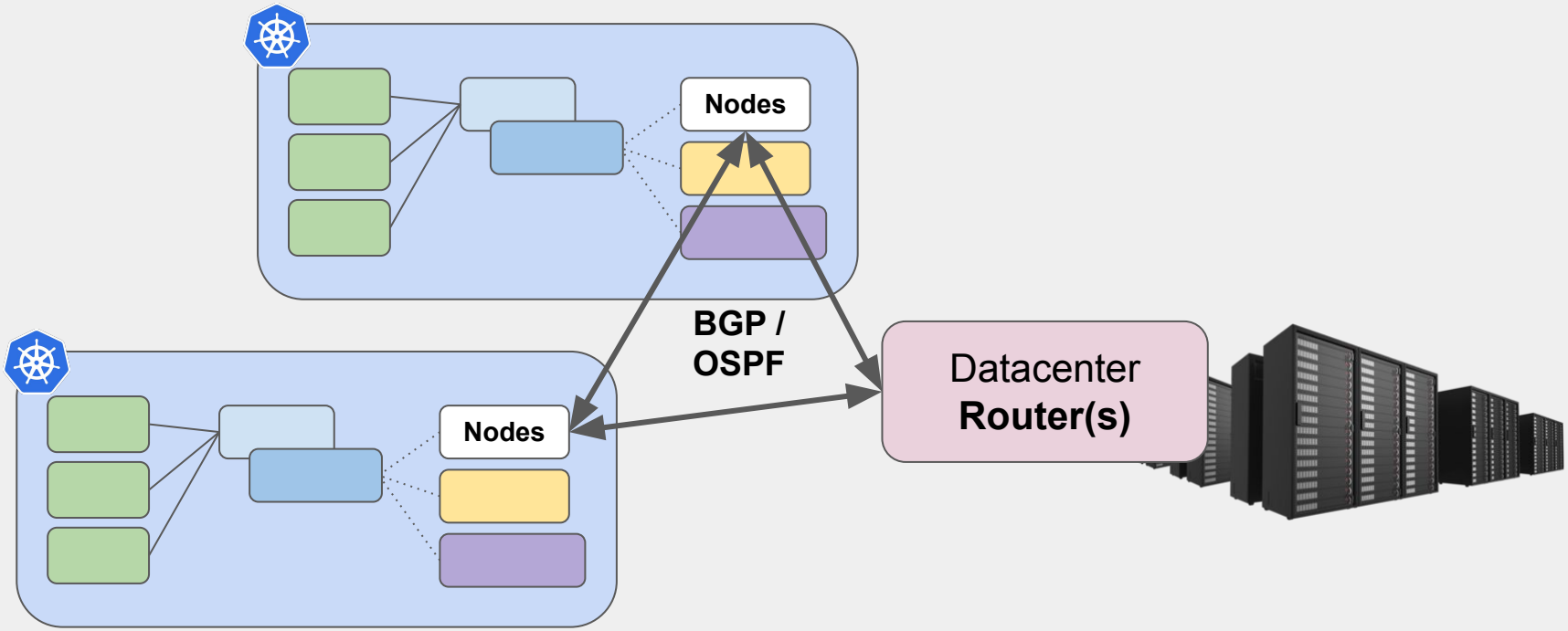
BGP - Border Gateway Protocol

Share routes between Autonomous Systems
Makes the Internet work (it's not that scary)

OSPF - Open Shortest Path First

Popular protocol within
private networks (Single AS)

Route Sharing



Route Sharing

Note:

Each cluster needs a **unique** Service-subnet and Pod-subnet for multi-cluster route-sharing to make sense!

Route Sharing

BGP CNI's:

- Calico (bird)
- Kube-router (can replace kube-proxy)
- Romana

Also see:

- FRRouting / Quagga (OSPF)
- Your own router

What about DNS?

`app.web.svc.cluster.local`

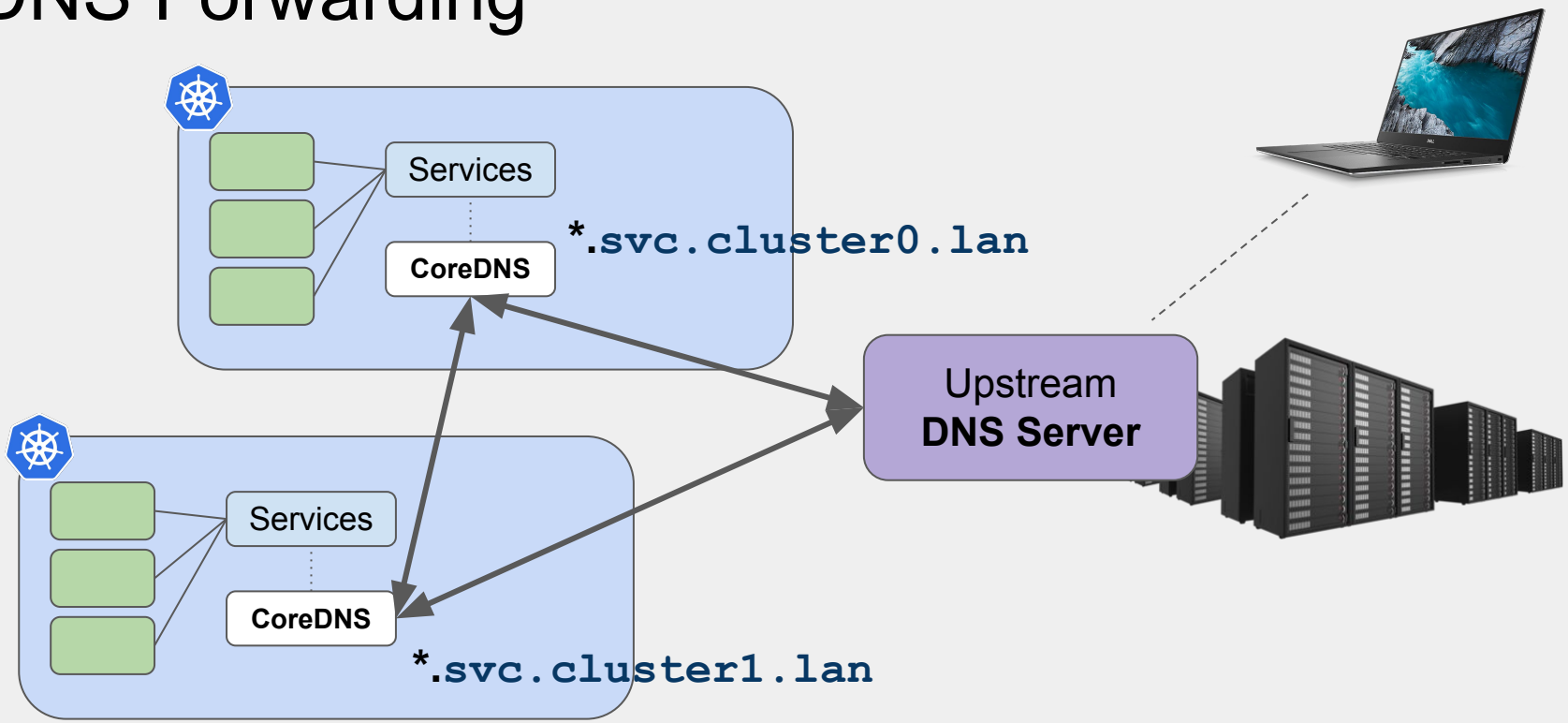
DNS Forwarding

CoreDNS is the primary resolver in many k8s clusters

CoreDNS is extensible and very configurable

Reachable via the 10th IP in the
Service Subnet of a cluster

DNS Forwarding



Route Sharing + DNS Forwarding

Extends the native k8s service-discovery and routing mesh beyond the cluster

1-hop Service routing for cross-cluster and non-Kubernetes workloads

Route Sharing + DNS Forwarding

Extends the native k8s service-discovery and routing mesh beyond the cluster

1-hop Service routing for cross-cluster and non-Kubernetes workloads

Layer 4 abstraction
No built-in TLS / mTLS
Prereq. for multi-cluster Service Meshes

Node-network Overlays / VPN's

CNI

- weave net multicast ok (L2)
- Cilium wireguard + multi-cluster mesh

2-way UDP hole-punching / multi-NAT traversal

- Tailscale wireguard
- slackhq/nebula multicast ok
- ZeroTier multicast ok

Going further:

Network Policy:

Controllers can be made aware of
Pod identity from other clusters

KEP-1645: kubernetes-sigs/mcs-api

Cross-Cluster Service Meshes

Inter-Cluster orchestration

Ex: Flux dependencies

+ Flagger kStatus (WIP)

https://github.com/
stealthybox/multicluster-gitops

@capileigh

stealthybox