KubeCon | CloudNativeCon

North America 2018

cilium

Connecting Kubernetes Clusters Across Cloud Providers

Thomas Graf, Co-Founder & CTO, Isovalent

@tgraf__

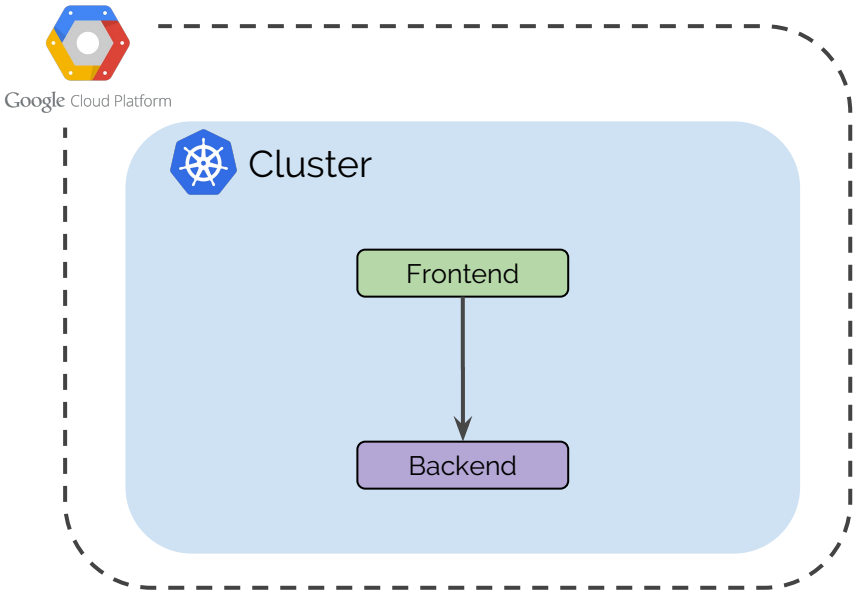
About the Speaker

Thomas Graf

- Linux kernel developer for many years at Red Hat
- Working on networking, security and BPF
- Founder of the Cilium project

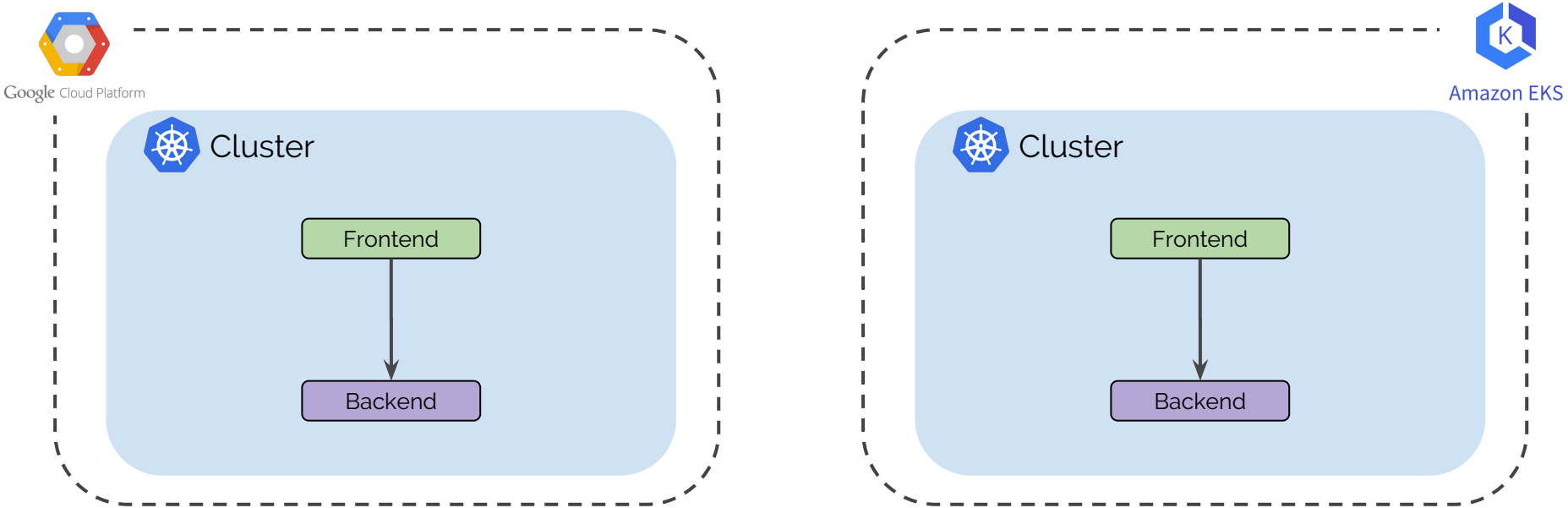
Goal of this Session:

Run Services Across Cloud providers



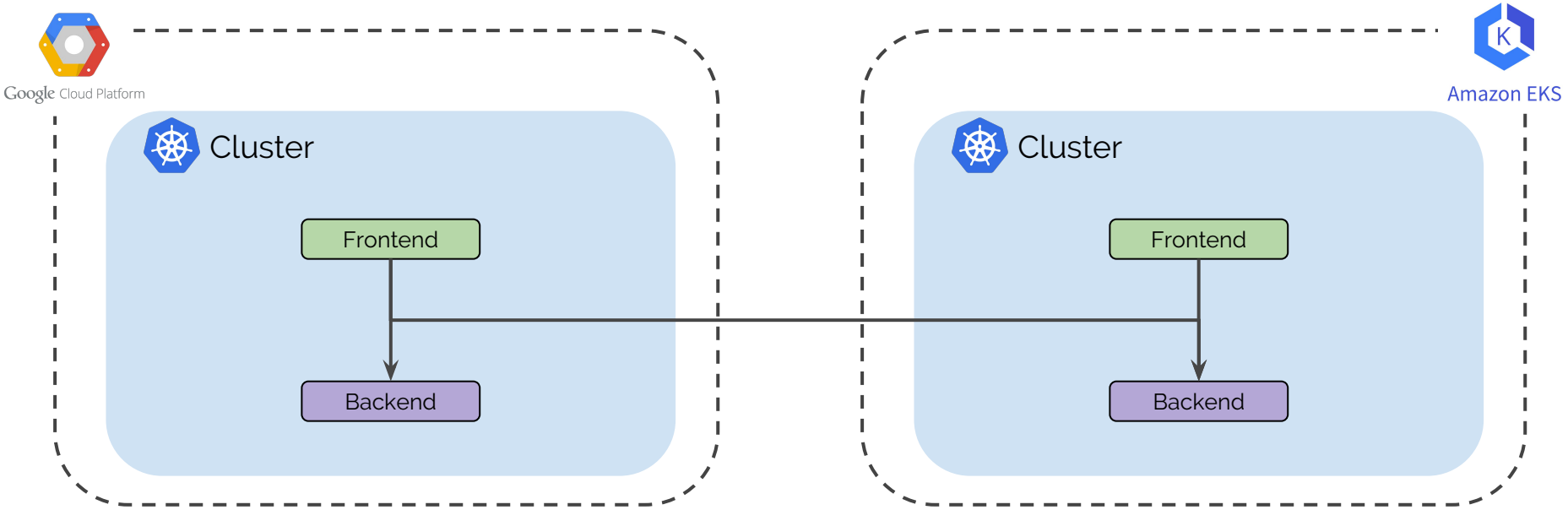
Goal of this Session:

Run Services Across Cloud providers

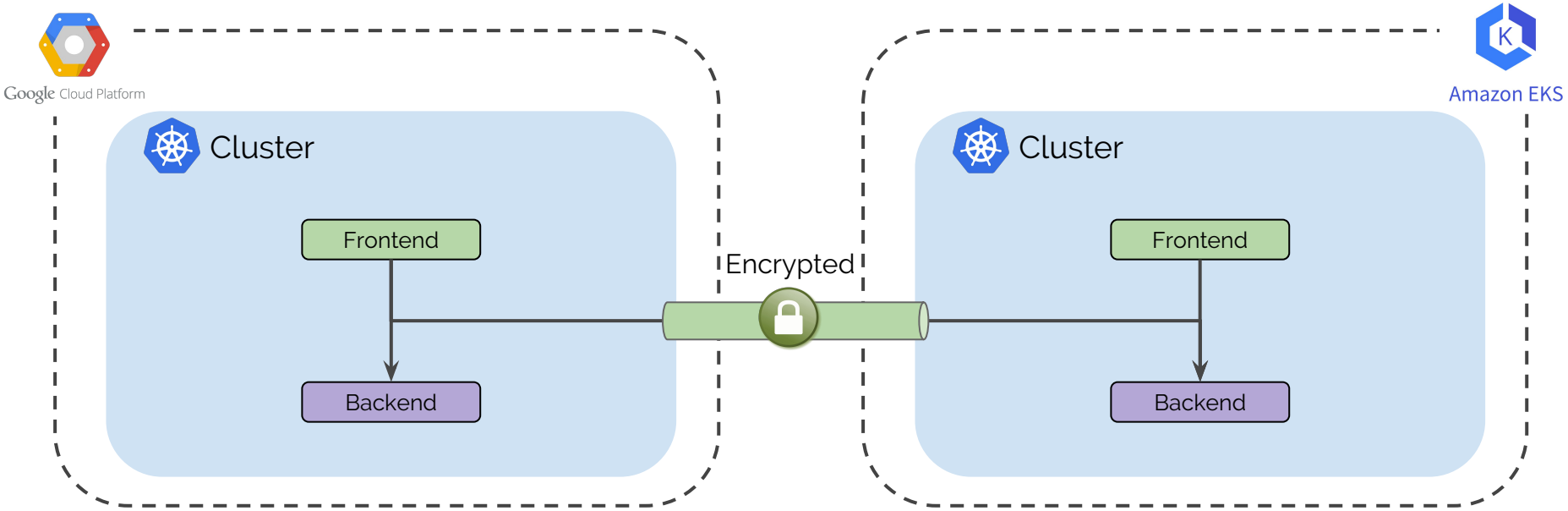


Goal of this Session:

Run Services Across Cloud providers

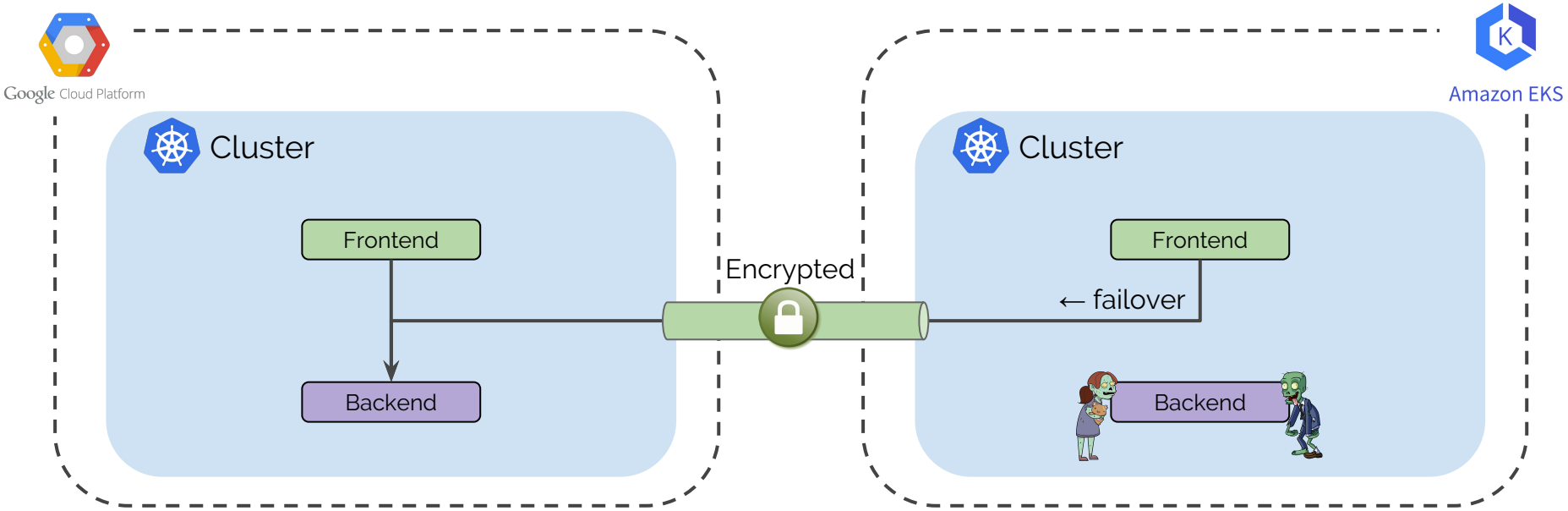


Goal of this Session:
Run Services Across Cloud providers



Goal of this Session:

Run Services Across Cloud providers



What Tools do we need?

Kubernetes

- Open Source (Apache)
- Managed or self-managed
- Kubernetes services

Infrastructure APIs

- VPC concept with routing
- IPSec compatible VPN Gateway with IKEv1 support

Cilium

- Open Source (Apache)
- Based on new BPF technology
- Networking (CNI)
- Kubernetes services
  - Replacing kube-proxy
  - Multi-cluster capability (1.4)
- Network security
  - Identity-based, DNS aware, API aware, data protocol aware
  - Transparent encryption (1.4)
- Envoy/Istio Integration
  - Sidecar Acceleration
  - Transparent SSL visibility (kTLS)

What is BPF?

Highly efficient sandboxed virtual machine in the Linux kernel. Making the Linux kernel programmable at native execution speed.

Jointly maintained by Cilium and Facebook engineers with collaborations from Google, Red Hat, Netflix, and many others.

```
$ clang -target bpf -emit-llvm -S \
   32-bit-example.c
$ llc -march=bpf 32-bit-example.ll
$ cat 32-bit-example.s
    cal:
      r1 = *(u32 *)(r1 + 0)
      r2 = *(u32 *)(r2 + 0)
      r2 += r1
      *(u32 *)(r3 + 0) = r2
      exit
```

Who uses BPF?

Every packet toward facebook.com has been processed by BPF/XDP enabled application since May, 2017

Nikita V. Shirokov, Facebook Traffic team
Linux Networking Summit 2018

Source:
http://vger.kernel.org/lpc_net2018_talks/LPC_XDP_Shirokov_v2.pdf

Who uses BPF?

- L3-L4 Load balancing
- Network security
- Traffic optimization
- Profiling

- Working upstream to replacing iptables with BPF
- Profiling & Tracing

Google

- QoS & Traffic optimization
- Network Security
- Profiling

NETFLIX

- Performance Troubleshooting & Monitoring
- Check out bpftrace and Brendan Gregg's blog posts

Who contributes to BPF?

380 Daniel Borkmann (Cilium, Maintainer)
161 Alexei Starovoitov (Facebook, Maintainer)
160 Jakub Kicinski Netronome
110 John Fastabend (Cilium)
96 Yonghong Song (Facebook)
95 Martin KaFai Lau (Facebook)
94 Jesper Dangaard Brouer (Red Hat)
74 Quentin Monnet (Netronome)
45 Roman Gushchin (Facebook)
45 Andrey Ignatov (Facebook)

Top contributors of the total 186 contributors to BPF from January 2016 to November 2018.

Connecting  Clusters with  cilium &  

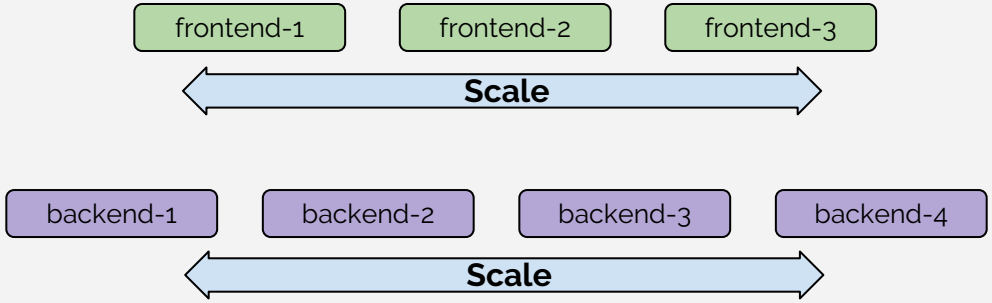
Deployments & Pods

Deployment View

Frontend
replicas=3

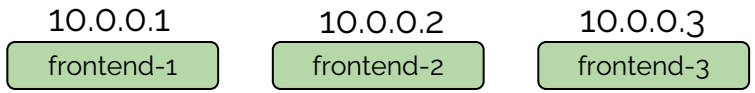
Backend
replicas=4

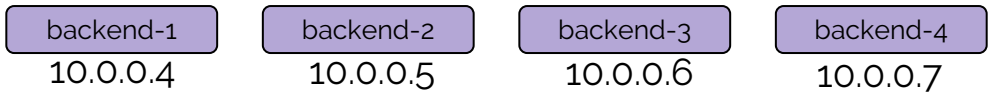
Pod View



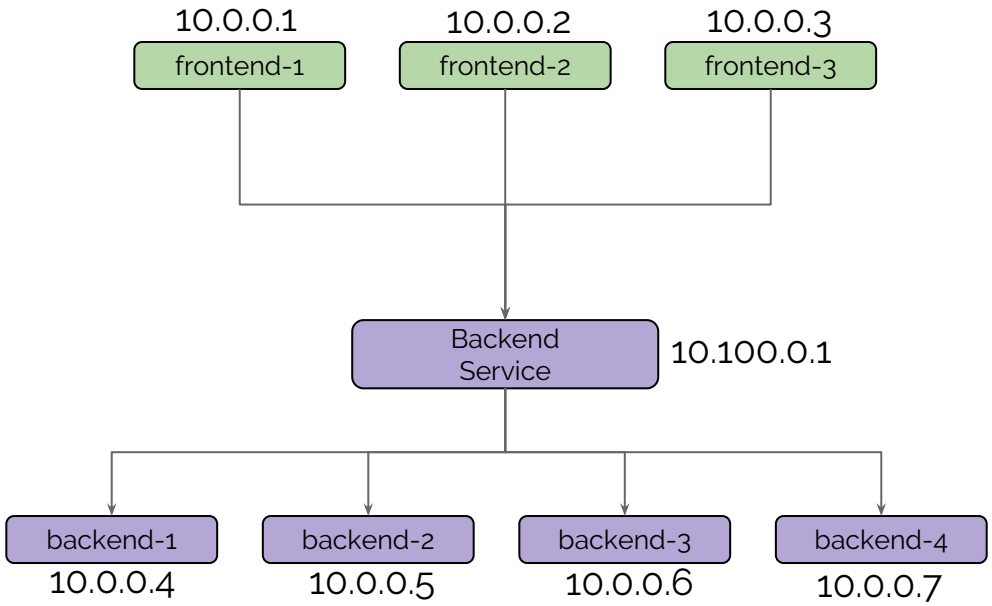

```
$ kubectl get deployment backend
NAME         DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
backend      4         4         4            4           1d
```

What is a Service?



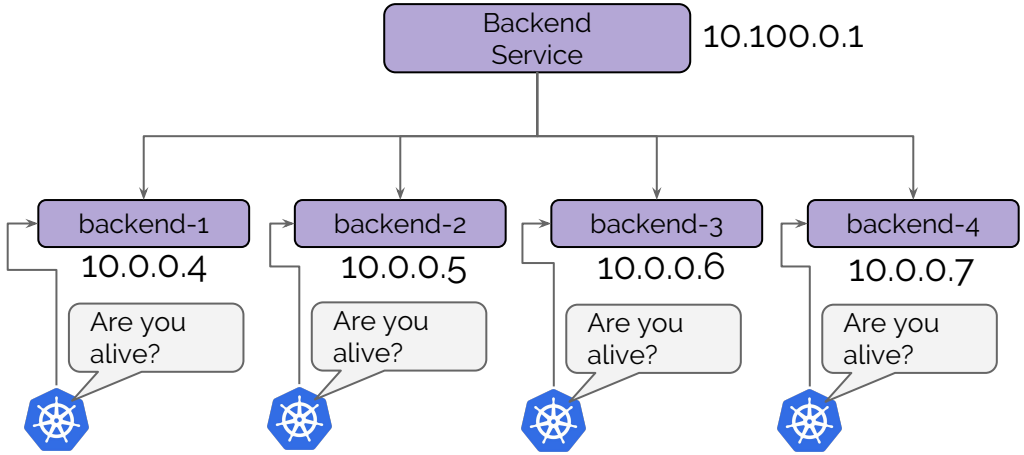


What is a Service?



Health Checks

```
[...]
livenessProbe:
  exec:
    command:
    - check-status
  failureThreshold: 3
  periodSeconds: 2
```



What are Endpoints?

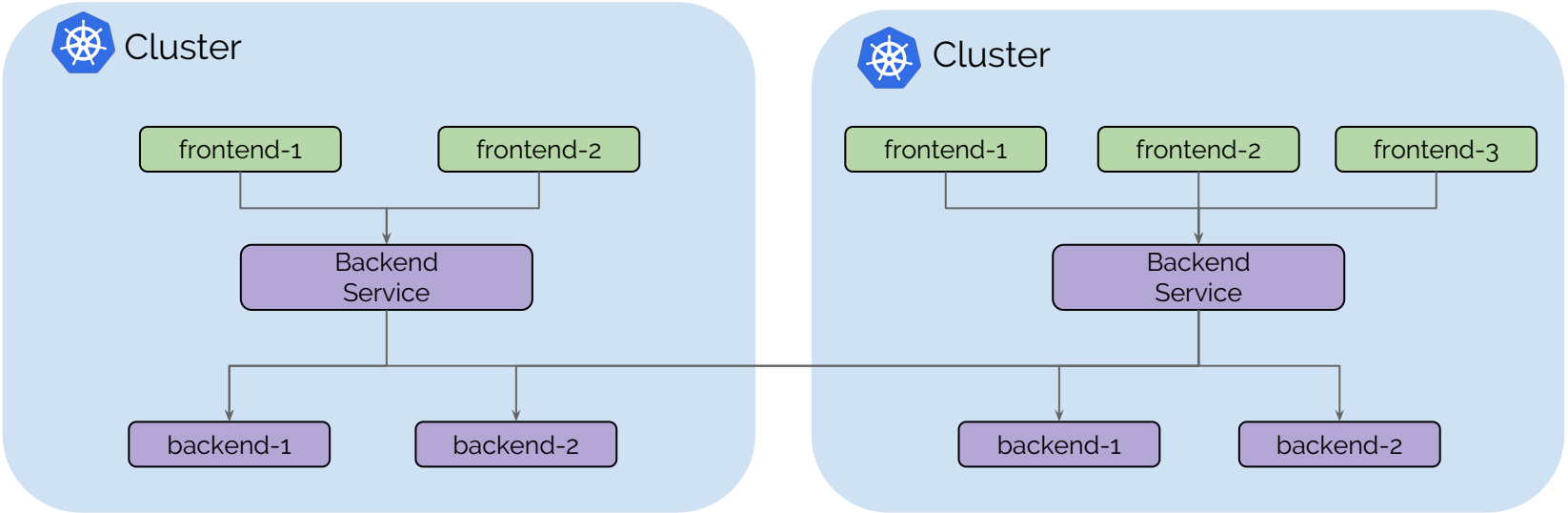

```
$ kubectl get svc backend
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
Backend      ClusterIP   10.39.245.245   <none>        80/TCP    1d

$ kubectl get endpoints backend
NAME         ENDPOINTS                         AGE
Backend      10.36.1.219:80,10.36.2.249:80     1d
```

- Kubernetes creates a shadow `Endpoints` object for every Service.
- The Endpoints object lists all pod IPs and port mappings of healthy pods based on the liveness health-check.

Advanced:

- You can maintain the `Endpoints` object as a user as well.

Global Services




```
metadata:
  annotations:
    io.cilium/global-service: "true"
```

Demo

Design Principles

Simple

- Simple to use
  - Standard Kubernetes Services)
  - Avoid need for networking degree
- Simple to troubleshoot & debug

Secure

- Encryption
- Security policies spanning clusters with identity-based enforcement
- Mutual TLS compatibility

Resilient

- Preserve and respect availability zones and failure domains.
- Failures in one cluster should not impact other clusters.
- Avoid requirement of Kubernetes clusters to be aware of each other

Efficient

- Native networking speeds
- Direct pod to pod connections without intermediate termination (proxies).

Supported Service Annotations

`io.cilium/global-service: {true|false}`

Whether to include endpoints of other clusters.

`io.cilium/shared-service: {true|false}`

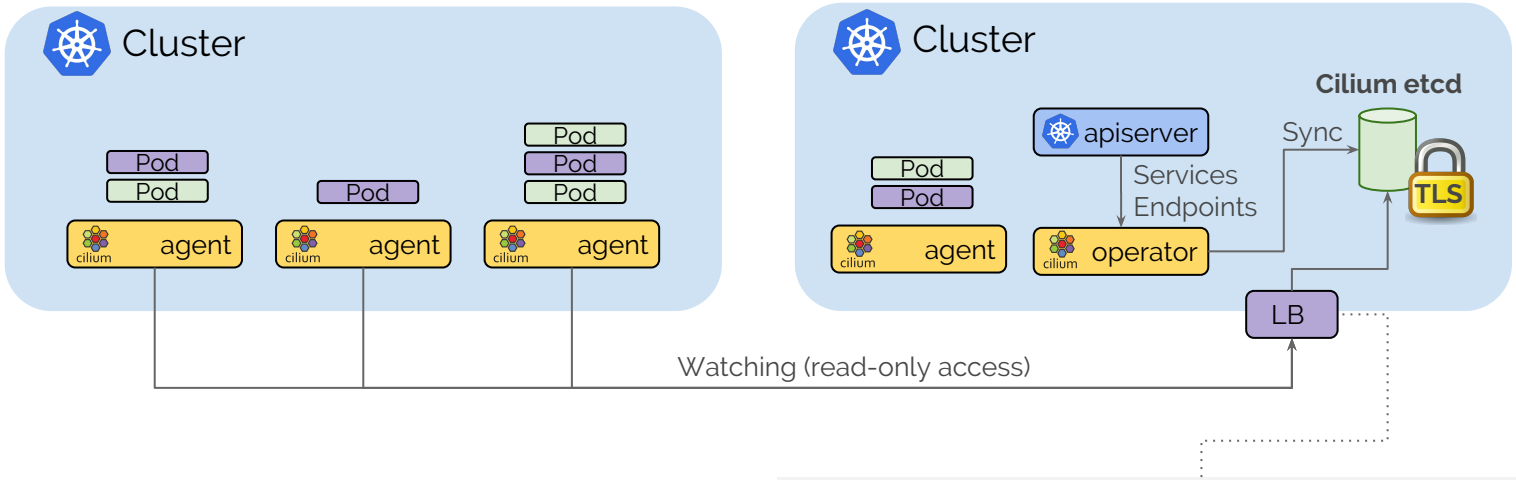
Whether to share local with other clusters. Defaults to true if global-service.

Coming Soon:

`io.cilium/service-affinity: {local-cluster|local-node|remote|none}`

Whether to prefer local or remote endpoints.

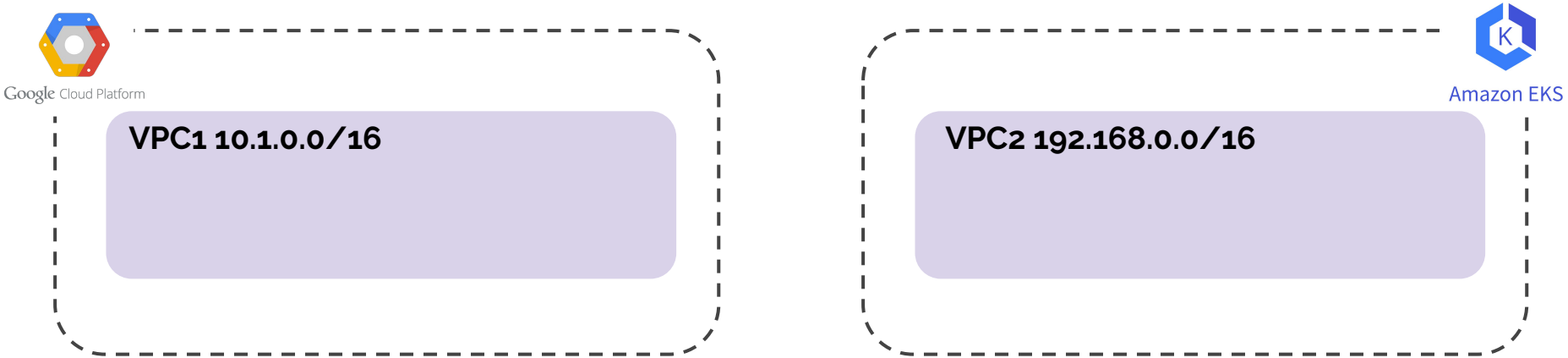
ClusterMesh Control Plane



- Control plane access to other clusters is always read-only (notification stream).
- Not all clusters must be aware of each other

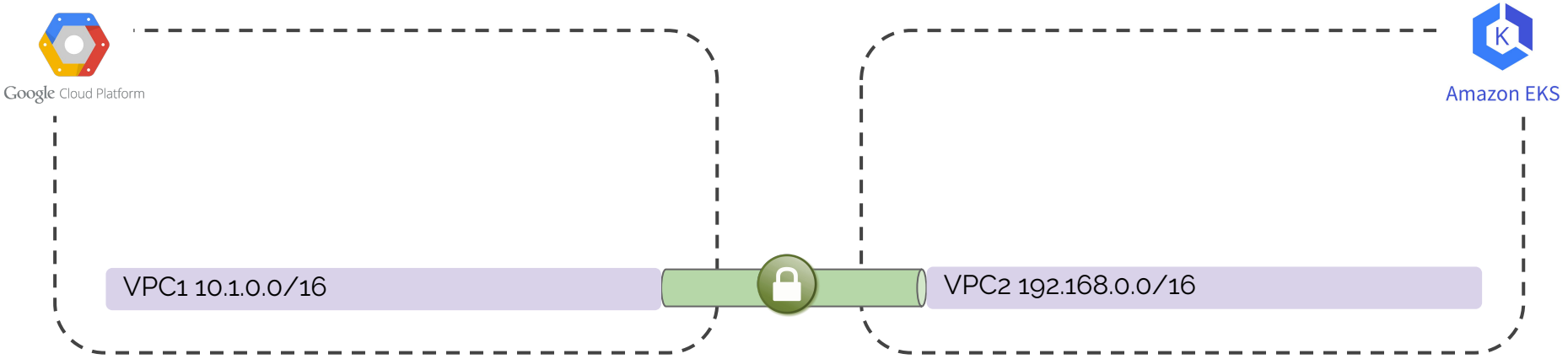
```
apiVersion: v1
kind: Service
metadata:
  name: cilium-etcd-external
  annotations:
    cloud.google.com/load-balancer-type: "Internal"
```

Step-by-step Setup



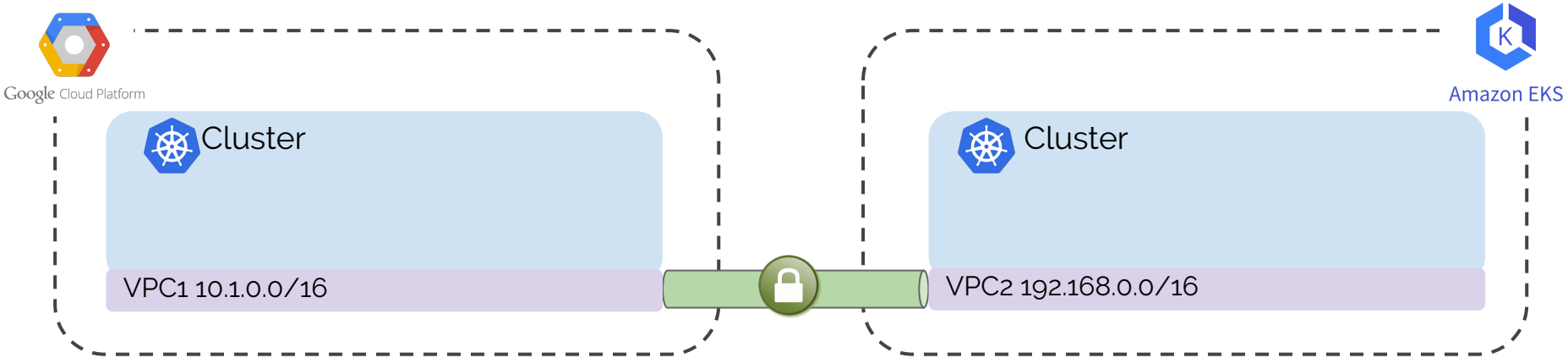
- Create VPCs in each cloud provider with non-overlapping CIDR ranges.
- Support for overlapping CIDR ranges is possible but only complicates the setup.

Step-by-step Setup



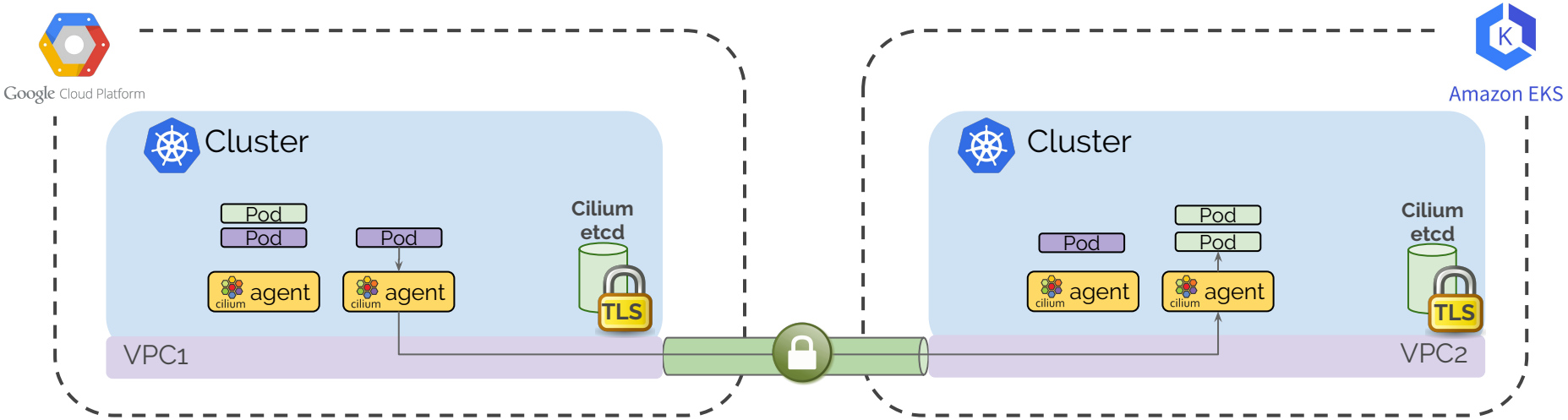
- Create a VPN gateway and redundant VPN tunnels using IPSec. (Instructions in references)
- IPSec is the standard. All cloud providers support it.
- Setup routing to route from VPC1 to VPC2 via VPN and vice versa.

Step-by-step Setup



- Setup Kubernetes clusters using the created VPCs.

Step-by-step Setup

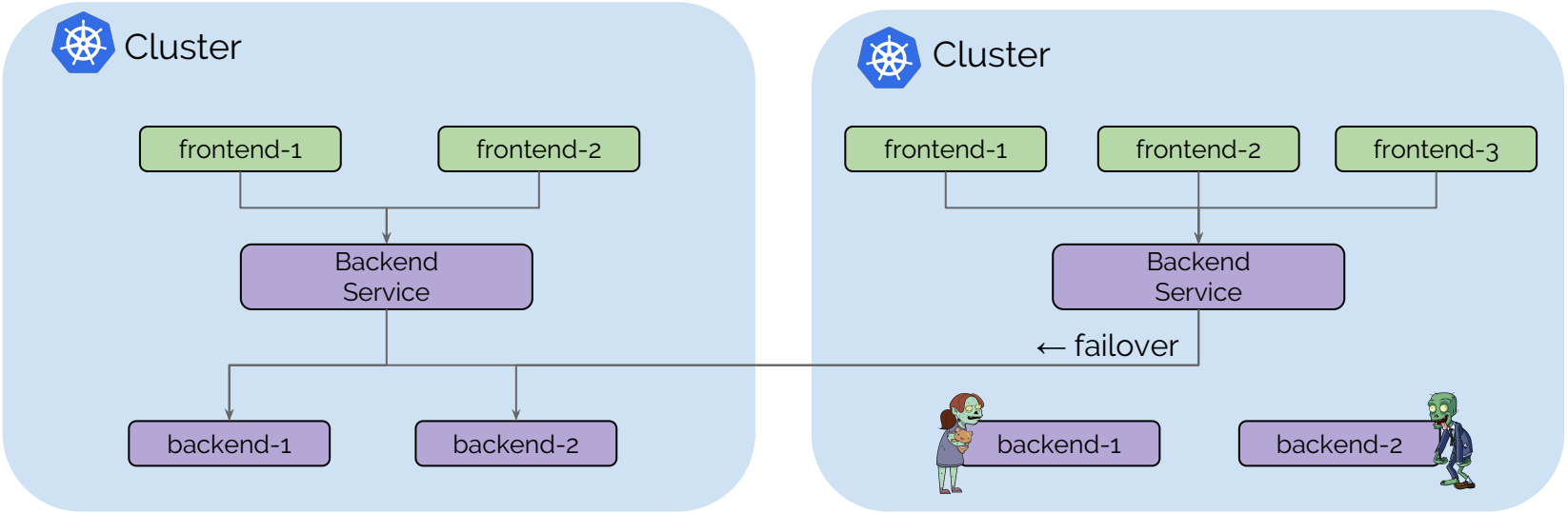


- Deploy Cilium
- Setup internal LoadBalancer to expose control plane on VPC
- Deploy Kubernetes secrets (`clustermesh-secrets`) to establish connections

Use Cases

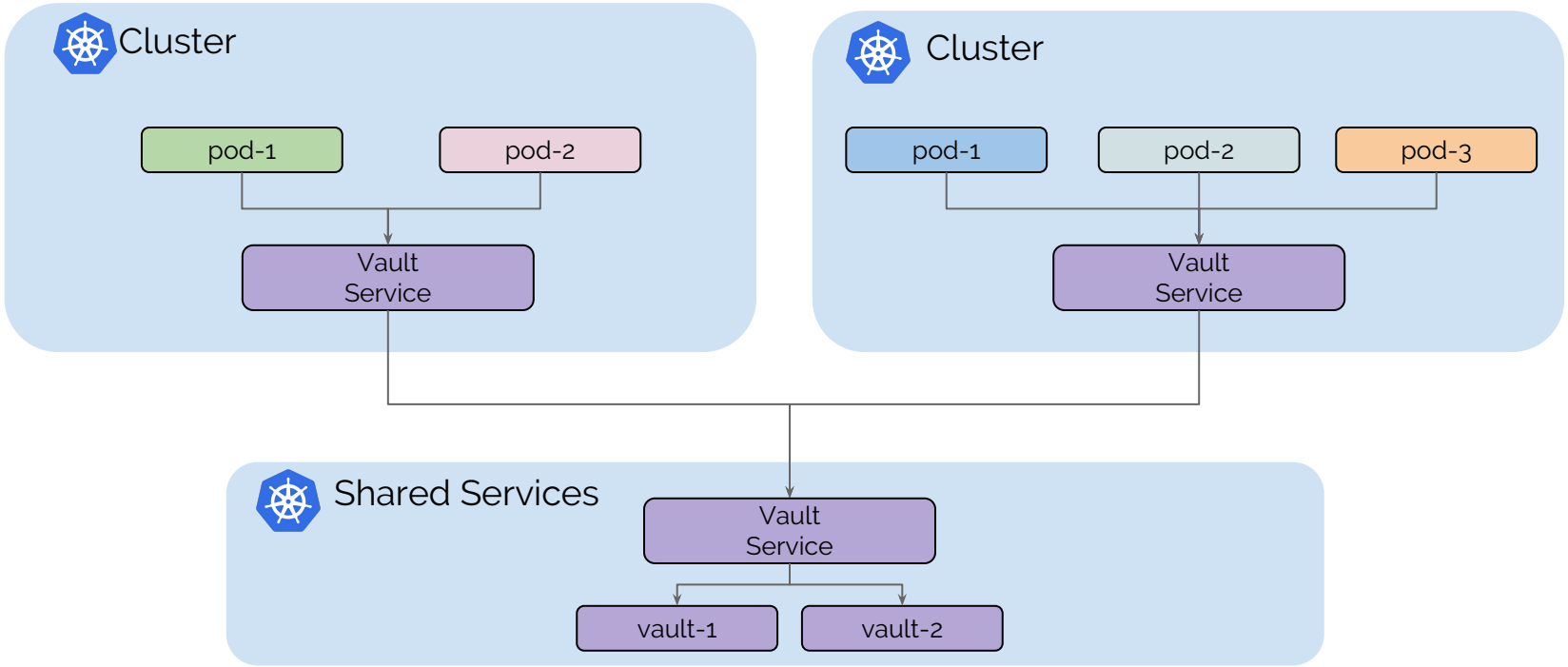
High Availability

Fail over to another cluster



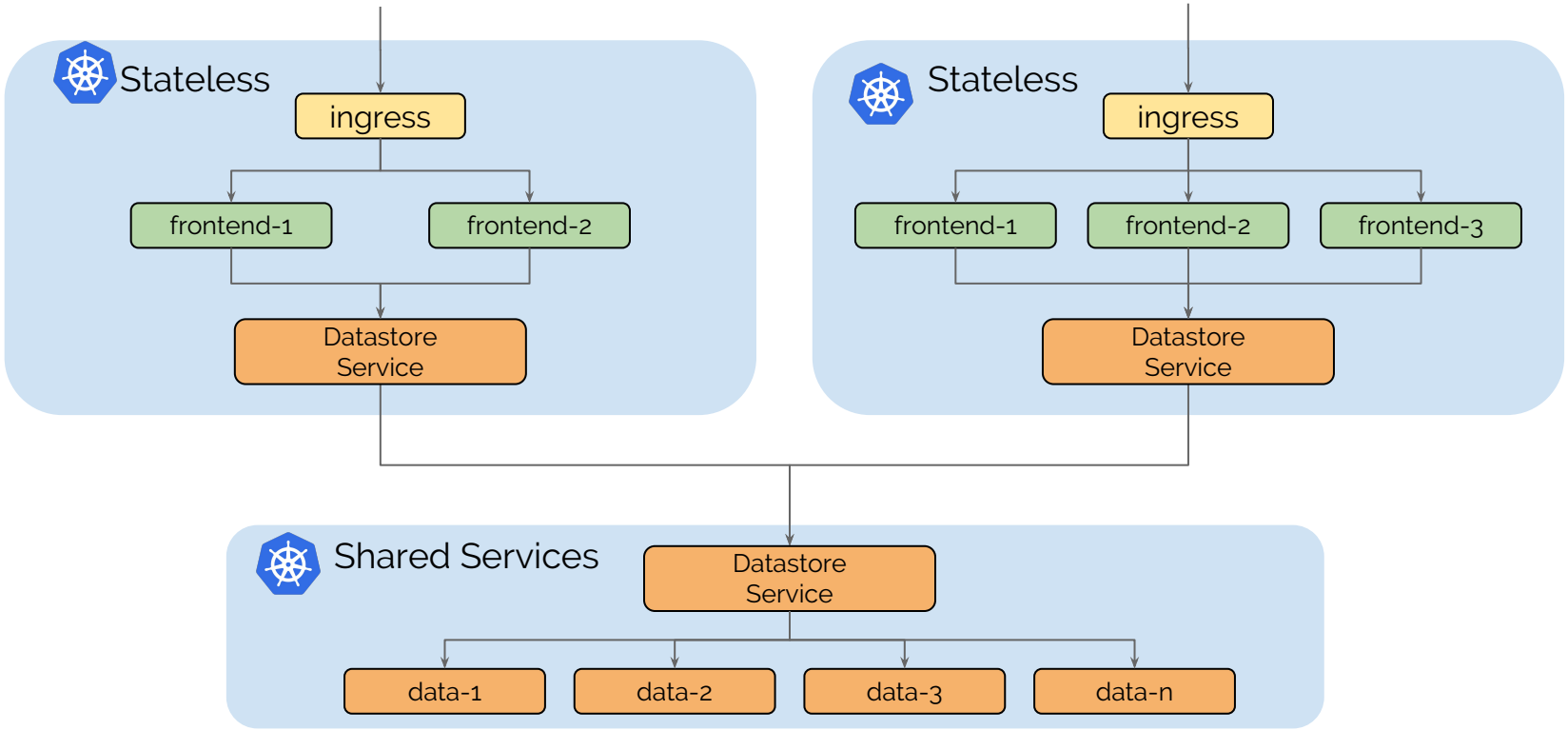
Shared Services

Not all services need to be run in every cluster



Split Stateless and Stateful

Keep your clusters dependency free



CVE-2018-1002105

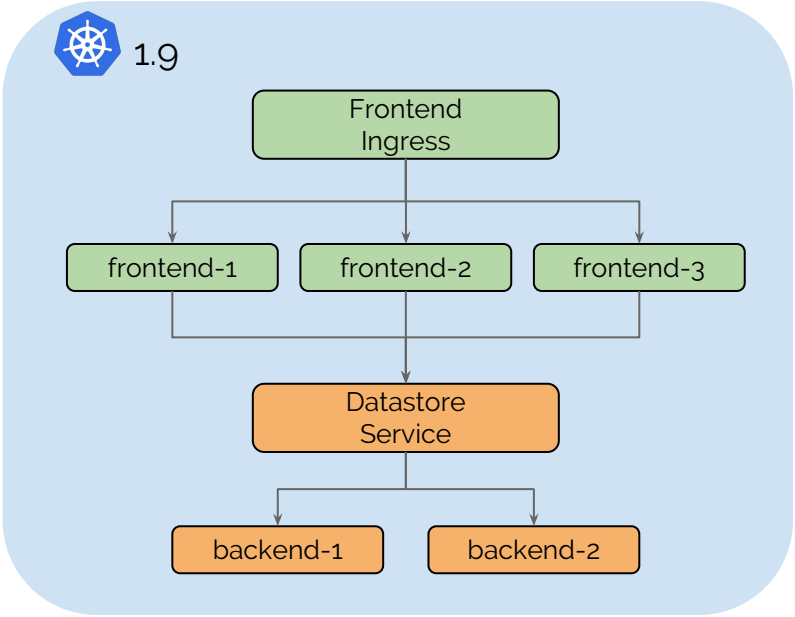
CVE-2018-1002105: proxy request handling in kube-apiserver can leave vulnerable TCP connections

Affected versions:

- Kubernetes v1.0.x-1.9.x
- Kubernetes v1.10.0-1.10.10 (fixed in v1.10.11)
- Kubernetes v1.11.0-1.11.4 (fixed in v1.11.5)
- Kubernetes v1.12.0-1.12.2 (fixed in v1.12.3)

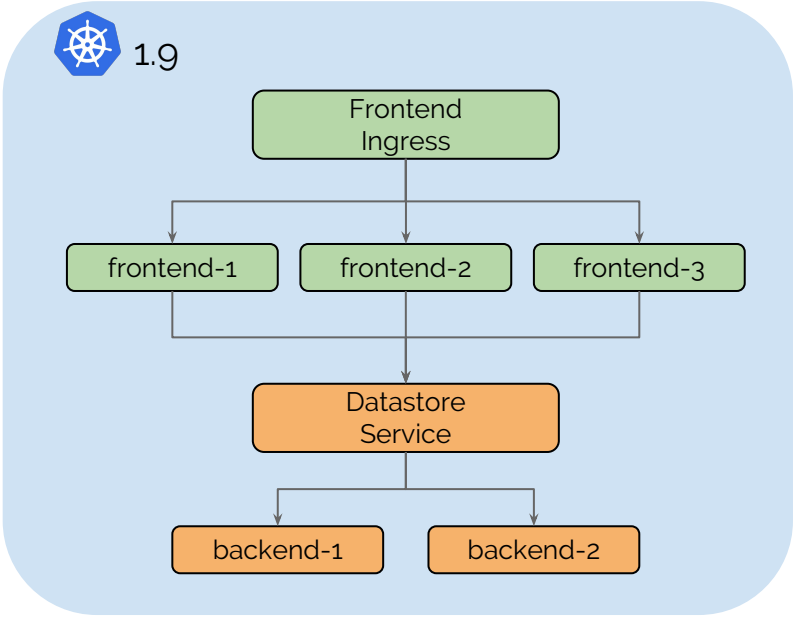
Evacuation

Move the stateless pieces



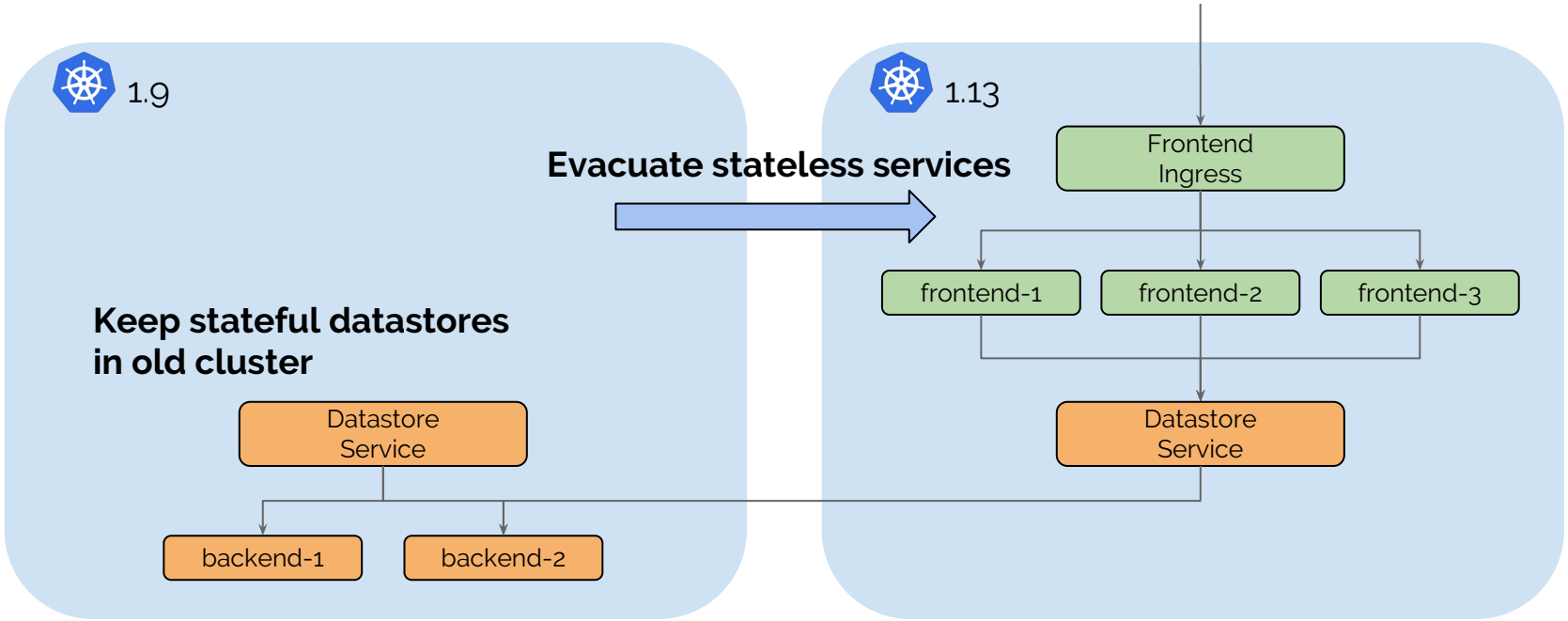
Evacuation

Move the stateless pieces



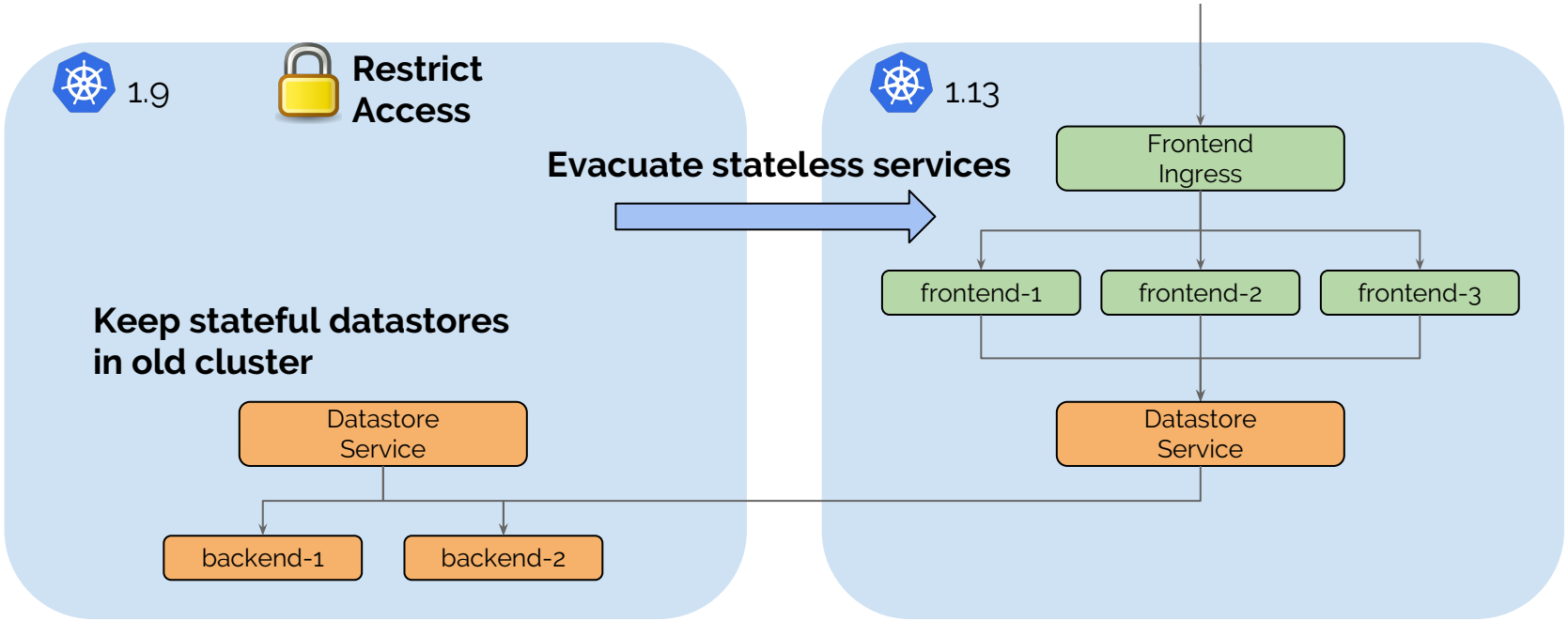
Evacuation

Move the stateless pieces



Evacuation

Move the stateless pieces



Istio Integration

Control Plane

- Service Management
- Identity Provider Integration
- Telemetry Collection
- ...

Knows what to do

Dataplane

- Handling network packets and L7 requests
- Routing & load balancing
- Security policy enforcement
- ...

Knows how to do it

- Both Istio and Kubernetes currently provide service definitions
- We chose to implement Kubernetes services first for simplicity
- Istio services can be supported with the same simplicity and performance

Summary

- **Based on new BPF technology**
- **Networking (CNI)**
- **Kubernetes services**
  - Replacing kube-proxy
  - Multi-cluster capability (1.4)
- **Network security**
  - Identity-based, DNS aware, API aware, data protocol aware
  - Transparent encryption (1.4)
- **Envoy/Istio Integration**
  - Sidecar Acceleration
  - Transparent SSL visibility (kTLS)

Thank You!

More Information:

Slack: https://cilium.io/slack

GitHub: https://github.com/cilium/cilium

Docs: https://docs.cilium.io/

Twitter: @ciliumproject

Want to hear more about Cilium at KubeCon?

Implementing Least Privilege Security and Networking with BPF on Kubernetes

1:45pm - 2:20pm, Ballroom 6C


KubeCon
CloudNativeCon
North America 2018