

# Scaling Network Policy Enforcement Beyond the Cluster Boundary



CiliumCon  
NORTH AMERICA



DATADOG

# Scaling Network Policy Enforcement Beyond the Cluster Boundary



## Hemanth Malla

@hemanthmalla



## Maxime Visonneau

@mvisonneau





100,000,000,000,000  
events/day



# Agenda

---

**01** Background and Concepts

---

**02** State of Cross Cluster Policies

---

**03** Evolution of Meshing

---

**04** KV KV Mesh?

---

**05** Migration, Monitoring and Best Practices

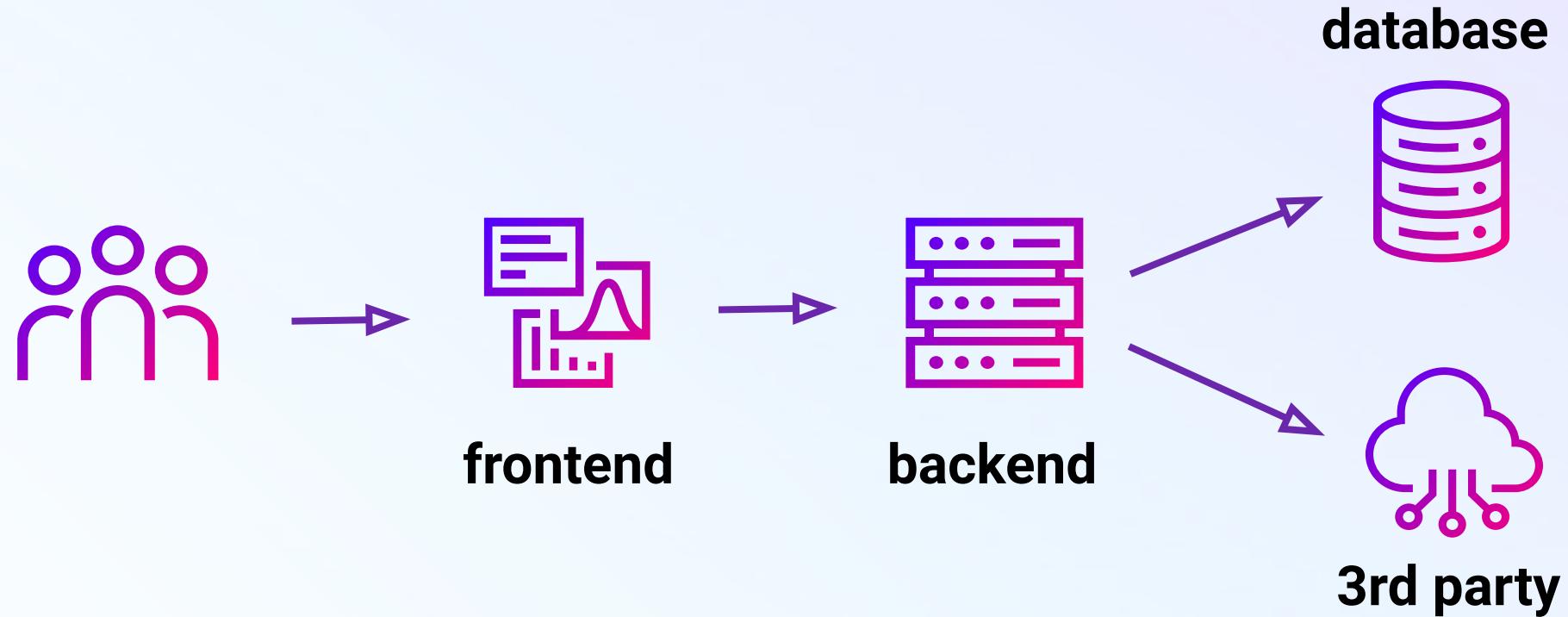
---

**06** Future Work

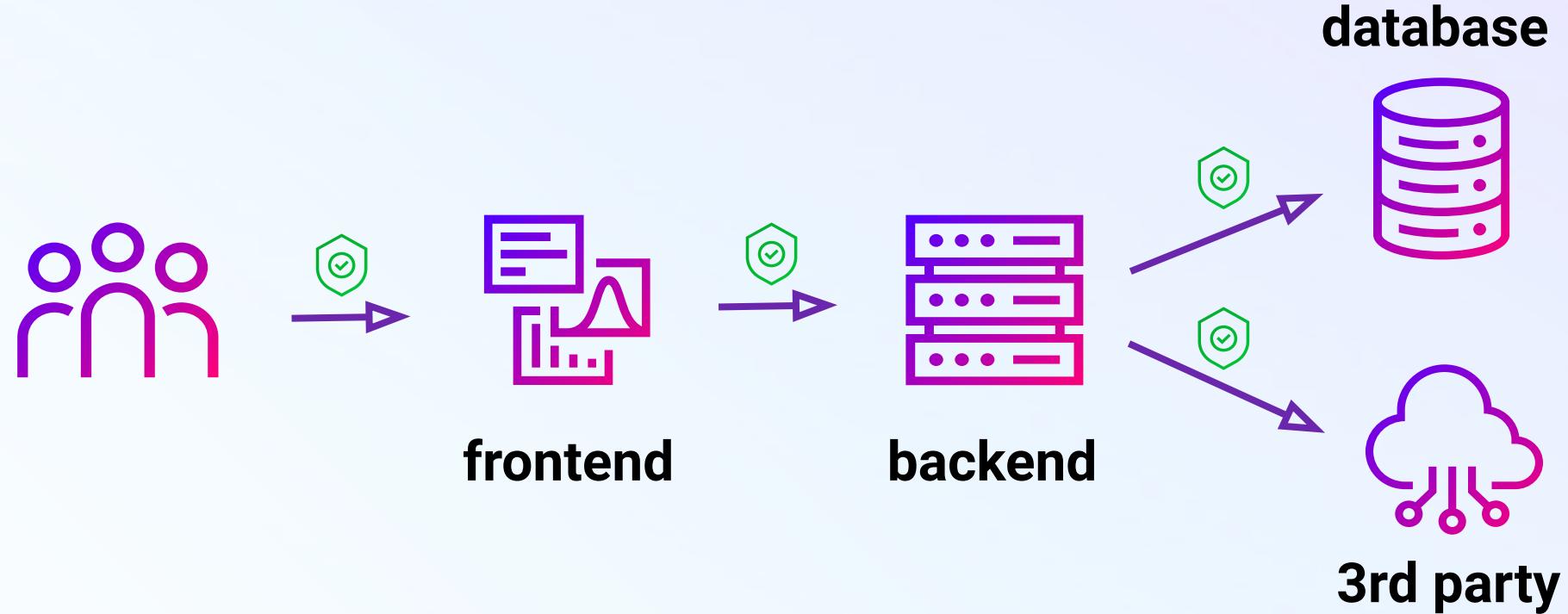
---

# 01 Background and Concepts

# A basic example



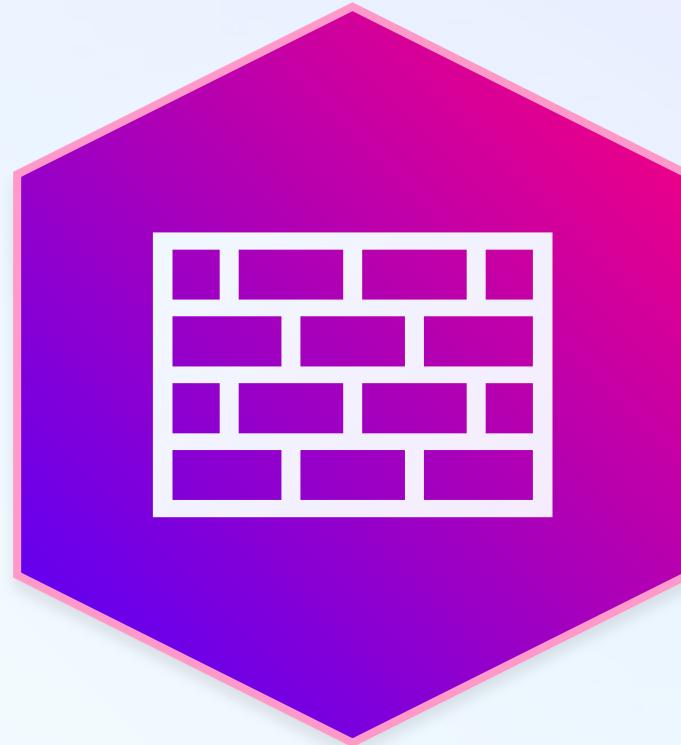
# Scoping down accesses to what is necessary



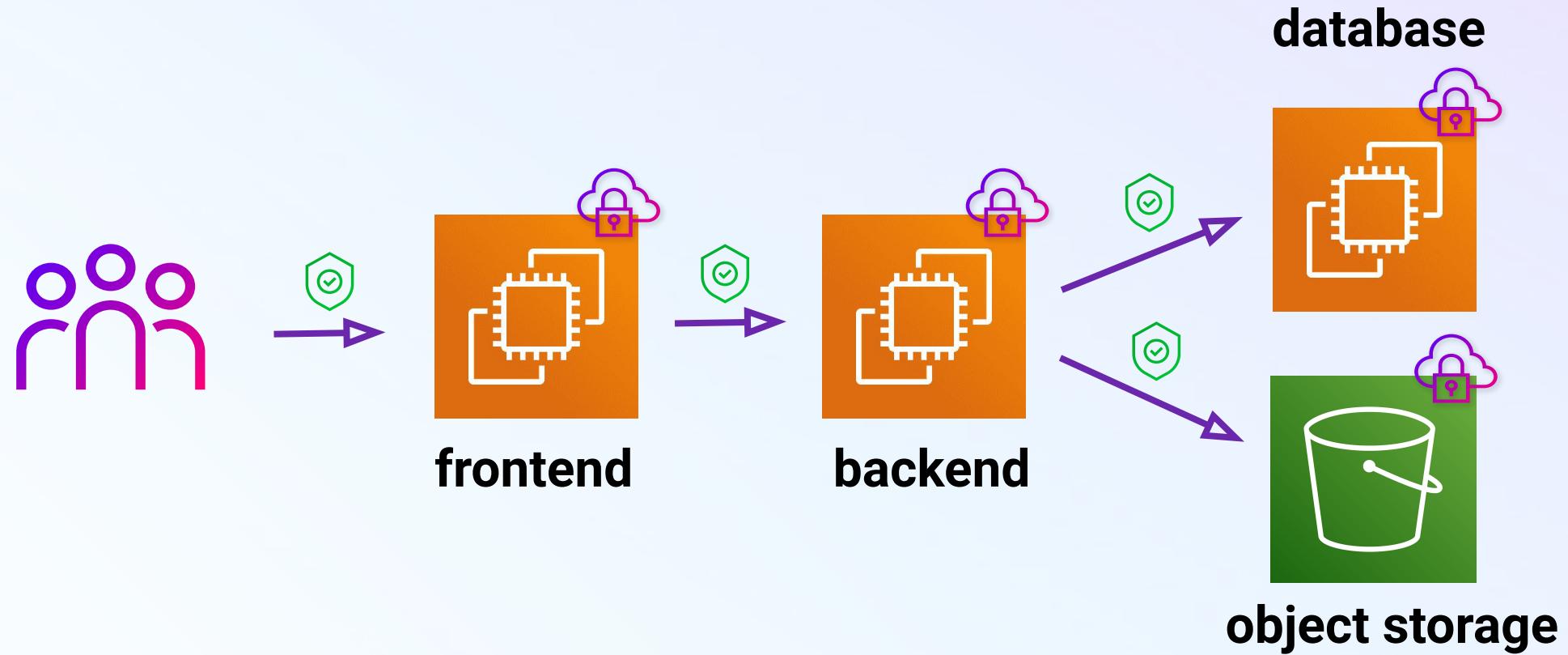




# Security primitives

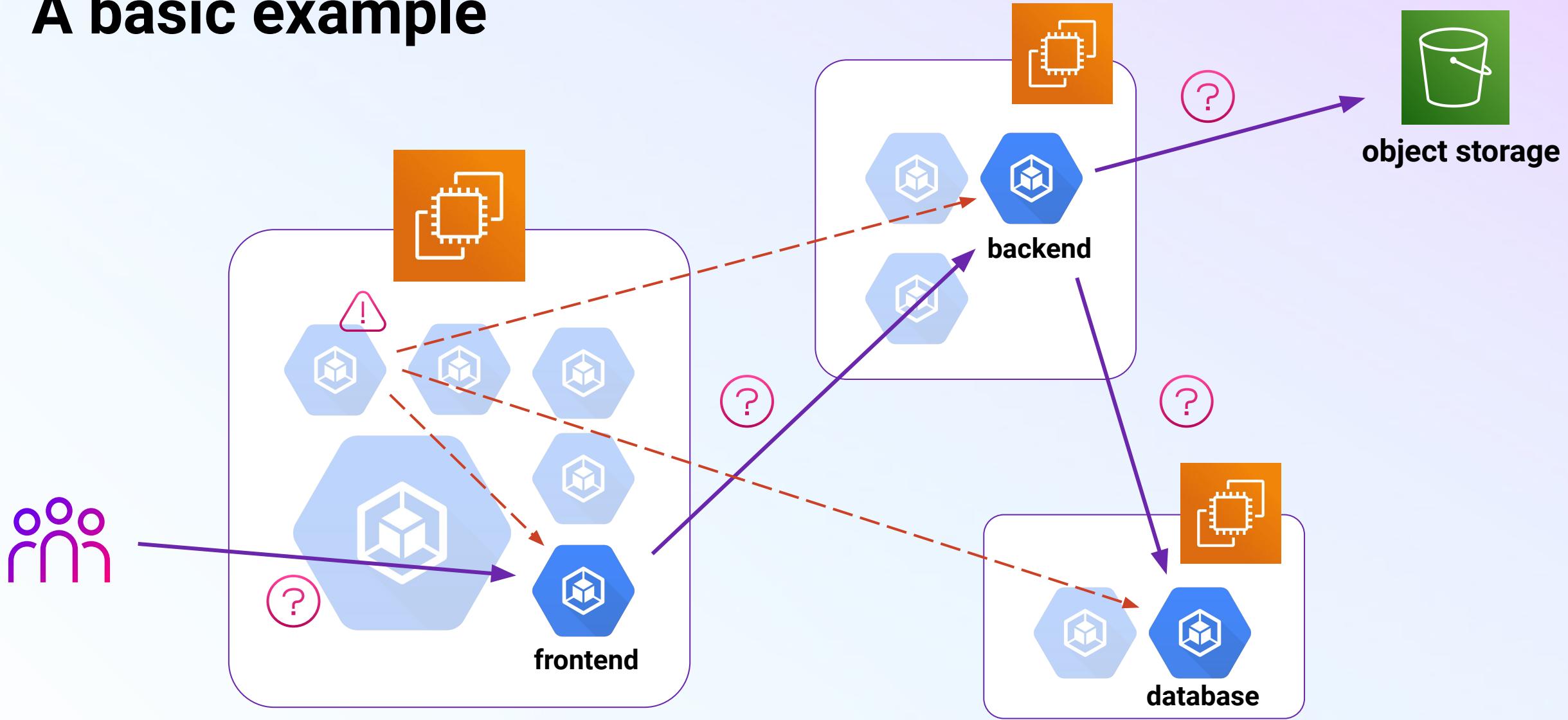


# A basic example





# A basic example



View Amazon EKS security group requirements for clusters

This topic describes the security group requirements of an Amazon EKS cluster.

## Default cluster security group

When you create a cluster, Amazon EKS creates a security group that's named `eks-cluster-sg-my-cluster-uniqueID`. This security group has the following default rules:

Rule type	Protocol	Ports	Source	Destination
Inbound	All	All	Self	
Outbound	All	All		0.0.0.0/0(IPv4) or ::/0(IPv6)

**Important**  
If your cluster doesn't need the outbound rule, you can remove it. If you remove it, you must still have the minimum rules listed in [Restricting cluster traffic](#). If you remove the inbound rule, Amazon EKS recreates it whenever the cluster is updated.

Amazon EKS adds the following tags to the security group. If you remove the tags, Amazon EKS adds them back to the security group whenever your cluster is updated.

Key	Value
<code>kubernetes.io/cluster/<i>my-cluster</i></code>	<code>owned</code>
<code>aws:eks:cluster-name</code>	<code>my-cluster</code>
Name	<code>eks-cluster-sg-<i>my-cluster-uniqueid</i></code>

Amazon EKS automatically associates this security group to the following resources that it also creates:

- 2–4 elastic network interfaces (referred to for the rest of this document as *network interface*) that are created when you create your cluster.
- Network interfaces of the nodes in any managed node group that you create.

The default rules allow all traffic to flow freely between your cluster and nodes, and allows all outbound traffic to any destination.

On this page

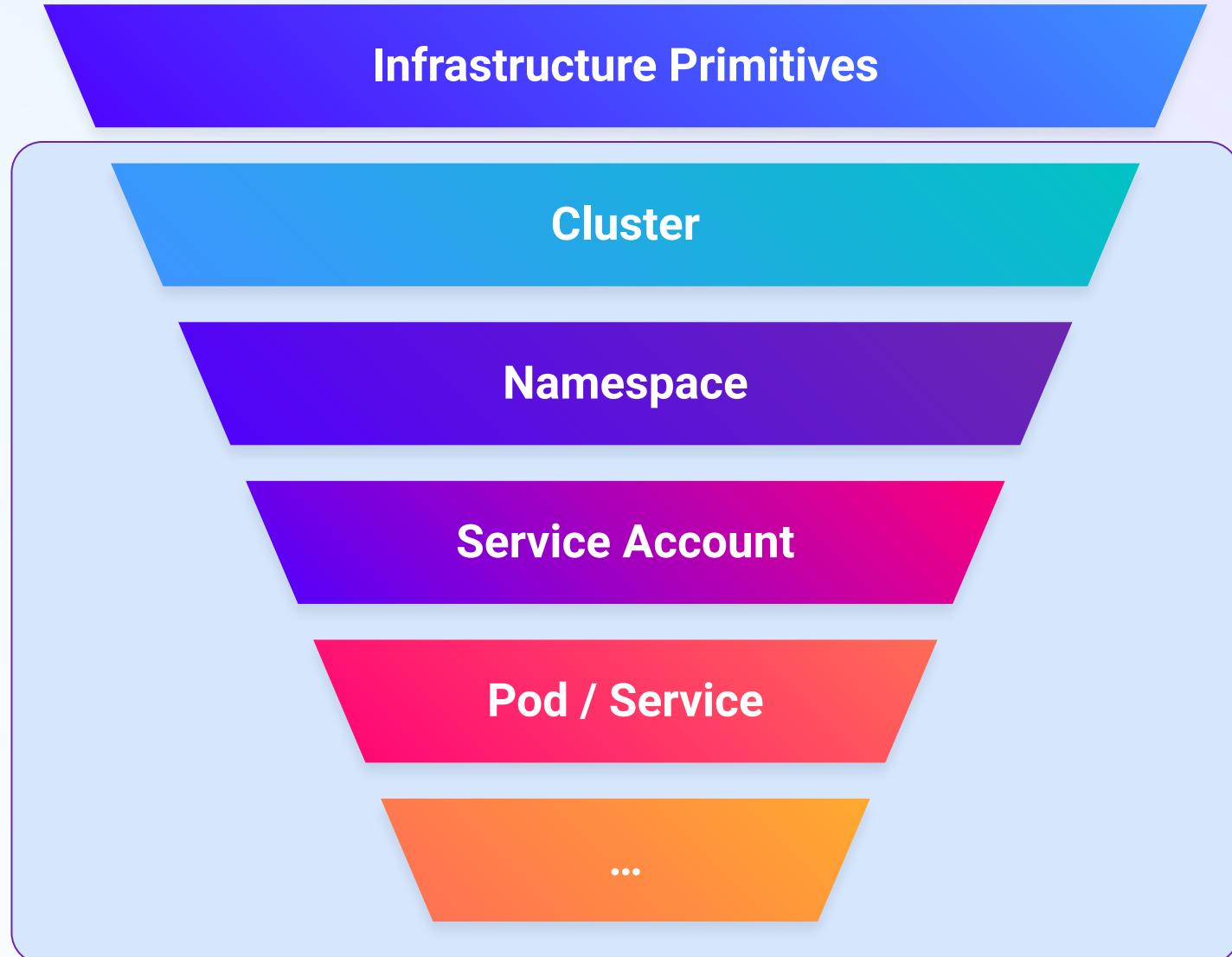
Default cluster security group  
Restricting cluster traffic

Feedback  Preferences 



# Network Policies

# Finer-grained control



# Network Policy

Applies to **backend** pods

Allow **TCP/443 from frontend** pods

Allow **TCP/5432 to database** pods

Allow **TCP/443 to the object storage endpoint**



```
● ● ●  
---  
apiVersion: cilium.io/v2  
kind: CiliumNetworkPolicy  
metadata:  
  name: backend-policy  
  namespace: backends  
spec:  
  endpointSelector:  
    matchLabels:  
      role: backend  
  
  ingress:  
    - fromEndpoints:  
        - matchLabels:  
            role: frontend  
            io.kubernetes.pod.namespace: frontends  
        toPorts:  
          - ports:  
              - port: "443"  
                protocol: TCP  
  
  egress:  
    - toEndpoints:  
        - matchLabels:  
            role: database  
            io.kubernetes.pod.namespace: databases  
        toPorts:  
          - ports:  
              - port: "5432"  
                protocol: TCP  
  
    - toFQDNs:  
        - matchName: some-bucket.s3.us-east-1.amazonaws.com  
    toPorts:  
      - ports:  
          - port: "443"  
            protocol: TCP
```



Any other network traffic not specified is **denied**

# Ways to restrict traffic to/from all pods by default



```
apiVersion: cilium.io/v2
kind: CiliumClusterwideNetworkPolicy
metadata:
  name: default-deny-all
spec:
  endpointSelector: {}
  egress: {}
  ingress: {}
```

or

The configuration of the Cilium agent and the Cilium Network Policy determines whether an endpoint accepts traffic from a source or not. The agent can be put into the following three policy enforcement modes:

#### default

This is the default behavior for policy enforcement. In this mode, endpoints have unrestricted network access until selected by policy. Upon being selected by a policy, the endpoint permits only allowed traffic. This state is per-direction and can be adjusted on a per-policy basis. For more details, [see the dedicated section on default mode](#).

#### always

With always mode, policy enforcement is enabled on all endpoints even if no rules select specific endpoints.

If you want to configure health entity to check cluster-wide connectivity when you start `cilium-agent` with `enable-policy: always`, you will likely want to enable communications to and from the health endpoint. See [Example: Add Health Endpoint](#).

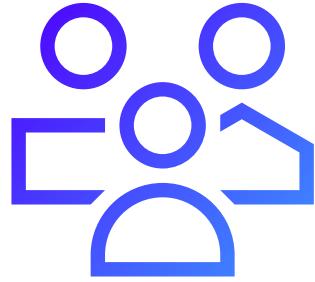
#### never

With never mode, policy enforcement is disabled on all endpoints, even if rules do select specific endpoints. In other words, all traffic is allowed from any source (on ingress) or destination (on egress).

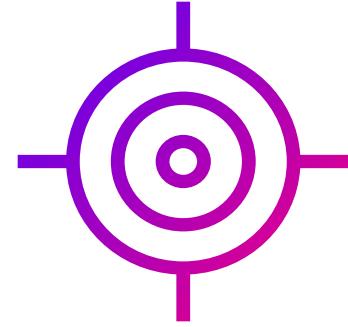


[docs.cilium.io/en/latest/security/policy/intro](https://docs.cilium.io/en/latest/security/policy/intro)

# Two important concepts



**Identities**



**Endpoints**

# Identity

A **unique identifier** within the cluster

A **set of security labels** derived from the resource we are looking to represent



```
---  
apiVersion: cilium.io/v2  
kind: CiliumIdentity  
metadata:  
  labels: [...]  
  name: "12345"  
  security-labels:  
    k8s:io.cilium.k8s.policy.cluster: cluster-a  
    k8s:io.cilium.k8s.policy.serviceaccount: backend  
    k8s:io.kubernetes.pod.namespace: backends  
    k8s:tags.datadoghq.com/env: staging  
    k8s:tags.datadoghq.com/service: backend  
    k8s:tags.datadoghq.com/version: 1.2.3
```

# Endpoint

Some references to which pod leverages it

A **unique identifier** within the cluster

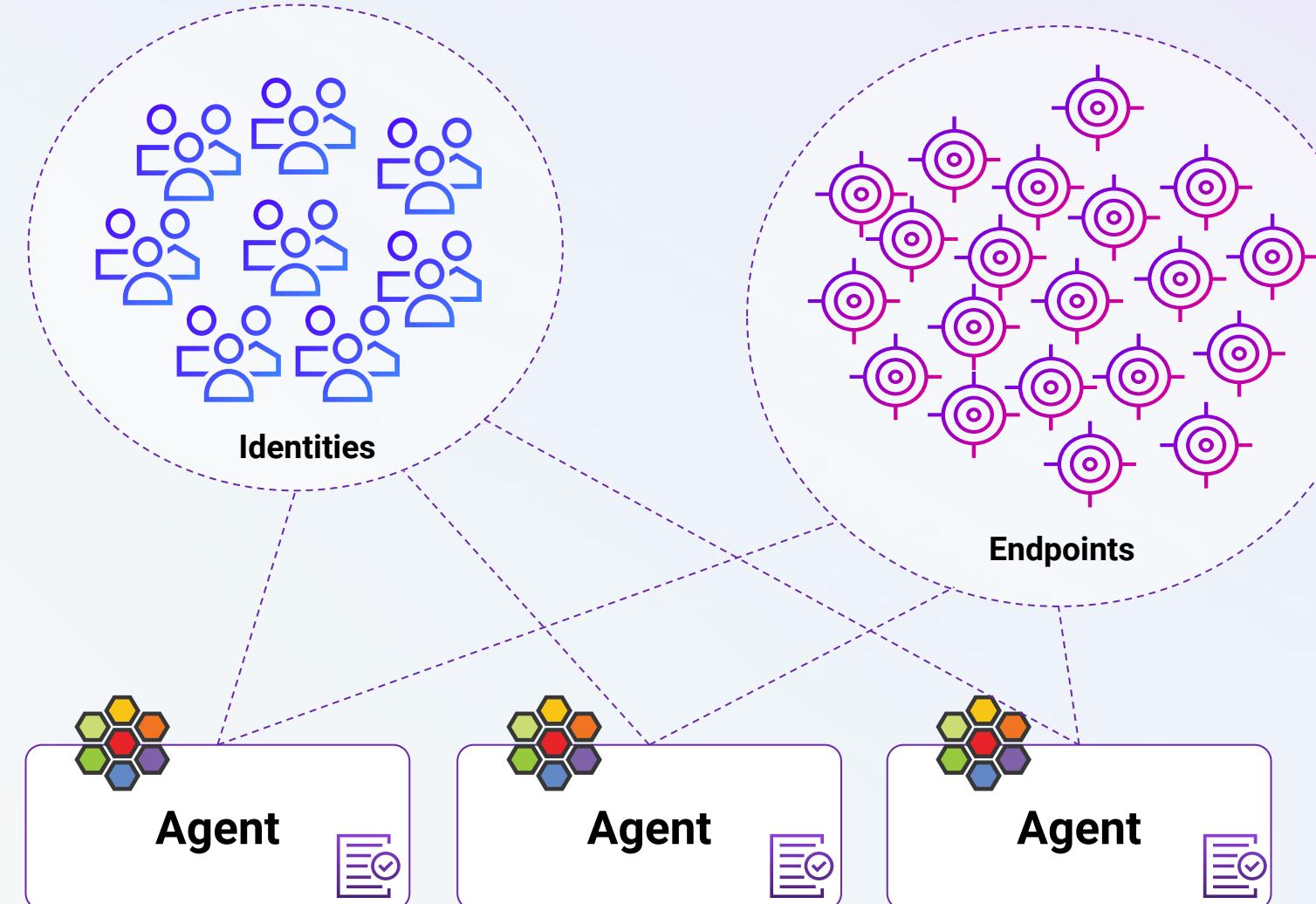
The **identity** it is associated with

**Allocated IP addresses** and its current state

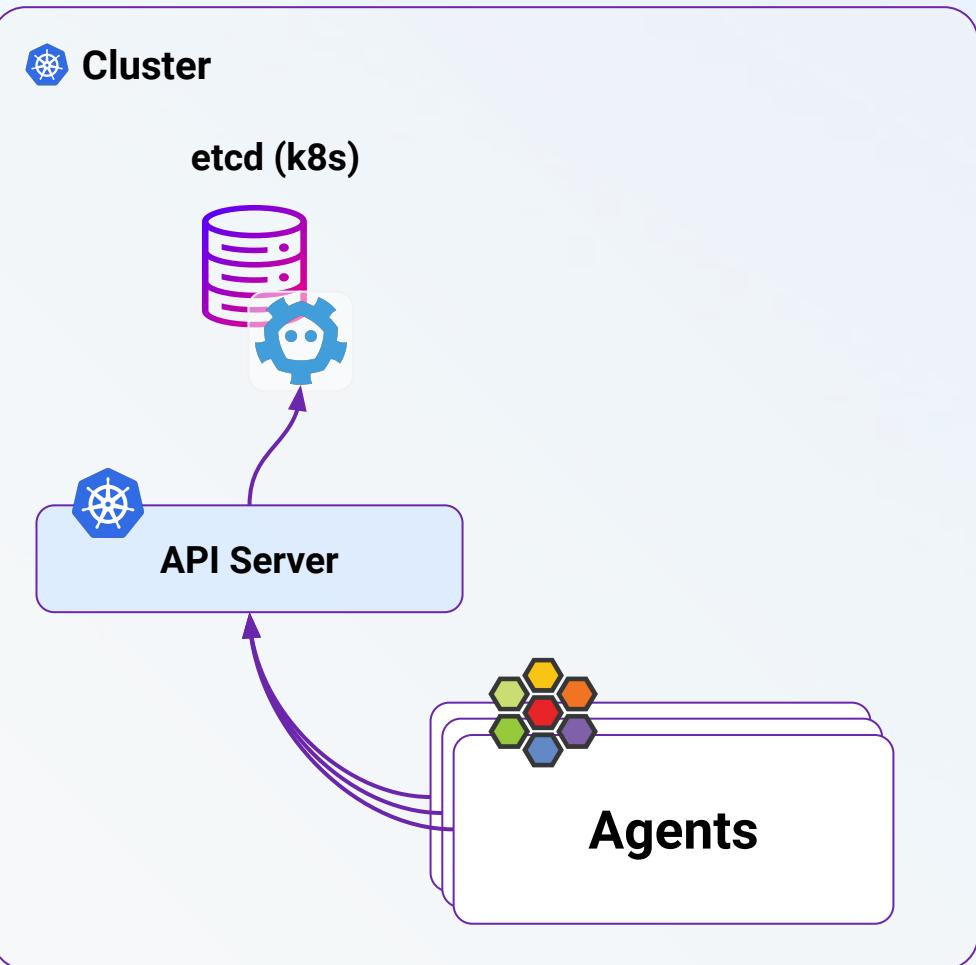


```
---  
apiVersion: cilium.io/v2  
kind: CiliumEndpoint  
metadata:  
  labels: [...]  
  name: backend-abcd-1234  
  namespace: backends  
  ownerReferences:  
    - apiVersion: v1  
      kind: Pod  
      name: backend-abcd-1234  
status:  
  encryption: {}  
  external-identifiers:  
    cni-attachment-id: 55d8d2348907dd2449b57de09cb07908d2:eth0  
    container-id: 55d8d2348907dd2449b57de09cb07908d2  
    k8s-namespace: backends  
    k8s-pod-name: backend-abcd-1234  
    pod-name: backends/backend-abcd-1234  
  id: 101  
  identity:  
    id: 12345  
  labels:  
    - k8s:io.cilium.k8s.policy.cluster=cluster-a  
    - k8s:io.cilium.k8s.policy.serviceaccount=backend  
    - k8s:io.kubernetes.pod.namespace=backends  
    - k8s:tags.datadoghq.com/env=staging  
    - k8s:tags.datadoghq.com/service=backend  
    - k8s:tags.datadoghq.com/version=1.2.3  
networking:  
  addressing:  
    - ipv4: 172.16.4.147  
  node: 172.16.6.134  
state: ready
```

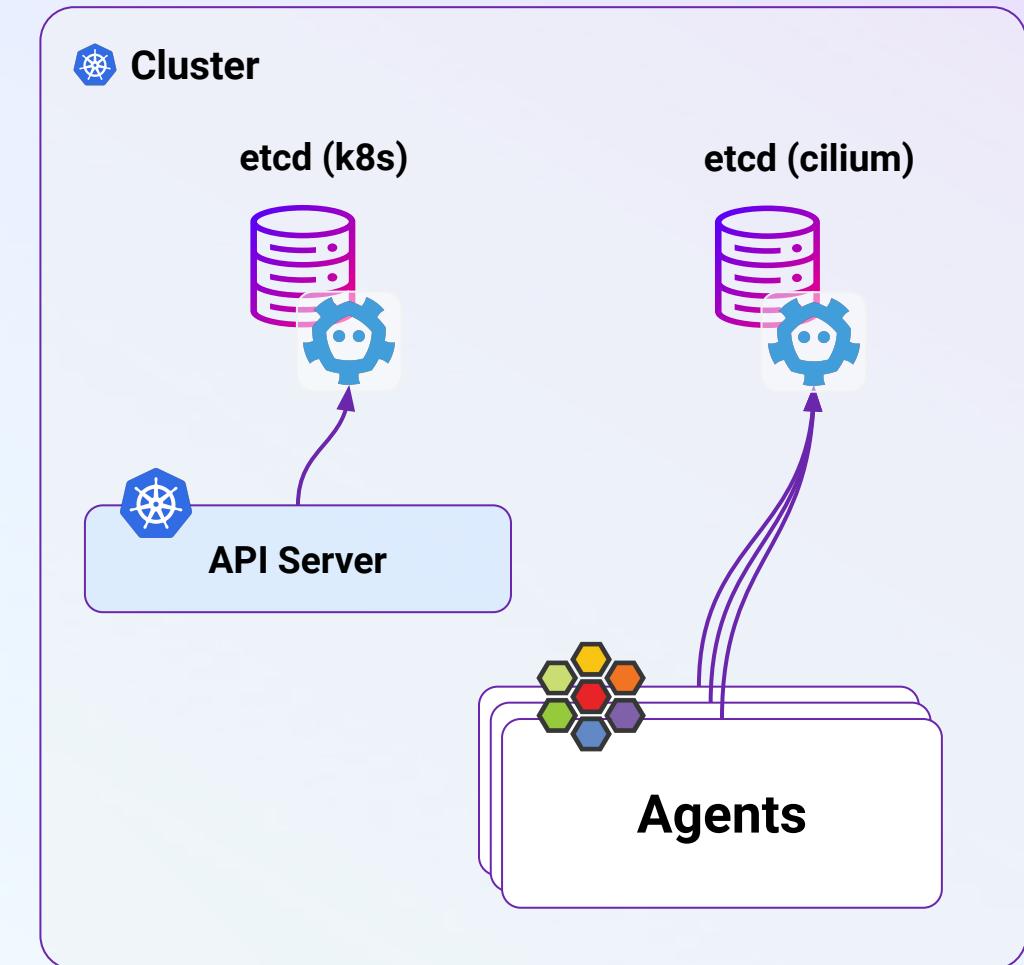
 Cluster



# Where is it all stored?



CRD Mode (default)



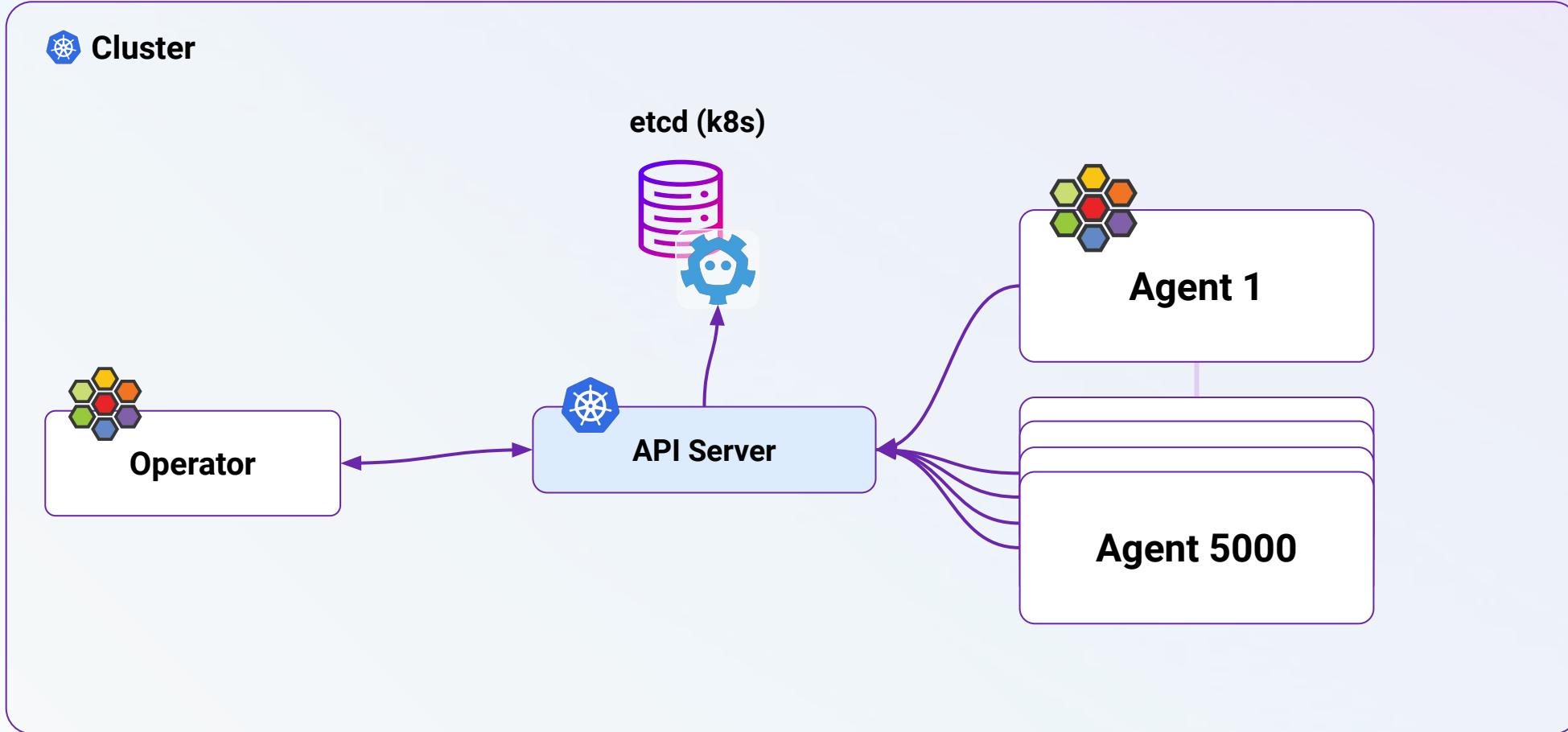
External Key-Value Store

# Scala



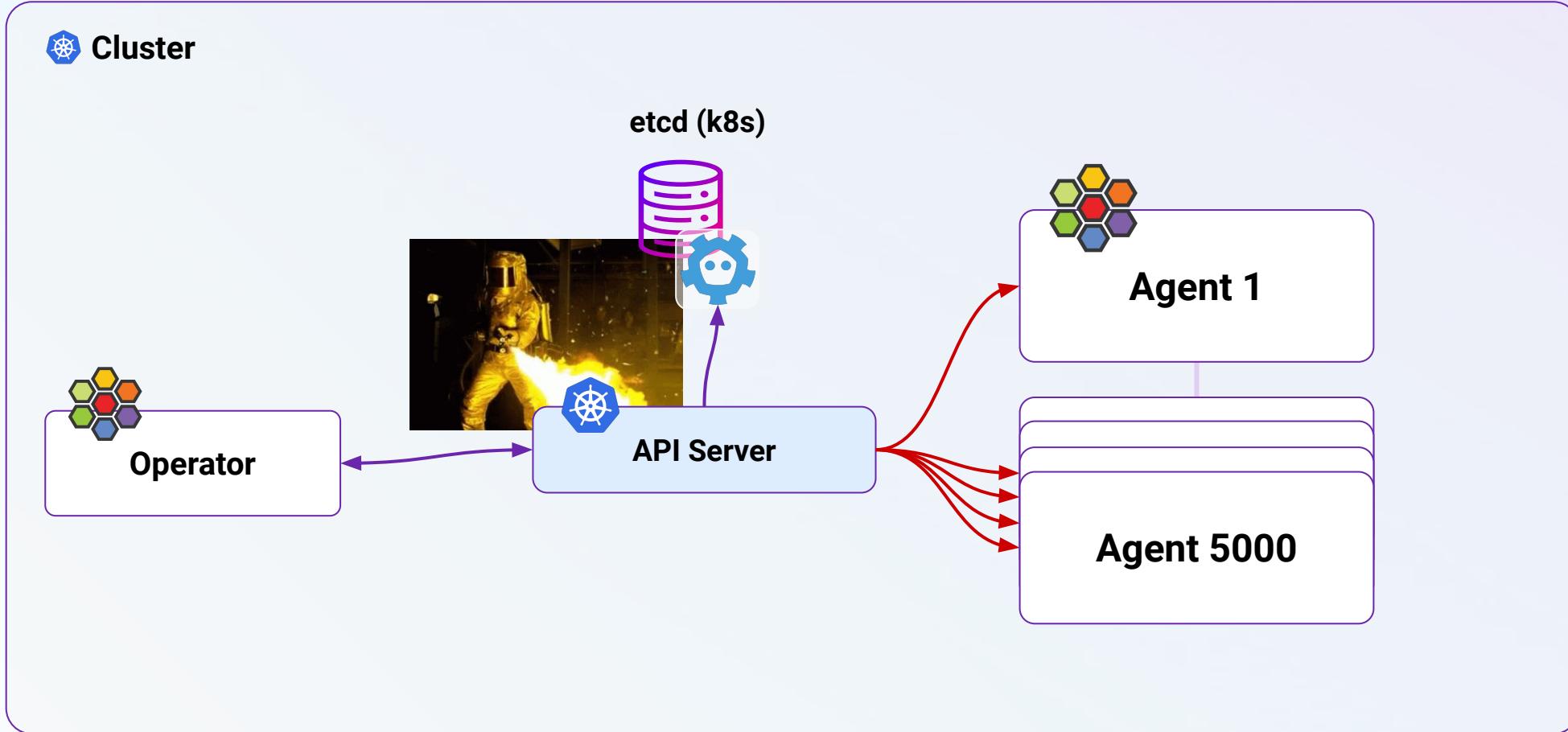
# lity

# Scalability



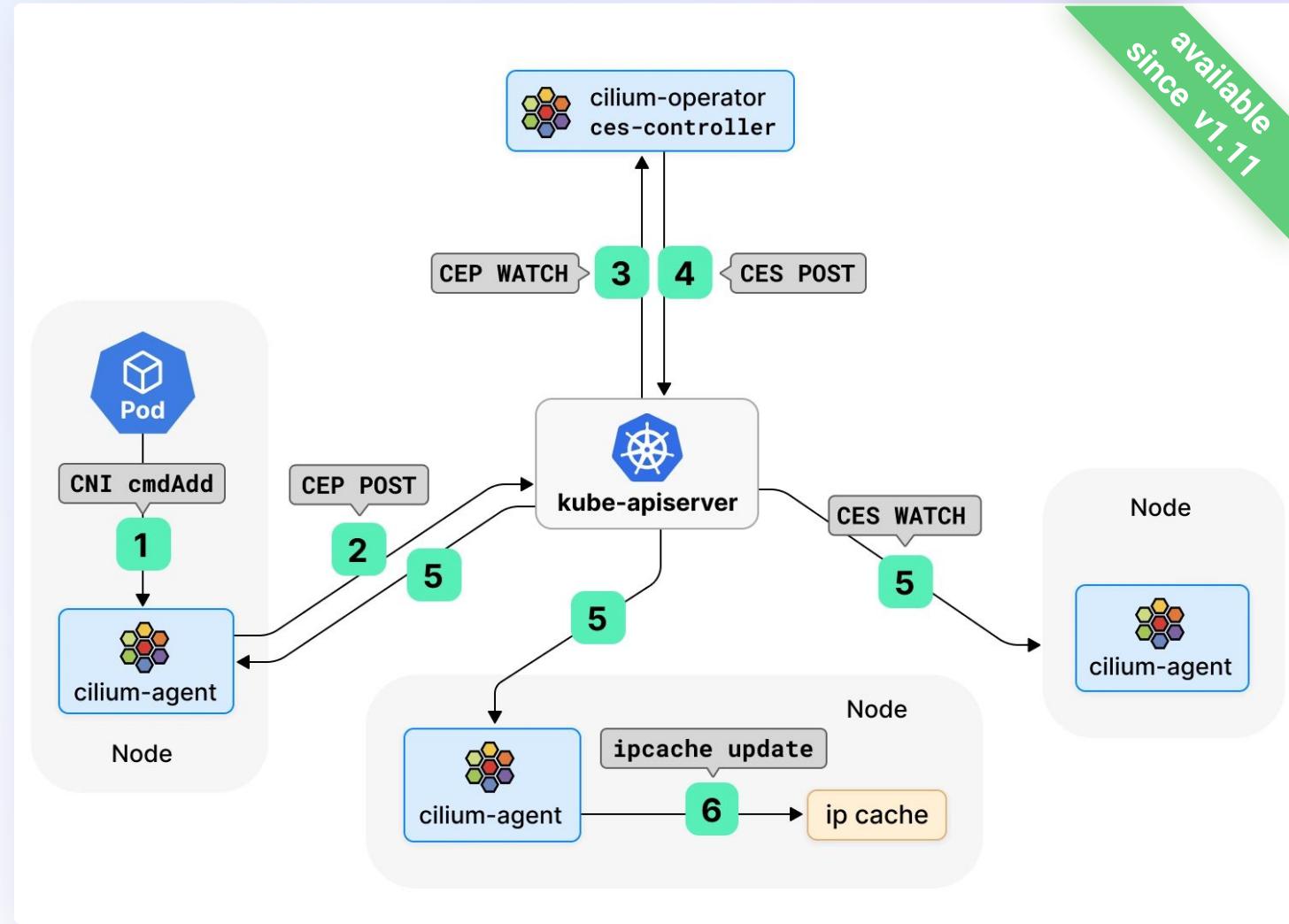
```
~$ kubectl scale deploy/backend \
--current-replicas 0 \
--replicas 100
```

# Scalability



**100 endpoints \* 5000 nodes = 500,000 watch updates**

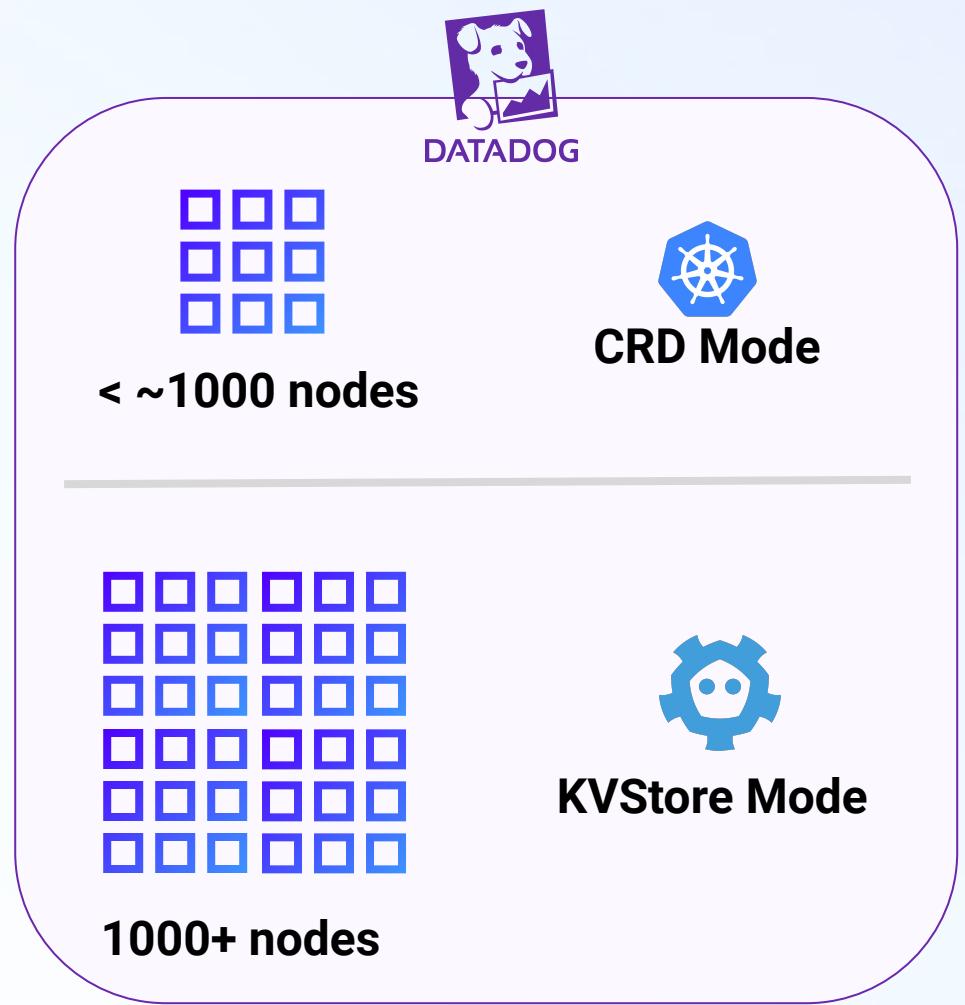
# Batching to the rescue



[isovalent.com/blog/post/2021-12-release-111#cilium-endpoint-slices](https://isovalent.com/blog/post/2021-12-release-111#cilium-endpoint-slices)



# At Datadog, for now we use both



@antonipp



--identity-allocation-mode=double-writeread{kvstore,crd}

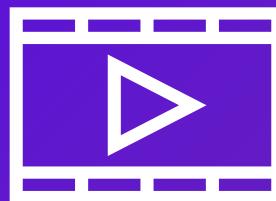


[github.com/cilium/cilium/pull/31920](https://github.com/cilium/cilium/pull/31920)

# Marcel Zięba

# Hemanth Malla

## KubeCon Europe 2024



## CRD Vs Dedicated etcd as Storage Backend : Lessons from Taming High Churn Clusters

*Hemanth Malla, Datadog & Marcel Zięba, Isovalent*



[youtube.com/watch?v=o\\_LutlRlv-A](https://youtube.com/watch?v=o_LutlRlv-A)



A screenshot of a web browser window displaying the Kubernetes Documentation. The title bar shows the page is titled "Considerations for large clusters". The address bar shows the URL is [kubernetes.io/docs/setup/best-practices/cluster-large/](https://kubernetes.io/docs/setup/best-practices/cluster-large/). The page header features the Kubernetes logo and navigation links for Documentation, Kubernetes Blog, Training, Partners, Community, Case Studies, Versions, and English. The main content area shows the breadcrumb navigation: Kubernetes Documentation / Getting started / Best practices / Considerations for large clusters. The main heading is "Considerations for large clusters". Below it, a text block explains that a cluster is a set of nodes (physical or virtual machines) running Kubernetes agents, managed by the control plane. It states that Kubernetes v1.31 supports clusters with up to 5,000 nodes and is designed to accommodate configurations that meet *all* of the following criteria. A purple-bordered box contains a bulleted list of these criteria: No more than 110 pods per node, No more than 5,000 nodes, No more than 150,000 total pods, and No more than 300,000 total containers. A final text block notes that you can scale your cluster by adding or removing nodes, depending on how your cluster is deployed.

Kubernetes Documentation / Getting started / Best practices / Considerations for large clusters

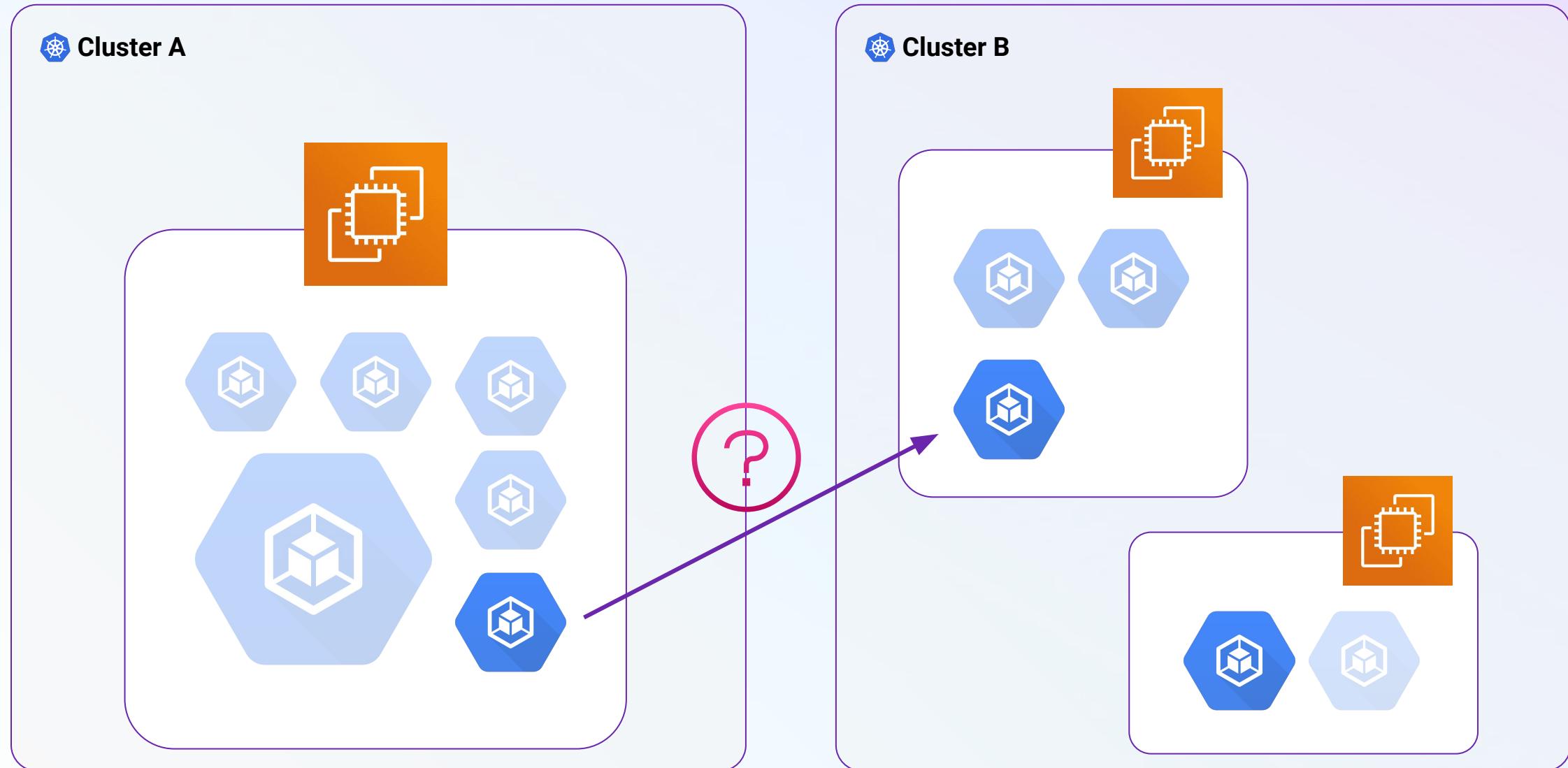
# Considerations for large clusters

A cluster is a set of nodes (physical or virtual machines) running Kubernetes agents, managed by the control plane. Kubernetes v1.31 supports clusters with up to 5,000 nodes. More specifically, Kubernetes is designed to accommodate configurations that meet *all* of the following criteria:

- No more than 110 pods per node
- No more than 5,000 nodes
- No more than 150,000 total pods
- No more than 300,000 total containers

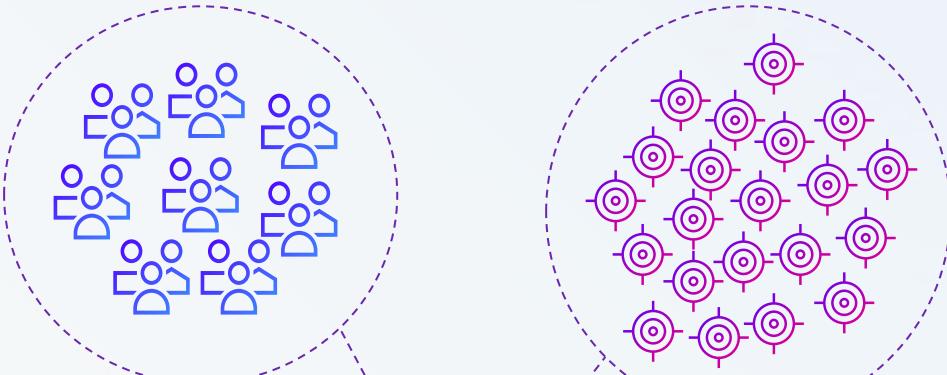
You can scale your cluster by adding or removing nodes. The way you do this depends on how your cluster is deployed.

# Moving abroad



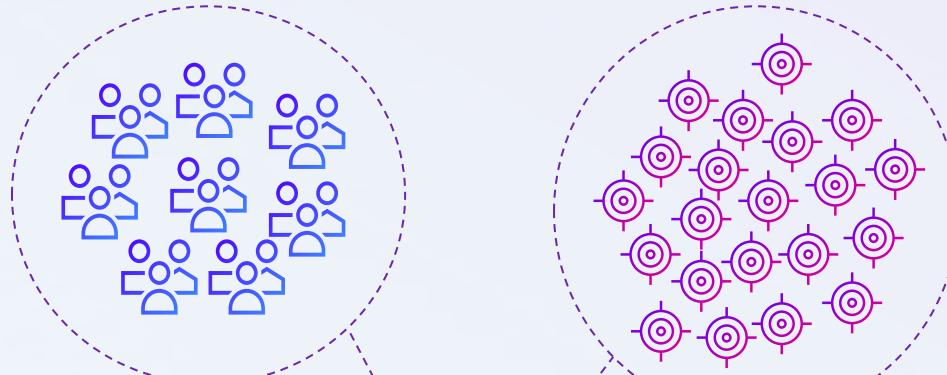
# Moving abroad

Cluster A



Agents

Cluster B



Agents

---

# 02 State of Cross Cluster Policies

# Back to leveraging lower-down primitives?

## Allow to external CIDR block

This example shows how to allow all endpoints with the label `app=myService` to talk to the external IP `20.1.1.1`, as well as the CIDR prefix `10.0.0.0/8`, but not CIDR prefix `10.96.0.0/12`

k8s YAML

JSON

Copy First Line  Copy All

```
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "cidr-rule"
spec:
  endpointSelector:
    matchLabels:
      app: myService
  egress:
  - toCIDR:
    - 20.1.1.1/32
  - toCIDRSet:
    - cidr: 10.0.0.0/8
    except:
    - 10.96.0.0/12
```



# Back to leveraging lower-down primitives?

**ToGroups** rules can be used to define policy in relation to cloud providers, like AWS.

```
---  
kind: CiliumNetworkPolicy  
apiVersion: cilium.io/v2  
metadata:  
  name: to-groups-sample  
  namespace: default  
spec:  
  endpointSelector:  
    matchLabels:  
      org: alliance  
      class: xwing  
  egress:  
  - toPorts:  
    - ports:  
      - port: '80'  
        protocol: TCP  
    toGroups:  
    - aws:  
      securityGroupsIds:  
      - 'sg-0f2146100a88d03c3'
```

Copy First Line  Copy All

This policy allows traffic from pod *xwing* to any AWS elastic network interface in the security group with ID `sg-0f2146100a88d03c3`.



# Or alternative ones?

## DNS based

The example below allows all DNS traffic on port 53 to the DNS service and intercepts it via the [DNS Proxy](#). If using a non-standard DNS port for a DNS application behind a Kubernetes service, the port must match the backend port. When the application makes a request for my-remote-service.com, Cilium learns the IP address and will allow traffic due to the match on the name under the `toFQDNs.matchName` rule.

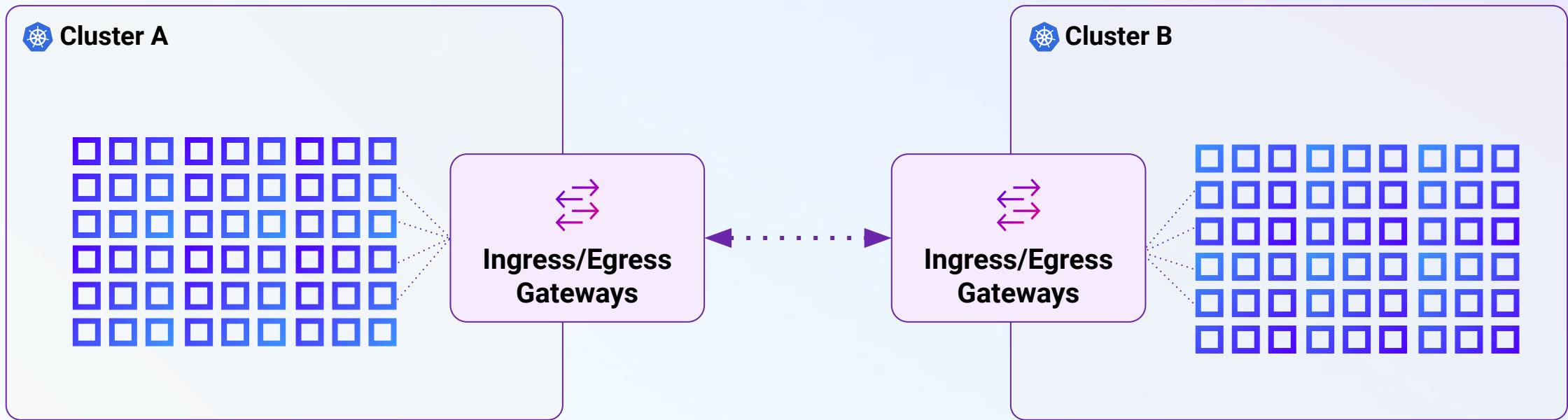
k8s YAML

JSON

Copy First Line  Copy All

```
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "to-fqdn"
spec:
  endpointSelector:
    matchLabels:
      app: test-app
  egress:
    - toEndpoints:
        - matchLabels:
            "k8s:io.kubernetes.pod.namespace": kube-system
            "k8s:k8s-app": kube-dns
        toPorts:
          - ports:
              - port: "53"
                protocol: ANY
            rules:
              dns:
                - matchPattern: "*"
    - toFQDNs:
        - matchName: "my-remote-service.com"
```

# Or alternative ones?

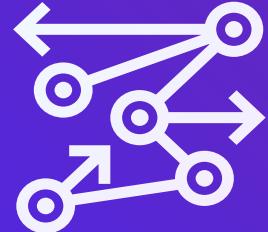


[docs.cilium.io/en/v1.16/network/servicemesh/ingress-to-gateway/ingress-to-gateway](https://docs.cilium.io/en/v1.16/network/servicemesh/ingress-to-gateway/ingress-to-gateway)  
[docs.cilium.io/en/v1.16/network/egress-gateway/egress-gateway](https://docs.cilium.io/en/v1.16/network/egress-gateway/egress-gateway)

# Common limitations



User Change



Loose Coupling

# Our ideal world

 Cluster A



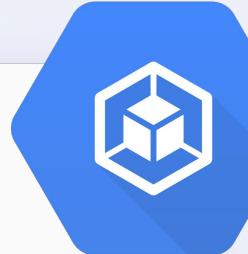
```
apiVersion: cilium.io/v2
kind: CiliumNetworkPolicy
metadata:
  name: backend-policy
  namespace: backends
spec:
  endpointSelector:
    matchLabels:
      role: backend

  ingress:
    - fromEndpoints:
        - matchLabels:
          role: frontend
          io.kubernetes.pod.namespace: frontends
        toPorts:
          - ports:
              - port: "443"
                protocol: TCP

  egress:
    - toEndpoints:
        - matchLabels:
          role: database
          io.kubernetes.pod.namespace: databases
        toPorts:
          - ports:
              - port: "5432"
                protocol: TCP

    - toFQDNs:
        - matchName: some-bucket.s3.us-east-1.amazonaws.com
      toPorts:
        - ports:
            - port: "443"
              protocol: TCP
```

 Cluster B



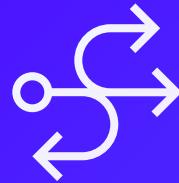


# Cluster Mesh

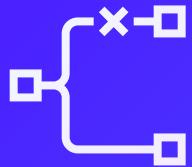
# Cluster Mesh



Seamless Pod IP Routing Across Clusters



Transparent Service Discovery



High Availability and Fault Tolerance



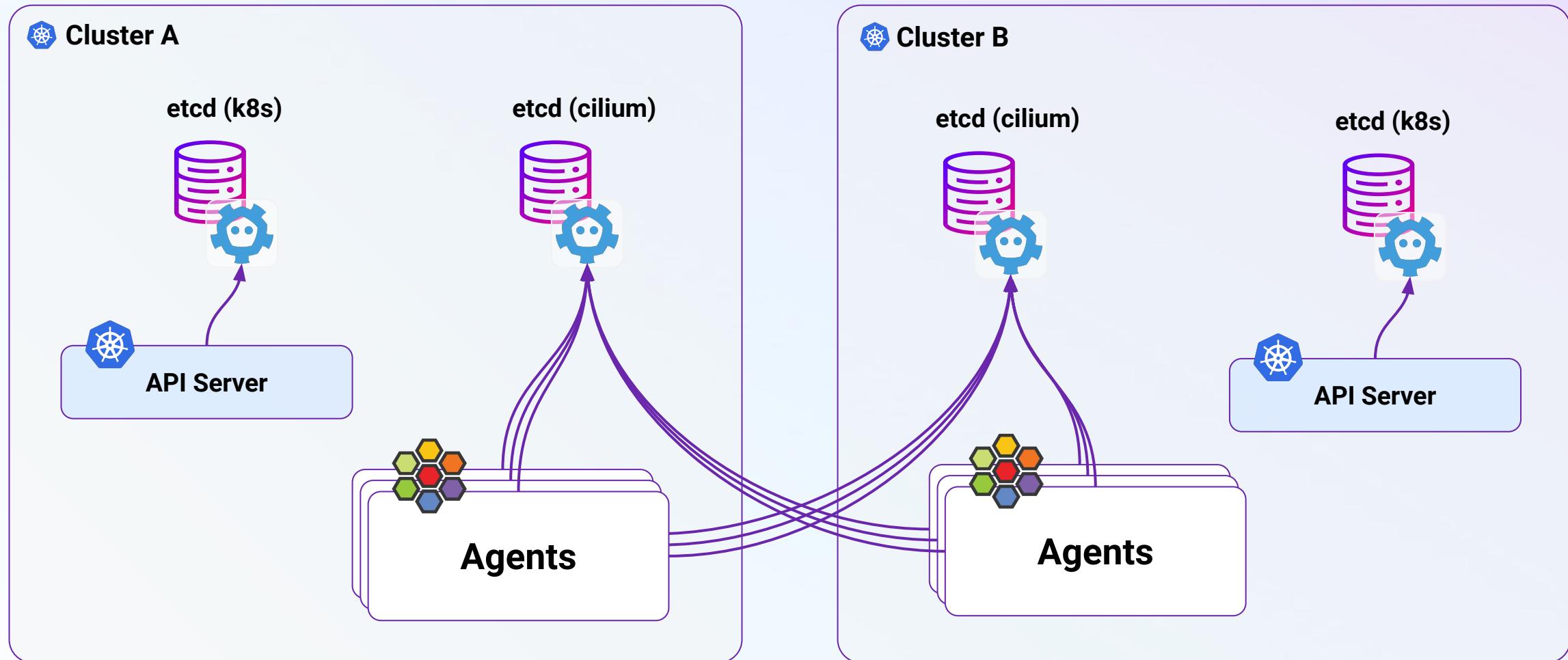
Uniform Network Policy Enforcement

# 03 Evolution of Meshing

# Evolution of Meshing



# ClusterMesh



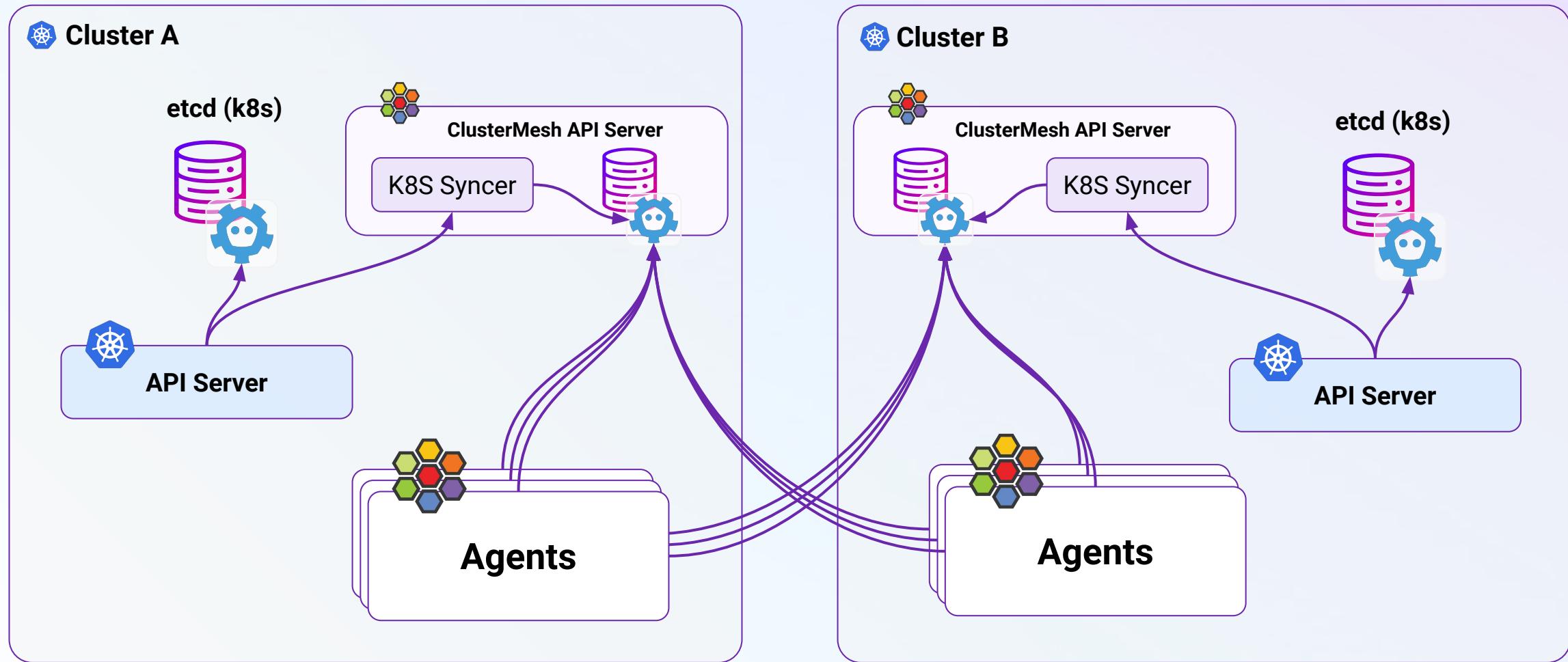
Requires using an External Key-Value Store implementation

# **m\*n updates**

**m = #clusters in mesh**

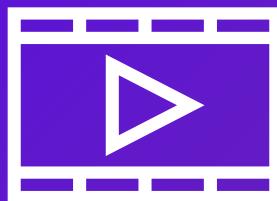
**n = #agents per cluster**

# ClusterMesh + ClusterMesh API Server



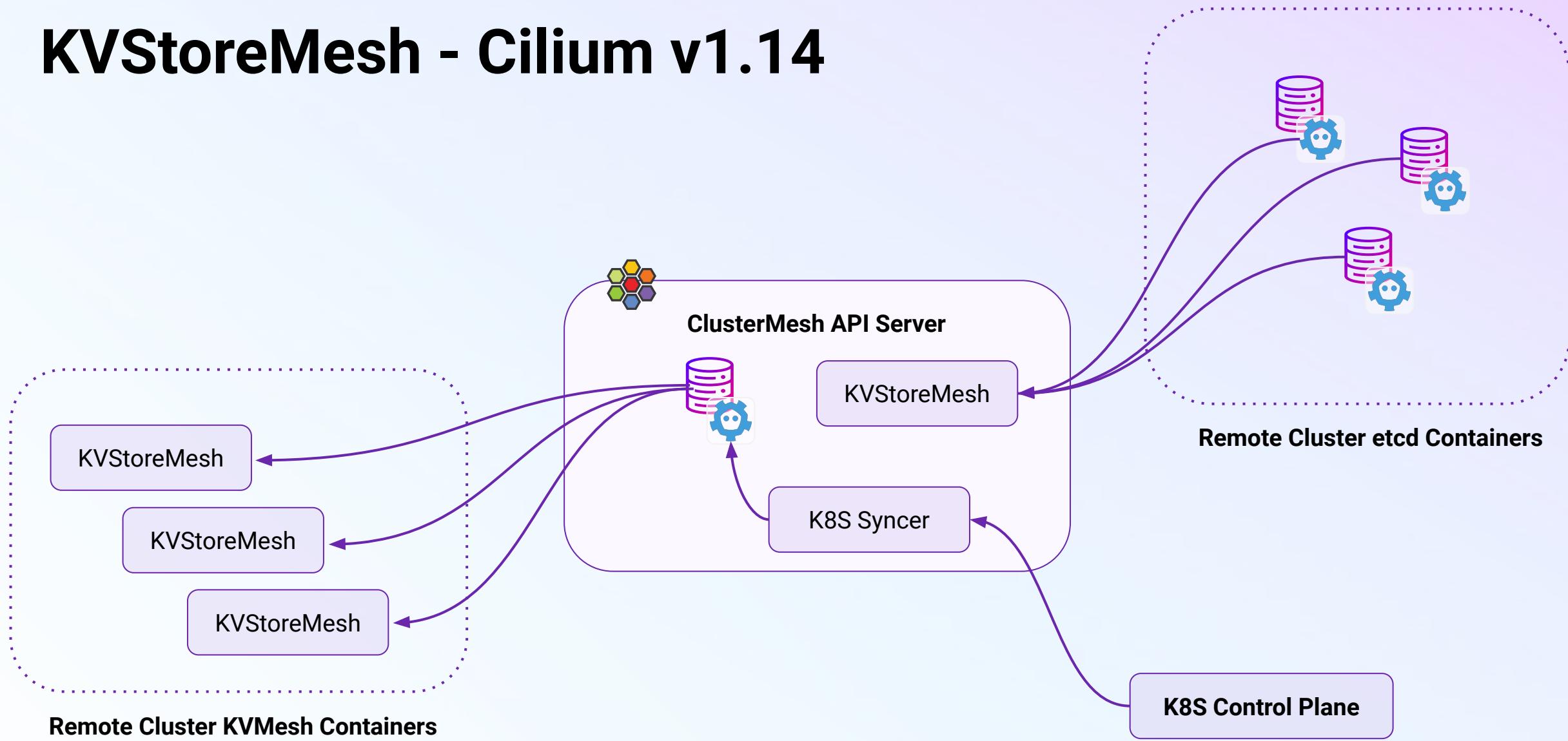
# Arthur Chiao

## eBPF Summit 2022



[youtube.com/watch?v=NIsU4I950I4](https://youtube.com/watch?v=NIsU4I950I4)

# KVStoreMesh - Cilium v1.14



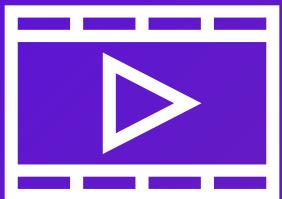
**m+n** updates

**m = #clusters in mesh**

**n = #agents per cluster**

# Ryan Drew

## CiliumCon NA 2023



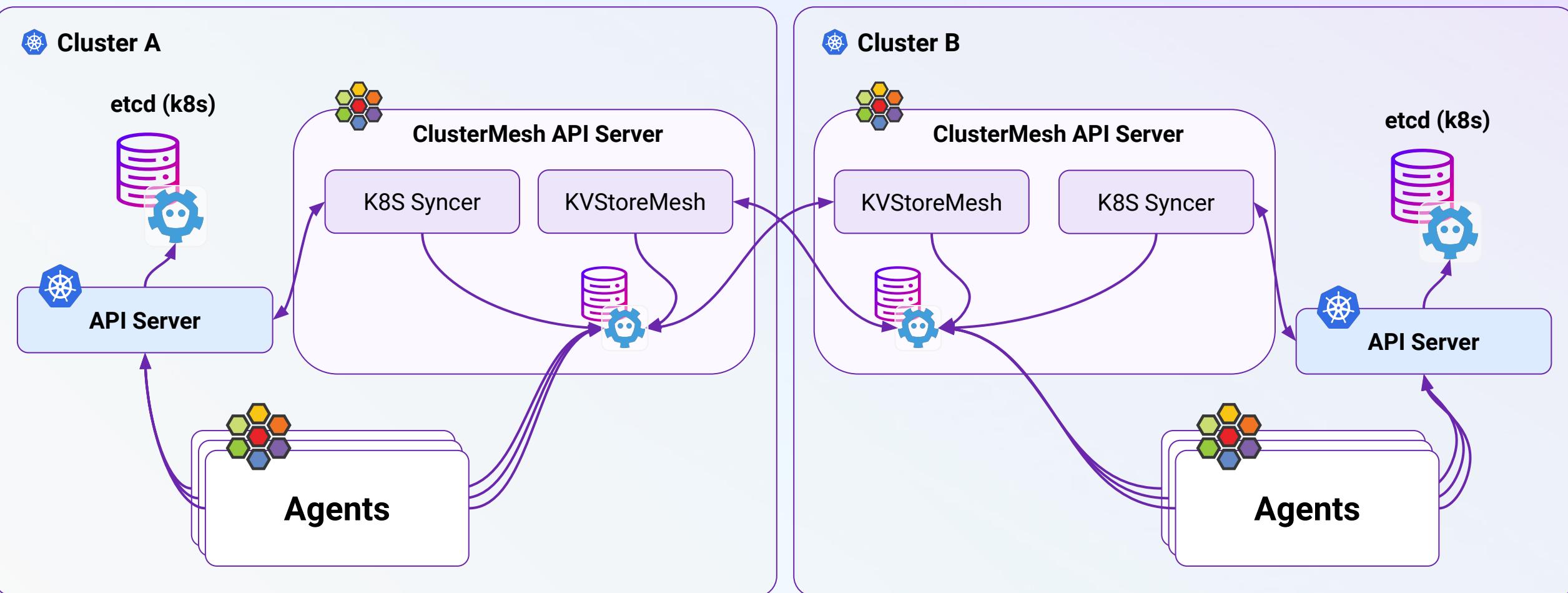
### Why KVStoreMesh? Lessons Learned from Scale Testing Cluster Mesh with 50k Nodes Across 255 Clusters

*Ryan Drew, Isovalent*



[youtube.com/watch?v=biZMCj1rLoM](https://youtube.com/watch?v=biZMCj1rLoM)

# KVStoreMesh - CRD Mode Only?



# Let's take a look at the data format

```
~$ kubectl get pods -l app=backend -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE
backend-6b8d95757c-wmsxg   2/2     Running   0          3d18h   10.1.1.53   some-node

~$ kubectl get ciliumendpoints backend-6b8d95757c-wmsxg -o wide
NAME           SECURITY   IDENTITY   ENDPOINT STATE   IPV4          IPV6
backend-6b8d95757c-wmsxg   91392      ready      10.1.1.53 fd00:10:1:1::4a0b
```

# Let's take a look at the data format

```
~$ kubectl get pods -l app=backend -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE
backend-6b8d95757c-wmsxg  2/2     Running   0          3d18h   10.1.1.53   some-node

~$ kubectl get ciliumendpoints backend-6b8d95757c-wmsxg -o wide
NAME           SECURITY   IDENTITY   ENDPOINT STATE   IPV4          IPV6
backend-6b8d95757c-wmsxg  91392      ready      10.1.1.53 fd00:10:1:1::4a0b
```

Key	Value
cilium/cache/identities/v1/cluster-a/id/91392	k8s:io.cilium.k8s.namespace.labels.kubernetes.io/metadata.name=backends; k8s:io.cilium.k8s.policy.cluster=cluster-a; k8s:io.cilium.k8s.policy.serviceaccount=default; k8s:io.kubernetes.pod.namespace=backends; k8s:name=backend;
cilium/cache/ip/v1/cluster-a/10.1.1.53	{ "IP": "10.1.1.53", "Mask": null, "HostIP": "192.168.16.2", "ID": 91392, "Key": 0, "Metadata": "", "K8sNamespace": "backends", "K8sPodName": "backend-6b8d95757c-wmsxg" }

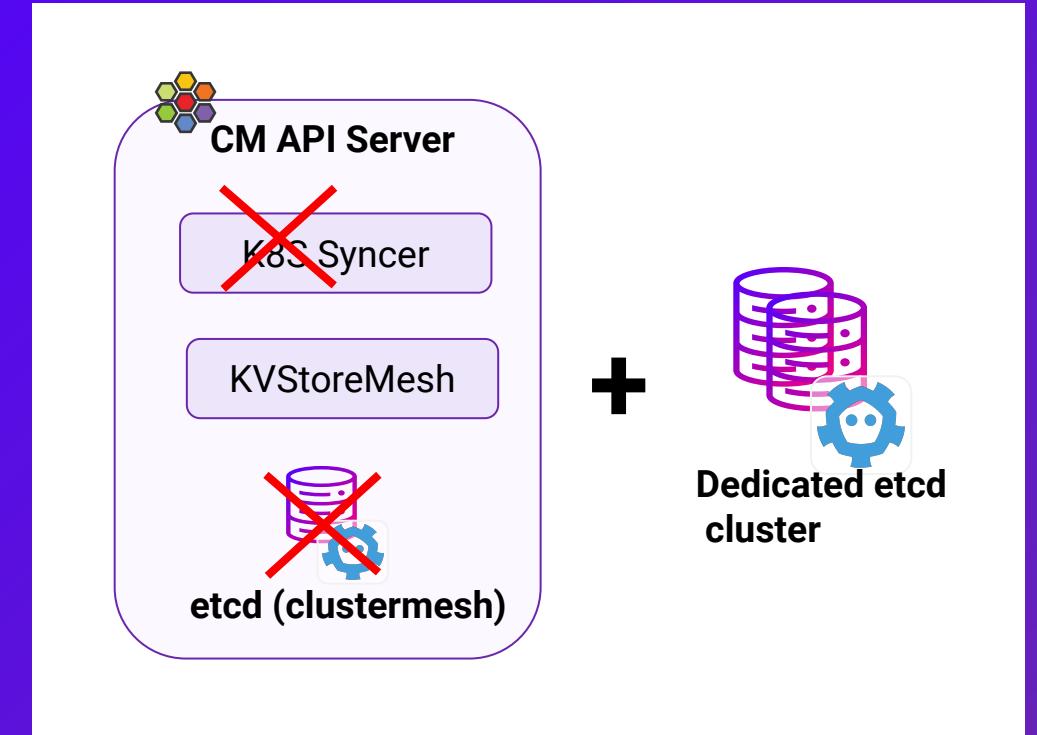
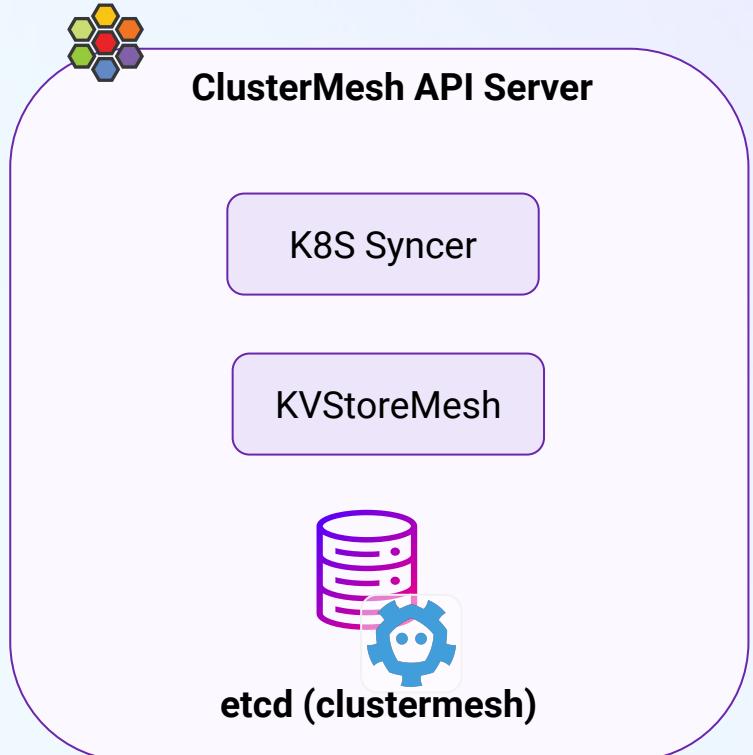
# 04 KV KV Mesh?

# KVStoreMesh

KVStoreMesh

+

KVStore Identity Mode



# Last Piece of the puzzle

[1.14] operator: propagate CiliumClusterConfig when in kvstore mode #32349



Merged

idelossa merged 1 commit into [cilium:v1.14](#) from [DataDog:hmalla/op\\_cluster\\_config](#) on May 7

Conversation 9

Commits 1

Checks 45

Files changed 1



hemanthmalla commented on May 3 · edited

Member ...

1.14 backport of [#27109](#) to support running `kvstoremesh` in `kvstore` identity allocation mode. Currently cluster-config in local kvstore is only set by apiserver component of clustermesh-apiserver. In this config, we don't need to run apiserver component. So, without this commit remote clusters cannot discover cluster-id and will default to using 0 as cluster ID.



hemanthmalla requested a review from [cilium/tophat](#) as a code owner 7 months ago

Reviewers

idelossa

youngnick

giorio94

Assignees

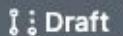
No one—[assign yourself](#)



[github.com/cilium/cilium/pull/32349](https://github.com/cilium/cilium/pull/32349)

# Last Piece of the puzzle

## helm: kvstoremesh with external etcd #34174

 Draft

calcium6 wants to merge 3 commits into `cilium:main` from `calcium6:kvstoremesh-deploy` 

 Conversation 24

 Commits 3

 Checks 11

 Files changed 14



calcium6 commented on Aug 5

First-time contributor 

When running clustermesh-apiserver, there should be an option to enable and disable containers that are not needed.  
For example the etcd and apiserver containers are not necessary when running kvstoremesh with an external kvstore.

This commit adds the options to disable these containers.



 calcium6 requested review from `cilium/sig-k8s`, `cilium/helm`, `cilium/sig-clustermesh` and `cilium/docs-structure` as code owners 3 months ago



[github.com/cilium/cilium/pull/34174](https://github.com/cilium/cilium/pull/34174)

# KVStoreMesh

Cluster A

etcd (k8s)



API Server

ClusterMesh API Server

K8S Syncer

KVStoreMesh



Agents

Cluster B

ClusterMesh API Server

KVStoreMesh

K8S Syncer



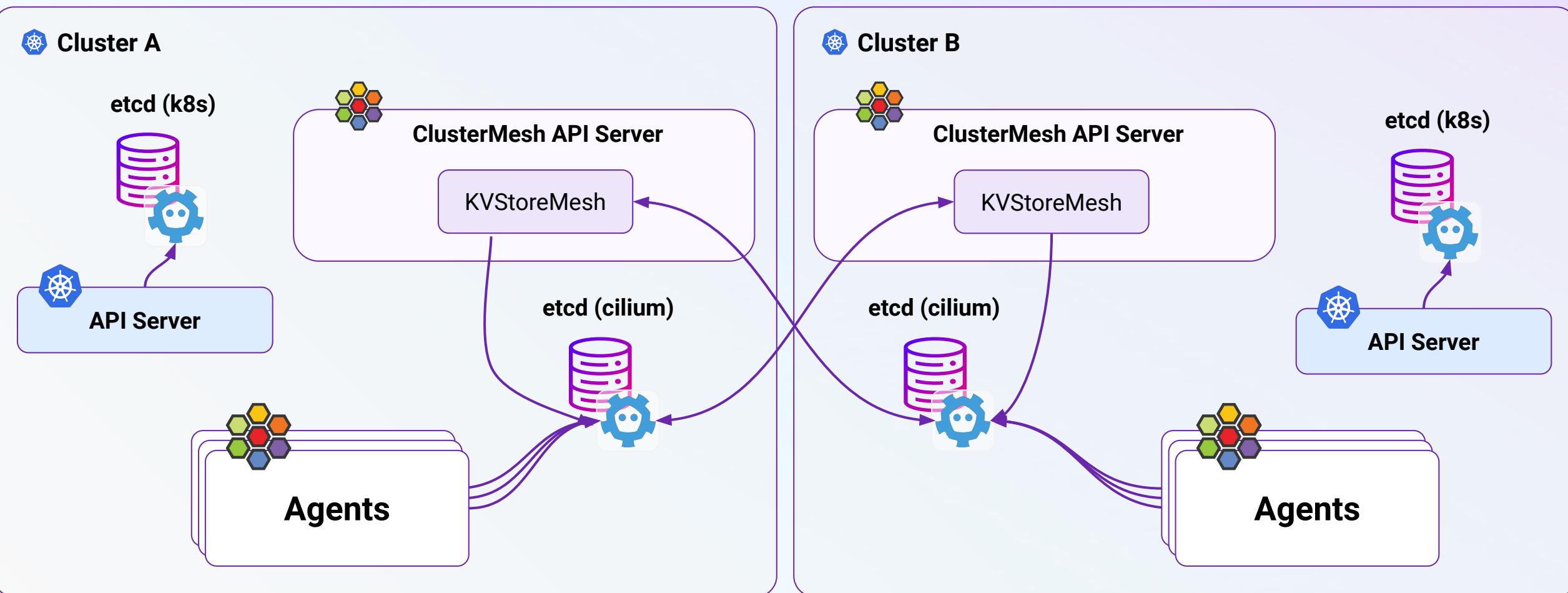
etcd (k8s)



API Server

Agents

# KV KV Mesh



# Mesh Modes

**ClusterMesh**  
+  
**Clustermesh API Server**

Custom Resource Definitions

default  
since v1.16

**KVStoreMesh**

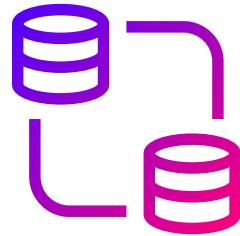
**Original ClusterMesh**

**KVKVMesh?**

External Key-Value Store

# 05 Migration, Monitoring and Best Practices

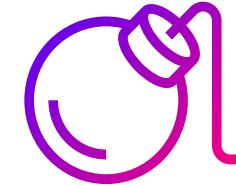
# Moving Parts



What is getting replicated?



Shuffling too much information too fast?



What happens when kvstoremesh dies?

# HA Mode

## Add support for multiple clustermesh-apiserver replicas (ClusterMesh HA)

#31677 [View](#)

**Merged** joamaki merged 3 commits into [main](#) from [pr/thorn3r/clustermeshHA](#) [View](#) on Apr 9

[Conversation 46](#) [Commits 3](#) [Checks 43](#) [Files changed 9](#)

 **thorn3r** commented on Mar 29 · edited [View](#) Member [...](#)

This adds support for running clustermesh-apiserver deployments with multiple replicas for high availability.

Each clustermesh-apiserver pod runs its own etcd cluster. Depending on configuration, either the Cilium Agent or KVStoreMesh instance watches etcd in a remote cluster. All responses from the remote etcd cluster are intercepted and the header is inspected to retrieve the etcd cluster ID. If a failover event occurs and the cluster ID has changed, the remote connection is restarted to ensure that no events are missed and that no invalid data is retained. See individual commit messages for additional details.

[Add support for deploying clustermesh-apiserver with multiple replicas for high availability.](#) [View](#)

  3

**Reviewers**

-  **squeed**
-  **joamaki**
-  **marseel**
-  **nbusseneau**
-  **giorio94**
-  **viktor-kurchenko**

**Assignees**

No one—[assign yourself](#)

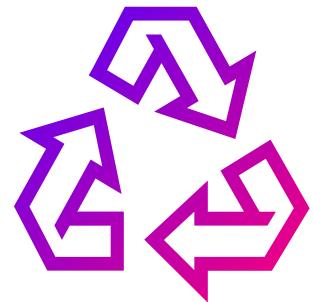
**Labels**



[github.com/cilium/cilium/pull/31677](https://github.com/cilium/cilium/pull/31677)

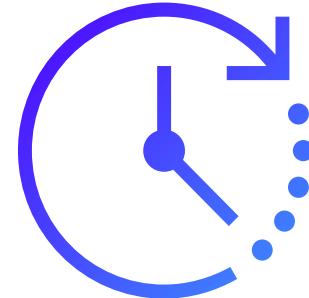
# Primer on etcd storage / garbage collection

## Local Keys



Classic Garbage Collection

## Remote Keys



Leases with TTLs

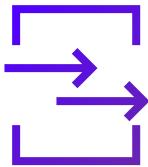
# Safely meshing with existing clusters



**Key-Value leases TTL tuning**



**KVStore QPS  
Bootstrap + Regular**



**Unidirectional Meshing**

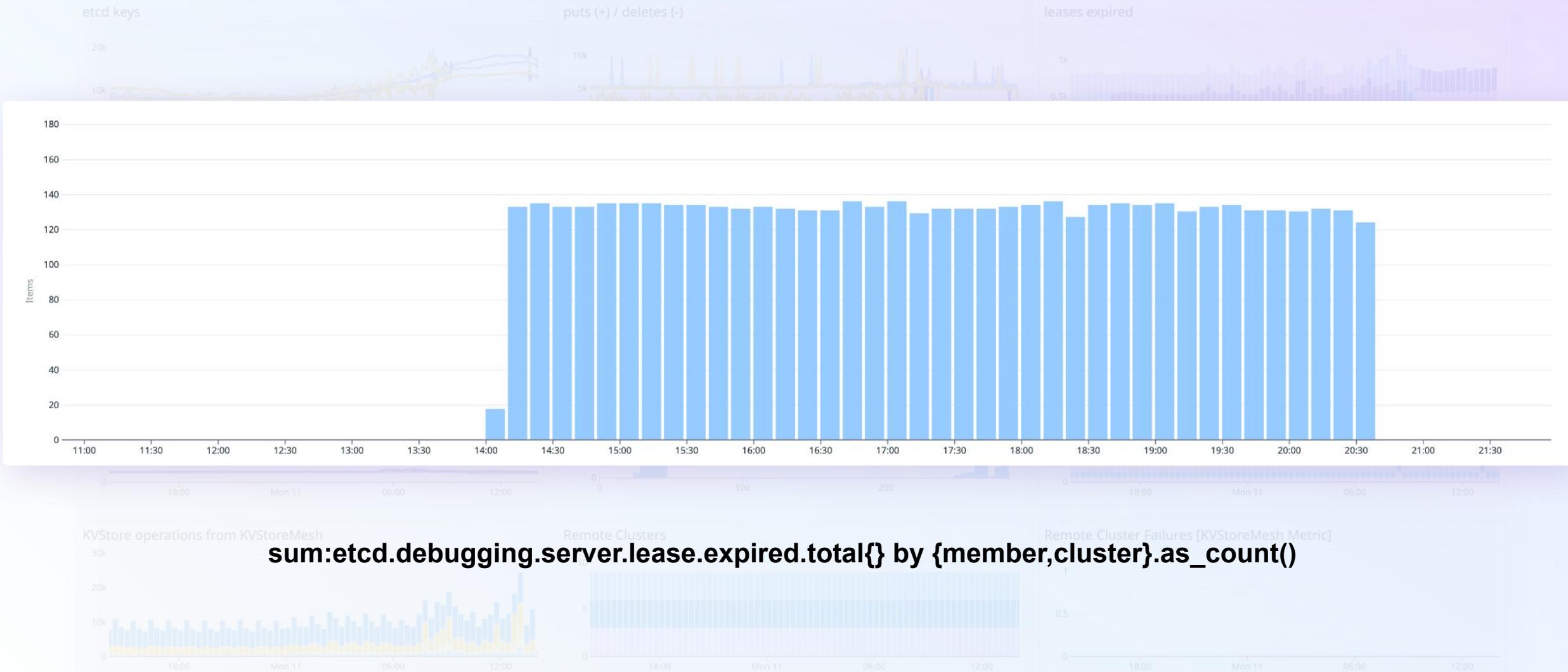


**World -> Known Identity**

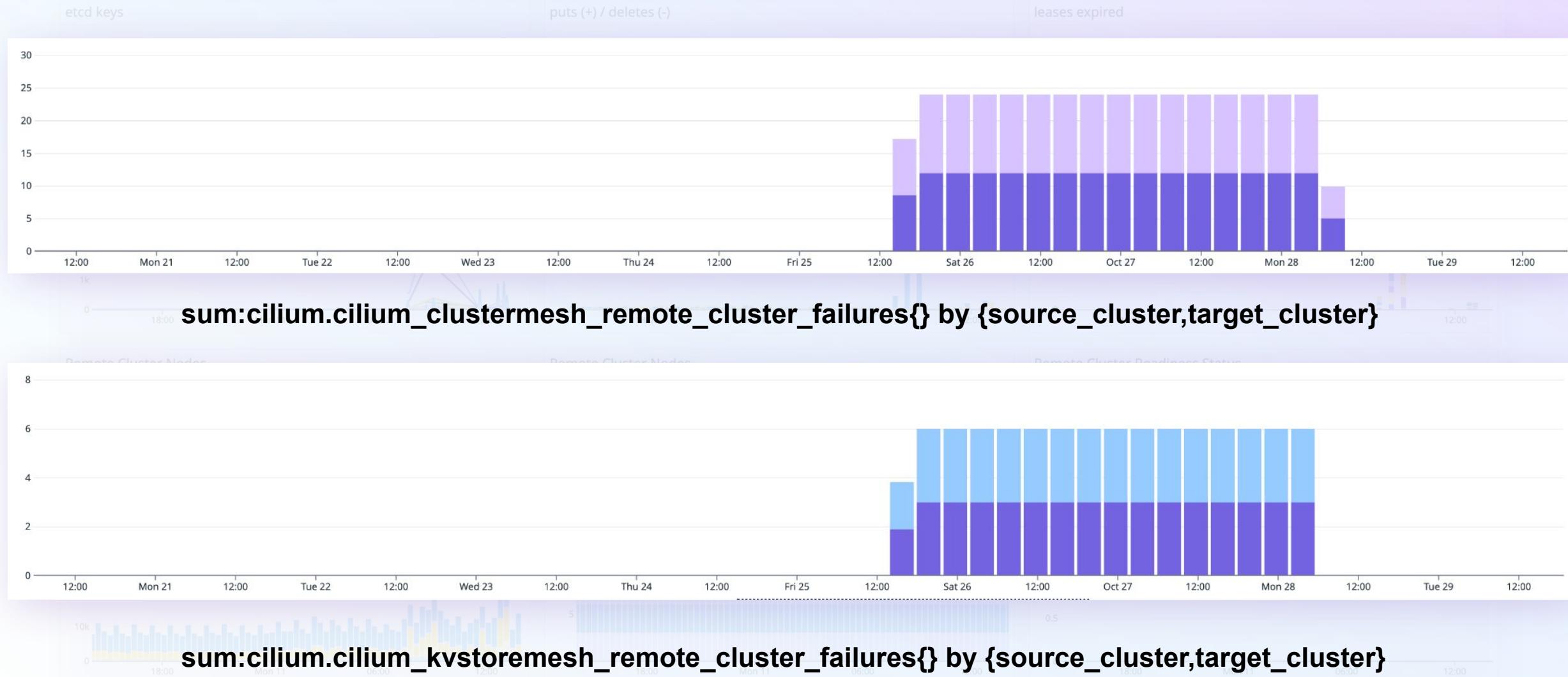
# Metrics to watch



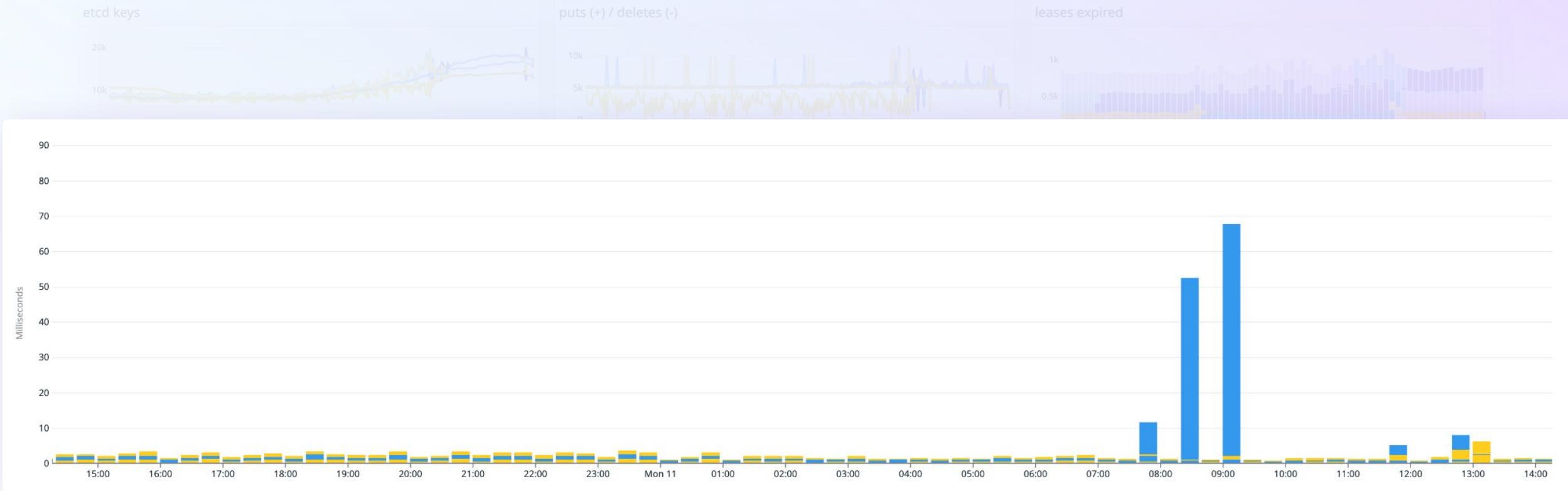
# etcd MVCC lease expirations



# ClusterMesh & KVStoreMesh failures



# etcd operations and rate limiting



KVStore operations from KVStoreMesh  
**max:cilium.cilium\_kvstoremesh\_api\_limiter\_wait\_duration\_seconds{} by {api\_call,value,cluster\_name}**



# Debugging

```
> k exec -it cilium-kvstoremesh-6884dd75f6-xkjmn -- clustermesh-apiserver kvstoremesh-dbg troubleshoot  
Found 4 cluster configurations
```

Cluster "ciliumcon":

```
  Configuration path: /var/lib/cilium/clustermesh/ciliumcon
```

↳ Endpoints:

- https://etcd-0.etcd.ciliumcon-k8s-etcd-cilium.svc.kubecon.dog:2379
  - ✓ Hostname resolved to: 10.165.171.105
  - ✓ TCP connection successfully established to 10.165.171.105:2379
  - ✓ TLS connection successfully established to 10.165.171.105:2379
  - ⓘ Negotiated TLS version: TLS 1.3, ciphersuite TLS\_AES\_128\_GCM\_SHA256
  - △ Could not retrieve etcd server version
- https://etcd-1.etcd.ciliumcon-k8s-etcd-cilium.svc.kubecon.dog:2379
  - ✓ Hostname resolved to: 10.165.184.210

....

....

....

⚡ Digital certificates:

- ✓ TLS Root CA certificates:

- Serial number: 77:38:02:43:e4:e7:13:be:a6:bd:16:d1:ce:aa:da:30:3f:7e:54:4d
- Subject: CN=ciliumcon-k8s-etcd-cilium
- Issuer: CN=ciliumcon-k8s-etcd-cilium
- Validity:
  - Not before: xxxx-04-10 20:35:11 +0000 UTC
  - Not after: xxxx-04-09 20:35:41 +0000 UTC

- ✓ TLS client certificates:

- Serial number: 79:d0:cf:b7:cf:96:6d:65:90:02:a8:29:7f:2b:42:1b:6c:f7:77:58
- Subject: CN=cilium-kvstoremesh
- Issuer: CN=ciliumcon-k8s-etcd-cilium
- Validity:
  - Not before: xxxx-11-03 18:56:21 +0000 UTC
  - Not after: xxxx-11-10 18:56:51 +0000 UTC

ⓘ Etcd client:

- ✓ Etcd connection successfully established
- ⓘ Etcd cluster ID: ce017a81fcff3e8b

# Future Work

Making `kvstoremesh` process aware of “relevant” identities

---

[CFP-27752](#): Operator Manages Cilium Identities

---

Only replicating keys relevant for enabled features

# Thanks!

Blog: <https://www.datadoghq.com/blog/engineering/>

We're hiring! <https://www.datadoghq.com/careers/>

@mvisionneau

@hemanthmalla

#sig-scalability / #sig-clustermesh on Cilium Slack

Feedback



[sched.co/1izs0](https://sched.co/1izs0)



DATADOG