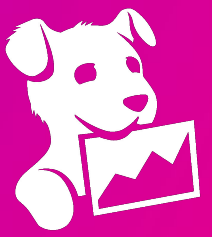
Scaling Network Policy Enforcement Beyond the Cluster Boundary

CiliumCon
NORTH AMERICA



DATADOG

Scaling Network Policy Enforcement Beyond the Cluster Boundary

Hemanth Malla

@hemanthmalla

Maxime Visonneau

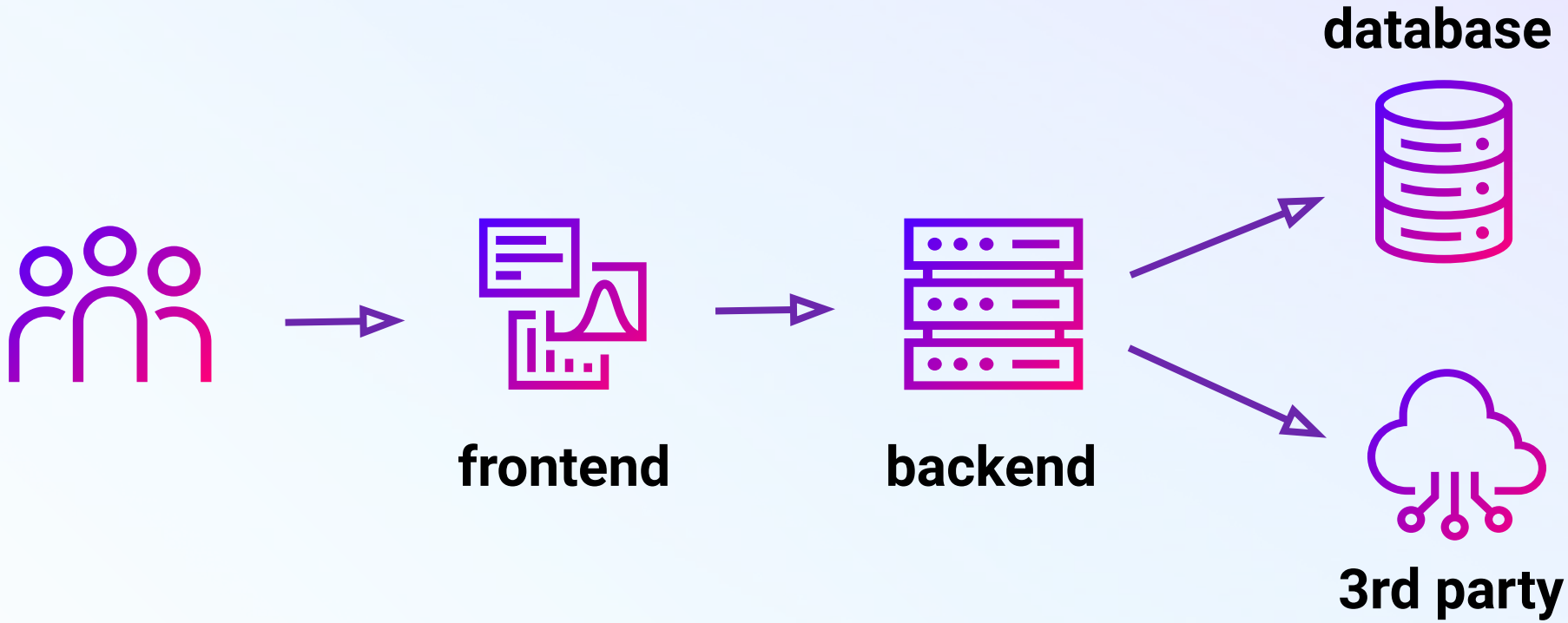
@mvisonneau

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

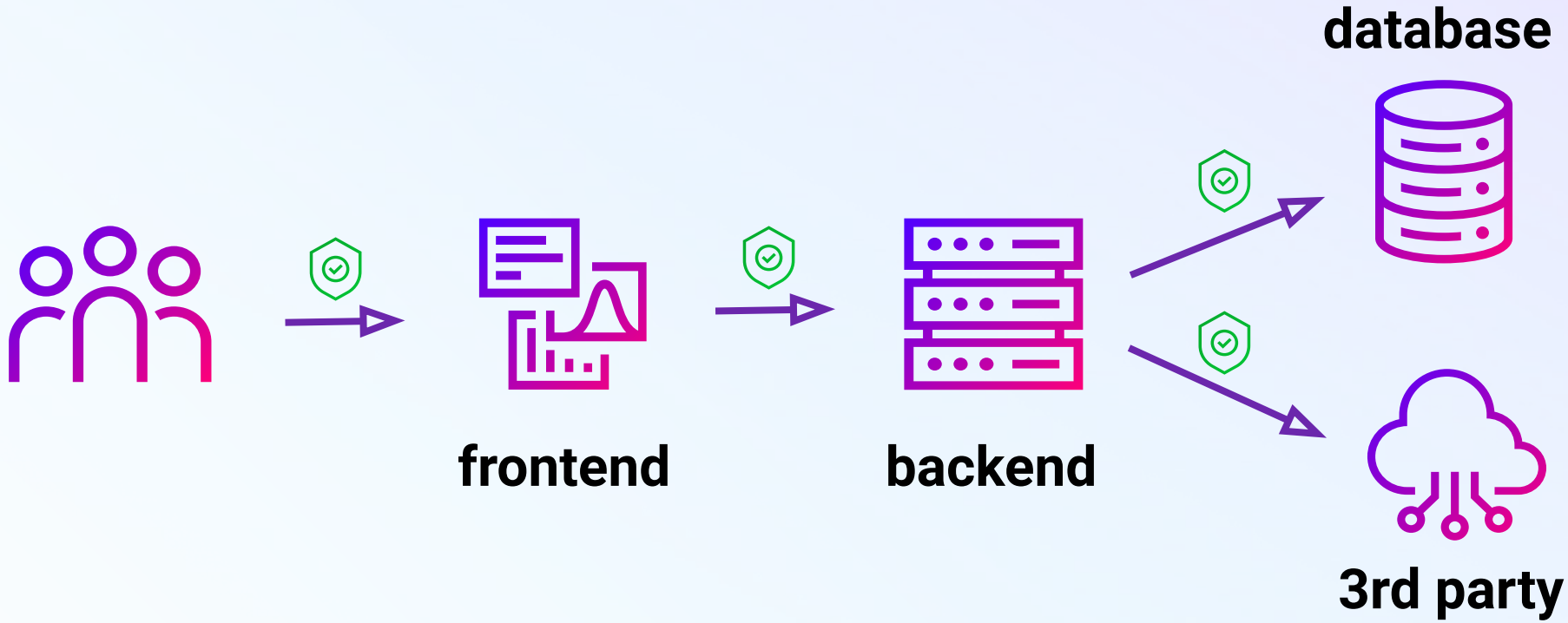
Agenda

01 Background and Concepts

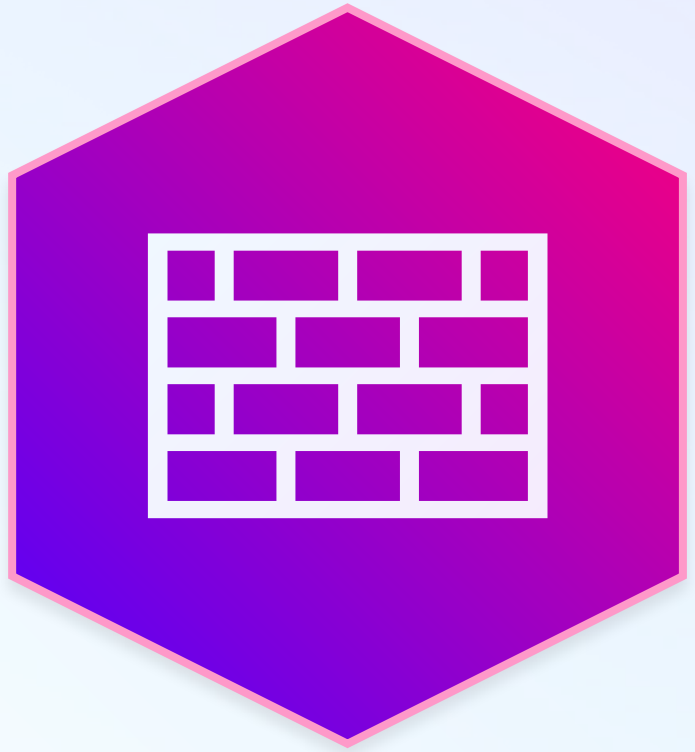
A basic example



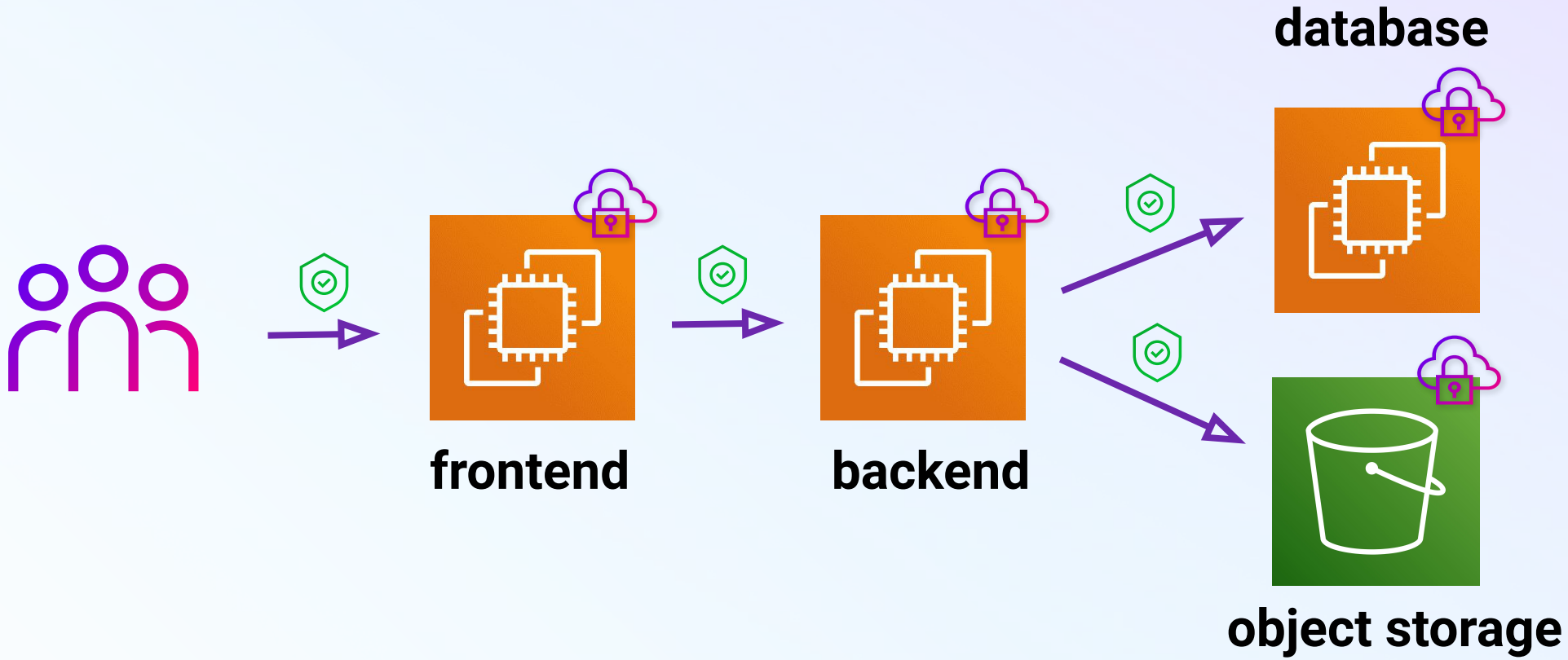
Scoping down accesses to what is necessary



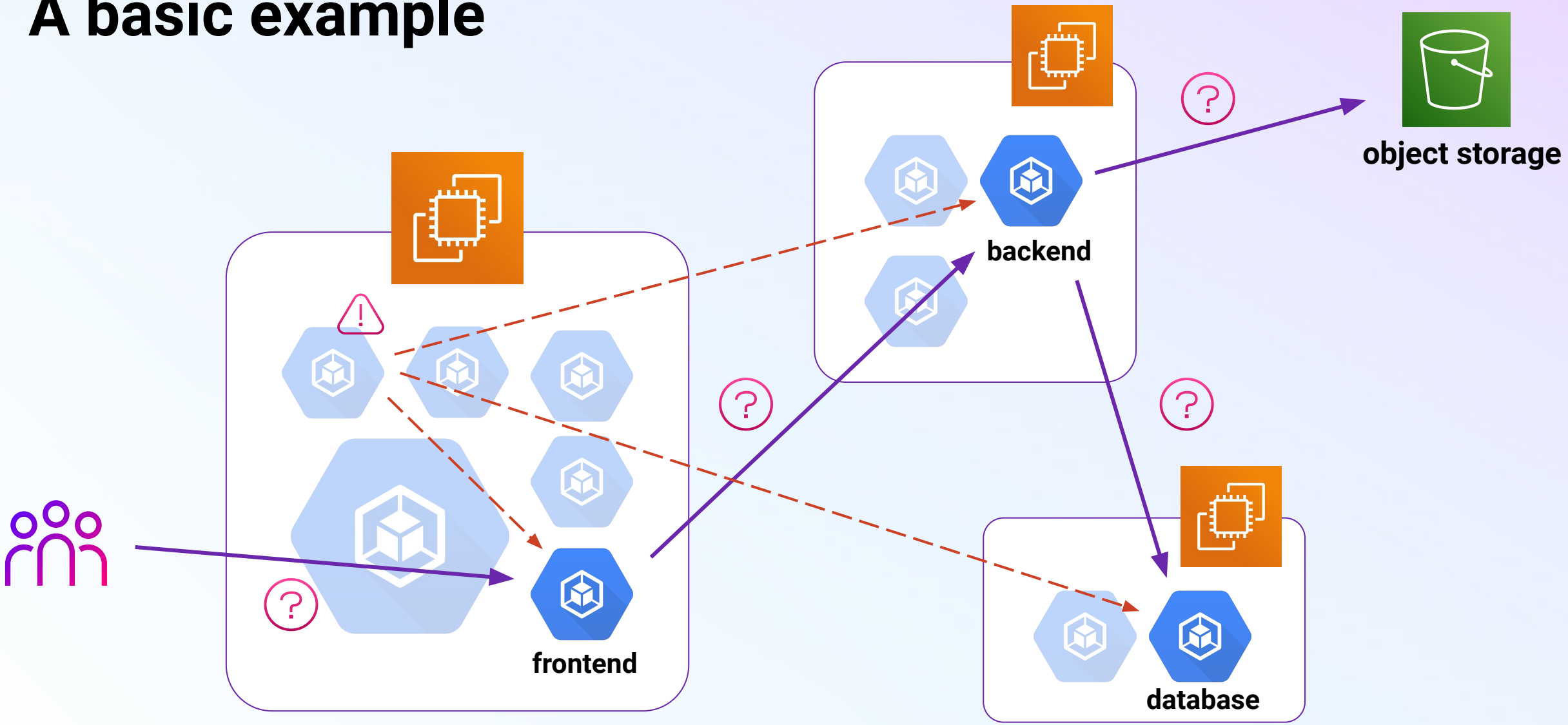
Security primitives

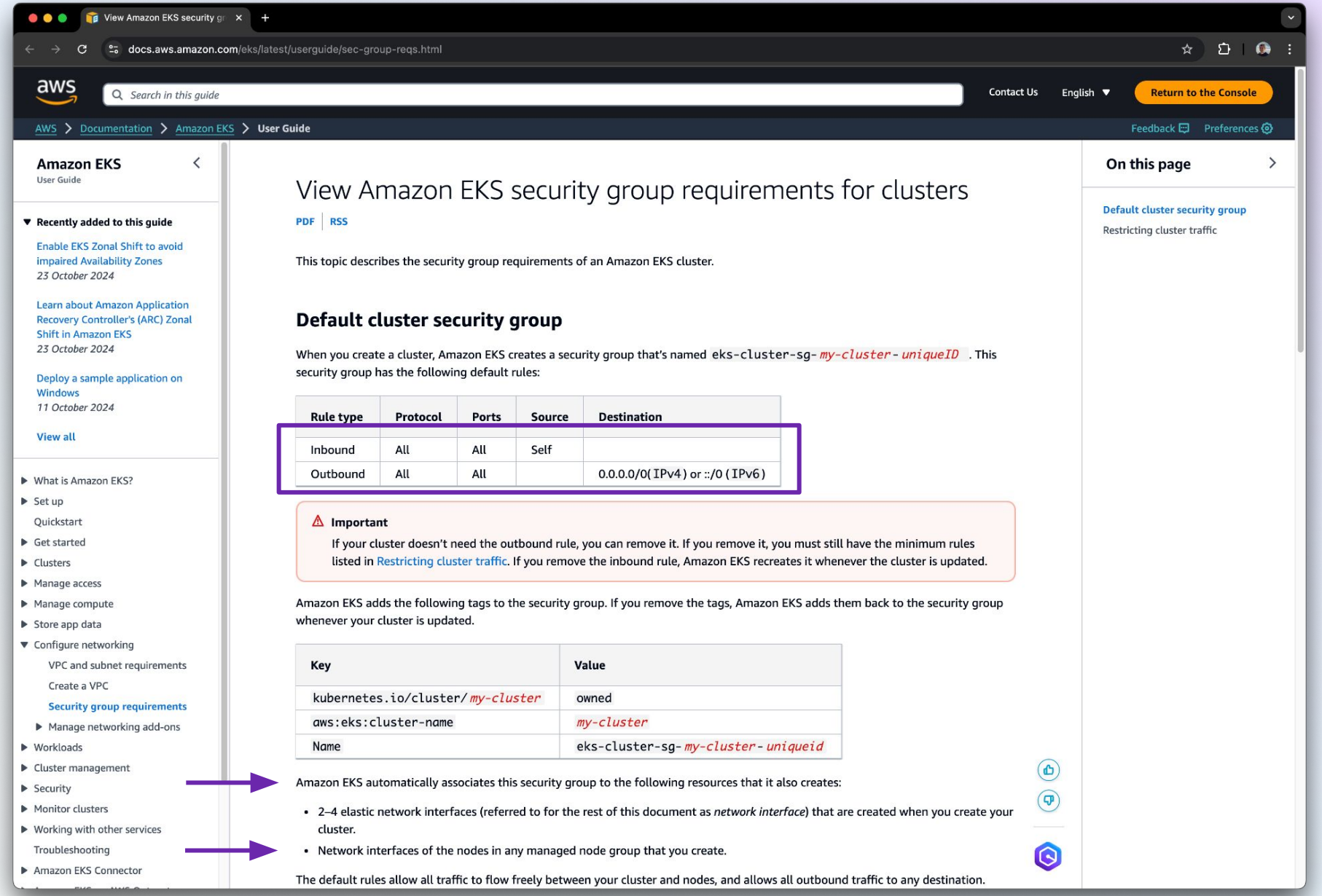


A basic example



A basic example

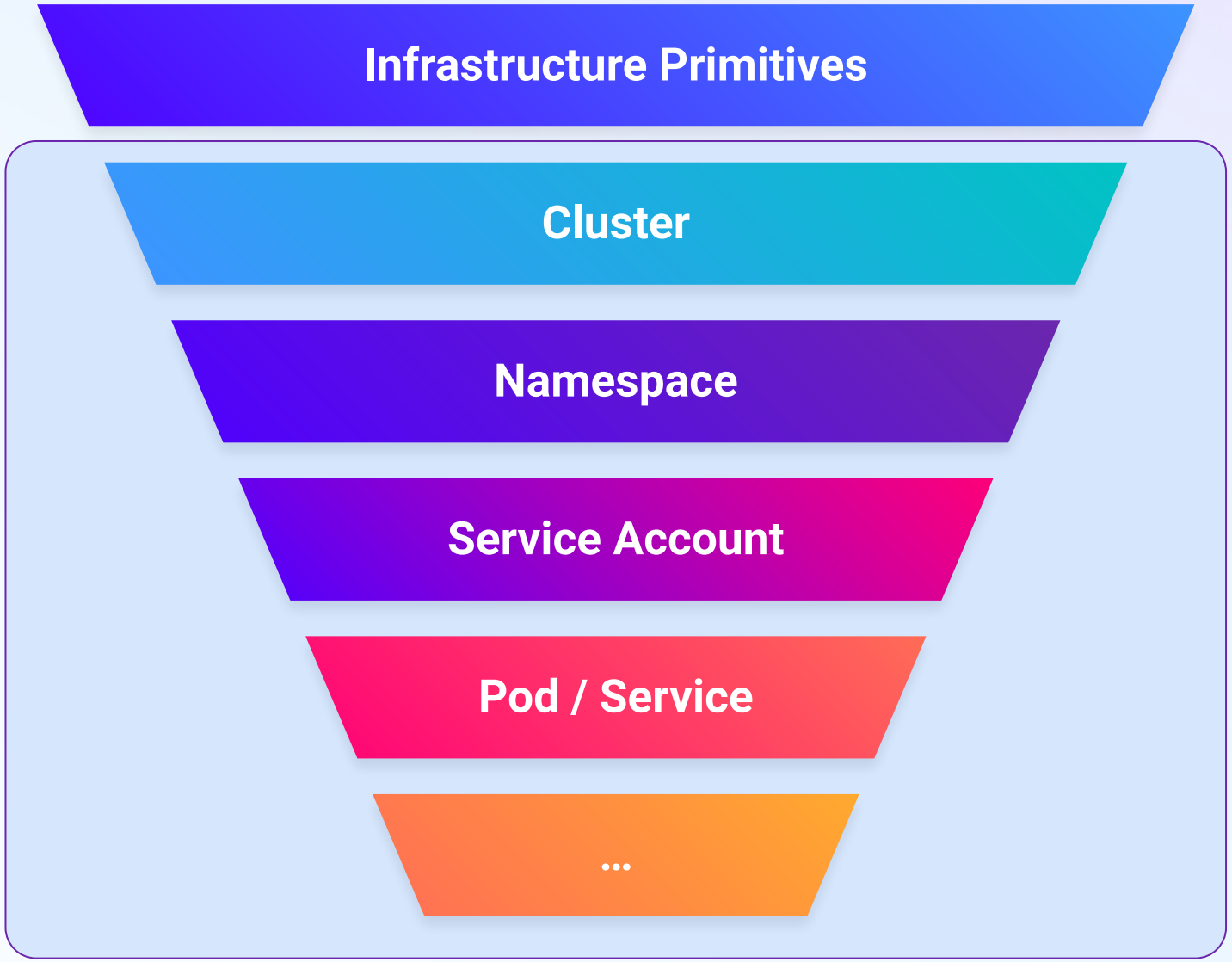




aws docs.aws.amazon.com/eks/latest/userguide/sec-group-reqs.html

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

```yaml
---
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

```yaml
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.

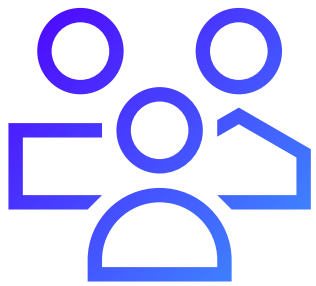
If you want to configure health entity to check cluster-wide connectivity when you start cilium-agent with `enable-policy: always`, you will likely want to enable communications to and from the health endpoint. See Example: Add Health Endpoint.

never

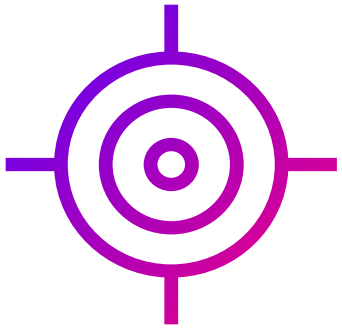
With never mode, policy enforcement is disabled on all endpoints, even if rules do select specific endpoints. In other words, all traffic is allowed from any source (on ingress) or destination (on egress).

docs.cilium.io/en/latest/security/policy/intro

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

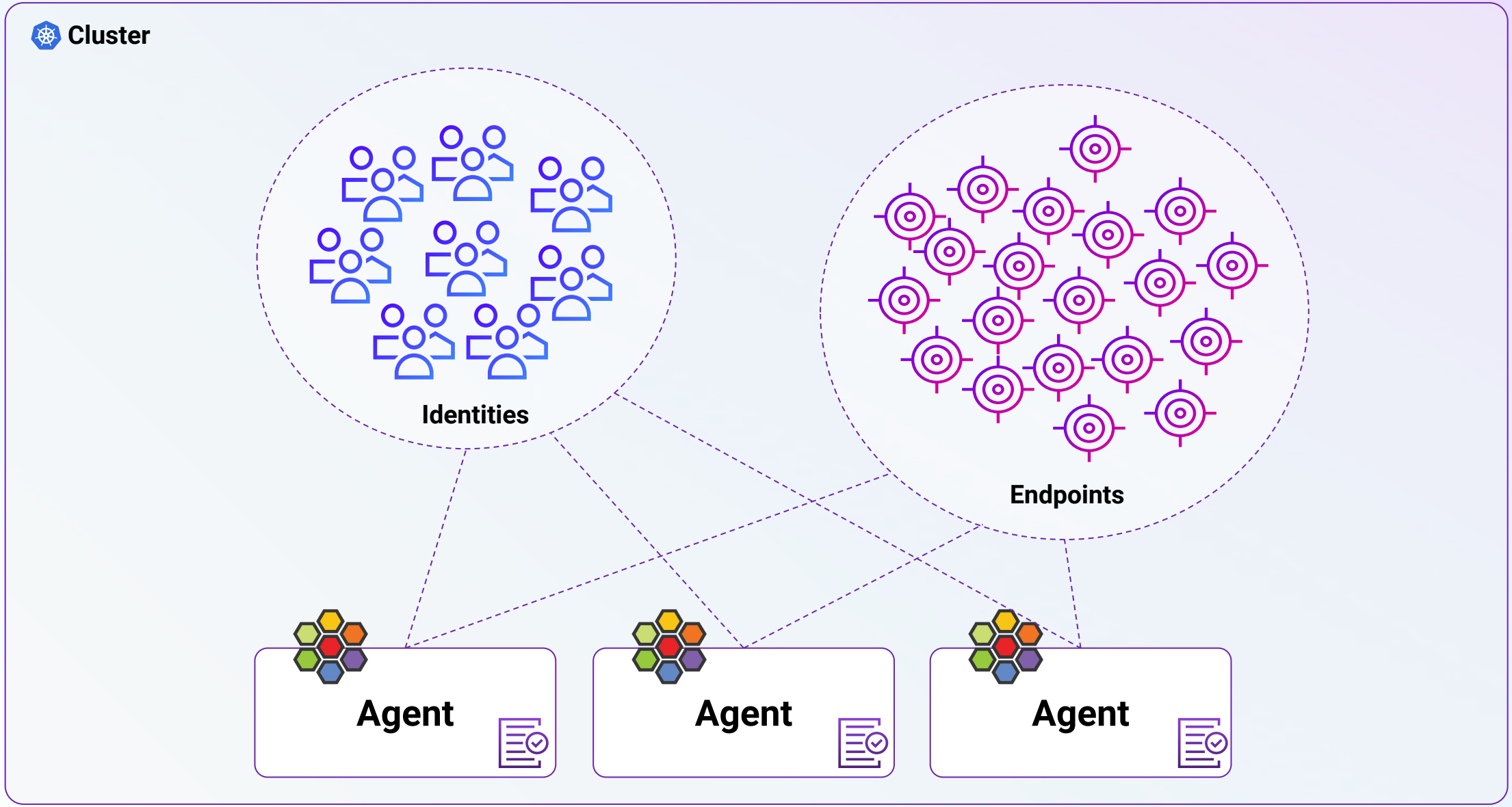
Some references to which pod leverages it

A **unique identifier** within the cluster

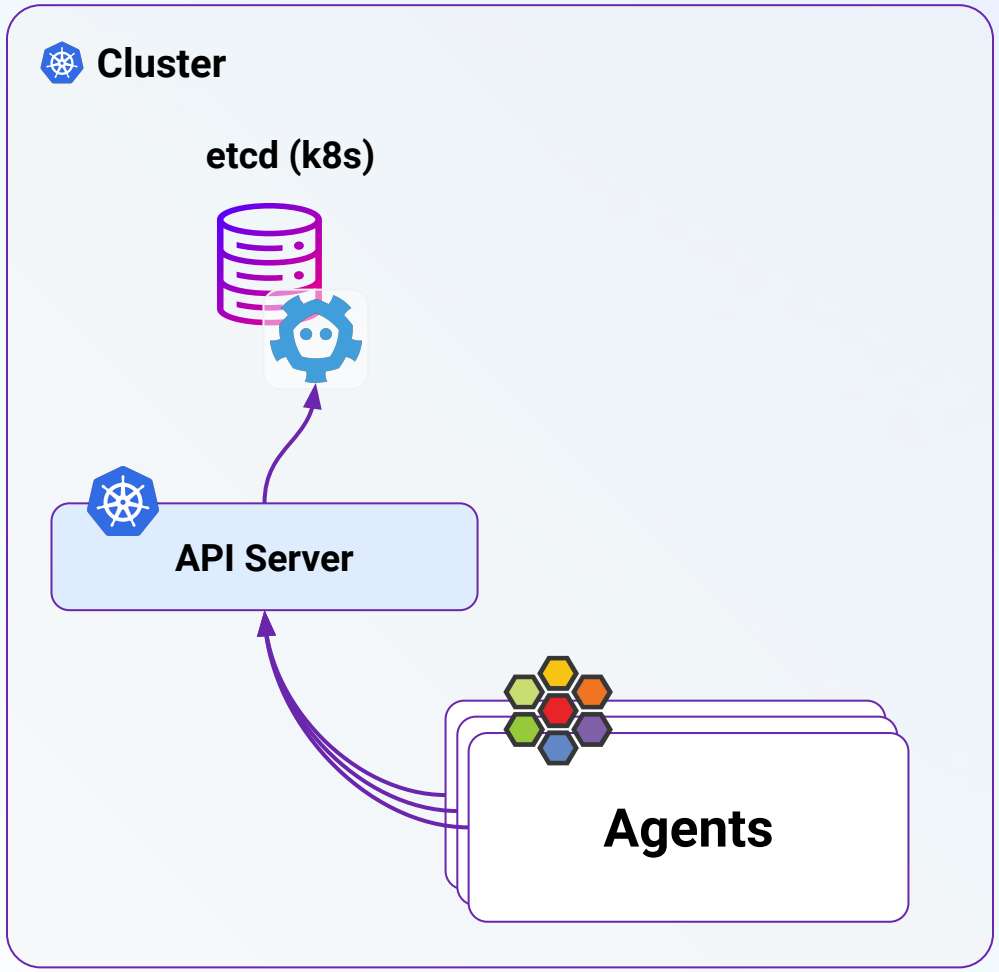
The **identity** it is associated with

Allocated IP addresses and its current **state**

```yaml
---
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
```

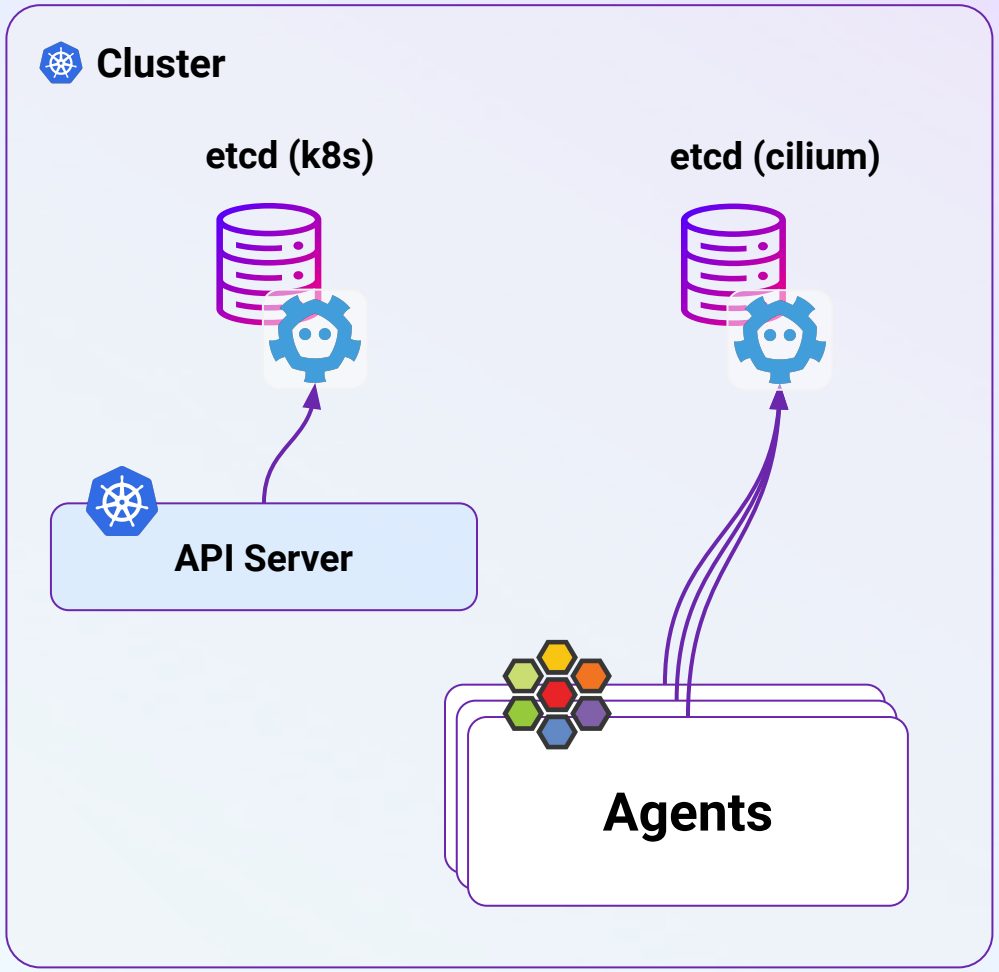

Cluster
Identities
Endpoints
Agent
Agent
Agent

Where is it all stored?



CRD Mode (default)

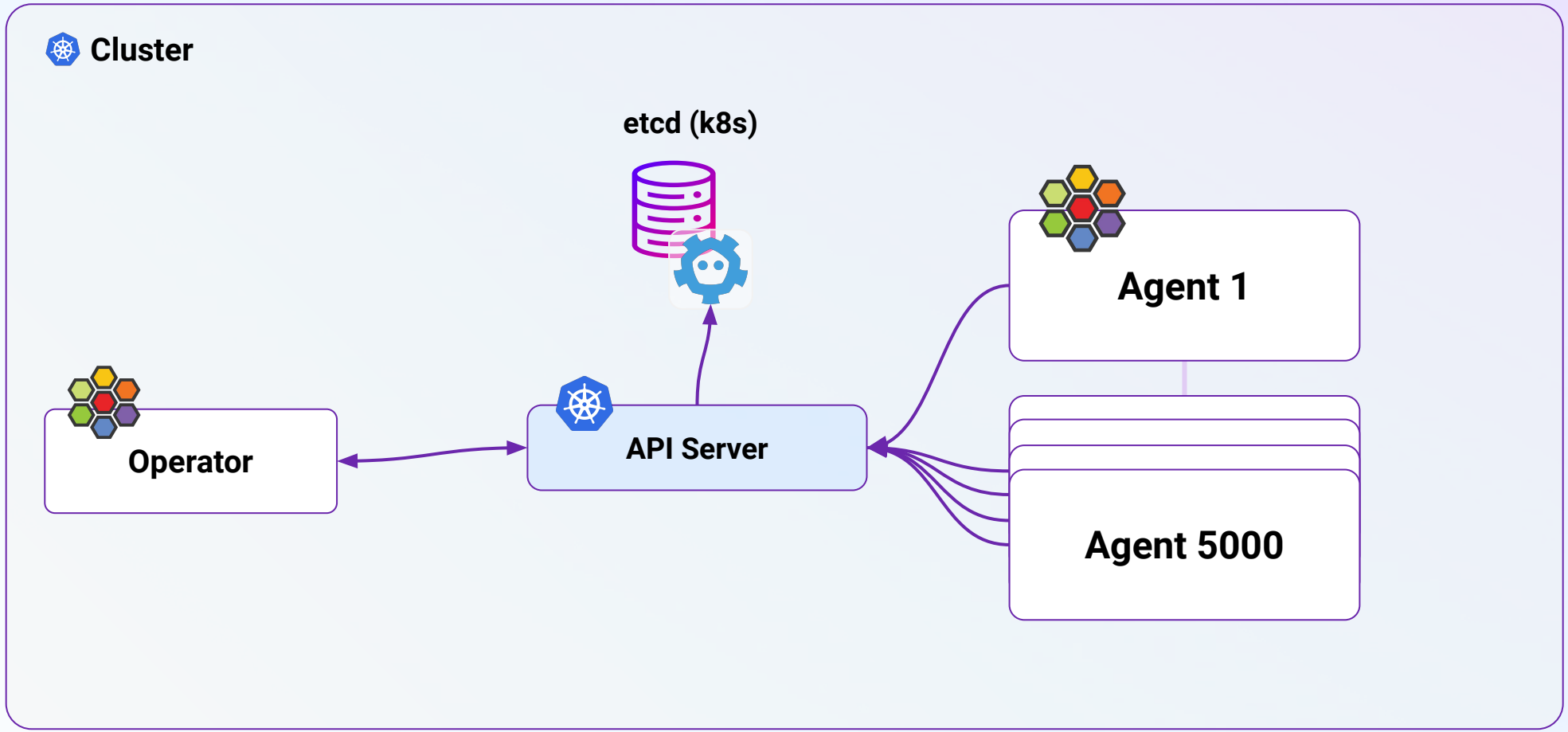
or



External Key-Value Store

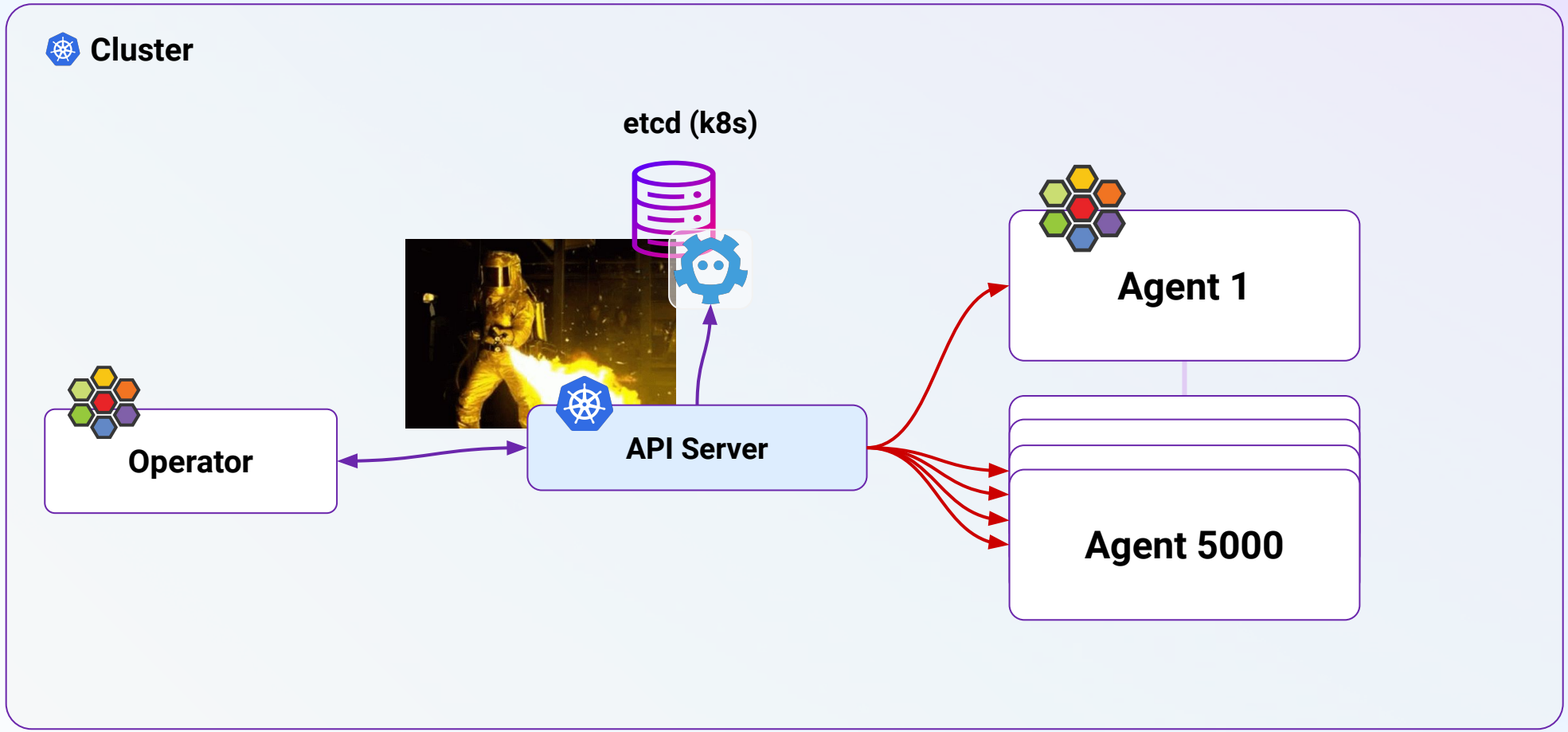
Scala  lity

Scalability



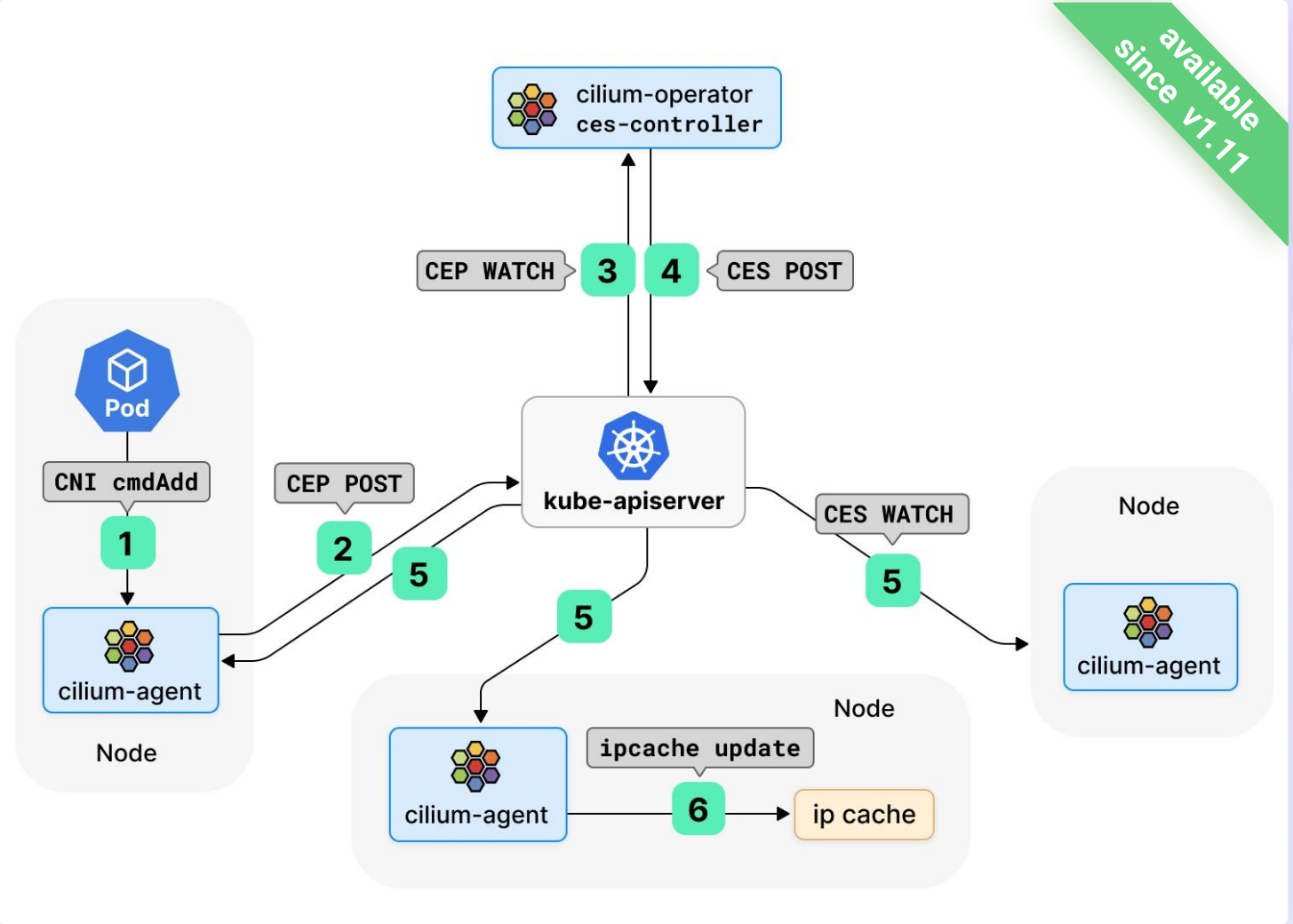

```
~$ kubectl scale deploy/backend \
    --current-replicas 0 \
    --replicas 100
```

Scalability



100 endpoints * 5000 nodes = 500,000 watch updates

Batching to the rescue



isovalent.com/blog/post/2021-12-release-111#cilium-endpoint-slices

At Datadog, for now we use both

DATADOG

< ~1000 nodes

CRD Mode

1000+ nodes

KVStore Mode

@antonipp

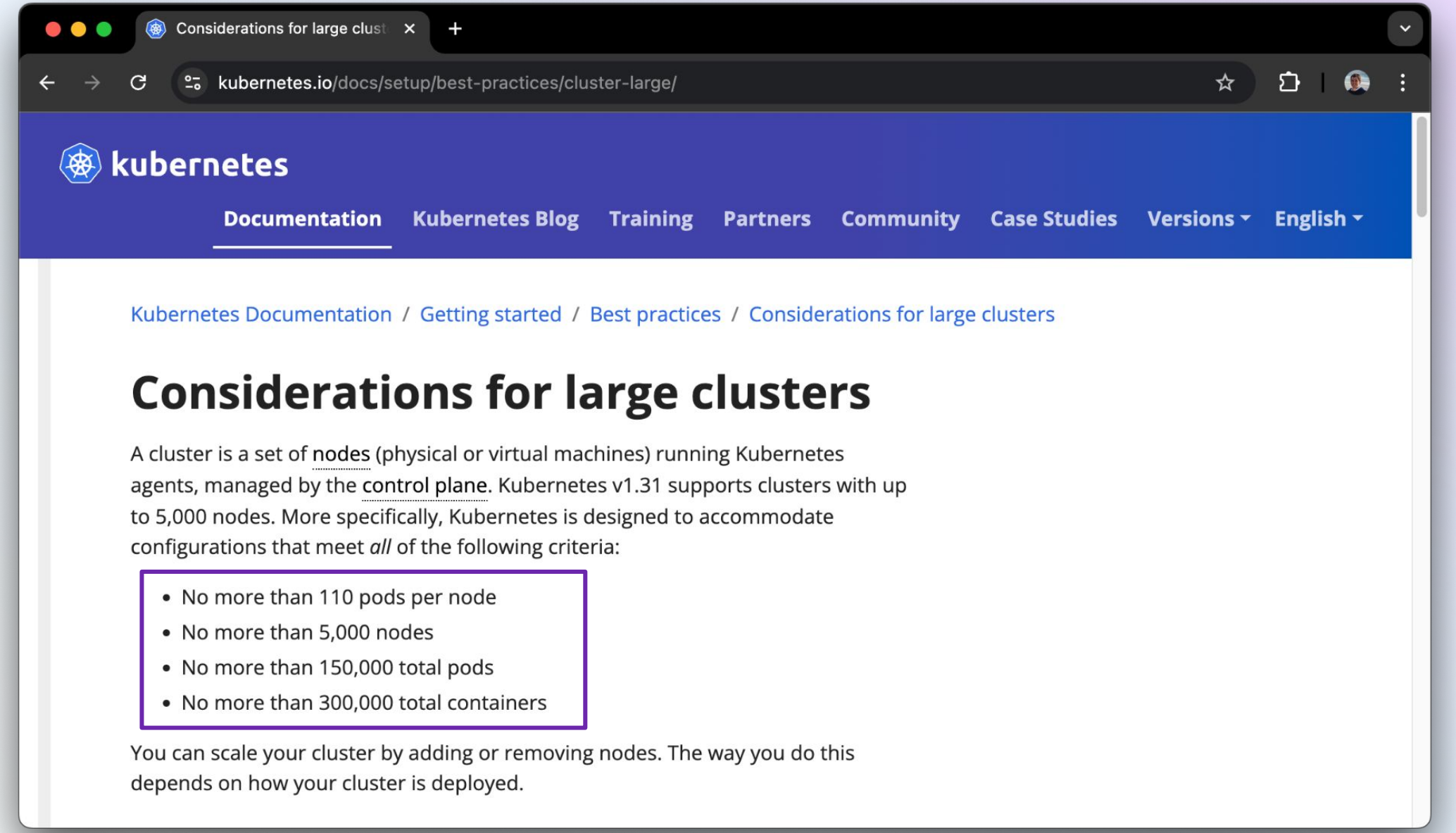
```
--identity-allocation-mode=double-writeread{kvstore,crd}
```

github.com/cilium/cilium/pull/31920

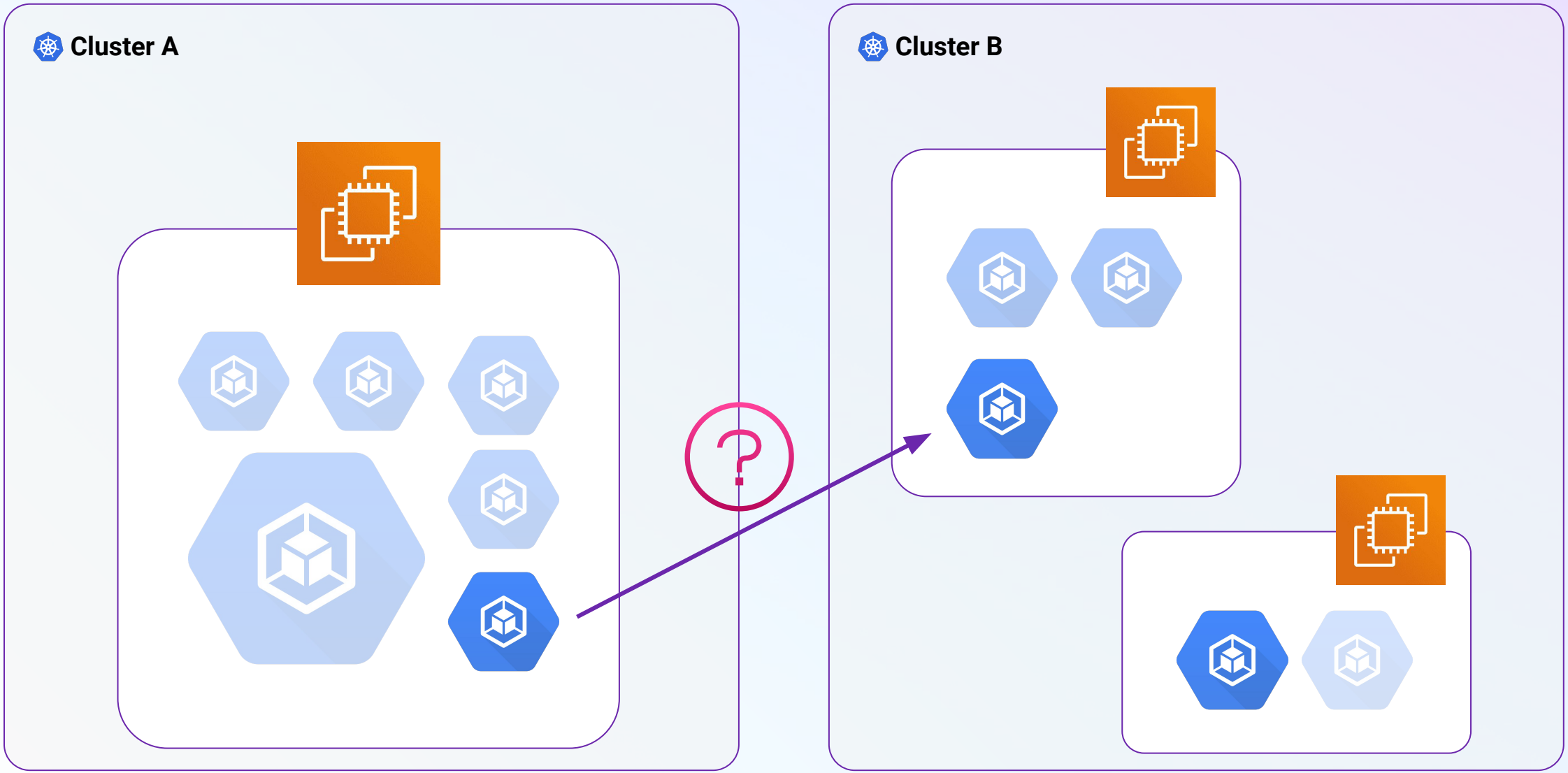
Marcel Zięba
Hemanth Malla

KubeCon Europe 2024

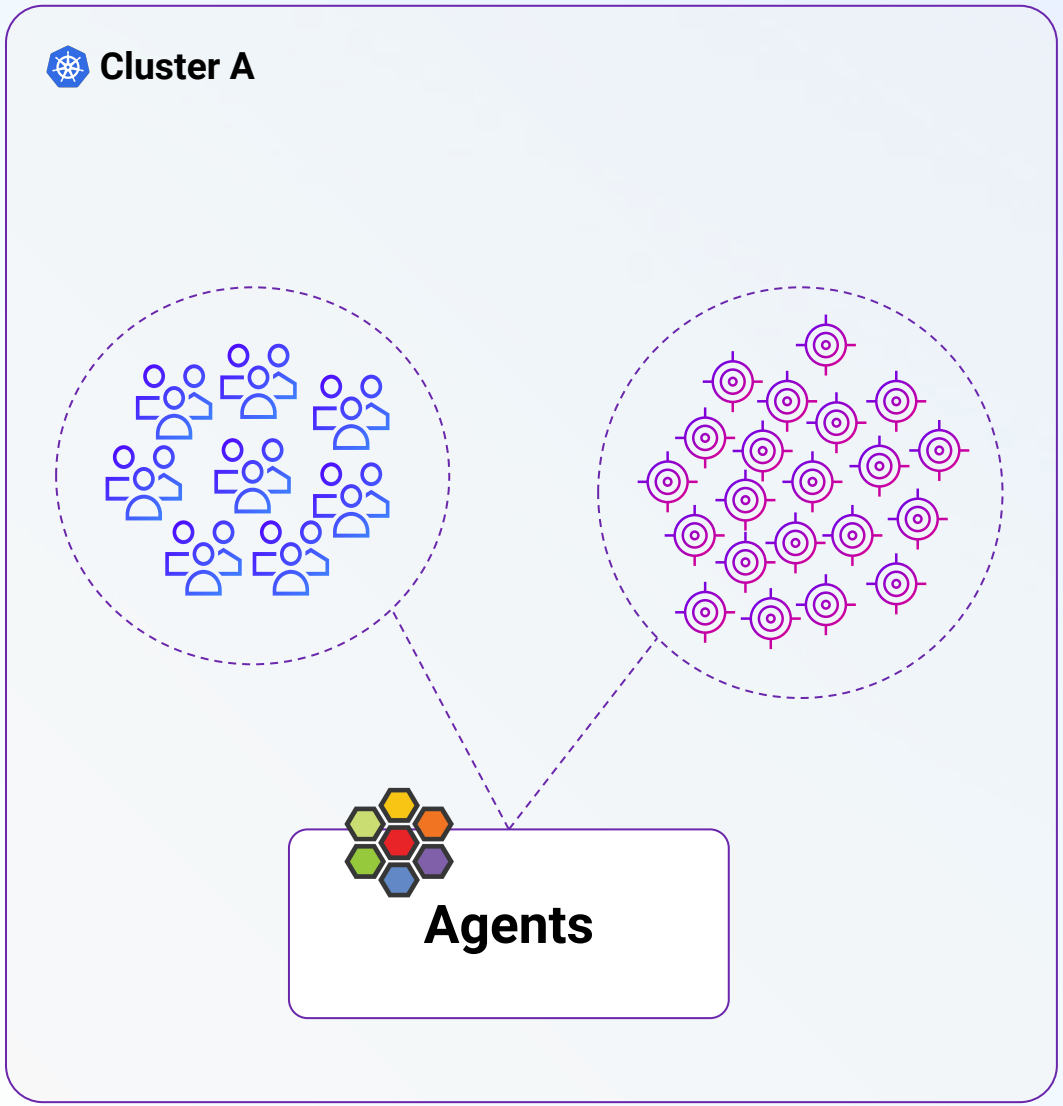
youtube.com/watch?v=o_LutIRlv-A

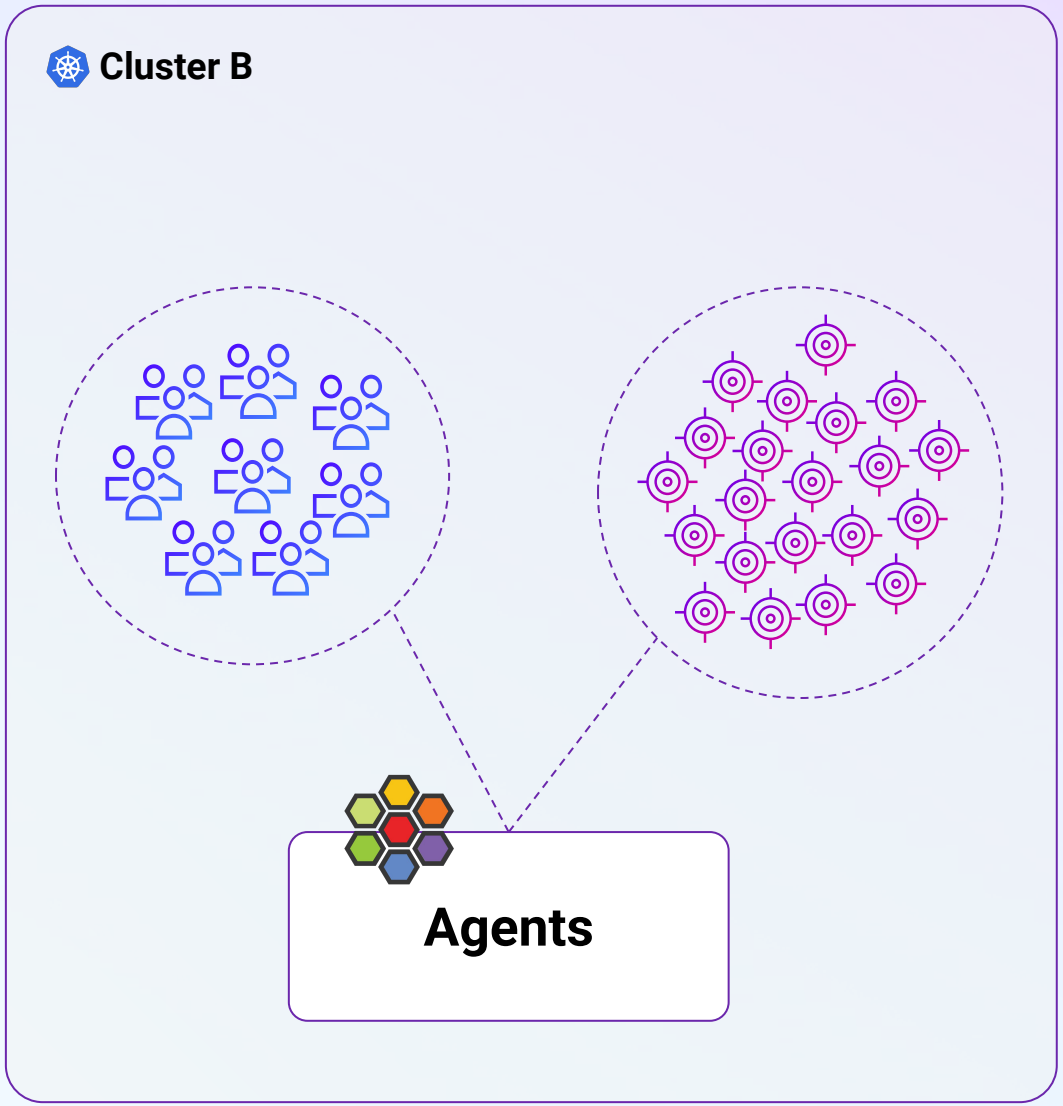

Considerations for large clust
kubernetes.io/docs/setup/best-practices/cluster-large/
kubernetes
Documentation
Kubernetes Blog
Training
Partners
Community
Case Studies
Versions
English
Kubernetes Documentation / Getting started / Best practices / Considerations for large clusters
Considerations for large 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





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
```

docs.cilium.io/en/v1.16/security/policy/language/#allow-to-external-cidr-block

Back to leveraging lower-down primitives?

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

Copy First Line Copy All

```
---
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'
```

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

docs.cilium.io/en/v1.16/security/aws/#policy-language

Or alternative ones?

DNS based

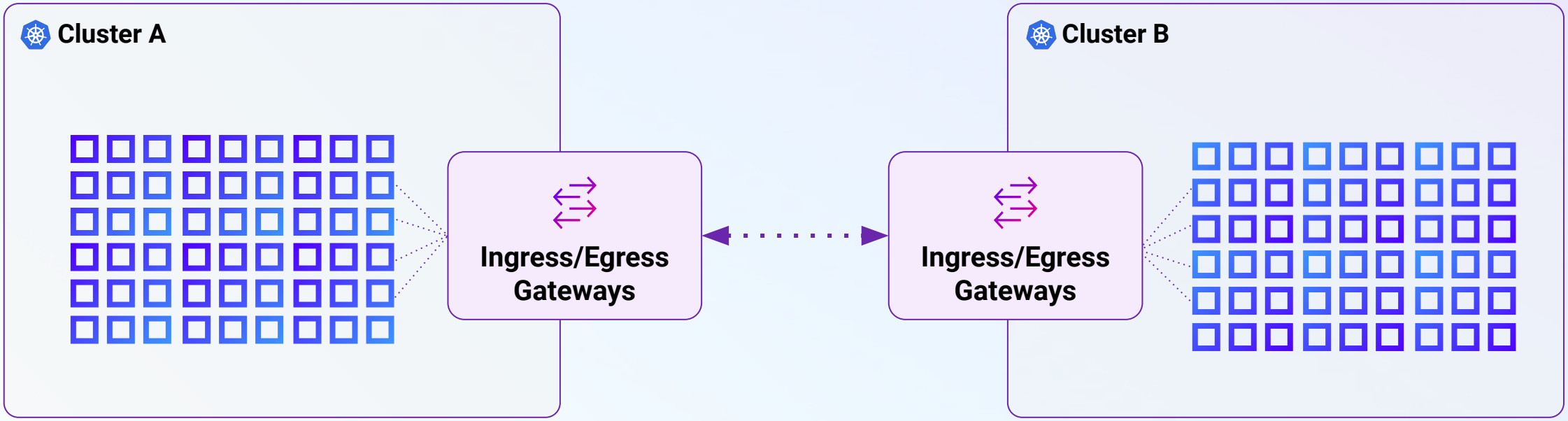
The example below allows all DNS traffic on port 53 to the DNS service and intercepts it via the DNS Proxy. If using a non-standard DNS port for a DNS application behind a Kubernetes service, the port must match the backend port. When the application makes a request for my-remote-service.com, Cilium 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
docs.cilium.io/en/v1.16/network/egress-gateway/egress-gateway

Common limitations

Our ideal world

Cluster A

Cluster B

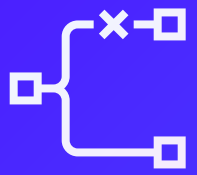
```yaml
---
apiVersion: cilium.io/v2
kind: CiliumNetworkPolicy
metadata:
  name: backend-policy
  namespace: backends
spec:
  endpointSelector:
    matchLabels:
      role: backend

  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 Mesh

Cluster Mesh

Seamless Pod IP Routing Across Clusters

Transparent Service Discovery

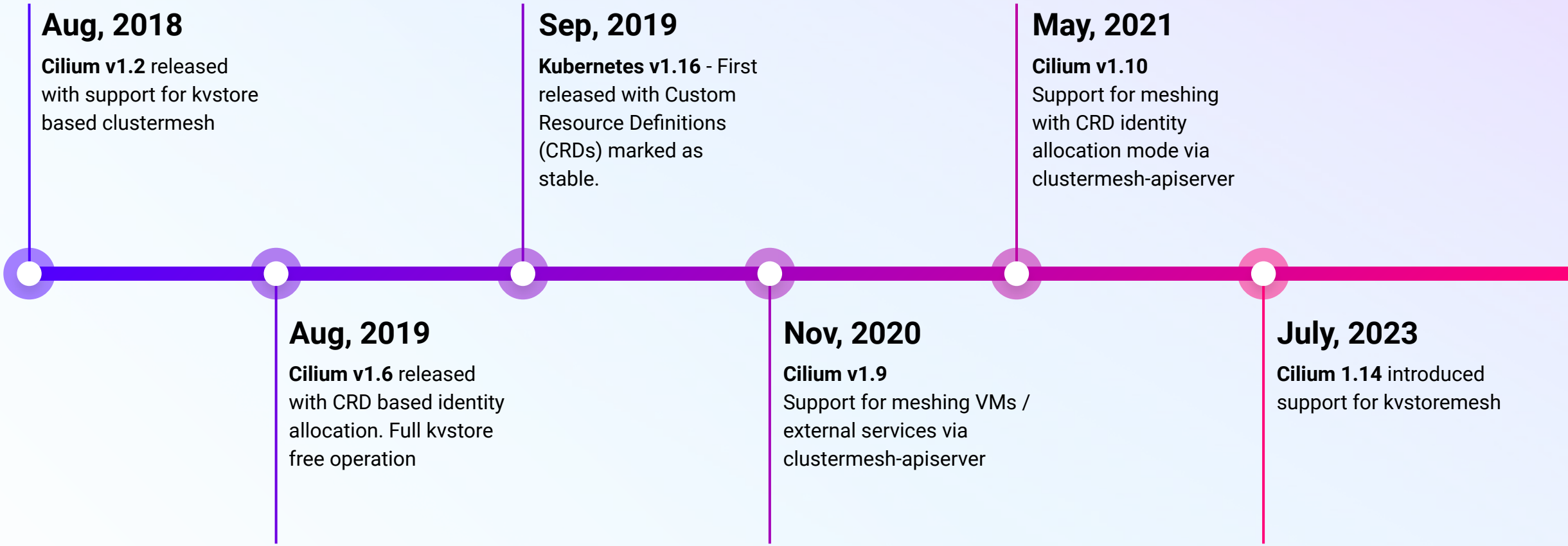


High Availability and Fault Tolerance

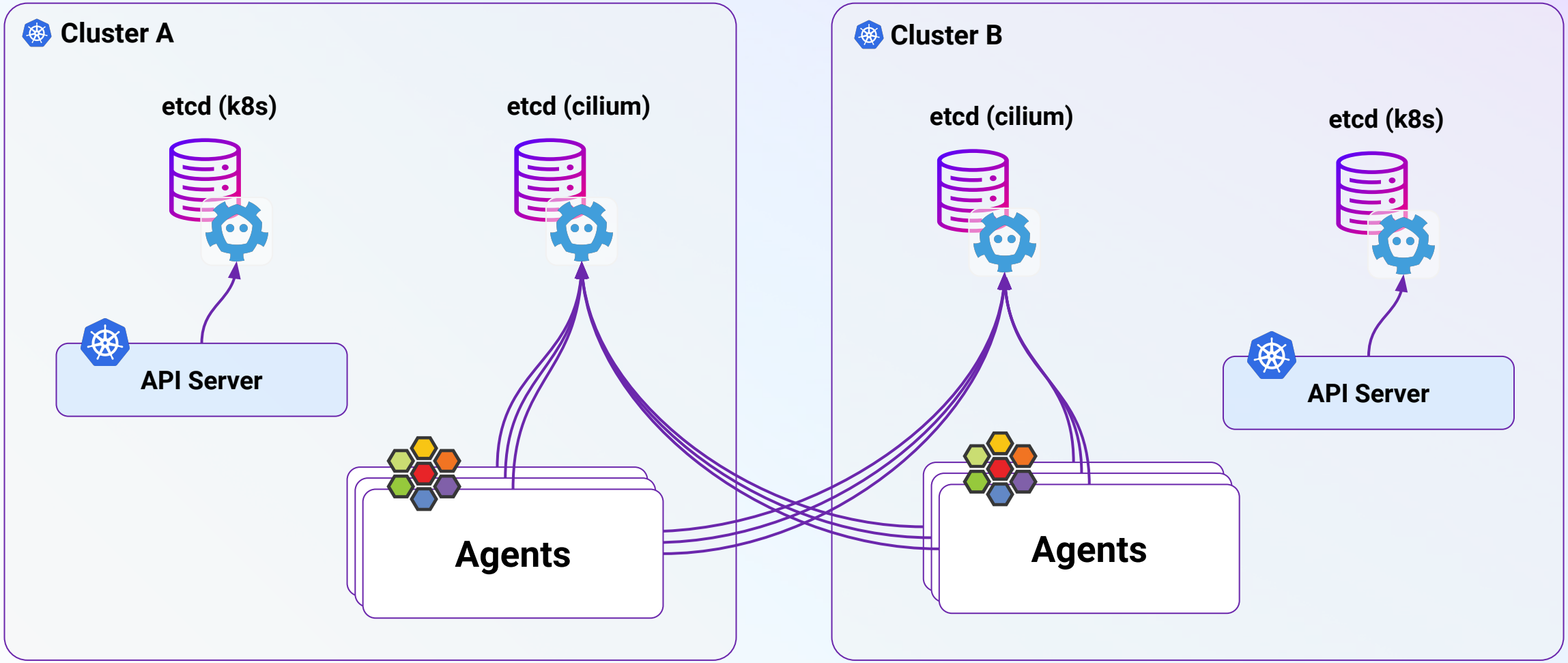
Uniform Network Policy Enforcement

03 Evolution of Meshing

Evolution of Meshing



ClusterMesh



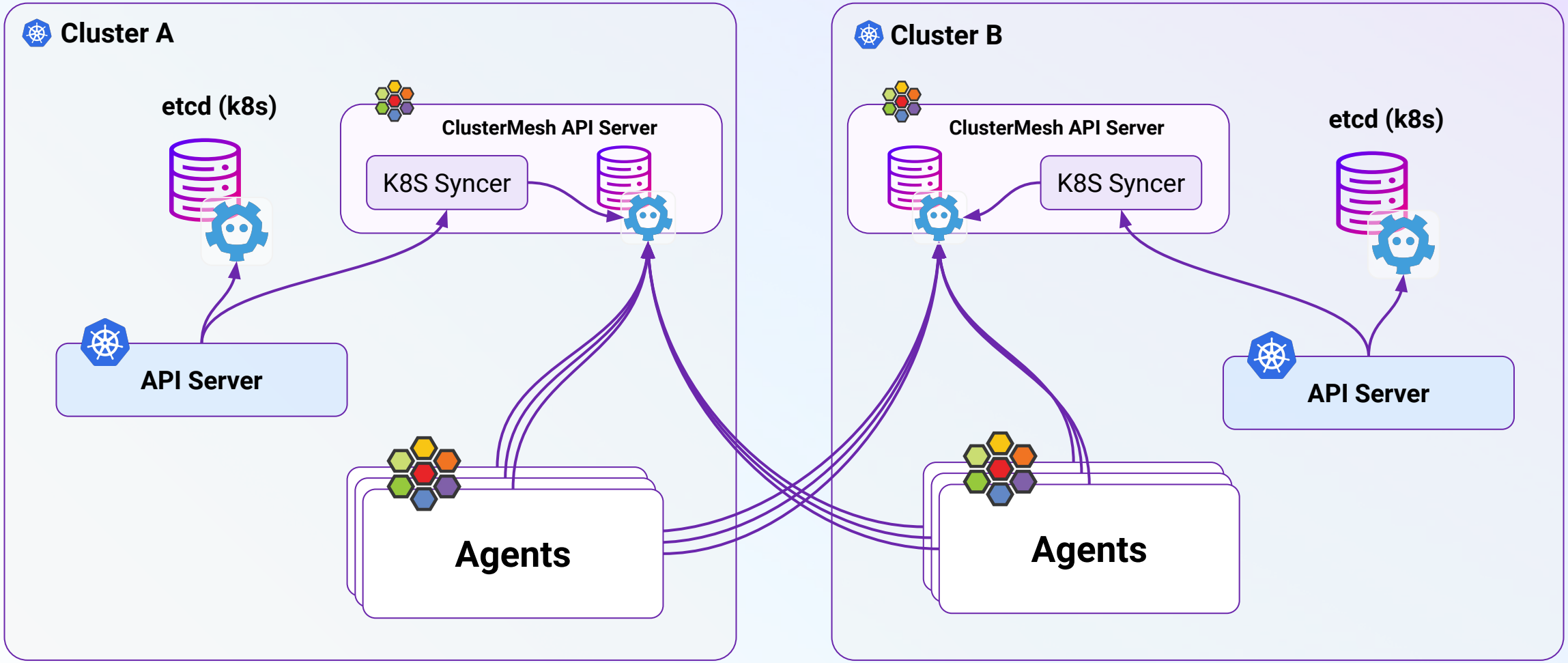
Requires using an External Key-Value Store implementation

m*n updates

m = #clusters in mesh

n = #agents per cluster

ClusterMesh + ClusterMesh API Server

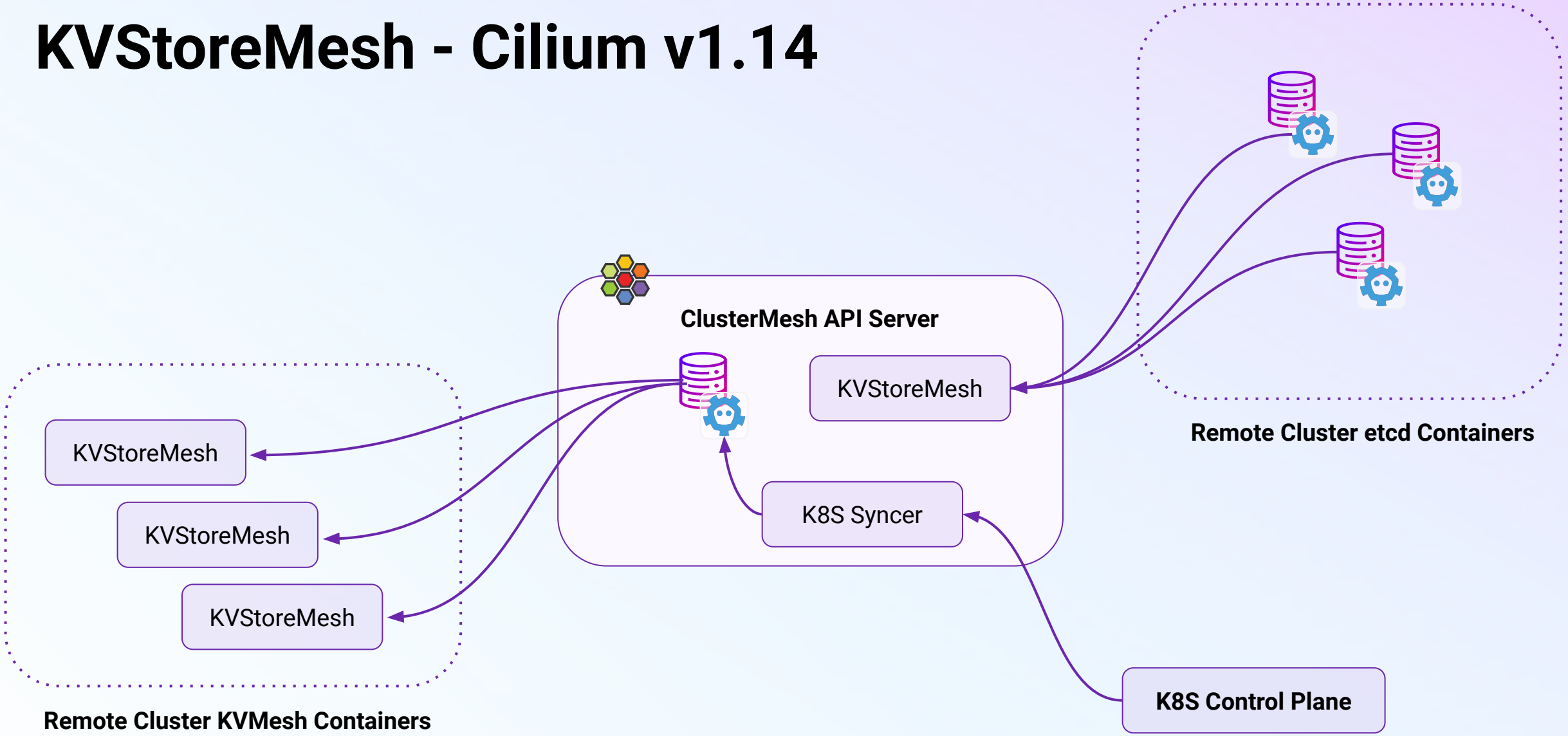


Arthur Chiao

eBPF Summit 2022

youtube.com/watch?v=NIsU4l950l4

KVStoreMesh - Cilium v1.14



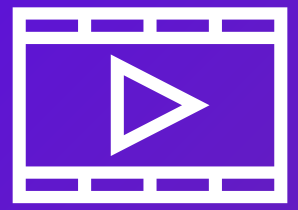
m+n updates

m = #clusters in mesh

n = #agents per cluster

Ryan Drew

CiliumCon NA 2023



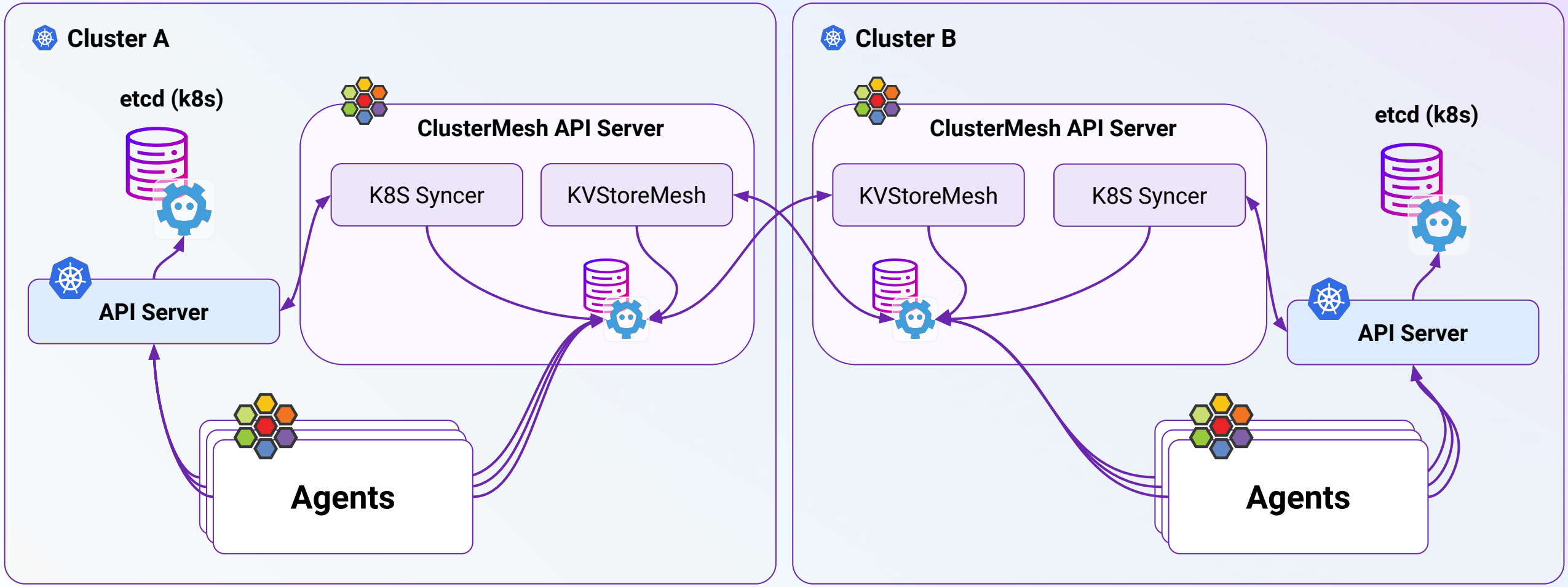
CiliumCon NORTH AMERICA

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

Ryan Drew, Isovalent

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
```

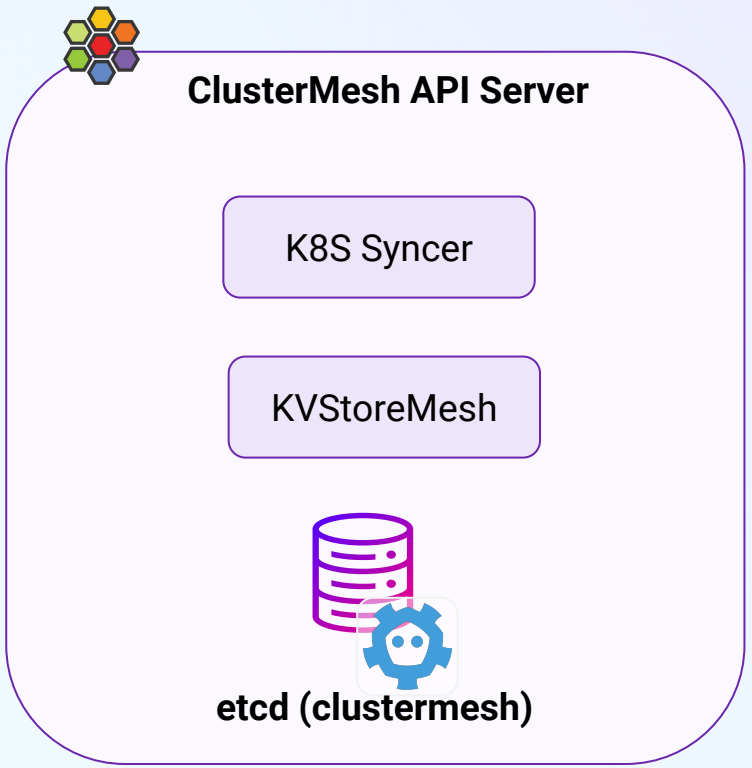
Key	Value

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;

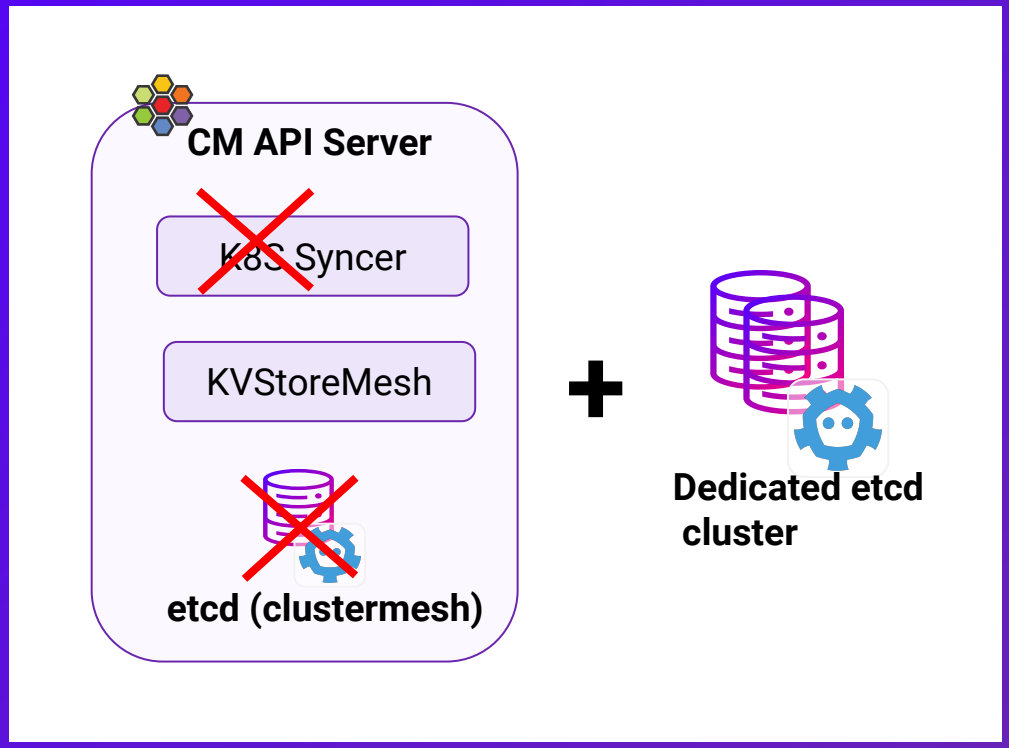
cilium/cache/ip/v1/cluster-a/10.1.1.53	{ "IP": "10.1.1.53", "Mask": null, "HostIP": "192.168.16.2", "ID": 91392, "Key": 0, "Metadata": "", "K8sNamespace": "backends", "K8sPodName": "backend-6b8d95757c-wmsxg" }

04 KV KV Mesh?

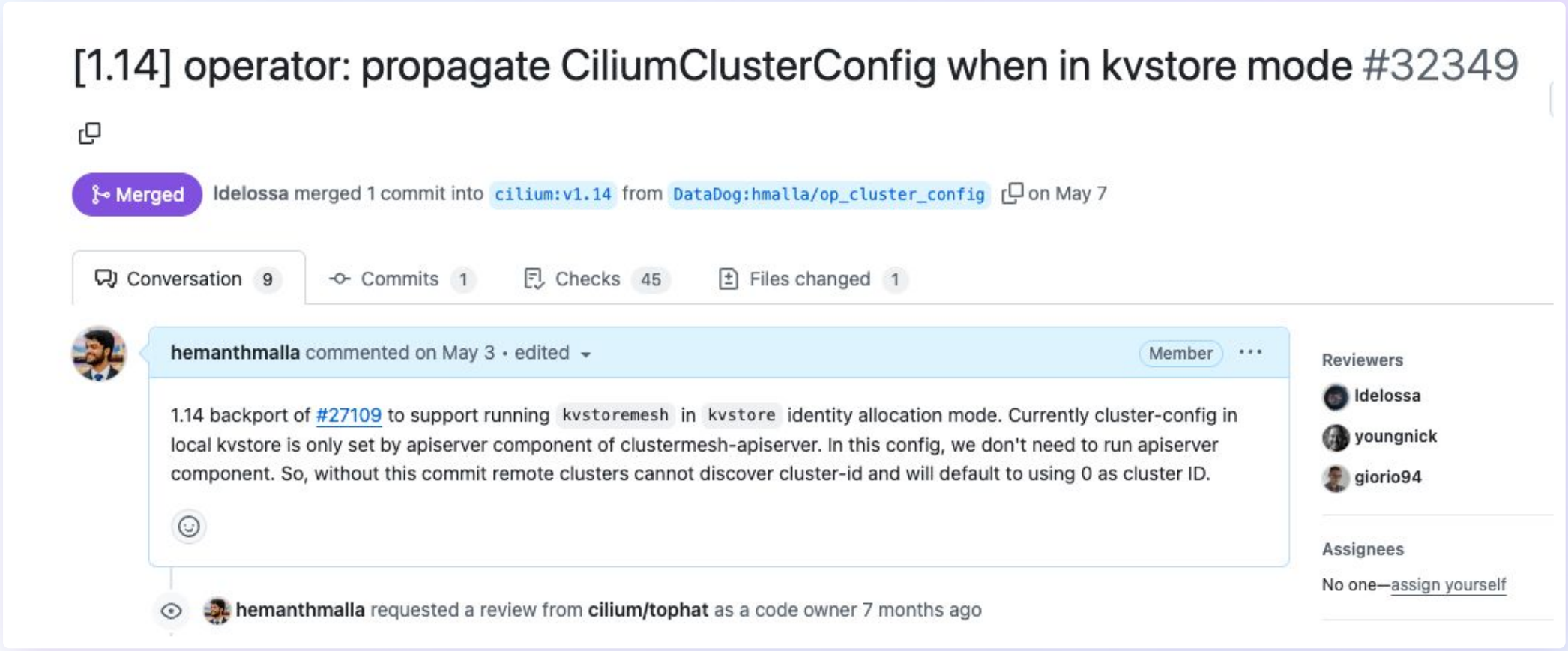
KVStoreMesh



KVStoreMesh + KVStore Identity Mode



Last Piece of the puzzle 🧩



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

Merged ldelossa 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.

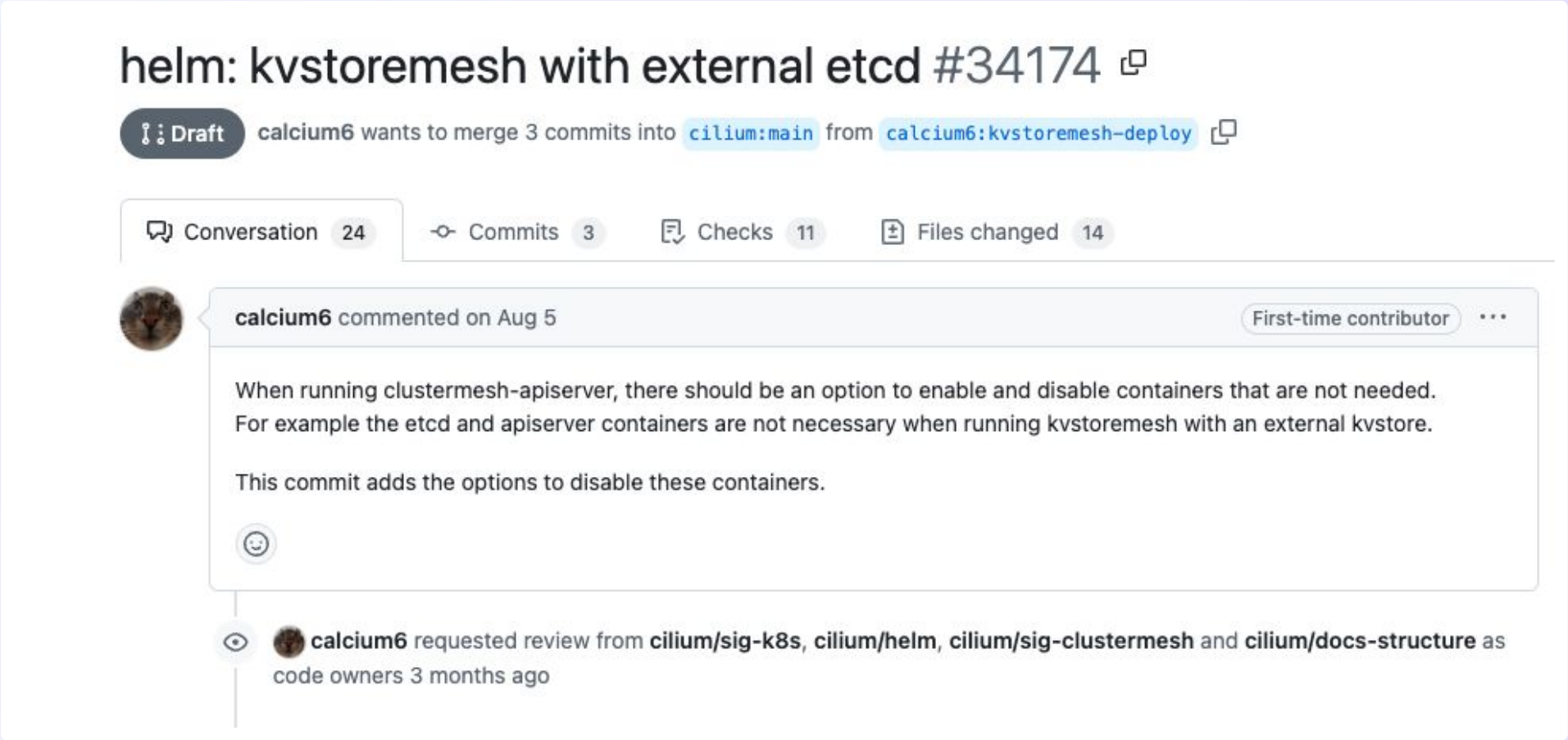
hemanthmalla requested a review from **cilium/tophat** as a code owner 7 months ago

Reviewers
ldelossa
youngnick
giorio94

Assignees
No one—assign yourself

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

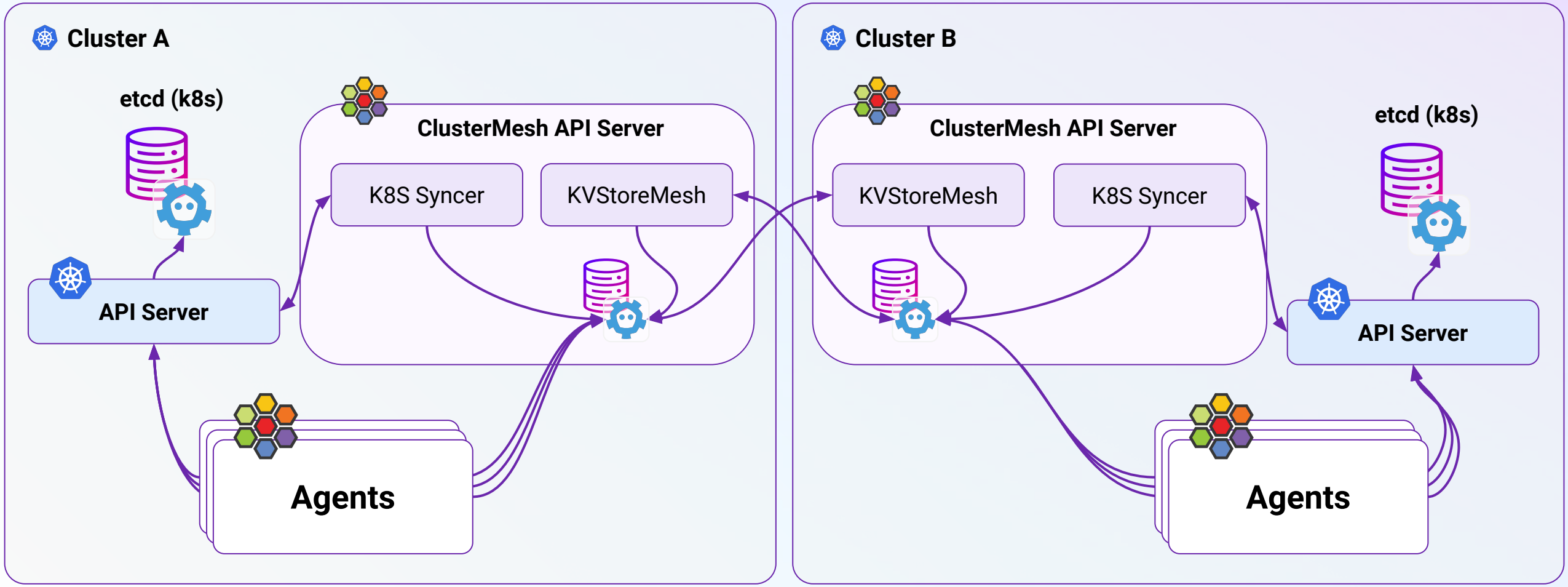
When running clustermesh-apiserver, there should be an option to enable and disable containers that are not needed.
For example the etcd and apiserver containers are not necessary when running kvstoremesh with an external kvstore.

This commit adds the options to disable these containers.

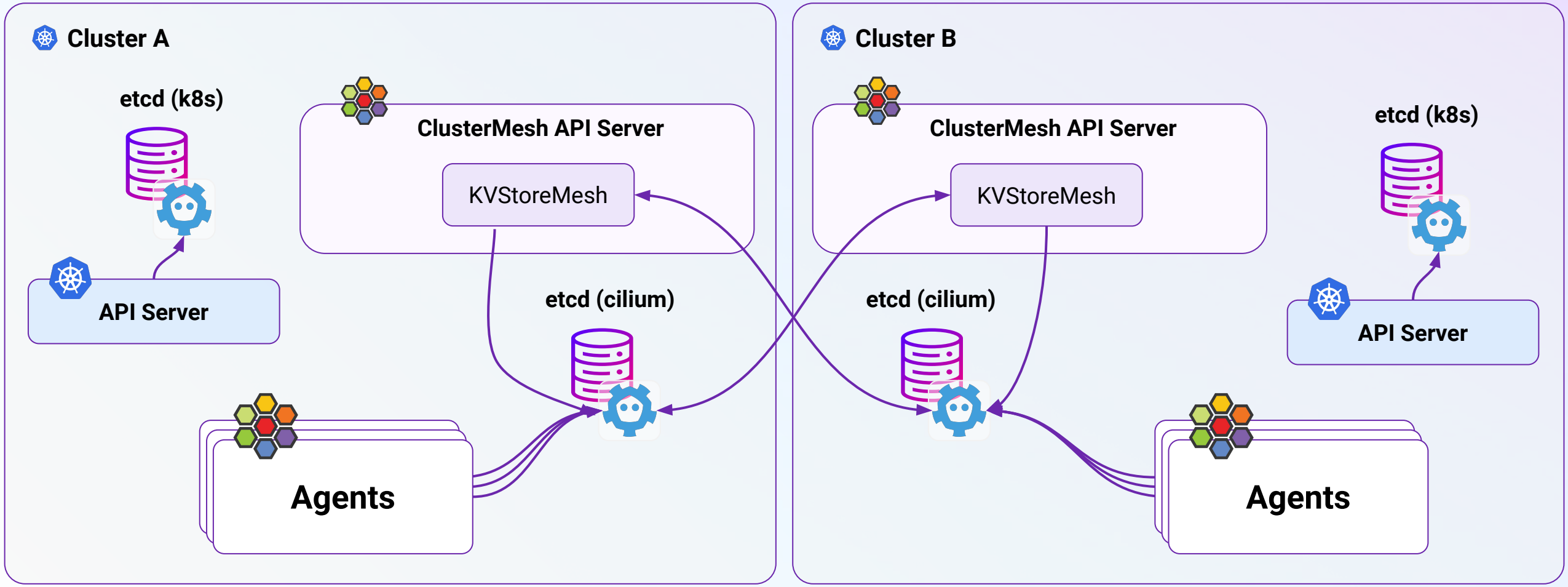
calcium6 requested review from **cilium/sig-k8s**, **cilium/helm**, **cilium/sig-clustermesh** and **cilium/docs-structure** as code owners 3 months ago

github.com/cilium/cilium/pull/34174

KVStoreMesh



KV KV Mesh


Cluster A
etcd (k8s)
ClusterMesh API Server
KVStoreMesh
API Server
etcd (cilium)
Agents
Cluster B
ClusterMesh API Server
KVStoreMesh
etcd (k8s)
API Server
etcd (cilium)
Agents

Mesh Modes

ClusterMesh + Clustermesh API Server

KVStoreMesh

default since v1.16

Custom Resource Definitions

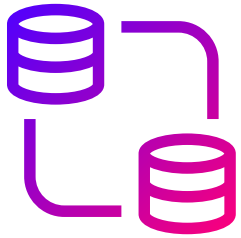
Original ClusterMesh

KVKVMesh?

External Key-Value Store

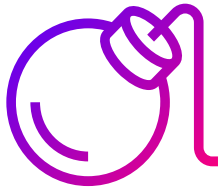
05 Migration, Monitoring and Best Practices

Moving Parts 🚚



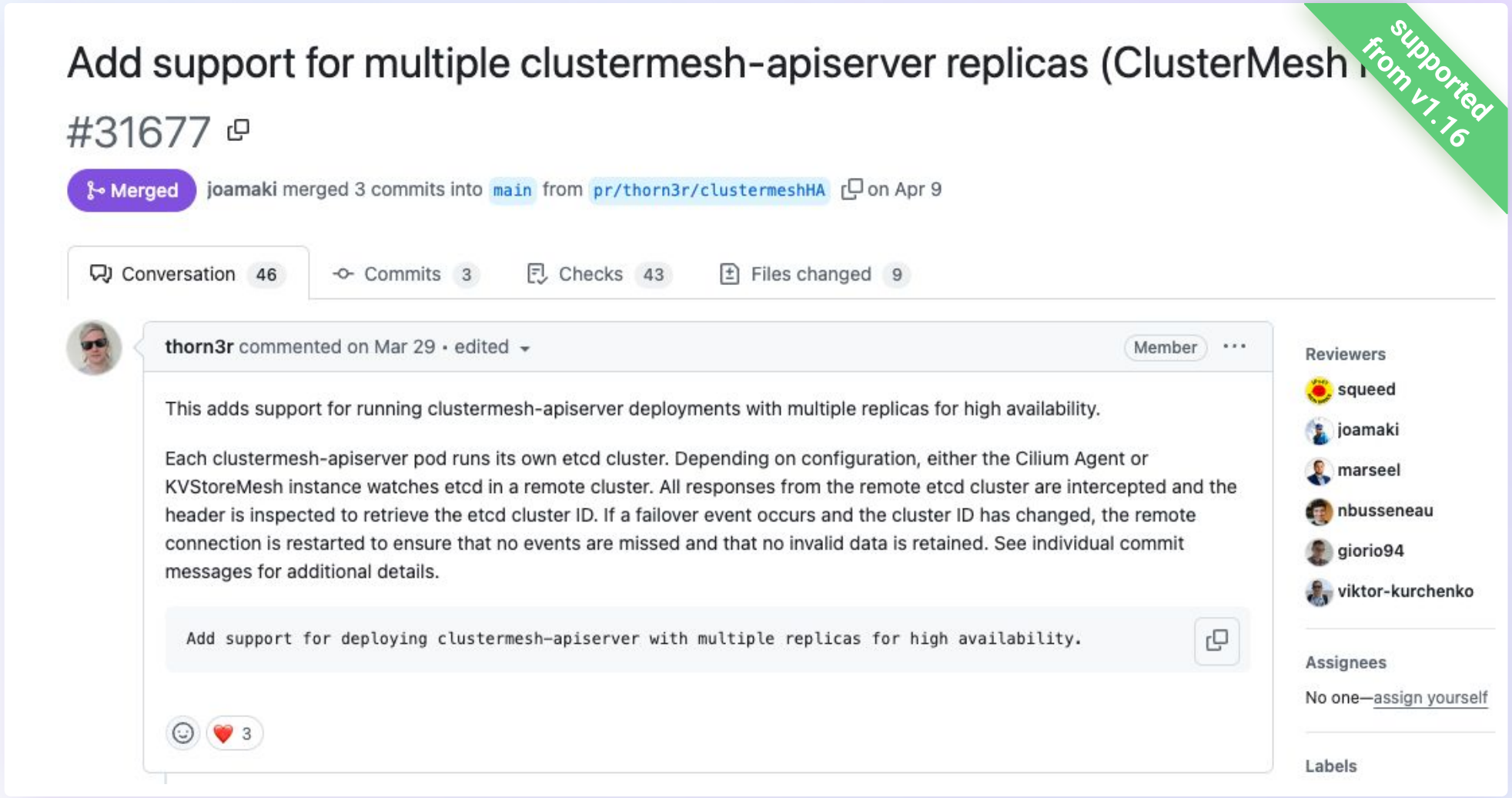
What is getting replicated?

Shuffling too much information too fast?



What happens when kvstoremesh dies?

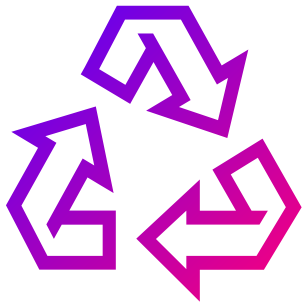
HA Mode



github.com/cilium/cilium/pull/31677

Primer on etcd storage / garbage collection

Local Keys



Classic Garbage Collection

Remote Keys

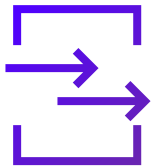
Leases with TTLs

Safely meshing with existing clusters

Key-Value leases TTL tuning

KVStore QPS

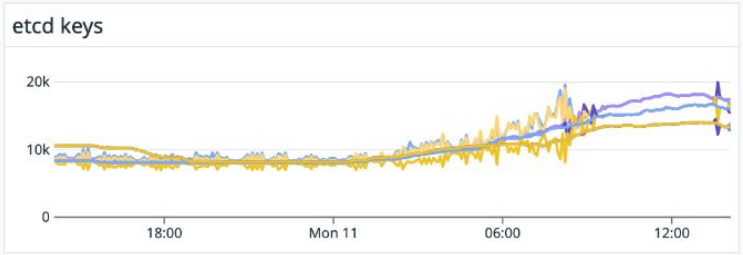
Bootstrap + Regular

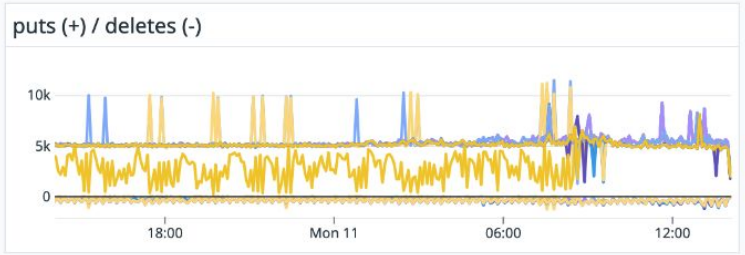


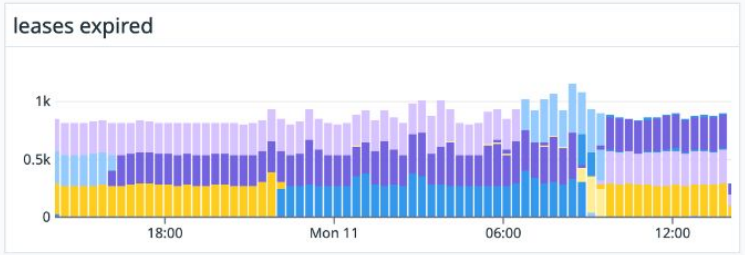
Unidirectional Meshing

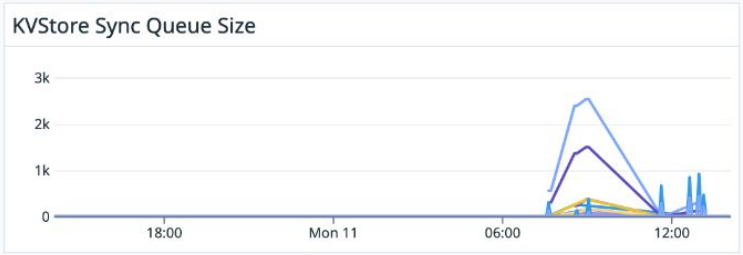
World -> Known Identity

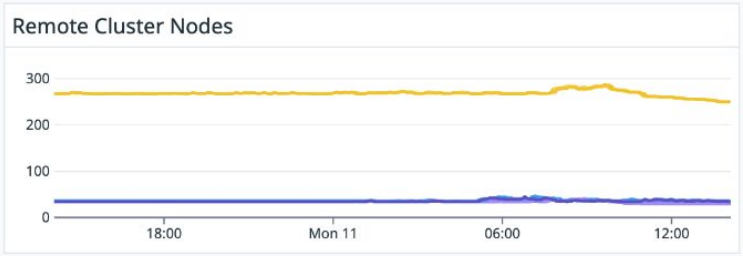
Metrics to watch

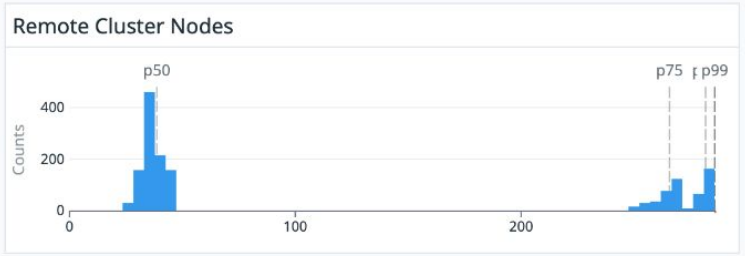


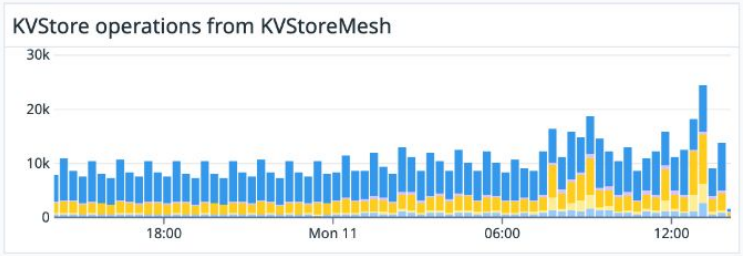


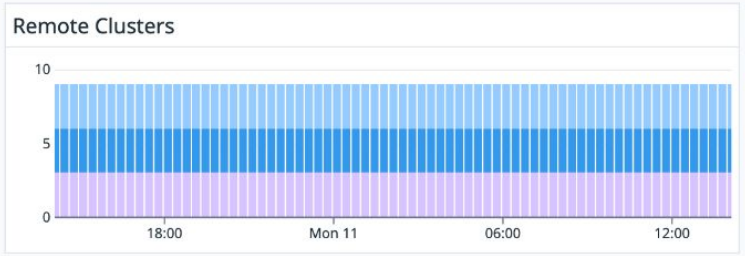


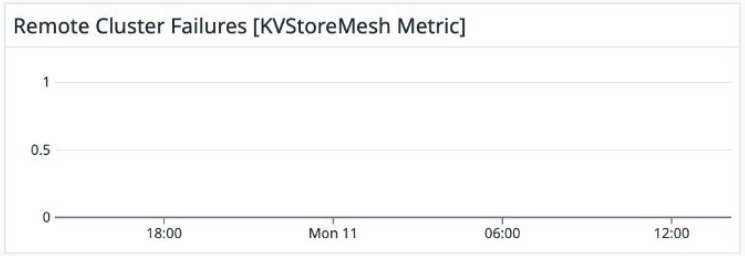




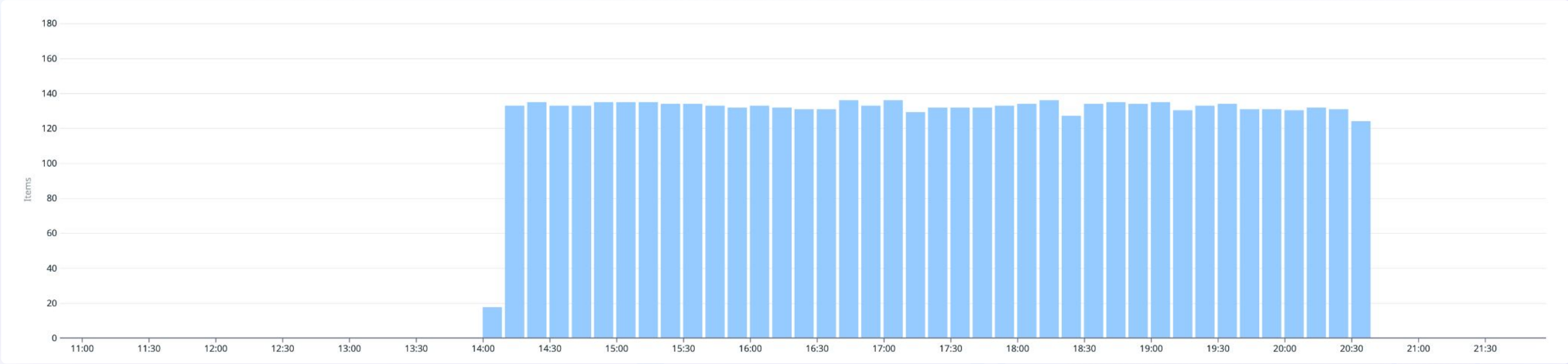






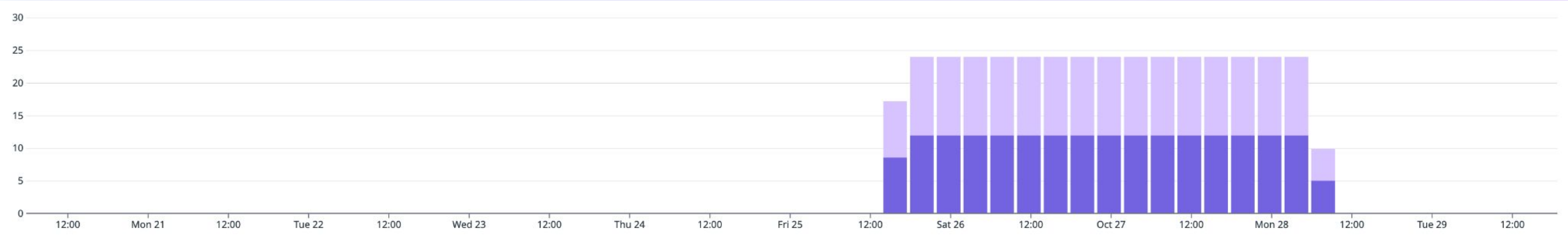


etcd MVCC lease expirations

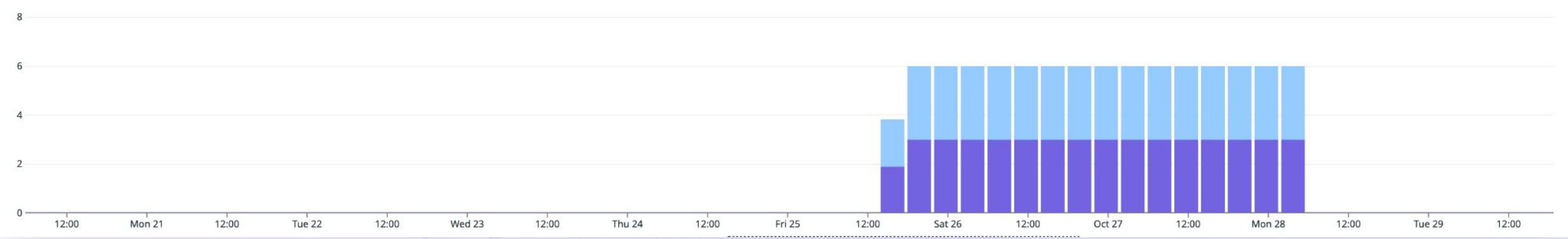


sum:etcd.debugging.server.lease.expired.total{} by {member,cluster}.as_count()

ClusterMesh & KVStoreMesh failures

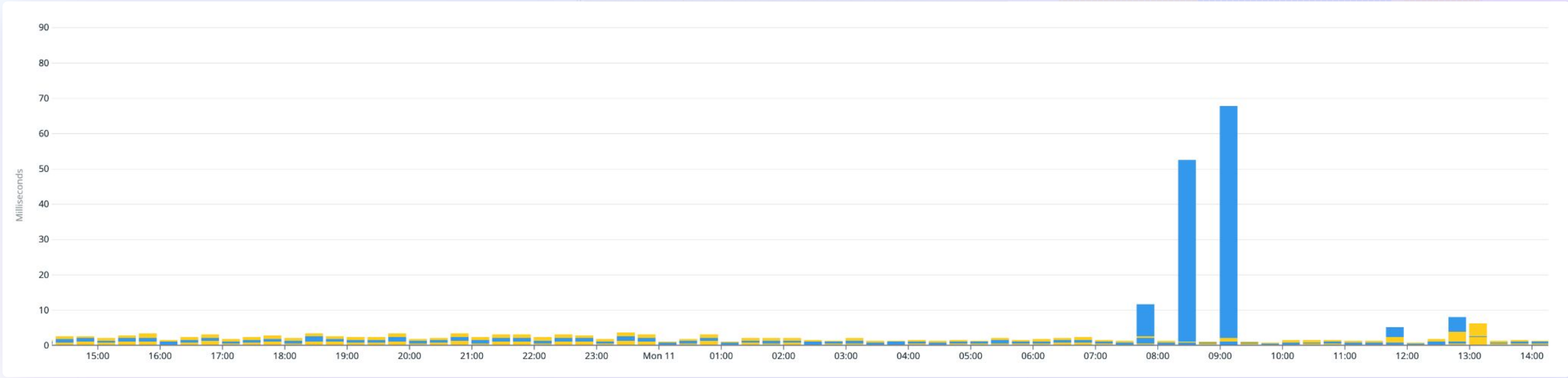


sum:cilium.cilium_clustermesh_remote_cluster_failures{} by {source_cluster,target_cluster}



sum:cilium.cilium_kvstoremesh_remote_cluster_failures{} by {source_cluster,target_cluster}

etcd operations and rate limiting



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
    ℹ️  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
  ℹ️  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/

@mvisonneau
@hemanthmalla

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

Feedback

sched.co/1izs0

DATADOG