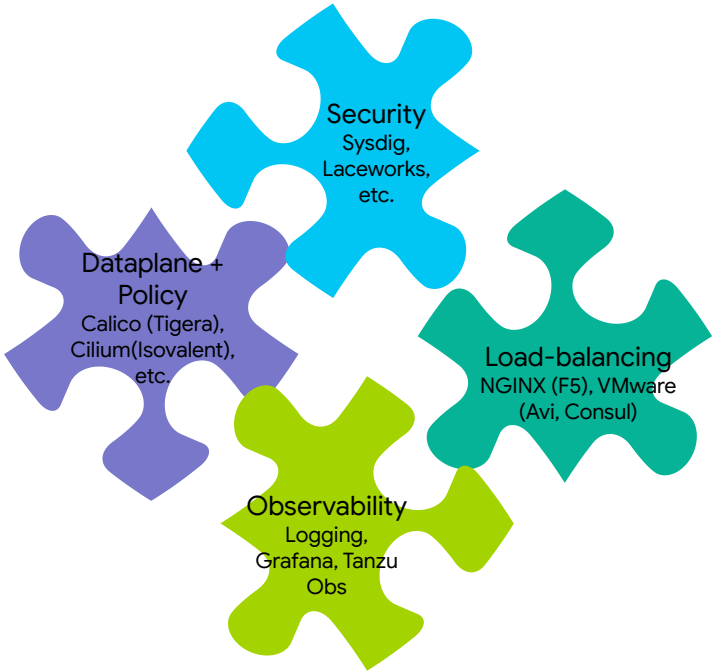
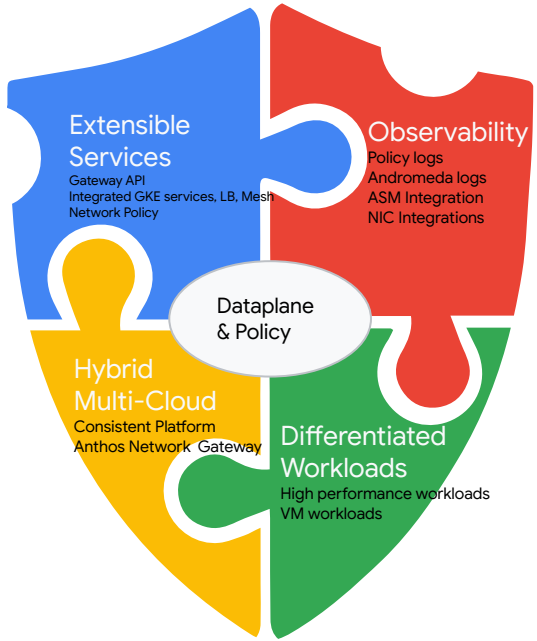
GKE & Anthos Networking

Industry Landscape and Value

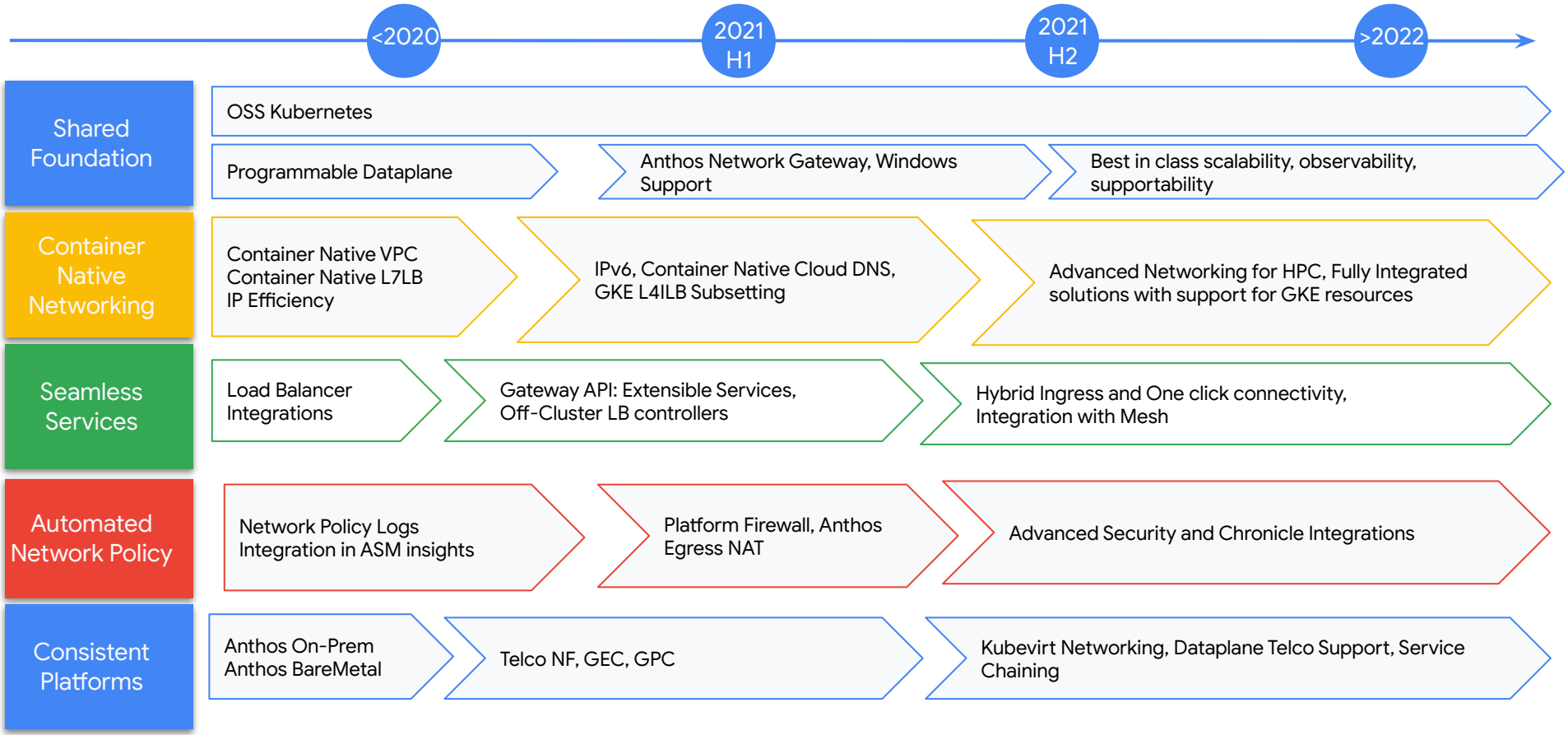
K8S Landscape: Fragmented Experience



GKE Networking: Integrated and Differentiated



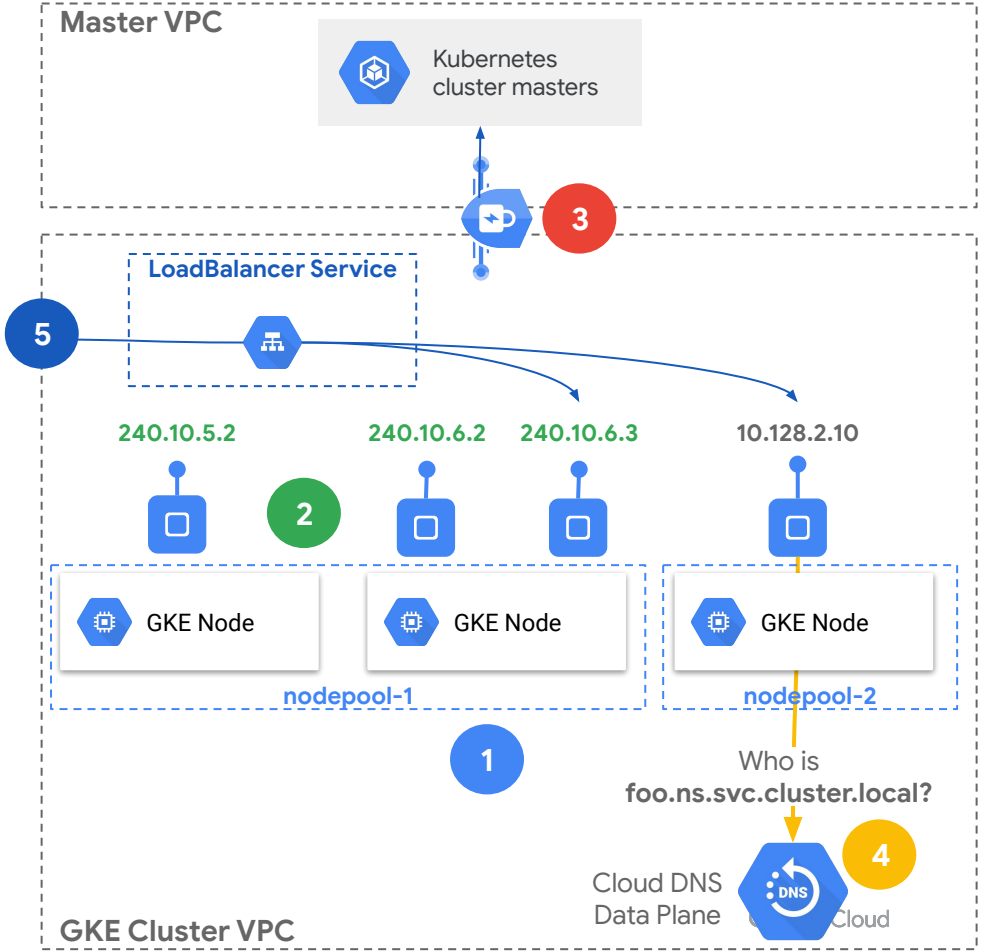
Multi Year Strategy, Grounded in Customer Focus



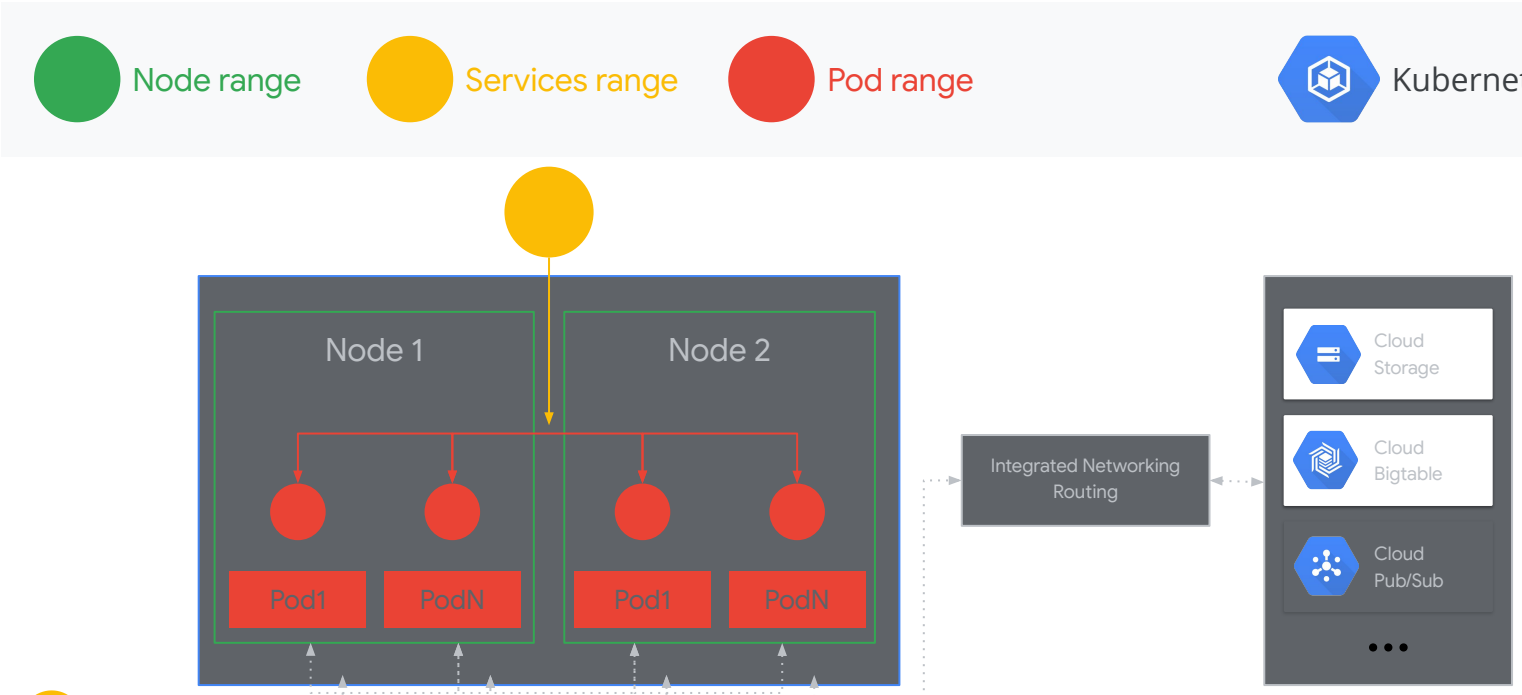
GKE Networking

Google Network Orchestration with GKE

1. Multi-pod CIDR
2. Non-RFC 1918
3. Private clusters 2.0
4. Cloud DNS for GKE
5. IPv6



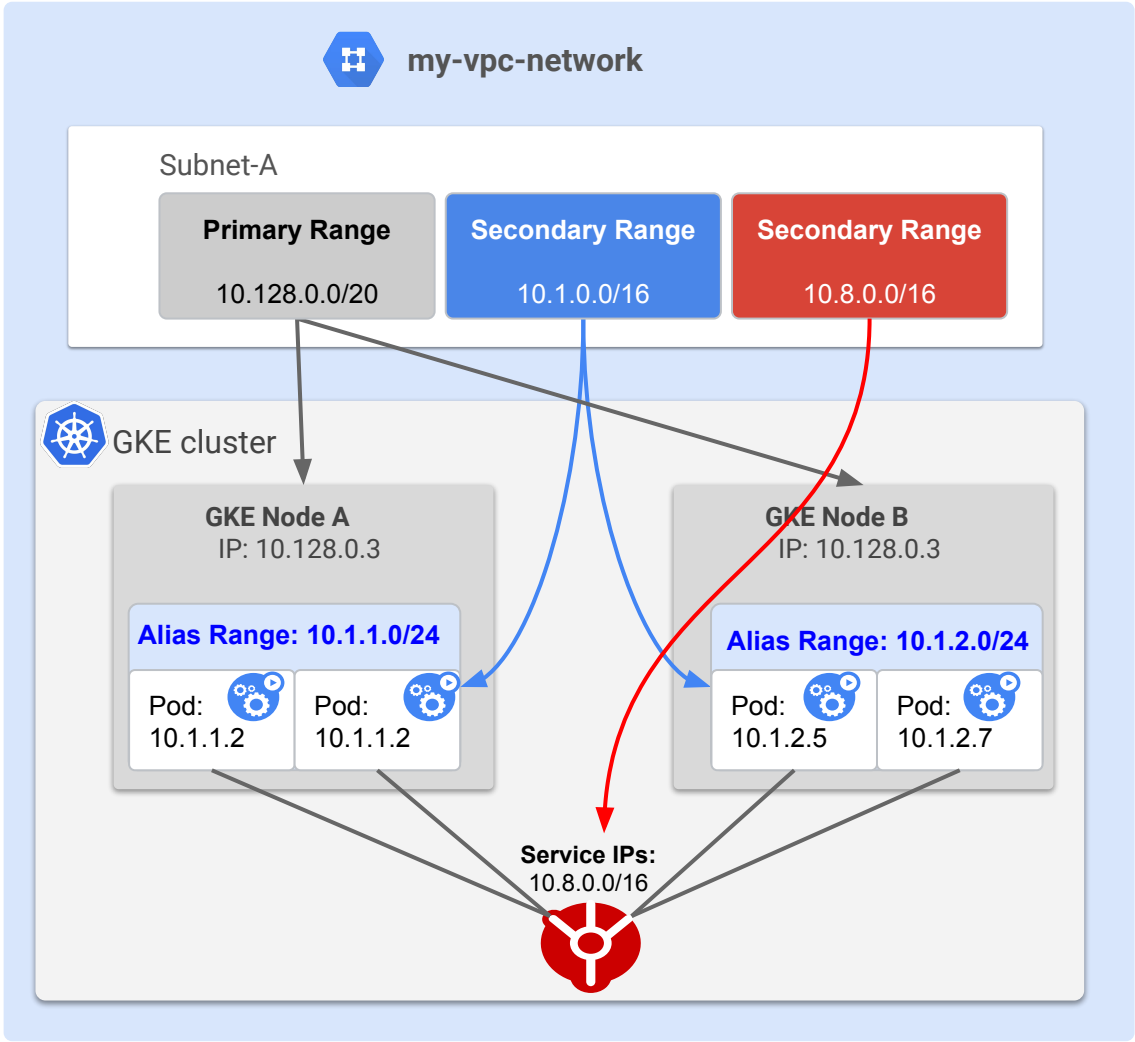
Integrated IP allocation



VPC-Native Clusters

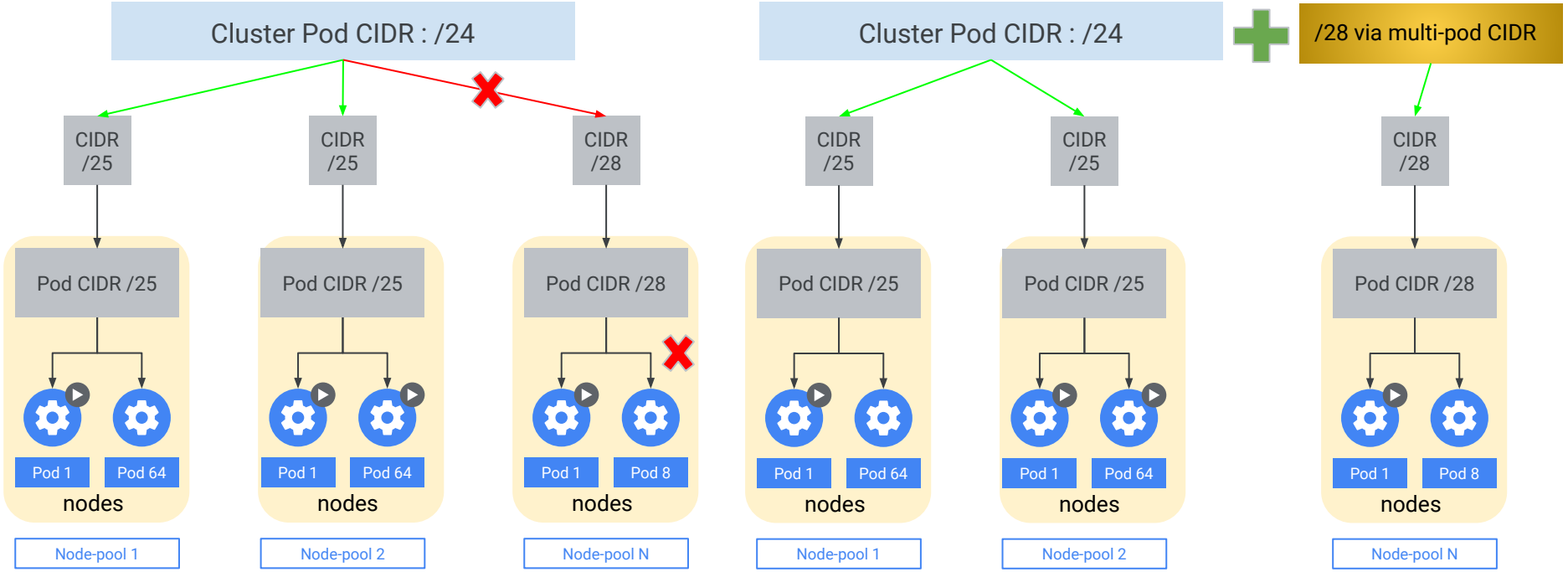
Benefits

- **Higher Scale** thanks to less routes
- **Flat Network** between Pod IPs and any IP in the VPC
- **Direct access to GCP services** such as
  - Private access to Google services like GCS, BigQuery
  - Private clusters, shared-VPC,
  - IP management, flex CIDR etc.



Google Cloud

Discontiguous Multi-Pod CIDR



- Node-Pool N doesn't come up as we are out of Pod IPs.
- Users have to delete and recreate Cluster

- A new CIDR is added & the new Node pool references this to come up
- Users don't have to delete and recreate; cluster size can change dynamically

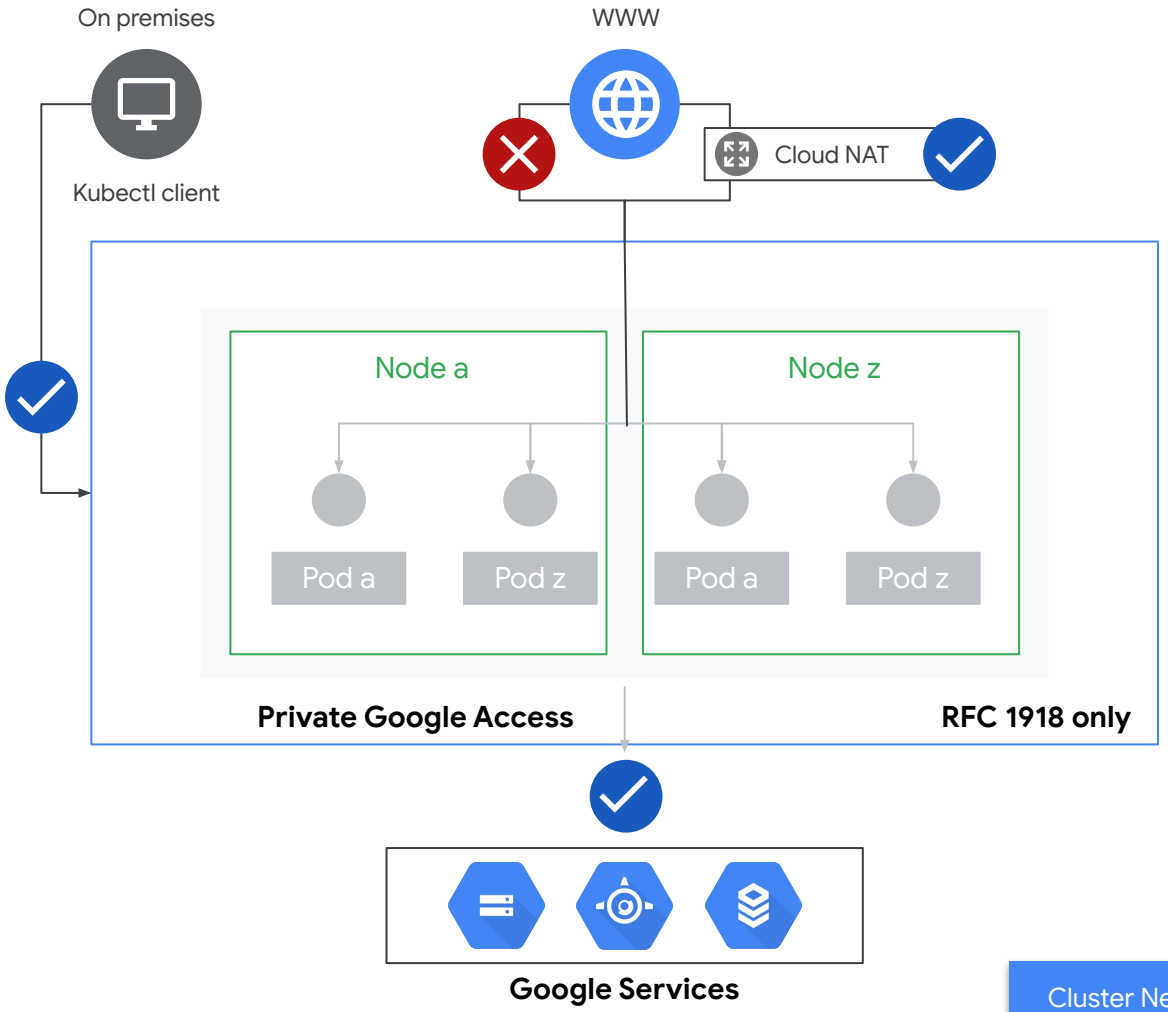
Google Cloud

Non-RFC1918 Addresses

- Optimizations have a threshold and sometimes customers are just out of RFC 1918
- New block of IPs to replenish IP inventory

- GKE will support new Reserved ranges that can be used similar to RFC1918 Private Addresses
  - Reserved RFC 6598 address - 100.64.0.0/10 aka CGN addresses
  - IANA IPv4 Special Purpose Address Registry [RFC5736] - 192.0.0.0/24
  - TEST-NET-1 [RFC5737] - 192.0.2.0/24
  - 6to4 Relay Anycast [RFC3068] - 192.88.99.0/24
  - Network Interconnect Device Benchmark Testing [RFC2544] - 198.18.0.0/15
  - TEST-NET-2 [RFC5737] - 198.51.100.0/24
  - TEST-NET-3 [RFC5737] - 203.0.113.0/24
  - Reserved Class E address - 240.0.0.0/4
    - Applies to both Private and Public Clusters
    - Caveats : On-Prem Networking devices like Routers and Switches don't have full support, Windows OS doesn't support Class E.
- GKE also supports **P**rivately **U**sed **P**ublic **I**Ps (PUPI). This means any non-Google IP can be used as a Private IP in your Clusters
  - Applies only to GKE Private Clusters, Windows OS currently doesn't work with PUPI

Private clusters



Google Cloud

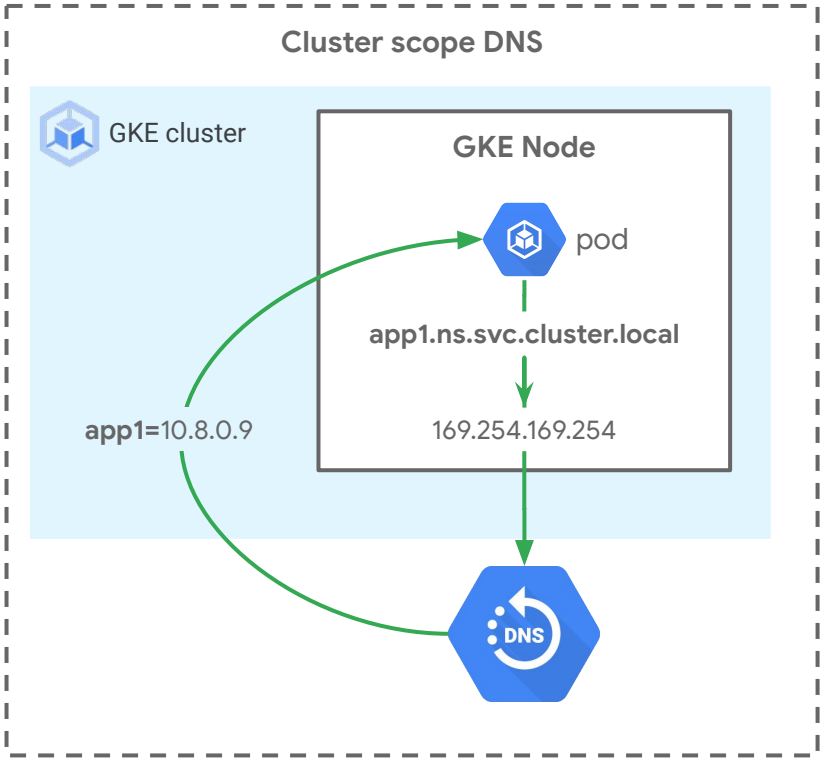
Public Preview

Container-native Cloud DNS

Cloud DNS for GKE

The full power of Cloud DNS, natively integrated for GKE

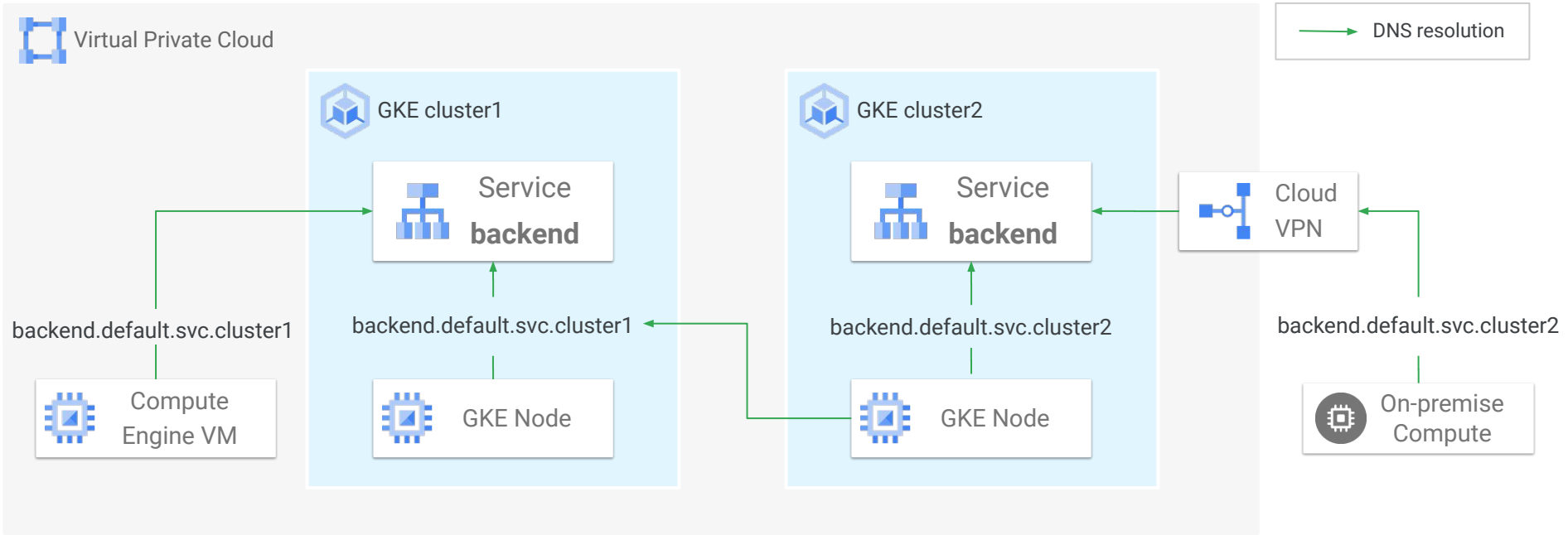
- Scales to millions of DNS QPS
- Local DNS resolution on every node
- K8s Service and Pod resolution outside of the cluster
- DNS query logging and monitoring
- Globally distributed, Google-managed DNS
- User guide



**Cloud DNS for GKE only supported for GKE on GCP clusters

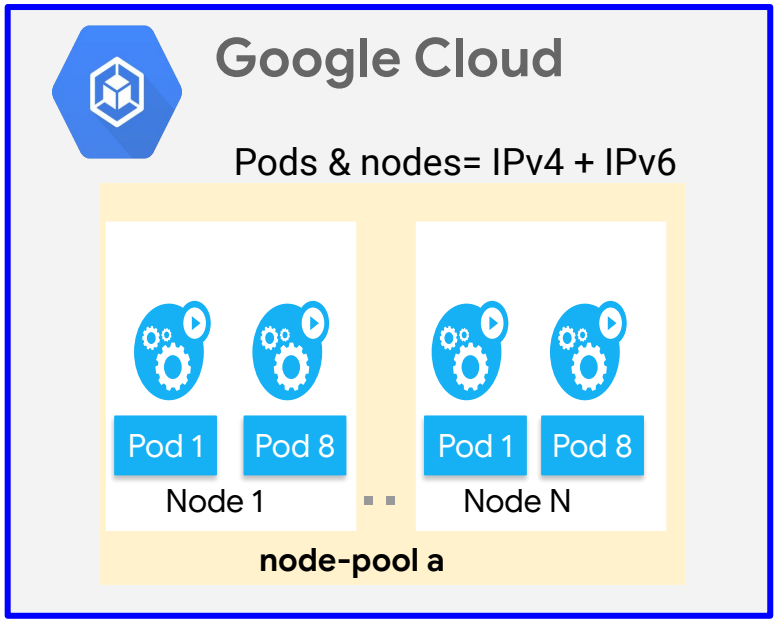
Public Preview

VPC Scope DNS



IPv6 Dual Stack Use Cases - GCP

1. Connectivity to an external IPv6 endpoint over the internet
   a. Security/Scanning purposes
   b. Participation in a peer-to-peer network requiring IPv6
2. IPv4 address exhaustion
3. Performance

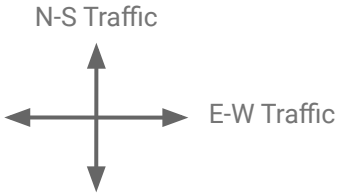


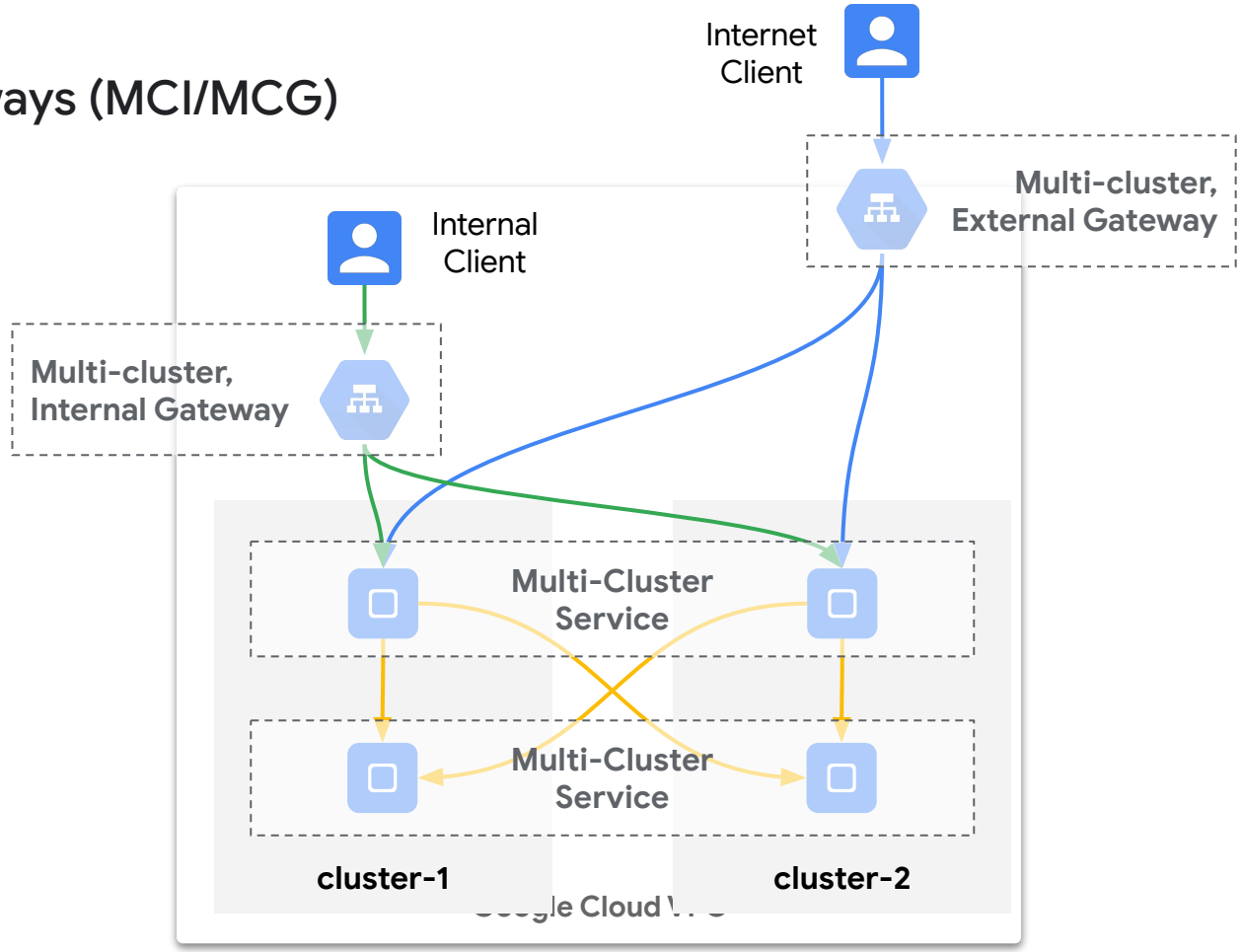
Request:

1) If your customer has interest in GKE + IPv6, please fill this form.
2) Please add your customer info to this tracker.

Multi-cluster Services & Multi-cluster Ingress/Gateways (MCI/MCG)

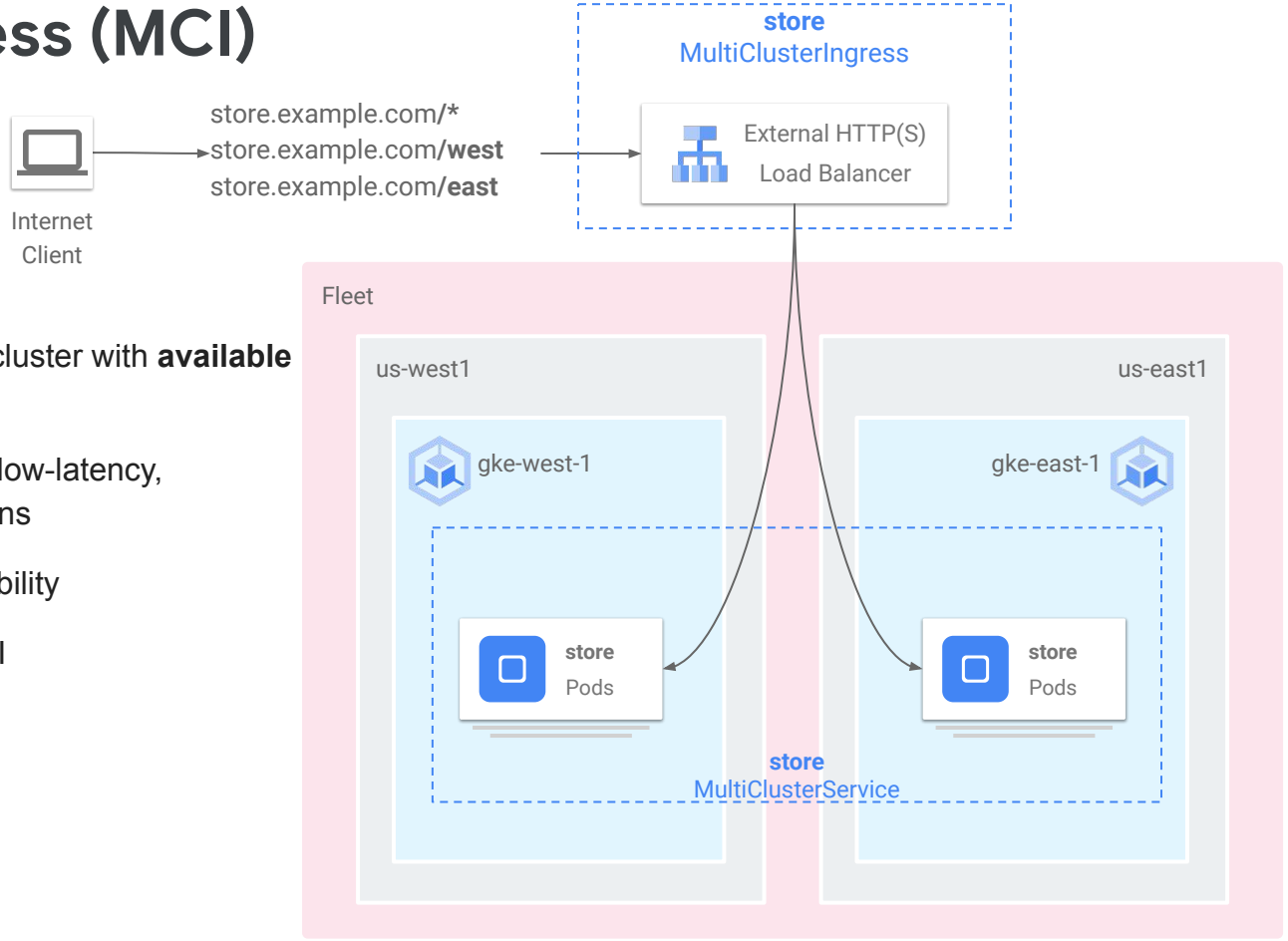
- MCS is for east-west traffic (between clusters)
- MCI is for north-south traffic (from outside the cluster to across many clusters)
- Both are complementary and often used together





Multi-cluster Ingress (MCI)

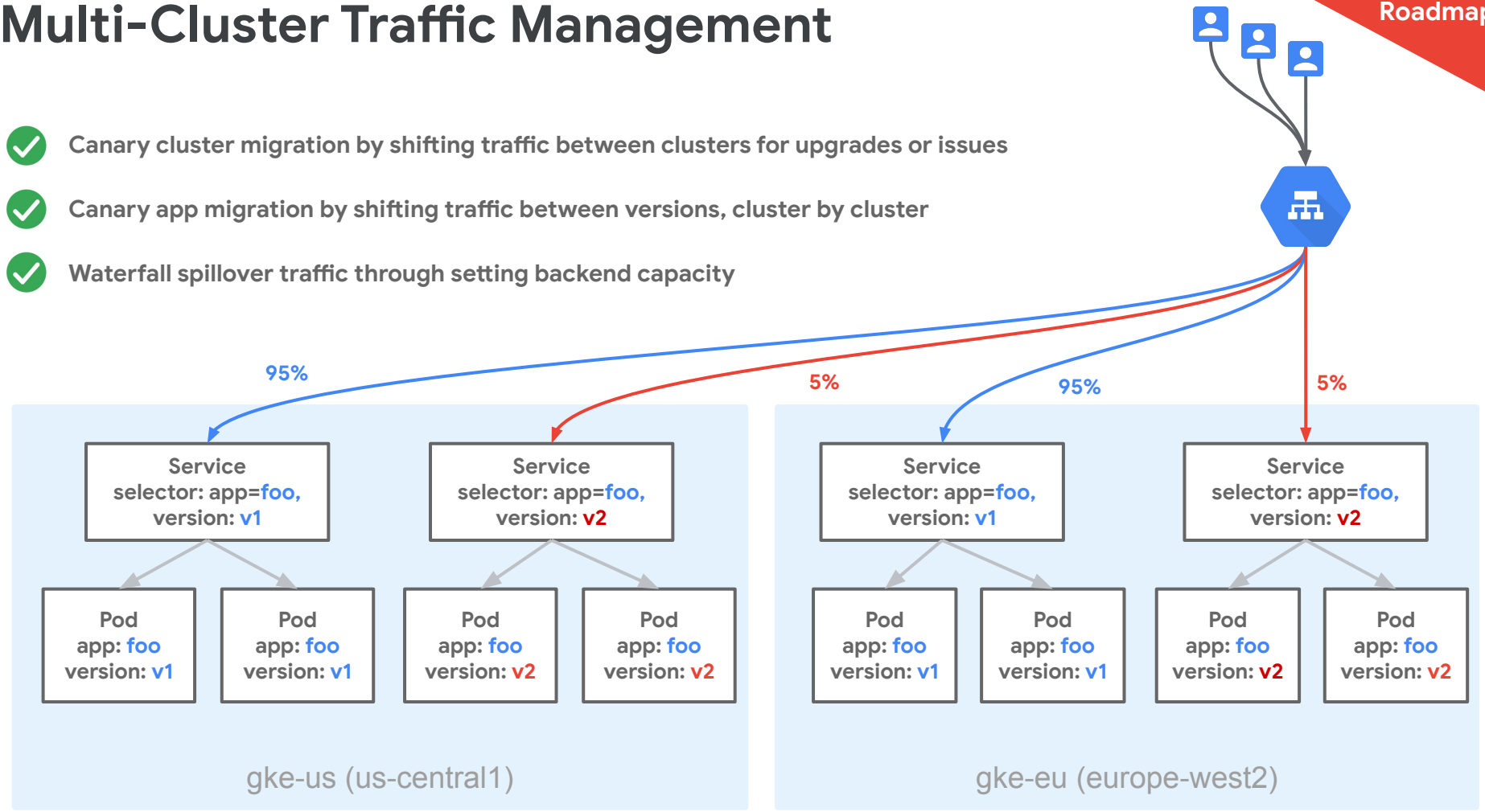
- Routing to **closest, healthy** GKE cluster with **available capacity**
- A single VIP for globally available, low-latency, multi-cluster Kubernetes applications
- Provides multi-regional high availability
- Simple, declarative, K8s-native API



Multi-Cluster Traffic Management

Roadmap

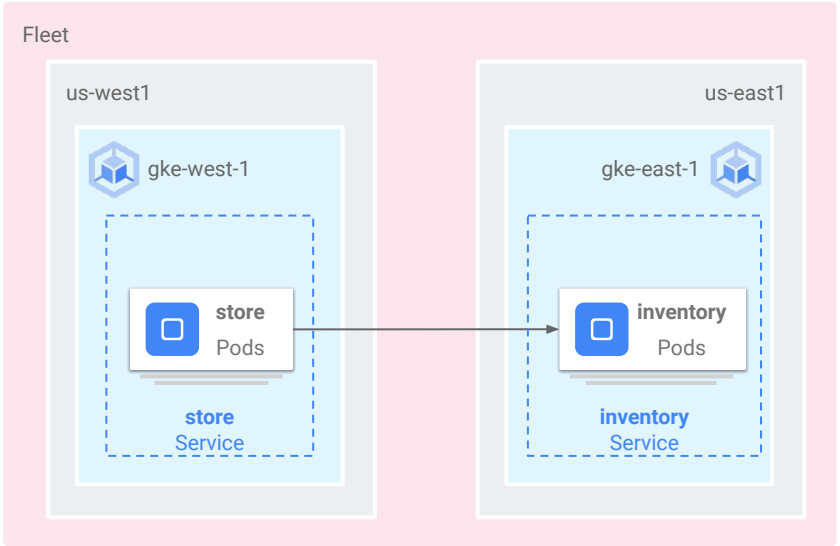
- Canary cluster migration by shifting traffic between clusters for upgrades or issues
- Canary app migration by shifting traffic between versions, cluster by cluster
- Waterfall spillover traffic through setting backend capacity



GKE Multi-cluster Services

- Services can span multiple zones, regions, or GCP projects
- Clusters must reside in the same VPC or in peered VPCs
- Global scale endpoint management powered by Traffic Director
- Implements open source API

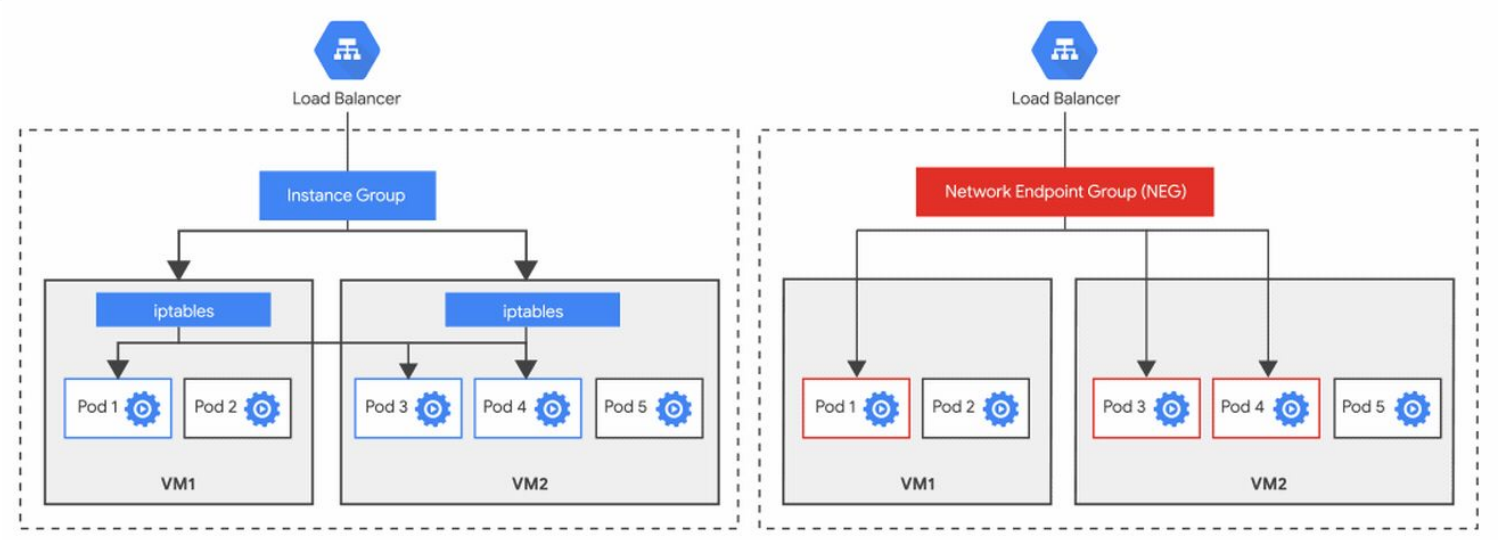
Simple setup and configuration



Container Native Load Balancing

with Network Endpoint Groups (NEGs)

- Uses routable Pod IPs to send traffic from load balancers
- Pod readiness to safely control traffic based on pod **and** network availability
- Direct visibility of every Pod from the load balancer with no intermediate hops
- Simpler traffic path with no NAT from load balancer to Pod

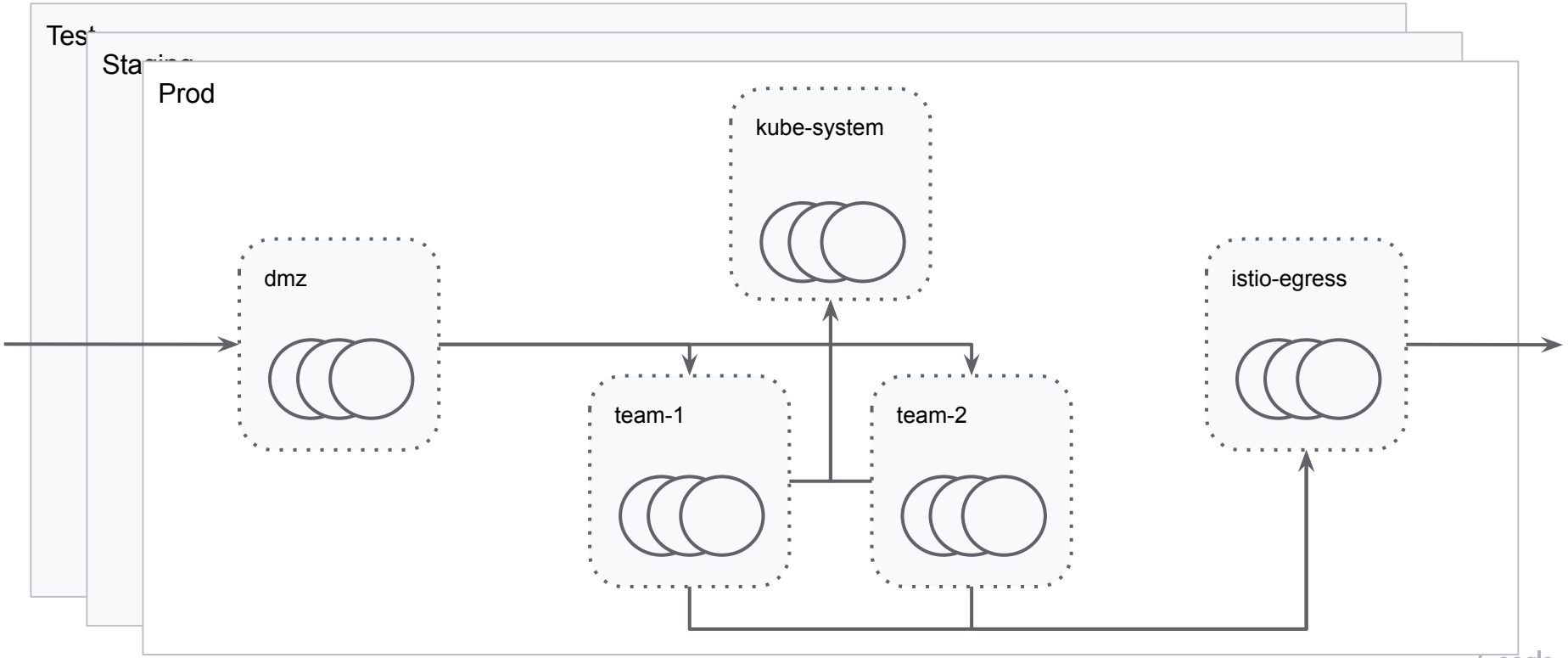


Kubernetes Network Policy

Kubernetes Network Policy is a versatile L3/L4 **firewall** for pods.

- OSS K8S API to write pod-level firewalls to achieve microsegmentation
- A key requirement for security-conscious enterprise customers
- Used in conjunction with VPC firewalls and Istio security policies
- Optional on GKE on GCP, always on for Anthos on-prem
- Automatically enabled with **GKE Dataplane V2** clusters

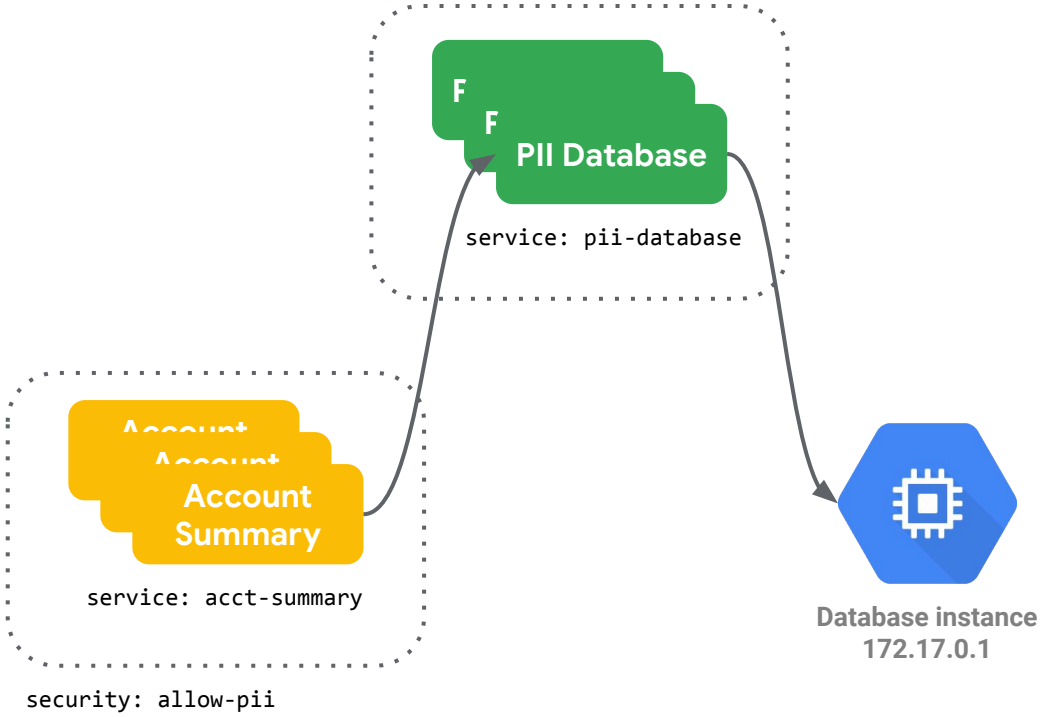
Network Policy can be used to achieve **segmenting**, **routing** and other advanced use cases.



Check out the **default deny** and **kube-system isolation** network policies on GitHub.

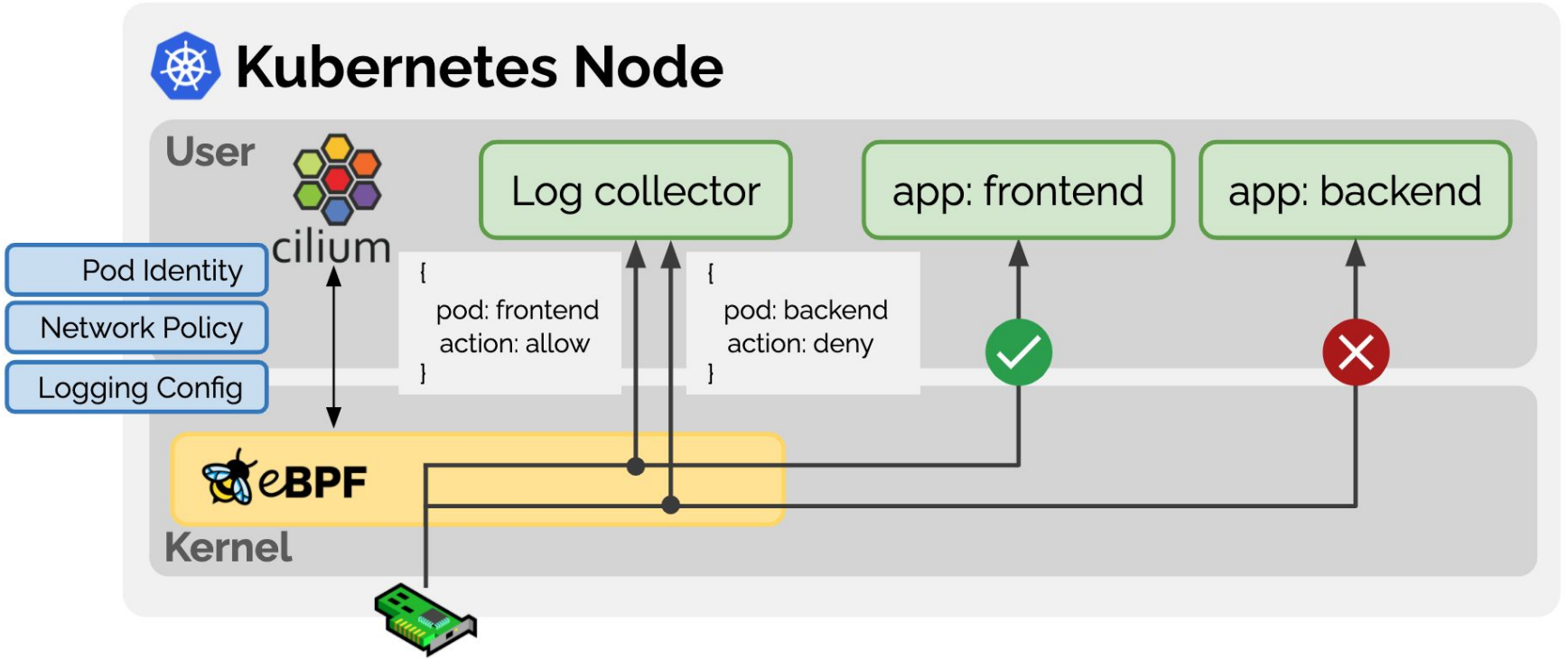
Use Network Policy to **segment** pods, namespaces and even CIDR blocks.


```yaml
kind: NetworkPolicy
spec:
  podSelector:
    matchLabels:
      service: pii-database
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          security: allow-pii
      podSelector:
        matchLabels:
          service: acct-summary
  egress:
  - to:
    - ipBlock
        cidr: 172.17.0.1
    ports:
    - protocol: TCP
      port: 5978
```



Check out the K8S Network Policy spec here.

Network Policy logging harnesses the power of GKE Dataplane V2.



Check out our blog post that explains how this works here.

Network Policy logging will **show** customers what connections are accepted and denied.

- Log all allowed and denied connections to/from pods
- Make sense of networking (IPs) and Kubernetes (labels) in the same view
- Can be configured to log for the entire cluster or per object
- Works on GCP (Cloud Logging) and on-prem (3rd party sinks like Splunk)
- Looks and feels like GCP Firewall Logs

Check out **GKE annotations** in VPC Flow Logs for more visibility inside your clusters.

Policy, pod, namespace, node, and IP/port - all in **one** place!

This feature can be used to satisfy key **network** and **security** use cases.

- Which pods are talking to the internet (avoids the ip-masq-agent problem)
- Which namespaces are talking to each other (tiered multi-tenant deployments)
- Which policies are working as expected, which policies are missing (using a generic allow-all policy)
- Which pods are under a DoS attack (based on the number of deny logs)

Introducing GKE Dataplane V2

GKE Dataplane V2 is the **foundation** we need for advanced networking use cases.

IPtables

- Today's data plane is based on iptables
- Kubeproxy performs K8S service resolution
- Calico enforces K8S network policy

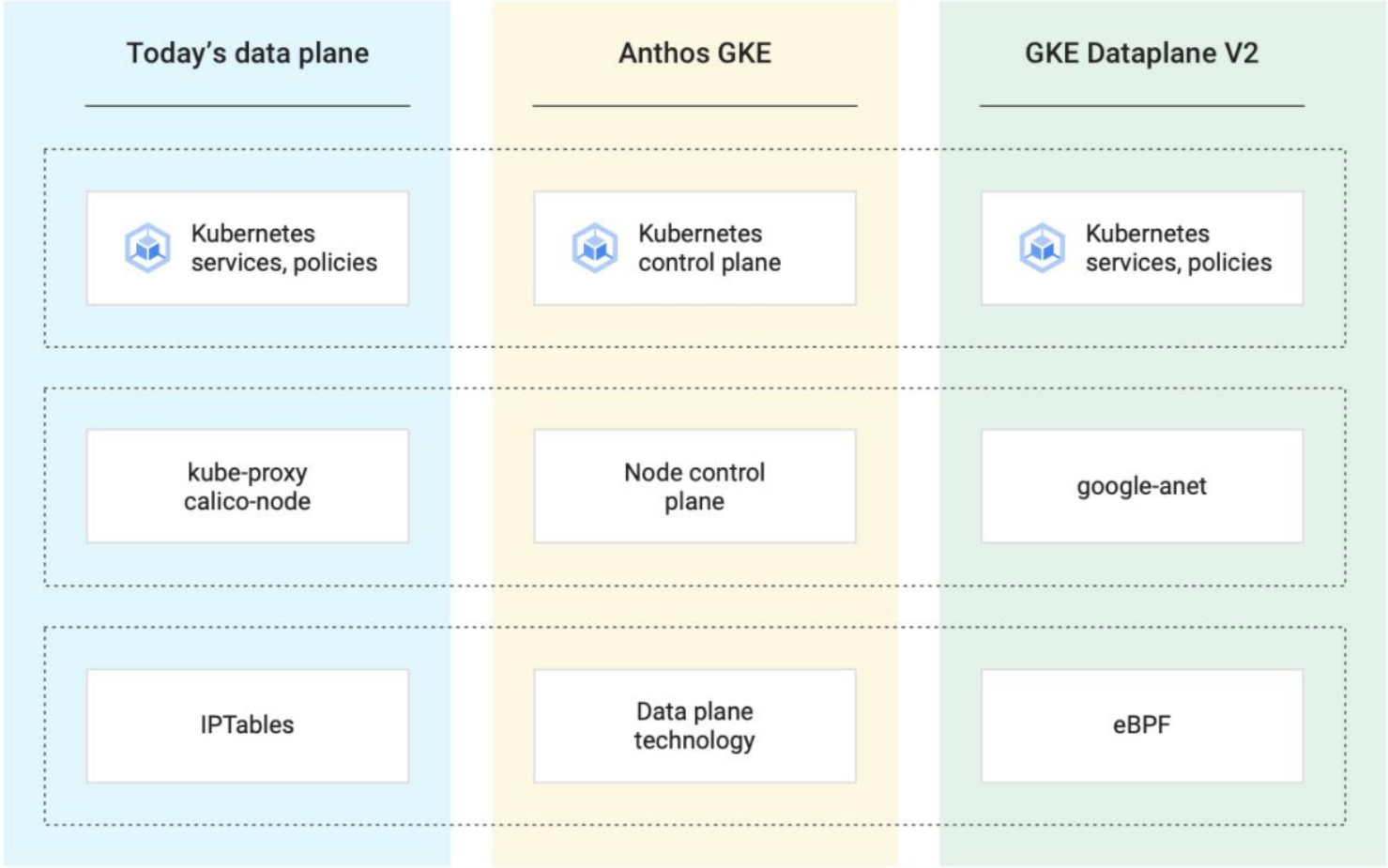
eBPF

- eBPF provides more flexibility, visibility in the data plane
- Building on top of existing Cilium OSS
- Maintaining **parity** with existing GKE features

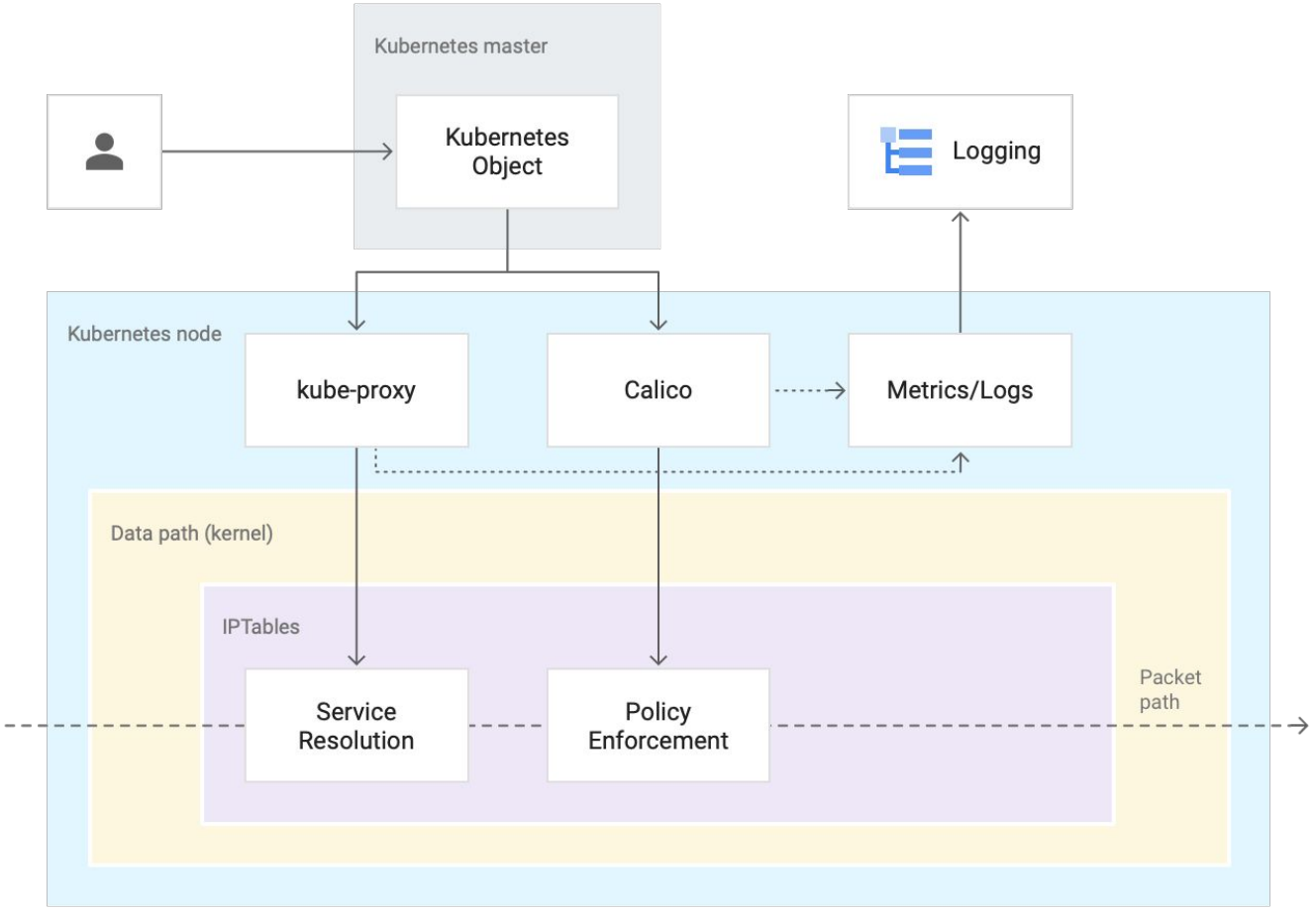
GKE Dataplane V2

- Security (pod identity)
- Scalability (15K nodes)
- Extensibility (IPv6)
- Consistency (hybrid multi-cluster policy)

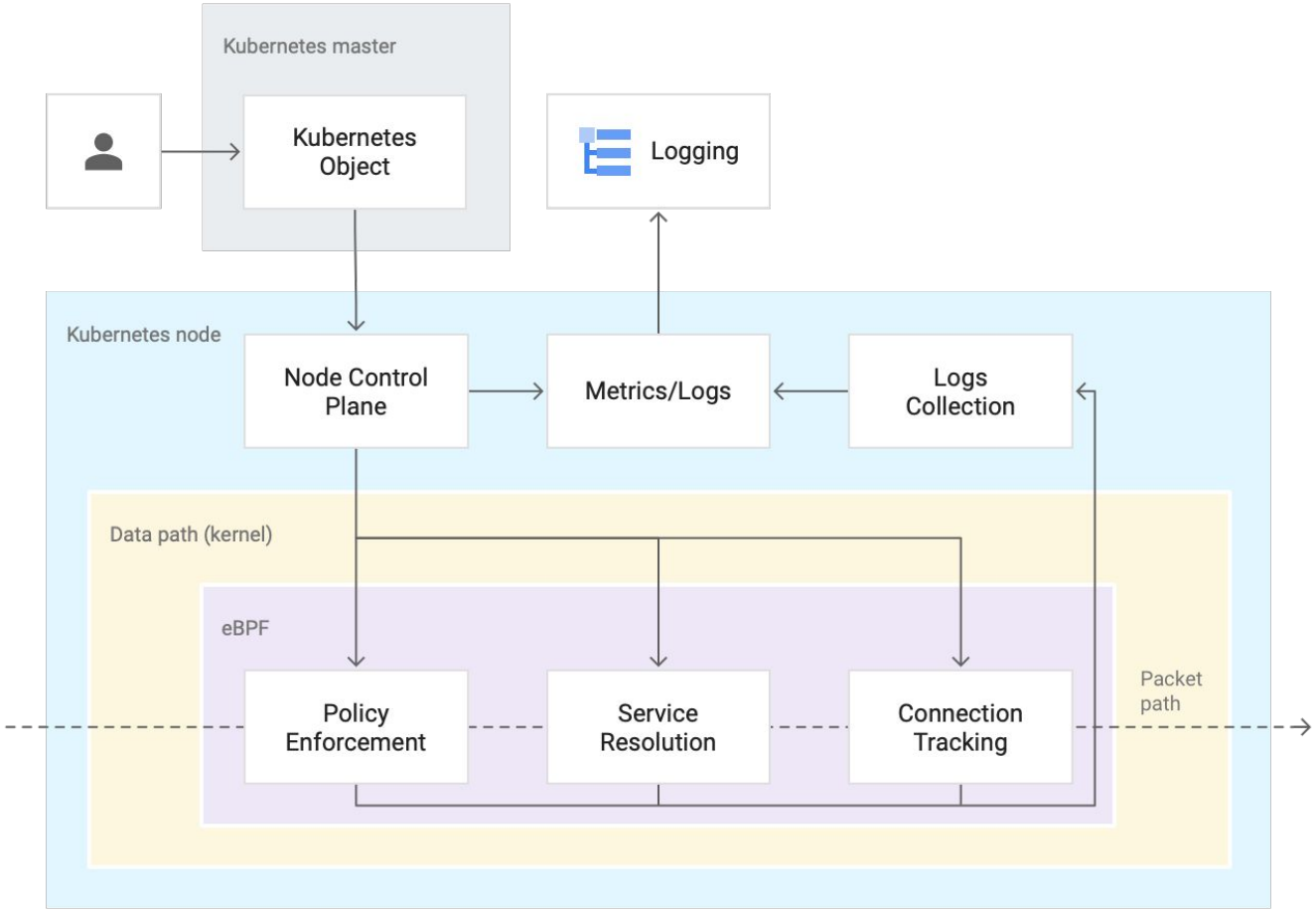
GKE Dataplane V2 is built on **eBPF**.



Functional diagram using **iptables**



Functional diagram using **eBPF**



kube-system looks very different with Dataplane V2


```
NAME
calico-node-2qsrg
calico-node-dmxw4
calico-node-vertical-autoscaler-bcc7978d-nkcd6
calico-node-wnx2p
calico-typha-5cdd59dc44-kkwfp
calico-typha-horizontal-autoscaler-7cd7856b7b-95lhp
calico-typha-vertical-autoscaler-7c4b89c9-stbvr
event-exporter-gke-59b99fdd9c-nx7cm
fluentbit-gke-8p8zg
fluentbit-gke-hl2v5
fluentbit-gke-jtpsr
gke-metrics-agent-747wx
gke-metrics-agent-pzxrq
gke-metrics-agent-qtzh9
ip-masq-agent-5bxpw
ip-masq-agent-k2jh6
ip-masq-agent-t7xjh
kube-dns-7c976ddbdb-26krf
kube-dns-7c976ddbdb-fqnck
kube-dns-autoscaler-5c78d65cd9-f2x72
kube-proxy-gke-legacy-netpol-demo-default-pool-24d57700-ptvv
kube-proxy-gke-legacy-netpol-demo-default-pool-24d57700-q90t
kube-proxy-gke-legacy-netpol-demo-default-pool-24d57700-sz3p
l7-default-backend-5b76b455d-prklq
metrics-server-v0.3.6-547dc87f5f-8m9j5
prometheus-to-sd-6sk87
prometheus-to-sd-v6lht
prometheus-to-sd-vjtfr
stackdriver-metadata-agent-cluster-level-5b8988d77f-pskln
```

```
NAME
anet-operator-78cf8bd5-lzmk6
anetd-k4fv4
anetd-qkmlp
anetd-xvqc8
event-exporter-gke-59b99fdd9c-94c5c
fluentbit-gke-6pztn
fluentbit-gke-jtjxx
fluentbit-gke-pr9pw
gke-metrics-agent-256b5
gke-metrics-agent-6f86v
gke-metrics-agent-j2l7w
kube-dns-7c976ddbdb-h6mqm
kube-dns-7c976ddbdb-tz4fb
kube-dns-autoscaler-5c78d65cd9-xm2nx
l7-default-backend-5b76b455d-8nzbh
metrics-server-v0.3.6-547dc87f5f-rl45l
netd-8wcgm
netd-c5qdl
netd-fgr7c
prometheus-to-sd-75fbq
prometheus-to-sd-85dbx
prometheus-to-sd-rhxp4
stackdriver-metadata-agent-cluster-level-8474b45b55-g5tz8
```

GKE Dataplane V2 increases **consistency** across different environments.

1 Security

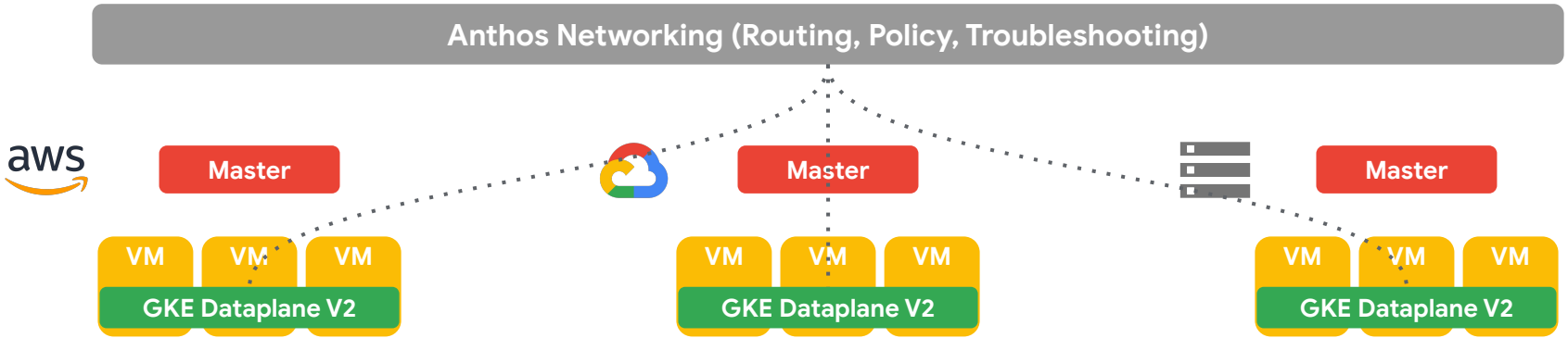
- Kubernetes Network Policy is **always on**, in every cluster
- Node Network Policy abstracts underlying infra with YAML based **node firewalls**

2 Operations

- **Network Policy logging** is built into GKE Dataplane V2
- Same **troubleshooting** tools across different environments

3 Scale

- **Kube-proxy less** implementation, regardless of underlying infra
- Same cluster network **performance** no matter where it is running



What’s next for GKE Dataplane V2

- In-place upgrade to Dataplane V2
- 5K+ nodes per cluster
- Dataplane V2 as the default dataplane for GKE/Anthos

Next steps

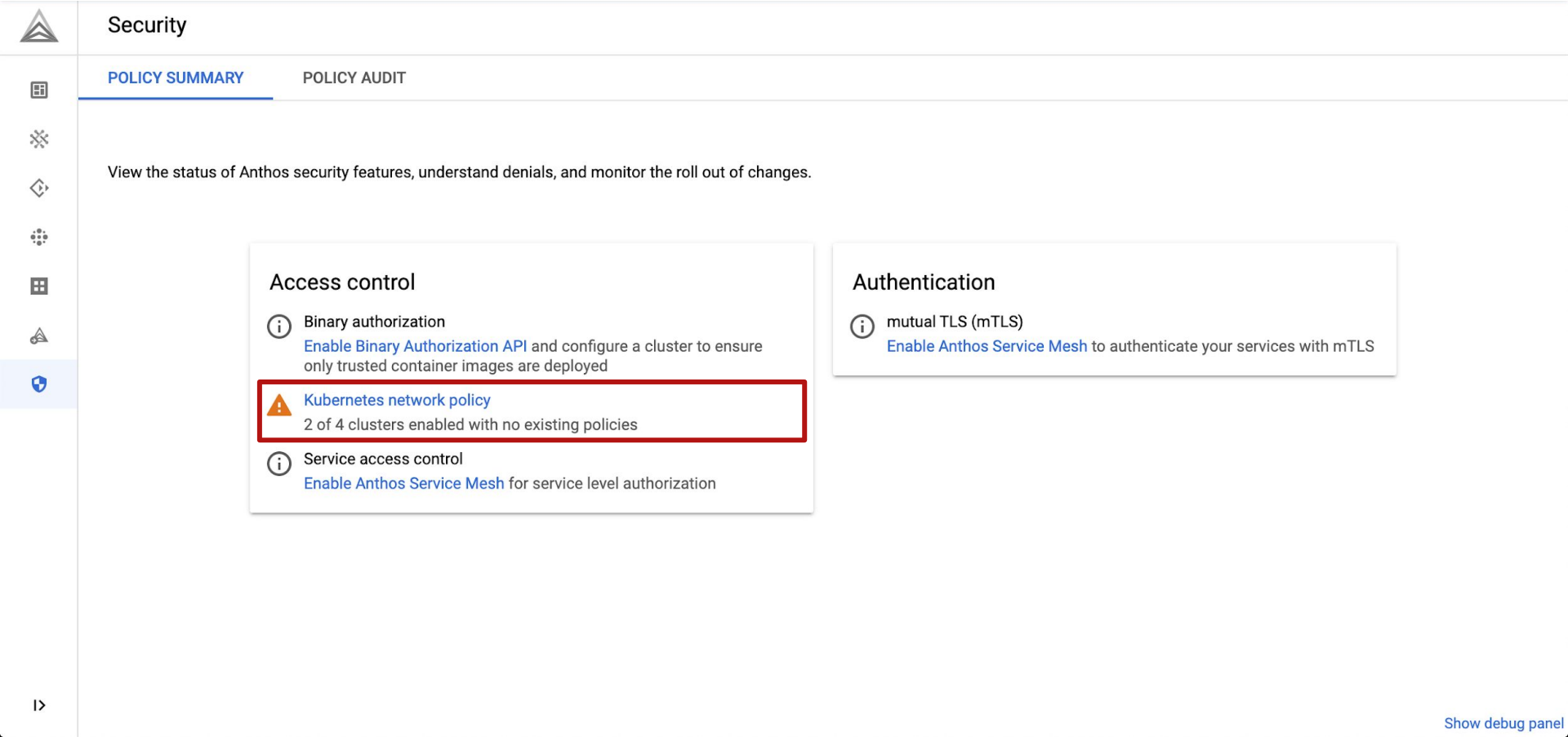
Try out Dataplane V2 and Network Policy Logging!

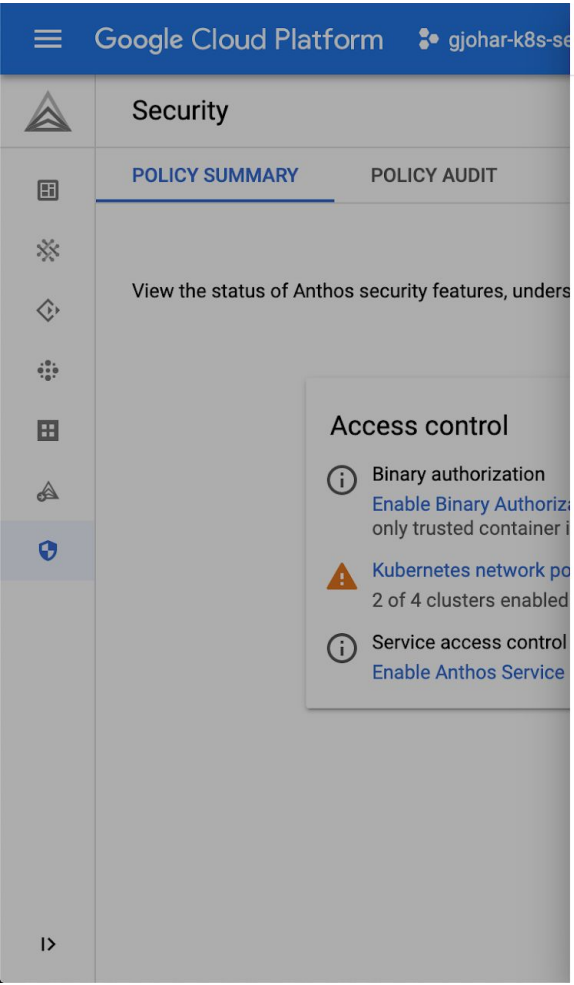
- Try the features out

```
gcloud container clusters create <cluster name> \
--enable-dataplane-v2 \
--release-channel rapid \
--enable-ip-alias \
{--region region-name | --zone zone-name}
```

- Read the public docs and blog post for more details

Introducing **Security Insights UI** for GKE and Anthos





Kubernetes network policy

A Kubernetes network policy is a specification of how groups of pods are allowed to communicate with each other and other network endpoints.
Learn more

Cluster status

Time Span: 1 week

Cluster	Location	Status	View policy	Denials
legacy-netpol-demo	us-east1-b	Enabled	No policy defined	-
netpol-configsync-demo	us-east1-b	Enabled	Policy details	37
prod-1-test-1179600	us-east1-b	Enabled	Policy details	0
prod-test-1179600	us-east1-b	Enabled	No policy defined	-

CLOSE

Security

POLICY SUMMARY | POLICY AUDIT

Cluster: netpol-configsync-demo

Namespace: default

Show system objects

Binary authorization

Enable Binary Authorization API to ensure only trusted container images are deployed

Kubernetes network policy

All workloads adhering to a network policy

Service access control

Enable ASM for service level authorization
Learn more

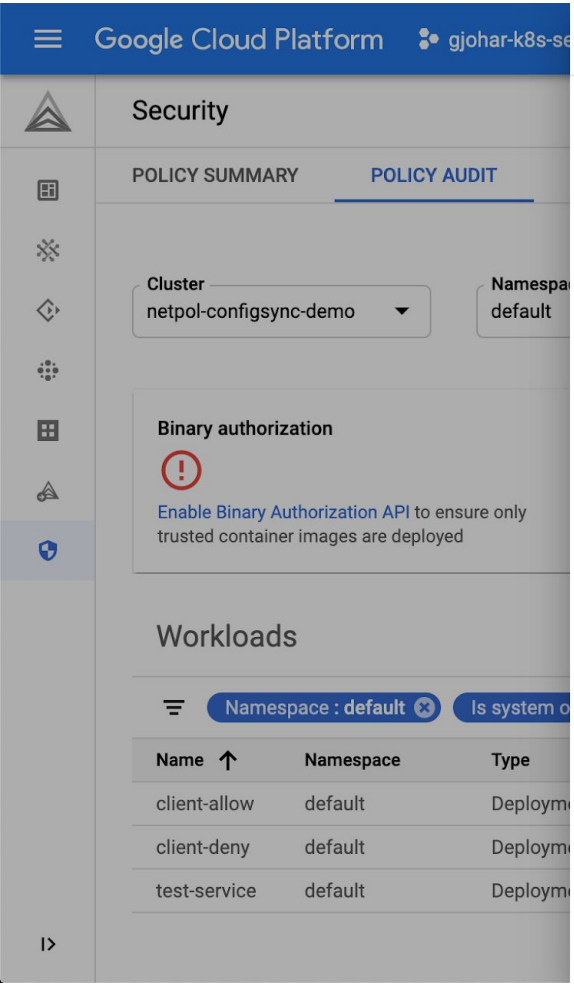
mTLS status

Enable ASM to authenticate services
Learn more

Workloads

Is system object : False | Namespace : default | Filter table

Name	Namespace	Type	Kubernetes network policy	Service access control	mTLS details
client-allow	default	Deployment	Enabled	Disabled	Partial mTLS
client-deny	default	Deployment	Enabled	Disabled	Partial mTLS
test-service	default	Deployment	Enabled	Disabled	Partial mTLS

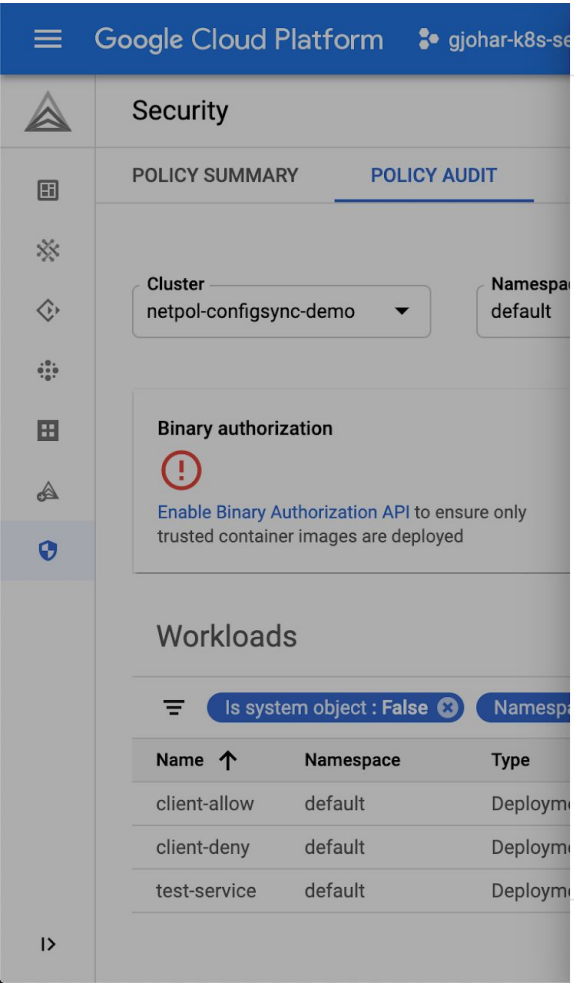


Kubernetes network policy

View the YAML file to see how network policy is configured for this workload.

Kind	Scope	Created
Network Policy	workload	9/1/2020, 4:47:33 PM
Network Policy	namespace	9/1/2020, 4:44:26 PM

CLOSE



Kubernetes network policy

View the YAML file to see how network policy is configured for this workload.

Kind	Scope	Created

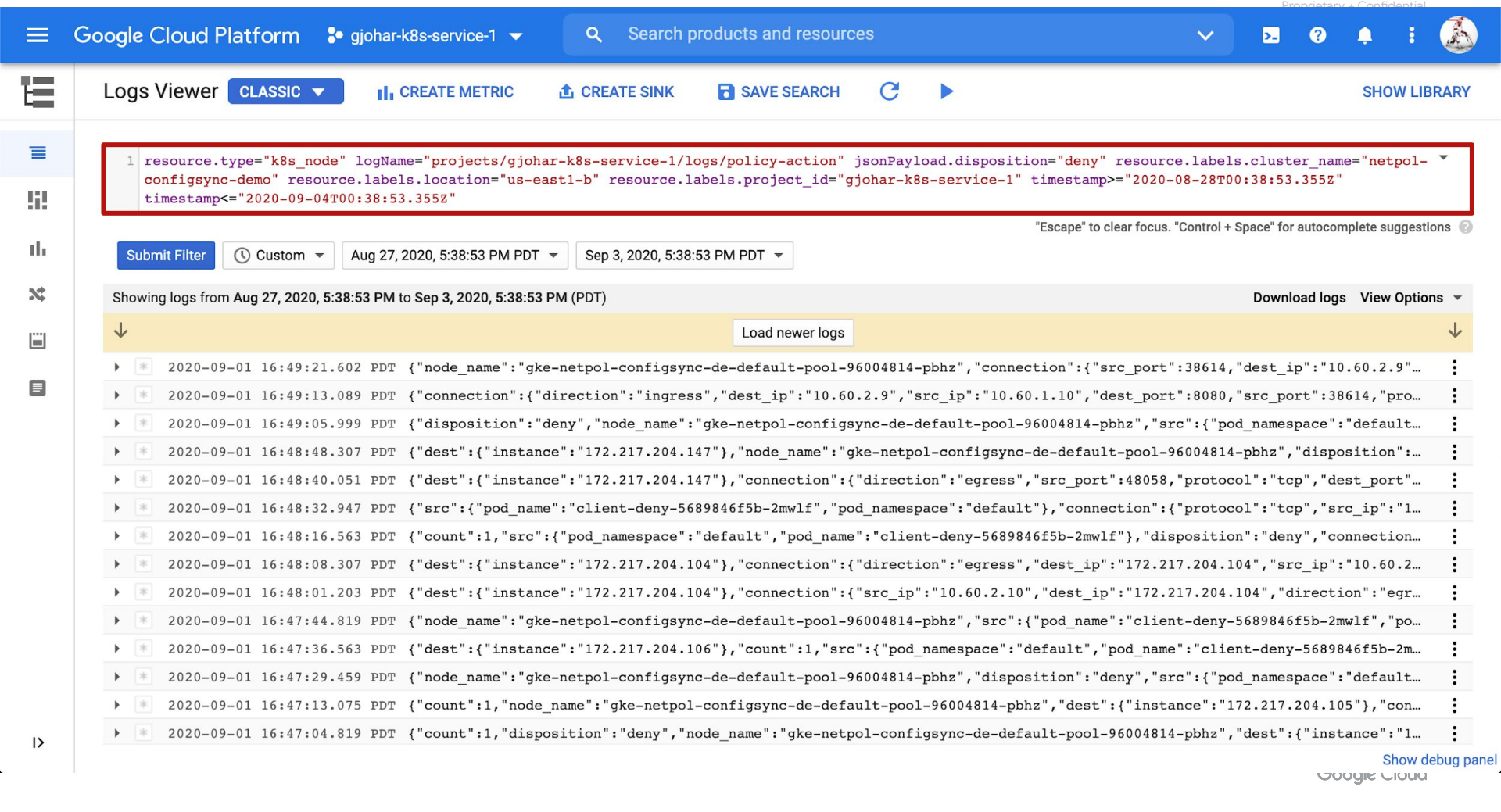
Network Policy

Scope: workload

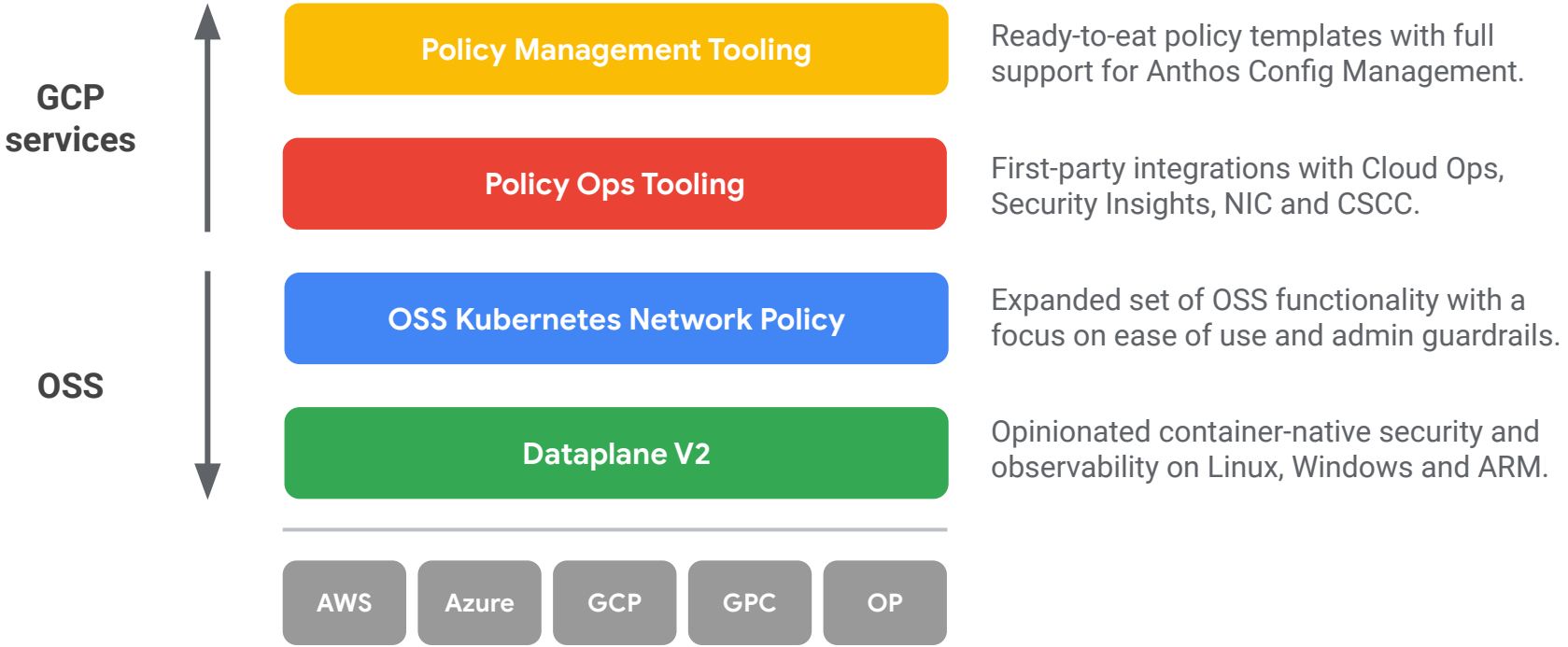
Created: 9/1/2020, 4:47:33 PM

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"networking.k8s.io/v1","kind":"NetworkPolicy","metadata":{"annotatio
    policy.network.gke.io/enable-logging: "true"
  creationTimestamp: "2020-09-01T23:47:33Z"
  generation: 1
  name: allow-all
  namespace: default
  resourceVersion: "5850575"
  selfLink: /apis/networking.k8s.io/v1/namespaces/default/networkpolicies/allow-all
  uid: ca1fee83-7d24-4e58-910d-439e49a36cf5
spec:
  egress:
  - {}
  ingress:
  - {}
```

CLOSE



Putting it all together...



Ready-to-eat policy templates with full support for Anthos Config Management.

First-party integrations with Cloud Ops, Security Insights, NIC and CSCC.

Expanded set of OSS functionality with a focus on ease of use and admin guardrails.

Opinionated container-native security and observability on Linux, Windows and ARM.

Tutorial

https://github.com/nbrandaleone/dataplane-v2

Thank you.

Google Cloud