KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2019

# KAS Proxy Service

## A new networking traffic abstraction in Kubernetes

Walter Fender: cheftako@google.com
GitHub: cheftako
Software Engineer @ Google

Yongkun Gui: ygui@google.com
GitHub: anfernee
Software Engineer @ Google

# KAS Proxy Service

Software Engineer - Google
Cheftako - GitHub
Sig Cloud Provider - TL
Sig API Machinery - Contributor

Software Engineer - Google
Anfernee - GitHub
Sig Networking - Contributor

# KAS Proxy Service

We are adding a configurable, extensible proxy service for connections outbound from the K8s API Server. This includes a reference implementation of the Proxy Server and a Proxy Agent.

- But why?
- How does it work?
- How do I use it?
- What’s next?
- Can I help?

# KAS Proxy Service

Kubernetes API Server gets a network proxy AKA connectivity service? But why?
Is the Proxy Service a requirement? **YES**!

- There should be NO preferential cloud providers in Kubernetes.
    - Cloud provider code does not belong in the Kubernetes API Server
    - https://github.com/kubernetes/kubernetes/blob/release-1.15/cmd/kube-apiserver/app/server.go#L239

- The Kubernetes API Server should always be able to talk to the Cluster.
    - To always be able to do that the KAS needs Contextual Routing
    - IP address overlap in cluster network vs control plane or hybrid scenarios

- We need to remove all vulnerabilities we are aware of
    - https://groups.google.com/d/msg/kubernetes-security-announce/tyd-MVR-tY4/tyREP9-qAwAJ
    - Secure communication (tunnel) between Kubernetes API Server and the systems it talks to.

# KAS Proxy Service

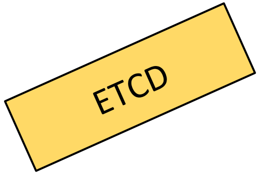
Kubernetes API Server gets a network proxy AKA connectivity service? But why?
Is the Proxy Server also a nice to have? **YES**!

- No one likes SSH Tunnels
  - https://github.com/kubernetes/kubernetes/issues/54076
  - Code complexity and wiring
  - Requires SSHD running on Nodes, accepting requests

- Extensibility for Control Plane Communication
  - Alternate connection direction (cluster to master)
  - Alternate connection protocol (grpc, vpn, ....)

# KAS Proxy Service

What outbound requests does the Kubernetes API Server make?



ETCD

# KAS Proxy Service

What outbound requests does the Kubernetes API Server make?

ETCD

Image Policy Webhook

Aggregate API Server

Admission Webhook

Authentication, Authorization and Audit Webhooks

KMS GRPC Service

# KAS Proxy Service

What outbound requests does the Kubernetes API Server make?

ETCD

Nodes Proxy

Image Policy Webhook

Pods Exec

Admission Webhook

Pods Portforward

Services Proxy

Aggregate API Server

Admission Webhook

Pods Proxy

Pods Attach

Pods Logs

KMS GRPC Service

Authentication, Authorization and Audit Webhooks

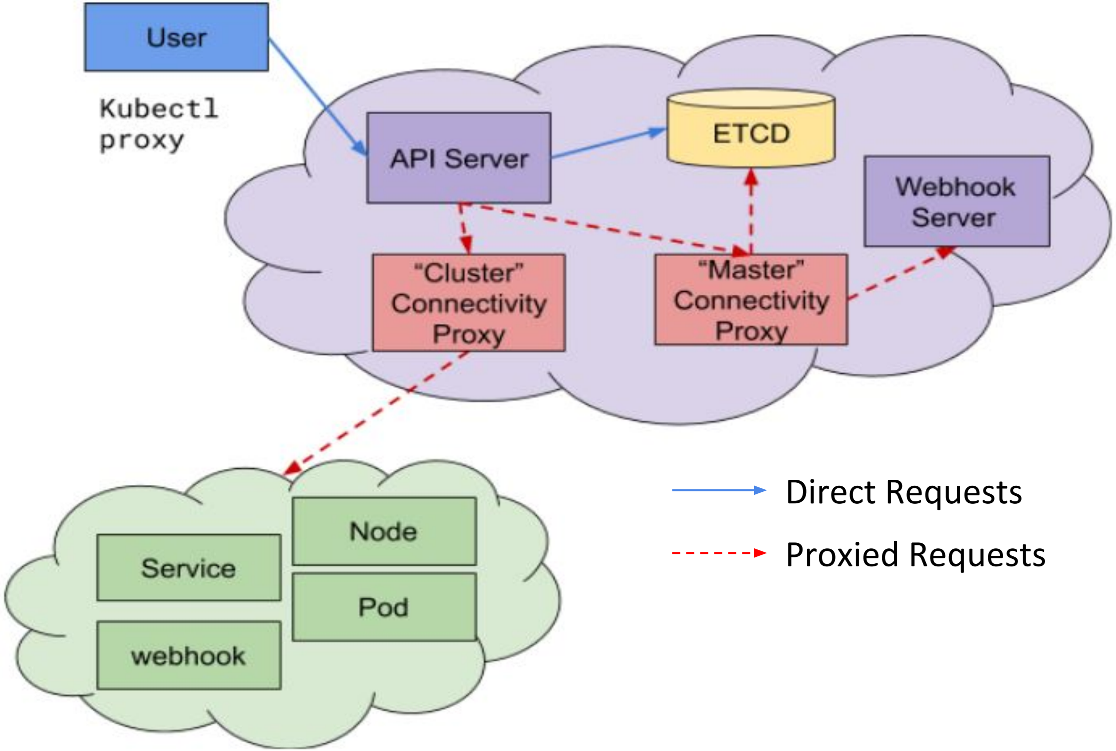
Aggregate API Server

# KAS Proxy Service

- “Cluster” Connectivity Service is used for all traffic destined for the cluster or data network. This should include all the nodes which do the work of the cluster.

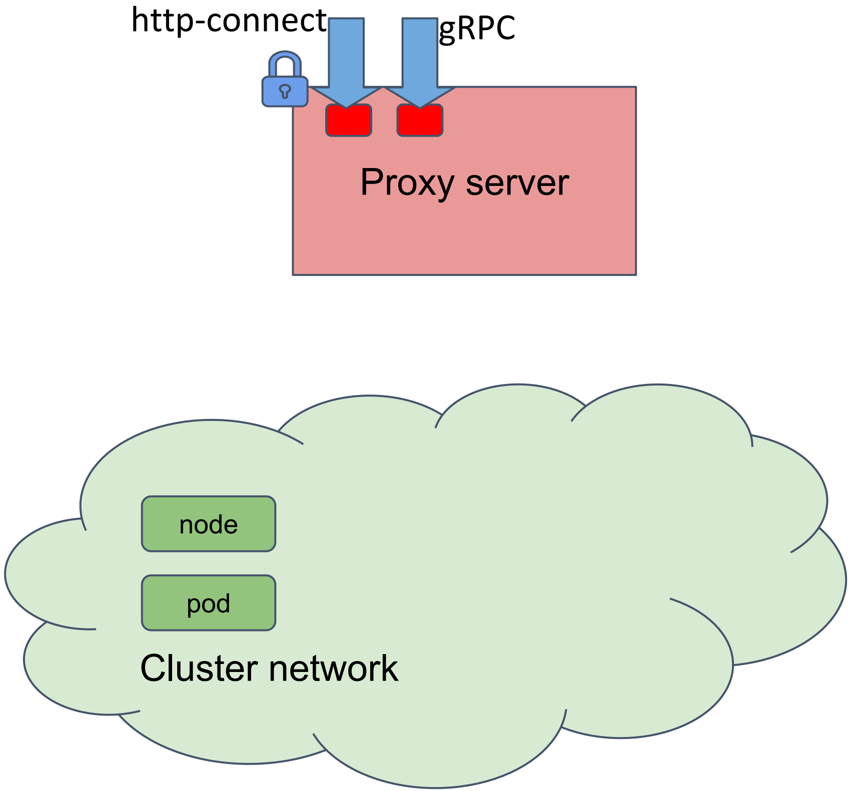
- “Master” Connectivity Service is used for all traffic destined for the control plane or network. This should include anything maintaining the cluster.

- “ETCD” Connectivity Service for all ETCD traffic.



# KAS Proxy Service

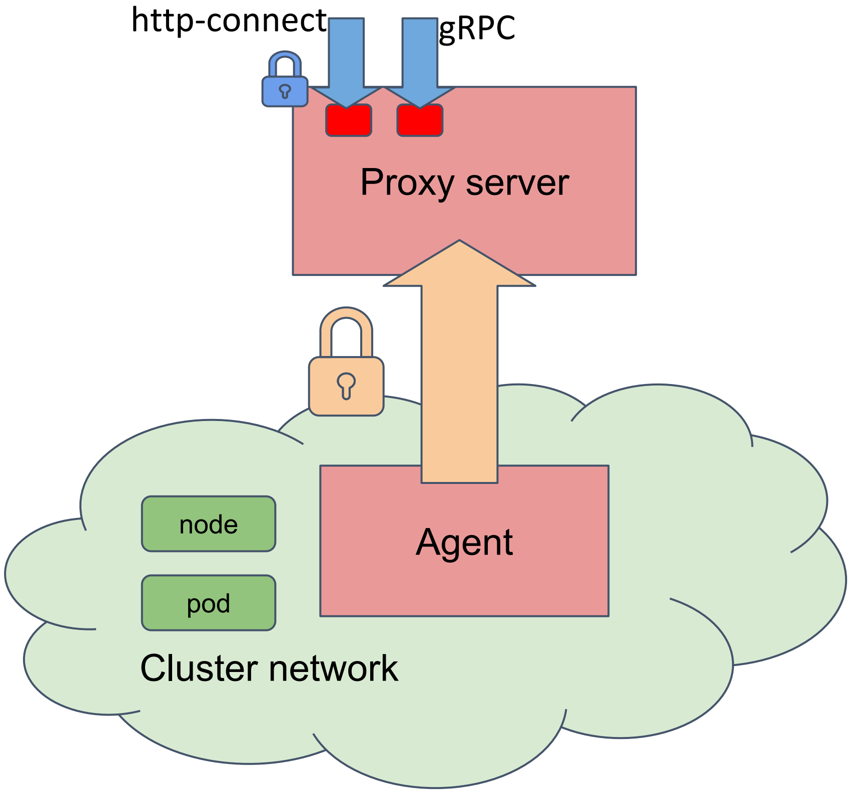
- Connectivity Proxy server
  - Support both HTTP Connect and gRPC as proxy interface
  - Secured with mTLS



# KAS Proxy Service

- Connectivity Proxy server
  - Support both HTTP Connect and gRPC as proxy interface
  - Secured with mTLS

- Agent-to-Proxy Tunnel
  - Dialback
  - Secured with mTLS

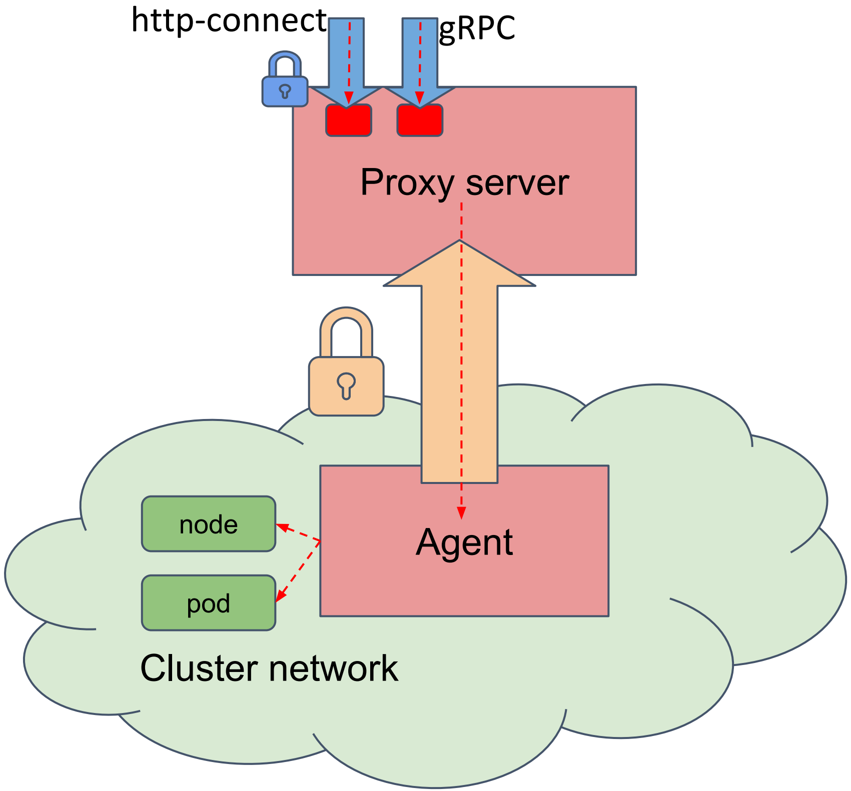


# KAS Proxy Service

- Connectivity Proxy server
  - Support both HTTP Connect and gRPC as proxy interface
  - Secured with mTLS

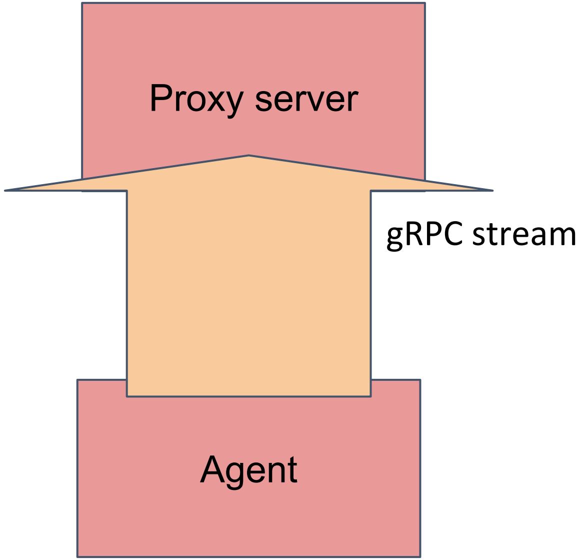
- Agent-to-Proxy Tunnel
  - Dialback
  - Secured with mTLS

- Agent
  - Distribute stream to different endpoints



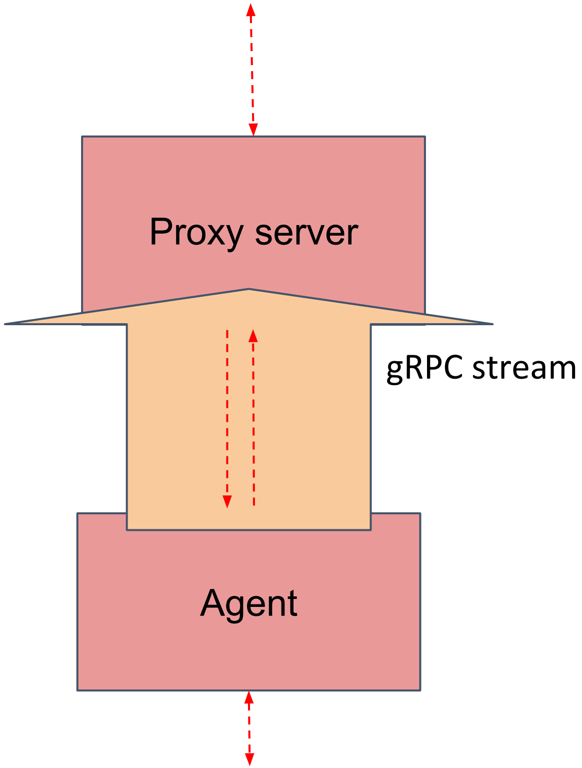
# KAS Proxy Service

- Proxy-Agent tunnel
  - gRPC bidirectional streaming API



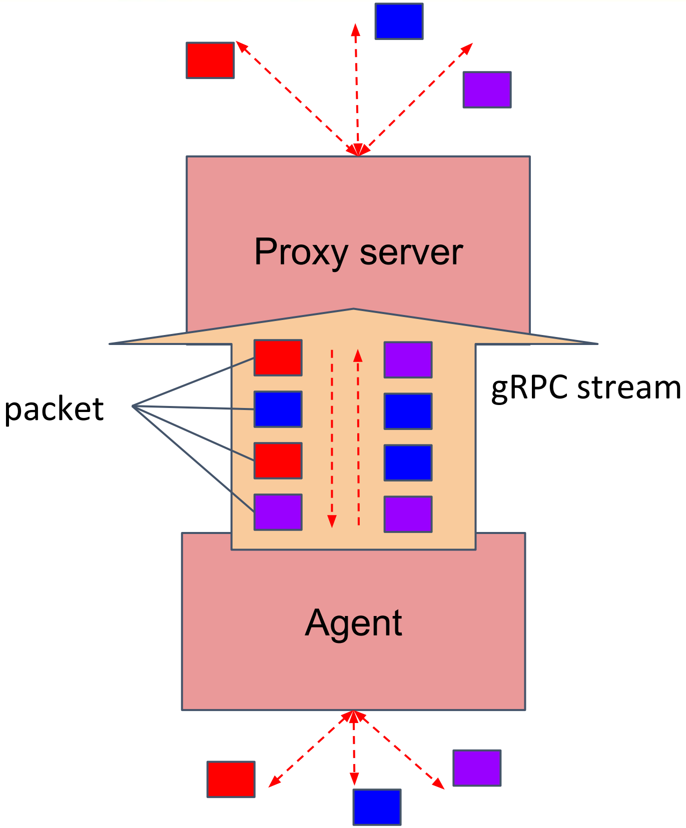
# KAS Proxy Service

- Proxy-Agent tunnel
  - gRPC bidirectional streaming API
  - Full duplex connection



# KAS Proxy Service

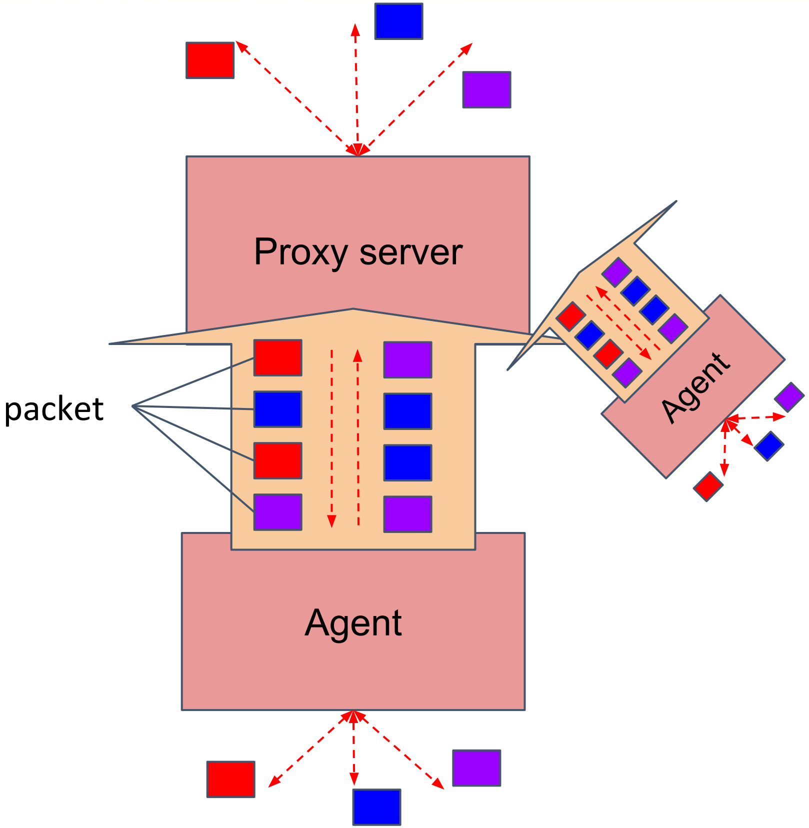
- Proxy-Agent tunnel
  - gRPC bidirectional streaming API
  - Full duplex connection
  - Multiplex over one tunnel
  - Resumable connection



# KAS Proxy Service

- Proxy-Agent tunnel
  - gRPC bidirectional streaming API
  - Full duplex connection
  - Multiplex over one tunnel
  - Resumable connection

- Advanced features
  - Load Balancer
  - Monitoring
  - Auditing
  - Throttling



# KAS Proxy Service

**Connectivity Configuration (**--network-proxy-config-file=**connectivity_service_configuration.yaml):**

```
apiVersion: apiserver.k8s.io/v1alpha1
kind: ConnectivityServiceConfiguration
connectionServices:
- name: "cluster"
  connection:
    type: "http-connect"
    url: "https://127.0.0.1:8131"
    caBundle: "/etc/srv/kubernetes/pki/proxy-server/ca.crt"
    clientKeyFile: "/etc/srv/kubernetes/pki/proxy-server/client.key"
    clientCertFile: "/etc/srv/kubernetes/pki/proxy-server/client.crt"
- name: "master"
  connection:
    type: "direct"
- name: "etcd"
  connection:
    type: "direct"
```

# KAS Proxy Service

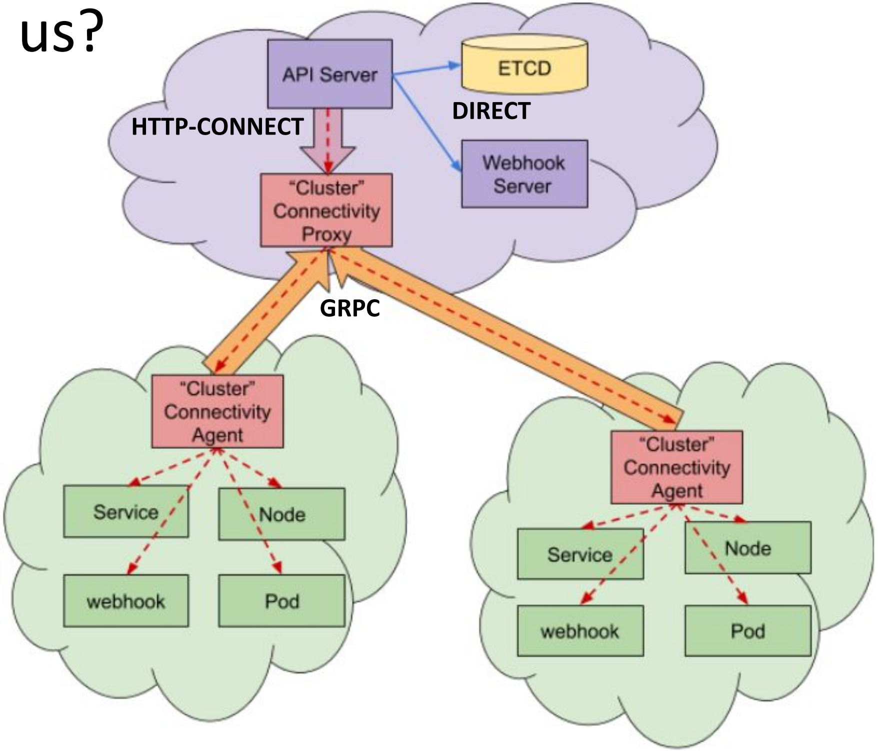
What does this configuration give us?

Where can I find it and the code?

# KAS Proxy Service

What does this configuration give us?

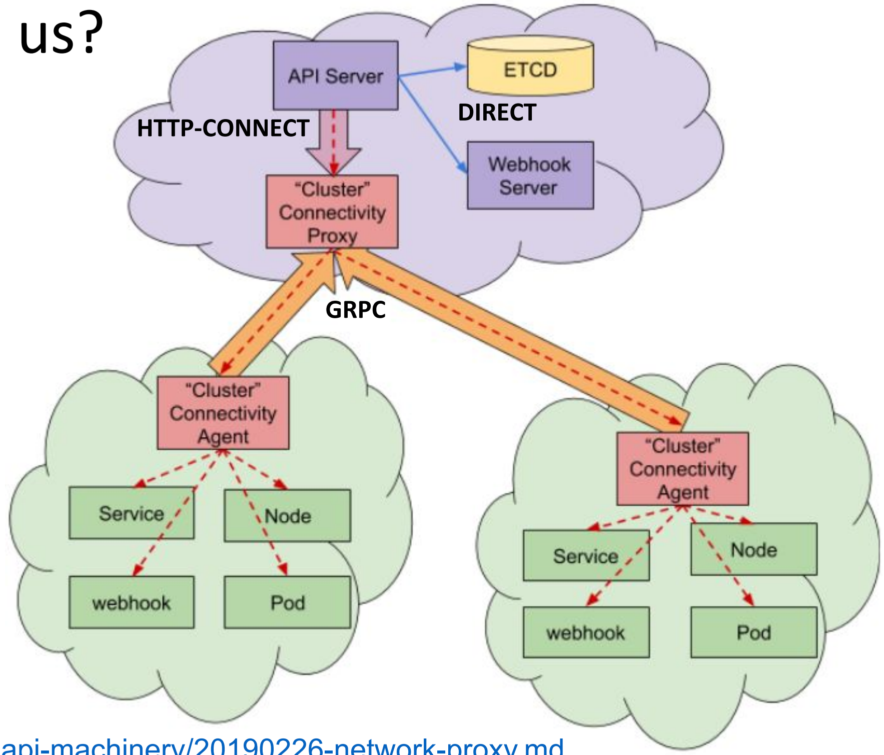
Where can I find it and the code?



# KAS Proxy Service

## What does this configuration give us?

- Secured tunnel to the data plane
- Direct connection to the ETCD server
- Direct connection to other master components



## Where can I find it and the code?

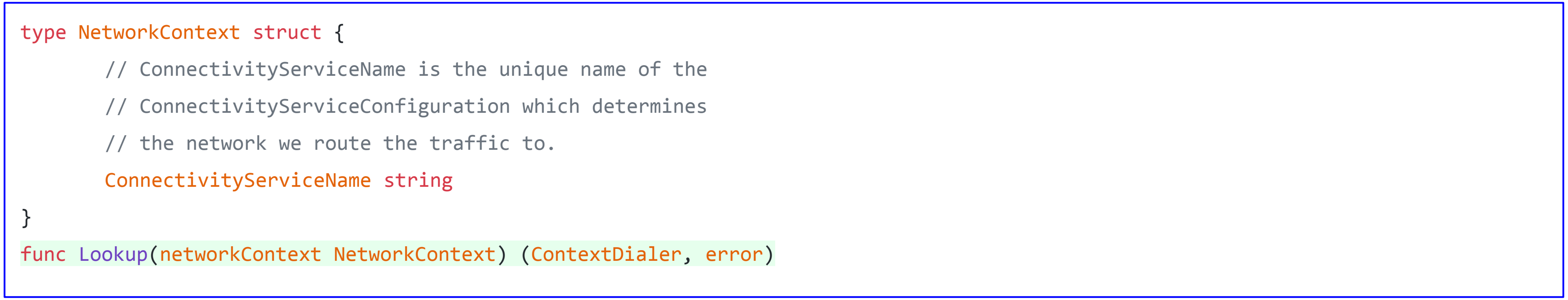
https://github.com/kubernetes/kubernetes/pull/78543

https://github.com/kubernetes-sigs/apiserver-network-proxy

https://github.com/kubernetes/enhancements/blob/master/keps/sig-api-machinery/20190226-network-proxy.md

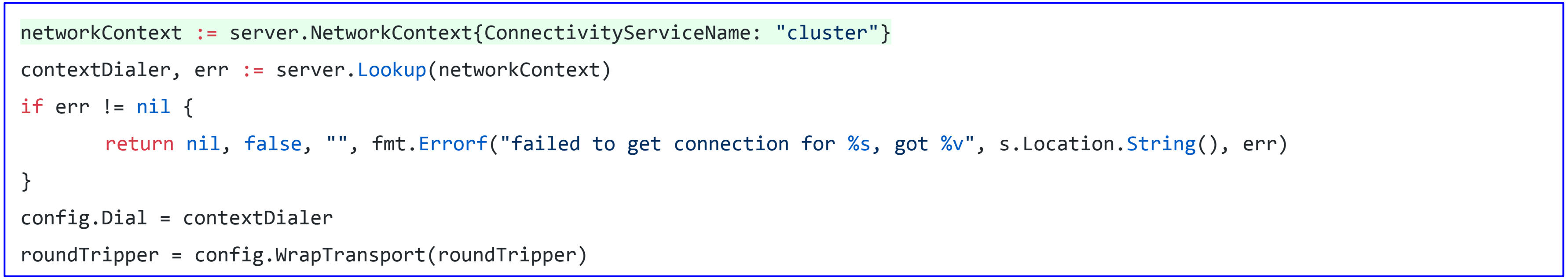
# KAS Proxy Service

## Connectivity API:



```go
type NetworkContext struct {
        // ConnectivityServiceName is the unique name of the
        // ConnectivityServiceConfiguration which determines
        // the network we route the traffic to.
        ConnectivityServiceName string
}
func Lookup(networkContext NetworkContext) (ContextDialer, error)
```

## Connectivity Usage:



```go
networkContext := server.NetworkContext{ConnectivityServiceName: "cluster"}
contextDialer, err := server.Lookup(networkContext)
if err != nil {
        return nil, false, "", fmt.Errorf("failed to get connection for %s, got %v", s.Location.String(), err)
}
config.Dial = contextDialer
roundTripper = config.WrapTransport(roundTripper)
```

# KAS Proxy Service

## (Possible) Futures:

- Alternate communication from KAS to “Cluster”. Securing Server and Agent
    - OpenVPN, StrongSwan, WireGuard, …
    - Server to Agent communication is an extensibility point

- Connectivity Service Configurations beyond “master” and “cluster”.
    - Connectivity by Label?
    - Connectivity by Service?
    - Multi-tenant use cases?

- Allow better connections than HTTP-CONNECT from KAS to Connectivity Server
    - GRPC, already supported by the reference Connectivity Server

- Allow the K8s API Server to not be public
    - Have cluster traffic sent to Agent and be reverse tunneled to the master

# KAS Proxy Service

Interested in Contributing?

- SIG API MACHINERY
- SIG NETWORKING
- SIG CLOUD PROVIDER

Get involved!

Kubernetes SIGs/apiserver-network-proxy