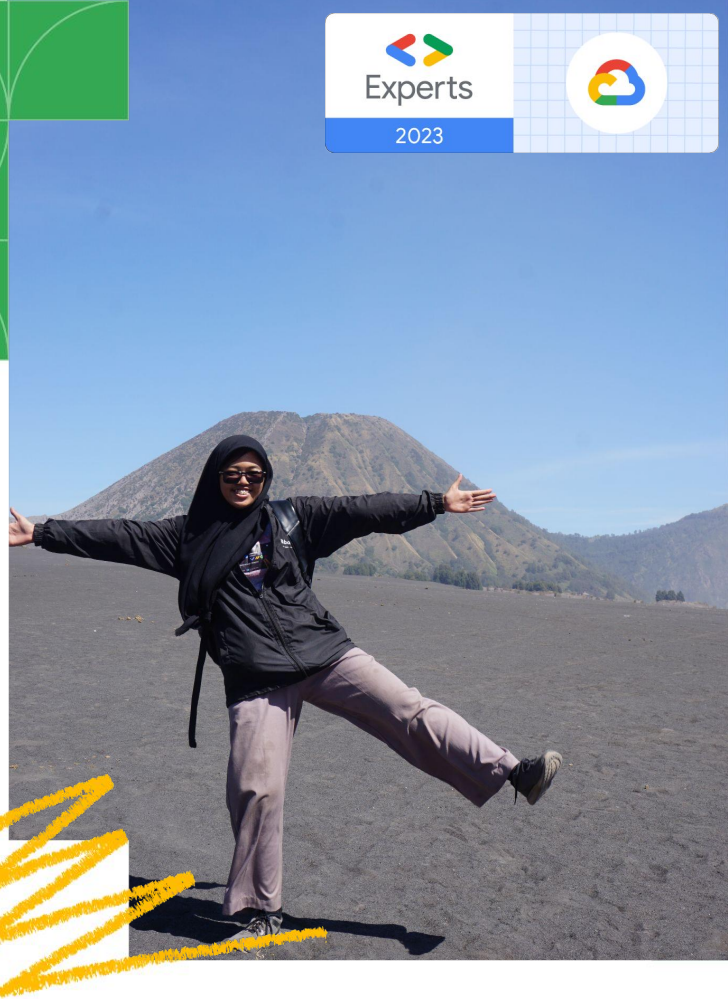
devfest

Experts 2023

Simplifying GKE Multi-Cluster for Disaster Recovery and Failover



Google Developer Groups

GDG Depok

Hello Depok!

Ananda Dwi Rahmawati

- Cloud Engineer @ Activate Interactive Pte Ltd Singapore | 2023 - present
- 4+ years experience
- GDE Cloud Modernization Apps
- Tech background: System, Networking, IaaS & PaaS Cloud, DevOps, a bit of Programming
- https://linktr.ee/misskecupbung

Today’s Agenda

Challenges

Multi-Cluster Solutions in Google Cloud

Demo

Google Developer Groups

GDG Depok

Challenges

Challenges

There are a number of challenges that we may face when using a standalone GKE cluster, including:

- **Scalability**: A standalone GKE cluster can only scale to a limited number of nodes.
- **Availability**: A standalone GKE cluster is susceptible to outages. If there is an outage in the region or zone where your cluster is deployed, your application will be unavailable.
- **Security**: A standalone GKE cluster is a single point of failure. If your cluster is compromised, your entire application could be at risk.
- **Compliance**: If you have data sovereignty requirements, you may need to deploy your application in specific regions or zones.

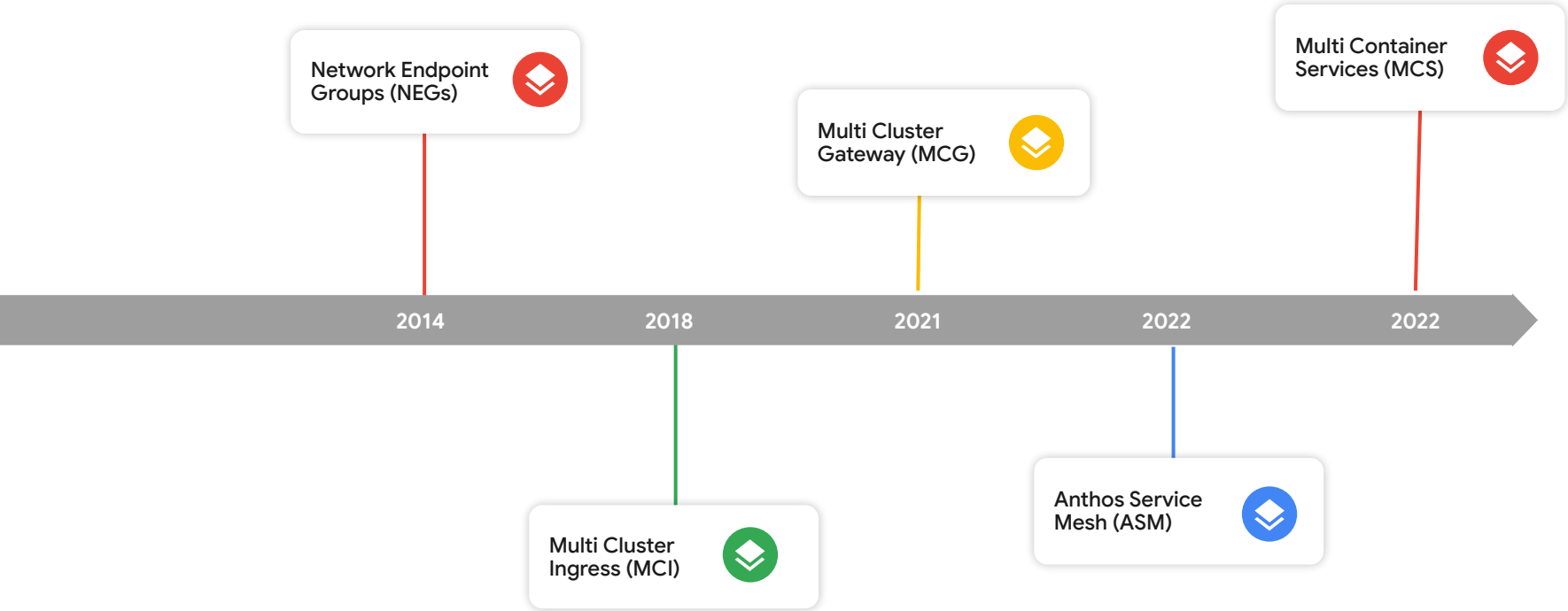

'Simple Statement or URL',
style: TextStyle(
color: Colors.green[200],
),
),
devfest
s.star,
r: Colors.green[500],
Text('23'),
“Multiple clusters to address state management, privacy, scalability, availability, and data sovereignty requirements”

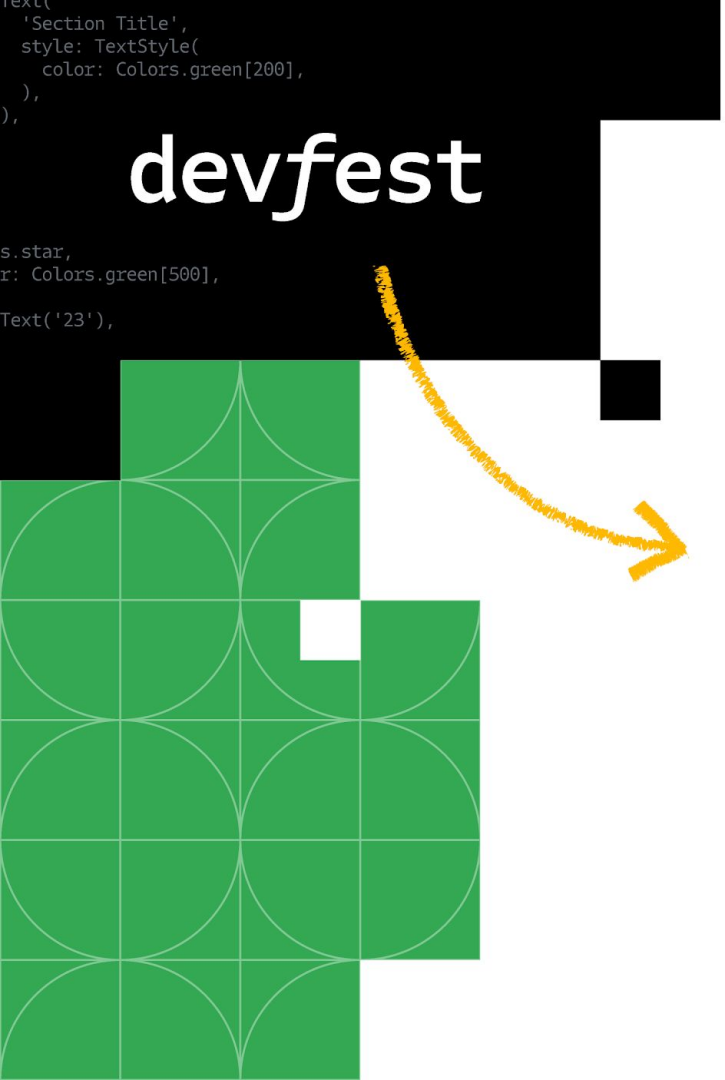
Google Developer Groups

GDG Depok

Multi-Cluster Solutions in Google Cloud

Multi-Cluster Solutions Offered for GKE





Google Developer Groups

GDG Depok

Network Endpoint Groups (NEGs)

Network Endpoint Groups (NEGs)

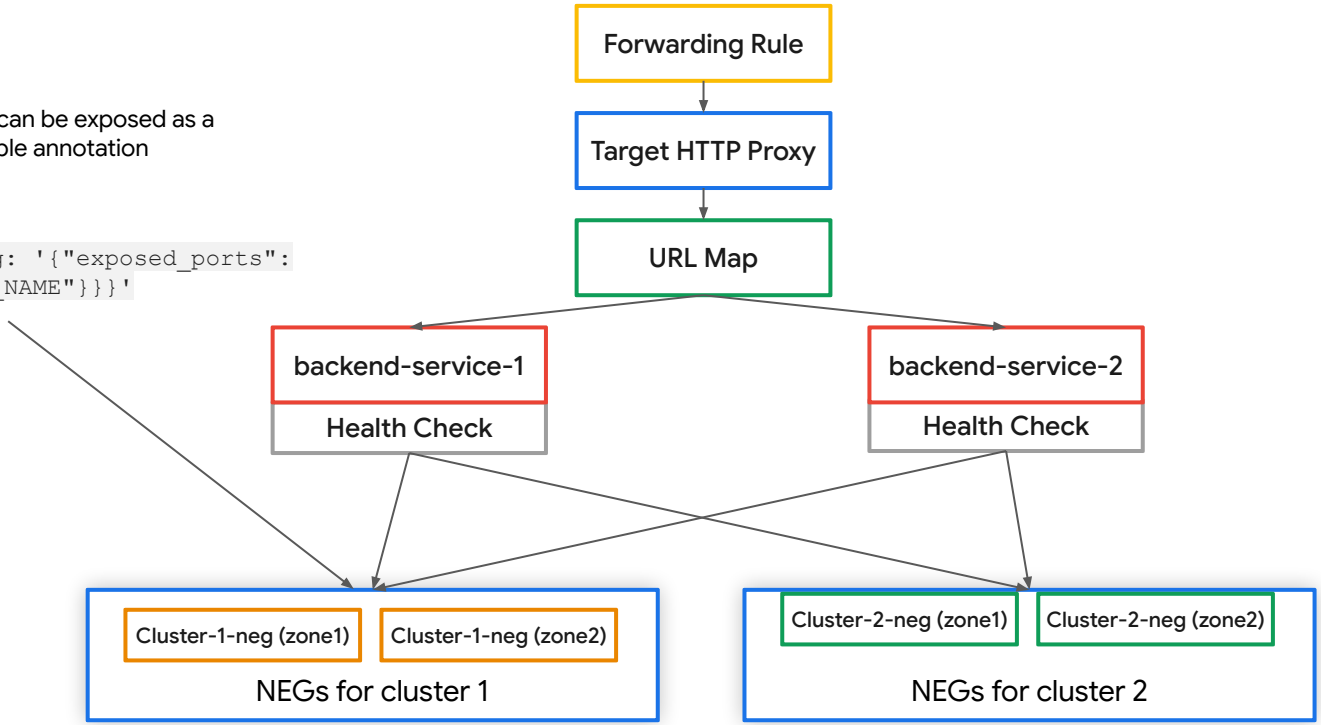
Network Endpoint Groups (NEGs) are a **basic load balancing feature** in Google Cloud Platform that allows you to **distribute traffic across multiple resources** of a service.

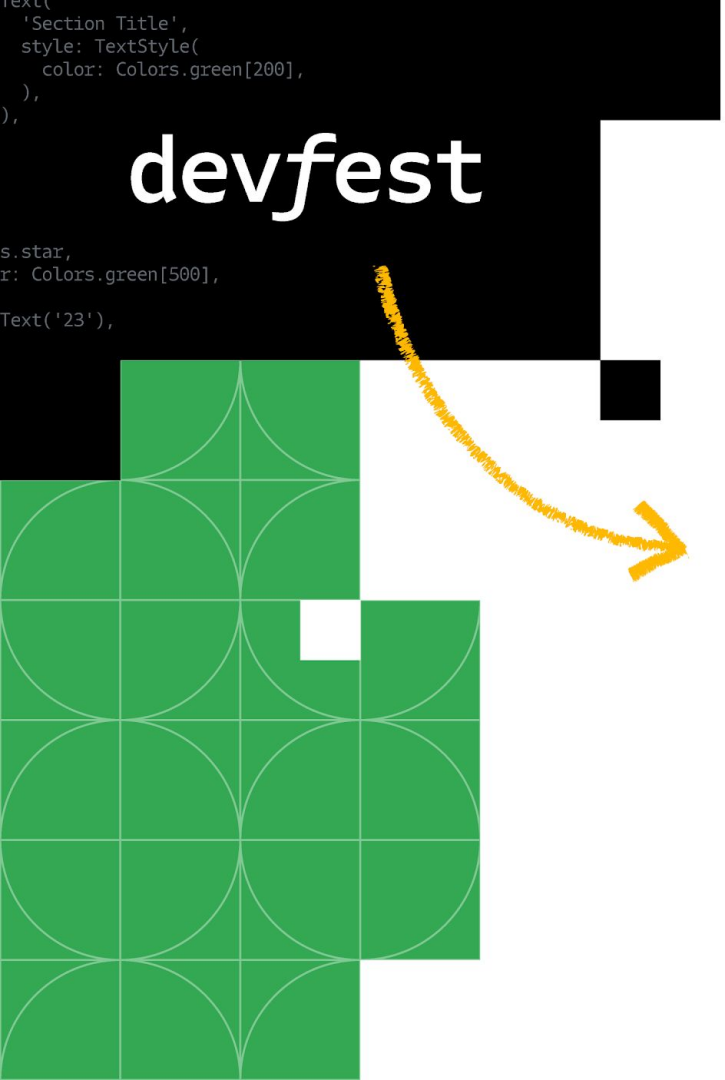
NEGs are relatively easy to set up and use, but they d**o not support all of the features of a full-fledged service mesh**, such as traffic splitting and fault tolerance.

Network Endpoint Group (NEGs)

A **Kubernetes Service** can be exposed as a NEG in GKE using a simple annotation

```
annotations:
cloud.google.com/neg: '{"exposed_ports":
{"80":{"name": "NEG_NAME"}}}'
```





Google Developer Groups

GDG Depok

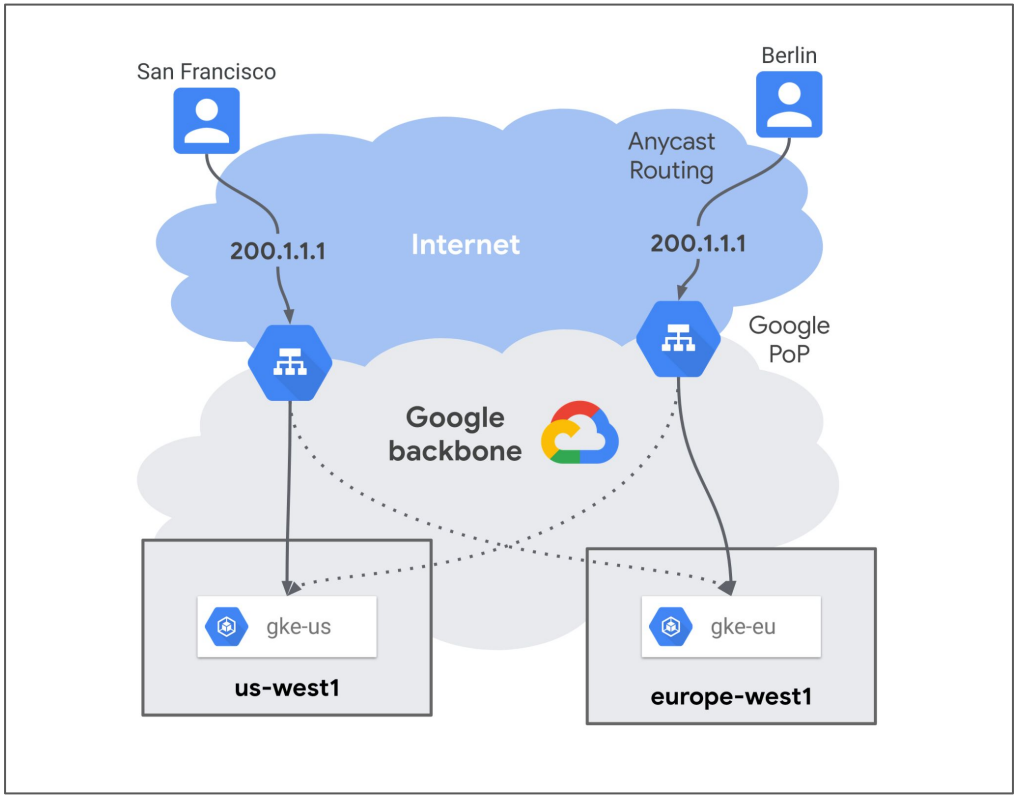
Multi-Container Ingress (MCI)

Multi Cluster Ingress

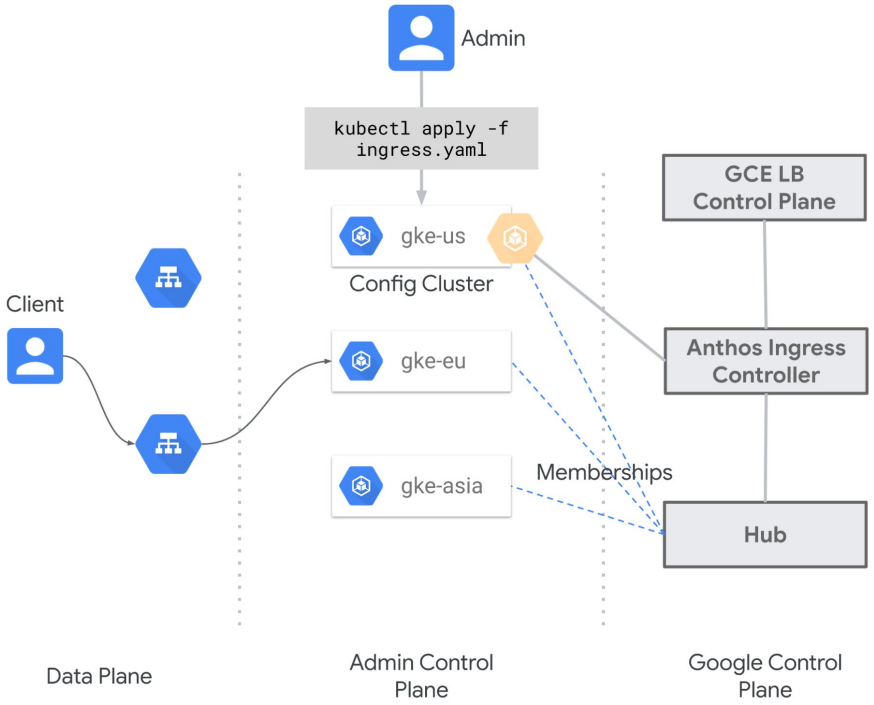
MCI is a native GKE feature that allows you to **expose services deployed in multiple clusters** using a single external IP address and load balancer. MCI is a good choice for applications that need to be accessible to external clients, such as web applications and APIs.

MCI is relatively **easy to set up and use**, and it is **well-integrated with other GKE features**. However, MCI does not support all of the features of a full-fledged service mesh, such as traffic splitting and fault tolerance.

How Multi Cluster Ingress works

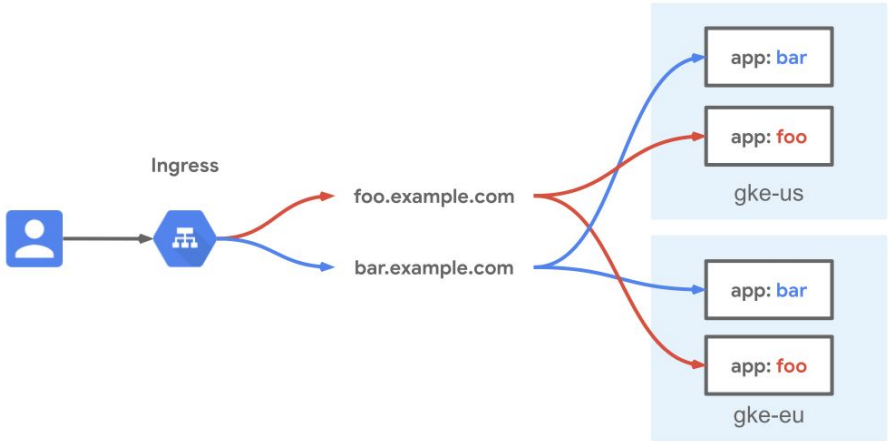


Multi Cluster Ingress architecture



MCI Configuration

```yaml
apiVersion: networking.gke.io/v1
kind: MultiClusterIngress
metadata:
  name: foobar-ingress
  namespace: blue
spec:
  template:
    spec:
      backend:
        serviceName: default-backend
        servicePort: 80
      rules:
      - host: foo.example.com
        backend:
          serviceName: foo
          servicePort: 80
      - host: bar.example.com
        backend:
          serviceName: bar
          servicePort: 80
```



Google Developer Groups

GDG Depok

Multi-Cluster Gateway

Multi-Cluster Gateway

Multi Cluster Gateway (MCG) is a Google Kubernetes Engine (GKE) feature that allows you to **manage traffic across multiple Kubernetes clusters**. It is a global service that can be used to **expose services deployed in multiple clusters** to external clients. MCG is based on the GKE Gateway controller, but it provides additional features for managing traffic across multiple clusters.

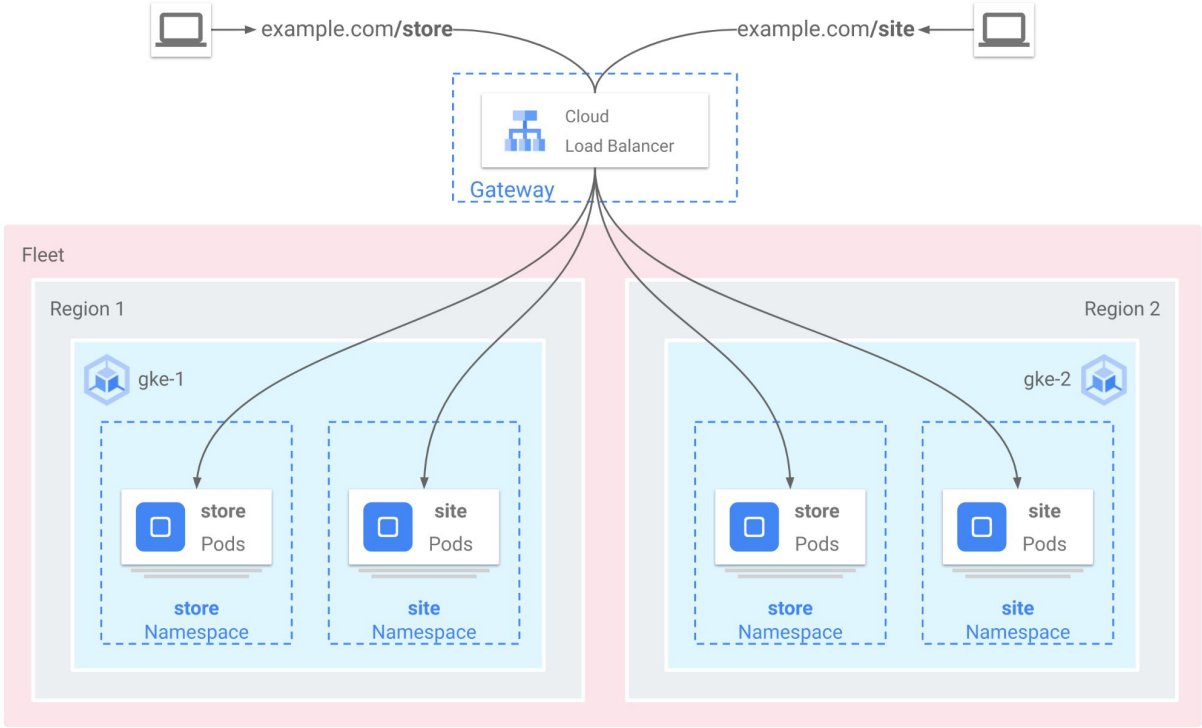
Containers & Kubernetes

Ingress traffic to your GKE fleet with the Multi-cluster Gateway controller, now GA

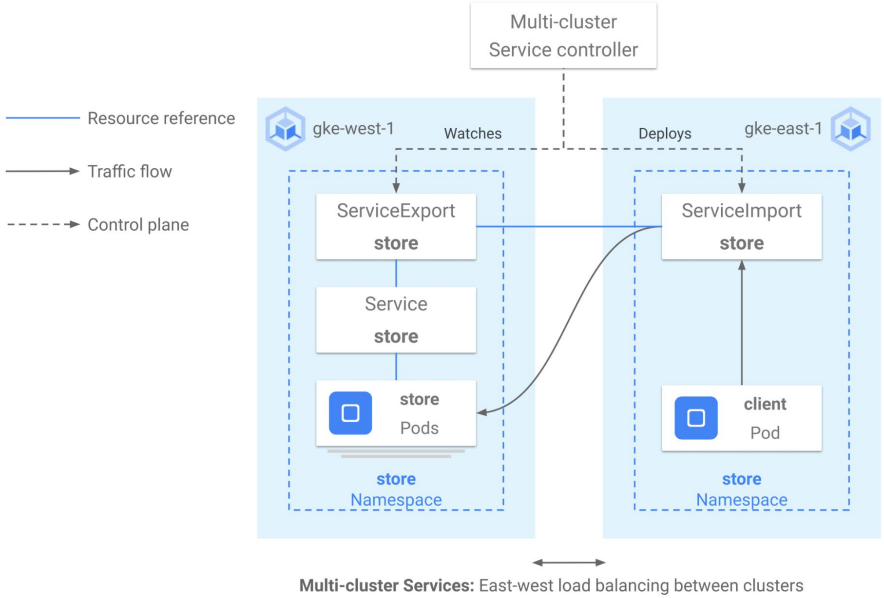
November 1, 2023

https://cloud.google.com/blog/products/containers-kubernetes/multi-cluster-gateway-controller-for-gke-is-now-ga

Multi-Cluster Gateway Example



Example architecture



Multi-cluster Services: East-west load balancing between clusters

Enable the required APIs

```
gcloud services enable \
  trafficdirector.googleapis.com \
  multiclusterservicediscovery.googleapis.com \
  multiclusteringress.googleapis.com \
  --project=PROJECT_ID
```

(example) HTTPRoute resource to forward traffic to a group of backend Pods that run across one or more clusters

```
kind: HTTPRoute
apiVersion: gateway.networking.k8s.io/v1beta1
metadata:
  name: store-route
  namespace: store
  labels:
    gateway: multi-cluster-gateway
spec:
  parentRefs:
  - kind: Gateway
    namespace: store
    name: external-http
  hostnames:
  - "store.example.com"
  rules:
  - backendRefs:
    - group: net.gke.io
      kind: ServiceImport
      name: store
      port: 8080
```

Google Developer Groups

GDG Depok

Anthos Service Mesh

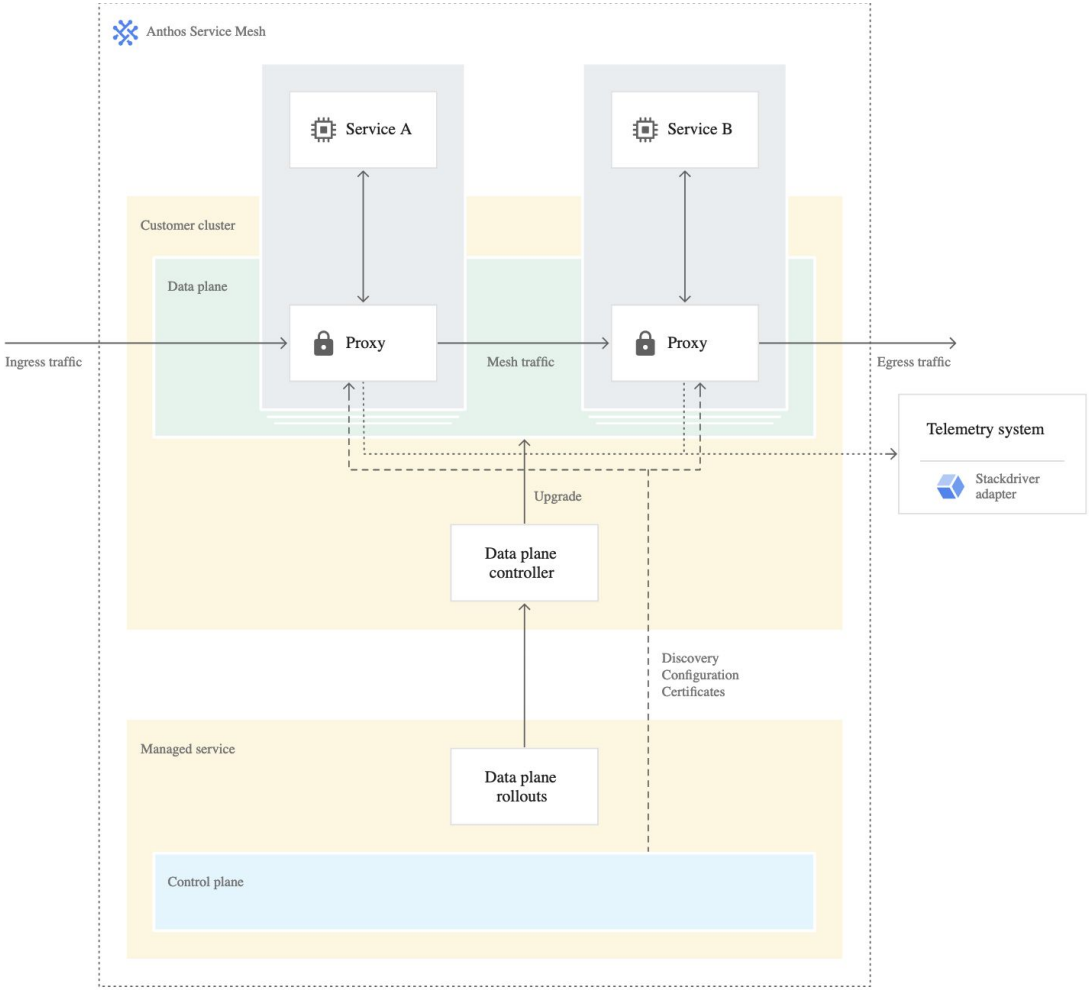
Antos Service Mesh

Anthos Service Mesh (ASM) is a managed service mesh built on top of Istio that helps you manage, monitor, and secure microservices architectures.

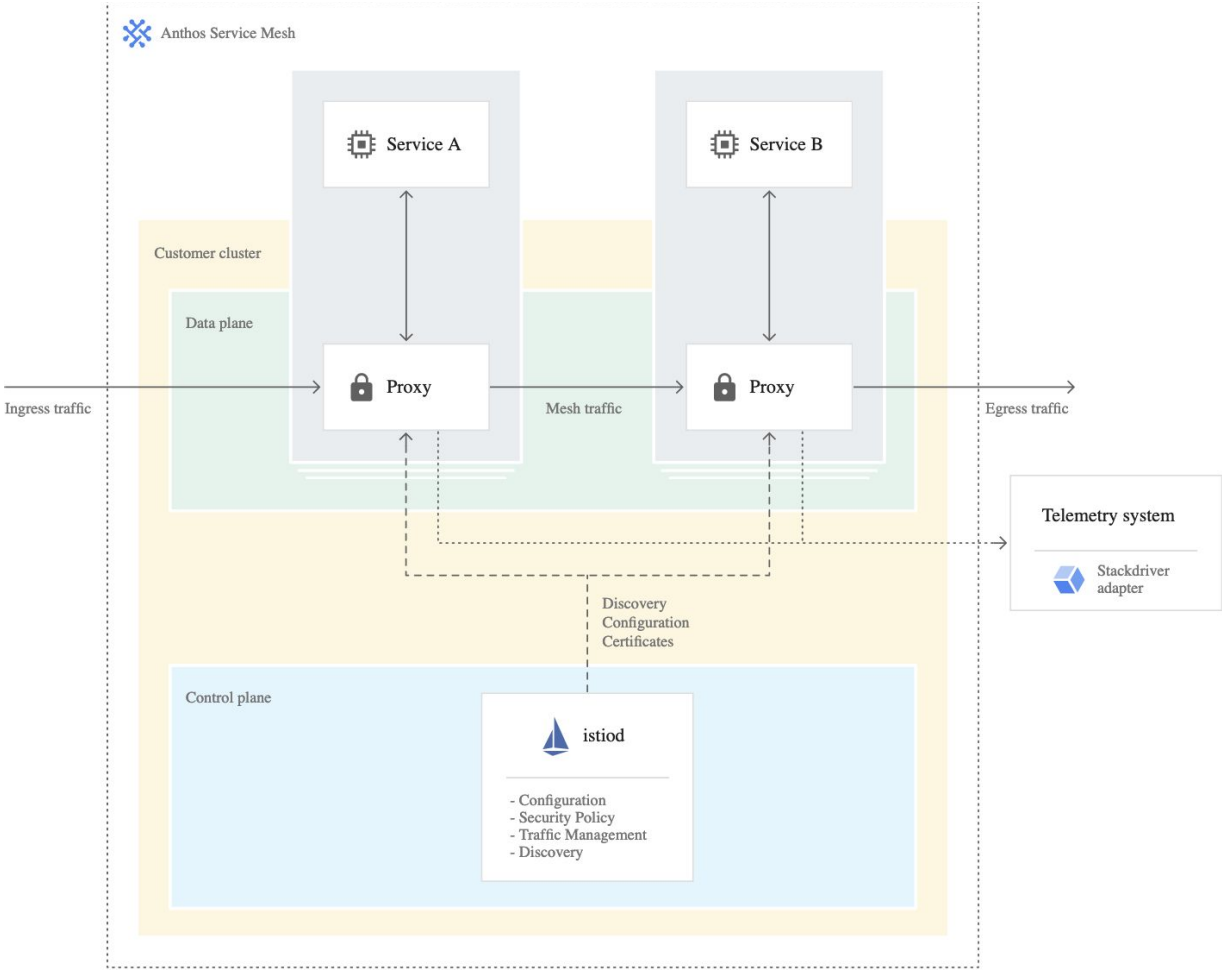
ASM is available in two deployment options:

- **Managed Anthos Service Mesh:** This is the simplest and most recommended deployment option. ASM provisions and manages a dedicated control plane for your mesh. You only need to install the ASM agent on your workloads.
- **In-cluster Anthos Service Mesh**: This deployment option allows you to run ASM on your own Kubernetes clusters. This gives you more control over the deployment and management of ASM, but it also requires more effort to set up and maintain.

Managed Anthos Service Mesh



In-cluster Control Plane



ASM Features

- **Service discovery and load balancing**: ASM automatically discovers all services in your mesh and load balances traffic between them.
- **Traffic management:** ASM allows you to route traffic between your services in a variety of ways, including based on path, header, or load balancing policy.
- **Security**: ASM provides a number of security features, including mutual TLS encryption, service identity, and traffic authorization.
- **Observability**: ASM provides a rich set of observability features, including tracing, monitoring, and logging.

Enable the Anthos Service Mesh fleet feature

```
gcloud container fleet mesh enable --project FLEET_PROJECT_ID
```

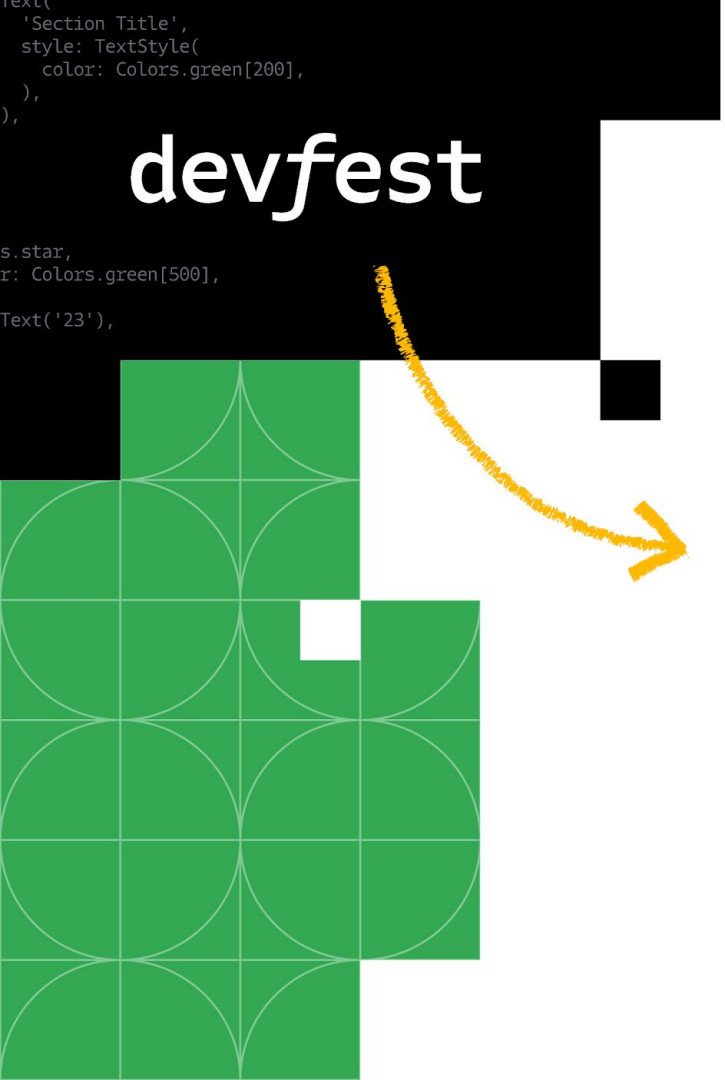
Create an operator manifest for the egress gateway

```
cat << EOF > egressgateway-operator.yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  name: egressgateway-operator
  annotations:
    config.kubernetes.io/local-config: "true"
spec:
  profile: empty
  revision: REVISION
  components:
    egressGateways:
    - name: istio-egressgateway
      namespace: istio-egress
      enabled: true
  values:
    gateways:
      istio-egressgateway:
        injectionTemplate: gateway
        tolerations:
          - key: "dedicated"
            operator: "Equal"
            value: "gateway"
        nodeSelector:
          cloud.google.com/gke-nodepool: "gateway"
EOF
```

Apply a default PeerAuthentication policy for the mesh

```
cat <<EOF | kubectl apply -f -
apiVersion: "security.istio.io/v1beta1"
kind: "PeerAuthentication"
metadata:
  name: "default"
  namespace: "istio-system"
spec:
  mtls:
    mode: STRICT
EOF
```

Google Developer Groups



Google Developer Groups

GDG Depok

Multi-Container Services (MCS)

Multi-Container Services (MCS)

MCS is a feature of GKE that allows you to **expose services deployed in multiple clusters** to external clients using a **single external IP address and load balancer**.

MCS is similar to MCI, but it provides **some additional features**, such as the ability to expose stateful services and the ability to route traffic to specific clusters based on various factors, such as user location or service version.

Enable the MCS, fleet (hub), Resource Manager, Traffic Director, and Cloud DNS APIs

```
gcloud services enable \
    multiclusterservicediscovery.googleapis.com \
    gkehub.googleapis.com \
    cloudresourcemanager.googleapis.com \
    trafficdirector.googleapis.com \
    dns.googleapis.com \
    --project=PROJECT_ID
```

Enabling MCS on your GKE cluster

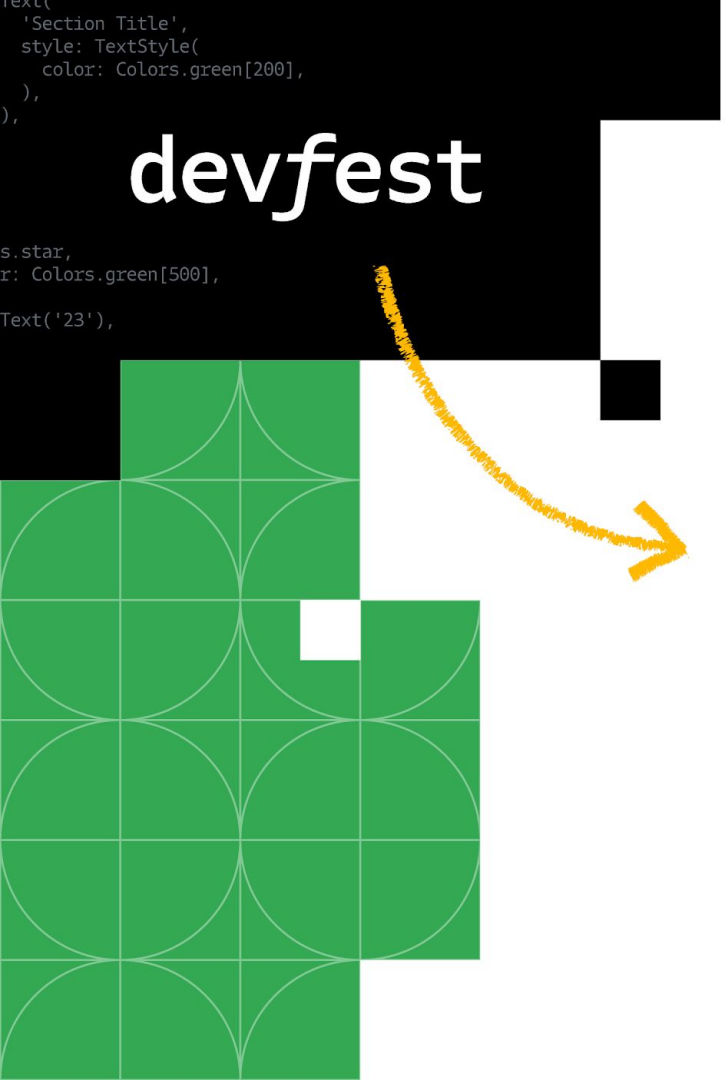
```
gcloud container fleet multi-cluster-services enable \
    --project PROJECT_ID
```

Registering a Service for export

```
# export.yaml
kind: ServiceExport
apiVersion: net.gke.io/v1
metadata:
 namespace: NAMESPACE
 name: SERVICE_EXPORT_NAME
```

Consuming cross-cluster Services

```
kind: ServiceImport
apiVersion: net.gke.io/v1
metadata:
 namespace: EXPORTED-SERVICE-NAMESPACE
 name: external-svc-SERVICE-EXPORT-TARGET
status:
 ports:
 - name: https
   port: 443
   protocol: TCP
   targetPort: 443
 ips: CLUSTER_SET_IP
```

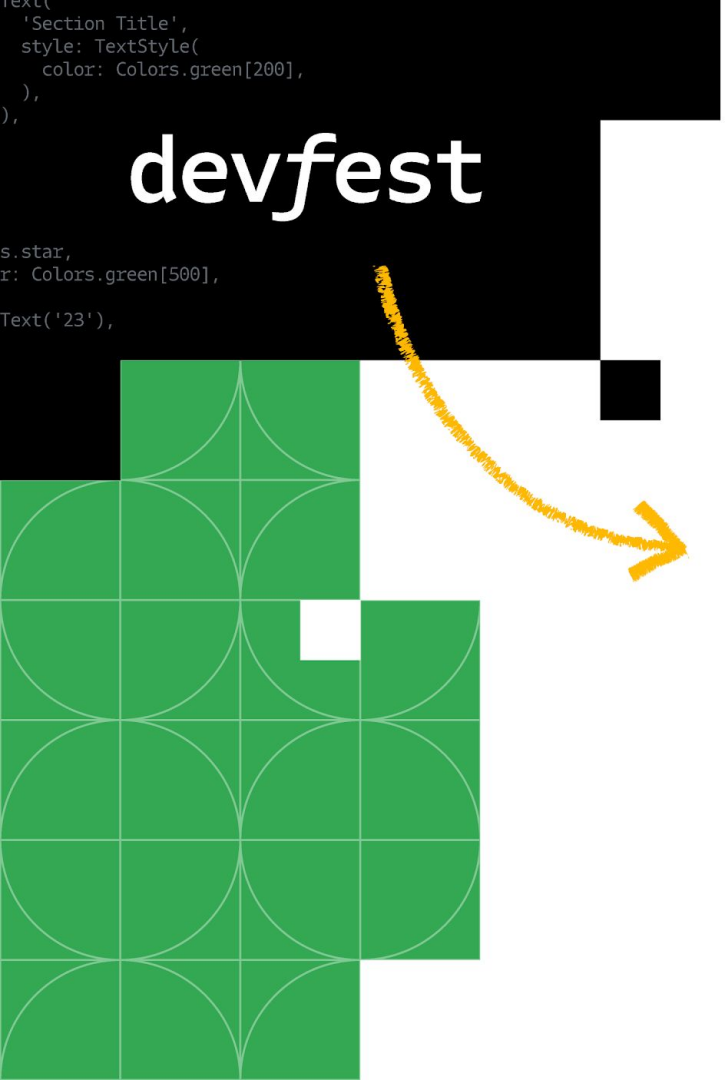


Google Developer Groups

GDG Depok

Key Takeways

	NEGs	MCI	MCG	ASM	MCS
Maturity	Production	Production	Production	Production	Production
Ease of use	Easy	Easy	Medium	Hard	Hard
Features	Basic load balancing	Basic load balancing and ingress	Advanced traffic management	Advanced traffic management, security, and observability	Advanced traffic management, stateful services, and traffic routing
Integration with other GKE features	Good	Good	Limited	Limited	Limited
Best use cases	Exposing applications to external clients	Exposing applications to external clients	Applications that require a high level of reliability and availability	Applications that require a high level of reliability, availability, and security	Highly available applications that need to be accessible to external clients and exposed to multiple clusters

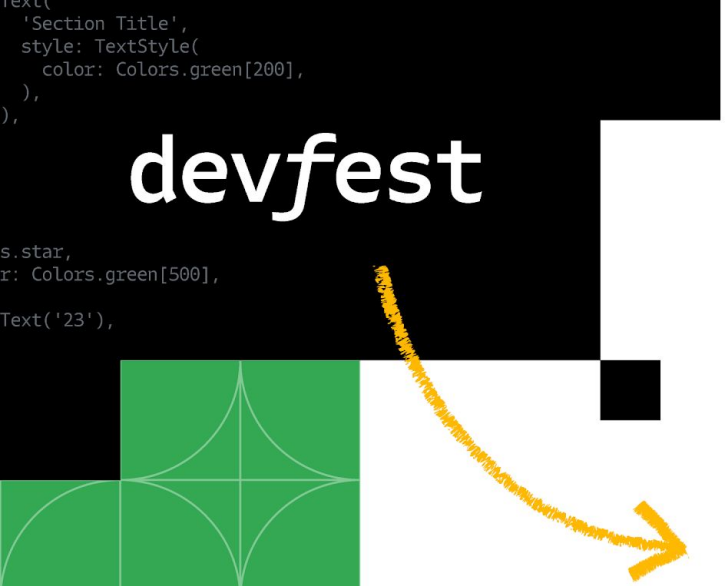


Google Developer Groups

GDG Depok

Demo

Antosh Service Mesh:
https://s.id/anthosmeshguide01



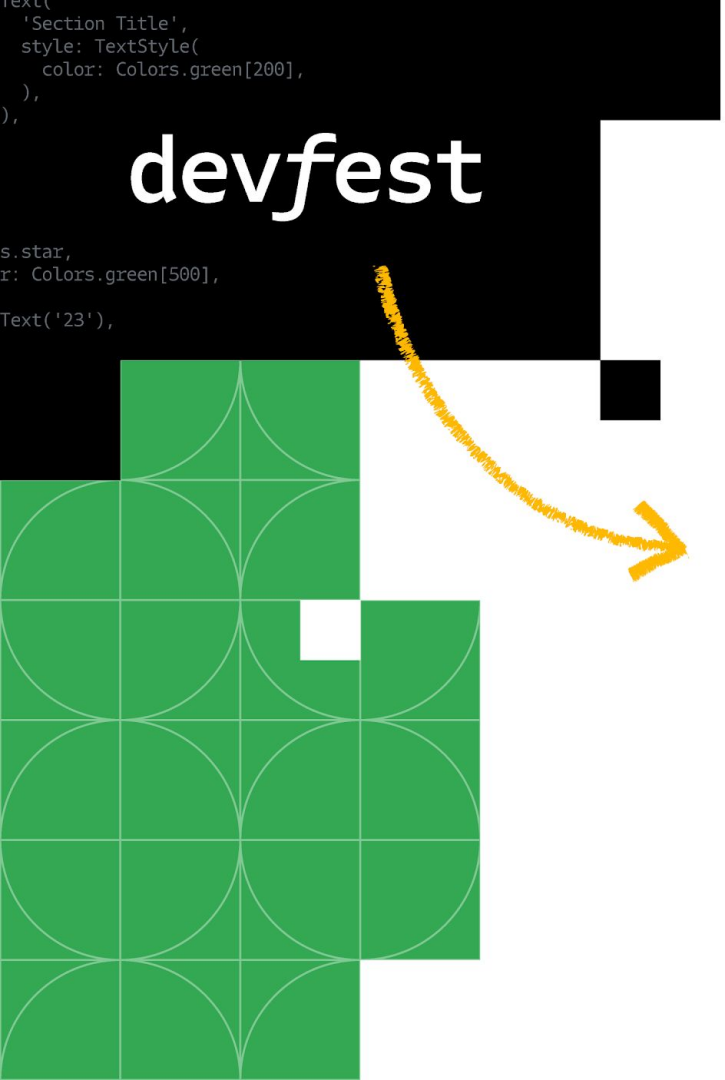
Google Developer Groups

GDG Depok

References

References

1. https://cloud.google.com/load-balancing/docs/negs/
2. https://cloud.google.com/kubernetes-engine/docs/concepts/multi-cluster-ingress
3. https://cloud.google.com/kubernetes-engine/docs/how-to/enabling-multi-cluster-gateways
4. https://cloud.google.com/kubernetes-engine/docs/how-to/deploying-multi-cluster-gateways
5. https://cloud.google.com/kubernetes-engine/docs/how-to/multi-cluster-services
6. https://cloud.google.com/blog/products/containers-kubernetes/multi-cluster-gateway-controller-for-gke-is-now-ga



Google Developer Groups

GDG Depok

Thank you!