
Giant Swarm
At Your Service, Mesh
26/08/2021
KUBERNETES

Giant Swarm

At Your Service, Mesh



Simon Weald
SRE

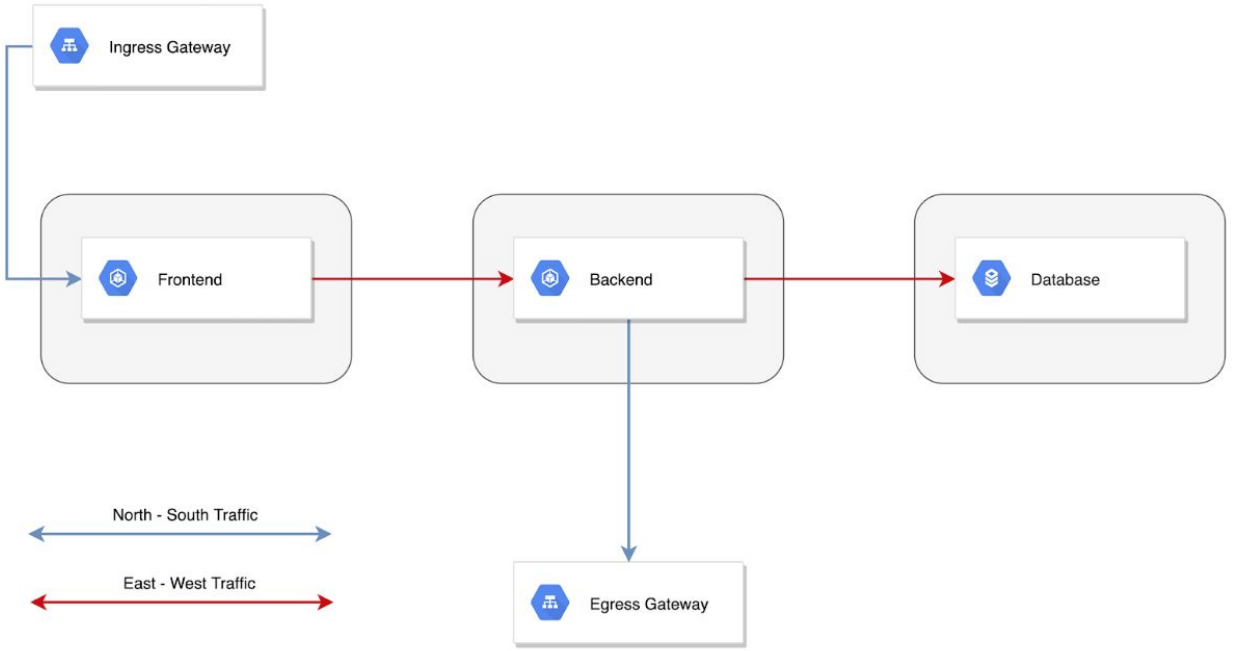

EXCUSE ME SIR,
DO YOU HAVE A MOMENT
TO TALK ABOUT OUR LORD
AND SAVIOUR, THE SERVICE MESH?
imgflip.com

Agenda

- What is a service mesh?
- The hype is real; but do you *need* it?
- How can a service mesh help you?
- A quick comparison: Linkerd vs Istio
- Demo time


So what exactly *is* a service mesh?
KUBERNETES

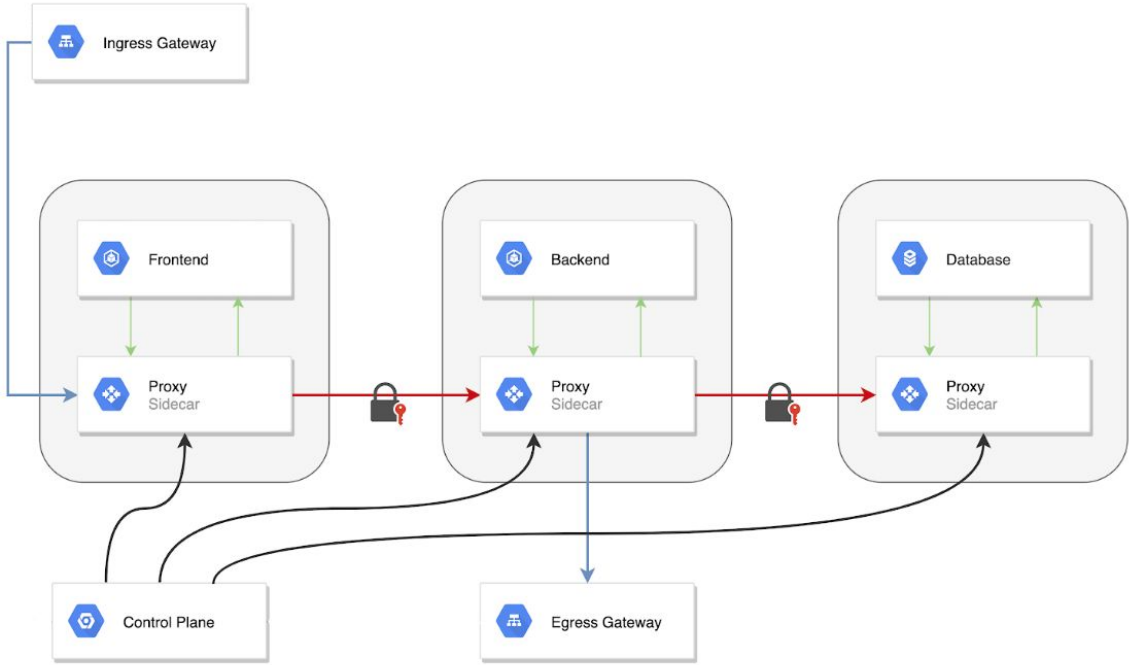
BM (Before Mesh)



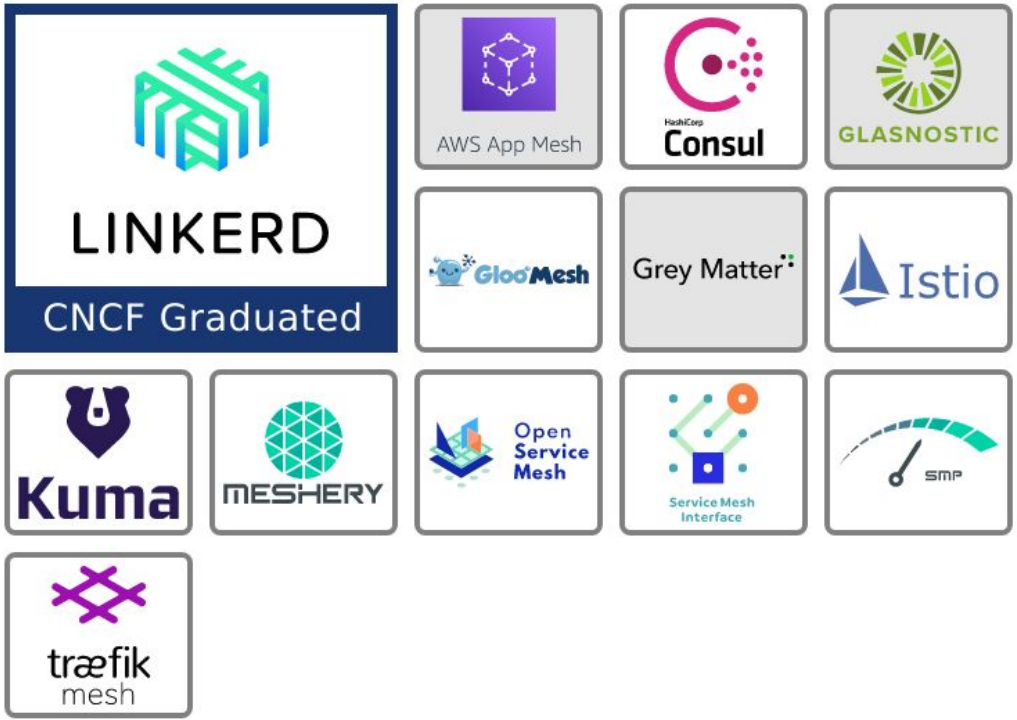
AM (After Mesh)

Terminology

Control plane
Data plane
Proxy
Sidecar



Mesh providers



Do you need a service mesh?



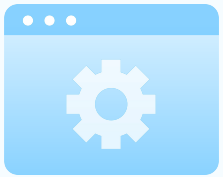
Yes.



To mesh, or not to mesh; that is the question.

(William Shakespeare, 1602)

A typical Kubernetes journey



Initial deployment

My First Kubernetes



Day 2 Operations

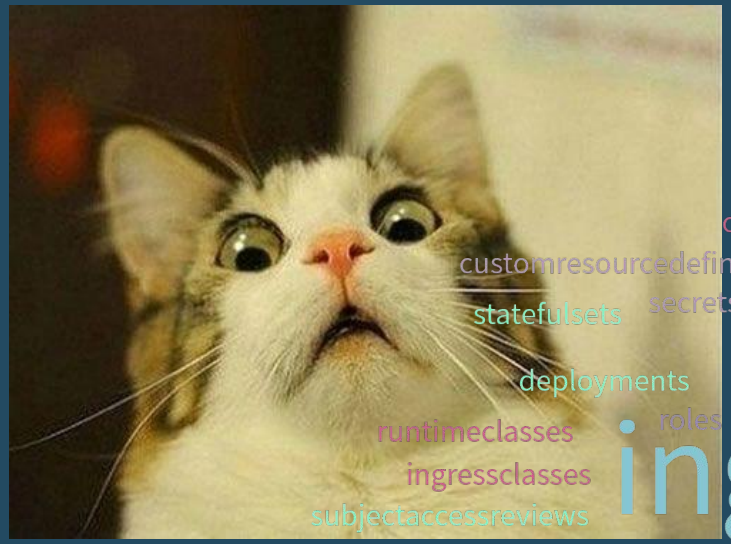
Kubernetes upgrades, scaling, monitoring with Prometheus, deploying workloads according to best practices



Augmentation

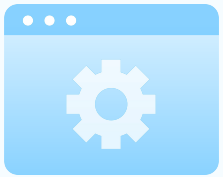
CI/CD pipelines, serverless, self-service application catalogs, GitOps


volumeattachments
clusterrolebindings
certificatesigningrequests
customresourcedefinitions
apiservices
mutatingwebhookconfigurations
statefulsets
secrets
persistentvolumes
pods
networkpolicies
resourcequotas
deployments
controllerrevisions
endpointslices
namespaces
csidrivers
runtimeclasses
roles
services
serviceaccounts
daemonsets
ingresses
leases
storageclasses
ingressclasses
subjectaccessreviews
priorityclasses
replicasets
configmaps
poddisruptionbudgets
podtemplates
endpoints
csinodes
horizontalpodautoscalers
bindings
replicationcontrollers
events
persistentvolumeclaims
podsecuritypolicies
cronjobs
limitranges
localsubjectaccessreviews
flowschemas
jobs
rolebindings
nodes
clusterroles
prioritylevelconfigurations
tokenreviews
csistoragecapacities
validatingwebhookconfigurations
componentstatuses
selfsubjectaccessreviews
selfsubjectrulesreviews



volumeattachments
clusterrolebindings
certificatesigningrequests
customresourcedefinitions
apiservices
mutatingwebhookconfigurations
secrets
pods
networkpolicies
statefulsets
persistentvolumes
resourcequotas
deployments
controllerrevisions
endpointslices
namespaces
csidrivers
roles
services
serviceaccounts
daemonsets
runtimeclasses
ingresses
leases
storageclasses
ingressclasses
priorityclasses
subjectaccessreviews
replicasets
configmaps
poddisruptionbudgets
podtemplates
endpoints
csinodes
horizontalpodautoscalers
bindings
replicationcontrollers
events
persistentvolumeclaims
podsecuritypolicies
cronjobs
flowschemas
limitranges
localsubjectaccessreviews
jobs
rolebindings
nodes
clusterroles
prioritylevelconfigurations
tokenreviews
csistoragecapacities
validatingwebhookconfigurations
componentstatuses
selfsubjectaccessreviews
selfsubjectrulesreviews

A typical Kubernetes journey



Initial deployment

My First Kubernetes



Day 2 Operations

Kubernetes upgrades, scaling, monitoring with Prometheus, deploying workloads according to best practices



Augmentation

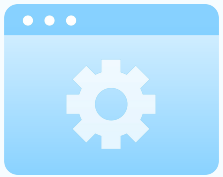
CI/CD pipelines, serverless, self-service application catalogs, GitOps



Day 2: smashed it 💪



A typical Kubernetes journey



Initial deployment

My First Kubernetes



Day 2 Operations

Kubernetes upgrades, scaling, monitoring with Prometheus, deploying workloads according to best practices



Augmentation

CI/CD pipelines, serverless, self-service application catalogs, GitOps



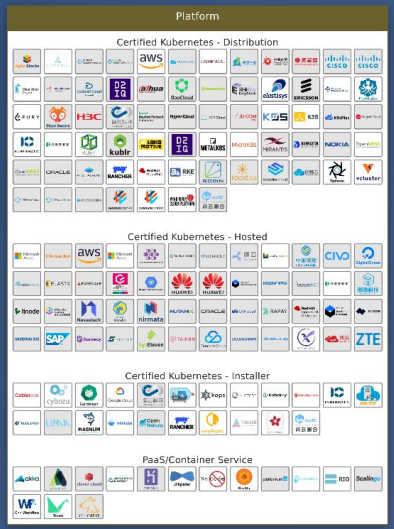


CNCF Cloud Native Landscape

2021-08-21T08:40:13Z bdc4ccdd

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io







Giant Swarm

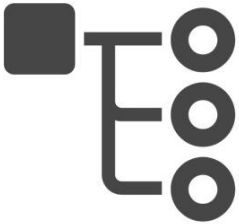
What can a service mesh do for you?



The Observability Trifecta

https://swarm.ly/writing-minimal-services

Tracing



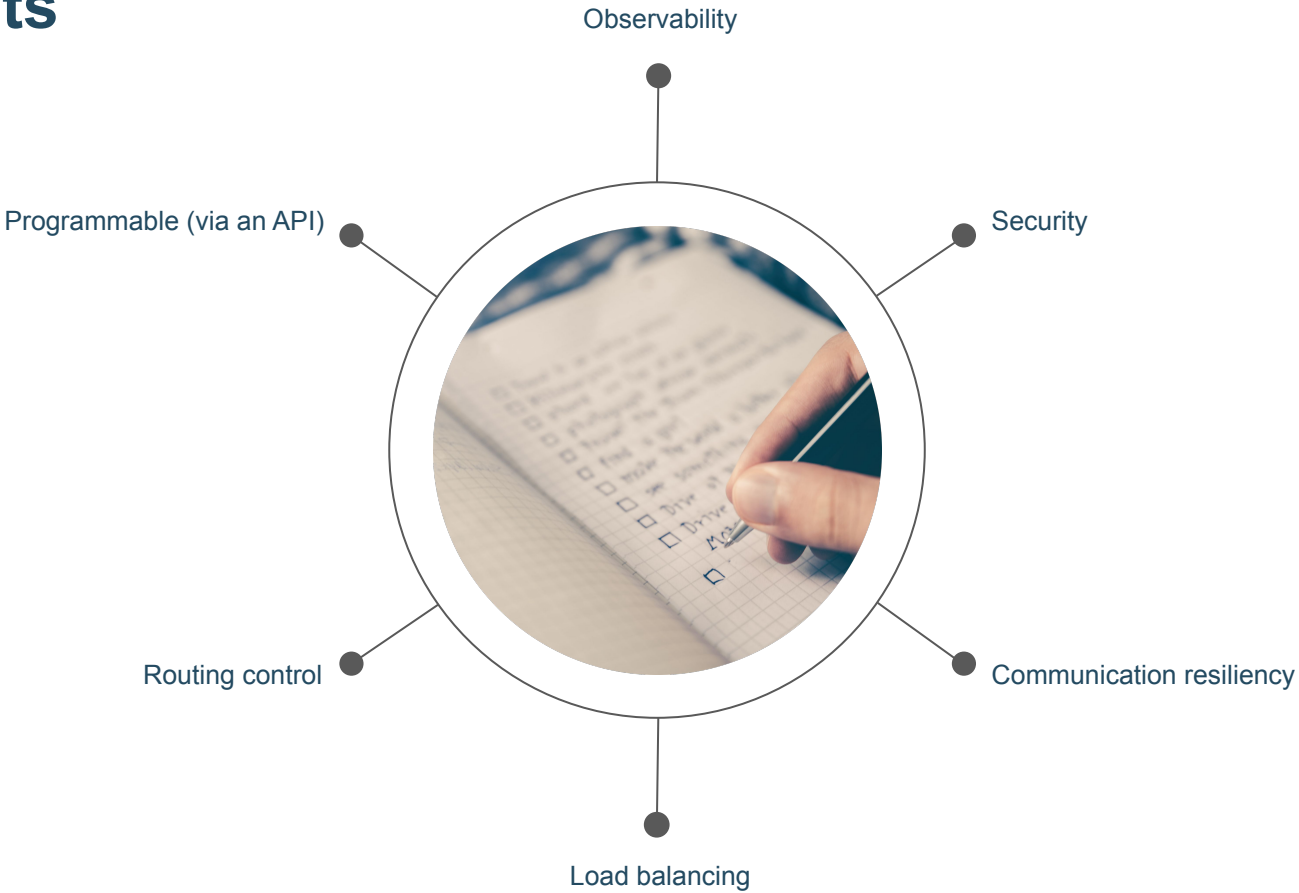
Logging



Metrics



Highlights


Observability
Programmable (via an API)
Security
Routing control
Communication resiliency
Load balancing

Big Corp: a case study



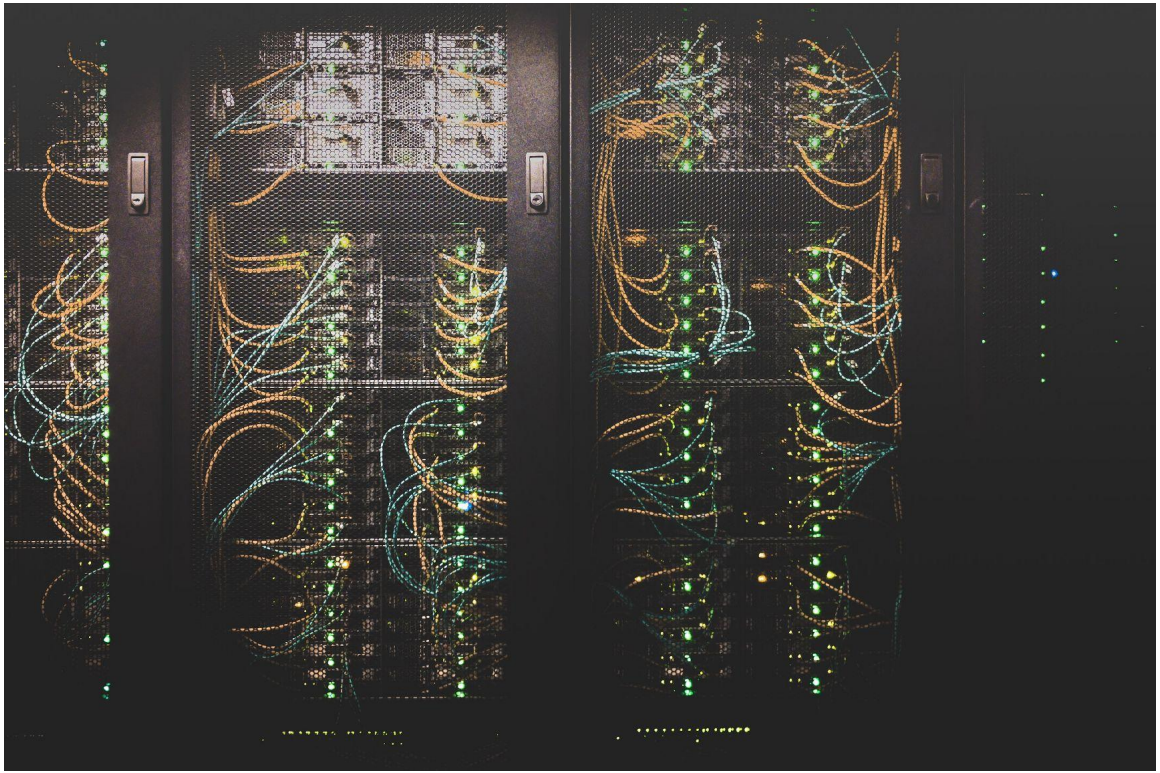
Big Corp



Developers



Platform team



Security team



All that glitters is not gold



- Increased resource usage (financial cost)
- Increase in manpower required to manage it (human cost)
- Adds more network hops (latency)
- Another component to have to investigate when troubleshooting
- Designed for use with the microservice architecture pattern

Linkerd vs Istio



Fight!



A brief comparison

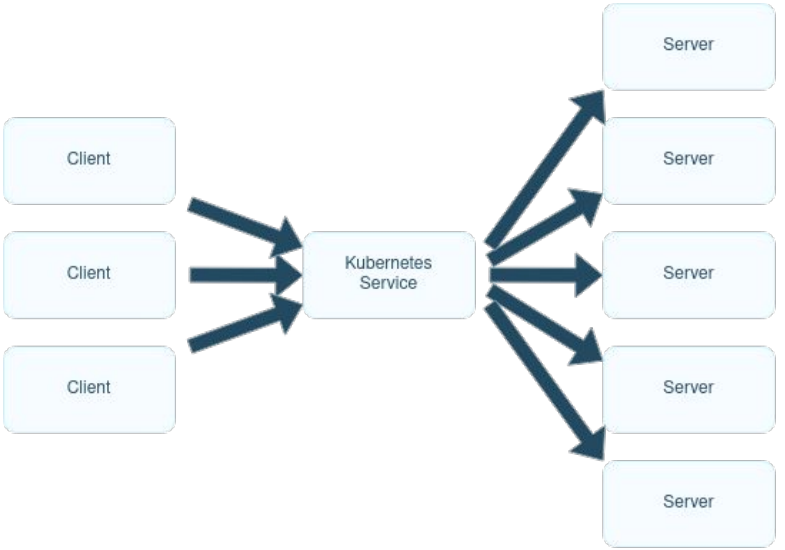
	Istio	Linkerd
Ease of installation		✓
Resource usage		✓
Platforms	✓	
Multi-cluster support	✓	✓
Network resiliency	✓	
Security	✓	
Observability		✓
Enterprise support	✓	✓

Demo time



Demo details

- Very basic; < 100 lines of Go
- Typical microservice architecture
- Client makes HTTP calls
- All pods in a deployment are the same
- Server pods are occasionally faulty



Thanks to the demo gods / the demo gods were unkind

(delete as appropriate)




IT WAS AT THIS MOMENT
SIMON REALISED
HE'D RUN OUT OF MEMES
imgflip.com

Wrap-up

- Any organisation using microservices will benefit from a mesh.
- Meshes are typically application-aware and therefore they compliment your existing metrics - they don’t replace them.
- They provide vastly more flexibility when it comes to networking and traffic management than vanilla Kubernetes.
- You get mTLS between all your services (east-west traffic).
- It will introduce more complexity and resource requirements (both in terms of compute and man hours).


Giant Swarm
Thank you!
@glitchcrab
simonweald@giantswarm.io
KUBERNETES