

# KubeCon '19 Barcelona

Daniel-Andrei Minca

Site Reliability Engineer @ GfK SE

20-23 May '19

# Outline

1. D1 - Intro
2. D2 - Deep-dive
3. D3 - Wrap-up
4. Demo: 'kind' (Kubernetes in Docker)<sup>1</sup>

---

<sup>1</sup>pray to the demo gods

## Beer?



Figure: Alhambra - mana potion

# State of CNCF projects

2

## 1. Incubating:

- ▶ Linkerd
- ▶ Helm v3.0.0-alpha1 w/o Tiller
- ▶ Rook.io
- ▶ Harbor etc.

## 2. Graduated:

- ▶ fluentd

## Top players at KubeCon '19

- ▶ Google Cloud Platform
- ▶ Amazon Web Services
- ▶ WeaveWorks



Figure: Huawei was also there

## Loki (by Grafana)<sup>3</sup>

- ▶ logs: groups them into 'streams' & indexes them with labels
- ▶ LogQL allows filter chaining
- ▶ extract labels from logs
- ▶ live tailing (demo failed during KubeCon pres.)



---

<sup>3</sup><https://youtu.be/CQiawXlgabQ>

## Linkerd<sup>4</sup>

1. service mesh: sidecar, data-plane etc.
2. auto golden-metrics
3. transparent mTLS



---

<sup>4</sup><https://youtu.be/Z3nfLI3z0hc>

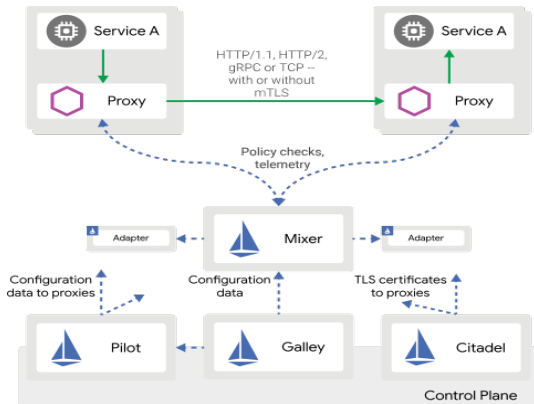
## Kubernetes Dashboard (by Grafana)<sup>5</sup>

1. DMM (Dashboard Maturity Model)
2. template variables (use more)
3. Use scripting libraries to generate dashboards (mixins)
  - ▶ grafonnet (Jsonnet)
  - ▶ grafanalib (Python)

---

<sup>4</sup><https://youtu.be/YE2aQFiMGfY>



Istio<sup>6</sup>

1. Grants: mTLS, ACL, traffic auditing, passthrough mTLS, auto LB
2. Conn's multiple net's through Gateways

<sup>5</sup><https://youtu.be/-t2BfT59zJA>

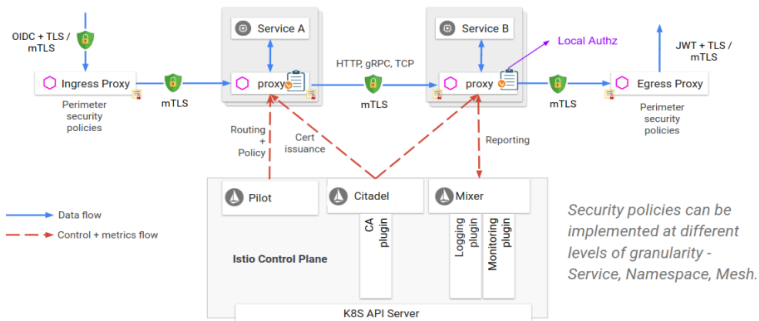


Figure: Istio mTLS (security)

## CI/CD with 'kind'<sup>7</sup>

1. e2e tests:
  - ▶ start full k8s Cluster
  - ▶ black-box Testing
  - ▶ 'kind create cluster'
2. exposes full kubeadm config surface
3. enable Alpha features, Network Driver etc.

```
1 # this config file contains all config fields with comments
2 kind: Cluster
3 apiVersion: kind.sigs.k8s.io/v1alpha3
4 # patch the generated kubeadm config with some extra settings
5 kubeadmConfigPatches:
6 - |
7   apiVersion: kubeadm.k8s.io/v1beta1
8   kind: ClusterConfiguration
9   metadata:
10     name: config
11   networking:
12     serviceSubnet: 10.0.0.0/16
13 # patch it further using a JSON 6902 patch
14 kubeadmConfigPatchesJson6902:
15 - group: kubeadm.k8s.io
16   version: v1beta1
17   kind: ClusterConfiguration
18   patch: |
19     - op: add
20       path: /apiServer/certSANs/-
21       value: my-hostname
```

Figure: sample 'kind' config file

---

<sup>3</sup><https://youtu.be/8KtmevMFfxA>

1. examples of kind running on XYZ CI
2. other ways to run kind:
  - ▶ as a library

```
// create a cluster context
ctx := cluster.NewContext("my-test-cluster")
// defer tearing down the cluster
defer ctx.Delete()

// create the cluster
if err := ctx.Create(&config.Config{}); err != nil {
    t.Errorf("Failed to create cluster: %v", err)
    t.FailNow()
}

// your tests here
```

DEMO: kind



## Refs & links

- ▶ [https://static.sched.com/hosted\\_files/kccnceu19/31/Grafana%20Loki.pdf](https://static.sched.com/hosted_files/kccnceu19/31/Grafana%20Loki.pdf)
- ▶ [https://static.sched.com/hosted\\_files/kccnceu19/e8/Intro%20to%20Linkerd%20KCCNCEU19.pdf](https://static.sched.com/hosted_files/kccnceu19/e8/Intro%20to%20Linkerd%20KCCNCEU19.pdf)
- ▶ [https://static.sched.com/hosted\\_files/kccnceu19/27/Kubecon%202019\\_%20Kubernetes%20dashboards%20final.pdf](https://static.sched.com/hosted_files/kccnceu19/27/Kubecon%202019_%20Kubernetes%20dashboards%20final.pdf)
- ▶ <https://www.youtube.com/playlist?list=PLj6h78yzYM2PpmMAnvpvsnr4c27wJePh3>