Istio

and why do we need it

(blowing up clusters with style)
Speaking DevOps

- Cooperation in a cross-functional team
- Lower the entry barriers to technology
- Declarativeness
- Easy to start, yet hard to master
The tale of Kubernetes
Deploying a workload

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: our-nginx
spec:
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14
        ports:
          - containerPort: 80
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: our-nginx
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14
        ports:
          - containerPort: 80
Exposing

User or a Happy Computer

Nginx Service

Nginx:1.14
Nginx:1.14
Nginx:1.14

---

kind: Deployment
metadata:
  name: our-nginx
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 3
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
      - name: nginx
        image: nginx:1.14
        ports:
          - containerPort: 80

kind: Service
metadata:
  name: our-nginx-service
spec:
  selector:
    app: nginx
  ports:
    - port: 80
Mounting volumes

- Secrets such as certificates
- Configurations
One of the pods suddenly fails

- Node failure
- Pod delete
- Probe unsuccess
Kubernetes replaces this pod

- Pod is replaced to match the desired count
What about the advanced scenarios?

A/B testing
Example of Istio configuration

Goals:

- Create Subsets for 2 versions
- Split traffic based on a HTTP header
- Accept traffic from the Internet
Istio Destination Rule

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: our-nginx-dr
spec:
  host: our-nginx-service.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: LEAST_CONN
```

Diagram:
- **Destination Rule Global Subset**
- **Nginx Service**
- **Nginx:1.14** (version=v1)
- **Nginx:1.14** (version=v1)
- **Nginx:1.15** (version=v2)
Istio Destination Rule (Subsets)

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: our-nginx-dr
spec:
  host: our-nginx-service.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: LEAST_CONN
  subsets:
  - name: stable-version
    labels:
      version: v1
  - name: new-version
    labels:
      version: v2
```

Diagram:
- Nginx Service
- Destination Rule Global Subset
  - Subset v1
    - Nginx:1.14 (version=v1)
  - Subset v2
    - Nginx:1.15 (version=v2)
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews-route
spec:
  hosts:
  - our-nginx-service.prod.svc.cluster.local
gateways:
  - our-gateway
http:
  - match:
    - headers:
      service-version:
        exact: v2
    route:
      - destination:
        host: our-nginx-service.prod.svc.cluster.local
        subset: v2
  - route:
    - destination:
      host: our-nginx-service.prod.svc.cluster.local
      subset: v1
```yaml
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: our-gateway
spec:
  selector:
    istio: ingressgateway
  servers:
    - port:
        number: 80
        name: http
        protocol: HTTP
      hosts:
        - 'example.com'
```
Istio Summary

- Allows continuous adoption
- Provides declarative networking API
- Components
  - Gateway (L4-L6 entrypoint)
  - Virtual Service (L4-L7 router)
  - Destination Rule (service traffic policies)

Note: it’s hard to master, really.
Learn more at ...

- https://istio.io/
- https://istio.io/docs/concepts/traffic-management/
Thank you!

Peter Malina
CTO @FlowUp
Socials: @petomalina