# Building Platforms with Istio

**Murugappan Sevugan Chetty**

**IstioCon - 2021**

Feb 26, 2021

OPTUM®

# About Me

## Murugappan Sevugan Chetty

**Opensource contributor and Principal Engineer @ Optum**

**Twitter : @itsmurugappan**
**Github : @itsmurugappan**

# In this talk …

# Platform Intro

# What we set out to build

**On-Premises**, **Multitenant** and **fully managed Serverless platform** with…

#1 Simple UX

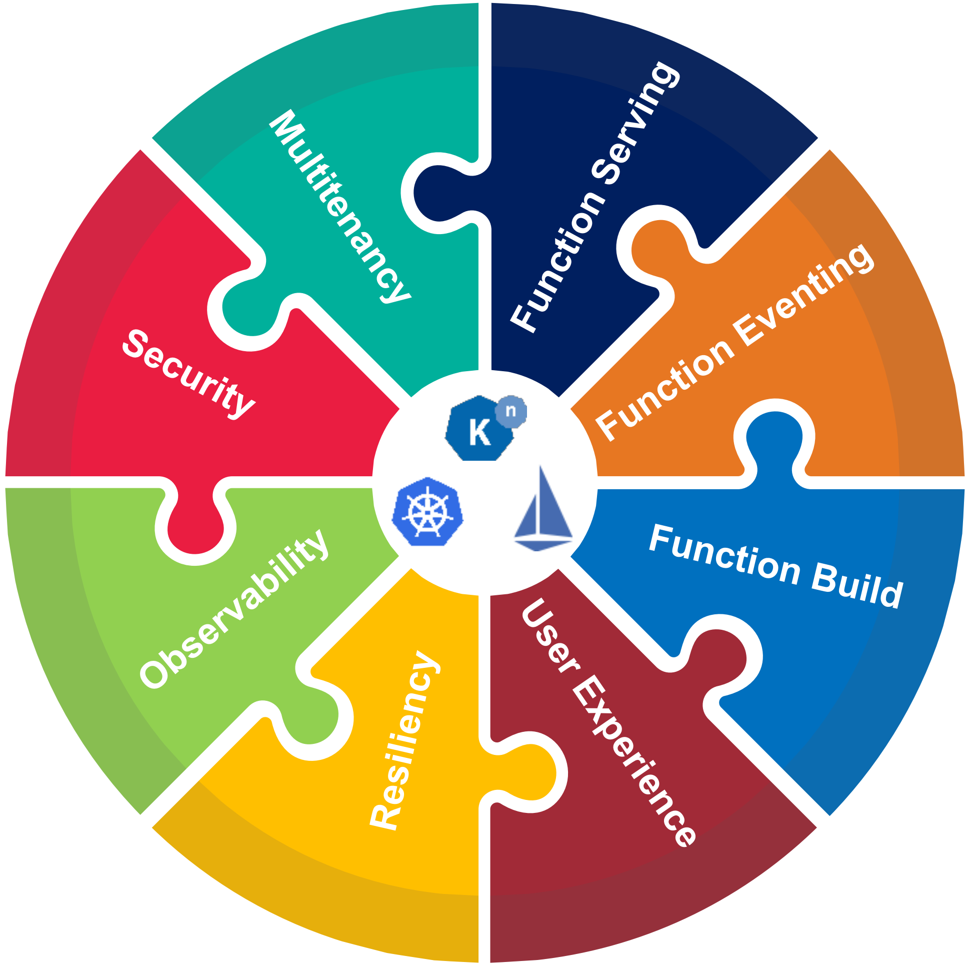
#2 Low barrier of entry

#3 Handle cross cutting concerns like Authz/Authn

#4 Resilient

#5 End to End Observability

# Platform Capabilities



# Technology picks

OPTUM®

# Technology – Capability Matrix

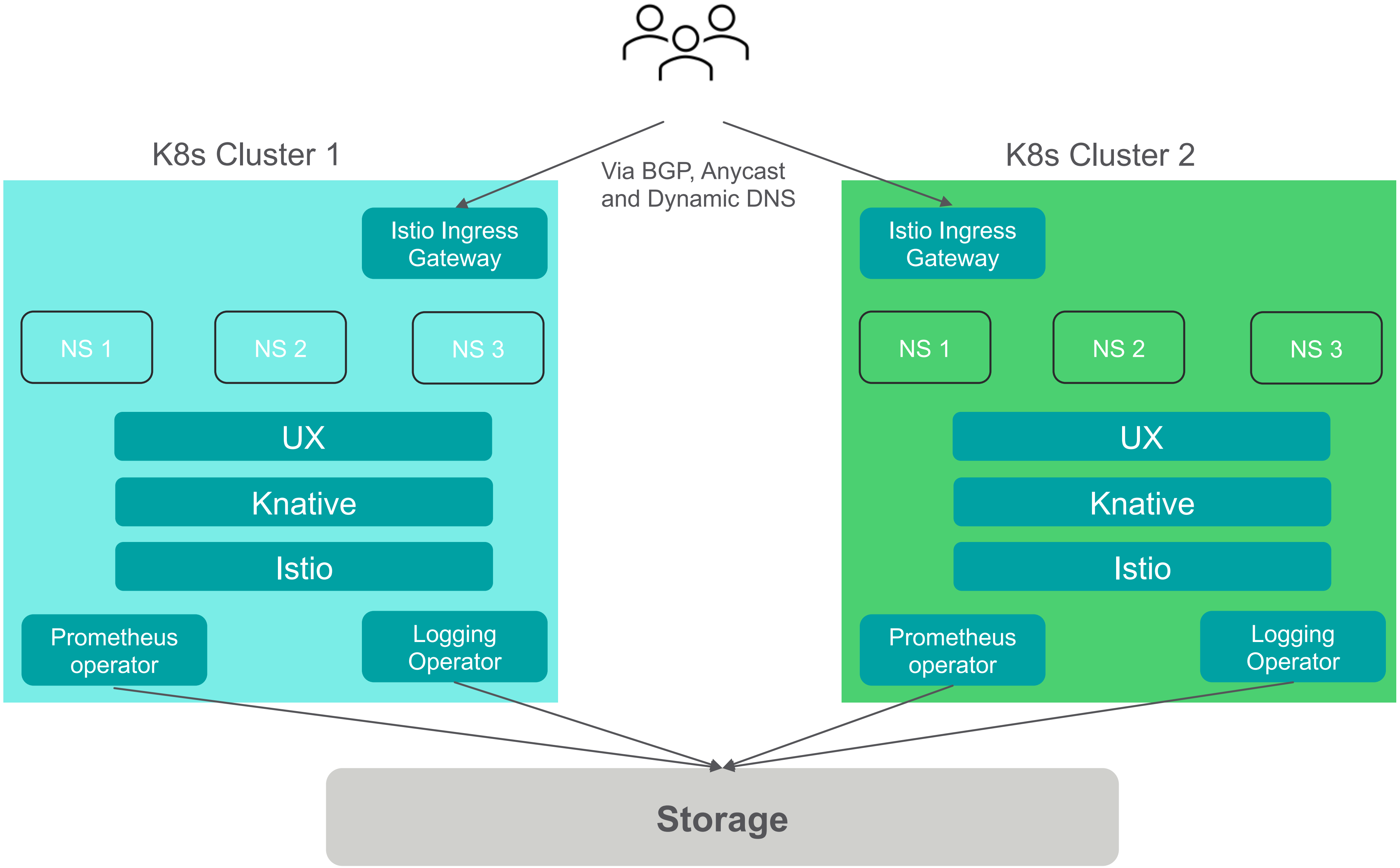
| | Function - Serving | Function - Build | Function - Eventing | Multitenancy |
|---|---|---|---|---|
| Kubernetes | | | | ✓ |
| Istio | | | | ✓ |
| Knative | ✓ | | ✓ | ✓ |
| Tekton | | ✓ | | ✓ |
| Prometheus | | | | ✓ |
| Grafana | | | | ✓ |
| Banzai Cloud Logging Operator | | | | ✓ |
| In-house Tools and UX | | ✓ | | ✓ |

# Technology – Capability Matrix

| | Security | Observability | Resiliency | UX |
|---|---|---|---|---|
| Kubernetes | ✓ | ✓ | ✓ | |
| Istio | ✓ | ✓ | ✓ | |
| Knative | | ✓ | ✓ | |
| Tekton | | ✓ | | |
| Prometheus | | ✓ | | |
| Grafana | | ✓ | | |
| Banzai Cloud Logging Operator | | ✓ | | |
| In-house Tools and UX | | ✓ | ✓ | ✓ |

# How they stack up ?

# Platform Architecture



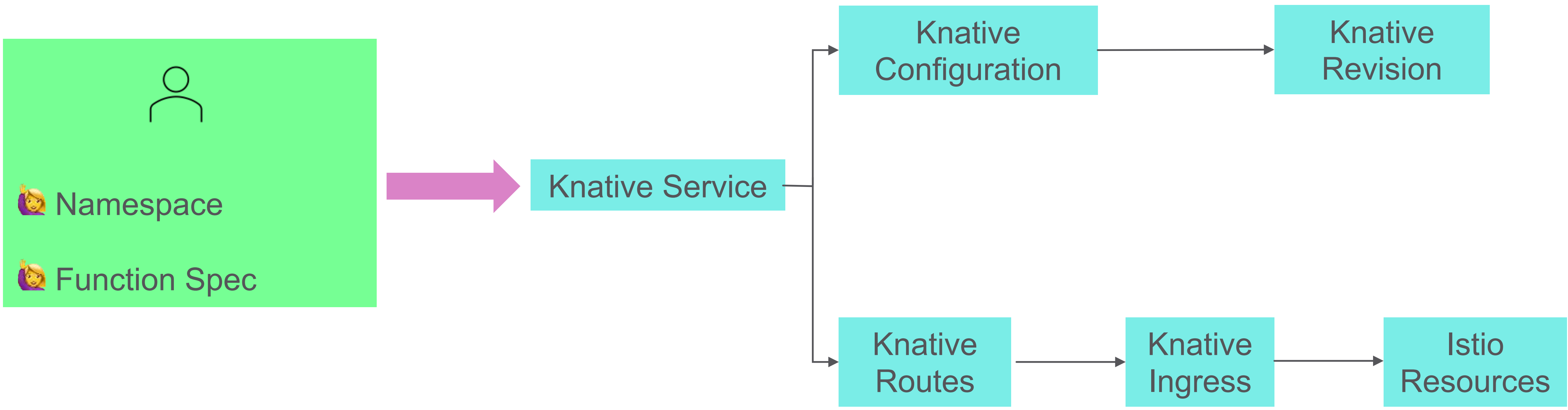
OPTUM®

# Istio in Action

# Knative and Istio

Knative requires a L7 load balancer for

- Routing the traffic to the correct Knative service
- Split traffic
- per-request load balancing between Pods



# Knative Routes

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: java-knative-svc-test-ingress
  namespace: test-alpha
  ownerReferences:
  - apiVersion: networking.internal.knative.dev/v1alpha1
    blockOwnerDeletion: true
    controller: true
    kind: Ingress
    name: java-knative-svc-test
spec:
  gateways:
  - knative-serving/knative-ingress-gateway
  - knative-serving/knative-local-gateway
  hosts:
  - java-knative-svc-test.test-alpha
  - java-knative-svc-test.test-alpha.svc
  - java-knative-svc-test.test-alpha.svc.cluster.local
```

# Knative Traffic Splitting

```
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
  name: hello
  namespace: test-alpha
spec:
  template:
      name: hello-v2
    spec:
      containers:
      - image: docker.repo1.uhc.com/seset/java-knative
  traffic:
  - latestRevision: true
    percent: 50
  - latestRevision: false
    percent: 50
    revisionName: hello-skblr-1
    tag: staging
```

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: hello-ingress
  namespace: test-alpha
hosts:
  - hello.test-alpha
  - hello.test-alpha.svc
  - hello.test-alpha.svc.cluster.local
  - staging-hello.test-alpha
  - staging-hello.test-alpha.svc
  - staging-hello.test-alpha.svc.cluster.local
  http:
    route:
    - destination:
        host: hello-v2.test-alpha.svc.cluster.local
        port:
          number: 80
      headers:
        request:
          set:
            Knative-Serving-Namespace: test-alpha
            Knative-Serving-Revision: hello-v2
      weight: 50
    - destination:
        host: hello-skblr-1.test-alpha.svc.cluster.local
        port:
          number: 80
      headers:
        request:
          set:
            Knative-Serving-Namespace: test-alpha
            Knative-Serving-Revision: hello-skblr-1
      weight: 50
```

# Knative TLS

```yaml
apiVersion: networking.istio.io/v1beta1
kind: Gateway
metadata:
  name: hello-3797421420
  namespace: test-alpha
  ownerReferences:
  - apiVersion: networking.internal.knative.dev/v1alpha1
    blockOwnerDeletion: true
    controller: true
    kind: Ingress
    name: hello
spec:
  selector:
    app: istio-ingressgateway
    istio: ingressgateway
  servers:
  - hosts:
    - hello-test-alpha.domain.com
    port:
      name: test-alpha/hello:0
      number: 443
      protocol: HTTPS
    tls:
      credentialName: hello-3c74709c-0a4f-4aba-a346-3e555a3ce0ee
      mode: SIMPLE
      privateKey: tls.key
      serverCertificate: tls.crt
```

```yaml
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: route-5b6d939d-a52d-47d4-9912-bc9267dca0ef-199426798
  namespace: test-alpha
  ownerReferences:
  - apiVersion: networking.internal.knative.dev/v1alpha1
    blockOwnerDeletion: true
    controller: true
    kind: Certificate
    name: route-5b6d939d-a52d-47d4-9912-bc9267dca0ef-199426798
spec:
  commonName: hello-test-alpha
  dnsNames:
  - hello-test-alpha.domain.com
  issuerRef:
    kind: ClusterIssuer
    name: tpp-venafi-issuer
  secretName: route-5b6d939d-a52d-47d4-9912-bc9267dca0ef-199426798
```

# Authz & Authn

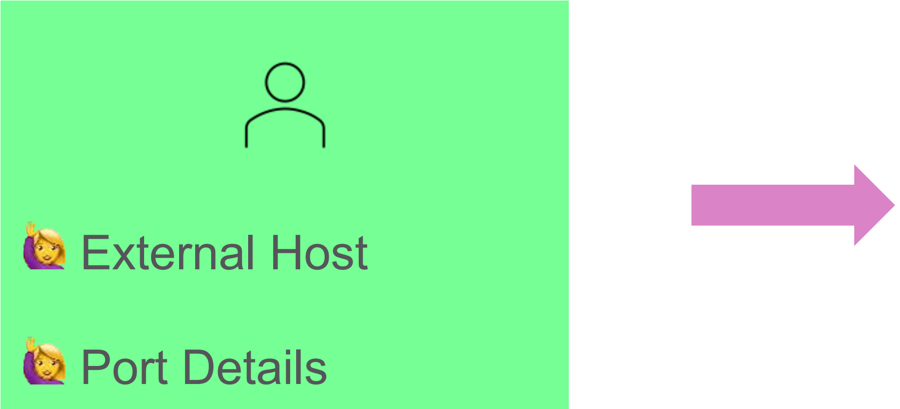
- 🙋 Namespace
- 🙋 Function Name
- 🙋 JWT Claim

```
apiVersion: security.istio.io/v1beta1
kind: RequestAuthentication
metadata:
  name: java-knative-svc-policy
spec:
  jwtRules:
  - issuer: https://issuer-url
    jwksUri: http://issuer-certs-url
  selector:
    matchLabels:
      serving.knative.dev/service: java-knative-svc
```

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: java-knative-svc-authz-policy
spec:
  rules:
  - when:
    - key: request.auth.claims[groups]
      values:
      - secure-group
  selector:
    matchLabels:
      serving.knative.dev/service: java-knative-svc
```

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: java-knative-svc-def-authz-policy
spec:
  rules:
  - to:
    - operation:
        paths:
        - /_internal/knative/activator/probe
        - /metrics
        - /healthz
        - /_/health
  selector:
    matchLabels:
      serving.knative.dev/service: java-knative-svc
```

# External Service Registration

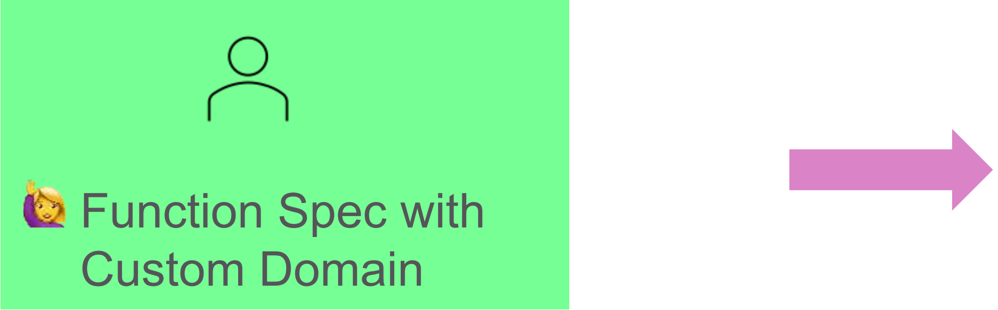


🙋 External Host

🙋 Port Details

```
apiVersion: networking.istio.io/v1beta1
kind: ServiceEntry
metadata:
  name: service-entry-request
spec:
  hosts:
  - mesh-external.host.com
  location: MESH_EXTERNAL
  ports:
  - name: http
    number: 80
    protocol: HTTP
  resolution: DNS
```

# Custom Domains



🙋 Function Spec with Custom Domain

Configure Knative to create Knative service url with custom domain

Update Knative ingress gateway to accept this host

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    external-dns.alpha.kubernetes.io/hostname: gateway.custom.domain.com
  labels:
    app: istio-custom-gateway
    istio: istio-custom-gateway
    release: istio
  name: istio-custom-gateway-svc
  namespace: istio-system
spec:
  ports:
  - name: http2
    port: 80
    targetPort: 8080
  - name: https
    port: 443
    targetPort: 8443
  selector:
    app: istio-ingressgateway
    istio: ingressgateway
  type: LoadBalancer
```

# And..

End to End Tracing

Metering

Kiali

Istio CNI

- In House – UX – Only point of interaction for users
  - Opinionated
  - Low barrier for entry
  - Eliminate bad actors
  - Handle deprecations and version upgrades
    - Service role/bindings to request auth/auth policy

OPTUM®

# Managing Istio

OPTUM®

# Istioctl

- **Installation**

```
$ istioctl manifest generate -f istio-template.yaml > istio.yaml
```

- **Proxy status**

```
$ istioctl proxy-status
```

- **Dashboard**

```
$ istioctl dashboard envoy
```

# Closing thoughts

# Summary

- Right set of tools for building and managing platform
    - Istioctl
    - Istio Go Client
- Active community
- Start small

OPTUM®