Who am I?

Kamesh Sampath
Director of Developer Experience at Red Hat

- Active Open Source Contributor
  - OpenWhisk
  - Eclipse Che


kamesh.sampath@hotmail.com
kameshsampath
@kamesh_sampath
bit.ly/istio-tutorial
What is a microservice?

The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API.

These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

Martin Fowler
Microservices Introductory Materials

Video Training: [bit.ly/microservicesvideo](bit.ly/microservicesvideo)

Kubernetes for Java Developers

Microservices Advanced Materials

[bit.ly/mono2microdb](bit.ly/mono2microdb)

Istio.io

@kamesh_sampath
Short History of Microservices

- Continuous Integration via XP: 1999
- Agile Manifesto: Feb 2001
- DevOps: 2009
- Java EE6: 2009
- DropWizard: May 2011
- Hystrix: March 2012
- Eureka: July 2012
- Spring Boot: Sept 2013
- Kubernetes: June 2014

- AWS EC2: 2006
- NETFLIX: to AWS 2010
- Ribbon: March 2012
- Microservices Access Thoughtworks Radar March 2012
- Docker: March 2013
- Microservices Defined Thoughtworks Fowler, Lewis March 2014

NETFLIX OSS
Microservices == Distributed Computing
Distributed Computing == Network of Services

@kamesh_sampath
Microservices'ilities

MyService

API
Tracing
Discovery
Invocation
Elasticity
Resilience
Pipeline
Authentication
Logging
Monitoring
Some Action!!!!
Microservices'ilities + Kubernetes
Microservices'ilities + OpenShift

- MyService
- Monitoring
- Tracing
- Discovery
- Invocation
- Authentication
- Logging
- Resilience
- Pipeline
- Elasticity
Microservice (Yes) Pain Points

● Discovery
● Distributed Tracing
● Circuit Breakers
● Metrics and Monitoring
● Operational Requirements
  ○ A/B Testing
  ○ Canary Release
  ○ Rate Limiting
  ○ Access Policies
What's Wrong with Netflix OSS?

Java Only

Adds a lot of libraries to YOUR code
Istio - Sail

(Kubernetes - Helmsman or ship’s pilot)
A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It’s responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as an array of lightweight network proxies that are deployed alongside application code, without the application needing to be aware of this infrastructure layer.

Microservices'ilities + Istio
Microservices embedding Capabilities

Before Istio

@kamesh_sampath
Microservices externalizing Capabilities

After Istio

@kamesh_sampath
SideCars

- Two or more containers deployed to same pod
- Share
  - same namespace
  - same Pod IP
  - Shared lifecycle
- Used to enhance the containers

Source: [http://blog.kubernetes.io/2015/06/the-distributed-system-toolkit-patterns.html](http://blog.kubernetes.io/2015/06/the-distributed-system-toolkit-patterns.html)
Kubernetes, Istio, Envoy

Pod
Container
JVM
Service A
Sidecar Container

Pod
Container
JVM
Service B
Sidecar Container

Pod
Container
JVM
Service C
Sidecar Container

@kamesh_sampath
Istio Control Plane

Pod
Container
JVM
Service A
Envoy Side-car

HTTP1.1, HTTP2, gRPC, TCP w/TLS

Pod
Container
JVM
Service B
Envoy Side-car

HTTP1.1, HTTP2, gRPC, TCP w/TLS

Pod
Container
JVM
Service C
Envoy Side-car

HTTP1.1, HTTP2, gRPC, TCP w/TLS

Istio Pilot
istioctl, API, config

Istio Mixer
Quota, Telemetry, Rate Limiting, ACL

Istio Auth
CA, SPIFFE

@kamesh_sampath
- annotations:
  alpha.istio.io/sidecar: injected
  alpha.istio.io/version: jenkins@ubuntu-16-04-build
  pod.beta.kubernetes.io/init-containers: '["args":
    "--kernel.core_pattern=/tmp/core.%e,%p,%t\u0020"
  ]'
- creationTimestamp: null

- labels:
  app: details
  version: v1
- spec:
  containers:
  - image: istio/examples-bookinfo-details-v1
    imagePullPolicy: IfNotPresent
    name: details
    ports:
      - containerPort: 9080
    resources: {}
    args:
      - proxy
      - sidecar
      - "-v"
      - "-u"
    env:
      - name: POD_NAME
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
      - name: POD_NAMESPACE
        valueFrom:
          fieldRef:
            fieldPath: metadata.namespace
      - name: POD_IP
        valueFrom:
          fieldRef:
            fieldPath: status.podIP
    image: docker.io/istio/proxy_debug:0.1
    imagePullPolicy: Always
Canary Deployment
Canary Deployment
Canary Deployment
Canary Deployment
Canary Deployment
Canary Deployment
Canary Deployment
Canary Deployment
Canary Deployment
Canary Deployment

- SCM
- DEVELOPMENT
- QA
- STAGING
- PRODUCTION
- ROUTER
- USERS
Canaries with Kubernetes

![Diagram showing a route/ingress to a service with two pods, each containing a JVM and a service. The service A version 1 is on the left with 50% traffic, and service A version 2 is on the right with 50% traffic.](image-url)
Canaries with Istio

- Route/Ingress
- Service
- Pod
  - Container
  - JVM
  - Service A v1 (90%)
- Pod
  - Container
  - JVM
  - Service A v2 (10%)
Next Generation Microservices - Service Mesh

Code Independent (Polyglot)

- Intelligent Routing and Load-Balancing
  - A/B Tests
  - Smarter Canary Releases
- Chaos: Fault Injection
- Resilience: Circuit Breakers
- Observability: Metrics and Tracing
- Fleet wide policy enforcement
bit.ly/istio-tutorial
Free eBooks from developers.redhat.com
Microservices Introductory materials
Demo: bit.ly/msa-instructions
Slides: bit.ly/microservicesdeepdive
Video Training: bit.ly/microservicesvideo

Microservices Advanced Materials
bit.ly/istio-tutorial
bit.ly/mono2microdb

@kamesh_sampath
Some more Action!!!!
GREAT INDIAN DEVELOPER SUMMIT 2019

Conference: April 23-26, Bangalore

Register early and get the best discounts!

www.developersummit.com  @greatindiandev  bit.ly/gidslinkedln