Microservices, Service Mesh, and CI/CD Pipelines - Making it All Work Together

Brian Redmond, Microsoft
Who am I?

Brian Redmond

• Cloud Architect @ Microsoft (17 years)
• Azure Global Black Belt Team
• Live in Pittsburgh, PA, USA
• Avid marathon runner and outdoors enthusiast
• World traveler

@chzbrgr71
DO WHAT IS GREAT.
Microservices

The answer to all problems.
The answer to all problems.
Kubernetes

The answer to all problems.
I DON'T ALWAYS TEST MY CODE

BUT WHEN I DO, I DO IT IN PRODUCTION
KEEP CALM AND CROSS YOUR FINGERS
Deployment Strategies

- Container Native Development
- Continuous Integration
- Continuous Deployment
- Blue/Green testing
- Canary testing
- A/B testing
Blue / Green Testing
Canary Testing

proxy

80%

v1

100%

v2

@chzbrgr71
A / B Testing

proxy

Live Traffic

Duplicate for analysis

v1

v2
Microservices Deployment
What's missing?
advanced routing
observability
chaos testing
Enter Istio

- Service discovery and routing
- Health checking
- Observability
- Authentication and authorization
- Resiliency
- Policy enforcement
What can I do with Istio?

- advanced routing → route rules, traffic shaping
- observability → metrics, logs, and tracing
- chaos testing → fault injection (delays, faults)
what about a CI/CD tool?
I SEE YOU'RE CREATING A PIPELINE
LET ME HELP YOU WITH THAT
brigade = event-driven scripting for kubernetes

- Encapsulate functions in containers
- Run in parallel or serial
- Trigger workflow from Github, Docker registry, etc.
- Javascript (pipeline as code)
- Project config stored as secrets
- Well suited for CI/CD pipelines
1. Trigger

Gateway

2. GW loads project, creates event.

Event

Controller

3. Controller reads event, creates worker

Worker

4. Worker runs script, managing jobs

Kubernetes APIs

Project

5. API query handled by API server inspecting Kubernetes

Jobs

API
Demo time!

@chzbrgr71
Demo App Architecture

Azure Kubernetes Service

- api-earthquakes
- api-fracking
- api-weather
- api-flights
- api-location

web-earth

Azure CosmosDB

@chzbrgr71
Demo - What you will see

- demo-earth
- api-earthquakes
- version: prod
- version: canary
- virtualservice
- Azure Container Registry
- ACR Build
- helm
- slack
- webhook
- Brigade
- Istio
- Grafana
- @chzbrgr71
Summary

• Microservices and containers bring added challenges for deployment/testing
• Testing in production is a good idea (with canary)
• Service Mesh solutions help with:
  – Observability
  – Advanced Routing
  – Chaos Testing
• CI/CD can automate the process
you can find me at:

• Twitter, Github: @chzbrgr71
• brian.redmond@microsoft.com
• Demo code: https://github.com/chzbrgr71/os-summit-tokyo