



Managing Microservices with Istio on OpenShift



Microservicios Meetup



oc whoami

\$ oc whoami
rmarting, jromanmartin
\$ oc describe user rmarting jromanmartin
Name: Jose Roman Martin Gil
Created: 43 years ago
Labels: father, husband, friend, runner, curious, red hatter,
developer (in any order)
Annotations: Principal Middleware Architect @ Red Hat
Identities:

- mailto: rmarting@redhat.com
- GitHub: https://github.com/rmarting
- Twitter: https://twitter.com/jromanmartin

LinkedIn: <u>https://www.linkedin.com/in/jromanmartin/</u>



oc whoami

\$ oc whoami
cvicens, _cvicens, carlosvicens
\$ oc describe user cvicens
Name: Carlos Vicens Alonso
Created: 43 years ago
Labels: father, husband, red hatter, developer, tinkerer (in
any order)
Annotations: Senior Specialist Solution Architect @ Red Hat
Identities:

- mailto: cvicensa@redhat.com
- GitHub: https://github.com/cvicens
- Twitter: https://twitter.com/_cvicens
- LinkedIn: <u>https://www.linkedin.com/in/carlosvicens/</u>
- Medium: <u>https://medium.com/@carlos.vicens.alonso</u>



True story ...any given night

Have you ever woken up at night feeling uneasy and sweaty wondering if your app is a monolith?



_shy_developer



True story ... same night, but later

Now I'm a believer... I love microservices...

I hate monoliths...

What next? How do I move on?

Same _shy_developer, now converted

Little spoiler! (although we hate spoilers!)

We'll break something in a mesh...

We'll walk you through a plausible route to find the root cause of the problem

> ...the butler did it! Sorry I couldn't help it

MICROSERVICES ARCHITECTURE

Microservices Architecture





Microservices Architecture





Distributed Architecture





Eight Fallacies of Distributed Computing

- 1. The network is reliable
- 2. Latency is zero
- 3. Bandwidth is infinite
- 4. The network is secure
- 5. Topology doesn't change
- 6. There is one administrator
- 7. Transport cost is zero
- 8. The network is homogeneous



Source: https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing Photo: Icon made by Freepik from www.flaticon.com



HOW TO DEAL WITH THE COMPLEXITY?

Deployment





Configuration





Service Discovery





Dynamic Routing





Fault Tolerance





Tracing and Visibility





What About ...?





Then?

There should be a BETTER WAY

Mmm maybe ... addressing the complexity WITHIN the infrastructure

SERVICE MESH A Dedicated Network for Service-to-Service Communications

Microservices Evolution





Microservices without Service Mesh





Service Mesh Architecture





OpenShift Service Mesh



MAISTRA (Kubernetes Operator)



Service Mesh Ecosystem





SERVICE MESH FEATURES

FAULT TOLERANCE

Circuit Breaker without Service Mesh



coupled to the service code



Circuit Breaker with Service Mesh



transparent to the services



Circuit Breaker with Service Mesh



improved response time with global circuit status



Timeouts and Retries with Service Mesh



configure timeouts and retries, transparent to the services



Rate Limits with Service Mesh



limit invocation rates, transparent to the services



DYNAMIC ROUTING

Dynamic Routing without Service Mesh



custom code to enable dynamic routing



Canary Deployment with Service Mesh





Dark Launches with Service Mesh





SERVICE SECURITY

Secure Communication without Service Mesh



coupled to the service code



Secure Communication with Service Mesh



mutual TLS authentication, transparent to the services



Control Service Access with Service Mesh



control the service access flow, transparent to the services



CHAOS ENGINEERING

Chaos Engineering without Service Mesh





Chaos Engineering with Service Mesh



inject delays, transparent to the services



Chaos Engineering with Service Mesh



inject protocol-specific errors, transparent to the services



DISTRIBUTED TRACING

Distributed Tracing without Service Mesh



code to enable dynamic tracing



Distributed Tracing with Service Mesh



discovers service relationships and process times, transparent to the services





OBSERVABILITY

Metrics

- Mixer collects metrics from Envoy
- Adapters in the Mixer normalize and forward to monitoring backends
- Metrics backend can be swapped at runtime





KIALI

What are my microservices doing?



Service graph with dynamic traffic visualization





vices > Namespace: coolstore > Service: Inventory > Service Info p Inbound Metrics Traces							Istio Config Namespace Filter by Namespace I1 Active Filters: Namespace coolstore × Clear All Filters		
⊕* inventory							v coolstore-api coolstore	VirtualService	Config: 🔗
Labels Ports app inventory demo coolstore-msa. expose true group com.redhat.cloudnative TCP http https://service-msak-msetime provider fabrics service-msak-msetime provider TCP http			Ports TCP http (8080)	Endpoints 172.17.0.27 : inventory-2-2tj4v 172.17.0.28 : inventory-v2-2-s997x		Health	coolstore-gateway	Gateway	
Type ClusterIP IP 172 30 59 123 Created at 2/4/2019, 4:52:19 PM						Error rate degraded: 0.24%>=0.1%	Coolstore-inventory	VirtualService	Config: 🥑
Resource Version 73	134						(P) coolstore-web	VirtualService	Config: ⊘
Workloads (2) So	ource Workloads (1) Virt	ual Services (2) Destination Rules (1)							
Name	Туре	Labels			Created at	Resource version	(P) coolstore	VirtualService	Config: 🕑
inventory	DeploymentConfig	app inventory group com.redhat.cloudna	tive provider fabric8 version 1.0.0		2/4/2019, 4:52:18 PM	343548			
inventory-v2	DeploymentConfig	app inventory group com.redhat.cloudna	tive provider fabric8 version 2.0.0		2/4/2019, 4:52:18 PM	343079	coolstore	DestinationRule	Config: 🕢

Services view focus on Istio configuration Validation of Istio objects.



Tracing with Jaeger

Jaeger UI Lookup by Trace ID	Search	Compare	About Jaeger \sim
Find Traces Service (6) inventory Operation (0) all		100ms 9 50ms 05.16.40 am 06.10.00 am 06.03.20 am	° ∎ Tim#
Tags ⑦ http.status_code=200 error=true		100 Traces	Sort: Most Recent V
Lookback		Compare traces by selecting result items	
Lustriou		gateway: inventory:8080/* da8/41b	13.38ms
Min Duration e.g. 1.2s, 100ms, 500us		2 Spans gateway (1) inventory (1)	Today 11:31:09 am a few seconds ago
Max Duration		gateway: inventory:8080/* 3b6428d	11.16ms
e.g. 1.1s Limit Results		2 Spans gateway (1) inventory (1)	Today 11:31:09 am a few seconds ago

Jaeger UI integrated into Kiali for 360° observability view



Then?



DEMO TIME!

NO ANIMALS were harmed during the making of this demo NO CODE was changed either... almost :-)

Demo explained...

1. Coolstore works properly

- 2. Break Catalog (2 pods, 1 down)
- 3. Notice (Kiali, Grafana, Jaeger)
- 4. Find the root cause
- **5.** Fix \rightarrow retry policy (2 pods, 1 down, avoid the one down!)
- 2. New version Inventory
- 3. Distribute load
- 4. Break inventory (2 versions, 1 version down)
- 5. Notice (Kiali, Grafana, Jaeger)
- 6. Find ...
- **7.** Fix \rightarrow circuit breaker (avoid damaged version)



TAKEWAY

Distributed Services Platform



try it at http://learn.openshift.com



Get Started!

- <u>https://istio.io/docs/</u>
- <u>https://maistra.io/</u>
- <u>https://www.kiali.io/</u>
- <u>https://github.com/redhat-developer-demos/istio-tutorial</u>
- <u>https://github.com/rmarting/service-mesh-meetup</u> (This Demo)





OPENSHIFT Meetup

Questions?



Microservicios Meetup





OPENSHIFT Meetup

Thank you!



Microservicios Meetup

