Red Hat

Red Hat Dallas Emerging Tech Summit

December 5, 2019

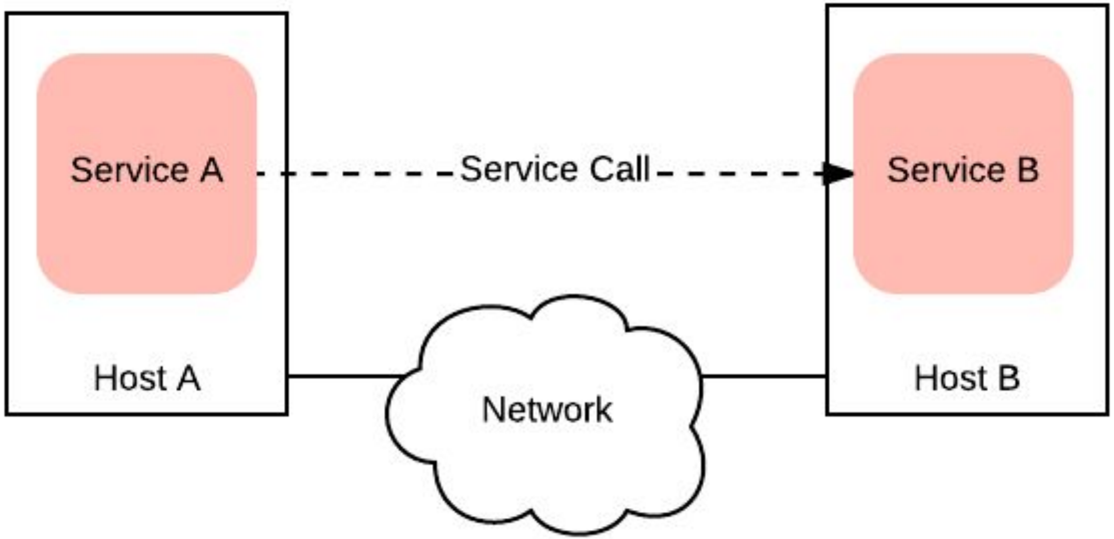
OpenShift Service Mesh

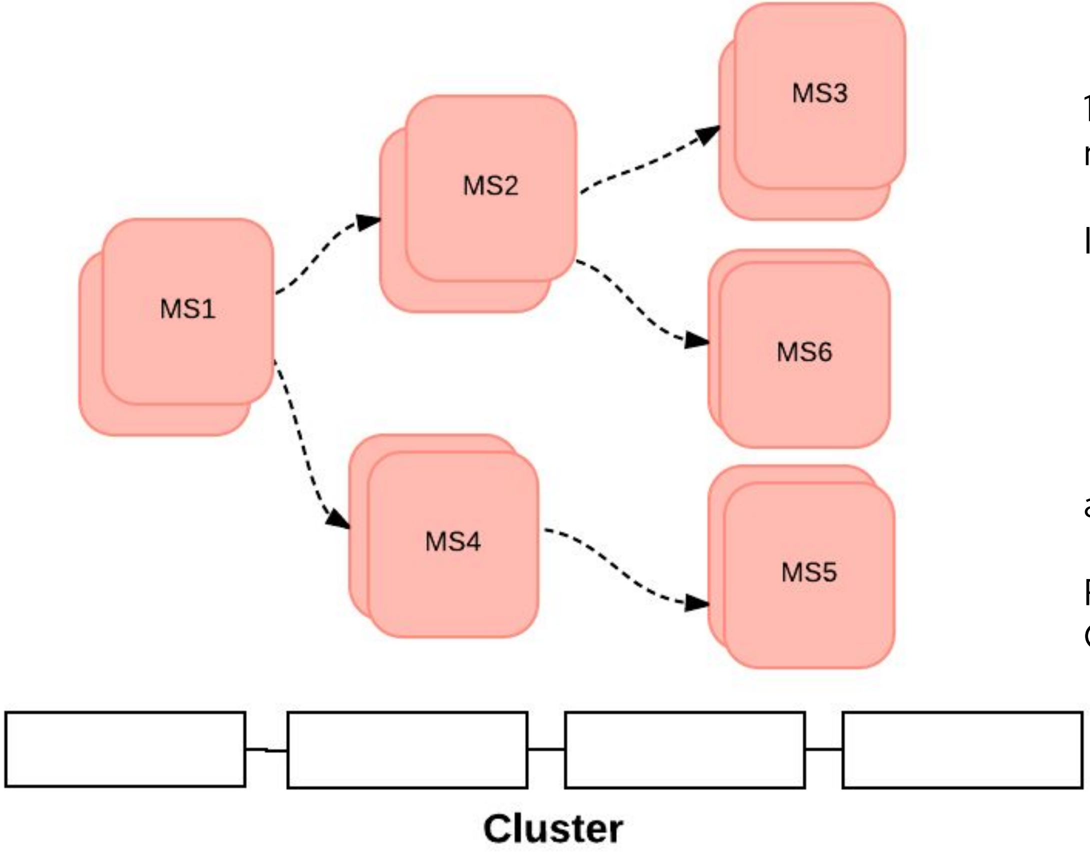
Veer Muchandi, Chief Architect - Container Solutions, NA Commercial

Introductions

Name: Veer Muchandi
Role/team: Field Chief Architect, Container Solutions, NACS
Where you're from: Lives in Alpharetta, GA; Travels all around spreading awesomeness of Containers!!
Helps enterprises with Container Strategy, DevOps, Microservices strategy. He conducts strategy sessions, workshops and education sessions and drives the change. Veer is a well known speaker and blogger. He loves to learn and teach. Veer also runs OpenShift Meetup groups in Atlanta and Jacksonville.

Calls with Distributed Computing





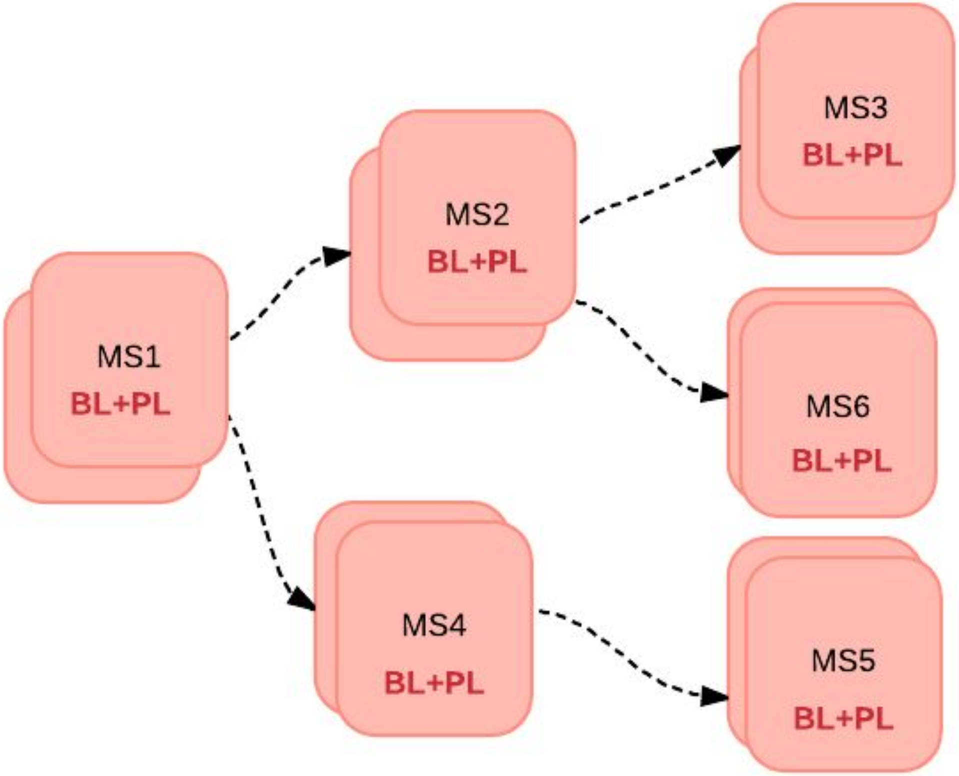
100s of Microservices running on cluster

Issues to address:

- Load balancing
- Network Faults
- Circuit Breakers
- Service Discovery
- QOS & SLA

and more..

Requires “Plumbing Code”



BL - Business Logic

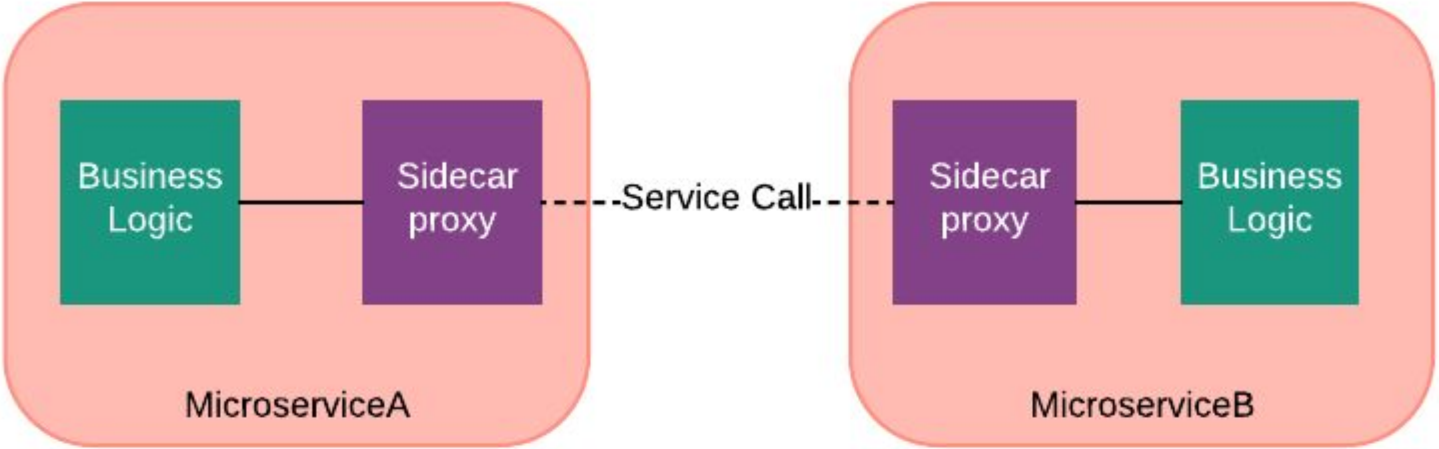
PL- Plumbing Logic to solve the infrastructure issues such as Service discovery, load balancing, fault-tolerance, rate limiting, QoS etc.

Examples: Frameworks such as NetflixOSS

Issues with libraries/frameworks embedded in code

Developers have to worry about the plumbing code

- Code intrusive
- Learning curve for such frameworks
- Not language-agnostic; hampers polyglot microservices
- Maintenance overhead
  - thousands of services using version of libraries
  - updates to infra libraries require integration, testing and re-deployment of all services

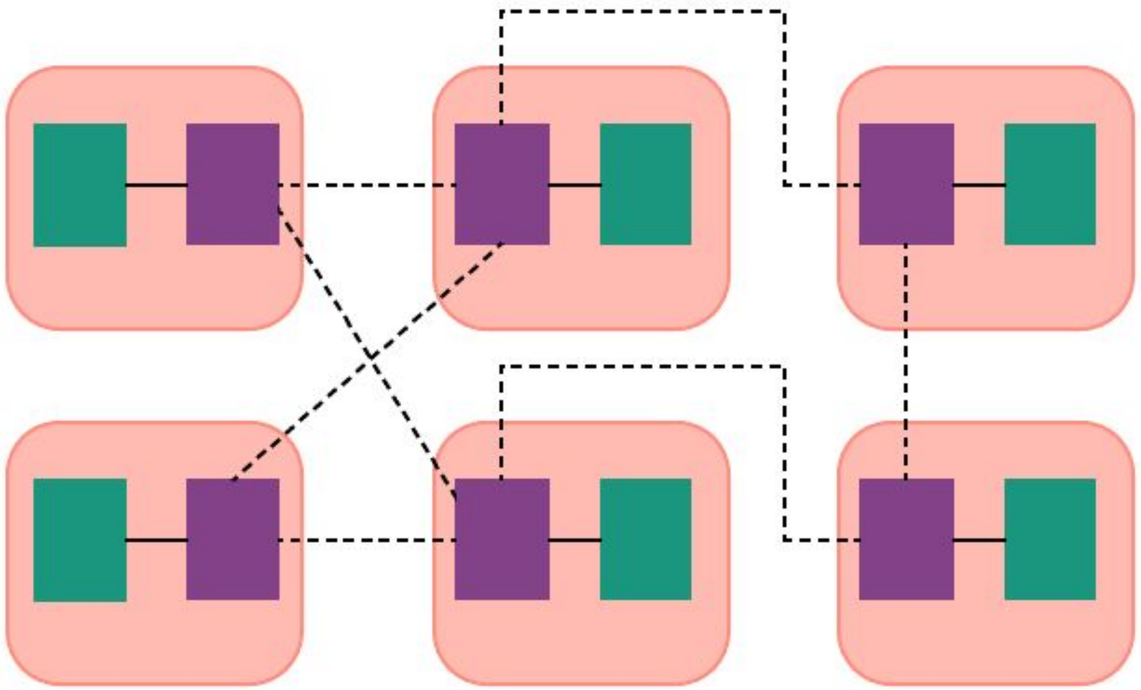


SideCar Proxy

- Intercepts all network communication between microservices
- Encapsulates Service Infrastructure code
- Application code (business logic) unaware of Sidecar proxy
- Examples – Linkerd, Envoy

Service Mesh

Network of Microservices



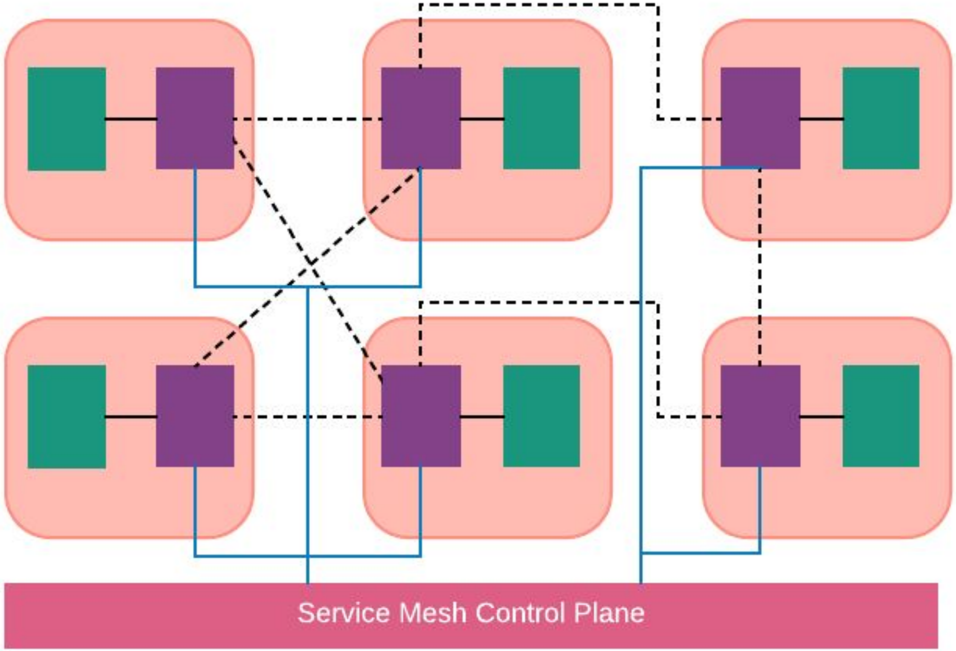
Service Mesh is a dedicated infrastructure layer to handle service-service communications

Typically implemented as an array of lightweight network proxies deployed alongside application code

Interconnected Proxies form a mesh network

Control Plane

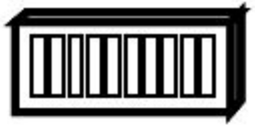
Service Mesh Implementation



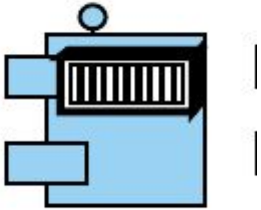
Implementation of mesh network involves a “Control Plane”

Proxies managed by centralized Control Plane

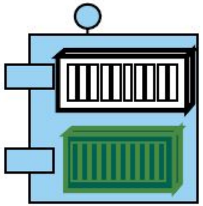
Kubernetes/OpenShift Pods



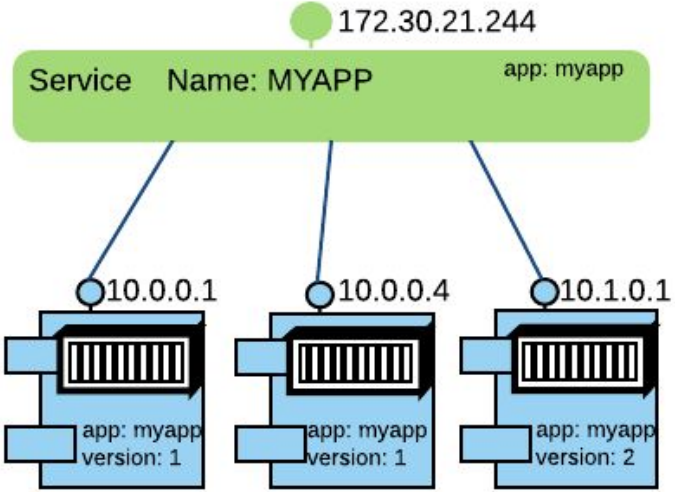
Container



Pod

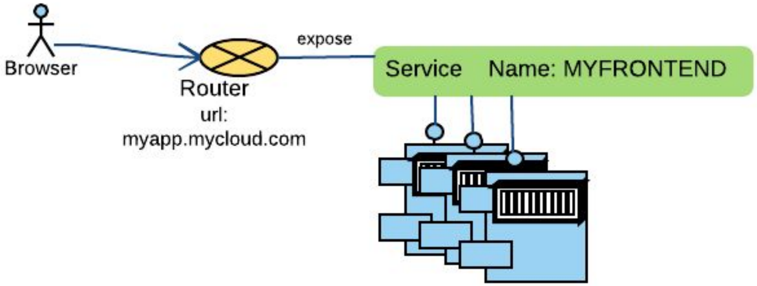


Pods support multiple containers. Sidecars are naturally available

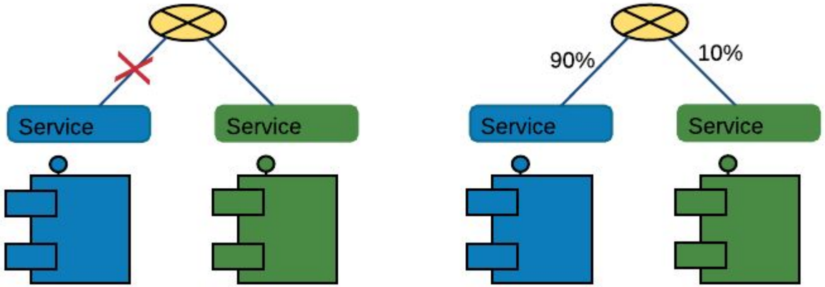


Pods scale and are typically front-ended by a Kubernetes Service. Service Discovery is built inside K8S using SkyDNS

Load balancing, Traffic Routing in OpenShift vs K8S



OpenShift provides routing as ingress into the cluster which does load balancing



Also implements patterns such as blue/green and A/B Testing via Router

Whereas, Kubernetes depends on an external load balancer
For Traffic Splitting, you can emulate it by changing the pod proportions behind services

While Kubernetes goes to an extent and OpenShift a little more in terms of the handling infrastructure needs such as service discovery, load balancing and some request routing, we need more OOTB features -

- Content based routing
- Canary, AB deployments
- Rate Limiting
- End-to-End access control
- Fault tolerance and fault injection
- Routing, Ingress and Egress rules
- Circuit Breakers
- Integration with tools for logging, monitoring, quotas, ACLs and more

Welcome to Istio

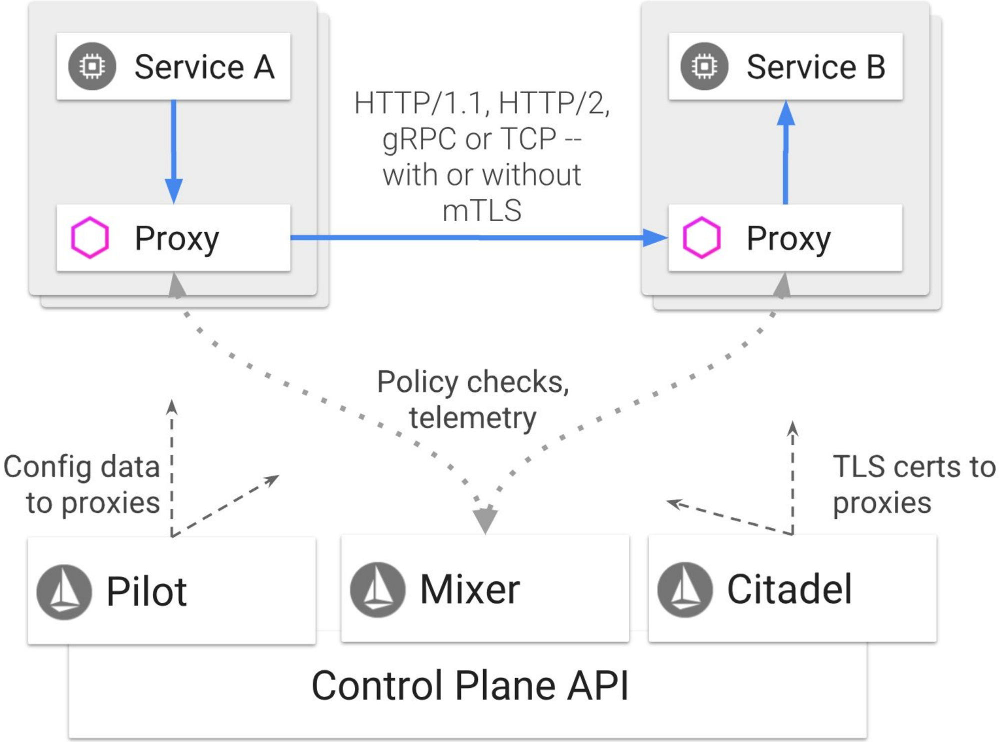
Istio implements Service Mesh and adds

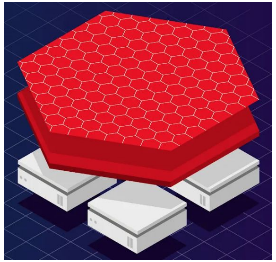
- **Traffic Management:** Load balancing, Failure Recovery, Circuit breakers etc
- **Observability:** Monitoring, Metrics, Tracing etc
- **Policy Enforcement:** Routing Rules, Ingress rules, Egress Rules, Canary, Blue/Green, A/B, Rate Limiting etc
- **Service Identity, Security:** verifiable identity to services, Access control, End-to-end authentication etc

Currently built for Kubernetes and other platforms such as Mesos coming up

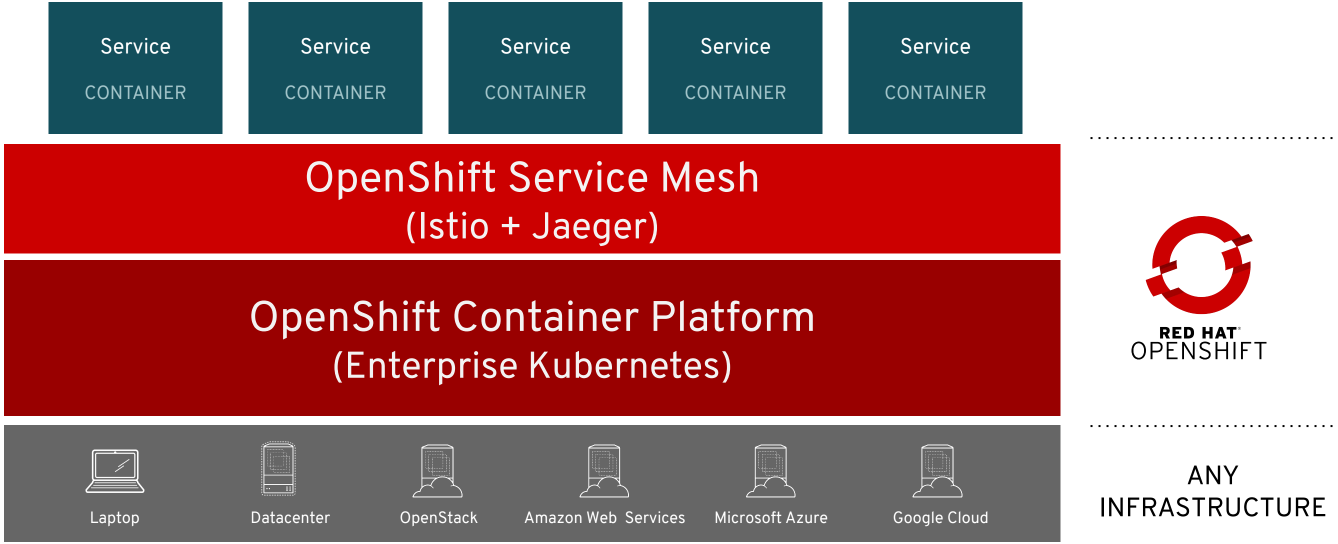
Integrates with existing solutions for ACL, Logging, Monitoring, Quotas, Auditing etc.

Istio Architecture





OpenShift Service Mesh



Service CONTAINER

Service CONTAINER

Service CONTAINER

Service CONTAINER

Service CONTAINER

OpenShift Service Mesh
(Istio + Jaeger)

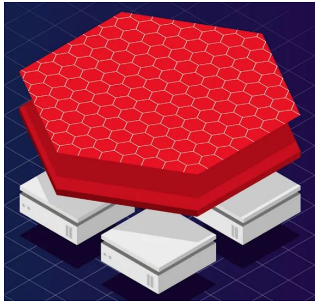
OpenShift Container Platform
(Enterprise Kubernetes)

Laptop | Datacenter | OpenStack | Amazon Web Services | Microsoft Azure | Google Cloud

RED HAT OPENSHIFT

ANY INFRASTRUCTURE

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

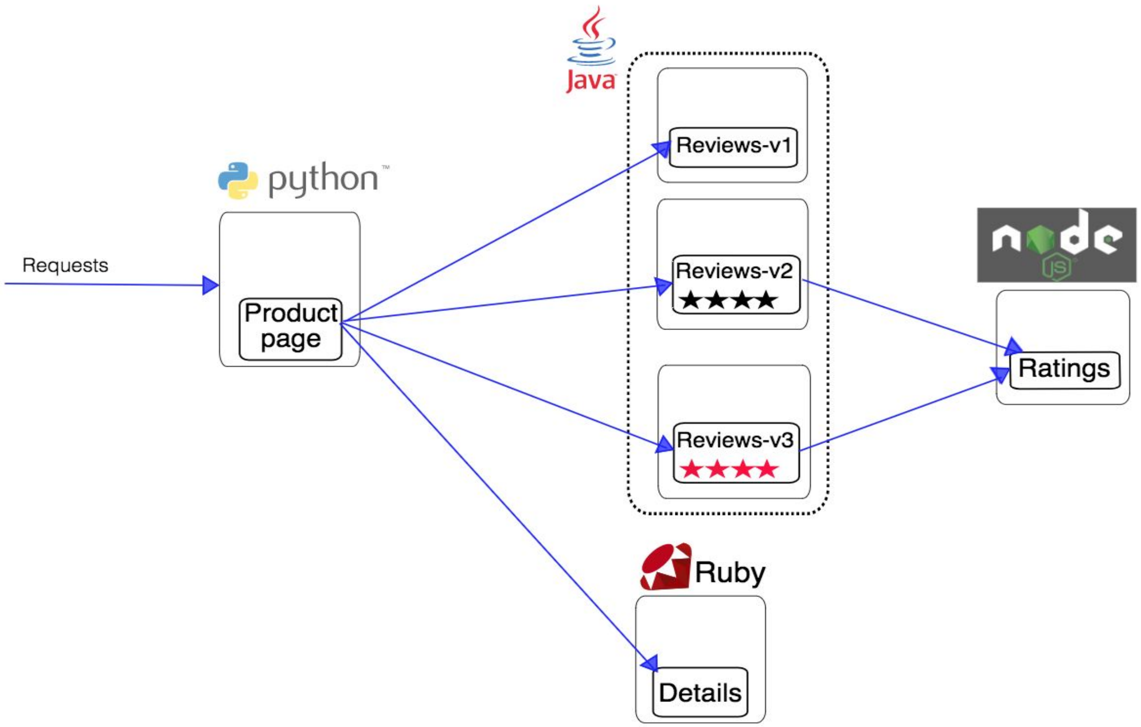


OpenShift Service Mesh

- Built in Tracing and Visualization with Kiali and Jaeger
- Operator based: Installed and managed by Service Mesh Operator
- Multi-tenant
  - Run separate service meshes for the set of applications that need to communicate
- Security of your applications the top priority
  - No need for elevated access privileges to service accounts in namespaces to inject sidecars
  - CNI plug-in replaces the init-container network configuration
- Network policies auto-configured to enable communications only with ServiceMesh control plane by default

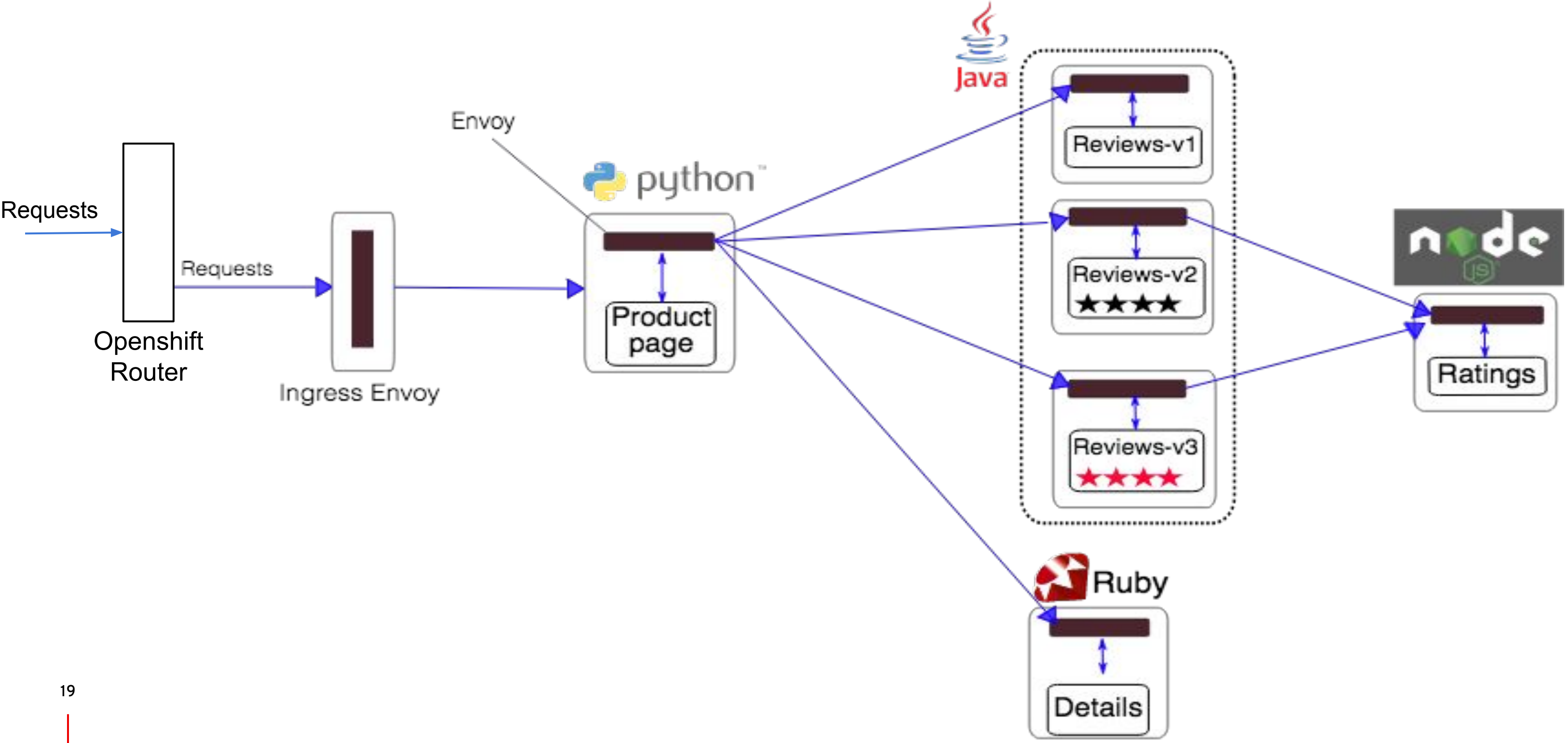
Demos

Understanding Sample BookInfo Application



BookInfo Application without Istio

BookInfo Control Flow



Red Hat

STAY ENGAGED

Developers. redhat.com

Your access point for no-cost developer tools and product subscriptions, how-tos, and demos

Red Hat User Groups

Meetups for networking and tech deep dives
www.meetup.com/Dallas-Red-Hat-Users-Group/

DevNation

Virtual and live events
Catch replays at
https://developers.redhat.com/devnation/

Next.redhat.com

Stay in touch with the Office of the CTO

Thank you

Red Hat is the world's leading provider of

enterprise open source software solutions.

Award-winning support, training, and consulting

services make

Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

twitter.com/RedHat