MANAGE MICROSERVICES AT THE APPLICATION LAYER WITH ISTIO

Discover, Authenticate, and Observe Service Interaction with a Service Mesh

In an era when every business is a software business, or needs to become one, using microservices is one of the keys to building a distributed application that can be scaled to meet changes in demand, modified to better engage customers, or extended to pursue new market opportunities.

A microservices architecture breaks up the functions of an application into a set of small, discrete, decentralized, goal-oriented processes, each of which can be independently developed, tested, deployed, replaced, and scaled. Each instance of a microservice typically resides its own container.

A service mesh manages the interactions of microservices at the application layer above virtual IP addresses and ports. A service mesh delivers service discovery, forwarding, monitoring, and service-to-service authentication.

Istio is an example of a service mesh. Istio intercepts network communications among the microservices that make up a containerized application deployed on Kubernetes to manage and help secure the microservices as they interact. Istio lets you oversee the interactions of microservices at a microscopic level.

Istio Architecture

The architectue of Istio divides the service mesh into a control plane and a data plane. The control plane manages proxies in the data plane, enforces policies, and collects data. In the data plane, the proxies are provided by Envoy, which mediates the traffic for the services in the service mesh. Envoy can dynamically discover services, handle TLS termination, check health, and provide metrics.

Istio deploys Envoy as a sidecar so you can add a service mesh to a Kubernetes deployment without rearchitecting your application or rewriting your code. A sidecar is a cloud-native architectural pattern that isolates a component or service of an application by placing it in a separate but co-located container for independence and flexibility. Istio can be deployed manually or by using Ansible or Helm.

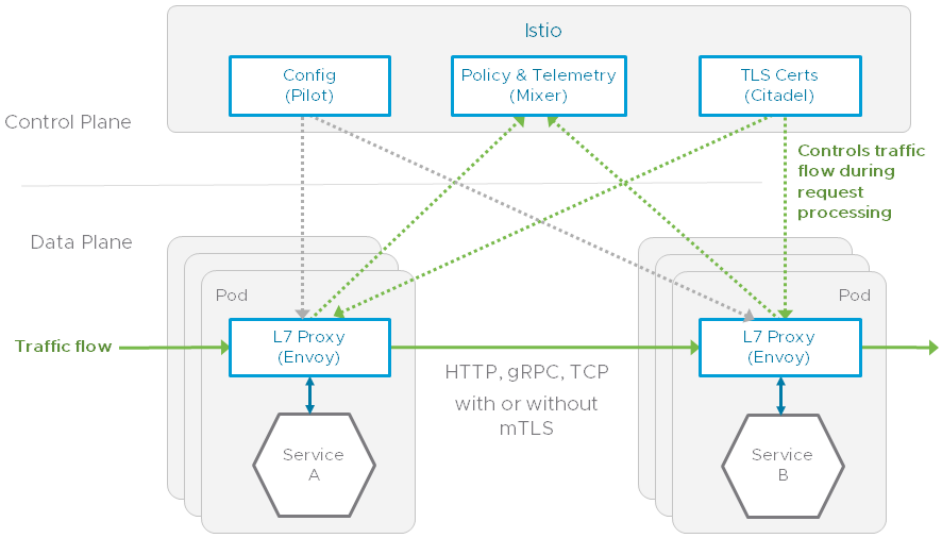

Figure 1: The architecture of Istio controls, secures, and observes the interaction of microservices at the application layer in a Kubernetes cluster.

MICROSERVICES ARCHITECTURE VS. MONOLITHIC ARCHITECTURE

A microservices architecture is the antithesis of a monolithic architecture. By building an app with small, modular components, developers gain flexibility and velocity: They can work with different languages and technologies, and they can improve productivity, especially in the face of rapidly changing requirements or evolving feedback.

But another difference between a monolithic architecture and a microservices architecture is that with microservices, you now must address such issues as security, rate limits, and monitoring for each microservice, rather than having to address the issues only once for a monolithic app.

vmware®

VMWARE PKS AT A GLANCE

VMware® PKS provides a highly available, production-grade Kubernetes platform equipped with advanced networking from VMware NSX® Data Center, a secure image registry, and life cycle management with BOSH. The solution radically simplifies the deployment and operation of Kubernetes clusters so you can run, orchestrate, secure, and maintain containers at scale on VMware vSphere®, GCP, or AWS.

VMWARE CLOUD PKS AT A GLANCE

VMware® Cloud PKS delivers Kubernetes as an easy-to-use cloud service that is managed by VMware so you can deploy, orchestrate, and scale containerized applications without the burden of implementing, operating, and maintaining Kubernetes. VMware Cloud PKS uses AWS or, soon, Azure as its underlying infrastructure. You can use Istio with VMware Cloud PKS to discover and monitor microservices.

LEARN MORE ABOUT VMWARE PKS

To learn about how VMware can help you deploy, manage, and secure cloud-native applications on VMware PKS, visit:

cloud.vmware.com

LEARN MORE ABOUT VMWARE CLOUD PKS

To learn how VMware can help you deploy, manage, and secure cloud-native applications on VMware Cloud PKS, visit:

cloud.vmware.com

Istio also includes several components that reside in the control plane:

- Mixer enforces access control, sets usage policies, and collects data from the proxies.
- Pilot helps manage and route traffic, including retries and timeouts.
- Citadel authenticates inter-service connections and controls access to services.

Application-Layer Requirements of Microservices

Microservices have requirements that go beyond networking and segmentation. To securely interact in the dynamic environment of containers and Kubernetes, microservices require the following:

- Service discovery
- Service forwarding
- Failure handling
- Visibility and monitoring with traces and metrics
- Rate limits
- Service-to-service authentication and authorization

Fulfilling these requirements can help answer low-level questions posed by application developers, questions such as the following that developers need to address as they build cloud-native applications:

- How is traffic entering the service mesh through ingress when mutual TLS authentication is enabled?
- How is end-to-end encryption enforced to bridge the gap between higher-level networking infrastructure and the service mesh?
- How can you shorten the mean time to repair (MTTR) when something goes wrong?

A service mesh such as Istio can help fulfill these application-oriented requirements. Istio works at the application layer to help developers regulate and observe how microservices interact at a microscopic level.

Using Istio with VMware PKS or VMware Cloud PKS

Istio can be used with VMware PKS as well as VMware Cloud PKS. When you deploy a containerized applications on a Kubernetes cluster, you can include Istio to help manage and secure the interaction of microservices. Deploying Istio helps manage microservices by adding the following functionality:

- Control of retries, failovers, and responses to faults.
- Policies for access control, quotas, and rate limits.
- Metrics collection for interactions, such as success and failure rates. The metrics can be displayed in the Istio dashboard by using Grafana.
- Authentication and authorization for service-to-service communication.

The result of using Istio with VMware PKS or VMware Cloud PKS is that you gain tighter control and observability over the interactions of services as you deploy, scale, and maintain an application built with a microservices architecture.

vmware®

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
Copyright © 2018 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents. VMware is a registered trademark or trademark of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: vmware-service-mesh-pks