SOLUTION BRIEF – **NOVEMBER 2019**

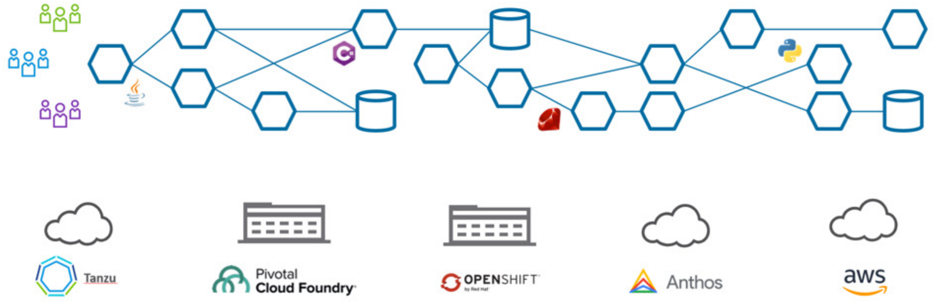
NSX Service Mesh Solution Brief

The Rise of Microservices Architectures Brings New Challenges

Application architectures are constantly changing. Over the past several years, the application space has evolved from monolithic to service-oriented architecture (SOA) to microservices. In a microservices architecture, latency is introduced between services in the service chain, which will affect the entire application and the user experience. Another challenge is troubleshooting and identifying the root cause of issues when they occur. This is usually done using tracing, which allows you to understand the communication and transaction flow of the applications. Troubleshooting is also hampered by the fact that applications built in microservices architectures are composed of many different services, often written in different coding languages. Some of the workarounds to address these problems -- like API gateways, language specific libraries, and northside proxies – do not fully address these challenges, but a service mesh can.

Introduction to NSX Service Mesh

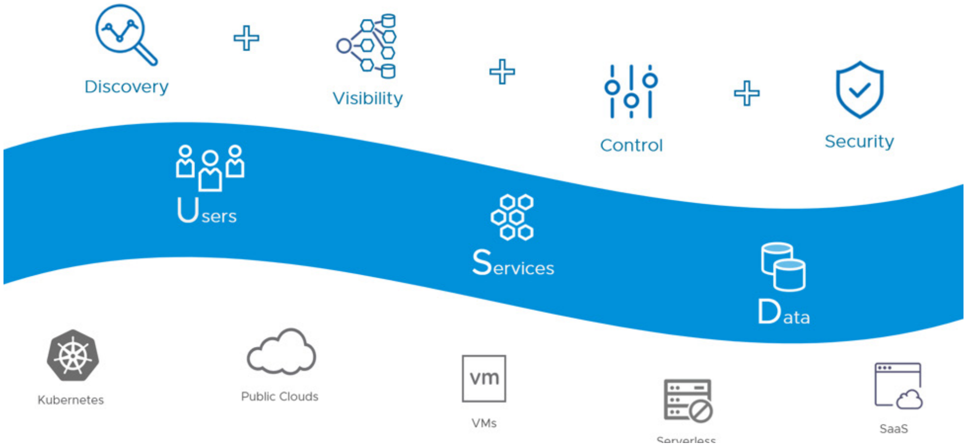
A service mesh addresses challenges associated with a microservices architecture. However, the service mesh itself introduces new challenges related to multiple Kubernetes cluster/multi-cloud operations and the limited scope of most service meshes today, which focus on services alone. NSX Service Mesh solves these challenges and more. By abstracting the service mesh from the physical boundaries of a single Kubernetes cluster and a single cloud, and by extending the scope from service-to-service communication to users-to-service-to data communication, NSX Service Mesh is able to control, secure, and operate applications, no matter where their components are deployed.



Widening the Scope of the Mesh

All service mesh implementations are focused on bringing visibility, traffic management, and security to service-to service communications at Layer 7. But application flows are not limited to interservice communications; users also access services and data (via those services). In a multi-cloud, multi-platform environment, where you may not have access to the underlying infrastructure, you need to move up the stack to manage communication and access between users, services, and data, by abstracting out the underlying physical infrastructure.

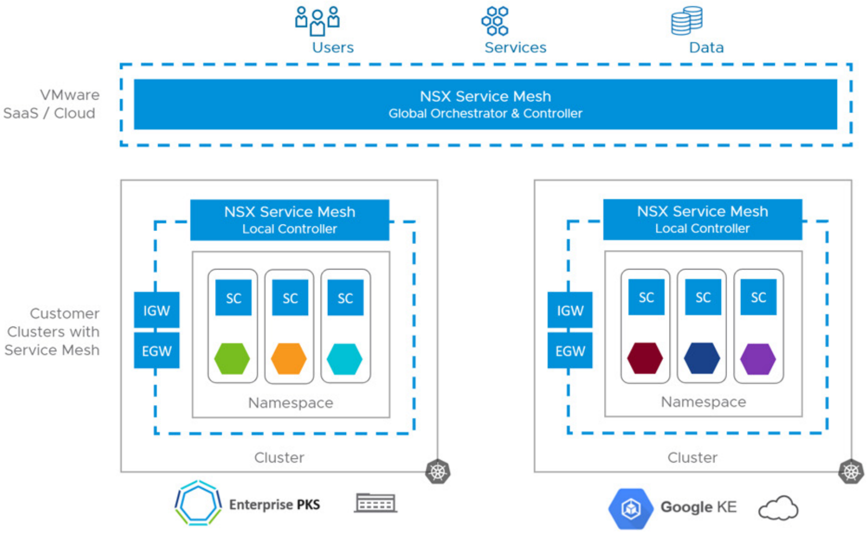
VMware is leading an effort in the open-source community to extend the visibility and control of communication from just service-to-service communications to include users, services, and data. This effort includes work on the data services proxy filter, which will allow the Envoy proxy to decode the data wire protocols and SQL queries. This allows NSX Service Mesh to have visibility into the entire transaction all the way to the datastore. With this visibility, access can be logged for auditing and security purposes. When this level of visibility is achieved, policies can be used to control access to the data.



NSX Service Mesh Architectural Overview

VMware NSX Service Mesh uses Istio as a data plane abstraction for Kubernetes workloads. When deploying Istio, it's typically tied to a single Kubernetes cluster. Istio users don't stretch it across more than one Kubernetes cluster, as most prefer each cluster to be able to operate independently from other Kubernetes clusters. For this reason, NSX Service Mesh acts as a control plane for many data plane Istio deployments managing the life cycle of Istio from onboarding to Day 2 and Day 3 operations. NSX Service Mesh only handles the life cycle of the service mesh (Istio, in this case); it does not handle the lifecycle of Kubernetes. When onboarding a new cluster on NSX Service Mesh, the service will perform the deployment of a curated version of Istio, which is signed and supported by VMware. This Istio deployment is the same as the upstream Istio in every way, but it also includes an agent that communicates with the NSX Service Mesh control plane.

Istio installation is not the most intuitive, but the onboarding process of NSX Service Mesh simplifies the process significantly. NSX Service Mesh acts as an abstraction layer on many data plane service meshes. The solution applies the same concepts of service mesh such as traffic control, security, and observability — not to a single Kubernetes cluster or cloud, but across Kubernetes clusters, clouds, and third-party service meshes. The architecture is constructed of a local Istio data plane with its own local control plane and a central control plane, which is the NSX Service Mesh service.



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com