WHITE PAPER

Zero Trust with cert-manager, Istio and Kubernetes

A 5-Step Practical Guide to Building a Production-Ready, Zero Trust Environment

Venafi

Written by cloud native experts, this white paper provides cloud native platform teams and enterprise security teams a five-step guide on deploying Istio service mesh based on industry best practices.

Executive Overview

This white paper addresses the primary cloud native factors organizations must consider when adopting Istio, including how to align with modern Zero Trust networking principles. In addition, it delves into Istio's many security features, particularly the ways in which mutual TLS (mTLS) can be leveraged within cloud native and legacy applications. Finally, it looks at integration points to sync Istio with cert-manager so that TLS certificates are automatically signed by trusted Certificate Authorities (CAs) that are then distributed to workloads in a service mesh.

Build	Deploy	Run
Container images	Kubernetes platform lifecycle	Container isolation
Private registry	Identity and access management	Network isolation
Build management	Platform data	Application access and data
CD/CI pipeline	Deployment policies	Observability

Introduction

Many enterprises see great value in using Istio for its range of advanced development, operations and security features, which solves many significant issues when running polyglot, distributed applications at scale. Service meshes are an abstraction layer on top of Kubernetes that consolidate crosscutting concerns when running production-grade distributed applications. Istio is one of the leading service mesh implementations, offering a feature-rich experience for securing workloads, managing network traffic and observing application behavior.

Zero Trust is a holistic approach to network security that posits that strict identity is required for each actor on the network, whether they are external or internal to the perimeter. Istio helps to manage workload identities by minting X.509 certificates that are used for mTLS, centralizing the process of certificate issuance and renewal.

Why Kubernetes?

Kubernetes is a cloud native container orchestrator for distributed systems that has become increasingly popular with organizations looking to optimize for scalability and application development. By using its principles of declarative resource descriptions, state reconciliation and using an extensible API, it has many benefits, especially in relation to environment interoperability and workload portability.

This provides good leverage for adopting microservices architectures, which, in itself, presents different challenges for security. For example, when running polyglot applications and frameworks, each may require different methods for implementing security, observability and traffic management.

This creates the need to focus on having consistent approaches to authorization and authentication, transport layer security as well as auditing environments, some of which may be multitenant. Additionally, consideration needs to be made for how to enforce governance and policy, as well as how to defend against threats such as man-in-the-middle attacks.

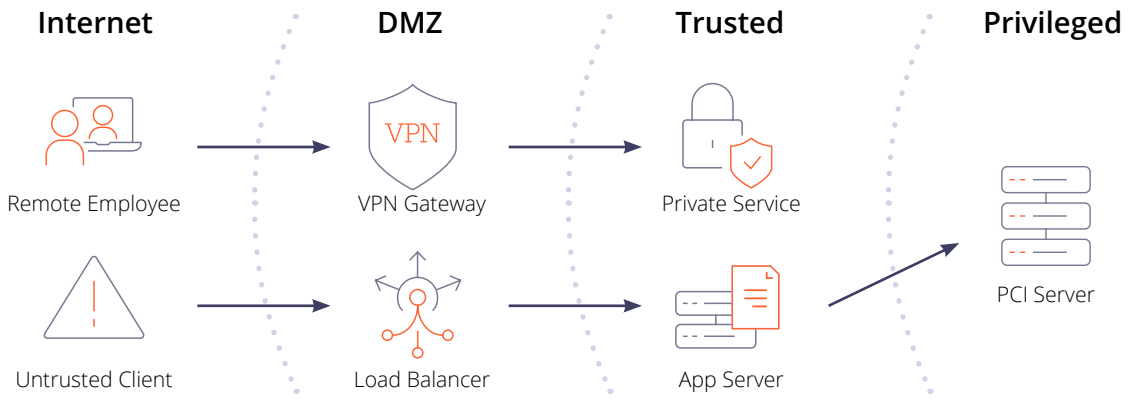
Build	Deploy	Run
Container images	Kubernetes platform lifecycle	Container isolation
Private registry	Identity and access management	Network isolation
Build management	Platform data	Application access and data
CD/CI pipeline	Deployment policies	Observability

The traditional network security model and its challenges

Traditional network security uses a perimeter model that focuses on a ring-fenced approach by breaking down different segments of the network into zones, usually contained by one or multiple firewalls. Zones are given different levels of trust that determines the different resources each is permitted to access.

This perimeter provides gates to resources deemed riskier, such as a public-facing web server, and places them in an exclusion zone like a DMZ, where traffic can be tightly controlled.

Because the threat landscape has evolved, the traditional approach to Zero Trust is no longer sufficient for modern cloud environments.



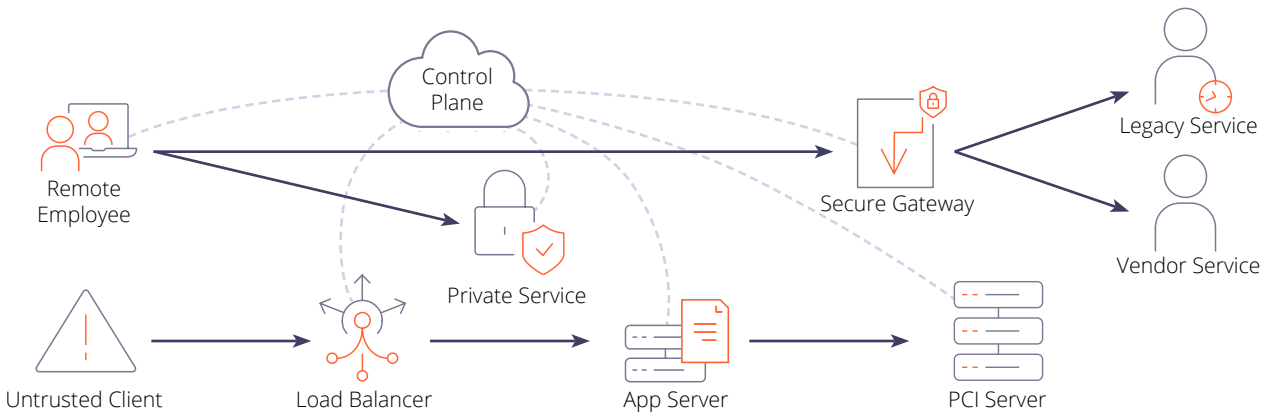
What is Zero Trust, and why does it matter?

In modern cloud infrastructure, a Zero Trust environment behaves as though external and internal threats exist on the network at all times. Network locality is no longer a basis for trust; therefore, all hosts must provide proper identities. Threats must be prevented from both active and passive attacks. For example, a passive attack would be the ability for malware to sniff traffic within a cluster for sensitive information. This requires that strong encryption is used, in addition to host identity.

Every device, user and network flow must be authenticated and authorized on the network using cryptographically verifiable identities, which are required to allow communication between actors on the network.

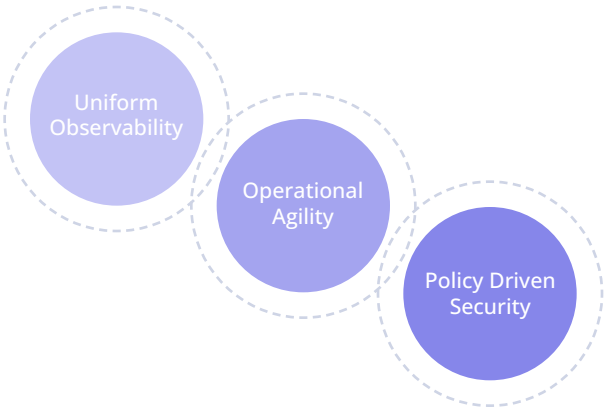
There are three main premises to a Zero Trust network: user and application authentication, device or machine authentication, and trust.

An increasingly common way to achieve Zero Trust is to use a layer of platform infrastructure that abstracts away the complexity and management overhead in order to transparently implement a Zero Trust network environment.



Istio service mesh

In a cloud native Kubernetes context, a service mesh is a dedicated infrastructure layer that consolidates crosscutting concerns by controlling service-to-service communication over the network. It provides a transparent and language-agnostic framework to easily and flexibly automate application workload functions and allow policy separation between application stacks and network stacks.

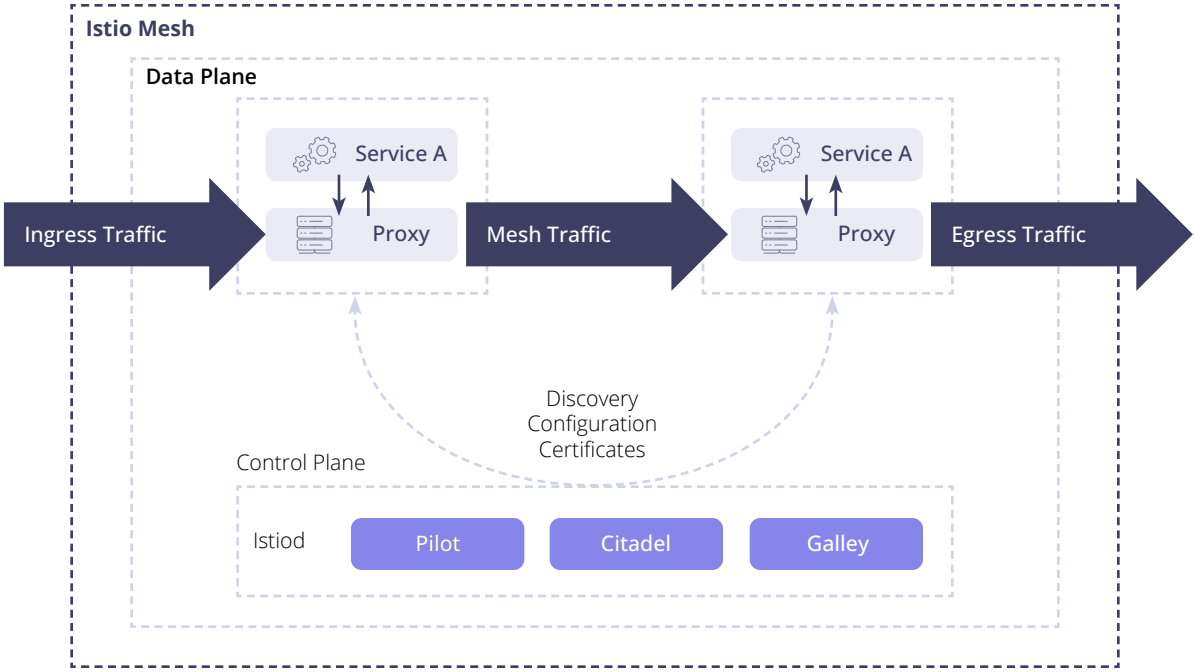


Istio is based around three pillars to provide uniform observability, operational agility and policy-driven security.

Istio architecture

At a high level, the Istio architecture comprises a control plane and a data plane. The control plane has components for configuration, such as traffic routing rules and network policies, that is distributed to proxies in the data plane. Additionally, service-to-service authentication and authorization is enabled by issuing TLS certificates to workloads in the mesh.

In the Istio data plane, high-performance and programmable container instances of the Envoy proxy are injected into Kubernetes Pods. This follows the sidecar pattern in which the proxy is colocated with each workload. All inbound and outbound network traffic is intercepted by these proxy containers before being routed to backend services, thus acting as a policy enforcement point. As the Envoy proxies operate at Layer 7, policies can be used to enforce traffic routing configurations, which can route traffic to specific back ends based on attributes such as HTTP headers, or enforce authorization policies.



Istio security principles

The Istio security posture is built around three principal factors.

The first is strong workload identities driven by powerful policy enforcement and transparent mTLS encryption, which is essential for authentication and authorization. These are abstracted away from application workloads, meaning there are no changes required to business logic or the underlying infrastructure.

Second, defense-in-depth allows for Istio policies to be layered with existing security systems that further strengthens the security of the mesh and communications outside of the network perimeter.

Third, the service mesh is able to implement and adhere to the three Zero Trust principles of user and application authentication, device or machine authentication, and trust. This is facilitated by having a root trust enabling transparent TLS, as well as secure identity management, certificate issuance and rotation.

mTLS for transparent service-to-service encryption

Mutual TLS (mTLS) is a mechanism for authentication between two parties using public key certificates. It enables bidirectional encryption of traffic and provides transparent service-to-service encryption.

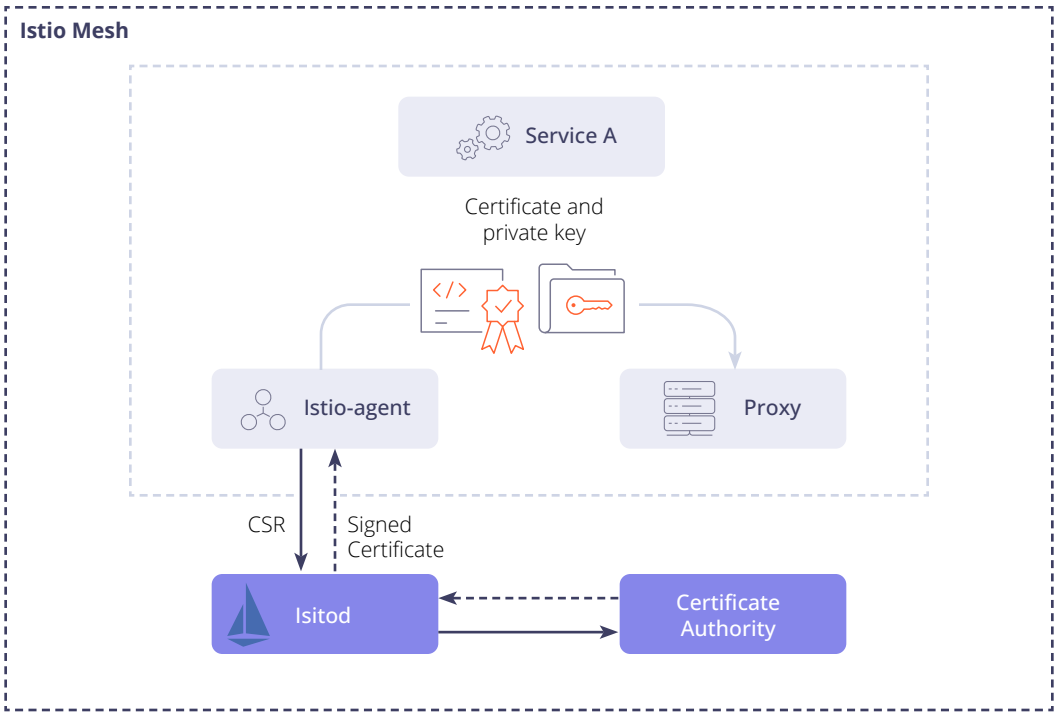
In Istio's case, identity management using mTLS is provided using the SPIFFE format as well as automating key and certificate generation, distribution and rotation. In contrast to end-user authentication where only the server certificate is used to establish authentication, in service-to-service authentication, the client and server must use the X.509 certificates in order to establish proof of both identities.

There are essentially two options when looking to adopt Istio as a service mesh that depend on the application requirements. The permissive mTLS option uses a combination of mTLS encrypted and unencrypted traffic between services which can be useful for supporting legacy services that are not yet migrated to Istio and enable access for legacy clients that reside outside of the mesh. By comparison, the strict mTLS option enforces traffic encryption between all services, as well as control plane components by default.

Certificate management

Certificate management is a key enabler of the service mesh model and is responsible for X.509 certificate issuance and lifecycle management. It facilitates identity in the mesh and is the critical framework for establishing trust. It provides the means for transparent adoption, as well as automation and rotation at scale. Within a single island mesh, this works well for service-to-service communication; however, this model does not work well for infrastructure that requires intermesh communication. To support service-to-service communication between meshes, the trust solution must support a different set of identities from another mesh environment. This requires the Istio CA to extend support for communication using intermediate certificates so clusters from each mesh have a root CA out of cluster that CAs from each mesh implicitly trusts. In reality for the enterprise, the model to support cross-cluster, multi-mesh trust should be bootstrapped using certificates from a preferred private issuer. This is a common requirement for modern enterprise environments.

A real-world challenge with certificate management and Istio is how to integrate with existing enterprise PKI solutions and ensure certificates in the control and data plane are rooted in the enterprise chain of trust.



cert-manager machine identity management

cert-manager is the industry leading open source machine identity management solution for full X.509 certificate (i.e., TLS) lifecycle management in Kubernetes.

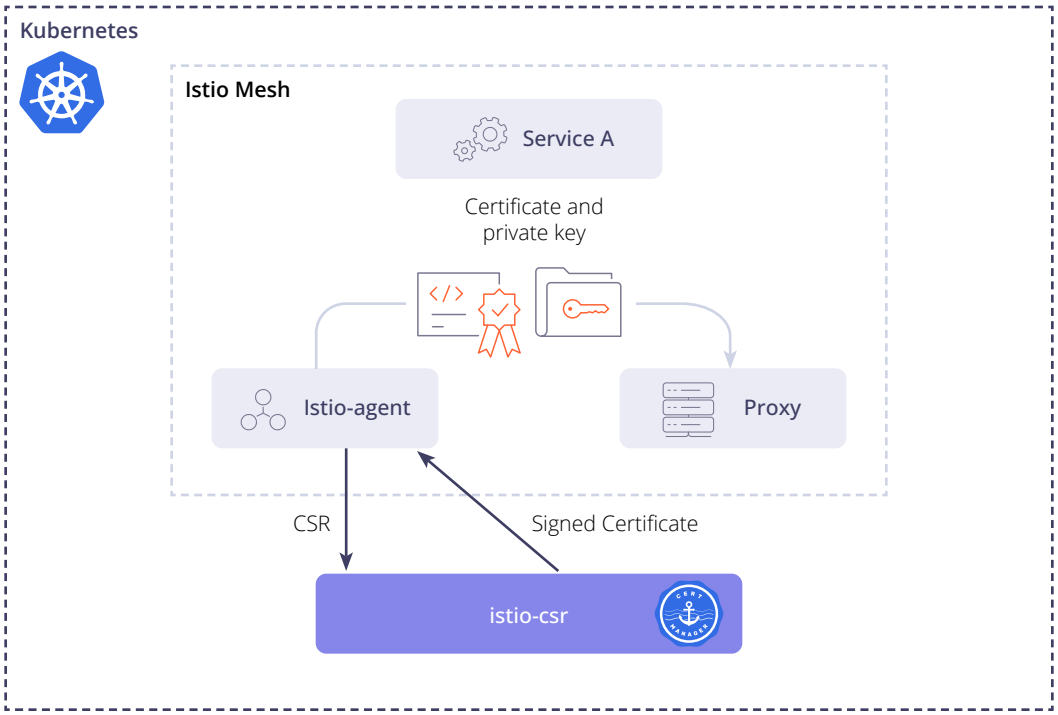
cert-manager was created and invented by the Jetstack team at Venafi. In 2020, it was accepted into the CNCF (Cloud Native Computing Foundation). In 2021, cert-manager was downloaded more than half a billion times from organizations around the world.

cert-manager is built with integrated support for a range of issuers, including ACME and Venafi, and enables a seamless integration to automate certificate issuance and renewal for workloads in clusters. It automates the full process for certificate issuance and certificate lifecycle management with built-in support for issuers such as ACME certificates from Let's Encrypt, as well as HashiCorp Vault and Venafi. With an extensible framework for creating new issuers, cert-manager also easily enables private issuers that allows enterprises to easily enable cert-manager to work with their own preferred PKI solutions.

In an Istio context, istiod has its own Registration Authority (RA), as well as a CA to manage certificate authentication and authorization. The Istio-agent, colocated with workloads, directs certificate signing requests at istiod. Once authenticated and authorized, a local CA signs the certificates.

To allow enterprise to deploy Istio with its own private PKI solution, cert-manager is used in place of the built-in istiod RA/CA and replaces them with a new service called istio-csr. This agent performs the RA function, complete with fine-grained policy controls, and integrates directly with cert-manager to work with whichever issuer is needed by the enterprise, for example, Venafi TPP or Vault.

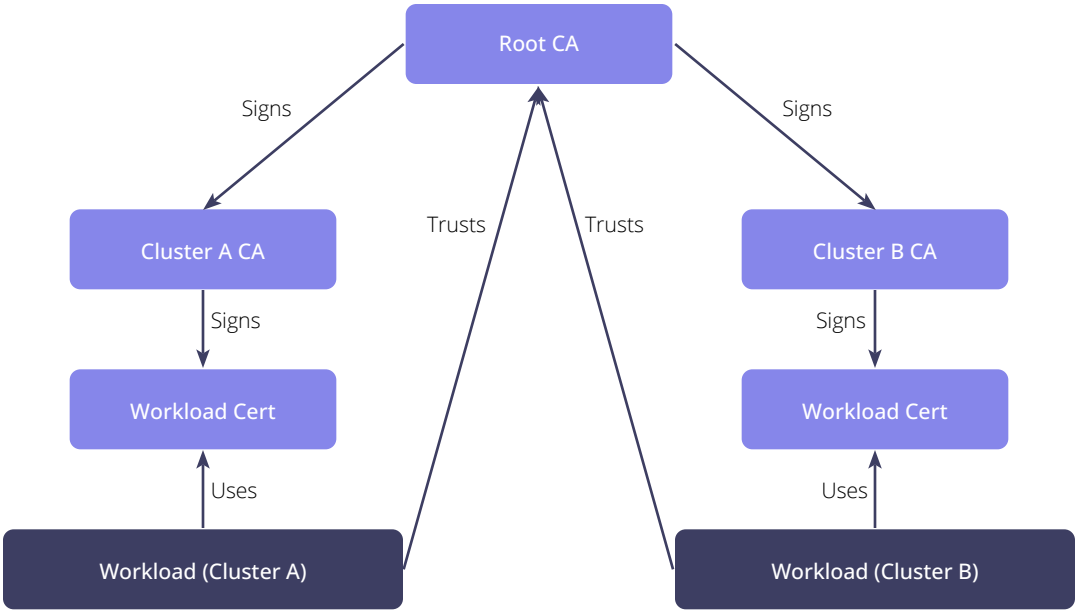
cert-manager's huge popularity is due to its demonstrable appeal as a consistent, reliable and agnostic solution for machine identity automation.



Service meshes in multi-Kubernetes clusters

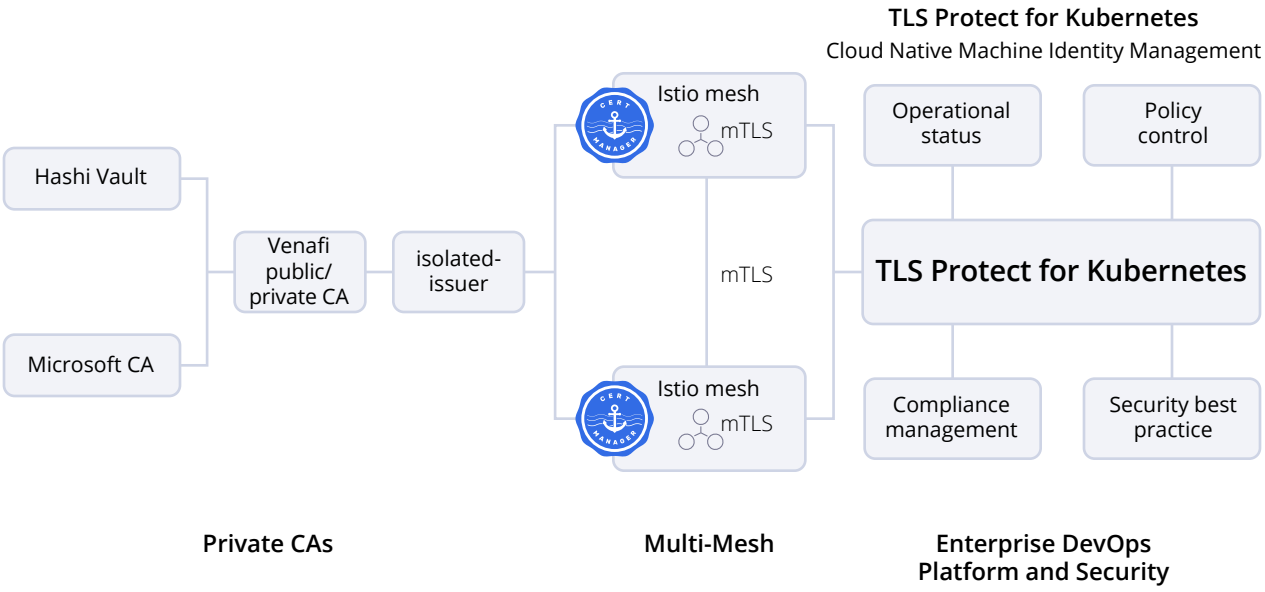
In a multi-primary control plane model, while each cluster has its own mesh, they share a root of trust. Thus, workloads from any mesh that share this root of trust can communicate with each other. Communication between clusters then goes through an ingress gateway configured to pass the TLS connection straight through to the workload's Istio sidecars.

To do this each cluster's Istio control plane must use a plug-in CA that shares a common root CA. Istio uses this CA to sign certificates for each workload in its mesh. All workloads are configured to trust the root certificate, so that they can trust workloads in other clusters signed by other cluster CAs. In the visual example above, each workload trusts the root CA and uses a workload certificate with access to the private key. If the workload in cluster A were to communicate with the workload in cluster B, it would present a certificate chain from the root CA, which the workload in cluster B would be able to validate and then trust, as it trusts the root CA.



TLS Protect for Kubernetes with Istio service mesh

When operating at scale, the capability to centralize operations and consolidate visibility becomes essential to cluster management, monitoring and policy compliance. TLS Protect for Kubernetes is an enterprise-grade machine identity management solution built for cert-manager. It provides a control plane for cross-cluster visibility and configuration controls, providing both platform and security teams with detailed views of the operational health and status of cert-manager and issuers, as well as the overall security posture. TLS Protect for Kubernetes provides extra visibility of each X.509 certificate in relation to its configuration and status and displays errors and warnings and recommended remediation. This helps to prevent future misuse of badly configured certificates and provides consistency at scale to manage increasing volumes and variations of certificate requests. This hardens the enterprise security posture by supporting the platform team's need to implement security best practices for certificate management and utilization.



To support the enterprise requirements for additional security benefits using Istio, TLS Protect for Kubernetes provides a unique feature to enhance the security of cert-manager: isolated-issuer. This component acts as an external cert-manager issuer, and it is designed to be deployed outside clusters in hardened environments. The component is responsible for signing machine identities in the mesh, while protecting private keys and CA credentials for third party platforms such as HashiCorp Vault or Venafi TPP. The isolated-issuer is typically bootstrapped with an intermediate CA certificate, issued from an enterprise CA, and this is used to sign certificates for workloads in the mesh, together with the istio-csr component. The private key material may be stored in memory or in Vault to enhance security.

The isolated-issuer and TLS Protect for Kubernetes work with Istio to provide trust for multicluster and multi-mesh architecture. Using an out-of-cluster isolated-issuer and CA means workloads can obtain identities for shared trust between clusters and meshes, while meeting requirements to use an enterprise-backed CA. It also enables a more seamless means to authenticate and authorize with existing on-premises, non-containerized workloads in other environments across the enterprise estate.

5 levels to service mesh adoption for Zero Trust

The following practical guide is a suggested set of progressive action points that can be used to easily build a high level of proficiency and learning using Istio and TLS Protect for Kubernetes to build a cloud native Zero Trust environment.

	Zero Trust Adoption Level	Action Plan	Objective
LEVEL 1	Automate certificate management and lifecycle to secure ingress gateways	Deploy cert-manager to manage and automate WebPKI using public CAs, such as Let's Encrypt, and use TLS Protect for Kubernetes for ingress and certificate visibility.	Establish machine identity automation for WebPKI and secure all web-facing workloads.
LEVEL 2	Automate secure intracluster pod-to-pod communication	Use cert-manager with the CSI driver to automate certificates for each pod in a cluster and use this in code to implement mTLS between workloads.	Implement the Zero Trust security principle of authentication for workloads running in pods in a Kubernetes cluster using certificates issued from a private PKI provider.
LEVEL 3	Adoption of Istio service mesh for seamless interservice mTLS	Implement Istio with cert-manager and the istio-csr agent to automate certificate generation and renewal for Istio sidecar proxies, enforcing mTLS between pods residing in Istio.	Abstract and automate the orchestration of service identity for encrypted traffic in transit and mutual authentication within Kubernetes environments.
LEVEL 4	Use identity- and policy-based authorization controls for edge and inter-service traffic flows	Use Istio Authorization features for dynamic workload-to-workload and enduser-to-workload access control enforcement.	Implement a key tenet of Zero Trust, authorization that builds on the service identity and authentication capabilities of Istio for fine-grained, intramesh access control.
LEVEL 5	Multicluster/multi-mesh integrated with an enterprise CA, such as HashiCorp Vault or Venafi, with TLS Protect for Kubernetes	Use the TLS Protect for Kubernetes isolated issuer for cert-manager to sign certificates outside of clusters, ensuring protection of sensitive private CA key material, as well as automation of intermediate CAs rooted in the enterprise chain of trust.	Fully automated machine identities, mutual authentication and authorization controls, for workloads across multiple meshes, integrated with enterprise PKI for end-to-end cloud native Zero Trust.

For more information on how Venafi can help you with Kubernetes and Istio implementations please visit **venafi.com**.

Trusted by

5 OF THE 5 Top U.S. Health Insurers
5 OF THE 5 Top U.S. Airlines
3 OF THE 5 Top U.S. Retailers
3 OF THE 5 Top Accounting/Consulting Firms
4 OF THE 5 Top Payment Card Issuers
4 OF THE 5 Top U.S. Banks
4 OF THE 5 Top U.K. Banks
4 OF THE 5 Top S. African Banks
4 OF THE 5 Top AU Banks

Venafi is the cybersecurity market leader in identity management for machines. From the ground to the cloud, Venafi solutions automate the lifecycle of identities for all types of machines—from physical devices to software applications, APIs and containers. With more than 30 patents, Venafi delivers innovative solutions for the most demanding, security-conscious organizations in the world. **To learn more, visit venafi.com**.