TIGERA

White Paper

Using Calico to create a Kubernetes cluster mesh for multi-cluster environments

Security, observability, and networking for microservices spanning multiple clusters across hybrid and multi-cloud environments

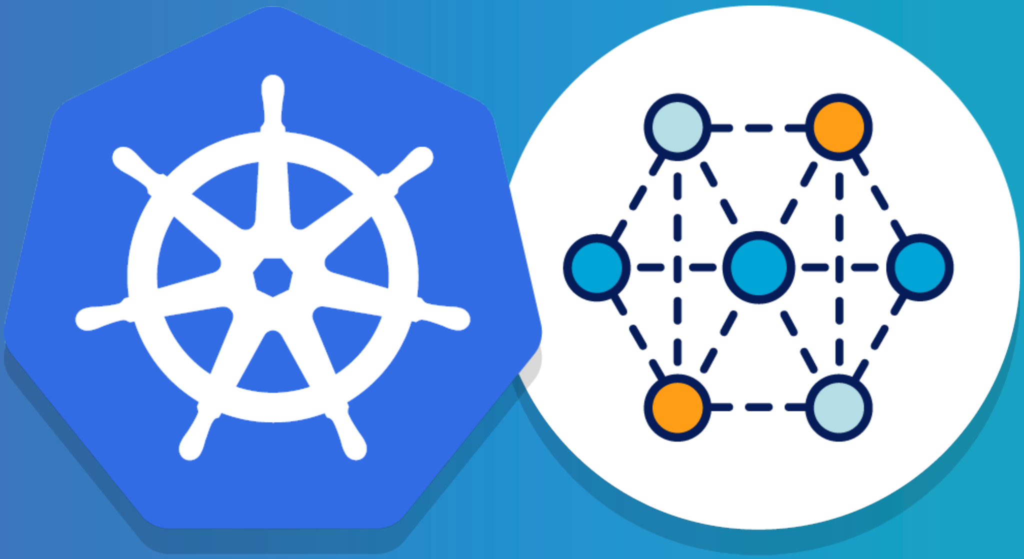


Table of Contents

Background 3

Why do you need microservices spanning multiple clusters? 4

Security, observability, and networking requirements for microservices running in multi-cluster environments 5

How Calico enables a Kubernetes cluster mesh for security, observability, and networking in multi-cluster environments 6

How companies are using Calico for multi-cluster Kubernetes environments 10

Conclusion 11

Additional resources 12

Getting started 12

Background

Most organizations start and often stay with a single Kubernetes cluster. In Kubernetes literature, the cluster boundary is how far the Kubernetes control plane reaches. However, the usage of Kubernetes, especially at leading-edge organizations operating at scale, has crossed the single-cluster threshold. Organizations are building and deploying services across multiple clusters for high availability, disaster recovery, application isolation, compliance, latency concerns, staged migration, and multi-tenancy reasons.

Regardless of the reasons to deploy multiple clusters, platform and application teams must address networking, security, and observability issues related to microservices deployed across multi-clusters, sometimes spanning hybrid and multi-cloud environments.

Calico is the most widely adopted container networking and security solution powering more than 100M containers across 2M+ nodes in 166 countries. Calico is supported across all major cloud providers and Kubernetes distributions, and is used by leading companies to build Kubernetes cluster meshes for running multi-cluster Kubernetes environments.

After explaining the need for microservices spanning multiple clusters, this paper details the networking, security, and observability challenges this creates, and how you can address these challenges by using Calico to create a Kubernetes cluster mesh.

Why do you need microservices spanning multiple clusters?

There are several use cases for the use of microservices across multiple clusters. Some of these use cases are listed below.

- **Centralized services** – Cloud-native architecture relies on the use of interdependent and often centralized microservices that are built, deployed, and maintained by completely different teams and over different periods of time. Let's look at an example of this. Hipster Shop (included with Calico Cloud) offers an inventory service to show if an item is in stock. To offer this service, the inventory service needs to communicate with the inventory database service, which might be developed by another team and deployed in a new cluster. Microservices-based applications are often broken down into purpose-built units that serve a specific business function. These centralized services may be developed and deployed independently in their clusters.

- **High availability and disaster recovery** – For certain services, redundancy of workloads is essential to ensure high availability and disaster recovery. Examples of such services are database-as-a-service for popular databases such as MongoDB, Postgres, Kafka, Elastic, RabbitMQ, and others.

- **Geographic proximity and compliance requirements** – For certain services, the workloads need to be deployed in the geographic proximity of the service user for performance and latency reasons. Also, certain countries require that data is stored within the country's sovereign borders.

- **Isolation** – Multiple applications deployed inside a single cluster introduce the risk of threats moving laterally from one application to the other. To curb the blast radius of any one application or infrastructure problem within a cluster, an organization may choose a multi-cluster architecture instead of deploying multiple applications within the same cluster and risking cross-infection.

- **Scalability** – Certain businesses, especially those that depend on retail or business functions like AI and ML, have architectures with bursty characteristics that require provisioning of additional capacity through additional clusters in a new hybrid or cloud environment.

- **Phased upgrade and migration (blue/green deployments)** – With the emergence of feature-flagging and continuous deployment of new capabilities, organizations often adopt a phased approach to deploy these capabilities. They start with one or two clusters to limit the impact of unforeseen changes before deploying these changes across all clusters.

Security, observability, and networking requirements for microservices running in multi-cluster environments

Since Kubernetes does not natively support inter-cluster communication, the first and foremost requirement is enabling communication between Kubernetes clusters. Once communication is established, additional security and observability requirements need to be addressed. Some common requirements for enabling security, observability, and networking across multiple clusters are listed below.

- **Inter-cluster communication** – Communication across pods located in different clusters is essential for all use cases for microservices spanning multiple clusters.
- **Service discovery** – To enable the consumption of services deployed in different clusters, each cluster must be aware of the workloads deployed in the other clusters.
- **Load balancing** – If a multi-cluster architecture is used to deploy highly scalable and available services, it should be possible to load balance the traffic across multiple clusters.
- **Network and security policy** – Network and security policies that protect access to and from workloads, as well as help achieve microsegmentation, should extend to workloads deployed across multiple clusters.
- **Compliance** – Compliance is another challenge when working with a multi-cluster environment. For example, GDPR may limit communication between clusters based on the geographic location of the clusters. In that case, the security policies governing compliance in a multi-cluster environment must ensure the right level of isolation.
- **Encryption** – Encryption of communication within and across clusters is often required as part of internal security controls as well as some compliance regulations.
- **Observability** – Observability is challenging in multi-cluster architectures where services are deployed across multiple clusters in hybrid and multi-cloud environments.

How Calico enables a Kubernetes cluster mesh for security, observability, and networking in multi-cluster environments

As enterprises build out large, multi-cluster Kubernetes environments, they encounter an entirely new set of security challenges, requiring solutions that operate at scale and can be deployed both on-premises and across multiple clouds.

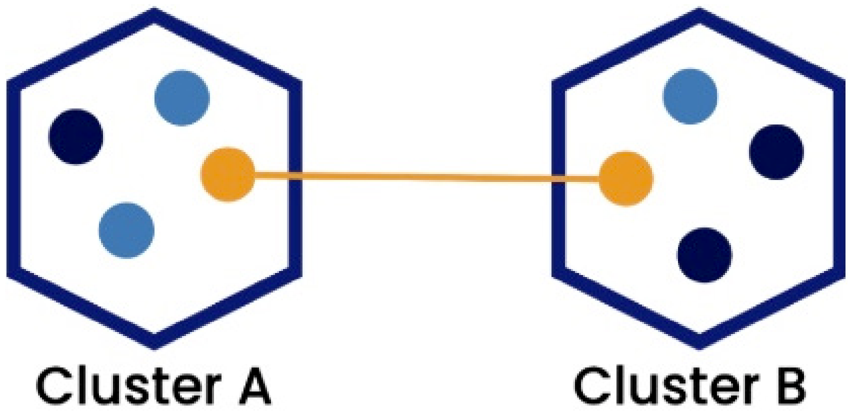
Calico provides a centralized management plane and single point of control for security, observability, and networking for a multi-cluster Kubernetes deployment. Let's take a closer look at Calico's capabilities in this regard.

Centralized management plane for multiple clusters

Managing a standalone cluster with multiple nodes is not onerous; however, when you add multiple clusters, the complexity compounds. Calico provides a multi-cluster management plane to enable security and observability across multiple clusters in hybrid and multi-cloud environments and user access control using RBAC. This architecture also supports the federation of network policy resources across clusters and lays the foundation for a single pane of glass.

Federation

With Calico, you can create policies in one cluster that reference pods in another cluster using federated identity. Federated services provide service discovery of remote pods in another cluster. With these two features, you can define fine-grained security controls between multiple clusters.



Service discovery with federated services

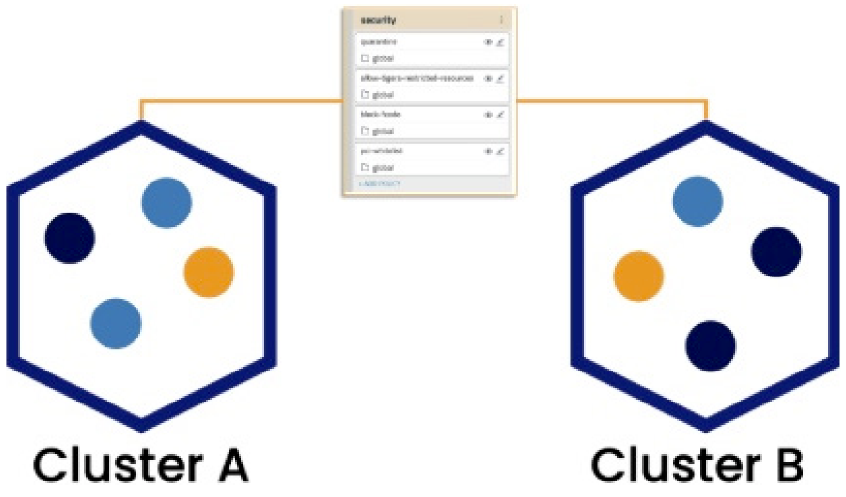
Calico supports the creation of a federated service that consolidates services from different endpoints. A federated service (also called a backing service) is a set of services with consolidated endpoints. Calico discovers services across all clusters (both local and remote clusters) and creates a federated service on the local cluster that encompasses all individual services.

Federated services are managed by the Tigera Federated Service Controller, which monitors and maintains endpoints for each locally-federated service. The controller does not change the configuration on remote clusters.

A federated service looks similar to a regular Kubernetes service, but instead of using a pod selector, it uses an annotation. Local cluster workloads can use Kubernetes service discovery to discover federated services with local and remote cluster workloads as backing endpoints.

Federated tiers and policies

Using federated tiers and federated policies, you can define security policies that apply across all clusters, or to a specific group of clusters. If you plan to deploy multiple clusters, federated tiers, and policies, you can extend your security controls to each existing and new cluster. This reduces duplication of policies (and maintenance of identical policies per cluster) to simplify the creation and maintenance of your security controls.



Unified security policy controls for application and network layers that work for local and federated microservices

Calico applies policies on federated services in the same way it applies policies on local services within a cluster, thus creating a seamless set of policies that authorize access to both local and remote microservices. Policy definitions within a cluster can reference labels on workload and host endpoints from remote clusters. Calico policies can be applied to both local and remote services. Calico's policy recommender will factor communications with remote cluster workloads and can generate security policy rules to permit inter-cluster communications.

Over the years, Calico has become well known for its capabilities around implementing controls across the full stack, from the network layer up through the application layer, providing the capability to combine application-layer controls with controls at the network or transport layer. You can implement policies you've defined by assigning them to tiers that enforce an order of precedence, and each of those tiers can be tied to role-based access control (RBAC).

For example, if you want to have a security team manage certain tiers that might be geared toward PCI compliance, or monitor egress traffic and compare it against known threat feeds, you can easily do so. What's more, you can put those policies in place without interfering with what you might want to enable your application and development teams to do for east-west traffic and service-to-service communication, which service mesh typically handles.

Calico also offers some powerful capabilities related to egress access controls. The capabilities make it easy to integrate with firewalls or other kinds of controls, such as DNS rules, where you might want to understand the origin of egress traffic and implement certain controls around that. Suppose you're working with a SIEM or other log management system or monitoring tool. In that case, it's really helpful to be able to identify the origin of egress traffic to the point where you have visibility into the specific application or namespace from which egress traffic seen outside the cluster came.

Unified security policy controls across bare metal, hosts, VMs and containers

In the complex world of digital transformation, application modernization often happens in phases over a long period of time. During this time, it is common to have legacy monolithic services on VMs and hosts, run along more modern microservices in containers. Calico provides a unified policy framework that works across bare metal, hosts, VMs, and containers enabling the co-existence of legacy and modern environments.

Policy replication

For organizations deploying data as a service in a multi-cluster environment for technologies such as MongoDB, Redis, Elastic, RabbitMQ, Kafka, and others, Calico provides an easy way to export the policies of a cluster and apply them to another identical cluster. This helps enforce consistent policies across all clusters and prevent drift.

In addition, Calico provides a single, centralized management plane to manage network security across all Kubernetes clusters, with capabilities such as:

- Central login for everything
- Centralized point of control
- Centralized log management
- Centralized storage management
- Report compliance of those controls
- Troubleshooting tools; flow visualizer
- IDS applied across all clusters under management; identifies potentially malicious activities
- Alert on non-compliance and indicators of compromise
- Alerts sent to SIEM (Splunk, Sumo Logic, and others)

Encryption

Calico offers encryption for data in transit that leverages the latest in crypto technology, using open-source WireGuard. As a result, Calico's encryption is highly performant and leverages an architecture that still allows visibility into workload communication.

One of the areas that introduces considerable operational complexity is the certificate management associated with mTLS that is used in most service meshes. WireGuard provides a highly performant alternative that requires zero configuration—even aspects such as key rotation are built into the protocol.

Observability and troubleshooting

Calico offers visibility into service-to-service communication in a way that is resource efficient and cost effective. It provides Kubernetes-native visualizations of all the data it collects, in the form of a Dynamic Service and Threat Graph. The graph allows the user to visualize communication flows across services and team spaces, to facilitate troubleshooting. This is beneficial to platform operators, service owners, and development teams.

With Calico, Envoy is integrated into the data plane to provide operational simplicity. No matter which data plane you're using (standard Linux iptables, Windows, or eBPF), Calico provides observability, traffic flow management, and control by deploying a single instance of Envoy as a daemon set on each node of your cluster. Instead of having an Envoy container in every pod as a sidecar, there is one Envoy container per node. This provides performance advantages, as it's more resource-efficient and cost-effective.

For federated services, the Dynamic Service and Threat Graph can visualize microservices communications across clusters. This visibility can easily be extended to workloads outside the cluster in other remote clusters as well.

In addition to the Dynamic Service and Threat Graph, the flow logs generated by Calico are enriched with contextual information on both local and remote cluster workloads for inter-cluster communication.

How companies are using Calico for multi-cluster Kubernetes environments

Companies that are some of the world's largest users of Kubernetes use Calico to create a Kubernetes cluster mesh to enable the use cases described below.

High availability for data/AI/ML as a service deployed across multiple clusters

Kubernetes has enabled the deployment of data and AI/ML services based on technologies such as MongoDB, Cassandra, Neo4j, Kafka, Elastic, RabbitMQ, Spark, Presto, and others. Given the large-scale use of these services and often the bursty nature of the usage, a multi-cluster architecture makes the most sense for high availability and scalability.

Calico is the perfect solution for such services. Calico supports the creation of a federated service that consolidates services from different endpoints, uses native Kubernetes capabilities to load balance traffic across these services for high-availability and scalability, enforces a consistent set of policies, and provides granular observability for services running across clusters.

Compliance

Some compliance regulations like GDPR require that access to services is limited to specific geographic regions. For a multi-cluster microservices deployment, this means that communication between certain clusters has to be limited. For example if a MongoDB service has been deployed in clusters in EMEA and North America, GDPR rules might require that EMEA clusters are prevented from communicating with North America clusters.

Calico enables the creation of a single federated service and the enforcement of policies that take into consideration the geographic location of the cluster to ensure that traffic between different geographic clusters is disallowed.

Phased upgrade and migration (rolling, blue/green, canary deployments)

Modern applications are often distributed and cloud based. They can scale elastically to meet demand and are more resilient to failure thanks to highly available architectures. These applications almost always have frequent deployments. To mitigate risks, organizations use a variety of deployment models, including rolling, blue/green, and canary models.

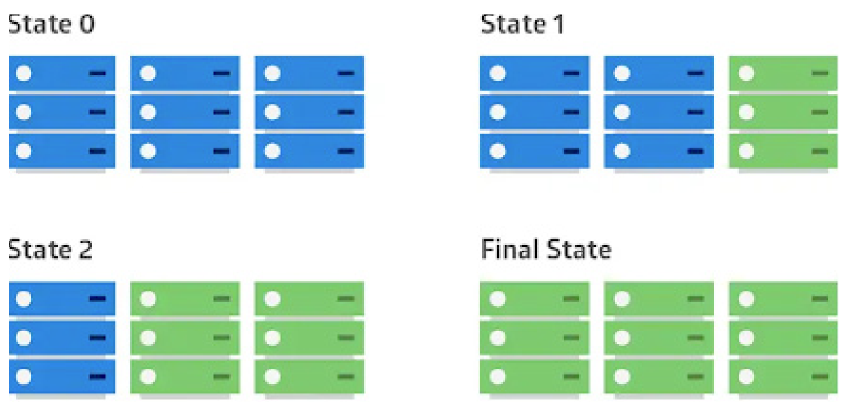


Figure: Rolling Deployment

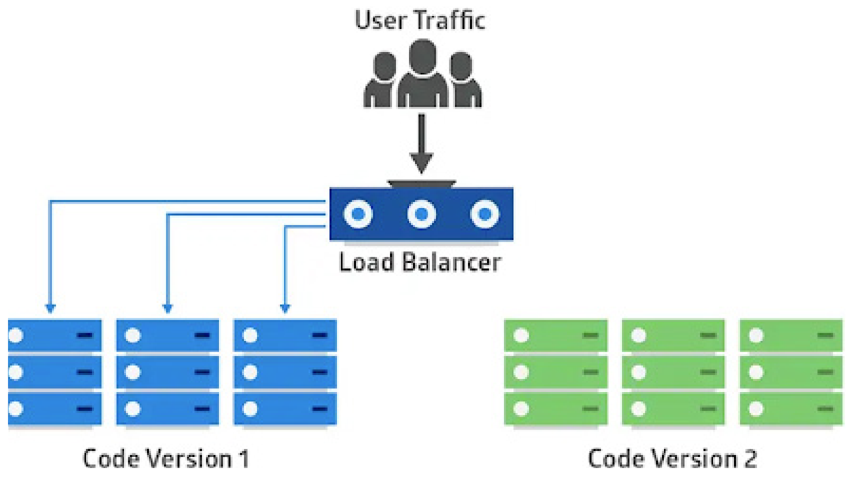


Figure: Blue/Green Deployment

Calico enables the use of any of the deployment models to upgrade multi-cluster applications.

Conclusion

Calico offers an elegant and operationally simple solution for organizations running Kubernetes at scale. The ability to create a Kubernetes cluster mesh that enables networking, security, and observability unlocks the scale at which microservices can be operated. Some of the advantages Calico offers for building a Kubernetes cluster mesh include:

- No need for an additional management plane
- Sidecar-less data plane with Envoy as a daemonset
- Multi-cluster federation across hybrid and multi-cloud
- Multi-cluster observability from network to application layer
- Blazing-fast encryption
- Industry's most advanced zero-trust security policy engine

Additional resources

O'Reilly ebook: Kubernetes security and observability – Adopt a holistic security and observability strategy for building and securing cloud-native applications running on containers and Kubernetes.

Download now

ebook: Kubernetes networking and security – Confidently approach Kubernetes networking and security, starting with basic networking concepts, all the way through to advanced Kubernetes networking with eBPF.

Download now

Do you have a question about security, observability, or compliance for containers or Kubernetes?

Contact us

Getting started

Get hands-on experience with using Calico for Kubernetes multi-cluster mesh.

Try now

Become a container and Kubernetes security and observability expert. Sign up for our weekly online webinars and workshops.

Click here

TIGERA

Tigera, Inc.

58 Maiden Lane, Fl 5
San Francisco, CA 94108

+1 (415) 612-9546 / www.tigera.io