Secure Kubernetes Ingress and Egress Controller

OVERVIEW

VoltMesh provides a secure Kubernetes ingress and egress controller with a comprehensive set of load balancing, API gateway, and multi-layer security capabilities to "connect, secure, and observe" one or multiple Kubernetes clusters. It is deployed in front of any Kubernetes cluster with automated software life-cycle management, unified policy enforcement, and aggregated metrics and logs for observability.

In addition, Volterra provides the ability to use the Volterra global network for secure, high-performance connectivity across multiple Kubernetes clusters and clouds. It allows you to directly expose select services to the public Internet, perform global traffic management, and get robust DDoS and WAF protection without using additional service providers.

OBJECTIVE

The goal of this document is to describe how to deploy a VoltMesh cluster to act as a highly available ingress and egress controller for application workloads deployed in existing Kubernetes infrastructure.

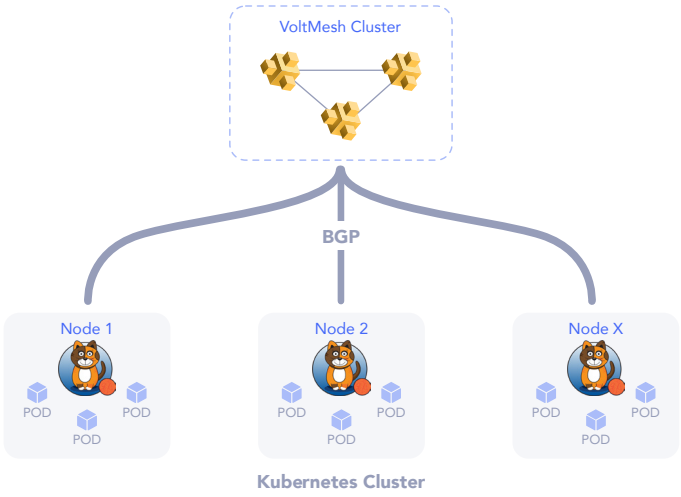
Two designs are provided in this document:

- Design One utilizes BGP (Calico) to achieve optimal load balancing performance while providing maximum redundancy
- Design Two utilizes NodePort services for maximum compatibility but with less optimal load balancing characteristics and a potential loss of redundancy for egress traffic

Configuration guidelines and examples are included for easy deployment and can be adapted where necessary.

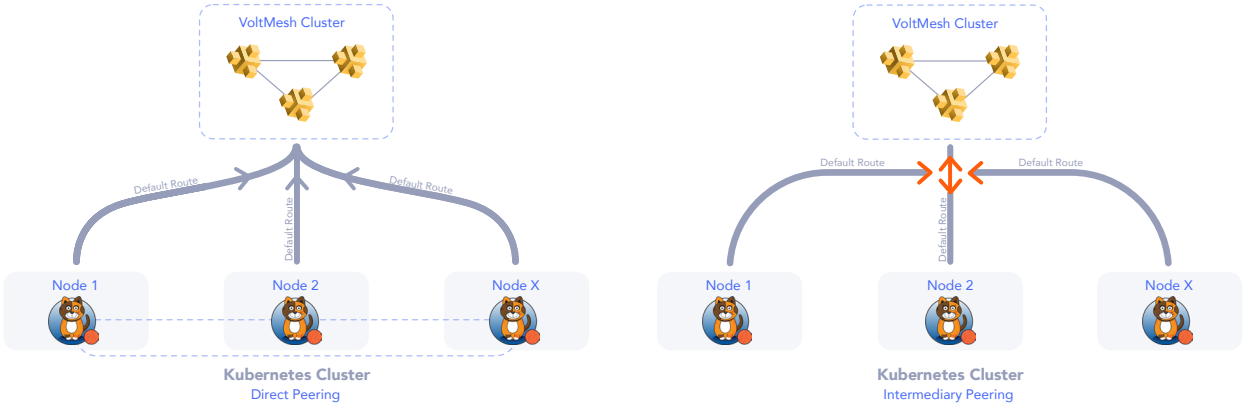
DESIGN ONE: POD ROUTING WITH CALICO + BGP

In this design, a three node VoltMesh cluster is deployed in front of an existing Kubernetes cluster with the Calico CNI network plug-in installed to provide ingress/egress traffic routing and network security to application workloads. While this solution design focuses on Calico, it may be possible to substitute other CNI plug-ins that support BGP without encapsulation.



Application endpoints are automatically discovered through Volterra's Service Discovery integration with the Kubernetes API Server for real-time detection. As application workloads are horizontally scaled in and out, these application endpoints, or Pod IPs, are automatically added and removed from the VoltMesh origin pools for zero touch load balancer reconfiguration.

Once application Pod IPs are identified through Service Discovery, Layer 3 routing between the VoltMesh cluster and application pods is enabled through BGP advertisement of Pod IP routes. All VoltMesh nodes in the cluster will peer directly with the Calico nodes or through an intermediary "top-of-rack" router. VoltMesh supports both eBGP and iBGP peering.



When directly peering VoltMesh nodes with Calico nodes, it is important to note that VoltMesh is not a route reflector server—internal peering between Calico nodes is essential to share Pod IP information within the Kubernetes cluster to avoid adding an additional hop to each Pod-to-Pod request or worse, breaking Pod-to-Pod communication. Internal peering can be enabled via full mesh (for small-scale deployments) or through designated Calico nodes acting as route reflectors (for medium to large-scale deployments).

To enable the VoltMesh egress controller functionality for application workloads, the VoltMesh cluster needs to inspect all traffic leaving the Kubernetes cluster in order to apply configurable security policies at Layers 3, 4, and 7.

If directly peering VoltMesh nodes with Calico nodes, a single VoltMesh node in the cluster becomes the default gateway or "next hop" for all traffic leaving the Kubernetes cluster. A default route is advertised from the VoltMesh cluster to all Kubernetes nodes through BGP and is installed on the host OS by Calico. If the VoltMesh node acting as the default gateway fails for any reason, the failure is detected by the VoltMesh cluster and a new gateway address is automatically propagated to the Kubernetes nodes via BGP.

Peering VoltMesh nodes with an intermediary router enables equal-cost multi-path (ECMP) between the VoltMesh cluster and the router, allowing egress traffic to be routed through all three VoltMesh nodes instead of a single node, thus taking advantage of an all active configuration instead of active/standby/standby. The default gateway propagated by the VoltMesh cluster through BGP is not advertised to the Kubernetes nodes and the next hop for each Kubernetes node remains the router.

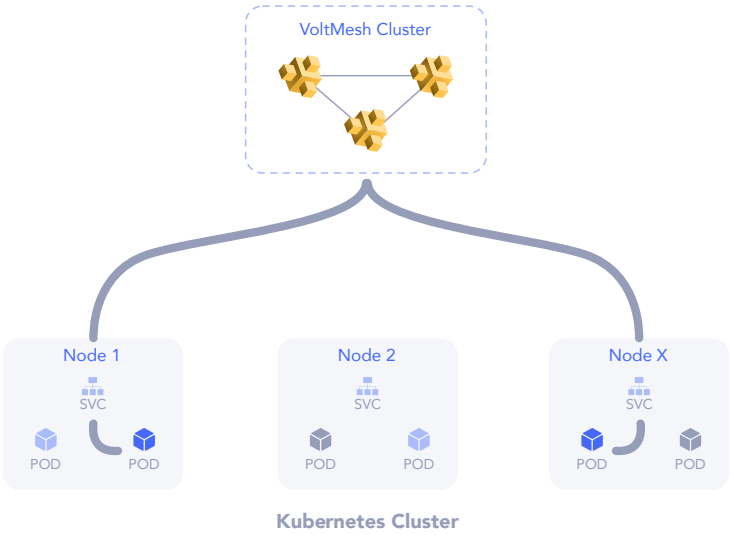
Virtual IPs, or VIPs, are leveraged for all HTTP(S) and TCP load balancers that are deployed on the VoltMesh cluster. Each VIP utilizes VRRP for high availability in case of a VoltMesh node failure and is also advertised through BGP.

Summary

This design provides high availability and fault tolerance for mission-critical applications and is the recommended deployment profile.

DESIGN TWO: NODEPORT SERVICES

In this design, a three node VoltMesh cluster is deployed in front of an existing Kubernetes cluster with any CNI network plug-in installed to provide ingress/egress traffic routing and network security to application workloads.



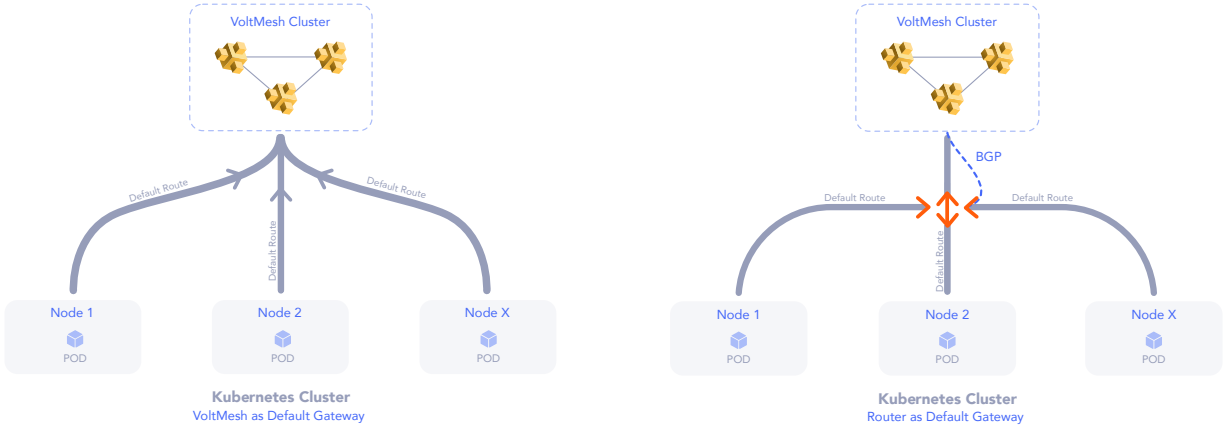
Application workloads are automatically discovered through Volterra's Service Discovery integration with the Kubernetes API Server for real-time detection. As applications are horizontally scaled in and out, VoltMesh discovers the Kubernetes node IP addresses that are hosting the application pods and automatically adds and removes the host IP addresses from the origin pool for zero touch load balancing reconfiguration.

Unlike traditional external load balancing with NodePort services, VoltMesh only discovers the Kubernetes nodes that are hosting the application, thus eliminating the need to add all Kubernetes nodes to your external load balancer server pool. In addition, deploying NodePort services with an external traffic policy set to local removes additional hops in the request path by disabling kube-proxy from proxying requests to other Kubernetes nodes in the cluster that are hosting the same application.

To enable the VoltMesh egress controller functionality for application workloads, the VoltMesh cluster needs to inspect all traffic leaving the Kubernetes cluster in order to apply configurable security policies at Layers 3, 4, and 7.

Unlike Design One, advertising a default route via BGP and enabling equal-cost multi-path (ECMP) for gateway redundancy in case of a VoltMesh node failure is not possible since the Kubernetes nodes are unable to advertise and receive BGP routes. However, the VoltMesh cluster can still act as an egress controller if each Kubernetes node designates a single VoltMesh node as its default gateway. This can be done statically or dynamically through DHCP. This approach, though, is not failure redundant—a single VoltMesh node failure may cause some or all egress traffic to be dropped.

A hybrid solution is to deploy a router between the Kubernetes cluster and VoltMesh cluster. The router exchanges BGP route information with the VoltMesh cluster and each VoltMesh node becomes part of an all active configuration through ECMP. If a VoltMesh node fails, the IP address is automatically removed from the ECMP pool on the router. The router remains the next hop for all Kubernetes nodes, thereby eliminating the need for Kubernetes to support BGP while adding a layer of high availability.



Virtual IPs, or VIPs, are leveraged for all HTTP(S) and TCP load balancers that are deployed on the VoltMesh cluster with each VIP utilizing VRRP for high availability in case of a VoltMesh node failure.

When deploying VoltMesh load balancers with NodePort services, it is not advised to run two or more of the same application pod on the same Kubernetes node due to the way kube-proxy handles persistent connections. If application pods are deployed in this manner, all client requests will be proxied to the same application pod on the Kubernetes node if a local external traffic policy is defined, or the same application pod on other Kubernetes nodes if a cluster external traffic policy is defined. This behavior diminishes the overall effectiveness of load balancing and may result in an uneven distribution of requests.

Summary

This design provides an alternative for Kubernetes deployments where Calico is not the CNI network plug-in installed, and allows application pods to be exposed through NodePort services for maximum compatibility. Note that this comes at the cost of reduced load balancing efficacy and lower fault tolerance for egress traffic.

About Volterra

Volterra provides a distributed cloud services platform to deploy, network and secure applications across multi-cloud and the edge.

Learn more about Volterra multi-cloud solutions

Visit: volterra.io
Contact Technical Sales: sales@volterra.io