

MetalLB

MetalLB is a load-balancer implementation designed for bare-metal Kubernetes clusters. It allows Kubernetes services of type LoadBalancer to be exposed with an external IP address, similar to how cloud providers assign IPs for load balancers, but MetalLB works in environments that do not have built-in cloud load balancer support. It uses standard networking protocols like ARP for Layer 2 or BGP for Layer 3 to handle traffic routing.

MetalLB manages IP address allocation from configured IP pools, which means you provide IP ranges that MetalLB can assign to services. This is useful when running Kubernetes clusters on bare metal or private networks where cloud load balancers are not an option. It supports scenarios such as colocation facilities with leased IPs or private LANs using RFC1918 address spaces.

It has two main features: IP address allocation and external announcement for service accessibility. MetalLB can be run as an operator or addon, and is compatible with Kubernetes clusters requiring fault-tolerant access to services via external IPs.

For configuration, you define pools of IP addresses, and MetalLB assigns IPs dynamically or lets you specify IPs manually for services. It integrates with native Kubernetes service types and ensures consistent and reliable load balancing.

In summary, MetalLB is a lightweight, open-source solution for adding load balancer functionality to bare-metal Kubernetes environments using standard network protocols and IP pooling.

Purpose of MetalLB: Adding the LoadBalancer service to self-hosted or on-prem Kubernetes clusters. It's an alternative when you're not using cloud-provided Kubernetes services like EKS.

Benefits: It's one of the most popular options for adding load balancers to Kubernetes.

Core Features:

Address Allocation: Configurable pools of IP addresses for load balancer services. These pools can include public, private, or a mix of IPs, and you can define multiple ranges. Auto-assignment of IPs is also mentioned.

External Announcement: How these allocated IPs are advertised on the network.

Layer 2 (L2): One machine takes ownership of the service using ARP (IPv4) or NDP (IPv6) for local traffic.

BGP (Border Gateway Protocol): Establishes peering sessions with nearby routers for true load balancing across multiple nodes.

Installation: Using kubectl apply with a manifest file.

Configuration: Defining IP address pools and advertisement methods (e.g., Layer 2) in a YAML file.

Usage Example: Showing how MetalLB assigns an IP from the defined range to a service like NGINX Ingress Controller.

Official Documentation: The video mentions the MetalLB website for more information.

MetalLB uses two main methods for **external announcement** of IP addresses:

1. **Layer 2 (L2):** With Layer 2, one machine in your Kubernetes cluster takes ownership of a service's IP address. It uses protocols like ARP (for IPv4) or NDP (for IPv6) to announce that it owns the IP address on your local network. This is effective for local traffic but means only one node actively handles the traffic for a given service at a time, even if you have multiple controller nodes.
2. **BGP (Border Gateway Protocol):** BGP allows MetalLB to establish peering sessions with nearby routers that you own. It tells these routers how to forward traffic to the service IPs. The key advantage of BGP is that it enables true load balancing across multiple nodes because the routers can distribute traffic more intelligently.

In short, Layer 2 is simpler and ideal for local network environments, but BGP offers more advanced, true load balancing capabilities by integrating with your network's routing infrastructure.