

MetalLB Installation Guide

This guide provides step-by-step instructions for installing MetalLB, a bare metal load-balancer for Kubernetes.

Prerequisites

Before installing MetalLB, ensure you:

- Meet all [requirements](#)
- Check [network addon compatibility](#)
- Review [cloud compatibility](#) if running on a cloud platform

Step 1: Prepare kube-proxy (Required for IPVS mode)

If you're using kube-proxy in IPVS mode (Kubernetes v1.14.2+), you must enable strict ARP mode.

Option A: Manual Edit

Edit the kube-proxy configmap:

```
kubectl edit configmap -n kube-system kube-proxy
```

Add or update the following configuration:

```
apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration
mode: "ipvs"
ipvs:
  strictARP: true
```

Option B: Automated Update

Check what changes would be made:

```
kubectl get configmap kube-proxy -n kube-system -o yaml | \
sed -e "s/strictARP: false/strictARP: true/" | \
kubectl diff -f - -n kube-system
```

Apply the changes:

```
kubectl get configmap kube-proxy -n kube-system -o yaml | \
sed -e "s/strictARP: false/strictARP: true/" | \
kubectl apply -f - -n kube-system
```

Step 2: Choose Your Installation Method

MetalLB supports three installation methods:

1. Kubernetes Manifests (simplest)
2. Kustomize (flexible)
3. Helm (most features)

Method 1: Installation by Manifest

Standard Installation (Native BGP)

Deploy MetalLB with native BGP implementation:

```
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.15.2/config/manifests/metallb-native.yaml
```

FRR Mode Installation

For FRR mode with BFD support:

```
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.15.2/config/manifests/metallb-frr.yaml
```

Experimental FRR-K8s Mode

For experimental FRR-K8s mode:

```
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.15.2/config/manifests/metallb-frr-k8s.yaml
```

With Prometheus Integration

For native BGP with Prometheus:

```
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.15.2/config/manifests/metallb-native-prometheus.yaml
```

Method 2: Installation with Kustomize

Step 1: Create kustomization.yml

For native BGP implementation:

```
# kustomization.yml
namespace: metallb-system

resources:
- github.com/metallb/metallb/config/native?ref=v0.15.2
```

For FRR mode:

```
# kustomization.yml
namespace: metallb-system

resources:
- github.com/metallb/metallb/config/frr?ref=v0.15.2
```

For experimental FRR-K8s mode:

```
# kustomization.yml
namespace: metallb-system

resources:
- github.com/metallb/metallb/config/frr-k8s?ref=main
```

Step 2: Apply with Kustomize

```
kubectl apply -k .
```

Method 3: Installation with Helm

Step 1: Add Helm Repository

```
helm repo add metallb https://metallb.github.io/metallb
```

Step 2: Update Helm Repositories

```
helm repo update
```

Step 3: Install MetalLB

Basic installation:

```
helm install metallb metallb/metallb
```

Installation with custom values file:

```
helm install metallb metallb/metallb -f values.yaml
```

Step 4: Configure for FRR Mode (Optional)

Create a `values.yaml` file for FRR mode:

```
speaker:
  frr:
    enabled: true
```

For experimental FRR-K8s mode:

```
frrk8s:
  enabled: true
```

Then install with:

```
helm install metallb metallb/metallb -f values.yaml
```

Step 3: Pod Security Configuration (Kubernetes 1.23+)

For Kubernetes versions with pod security admission, label the metallb-system namespace:

```
kubectl label namespace metallb-system \
  pod-security.kubernetes.io/enforce=privileged \
  pod-security.kubernetes.io/audit=privileged \
  pod-security.kubernetes.io/warn=privileged
```

Step 4: Verify Installation

Check that all MetalLB components are running:

```
kubectl get pods -n metallb-system
```

Expected output should show:

- `controller` deployment pod(s) running
- `speaker` daemonset pods running on each node

Check the deployed resources:

```
kubectl get all -n metallb-system
```

Using MetalLB Operator (Alternative)

Step 1: Install from OperatorHub

Visit operatorhub.io/operator/metallb-operator and follow the installation instructions for your cluster.

Step 2: Configure BGP Type (Optional)

For FRR mode, edit the ClusterServiceVersion:

```
kubectrl edit csv metallb-operator
```

Change the `BGP_TYPE` environment variable:

```
- name: METALLB_BGP_TYPE
  value: frr # or 'frr-k8s' for experimental mode
```

Advanced Configuration

Setting LoadBalancer Class

To use a specific LoadBalancer class, add the `--lb-class=<CLASS_NAME>` parameter to both speaker and controller.

For Helm installations, use:

```
loadBalancerClass: <CLASS_NAME>
```

FRR Logging Level Configuration

Configure FRR daemons logging by setting the speaker `--log-level` or use the `FRR_LOGGING_LEVEL` environment variable.

Mapping:

- `all, debug` → `debugging`
- `info` → `informational`
- `warn` → `warnings`
- `error` → `error`
- `none` → `emergencies`

Upgrading MetalLB

Step 1: Check Release Notes

Always review the [release notes](#) before upgrading.

Step 2: Upgrade Using Your Installation Method

For Manifest:

```
kubectrl apply -f https://raw.githubusercontent.com/metallb/metallb/<NEW_VERSION>/config/manifests/metallb-native.yaml
```

For Helm:

```
helm upgrade metallb metallb/metallb
```

For Kustomize: Update the version in `kustomization.yaml` and reapply.

Next Steps

After installation, MetalLB will remain idle until configured. You need to:

1. Create an `IPAddressPool` resource to define available IP addresses
2. Create an `L2Advertisement` or `BGPAdvertisement` resource
3. Deploy services with `type: LoadBalancer`

Example configuration:

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: first-pool
  namespace: metallb-system
spec:
  addresses:
    - 192.168.1.240-192.168.1.250
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: example
  namespace: metallb-system
```

Apply configuration:

```
kubectl apply -f config.yaml
```

Troubleshooting

Check Controller Logs

```
kubectl logs -n metallb-system deployment/controller
```

Check Speaker Logs

```
kubectl logs -n metallb-system daemonset/speaker
```

Verify Service Assignment

```
kubectl get svc
```

Check Events

```
kubectl get events -n metallb-system --sort-by='.lastTimestamp'
```

Quick Reference Commands

```
# Check MetalLB version
kubectl get deployment -n metallb-system controller -o jsonpath='{.spec.template.spec.containers[0].image}'

# Get all MetalLB resources
kubectl get all -n metallb-system

# Describe a service to see LoadBalancer details
kubectl describe svc <service-name>

# Get IPAddressPools
kubectl get ipaddresspool -n metallb-system

# Get L2Advertisements
kubectl get l2advertisement -n metallb-system

# Get BGPAdvertisements
kubectl get bgpadvertisement -n metallb-system
```