

19-metallb

MetalLB Goal Set up a LoadBalancer service on a bare-metal or local Kubernetes cluster using MetalLB. Prerequisites Kubernetes cluster is running (`kubectl works`) You can reach cluster nodes on your LAN You have a free IP range in the same subnet as your nodes Step 1: Install MetalLB Create the namespace and components: `bash kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.14.9/config/manifests/metallb-native.yaml` Wait for controller and speaker pods: `bash kubectl -n metallb-system get pods` Step 2: Choose an address pool Pick a range that is **not** used by DHCP, routers, or existing hosts. For **Minikube**, use the Minikube node subnet (check with `minikube ip`). Example (Minikube IP is 192.168.49.2): - Node subnet: 192.168.49.0/24 - Reserved MetalLB pool: 192.168.49.100-192.168.49.120 Do **not** use 10.96.0.0/12 for MetalLB. That range is usually the Kubernetes Service CIDR. Step 3: Create IPAddressPool yaml `apiVersion: metallb.io/v1beta1 kind: IPAddressPool metadata: name: first-pool namespace: metallb-system spec: addresses: - 192.168.49.100-192.168.49.120` Apply it: `bash kubectl apply -f ipaddresspool.yaml` Step 4: Create L2Advertisement yaml `apiVersion: metallb.io/v1beta1 kind: L2Advertisement metadata: name: l2 namespace: metallb-system spec: ipAddressPools: - first-pool` Apply it: `bash kubectl apply -f l2advertisement.yaml` Step 5: Expose an app with LoadBalancer yaml `apiVersion: v1 kind: Service metadata: name: web-lb spec: type: LoadBalancer selector: app: web ports: - port: 80 targetPort: 80` Apply and check: `bash kubectl apply -f service-lb.yaml` `kubectl get svc web-lb` The EXTERNAL-IP should be assigned from your MetalLB pool. Step 6: Verify traffic Open `http://<EXTERNAL-IP>` in browser, or Run: `bash curl http://<EXTERNAL-IP>` Troubleshooting bash `kubectl -n metallb-system get all` `kubectl -n metallb-system logs deploy/controller` `kubectl -n metallb-system logs daemonset/speaker` `kubectl describe svc web-lb` Cleanup `bash kubectl delete svc web-lb` `kubectl delete -f l2advertisement.yaml` `kubectl delete -f ipaddresspool.yaml` Practical notes MetalLB does not create cloud load balancers; it announces IPs on your network. In most home/lab clusters, L2Advertisement mode is the simplest option. If no external IP is assigned, first check IP pool overlap/conflicts. Minikube note (tunnel vs MetalLB) route: 10.96.0.0/12 -> 192.168.49.2 from minikube tunnel is expected and refers to Service CIDR routing. If you use MetalLB, assign from Minikube subnet (for example 192.168.49.x), not from 10.96.0.0/12. In local labs, use either minikube tunnel path or MetalLB path for LoadBalancer testing; avoid mixing both until basic flow works.