

MetalLB Load Balancing Configuration Guide for OpenShift 4.17

Prerequisites

1. OpenShift Container Platform 4.17 cluster installed
2. OpenShift CLI (oc) installed
3. User with `cluster-admin` privileges
4. MetalLB Operator installed (refer to networking operators documentation)
5. Network configured to route traffic from clients to the cluster host network

Step 1: Install MetalLB Operator

Before configuring MetalLB, ensure the MetalLB Operator is installed in your cluster. The operator should be deployed in the `metallb-system` namespace.

Step 2: Create IP Address Pool

Create an IP address pool that MetalLB will use to assign IP addresses to LoadBalancer services.

Basic IP Address Pool Configuration

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  namespace: metallb-system
  name: doc-example
  labels:
    zone: east
spec:
  addresses:
    - 203.0.113.1-203.0.113.10
    - 203.0.113.65-203.0.113.75
```

Apply the configuration:

```
oc apply -f ipaddresspool.yaml
```

IP Address Pool with VLAN Support

For VLAN-specific configurations:

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  namespace: metallb-system
  name: doc-example-vlan
  labels:
    zone: east
spec:
  addresses:
    - 203.0.114.1-203.0.114.10 # Must match VLAN subnet
```

After creating the pool with VLAN, configure IP forwarding:

```
oc edit network.operator.openshift/cluster
```

Add under `spec.defaultNetwork.ovnKubernetesConfig`:

```
gatewayConfig:
  ipForwarding: Global
```

Step 3: Configure Advertisement Method

Choose between Layer 2 (L2) or BGP advertisement based on your network requirements.

Option A: Layer 2 Advertisement

For simple deployments without BGP:

```
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: example-l2
  namespace: metallb-system
spec:
  ipAddressPools:
    - doc-example
```

Apply the L2 advertisement:

```
oc apply -f l2advertisement.yaml
```

Option B: BGP Advertisement

For production deployments with BGP routing:

Step 3.1: Create BGP Peer

```
apiVersion: metallb.io/v1beta2
kind: BGPPeer
metadata:
  namespace: metallb-system
  name: doc-example-peer
spec:
  peerAddress: 10.0.0.1
  peerASN: 64512
  myASN: 64513
  routerID: 10.10.10.10
  peerPort: 179
```

Apply BGP peer:

```
oc apply -f bgppeer.yaml
```

Step 3.2: Create BGP Advertisement

```
apiVersion: metallb.io/v1beta1
kind: BGPAdvertisement
metadata:
  name: example-bgp
  namespace: metallb-system
spec:
  ipAddressPools:
    - doc-example
  peers:
    - doc-example-peer
  aggregationLength: 32
```

Apply BGP advertisement:

```
oc apply -f bgpadvertisement.yaml
```

Step 4: Configure Advanced Features (Optional)

BFD Profile for Faster Failure Detection

```
apiVersion: metallb.io/v1beta1
kind: BFDProfile
metadata:
  name: example-bfd
  namespace: metallb-system
spec:
  receiveInterval: 200
  transmitInterval: 200
  detectMultiplier: 3
  echoInterval: 100
  echoMode: false
  passiveMode: false
  minimumTtl: 254
```

Apply and associate with BGP peer:

```
apiVersion: metallb.io/v1beta2
kind: BGPPeer
metadata:
  namespace: metallb-system
  name: doc-example-peer-bfd
spec:
  peerAddress: 10.0.0.1
  peerASN: 64512
  myASN: 64513
  bfdProfile: example-bfd
```

Community Aliases for BGP

```
apiVersion: metallb.io/v1beta1
kind: Community
metadata:
  name: community-example
  namespace: metallb-system
spec:
  communities:
    - name: NO_ADVERTISE
      value: "65535:65282"
```

Step 5: Deploy a Service

Create a LoadBalancer service to test the configuration:

```
apiVersion: v1
kind: Service
metadata:
  name: example-service
  annotations:
    metallb.io/address-pool: doc-example # Optional: specify
    pool
spec:
  type: LoadBalancer
  loadBalancerIP: 203.0.113.5 # Optional: request specific IP
  selector:
    app: example
  ports:
    - port: 80
      targetPort: 8080
      protocol: TCP
```

Apply the service:

```
oc apply -f service.yaml
```

Step 6: Verification

Verify IP Address Pool

```
oc describe -n metallb-system IPAddressPool doc-example
```

Verify Service External IP

```
oc get service example-service
```

Expected output should show EXTERNAL-IP assigned from your pool.

For BGP Deployments - Check BGP Session

```
# Get speaker pod
```

```
oc get -n metallb-system pods -l component=speaker
```

```
# Check BGP summary
```

```
oc exec -n metallb-system <speaker-pod> -c frr -- vtysh -c "show
  bgp summary"
```

```
# Check BGP configuration
oc exec -n metallb-system <speaker-pod> -c frr -- vtysh -c "show
    running-config"
```

For L2 Deployments - Check L2 Status

```
oc describe -n metallb-system L2Advertisement example-l2
```

Step 7: Advanced Configurations

Enable IP Forwarding for Secondary Interfaces

For secondary interfaces, enable IP forwarding:

```
# For specific interface
echo -e "net.ipv4.conf.bridge-net.forwarding =
    1\nnet.ipv6.conf.bridge-net.forwarding = 1" | base64 -w0
```

Create MachineConfig:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 99-enable-ip-forward
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - path: /etc/sysctl.d/enable-ip-forward.conf
          mode: 0644
          contents:
            source: data:text/plain;charset=utf-8;base64,<BASE64_ENCODED_STRING>
```

Configure VRF for Network Isolation

For VRF-based routing (Technology Preview):

1. Create NodeNetworkConfigurationPolicy for VRF
2. Configure BGPPeer with VRF
3. Set up EgressService for symmetric routing

Troubleshooting

Enable Debug Logging

```
apiVersion: metallb.io/v1beta1
kind: MetalLB
metadata:
  name: metallb
  namespace: metallb-system
spec:
  logLevel: debug
```

Check Speaker Logs

```
oc logs -n metallb-system <speaker-pod> -c speaker
oc logs -n metallb-system <speaker-pod> -c frr
```

Collect Diagnostic Information

```
oc adm must-gather
```

Best Practices

1. **IP Address Planning:** Ensure IP ranges don't overlap with existing network infrastructure
2. **High Availability:** Use multiple speaker pods across different nodes
3. **BGP Configuration:** For production, use BGP with BFD for faster failover
4. **Security:** Use BGP authentication with passwords or secrets
5. **Monitoring:** Set up Prometheus metrics for MetalLB components
6. **Network Isolation:** Consider VRF for multi-tenant scenarios
7. **Documentation:** Document your IP address allocations and BGP peer configurations

Common Issues and Solutions

Issue: Service stuck in Pending state

- Verify IP address pool has available addresses
- Check speaker pod logs for allocation errors
- Ensure MetalLB pods are running

Issue: BGP session not established

- Verify network connectivity to BGP peer
- Check BGP peer configuration matches
- Review firewall rules for port 179

Issue: Traffic not reaching service

- Verify routing configuration
- Check if IP forwarding is enabled (for secondary interfaces)
- Ensure network allows ARP/NDP for L2 mode

Summary

This guide covers the essential steps to configure MetalLB for load balancing on OpenShift: 1. Create IP address pools 2. Configure advertisement (L2 or BGP) 3. Deploy services with type LoadBalancer 4. Verify and troubleshoot the configuration

For production environments, BGP with BFD is recommended for better scalability and faster failure detection. For simpler deployments, L2 mode provides an easy-to-configure solution without requiring BGP infrastructure.