Oracle Cloud Native Environment

MetalLB Module for Release 1.9

F93855-01
May 2024

Oracle Cloud Native Environment MetalLB Module for Release 1.9,

F93855-01

Contents

Preface

This document contains information about setting up and using the MetalLB module to set up network load balancers for Kubernetes applications in Oracle Cloud Native Environment. MetalLB can provide load balancers that can be used with Kubernetes LoadBalancer services to externalize applications outside of the Kubernetes cluster. MetalLB is used to provide load balancers on bare metal hosts.

Documentation License

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

Introduction to the MetalLB Module

Network load balancers provide a method of externally exposing Kubernetes applications. A Kubernetes LoadBalancer service is used to create a network load balancer that provides and exposes an external IP address that can be used to connect to an application from outside the cluster.

More information on Kubernetes services, including the LoadBalancer service, is available in the upstream Kubernetes documentation.

MetalLB is a network load balancer for Kubernetes applications running on bare metal hosts. MetalLB lets you use Kubernetes LoadBalancer services, which traditionally use a cloud provider network load balancer, in a bare metal environment.

MetalLB has two features that enable the network load balancer: address allocation, and external announcement.

Address allocation provides IP addresses to Kubernetes applications from the pool of IP addresses you provide in the MetalLB configuration file.

External announcement makes the network beyond the Kubernetes cluster aware that the IP is available in the cluster. This is provided using either Address Resolution Protocol (ARP) and Neighbor Discover Protocol (NDP) in Layer 2 mode, or Border Gateway Protocol (BGP) in BGP mode.

For more information on MetalLB, see the upstream MetalLB documentation.

The MetalLB module is used to set up network load balancers for Kubernetes applications using MetalLB.

Oracle Cloud Native Environment deploys MetalLB onto the control plane nodes using a configuration file you set up beforehand.

2
Installing the MetalLB Module

This chapter discusses how to install the MetalLB module to set up application load balancers for Kubernetes applications using MetalLB on bare metal Oracle Cloud Native Environment instances.

Prerequisites

This section contains the prerequisite information you need to set up the MetalLB module.

Setting up the Health Check Endpoint Network Ports

When using a Kubernetes LoadBalancer service with the `ServiceInternalTrafficPolicy` set to `Cluster` (the default), a health check endpoint is expected to be available on TCP port 10256. `kube-proxy` creates a listener on this port, which provides access to the LoadBalancer service to verify that `kube-proxy` is healthy on the nodes. The LoadBalancer service decides which nodes can have traffic routed to them using this policy. To allow traffic on this port, you must open TCP port 10256 on all Kubernetes nodes. On each Kubernetes node, run:

```
sudo firewall-cmd --zone=public --add-port=10256/tcp
sudo firewall-cmd --zone=public --add-port=10256/tcp --permanent
sudo systemctl restart firewalld.service
```

For more information on the `ServiceInternalTrafficPolicy`, see the upstream Kubernetes documentation.

Ensure traffic is allowed for TCP port 10256 in the network security list.

Setting up the Network Ports

You must open the following ports on Kubernetes worker nodes. On each worker node, run:

```
sudo firewall-cmd --zone=public --add-port=7946/tcp --permanent
sudo firewall-cmd --zone=public --add-port=7946/udp --permanent
sudo systemctl restart firewalld.service
```

Creating a MetalLB Configuration File

You must provide a MetalLB configuration file on the **operator** node to configure your MetalLB instance. In the configuration file, which must be in the `YAML` custom resource definition (CRD) file format, you can configure MetalLB using options such as `IPAddressPools`, `BGPPeer`, `BGPAdvertisement`, and `L2Advertisement`.

The Platform API Server uses the information contained in the configuration file when creating the MetalLB module.

For information on the options available to use in the configuration file, see the upstream MetalLB documentation.

The following example configuration file uses a MetalLB Layer 2 configuration and provides the IP address range from 192.168.1.240 to 192.168.1.250 to MetalLB to create load balancer IPs for Kubernetes applications. This example file is named `metallb-config.yaml` and contains:

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  creationTimestamp: null
  name: default
  namespace: metallb
spec:
  addresses:
  - 192.168.1.240-192.168.1.250
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  creationTimestamp: null
  name: l2advertisement1
  namespace: metallb
spec:
  ipAddressPools:
  - default
```

Deploying the MetalLB Module

You can deploy all the modules required to set up MetalLB for a Kubernetes cluster using a single `olcnectl module create` command. This method might be useful to deploy the MetalLB module at the same time as deploying a Kubernetes cluster.

If you have an existing deployment of the Kubernetes module, you can specify that instance when deploying the MetalLB module.

This section guides you through installing each component required to deploy the MetalLB module.

For the full list of the Platform CLI command options available when creating modules, see the `olcnectl module create` command in Platform Command-Line Interface.

To deploy the MetalLB module:

1. If you don't already have an environment set up, create one into which the modules can be deployed. For information on setting up an environment, see Installation. The name of the environment in this example is `myenvironment`.
2. If you don't already have a Kubernetes module set up or deployed, set one up. For information on adding a Kubernetes module to an environment, see Kubernetes Module. The name of the Kubernetes module in this example is `mycluster`.
3. Create a MetalLB module and associate it with the Kubernetes module named `mycluster` using the `--metallb-kubernetes-module` option. In this example, the MetalLB module is named `mymetallb`.

   ```
   olcnectl module create \
   --environment-name myenvironment \
   ```

```
--module metallb \
--name mymetallb \
--metallb-kubernetes-module mycluster \
--metallb-config metallb-config.yaml
```

The `--module` option sets the module type to create, which is `metallb`. You define the name of the MetalLB module using the `--name` option, which in this case is `mymetallb`.

The `--metallb-kubernetes-module` option sets the name of the Kubernetes module.

The `--metallb-config` option sets the location for the MetalLB configuration file. This file must be available on the operator node under the provided path. For information on creating this configuration file, see Creating a MetalLB Configuration File.

If you do not include all the required options when adding the module, you're prompted to provide them.

4. Use the `olcnectl module install` command to install the MetalLB module. For example:

```
olcnectl module install \
--environment-name myenvironment \
--name mymetallb
```

You can optionally use the `--log-level` option to set the level of logging displayed in the command output. By default, error messages are displayed. For example, you can set the logging level to show all messages when you include:

```
--log-level debug
```

The log messages are also saved as an operation log. You can view operation logs as commands are running, or when they've completed. For more information using operation logs, see Platform Command-Line Interface.

The MetalLB module is deployed into the Kubernetes cluster.

Verifying the MetalLB Module Deployment

You can verify the MetalLB module is deployed using the `olcnectl module instances` command on the operator node. For example:

```
olcnectl module instances \
--environment-name myenvironment
```

The output looks similar to:

```
INSTANCE                 MODULE        STATE
mymetallb                metallb       installed
mycluster                kubernetes    installed
...
```

Note the entry for `metallb` in the `MODULE` column is in the `installed` state.

In addition, use the `olcnectl module report` command to review information about the module. For example, use the following command to review the MetalLB module named `mymetallb` in `myenvironment`:

```
olcnectl module report \
--environment-name myenvironment \
--name mymetallb \
--children
```

For more information on the syntax for the `olcnectl module report` command, see Platform Command-Line Interface.

3
Using MetalLB Load Balancers

This chapter discusses how to install and configure the MetalLB module, as a custom resource, to set up a network load balancer for Kubernetes applications in a bare metal environment.

Creating an Application Using MetalLB

This section contains a basic test to verify you can create a Kubernetes application that uses MetalLB to provide external IP addresses.

To create a test application to use MetalLB:

1. Create a Kubernetes application that uses a LoadBalancer service. The deployment in this example creates an NGINX application with a replica count of 2, and an associated LoadBalancer service.

   On a control plane node, create a file named `nginx-metallb.yaml` and copy the following into the file.

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: container-registry.oracle.com/olcne/nginx:1.17.7
        ports:
        - containerPort: 80
---
kind: Service
apiVersion: v1
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
```

```
  type: LoadBalancer
  ports:
  - name: http
    port: 80
    targetPort: 80
```

2. Start the NGINX deployment and LoadBalancer service:

   ```
   kubectl apply -f nginx-metallb.yaml
   ```

3. You can see the `nginx-deployment` application is running using the `kubectl get deployment` command:

   ```
   kubectl get deployments.apps
   ```

   The output looks similar to:

   ```
   NAME               READY   UP-TO-DATE   AVAILABLE   AGE
   nginx-deployment   2/2     2            2           31s
   ```

4. You can see the `nginx-deployment` service is running using the `kubectl get svc` command:

   ```
   kubectl get svc
   ```

   The output looks similar to:

   ```
   NAME            TYPE           CLUSTER-IP     EXTERNAL-IP
   PORT(S)        AGE
   kubernetes      ClusterIP      10.96.0.1      <none>
   443/TCP        25h
   nginx-service   LoadBalancer   10.99.253.99   192.168.1.240
   80:31875/TCP   70s
   ```

   You can see the EXTERNAL-IP for the `nginx-service` LoadBalancer has an IP address of `192.168.1.240`. This IP address is provided by MetalLB and is the external IP address that you can use to connect to the application.

5. Use `curl` to connect to the NGINX application's IP address and add the port for the application (`192.168.1.240:80` in this example) to show the NGINX default page.

   ```
   curl 192.168.1.240:80
   ```

   The output looks similar to:

   ```
   <!DOCTYPE html>
   <html>
   <head>
   <title>Welcome to nginx!</title>
   <style>
       body {
   ```

```
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed
and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

6. You can delete the `nginx-service` LoadBalancer service using:

```
kubectl delete svc nginx-service
```

7. You can delete the `nginx-deployment` application using:

```
kubectl delete deployments.apps nginx-deployment
```

4
Removing the MetalLB Module

You can remove a deployment of the MetalLB module and leave the Kubernetes cluster in place. To do this, you remove the MetalLB module from the environment.

Use the `olcnectl module uninstall` command to remove the MetalLB module. For example, to uninstall the MetalLB module named `mymetallb` in the environment named `myenvironment`:

```
olcnectl module uninstall \
--environment-name myenvironment \
--name mymetallb
```

The MetalLB module is removed from the environment.