MetalLB 介绍与使用

AUTHOR: 彭玲 TIME: 2022/1/26

MetalLB 简介

MetalLB 是一个 Kubernetes 感知的解决方案，它将监视 `LoadBalancer` 类型的 services，并且为这些 services 分配一个来自虚拟池的 IP 地址。

K8s `LoadBalancer` 类型的 `Service` 依赖于外部的云提供的 Load Balancer。当我们把 k8s 部署在裸机上面时，或者是测试环境时，需要简单的 LoadBalancer 来验证工作，开源的 metallb 就是一个不错的选择。

MetalLB 工作模式

Metallb 支持两种工作模式：Layer2 模式、BGP 模式（需要上层路由器支持）。

- MetalLB 支持本地流量，这意味着接收数据的机器将是为请求提供服务的机器。不建议使用具有高流量工作负载的虚拟 IP，因为只有一台机器会接收服务的流量 - 其他机器仅用于 `故障转移`。
- BGP 没有此限制，BGP 将 `节点` 视为原子单元。这意味着如果 Service 在 5 个节点中的 2 个节点上运行，那么只有这 2 个节点会接收流量，但即使其中一个节点有 3 个 pod 而另一个节点上只有 1 个 pod 运行，它们也会各自接收 50% 的流量。建议使用节点反亲和防止 Kubernetes pod 堆叠在单个节点上。

MetalLB 安装

安装 MetalLB 支持三种方式：使用 Kubernetes 清单、使用 Kustomize 或 Helm。下面使用 K8s 清单进行安装。

1. 安装 MetalLB pod

K8s 清单 将 MetalLB 部署到`metallb-system`命名空间下的集群，相关组件：

- `metallb-system/controller`部署：处理 IP 地址分配的集群范围的控制器。
- `metallb-system/speaker`守护进程：使用您选择的协议以使服务可访问。
- `controller`和`speaker`对应的 Service accounts，以及组件运行所需的 RBAC 权限。

安装清单不包含配置文件。MetalLB 的组件仍将启动，但将保持`空闲`状态，直到定义和部署 ConifgMap。

```
# 创建命名空间：metallb-system
anxin@node38:~$ kubectl apply -f
https://raw.githubusercontent.com/metallb/metallb/v0.11.0/manifests/namespac
e.yaml
namespace/metallb-system created

# 安装 metallb pod
anxin@node38:~$ kubectl apply -f
https://raw.githubusercontent.com/metallb/metallb/v0.11.0/manifests/metallb.
yaml
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created # serviceaccount controller
serviceaccount/speaker created # serviceaccount speaker
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
role.rbac.authorization.k8s.io/controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller
created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
rolebinding.rbac.authorization.k8s.io/controller created
daemonset.apps/speaker created # metallb-system/speaker 守护进程
deployment.apps/controller created # metallb-system/controller 部署
```

2. 配置 MetalLB

layer2 模式不需要任何特定于协议的配置，只需要 IP 地址。它通过直接响应本地网络上的 ARP 请求来工作，将机器的 MAC 地址提供给客户端。

下面对`layer2 模式`进行配置。以下配置使 MetalLB 可以控制`10.244.1.240-10.244.1.250`的 IP，并配置 Layer 2 模式：

```
anxin@node38:~/pengling/k8s/metallb-demo$ vi metallb-config.yaml

apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
```

```
  name: metallb-config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - 10.244.1.240-10.244.1.250 # 跟 K8s 节点的管理网是同一个 /24 地址段 (node
.spec.podCIDR)
```

应用 ConfigMap 配置：

```
anxin@node38:~/pengling/k8s/metallb-demo$ kubectl apply -f metallb-
config.yaml
configmap/metallb-config created
```

MetalLB 用法

在安装和配置 MetalLB 之后，要向外部公开服务，只需将其创建并`spec.type`设置为`LoadBalancer`，然后 MetalLB 将完成剩下的工作。

请求特定 IP

MetalLB 遵守`spec.loadBalancerIP`参数，因此如果希望使用特定地址设置您的服务，您可以通过设置该参数来请求它。

如果您想要某种地址但不关心具体是哪一个，MetalLB 还支持请求特定的`地址池`。

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    metallb.universe.tf/address-pool: production-public-ips # 请求特定的地址池
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

流量策略

MetalLB 理解并遵守 Service 的`externalTrafficPolicy`选项，并根据您选择的策略和`公告协议`实现不同的`公告模式`。下面对 Layer2 mode 进行说明。

在 layer2 模式下`发布`时，集群中的一个节点将吸引服务 IP 的流量。从那里开始，行为取决于所选的流量策略。

“Cluster” 流量策略

使用默认`Cluster`流量策略，`kube-proxy`在接收到流量的节点上进行负载均衡，并将流量分发到服务中的所有 Pod。

此策略导致服务中所有 pod 的流量分布均匀。但是，`kube-proxy`在进行负载均衡时会掩盖连接的源 IP 地址，因此您的 pod 日志将显示外部流量似乎来自`服务`的`leader`节点。

“Local” 流量策略

使用`Local`流量策略，`kube-proxy`在接收流量的节点上，仅将流量发送到同一节点上的 service’s pod(s)。节点之间没有“水平”的流量。

因为`kube-proxy`不需要在集群节点之间发送流量，所以你的 Pod 可以看到传入连接的真实源 IP 地址。

此策略的缺点是`传入流量`仅流向`service`中的某些 pods。不在当前`leader`节点上的 Pod 不会接收流量，它们只是作为副本存在，以防需要`故障转移`。

IP 地址共享

默认情况下，服务不共享 IP 地址。如果需要`在单个 IP 上托管服务`，您可以通过向 services 添加`metallb.universe.tf/allow-shared-ip`注解来启用选择性 IP 共享，注解的值是一个`共享密钥`。

应用场景：解决 Kubernetes 的限制，以及使用有限的 IP 地址。

解决 Kubernetes 的限制

Kubernetes 目前不允许多协议 LoadBalancer 服务。这通常会使像 DNS 这样的服务无法运行，因为它们必须同时侦听 TCP 和 UDP。要使用 MetalLB 解决 Kubernetes 的这一限制：

- 创建两个服务 (一个用于 TCP，一个用于 UDP)，它们都具有相同的 pod 选择器。
- 然后，为它们提供`相同的共享密钥`和`spec.loadBalancerIP`，以将 TCP 和 UDP 服务端口配置在同一个 IP 地址上。

使用有限的 IP 地址

如果您拥有的服务多于可用 IP 地址，并且您不能或不想获得更多地址，唯一的选择是为每个 IP 地址`托管`多个服务。

MetalLB 使用示例

实验环境：node38 集群。

nginx 清单

```
anxin@node38:~/pengling/k8s/metallb-demo$ vi example-nginx-lb.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 3
```

```
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-container
        image: nginx
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-lb
  annotations:
    metallb.universe.tf/address-pool: production-public-ips
spec:
  type: LoadBalancer
  selector:
    app: nginx
  ports:
  - port: 80
    targetPort: 80
```

nginx 部署

```
anxin@node38:~/pengling/k8s/metallb-demo$ kubectl apply -f example-nginx-lb.yaml
deployment.apps/nginx-deployment created
service/nginx-lb created
```

nginx 部署结果

```
# EXTERNAL-IP 为 <pending>
anxin@node38:~$ kubectl get svc nginx-lb -o wide
NAME       TYPE         CLUSTER-IP    EXTERNAL-IP   PORT(S)       AGE   SELECTOR
nginx-lb   LoadBalancer   10.103.7.18   <pending>     80:30432/TCP   105s   app=nginx
```

使用浏览器访问：http://10.8.30.38:30432/ 或 http://10.8.30.37:30432/ 或 http://10.8.30.35:30432/。