

Installation von MetalLB (Lokaler LoadBalancer)

Einleitung

In dieser kurzen Anleitung beschreibe ich kurz, wie wir mit der Hilfe von MetalLB einen lokalen LoadBalancer betreiben können. In der Regel verwendet man einen LoadBalancer innerhalb eines Cloud-Providers und dieser stellt dann einen kostenpflichtigen LoadBalancer zur Verfügung. Sobald wir aber im LAN einen solchen LoadBalancer über die Service-Konfiguration anfordern, sollte die Anfrage auf `"pending"` stehen bleiben und keine IP erhalten. Mit diesem LoadBalancer sind die internen Anwendungen dann erreichbar über eine dedizierte IP-Adresse.

Durchführung

Im ersten Schritt führen wir den folgenden Befehl auf unserem Kubernetes Master Node aus:

```
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.14.8/config/manifests/metallb-native.yaml
```

Damit wird dann die Konfiguration für den LoadBalancer Pod heruntergeladen und gestartet. Um zu überprüfen, ob die Pods erfolgreich gestartet wurden, können wir den folgenden Befehl ausführen:

```
kubectl get pods --namespace metallb-system
```

Wenn hier bei allen Containern der Status auf `Running` steht, sollten die Pods ordnungsgemäß hochgefahren sein. Wir müssen dann im Anschluss eine neue YAML-Datei anlegen, in dem wir den IPv4-Bereich definieren, welcher vom LoadBalancer verwendet werden darf. Die Datei sieht folgendermaßen aus:

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: <pool-name>
  namespace: metallb-system
spec:
  addresses:
    - <ip-von_ip-bis>
```

Jetzt müssen wir eine weitere YAML-Datei erstellen. Diese enthält das "L2Advertisement" und enthält den entsprechenden IP-Pool, der zur Vergabe der IP-Adressen verwendet werden darf. Die Datei sieht wie folgt aus:

```
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: l2metallb
  namespace: metallb-system
spec:
  ipAddressPools:
  - <pool-name>
```

Diese beiden Konfigurationen werden dann auch wieder einmal über `kubectly apply -f <dateiname>` aktiviert. Wenn alles geklappt hat, können wir in der Service-Konfiguration den Typen auf LoadBalancer setzen. Wenn wir jetzt die Service-Konfiguration aktualisieren, sollten wir mit dem folgenden Befehl dann die IP-Adresse unseres LoadBalancers sehen können.

```
kubectly get services --all-namespaces
```

Revision #3

Created 23 October 2024 18:48:36 by Phillip U.

Updated 29 October 2024 10:17:46 by Phillip U.