Kubernetes Headless-Service

Übersicht

Ein Service ist eine Kubernetes-Ressource, die einen einzigen Einstiegspunkt zu einem oder mehreren Pods bereitstellt. Genauer gesagt stellt ein Service eine Gruppe von Pods über einen einzigen Domänennamen und eine IP-Adresse im Netzwerk zur Verfügung. Sobald eine Anfrage den Service erreicht, leitet kube-proxy die Anfrage an einen der dahinterliegenden Pods weiter.

In diesem Tutorial lernen wir eine spezialisierte Form von Service-Ressourcen kennen, den sogenannten Headless Service. Im Gegensatz zum herkömmlichen Service erlaubt uns ein Headless Service, die IP-Adressen der einzelnen unterstützenden Pods direkt zu erhalten. Diese Fähigkeit eröffnet interessante Anwendungsfälle, die mit einem normalen Service nicht möglich wären.

Wir werden zunächst die grundlegende Konfiguration eines Headless Service behandeln und anschließend an einer praktischen Demonstration sehen, wie man ihn in der Praxis einsetzen kann.

Headless Service

In Kubernetes bezeichnet man einen Headless Service als eine Service-Ressource, der keine Cluster-IP-Adresse zugewiesen wird.

Um einen solchen Headless Service zu definieren, setzt man in der Ressourcen-Definition das Feld `spec.clusterIP` auf den Wert `None`:

```
$ cat headless-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: headless-svc
spec:
  clusterIP: None
  selector:
    app: web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```

Nachdem wir die Ressource auf unserem Cluster erstellt haben, können wir die Details mit dem Befehl kubectl get svc überprüfen:

```
$ kubectl get svc -o go-template='{{ .spec.clusterIPs }}' headless-svc
[None]
```

Dabei verwenden wir die Option -o go-template, um speziell das Feld clusterIP des Services auszulesen. Wie zu erwarten war, weist die Kubernetes control plane einer Headless-Service-Ressource keine IP-Adresse zu.

Eigenschaften eines Headless Service

Bei der Namensauflösung eines typischen Service liefert der DNS-Server eine einzelne IP-Adresse zurück – die sogenannte Cluster-IP, die von der Control Plane zugewiesen wurde. Anders verhält es sich bei einem Headless Service: Hier gibt eine DNS-Abfrage eine Liste von IP-Adressen zurück, die zu den einzelnen, unterstützenden Pods gehören.

Diese Funktionsweise ermöglicht Anwendungsfälle, die mit einem normalen Service nicht realisierbar wären. Beispielsweise kann ein Monitoring-Dienst mithilfe eines Headless Service gezielt alle IP-Adressen der zugehörigen Pods ermitteln und so individuelle Health-Check-Anfragen verschicken. Bei einem regulären Service wäre nicht sichergestellt, welcher Pod die Anfrage entgegennimmt.

Headless Service in Aktion

In diesem Abschnitt richten wir die notwendigen Ressourcen ein, um die Funktionsweise eines Headless-Services zu demonstrieren.

Zunächst erstellen wir einen Headless-Service:

```
$ kubectl apply -f - <<EOF
apiVersion: v1
kind: Service
metadata:
  name: headless-svc-stateful
spec:
  clusterIP: None
  selector:
    app: web-stateful
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
EOF
```

Als Nächstes erstellen wir ein StatefulSet-Objekt, das drei Pods im Kubernetes-Cluster bereitstellt:

```
$ kubectl apply -f - <<EOF
apiVersion: apps/v1
```

```
kind: StatefulSet
metadata:
  name: app-stateful
  labels:
    app: server-stateful
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web-stateful
  serviceName: headless-svc-stateful
  template:
    metadata:
      labels:
        app: web-stateful
    spec:
      containers:
      - name: nginx
        image: nginx:alpine
        ports:
        - containerPort: 80
EOF
```

Wichtig ist hierbei, dass das StatefulSet einen Wert für das Feld `serviceName` benötigt. Dieser `serviceName` muss auf den Headless Service verweisen, der das StatefulSet im Netzwerk verfügbar macht. In unserem Fall handelt es sich um die Ressource mit dem Namen `headless-svc-stateful`.

Nachdem das Manifest angewendet wurde, wird die Kubernetes Control Plane drei Pods erzeugen:

```
$ kubectl get pods -l app=web-stateful
NAME             READY   STATUS    RESTARTS   AGE
app-stateful-0   1/1     Running   0          32m
app-stateful-1   1/1     Running   0          31m
app-stateful-2   1/1     Running   0          31m
```

Erstellen eines Ephemeral Containers

Ein Ephemeral Container ist eine praktische Möglichkeit, einem bestehenden Pod einen zusätzlichen Container mit einem anderen Image hinzuzufügen. Häufig wird hierfür ein Image mit erweiterten Werkzeugen verwendet, um Probleme innerhalb des Pods zu debuggen. In unserem Beispiel werden wir einen Ephemeral Container mit DNS-Tools auf einem bestehenden Pod erstellen, um verschiedene Funktionen unseres Headless Services zu testen.

Dazu verwenden wir den Befehl kubectl debug:

```
$ kubectl debug -it app-stateful-0 --image=slongstreet/bind-utils:latest --
bash
Defaulting debug container name to debugger-2crmp.
```

```
$
```

Der oben gezeigte Befehl startet einen Ephemeral Container im Pod `app-stateful-0` unter Verwendung des Images slongstreet/bind-utils. Dieses Image enthält unter anderem das Werkzeug nslookup, mit dem wir die Auflösung von Service-Domainnamen überprüfen können. Anschließend öffnen wir eine Bash-Sitzung innerhalb des Containers.

Auflösung der IP-Adresse des Headless-Services

Innerhalb des flüchtigen Containers wollen wir die IP-Adresse des Headless-Services `headless-svc-stateful` mittels des Befehls `nslookup` auflösen:

```
$ nslookup headless-svc-stateful
Server:         10.96.0.10
Address:        10.96.0.10#53

Name:   headless-svc-stateful.default.svc.cluster.local
Address: 10.244.0.11
Name:   headless-svc-stateful.default.svc.cluster.local
Address: 10.244.0.12
Name:   headless-svc-stateful.default.svc.cluster.local
Address: 10.244.0.10
```

Die Abfrage liefert eine Liste von drei IP-Adressen für diesen Domainnamen. Jede dieser Adressen gehört zu einem Pod, der durch den Service angesprochen wird.

Ein kleiner Haken an der Antwort ist, dass wir anhand der IP-Adressen nicht direkt erkennen können, zu welchen Pods sie jeweils gehören. Glücklicherweise ist es mit einem Headless-Service einfach, diese Zuordnung herauszufinden.

Auflösung der IP-Adresse eines bestimmten Pods

Möchten wir die IP-Adresse eines bestimmten Pods ermitteln, der dem Headless-Service zugeordnet ist, fügen wir den Namen des Pods als Subdomain zum Domainnamen des Headless-Services hinzu.

Konkret können wir so die IP-Adresse des Pods `app-stateful-1` mit folgendem Domainnamen auflösen `app-stateful-1.headless-svc-stateful`:

```
$ nslookup app-stateful-1.headless-svc-stateful
Server:         10.96.0.10
Address:        10.96.0.10#53

Name:   app-stateful-1.headless-svc-stateful.default.svc.cluster.local
Address: 10.244.0.11
```

Anstatt wie zuvor alle IP-Adressen der zugehörigen Pods zurückzugeben, liefert der Befehl nun lediglich die IP-Adresse des spezifischen Pods `app-stateful-1`.

Fazit

In diesem Tutorial haben wir uns kurz mit dem Service-Objekt im Kubernetes-Umfeld beschäftigt und dabei erfahren, dass es sich bei einem Headless-Service im Grunde um einen Service ohne Cluster-IP handelt. Aufgrund des Fehlens einer Cluster-IP verhalten sich Headless-Services anders als reguläre Services.

Abschließend haben wir ein Beispiel durchgespielt, das zeigt, wie Headless-Services besonders in StatefulSet-Pods nützlich sind, um gezielt die IP-Adresse eines bestimmten Pods zu ermitteln.