

What is a headless service, what does it do/accomplish, and what are some legitimate use cases for it?

Asked 2 years, 11 months ago Active 2 months ago Viewed 32k times



111



kubernetes

24



Share Improve this question

Follow

edited Jun 6 at 10:28



Gupta

5,480

4

32

48

asked Oct 8 '18 at 18:04



John Lexus

2,514

3

10

22

4 Answers

Active

Oldest

Votes



223



Well, I think you need some theory. There are many explanations (including the official docs) across the whole internet, but I think Marco Luksa did it the best:

Each connection to the service is forwarded to one randomly selected backing pod. But what if the client needs to connect to all of those pods? What if the backing pods themselves need to each connect to all the other backing pods. Connecting through the service clearly isn't the way to do this. What is?

For a client to connect to all pods, it needs to figure out the the IP of each individual pod. One option is to have the client call the Kubernetes API server and get the list of pods and their IP addresses through an API call, but because you should always strive to keep your apps Kubernetes-agnostic, using the API server isn't ideal

Luckily, Kubernetes allows clients to discover pod IPs through DNS lookups. Usually, when you perform a DNS lookup for a service, the DNS server returns a single IP — the service's cluster IP. But if you tell Kubernetes you don't need a cluster IP for your service (you do this by setting the clusterIP field to None in the service specification), the DNS server will return the pod IPs instead of the single service IP. Instead of returning a single DNS A record, the DNS server will return multiple A records for the service, each pointing to the IP of an individual pod backing the service at that moment. Clients can therefore do a simple DNS A record lookup and get the IPs of all the pods that are part of the service. The client

Setting the clusterIP field in a service spec to None makes the service headless, as Kubernetes won't assign it a cluster IP through which clients could connect to the pods backing it.

"Kubernetes in Action" by Marco Luksa

Share Improve this answer Follow

answered Oct 9 '18 at 4:39



[Konstantin Vustin](#)

4,116 2 10 26

34 I was happy to help you out, but i'd like to advise you to buy this book. The author has unquestionably done a fine job and should be encouraged. – [Konstantin Vustin](#) Oct 10 '18 at 3:58



18



Simply put, a **Headless service** is the same as **default ClusterIP service**, but *lacks load balancing or proxying*. Allowing you to connect to a Pod directly.

Share Improve this answer Follow

answered Feb 18 '20 at 13:27



[Raff](#)

319 2 6

6 Headless services still provide load balancing across pods but through the DNS round-robin mechanism instead of through the service proxy. – [Gupta](#) Apr 19 at 10:18



10



Let me break this question into each sub-parts the way we do in agile.

What exactly is a headless service.



It is used for discovering individual pods(especially IPs) which allows another service to interact directly with the Pods instead of a proxy. With NodePort, LoadBalancer, ExternalName, and ClusterIP clients usually connect to the pods through a Service ([Kubernetes Services simply visually explained](#)) rather than connecting directly.

What does it accomplish?

The requirement is not to make single IP like in the case of other service types. We need all the pod's IP sitting behind the service.

What are some legitimate use cases for it?

- Create Stateful service.

- Deployment of Relational databases
- and many more

Some Practical in Action

Deployment config

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app
  labels:
    app: server
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
      - name: nginx
        image: nginx:alpine
        ports:
        - containerPort: 80
```

Regular Service

```
apiVersion: v1
kind: Service
metadata:
  name: regular-svc
spec:
  selector:
    app: web
  ports:
  - protocol: TCP
    port: 80
    targetPort: 8080
```

Headless Service

```
apiVersion: v1
kind: Service
metadata:
  name: headless-svc
spec:
  clusterIP: None # <= Don't forget!!
  selector:
    app: web
  ports:
  - protocol: TCP
    port: 80
    targetPort: 8080
```

Create all resources and run tmp pod.

```
k run tmp01 --image=tutum/dnsutils -- sleep infinity
```

```
k exec tmp01 -it -- /bin/sh
```

```
#=> nslookup regular-svc
```

```
Server:      10.96.0.10
```

```
Address:     10.96.0.10#53
```

```
Name:       regular-svc.moon.svc.cluster.local
```

```
Address:    10.109.150.46
```

```
#=> nslookup headless-svc
```

```
Server:      10.96.0.10
```

```
Address:     10.96.0.10#53
```

```
Name:       headless-svc.moon.svc.cluster.local
```

```
Address:    172.17.0.31
```

```
Name:       headless-svc.moon.svc.cluster.local
```

```
Address:    172.17.0.30
```

```
Name:       headless-svc.moon.svc.cluster.local
```

```
Address:    172.17.0.32
```

The DNS server returns three different IPs for the `headless-svc.moon.svc.cluster.local` FQDN .

Note 1: with a headless service, clients can connect to its pods by connecting to the service's DNS name, as they can with regular services. But with headless services, because DNS returns the pods' IPs, clients connect directly to the pods, instead of through the service proxy.

Note 2: Headless services still provide load balancing across pods but through the DNS round-robin mechanism instead of through the service proxy.

Share Improve this answer

edited Jun 16 at 5:44

answered Apr 19 at 10:17

Follow



Gupta

5,480

4 32 48



5

I think the most common use case is mainly attributable to StatefulSets, that currently required a Headless Service. See [why-statefulsets-cant-a-stateless-pod-use-persistent-volumes](#) when you might use one...



Share Improve this answer Follow

answered Sep 3 '19 at 10:18



ama

1,254

1 14 29

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up

