

Kubernetes DNS

在Kubernetes中, DNS服务是非必须的, 所以无论是Kube-dns还是CoreDNS通常是以插件的形式安装, 但涉及到域名访问和服务发现时, DNS服务又是必须的. 用以解析Kubernetes集群内的pod和Service域名.

Service服务发现的两种方式:

- 环境变量: Pod创建的时候, 服务的ip和port会以环境变量的形式注入到pod里
- DNS: Service创建成功后, 会在dns服务器里导入一些记录, 想要访问某个服务, 通过dns服务器解析出对应的ip和port, 从而实现服务访问

环境变量

```
# 创建nginx deploy
[root@k8s-master manifests]# vi nginx_deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.18.0
          ports:
            - containerPort: 80

# 创建nginx service
[root@k8s-master manifests]# vi nginx_svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  labels:
    name: nginx-service
spec:
  ports:
    - port: 8180
      targetPort: 80
  selector:
    app: nginx
```

查看pod/svc/deploy创建情况

```
[root@k8s-master manifests]# kubectl get po,svc,deploy
```

NAME	READY	STATUS	RESTARTS	AGE
pod/nginx-deploy-67dfd6c8f9-jhvst	1/1	Running	0	4m2s
pod/nginx-deploy-67dfd6c8f9-zd5zl	1/1	Running	0	4m2s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/nginx-service	ClusterIP	192.168.33.47	<none>	8180/TCP	2m22s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/nginx-deploy	2/2	2	2	4m2s

创建查看svc环境变量的pod

```
[root@k8s-master manifests]# vi look_svc_env.yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: look-svc-env
```

```
spec:
```

```
  containers:
```

```
    - name: look-svc-env
```

```
      image: busybox
```

```
      command: ["/bin/sh", "-c", "env"]
```

查看日志

```
[root@k8s-master manifests]# kubectl logs look-svc-env
```

```
KUBERNETES_SERVICE_PORT=443
```

```
KUBERNETES_PORT=tcp://192.168.0.1:443
```

```
HOSTNAME=look-svc-env
```

```
...
```

```
NGINX_SERVICE_SERVICE_HOST=192.168.33.47
```

```
NGINX_SERVICE_PORT_8180_TCP_ADDR=192.168.33.47
```

```
KUBERNETES_PORT_443_TCP_PROTO=tcp
```

```
NGINX_SERVICE_PORT_8180_TCP_PORT=8180
```

```
NGINX_SERVICE_PORT_8180_TCP_PROTO=tcp
```

```
NGINX_SERVICE_SERVICE_PORT=8180
```

```
NGINX_SERVICE_PORT=tcp://192.168.33.47:8180
```

```
PWD=/
```

```
NGINX_SERVICE_PORT_8180_TCP=tcp://192.168.33.47:8180
```

nginx svc停掉后重启look-svc-env,就会发现环境变量已消失

```
[root@k8s-master manifests]# kubectl logs look-svc-env
```

```
KUBERNETES_SERVICE_PORT=443
```

```
KUBERNETES_PORT=tcp://192.168.0.1:443
```

```
HOSTNAME=look-svc-env
```

```
SHLVL=1
```

```
HOME=/root
```

```
KUBERNETES_PORT_443_TCP_ADDR=192.168.0.1
```

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

```
KUBERNETES_PORT_443_TCP_PORT=443
```

```
KUBERNETES_PORT_443_TCP_PROTO=tcp
```

```
KUBERNETES_SERVICE_PORT_HTTPS=443
```

```
KUBERNETES_PORT_443_TCP=tcp://192.168.0.1:443
```

```
KUBERNETES_SERVICE_HOST=192.168.0.1
```

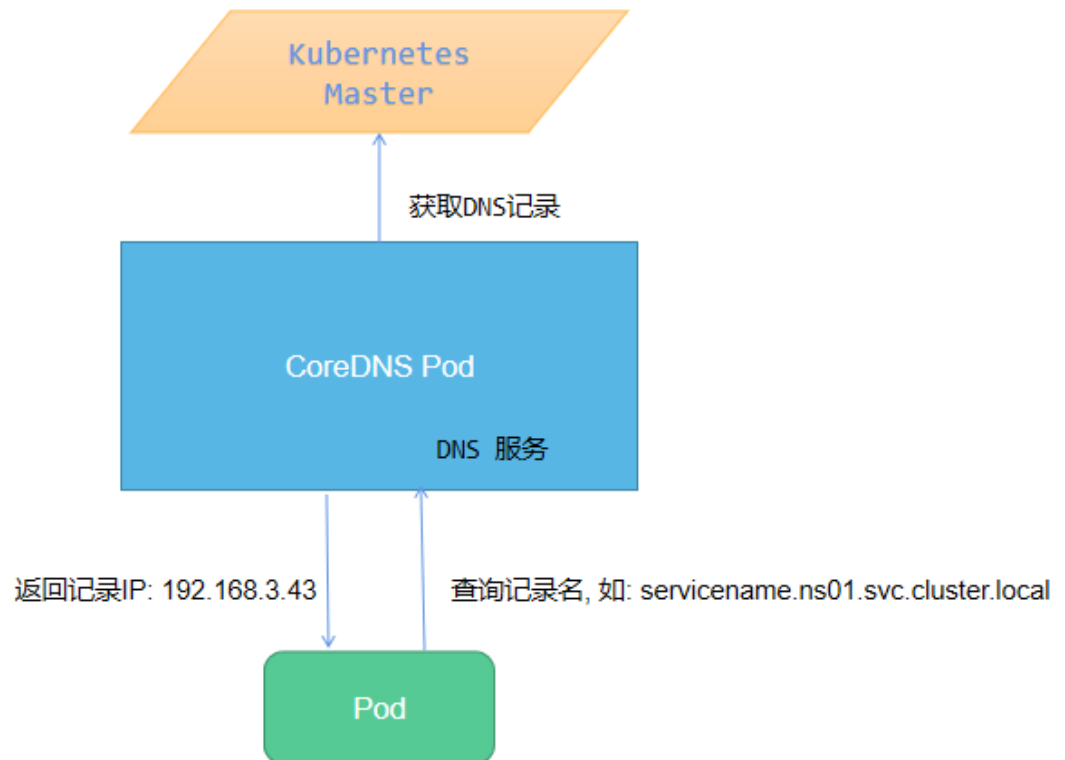
```
PWD=/
```

可以看到之前创建的nginx service已经写到了环境变量里, 从这里的

NGINX_SERVICE_SERVICE_HOST=192.168.33.47和**NGINX_SERVICE_SERVICE_PORT=8180**两个环境变量就能发现nginx服务, 但这里有个问题, 获取环境变量前提必须是先有对应的Service创建成功并写入环境变量, 如果Service没有启动, 这些环境变量便无法获取, 稳定性无法保证.

DNS

Kubernetes 从 v1.11 开始可以使用 CoreDNS 来提供命名服务, 并从 v1.13 开始成为默认 DNS 服务。CoreDNS 的特点是效率更高, 资源占用率更小, 推荐使用 CoreDNS 替代 kube-dns 为集群提供 DNS 服务。CoreDNS 基本架构如下:



如果需要定制DNS服务,可参考下面官方给的方案: [自定义DNS服务,定制DNS:dns-custom-nameservers](#)

```
# 查看namespace=kube-system的pod
[root@k8s-master manifests]# kubectl get pods -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
...
coredns-6d56c8448f-m92h8            1/1     Running   16          27d
coredns-6d56c8448f-wh66t            1/1     Running   16          27d
...
# 已安装CoreDNS
[root@k8s-master manifests]# docker images
REPOSITORY                                TAG
IMAGE ID                                CREATED                                SIZE
...
registry.aliyuncs.com/google_containers/coredns    1.7.0
bfe3a36ebd25    5 months ago                                45.2MB
...
### 域名格式
```

- 普通的 Service：会生成`servicename.namespace.svc.cluster.local`的域名，会解析到 Service 对应的 ClusterIP 上，在 Pod 之间的调用可以简写成 `servicename.namespace`，如果处于同一个命名空间下面，甚至可以只写成 `servicename` 即可访问
- Headless Service：无头服务，就是把 `clusterIP` 设置为 `None` 的，会被解析为指定 Pod 的 IP 列表，同样还可以通过`podname.servicename.namespace.svc.cluster.local`访问到具体的某一个 Pod

创建带nslookup命令工具的busybox Pod

```
[root@k8s-master manifests]# vi busybox-dns.yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: busybox
```

```
  namespace: default
```

```
spec:
```

```
  containers:
```

```
    - name: busybox
```

```
      image: busybox:1.28
```

```
      command:
```

```
        - sleep
```

```
        - "3600"
```

```
      imagePullPolicy: IfNotPresent
```

```
      restartPolicy: Always
```

```
[root@k8s-master manifests]# kubectl apply -f busybox-dns.yaml
```

```
pod/busybox created
```

```
[root@k8s-master manifests]# kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
busybox	1/1	Running	0	39s

查看默认的API Server DNS信息

```
[root@k8s-master manifests]# kubectl exec -it busybox -- nslookup
```

```
kubernetes.default
```

```
Server:      192.168.0.10
```

```
Address 1: 192.168.0.10 kube-dns.kube-system.svc.cluster.local
```

```
Name:        kubernetes.default
```

```
Address 1: 192.168.0.1 kubernetes.default.svc.cluster.local
```

Pod 的 DNS 配置

Pod 的 DNS 策略

DNS 策略可以逐个 Pod 来设定。目前 Kubernetes 支持以下特定 Pod 的 DNS 策略。这些策略可以在 Pod 规约中的 `dnsPolicy` 字段设置：

- "Default": Pod 从运行所在的节点继承名称解析配置
- "ClusterFirst": 与配置的集群域后缀不匹配的任何 DNS 查询（例如 "www.kubernetes.io"）都将转发到从节点继承的上游名称服务器。集群管理员可能配置了额外的存根域和上游 DNS 服务器
- "ClusterFirstWithHostNet"：对于以 `hostNetwork` 方式运行的 Pod，应显式设置其 DNS 策略 "ClusterFirstWithHostNet"

- "None": 此设置允许 Pod 忽略 Kubernetes 环境中的 DNS 设置。Pod 会使用其 dnsConfig 字段 所提供的 DNS 设置

说明：“Default”不是默认的 DNS 策略。如果未明确指定 dnsPolicy，则使用 "ClusterFirst"。

Pod 的 DNS 配置可让用户对 Pod 的 DNS 设置进行更多控制。

dnsConfig 字段是可选的，它可以与任何 dnsPolicy 设置一起使用。但是，当 Pod 的 dnsPolicy 设置为 "None" 时，必须指定 dnsConfig 字段。

用户可以在 dnsConfig 字段中指定以下属性：

- nameservers：将用作于 Pod 的 DNS 服务器的 IP 地址列表。最多可以指定 3 个 IP 地址。当 Pod 的 dnsPolicy 设置为 "None" 时，列表必须至少包含一个 IP 地址，否则此属性是可选的
- searches：用于在 Pod 中查找主机名的 DNS 搜索域的列表。此属性是可选的, Kubernetes 最多允许 6 个搜索域
- options：可选的对象列表，其中每个对象可能具有 name 属性（必需）和 value 属性（可选）

```
# 创建自定义DNS的Pod
[root@k8s-master manifests]# vi nginx_pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.18.0
    ports:
    - containerPort: 80
  dnsPolicy: "None"
  dnsConfig:
    nameservers:
    - 8.8.8.8
    searches:
    - default.svc.cluster-domain.example
    - cluster-domain.example
    options:
    - name: pod_num
      value: "1"
[root@k8s-master manifests]# kubectl apply -f nginx_pod.yaml
pod/nginx created
# 查看DNS信息
[root@k8s-master manifests]# kubectl exec -it nginx -- cat /etc/resolv.conf
nameserver 8.8.8.8
search default.svc.cluster-domain.example cluster-domain.example
options pod_num:1
```

验证DNS配置

查看之前讲解环境变量时创建的Service

```
[root@k8s-master manifests]# kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
nginx-service	ClusterIP	192.168.202.76	<none>	8180/TCP	112m

查看默认kube-dns

```
[root@k8s-master manifests]# kubectl get svc kube-dns -n kube-system
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kube-dns	ClusterIP	192.168.0.10	<none>	53/UDP,53/TCP,9153/TCP	28d

查看该nginx service的DNS配置

```
[root@k8s-master manifests]# kubectl exec -it nginx-deploy-67dfd6c8f9-jhvt -- cat /etc/resolv.conf
```

这个默认的IP是kube-dns创建时分配的固定IP地址

```
nameserver 192.168.0.10
```

```
search default.svc.cluster.local svc.cluster.local cluster.local
```

```
options ndots:5
```

进入busybox工具容器内部验证一下之前配置的nginx service

```
[root@k8s-master manifests]# kubectl run --rm -i --tty test-dns --image=busybox /bin/sh
```

```
/ # cat /etc/resolv.conf
```

```
nameserver 192.168.0.10
```

```
search default.svc.cluster.local svc.cluster.local cluster.local
```

```
options ndots:5
```

访问

```
/ # wget -q -O- nginx-service.default.svc.cluster.local:8180
```

```
<!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>
```