

Ingress 部署指北

2021/12/16

Ingress 部署指北

[Ingress 简介](#)

[环境准备](#)

[1. K8s 集群](#)

[2. ingress-nginx 安装](#)

[安装指南](#)

[使用 YAML 配置文件安装](#)

[Ingress 控制器预检](#)

[本地测试](#)

[1. 创建 Web 服务](#)

[2. 创建 ingress](#)

[3. 浏览器访问](#)

Ingress 简介

Ingress 是对集群中服务的外部访问进行管理的 API 对象，典型的访问方式是 HTTP。Ingress 可以提供负载均衡、SSL 终结和基于名称的虚拟托管。

为了让 Ingress 资源工作，集群必须有一个正在运行的 Ingress 控制器。仅创建 Ingress 资源本身没有任何效果。

与 kube-controller-manager 管理的 控制器 不同，Ingress 控制器不是随集群自动启动的。可以根据需要选择最适合你的集群的 ingress 控制器。

环境准备

1. K8s 集群

生产环境下选择 K8s (> v1.18) 集群，开发环境下可以选择 MicroK8s。本地实验环境选择 MicroK8s 开发集群：

```
1 dragon@test-pc:~$ microk8s kubectl version
2 Client Version: version.Info{Major:"1", Minor:"22+", GitVersion:"v1.22.4-3+adc4115d990346", GitCommit:"adc4115d990346b87714cc4f033d225711bf744d", GitTreeState:"clean", BuildDate:"2021-11-17T22:07:39Z", GoVersion:"go1.16.10", Compiler:"gc", Platform:"linux/amd64"}
3 Server Version: version.Info{Major:"1", Minor:"22+", GitVersion:"v1.22.4-3+adc4115d990346", GitCommit:"adc4115d990346b87714cc4f033d225711bf744d", GitTreeState:"clean", BuildDate:"2021-11-17T22:03:17Z", GoVersion:"go1.16.10", Compiler:"gc", Platform:"linux/amd64"}
```

2. [ingress-nginx](#) 安装

注意：microk8s 环境下，只需要执行下面的命令启用 ingress 即可，无需手动安装 ingress-nginx。

```
1 | $ microk8s enable ingress
```

安装指南

根据需要可以从 [Ingress 控制器](#) 列表中选择合适的 Ingress 控制器进行安装，例如 ingress-nginx。NGINX ingress controller 的安装方法有多种：

- 使用 Helm，使用项目 chart 仓库。
- 使用 kubectl apply 通过 YAML 清单进行。
- 使用诸如 minikube 或 MicroK8s 特定的插件 (specific addons)。

在大多数 Kubernetes 集群上，ingress 控制器无需任何额外配置即可工作。如果您想尽快上手，可以查看 [快速入门](#) 说明。

但是，在许多环境中，您可以通过启用额外功能来提高性能或获得更好的日志。建议您查看 [特定于环境](#) 的说明，了解 特定环境 或 云提供商 关于优化 Ingress 控制器的详细信息。

使用 YAML 配置文件安装

在 K8s (> v1.18) 集群环境下，以下命令会在 ingress-nginx 命名空间 (自动创建) 中安装 controller：

```
1 | $ kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.1.0/deploy/static/provider/cloud/deploy.yaml
```

Ingress 控制器预检

```
1 | # k8s v1.18
2 | anxin@node38:~$ kubectl get pods --namespace=ingress-nginx
3 | NAME                                READY  STATUS   RESTARTS  AGE      286d
4 | ingress-nginx-controller-d6dc77594-ckqgx  1/1    Running  1218
5 |
6 | # microk8s v1.22+
7 | dragon@test-pc:~$ kubectl get pods --namespace=ingress
8 | NAME                                READY  STATUS   RESTARTS  AGE
9 | nginx-ingress-microk8s-controller-xbklb  1/1    Running  0          33m
```

本地测试

1. 创建 Web 服务

创建一个简单的 Web 服务器和相关的服务：

```

1 # 1. 创建 web 服务器
2 dragon@test-pc:~$ kubectl create deployment demo --image=httpd --port=80
3 deployment.apps/demo created
4 # 2. Expose a resource as a new Kubernetes service.
5 dragon@test-pc:~$ kubectl expose deployment demo
6 service/demo exposed

```

查看结果:

```

1 dragon@test-pc:~$ kubectl get deploy
2 NAME      READY   UP-TO-DATE   AVAILABLE   AGE
3 demo      1/1     1             1           31h
4
5 dragon@test-pc:~$ kubectl get svc
6 NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
7 demo      ClusterIP   10.152.183.106 <none>        80/TCP     30h

```

2. 创建 ingress

创建一个 ingress 资源，映射到 localhost。

- K8s 集群环境

```

1 $ kubectl create ingress demo-localhost --class=nginx --
  rule=demo.localdev.me/*=demo:80

```

- microk8s 环境

```

1 # 可以省略 --class=public (默认的 IngressClass)
2 dragon@test-pc:~$ kubectl create ingress demo-localhost --class=public --
  rule=demo.localdev.me/*=demo:80
3 ingress.networking.k8s.io/demo-localhost created

```

查看 microk8s 环境下的结果:

```

1 # 查看创建的 ingress
2 dragon@test-pc:~$ kubectl get ingress
3 NAME          CLASS    HOSTS          ADDRESS          PORTS    AGE
4 demo-localhost public   demo.localdev.me          80           26h
5
6 # 查看 ingress 实例 demo-localhost 配置
7 dragon@test-pc:~$ kubectl get ingress demo-localhost -o yaml
8 apiVersion: networking.k8s.io/v1
9 kind: Ingress
10 metadata:
11   creationTimestamp: "2021-12-15T07:31:10Z"
12   generation: 1
13   name: demo-localhost
14   namespace: default
15   resourceVersion: "2966486"
16   selfLink: /apis/networking.k8s.io/v1/namespaces/default/ingresses/demo-
  localhost
17   uid: 1baee2ac-ca5b-448e-9749-3428208c03ef
18 spec:
19   ingressClassName: public # --class=public

```

```
20 rules:
21   - host: demo.localdev.me # --rule=demo.localdev.me/*=demo:80
22     http:
23       paths:
24         - backend:
25             service:
26               name: demo
27               port:
28                 number: 80
29               path: /
30               pathType: Prefix
31 status:
32   loadBalancer: {}
```

3. 浏览器访问

访问 <http://demo.localdev.me/>, 应该会看到一个 HTML 页面: "It works!"。

