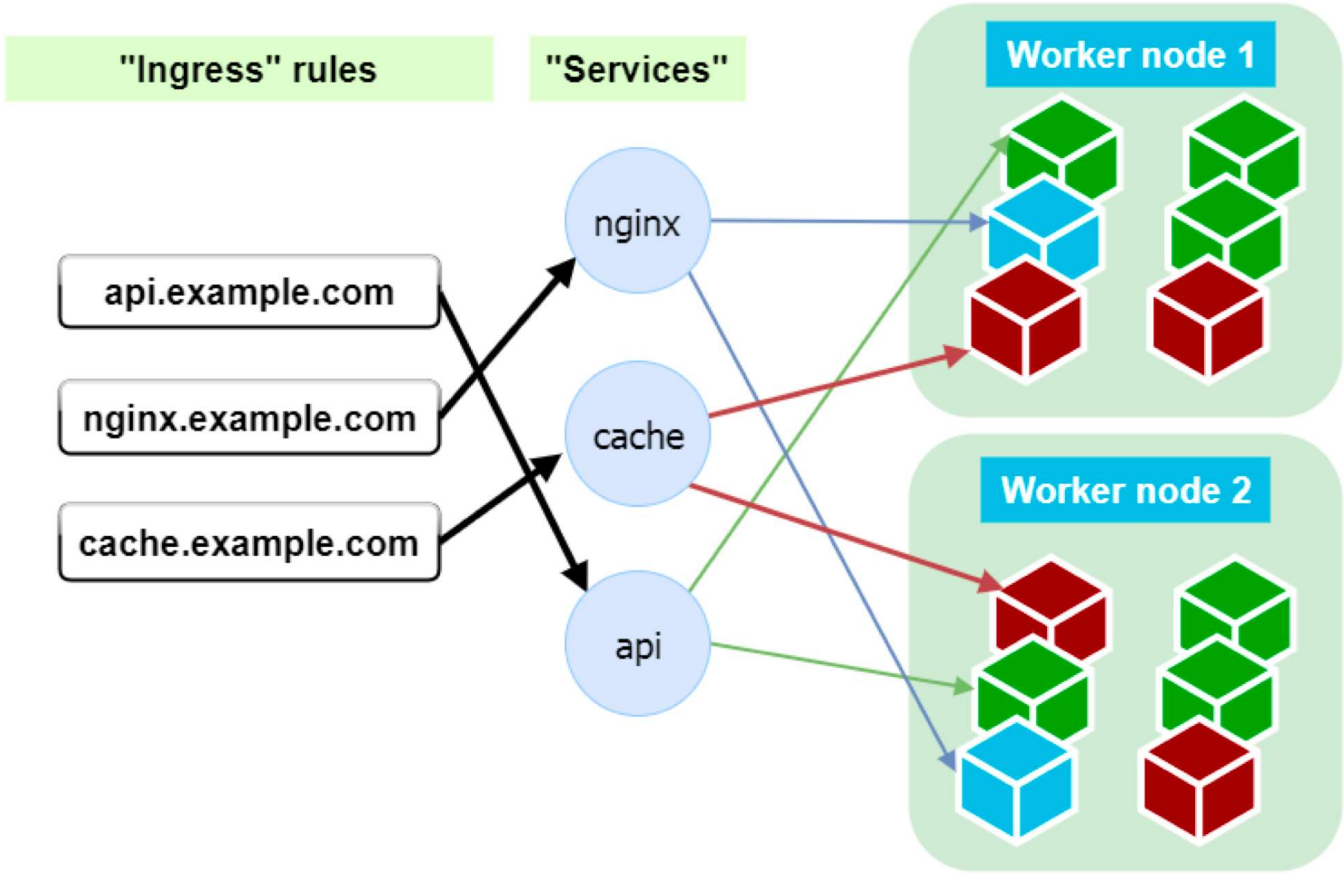
Ingress service types in Kubernetes

Kubernetes Advocate
Nov 19, 2020 · 4 min read

Kubernetes has a built-in configuration object for **HTTP load balancing** called **Ingress**.



It defines rules for external connectivity to the pods represented by one or more Kubernetes services.

Ingress can provide **SSL termination** and **name-based virtual hosting**.

The traffic routing is controlled by rules defined on the Ingress resource.

```
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: app-ingress
spec:
  rules:
    -
      http:
        paths:
          -
            backend:
              serviceName: app-service
              servicePort: 80
            path: /
```

Ingress resource solely supports rules for steering communications protocol (HTTP)traffic. The Ingress specification has all the data required to put together a load balancer or proxy server. most significantly, it contains a listing of rules matched against all incoming requests.

Ingress rules

Each HTTP rule contains an optional host, a list of paths each of which has an associated backend defined with a serviceName and service port.

If the traffic path is not matched to any rules, then traffic sends to the default backend.

Default Backend

The default backend is often a configuration possibility of the Ingress controller and isn't laid out in your Ingress resources. If none of the hosts or methods match the **HTTP** request within the Ingress objects, the traffic is routed to your default backend.

Types of Ingress

Single Service Ingress

It doesn't have any rules and it sends traffic to a single service. You can use this to create a default backend with no rules.

```
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: frontend-ingress
spec:
```

```
  backend:
    serviceName: frontend-service
    servicePort: 80
```

Simple fanout

A fanout configuration routes traffic to more than one service, based on the HTTP URI being requested.

```
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple-fanout-example
spec:
  rules:
    -
      host: shopping.example.com
      http:
        paths:
          -
            backend:
              serviceName: clothes-service
              servicePort: 8080
            path: /clothes
          -
            backend:
              serviceName: House-service
              servicePort: 8081
            path: /kitchen
```

Name-based virtual hosting

Name-based virtual hosts support routing HTTP traffic to multiple hostnames.

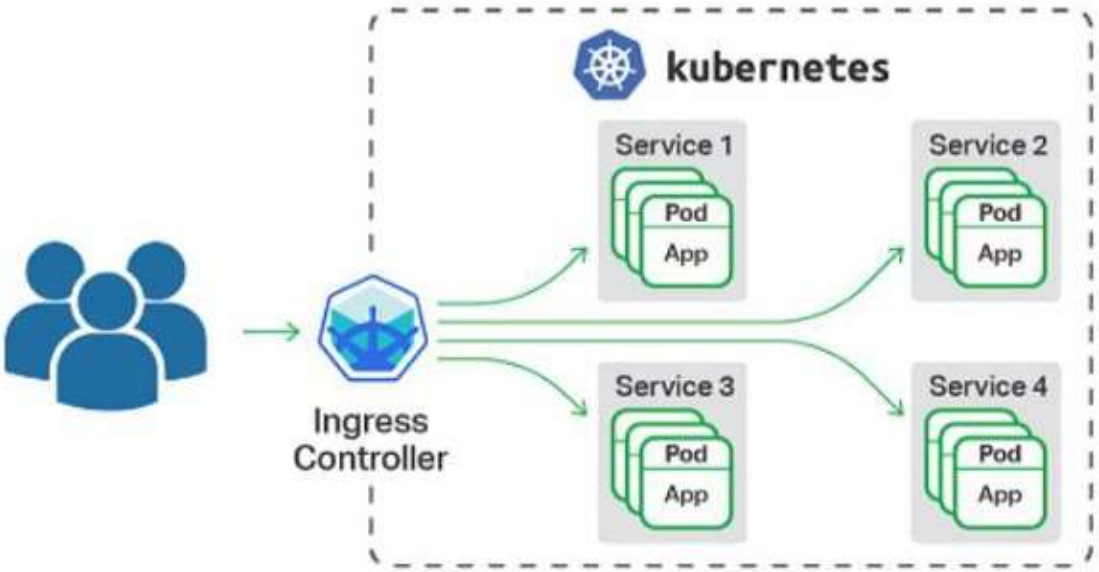
```
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: name-virtual-host-ingress
spec:
  rules:
    -
      host: shopping.example.com
      http:
        paths:
          -
            backend:
              serviceName: clothes-service
              servicePort: 8080
```

```
        path: /clothes
      -
        backend:
          serviceName: House-service
          servicePort: 8081
        path: /kitchen
-
  host: music.example.com
  http:
    paths:
      -
        backend:
          serviceName: Hindi-service
          servicePort: 9090
        path: /hindhi
      -
        backend:
          serviceName: English-service
          servicePort: 9091
        path: /english
```

Ingress controller

In order to work the ingress resource, the **Kubernetes** cluster must have an ingress controller running.

It runs as part of the **Kube-controller-manager** and is typically started automatically with a cluster.



There are so many ingress controller implementations and choices that best fit your cluster.

Additional controllers include:

NGINX, Inc. provides overall maintenance and customer support for the NGINX controller for K8s. Contour is an **associate** Envoy **primarily based** ingress controller provided and supported by Heptio.

Traefik is a fully-featured ingress controller (Let's Encrypt, secrets, http2, WebSocket), and it also comes with commercial support by Continuous.

Ambassador API Gateway is an Envoy based ingress controller with the community or commercial support from Datawire.

Citrix provides an Ingress Controller for its hardware (MPX), virtualized (VPX), and free containerized (CPX) ADC for bare-metal and cloud deployments.

F5 Networks always provides support and maintenance for the F5 BIG-IP Controller for K8s clusters. Gloo is an open-source ingress controller supported Envoy that offers API entranceway practicality with enterprise support from solo.io. HAProxy based mostly ingress controller DevOps/ ha proxy-ingress that is mentioned on the weblog post-HAProxy Ingress Controller for Kubernetes. HAProxy Technologies offers support and maintenance for HAProxy Enterprise and therefore the ingress controller DevOps/ha proxy-ingress.

Istio based ingress controller Control Ingress Traffic.

You can deploy multiple ingress controllers within a cluster.

When you create an ingress resource, you should annotate each ingress with the appropriate ingress.

Ex: Ingress with Nginx ingress controller:

```
---
metadata:
  annotations:
    kubernetes.io/ingress.class: nginx
  name: my-ingress
```

If you do not define ingress.class, your cloud provider will use a default ingress provider.

For more Reference

Service

An abstract way to expose an application running on a set of Pods as a network service. With Kubernetes you don't need...

kubernetes.io

Ingress

FEATURE STATE: Kubernetes v1.19 [stable] An API object that manages external access to the services in a cluster...

kubernetes.io

Ingress Controllers

In order for the Ingress resource to work, the cluster must have an ingress controller running. Unlike other types of...

kubernetes.io