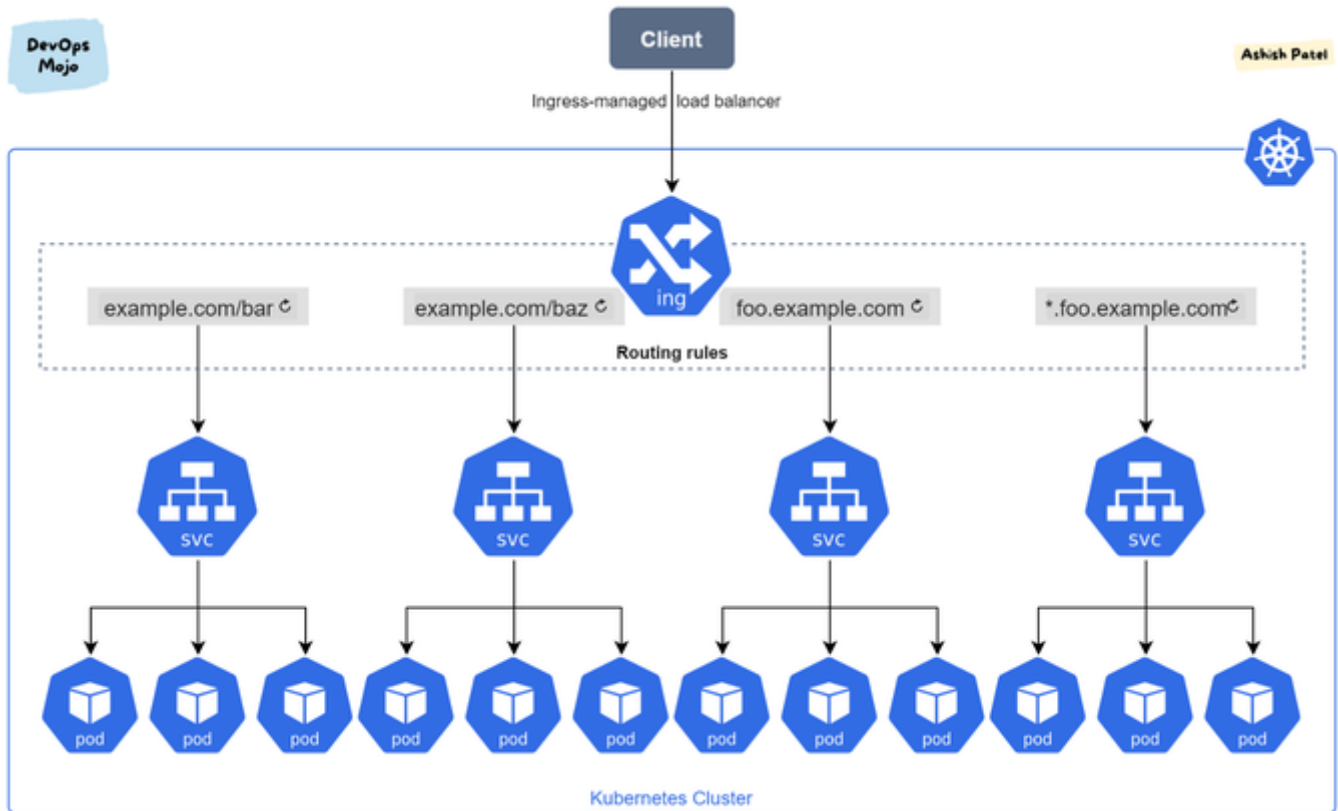


# Kubernetes — Ingress Overview

What is K8s Ingress? — Introduction to Kubernetes Ingress.



Kubernetes — Ingress

TL;DR

[Ingress](#) exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.

What is Ingress in Kubernetes?

In Kubernetes, an Ingress is an object that **allows access to Kubernetes services from outside the Kubernetes cluster**. You can configure access by creating a collection of rules that define which inbound connections reach which services.

An Ingress can be configured to give Services externally-reachable URLs, load balance traffic, terminate SSL/TLS, and offer name-based virtual hosting. Ingress lets you configure an HTTP load balancer for applications running on Kubernetes, represented by one or more Kubernetes internal Services.

An [Ingress controller](#) is responsible for fulfilling the Ingress, usually with a load balancer, though it may also configure your edge router or additional frontends to help handle the traffic.

The Ingress [spec](#) has all the information needed to configure a load balancer or proxy server. It contains a list of rules matched against all incoming requests. Ingress provides routing rules to manage external users' access to the services in a Kubernetes cluster, typically via HTTPS /HTTP. With Ingress, you can easily set up rules for routing traffic without creating a bunch of Load Balancers or exposing each service on the node. This makes it the best option to use in production environments.

An Ingress does not expose arbitrary ports or protocols. Exposing services other than HTTP and HTTPS to the internet typically uses a service of type *NodePort* or *LoadBalancer*.

Ingress controllers

- Ingress controller is an application that runs in a cluster and configures an HTTP load balancer according to Ingress resources. The load balancer can be a software load balancer running in the cluster or hardware or cloud load balancer running externally. Different load balancers require different Ingress controller implementations.

- In order to Ingress resource work, the cluster must have an ingress controller running.
- You can deploy any number of ingress controllers within a cluster.
- There are many different Ingress controllers, and there's support for cloud-native load balancers (from GCP, AWS, and Azure).  
e.g. **Nginx**, **Ambassador**, **EnRoute**, **HAProxy**, **AWS ALB**, **AKS Application Gateway**

| You must have an [Ingress controller](#) to satisfy an Ingress. Only creating an Ingress resource has no effect.

## Key Points

- An ingress is an API object that manages external access to the services in a cluster, typically HTTP. It means you can use Ingress to make your Service accessible from the outside.
- Ingress is not a Service type, but it acts as the entry point for the cluster.
- Ingress offers simplistic gateway-type solutions.
- Ingress lets you consolidate your routing rules into a single resource and expose multiple services under the same IP address, using the same load balancers.
- Ingress also enables the configuration of resilience (time-outs, rate-limiting), content-based routing, authentication, and much more.

## Use Cases

- Externally reachable URLs for applications deployed in Kubernetes clusters.
- Load balancing rules and traffic, as well as TLS/SSL termination for each hostname, such as `foo.example.com`.
- Content-based routing:
  1. *Host-based routing*. For example, routing requests with the host header `foo.example.com` to one group of services and the host header `bar.example.com` to another group.
  2. *Path-based routing*. For example, routing requests with the URI that starts with `/serviceA` to service A and requests with the URI that starts with `/serviceB` to service B.

## Example

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: example.com
    http:
      paths:
      - path: /foo
        pathType: Prefix
        backend:
          service:
            name: foo-service
            port:
              number: 3000
      - path: /bar
        pathType: Prefix
        backend:
          service:
            name: bar-service
            port:
              number: 6000
  - host: foo.example.com
    http:
      paths:
      - pathType: Prefix
        path: "/foo"
        backend:
          service:
            name: foo-service-2
            port:
              number: 80
  - host: "*.foo.example.com"
    http:
      paths:
      - pathType: Prefix
        path: "/foo"
        backend:
          service:
            name: foo-service-3
            port:
              number: 8080

```

## Summary

**A Kubernetes Ingress is a robust way to expose Kubernetes services outside the cluster.** It lets you consolidate your routing rules to a single resource, and gives powerful options for configuring these rules, by allowing an API Gateway style of traffic routing.