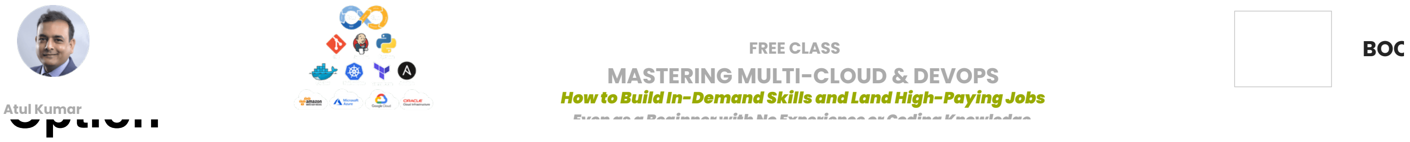
Cloud Option



June 2, 2023 by Sahid (https://k21academy.com/author/sahidk21academy-com/)

17145 views

As Kubernetes continues to dominate the container orchestration landscape, managing and exposing services to the outside world has become a crucial aspect of application deployment. Kubernetes Ingress Controllers have emerged as a vital component for achieving this, simplifying the process of routing external traffic to services running within a Kubernetes cluster.

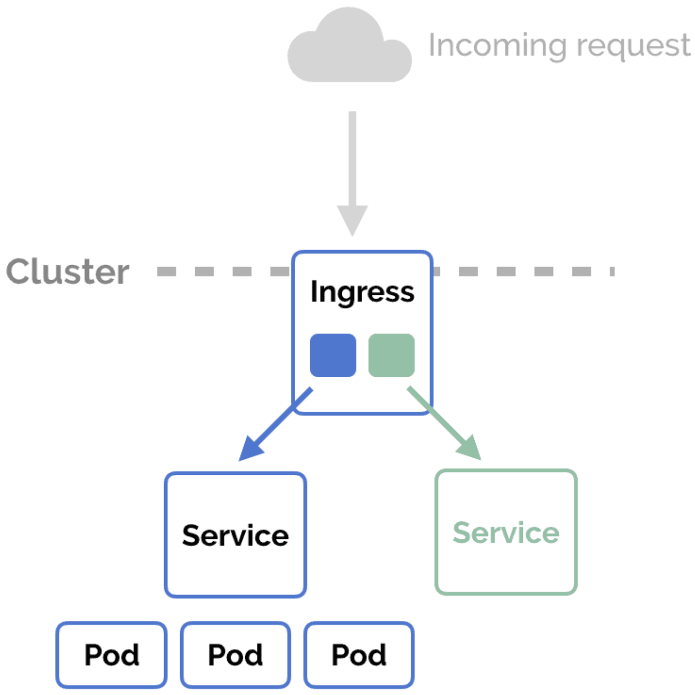
Kubernetes has 3 types of services viz. ClusterIP, NodePort, and LoadBalancer. The Ingress exposes HTTP/S routes from outside the cluster to services inside the cluster. So, one thing must be clear to you now that the ingress isn't a type of service that Kubernetes offers. Whereas the Kubernetes Ingress Controller is pledged to manage the traffic and the ingress mostly with a LoadBalancer. In this blog post, we will explore the concept of Kubernetes Ingress Controllers, their importance, and how they facilitate seamless service exposure.

Kubernetes Ingress Controller | Load Balancer Servi...

In this blog, we are going to talk about:

What is Ingress Controller?

Ingress Controller is an intelligent Load Balancer. Ingress is a high-level abstraction responsible for allowing simple host or URL based HTTP routing. It is always implemented using a third-party proxy. These implementations are nothing but **Ingress Controller**. It is a **Layer-7** load balancer.



Ingress Controller

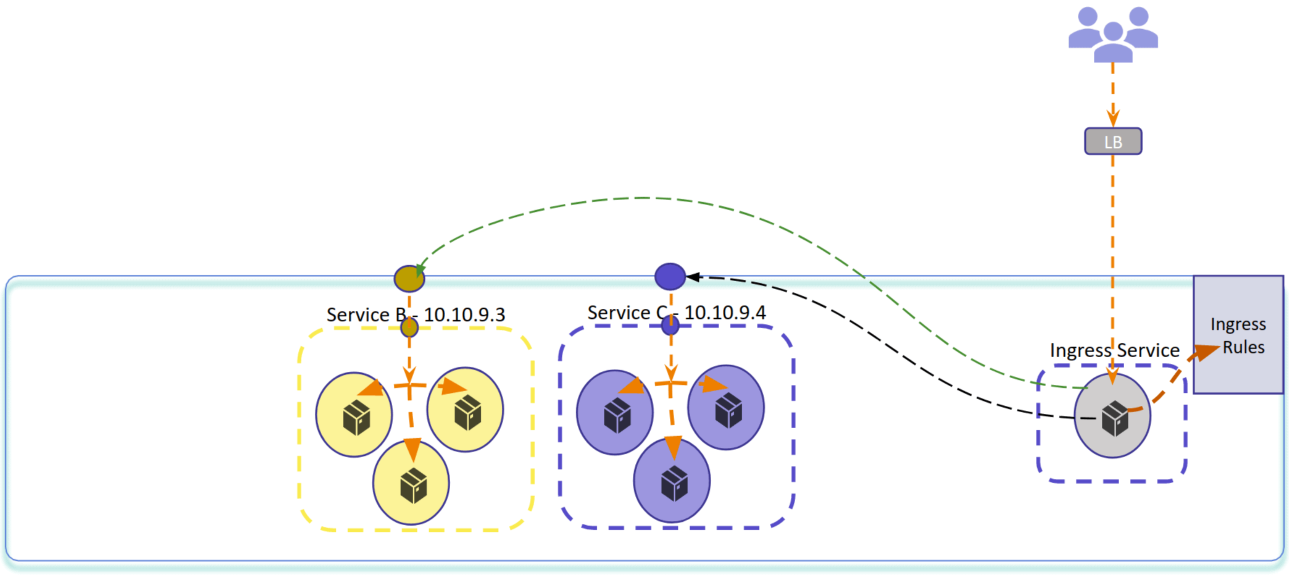
Kubernetes ingress controller is accountable for reading the Ingress resource information and processing that data. Ingress is tightly integrated into K8s, meaning that your existing workflows around kubectl will likely extend nicely to managing ingress. One thing to keep in mind is that an ingress controller customarily doesn't eliminate the need for an external load balancer instead the ingress controller simply adds an additional layer of routing & control for the load balancer.

learn with us join our **Kubernetes training** (https://k21academy.com/kubernetes/) and Understand **Kubernetes** in an easy way

Kubernetes Ingress Controller with a LoadBalancer

Whatever is your ingress strategy, you presumably will need to start with an external load balancer. This will route traffic to a K8s service on the cluster that will perform service-specific routing. In this setup, your load balancer provides a stable endpoint which is nothing but an IP address for external traffic to access.

Both ingress controllers and K8s services require an external load balancer. So, this concludes that NodePort is not designed to be directly used for production.



Ingress Controller with a LoadBalancer

Also Read: Our blog post on **Kubernetes ReadinessProbe** (https://k21academy.com/docker-kubernetes/kubernetes-readiness-and-livenessprobe/). Click here

Cloud-Based Ingress Controller

These days most of the cloud providers have Kubernetes services, below are the major cloud providers' Ingress Controller links:

- AKS Application Gateway Ingress Controller (https://github.com/Azure/application-gateway-kubernetes-ingress)
- AWS ALB Ingress Controller (https://github.com/kubernetes-sigs/aws-alb-ingress-controller)
- GCP GLBC/GCE-Ingress Controller (https://github.com/kubernetes/ingress-gce/blob/master/README.md)

The major advantage of using a cloud-based Ingress Controller is native integration with other cloud services. For instance, GCE Ingress Controller supports Cloud IAP for Google Kubernetes Engine (https://cloud.google.com/iap/docs/enabling-kubernetes-howto) to easily turn on Identity-Aware Proxy to protect internal K8s applications. As for ALB Ingress Controller, it creates an Application Load Balancer by default and integrates well with Route 53, Cognito, and AWS WAF. If you are using Azure Pipelines to manage your DevOps on Azure, AKS Application Gateway Ingress Controller is the best for Azure CI/CD workflow.

Also Check: What is a Deployment in Kubernetes (https://k21academy.com/docker-kubernetes/kubernetes-deployment/)? Click here

Opensource Ingress Controller

Opensource is always there to the rescue no matter what. Kubernetes has listed quite a few of them on their website, here are some of them:

- Voyager (https://voyagermesh.com/): HAProxy based Ingress Controller from AppsCode (https://appscode.com/)
- F5 (https://clouddocs.f5.com/containers/latest/userguide/kubernetes/): Supports F5's BIG-IP Container Ingress Services
- HAProxy (https://haproxy-ingress.github.io/): Community-driven HAProxy Ingress Controller as well as enterprise offering from HAProxy Tech (https://github.com/haproxytech/kubernetes-ingress)
- Istio (https://istio.io/): Ingress Gateway for Istio-enabled clusters
- Kong (https://github.com/Kong/kubernetes-ingress-controller): nginx-based API gateway with community or enterprise options from KongHQ (https://konghq.com/solutions/kubernetes-ingress/)
- NGINX (https://github.com/nginxinc/kubernetes-ingress): official Ingress for NGINX and NGINX Plus
- Skipper (https://opensource.zalando.com/skipper/kubernetes/ingress-controller/): HTTP router and reverse proxy from Zalando (https://jobs.zalando.com/en/tech/)

Most used Ingress Controller (credits: CNCF)

Also Read: **Docker vs Kubernetes** (https://k21academy.com/docker-kubernetes/kubernetes-vs-docker/), to know the major differences between them

NGINX-Based Ingress Controllers

This is the most popular and only open-source Ingress Controller maintained by the K8s team, built on top of NGINX reverse proxy. It is a popular option for simple HTTP/S routing and SSL termination use case. Hence of the popularity, there is comprehensive documentation and tutorials available for common ingress tasks and related tools.

If you need a straightforward simple reverse proxy, ingress-nginx is a safe and decisive option. On the other hand, if you are looking for high performance and also additional features supported by NGINX, consider using the Ingress Controller from NGINX instead.

HAProxy-Based Ingress Controllers

HAProxy is also one of the popular TCP/HTTP reverse proxy solution that existed before Kubernetes. So, if configuring the load balancing algorithm is your primary deciding factor, HAProxy Ingress is a great option with a proven record of high performance. As an Ingress Controller, HAProxy Ingress offers dynamic configuration update via API to address reliance on static configuration files with HAProxy.

Also Check: Our blog post on **Kubernetes labels and Annotations** (https://k21academy.com/docker-kubernetes/labels-and-annotations-in-kubernetes/). Click here

F5 Container Ingress

The F5 BIG-IP Controller (k8s-bigip-ctlr) is a cloud-native connector that can use either Kubernetes or OpenShift as a BIG-IP orchestration platform. The BIG-IP Controller watches the Kubernetes API (https://kubernetes.io/docs/api/) for specially formatted resources and updates the BIG-IP system configuration accordingly.

Envoy Proxy

Envoy Proxy (https://www.envoyproxy.io/) is a modern, high-performance service proxy. Envoy is similar to software load balancers such as NGINX and HAProxy. It was originally written and deployed at Lyft, Envoy now has a vigorous contributor base and is an official CNCF project. It provides the foundation for a service mesh.

GCP Ingress Controller

Google cloud platform is also one of the top-notch ingress controller providers. When an Ingress object is created on the GCP, the GKE Ingress controller (https://github.com/kubernetes/ingress-gce) creates a Google Cloud HTTP(S) Load Balancer (https://cloud.google.com/load-balancing/docs/https) and configures it according to the information in the Ingress and its associated Services. To use Ingress, the HTTP load balancing add-on must be enabled.

Also Check: Our blog post on **Kubernetes Certified Administrator Training** (https://k21academy.com/docker-kubernetes/certified-kubernetes-administrator-cka-exam/).

Traefik

The Traefik Kubernetes Ingress provider is a Kubernetes Ingress controller; that is to say, it manages access to cluster services by supporting the Ingress specification.

Istio Ingress

Istio makes heavy use of Envoy proxies to mediate all traffic within the service mesh. Istio provides the best integration with existing Istio fabric and services with traffic routing, observability, security, and deployment models. However, Istio is not lightweight and has a fairly large learning curve, so if Envoy proxy is the only functionality you are looking for, use the following options instead.

Ambassador

Ambassador is an API Gateway and L7 load balancer with Kubernetes Ingress support. It is fully-featured with various protocol supports, security, high availability, and even Knativ serverless integration. Although it's based on Envoy, it connects nicely with other service mesh solutions besides Istio (e.g. Consul, Linkerd). The benchmark results (https://www.getambassador.io/resources/envoyproxy-performance-on-k8s/) posted on their blog compares favourably to NGINX and HAProxy, although it has not been updated for several months.

Also Read: Our previous blog post on **Kubernetes dashboard** (https://k21academy.com/docker-kubernetes/kubernetes-dashboard/). Click here

Which is the best ingress controller for Kubernetes?

With so many options on the market, how do I choose which Ingress Controller is right for my use case? You must have realised by now that **ingress-nginx** is a safe and one of the most popular choices when you need a simple solution to get started. If you are using Istio as your service mesh, **Istio Ingress** is a natural fit; otherwise, consider an Envoy-based solution that works with Consul or Linkerd. Here are some considerations before choosing a solution:

1. **Protocol Support**: Is there a requirement for TCP/UDP or gRPC integration?
2. **Enterprise Support**: Do you need a commercial/enterprise support for a mission-critical system?
3. **Advanced Features:** Are you looking for a lightweight solution or are canary deployments or circuit breakers must-haves for your use case?
4. **API Gateway Features:** Do you need API Gateway functionalities or a pure Kubernetes Ingress?

Benefits of Using Ingress Controllers

Below are the benefits of using an Ingress Controllers:

1. Simplified Service Exposure: Ingress Controllers abstract away the complexity of managing network routing and load balancing, providing a straightforward way to expose services to the outside world. Developers can focus on writing application code rather than dealing with networking intricacies.
2. Traffic Management and Load Balancing: Ingress Controllers allow for advanced traffic routing and load balancing strategies. This enables efficient distribution of incoming requests across multiple instances of a service, ensuring high availability and scalability.
3. Enhanced Security: Ingress Controllers often provide SSL termination, enabling secure communication between clients and services. They also support authentication mechanisms and request filtering, protecting the exposed services from unauthorized access and potential security threats.
4. Scalability and Flexibility: Ingress Controllers seamlessly scale alongside the Kubernetes cluster, adapting to changes in the number of services or replicas. They provide a flexible and extensible framework for managing ingress rules, allowing for easy customization and integration with other tools and services.

Frequently Asked Questions

What is a Kubernetes Ingress Controller?

A Kubernetes Ingress Controller is a software component responsible for implementing the rules specified in the Ingress resource. It acts as a gateway between external traffic and services running within a Kubernetes cluster, enabling seamless routing and service exposure.

Why do I need an Ingress Controller?

An Ingress Controller simplifies the process of managing external traffic and service exposure in Kubernetes. It provides advanced routing capabilities, load balancing, SSL termination, and security features, making it easier to expose services to the outside world while ensuring scalability and reliability.

What are some popular Ingress Controllers?

Some widely used Ingress Controllers in the Kubernetes ecosystem include Nginx Ingress Controller, Traefik, and HAProxy Ingress Controller. These controllers offer various features and integrations, allowing you to choose the one that best fits your requirements.

What benefits do Ingress Controllers offer?

Ingress Controllers provide several benefits, including simplified service exposure, advanced traffic management, load balancing, enhanced security through SSL termination and authentication mechanisms, scalability, and flexibility to adapt to changes in the Kubernetes cluster.

Can I use multiple Ingress Controllers in a Kubernetes cluster?

While it is technically possible to use multiple Ingress Controllers in a Kubernetes cluster, it is generally recommended to choose one as the primary controller to avoid conflicts and ensure consistent routing rules and behavior.

Are Ingress Controllers compatible with cloud-based load balancers?

Yes, Ingress Controllers can work with cloud-based load balancers. In such cases, the Ingress Controller acts as an interface between Kubernetes and the cloud load balancer, providing additional routing and management capabilities specific to Kubernetes services.

Conclusion

Kubernetes Ingress Controllers play a crucial role in simplifying the process of exposing services and managing external traffic within a Kubernetes cluster. They provide a unified and flexible approach to service exposure, allowing for advanced traffic routing, load balancing, and security features. Whether you choose Nginx, Traefik, HAProxy, or any other Ingress Controller, leveraging these powerful tools will enable you to effortlessly expose your services and build scalable, reliable applications

Related Post

- **Kubernetes Networking: Container-to-container, Pod-to-Pod, Pod-to-Service** (https://k21academy.com/docker-kubernetes/kubernetes-networking/)
- **Kubernetes Architecture | An Introduction to Kubernetes Components** (https://k21academy.com/docker-kubernetes/kubernetes-architecture-components-overview-for-beginners/)
- **Kubernetes for Beginners** (https://k21academy.com/docker-kubernetes/docker-and-kubernetes-important-blogs-for-beginners/)
- **Visit our YouTube channel on "Docker & Kubernetes"** (https://k21academy.com/dockerkubernetes?sub_confirmation=1)
- **Certified Kubernetes Administrator (CKA) Certification Exam: Everything You Must Know (https://k21academy.com/docker-kubernetes/certified-kubernetes-administrator-cka-certification-exam-everything-you-must-know/)**
- **Certified Kubernetes Administrator (CKA) Certification: Step By Step Activity Guides/Hands-On Lab Exercise (https://k21academy.com/docker-kubernetes/certified-kubernetes-administrator-cka-certification-training-step-by-step-activity-guides-hands-on-lab-exercise/)**
- **Kubernetes vs Docker – Understand the Difference** (https://k21academy.com/docker-kubernetes/kubernetes-vs-docker/)
- **Install Docker on Windows, Ubuntu and Mac: A Complete Step-by-Step Guide** (https://k21academy.com/docker-kubernetes/docker-installation-overview/)
- **Docker Compose Overview & Steps to Install Docker Compose (https://k21academy.com/docker-kubernetes/docker-compose/)**
- **CKA Certification Exam (Certified Kubernetes Administrator)** (https://k21academy.com/docker-kubernetes/certified-kubernetes-administrator-cka-exam/)
- **Container (Docker) vs Virtual Machines (VM): What Is The Difference?** (https://k21academy.com/docker-kubernetes/docker-vs-virtual-machine/)

- **How To Setup A Three Node Kubernetes Cluster For CKA: Step By Step** (https://k21academy.com/docker-kubernetes/three-node-kubernetes-cluster/)

Join FREE Masterclass

Discover the **Power of Kubernetes, Docker & DevOps** – Join Our Free Masterclass (https://k21academy.com/free-masterclass-dockers-kubernetes-administrations-with-certifications/?utm_source=blogcontentupgrade&utm_medium=referral&utm_campaign=kubernetes02_may23). Unlock the secrets of Kubernetes, Docker, and DevOps in our exclusive, no-cost masterclass. Take the first step towards building highly sought-after skills and securing lucrative job opportunities. **Click on the below image** to Register Our **FREE Masterclass** (https://k21academy.com/free-masterclass-dockers-kubernetes-administrations-with-certifications/?utm_source=blogcontentupgrade&utm_medium=referral&utm_campaign=kubernetes02_march21) Now!

(https://k21academy.com/free-masterclass-dockers-kubernetes-administrations-with-certifications/?utm_source=blogcontentupgrade&utm_medium=referral&utm_campaign=kubernetes02_may23)

(/)

"Learn Cloud From Experts"

(https://www.facebook.com/k21academy/)

(https://twitter.com/k21academy)

(https://www.linkedin.com/company/k21academy)

(https://www.instagram.com/k21academy/)

(https://www.youtube.com/k21academy)

Site Map

Home (https://k21academy.com/)
Courses (/courses/)
Blog (https://k21academy.com/blog/)
Partner with Us (https://k21academy.com/affiliate/)
Careers (https://k21academy.com/careers/)
Contact Us (https://k21academy.com/contact-us/)
Terms and Conditions (https://k21academy.com/terms-and-conditions/)
Privacy Policy (https://k21academy.com/privacy-policy/)

Courses

Docker and Kubernetes Job Oriented Program (https://k21academy.com/docker-kubernetes-training/)
AWS Job Oriented Program (https://k21academy.com/awsjob-oriented-program/)
Azure Job Oriented Program (https://k21academy.com/azurejob-oriented-program/)
Azure Data Job Oriented Program (https://k21academy.com/azure-data-cloud-job-program/)
DevOps Job Oriented Program (https://k21academy.com/devops-job-program/)
Oracle Cloud Job Oriented Program (https://k21academy.com/oracle-cloud-job-program/)
Terraform Job Oriented (https://k21academy.com/infra-automation-certification-terraform-training/)

Get in touch with us

8 Magnolia Pl, Harrow HA2 6DS,
United Kingdom

Phone:
US: +1 530 264 8480 (tel:+15302648480)

Email:
contact@k21academy.com (mailto:contact@k21academy.com)