13 Key Considerations for Choosing the Right Ingress Controller

An ingress controller acts as a reverse proxy and load balancer to implement a Kubernetes Ingress, or set of API routing rules. Your choice in ingress controller greatly impacts the efficiency of your deployments and the effectiveness of your team.

There are 13 considerations to take into account when evaluating ingress controller features. Here they are.

1. Traffic profile
   - A reverse proxy's performance is context-dependent. Different ingress choices have varying resource requirements.

2. Dynamic configuration updates
   - Are you using dynamic, load-based algorithms? Some ingress controllers support these features, so you won't have to code them yourself.

3. Resilience
   - If you need to observe or trace traffic, you may need a service mesh. If you require a service mesh, one that doubles as an ingress controller will increase efficiency and reduce technical bloat.

4. API gateway
   - Are you integrating with an external, managed cloud-based load balancer? Make sure the ingress controller you select integrates well with your external load balancer for efficiency.

5. High availability
   - Do you need to integrate with existing metrics and log collection systems? Some ingress controllers don't support all specific monitoring and logging tooling.

6. Load balancing algorithms
   - What sort of algorithm-based routing do you need? Most ingress controllers support round-robin. If you want least connection, you'll need an ingress controller with advanced load balancing.

7. Advanced traffic shifting
   - Leverage policies to spread traffic among services evenly. Some ingress controllers can split traffic with sophisticated rules. If you need to perform canary testing, choose an ingress controller that supports traffic shifting.

8. Resources constraints
   - Ingress controllers can be resource-intensive. If you are cost-sensitive, choose a lightweight ingress controller. Some ingress controllers support scaling up and down, while others do not.

9. Advanced monitoring
   - Some ingress controllers support advanced monitoring, allowing you to observe or trace traffic, adjust policies, and analyze performance.

10. Support
    - Most vendors offer unlimited support. Make sure you have the right support level. Consider the cost implications of an enterprise support plan.

11. Ecosystem
    - Make sure the ingress controller you choose integrates well with Kubernetes and Kubernetes applications.

12. API gateway
    - Do you require an ingress controller, API gateway, or both? If you require business logic at the edge, choose an API gateway—or an ingress controller built for API management.

13. Open source
    - Open source ingress controllers are easy on the checkbook, but what if you need support in the middle of the night? If you're going open source, see if the vendor offers an enterprise support plan.

Want an ingress controller that checks all the boxes?

Explore TrafficKubes, an all-in-one ingress controller, API gateway, and service mesh.

REQUEST A DEMO
TRY IT FREE