

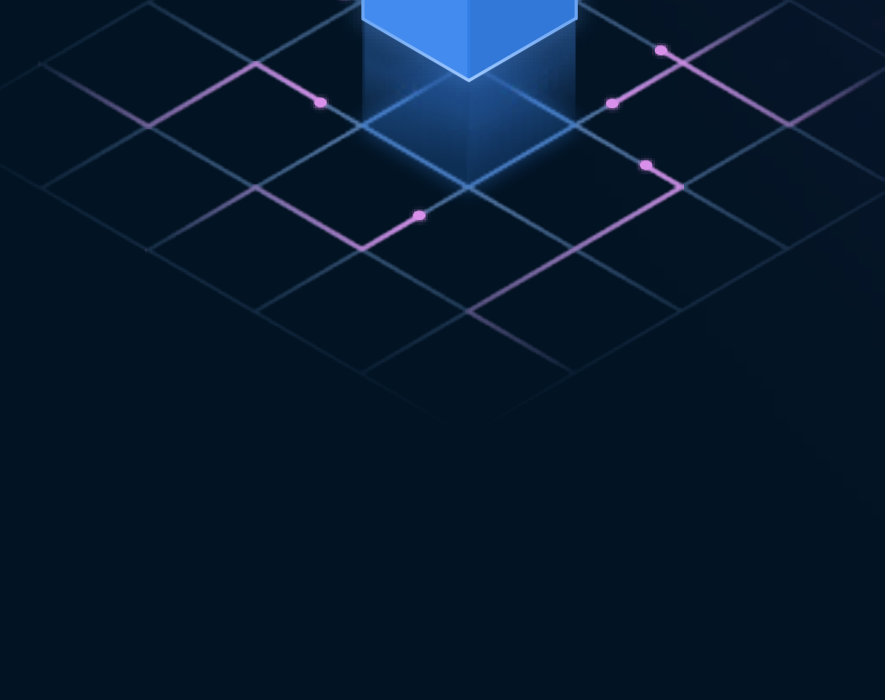
13 Key Considerations for Choosing the Right Ingress Controller

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 deployments and effectiveness of your team.

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

1. Traffic protocol

Are you routing HTTP(S), HTTP/2, websockets, TCP/UDP, or gRPC? Check which protocols an ingress controller supports.

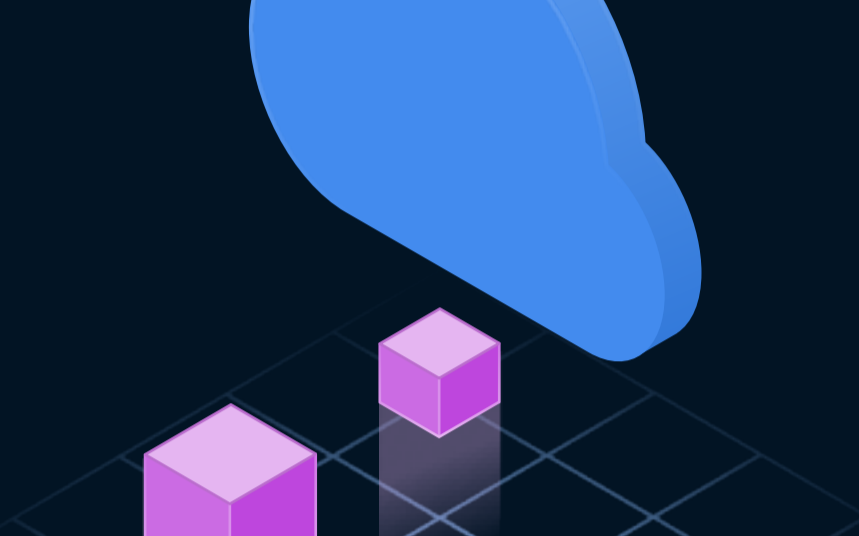


2. Dynamic configuration updates

Do you require dynamic, zero-downtime configuration changes? Some ingress controllers require downtime to update configuration, while others update dynamically without.

3. Resilience

Do you need rate limiting, retries, or circuit breakers at the edge? Some ingress controllers support these features, so you won't have to code them yourself.

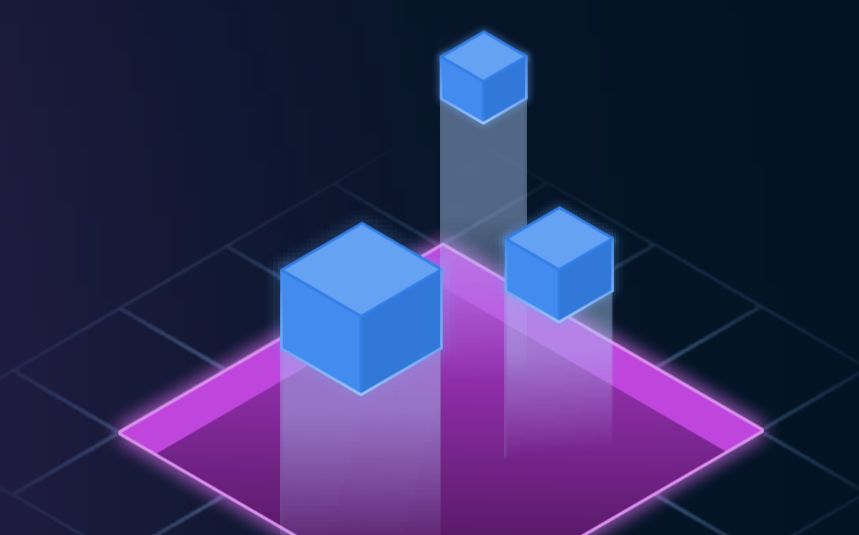
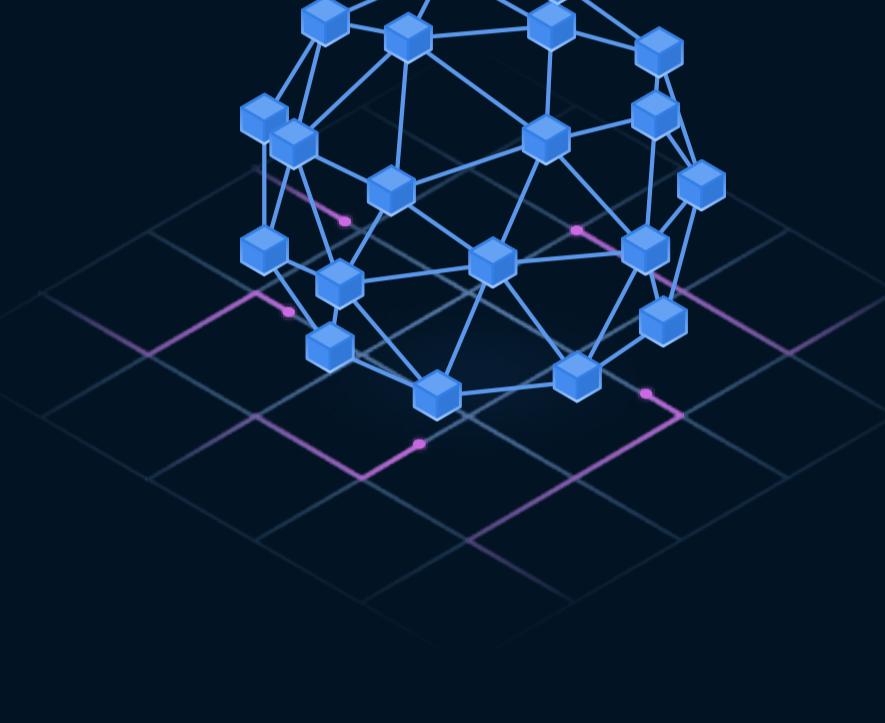


4. External load balancer integration

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. Service mesh

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

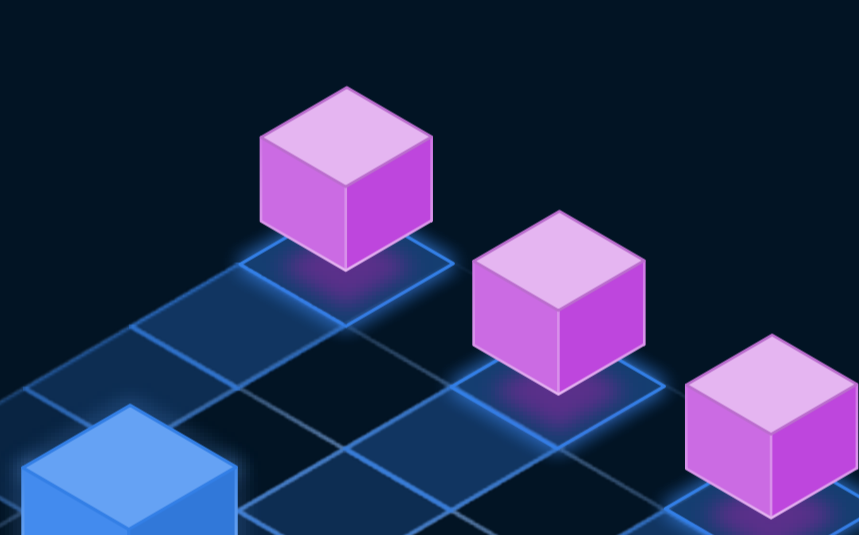
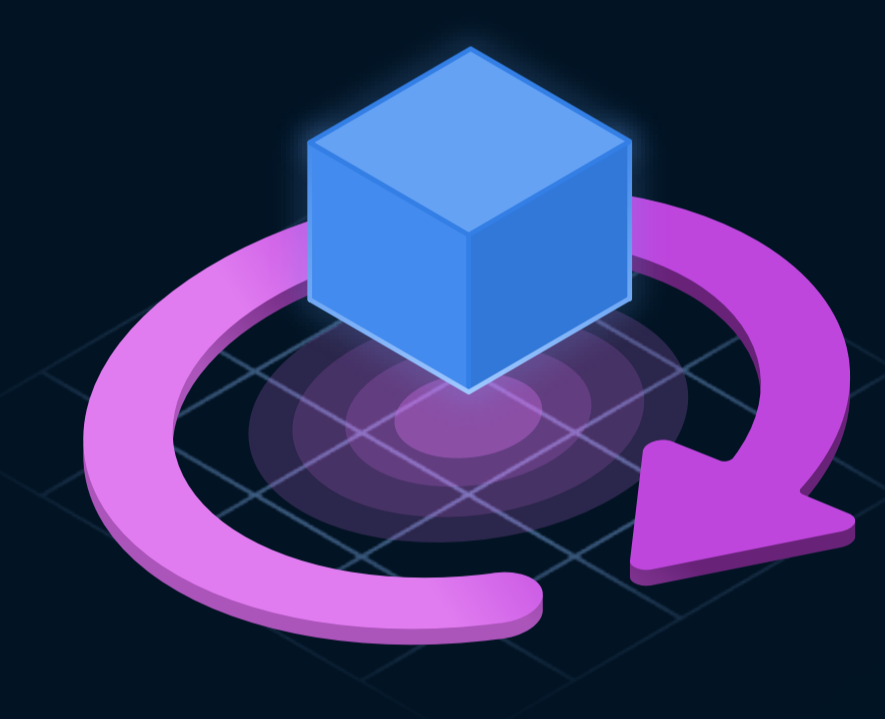


6. API gateway

Do you need 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.

7. High availability

Can you afford downtime when a server restarts for either planned or unplanned maintenance? If not, you need high availability, which not all ingress controllers support.



8. Load balancing algorithms

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

9. Advanced traffic shifting

Load balancing lets you spread the load of a service, but not all load balancers can split traffic with sophisticated rules. If you need to perform canary testing, choose an ingress controller that supports traffic shifting.

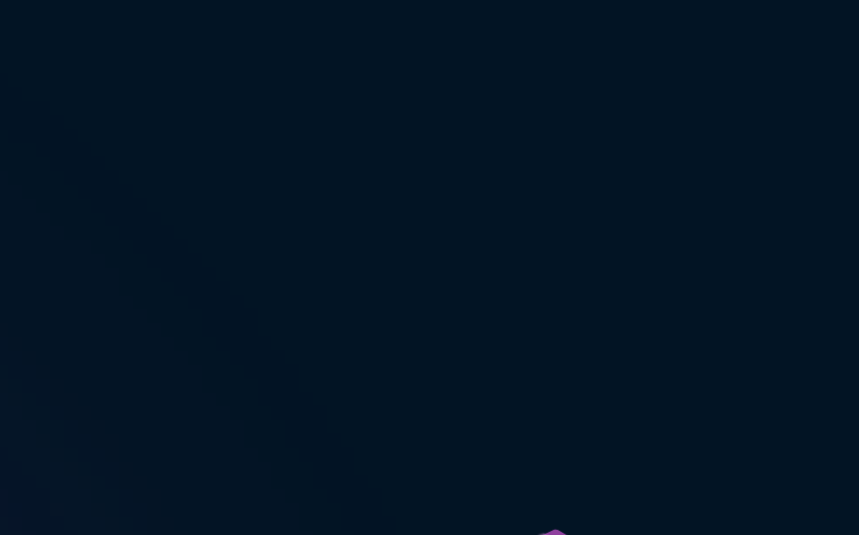
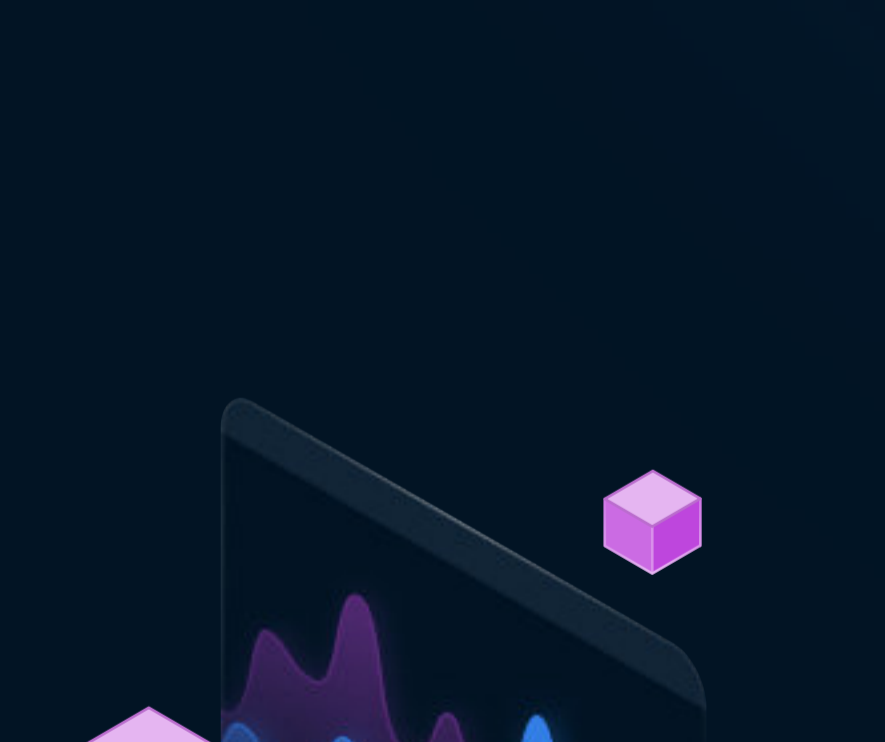


10. Resources constraints

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.

11. Monitoring

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

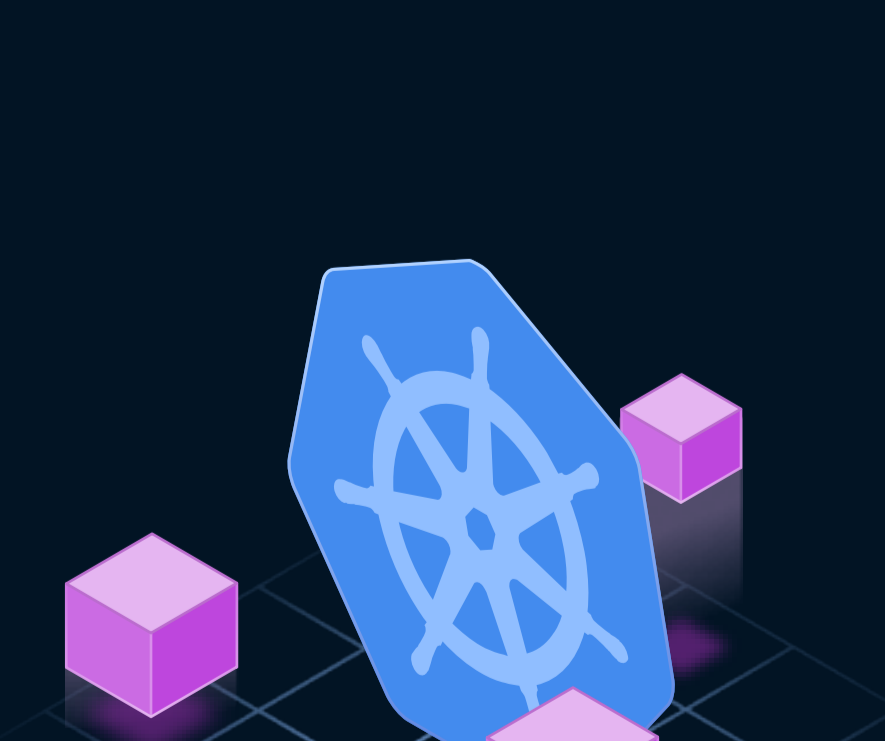


12. Support

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 OS, see if the vendor offers an enterprise support plan.

13. Ecosystem

Make sure the ingress controller you are considering has support in the Kubernetes partner ecosystem.



Want an ingress controller that checks all the boxes?

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

[REQUEST A DEMO](#)

[TRY IT FREE](#)