Traefik & Kubernetes

- Configuration Example
- Configuration Options
  - endpoint
  - ingressEndpoint.publishedService
  - reportNodeInternalIPs
- Routing Configuration
- Further

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

??? warning "Ingress Backend Resource not supported"

```
Referencing backend service endpoints using
[`spec.rules.http.paths.backend.resource`](https://kubernetes.io/docs/reference/kubernetes-
api/service-resources/ingress-v1/#IngressBackend) is not supported.
Use `spec.rules.http.paths.backend.service` instead.
```

??? warning "Ingress API frozen"

```
The Kubernetes Ingress API is [frozen](https://kubernetes.io/docs/concepts/services-
networking/ingress/) and will not receive further changes or updates.
You are encouraged to use the [Kubernetes Gateway provider](./kubernetes-gateway.md) instead.
```

Configuration Example

You can enable the `kubernetesIngress` provider as detailed below:

```
providers:
  kubernetesIngress: {}
```

```
[providers.kubernetesIngress]
```

```
--providers.kubernetesingress=true
```

```
## Values file
providers:
  kubernetesIngress:
    enabled: true
```

The provider then watches for incoming ingresses events, such as the example below, and derives the corresponding dynamic configuration from it, which in turn creates the resulting routers, services, handlers, etc.

Configuration Options

Field	Description	Default	Required
`providers.providersThrottleDuration`	Minimum amount of time to wait for, after a configuration reload, before taking into account any new configuration refresh event. If multiple events occur within this time, only the most recent one is taken into account, and all others are discarded. **This option cannot be set per provider, but the throttling algorithm applies to each of them independently.**	2s	No
`providers.kubernetesIngress.endpoint`	Server endpoint URL. More information here.	""	No
`providers.kubernetesIngress.token`	Bearer token used for the Kubernetes client configuration. Accepts either the token value directly or a path to a file containing the token.	""	No
`providers.kubernetesIngress.certAuthFilePath`	Path to the certificate authority file. Used for the Kubernetes client configuration.	""	No

Field	Description	Default	Required
`providers.kubernetesIngress.namespaces`	Array of namespaces to watch. If left empty, watch all namespaces.		No
`providers.kubernetesIngress.labelSelector`	Allow filtering on `Ingress` objects using label selectors. No effect on Kubernetes `Secrets`, `EndpointSlices` and `Services`. See label-selectors for details.	""	No
`providers.kubernetesIngress.ingressClass`	The `IngressClass` resource name or the `kubernetes.io/ingress.class` annotation value that identifies resource objects to be processed. If empty, resources missing the annotation, having an empty value, or the value `traefik` are processed.	""	No
`providers.kubernetesIngress.disableIngressClassLookup`	**Deprecated:** please use `disableClusterScopeResources` instead. Prevent to discover IngressClasses in the cluster. It alleviates the requirement of giving Traefik the rights to look IngressClasses up. Ignore Ingresses with IngressClass. Annotations are not affected by this option.	false	No
`providers.kubernetesIngress.ingressEndpoint.hostname`	Hostname used for Kubernetes Ingress endpoints.	""	No
`providers.kubernetesIngress.ingressEndpoint.ip`	This IP will get copied to the Ingress `status.loadbalancer.ip`, and currently only supports one IP value (IPv4 or IPv6).	""	No
`providers.kubernetesIngress.ingressEndpoint.publishedService`	The Kubernetes service to copy status from. More information here.	""	No

Field	Description	Default	Required
providers.kubernetesIngress.reportNodeInternalIPs	Report node internal IPs in Ingress status. Incompatible with ingressEndpoint and disableClusterScopeResources. More information here.	false	No
providers.kubernetesIngress.throttleDuration	Minimum amount of time to wait between two Kubernetes events before producing a new configuration. This prevents a Kubernetes cluster that updates many times per second from continuously changing your Traefik configuration. If empty, every event is caught.	0s	No
providers.kubernetesIngress.allowEmptyServices	Allows creating a route to reach a service that has no endpoint available. It allows Traefik to handle the requests and responses targeting this service (applying middleware or observability operations) before returning a 503 HTTP Status.	false	No
providers.kubernetesIngress.allowExternalNameServices	Allows the Ingress to reference ExternalName services.	false	No
providers.kubernetesIngress.crossProviderNamespaces	List of namespaces from which Ingresses or Services are allowed to use traefik.ingress.kubernetes.io/router.middlewares, traefik.ingress.kubernetes.io/router.tls.options, traefik.ingress.kubernetes.io/service.middlewares, or traefik.ingress.kubernetes.io/service.serverstransport annotations. When unset, all namespaces are allowed. When set to [], every cross-provider reference is rejected.	[]	No

Field	Description	Default	Required
`providers.kubernetesIngress.nativeLBByDefault`	Allow using the Kubernetes Service load balancing between the pods instead of the one provided by Traefik for every `Ingress` by default. It can be overridden in the `Service`	false	No
`providers.kubernetesIngress.disableClusterScopeResources`	Prevent from discovering cluster scope resources (`IngressClass` and `Nodes`). By doing so, it alleviates the requirement of giving Traefik the rights to look up for cluster resources. Furthermore, Traefik will not handle Ingresses with IngressClass references, therefore such Ingresses will be ignored (please note that annotations are not affected by this option). This will also prevent from using the `NodePortLB` options on services and is incompatible with `reportNodeInternalIPs`.	false	No
`providers.kubernetesIngress.strictPrefixMatching`	Make prefix matching strictly comply with the Kubernetes Ingress specification (path-element-wise matching instead of character-by-character string matching). For example, a PathPrefix of `/foo` will match `/foo` , `/foo/` , and `/foo/bar` but not `/foobar` .	false	No

`endpoint`

The Kubernetes server endpoint URL.

When deployed into Kubernetes, Traefik reads the environment variables `KUBERNETES_SERVICE_HOST` and `KUBERNETES_SERVICE_PORT` or `KUBECONFIG` to construct the endpoint.

The access token is looked up in `/var/run/secrets/kubernetes.io/serviceaccount/token` and the SSL CA certificate in `/var/run/secrets/kubernetes.io/serviceaccount/ca.crt` . Both are mounted automatically when deployed inside Kubernetes.

The endpoint may be specified to override the environment variable values inside a cluster.

When the environment variables are not found, Traefik tries to connect to the Kubernetes API server with an external-cluster client.

In this case, the endpoint is required. Specifically, it may be set to the URL used by `kubectl proxy` to connect to a Kubernetes cluster using the granted authentication and authorization of the associated kubeconfig.

```
providers:
  kubernetesIngress:
    endpoint: "http://localhost:8080"
    # ...
```

```
[providers.kubernetesIngress]
  endpoint = "http://localhost:8080"
  # ...
```

```
--providers.kubernetesingress.endpoint=http://localhost:8080
```

`ingressEndpoint.publishedService`

Format: `namespace/servicename`.

The Kubernetes service to copy status from, depending on the service type:

- **ClusterIP:** The ExternalIPs of the service will be propagated to the ingress status.
- **NodePort:** The ExternalIP addresses of the nodes in the cluster will be propagated to the ingress status.
- **LoadBalancer:** The IPs from the service's `loadBalancer.status` field (which contains the endpoints provided by the load balancer) will be propagated to the ingress status.
- **ExternalName:** The hostname from the service's `spec.externalName` field will be propagated to the ingress status.

When using third-party tools such as External-DNS, this option enables the copying of external service IPs to the ingress resources.

```
providers:
  kubernetesIngress:
    ingressEndpoint:
      publishedService: "namespace/foo-service"
    # ...
```

```
[providers.kubernetesIngress.ingressEndpoint]
  publishedService = "namespace/foo-service"
  # ...
```

```
--providers.kubernetesingress.ingressendpoint.publishedservice=namespace/foo-service
```

`reportNodeInternalIPs`

When set to `true`, Traefik reports the internal IPs of all nodes in the cluster into the `status.loadBalancer.ingress` field of each managed Ingress resource.

This is the equivalent of ingress-nginx's `--report-node-internal-ip-address` flag and is the recommended approach for bare-metal Kubernetes deployments where Traefik runs as a DaemonSet without a cloud LoadBalancer or MetalLB.

This option requires cluster-scope access to Node resources and is mutually exclusive with `ingressEndpoint` and `disableClusterScopeResources`.

```
providers:
  kubernetesIngress:
    reportNodeInternalIPs: true
    # ...
```

```
[providers.kubernetesIngress]
  reportNodeInternalIPs = true
  # ...
```

```
--providers.kubernetesingress.reportnodeinternalips=true
```

Routing Configuration

See the dedicated section in routing.

Further

To learn more about the various aspects of the Ingress specification that Traefik supports, many examples of Ingresses definitions are located in the test examples of the Traefik repository.

{% include-markdown "includes/traefik-for-business-applications.md" %}

□□□ #2
□ □□□ □□□ 2026-10-31 14:00:55 UTC
□ □□□ □□□ 2026-10-31 14:00:56 UTC