IngressRouteUDP

- Configuration Example
- Configuration Options
  - ExternalName Service
  - NativeLB

`IngressRouteUDP` is the CRD implementation of a Traefik UDP router.

Before creating `IngressRouteUDP` objects, you need to apply the Traefik Kubernetes CRDs to your Kubernetes cluster.

This registers the `IngressRouteUDP` kind and other Traefik-specific resources.

Configuration Example

```
apiVersion: traefik.io/v1alpha1
kind: IngressRouteUDP
metadata:
  name: ingressrouteudpfoo
  namespace: apps
spec:
  ingressClassName: traefik-lb
  entryPoints:
    - fooudp  # The entry point where Traefik listens for incoming traffic.
  routes:
  - services:
    - name: foo # The name of the Kubernetes Service to route to.
      port: 8080
      weight: 10
      nativeLB: true # Enables native load balancing between pods.
```

Configuration Options

Field	Description	Default	Required
`ingressClassName`	Defines the IngressClass cluster resource to use. It replaces the deprecated `kubernetes.io/ingress.class` annotation. The spec field takes precedence over the annotation.		No
`entryPoints`	List of entrypoints names.		No
`routes`	List of routes.		Yes
`routes[n].services`	List of Kubernetes service definitions. See here for `ExternalName Service` setup.		No
`routes[n].services[n].name`	Defines the name of a Kubernetes service.		Yes
`routes[n].services[n].port`	Defines the port of a Kubernetes service. This can be a reference to a named port.		Yes
`routes[n].services[n].weight`	Defines the weight to apply to the server load balancing.	1	No
`routes[n].services[n].nativeLB`	Controls, when creating the load-balancer, whether the LB's children are directly the pods IPs or if the only child is the Kubernetes Service clusterIP.	false	No
`routes[n].services[n].nodePortLB`	Controls, when creating the load-balancer, whether the LB's children are directly the nodes internal IPs using the nodePort when the service type is NodePort. It allows services to be reachable when Traefik runs externally from the Kubernetes cluster but within the same network of the nodes. See here for more information.	false	No

ExternalName Service

Traefik backends creation needs a port to be set, however Kubernetes ExternalName Service could be defined without any port. Accordingly, Traefik supports defining a port in two ways:

- only on `IngressRouteUDP` service
- on both sides, you'll be warned if the ports don't match, and the `IngressRouteUDP` service port is used

Thus, in case of two sides port definition, Traefik expects a match between ports.

=== "Ports defined on Resource"

````
```yaml tab="IngressRouteUDP"
apiVersion: traefik.io/v1alpha1
kind: IngressRouteUDP
metadata:
  name: test.route
  namespace: apps

spec:
  entryPoints:
    - foo
  routes:
  - services:
    - name: external-svc
      port: 80
```

```yaml tab="Service ExternalName"
apiVersion: v1
kind: Service
metadata:
  name: external-svc
  namespace: apps

spec:
  externalName: external.domain
  type: ExternalName
```
````

=== "Port defined on the Service"

```yaml tab="IngressRouteUDP"
apiVersion: traefik.io/v1alpha1
kind: IngressRouteUDP
metadata:
  name: test.route
  namespace: apps

spec:
  entryPoints:
    - foo
  routes:
  - services:
    - name: external-svc
```

```yaml tab="Service ExternalName"
apiVersion: v1
kind: Service
metadata:
  name: external-svc
  namespace: apps

spec:
  externalName: external.domain
  type: ExternalName
  ports:
    - port: 80
```

=== "Port defined on both sides"

```yaml tab="IngressRouteUDP"
apiVersion: traefik.io/v1alpha1
kind: IngressRouteUDP
metadata:
  name: test.route
  namespace: apps
```

```
spec:
  entryPoints:
    - foo
  routes:
  - services:
    - name: external-svc
      port: 80
```

```yaml tab="Service ExternalName"
apiVersion: v1
kind: Service
metadata:
  name: external-svc
  namespace: apps

spec:
  externalName: external.domain
  type: ExternalName
  ports:
    - port: 80
```

NativeLB

To avoid creating the server load-balancer with the pods IPs and use Kubernetes Service `clusterIP` directly, one should set the `NativeLB` option to true. By default, `NativeLB` is false.

```
apiVersion: traefik.io/v1alpha1
kind: IngressRouteUDP
metadata:
  name: test.route
  namespace: default
spec:
  entryPoints:
    - foo
routes:
- services:
  - name: svc
```

```
    port: 80
    # Here, nativeLB instructs to build the servers load balancer with the Kubernetes Service clusterIP only.
    nativeLB: true
```

```
apiVersion: v1
kind: Service
metadata:
  name: svc
  namespace: default
spec:
  type: ClusterIP
  ...
```

□□□ #2
□ □□□ □□□ 2026-10-31 14:05:02 UTC
□ □□□ □□□ 2026-10-31 14:05:03 UTC