ServersTransport TCP

- Configuration Example
- Configuration Options
  - terminationDelay
  - proxyProtocol.version

ServersTransport allows configuring the transport between Traefik and your TCP servers.

Configuration Example

Declare the serversTransport:

```
tcp:
  serversTransports:
    mytransport:
      dialTimeout: "30s"
      dialKeepAlive: "20s"
      terminationDelay: "200ms"
      tls:
        serverName: "example.com"
        certificates:
          - "/path/to/cert1.pem"
          - "/path/to/cert2.pem"
        insecureSkipVerify: true
        rootCAs:
          - "/path/to/rootca.pem"
        peerCertSANs:
          - type: DNSName
            value: foo.com
          - type: URI
            value: spiffe://example.org/peer
        spiffe:
```

```
          ids:
            - "spiffe://example.org/id1"
            - "spiffe://example.org/id2"
          trustDomain: "example.org"
```

```
[tcp.serversTransports.mytransport]
  dialTimeout = "30s"
  dialKeepAlive = "20s"
  terminationDelay = "200ms"

  [tcp.serversTransports.mytransport.tls]
    serverName = "example.com"
    certificates = ["/path/to/cert1.pem", "/path/to/cert2.pem"]
    insecureSkipVerify = true
    rootCAs = ["/path/to/rootca.pem"]
    peerCertSANs = [{type = "DNSName", value = "foo.com"}, {type = "URI", value =
"spiffe://example.org/peer"}]

  [tcp.serversTransports.mytransport.tls.spiffe]
    ids = ["spiffe://example.org/id1", "spiffe://example.org/id2"]
    trustDomain = "example.org"
```

Attach the serversTransport to a service:

```
tcp:
  services:
    Service01:
      loadBalancer:
        serversTransport: mytransport
```

```
## Dynamic configuration
[tcp.services]
  [tcp.services.Service01]
    [tcp.services.Service01.loadBalancer]
      serversTransport = "mytransport"
```

```
labels:
  - "traefik.tcp.services.Service01.loadBalancer.serversTransport=mytransport"
```

```
{
  // ...
  "Tags": [
    "traefik.tcp.services.Service01.loadBalancer.serversTransport=mytransport"
  ]
}
```

Configuration Options

Field	Description	Default	Required
`serverstransport.dialTimeout`	Defines the timeout when dialing the backend TCP service. If zero, no timeout exists.	30s	No
`serverstransport.dialKeepAlive`	Defines the interval between keep-alive probes for an active network connection.	15s	No
`serverstransport.terminationDelay`	Sets the time limit for the proxy to fully terminate connections on both sides after initiating the termination sequence, with a negative value indicating no deadline. More Information here	100ms	No
`serverstransport.proxyProtocol`	Defines the Proxy Protocol configuration. An empty `proxyProtocol` section enables Proxy Protocol version 2.		No
`serverstransport.proxyProtocol.version`	Traefik supports PROXY Protocol version 1 and 2 on TCP Services. More Information here	2	No
`serverstransport.tls`	Defines the TLS configuration. An empty `tls` section enables TLS.		No
`serverstransport.tls.serverName`	Configures the server name that will be used for SNI.		No

Field	Description	Default	Required
`serverstransport.tls.certificates`	Defines the list of certificates (as file paths, or data bytes) that will be set as client certificates for mTLS.		No
`serverstransport.tls.insecureSkipVerify`	Controls whether the server's certificate chain and host name is verified.	false	No
`serverstransport.tls.rootCAs`	Defines the root certificate authorities to use when verifying server certificates. (for mTLS connections).		No
`serverstransport.tls.peerCertSANs`	Defines the SANs (Subject Alternative Names) used to match against SANs during the peer certificate verification.	[]	No
`serverstransport.tls.peerCertSANs[].type`	Defines the SAN type (`URI` or `DNSName`) to match against the peer certificate's Subject Alternative Names.	""	No
`serverstransport.tls.peerCertSANs[].value`	Defines the SAN value to match against the peer certificate's Subject Alternative Names.	""	No
`serverstransport.tls.spiffe`	Defines the SPIFFE configuration. An empty `spiffe` section enables SPIFFE (that allows any SPIFFE ID).		No
`serverstransport.tls.spiffe.ids`	Allow SPIFFE IDs. This takes precedence over the SPIFFE TrustDomain.		No
`serverstransport.tls.spiffe.trustDomain`	Allow SPIFFE trust domain.	""	No

!!! note "SPIFFE"

```
Please note that SPIFFE must be enabled in the [install configuration](../../install-configuration/tls/spiffe.md) (formerly known as static configuration) before using it to secure the connection between Traefik and the backends.
```

`terminationDelay`

As a proxy between a client and a server, it can happen that either side (e.g. client side) decides to terminate its writing capability on the connection (i.e. issuance of a FIN packet). The proxy needs to propagate that intent to the other side, and so when that happens, it also does the same on its connection with the other side (e.g. backend side).

However, if for some reason (bad implementation, or malicious intent) the other side does not eventually do the same as well, the connection would stay half-open, which would lock resources for however long.

To that end, as soon as the proxy enters this termination sequence, it sets a deadline on fully terminating the connections on both sides.

The termination delay controls that deadline. A negative value means an infinite deadline (i.e. the connection is never fully terminated by the proxy itself).

`proxyProtocol.version`

Traefik supports PROXY Protocol version 1 and 2 on TCP Services. It can be configured by setting `proxyProtocol.version` on the serversTransport. The option specifies the version of the protocol to be used. Either 1 or 2.

□□□ #2
□ □□□ □□□ 2026-10-31 14:05:47 UTC
□ □□□ □□□ 2026-10-31 14:05:48 UTC