

gNMI Dial-Out Using the gRPC Tunnel Service

This module describes how to configure a tunnel service for gNMI dial-out connections. You can use the gRPC tunnel server to forward connections from external clients, such as a gRPC Network Management Interface (gNMI) or gRPC Network Operations interface (gNOI), to connect to a network device without establishing a direct connection.

- gNMI Dial-Out Using gRPC Tunnel Service, on page 1
- Information About gNMI Dial-Out Using gRPC Tunnel Service, on page 2
- How to Configure gNMI Dial-Out Using gRPC Tunnel Service, on page 3
- Verifying the gNMI Dial-Out Using gRPC Tunnel Service Configuration, on page 6
- Feature Information for gNMI Dial-Out Using gRPC Tunnel Service, on page 7

gNMI Dial-Out Using gRPC Tunnel Service

In releases prior to Cisco IOS XE Dublin 17.11.1, gNMI supports a dial-in session, where the data collector sends RPCs directly to a network device. From Cisco IOS XE Dublin 17.11.1, gNMI uses a tunnel service for gNMI dial-out connections based on the recommendation from the OpenConfig forum.

With gNMI dial-out through gRPC tunnel service, you can use a router (tunnel client) to dial out to a collector (tunnel server). After establishing a session, the tunnel server acts as a client and requests gNMI services. The tunnel server then forwards requests from one or more gNMI or gNOI clients. Note that the gRPC tunnel server and the gNMI or gNOI client may be distinct entities.

Note The gRPC tunnel design is based on the feature specifications provided in the **tunnel.proto** file.

For more information about gNMI dial-out using gRPC tunnel, see the Github repository.

Note The tunnel service supports only Transport Layer Security (TLS) sessions.

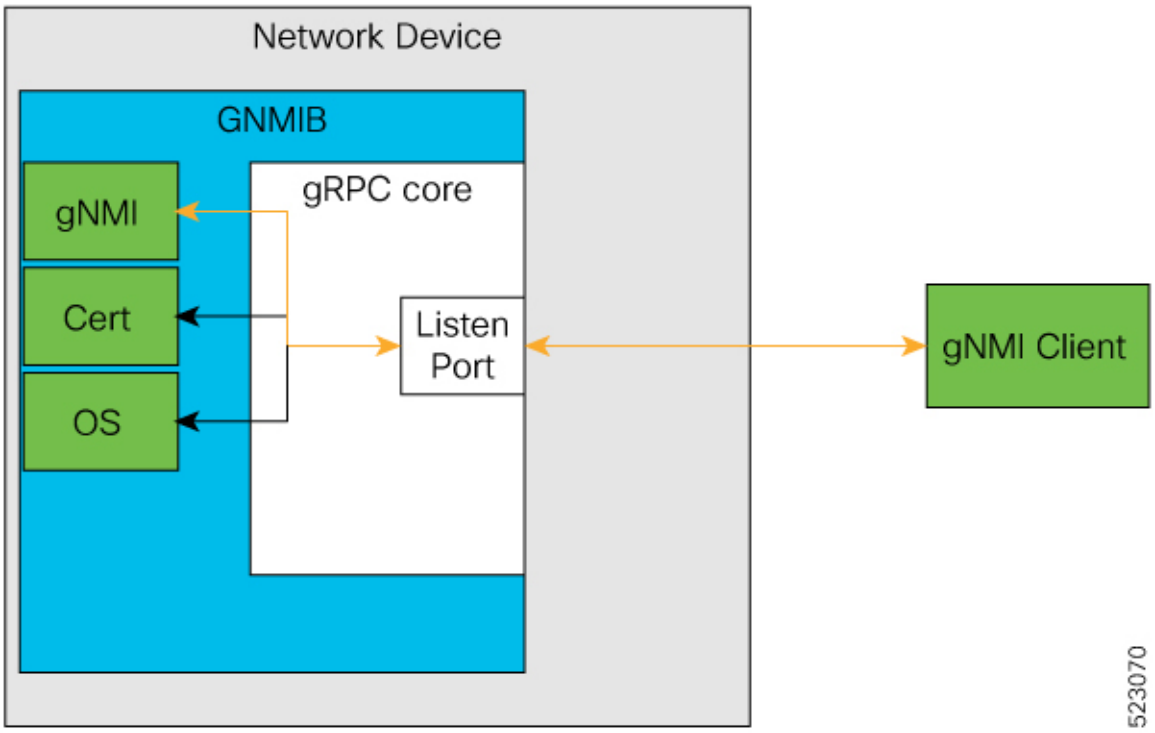
Information About gNMI Dial-Out Using gRPC Tunnel Service

The following sections provide detailed information about a traditional gRPC connection, a gRC tunnel, and connecting to GNMIB using a gRPC tunnel.

Traditional gRPC Connection

The yellow arrow in the following image shows the traditional method of connecting to a network device to access gRPC or gNOI services. The gNMI client connects to the network device only when the gNMI client is allowed to establish a direct connection.

Figure 1: Traditional gRPC Connection



gRPC Tunnel

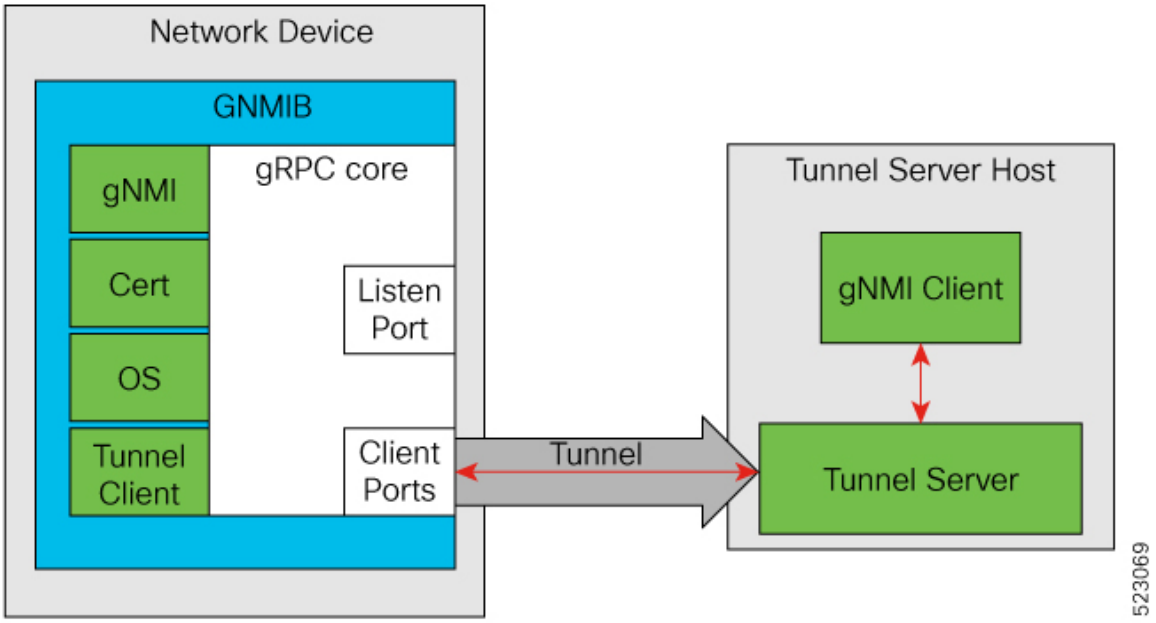
The following are the main components of a gRPC tunnel:

- Target: Represents a single service on a network device. For example, gNMI or gNOI is one target type. A tunnel client can register one or more target types with a tunnel server.
- Tunnel: A bidirectional stream in which data can be forwarded between the tunnel client and the server.
- Tunnel Server: The gRPC server that manages the target subscriptions and registrations.
- Tunnel Client: GNMIB is the gRPC tunnel client.

Connecting to GNMIB Using the gRPC Tunnel

In the gRPC tunnel design, the traditional flow is reversed. The network device dials out to the gRPC tunnel server. This leads to the gRPC tunnel server and any gNMI or gNOI clients to be unaware of the network device addresses and locations. Also, the network devices can access the gRPC tunnel server even if its outgoing connections are blocked.

Figure 2: A New Method of Connecting to GNMIB



How to Configure gNMI Dial-Out Using gRPC Tunnel Service

The following sections provide detailed information about the configurations that comprise the larger gNMI dial-out using gRPC tunnel service configuration.

Configuring and Enabling a Target

Run the following commands on a network device to configure and enable the target:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **gnxi grpctunnel target** {**GNMI_GNOI** | **GNMI_GNOI_INSECURE**}
4. **enable**
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**gnxi grpctunnel target** {**GNMI_GNOI** | **GNMI_GNOI_INSECURE**} **Example:** `Device(config)# gnxi grpctunnel target GNMI_GNOI`	Configures a gRPC tunnel target, and enters target configuration mode. **Note** Only **GNMI_GNOI** targets are supported. The **GNMI_GNOI_INSECURE** target is for testing purposes only and always connects to the GNMIB's insecure port.
Step 4	**enable** **Example:** `Device(config-target)# enable`	Enables the tunnel target.
Step 5	**end** **Example:** `Device(config-target)# end`	Exits target configuration mode and returns to privileged EXEC mode.

Configuring a gRPC Tunnel

Run the following commands to configure and enable a target in a network device that is part of a gRPC tunnel. Configure the IP address of the tunnel server, the port the tunnel server listens on, and the source or outgoing VRF. The following configuration task shows how the target sends data to the server:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **gnxi grpctunnel destination** *destination-name*
4. **enable**
5. **address** *IP-address*
6. **port** *port-number*
7. **identity-trustpoint** *trustpoint-name*
8. **source-address** *IP-address*
9. **source-vrf** *VRF-name*

10. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**gnxi grpctunnel destination** *destination-name* **Example:** `Device(config)# gnxi grpctunnel destination foobar`	Configures a gRPC tunnel destination and enters destination configuration mode.

Step 4	**enable**
	Example:
	`Device(config-destination)# enable`
	
	`% node-1:dbm:gnmi:Destination address must be set when destination is enabled`
		Enables the tunnel destination.

Step 5	**address** *IP-address* **Example:** `Device(config-destination)# address 209.165.200.225`	Configures the destination IP address.
Step 6	**port** *port-number* **Example:** `Device(config-destination)# port 1234`	Configures the destination port. • Valid values for the *port-number* argument are from 0 to 65535.
Step 7	**identity-trustpoint** *trustpoint-name* **Example:** `Device(config-destination)# identity-trustpoint trustpoint1`	Configures the specified trustpoint certificate as the TLS identity when connecting securely to a destination.
Step 8	**source-address** *IP-address* **Example:** `Device(config-destination)# source-address 209.165.201.30`	Configures the outgoing source address to use when connecting to the tunnel server or destination.

	Command or Action	Purpose
Step 9	**source-vrf** *VRF-name* **Example:** `Device(config-destination)# source-vrf Mgmt-vrf`	Configures a source Virtual Routing and Forwarding (VRF) instance when connecting to the tunnel server or destination.
Step 10	**end** **Example:** `Device(config-destination)# end`	Exits destination configuration mode and returns to privileged EXEC mode.

Verifying the gNMI Dial-Out Using gRPC Tunnel Service Configuration

Use the following command to verify the state of the gRPC tunnel service interface:

```
Device# show gnxi state detail

Settings
========
Server: Enabled
Server port: 50052
Secure server: Enabled
Secure server port: 9339
Secure client authentication: Disabled
Secure trustpoint: gnoi_pyats
Secure client trustpoint:
Secure password authentication: Disabled

GNMI
====
Admin state: Enabled
Oper status: Up
State: Provisioned
gRPC Server
-----------
Admin state: Enabled
Oper status: Up
Configuration service
---------------------
Admin state: Enabled
Oper status: Up
Telemetry service
-----------------
Admin state: Enabled
Oper status: Up

GNOI
====
Cert Management service
-----------------
Admin state: Enabled
Oper status: Up
OS Image service
----------------
Admin state: Enabled
Oper status: Up
```

```
Supported: Supported
Factory Reset service
---------------------
Admin state: Enabled
Oper status: Up
Supported: Supported

GRPC Tunnel
===========
Admin state: Enabled
Oper status: Up
```

Use the following command to display the statuses of all the currently configured gRPC tunnel servers:

```
Device# show gnxi grpctunnel destinations

All configured destinations
Destination Name: foobar
Target: GNMI_GNOI
Tag: 1
Registered: Yes
Session Started: Yes
Tunnel Active: Yes
Error:
Destination Name: example
Target: GNMI_GNOI
Tag: 1
Registered: Yes
Session Started: Yes
Tunnel Active: Yes
Error:
```

Feature Information for gNMI Dial-Out Using gRPC Tunnel Service

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for gNMI Dial-Out Using gRPC Tunnel Service

Feature Name	Release	Feature Information
gNMI Dial-Out Using gRPC Tunnel Service	Cisco IOS XE Dublin 17.11.1	This feature allows you to configure a network device (tunnel client) to register certain targets (preapproved services) with a gRPC tunnel server through the CLI. The following commands were introduced for this feature: • **gnxi grpctunnel destination** • **gnxi grpctunnel target** This feature was introduced on the following platforms: • Cisco Catalyst 9200, 9200L, and 9200CX Series Switches • Cisco Catalyst 9300, 9300L, and 9300X Series Switches • Cisco Catalyst 9400 and 9400X Series Switches • Cisco Catalyst 9500 and 9500-High Performance Series Switches • Cisco Catalyst 9600 and 9600X Series Switches • Cisco Network Convergence System 4200 Series