

Revised: April 13, 2026

Cisco Nexus Hyperfabric — Hyperfabricbot

Hyperfabricbot

Hyperfabricbot is a bot designed to help you monitor, query, and troubleshoot your network fabric through integrated messaging platforms like Webex and Discord.

Once you add the bot to your messaging platform, you interact with its interface by entering `@AskHyperfabric` in Webex or using `/ask` or `/` commands in Discord. By using conversational language powered by Natural Language Processing (NLP), and structured commands, the bot provides a way to access real-time fabric data and product documentation without requiring constant access to the Nexus Hyperfabric GUI.

Key advantages of using the Hyperfabricbot:

- **GUI independence:** Retrieve real-time telemetry, fabric status, and assertion details directly within your messaging platform. This reduces the need to log into the Nexus Hyperfabric GUI for routine monitoring or rapid troubleshooting.
- **External visibility:** Administrators can provide temporary, read-only access to users who do not have a Nexus Hyperfabric account for specific troubleshooting or inquiry sessions without creating permanent system credentials.
- **Documentation access:** Ask Hyperfabricbot for configuration steps, troubleshooting procedures, and similar tasks.

Hyperfabricbot specifications and limitations

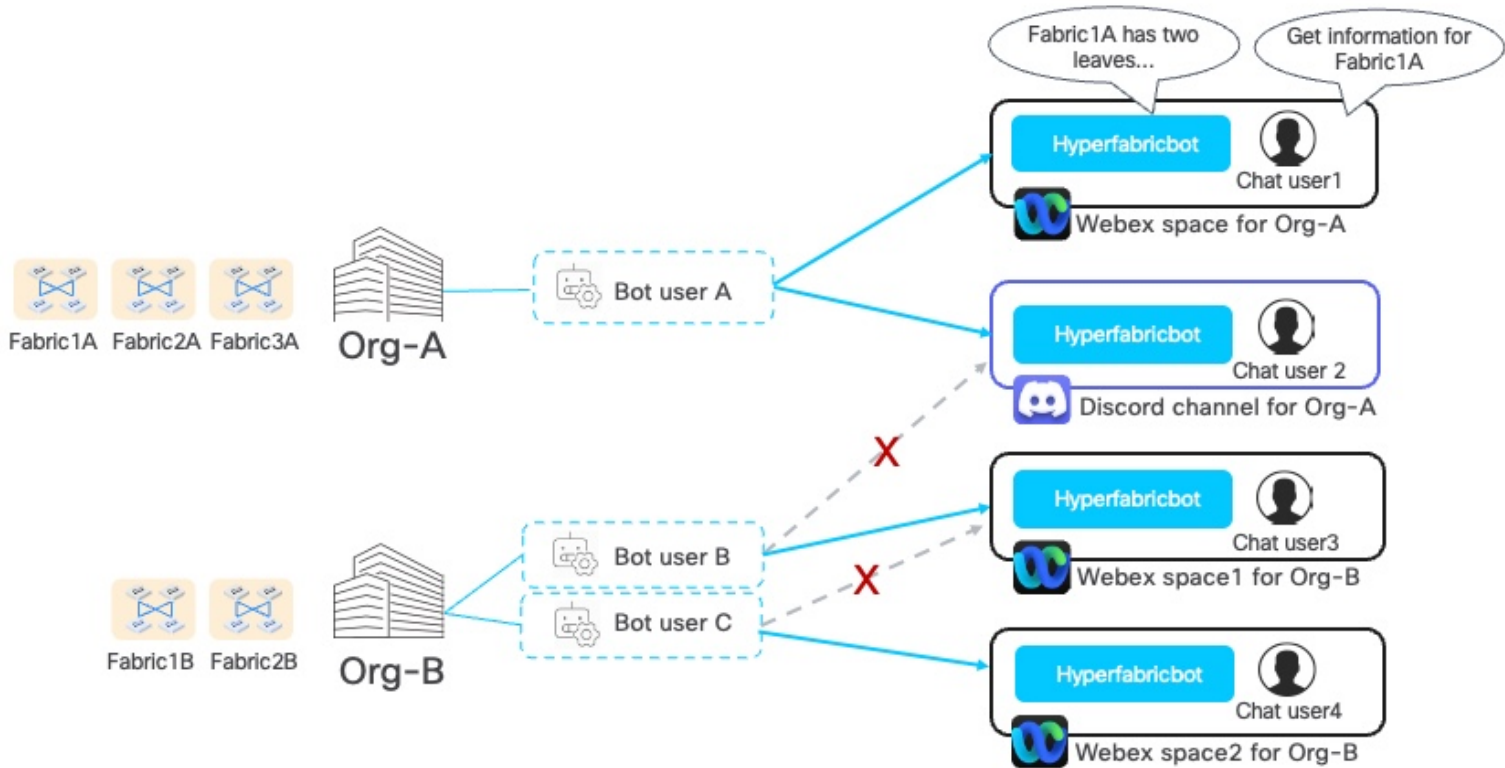
Review these specifications to understand bot limits and user access.

Operational scope

Hyperfabricbot supports read-only monitoring and troubleshooting queries on Webex and Discord. When you are added to a chat room, you can query individual fabrics or retrieve a list of all fabrics within your organization. You cannot make configuration changes using Hyperfabricbot. For more information, refer to [Supported structured commands and query scope, on page 10](#).

Organizations, bot users, and chat rooms

Figure 1: Organizations, bot users, and chat rooms



You can configure bot users, organizations, and chat rooms (Webex spaces or Discord channels) in these ways:

- You can bind multiple bot users (Bot user B and Bot user C) to an organization.
- Each bot user can be bound to only one organization.
- Each chat room can be associated to only one bot user and one organization.
- A bot user can be bound to multiple chat rooms if it is the only bot user in each room and each room is using a different messaging platform (Bot user A is bound to two chatrooms: a Webex space and a Discord channel). This approach allows you to separate chat rooms for specific needs. For example, use one chat room for a team handling a specific issue, and another chat room for monitoring a fabric.

For setup information and details about bot users, refer to [Configuring Hyperfabricbot, on page 3](#).

Security and access

The bot uses a linked account model where Nexus Hyperfabric administrators manage user access. The system establishes a connection through a unique binding between a specific chat room (space or channel) and a virtual bot user. When you add a user to a chat room, they can query Hyperfabricbot, regardless of their individual Nexus Hyperfabric account status.

Configuring Hyperfabricbot

To effectively deploy the Nexus Hyperfabric bot, it is essential to understand the distinction between Hyperfabricbot and the virtual bot user. Hyperfabricbot is the bot that you interact with and invite into your chat room (Webex space or Discord channel). The bot user is the "account" created to represent your organization.

Configuring Hyperfabricbot requires three high-level steps.

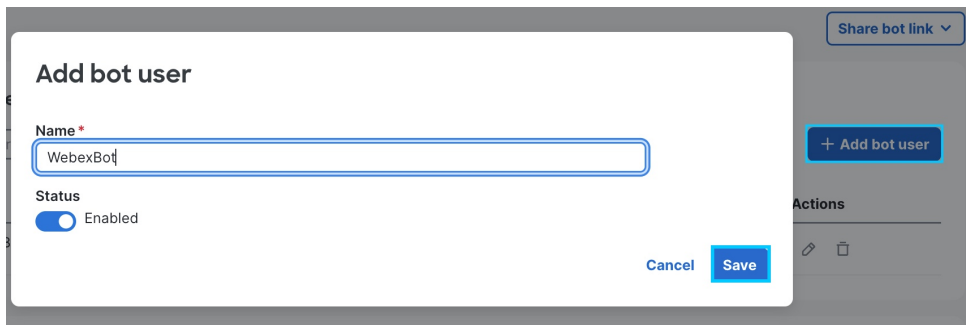
1. **Create a bot user:** Create a bot user in Nexus Hyperfabric for an organization. You can create multiple bot users in an organization.
2. **Invite and board the bot:** Invite Hyperfabricbot into your specific messaging platform space or channel.
3. **Bind the bot user (account binding):** Link the messaging platform to the bot user using a unique verification ID. You can bind only one bot user for each chat room.

Configure Hyperfabricbot for Webex

Use this procedure to create a virtual bot user, invite Hyperfabricbot to the Webex space, and bind the account for active communication.

Step 1 Create a virtual bot user in Nexus Hyperfabric.

- a) Choose **Administration > User and bot management > Bots**.
- b) Select **+Add bot user**.
- c) Enter a name for the bot user and select **Save**.



The system appends letters and numbers to the name. This ensures that each bot user remains unique.

Bot users

Q Search by name 1 result

Name	Bindings	User status
WebexBot-2bb52b65-2bb52b65	1	Enabled

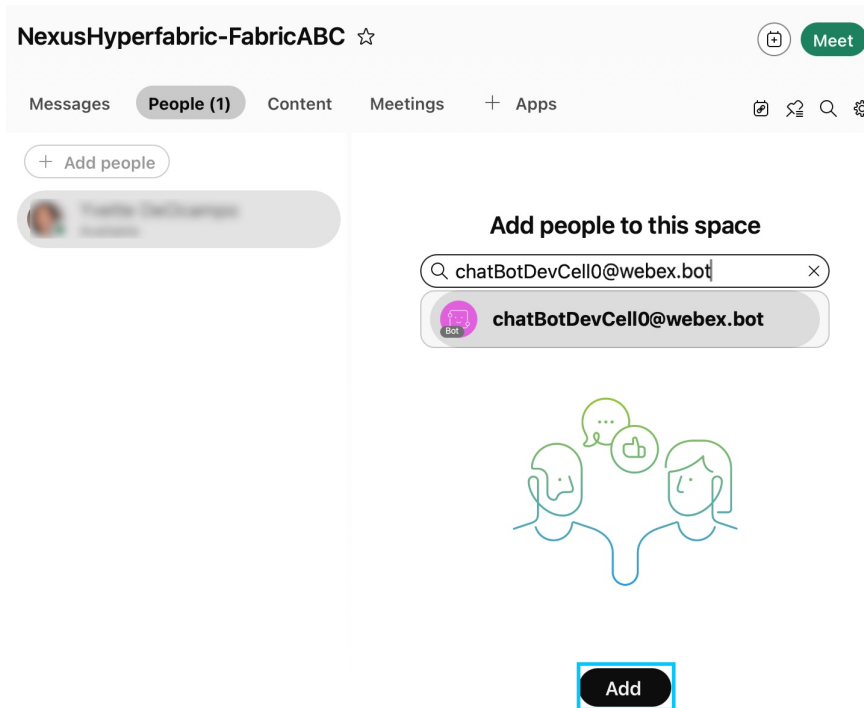
- d) From the **Share bot link** drop-down list, select **Copy Webex bot ID**. You will use this ID in Webex to add Hyperfabricbot.



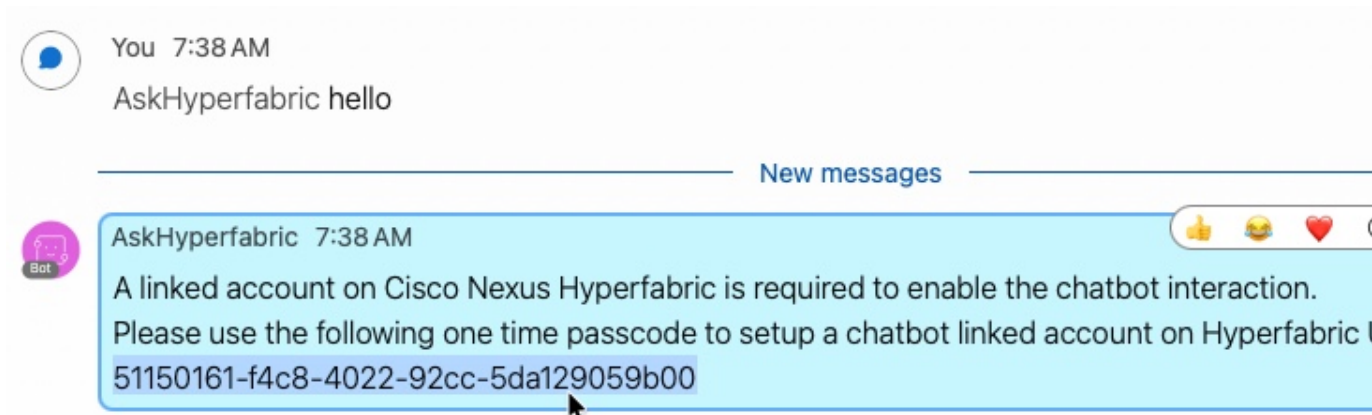
Step 2

Invite Hyperfabricbot to Webex.

- In Webex, create a dedicated space with a descriptive name for Nexus Hyperfabric operations.
- In the Webex space, choose **People** > + **Add people**.
- Paste the Webex bot ID and select **Add**.



- In the Webex messaging space, enter **@AskHyperfabric hello**.



- e) Copy the passcode (binding code) provided by Hyperfabricbot. You will use this binding code in Nexus Hyperfabric to bind the bot user.

Step 3 Bind the bot user to the Webex space.

- a) In Nexus Hyperfabric (**Administration > User and bot management > Bots**), select + **Bind bot user**.
- b) From the **Select bot user** drop-down list, select the Webex bot user.
- c) In the **Enter binding code** field, paste the binding code.
- d) Select **Save**.

Bind bot user

Select bot user*

WebexBotUser1-2bb52b65-2bb52b65

Enter binding code * ⓘ

13f01b01-627a-439b-bd9d-3855bb7e4d87

✓ Binding code is valid

Confirm that you want to bind **WebexBotUser1-2bb52b65-2bb52b65** with the following service session.

Service Webex
Service Details Space Name: NexusHyperfabric-Org1-FabricABC
Space ID: 9423b990-0204-11f1-9b11-7b2c95533e6a

Enable binding immediately

Enabled

Cancel Save

Step 4 Start interacting with Hyperfabricbot. Enter **@AskHyperfabric** followed by a command or question.

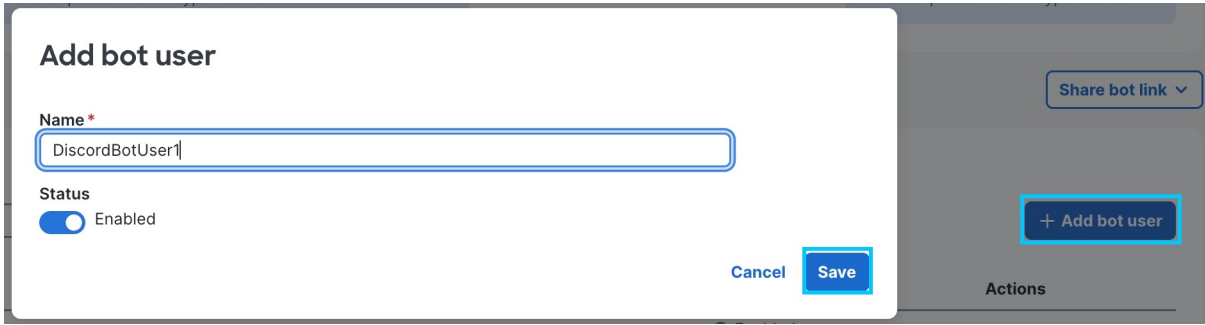
Configure Hyperfabricbot for Discord

This procedure includes authorizing Discord bot access using an invitation link and binding the connection through Nexus Hyperfabric. Images illustrate the process, but they may not match your version of Discord. For the most recent Discord information, review the official Discord documentation.

Create a server and channel in Discord where you want to host Hyperfabricbot before proceeding with this procedure.

Step 1 Create a virtual bot user in Nexus Hyperfabric.

- a) Choose **Administration > User and bot management > Bots**.
- b) Select +**Add bot user**.
- c) Enter a name for the bot user and select **Save**.



The system adds letters and numbers to the name. This ensures that each bot user remains unique.

Bot users

Search by name 2 results [+ Add bot user](#)

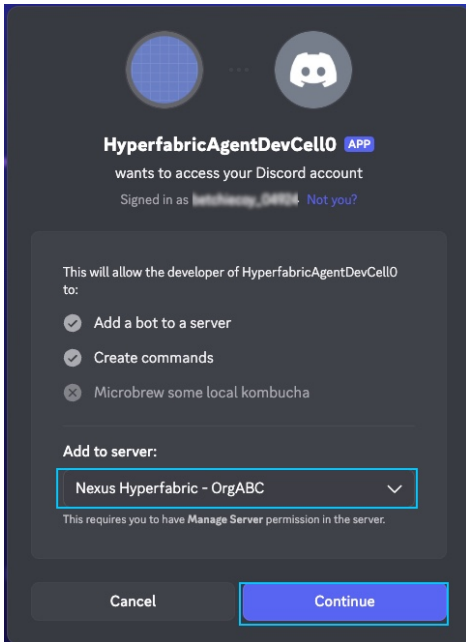
Name	Bindings	User status	Actions
DiscordBotUser1-2bb52b65-2bb52b65	0	Enabled	
WebexBotUser1-2bb52b65-2bb52b65	1	Enabled	

d) From the **Share bot link** drop-down list, select **Copy Discord bot link**.

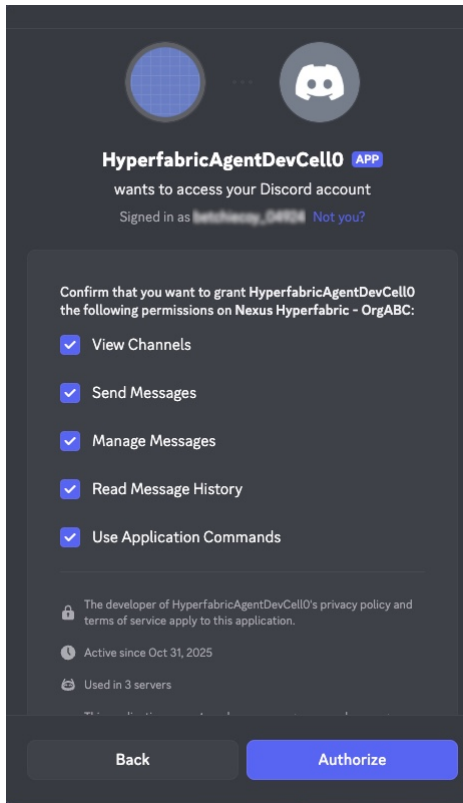
Step 2

Invite the Hyperfabricbot to Discord.

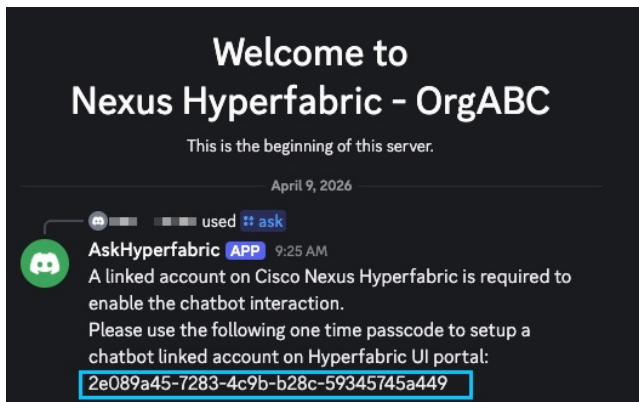
- Paste the invitation link into a web browser. The Discord application opens in the web browser.
- Navigate through the prompts, until you can select a server where you want to add Hyperfabricbot.
- Select the dedicated Discord server where you want to host Hyperfabricbot and select **Continue**.



d) Check all permissions and select **Authorize**. You can choose to close the browser window where the success message appears.



- e) Open Discord and select the server and channel where Hyperfabricbot was added.
- f) In the Discord messaging space, enter `/ask help`.
- g) Copy the passcode (binding code) provided by Hyperfabricbot. You will use this binding code in Nexus Hyperfabric to bind the bot user.



Step 3

- Bind the bot user to the Discord channel.
 - a) In Nexus Hyperfabric (**Administration** > **User and bot management** > **Bots**), select + **Bind bot user**.
 - b) From the **Select bot user** drop-down list, select the Discord bot user.
 - c) In the **Enter binding code** field, paste the binding code.
 - d) Select **Save**.

Bind bot user

Select bot user*

DiscordBot-2bb52b65-2bb52b65

Enter binding code* ⓘ

37de94ef-af0c-4001-af34-b836f836a14a

✔ Binding code is valid

Confirm that you want to bind **DiscordBot-2bb52b65-2bb52b65** with the following service session.

Service Discord

Service Details Server: Nexus Hyperfabric - OrgABC; 1465863828439826525

Channel: fabric-123; 1465870020407201964

Enable binding immediately

Enabled

Cancel Save

Step 4 Start interacting with Hyperfabricbot.

- a) Enter / and scroll down to see a list of structured commands.
- b) Enter **/ask list** to display all structured commands.
- c) Enter **/ask** and a conversational prompt. For example: **/ask Show all fabrics.**

Disable Hyperfabricbot

Use this procedure to temporarily disable Hyperfabricbot.

Step 1 Choose **Administration > User and bot management > Bots.**

Step 2 In the **Bot bindings** area, locate the bot user you want to disable.

Step 3 From the **Action** column, choose **...** > **Disable**. The bot user account is immediately disabled from the associated messaging platform. To enable it, perform this procedure again, but select **Enable**.

Unbind and delete Hyperfabricbot


To delete Hyperfabricbot from a Webex space or Discord channel, first remove the chatbot binding and then delete the bot user.

Step 1 Choose **Administration > User and bot management > Bots.**

Step 2 In the **Bot bindings** area, locate the bot user you want to delete.

Step 3 From the **Action** column, choose **...** > **Delete bot binding.**

Step 4 To confirm deletion, select **Delete**. The system disables the bot user account in the associated messaging platform.

Step 5 In the **Bot users** area, locate the bot user you want to delete and select .

Step 6 To confirm deletion, select **Delete**. The bot user is deleted.

Interacting with Hyperfabricbot

Learn to query Hyperfabricbot using natural language or structured commands to monitor your network fabric. Use the specific syntax required for your messaging platform.

Webex interaction

In Webex, you must enter `@AskHyperfabric` to initiate a request.

- **Method:** Type `@AskHyperfabric` followed by your command or question.
- **Example:** `@AskHyperfabric Show me information about FabricABC.`

Discord interaction

Discord supports both structured slash commands for direct actions and the `/ask` command for free-form questions.

- **Slash commands:** Type `/` to view and select from the list of available structured commands, such as `/watch_fabric` or `/list_fabric`.
- **Free-form queries:** Type `/ask` followed by your question to initiate a natural language request.
- **Example:** `/ask Show me the port counters.`

Query scope and help

The Nexus Hyperfabric bot is designed specifically for fabric operations. If you submit a query that falls outside the scope of Nexus Hyperfabric, Hyperfabricbot will return a "No matching command found" or similar message. To get a list of supported commands, type `@AskHyperfabric help` in Webex or `/ask list` in Discord.

Product information queries

You can ask Hyperfabricbot about Nexus Hyperfabric configuration guidelines, troubleshooting procedures, and operational workflows using conversational prompts. Hyperfabricbot provides context-specific answers based on product documentation.

Table 1: Examples of product queries

Category	Prompt examples
General	Tell me about Nexus Hyperfabric.
Configuration	How do I create a BGP peering?
Operational tasks and requirements	What are the requirements for AI infrastructure?
Troubleshooting	How do I troubleshoot BGP session states?

Fabric management queries

Use fabric management queries to monitor, troubleshoot, and verify your network fabric using conversational prompts or [structured commands](#). The Hyperfabricbot supports these type of fabric management queries:

- **Monitor fabric health:** You can identify and track system errors by telling the Hyperfabricbot to "Show me all issues in [fabric-name]" or by using the structured command `list_assert fabric_entity=[fabric-name]`.

- **Troubleshoot and resolve issues:** When you need to restore fabric health, you can ask, "How do I fix the error on [device-name]?" or use the `show_remedy_action fabric_entity=[fabric-name]` command to receive clear, actionable steps.
- **Track fabric state proactively:** You can stay informed about real-time changes by telling the Hyperfabricbot to "Start watching [fabric-name] for events," or by executing the `watch_fabric fabric_entity=[fabric-name]` command to receive automated updates.
- **Verify device configurations:** You can confirm that your devices are set up correctly by asking, "Show me the configuration for [device-name]," or by using the `show_node_config device_entity=[device-name]` command to retrieve current settings.
- **Access real-time telemetry:** You can gain immediate visibility into device-level data by asking, "Can I see the port counters for [device-name]?" or by using the `show_port_counters device_entity=[device-name]` command to pull interface statistics.

Supported structured commands and query scope

The Hyperfabricbot utilizes a specific set of structured commands. All specific fabric inquiries—whether initiated via Natural Language Processing (NLP) or direct structured commands—are strictly limited to these functions.

To see all supported structured commands and their descriptions, type `@AskHyperfabric help` in Webex or `/ask help` in Discord.

Table 2: Supported structured commands

Structured command	Description
<code>list_fabric</code>	Returns a list of all fabrics in the organization.
<code>list_assert fabric_entity=fabric-name</code>	Lists assertions for a specified fabric.
<code>show_fabric fabric_entity=fabric-name</code>	Displays high-level details for a specified fabric.
<code>show_neighbor entity=fabric-name device-name</code>	Displays neighbor discovery information.
<code>show_node_config device_entity=device-name</code>	Retrieves configuration details for a specific device.
<code>Show_assert device_entity=device-name assert_type=assert-type [assert_state=all true false]</code>	Shows detailed assertions for a device.
<code>show_l2fdb entity=fabric-name device-name</code>	Displays the Layer 2 Forwarding Database.
<code>show_l3table entity=fabric-name device-name</code>	Displays the Layer 3 routing table.
<code>show_port_counters device_entity=device-name[port_name=interface-name]</code>	Retrieves interface statistics.
<code>show_port_state device_entity=device-name[port_name=interface-name]</code>	Displays the operational state of ports.
<code>show_proc_table device_entity=device-name</code>	Retrieves the process table for a specific device.
<code>show_remedy_action fabric_entity=fabric-name</code>	Provides actionable remediation steps.
<code>show_vni entity=fabric-name device-name</code>	Displays Virtual Network Instance (VNI) details.



Structured command	Description
<code>show_vlan entity=fabric-name device-name [vlan_id=vlan id]</code>	Shows VLAN configurations.
<code>show_vrf entity=fabric-name device-name [ip_prefix=ip-prefix]</code>	Displays VRF details.
<code>watch_fabric fabric_entity=fabric-name</code>	Initiates proactive monitoring for a specific fabric
<code>cancel_fabric fabric_entity=fabric-name</code>	Stops proactive monitoring for a specific fabric
<code>show_watch_fabric</code>	Lists all currently active fabric watches.
<code>clear_session</code>	Clears current session context.

Getting started with Hyperfabricbot

Use these conversational examples to retrieve real-time data on fabric health, assertions, and interface statistics without memorizing complex command syntax.

Table 3: Example of plain language questions

Category	Example prompts
Fabric information and health	<ul style="list-style-type: none"> • Show me information about <i>fabric-name</i>. • What is the current status of <i>fabric-name</i>?
Assertions	<ul style="list-style-type: none"> • Show me all issues in <i>fabric-name</i>. • Show fixes for <i>fabric-name</i>..
Devices	Show me all devices for <i>fabric-name</i> .
Device issues	Show me the errors for <i>device-name</i> .
Neighbor discovery	Who are the neighbors for <i>device-name</i> ?
Interface statistics	Can I see port counters for <i>device-name</i> ?
Node configuration	Show me the configuration for <i>device-name</i> .
VLANs	<ul style="list-style-type: none"> • What VLANs exist on <i>device-name</i>? • Show me the VNI mappings for <i>fabric-name</i>.
Monitoring	<ul style="list-style-type: none"> • Start watching <i>fabric-name</i> for events. • List all active fabric watches. • Stop watching <i>fabric-name</i>.

Category	Example prompts
Hyperfabricbot session interaction	<ul style="list-style-type: none"> <li data-bbox="834 296 1422 428"> • Hello <div style="display: inline-block; vertical-align: middle;">  Prompts Hyperfabricbot to introduce itself and provide some general information as to what it can do. <p>Note</p> </div> <li data-bbox="834 457 1321 590"> • Clear <div style="display: inline-block; vertical-align: middle;">  Clears the current session context. <p>Note</p> </div>

Hyperfabricbot interaction tips

Use these tips to get the best results when querying Hyperfabricbot.

- **Be specific:** When using natural language, provide clear, direct questions. Avoid vague or overly complex sentences to ensure the bot understands your intent.
- **Use structured commands for complex tasks:** While natural language is convenient, structured commands are more precise for retrieving specific telemetry or configuration data.
- **Verify your syntax:** Remember that interaction methods vary by platform. Use **@AskHyperfabric** in Webex and slash commands or **/ask** in Discord.
- **Stay within scope:** The bot is designed for fabric-related operations or questions that can be answered by documentation. If you receive an error message, verify that your request matches a supported command. Note that errors may also occur if you query for data that does not currently exist in your fabric, such as active assertions or specific device configurations.
- **Leverage the help command:** If you are unsure of the correct syntax or available options, type **@AskHyperfabric help** in Webex or **/ask list** in Discord to view a list of all supported structured commands.