

# HYPR Integration with SIEMs

Sending HYPR event data to to a SIEM

---

## Overview

HYPR Event Hooks integrates with virtually any SIEM to stream event data in real time from HYPR to the SIEM. All HYPR software services generate detailed events revealing insights into the state of services and transactions. While the HYPR events can be viewed in the HYPR Control Center administrative interface, it is best to ingest the data into a purpose-built SIEM for long-term storage and data mining.

## Technology

HYPR Event Hooks utilize web hook technology to stream HYPR events to an external SIEM. Simply configure an HTTP Event Collector (HEC) in the target SIEM to receive the HYPR events. Each time HYPR generates an event, it is immediately added to the event stream

## Configuration

HYPR supports Splunk and Datadog out of the box, but any SIEM that supports web hooks can be configured as a target for HYPR event streaming.

Configuring HYPR Event Hooks involves setting up the SIEM to receive events with a HEC and configuring HYPR with the SIEM target information.

### Sending events to Splunk

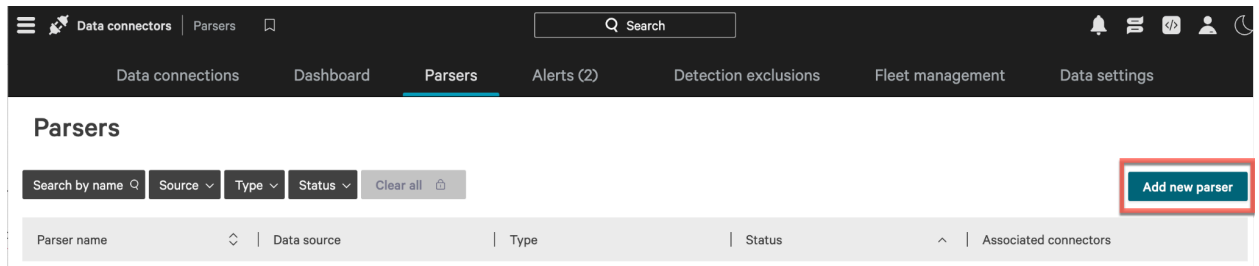
HYPR has a built-in integration with Splunk. The HYPR product documentation describes the steps to set up both Splunk and HYPR. See [Event Hook: Splunk](#) in the HYPR documentation.

### Sending events to Crowdstrike

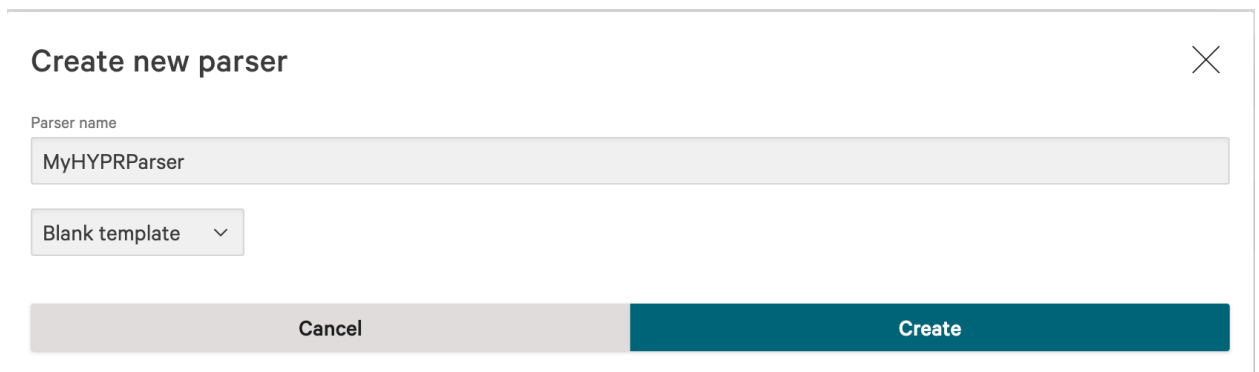
Crowdstrike supports receiving data over an HTTP Event Data Collector, which can be found in their [online documentation](#). As part of the event collector configuration, you must also [configure a parser](#) to transform the HYPR data into a common format for Crowdstrike.

## Configure the CrowdStrike parser

1. In the Falcon console, go to Next-Gen SIEM > Log Management > Data onboarding > Parsers. Click on the **Add new parser** button



2. In the dialog box, give the parser a name and select "Blank template" in the drop-down. Click the **Create** button.

A screenshot of the 'Create new parser' dialog box. The dialog has a title bar with a close button (X) on the right. Below the title, there is a 'Parser name' label and a text input field containing 'MyHYPRParser'. Below the input field is a dropdown menu currently showing 'Blank template'. At the bottom of the dialog, there are two buttons: a grey 'Cancel' button on the left and a teal 'Create' button on the right.

- You will be brought to a parser edit GUI, which comes pre-populated with some parser code. It looks like this:

**Edit parser**

Parser script  Generate parser

```

1 /*
2 # Log Parsing Template
3 This template implements CrowdStrike's Parsing Standard for log normalization.
4 Reference: https://falcon.crowdstrike.com/documentation/page/u05f69c9/crowdstrike-parsing-standard
5 https://falcon.us-2.crowdstrike.com/documentation/page/u05f69c9/crowdstrike-parsing-standard
6 https://falcon.eu-1.crowdstrike.com/documentation/page/u05f69c9/crowdstrike-parsing-standard
7 */
8
9 // =====
10 // STATIC METADATA DEFINITIONS
11 // =====
12 | Vendor := "REQUIRED_INPUT" // vendor name https://developer.crowdstrike.com/docs/ng-siem/vendors
13 | Parser.version := "1.0.0" // parser version
14 | ecs.version := "8.17.0" // ECS Version https://www.elastic.co/docs/reference/ecs
15 | Cps.version := "2.0.0" // CPS Version https://developer.crowdstrike.com/docs/ng-siem/cps-standard/
16 | event.module := "REQUIRED_INPUT" // product/module name https://developer.crowdstrike.com/docs/ng-siem/event-modules
17 | observer.type := "REQUIRED_INPUT" // https://www.elastic.co/docs/reference/ecs-observer#field-observer-type
18
19 // =====
20 // PREPARSE - Identify your log format and un-comment the relevant code
21 // =====
22
23 // JSON Format (https://library.humio.com/data-analysis/functions-parsejson.html)
24 // Example log: {"timestamp":"2023-04-23T14:30:45.123Z","level":"INFO","message":"User login","user_id":"12345"}
25 // | parseJson(prefix="Vendor.", excludeEmpty="true", handleNull="discard")
26
27 // Syslog Format - RFC5424 (https://library.humio.com/data-analysis/syntax-regex.html)
28 // Example log: <34>1 2023-04-23T14:30:45.123Z host1 app1 1234 ID47 [exampleSDID@32473

```

Fields to tag: Cps.version Vendor ecs.version event.dataset event.kind event.module event.outcome observer.type

Test data Failed: 3 + Add test Run tests

Show only failed tests

- 2018-10-15T12:51:40+00:00 [INFO] This is an example log entry. id=123 fruit=banana
- ! Failed to validate event against data schema. Please see the "Schema violations" tab for more information. ... and 1 more error
- 2018-10-15T12:52:42+01:30 [ERROR] Here is an error log entry. class=c.o.StringUtil fruit=pineapple
- ! Failed to validate event against data schema. Please see the "Schema violations" tab for more information. ... and 1 more error
- 2018-10-15T12:53:12+01:00 [INFO] User logged in. user\_id=1831923 protocol=http
- ! Failed to validate event against data schema. Please see the "Schema violations" tab for more information. ... and 1 more error

Cancel Save and exit

## 4. Select all the code on the left and deleted it

The screenshot shows the 'Edit parser' interface in the HYPR application. The top navigation bar includes 'Data connectors', 'Parsers > MyHYPRParser', a search bar, and user icons. The main area is split into two panes. The left pane, titled 'Parser script', contains a single line of code: '1 |'. The right pane, titled 'Test data', shows a list of test results. The first entry is an INFO log: '2018-10-15T12:51:40+00:00 [INFO] This is an example log entry. id=123 fruit=banana'. Below it is a red error message: '! Failed to validate event against data schema. Please see the "Schema violations" tab for more information. ... and 1 more error'. The second entry is an ERROR log: '2018-10-15T12:52:42+01:30 [ERROR] Here is an error log entry. class=c.o.StringUtil fruit=pineapple'. Below it is a red error message: '! Failed to validate event against data schema. Please see the "Schema violations" tab for more information. ... and 1 more error'. The third entry is an INFO log: '2018-10-15T12:53:12+01:00 [INFO] User logged in. user\_id=1831923 protocol=http'. Below it is a red error message: '! Failed to validate event against data schema. Please see the "Schema violations" tab for more information. ... and 1 more error'. At the bottom of the interface, there are 'Cancel' and 'Save and exit' buttons.

## 5. Next copy/paste this code into the Parser script text box:

```
// HYPR Parser

// #region PREPARSE
/*****
***** Parse JSON payload with prefix "Vendor."
***** This flattens the JSON so that all keys are prefixed with "Vendor."
*****/
| toJson(prefix="Vendor.", handleNull="discard", excludeEmpty=true)

// Attempt to parse the timestamp from detail.data.eventTimeInUTC (milliseconds)
// or, if not present, from Vendor.time using an explicit ISO8601 format.
| case {
  Vendor.detail.data.eventTimeInUTC = *
    | parseTimestamp(field="Vendor.detail.data.eventTimeInUTC",
format="milliseconds");
  Vendor.time = *
    | parseTimestamp(field="Vendor.time", format="yyyy-MM-dd'T'HH:mm:ssX");
  *;
}
```

```
}
// #endregion

// #region METADATA
/***** Static Metadata Definitions (required fields)
*****/
| Parser.version := "1.0.0"
| Vendor := "hypr"
| event.kind := "event"
| event.module := "fido2"
| ecs.version := "8.11.0"
| Cps.version := "1.0.0"
| event.dataset := "fido2.registration"
| event.category[0] := "authentication"
| event.type[0] := "user"
// #endregion

// #endregion

// #region NORMALIZATION
/***** Normalize additional fields from the JSON payload to CPS fields
*****/

// Core event fields
| message := rename(Vendor.detail.data.message)
| ip_address := rename(Vendor.detail.data.remoteIP)
| user.name := rename(Vendor.detail.data.machineUserName)

// Map additional vendor-specific fields for further context
| Vendor.eventId := rename(Vendor.detail.data.id)
| Vendor.dataVersion := rename(Vendor.detail.data.version)
| Vendor.eventDataType := rename(Vendor.detail.data.type)
| Vendor.subName := rename(Vendor.detail.data.subName)
| Vendor.loggedBy := rename(Vendor.detail.data.eventLoggedBy)
| Vendor.loggedTime := rename(Vendor.detail.data.loggedTimeInUTC)
| Vendor.tenantId := rename(Vendor.detail.data.tenantId)
| Vendor.userAgent := rename(Vendor.detail.data.userAgent)
| Vendor.traceId := rename(Vendor.detail.data.traceId)
| Vendor.deviceType := rename(Vendor.detail.data.deviceType)
| Vendor.rpAppId := rename(Vendor.detail.data.rpAppId)
| Vendor.machineId := rename(Vendor.detail.data.machineId)
| Vendor.sessionId := rename(Vendor.detail.data.sessionId)
| Vendor.deviceOS := rename(Vendor.detail.data.deviceOS)
| Vendor.serverRelVersion := rename(Vendor.detail.data.serverRelVersion)
| Vendor.origin := rename(Vendor.detail.data.origin)
| Vendor.eventTags := rename(Vendor.detail.data.eventTags)

// Map outer-level fields (if present)
| Vendor.account := rename(Vendor.account)
| Vendor.region := rename(Vendor.region)
| Vendor.dataSource := rename(Vendor.detail.dataSource)
| Vendor.customerUuid := rename(Vendor.detail.customerUuid)
```

```
| Vendor.tenantUuid := rename(Vendor.detail.tenantUuid)
| Vendor.detailEventTags := rename(Vendor.detail.eventTags)
```

## 6. Your screen will now look like this:

Parser script 38

```
38 // #region NORMALIZATION
39 /*****
40 ***** Normalize additional fields from the JSON payload to CPS fields
41 *****/
42
43 // Core event fields
44 | message := rename(Vendor.detail.data.message)
45 | ip_address := rename(Vendor.detail.data.remoteIP)
46 | user.name := rename(Vendor.detail.data.machineUserName)
47
48 // Map additional vendor-specific fields for further context
49 | Vendor.eventID := rename(Vendor.detail.data.id)
50 | Vendor.dataVersion := rename(Vendor.detail.data.version)
51 | Vendor.eventDataType := rename(Vendor.detail.data.type)
52 | Vendor.subName := rename(Vendor.detail.data.subName)
53 | Vendor.loggedBy := rename(Vendor.detail.data.eventLoggedBy)
54 | Vendor.loggedTime := rename(Vendor.detail.data.loggedTimeInUTC)
55 | Vendor.tenantId := rename(Vendor.detail.data.tenantId)
56 | Vendor.userAgent := rename(Vendor.detail.data.userAgent)
57 | Vendor.traceId := rename(Vendor.detail.data.traceId)
58 | Vendor.deviceType := rename(Vendor.detail.data.deviceType)
59 | Vendor.rpAppId := rename(Vendor.detail.data.rpAppId)
60 | Vendor.machineId := rename(Vendor.detail.data.machineId)
61 | Vendor.sessionId := rename(Vendor.detail.data.sessionId)
62 | Vendor.deviceOS := rename(Vendor.detail.data.deviceOS)
63 | Vendor.serverRelVersion := rename(Vendor.detail.data.serverRelVersion)
64 | Vendor.origin := rename(Vendor.detail.data.origin)
65 | Vendor.eventTags := rename(Vendor.detail.data.eventTags)
66
67 // Map outer-level fields (if present)
68 | Vendor.account := rename(Vendor.account)
69 | Vendor.region := rename(Vendor.region)
70 | Vendor.dataSource := rename(Vendor.detail.dataSource)
71 | Vendor.customerUuid := rename(Vendor.detail.customerUuid)
72 | Vendor.tenantUuid := rename(Vendor.detail.tenantUuid)
73 | Vendor.detailEventTags := rename(Vendor.detail.eventTags)
74
```

Fields

to Cps.version Vendor ecs.version event.dataset event.kind event.module event.outcome observer.type

tag:

Test data Failed: 3

2018-10-15T12:51:40+00:00 [INFO] This is an example log entry. id=123 fruit=banana

! Failed to validate event against data schema. Please see the "Schema violations" tab for more information.

2018-10-15T12:52:42+01:30 [ERROR] Here is an error log entry. class=c.o.StringUtil fruit=pineapple

! Failed to validate event against data schema. Please see the "Schema violations" tab for more information.

2018-10-15T12:53:12+01:00 [INFO] User logged in. user\_id=1831923 protocol=http

! Failed to validate event against data schema. Please see the "Schema violations" tab for more information.

Cancel Save and exit

## 7. Click the **Save and exit** button

Your parser has now been saved for use during setup of the data connector in the next section.

### Configure the Crowdstrike data connector

1. In the Falcon console, go to Next-Gen SIEM > Log Management > Data onboarding > Data connections. Click on the **+Add connection** button

**Status of connections**

- Active: 0
- Idle: 1
- Error: 0
- Disconnected: 0
- Pending: 1
- Paused: 0
- Total connections: 2

**Data ingest**

Next-Gen SIEM | Other

- Avg. ingest per day (30 day moving average): 22.22 KB / 10 GB
- Avg. ingest per day limit: 10 GB
- Daily ingest (Since 00:00 UTC): 10.86 kB today

**Connections 2 items**

Search | Status | Vendor | Product | Connector type | Parser | Subscription | Clear all

**+ Add connection**

Status	Connection name	Vendor	Product	Connector type	Parser	Subscription	Actions
Pending	test	Generic	HEC	Push	json-for-action(Ge...	Next-Gen SIEM	

2. In the Data Connectors page, filter or sort by Connector name, Vendor, Product, Connector Type, Author, or Subscription to find and select the HEC/HTTP Event Data Connector.

**Data connectors 1 items**

Filter by connector name: http | Vendor | Product | Connector type | Author | Subscription | Clear all

Connector name	Vendor	Product	Connector type	Author	Subscription
HEC / HTTP Event Connector	Generic	HEC	Push	Crowdstrike	Next-Gen SIEM

3. In the New connection dialog, review connector metadata, version, and description. Click Configure.

The screenshot shows the 'Data connectors' page in the HYPR interface. The breadcrumb trail is 'Data onboarding > New connection'. A search bar is present at the top right. Below the breadcrumb, there's a link to 'Data connections' and a heading 'Data connectors 1 items'. A filter bar allows filtering by connector name (http), vendor, product, connector type, author, and subscription, with a 'Clear all' button. Below the filter bar is a table of connectors. The table has columns for Connector name, Vendor, Product, Connector type, Author, and Subscription. The first row shows 'HEC / HTTP Eve...' as the connector name, 'Generic' as the vendor, 'HEC' as the product, 'Push' as the connector type, 'Crowdstrike' as the author, and 'Next-Gen SIEM' as the subscription. To the right of the table, a detailed view for the 'HEC / HTTP Event Connector' is shown, including a 'Configure' button. The details include: Vendor: Generic, Product: HEC, Connector type: Push, Author: Crowdstrike, Parser name: centrix-iot-json, Subscription: Next-Gen SIEM, Version: v1.0.0, and Description: Ingesting data from any data source that uses the HTTP/HTTPS protocol with...

1 result (1-1 shown) | Items per page 20 | Page 1 of 1

4. In the Add new connector page, enter or select these details:

- Data source: Enter a name for the data source to display on the connection's Details page.
- Connector name: Enter a name to identify the connector. This name displays in the Connections list.
- Description: Optional. Enter a description of the connector.
- Parsers: Select a parser to use for this connection. In the Parsers dropdown menu, search for an existing parser that aligns with the data source. If such a parser does not exist, you need to create a custom parser. To create a custom parser, click Create new parser. For more info, see [Add a new parser](#). For custom parser requirements, see [Understanding the CrowdStrike Parsing Standard](#).

**Add new connector**

To add a connector, provide data and connector details.

[Learn more about data connectors](#)

**Data details**

Data source  
HYPR Event Hook

**Connector details**

Connector name  
My HYPR Tenant

Description (optional)  
events from mytenant.hypr.com

**Parser details**

Parsers  
MyHYPRParser Create new parser

I affirm that any data shared with CrowdStrike using connectors or 3rd party applications will be done so in accordance with the [Terms and Conditions](#).

Cancel Create connection

**Additional resources**

- Learn how to set up the**  
To learn more about this data connector, view our [documentation](#).
- Learn more about data connectors**  
For general information about data connectors, see [data connectors documentation](#).
- Learn more about parsers**  
For more information about parsers, see [parser documentation](#).
- Download Logscale Collector**  
Aggregate logs from different sources using our Logscale Collector. Install it by finding the appropriate Logscale Collector for your device on our [tool downloads list](#).

- Click the Terms and Conditions box, then click the **Create connection**.
- A banner message appears in the Falcon console when your API key and API URL are ready to be generated. Click the Close button.

**Connector setup in progress**

The connector is being configured to receive your data, however you will first need to enter an API key into Generic or an appropriate service in order to begin sending data. The API key will be generated shortly. [Learn more](#)

Close

- You will be returned to the configuration screen for the data connector you just created. At the top of the page, you will see a message that you need to generate an API key.

Click the **Generate API key** button.

The screenshot shows the 'My HYPR Tenant' data connector page. At the top, a message states: 'This connector is ready to receive data. To begin sending data, select the Generate API key button and enter it into Generic or an appropriate service.' A red arrow points to the 'Generate API key' button in the top right corner of this message box. Below the message, the page is divided into sections: 'Data connections', 'My HYPR Tenant' (with a 'Pending' status and a link to 'Learn more about data connectors'), 'Data details' (showing 'Data source: HYPR Event Hook', 'Last ingested (UTC): --', and 'Total ingested amount in last 24 Hours: 0 B'), 'API authorization' (showing 'API URL: https://6cb97810a8174f48913b128eca269752.ingest.us-2.crowdstrike.com/services/collector'), 'Connector details' (showing 'Connector name: My HYPR Tenant', 'Connection ID: 6cb97810a8174f48913b128eca269752', and 'Description: events from mytenant.hypr.com'), and 'Parser details' (showing 'Parsers: MyHYPRParser'). On the right side, there is an 'Additional resources' section with links to 'Learn how to set up the HYPR Event Hook', 'Learn more about data connectors', 'Learn more about parsers', and 'Download Logscale Collector'.

- You will see a dialog box with the API key and API URL. Copy and safely store the API key and API URL to use during connector configuration. Click the **Close** button.

The screenshot shows a 'Connection setup' dialog box. It contains the following text: 'The system is ready to receive your data, however you will first need to enter this information into HEC / HTTP Event Connector or an appropriate service in order to begin sending data. This API key will only be shown once.' Below this is a link to 'Learn more'. The 'API key' is displayed as 'c77' followed by a masked section and '79'. The 'API URL' is 'https://6cb97810a8174f48913b128eca269752.ingest.us-2.crowdstrike.com/services/collector'. There are copy icons next to both the API key and the API URL. A 'Close' button is located at the bottom of the dialog box.

Return to the Next-Gen SIEM > Log Management > Data onboarding > Data connections screen to view your new data connector.

Connections 4 items

Search  Status  Vendor  Product  Connector type  Parser  Subscription  Clear all

Status	Connection name	Vendor	Product	Connector type	Parser	Subscription	Actions
Pending	<a href="#">My HYPR Tenant</a>	Generic	HEC	Push	<a href="#">MyHYPRParser</a>	Next-Gen SIEM	<input type="button" value="v"/>

The status will remain "Pending" until HYPR has been configured to send data to this connector.

This completes the CrowdStrike data connection configuration.

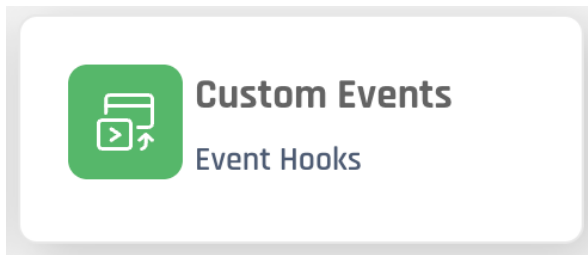
### Configuring Event Hook in HYPR

Use the AP key and API URL from the previous section to configure the JSON for the HYPR event hook.

Follow the procedure for configuring a custom event hook in the [Event Hook: Custom](#) page in the HYPR documentation. Copy the below JSON data into a text editor and add your path, port and token.

```
{
  "name": "Crowdstrike Event Hook",
  "eventType": "ALL",
  "invocationEndpoint": "<API URL>/raw",
  "httpMethod": "POST",
  "authType": "API_KEY",
  "authParams": {
    "apiKeyAuthParameters": {
      "apiKeyName": "Authorization",
      "apiKeyValue": "Bearer XXXXX"
    },
    "invocationHttpParameters": {
      "headerParameters": [
        {
          "key": "Content-Type",
          "value": "application/json",
          "isValueSecret": false
        }
      ]
    }
  }
}
```

In the HYPR Control Center, navigate to Integrations. Click the **Add new integration** button and choose Custom Events.



Copy the JSON into the dialog box and click the **Add Event Hook** button.

## Add New Event Hook



```
{
  "eventType": "ALL",
  "invocationEndpoint": "https://f31fabf97c49464986fc983c41301d9e.ingest.us-2.crowdstrike.com/services/collector/raw",
  "httpMethod": "POST",
  "authType": "API_KEY",
  "authParams": {
    "apiKeyAuthParameters": {
      "apiKeyName": "CSAPIKEY"
    },
    "invocationHttpParameters": {
      "headerParameters": [
```

Cancel

Add Event Hook

At this point new HYPR events will be sent to CrowdStrike.

## Sending events to Cribl

HYPR can be configured to send events to Cribl using HYPR's custom event hook ([Event Hook: Custom](#)) functionality. In this case, Cribl is configured to receive data over HTTP/S and HYPR is configured to send event data to the Cribl endpoint.

see <https://docs.cribl.io/stream/sources-raw-http/>  
see free trial at <https://cribl.io/try-cribl/>

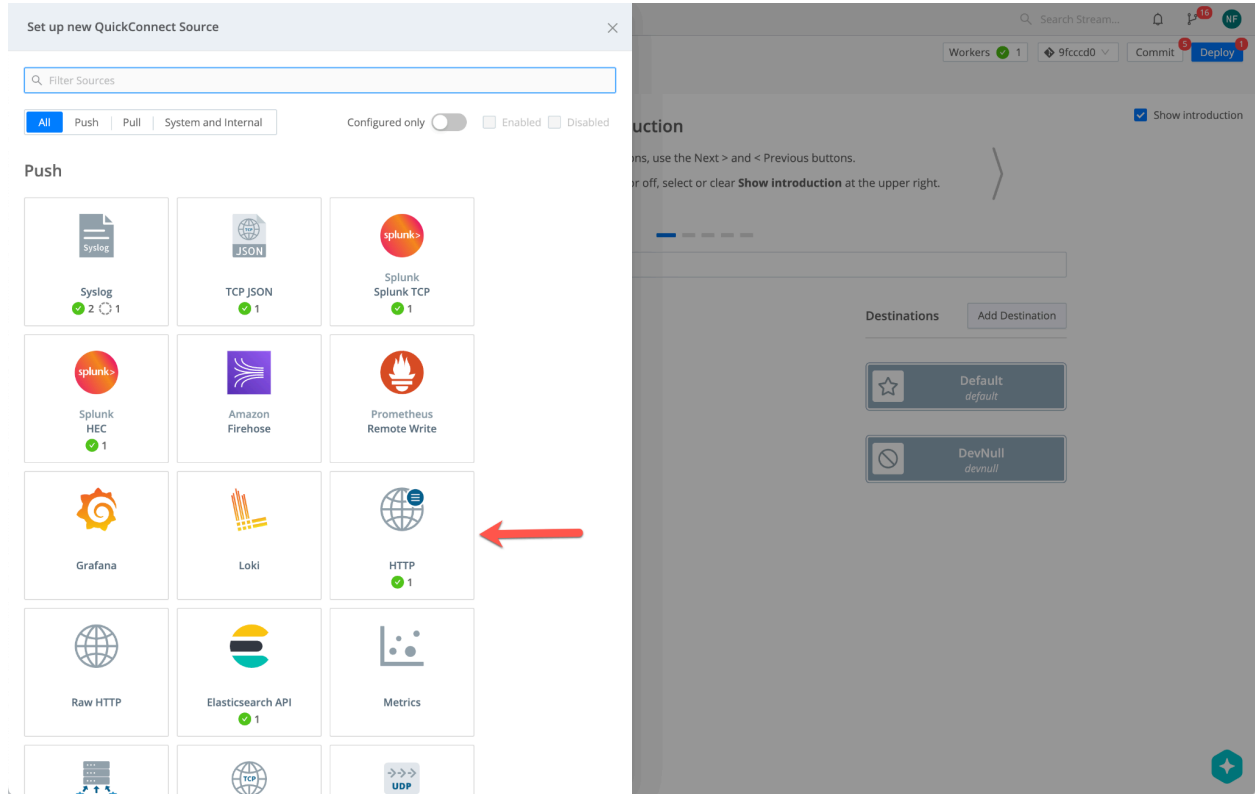
## Setting up Cribl

Before HYPR can be configured, you must first set up Cribl to receive data over HTTP/S. Cribl provides documentation for this procedure in their [product documentation](#). This section provides some example screenshots for configuring Cribl.

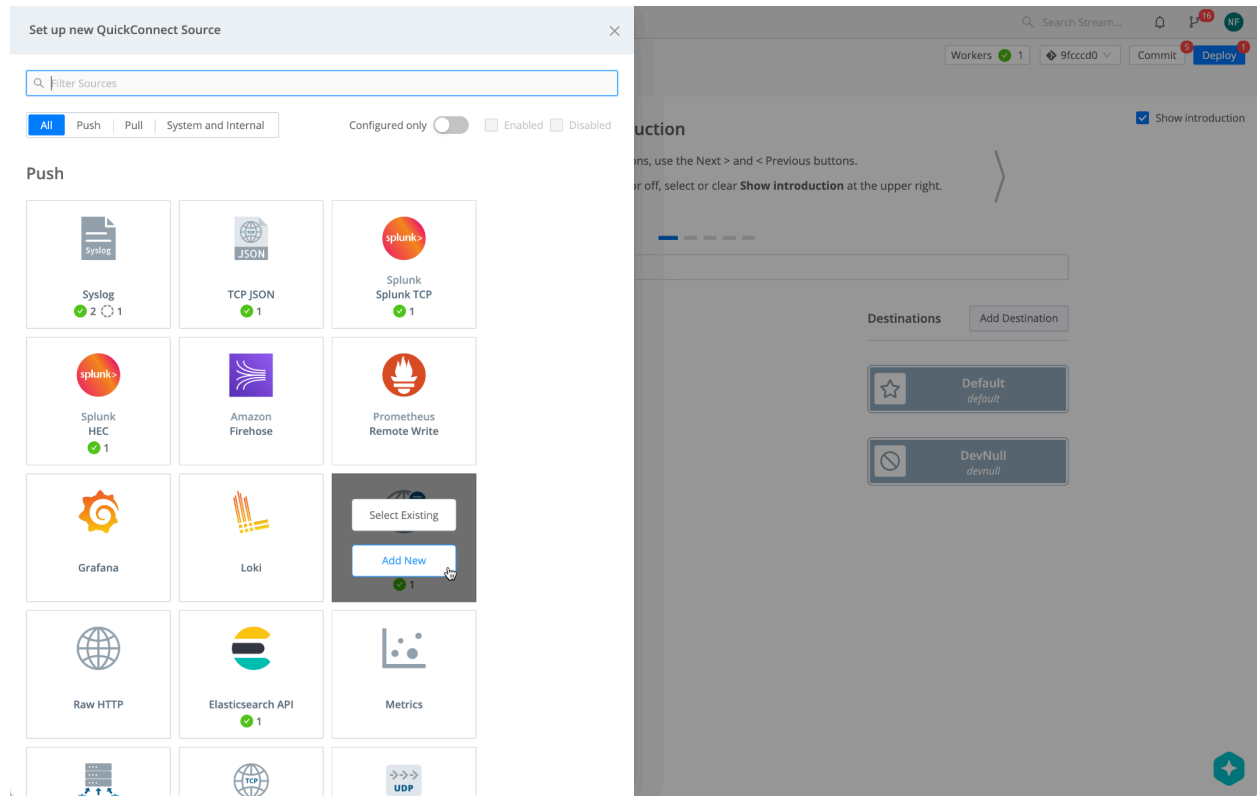
Login to your Cribl tenant and click Worker Groups in the left-hand menu and then click on your desired worker group (you may only have one called "default"). To configure via [QuickConnect](#), navigate to **Routing > QuickConnect** (Stream).

The screenshot displays the Cribl QuickConnect interface. The top navigation bar includes 'Stream', 'Products', 'HYPR', and 'main'. The main content area is titled 'QuickConnect introduction' and contains instructions for navigating through the setup process. Below the introduction, there is a 'Sources' section with an 'Add Source' button highlighted by a red arrow. To the right, there is a 'Destinations' section with two buttons: 'Default' and 'DevNull'. The interface also features a search bar, a 'Workers' count, and 'Commit' and 'Deploy' buttons.

Click **Add Source** and you will be presented with a list of available sources.



Locate the HTTP source, click it and select Add New



On the General Settings tab, give it a Name, Description and Port. Click on Add Token button to create an access token.

🌐 Source > HTTP  
New HTTP

- General Settings
- TLS Settings (Server Side)
- Processing Settings ^
- Fields
- Pre-Processing
- Advanced Settings

**Input ID\*** ?

`__inputId.startsWith('http:HYPREventHook:')` 📄

**Description**

**Address\*** ?

**Port\*** ?

▼ **Authentication**

**Auth tokens** ?

←

▼ **Optional Settings**

**Cribl HTTP event API** ?

**Elasticsearch API endpoint (Bulk API)** ?

**Splunk HEC endpoint** ?

**Enable Splunk HEC acknowledgements**

**Tags** ?

Enabled

Generate a token and record it in a secure location such as a password manager. You can also enter a description of the token.



Source > HTTP  
New HTTP

General Settings

TLS Settings (Server Side)

Processing Settings

Fields

Pre-Processing

Advanced Settings

Input ID\* ?

Enabled

HYPREventHook

\_\_inputId.startsWith('http:HYPREventHook:')

Description

Event hook from HYPR CC

Address\* ?

0.0.0.0

Port\* ?

20001

Authentication

Auth tokens ?

Used by HYPR to auth to Cribl

Clone

Token\* ?

.....



Generate

Description

Used by HYPR to auth to Cribl

Fields ?

Add Field

Add Token

Optional Settings

Cribl HTTP event API ?

/cribl

Elasticsearch API endpoint (Bulk API) ?

/elastic

Splunk HEC endpoint ?

/services/collector

Enable Splunk HEC acknowledgements

Tags ?

Enter tags

Prev

Next

Cancel

Save

Next select the TLS settings tab. You can use the default Cribl server certificate or configure your own. In this example, we use the default Cribl certificate. Enter the following values in the form

Name	Value
Private key path	/opt/criblcerts/criblcloud.key
Certificate path	/opt/criblcerts/criblcloud.crt

Leave the remaining values as default.

Sources > HTTP  
HYPREventHook

Configure Status Charts Live Data Logs Notifications

General Settings

**TLS Settings (Server Side)**

Processing Settings ^

Fields

Pre-Processing

Advanced Settings

Connected Destinations 1

Enabled  Autofill?

Certificate ⓘ  
Select one  Create

Private key path\* ⓘ  
/opt/cribcloud/cribcloud.key

Passphrase ⓘ  
Enter passphrase

Certificate path\* ⓘ  
/opt/cribcloud/cribcloud.crt

CA certificate path ⓘ  
Enter CA certificate path

Authenticate client (mutual auth) ⓘ

Minimum TLS version  
TLSv1.2

Maximum TLS version  
Select one

Delete Source Cancel Save

Next click on the Connected Destinations tab. In this example, we create a “passthu” pipeline to devnull.

Sources > HTTP  
HYPREventHook

Configure Status Charts Live Data Logs Notifications

General Settings

TLS Settings (Server Side)

Processing Settings ^

Fields

Pre-Processing

Advanced Settings

Connected Destinations 1

Send to Routes QuickConnect

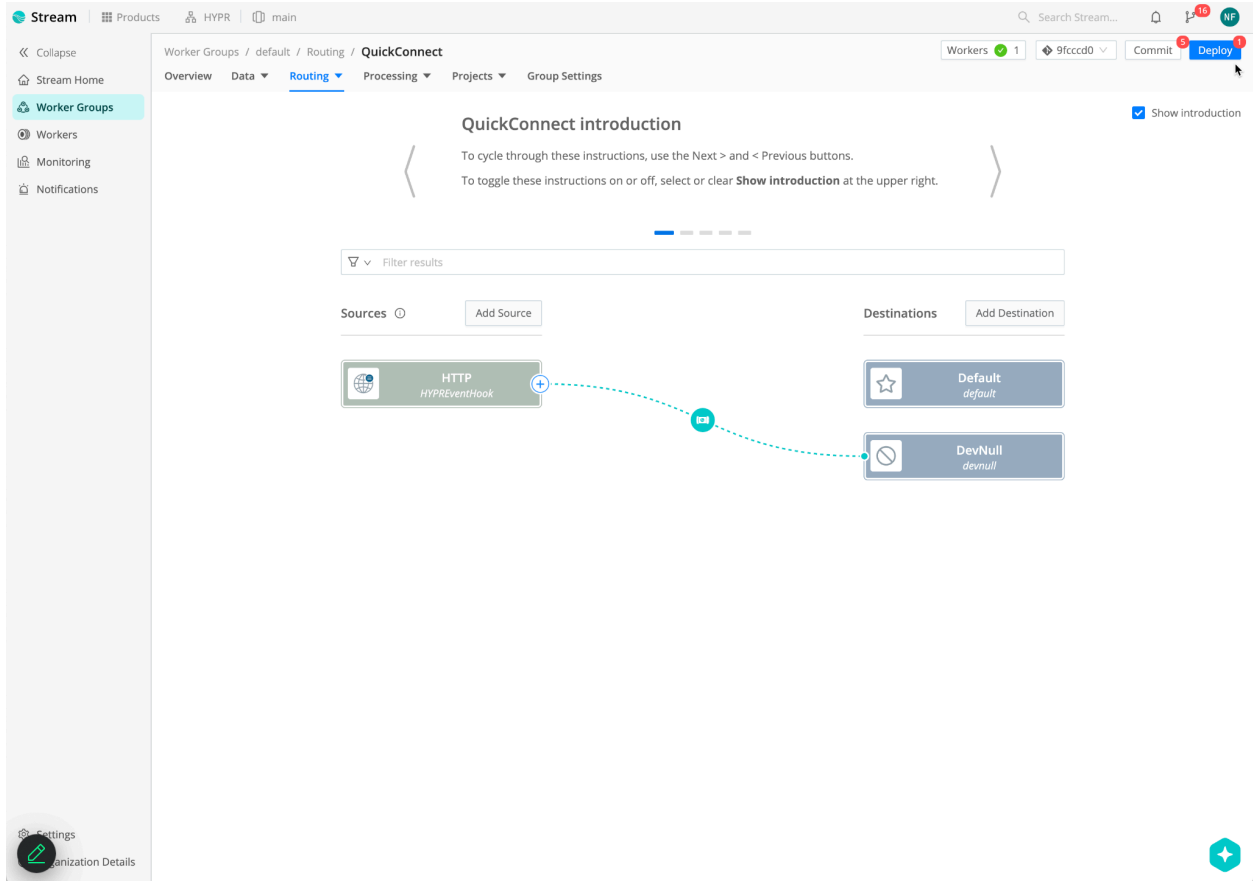
Use QuickConnect

Pipeline or Pack	Destination
passthru	devnull:devnull

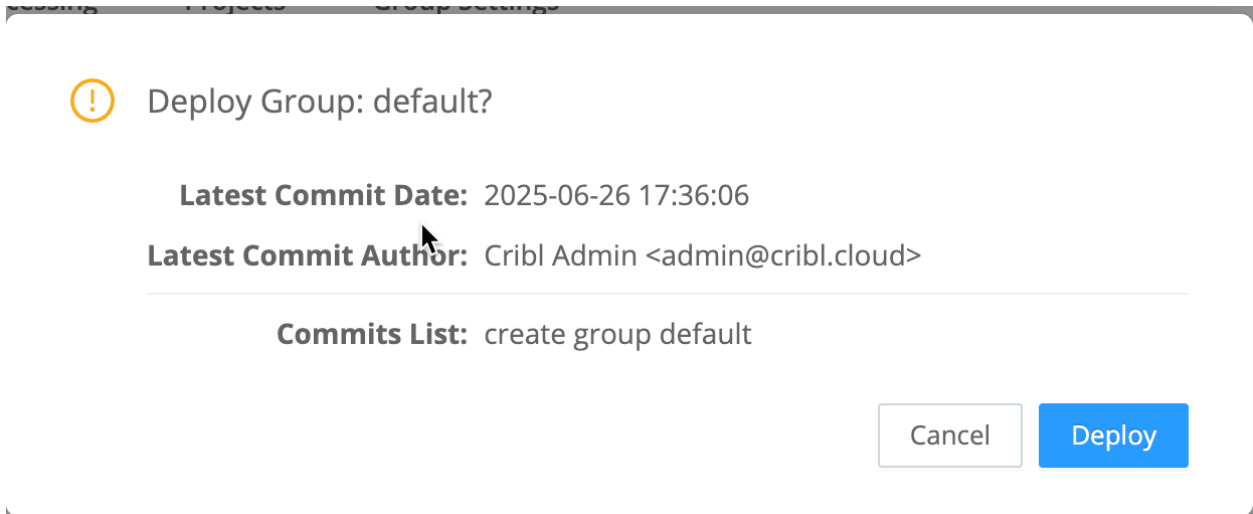
Add Quick Connection

Create Source Cancel Save

Click Save to save the configuration.

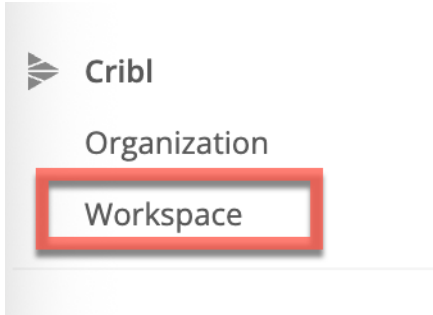


In the upper right corner, click the deploy button to push the changes out.

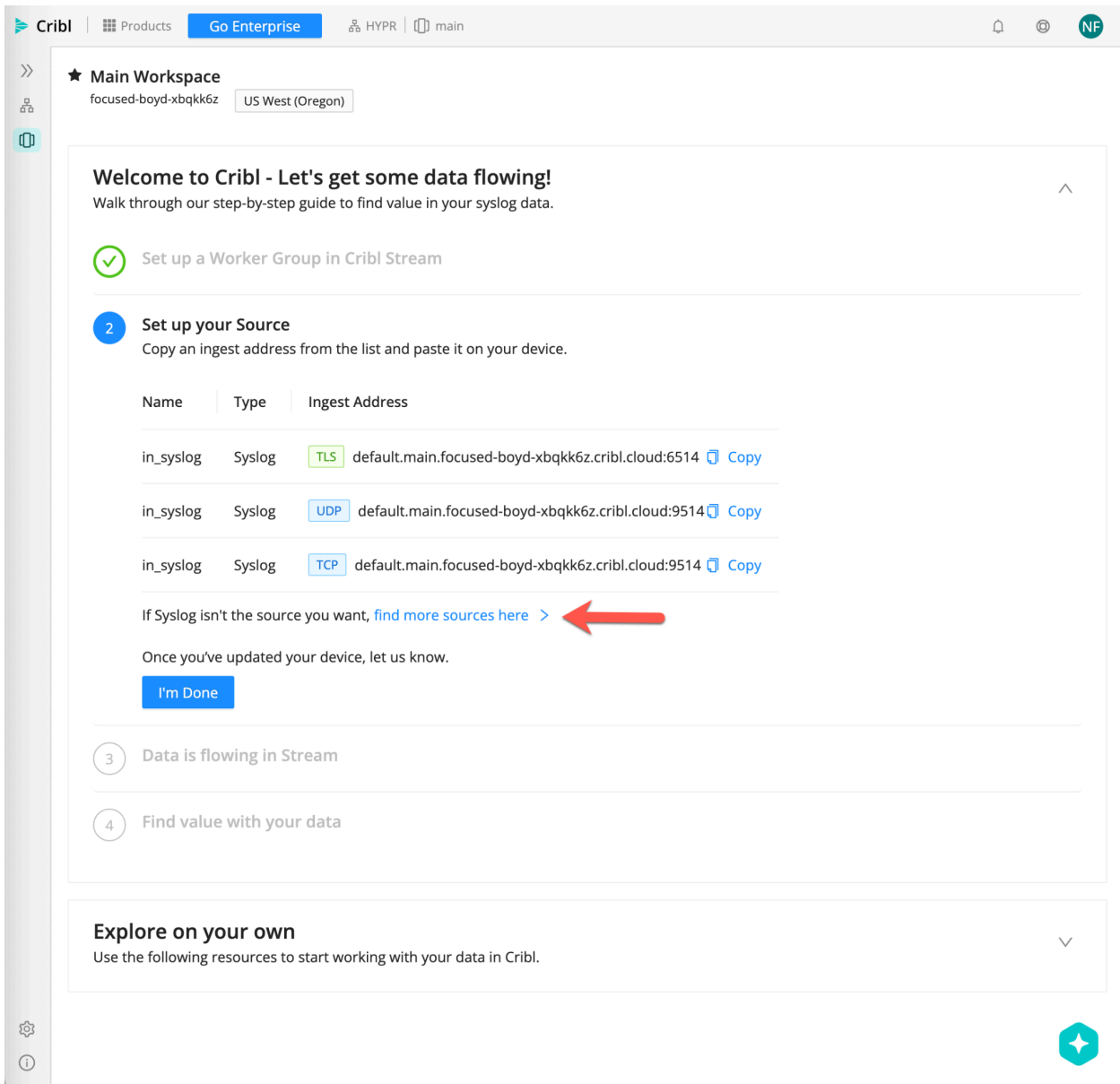


This completes the configuration.

In order to configure HYPR, you will need to get the URL for the Cribl HTTP endpoint. At the top of the screen, click Products and then click Workspace



You will see a screen like this



Click “find more sources here”. Locate the HTTP URL and copy it



Put the URL in your notes and change the port value to the one you defined when creating the HTTP source (found in General Settings). You will need this URL and the access token you created to configure HYPR.

### Configuring Event Hook in HYPR

Follow the procedure for configuring a custom event hook in the [Event Hook: Custom](#) page in the HYPR documentation. Copy the below JSON data into a text editor and add your path, port and token.

```
{
  "name": "HYPR-CRIBL",
  "eventType": "ALL",
  "invocationEndpoint":
  "https://<yourpath>.cribl.cloud:<yourport>/cribl/_bulk",
  "httpMethod": "POST",
  "authType": "API_KEY",
  "authParams": {
    "apiKeyAuthParameters": {
      "apiKeyName": "Authorization",
      "apiKeyValue": "Bearer <yourtoken>"
    },
    "invocationHttpParameters": {
      "headerParameters": [
        {
          "key": "Content-Type",
          "value": "application/json",
          "isValueSecret": false
        }
      ]
    }
  }
}
```

In the HYPR Control Center, navigate to Integrations. Click the **Add new integration** button and choose Custom Events.



## Custom Events

Event Hooks

Copy the JSON into the dialog box and click the **Add Event Hook** button

### Add New Event Hook



```
{
  "name": "HYPR-CRIBL",
  "eventType": "ALL",
  "invocationEndpoint": "https://<yourpath>.cribl.cloud:<yourport>/cribl/_bulk",
  "httpMethod": "POST",
  "authType": "API_KEY",
  "authParams": {
    "apiKeyAuthParameters": {
      "apiKeyName": "Authorization",
      "apiKeyValue": "Bearer <yourtoken>"
    },
    "invocationHttpParameters": {
```

Cancel

Add Event Hook

At this point new HYPR events will be sent to Cribl.