

INTEGRATIONS • FOR IT ADMINS

Get Cloudraw alerts in Slack, Telegram and e-mail

Send security, connector and app firewall alerts to the places your team already watches, and choose which alerts go where.

Last updated 7 October 2026

Overview

Every Cloudraw alert appears in the admin console. You can also send alerts to three outside channels:

- **Slack**, through an incoming webhook. Any chat tool that accepts a Slack-style webhook with a `text` field also works.
- **Telegram**, to a private chat, a group or a channel.
- **E-mail**, to up to 10 addresses.

The setup has two parts:

1. **Connect the channels** you want. Each one takes about 5 minutes.
2. **Choose the routing**: which alert categories go to which channel. Cloudraw starts with sensible defaults, so this step is optional.

Both parts are in **Settings** → **Alerts** in the admin console.

You need

- The *Owner* or *Security admin* role in the Cloudraw admin console.
- For Slack: permission to add apps to your Slack workspace, or a Slack admin who can approve the app.
- For Telegram: the Telegram app, and admin rights in the group or channel that should receive alerts.

How delivery works

- **Only verified destinations receive alerts.** E-mail addresses must confirm by link. A Telegram chat on the Cloudraw bot must be linked with a code. Your own Telegram bot and your Slack webhook are yours, so they work at once.
- **Firing and resolved.** Conditions that last, such as a connector going offline or a certificate close to expiry, send one alert when they start and one **RESOLVED** message when they clear.

- **At most 30 messages per channel per hour.** The 30th message says that further alerts are paused until the next hour. They still appear in the console.
- **Retries.** If a channel does not answer, Cloudraw retries with growing delays, from 5 seconds up to 30 minutes. After 6 failed attempts the message is dropped and the channel shows the last error.
- **Secrets are write-only.** After you save a Slack webhook URL or a Telegram bot token, the console never shows it again. Both are encrypted at rest.

Slack and Telegram messages look like this:

● WARNING – Your workspace
Connector Office-1 is offline.

2026-10-07T09:14:03.512Z

The first line shows the severity: ● CRITICAL, ● WARNING or ● INFO. When the condition clears you get ● RESOLVED.

NOTE

For a SOC or SIEM, use **Settings** → **SIEM** or signed webhooks instead. They receive every event, with no hourly limit.

Alert categories and defaults

Each alert belongs to one category. For each category you choose the channels that receive it. Cloudraw uses these defaults until you change them:

Category	What it covers	Console	E-mail	Slack	Telegram
Security	Person or device quarantined, remote assist started or ended, a person removed by SCIM, a new device added, log shipping to your SIEM failing, the same program blocked on 3 or more devices	On	On	On	On
Connector health	A connector goes offline, or Cloudraw detects that a resource cannot be reached. A RESOLVED message follows when it recovers.	On	Off	Off	On
Devices	Device trust changes and device admission decisions	On	Off	Off	Off

Category	What it covers	Console	E-mail	Slack	Telegram
Access	Just-in-time access requests and approvals, people who signed up and wait for approval, people created by SCIM	On	Off	Off	Off
Billing	Plan and billing notices	On	On	Off	Off
System	Certificates that renew within 14 days, changes to your e-mail sender, and other platform notices	On	Off	Off	Off
App firewall	A program blocked from an app (or, if you choose, one that would have been blocked). See App firewall alerts .	On	Off	On	On

To change the routing, tick or clear the boxes in the table in **Settings** → **Alerts** and save. A channel you have not connected yet receives nothing, even if its box is ticked.

TIP

Start with the defaults. If a channel gets too noisy, turn off one category at a time rather than the whole channel.

Slack

Clouddraw posts to Slack through an *incoming webhook*. A webhook posts to one channel. To use several Slack channels, pick one for Clouddraw alerts.

Step 1. Create a Slack app with an incoming webhook

1 Create the app

Go to api.slack.com/apps (<https://api.slack.com/apps>) and sign in. Click **Create New App**, then **From scratch**.

2 Name and workspace

App Name: . Under **Pick a workspace to develop your app in**, choose your Slack workspace. Click **Create App**.

3 Turn on incoming webhooks

In the left menu, under **Features**, click **Incoming Webhooks**. Switch **Activate Incoming Webhooks** to **On**.

4 Add a webhook

Scroll down and click **Add New Webhook**. Choose the channel that should receive alerts and click **Allow**. If your workspace needs admin approval for apps, Slack sends a request to your Slack admin first.

5 Copy the URL

Back on the **Incoming Webhooks** page, find the new row under **Webhook URL** and click **Copy**. It looks like this:

```
https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXX
```

IMPORTANT

Treat the webhook URL like a password. Anyone who has it can post to your channel. If it leaks, remove the webhook in Slack and add a new one.

For a private channel, you must be a member of it before you can pick it in step 4.

Step 2. Paste the URL into Clouddraw

1 Open Slack settings

In the admin console go to **Settings** → **Alerts** → **Slack**.

2 Paste and save

Paste the URL into **Webhook URL** and click **Save**.

3**Send a test**

Click **Send test**. A test message should appear in the channel within a few seconds. See [Send a test alert](#).

When you save, Cloudraw checks the URL:

- It must start with `https://`.
- It must not contain a user name or password.
- It must not point to a private, loopback or internal address, or to a Cloudraw host.
- It must use port 443 (or 8088, 8443 or 6514). Slack URLs use 443.

Cloudraw does not post anything on save. Use **Send test** to prove the URL works. After saving, the console shows only the host name, `hooks.slack.com`. To change the URL, paste a new one and save. To stop Slack alerts, remove the Slack channel.

NOTE

Cloudraw sends plain text in the `text` field, up to 3,500 characters. Microsoft Teams workflows and Mattermost incoming webhooks that accept this format also work in this field.

Telegram

There are two ways to connect Telegram:

- **The Cloudraw bot.** No bot to create. You link your chat with a one-time code. Available when your console offers it.
- **Your own bot.** You create a bot with BotFather and give Cloudraw its token and the chat id. Messages come from your bot.

Option A. Use the Cloudraw bot

1 Start linking

In **Settings** → **Alerts** → **Telegram**, choose the Cloudraw bot and save. The console shows a code and two links. The code is shown once and is valid for 24 hours.

2 Link a private chat

Open the private-chat link on a device with Telegram and press **Start**. The bot receives `/start <code>`.

3 Or link a group

Open the group link, pick the group and add the bot. If that does not send the code, send this message in the group: `/start@<bot name> <code>`.

4 Check the result

The bot answers "✅ *This chat now receives Cloudraw alerts for <your workspace>.*" The console shows the chat as linked, with its title.

If the code expires, ask for a new one in the same place. You can ask for up to 10 codes per hour.

Option B. Use your own bot

1. Create the bot and copy its token

1 Open BotFather

In Telegram, open a chat with **@BotFather** (<https://t.me/BotFather>). Check for the blue verified mark.

2 Create the bot

Send `/newbot`. Enter a display name, for example `Cloudraw alerts`. Then enter a user name that ends in `bot`, for example `acme_cloudraw_alerts_bot`.

3 Copy the token

BotFather replies with the HTTP API token. It looks like this:

```
123456789:AAExampleExampleExampleExample
```

Keep it secret. Anyone with the token can post as your bot. If it leaks, send `/revoke` to BotFather and save the new token in Cloudraw.

2. Add the bot to the chat

- **Group:** add the bot to the group as a member.
- **Channel:** add the bot as an **administrator** of the channel with the right to **Post messages**. Bots cannot post in a channel as normal members.
- **Private chat:** open the bot and press **Start**. A bot cannot write to someone who never started it.

3. Find the chat id

Cloudraw needs the chat id. It accepts two forms:

- A **number**. Groups and channels have negative numbers, for example `-1001234567890`. Private chats have positive numbers.
- For a **public channel**, its name with `@`, for example `@acme_alerts`.

To find the number of a group or private chat:

1 Send a message

In the group, send a command to the bot, for example `/start@acme_cloudraw_alerts_bot`. By default bots in groups only see commands and messages that mention them.

2 Open getUpdates

In a browser, open this address with your token in place of `<token>`:

```
https://api.telegram.org/bot<token>/getUpdates
```

3 Read the id

Find `"chat":{"id":-100...` in the answer. Copy the number, including the minus sign. If the answer is `{"ok":true,"result":[]}`, send the message again and reload the page.

For a private channel, forward one of its posts to the bot, open `getUpdates` again, and copy the number under `forward_from_chat`.

IMPORTANT

When Telegram upgrades a group to a supergroup, its chat id changes to a new number that starts with `-100`. Save the new id in Cloudraw, or alerts stop.

4. Save in Cloudraw

1

Choose your own bot

In **Settings** → **Alerts** → **Telegram**, choose to use your own bot.

2

Fill in the fields

Use the table below, then click **Save**. The chat is linked at once.

3

Send a test

Click **Send test**. The message should appear in the chat.

Cloudraw field	Value	Cloudraw checks
Bot token	The token from BotFather	Digits, a colon, then the secret part, for example <code>123456789:AAE...</code>
Chat id	The chat id from step 3	A number (negative for groups and channels) or <code>@channelname</code>

The token is write-only. To change only the chat id later, enter the new chat id and leave the token empty. Cloudraw keeps the saved token.

E-mail

1

Add the recipients

In **Settings** → **Alerts** → **E-mail**, enter 1 to 10 addresses and click **Save**. A shared mailbox such as `soc@yourcompany.com` works well.

2

Each recipient confirms

Every new address gets an e-mail titled *"Confirm security alert e-mails"*. The recipient clicks **Review and confirm**, then presses **Confirm** on the page that opens. The link is valid for 7 days.

3

Check the status

Each address shows *pending* until it is confirmed, then *confirmed*. Only confirmed addresses receive alerts.

Addresses that were already confirmed stay confirmed when you edit the list. Removing an address stops its alerts at once.

NOTE

The confirmation page has a **Confirm** button on purpose. Mail security scanners that open links in advance cannot confirm an address on their own.

What is sent

Each alert is a branded Cloudraw e-mail with:

- A subject that starts with the severity: *Critical*, *Warning*, *Notice* or *Resolved*, followed by the alert text.
- The alert text, your workspace name, the severity and the time in UTC.
- A button to open the Cloudraw console.

Alerts come from the Cloudraw sender address. To send them from your own domain, see [Send Cloudraw e-mails from your own domain](#).

Resending confirmations

If a recipient did not get the e-mail, check their spam folder, then use the resend option for pending addresses. To prevent abuse there are limits:

- One confirmation e-mail per address per workspace every 24 hours.
- At most 3 confirmation e-mails per address every 24 hours, across all workspaces.
- At most 30 confirmation e-mails per workspace per day.

Send a test alert

Each connected channel has a **Send test** button in **Settings** → **Alerts**. It sends one real message right away and shows what the channel answered: delivered, or the error.

The test message reads:

● INFO – Your workspace
Test alert – this channel receives Cloudraw alerts.

- **E-mail**: the test goes only to confirmed addresses.
- **Telegram**: the chat must be linked first.
- You can send up to 10 tests per channel per hour.

For automation, the same test is available in the API: `POST /v0/tenants/{tenant}/notification-channels/{channel}/test`, where `{channel}` is `email`, `telegram` or `slack`. It returns `{"delivered": true, "error": null}` or the error.

App firewall alerts

The app firewall decides which programs on a device may open an app. Each app has its own **Alert me** switch, on the app's **Allowed apps** tab:

Alert me	What you get
When something is blocked (default)	An alert each time a program is blocked from this app.
Also when it would be blocked	Also alerts for report-only decisions: programs that would be blocked once you switch the app to Enforce. Useful while you test the allowed list.
Never	No alerts for this app.

These alerts use the **App firewall** category. By default they go to the console, Slack and Telegram, but not e-mail. Change that in the routing table (see [Alert categories and defaults](#)).

To keep the noise down:

- You get at most one alert per device, app and program per hour.
- After 30 app firewall alerts in one hour, Clouddraw sends one summary, *"Many app-firewall alerts this hour..."*, and then stays quiet until the next hour.
- If the same program is refused on 3 or more devices within an hour, Clouddraw sends a **critical** alert in the **Security** category. It may be a tool spreading in your network.

NOTE

Your SIEM receives every app firewall decision, with no hourly limit, whatever the **Alert me** switch says.

Troubleshooting

When you save a channel

Message	Cause and fix
<i>"https:// required"</i>	The Slack URL does not start with <code>https://</code> . Copy it again from Slack.
<i>"not a valid URL"</i>	The pasted text is not a full URL. Copy the whole Webhook URL from Slack.

Message	Cause and fix
<i>"credentials in the URL are not allowed – use the token fields"</i>	The URL contains <code>user:password@</code> . Use the plain webhook URL.
<i>"private, loopback and link-local addresses are not allowed" or "internal host names are not allowed"</i>	The URL points to an internal address or host. Clouddraw only posts to public hosts such as <code>hooks.slack.com</code> .
<i>"port ... not allowed (allowed: 443, 8088, 8443, 6514)"</i>	The URL uses another port. Slack webhooks use 443.
<i>"webhook_url required"</i>	The Slack field was empty when you saved.
<i>"bot_token format is <digits>:<secret>"</i>	The token is incomplete or has extra characters. Copy it again from BotFather.
<i>"chat_id: numeric id (groups are negative) or @channelname"</i>	The chat id is wrong. Use the number from <code>getUpdates</code> , with its minus sign, or <code>@channelname</code> for a public channel.
<i>"the Clouddraw bot is not available yet – use your own bot (chat_id + bot_token)"</i>	The Clouddraw bot is not offered on your workspace yet. Use Option B .
<i>"with the Clouddraw bot the chat is linked by code, not by chat_id"</i>	You entered a chat id but no bot token. Either add your bot's token, or clear the chat id and link with a code.
<i>"recipients: 1-10 addresses"</i>	The e-mail list is empty or has more than 10 addresses.
<i>"not an email address: ..."</i>	One of the addresses has a typo. The message lists it.
<i>"Secrets cannot be stored safely right now (encryption key missing on this server). Retry later; Clouddraw operations has been alerted."</i>	A temporary problem on the Clouddraw side. Nothing was saved. Try again later.

When you send a test, or in the channel's delivery status

Message	Cause and fix
<i>"configure slack first" (or telegram, email)</i>	The channel is not saved yet. Save it, then test.
<i>"no confirmed email address yet — recipients must click their confirmation link"</i>	No recipient has confirmed. Ask them to open the confirmation e-mail and press Confirm .
<i>"the chat is not linked yet — send the /start code to the bot"</i>	With the Cloudraw bot, the chat has not sent the code yet. Send <code>/start <code></code> , or ask for a new code if it expired.
<i>"the platform email relay is not configured yet"</i>	E-mail delivery is not available on the Cloudraw side yet. Contact support.
<i>"too many test messages this hour"</i>	More than 10 tests on this channel this hour. Wait for the next hour.
<i>"hooks.slack.com: HTTP 404", "HTTP 403" or "HTTP 410"</i>	Slack refused the webhook. It was removed, the app was uninstalled, or the channel was archived. Create a new webhook and save the new URL.
<i>"telegram: Unauthorized"</i>	The bot token is wrong or was revoked. Save the current token from BotFather.
<i>"telegram: Bad Request: chat not found"</i>	Wrong chat id, or the bot is not in the chat. Add the bot and check the id.
<i>"telegram: Forbidden: bot was kicked from the group chat" or "bot is not a member of the channel chat"</i>	Add the bot back. In a channel, make it an administrator with Post messages .
<i>"telegram: Bad Request: group chat was upgraded to a supergroup chat"</i>	The group got a new chat id. Find it with <code>getUpdates</code> and save it.
<i>"telegram: Forbidden: bot can't initiate conversation with a user"</i>	For a private chat, open the bot and press Start first.

Other symptoms

- **Some alerts never reach a channel.** Check the routing table. Only categories ticked for that channel are sent.
- **Alerts stop for the rest of the hour.** The channel reached 30 messages this hour. The alerts are still in the console.
- **A confirmation e-mail never arrives.** Check spam and quarantine. The address may also have hit the limit of confirmation e-mails for 24 hours. Try again the next day.
- **A recipient sees "Link expired".** The link is older than 7 days. Resend the confirmation.