

INTEGRATIONS • FOR IT ADMINS

Lock a SaaS app to your connector's IP (egress)

Send a SaaS app's traffic out through your connector, so the app sees one fixed IP, then allow only that IP at the provider.

Last updated 7 October 2026

Overview

Many SaaS apps can refuse sign-ins that do not come from an IP address you trust. With Clouddraw you can make that IP the public IP of your **connector**. People with access reach the app from anywhere, and the app always sees the connector's IP. Everyone else, and every stolen password used from somewhere else, is refused by the app itself.

The setup has three steps:

1. Publish the app in Clouddraw as an **egress** resource, and give people access.
2. Find the connector's public IP in Clouddraw.
3. Allow only that IP in the SaaS admin settings.

You need

- A connector on a machine with a **fixed public IP**. If the IP changes, people are locked out. See [Install a Clouddraw connector](#).
- The connector installed with the current Windows or Linux command, so that automatic updates are on. The connector reports its public IP through its update checks.
- Clouddraw Connect on the devices of the people who use the app.
- Admin rights in the SaaS app.

How egress works

An egress resource is a list of the app's **real hostnames**, for example `login.salesforce.com`, hosted by a connector.

1. A person who has access opens the app as usual, by its real name.
2. Clouddraw Connect on their device catches the connections to those names and sends them through the Clouddraw network to your connector.
3. The connector looks up the real name in public DNS and connects to the app.
4. The app sees the connector's public IP.

The browser still talks to the real site with its real certificate. Cloudraw does not decrypt the traffic. Names that are not in the list go out from the device as usual.

NOTE

An egress resource is **deny by default**, like any application. Until you add an access rule, nobody's traffic goes through the connector, and the allow-list at the provider blocks everyone.

Step 1. Publish the SaaS as egress

- 1 New application**
In the admin console go to **Applications** → **New application** and choose **Website / SaaS via connector IP**.
- 2 Hostnames**
Add every hostname the app uses for sign-in and for the pages you want to protect. You can use a wildcard such as `*.example.com`. The provider sections below list a starting set.
- 3 Ports**
Leave the default, **443** and **80**, unless the app needs another TCP port. For example, add **22** for Git over SSH.
- 4 Hosted by**
Pick the connector whose IP the app will see. You can pick several. Then the app may see any of their IPs, so you must allow all of them.
- 5 Publish and add an access rule**
Give the people or groups who use the app access to it.

The application page then says *"People reach <hostname> by its real name; the site sees your connector's IP..."* and the app carries a **via connector IP** badge.

You can change the hostnames and ports later. Changes take effect in place. To rename the app, publish it again under the new name.

Step 2. Find the connector's public IP

- 1 Open Connectors**
In the admin console open **Connectors**. Switch to **Expert** view if the **Public IP** column is not shown.
- 2 Copy the IP**
Copy the address in the **Public IP** column. Under it you see when it was last seen.

The connector reports this IP every 15 minutes. If the column says *not reported*, wait 15 minutes after installing. Connectors without automatic updates never report it. Then ask your network team for the public IP the machine uses to reach the internet.

IMPORTANT

Clouddraw sees the IP of the connector's update checks. If your network sends web traffic through a proxy or a different NAT address, the SaaS may see another IP. Check this with your network team, or confirm it in the provider's sign-in log after your first test (see [Troubleshooting](#)).

At the provider, enter one IP as a single address or as `/32`, for example `203.0.113.10/32`. The examples below use `203.0.113.10`. Use your own IP.

Step 3. Allow the IP at the provider

WARNING

An IP allow-list can lock you out. Before you turn it on, keep one admin account that is not restricted (a break-glass account), test with one user first, and keep a browser session open as admin until the test works.

IMPORTANT

If people sign in to Cloudraw Connect with the same provider (Microsoft, Google or Okta), never restrict the *Cloudraw* app itself. People sign in to Cloudraw **before** their traffic goes through the connector. If the provider blocks that sign-in, nobody can connect.

Provider menus change from time to time. The paths below were checked in October 2026.

Microsoft 365 (Entra Conditional Access)

Needs Microsoft Entra ID P1 or higher.

Hostnames to publish (sign-in, then the apps you protect): `login.microsoftonline.com`, `login.microsoft.com`, and for example `outlook.office.com`, `outlook.office365.com`, `*.sharepoint.com`. Conditional Access checks the IP when the person signs in, so the sign-in names matter most.

1 Create a named location

In the [Microsoft Entra admin center](https://entra.microsoft.com) (<https://entra.microsoft.com>) go to **Entra ID** → **Conditional Access** → **Named locations** (older menus: **Protection** → **Conditional Access**). Click **+ IP ranges location**, name it `Cloudraw connector`, add `203.0.113.10/32`, tick **Mark as trusted location** and click **Create**.

2 Create a policy

Go to **Conditional Access** → **Policies** → **+ New policy**. Name it `Block outside Cloudraw`.

3 Users

Include a test user or group. Exclude your break-glass account.

4 Target resources

Include **Office 365** (or the apps you protect). If people sign in to Cloudraw with Microsoft, do not include **All resources** unless you exclude the *Cloudraw* app registration.

5 Network

Under **Conditions** → **Locations** (newer menus: **Network**) set **Configure** to **Yes**. Include **Any location** (Any network). Exclude the `Cloudraw connector` location.

6

Grant

Choose **Block access**.

7

Test, then turn on

Start with **Report-only**. Check the sign-in logs, then switch the policy to **On**.

Google Workspace (Context-Aware Access)

Context-Aware Access needs Enterprise Standard or Plus, Education Standard or Plus, or Cloud Identity Premium. Other editions cannot restrict by IP.

Hostnames to publish: the apps you protect, for example `mail.google.com`, `drive.google.com`, `docs.google.com`, `calendar.google.com`, and the sign-in page `accounts.google.com`. Do not use `*.google.com` unless you want all Google traffic, search included, to go through the connector.

1

Create an access level

In the [Google Admin console](https://admin.google.com) (<https://admin.google.com>) go to **Security** → **Access and data control** → **Context-Aware Access** → **Access levels**. Click **Create access level**, name it `Cloudraw connector`, add the condition **IP subnet** = `203.0.113.10/32`, and save.

2

Assign it

Go to **Context-Aware Access** → **Assign access levels**. Pick a test organizational unit or group, select the apps (Gmail, Drive and Docs, Calendar...), click **Assign** and choose `Cloudraw connector`.

3

Monitor, then enforce

Use **Monitor mode** first if your console offers it. Check the Context-Aware Access log, then switch to **Active mode**.

Context-Aware Access applies to Google's apps. It does not block the Google sign-in page itself, so it does not affect signing in to Cloudraw with Google.

Salesforce (Login IP ranges)

Hostnames to publish: `login.salesforce.com`, `yourcompany.my.salesforce.com`, `yourcompany.lightning.force.com`, plus `*.force.com` and `*.salesforce.com` for the other Salesforce pages (Visualforce, files).

1

Profile Login IP Ranges (blocks)

In **Setup**, type **Profiles** in Quick Find, open the profile your users have, find **Login IP Ranges** and click **New**. Enter `203.0.113.10` as both **Start IP Address** and **End IP Address** and save. Users of that profile can no longer sign in from any other IP.

2 Every request, not only sign-in (optional)

In **Setup** → **Session Settings**, tick **Enforce login IP ranges on every request**. Then every page must come through the connector, so publish all the Salesforce hostnames you use.

NOTE

Setup → **Network Access** (Trusted IP Ranges) does **not** block anyone. It only skips the e-mail verification code for sign-ins from that IP. Use Profile Login IP Ranges to restrict access.

GitHub (IP allow list)

Needs GitHub Enterprise Cloud. The list protects your organization's private resources on the web, in the API and over Git.

Hostnames to publish: `github.com`, `api.github.com`, `codeload.github.com`, `uploads.github.com`, `*.githubusercontent.com`. **Ports:** 443 and 80, plus **22** for Git over SSH.

1 Add the IP

For an organization: open the organization, click **Settings** → **Authentication security**. For an enterprise: **Your enterprises** → **Settings** → **Authentication security**. Under **IP allow list**, enter `203.0.113.10`, a description such as `Cloudraw connector`, and click **Add**.

2 Turn it on

Tick **Enable IP allow list** and click **Save**.

NOTE

GitHub-hosted Actions runners, webhooks and GitHub Apps do not come from your connector. If they need your private repositories, add their addresses too, or tick **Enable IP allow list configuration for installed GitHub Apps**.

AWS (IAM policy with `aws:SourceIp`)

Hostnames to publish: `signin.aws.amazon.com`, `console.aws.amazon.com`, `*.console.aws.amazon.com`, `*.signin.aws.amazon.com`, and for the AWS CLI and SDKs `*.amazonaws.com`. If you use IAM Identity Center, add your portal, for example `yourcompany.awsapps.com`.

1 Create the policy

In the IAM console go to **Policies** → **Create policy** → **JSON**, paste the policy below with your IP, click **Next**, name it `DenyOutsideCloudraw` and click **Create policy**.

2

Attach it

Go to **User groups**, open the group, then **Permissions** → **Add permissions** → **Attach policies**. For a whole AWS organization, use the same statement as a service control policy in **AWS Organizations** → **Policies** → **Service control policies**.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DenyOutsideCloudrawConnector",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "NotIpAddress": { "aws:SourceIp": ["203.0.113.10/32"] },
      "Bool": { "aws:ViaAWSService": "false" }
    }
  }]
}
```

The `aws:ViaAWSService` line keeps AWS services working when they call other services for you. Requests through a VPC endpoint carry no public source IP, so this policy denies them. Do not attach it to roles that your workloads use.

Okta (network zones)

Hostnames to publish: your Okta org, for example `yourcompany.okta.com` and `yourcompany-admin.okta.com`, or your custom Okta domain, plus the apps you protect.

1

Create a network zone

In the Okta Admin Console go to **Security** → **Networks**, click **Add zone** → **IP Zone**, name it `Cloudraw connector`, enter `203.0.113.10` under **Gateway IPs** and save.

2

Add a rule to the app's policy

Go to **Security** → **Authentication Policies**, open the policy of the app you protect and click **Add rule**. Set **User's IP is** to **Not in any of the following zones** with `Cloudraw connector`, set **Access is** to **Denied**, save, and move the rule to the top.

Put the rule on the apps you protect, not on the Global Session Policy, if people sign in to Cloudraw with Okta. A global rule would also block the Cloudraw sign-in.

Any other app

- 1 Find the setting**

Look in the app's admin or security settings for *IP allow list*, *trusted IPs*, *IP restrictions*, *network restrictions* or *login IP ranges*.
- 2 Find all hostnames**

Sign in once with your browser's developer tools open (**F12**, **Network** tab). Note the hostnames for the sign-in page, the app and its API. Publish those, using a wildcard where there are many.
- 3 Add the IP and test**

Add the connector's IP as a single address or **/32**. Test with one user, then check the app's sign-in or audit log shows the connector's IP.

Limits

Clouddraw checks the egress resource when you publish or change it. It refuses:

Rule	Example of what is refused
1 to 20 hostnames per egress resource. For more, create a second egress resource.	21 hostnames
Only plain hostnames: letters, digits, hyphens and dots. No <code>https://</code> , paths, ports or IP addresses.	<code>https://crm.example.com/login</code> , <code>203.0.113.5</code>
A wildcard only as <code>*.</code> at the start, and scoped to a domain.	<code>*.com</code> , <code>crm.*.com</code>
No Clouddraw names.	<code>app.clouddraw.com</code> , names ending in <code>.clouddraw</code>
Every name must resolve in public DNS. For a wildcard, the domain itself must resolve.	A name that does not exist yet
No private hostnames. A name that resolves to a private, loopback or link-local address is refused. Publish it as an application or a private network instead.	A name that resolves to <code>10.0.0.5</code> or <code>192.168.1.20</code>
1 to 10 ports, each 1 to 65535. TCP only. Default: 443 and 80.	Port <code>0</code> , port <code>70000</code> , 11 ports
1 to 5 connectors under Hosted by .	6 connectors

The error message names the hostname or port that was refused. Fix that entry and publish again.

TIP

If Clouddraw refuses a wildcard because its domain does not resolve, list the exact hostnames instead.

Troubleshooting

Message or symptom	Cause and fix
The provider blocks a user who should have access	Check that Cloudraw Connect is connected on their device and that an access rule gives them the egress resource. Then check the provider's sign-in log for the IP it saw.
The sign-in log shows the user's home or office IP	The page they used is on a hostname that is not in your list. Find it (see Any other app) and add it to the egress resource.
The sign-in log shows an IP you do not recognize, not the connector's	The connector machine leaves through a different NAT address or proxy than Cloudraw saw. Ask your network team for the address, and allow that one.
Blocked sometimes, not always	The resource is hosted by several connectors and only some of their IPs are allowed. Allow every connector's IP.
Everyone was locked out after the connector's IP changed	The machine has no fixed public IP. Sign in with your break-glass account, update the allow-list, and move the connector to a machine with a fixed IP.
Nobody can sign in to Cloudraw Connect after the change	The provider policy also covers the Cloudraw sign-in. Exclude the <i>Cloudraw</i> app from the policy (see Step 3).
Public IP shows <i>not reported</i>	Wait 15 minutes after installing. Connectors without automatic updates (older installs, Docker) do not report it. See Updates .
<i>"... resolves to a private address ... publish it as an application or private network instead"</i>	The name is internal. Egress is for public sites. Publish it as an application.
<i>"a wildcard must be scoped to a domain"</i>	Use <code>*.crm.example.com</code> or <code>*.example.com</code> , not <code>*.com</code> .
<i>"Cloudraw names cannot be intercepted"</i>	Remove the Cloudraw hostname from the list.
The app works but some parts do not load	Those parts use another hostname or port. Add it, or a wildcard for the domain. Only TCP is carried.

Message or symptom	Cause and fix
The app is unreachable for everyone with access	The connector is offline. With one connector there is no failover. Add a second connector under Hosted by , and allow its IP too.