Parameters

- Required for full Cloudflare Zero Trust features
  - organization
- Required for DNS-only policy enforcement
  - gateway_unique_id
- Organization parameters
  - allow_managed_deployments
  - allow_updates
  - auth_client_id
  - auth_client_secret
  - auto_connect
  - display_name
  - enable_netbt
  - enable_pmtud
  - enable_post_quantum
  - environment
  - external_emergency_signal_fingerprint
  - external_emergency_signal_interval
  - external_emergency_signal_url
  - hardware_backed_registration
  - local_emergency_signal_enabled
  - managed_network_probe_count
  - managed_network_probe_interval_secs
  - mdm_file_debounce_secs
  - mdm_file_unavailable_debounce_secs
  - onboarding
  - override_api_endpoint
  - override_doh_endpoint
  - override_warp_endpoint

- service_mode
- support_url
- switch_locked
- unique_client_id
- use_in_app_webview
- warp_tunnel_protocol
- Top-level parameters
  - configs
  - multi_user
  - organization_configs
  - pre_login
- Per-app VPN parameters (Android)
  - app_identifier
  - is_browser

---

import { GlossaryTooltip, Details, InlineBadge } from "~/components";

Each Cloudflare One Client (formerly WARP) supports the following set of parameters as part of their deployment, regardless of the deployment mechanism.

:::note

Most of the parameters listed below are also configurable in Cloudflare One under **Team & Resources** > **Devices**. In the event of conflicting settings, the Cloudflare One Client will always give precedence to settings on the local device (for example, in your `mdm.xml` or `com.cloudflare.warp.plist` files).

:::

Required for full Cloudflare Zero Trust features

For the majority of Cloudflare Zero Trust features to work, you need to specify a team name. Examples of Cloudflare Zero Trust features which depend on the team name are HTTP policies, Browser Isolation, and device posture.

`organization`

Instructs the client to register the device with your organization. Registration requires authentication via an IdP or Service Auth.

Value Type: `string`

Value: Your team name.

Required for DNS-only policy enforcement

This field is used to enforce DNS policies when deploying the client in DoH-only mode.

`gateway_unique_id`

Instructs the client to direct all DNS queries to a specific Gateway DNS location. This value is only necessary if deploying without a team name or in an organization with multiple DNS locations. If you do not supply a DoH subdomain, we will automatically use the default Gateway DNS location for your organization.

Value Type: `string`

Value: Your DoH subdomain.

Organization parameters

You can use the following parameters to configure a specific Zero Trust organization.

`allow_managed_deployments`

Client modes	Zero Trust plans
All modes	All plans

System	Availability	Minimum client version
Windows		2026.6.0
macOS		2026.6.0
Linux		—
iOS		—
Android		—
ChromeOS		—

Local kill switch for client version assignments pushed from the Cloudflare dashboard. Use this parameter to opt a device out of dashboard-managed version changes regardless of what is configured in the dashboard.

By default, the device applies any assignment the dashboard sends. If no assignment targets the device, the parameter has no effect.

Value Type: `boolean`

Value:

- `true` — (default) The device applies client version assignments configured for it in the Cloudflare dashboard. If no assignment targets the device, this setting has no effect.
- `false` — The device ignores client version assignments and stays on its current installed version, regardless of what is configured in the dashboard.

`allow_updates`

Specifies whether the client should check for updates prior to initial registration.

Value Type: `boolean`

Value:

- `true` — (default) The client automatically checks for updates, prompting the user to install them.
- `false` — The client does not automatically check for updates.

`auth_client_id`

Enrolls the device in your Zero Trust organization using a service token. Requires the `auth_client_secret` parameter.

Value Type: `string`

Value: Client ID of the service token.

Example configuration:

```
<key>auth_client_id</key>
<string>88bf3b6d86161464f6509f7219099e57.access</string>
<key>auth_client_secret</key>
<string>bdd31cbc4dec990953e39163fbbb194c93313ca9f0a6e420346af9d326b1d2a5</string>
```

:::note The service token must have *Service Auth* device enrollment permissions. Allow permissions will not work for service tokens. :::

`auth_client_secret`

Enrolls the device in your Zero Trust organization using a service token. Requires the `auth_client_id` parameter.

Value Type: `string`

Value: Client Secret of the service token.

`auto_connect`

If switch has been turned off by user, the client will automatically turn itself back on after the specified number of minutes. We recommend keeping this set to a very low value — usually just enough time for a user to log in to hotel or airport Wi-Fi. If any value is specified for `auto_connect` the default state of the Cloudflare One Client will always be Connected (for example, after the initial install or a reboot).

Value Type: `integer`

Value:

- `0` — Allow the switch to stay in the off position indefinitely until the user turns it back on.
- `1` to `1440` — Turn switch back on automatically after the specified number of minutes.

:::note

This parameter replaces the old `enabled` property, which can no longer be used in conjunction with the new `switch_locked` and `auto_connect`. If you want to use these parameters, you must remove `enabled`. :::

`display_name`

Identifies a Zero Trust organization in the Cloudflare One Client GUI when the client is deployed with multiple organizations. Required if the `organization` parameter is specified within a `configs` array.

Value Type: `string`

Value: Organization nickname shown to users in the Cloudflare One Client GUI (for example, `Test environment`).

`enable_netbt`

NetBIOS over TCP/IP (NetBT) is a legacy feature in Windows primarily used for name resolution in some rare scenarios. The Cloudflare One Client disables NetBT on the tunnel interface by default for security reasons. If your organization still relies on legacy applications that require NetBT, you can override the default behavior and enable NetBT.

Value Type: `boolean`

Value:

- `false` — (default) Disables NetBT on the Cloudflare One Client tunnel interface.
- `true` — Enables NetBT on the Cloudflare One Client tunnel interface.

`enable_pmtud`

Path MTU Discovery (PMTUD) allows the Cloudflare One Client to discover the largest packet size that can be sent over the current network and optimize connection performance.

Value Type: `boolean`

Value:

- `false` — Disables PMTUD.
- `true` — Enables PMTUD on the Cloudflare One Client tunnel interface. This is the default in Cloudflare One Client version 2026.6.822.0 and later.

In earlier versions, the default is `false`.

`enable_post_quantum`

Client modes	Zero Trust plans
• Traffic and DNS mode • Traffic only mode	All plans

System	Availability	Minimum WARP version
Windows	✅	2025.5.735.1
macOS	✅	2025.5.735.1
Linux	✅	2025.5.735.1
iOS	✅	1.10
Android	✅	2.4
ChromeOS	✅	2.4

The Cloudflare One Client uses post-quantum cryptography to secure connections from the device to Cloudflare's network. Post-quantum cryptography requires the MASQUE protocol and is enabled by default on all devices using MASQUE.

Value Type: `boolean`

Value:

- `false` — Disables post-quantum key agreement.
- `true` — Enables post-quantum key agreement for all traffic through the WARP tunnel.

`environment`

Client modes	Zero Trust plans
All modes	All plans

System	Availability	Minimum WARP version
Windows	✅	2025.9.558.0
macOS	✅	2025.9.558.0
Linux	✅	2025.9.558.0
iOS	✅	1.12.0

System	Availability	Minimum WARP version
Android		2.5.1
ChromeOS		2.5.1

Configures the Cloudflare One Client to connect to Cloudflare's FedRAMP High authorized environment.

Value Type: `string`

Value:

- `normal` — (default) The Cloudflare One Client connects to the standard API endpoints, IPs, and domains (like `<ACCOUNT_ID>.cloudflare-gateway.com`) and forwards traffic to Cloudflare data centers worldwide.
- `fedramp_high` — The Cloudflare One Client connects to FedRAMP-specific API endpoints, IPs, and domains (like `<ACCOUNT_ID>.fed.cloudflare-gateway.com`). Traffic is forwarded to FedRAMP High compliant data centers for processing. To configure the FedRAMP High environment, you must allow the FedRAMP-specific endpoints, IPs, and domains through your firewall.

When using multiple configurations for the same organization, all configurations must specify the same `environment` value. A single organization cannot operate in both the normal and FedRAMP High environments.

In version 2026.5.0 and above, use `organization_configs` to set `environment` once per organization rather than repeating it in every config entry:

```
<dict>
  <key>organization_configs</key>
  <dict>
    <key>mycompany-gov</key>
    <dict>
      <key>environment</key>
      <string>fedramp_high</string>
    </dict>
  </dict>
  <key>configs</key>
  <array>
    <dict>
      <key>organization</key>
      <string>mycompany-gov</string>
```

```
      <key>display_name</key>
      <string>Production</string>
    </dict>
    <dict>
      <key>organization</key>
      <string>mycompany-gov</string>
      <key>display_name</key>
      <string>Staging</string>
    </dict>
    <dict>
      <key>organization</key>
      <string>test-org</string>
      <key>display_name</key>
      <string>Test</string>
      <key>environment</key>
      <string>normal</string>
    </dict>
  </array>
</dict>
```

In earlier versions, each configuration for that organization must include `environment` explicitly:

```
<dict>
  <key>configs</key>
  <array>
    <dict>
      <key>organization</key>
      <string>mycompany-gov</string>
      <key>display_name</key>
      <string>Production</string>
      <key>environment</key>
      <string>fedramp_high</string>
    </dict>
    <dict>
      <key>organization</key>
      <string>mycompany-gov</string>
      <key>display_name</key>
      <string>Staging</string>
      <key>environment</key>
      <string>fedramp_high</string>
```

```
    </dict>
    <dict>
      <key>organization</key>
      <string>test-org</string>
      <key>display_name</key>
      <string>Test</string>
      <key>environment</key>
      <string>normal</string>
    </dict>
  </array>
</dict>
```

`external_emergency_signal_fingerprint`

The SHA-256 fingerprint that the Cloudflare One Client will use to validate the `external_emergency_signal_url` HTTPS endpoint. Refer to External Emergency Disconnect for details on how to extract this fingerprint.

Value Type: `string`

Value: SHA-256 fingerprint of the HTTPS server certificate (for example, `DD4F4806C57A5BBAF1AA5B080F0541DA75DB468D0A1FE731310149500CCD8662`)

`external_emergency_signal_interval`

How often the Cloudflare One Client will poll `external_emergency_signal_url` for an External Emergency Disconnect signal.

Value Type: `integer`

Value: Polling frequency in seconds (minimum `30`, default `300`)

`external_emergency_signal_url`

The HTTPS endpoint that the Cloudflare One Client will poll for an External Emergency Disconnect signal.

Value Type: `string`

Value: `https://192.0.2.1:3333/status/disconnect`

The URL must use `https://` and use an IPv4 or IPv6 address as host (not a domain).

`hardware_backed_registration`

Client modes	Zero Trust plans
All modes	All plans

System	Availability	Minimum WARP version
Windows	□	2026.6.0
macOS	□ (Apple silicon only)	2026.6.0
Linux	□	2026.6.0
iOS	□	N/A
Android	□	N/A
ChromeOS	□	N/A

Binds the registration to a non-exportable key stored in device hardware (Secure Enclave or TPM 2.0) and authenticates API requests with mTLS. For more information, refer to Hardware-backed registration.

Value Type: `boolean`

Value:

- `false` — (default) The Cloudflare One Client stores its API token in the device keystore.
- `true` — The Cloudflare One Client generates a hardware-backed key during registration and uses mTLS for subsequent API calls.

:::caution Changing this value after a device has registered forces the device to register again. Existing registrations are not migrated. :::

`local_emergency_signal_enabled`

Enables Local Emergency Disconnect. When enabled, the Cloudflare One Client monitors a fixed-path JSON file on the device for an emergency disconnect signal.

Value Type: `boolean`

Value:

- `false` — (default) Local signal file monitoring is disabled.
- `true` — The client monitors the local signal file for emergency disconnect.

`managed_network_probe_count`

Client modes	Zero Trust plans
All modes	All plans

System	Availability	Minimum client version
Windows	✅	2026.9.1555.1
macOS	✅	2026.9.1555.1
Linux	✅	2026.9.1555.1
iOS	✅	—
Android	✅	—
ChromeOS	✅	—

Sets how many extra times the Cloudflare One Client checks for managed networks after a check finds none. The client checks for managed networks when the device registers and when the network changes. A temporary connection failure during that check can apply the wrong device profile until the next network change. Extra probes let the client correct the device profile without a network change. Each probe checks every managed network.

To set the wait between probes, use `managed_network_probe_interval_secs`.

Value Type: `integer`

Value:

- `0` — Turns off extra probes. The client only checks for managed networks on registration and network changes, as in versions before 2026.9.1555.1.
- `1` — (default) Runs one extra probe.
- `2` to `10` — Runs up to this number of extra probes. The client treats values above `10` as `10`.
- `-1` — Probes until the client finds a managed network.

`managed_network_probe_interval_secs`

Client modes	Zero Trust plans
All modes	All plans

System	Availability	Minimum client version
Windows		2026.9.1555.1
macOS		2026.9.1555.1
Linux		2026.9.1555.1
iOS		—
Android		—
ChromeOS		—

How long the Cloudflare One Client waits after a managed network check finds no managed network before it probes again. A network change that does not start a new check restarts a pending wait.

To set the number of probes, use `managed_network_probe_count`.

Value Type: `integer`

Value: Wait time in seconds (minimum `30`, maximum `300`, default `30`)

The client replaces values outside this range with the nearest limit.

`mdm_file_debounce_secs`

Client modes	Zero Trust plans
All modes	All plans

System	Availability	Minimum client version
Windows		2026.9.1555.1
macOS		2026.9.1555.1
Linux		2026.9.1555.1
iOS		—

System	Availability	Minimum client version
Android	✅	—
ChromeOS	✅	—

How long the Cloudflare One Client waits after the MDM file (`mdm.xml` or `com.cloudflare.warp.plist`) changes before it applies the new configuration. Each new change restarts the wait, so the client applies several quick changes together.

Value Type: `integer`

Value: Wait time in seconds (minimum `1`, maximum `60`, default `10`)

The client replaces values outside this range with the nearest limit.

`mdm_file_unavailable_debounce_secs`

Client modes	Zero Trust plans
All modes	All plans

System	Availability	Minimum client version
Windows	✅	2026.9.1555.1
macOS	✅	2026.9.1555.1
Linux	✅	2026.9.1555.1
iOS	✅	—
Android	✅	—
ChromeOS	✅	—

How long the Cloudflare One Client keeps the last applied MDM configuration after the MDM file goes missing or becomes unreadable. The wait starts at the first file change that leaves the file unavailable. If the file is still unavailable when the wait ends, the client removes the MDM configuration, including any managed registration.

Some MDM tools replace the file by deleting it and creating it again. A wait longer than that gap keeps the client from removing the registration.

Value Type: `integer`

Value: Wait time in seconds (minimum `0`, maximum `300`)

The client replaces values outside this range with the nearest limit. If you do not set this parameter, the client waits 20 seconds on macOS. On Windows and Linux, the client removes the MDM configuration as soon as it fails to read the file.

`onboarding`

Controls the visibility of the onboarding screens that ask the user to review the privacy policy during an application's first launch.

Value Type: `boolean`

Value:

- `false` — Screens hidden.
- `true` — (default) Screens visible.

`override_api_endpoint`

Overrides the IP address used by the Cloudflare One Client to communicate with the client orchestration API. If you set this parameter, be sure to update your organization's firewall to ensure the new IP is allowed through.

This functionality is intended for use with a Cloudflare China local network partner or any other third-party network partner that can maintain the integrity of network traffic. Most IT admins should not set this setting as it will redirect all API traffic to a new IP.

Value Type: `string`

Value: `1.2.3.4` — Redirect all client orchestration API calls to `1.2.3.4`.

The string must be a valid IPv4 or IPv6 address, otherwise the Cloudflare One Client will fail to parse the entire MDM file.

`override_doh_endpoint`

:::note Only supported in DNS only mode.[^1] :::

[^1]: Traffic and DNS mode is supported in client version 2025.2.664.0 and below. In version 2025.4.589.1 and above, this parameter does not apply to Traffic and DNS mode because all DoH traffic goes inside of the WARP tunnel.

Overrides the IP address used by the Cloudflare One Client to resolve DNS queries via DNS over HTTPS (DoH). If you set this parameter, be sure to update your organization's firewall to ensure the new IP is allowed through.

This functionality is intended for use with a Cloudflare China local network partner or any other third-party network partner that can maintain the integrity of network traffic. Most IT admins should not set this setting as it will redirect all DoH traffic to a new IP.

Value Type: `string`

Value: `1.2.3.4` — Redirect all DNS over HTTPS lookups to `1.2.3.4`.

The string must be a valid IPv4 or IPv6 address, otherwise the Cloudflare One Client will fail to parse the entire MDM file.

`override_warp_endpoint`

Overrides the IP address and UDP port used by the Cloudflare One Client to send traffic to Cloudflare's edge. If you set this parameter, be sure to update your organization's firewall to ensure the new IP is allowed through.

This functionality is intended for use with a Cloudflare China local network partner or any other third-party network partner that can maintain the integrity of network traffic. Most IT admins should not set this setting as it will redirect all Cloudflare One Client traffic to a new IP.

Value Type: `string`

Value: `203.0.113.0:500` — Redirect all Cloudflare One Client traffic to `203.0.113.0` on port `500`.

The string must be a valid IPv4 or IPv6 socket address (containing the IP address and port number), otherwise the Cloudflare One Client will fail to parse the entire MDM file.

`service_mode`

Allows you to choose the operational mode of the client.

Value Type: `string`

Value:

- `warp` — (default) Traffic and DNS mode.
- `1dot1` — DNS only mode.

- `proxy` — Local proxy mode. Use the `proxy_port` parameter to specify the localhost SOCKS proxy port (between `0`-`66535`). For example,

```
<key>service_mode</key>
<string>proxy</string>
<key>proxy_port</key>
<integer>44444</integer>
```

- `postureonly` — Posture only mode.
- `tunnelonly` - Traffic only mode.

`support_url`

When the Cloudflare One Client is deployed via MDM, the in-app **Send Feedback** button is disabled by default. This parameter allows you to re-enable the button and direct feedback towards your organization.

Value Type: `string`

Value:

- `https://<support.example.com>` — Use an `https://` link to open your company's internal help site.
- `mailto:<yoursupport@example.com>` — Use a `mailto:` link to open your default mail client.

`switch_locked`

Allows the user to turn off the client switch and disconnect the Cloudflare One Client.

Value Type: `boolean`

Value:

- `false` — (default) The user is able to turn the switch on/off at their discretion. When the switch is off, the user will not have the ability to reach sites protected by Access that leverage certain device posture checks.
- `true` — The user is prevented from turning off the switch. The Cloudflare One Client will automatically start in the connected state.

On new deployments, you must also include the `auto_connect` parameter with at least a value of `0`. This will prevent clients from being deployed in the off state without a way for users to manually enable them.

:::note This parameter replaces the old `enabled` property, which can no longer be used in conjunction with the new `switch_locked` and `auto_connect`. If you want to use these parameters, you must remove `enabled`. :::

`unique_client_id`

:::note Only valid for iOS and Android/ChromeOS. :::

Assigns a unique identifier to the device for the device UUID posture check.

Value Type: `string`

Value: UUID for the device (for example, `496c6124-db89-4735-bc4e-7f759109a6f1`).

`use_in_app_webview`

:::note Only valid for Android and ChromeOS. :::

Uses an in-app WebView for Zero Trust authentication instead of the default system browser. This setting is required when Always-on VPN with Lockdown mode is enabled.

Value Type: `boolean`

Value:

- `false` — (default) Uses the default system browser for Zero Trust authentication.
- `true` — Uses an in-app WebView for Zero Trust authentication.

`warp_tunnel_protocol`

Configures the protocol used to route IP traffic from the device to Cloudflare Gateway. For more information, refer to Device tunnel protocol.

Value Type: `string`

Value:

- `masque` — (default) MASQUE protocol
- `wireguard` — WireGuard protocol

Top-level parameters

Top-level parameters determine how the Cloudflare One Client manages device registrations.

`configs`

Allows a user to switch between Zero Trust organizations in the Cloudflare One Client GUI. The `configs` array is also required when using another top-level parameter such as `multi_user` or `pre_login`, even if only one organization is specified.

Value Type: `array`

Value: An array containing one or more Zero Trust organizations.

`multi_user`

Enables multiple user registrations on a Windows device.

Value Type: `boolean`

Value:

- `false` — (default) Only one Cloudflare One Client registration is stored per device. After a user logs in to the Cloudflare One Client, their settings and identity will apply to all traffic from the device.
- `true` — Each Windows user has their own Cloudflare One Client registration. For more information, refer to Multiple users on a Windows device.

`organization_configs`

Client modes	Zero Trust plans
All modes	All plans

System	Availability	Minimum WARP version
Windows		2026.5.0
macOS		2026.5.0
Linux		2026.5.0
iOS		N/A
Android		N/A

System	Availability	Minimum WARP version
ChromeOS		N/A

Defines organization-wide settings that apply to all `configs` entries for a given organization, overriding any conflicting values set within individual config entries.

Value Type: `dict`

Value: A dict where each key is an organization name (team name) and each value is a dict of organization-wide settings.

Supported settings:

Key	Description
`environment`	Overrides the `environment` value for all configs belonging to this organization.
`hardware_backed_registration`	Overrides the `hardware_backed_registration` value for all configs belonging to this organization.

Use `organization_configs` when you have multiple configurations for the same organization and want to enforce a single `environment` value across all of them. Any `environment` value set inside an individual `configs` entry for that organization will be overridden by the value specified in `organization_configs`.

Example configuration where `test-org` has two configs (Production and Staging), both enforced to use `fedramp_high` via `organization_configs`:

```
<dict>
  <key>organization_configs</key>
  <dict>
    <key>test-org</key>
    <dict>
      <key>environment</key>
      <string>fedramp_high</string>
    </dict>
  </dict>
  <key>configs</key>
  <array>
    <dict>
      <key>organization</key>
      <string>test-org</string>
```

```
      <key>display_name</key>
      <string>Production</string>
    </dict>
    <dict>
      <key>organization</key>
      <string>test-org</string>
      <key>display_name</key>
      <string>Staging</string>
    </dict>
  </array>
</dict>
```

`pre_login`

Allows the Cloudflare One Client to connect with a service token before a user completes the initial Windows login. For more information, refer to Connect the Cloudflare One Client before Windows login.

Per-app VPN parameters (Android)

Per-app VPN parameters allow you to choose the Android apps that can send traffic through the WARP tunnel. Admins can configure these parameters via any MDM tool that supports deploying an Android app to managed devices or work profiles.

`app_identifier`

An application package name/bundle identifier which uniquely identifies the app on the Google Play Store. This application will be tunneled through the Cloudflare One Client service.

Value Type: `string`

Value: The app identifier can be found in the ID query parameter of the specific app's Play Store URL. For example: in the case of `https://play.google.com/store/apps/details?id=com.cloudflare.cloudflareoneagent`, the app identifier for the Cloudflare One Agent app is `com.cloudflare.cloudflareoneagent`.

`is_browser`

An optional property. `is_browser` will help the Cloudflare One Agent application decide which browser to open instead of the default browser for specific features such as re-authentication and Gateway block notifications. If needed, admins should explicitly indicate that a given `tunneled_app` is a browser, rather than relying on automatic browser detection.

Value Type: `boolean`

Value: If the value is `true`, identifies the application defined in `app_identifier` as a browser. The default value is `false` and `is_browser` is an optional property.

□□□ #2
□ □□□ □□□ 2026-11-01 01:17:49 UTC
□ □□□ □□□ 2026-11-01 01:17:52 UTC