Secrets Store

- Secrets Store access control
- Audit logs
- Cloudflare Secrets Store
- Secrets Store integrations
- Workers integration
- How to
- Manage account secrets

Secrets Store access control

Secrets Store allows security administrators to have more control by implementing role-based access. For details about roles at Cloudflare, refer to Fundamentals.

:::note[Availability] While all Cloudflare accounts will have access to the Secrets Store section on the dashboard, only users with the necessary permissions will be able to interact with it, as described below. :::

Access to a secret is controlled by two independent checks that must both pass:

1. **Authorization** — the caller must have permission to perform the action. The permission comes from either a user role (when the dashboard authenticates the request) or an API token permission (when the request is made with an API token). A given request is evaluated against one or the other, not both.
2. **Secret scope** — the scope list on the secret must include the consuming service.

For example, deploying a Worker that binds a secret requires a role or API token that can bind secrets, and a secret whose scope includes `workers`.

Relevant roles

Refer to the list below for default role definitions.

Super Administrator

- Can create, edit, duplicate, delete, and view secrets metadata.
- Can add a Secrets Store binding to a Worker.
- Can create an association between a secret and an AI gateway.

Secrets Store Admin

- Can create, edit, duplicate, delete, and view secrets metadata.

Secrets Store Deployer

- Can view secrets metadata but cannot create, edit, duplicate, nor delete secrets.
- Can add a Secrets Store binding to a Worker.

- Can create an association between a secret and an AI gateway.

Secrets Store Reporter

- Can view secrets metadata.
- Cannot perform any actions (create, edit, duplicate, delete secrets), nor use Secrets Store integrations with other Cloudflare products.

API token permissions

API tokens have two Secrets Store permission levels: **Read** and **Edit**. The permission you need depends on what the token is doing, not on whether you intend to modify the secret itself.

- **Account Secrets Store Read**: Allows the caller to view metadata for secrets (for example, listing secrets or fetching the name, ID, scopes, and comments of a secret). This permission does not grant access to the value of a secret, and does not allow binding a secret to another resource.
- **Account Secrets Store Edit**: Allows the caller to create, edit, duplicate, or delete secrets. **Edit is also required to bind a secret to another Cloudflare resource**, such as adding a Secrets Store binding to a Worker or associating a secret with an AI Gateway. Attaching a secret to a resource is treated as a write against the secret.

:::caution[Deploying Workers from CI/CD] If you use Wrangler in a CI/CD pipeline (for example, `wrangler-action` in GitHub Actions) to deploy a Worker that has a Secrets Store binding, the API token used by the pipeline must have **Account Secrets Store Edit** permission. A token with only **Read** will fail at deploy time with an authorization error such as `failed to fetch secrets store binding due to authorization error - check deploy permissions and secret scopes`. :::

Secret scopes

Each secret has a list of **scopes** that determine which Cloudflare services are allowed to consume it. Scopes are set when a secret is created, and can be updated later by editing the secret.

The currently supported scopes are:

- `workers` — allows the secret to be bound to a Worker.
- `ai-gateway` — allows the secret to be associated with an AI Gateway.

A request to bind or associate a secret with a service will be rejected if that service is not in the scope list, even if the caller has the correct role or API token permission. Deploying a Worker with a

Secrets Store binding therefore requires both:

- A user role or API token that can bind secrets (Super Administrator or Secrets Store Deployer role, or **Account Secrets Store Edit** API token permission).
- A scope list on the secret that includes `workers`.

You can set scopes when creating a secret using the dashboard, the API (`scopes` field), or Wrangler (`--scopes` flag).

Audit logs

Audit logs provide a comprehensive summary of changes made within your Cloudflare account. This page lists the actions that are logged for Secrets Store.

- Access
- Create
  - Duplicating a secret is presented as a `create` log with a field `duplicated_from_id`.
- Update
  - A boolean `"value_modified": true` is presented when the secret value is edited.
- Delete

For information on how to access and use audit logs, refer to Fundamentals.

Cloudflare Secrets Store

import { Description, Plan } from "~/components";

Encrypt and store sensitive information as secrets that are securely reusable across your account. Cloudflare Secrets Store is a secure, centralized location in which account-level secrets are stored and managed. The secrets are securely encrypted and stored across all Cloudflare data centers.

Secrets Store is currently compatible with Cloudflare Workers and AI Gateway. Integrations with other products will be added in the future.

:::note[China availability] Secrets Store is unavailable in the Cloudflare China Network, operated by Cloudflare's partner JD Cloud. :::

Secrets Store integrations

import { DirectoryListing } from "~/components";

Workers integration

import { WranglerConfig, Tabs, TabItem, APIRequest, DashButton } from "~/components";

Cloudflare Secrets Store is a secure, centralized location in which account-level secrets are stored and managed. The secrets are securely encrypted and stored across all Cloudflare data centers.

Consider the steps below to learn how to use values from your account secrets store with Cloudflare Workers. :::note This is different from Workers Variables and Secrets, where you define and manage your secrets on a per-Worker level. :::

Before you begin

- If using the Dashboard, make sure you already have a Workers application. Refer to the Workers get started for guidance.
- You should also have a store created under the **Secrets Store** tab on the Dashboard. The first store in your account is created automatically when a user with Super Administrator or Secrets Store Admin role interacts with it.

  ```
    - If no store exists in your account yet and you have the necessary permissions, you can use the [Wrangler command](/workers/wrangler/commands/secrets-store/#secrets-store-store) `secrets-store store create <name> --remote` to create your first store.
  ```

:::caution[Local development mode] This guide assumes you are working in production. To use Secrets Store locally, you must use `secrets-store secret` Wrangler commands without the `--remote` flag. :::

1. Set up account secrets in Secrets Store

Follow the steps below to create secrets. You must have a Super Administrator or a Secrets Store Admin role within your Cloudflare account.

:::note You may also add account secrets directly from the Workers settings on the dashboard. You can skip to step 2 to do that. :::

Use the Wrangler command `secrets-store secret create`.

To use the following example, replace the store ID and secret name by your actual data. You can find and copy the store ID from the Secrets Store tab on the dashboard or use `wrangler secrets-store store list`.

Note that a secret name cannot contain spaces.

```
npx wrangler secrets-store secret create <STORE_ID> --name MY_SECRET_NAME --scopes workers --remote
```

```
✓ Enter a secret value: › ***

Creating secret... (Name: MY_SECRET_NAME, Value: REDACTED, Scopes: workers, Comment: undefined)
✓ Select an account: › My account
Created secret! (ID: 13bc7498c6374a4e9d13be091c3c65f1)
```

1. In the Cloudflare dashboard, go to the **Secrets Store** page.
2. Select **Create secret**.
3. Fill in the required fields, choosing *Workers* as the **Permission scope**. Once the secret is saved, the secret value will no longer be available for viewing.
4. (Optional) Select **Add additional secret** to create more than one secret at a time.
5. Select **Save** to confirm.

You can find and copy the store ID from the Secrets Store tab on the dashboard or use the Wrangler command. Also, make sure your secret `name` does not contain spaces.

Refer to Secrets Store API for the full API documentation.

<APIRequest path="/accounts/{account_id}/secrets_store/stores/{store_id}/secrets" method="POST" json={[{ "name":"<MY_SECRET_NAME>", "value":"<SECRET_VALUE>", "scopes":["workers"], "comment":"" }, { "name":"<MY_SECRET_NAME_2>", "value":"<SECRET_VALUE>", "scopes":["workers"], "comment":"" }]} />

Refer to manage account secrets for further options.

2. Bind an account secret to your Worker

Bindings allow your Worker to interact with resources on your Cloudflare account.

To bind an account secret to your Worker, you must have one of the following roles within your Cloudflare account:

- Super Administrator
- Secrets Store Deployer

Via Wrangler

1. Add a Secrets Store binding to your Wrangler configuration file:
   - `binding`: a descriptive name for your binding. This will be used in the Workers application when accessing your secret on the `env` object.
   - `store_id`: the corresponding Secrets Store ID where your account secret was created.
   - `secret_name`: the unique secret name, defined when your account secret was created.

```
{
	"main": "./src/index.js",
	"upload_source_maps": true,
	"secrets_store_secrets": [
		{
			"binding": "<BINDING_VARIABLE>",
			"store_id": "<STORE_ID>",
			"secret_name": "<MY_SECRET_NAME>"
		}
	]
}
```

Via Dashboard

1. In the Cloudflare dashboard, go to **Workers & Pages**.
2. Select a Workers application.
3. Go to **Settings** > **Bindings** and select **Add**.
4. On the **Add a resource binding** side panel, choose **Secrets Store**.
5. Fill in the required fields:
   - **Variable name**: a name for the binding. This will be used for your Worker to access the secret (step 3 below).
   - **Secret name**: select from the list of available account secrets created in step 1.
   - (Optional - Admins only) If the secret you need does not exist yet, select **Create secret**. This will add an account level secret in the same way as if you had created it on the Secrets Store.
6. Select **Deploy** to deploy your binding. When deploying, there are two options:
   - **Deploy:** Immediately deploy the binding to 100% of your audience.
   - **Save version:** Save a version of the binding which you can deploy in the future.

3. Access the secret on the `env` object

Bindings are located on the `env` object. To access the secret you first need an asynchronous call.

Call `get()` on the binding variable

:::caution[Local development mode] You cannot access production secrets (created on the dashboard, via API, or with the `--remote` flag) from your local development setup. To use Secrets Store locally, you must use `secrets-store secret` Wrangler commands without the `--remote` flag. :::

```
export default {
  async fetch(request, env) {
    // Example of using the secret safely in an API request
□□const APIkey = await env.<BINDING_VARIABLE>.get()

    const response = await fetch("https://api.example.com/data", {
      headers: { "Authorization": `Bearer ${APIKey}` },
    });
```

```
    if (!response.ok) {
      return new Response("Failed to fetch data", { status: response.status });
    }

    const data = await response.json();
    return new Response(JSON.stringify(data), {
      headers: { "Content-Type": "application/json" },
    });
  },
};
```

How to

import { Tabs, TabItem, APIRequest, DashButton } from "~/components";

Refer to the sections below to learn about common actions you might want to take when managing your data in Secrets Store.

You must have a Super Administrator or Secrets Store Admin role within your Cloudflare account.

Manage via Wrangler

Wrangler is a command-line interface (CLI) that allows you to manage Cloudflare Workers projects.

Refer to Wrangler commands for guidance on how to use it with Secrets Store.

Create a secret

1. In the Cloudflare dashboard, go to the **Secrets Store** page.
2. Select **Create secret**.
3. Fill in the required fields. Note that, once the secret is saved, the secret value will no longer be available for viewing.
4. (Optional) Select **Add additional secret** to create more than one secret at a time.
5. Select **Save** to confirm.

:::note A secret `name` cannot contain spaces. Refer to Secrets Store API for the full API documentation. :::

<APIRequest path="/accounts/{account_id}/secrets_store/stores/{store_id}/secrets" method="POST" json={[{ "name":"<MY_SECRET_NAME>", "value":"<SECRET_VALUE>", "scopes":["workers"], "comment":"" }, { "name":"<MY_SECRET_NAME_2>", "value":"<SECRET_VALUE>", "scopes":["workers"], "comment":"" }]} />

Duplicate a secret

Duplicate a secret to keep the same secret value but change name, scope, or comments.

1. In the Cloudflare dashboard, go to the **Secrets Store** page.
2. Search for the secret you would like to duplicate within the existing secrets list.
3. Select the three dots next to the secret and choose **Duplicate**.
4. Edit the **Secret name**, **Permission scope**, or **Comment**, according to your needs.
5. Select **Save** to confirm.

:::note A secret `name` cannot contain spaces. Refer to Secrets Store API for the full API documentation. :::

```
curl
https://api.cloudflare.com/client/v4/accounts/$ACCOUNT_ID/secrets_store/stores/$STORE_ID/secre
ts/$SECRET_ID/duplicate \
--header "Authorization: Bearer $CLOUDFLARE_API_TOKEN" \
--header "Content-Type: application/json" \
--data '{
    "name":"<NEW_DUPLICATE_NAME>",
    "scopes":["workers"],
    "comment":""
}'
```

Edit a secret

Edit a secret to replace an existing value with a new one.

:::caution This action will cause the replacement in all services using the secret. :::

You can also edit the secret **Permission scope** and **Comment**.

1. In the Cloudflare dashboard, go to the **Secrets Store** page.

2. Search for the secret you would like to edit within the existing secrets list.
3. Select the three dots next to the secret and choose **Edit**.
4. Edit the available fields according to your needs and select **Save** to confirm.

Refer to Secrets Store API for the full API documentation.

```
curl --request PATCH \
https://api.cloudflare.com/client/v4/accounts/$ACCOUNT_ID/secrets_store/stores/$STORE_ID/secrets/$SECRET_ID \
--header "Authorization: Bearer $CLOUDFLARE_API_TOKEN" \
--header "Content-Type: application/json" \
--data '{
    "comment":"<NEW_COMMENT>",
    "value":"<NEW_SECRET_VALUE>",
    "scopes":["workers"]
}'
```

Delete a secret

:::caution Before deleting a secret, make sure it is not deployed in your Workers applications or AI gateways. :::

1. In the Cloudflare dashboard, go to the **Secrets Store** page.
2. Search for the secret you would like to delete within the existing secrets list.
3. Select the three dots next to the secret and choose **Delete**.
4. Type in the secret name and select **Delete** to confirm.

Refer to Secrets Store API for the full API documentation.

Manage account secrets

Secrets can be API tokens, public/private keys, authorization keys, passwords, or even code variables. The only limitation is that a secret must be a string that does not exceed 65,536 bytes.

Once a secret is added to the Secrets Store, it can no longer be decrypted or accessed via API or on the dashboard. Only the service associated with a given secret will be able to access it.

Limits

Customers who create a secrets store in the open beta can have up to 100 secrets per account. Also, there can only be one store per account.

:::note[Production secrets] If you use Wrangler, there is a difference between production secrets and secrets that are only created locally (without the `--remote` flag). The limit of 100 secrets per account only considers production secrets. :::

:::caution[Local secrets are not encrypted] Secrets that you create locally (without the `--remote` flag) are stored unencrypted in your project's local data directory, which is `.wrangler/state` by default. Use test values for local development, and do not commit the local data directory to version control. :::

Resources

- Manage via Wrangler
- Create a secret
- Duplicate a secret
- Edit a secret
- Delete a secret