# 1Password CLI Secrets from the shell

## CONCEPTS

Keep secrets out of files: store op:// references instead, and let op run or op inject resolve them at the last moment.

**Reference** op://vault/item/field
↓ op run / inject
**1Password CLI** signed-in account
↓ resolves
**Secret value** env var or file

## SIGN IN

`op signin`
Sign in (prompts only if needed)

`op whoami`
Show the signed-in account

## VAULTS

`op vault list`
List readable vaults

## ITEMS

`op item list`
List items

`op item list --vault <vault>`
List items in one vault

`op item get <item>`
Show item details

`op item get <item> --fields label=password`
Print one field

`op item get <item> --otp`
Current one-time password

`op item get <item> --format json`
Full item as JSON

`op item create --category login --title <title>`
Create a login item

`op item create ... --generate-password=20,letters,digits`
Create with a generated password

## ITEMS (cont.)

`op item create ... <field>=<value>`
Set a field on create

`op item edit <item> <field>=<value>`
Change a field

`op item delete <item>`
Delete an item

## SECRET REFERENCES

`op read op://<vault>/<item>/<field>`
Print a secret

`op read op://<vault>/<item>/<section>/<field>`
Read a field in a section

`op read "op://<vault>/<item>/one-time password?attribute=otp"`
Read the current OTP

References use names or IDs. Quote them if a name contains spaces.

## RUN AND INJECT

`op run -- <cmd>`
Run with env var references resolved

`op run --env-file=<file> -- <cmd>`
Resolve references from a .env file

`op inject -i <tpl> -o <out>`
Fill a template into a file

Templates use {{ op://vault/item/field }}. Env vars inside references, like op://$APP_ENV/db/password, switch vaults per environment.

## DOCUMENTS

`op document create <file> --title <title>`
Upload a file

`op document get <doc> -o <file>`
Download a document