Caddy Web server, auto HTTPS

CONCEPTS

> With a Caddyfile in the current directory, caddy run loads it with no flags.

Caddyfile human-friendly config
↓ adapt
JSON config native format
↓ load / reload
Admin API localhost:2019
↓ applies
Running server auto HTTPS

> A site address with a real domain gets a public certificate automatically; localhost gets a locally trusted one.

RUN AND CONTROL

`caddy run`
Run in the foreground

`caddy start`
Run in the background

`caddy stop`
Stop the running instance gracefully

`caddy reload`
Apply the config to the running instance

`caddy run --config <file>`
Run with a specific config file

`caddy run --watch`
Reload automatically when the config changes

`caddy version`
Print the version

CONFIG TOOLS

`caddy validate --config <file>`
Load and provision without running

`caddy fmt --overwrite <file>`
Format the file in place

`caddy adapt --config <file>`
Convert a Caddyfile to JSON

QUICK SERVERS

`caddy file-server`
Serve the current directory

`caddy file-server --browse`
Serve with a directory listing

`caddy file-server --root <dir> --listen :<port>`
Serve a folder on a port

`caddy reverse-proxy --from <addr> --to <backend>`
Proxy requests to a backend

CADDYFILE BASICS

`example.com { ... }`
Site block with automatic HTTPS

`:8080 { ... }`
Site block on plain HTTP, port 8080

`http://example.com { ... }`
Explicit HTTP, no auto-HTTPS

`{ email <addr> }`
Global options block, must come first

`reverse_proxy <backend>`
Send requests to a backend

`file_server`
Serve static files from the site root

`root * <dir>`
Set the document root for all paths

`encode zstd gzip`
Compress responses

`redir <to> permanent`
Redirect to another URL

`try_files {path} /index.html`
Single-page app fallback

`handle /api/* { ... }`
Exclusive block for a path prefix

CERTIFICATES AND PASSWORDS

`caddy hash-password`
Prompt for a password and print its hash