Building and Testing

- Building
  - Build Traefik
- Testing

Compile and Test Your Own Traefik! {: .subtitle }

You want to build your own Traefik binary from the sources? Let's see how.

Building

You need: - Docker - `make` - Go - misspell - shellcheck - Tailscale if you are using Docker Desktop

!!! tip "Source Directory"

```
It is recommended that you clone Traefik into the `~/go/src/github.com/traefik/traefik`
directory.
This is the official golang workspace hierarchy that will allow dependencies to be properly
resolved.
```

!!! note "Environment"

```
Set your `GOPATH` and `PATH` variable to be set to `~/go` via:

```bash
export GOPATH=~/go
export PATH=$PATH:$GOPATH/bin
```

For convenience, add `GOPATH` and `PATH` to your `.bashrc` or `.bash_profile`

Verify your environment is setup properly by running `$ go env`.
```

````
Depending on your OS and environment, you should see an output similar to:

```bash
GOARCH="amd64"
GOBIN=""
GOEXE=""
GOHOSTARCH="amd64"
GOHOSTOS="linux"
GOOS="linux"
GOPATH="/home/<yourusername>/go"
GORACE=""
## ... and the list goes on
```
````

Build Traefik

Once you've set up your go environment and cloned the source repository, you can build Traefik.

```
$ make binary
SHA: 8fddfe118288bb5280eb5e77fa952f52def360b4 cheddar 2024-01-11_03:14:57PM
CGO_ENABLED=0 GOGC=off GOOS=darwin GOARCH=arm64 go build  -ldflags "-s -w \
    -X
github.com/traefik/traefik/v2/pkg/version.Version=8fddfe118288bb5280eb5e77fa952f52def360b4 \
    -X github.com/traefik/traefik/v2/pkg/version.Codename=cheddar \
    -X github.com/traefik/traefik/v2/pkg/version.BuildDate=2024-01-11_03:14:57PM" \
    -installsuffix nocgo -o "./dist/darwin/arm64/traefik" ./cmd/traefik

$ ls dist/
traefik*
```

You will find the Traefik executable (`traefik`) in the `./dist` directory.

Testing

Run unit tests using the `test-unit` target. Run integration tests using the `test-integration` target. Run all tests (unit and integration) using the `test` target.

```
$ make test-unit
GOOS=darwin GOARCH=arm64 go test -cover "-coverprofile=cover.out" -v ./pkg/... ./cmd/...
+ go test -cover -coverprofile=cover.out .
ok      github.com/traefik/traefik   0.005s  coverage: 4.1% of statements

Test success
```

For development purposes, you can specify which tests to run by using (only works the `test-integration` target):

??? note "Configuring Tailscale for Docker Desktop user"

````
Create `tailscale.secret` file in `integration` directory.

This file needs to contain a [Tailscale auth key](https://tailscale.com/kb/1085/auth-keys)
(an ephemeral, but reusable, one is recommended).

Add this section to your tailscale ACLs to auto-approve the routes for the
containers in the docker subnet:

```json
    "autoApprovers": {
      // Allow myself to automatically
      // advertize routes for docker networks
      "routes": {
        "172.31.42.0/24": ["your_tailscale_identity"],
      },
    },
```
````

```
# Run every tests in the MyTest suite
TESTFLAGS="-test.run TestAccessLogSuite" make test-integration

# Run the test "MyTest" in the MyTest suite
TESTFLAGS="-test.run TestAccessLogSuite -testify.m ^TestAccessLog\$\$" make test-integration
```

□□□ #2
□ □□□ □□□ 2026-10-31 13:57:45 UTC
□ □□□ □□□ 2026-10-31 13:57:45 UTC