static website + go:embed => ./website

Amit Saha

https://echorand.me

Problem Statement

- There was <u>no problem</u> to solve
- I <u>wanted</u> to host my new static website in the cloud on a virtual machine

Thinking about the solution

Initial plan - just copy the HTML and CSS files to the server

Put in *nginx* and be done

Then… I thought of writing my server in Go and serving the files from the filesystem

And then… I thought of the *embed* package

Solution to the problem

markdown -> hugo -> html + css ->

Go server with everything embedded

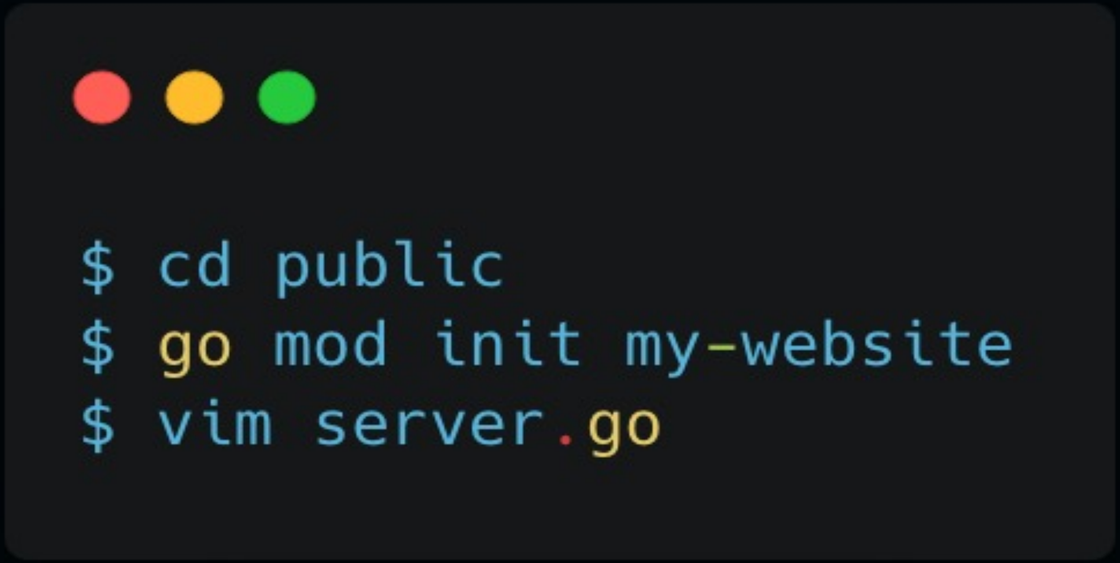
-> deploy

First pass

Create binary

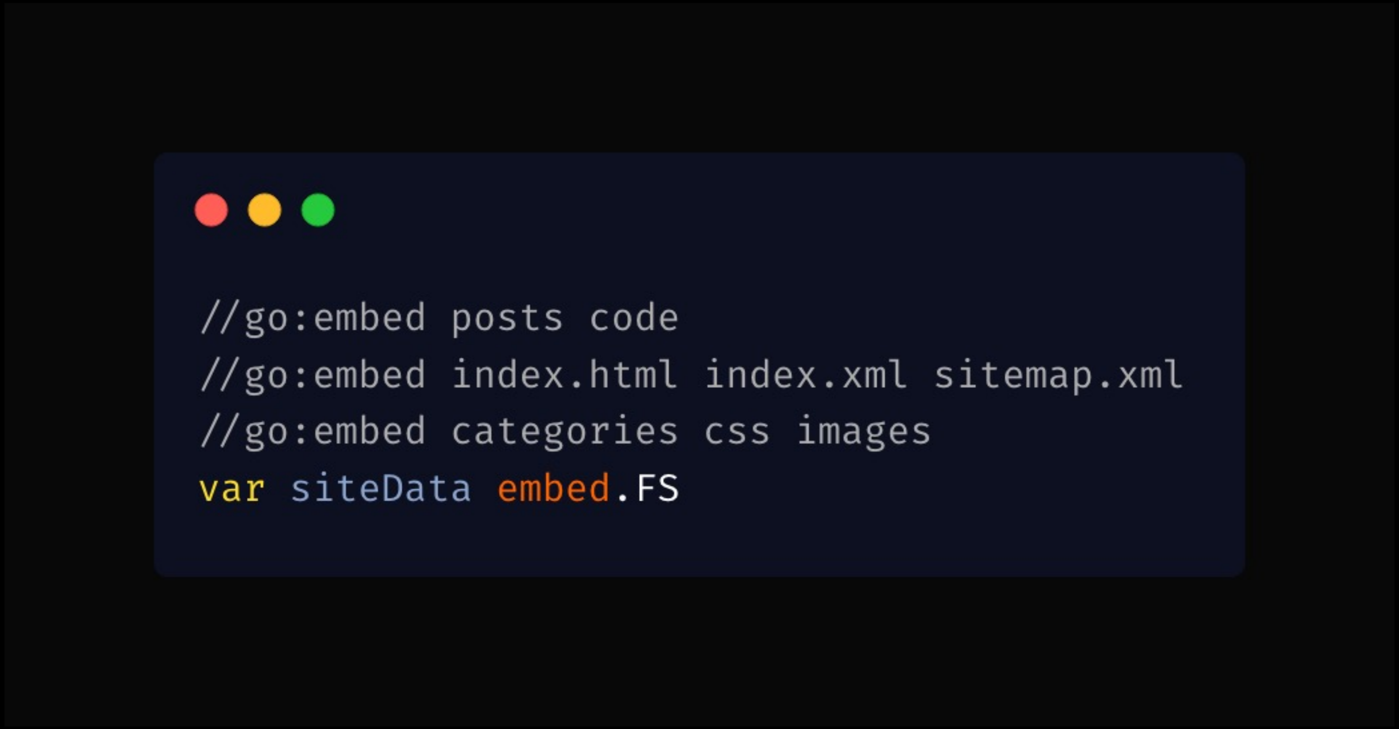
1. Create new hugo site
2. Put some content
3. Render the HTML
4. Then ..

Create a new module



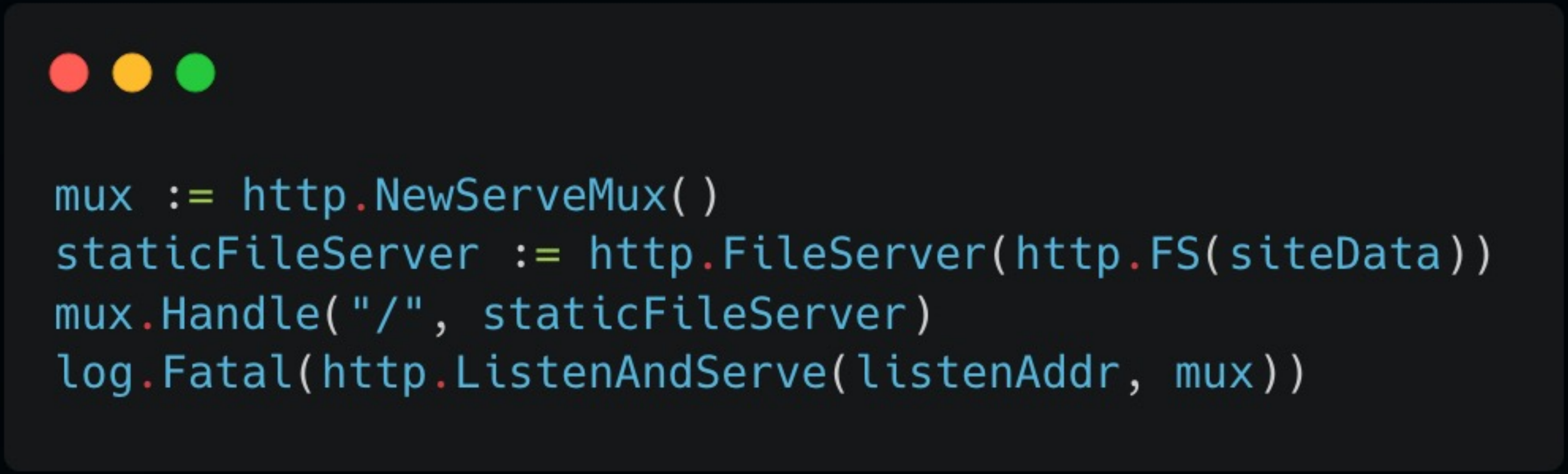
```
$ cd public
$ go mod init my-website
$ vim server.go
```

Write go:embed directives



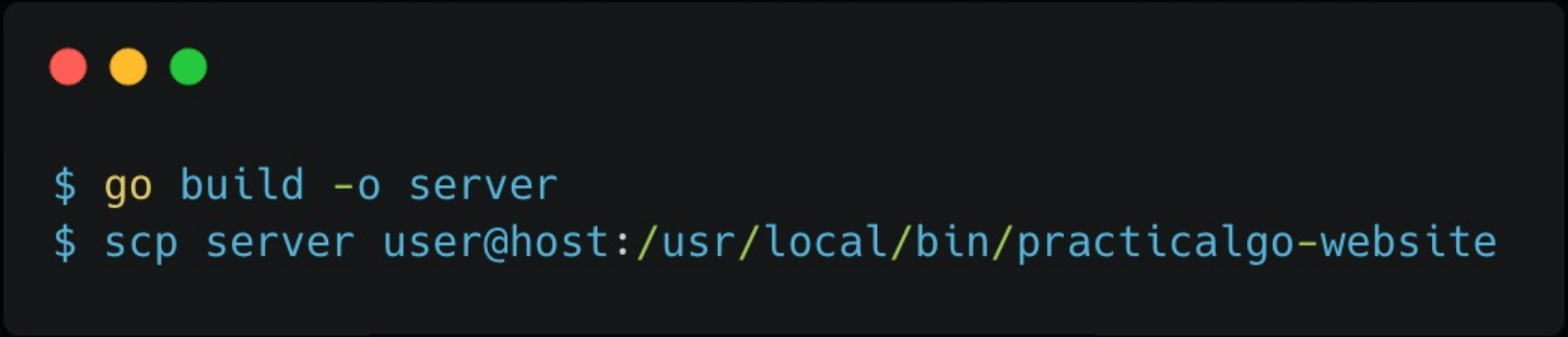
```go
//go:embed posts code
//go:embed index.html index.xml sitemap.xml
//go:embed categories css images
var siteData embed.FS
```

Write the server

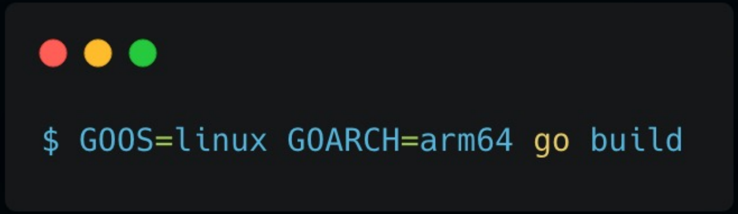


```go
mux := http.NewServeMux()
staticFileServer := http.FileServer(http.FS(siteData))
mux.Handle("/", staticFileServer)
log.Fatal(http.ListenAndServe(listenAddr, mux))
```

Build and deploy

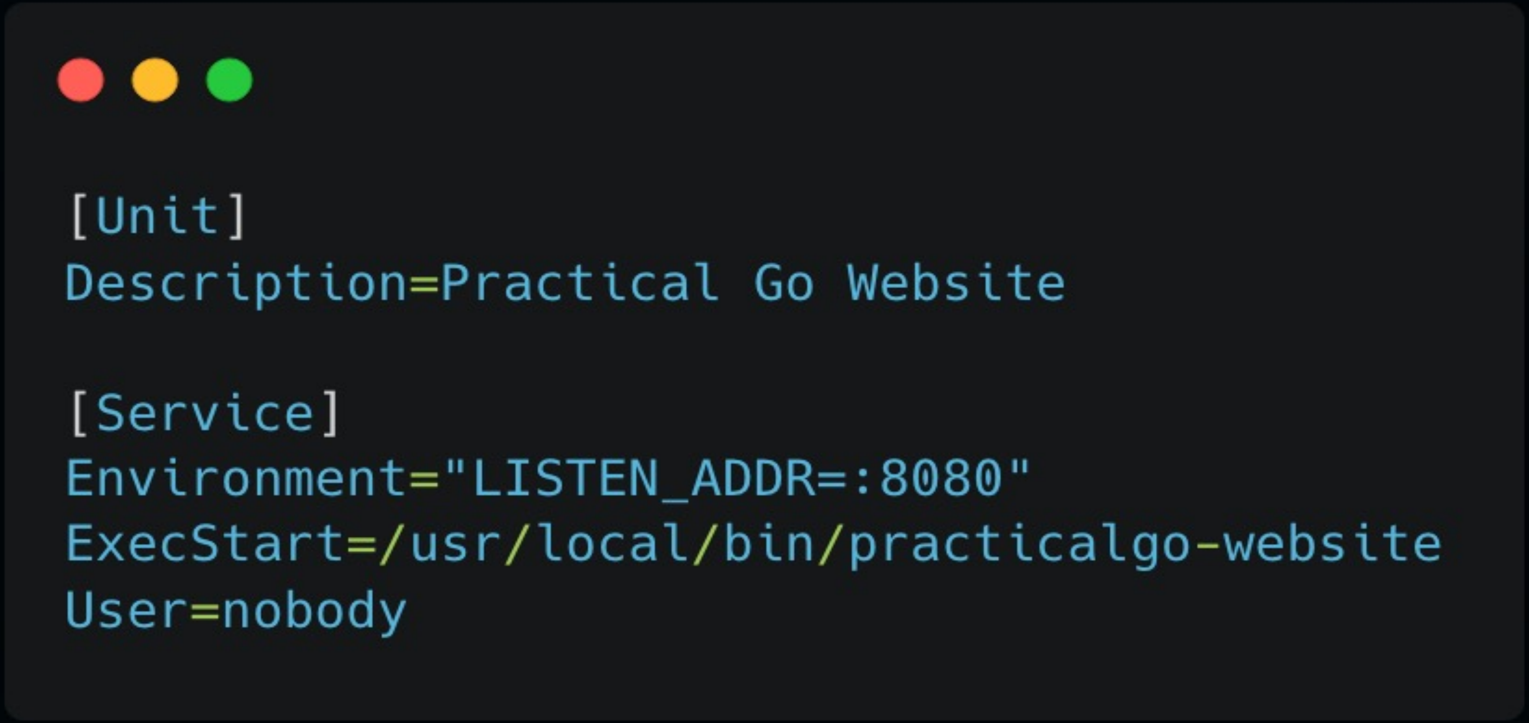


```
$ go build -o server
$ scp server user@host:/usr/local/bin/practicalgo-website
```



```
$ GOOS=linux GOARCH=arm64 go build
```

Systemd service



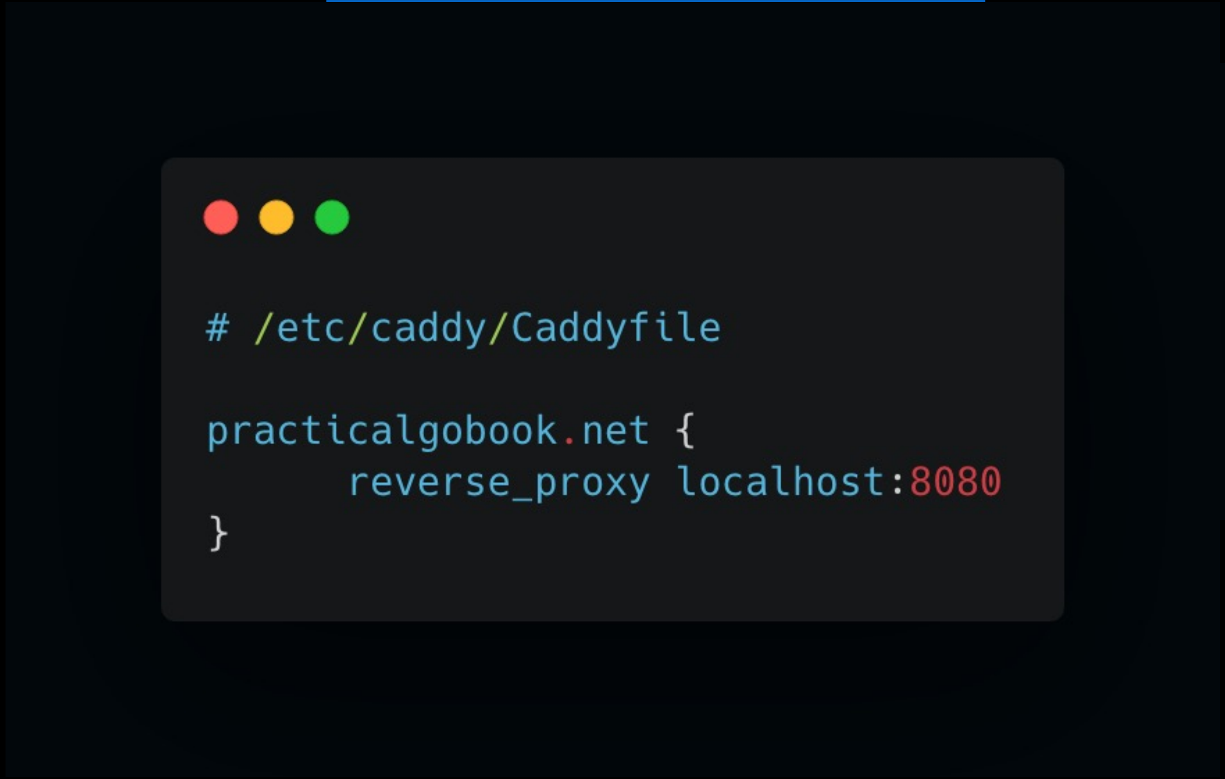
```
[Unit]
Description=Practical Go Website

[Service]
Environment="LISTEN_ADDR=:8080"
ExecStart=/usr/local/bin/practicalgo-website
User=nobody
```

DNS and IP address

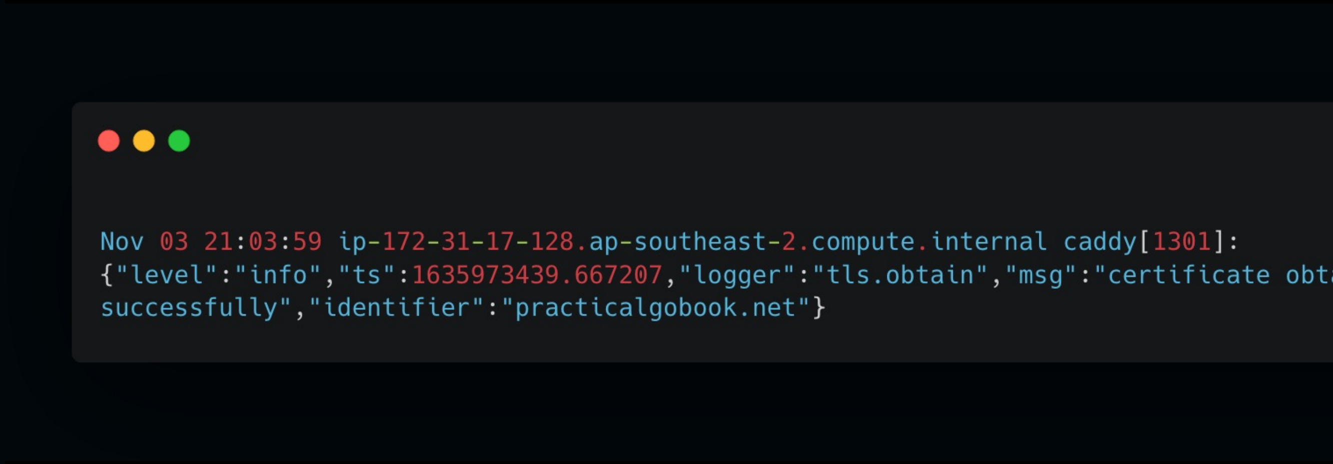
- Created a DNS record and pointed it to the public IP of my virtual machine

HTTPS and Reverse Proxy with Caddy

- Installed Caddy via the repository -> automatically systemd service
  - https://caddyserver.com/



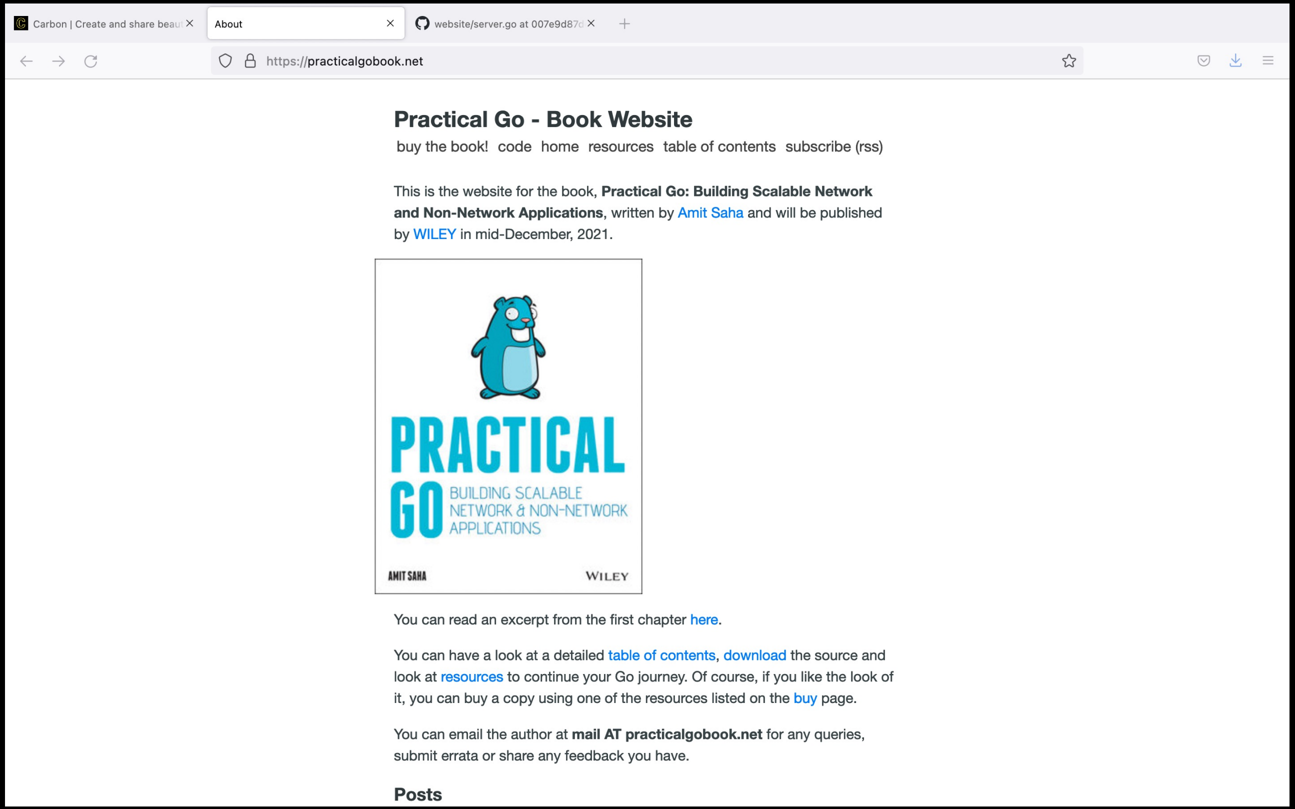
```
# /etc/caddy/Caddyfile

practicalgobook.net {
      reverse_proxy localhost:8080
}
```



```
Nov 03 21:03:59 ip-172-31-17-128.ap-southeast-2.compute.internal caddy[1301]:
{"level":"info","ts":1635973439.667207,"logger":"tls.obtain","msg":"certificate obt
successfully","identifier":"practicalgobook.net"}
```

Result - my new book's website!

https://practicalgobook.net


Carbon | Create and share beau
About
website/server.go at 007e9d87d

https://practicalgobook.net

Practical Go - Book Website

buy the book! code home resources table of contents subscribe (rss)

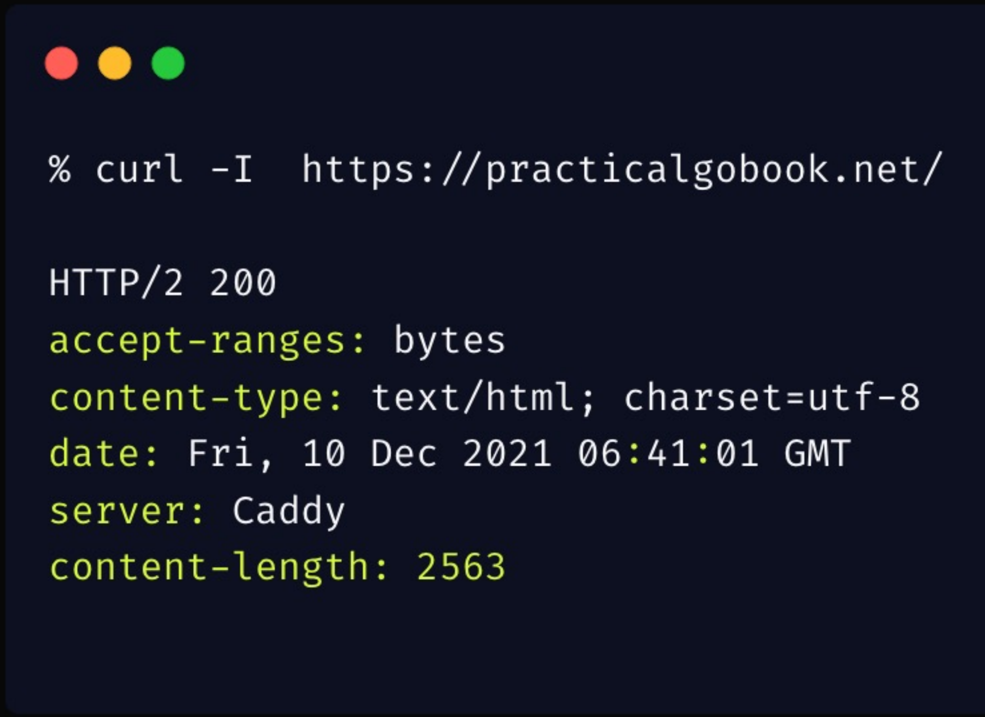
This is the website for the book, **Practical Go: Building Scalable Network and Non-Network Applications**, written by Amit Saha and will be published by WILEY in mid-December, 2021.

You can read an excerpt from the first chapter here.

You can have a look at a detailed table of contents, download the source and look at resources to continue your Go journey. Of course, if you like the look of it, you can buy a copy using one of the resources listed on the buy page.

You can email the author at **mail AT practicalgobook.net** for any queries, submit errata or share any feedback you have.

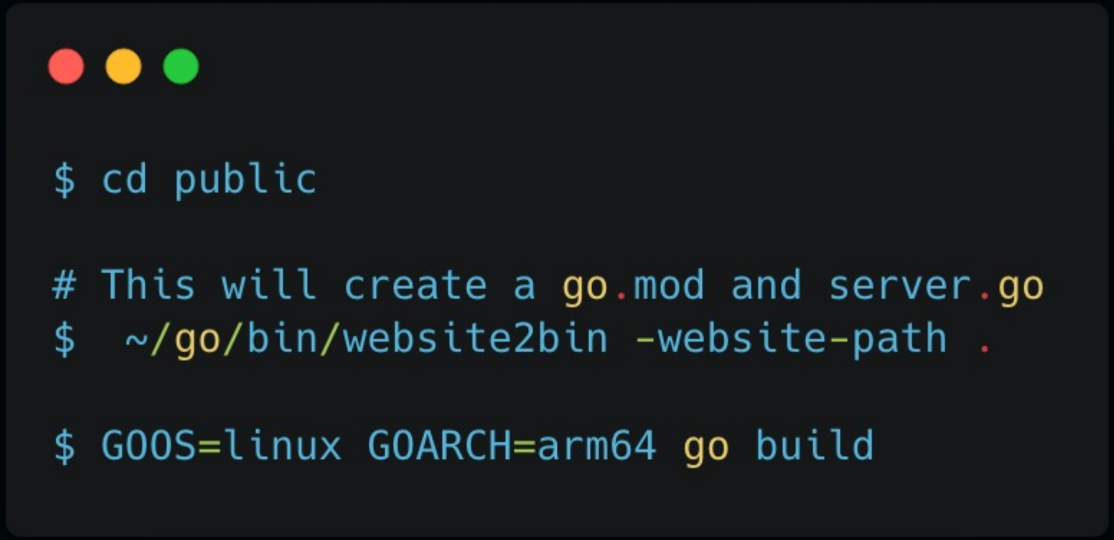
Posts


```
% curl -I  https://practicalgobook.net/

HTTP/2 200
accept-ranges: bytes
content-type: text/html; charset=utf-8
date: Fri, 10 Dec 2021 06:41:01 GMT
server: Caddy
content-length: 2563
```

Second pass

Automate the server creation

https://github.com/amitsaha/website2bin



```
$ cd public

# This will create a go.mod and server.go
$  ~/go/bin/website2bin -website-path .

$ GOOS=linux GOARCH=arm64 go build
```

More “problems” to solve

- The update/deployment story isn’t great

Thank you!

- https://echorand.me
- https://github.com/amitsaha