Go #GOLANG

Introduction and the Value of Learning to Code for SEs

Michael Gasch
Business Solution Architect
VMware Cloud-Native Applications SME

Agenda

Part #1

- Motivation and background
  - The value of learning to code
  - The value of coding in GOLANG
- Gopher time
- Language specification
- Go´s Playground

Part #2

- Setting up your workspace
- Coding time
  - “Hello World” - Your first GOLANG program
  - Building a simple web server
  - Leveraging Go´s built-in concurrency
  - Statically linking and the (fairly) unknown “FROM scratch” (Dockerfile)
- Useful resources

Motivation and background

Motivation and background

Why learn to code?

- Future-proof your career
- Understand the single source of truth (code)
- Look at your customers´ problems from the eye of a developer
- Understand why container technology is changing the SDLC
- Contribute (OSS)
- Towards the full stack engineer, well…

Why GOLANG?

- Rising star for writing distributed systems
- Concurrency built-in from the beginning
- Compiled, statically linking, fast builds
- Rich standard library (e.g. "net")
- Born out of Google, can it be wrong ☺ ?
- Feels fresh and makes fun
- Why you should learn Go
- How Go is making us faster

Gopher time

Open https://gopherize.me

Language specification

Language specification

- Web site: https://golang.org/
- Guided tutorial: https://tour.golang.org/welcome/1
- Digging deeper:
  - Documentation (all): https://golang.org/doc/
  - How to write Go code: https://golang.org/doc/code.html
  - Effective Go: https://golang.org/doc/effective_go.html
  - Command documentation: https://golang.org/doc/cmd

Go´s playground

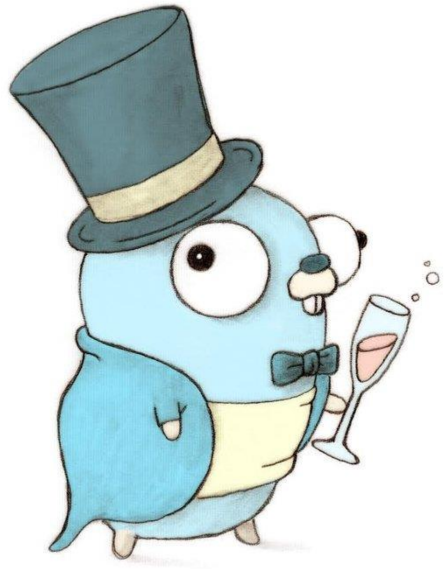
Open https://play.golang.org/

Setting up your workspace

Setting up your workspace

- Install the Go runtime
  - https://golang.org/doc/install or
  - (for OSX) I recommend Homebrew, following this guide
- Create and set up **GOPATH** (e.g. $HOME/_DEV)
  - You´d want to permanently set it, e.g. **export “GOPATH=$HOME/_DEV“** and also add GOBIN to your path (**export PATH=$PATH:$GOPATH/bin**)
- Go enforces this directory structure
  - $GOPATH/
    - src/
    - pkg/
    - bin/
- Use an editor with GOLANG tooling (fmt, import, etc.) support, e.g. Atom (more)

Coding time



vmware

“Hello World”

- Open and run https://play.golang.org/p/XEVnLJsWAe

Building a simple webserver

- Open and copy to your editor https://play.golang.org/p/dxuyoa-wMG
- Note: won´t run in playground

Simple webserver – Leveraging Go´s built-in concurrency

- Open and copy to your editor https://play.golang.org/p/7kYIiX4YIg
- Note: won´t run in playground

Statically linking and the (fairly) unknown “FROM scratch”

```
# Statically compile Go binary
CGO_ENABLED=0 GOOS=linux go build -a -ldflags '-w' -o ws1 .

# Dockerfile
FROM scratch
MAINTAINER Your Name <you@example.com>
ADD ws1 /
ENV PORT 8000
EXPOSE 8000
ENTRYPOINT ["/ws1"]

# Build and run docker image
docker build -t localhost:5000/ws1 .
docker run --rm -p 8000:8000 --name ws1 localhost:5000/ws1
```

Useful resources

Go-ing deeper

vmware®

Useful resources

- Example code on Github
  - git clone https://github.com/embano1/golang_training.git
- Read the Go documentation (see links “Language specification” in this presentation) many times
- Easy learning
  - https://gobyexample.com/
  - http://www.golangbootcamp.com/book/frontmatter
  - https://www.golang-book.com/books/intro
  - http://guzalexander.com/2013/12/06/golang-channels-tutorial.html
  - https://github.com/a8m/go-lang-cheat-sheet
- Tools like Sourcegraph help you to browse through (and understand) larger code bases
- Visualize Go concurrency https://divan.github.io/posts/go_concurrency_visualize/