

GOLANG AS COMMON LANGUAGE

---

**WELCOME TO GO**

**PHILIP, YOU HAVE TO CHECKOUT  
GOLANG, IT IS VERY POWERFUL  
FOR DOING DEVOPS!**

@rnrbarbosa **Roberto Barbosa**, December 2018

**GOT IT**



**I NEED TO TELL EVERYBODY**

[makeameme.org](http://makeameme.org)

## WHO AM I

- ▶ Philip Sahli
- ▶ DevOps Engineer @SwissPost
- ▶ MAS in Software Development at BFH Berne
- ▶ Containers (Dotcloud, Tutum, Kubernetes)
- ▶ Cloud (AWS, Digitalocean)
- ▶ Python and Django
- ▶ Go Programming and talking about it



ALSO KNOWN AS GOLANG

---

## WHAT IS GO

GO IS AN OPEN SOURCE PROGRAMMING LANGUAGE THAT MAKES IT EASY TO BUILD SIMPLE, RELIABLE, AND EFFICIENT SOFTWARE.

- ▶ Open source programming language
- ▶ Creates statically-linked binaries
- ▶ Statically typed → Type safety
- ▶ Automatic memory management



Go gopher

## WHY THEY CREATED GO

- ▶ Rob Pike  
Bell Labs, Unix,  
UTF-8
- ▶ Ken Thompson  
Bell Labs, B
- ▶ Robert Griesemer  
C++

Go was born out of **frustration** with existing languages and environments for the work we were doing at Google. Programming had become **too difficult** and the choice of languages was partly to blame. One had to choose either **efficient compilation**, **efficient execution**, or ease of **programming**; all three were not available **in the same mainstream language**.

Source: [https://golang.org/doc/faq#creating\\_a\\_new\\_language](https://golang.org/doc/faq#creating_a_new_language)

EVERYTHING STARTED IN 2007

## GO IS YOUNG

1987

Perl

1990

Python

1996

Java

2007

Idea for Go

2009

Go open sourced

2012

Go 1.0 released

2019

Go 1.13 (2019/09/03)

2020

Go 1.13.6 (2020/01/09)



## WHAT IS WRITTEN IN GO

- ▶ Docker
- ▶ Kubernetes
- ▶ Etc
- ▶ Hashicorp's Software
- ▶ Telegraf
- ▶ Hugo
- ▶ ... SDK's ...



## WHY TO USE GO

- ▶ One binary
- ▶ Standard Library
- ▶ Concurrency
- ▶ Broad tooling
- ▶ Simplicity
- ▶ Common language

# GO IS SIMPLE

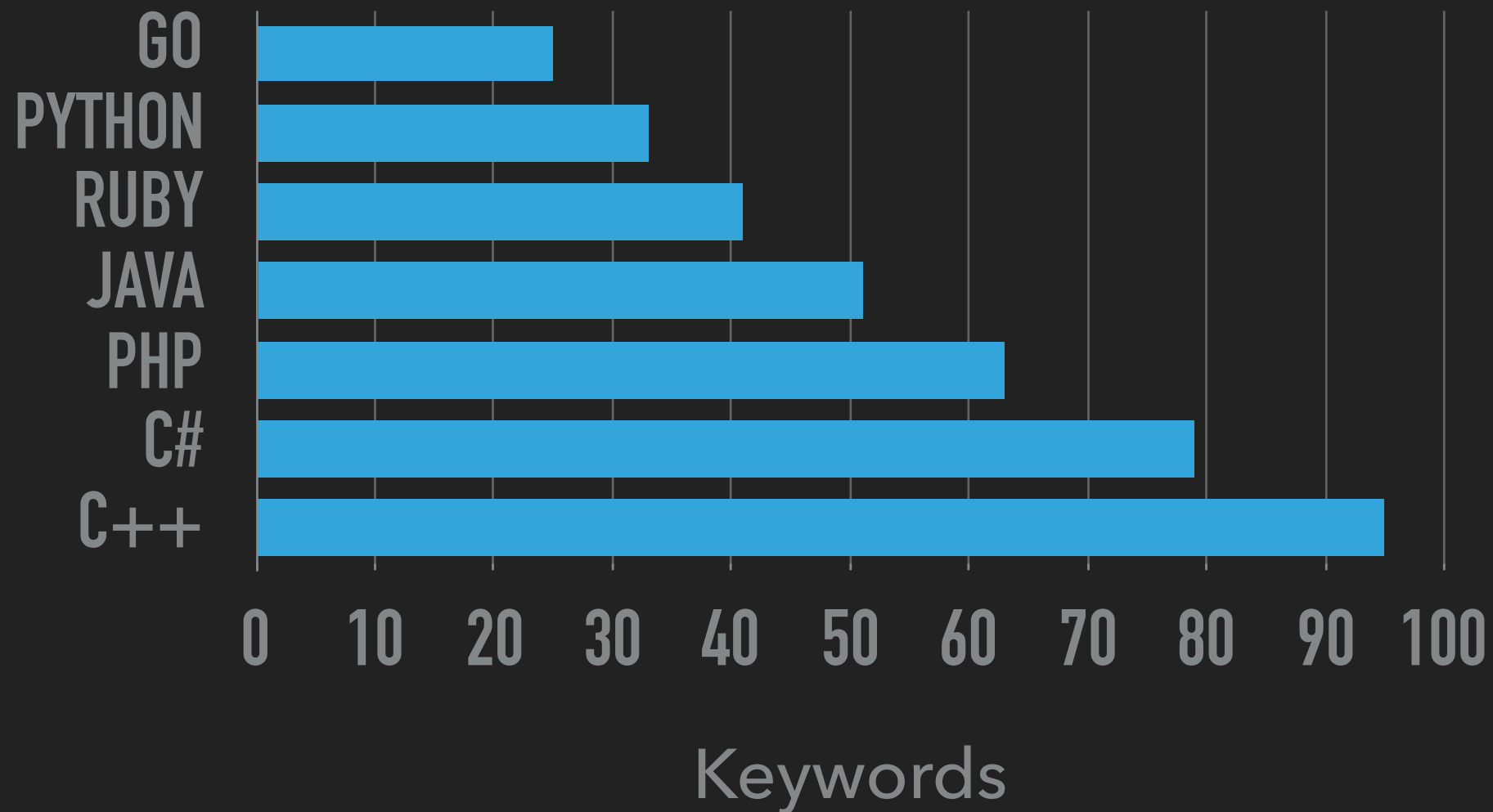
Read

Write

Run



Maintain



# WHERE TO USE GO

LINUX



DARWIN



WINDOWS



```
GOOS=linux GOARCH=amd64 go build cmd/cli
```

## CLIENTS

- ▶ Command-line interfaces
- ▶ Client daemons

## BACKEND

- ▶ API
- ▶ Microservices
- ▶ Serverless

## INFRASTRUCTURE

- ▶ Network daemons
- ▶ Automation
- ▶ Operators

## SDK

### BROWSER

- ▶ GopherJS
- ▶ WebAssembly

### MICROCONTROLLERS

- ▶ TinyGo

# HELLO BERNE



```
package main

import "fmt"

const text = "Hello Berne"

func main() {
    fmt.Println(text)
}
```



# IS GO OBJECT-ORIENTED?



- No classes, no inheritance
- Composition over inheritance
- Promotion technique



## COMPOSITION

- ▶ Embed other types
- ▶ Like Mixin's known in other languages

```
package main

import "fmt"

type Person struct {
    firstName string
    lastName  string
}

type Author struct {
    Person
    Publisher string
}

func main() {
    a := Author{}
    a.firstName = "John"
    a.lastName = "Doe"
    a.Publisher = "AuthorLab"
    fmt.Println(a)
}
```

## RECEIVER FUNCTIONS

```
package main

import "fmt"

type author struct {
    firstName string
    lastName  string
}

func (a *author) fullName() string {
    return fmt.Sprintf("%s %s", a.firstName, a.lastName)
}

func main() {
    ps := author{"Philip", "Sahli"}
    fmt.Println(ps.fullName())
}
```

## CONCURRENCY

- ▶ A goroutine is a lightweight thread managed by the Go runtime.
- ▶ Don't communicate by sharing memory, share memory by communicating



```
package main

import "fmt"

func main() {

    messages := make(chan string)

    go func() { messages <- "ping" }()

    msg := <-messages
    fmt.Println(msg)

}
```



# ERROR HANDLING



```
package main

import (
    "errors"
    "log"
)

func stringTooLong(s string) error {
    if len(s) > 10 {
        return errors.New("string is too long.")
    }
    return nil
}

func main() {
    err := stringTooLong("abcdefghijklm")
    if err != nil {
        log.Fatal(err)
        // Output: 2020/01/27 09:52:22 string is too long.
    }
}
```

## PACKAGE SYSTEM

- ▶ Packages in Go Core <https://golang.org/pkg/>

archive/zip, html/template, net/http, time, fmt, crypto

- ▶ Sub-repository packages <http://golang.org/x> by Go Project

[golang.org/x/tools](http://golang.org/x/tools), [golang.org/x/crypto](http://golang.org/x/crypto), [golang.org/x/sys](http://golang.org/x/sys), [golang.org/x/text](http://golang.org/x/text) ...

- ▶ Our Go packages

[github.com/lib/pq](https://github.com/lib/pq)

[github.com/aws/aws-sdk-go](https://github.com/aws/aws-sdk-go)

[github.com/philipsahli/client-go-wrapper](https://github.com/philipsahli/client-go-wrapper)

# DEPENDENCY MANAGEMENT

1

```
main.go:  
import "github.com/philipsahli/client-go-wrapper/pkg/wrapper"
```

IMPORT GO CODE FROM A GIT REPOSITORY

2

```
$ go mod init github.com/philipsahli/kubectl-images
```

ANALYSE IMPORTED PACKAGES AND PIN VERSIONS

3

```
$ go build cmd/kubectl-images
```

DOWNLOAD → COMPILE → PACK INTO BINARY

4

```
$ goreleaser
```

DEPLOY / DISTRIBUTE

## OUR JOURNEY

- ▶ Setting up a Gommunity
- ▶ 5-10 Gophers meets weekly for a Goffee
- ▶ More initiatives necessary to use Go as standard language
  - ▶ Guidelines
  - ▶ Education
- ▶ Keep talking to motivate people



## GO PROJECTS AT POST

- ▶ go-fw  
FW-Authenticator CLI
- ▶ terraform-provider-bitbucketserver  
TF Managed Bitbucket Server resources
- ▶ kafkactl  
Utility for Kafka Resources
- ▶ checkcert  
Utility for SSL certificates
- ▶ Namespace operator  
Onboard projects to K8s
- ▶ go-wait  
Binary waits for condition
- ▶ kafka-acl  
Ansible Module in Go
- ▶ ansible-overview  
inspired by Ansible Galaxy

## WANTED: DEVOPS ENGINEERS

- ▶ In einem **kollegialen** und kompetenten Team wirkst Du mit beim Aufbau und Management von **vollautomatisierten** Kubernetes Umgebungen. Diese laufen sowohl bei AWS als auch on premise.
- ▶ Du bist interessiert am Ansatz «**Infrastructure as Code**» und besitzt Erfahrung mit Automatisierung.
- ▶ Wir bieten Dir viel technischen Spielraum und **persönliche Entwicklungsmöglichkeiten**.

[post.ch/jobs](https://post.ch/jobs)

**DIE POST** 



## HOW TO LEARN GO

### 1. A tour of Go

<https://tour.golang.org/>

### 2. "Learn Go with Tests" by Chris James @quii

<https://quii.gitbook.io/learn-go-with-tests/>

### 3. Go: The complete Developer's Guide

<https://www.udemy.com/course/go-the-complete-developers-guide/>

### 4. GoDoc

<https://godoc.org/>

# HANDS-ON INTRODUCTION

- ▶ A co-worker needs to list frequently all Docker images used in a Kubernetes cluster.
- ▶ Help him out by providing him a binary powered by Go.



## HANDS-ON

Take client-go-wrapper as starting point, which already implements the method GetPods().

[github.com/philipsahli/client-go-wrapper](https://github.com/philipsahli/client-go-wrapper)

- ✓ Go 1.13.6
  - ✓ Visual Studio Code
    - Study on GoDoc
  - ✓ Go Extension ms-vscode.go
    - Initialize Go Module
  - ✓ Kubernetes Cluster
    - Write code
  - ✓ Goreleaser
    - Run & Build
  - ✓ Github Repository
    - Publish binary on Github
- <https://github.com/philipsahli/kubectl-images>



[post.ch/jobs](https://post.ch/jobs)

# READY TO GO?

[philip.sahli@post.ch](mailto:philip.sahli@post.ch)

[twitter.com/philipsahli](https://twitter.com/philipsahli)

[linkedin.com/in/philipsahli](https://linkedin.com/in/philipsahli)