Go In Grab

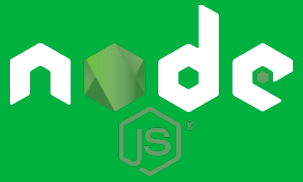
Golang的测试,持续集成以及部署策略

- By Zhao Chang, 一个在学习golang的程序猿

Grab是做什么的

1. Grab是东南亚最大的打车软件公司
2. 服务项目：
    a. GrabTaxi, GrabCar, GrabHitch
    b. GrabBike, GrabExpress ...
3. Grab的使命：
    a. 使出行更安全
    b. 使出行更便利

Grab从前的技术栈

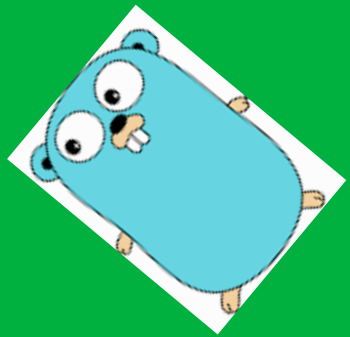


Grab现在的技术栈


RAILS


redis


Scalyr




MySQL


REDSHIFT


PostgreSQL


New
Relic


github


PHABRICATOR


DATADOG


docker

Grab未来可能增加的技术栈

Prometheus: 服务监控

数据流服务

事件响应，计算服务

Golang解决我们哪些问题：

1. 完善的工具链
2. 简洁的语言规范
3. 统一的技术栈
4. 方便的部署流程

Golang 使用前后的对比

- 使用前
  - 代码帮派林立
  - 存在很多难以debug的问题
  - 没有统一的面试，招聘，培训标准
- 使用后
  - Common code base
  - 在线机器数量大幅减少
  - 运维可以集中优化处理golang部署方案
  - 统一的新人培训bootcamp

Golang 使用中遇到的坑

- 第三方依赖库的更新
  - godeps
  - Vendor
- Buffered channels vs un-buffered channels
  - Time out and move on
  - 阻塞和非阻塞情况下的 select
- 要不要使用框架

举个栗子

```
resultCh := make(chan bool)
go func() {
    defer close(resultCh)
    resultCh <- funcReturnBoolean()
}()

select{
case result:=<-resultCh:
    println(result)
case <-time.After(1*time.Second):
}

```

再举个栗子

```go
package main

import "errors"

func test() {
    errCh := make(chan error, 1)
    outputCh := make(chan string)
    go func() {
        defer close(errCh)
        defer close(outputCh)
        errCh <- errors.New("error")
    }()
    select {
    case <-errCh:
    case <-outputCh:
        println("AHA!")
    }
}

func main() {
    for i := 0; i < 1000000; i++ {
        test()
    }
}
```

golang的代码规范

为什么要遵循代码规范：

代码规范是否限制了程序员的自由？

- C++, java
- python
- Golang

如何检查代码是否符合规范？

- golint go vet go errcheck goimports gofmt ...
- gometalinter

我们是工程师，我们应该负责测试吗？

工程师更应该负责测试：

- 软件工程师和传统工程师的区别
  - 一幢楼房 vs 一幢每天在变的楼房
  - 工程规范 vs 测试用例
  - 完工大吉 vs 没有尽头的更新
- 测试的益处？
  - 单元测试为了让函数的用户放心
  - 验收测试(UAT)为了让程序的用户放心
  - 为了持续集成

我们是如何测试golang程序的

stretchr / testify

rafaeljusto / redigomock

测试时替换函数指针

自动化的用户验收测试

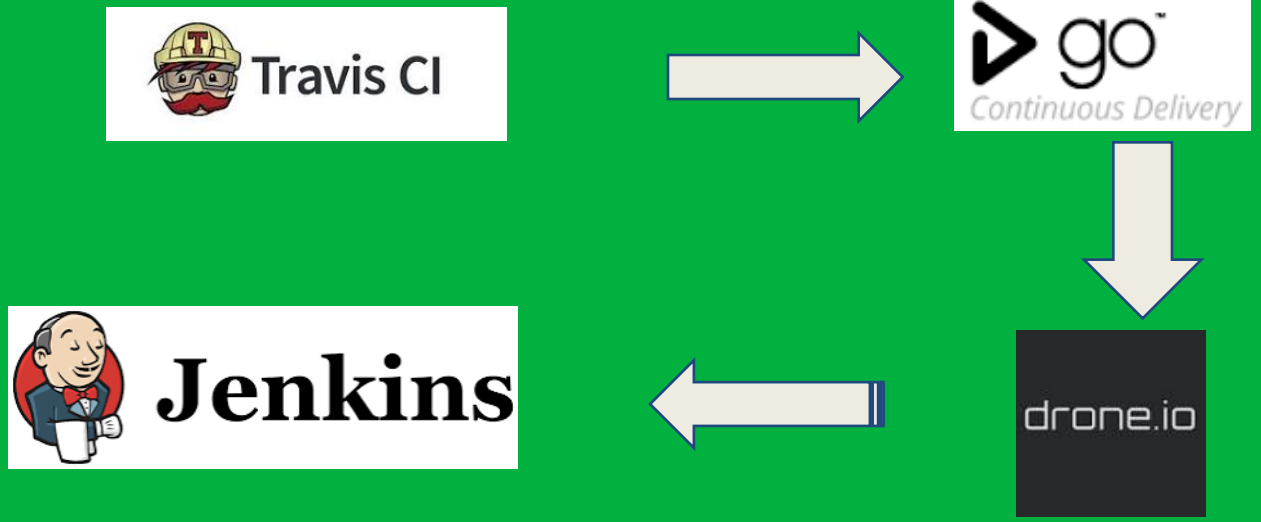
一个简单的例子

```go
package main

import "testing"

var realImpl = func() int {
    // do something for real
}

func TestSomething(t *testing.T) {
    defer func(org func() int) {
        realImpl = org
    }(realImpl)

    realImpl = func() int {
        return 0
    }
    //carry on with the testing
}
```

持续集成，持续集成

1. 好的工作流程：
    a. 让普通人表现的很牛
    b. 让牛人逆天
2. 糟糕的工作流程：
    a. 普通人想辞职
    b. 牛人已经辞职了

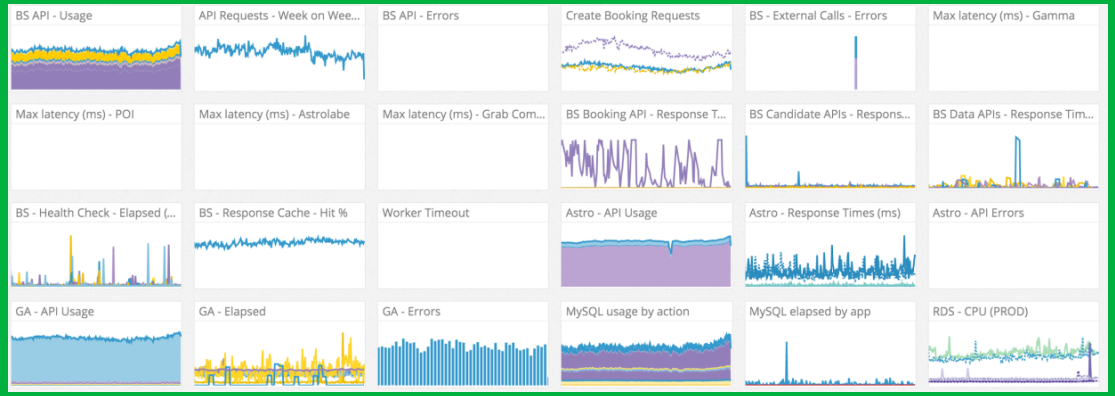
持续集成，持续集成（2）

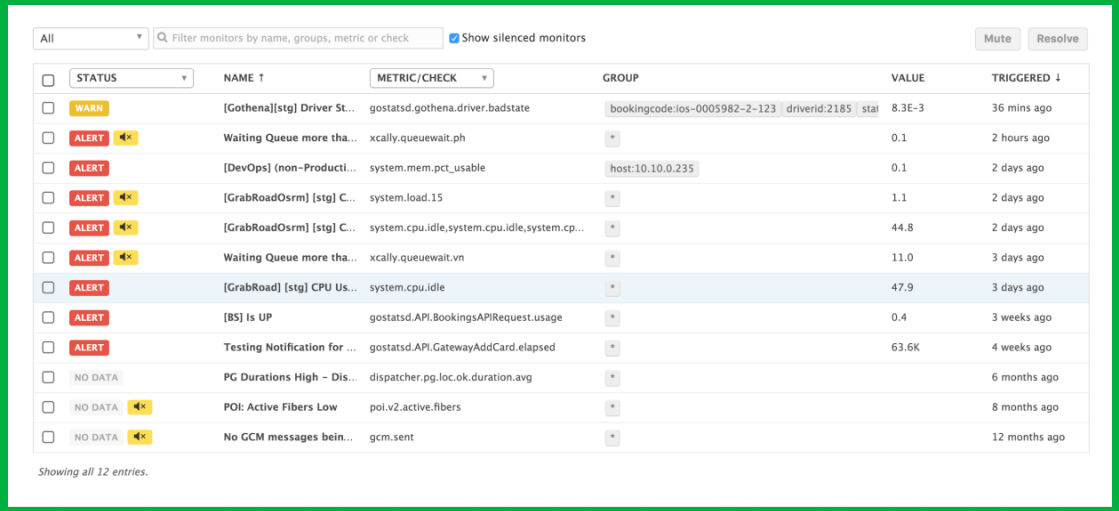


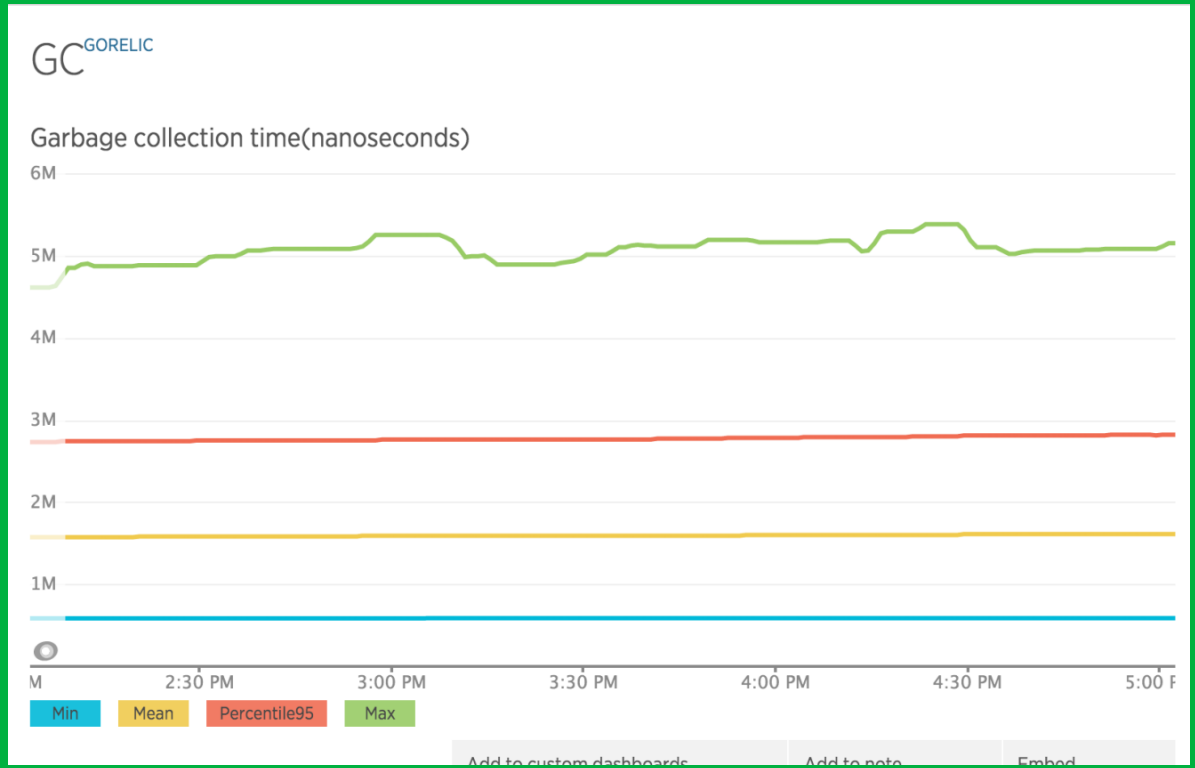
持续集成，持续集成（3）

持续部署 （staging）

Scalyr®


BS API - Usage
API Requests - Week on Wee...
BS API - Errors
Create Booking Requests
BS - External Calls - Errors
Max latency (ms) - Gamma
Max latency (ms) - POI
Max latency (ms) - Astrolabe
Max latency (ms) - Grab Com...
BS Booking API - Response T...
BS Candidate APIs - Respons...
BS Data APIs - Response Tim...
BS - Health Check - Elapsed (...
BS - Response Cache - Hit %
Worker Timeout
Astro - API Usage
Astro - Response Times (ms)
Astro - API Errors
GA - API Usage
GA - Elapsed
GA - Errors
MySQL usage by action
MySQL elapsed by app
RDS - CPU (PROD)


All
Filter monitors by name, groups, metric or check
Show silenced monitors
Mute
Resolve
STATUS
NAME ↑
METRIC/CHECK
GROUP
VALUE
TRIGGERED ↓
WARN [Gothena][stg] Driver St... gostatsd.gothena.driver.badstate bookingcode:ios-0005982-2-123 driverid:2185 stat 8.3E-3 36 mins ago
ALERT Waiting Queue more tha... xcally.queuewait.ph * 0.1 2 hours ago
ALERT [DevOps] (non-Producti... system.mem.pct_usable host:10.10.0.235 0.1 2 days ago
ALERT [GrabRoadOsrm] [stg] C... system.load.15 * 1.1 2 days ago
ALERT [GrabRoadOsrm] [stg] C... system.cpu.idle,system.cpu.idle,system.cp... * 44.8 2 days ago
ALERT Waiting Queue more tha... xcally.queuewait.vn * 11.0 3 days ago
ALERT [GrabRoad] [stg] CPU Us... system.cpu.idle * 47.9 3 days ago
ALERT [BS] Is UP gostatsd.API.BookingsAPIRequest.usage * 0.4 3 weeks ago
ALERT Testing Notification for ... gostatsd.API.GatewayAddCard.elapsed * 63.6K 4 weeks ago
NO DATA PG Durations High - Dis... dispatcher.pg.loc.ok.duration.avg * 6 months ago
NO DATA POI: Active Fibers Low poi.v2.active.fibers * 8 months ago
NO DATA No GCM messages bein... gcm.sent * 12 months ago
Showing all 12 entries.


GC GORELIC
Garbage collection time(nanoseconds)
6M
5M
4M
3M
2M
1M
M
2:30 PM
3:00 PM
3:30 PM
4:00 PM
4:30 PM
5:00 P
Min
Mean
Percentile95
Max

在结束之前稍微来点干货

一些Grab使用golang的经验

- go 编译大项目很慢
  - 试试 go build -i 或者 go install
- 资源和程序一起发布
  - https://github.com/jteeuwen/go-bindata
- debug 性能问题
  - pprof
  - https://github.com/prometheus/prometheus
  - https://github.com/tsenart/vegeta

我们使用golang的心路历程

- 从一个小服务到几十个后端服务
- 从分析log找问题到使用各种各种图形统计工具
- 从unit testing到UAT再到自己写测试覆盖报告
- 从一穷二白到参加各种gopher讨论组，组织golang bootcamp
- 从区区几个gopher到众志成城
- 持续集成和持续部署，我们是~~认真的！

Q&A