

Service Mesh Meetup #3 深圳站

Kubernetes、ServiceMesh、CI/CD 实践

杨文@JEX

2018.08.25

关于我

- JEX 技术VP
- 前小恩爱技术总监
- Gopher, 开源爱好者
- Go 夜读发起人
 - <https://github.com/developer-learning/night-reading-go>
 - <https://github.com/developer-learning/learning-kubernetes>

大纲

- 技术架构的演进
- 如何提升工程效率？DevOps
- CI/CD 实战
- ServiceMesh
- 踩过的一些坑
- Q&A

技术架构的演进

- 单体架构
 - 一个框架
 - 一个数据库
- 分模块整合架构（前后端分析）
 - 不同的框架或业务模块
 - 多种数据源
- 微服务架构
 - 各种语言、各种框架或子系统
 - 各种数据源
- ServiceMesh

	优点	缺点
单一架构	<ul style="list-style-type: none"> ➢ 简单 ➢ 实现快 	<ul style="list-style-type: none"> ➢ 高内聚 ➢ 系统复杂 ➢ 可扩展性差 ➢ 维护成本高
垂直架构	<ul style="list-style-type: none"> ➢ 已经演化成较多 mvc 架构 ➢ 前后端逻辑分离 ➢ 有一定模块化 ➢ 负载均衡 	<ul style="list-style-type: none"> ➢ 所有功能依然部署在同一个进程中 ➢ 应用越多，交互也越耦合
SOA 结构	<ul style="list-style-type: none"> ➢ 松耦合 ➢ 不同业务抽出单独服务 ➢ 前后端逻辑分离 	<ul style="list-style-type: none"> ➢ 服务划分粒度较粗 ➢ 中央调度中心管理服务
微服务	<ul style="list-style-type: none"> ➢ 独立部署 ➢ 服务更原子 ➢ 松耦合 ➢ 快速拓展 ➢ 技术选型灵活 	<ul style="list-style-type: none"> ➢ 部署复杂 ➢ 测试复杂 ➢ 消息通讯机制需要更完善

一般的开发流程

- 1. 开启一个新的 feature ；
- 2. Developer 从 develop 分支新建一个 feature/new_branch 来做特定 feature 的开发 ；
- 3. 开发并自测后，提交 merge request (MR) 请求合并到 develop 分支 ；（执行单元测试，测试状态呈现 MR 中）
- 4. Reviewer 对 MR 进行 code review ，批准合并之后，feature/new_branch 会合并到 develop ；
- 5. 部署负责人将 develop 分支代码部署到测试环境，然后再通知 QA 测试 ；（脚本或者人工）

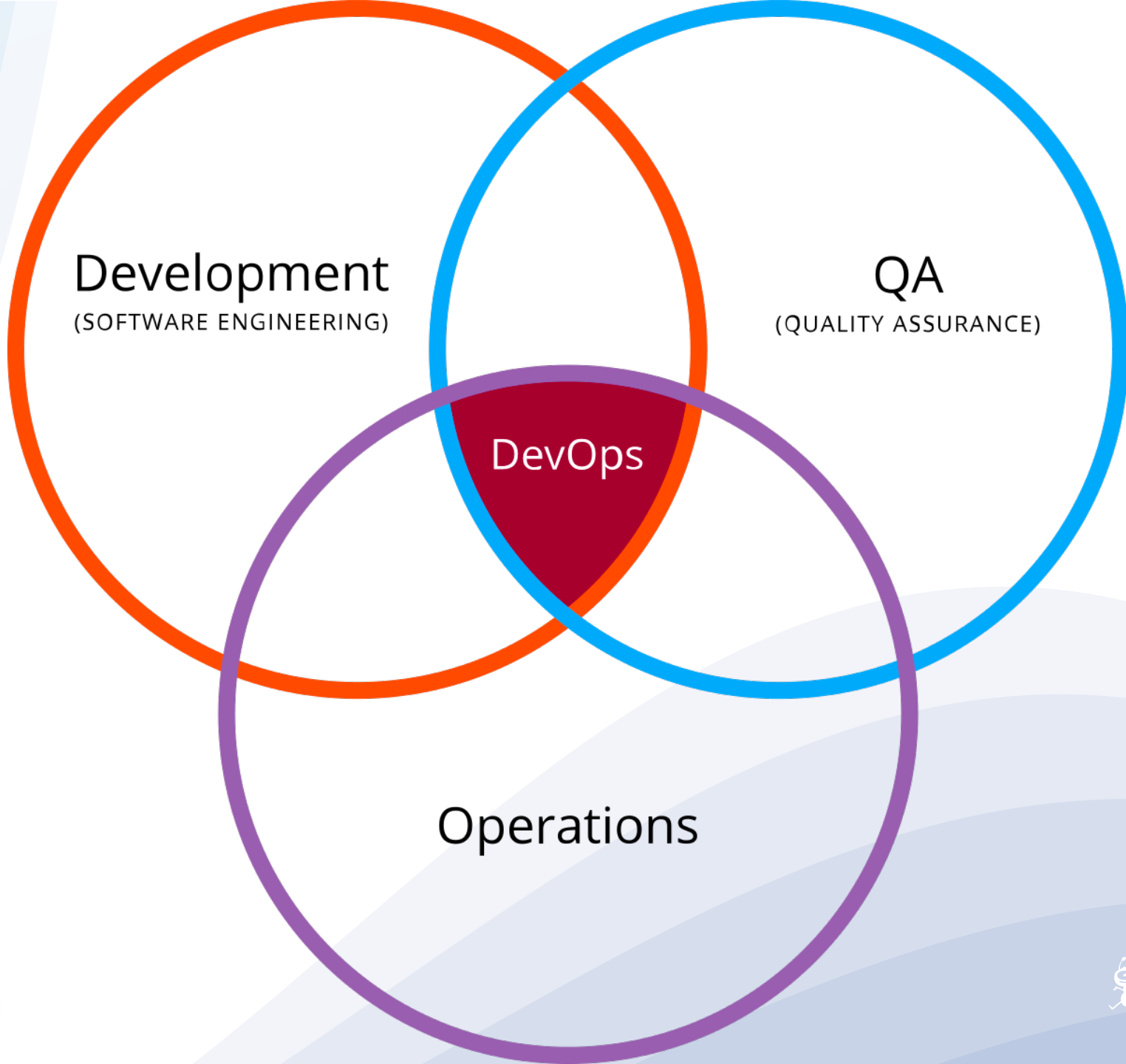
有什么问题？

- 效率低
 - 没有代码检查；
 - 没有自动化测试（包括单元测试）；
- 沟通成本高
 - 开发需要通知负责人、测试、产品等；（而且是每次构建/部署都需要）
- 依赖多

较好的开发流程（流程化、自动化）

- 1. 开启一个新的 feature ；
- 2. Developer 从 develop 分支新建一个 feature/new_branch 来做特定 feature 的开发 ；
- 3. 开发完成后，提交 merge request (MR) 请求合并到 develop 分支 ；
- 4. MR 触发 Jenkins, Jenkins/Drone 触发 Sonar 代码质量检测系统 ；
- 5. Sonar 将 report 和 issue 以 comments 的方式写到 Gitlab MR 中 ；
- 6. Developer 对 MR 进行反复修复直至通过 Sonar 的分析 ；
- 7. Reviewer 对 MR 进行 code review ， 批准合并之后， feature/new_branch 会合并到 develop ；
- 8. Merge 触发 Jenkins/Drone 自动构建 ；
- 9. 构建成功就执行你定义的工作流：打包镜像， 触发 deploy 以及其他后续的 Automation Testing 等流程 ；
- 10 . Drone 通知工作流程情况给开发/或者交付 QA 测试 ；

如何提升工程效率？

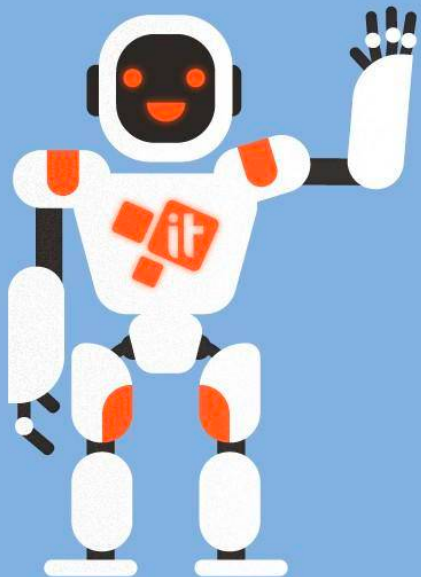


Development
(SOFTWARE ENGINEERING)

QA
(QUALITY ASSURANCE)

DevOps

Operations



46x
more frequent code deployment

440x
shorter time from commit to deploy

96x
faster TTR (time to recover)

5x
lower change failure rate

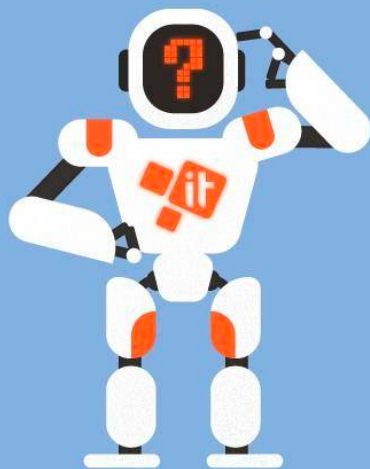
**High-performing teams love automation.
They do less manual work and have automated:**

33% more of their configuration management

30% more of their deployments

27% more of their change approval processes

27% more of their testing



2017 年 DevOps 现状调查报告

调查问题

高效能组织

中效能组织

低效能组织

发布频率

对于您处理的主要应用程序或服务，您的组织多长时间部署代码？

按需（每天多部署）

每周一次 & 每月一次

每周一次 & 每月一次

变更延迟

对于您处理的主要应用程序或服务，您的更改前置时间（即从代码提交到代码在生产中成功运行需要多长时间）？

< 1 小时

一周至一个月

一周至一个月

平均恢复时间 (MTTR)

对于您工作的主要应用程序或服务，发生服务事件（例如意外中断，服务受损）时恢复服务通常需要多长时间？

< 1 小时

少于一天

一天至一周

变更失败率

对于您工作的主要应用程序或服务还有哪些比例的结果呢？在降级服务中或随后需要修复（例如，导致服务损坏，服务中断，需要修补程序，回滚，修复前进，修补程

0 — 15%

0 — 15%

31 — 45%

DevOps

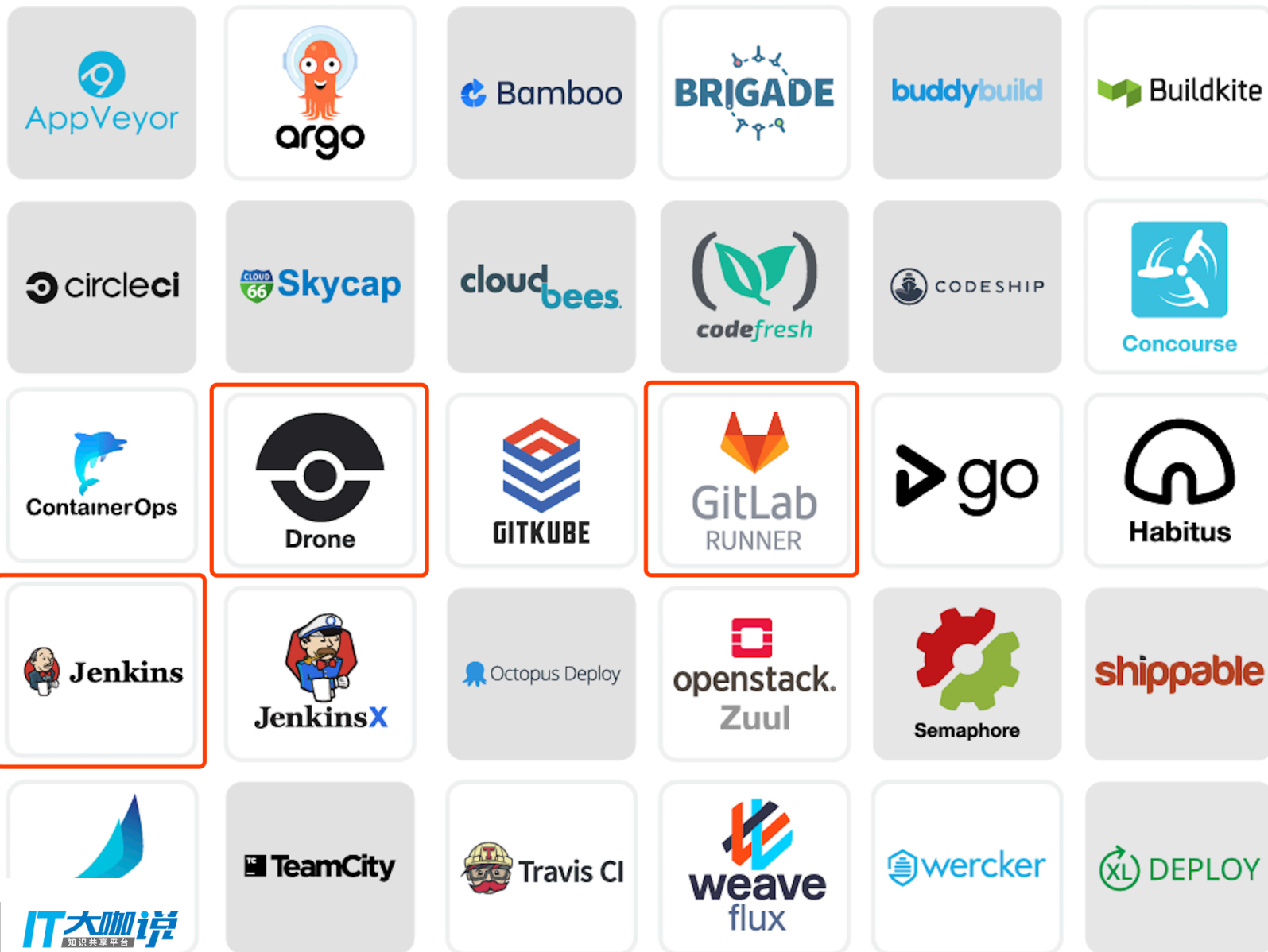
DevOps 不是一种新工具；

DevOps 不是一种新团队；

DevOps 不是一种新角色；

DevOps 是一种文化：一切自动化，工具化，规范化；

Continuous Integration / Continuous Delivery (CI/CD)



选择哪个 CI/CD 工具？



Jenkins



GitLab CI



Drone

CI/CD 的对比

工具	优势	不足
jenkins	社区庞大，插件丰富，文档丰富。	复杂，启动慢，运行慢，耗资源，需要专人维护管理，只能用 Java 写插件。
gitlab-ci	-	只支持 gitlab，yaml 不能扩展。
drone	轻量级，启动快，运行快，资源占用少，一个 .drone.yml 搞定，Go 开发，原生支持 Docker，任何语言都可以写插件，可以在本机运行测试。	插件不多，资源没有 jenkins 多。

Docker stats 查看 Docker 服务所占用的CPU和内存开销

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
2ef260b353b8	jenkins	1.32%	1.056GiB / 7.786GiB	13.57%	57.4MB / 2.35MB	72.2MB / 9.58MB	45
afa98b20e561	sonarqube	1.66%	1.703GiB / 7.786GiB	21.88%	73.3kB / 835kB	299MB / 708MB	187
6074fe0009c9	gitlab	10.69%	2.564GiB / 7.786GiB	32.93%	223kB / 865kB	267MB / 938kB	321
22cc7042f755	drone-mysql_drone-agent-1_1	0.00%	4.508MiB / 7.786GiB	0.06%	82.1kB / 65kB	209kB / 0B	9
d45625df93a3	drone-mysql_drone-agent-2_1	0.00%	3.898MiB / 7.786GiB	0.05%	225kB / 195kB	8.48MB / 0B	10
0f5acd9d75f8	drone-mysql_drone-server_1	0.00%	6.789MiB / 7.786GiB	0.09%	1.75MB / 1.14MB	16.7MB / 0B	10
dd3f0e8294fa	drone-mysql_mysql-server_1	0.08%	182.2MiB / 7.786GiB	2.29%	700kB / 1.45MB	29.2MB / 8.19kB	30
34fbf6057fa9	registry	0.00%	3.902MiB / 7.786GiB	0.05%	9.39kB / 0B	13.3MB / 0B	9

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
2ef260b353b8	jenkins	89.56%	1.073GiB / 7.786GiB	13.78%	58MB / 3.1MB	88.7MB / 19MB	188
afa98b20e561	sonarqube	1.62%	1.712GiB / 7.786GiB	21.99%	76.7kB / 835kB	299MB / 708MB	188
6074fe0009c9	gitlab	7.68%	2.552GiB / 7.786GiB	32.78%	248kB / 920kB	267MB / 938kB	322
22cc7042f755	drone-mysql_drone-agent-1_1	0.00%	4.453MiB / 7.786GiB	0.06%	84.6kB / 67.1kB	209kB / 0B	9
d45625df93a3	drone-mysql_drone-agent-2_1	0.00%	3.844MiB / 7.786GiB	0.05%	227kB / 197kB	8.48MB / 0B	10
0f5acd9d75f8	drone-mysql_drone-server_1	0.00%	6.766MiB / 7.786GiB	0.08%	1.75MB / 1.14MB	16.7MB / 0B	10
dd3f0e8294fa	drone-mysql_mysql-server_1	0.05%	182MiB / 7.786GiB	2.28%	700kB / 1.45MB	29.2MB / 8.19kB	30
34fbf6057fa9	registry	0.00%	3.902MiB / 7.786GiB	0.05%	12.9kB / 0B	13.3MB / 0B	9

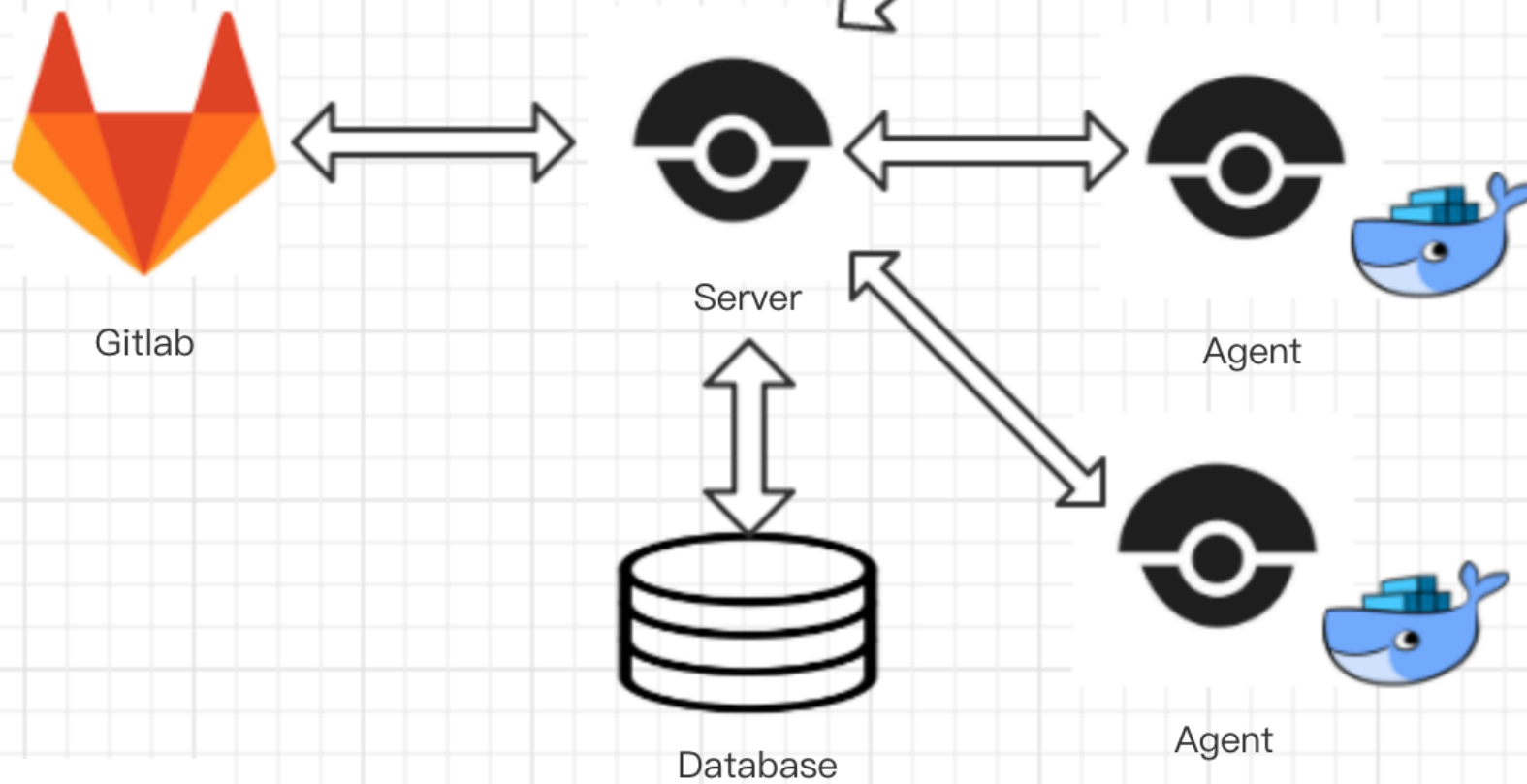
CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
bf35ddf48322	0_2356643662735021159_step_1	4.55%	36.92MiB / 7.786GiB	0.46%	19MB / 521kB	750kB / 74.7MB	20
a63bd7a4a37a	0_8118004982049157478_step_1	4.39%	42.42MiB / 7.786GiB	0.53%	18.4MB / 548kB	20.6MB / 47.3MB	18
2ef260b353b8	jenkins	0.19%	1.038GiB / 7.786GiB	13.33%	71.3MB / 6.1MB	91.3MB / 19.3MB	45
afa98b20e561	sonarqube	2.01%	1.782GiB / 7.786GiB	22.88%	719kB / 7.48MB	310MB / 708MB	197
6074fe0009c9	gitlab	3.76%	2.645GiB / 7.786GiB	33.97%	601kB / 2.43MB	281MB / 1.21MB	329
22cc7042f755	drone-mysql_drone-agent-1_1	0.00%	4.676MiB / 7.786GiB	0.06%	573kB / 411kB	360kB / 0B	11
d45625df93a3	drone-mysql_drone-agent-2_1	0.00%	5.746MiB / 7.786GiB	0.07%	1.2MB / 1.3MB	8.48MB / 0B	11
0f5acd9d75f8	drone-mysql_drone-server_1	0.00%	10.34MiB / 7.786GiB	0.13%	11.2MB / 7.93MB	17.1MB / 0B	10
dd3f0e8294fa	drone-mysql_mysql-server_1	0.09%	184.3MiB / 7.786GiB	2.31%	4.75MB / 9.34MB	29.8MB / 8.19kB	33
34fbf6057fa9	registry	0.00%	3.965MiB / 7.786GiB	0.05%	39.4kB / 0B	13.8MB / 0B	9

Drone

Drone











- 一款使用 Go 开发，基于容器技术的 CI/CD 系统，能够单独部署，支持几乎所有的 Git 平台（Github, Gitlab, Bitbucket, Gogs, Gitea 等）。
- 它的特点：
 - 一个 `.drone.yml` 搞定（简单）
 - 原生 Docker 支持: 任何步骤都是在 Docker 内执行的（环境隔离）
 - Pipeline as code
 - 丰富的插件支持
 - 极简的Web管理界面
- 其他：
 - Control PR build through comments([#2056](#)) —> [bradrydzewski](#) removed this from **To Do** in **Version 0.9** on Jun 30
 - ...

Drone 架构



Drone 社区 <https://discourse.drone.io/>

all categories ▾ **Categories** Latest Top + New Topic

Category	Topics	Latest
General Discussion General discussion, support, questions and answers	40 / month	 Build logs via REST 1 17h
Development Topics related to Drone development	2 / month	 How to build node projects? 5 20h
Announcements Topics announcing releases and important changes	3	  Drone Clone Error with Gitea (Duplicate) 2 1d
		 Spinning up a drone for integration testing 1 1d
		 Skip the clone step 2 2d
		 U I need help setting up drone.io 0 3d
		 Alternate names for secrets 1 4d
		 Showing the failed build step in drone-slack 0 4d
		 Couldn't find a way to use Nexus 1 5d

Drone 负责人



Drone 的安装

- docker pull drone/drone:0.8.6
- Docker pull drone/agent:0.8.6
- Docker run or Docker compose

```
version: '2'
```

```
services:
```

```
  drone-server:
```

```
    image: drone/drone:0.8
```

```
    ports:
```

```
      - 80:8000
```

```
      - 9000
```

```
    volumes:
```

```
      - /var/lib/drone:/var/lib/drone/
```

```
    restart: always
```

```
    environment:
```

```
      - DRONE_OPEN=true
```

```
      - DRONE_HOST=${DRONE_HOST}
```

```
      - DRONE_GITHUB=true
```

```
      - DRONE_GITHUB_CLIENT=${DRONE_GITHUB_CLIENT}
```

```
      - DRONE_GITHUB_SECRET=${DRONE_GITHUB_SECRET}
```

```
      - DRONE_SECRET=${DRONE_SECRET}
```

```
  drone-agent:
```

```
    image: drone/agent:0.8
```

```
    command: agent
```

```
    restart: always
```

```
    depends_on:
```

```
      - drone-server
```

```
    volumes:
```

```
      - /var/run/docker.sock:/var/run/docker.sock
```

```
    environment:
```

```
      - DRONE_SERVER=drone-server:9000
```

```
      - DRONE_SECRET=${DRONE_SECRET}
```

其他特性

- 支持Cache (vendor, node_modules) : 方便下次更快执行
- 支持触发其他CI服务 :
 - Jenkins(plugins/drone-jenkins)
 - Gitlab-CI(plugins/drone-gitlab-ci)
 - Drone (plugins/drone-downstream)
- 支持自定义插件 (你可以自己实现自己所需的插件)
- 本机测试 [.drone.yml](#) : drone exec

API 支持

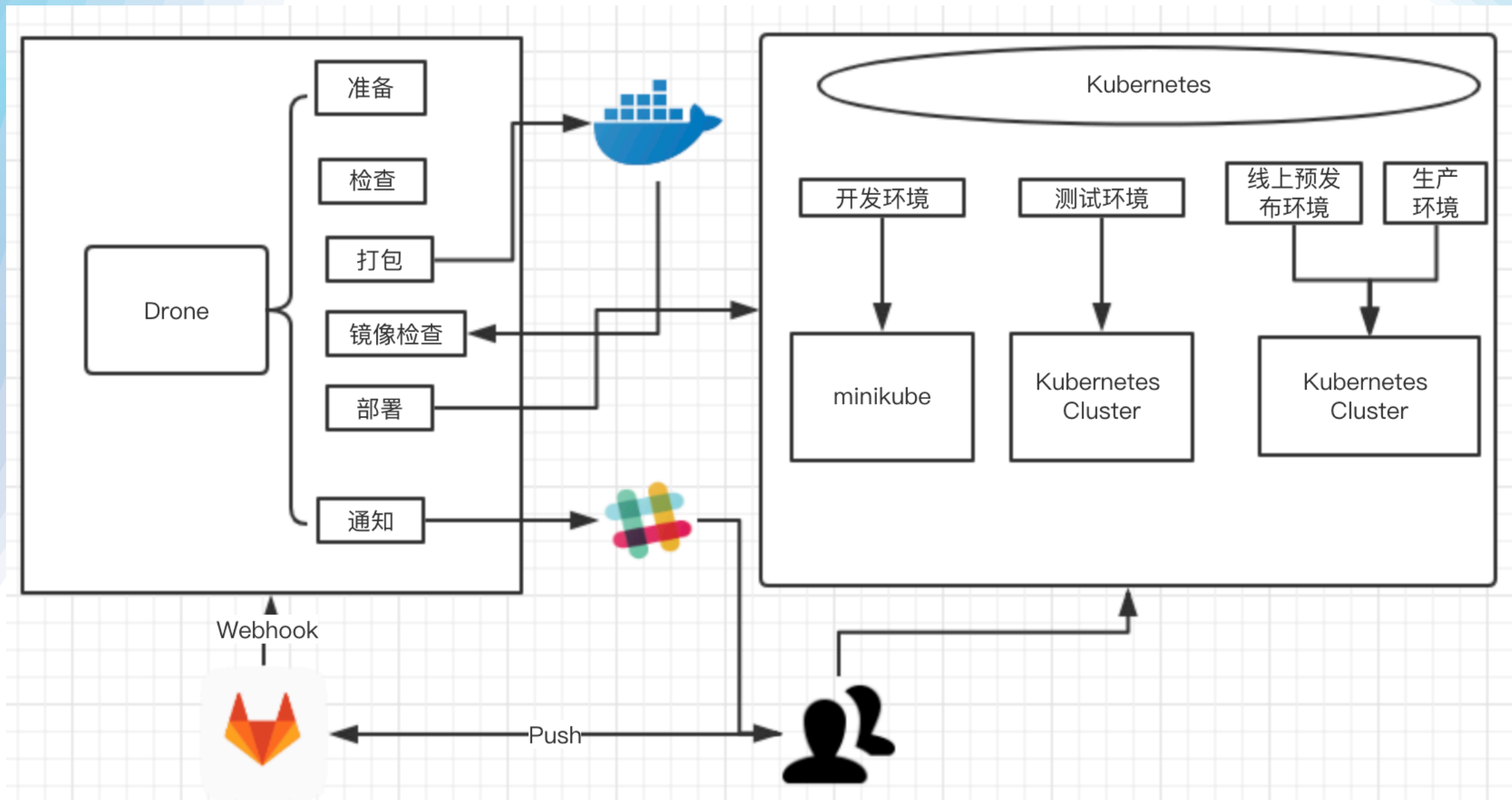
API Overview

Drone provides a comprehensive remote API for interacting with the Drone server. This section of the documents provides instructions for authenticating and using the remote API.

Libraries

Drone provides the following official libraries for integrating with the remote API:

Language	Repository
Go	https://github.com/drone/drone-go
Node	https://github.com/drone/drone-node
Python	https://github.com/drone/drone-python
	https://github.com/drone/drone-ruby



.drone.yml

```
1 workspace:
2   base: /go
3   path: src/maiyang.me/developer-learning/gcd
4
5 pipeline:
6   setup:
7     image: docker:git
8     commands:
9       - git clone xxx xxx
10
11   build:
12     image: golang:1.10.3
13     commands:
14       - go get
15       - go test ./...
16       - GOARCH=amd64 CGO_ENABLED=0 GOOS=linux go build
17
18   build_image:
19     image: plugins/docker
20     repo: yangwen13/gcd
21     dockerfile: Dockerfile
22     secrets: [docker_username, docker_password]
23     tags: [0.3]
24
25   run_server:
26     image: yangwen13/gcd:0.3
27     detach: true
28
29   deploy:
30     image: yangwen13/drone-kubernetes
31     kubernetes_template: k8s-gcd.yaml
32     kubernetes_namespace: default
33     secrets: [kubernetes_server, kubernetes_cert, kubernetes_token]
34
35   slack:
36     image: plugins/slack
37     webhook: your slack webhook url
38     when:
39       status: [success, failure]
40     icon_url: https://unsplash.it/256/256/?random
41     image_url: https://unsplash.it/256/256/?random
42     template: >
43       {{#success build.status}}
44         build {{build.number}} succeeded. Good Job.
45       {{else}}
46         build {{build.number}} failed. Fix me please.
47       {{/success}}
```

代码质量检测 SonarQube

参考资料：<https://github.com/developer-learning/night-reading-go/blob/master/articles/sonarqube-for-golang/2018-07-22-sonarqube-for-golang.md>

Drone 和 Jenkins 联调测试方法

本地即可测试 drone 上是否可以调用成功：

```
>docker run --rm \  
-e JENKINS_URL=<your jenkins URL>  
-e JENKINS_USER=<your jenkins username>  
-e JENKINS_TOKEN=<your jenkins APIToken>  
-e JENKINS_JOB=<your jenkins job name>  
appleboy/drone-jenkins
```

```

Started by user maiyang
Building in workspace /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning
Cloning the remote Git repository
Cloning repository http://192.168.0.56/developer-learning/gcd
> git init /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning/gcd # timeout=10
Fetching upstream changes from http://192.168.0.56/developer-learning/gcd
> git --version # timeout=10
using GIT_ASKPASS to set credentials
> git fetch --tags --progress http://192.168.0.56/developer-learning/gcd +refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url http://192.168.0.56/developer-learning/gcd # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url http://192.168.0.56/developer-learning/gcd # timeout=10
Fetching upstream changes from http://192.168.0.56/developer-learning/gcd
using GIT_ASKPASS to set credentials
> git fetch --tags --progress http://192.168.0.56/developer-learning/gcd +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 714bbb7917124439d55b61292fdce616401c378d (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 714bbb7917124439d55b61292fdce616401c378d
Commit message: "update"
> git rev-list --no-walk 714bbb7917124439d55b61292fdce616401c378d # timeout=10
Cloning the remote Git repository
Cloning repository http://192.168.0.56/developer-learning/pb
> git init /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning/pb # timeout=10
Fetching upstream changes from http://192.168.0.56/developer-learning/pb
> git --version # timeout=10
using GIT_ASKPASS to set credentials
> git fetch --tags --progress http://192.168.0.56/developer-learning/pb +refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url http://192.168.0.56/developer-learning/pb # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url http://192.168.0.56/developer-learning/pb # timeout=10
Fetching upstream changes from http://192.168.0.56/developer-learning/pb
using GIT_ASKPASS to set credentials
> git fetch --tags --progress http://192.168.0.56/developer-learning/pb +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision b2b0fb045fb40b6d94d986fdd36378c42330772f (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f b2b0fb045fb40b6d94d986fdd36378c42330772f
Commit message: "add Version props"
First time build. Skipping changelog.
No emails were triggered.
[developer-learning] $ /bin/bash /tmp/jenkins5260089194894508703.sh

```

```

INFO: Load metrics repository (done) | time=106ms
INFO: Project key: 192.168.0.56:gcd
INFO: Project base dir: /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning
INFO: ----- Scan gcd
INFO: Load server rules
INFO: Load server rules (done) | time=675ms
INFO: Base dir: /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning
INFO: Working dir: /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning/.scannerwork
INFO: Source paths: .
INFO: Test paths: .
INFO: Source encoding: UTF-8, default locale: en
INFO: Index files
INFO: Excluded sources:
INFO: **/_test.go
INFO: **/vendor/**
INFO: **/_test.go
INFO: Included tests:
INFO: **/_test.go
INFO: Excluded tests:
INFO: **/vendor/**
INFO: 16 files indexed
INFO: 16 files ignored because of inclusion/exclusion patterns
INFO: Quality profile for go: Sonar way
INFO: Sensor SonarGo [go]
INFO: Load coverage report from '/var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning/coverage.out'
INFO: Sensor SonarGo [go] (done) | time=690ms
INFO: Sensor Go Unit Test Report [go]
INFO: Sensor Go Unit Test Report [go] (done) | time=13ms
INFO: Sensor Import of go vet issues [go]
INFO: GoVetReportSensor: Importing /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning/govet-report.out
INFO: Sensor Import of go vet issues [go] (done) | time=5ms
INFO: Sensor Import of GoLint issues [go]
INFO: GoLintReportSensor: Importing /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning/golint-report.out
INFO: Sensor Import of GoLint issues [go] (done) | time=27ms
INFO: Sensor Import of GoMetaLinter issues [go]
INFO: GoMetaLinterReportSensor: Importing /var/jenkins_home/workspace/go/src/gogs.maiyang.me/developer-learning/gometalinter-report.out
INFO: Sensor Import of GoMetaLinter issues [go] (done) | time=6ms
INFO: Sensor SonarJavaXmlFileSensor [java]
INFO: Sensor SonarJavaXmlFileSensor [java] (done) | time=0ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=22ms
INFO: Sensor CPD Block Indexer
INFO: Sensor CPD Block Indexer (done) | time=0ms
INFO: SCM provider for this project is: git
INFO: 3 files to be analyzed
INFO: 1/3 files analyzed
WARN: Missing blame information for the following files:
WARN: * gcd/main.go
WARN: * pb/gcd.pb.go
WARN: This may lead to missing/broken features in SonarQube
INFO: Calculating CPD for 3 files
INFO: CPD calculation finished
INFO: Analysis report generated in 183ms, dir size=62 KB
INFO: Analysis reports compressed in 676ms, zip size=21 KB
INFO: Analysis report uploaded in 4193ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://192.168.0.56:8081/dashboard?id=192.168.0.56%3Agcd
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://192.168.0.56:8081/api/ce/task?id=AWVmBljF4SgXLzUr5d8n
INFO: Task total time: 12.319 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 15.894s
INFO: Final Memory: 15M/224M
INFO: -----
Email was triggered for: Always
Sending email for trigger: Always
An attempt to send an e-mail to empty list of recipients, ignored.
Finished: SUCCESS

```

☆ gcd

Passed

Last analysis: August 23, 2018, 5:03 PM

0 **A**
🐛 Bugs

0 **A**
🔒 Vulnerabilities

30 **A**
☢️ Code Smells

○ 0.0%
Coverage

● 89.6%
Duplications

365 **XS**
Go

Changes **1** Pipelines **1**

Showing **1** changed file ▾ with **3 additions** and **0 deletions**

```
▼ main.go
```

...	...	@@ -20,4 +20,7 @@ func main() {
20	20	var nilM *math.Math
21	21	nilM = nilM
22	22	fmt.Printf("result: %d\n", m.Divide())
	23	+
	24	+
	25	+
		var nilMap map[string]string
		nilMap["abc"]="def"
23	26	}



sonarqube @sonar · right now

SonarQube analysis reported 4 issues

▲ 3 major



Search ...

developer-learning/gcd ✓

🕒 2 minutes ago
🕒 1 minute, 47 seconds

developer-learning/api ✓

🕒 7 minutes ago
🕒 1 minute, 58 seconds

Successful

update

🕒 2 minutes ago
🕒 1 minute, 47 seconds

φ 4e174fd1f9 [🔗](#)
🔗 master

clone	00:02	✓
setup	00:03	✓
test	00:41	✓
build	00:12	✓
build_image	00:38	✓
run_server	00:10	✓
deploy	00:01	✓
slack	00:04	✓

Failure

custom template send

🕒 3 minutes ago
🕒 57 seconds

φ 521bac0d88 [🔗](#)
🔗 master

clone	00:02	✓
setup	00:03	✓
test	00:46	✗
build		⊖
build_image		⊖
run_server		⊖
deploy		⊖
slack	00:04	✓



DroneCI APP 4:18 PM

success [developer-learning/api#6eb58623](#) (master) by Administrator

success [developer-learning/gcd#21007964](#) (master) by Administrator



DroneCI APP 4:29 PM

success [developer-learning/gcd#08656be9](#) (master) by Administrator



DroneCI APP 5:04 PM

success [developer-learning/gcd#092ed225](#) (master) by Administrator



DroneCI APP 5:21 PM

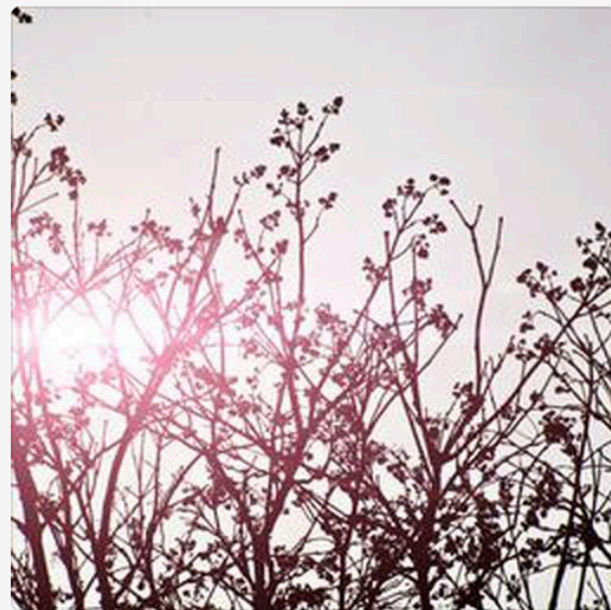
success [developer-learning/api#efbd6655](#) (master) by Administrator

success [developer-learning/gcd#4e174fd1](#) (master) by Administrator



DroneCI APP 7:15 PM

build 6 failed. Fix me please. (8 kB) ▾



.drone.yml 自定义

Talk is cheap,
Show me the code!

- 当使用一个客户端实例和多个后端实例进行部署时，所有的调用仅路由到单个后端实例。当部署第二个客户端时，它可能被路由到另一个后端实例。这不是所需的那种负载均衡，因为它不允许独立地扩展客户端和服务端。当客户端实例比服务器实例少时，一些服务器实例将处于空闲状态，所以 Kubernetes Service 不太适合 gRPC 负载均衡。—摘自：<http://www.k8smeetup.com/article/N1yW3gPNX>

gRPC LoadBalancing

- gRPC 负载均衡的文档：
<https://github.com/grpc/grpc/blob/master/doc/load-balancing.md>
- 解决办法：<https://github.com/jtattermusch/grpc-loadbalancing-kubernetes-examples>

什么是 ServiceMesh

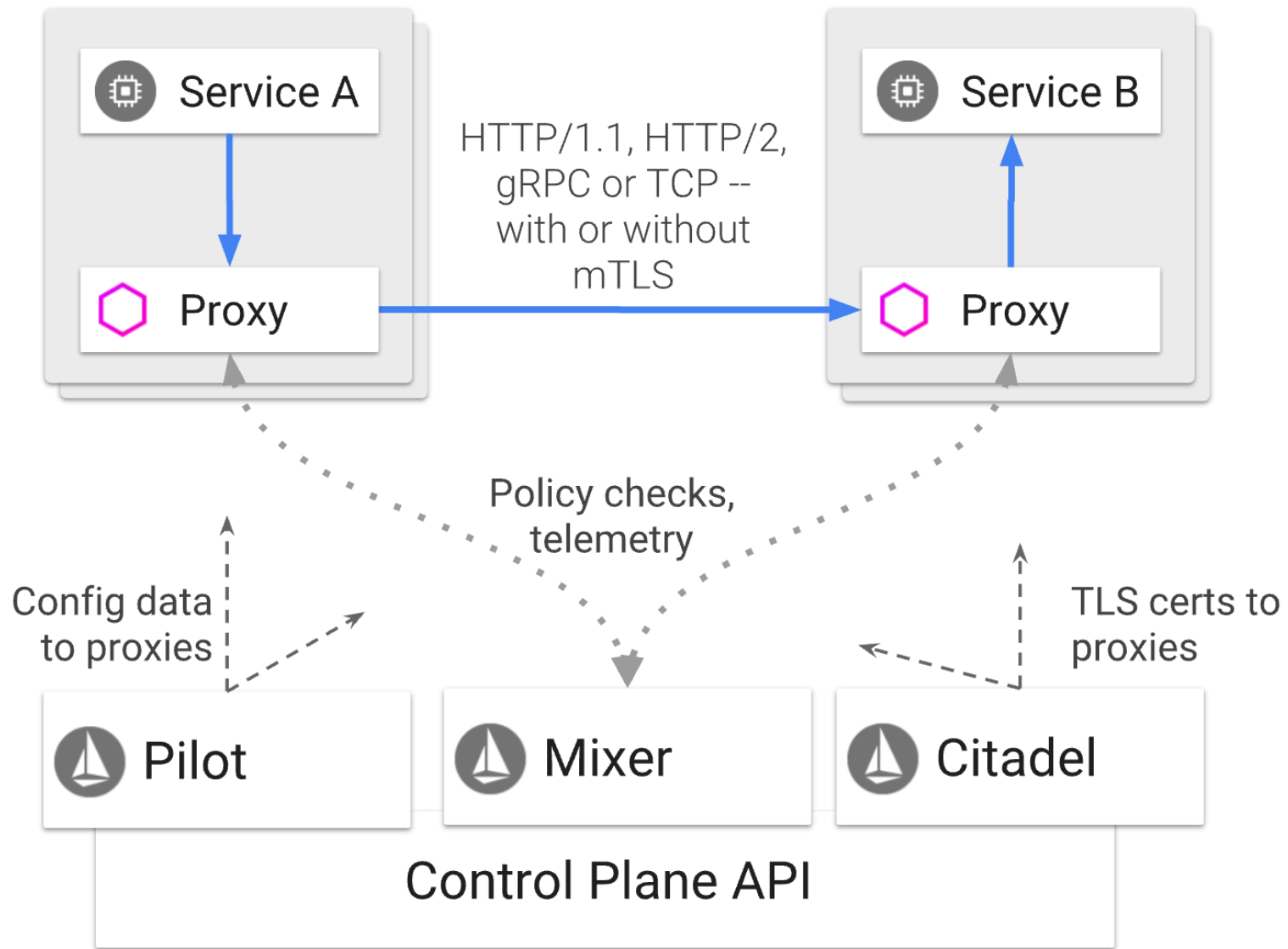
- 服务网格（Service Mesh）这个术语通常用于描述构成这些应用程序的微服务网络以及应用之间的交互。随着规模和复杂性的增长，服务网格越来越难以理解和管理。
- 它的需求包括服务发现、负载均衡、故障恢复、指标收集和监控以及通常更加复杂的运维需求，例如 A/B 测试、金丝雀发布、限流、访问控制和端到端认证等。

什么是 Istio

- Istio 提供一种简单的方式来为已部署的服务建立网络，该网络具有负载均衡、服务间认证、监控等功能，而不需要对服务的代码做任何改动。
- 想要让服务支持 Istio，只需要在您的环境中部署一个特殊的 sidecar 代理，使用 Istio 控制平面功能配置和管理代理，拦截微服务之间的所有网络通信。

Istio 的特点

- HTTP、gRPC、WebSocket 和 TCP 流量的自动负载均衡。
- 通过丰富的路由规则、重试、故障转移和故障注入，可以对流量行为进行细粒度控制。
- 可插入的策略层和配置 API，支持访问控制、速率限制和配额。
- 对出入集群入口和出口中所有流量的自动度量指标、日志记录和跟踪。
- 通过强大的基于身份的验证和授权，在集群中实现安全的服务间通信。



坑

- Gitlab
 - external_url
 - Outbound requests(Allow requests to the local network from hooks and services.)
- Drone
 - Drone 0.8 在 kubernetes 上支持的不好 (RPC error) ;
 - .drone.yml not .drone.yaml
 - Failed to activate your repository (检查网络, 以及代理配置)
 - plugins:
 - drone-kubernetes, 需要将 kubernetes token base64 解码(否则报错: Unauthorized)
- Jenkins
 - 默认是开启全局安全和防止跨站点请求伪造 (这个在测试时你可能需要关闭)
- Kubernetes(Minikube)
 - `—bootstrapper=localkube`
- Istio
- **GreatWall** (<http://blog.samemoment.com/articles/kubernetes/>)
 - 应该是给我造成最多最大的问题
 - SS

未来展望

CNCF (Cloud Native Computing Foundation)
云原生计算基金会

App Definition and Development

Database and Data Warehouse: Vitess, ArangoDB, BIGCHAM, CarbonData, Cockroach Labs, Couchbase, CRATE.IO, druid, FoundationDB, InfluxDB, Iguazio, Infrecon, MariaDB, MEMSQL, SQL Server, MongoDB, MySQL, Neo4j, Oracle, OrientDB, PostgreSQL, Presto, Qubole, Redis, RethinkDB, Scylla, Snowflake, Software.com, TIDB, Vertica, Yugabyte.

Streaming: CNCF Incubating, cloudevents, Amazon Kinesis, CNCF Sandbox, APEX.

Source Code Management: Bitbucket, GitHub, GitLab, Open Source Services.

Application Definition and Image Build: CNCF Sandbox, Apache Brooklyn, Bitnami, DRAFT, Habitat, HELM, Kaniko, Koodinot, Kubewirt, Minkube, OpenAPI Initiative, Packer, Google Cloud Shell, YIPPEO.

Continuous Integration / Continuous Delivery (CI/CD): Appveyor, Argo, Bamboo, BRIGADE, Buddybuild, Buildkite, CircleCI, Skycap, CloudBees, Codefresh, Concourse, ContainerOps, Drone, GitLab Runner, Go, Habitus, Jenkins, JenkinsX, OpenStack Zulu, Semaphore, Shipable, Sprinkler, TeamCity, Travis CI, Weave Flux, Wercker, Deploy.

Orchestration & Management

Scheduling & Orchestration: Kubernetes, Mesos, Nomad.

Coordination & Service Discovery: CoreDNS, Consul, etcd, ContainerPilot, Netflix Eureka, Airbnb Smartstack.

Service Management: GRPC, Envoy, Linkerd, Open Policy Agent, Ambassador, Apache Thrift, AVI Network, Backplane, Conduit, Haproxy, Heptio Contour, Istio, Kong, Netflix Ribbon, Netflix Zuul, NGINX, Traefik, Turbine Labs, Vamp.

Runtime

Cloud-Native Storage: Rook, Ceph, CSI, Datera, Dell EMC, Diamanti, Gluster, Hatchway, Hedvig, Kasten, Leofs, Minio, NetApp, OpenEBS, OpenIO, OpenSDS, Pure Storage, Quobyte, Robin, Sheeppdog, StorageOS, Swift, Triton Object Storage.

Container Runtime: containerd, rkt, cri-o, Google Cloud Gvisor, Intel Clear Containers, Kata, Ixd, Pouch, CNCF Incubating, Open SUSE, RunV, Singularity, SmartOS.

Cloud-Native Network: CNF, Alcid, Aporo, Avatrix, Cilium, Contiv, Cumulus, Flannel, GuardCore, Ligato, Midnet, VMware NSX, Open vSwitch, OpenContrail, PLUMgrid, Calico, Romana, Weave Net.

Provisioning

Host Management / Tooling: Ansible, CFEngine, Chef, kube-bench, LinkerKit, Puppet, Run Deck, SaltStack, StackStorm.

Infrastructure Automation: AWS CloudFormation, BOSH, Cloudify, Infratit, Juju, ManageIQ, Terraform.

Container Registries: Amazon ECR, Azure Registry, Codefresh Registry, Docker Registry, Harbor, IBM Cloud, Jfrog Artifactory, Portus, Atomic, Quay.

Secure Images: Anchore, Aqua, BlackDuck, Clair, Grafeas, NewVector, OpenSCAP, Scrapsys Nexus, Twistlock.

Key Management: Cybereark Conjur, KeyWhiz, KNOX, Spiffe, SPIRE, Lyr, Oracle Policy Automation, Teleport, Vault.

Public: Alibaba Cloud, AWS, Azure, Baidu Cloud, DigitalOcean, FUJITSU, HUAWEI, Tencent Cloud, IBM, Huawei.

Private: Digital Rebar, Foreman, IAAS, Openstack, ServiceMesh, 科大讯飞, nware.

Cloud Native Landscape

CLOUD NATIVE COMPUTING FOUNDATION

Redpoint, Amplify

l.cncf.io

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

Platforms

Certified Kubernetes - Distribution: Alibaba Cloud, AppCode, CalCloud, Canonical, Container Engine, DaoCloud, Diamanti, Docker, Ekecode, Giant Swarm, IBM Cloud, JD.COM, Joyent, PaaS, KubeSphere, Kubermatic, Kublr, Mesosphere, Mirantis, Navops, NetEase Cloud, OpenShift, Oracle, Pivotal, Platform, QFusion, Rancher, Stackpoint, SUSE, Tectonic, Telekom, Typhoon, VMware.

Certified Kubernetes - Hosted: Alibaba Cloud, Amazon AWS, Azure, Baidu Cloud, C3 Alibaba Cloud, Amazon EKS, EKS on Amazon EC2, Azure, IBM Cloud, CISCO, EasyStack, Alibaba Cloud, 腾讯云, 华为云, IBM Cloud, Nirmata, Oracle, SAP, Tencent Cloud, ZTE.

Certified Kubernetes - Installer: Apprenda, Core OS, Gardener, Google Cloud, Heptio, Oracle.

Non-Certified Kubernetes: Maestro, WePaaS, Huawei, Ektoscale, Supersigant.

PaaS/Container Service: Application Runtime, Flynn, Salastic FDC, Heroku, Hyper SH, JMeter, Kintena, Lightbend, Mantl, NoCode, PaaS, JHipster, Jpartner, Partner.io, Scalingo, Thuan.

Observability & Analysis

Monitoring: Prometheus, Grafana, Dynatrace, InfluxDB, Nagios, NetSIL, New Relic, NodeSource, Outlyer, Sensu, Sentry, SignalFx, StackRox, StarState, Sysdig, Thanos, Weave Cloud, Weave Scope.

Logging: Fluentd, Elastic, Graylog, Humio, Loggly, LogZilla, Loom, Sematext, Splunk, Sumologic.

Tracing: Jaeger, OpenTracing, Skywalking, Zipkin.

Kubernetes Certified Service Provider: Accenture, Alibaba Cloud, Amazon AWS, Argo, Baidu Cloud, Bitnami, BOC, CalCloud, Canonical, Clarinet, CloudOps, Concourse, Contino, Core OS, DaoCloud, Easystack, Ekecode, Giant Swarm, Heptio, Huawei, IBM Cloud, InuStack, JETSTACK, Kenzan, Kumina, LiveWier, Loadse, Mirantis, Natways, DaoCloud, InuStack, Loadse, NewContext, Nirmata, NITDATA, OCTO, OpenCloud, RX-ME, Samsung SDS, SAP, Stackpoint, Supersigant, 腾讯云, Treasure Data, WeaveWorks, X-ME, Linux Training.

Kubernetes Training Partner: DaoCloud, CloudOps, InuStack, DaoCloud, InuStack, Loadse, RX-ME, Linux Training.

Q&A



杨文 

广东 深圳



扫一扫上面的二维码图案，加我微信



金融级分布式架构



ServiceMesh

欢迎关注微信公众号，获取更多技术干货！

<http://www.servicemesh.com>