

# 持续集成

Subtitle

2022/10/05



# Table of Contents

持续集成 .....	1
构建范围控制 .....	1
变通方法 .....	1
多平台构建 .....	1
分支tag .....	2
submodule .....	2
kaniko cache .....	3
auto_tag 实现 .....	3
问题 .....	3
开发流程 .....	5
使用Ingress nginx作为反向代理 .....	5
相关资料 .....	6



# 持续集成

使用 drone 作为持续集成工具

其他工具

- [travis ci交叉编译](#)

## 构建范围控制

□ 根据目录触发CI

如果只更新了某个目录，可能只希望该目录下的内容重新构建

## 变通方法

见：[Github](#)

比较本次提交和上次提交变化的文件

`snippet.bash`

```
$ git diff --name-only 93ed3995823046b4d26088facdb028471f3e3e6e  
9f30117232a8fd090202e56b3e3843bfb3a19f23  
src/cv.md
```

对于相同性质，并且预期会不断增加的子文件夹，可以只用一个 step，比如用一个repo管理多个基于 PanBook 的简历或文章或幻灯片。

当子文件差异较大并且子文件夹长期不变的情况下，利用 exit code，检测本文件夹是否变化，如果没有变化，exit 78，见 [Drone论坛](#)，比如 PanBook 主项目可以用这种方式。

## 多平台构建

见：<https://discourse.drone.io/t/builds-are-stuck-in-pending-status/4437>

似乎要求有对应架构的 agent，不能交叉编译

## Undefined Platform when using Arm or Arm64

Drone assumes all pipelines are amd64 unless otherwise specified. If you are using Drone with arm or arm64 agents please be sure to specify the architecture to ensure builds are routed to the correct agent.

```
kind: pipeline
name: default

+platform:
+  os: linux
+  arch: arm

steps: ...
```

## 分支 tag

以下示例，在 k8s-sidecar 分支 push 或者创建 tag 时触发 CI

snippet.yaml

```
when:
  ref:
    - refs/heads/k8s-sidecar
    - refs/tags/*

---或者用trigger
trigger:
  ref:
    - refs/heads/test
    - refs/tags/*
```

参考：

- <https://discourse.drone.io/t/execute-a-build-step-only-when-a-feature-branch-is-tagged-with-a-specific-pattern/1237/3>
- <https://docker-runner.docs.drone.io/configuration/conditions/>

## submodule

<https://docker-runner.docs.drone.io/configuration/cloning/>

The default clone behavior does not use the -recursive flag and does not fetch submodules. If you would like to fetch submodules you should handle this as a step in your pipeline. For example:

snippet.yaml

```
kind: pipeline
type: docker
name: default

steps:
- name: submodules
  image: alpine/git
  commands:
    - git submodule update --recursive --remote
```

## kaniko cache

<https://github.com/GoogleContainerTools/kaniko#cache-repo>

## auto\_tag 实现

<https://github.com/banzaicloud/drone-kaniko/pull/27>

auto\_tag 是什么？

从目前看到的资料，似乎是在代码库打tag之后根据代码块的tag自动设置镜像的tag

比如 代码tag 1.0.1 镜像tag 1, 1.0, 1.0.1

在 drone-kaniko 中，考虑直接根据 \${DRONE\_TAG} 变量来实现 auto\_tag 功能，如果这个变量有效，那么按照以上规则加tag，否则用 latest。

## 问题

### go build ldflags

正常运行的

[snippet.yaml](#)

```
kind: pipeline
name: default

steps:
- name: build
  image: cr.registry.com/library/golang:1.13.3-alpine
  commands:
    - go build -ldflags "-X 'main.buildTime=`date`' -X 'main.goVersion=`go version`' -X
```

```
'main.gitHash=${DRONE_COMMIT_SHA}' -X 'main.version=${DRONE_TAG=latest}''' -a -o
bin/ep-healthcheck cmd/ep-healthcheck/main.go
environment:
  GOPROXY: https://goproxy.registry.com
  CGO_ENABLED: 0
  GOOS: linux
  GOARCH: amd64
- name: docker
  image: cr.registry.com/drone/kaniko
  settings:
    repo: op/kube-ep-healthcheck
    username:
      from_secret: username
    password:
      from_secret: password
    auto_tag: true
```

## plugin image 未更新

修复方案：<https://github.com/ops-itop/drone-runtime/tree/k8s-pullpolicy>

使用此分支重新编译 drone-controller.



drone 代码的 go.mod 中添加 replace github.com/drone/drone-runtime => github.com/ops-itop/drone-runtime k8s-pullpolicy 之后再编译

<https://discourse.drone.io/t/drone-1-3-1-on-kubernetes-not-pull-latest-plugin-image/6088>

手动设置 pull policy

snippet.yaml

```
pull: always
```

kubernets log

```
Container image "cr.private.com/drone/kaniko:latest" already present on machine
```

只能给repo添加secret，secret更改比较麻烦

考虑提issue，增加用户级别的secret，比如，用户级别的 docker push password，当repo中未定义 docker push password时yaml中又引用时，使用用户级别的docker push password。但是搜索到一个 issue，见：<https://github.com/drone/drone/issues/1619>，提出一个 Organization 级别的 secret 概念，

但是被作者给否了，作者提出一个变通的方案：使用命令行批量设置secret

### snippet.bash

```
drone repo ls | xargs -I{} drone secret add {} KEY VALUE
```

## 开发流程

参考：

- <https://avnpic.com/pages/drone-gitflow-kubernetes-for-cloud-native-ci>

## 使用Ingress nginx作为反向代理

参考文档：<https://0-8-0.docs.drone.io/setup-with-nginx/>

通过configmap修改了以下参数

```
proxy_read_timeout 3s;
proxy_next_upstream_tries 3;
```

3s 对于 drone 的 api/stream 接口可能太短。修改为60s

重试3次可能导致 websocket 连接断开，报以下错误

```
net::ERR_CONNECTION_CLOSED 200
proxy_read_timeout 3s时会报 net::ERR_SPDY_PROTOCOL_ERROR 200
```

改成1就不报错了，原因是？存疑，只增大 proxy\_read\_timeout，不动 proxy\_next\_upstream\_tries 问题也能解决。

This screenshot shows a configuration interface with the following tabs: 属性 (Properties), 联系人 (1) (Contacts), 自定义Nginx配置 (3) (Custom Nginx Configuration), 通知 (2) (Notifications), and 历史 (History). The '自定义Nginx配置' tab is selected.

The interface displays three custom Nginx configurations:

- 此负载均衡的所有自定义Nginx配置**: Total: 3 objects.
- 1**: 启用 (Enabled): 是 (Yes); 配置值 (Value): 60; 配置项 (Item): nginx.ingress.kubernetes.io/proxy-read-timeout; 用途 (Purpose): proxy\_read\_timeout; 配置值校验 (Validation): /^[0-9]{1,3}\$/; 默认值 (Default): 1.
- 2**: 启用 (Enabled): 是 (Yes); 配置值 (Value): 1; 配置项 (Item): nginx.ingress.kubernetes.io/proxy-next-upstream-tries; 用途 (Purpose): proxy\_next\_upstream\_tries; 配置值校验 (Validation): /^[1-3]\$/; 默认值 (Default): 2.
- 3**: 启用 (Enabled): 是 (Yes); 配置值 (Value): chunked\_transfer\_encoding off; 配置项 (Item): nginx.ingress.kubernetes.io/configuration-snippet; 用途 (Purpose): 自定义location; 配置值校验 (Validation): # no default value.

## 相关资料

api/stream 类型的 eventsource ,

参考文章 :

- <https://blog.csdn.net/brace924/article/details/92649471>
- <https://serverfault.com/questions/801628/for-server-sent-events-sse-what-nginx-proxy-configuration-is-appropriate>
- 速度优化 <https://avnpic.com/pages/speed-up-drone-ci-workflow>

Printed on: 2022/10/05 17:43

Convert to img Failed!