

薅一薅马叔叔的羊毛

使用 cnb 快速构建 AOSP

CNB 是什么？cnb.cool

- 云原生构建(Cloud Native Build)，马叔叔的新玩具
- 基于 Docker 生态，对环境、缓存、插件进行抽象，通过声明式的语法，帮助开发者以更酷的方式构建软件。
 - 声明式：声明式语法，可编程、易分享。
 - 易管理：与代码一起，同源管理。
 - 云原生：资源池化，屏蔽基础设施复杂性。

AOSP 是什么？

- Android Open Source Project （AOSP）
- 是根据Apache许可证授权的免费开源软件
- 然而，大多数设备使用谷歌开发的专有Android版本，预装谷歌移动服务等专有软件
- “Android”名称和徽标是谷歌的商标，谷歌可以限制未经认证的设备不得使用Android品牌和谷歌的专有版本

为什么不用自己的电脑？

- 大，源码100G+
- 内存需求高，32G是钱包的极限但不是 make 的极限
- 速度慢，🐢 速

云原生构建

[用量详情](#)

免费配额

29.51%

已用 47.22 核时 / 上限 160.00 核时

公测限免配额

0.00%

已用 0.00 核时 / 上限 1,600.00 核时

付费用量

0.00%

已用 0.00 核时 / 上限 ∞ 核时

云原生开发

[用量详情](#)

免费配额

9.98%

已用 159.61 核时 / 上限 1,600.00 核时

公测限免配额

0.00%

已用 0.00 核时 / 上限 16,000.00 核时

付费用量

0.00%

已用 0.00 核时 / 上限 ∞ 核时

我们需要怎么做？

1. 注册登陆
2. 创建组织
3. fork 仓库
4. 触发构建
5. 上传到 Docker 制品库

注册登陆：

- cnb.cool 使用微信扫码登陆

创建组织：

- 5位以下需要验证域名

fork 仓库：

- 提供的样例仓库里有 aosp: <https://cnb.cool/aosp/monorepo>

学习 CI:

以仓库为主体，由 配置文件 描述在哪个 分支 下发生什么 事件 时执行什么 任务。

! .cnb.yml U ×

! .cnb.yml > {} main > [] push > {} 0 > [] stages > {} 0

JSON schema for cnb configuration files (conf-schema-zh.json)

```
1  # 分支名
2  main:
3    # 事件名
4    push:
5      # 要执行的任务
6      - stages:
7        - name: echo
8          script: echo "hello world"
9
```


AOSP 的构建 CI:

- ctrl + C
- ctrl + V

```
$:
vscode:
  - runner:
      cpus: 64
  docker:
    build: .ide/Dockerfile
    volumes:
      - out:copy-on-write
  services:
    - vscode
    - docker
  stages:
    - name: df -h
      script: df -h
```

```
aosp_build_config:
  runner:
    cpus: 64
  docker:
    build: .ide/Dockerfile
    volumes:
      - out:copy-on-write
  stages:
    - name: build
      env:
        BUILD_HOSTNAME: cnb-build
        shell: |
          source build/envsetup.sh
          lunch aosp_arm-eng
          make -j64
      script: bash -e -c "${shell}"
```

```
main:
  push:
    - runner:
        cpus: 64
    docker:
      build: .ide/Dockerfile
      volumes:
        - out:copy-on-write
    stages:
      - name: build
        env:
          BUILD_HOSTNAME: cnb-build
          shell: |
            source build/envsetup.sh
            lunch aosp_arm-eng
            make -j64
        script: bash -e -c "${shell}"
```

workspace

workspace

workspace

EXPLORER

WORKSPACE

cts

dalvik

developers

development

device

external

frameworks

hardware

kernel

libcore

libnativehelper

out

packages

pdk

platform_testing

prebuilt

sd

system

test

toolchain

tools

.cnb.yml

.gitignore

Android.bp

bootstrap.bash

BUILD

README.md

WORKSPACE

OUTLINE

TIMELINE

CNB Welcome

.cnb.yml

.cnb.yml > {} main > [] push > {} 0 > [] stages > {} 0 > script

You, 3 minutes ago | 2 authors (You and one other)

1 main:

2 push:

3 stages:

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

1 [|||||] 100.0%

2 [|||||] 100.0%

3 [|||||] 100.0%

4 [|||||] 100.0%

5 [|||||] 100.0%

6 [|||||] 100.0%

7 [|||||] 100.0%

8 [|||||] 100.0%

9 [|||||] 100.0%

10 [|||||] 100.0%

11 [|||||] 100.0%

12 [|||||] 100.0%

13 [|||||] 100.0%

14 [|||||] 100.0%

15 [|||||] 100.0%

16 [|||||] 100.0%

17 [|||||] 100.0%

18 [|||||] 100.0%

19 [|||||] 100.0%

20 [|||||] 100.0%

21 [|||||] 100.0%

22 [|||||] 100.0%

23 [|||||] 100.0%

24 [|||||] 100.0%

25 [|||||] 100.0%

26 [|||||] 100.0%

27 [|||||] 100.0%

28 [|||||] 100.0%

29 [|||||] 100.0%

30 [|||||] 100.0%

31 [|||||] 100.0%

32 [|||||] 100.0%

33 [|||||] 100.0%

34 [|||||] 100.0%

35 [|||||] 100.0%

36 [|||||] 100.0%

37 [|||||] 100.0%

38 [|||||] 100.0%

39 [|||||] 100.0%

40 [|||||] 100.0%

41 [|||||] 100.0%

42 [|||||] 100.0%

43 [|||||] 100.0%

44 [|||||] 100.0%

45 [|||||] 100.0%

46 [|||||] 100.0%

47 [|||||] 100.0%

48 [|||||] 100.0%

49 [|||||] 100.0%

50 [|||||] 100.0%

51 [|||||] 100.0%

52 [|||||] 100.0%

53 [|||||] 100.0%

54 [|||||] 100.0%

55 [|||||] 100.0%

56 [|||||] 100.0%

57 [|||||] 100.0%

58 [|||||] 100.0%

59 [|||||] 100.0%

60 [|||||] 100.0%

61 [|||||] 100.0%

62 [|||||] 100.0%

63 [|||||] 100.0%

64 [|||||] 49.3%

Mem[|||||] 1.63G/128G

Swp[|||||] 768K/0K

Tasks: 27, 1021 thr; 19 running

Load average: 28.70 7.58 2.57

Uptime: 00:20:04

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
14908	root	20	0	14.4G	581M	23444	S	1521	0.4	0:32.48	/workspace/out/host/linux-x86/bin/soong_build --top /wo
13965	root	20	0	6992M	63380	18420	S	1057	0.0	0:46.38	/workspace/out/host/linux-x86/bin/go/soong-java/test/te
14445	root	20	0	8579M	100M	21012	S	847.	0.1	0:22.56	/workspace/out/host/linux-x86/bin/go/soong-apex/test/te
12385	root	20	0	10.5G	76440	14708	S	835.	0.1	1:25.93	/workspace/out/host/linux-x86/bin/go/soong-cc/test/test
12671	root	20	0	10.8G	90024	14888	S	824.	0.1	1:20.50	/workspace/out/host/linux-x86/bin/go/soong-rust/test/te
F312971	Esc	Cancel	Search	14.4G	581M	23444	S	15.2	0.4	0:02.31	prebuilt/go/linux-x86/pkg/tool/lisoong_build --top /wo

cnb

git

cnb

main

0

0

0

0

0

You, 1 second ago

Ln 5, Col 33

Spaces: 2

UTF-8

LF

YAML

Layout: us

No JSON Schema

和我一起说： 谢谢马叔叔