

# 更高效的构建 Docker 镜像

张晋涛

# 个人介绍

- Container/Docker/Kubernetes 开发者
- 知乎/微博: @张晋涛



# 目录

- 如何构建 Docker 镜像
- 优化标准
- 通用实践经验
- 深度优化
- CI 中的落地实践

# 范围

- 仅讨论 Docker 镜像
- 仅限利用 Docker 及其相关技术构建 Docker 镜像

# 如何构建 Docker 镜像

- 构建服务 (docker daemon)
- 构建规则 (Dockerfile)
- 构建请求 (docker CLI/docker client)

# 常规过程

```

(MoeLove) → foo docker build -t local/test:foo -f Dockerfile .
Sending build context to Docker daemon 2.048kB
Step 1/2 : FROM alpine
---> f70734b6a266
Step 2/2 : RUN apk add -q --no-cache bash
---> Running in 39f10d68922e
Removing intermediate container 39f10d68922e
---> 680f364783a8
Successfully built 680f364783a8
Successfully tagged local/test:foo
    
```

- Dockerfile 可使用本地目录中的文件
- 可通过远端 URL 构建，Git 仓库/标准 **tar** 归档/纯文本

# 从标准输入构建

```

(MoeLove) → foo docker build -t local/test:from-stdin - <<EOF
FROM alpine
RUN apk add --no-cache -q bash
EOF
Sending build context to Docker daemon  2.048kB
Step 1/2 : FROM alpine
---> f70734b6a266
Step 2/2 : RUN apk add --no-cache -q bash
---> Running in 2f140cfba585
Removing intermediate container 2f140cfba585
---> 81b4950fd49f
Successfully built 81b4950fd49f
Successfully tagged local/test:from-stdin
  
```

- 不支持同时从 stdin 读取 Dockerfile 与构建内容

# 查看构建结果

```
(MoeLove) → foo docker image ls local/test --format '{{.ID}}\t{{.Size}}'
81b4950fd49f      7.61MB      from-stdin
680f364783a8      7.61MB      foo
```



# 优化标准

- 构建耗时
- 镜像体积
- 用户体验

# 通用实践经验

# 排除不需要文件

- 使用 `.dockerignore`

# 极端案例，直接忽略全部，对需要文件单独放行

```
*  
!index.js
```

- 类似 `.gitignore`

# 利用构建缓存

```
(MoeLove) → foo docker build -t local/test:cache -f Dockerfile .  
Sending build context to Docker daemon 2.048kB  
Step 1/2 : FROM alpine  
---> f70734b6a266  
Step 2/2 : RUN apk add -q --no-cache bash  
---> Using cache  
---> 680f364783a8  
Successfully built 680f364783a8  
Successfully tagged local/test:cache
```

# 不建议的示例

```
FROM debian

COPY . /app

RUN apt update
RUN apt install -y openjdk-8-jdk

CMD [ "java", "-jar", "/app/target/gs-spring-boot-0.1.0.jar" ]
```

- 示例为本地预先构建，在构建过程中拷贝内容
- 此处无法利用构建缓存

# 将 COPY 动作后移

```
FROM debian
```

```
RUN apt update
```

```
RUN apt install -y openjdk-8-jdk
```

```
COPY . /app
```

```
CMD [ "java", "-jar", "/app/target/gs-spring-boot-0.1.0.jar" ]
```

# 部分拷贝

```
FROM debian
```

```
RUN apt update
```

```
RUN apt install -y openjdk-8-jdk
```

```
COPY target/gs-spring-boot-0.1.0.jar /app/
```

```
CMD [ "java", "-jar", "/app/gs-spring-boot-0.1.0.jar" ]
```

# 防止包缓存过期

**FROM** debian

**RUN** apt update && apt install -y openjdk-8-jdk

**COPY** target/gs-spring-boot-0.1.0.jar /app/

**CMD** [ "java", "-jar", "/app/gs-spring-boot-0.1.0.jar" ]

- 避免单独的 `apt update` 被缓存



# 慎用包管理器

```

root@5a23eb858163:/# apt install --no-install-recommends openjdk-8-jdk | gr
...
After this operation, 344 MB of additional disk space will be used.
^C
root@5a23eb858163:/# apt install openjdk-8-jdk | grep 'additional disk space
...
After this operation, 548 MB of additional disk space will be used.
^C
    
```

- 包管理器的额外“优化”

# 优化后

**FROM** debian

**RUN** apt update && apt install -y --no-install-recommends openjdk-8-jdk

**COPY** target/gs-spring-boot-0.1.0.jar /app/

**CMD** [ "java", "-jar", "/app/gs-spring-boot-0.1.0.jar" ]

# 效果对比

```
(MoeLove) → docker image ls local/foo --format '{{.ID}}\t{{.Size}}'
716523c83a26      497MB
178dacdaf015      600MB
```

- 通过增加 `--no-install-recommends` 参数，镜像体积减少 103M

# 清理包管理器缓存

**FROM** debian

**RUN** apt update && apt install -y --no-install-recommends openjdk-8-jdk \\  
 && rm -rf /var/lib/apt/lists/\*

**COPY** target/gs-spring-boot-0.1.0.jar /app/

**CMD** [ "java", "-jar", "/app/gs-spring-boot-0.1.0.jar" ]

```
(MoeLove) → docker image ls local/foo --format '{{.ID}}\t{{.Size}}'
ac272f3dcac2      481MB
716523c83a26      497MB
178dacdaf015      600MB
```

# 保证环境一致性

```
FROM maven:3.6.1-jdk-8-alpine AS builder
```

```
WORKDIR /app
```

```
COPY pom.xml /app/
```

```
RUN mvn dependency:go-offline
```

```
COPY src /app/src
```

```
RUN mvn -e -B package
```

```
FROM openjdk:8-jre-alpine
```

```
COPY --from=builder /app/target/gs-spring-boot-0.1.0.jar /
```

```
CMD [ "java", "-jar", "/gs-spring-boot-0.1.0.jar" ]
```

# 其他建议

- 选择合适的基础镜像
- 不做重复工作

# 深度优化

- 添加后删除

```
(MoeLove) → foo cat Dockerfile  
FROM alpine
```

```
RUN apk add -q --no-cache bash
```

```
RUN apk del -q bash
```

# 合并无用层

```
(MoeLove) → foo docker image ls local/test --format '{{.ID}}\t{{.Size}}\t{{.Architecture}}\n'
ae4b9effa140      5.63MB      squash
16ada640b0e2      7.63MB      add-and-del
```

- 通过 **squash** 将无用层合并



# 构建时删除

- 层内删除
- dive 分析

[Layers]				[Current Layer Contents]			
Cmp	Image ID	Size	Command	Permission	UID:GID	Size	Filetree
	sha256:3e207b409db364b595	5.6 MB	FROM sha256:3e207b409db364b5	drwxr-xr-x	0:0	841 kB	— bin
	sha256:60c88addb36dc14558	2.0 MB	apk add -q --no-cache bash	-rwxrwxrwx	0:0	0 B	arch → /bin/busybox
	sha256:6852453dfffb76adf94	22 kB	apk del -q bash	-rwxrwxrwx	0:0	0 B	ash → /bin/busybox
[Image & Layer Details]				-rwxrwxrwx	0:0	0 B	base64 → /bin/busybox
				-rwxr-xr-x	0:0	736 kB	bash
Layer Command /bin/sh -c apk del -q bash				-rwxrwxrwx	0:0	0 B	bbconfig → /bin/busybox
				-rwxr-xr-x	0:0	841 kB	busybox
Image efficiency score: 79 % Potential wasted space: 1.6 MB				-rwxrwxrwx	0:0	0 B	cat → /bin/busybox
				-rwxrwxrwx	0:0	0 B	chgrp → /bin/busybox
Count    Total Space    Path				-rwxrwxrwx	0:0	0 B	chmod → /bin/busybox
				-rwxrwxrwx	0:0	0 B	chown → /bin/busybox
2	736 kB	/bin/bash		-rwxrwxrwx	0:0	0 B	conspy → /bin/busybox
2	368 kB	/usr/lib/libncursesw.so.6.1		-rwxrwxrwx	0:0	0 B	cp → /bin/busybox
2	277 kB	/usr/lib/libreadline.so.8.0		-rwxrwxrwx	0:0	0 B	date → /bin/busybox
2	68 kB	/usr/lib/libformw.so.6.1		-rwxrwxrwx	0:0	0 B	dd → /bin/busybox
3	40 kB	/lib/apk/db/installad		-rwxrwxrwx	0:0	0 B	df → /bin/busybox
3	35 kB	/lib/apk/db/scripts.tar		-rwxrwxrwx	0:0	0 B	dmesg → /bin/busybox
2	35 kB	/usr/lib/libmenuw.so.6.1		-rwxrwxrwx	0:0	0 B	dnsdomainname → /bin/busy
2	18 kB	/usr/lib/libpanelw.so.6.1		-rwxrwxrwx	0:0	0 B	dumpkmap → /bin/busybox
2	1.7 kB	/etc/inputrc		-rwxrwxrwx	0:0	0 B	echo → /bin/busybox
3	228 B	/lib/apk/db/triggers		-rwxrwxrwx	0:0	0 B	ed → /bin/busybox
3	182 B	/etc/apk/world		-rwxrwxrwx	0:0	0 B	egrep → /bin/busybox
3	124 B	/etc/shells		-rwxrwxrwx	0:0	0 B	false → /bin/busybox
3	0 B	/var/cache/misc		-rwxrwxrwx	0:0	0 B	fatattr → /bin/busybox
2	0 B	/usr/lib/libncursesw.so.6		-rwxrwxrwx	0:0	0 B	fdflush → /bin/busybox
				-rwxrwxrwx	0:0	0 B	fgrep → /bin/busybox
				-rwxrwxrwx	0:0	0 B	fsync → /bin/busybox
^C	Quit	^Space	Switch view	^/	Filter files	^L	Show layer changes
				^A	Show aggregated changes		

# CI 中的落地实践

# 构建加速

```
image: harbor-registry.inner.youdao.com/devops/docker:19.03
services:
  - name: harbor-registry.inner.youdao.com/devops/docker:19.03-dind
    alias: docker
stages:
  - build
enable buildkit:
  stage: build
tags:
  - k8s
variables:
  DOCKER_BUILDKIT: 1
script:
  - docker build -t $CI_HARBOR_REGISTRY/$CI_PROJECT_PATH .
```

# GitLab CI 中的实践方案

- 缓存支持
- 私有项目支持
- 部署工具 - ydci
- <http://confluence.inner.youdao.com/x/KLQm>

# Q&A

## Thanks!

