

32Kubernetes 系列（三十）如何将镜像大小缩减 95%

Kubernetes 系列（三十）如何将镜像大小缩减 95%

使用多阶段构建可以减小 Docker 镜像的大小。

在编写大小为 0.5 GB 的镜像之前，编写 Dockerfile 似乎很容易。是的,这是真的。我最近在做一个 Vue 项目，在那里我正在构建一个前端应用程序。Vue 项目没什么新鲜的。我用的是同一个package.json 文件，这是典型前端应用会使用的相同依赖集。现在，我决定将应用程序部署到Kubernetes。因此，我将 docker 镜像推送到 DockerHub，并试图部署一个Pod。

我正在使用一个本地集群，pod 用了5分7秒启动。我很敬畏。然后我将其部署到AKS集群（Azure Kubernetes集群）。令人惊讶的是，花了3分40秒。这是巨大的变化。我试着给 Pod 施加压力，以触发Kubernetes HPA。新的pod需要同样的3分钟，最终应用程序会被太多的请求淹没。经过调查，我发现镜像的大小是瓶颈，当然，你不想拥有这么大的docker 镜像（0.5GB）。那是我意识到我必须对docker 镜像大小做一些事情的时候。因此，我使用了Docker多级构建的概念。

整个文章关于什么？

1. 使用多级构建缩小Docker Images的大小。
2. 了解多阶段构建的概念。

Docker 构建:

```
FROM node:alpine
WORKDIR /app
COPY package.json ./
COPY package-lock.json ./
COPY ./ ./
RUN npm i
CMD ["npm", "run", "start"]
```

这是一个典型的 Vue 应用程序的初始Docker镜像。如果你看到第1行，我使用的是 alpine 版本的node 镜像。在构建Dockerfile之后，这是它的大小：

```
(kubernetes-admin@kubernetes:default)[09:44:08 cloud-app-react]kubeadvocate@ mac$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
non-multi-stage     latest             21b194adda25       26 minutes ago     504MB
```

嗯，这是构建镜像的典型方式，而且编写这样的 dockerfile 似乎很容易。但从Kubernetes的角度来看，使用这种方法的缺点如下：a) 镜像的大小直接影响Pod的启动时间。b) 镜像越小，Pod 启动的速度越快。

Docker 多级构建:

Dockerfiles 中的多级构建特性使您能够创建更小的容器镜像，具有更好的缓存和更小的安全占用。通过多级构建，您可以在 Dockerfile 中使用多个FROM语句。每个FROM指令都可以使用不同的基础镜像，并且每个FROM指令都开始构建的新阶段。您可以有选择地将工件从一个阶段复制到另一个阶段，在最终镜像中丢弃您不想要的一切。

```

# Choose the Image which has Node installed already
FROM node:alpine as build

WORKDIR /code

COPY package.json package.json
COPY package-lock.json package-lock.json

RUN npm ci --production
# COPY all the files from Current Directory into the Container
COPY . .

# Install the Project Dependencies Like Express Framework
RUN npm run build

FROM nginx:1.22-alpine as prod

COPY --from=build /code/build /usr/share/nginx/html

# Tell that this image is going to Open a Port
EXPOSE 80

# Default Command to Launch the Application
CMD ["nginx", "-g", "daemon off;"]

```

看起来有什么不同？好了，让我们来理解这个文件。

第一行被称为一个阶段。若这个阶段没有命名，您通过它们的整数索引它们，对于第一个FROM指令，从0开始。但是，您可以通过在FROM指令中添加AS <name>来命名您的阶段。

剩下的步骤保持不变。其中一个改变游戏规则的步骤是COPY - from(第18行)。通过多阶段构建，您可以在Dockerfile中使用多个FROM语句。每个FROM指令都可以使用不同的基础镜像，并且每个FROM指令都开始构建的新阶段。您可以有选择地将工件从一个阶段复制到另一个阶段，在最终镜像中丢弃您不想要的一切。

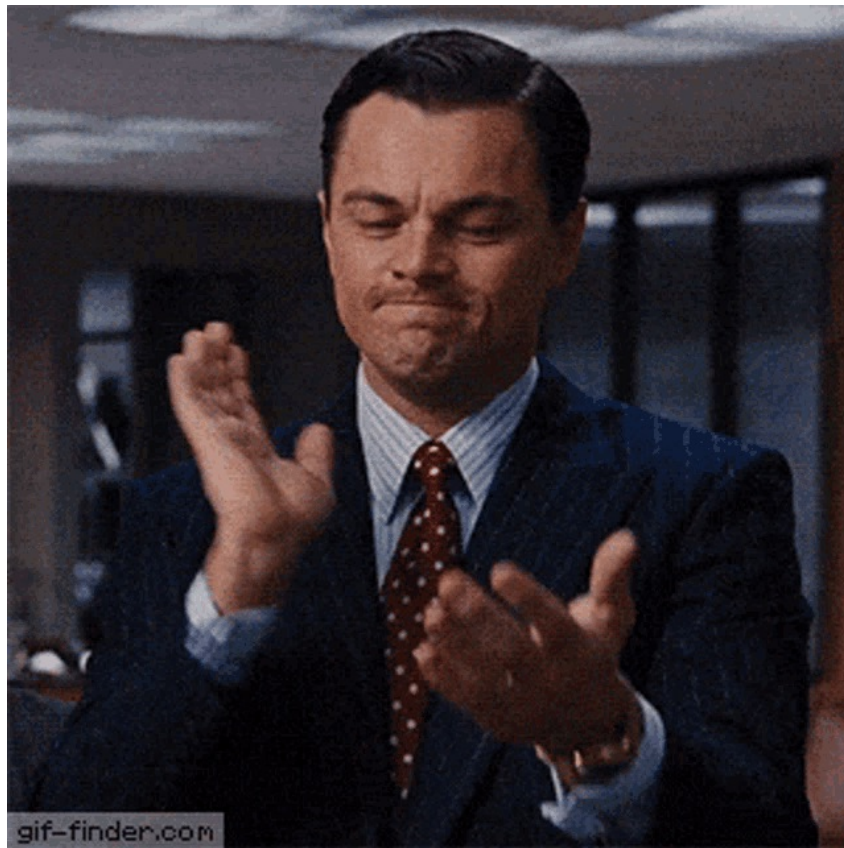
让我们现在开始构建新的镜像：

```

=> [build 5/7] RUN npm ci --production
=> [build 6/7] COPY . .
=> [build 7/7] RUN npm run build
=> [prod 2/2] COPY --from=build /code/build /usr/share/nginx/html
=> exporting layers
=> writing image sha256:365c29f1a75c08e4fc25cf2d94f98e4fb4616254cfc894079c12cf43c12b11ad
(⚡ [kubernetes-admin@kubernetes:default][10:40:45 cloud-app-react]kubeadvocate@ mac$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
<none> <none> 365c29f1a75c 6 seconds ago 27.2MB
(⚡ [kubernetes-admin@kubernetes:default][10:40:50 cloud-app-react]kubeadvocate@ mac$ docker tag 365c29f1a75c multi-stage-build
(⚡ [kubernetes-admin@kubernetes:default][10:41:03 cloud-app-react]kubeadvocate@ mac$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
multi-stage-build latest 365c29f1a75c 21 seconds ago 27.2MB
(⚡ [kubernetes-admin@kubernetes:default][10:41:05 cloud-app-react]kubeadvocate@ mac$

```

哇,是的!这是真的，镜像大小减小了95%。它的大小现在是27.2 MB。



我的本地集群花了25秒时间拉出镜像并启动。我的AKS Cluster花了10秒的时间拉出镜像并开始运行。

太棒了 :) !!

嗯，这就是你如何利用多级构建docker来减少你的docker镜像的大小。

欢迎关注我的公众号“[云原生拓展](#)”，原创技术文章第一时间推送。