

# kubernetes 部署 php 项目

作者: [fish2018](#)

原文链接: <https://ld246.com/article/1607674193897>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## php基础镜像构建

FROM daocloud.io/php:7.2-fpm-alpine

# 修改镜像源

RUN sed -i 's/dl-cdn.alpinelinux.org/mirrors.aliyun.com/g' /etc/apk/repositories

# 安装依赖,核心扩展,pecl扩展,git,composer,npm工具

RUN apk update && apk add --no-cache --virtual .build-deps

\$PHPIZE\_DEPS \

curl-dev \

imagemagick-dev \

libtool \

libxml2-dev \

postgresql-dev \

sqlite-dev \

libmcrypt-dev \

freetype-dev \

libjpeg-turbo-dev \

libpng-dev \

&& apk add --no-cache \

curl \

git \

imagemagick \

mysql-client \

postgresql-libs \

nodejs \

nodejs-npm \

# 配置npm中国镜像

&& npm config set registry <https://registry.npm.taobao.org> \

```

&& pecl install imagick \
&& pecl install mcrypt-1.0.1 \
&& docker-php-ext-enable mcrypt \
&& docker-php-ext-enable imagick \
&& docker-php-ext-install \
curl \
mbstring \
pdo \
pdo_mysql \
pdo_pgsql \
pdo_sqlite \
pcntl \
tokenizer \
xml \
zip \
&& docker-php-ext-install -j"$(getconf _NPROCESSORS_ONLN)" iconv
&& docker-php-ext-configure gd --with-freetype-dir=/usr/include/ --with-jpeg-dir=/usr/incl
de/
&& docker-php-ext-install -j"$(getconf _NPROCESSORS_ONLN)" gd
&& pecl install -o -f redis
&& rm -rf /tmp/pear
&& docker-php-ext-enable redis

# 安装composer并允许root用户运行

ENV COMPOSER_ALLOW_SUPERUSER=1
ENV COMPOSER_NO_INTERACTION=1
ENV COMPOSER_HOME=/usr/local/share/composer
RUN mkdir -p /usr/local/share/composer
&& curl -o /tmp/composer-setup.php https://getcomposer.org/installer
&& php /tmp/composer-setup.php --no-ansi --install-dir=/usr/local/bin --filename=compos
r --snapshot
&& rm -f /tmp/composer-setup.*
# 配置composer中国全量镜像
&& composer config -g repo.packagist composer https://packagist.phpcomposer.com

# 对其他容器开放9000端口

EXPOSE 9000

```

代码打包到php镜像，然后和nginx容器挂载相同目录。或者放到共享存储，php和nginx容器都挂载配置comfigmap、deployment、service、ingress

### 完整yaml

```

kind: Deployment # 对象类型
apiVersion: apps/v1 # api 版本
metadata: # 元数据
namespace: test-demo
name: test-demo # Deployment 对象名称
spec: # Deployment 对象规约
selector: # 选择器

```

```
matchLabels: # 标签匹配
app: test-demo
replicas: 1 # 副本数量
template: # 模版
metadata: # Pod 对象的元数据
labels: # Pod 对象的标签
app: test-demo
spec: # Pod 对象规约
imagePullSecrets:

- name: harbor-registry
  containers: # 这里设置了两个容器
- name: php-fpm # 第一个容器名称
  image: harbor.demo.com/demo:1.0.0 # php打包代码容器镜像
  ports:
- containerPort: 9000 # php-fpm 端口
  volumeMounts: # 挂载数据卷
- mountPath: /var/www/html # 挂载两个容器共享的 volume
  name: nginx-www
  lifecycle: # 生命周期
  postStart: # 当容器处于 postStart 阶段时, 执行一下命令
  exec:
  command: ["/bin/sh", "-c", "cp -r /test-demo /var/www/html && chown -R www-data.www-
ata /var/www/html/test-demo"] # 复制到挂载的 volume
- name: nginx # 第二个容器名称
  image: harbor.demo.com/nginx-1.16.1-baseimage:1.0.0 # 容器镜像
  ports:
- containerPort: 80 # nginx 端口
  volumeMounts: # nginx 容器挂载了两个 volume, 一个是与 php-fpm 容器共享的 volume, 另
一个是配置了 nginx.conf 的 volume
- mountPath: /var/www/html # 挂载两个容器共享的 volume
  name: nginx-www
- name: config-volume
  mountPath: /nginx/etc/conf.d
  volumes:
- name: nginx-www # 这个 volume 是 php-fpm 容器 和 nginx 容器所共享的, 两个容器都 volum
Mounts 了
  emptyDir: {}
- name: config-volume
  configMap:
  items:
- key: test-demo.conf
  path: test-demo.conf
  name: nginxconf

---

kind: Service # 对象类型
apiVersion: v1 # api 版本
metadata: # 元数据
namespace: test-demo
name: test-demo
spec:
```

```
selector:
app: test-demo
ports:
- port: 80
  targetPort: 80 # Service 将 nginx 容器的 80 端口暴露出来
```

---

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
name: test-demo
namespace: test-demo
annotations:
kubernetes.io/ingress.class: "nginx"
nginx.ingress.kubernetes.io/proxy-body-size: "50m"
spec:
rules:
- host: test-demo.demo.cn
  http:
  paths:
  - backend:
    serviceName: test-demo
    servicePort: 80
    path: /
    pathType: ImplementationSpecific
```

---

```
apiVersion: v1
kind: ConfigMap
metadata:
name: nginxconf
namespace: test-demo
data:
test-demo.conf: |-
server
{
listen 80 default_server;
listen [::]:80 default_server;
server_name _;
index index.php index.html index.htm default.php default.htm default.html;
root /var/www/html/test-demo/;

#PHP-INFO-START PHP引用配置，可以注释或修改
location ~ /\.php$ {
include fastcgi_params;
fastcgi_param REQUEST_METHOD $request_method;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
fastcgi_pass 127.0.0.1:9000;
}
```

```
#PHP-INFO-END
#REWRITE-START URL重写规则引用,修改后将导致面板设置的伪静态规则失效
location / {
    if (!-e $request_filename){
        rewrite ^(.*)$ /index.php?s=$1 last; break;
    }
}
#REWRITE-END

#禁止访问的文件或目录
location ~ ^/(\.user.ini|\.htaccess|\.git|\.svn|\.project|LICENSE|README.md)
{
    return 404;
}

#一键申请SSL证书验证目录相关设置
location ~ /\.well-known{
    allow all;
}

location ~ .*\.gif|jpg|jpeg|png|bmp|swf)$
{
    expires 30d;
    error_log off;
    access_log off;
}

location ~ .*\.js|css)?$
{
    expires 12h;
    error_log off;
    access_log off;
}
}
```