

# Minikube 和 Registry 和 Laravel 和 Vue 和

和 Minikube 和 registry 和  
Docker Hub和

Laravel 和 Vue 和

registry和

## 1. 和 Minikube 和 Registry

```
# 和 Minikube 和 registry 和
minikube start --driver=docker --insecure-registry="10.0.0.0/24"
minikube addons enable registry

# 和 registry 和
kubectl get svc -n kube-system | grep registry

# 和 registry
kubectl port-forward --namespace kube-system service/registry 5000:80 &
```

## 2. 和 Docker 和 Minikube 和 Registry

```
# 和 Minikube 和 Docker daemon
eval $(minikube docker-env)

# 和 Minikube 和 Docker
docker info | grep "Name: minikube"
```

## 3. 和 Laravel 和

# Laravel Dockerfile

```
# Dockerfile.laravel
FROM php:8.1-fpm

# 更新
RUN apt-get update && apt-get install -y \
    git \
    curl \
    libpng-dev \
    libonig-dev \
    libxml2-dev \
    zip \
    unzip \
    libzip-dev

# 安装 PHP 扩展
RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd zip

# 安装 Composer
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer

# 设置工作目录
WORKDIR /var/www

# 克隆代码
COPY . .

# 安装 Composer
RUN composer install --optimize-autoloader --no-dev

# 生成缓存
RUN php artisan config:cache && \
    php artisan route:cache && \
    php artisan view:cache

# 暴露端口
EXPOSE 9000
```

```
# 构建
```

```
CMD ["php-fpm"]
```



## Laravel 构建

```
# 构建
```

```
docker build -t localhost:5000/laravel-app:latest -f Dockerfile.laravel .
```

```
# 推送到 registry
```

```
docker push localhost:5000/laravel-app:latest
```

```
# 推送到 registry 并
```

```
curl http://localhost:5000/v2/_catalog
```

## 4. 构建

## Vue 构建

## Vue Dockerfile 构建

```
# Dockerfile.vue
```

```
FROM node:16 as builder
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install
```

```
COPY . .
```

```
RUN npm run build
```

```
FROM nginx:alpine
```

```
COPY --from=builder /app/dist /usr/share/nginx/html
```

```
COPY nginx.conf /etc/nginx/conf.d/default.conf
```

```
EXPOSE 80
```

```
CMD ["nginx", "-g", "daemon off;"]
```



## Nginx 构建

```
# nginx.conf
server {
    listen 80;
    server_name localhost;

    location / {
        root /usr/share/nginx/html;
        index index.html;
        try_files $uri $uri/ /index.html;
    }

    location /api {
        proxy_pass http://laravel-backend:9000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```



## Vue

```
# 
docker build -t localhost:5000/vue-app:latest -f Dockerfile.vue .

#  registry
docker push localhost:5000/vue-app:latest

# 
curl http://localhost:5000/v2/_catalog
```

## 5. Kubernetes

## Laravel

```
# laravel-deployment.yaml
apiVersion: apps/v1
```

kind: Deployment

metadata:

name: laravel-backend

namespace: laravel-vue-app

spec:

replicas: 2

selector:

matchLabels:

app: laravel-backend

template:

metadata:

labels:

app: laravel-backend

spec:

containers:

- name: laravel-app

image: localhost:5000/laravel-app:latest

imagePullPolicy: Always

env:

- name: DB\_HOST

value: "mysql"

- name: DB\_DATABASE

value: "laravel"

- name: DB\_USERNAME

value: "root"

- name: DB\_PASSWORD

valueFrom:

secretKeyRef:

name: laravel-secrets

key: db-password

ports:

- containerPort: 9000

resources:

requests:

cpu: "100m"

memory: "256Mi"

limits:


cpu: "500m"

memory: "512Mi"

# Vue

```
# vue-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: vue-frontend
  namespace: laravel-vue-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: vue-frontend
  template:
    metadata:
      labels:
        app: vue-frontend
    spec:
      containers:
        - name: vue-app
          image: localhost:5000/vue-app:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 80
      resources:
        requests:
          cpu: "50m"
          memory: "128Mi"
        limits:
          cpu: "200m"
          memory: "256Mi"
```

## 6.

```
# 
kubectl apply -f namespace.yaml
```

```
kubectl apply -f mysql-pvc.yaml
kubectl apply -f mysql-deployment.yaml
kubectl apply -f mysql-service.yaml
kubectl apply -f redis-deployment.yaml
kubectl apply -f redis-service.yaml
kubectl apply -f laravel-deployment.yaml
kubectl apply -f laravel-service.yaml
kubectl apply -f vue-deployment.yaml
kubectl apply -f vue-service.yaml
kubectl apply -f ingress.yaml
```

```
# pod
kubectl get pods -n laravel-vue-app -w
```

```
# URL
minikube service list
```

## 7.

 `build-and-deploy.sh` 

```
#!/bin/bash

# Laravel
echo "Building Laravel image..."
docker build -t localhost:5000/laravel-app:latest -f Dockerfile.laravel .
docker push localhost:5000/laravel-app:latest

# Vue
echo "Building Vue image..."
docker build -t localhost:5000/vue-app:latest -f Dockerfile.vue .
docker push localhost:5000/vue-app:latest



# Kubernetes
echo "Redeploying applications..."
kubectl rollout restart deployment/laravel-backend -n laravel-vue-app
kubectl rollout restart deployment/vue-frontend -n laravel-vue-app
```



```
echo "Deployment complete!"
```



## 8.







```
#  Minikube  Docker daemon
eval $(minikube docker-env)

#  registry 
kubectl get pods -n kube-system | grep registry

#  registry 
minikube addons disable registry
minikube addons enable registry
```





```
#  deployment 
resources:
  requests:
    cpu: "100m"
    memory: "128Mi"
  limits:
    cpu: "500m"
    memory: "512Mi"
```

## 9.



# 0000

kubectl delete -f .

# 01 registry

minikube addons disable registry

# 000000

docker rmi localhost:5000/laravel-app:latest

docker rmi localhost:5000/vue-app:latest

0000000000

Minikube 0000000000

Laravel 0 Vue 0000000000

registry

000000000000

Revision #1

Created 6 June 2025 08:51:23 by Admin

Updated 6 June 2025 08:51:38 by Admin