



```
minikube start --driver=docker --cpus=4 --memory=8192
```

## 2.

```
minikube addons enable ingress  
minikube addons enable metrics-server
```



## 1.

```
# namespace.yaml  
apiVersion: v1  
kind: Namespace  
metadata:  
  name: laravel-vue-app
```

## 2. MySQL

```
# mysql-deployment.yaml  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: mysql  
  namespace: laravel-vue-app  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      app: mysql  
  template:  
    metadata:  
      labels:  
        app: mysql
```

```
spec:
  containers:
    - name: mysql
      image: mysql:5.7
      env:
        - name: MYSQL_ROOT_PASSWORD
          value: yourpassword
        - name: MYSQL_DATABASE
          value: laravel
      ports:
        - containerPort: 3306
      volumeMounts:
        - name: mysql-persistent-storage
          mountPath: /var/lib/mysql
  volumes:
    - name: mysql-persistent-storage
      persistentVolumeClaim:
        claimName: mysql-pv-claim
```

---

```
# mysql-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: mysql
  namespace: laravel-vue-app
spec:
  ports:
    - port: 3306
  selector:
    app: mysql
```

---

```
# mysql-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
```

```
name: mysql-pv-claim
namespace: laravel-vue-app
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

## 3. Redis

```
# redis-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis
  namespace: laravel-vue-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
        - name: redis
          image: redis:alpine
          ports:
            - containerPort: 6379

---

# redis-service.yaml
apiVersion: v1
kind: Service
```

```
metadata:
  name: redis
  namespace: laravel-vue-app
spec:
  ports:
    - port: 6379
  selector:
    app: redis
```

## 4. Laravel

```
# laravel-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: laravel-backend
  namespace: laravel-vue-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: laravel-backend
  template:
    metadata:
      labels:
        app: laravel-backend
    spec:
      containers:
        - name: laravel-app
          image: your-registry/laravel-app:latest
          env:
            - name: DB_HOST
              value: mysql
            - name: DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: laravel-secrets
                  key: db-password
```

```
- name: REDIS_HOST
  value: redis
ports:
- containerPort: 9000
livenessProbe:
  httpGet:
    path: /api/health
    port: 9000
  initialDelaySeconds: 30
  periodSeconds: 10
readinessProbe:
  httpGet:
    path: /api/health
    port: 9000
  initialDelaySeconds: 5
  periodSeconds: 5
```

---

```
# laravel-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: laravel-backend
  namespace: laravel-vue-app
spec:
  ports:
  - port: 9000
  selector:
    app: laravel-backend
```

## 5. Vue

```
# vue-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: vue-frontend
```

```
namespace: laravel-vue-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: vue-frontend
  template:
    metadata:
      labels:
        app: vue-frontend
    spec:
      containers:
        - name: vue-app
          image: your-registry/vue-app:latest
          env:
            - name: VUE_APP_API_URL
              value: http://laravel-backend:9000
          ports:
            - containerPort: 8080
          livenessProbe:
            httpGet:
              path: /
              port: 8080
            initialDelaySeconds: 30
            periodSeconds: 10
```

---

```
# vue-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: vue-frontend
  namespace: laravel-vue-app
spec:
  ports:
    - port: 80
      targetPort: 8080
  selector:
```

```
app: vue-frontend
```

```
---
```

```
# ingress.yaml
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
metadata:
```

```
  name: laravel-vue-ingress
```

```
  namespace: laravel-vue-app
```

```
  annotations:
```

```
    nginx.ingress.kubernetes.io/rewrite-target: /
```

```
spec:
```

```
  rules:
```

```
  - host: laravel-vue.local
```

```
    http:
```

```
      paths:
```

```
      - path: /
```

```
        pathType: Prefix
```

```
        backend:
```

```
          service:
```

```
            name: vue-frontend
```

```
            port:
```

```
              number: 80
```

```
      - path: /api
```

```
        pathType: Prefix
```

```
        backend:
```

```
          service:
```

```
            name: laravel-backend
```

```
            port:
```

```
              number: 9000
```



GitHub Actions 

```
# .github/workflows/ci-cd.yaml
name: CI/CD Pipeline

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  build-test-backend:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2

      - name: Set up PHP
        uses: shivammathur/setup-php@v2
        with:
          php-version: '8.1'
          extensions: mbstring, ctype, fileinfo, tokenizer, mysql, pdo, xml, gd
          coverage: pcov

      - name: Install dependencies
        run: composer install --prefer-dist --no-progress

      - name: Copy .env
        run: cp .env.example .env

      - name: Generate key
        run: php artisan key:generate

      - name: Run tests
        run: php artisan test

      - name: Build Docker image
        run: docker build -t your-registry/laravel-app:${{ github.sha }} -f Dockerfile.backend .

      - name: Login to Docker Hub
        uses: docker/login-action@v1
```

with:

username: \${{ secrets.DOCKER\_HUB\_USERNAME }}

password: \${{ secrets.DOCKER\_HUB\_TOKEN }}

- name: Push Docker image

run: docker push your-registry/laravel-app:\${{ github.sha }}

build-test-frontend:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Set up Node.js

uses: actions/setup-node@v2

with:

node-version: '16'

- name: Install dependencies

run: npm install

- name: Run tests

run: npm run test:unit

- name: Build production

run: npm run build

- name: Build Docker image

run: docker build -t your-registry/vue-app:\${{ github.sha }} -f Dockerfile.frontend .

- name: Login to Docker Hub

uses: docker/login-action@v1

with:

username: \${{ secrets.DOCKER\_HUB\_USERNAME }}

password: \${{ secrets.DOCKER\_HUB\_TOKEN }}

- name: Push Docker image

run: docker push your-registry/vue-app:\${{ github.sha }}

deploy:

```
needs: [build-test-backend, build-test-frontend]
runs-on: ubuntu-latest
steps:
- name: Install kubectl
  uses: azure/setup-kubectl@v1

- name: Configure kubectl
  run: |
    mkdir -p ~/.kube
    echo "${{ secrets.KUBE_CONFIG }}" > ~/.kube/config

- name: Deploy to Kubernetes
  run: |
    kubectl set image deployment/laravel-backend laravel-app=your-registry/laravel-app:${{ github.sha }} -n
laravel-vue-app
    kubectl set image deployment/vue-frontend vue-app=your-registry/vue-app:${{ github.sha }} -n laravel-
vue-app
```

# Dockerfile

## Laravel Dockerfile

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

# Install dependencies
RUN apt-get update && apt-get install -y \
  build-essential \
  libpng-dev \
  libjpeg62-turbo-dev \
  libfreetype6-dev \
  locales \
  zip \
  jpegoptim optipng pngquant gifsicle \
  vim \
  unzip \
  git \
```

```
curl \  
libzip-dev \  
libonig-dev \  
libpq-dev  
  
# Clear cache  
RUN apt-get clean && rm -rf /var/lib/apt/lists/*  
  
# Install extensions  
RUN docker-php-ext-install pdo_mysql mbstring zip exif pcntl  
RUN docker-php-ext-configure gd --with-freetype --with-jpeg  
RUN docker-php-ext-install gd  
  
# Install composer  
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer  
  
# Set working directory  
WORKDIR /var/www  
  
# Copy existing application directory contents  
COPY . /var/www  
  
# Install dependencies  
RUN composer install  
  
# Expose port 9000 and start php-fpm server  
EXPOSE 9000  
CMD ["php-fpm"]
```

## Vue Dockerfile

```
# Dockerfile.frontend  
FROM node:16 as build-stage  
WORKDIR /app  
COPY package*.json ./  
RUN npm install  
COPY . .  
RUN npm run build
```





```
FROM nginx:stable-alpine as production-stage
COPY --from=build-stage /app/dist /usr/share/nginx/html
COPY nginx.conf /etc/nginx/conf.d/default.conf
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```



## 1. Kubernetes










```
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
```

## 2. Minikube IP hosts

```
minikube ip
#  IP  hosts 
#  : 192.168.49.2 laravel-vue.local
```

## 3. - : http://laravel-vue.local - API: http://laravel-vue.local/api



1. **Horizontal Pod Autoscaler (HPA):**  Laravel  Vue 
2.  :  Prometheus  Grafana
3.  :  ELK  Loki 