

☐☐ Minikube ☐☐ PHP+Vue



1.1 ☐☐☐☐☐

- ☐☐ ☐ Vue.js ☐☐
- ☐☐ ☐ PHP ☐☐☐☐ Laravel/Symfony☐
- ☐☐☐☐ ☐ MySQL/PostgreSQL
- ☐☐ ☐ Redis
- **CI/CD**☐ Jenkins/GitLab CI
- ☐☐☐☐☐ ☐ Minikube (Kubernetes)
- ☐☐ ☐ Prometheus + Grafana

1.2 ☐☐☐☐



2.1 ☐☐ Minikube ☐☐

```
# ☐☐☐ Ingress☐ Minikube☐☐
minikube start --driver=docker --cpus=4 --memory=8g --disk-size=50g \
  --addons=ingress,metrics-server

# ☐☐☐☐☐☐
```



```
&& docker-php-ext-install zip pdo_mysql
```

```
# [] Composer
```

```
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer
```

```
WORKDIR /var/www
```

```
COPY . .
```

```
# [] PHP[]
```

```
RUN composer install --no-dev --optimize-autoloader
```

```
# [] []
```

```
RUN chown -R www-data:www-data /var/www/storage
```

```
EXPOSE 9000
```

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

3.3 Nginx [] ([] +PHP[])

```
server {  
    listen 80;  
    server_name localhost;  
  
    root /usr/share/nginx/html;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ /index.html;  
    }  
  
    location /api {  
        try_files $uri $uri/ /index.php?$query_string;  
    }  
  
    location ~ /\.php$ {  
        fastcgi_pass php-backend:9000;  
        include fastcgi_params;  
        fastcgi_param SCRIPT_FILENAME /var/www/public$fastcgi_script_name;
```

```
}  
}
```

☐☐ Kubernetes ☐☐☐☐

4.1 ☐☐☐☐ (frontend-deployment.yaml)

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: vue-frontend  
  namespace: php-vue-prod  
spec:  
  replicas: 2  
  selector:  
    matchLabels:  
      app: vue-frontend  
  template:  
    metadata:  
      labels:  
        app: vue-frontend  
    spec:  
      containers:  
        - name: vue-app  
          image: your-registry/vue-app:${IMAGE_TAG}  
          imagePullPolicy: IfNotPresent  
          ports:  
            - containerPort: 80  
      resources:  
        requests:  
          cpu: "100m"  
          memory: "128Mi"  
        limits:  
          cpu: "500m"  
          memory: "512Mi"  
      livenessProbe:  
        httpGet:
```

```
    path: /
    port: 80
    initialDelaySeconds: 30
    periodSeconds: 10
---
apiVersion: v1
kind: Service
metadata:
  name: vue-frontend
  namespace: php-vue-prod
spec:
  selector:
    app: vue-frontend
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

4.2 PHP `deployment.yaml` (backend- deployment.yaml)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: php-backend
  namespace: php-vue-prod
spec:
  replicas: 3
  selector:
    matchLabels:
      app: php-backend
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 0
  template:
    metadata:
```

labels:

app: php-backend

spec:

containers:

- name: php-app

image: your-registry/php-app:\${IMAGE_TAG}

imagePullPolicy: IfNotPresent

ports:

- containerPort: 9000

envFrom:

- configMapRef:

name: php-config

- secretRef:

name: php-secrets

volumeMounts:

- name: storage

mountPath: /var/www/storage

resources:

requests:

cpu: "200m"

memory: "256Mi"

limits:

cpu: "1000m"

memory: "1Gi"

livenessProbe:

exec:

command:

- sh

- -c

- "php /var/www/artisan health:check"

initialDelaySeconds: 60

periodSeconds: 30

volumes:

- name: storage

emptyDir: {}

apiVersion: v1

kind: Service

metadata:

```
name: php-backend
namespace: php-vue-prod
spec:
  selector:
    app: php-backend
  ports:
    - protocol: TCP
      port: 9000
      targetPort: 9000
```

4.3 MySQL (mysql-statefulset.yaml)

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
  namespace: php-vue-prod
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:8.0
          envFrom:
            - secretRef:
                name: mysql-secrets
          ports:
            - containerPort: 3306
```

```
volumeMounts:
- name: mysql-data
  mountPath: /var/lib/mysql
resources:
  requests:
    cpu: "500m"
    memory: "1Gi"
  limits:
    cpu: "2000m"
    memory: "2Gi"
volumeClaimTemplates:
- metadata:
  name: mysql-data
  spec:
    accessModes: [ "ReadWriteOnce" ]
    resources:
      requests:
        storage: 10Gi
```

4.4 Ingress (ingress.yaml)

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: php-vue-ingress
  namespace: php-vue-prod
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
  - host: php-vue.local
    http:
      paths:
      - path: /(.*)
        pathType: Prefix
      backend:
        service:
          name: vue-frontend
```

```
    port:
      number: 80
  - path: /api/(.*)
    pathType: Prefix
  backend:
    service:
      name: php-backend
      port:
        number: 9000
```

☐☐ CI/CD ☐☐☐☐☐

5.1 Jenkinsfile ☐☐

```
pipeline {
  agent {
    kubernetes {
      yml """
apiVersion: v1
kind: Pod
spec:
  containers:
  - name: node
    image: node:16
    command: ["cat"]
    tty: true
  - name: php
    image: composer:2
    command: ["cat"]
    tty: true
  - name: docker
    image: docker:20.10
    command: ["cat"]
    tty: true
  volumeMounts:
  - name: docker-sock
```

```
    mountPath: /var/run/docker.sock
- name: kubectl
  image: bitnami/kubectl:latest
  command: ["cat"]
  tty: true
volumes:
- name: docker-sock
  hostPath:
    path: /var/run/docker.sock
""""
  }
}

environment {
  DOCKER_REGISTRY = "your-registry"
  FRONTEND_IMAGE = "${DOCKER_REGISTRY}/vue-app"
  BACKEND_IMAGE = "${DOCKER_REGISTRY}/php-app"
  K8S_NAMESPACE = "php-vue-prod"
  // Jenkins
  DOCKER_CREDS = credentials('docker-hub-creds')
  KUBE_CONFIG = credentials('kube-config')
}

stages {
  stage('Checkout') {
    steps {
      git branch: 'main', url: 'https://your-repo.git'
    }
  }

  stage('Build Frontend') {
    steps {
      container('node') {
        sh 'npm install'
        sh 'npm run build'
      }
    }
  }
}
```



```

sh "sed -i 's|\\${IMAGE_TAG}|${GIT_COMMIT}|g' k8s/frontend-deployment.yaml"
sh "sed -i 's|\\${IMAGE_TAG}|${GIT_COMMIT}|g' k8s/backend-deployment.yaml"

// 📦 K8S📦
sh "kubectl apply -f k8s/ -n ${K8S_NAMESPACE}"

// 📦📦📦📦
sh "kubectl rollout status deployment/vue-frontend -n ${K8S_NAMESPACE}"
sh "kubectl rollout status deployment/php-backend -n ${K8S_NAMESPACE}"
}
}
}
}
}

post {
  always {
    cleanWs()
  }
  failure {
    slackSend channel: '#ci-alerts', message: "Build Failed: ${env.JOB_NAME} ${env.BUILD_NUMBER}"
  }
  success {
    slackSend channel: '#ci-alerts', message: "Build Succeeded: ${env.JOB_NAME} ${env.BUILD_NUMBER}"
  }
}
}
}

```



6.1 📦📦📦📦📦📦





```






# 📦 backend-deployment.yaml📦📦  initContainer
initContainers:
- name: migrate
  image: your-registry/php-app:${IMAGE_TAG}
  command: ["sh", "-c", "php artisan migrate --force"]

```





```
envFrom:
- configMapRef:
  name: php-config
- secretRef:
  name: php-secrets
```

6.2

```
#   ConfigMap 
kubectl create configmap php-config --from-env-file=.env -n php-vue-prod

#   ConfigMap  
annotations:
  configmap.reloader.stakater.com/reload: "php-config"
```

6.3

```
#   Prometheus 
metadata:
  annotations:
    prometheus.io/scrape: "true"
    prometheus.io/port: "80"
```







7.1 Skaffold


```
# skaffold.yaml
apiVersion: skaffold/v2beta16
kind: Config
profiles:
- name: dev
  activation:
    - kubeContext: minikube
```

```
deploy:
  kubectl:
    manifests:
      enabled: false
portForward:
- resourceType: service
  resourceName: vue-frontend
  port: 80
  localPort: 8080
- resourceType: service
  resourceName: php-backend
  port: 9000
  localPort: 9000
```

7.2



```
#  Ingress
minikube tunnel


#  hosts
echo "$(minikube ip) php-vue.local" | sudo tee -a /etc/hosts

# 
open http://php-vue.local
```



8.1

```
#  Pod
kubectl get pods -n php-vue-prod -w

# 
kubectl logs -f deployment/php-backend -n php-vue-prod
```

