

# 部署文档

---

## 一、部署环境

- 系统部署结构

- 三节点 Kubernetes 集群

- 服务器配置

- 主节点
    - CentOS 7.8 (只要 cgroups 支持 systemd 的 linux 系统即可)
    - 4 核 CPU, 64G 内存处理器
    - Docker + Kubernetes三件套, 创建主节点
    - 参与负载均衡
  - 从节点 (两台)
    - CentOS 7.8 (只要 cgroupdriver 支持 systemd 的 linux 系统即可)
    - 4 核 CPU, 64G 内存处理器
    - Docker + Kubernetes三件套, 加入主节点创建的集群
  - Mysql 服务器
    - CentOS 7.8
    - 4 核 CPU, 64G 内存处理器
  - Redis 服务器
    - CentOS 7.8
    - 4 核 CPU, 64G 内存处理器
  - Gitlab 服务器
    - CentOS 7.8
    - 4 核 CPU, 64G 内存处理器
    - 安装 Gitlab 并连接镜像站, 在有新的项目提交的时候打包更新项目为新的 Docker 镜像并对镜像站上的原镜像进行更新, 实现持续集成
  - harbor 镜像站
    - CentOS 7.8
    - 4 核 CPU, 64G 内存处理器
    - 存储项目 docker 镜像
    - 连接 Kubernetes 集群, 在 Docker 镜像更新或 pod 损坏时将对应的 Docker 镜像转为 Runtime 部署到集群中, 实现持续交付。
- 客户端配置
  - Chrome 或 Firefox 浏览器

## 二、部署方式

因为 laravel 框架基于 php, 所以不需要像springboot框架项目一样打包运行, 项目配置完成仅需要做好代理就能运行

前端 build 文件放在后端入口文件夹中, 将入口代理到后端入口文件夹中的前端入口文件

- 仅需要对后端项目进行 CI/CD
- 生成两个pod

- 需要一个域名

### 三、部署流程

- **初始化（以下步骤仅需设置一次）**

- 编辑与提交 gitlab 持续集成所需要用到的 ops 文件以及项目所需要的 Dockerfile 文件
- 设置 gitlab CI/CD 步骤
- 在需要部署的项目中编辑对于上述步骤的映射文件
- 设置项目具体env参数（如laravel框架的.env文件，springboot框架不需要设置）

- **自动部署**

- 当发现新的提交，gitlab 会通过当前服务器上的 docker 和存储的 Dockerfile 文件对最新提交的代码打包为镜像并对镜像站上的包进行添加或更新。（CI）
- 当持续集成完成，镜像站镜像更新，镜像站会对集群中的运行的 pod 进行更新（更新中不会影响原来的服务，这是持续交付的特性）。（CD）