



WAKEDATA

惟客数据

# 标准化项目创建、构建和部署

入门+实战

# Content

**/01 项目创建**

**/02 代码检查**

**/03 项目构建**

**/04 项目部署**

# /00

## 背景

客户经营，找惟客



# 项目开发流程的标准化

过去两年我们做的一些事情：

- 项目迁移到 gerrit, 开启 Code Review 的大门
- 自动化代码检查
- [规范](#)
  - 技术栈规范
  - UI 规范
  - 代码规范
  - [分支规范](#)
- [CCC 认证](#)
- 项目部署容器化
- ....

# 项目开发流程的标准化

## 项目工程化进一步深化

- 微前端框架/基础库
- 运行容器
- 镜像构建规范
- 一键部署



## 镜像构建规范

- 镜像 `wkfe/<type>-<name>-[INDUSTRY]`
- 版本号
  - 遵循[惟客云产品版本规范](#)
  - 后端对应模块的版本号保持同步
  - 测试版本以 `-snapshot-<BUILD>` 为后缀
- 发布
  - 测试版本 `172.26.59.200/wkfe/*`
  - 生产版本 `ccr.ccs.tencentyun.com/wkfe/*`
- K8S 工作负载、Sidecar
  - `wkeadmin` 微前端基座
  - `wkeadmin-*` 其他后台独立服务，例如 `wkeadmin-ma`
  - `wkeapp-*` C 端服务，比如 H5 页面、小程序 CDN 资源
  - `wkeadmin-apps` 微前端子应用
    - sidecar 使用镜像名称命名，`wkfe/app-dsp`，sidecar 就命名为 `app-dsp`
- Jenkins Job
  - `check-<仓库名称>` 代码检查
  - `build-<仓库名称>` 项目构建



## 运行容器与镜像的标准化

现有的问题？

- 深度依赖编译，前端项目部署到具体环境需要重新构建，比如设置 cdn 地址
- 没有开箱即用的镜像
- 没有版本管理

运行容器？

- 运行容器是一个前端的运行环境，支持在部署时做一些‘动态’的事情，比如替换域名、注入监控脚本、变更主题

标准化？

- 只有一个镜像。不管是哪个环境，只有一个镜像
- 标准化也有规范化的意思

# /01

## 项目创建

客户经营，找惟客





## 项目脚手架

```
pnpm create wakeadmin
```

# /02

## 代码检查

客户经营，找惟客



```
sh ./scripts/check.sh
```



## 有什么区别?

```
sh ./scripts/check.sh
```

- 由仓库自己来管控检查的过程，更加灵活，尤其是不同分支行为不一致的情况下
- 持久化、自包含

# /03

## 项目构建

客户经营，找惟客



# 项目构建

```
sh ./scripts/build.sh
```

Docker 构建、发布、更新

# /03

## 项目部署

客户经营，找惟客



Rancher: 创建工作负载

Rancher: Ingress 负载均衡配置

[案例](#)

# Thanks

