# 部署与 CI/CD（含移动端上架）

目标：提供可重复部署流程，并以自动化测试作为门禁，确保质量；同时覆盖 Android/iOS 客户端的签名、打包与商店上架。

## 任务清单

1. Docker 化后端；多阶段构建
2. 移动端构建流水线（Android/iOS）与签名管理
3. 环境配置与密钥注入（后端与客户端各自策略）
4. CI：安装依赖、构建、测试、产物缓存
5. CD：后端部署（Cloud Run/Fly.io/EC2 等）与客户端分发（内部测试/商店）
6. 监控与告警（日志、指标、崩溃/性能）

## 后端实施建议

* GitHub Actions：
* on: push 到 main；
* 任务：lint -> test -> build -> docker push -> deploy；
* 缓存 pnpm/poetry 等依赖。
* 环境：Cloud Run/Fly.io/EC2/K8s；
* 测试：部署后健康检查 /healthz 与合成监控。

## 移动端流水线

* Android：
* 签名：keystore + gradle 签名配置（使用 Secret 管理）；
* 构建：flutter build appbundle --release 产出 AAB；
* 分发：Google Play Console 内测渠道（Alpha/Beta）；
* 自动化：Actions 使用 subosito/flutter-action 安装 SDK，缓存 pub 依赖；上传 AAB 到 Play（r0adkll/upload-google-play）。
* iOS：
* 证书/描述文件：使用 App Store Connect API + fastlane match/app-store-connect；
* 构建：flutter build ipa --release；
* 分发：TestFlight 内测；
* 自动化：fastlane gym/deliver 或 app-store-connect 的 CLI/Actions。

## 安全与配置

* 客户端敏感信息不硬编码；仅使用后端下发可公开配置；
* 后端密钥托管于 Secrets Manager；
* 证书与签名材料通过专用仓库存取（只读权限），拉取到 CI 运行器的安全路径。

## 质量门禁

* 客户端：flutter analyze、flutter test、golden 测试、集成测试；
* 服务端：单元/集成/契约测试；
* 覆盖率阈值与变更集风险报告。

## 发布策略

* 分支：main 合并触发后端生产部署与客户端内测分发；
* 灰度：小流量内测，采集 Crashlytics/Performance 指标；
* 审核：通过后再面向全量商店发布。