DevOpsDays Beijing

DevOpsDays

2017 DevOpsDays Beijing

时间：2017.03.18 地点：北京朝阳区万达索菲特大酒店

主办单位：高效运维社区 GreatOPS Community　Best Practice 最佳实践

基于Docker的DevOps流水线

徐磊
英捷创软 首席架构师

讲师简介

Speaker Introduction

徐磊

英捷创软 CEO
首席架构师
微软最有价值专家 MVP
Microsoft Regional Director
Certified ScrumMaster

LEANSOFT首席架构师

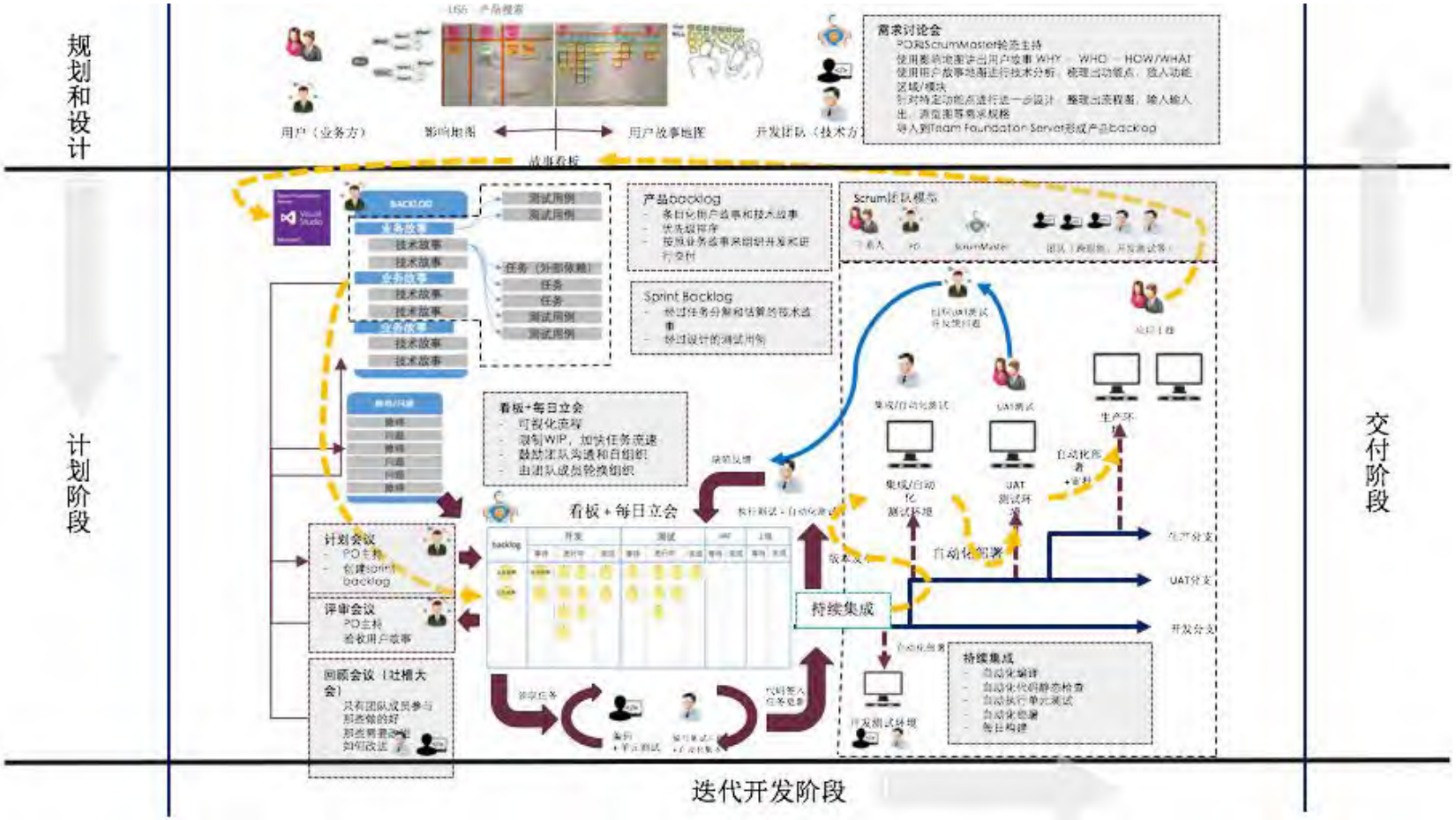
徐磊

英捷创软 CEO 首席架构师 | 资深ALM顾问和解决方案专家 | 微软最有价值专家 | 大中华区域社区技术总监 | ScrumMaster | 敏捷教练 | 超过10年的软件研发项目管理经验 | 曾任SSW中国研发中心总经理

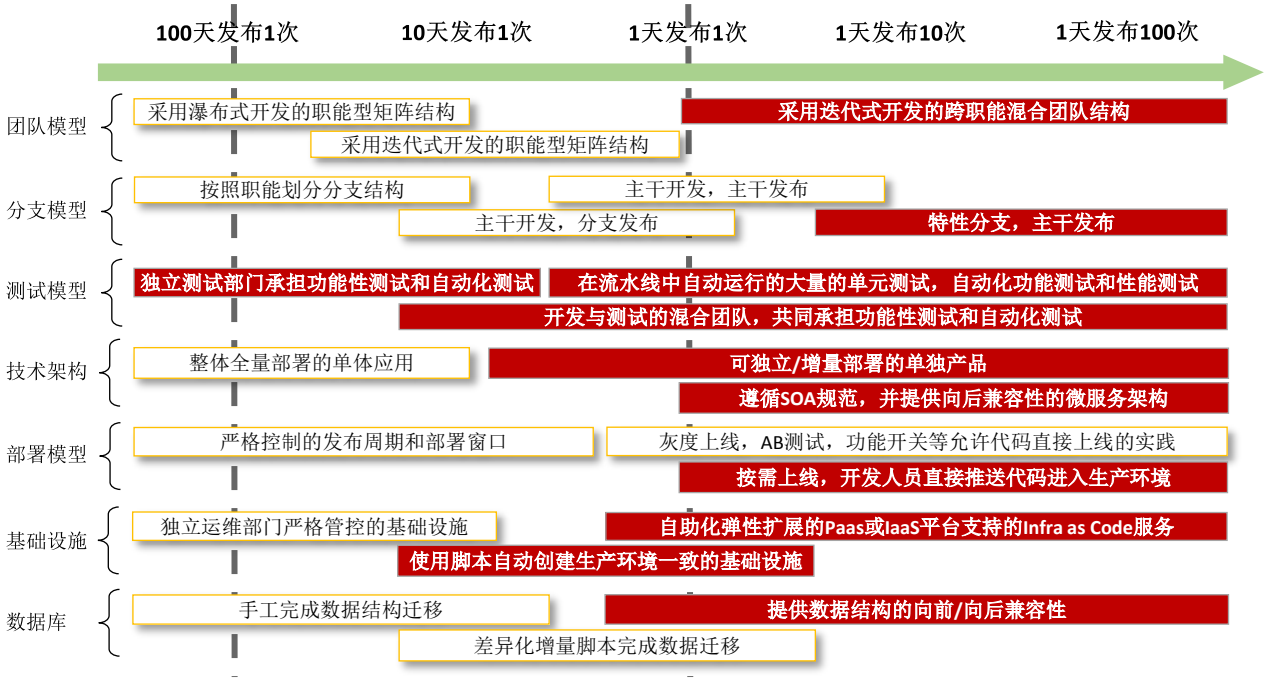
项目经验

- 高级ALM/DevOps顾问, 中国农业银行，互联网金融项目
- 敏捷教练, 兴业银行，海外网银敏捷试点项目
- 高级管理顾问，中国移动南方基地，互联网研发基地外包管理改进
- 高级ALM/DevOps顾问，上海通用汽车有限公司，软件研发过程改进
- 高级ALM/DevOps顾问，上海汇众汽车，软件研发过程改进和ALM平台落地
- 高级ALM/DevOps顾问，中国人民保险公司软件研发中心，软件研发过程改进和ALM平台落地
- 高级ALM/DevOps顾问，斯伦贝谢中国研发中心，敏捷开发与ALM落地
- 高级ALM/DevOps顾问，京东商城，PMO管理系统与ALM系统集成 高级ALM/DevOps顾问，
- 华为2012实验室（深圳），rDelta系统选型
- 高级ALM/DevOps顾问，百威英博，软件外包管理系统ALM落地实施

UDAD 用户故事驱动的DevOps实施框架



持续交付 – 容器的定位/价值



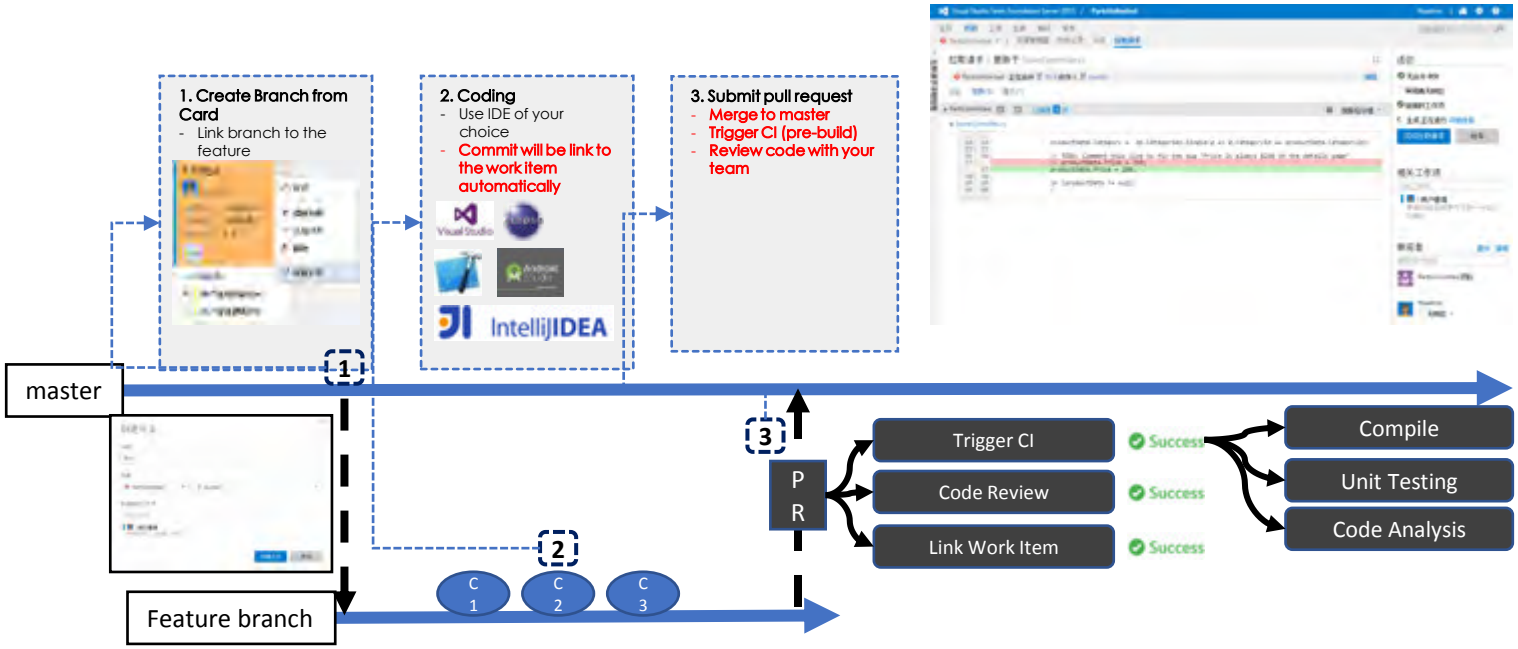
话题1：分支对持续交付的影响

传统分支模型: M-D-R

- 基于团队内部职能 / 技能的分支模型
- 瀑布开发
- 对迭代式开发不能很好支持
- 限制了交付速度，造成在制品积压
- 减缓反馈速度，增加了团队内部沟通成本

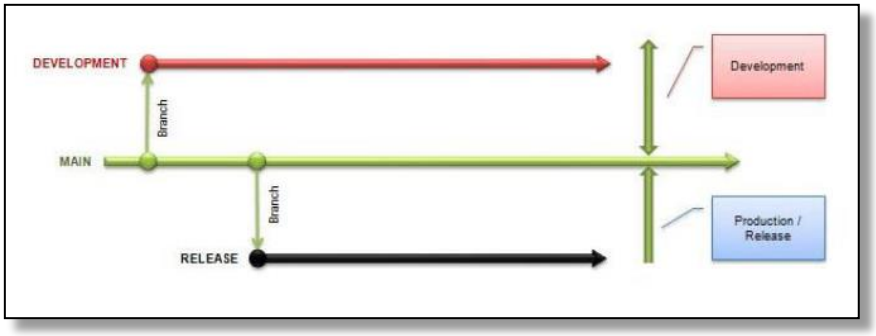
特性分支 – PR – 质量门

- 基于对外交付的分支模型，为按特性交付提供了支撑
- 跨职能团队模型，高效率的内部沟通减少中间环境浪费，快速消耗在制品
- 保持主分支一直处于可发布状态

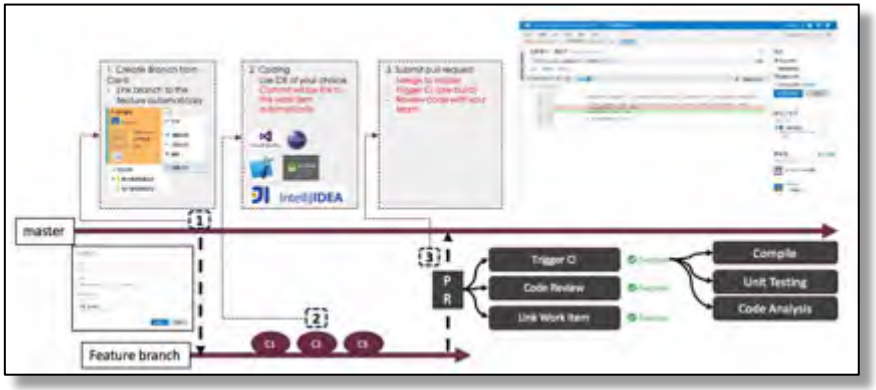


2种分支模型的比较

- 传统 M-D-R 模型
  - 面向内部流程
  - 与阶段绑定
  - 无法按照交付挑选代码修改
- 特性模型
  - 面向外部交付
  - 与用户场景绑定
  - 延迟决策



Traditional M-D-R Dodel



Feature Branch Model

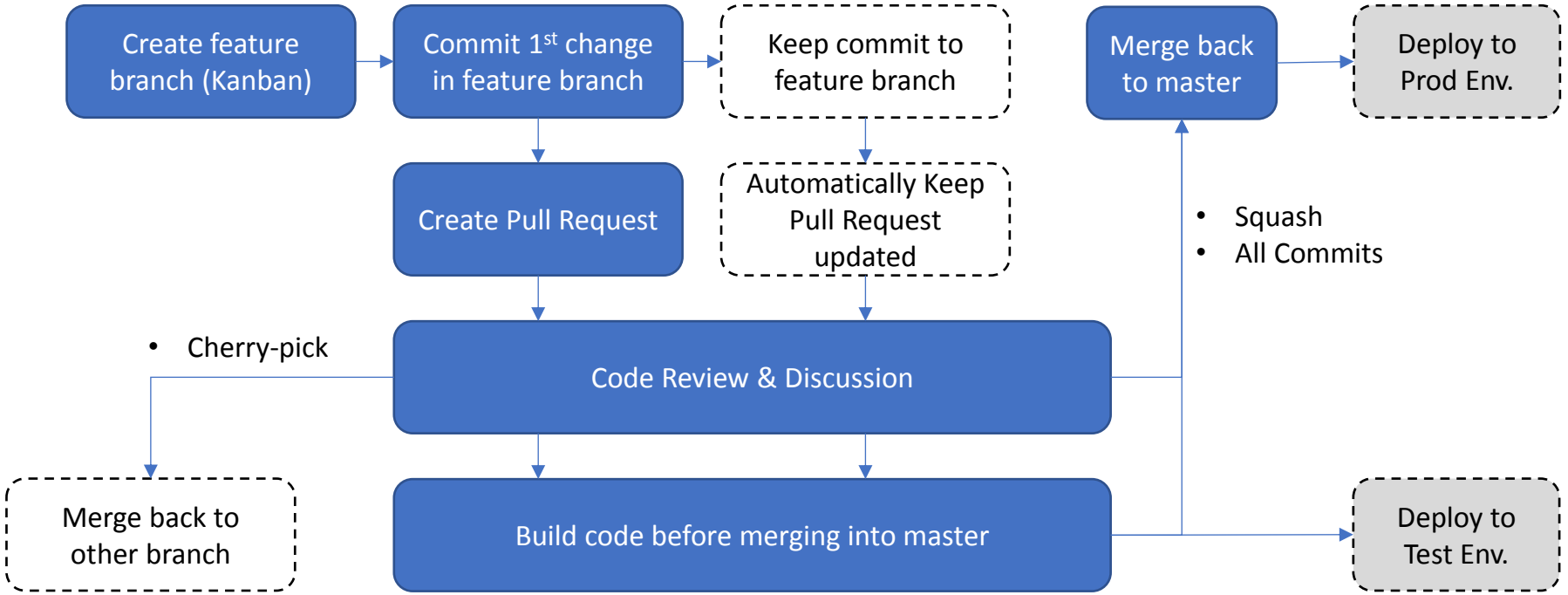
特性分支帮助你更好适应这些场景

- 并行开发，团队解耦
- 按用户需求进行交付
- 独立测试每个特性
- 保持主干代码的整洁
- 简化Ci流程
  - 为每个feature建立独立的Ci不在需要在每个feature branch上单独配置
  - 及时与生产代码进行集成

为什么要使用Pull Request?

- 审核改动，不要审核提交
- 可视化变更过程
- 预构建（在代码没有合并之前验证合并后的代码）

Pull Request 工作流



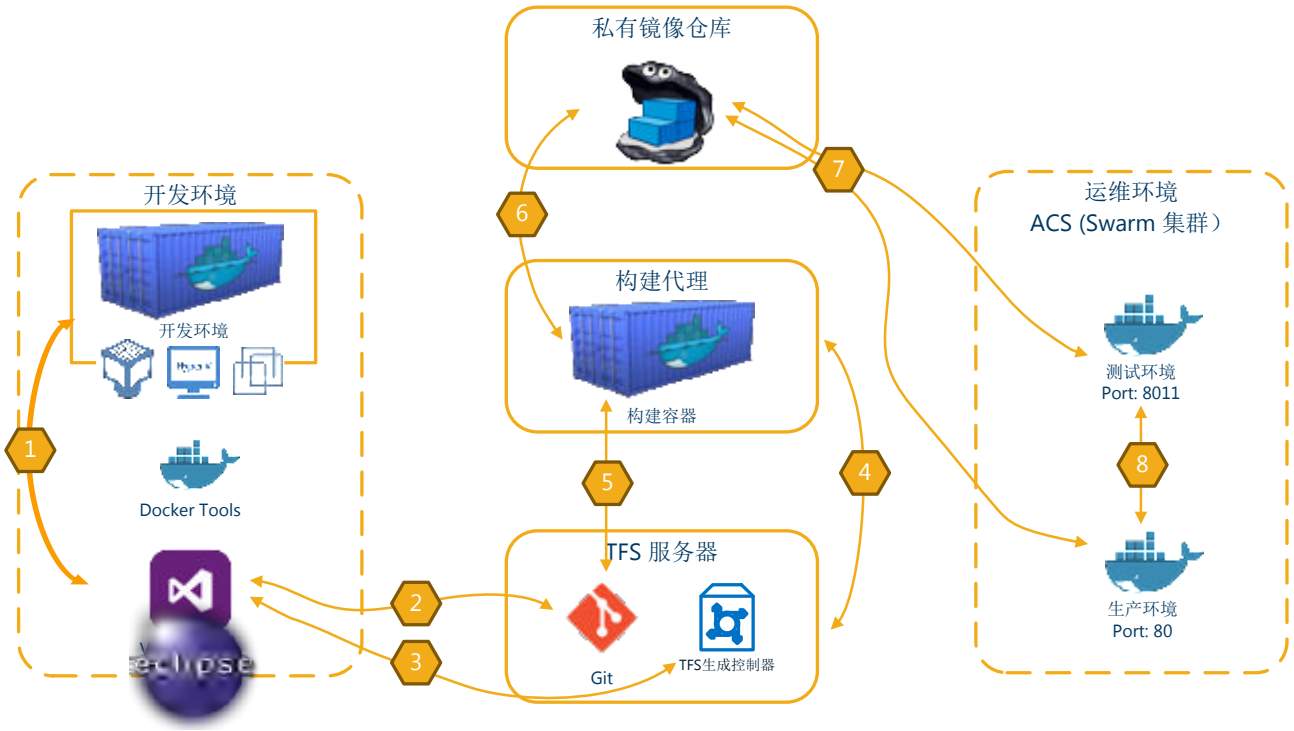
话题2：容器化开发调试的挑战

容器对DevOps的价值

- 一次构建，多处运行
  - 干净的，环境独立的，可迁移的运行平台
  - 在多次部署中不必担心依赖，包含环境相关配置
  - 隔离性，同时运行不同版本的库和依赖环境，而不用担心他们互相影响
  - 自动化测试，集成，打包过程；全部可以通过简单的脚本实现
  - 降低与不同应用运行平台的兼容性问题
  - 享受VM所提供的隔离性，快照等能力，同时又不被笨重的VM所拖累
- 配置一次，运行任何应用
  - 让应用生命周期管理变得更加高效，统一可复制
  - 提升开发人员的代码质量
  - 消除开发，测试，生产和客户定制化环境的差异性
  - 为不同职能/技能的人员各司其职提供了条件
  - 大大提升CI/CD的可靠性，速度和可复制性
  - 应为容器非常轻量，VM所存在的性能，成本，部署和可迁移性问题都迎刃而解

基于Docker的DevOps流水线

1. 编码并使用本地容器环境进行调试
2. 提交代码到git
3. 开发人员触发CI/CD过程
4. TFS 驱动构建代理执行编译，打包
5. 构建引擎从git获取代码和配置
6. 将容器镜像上传到私有仓库
7. 将容器镜像部署到不同个环境
8. QA/管理人员控制版本升级过程

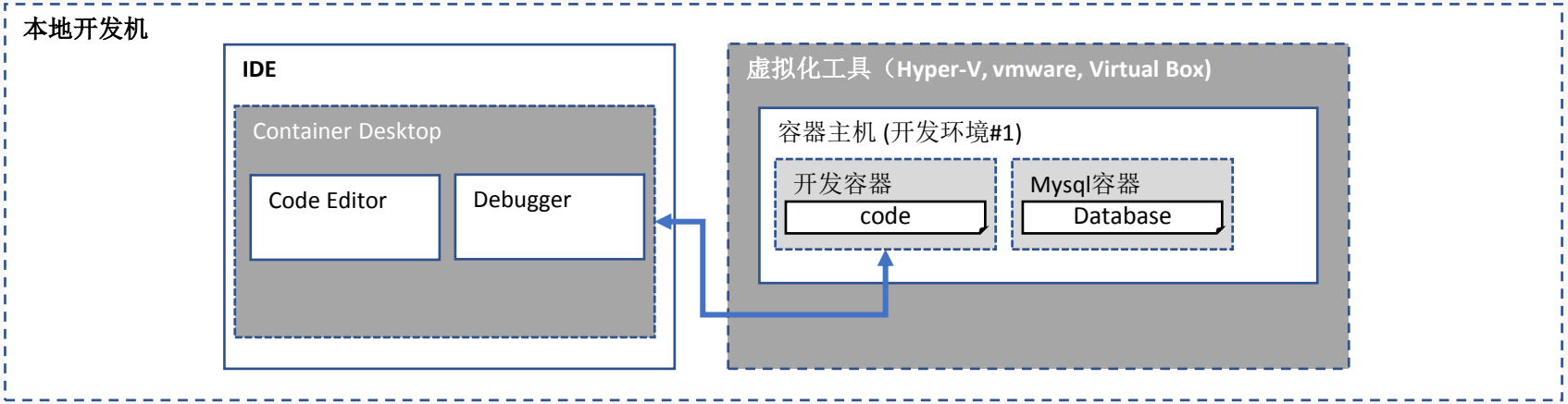


开发/测试环境放入容器？

- 接手新项目，半天时间都搭不起环境？
- 改了项目配置，别的开发人员就跑不起来？
- 版本准备好了，测试环境坏掉了？
- 想使用Docker，担心直接上生产有风险？

容器化你的开发环境

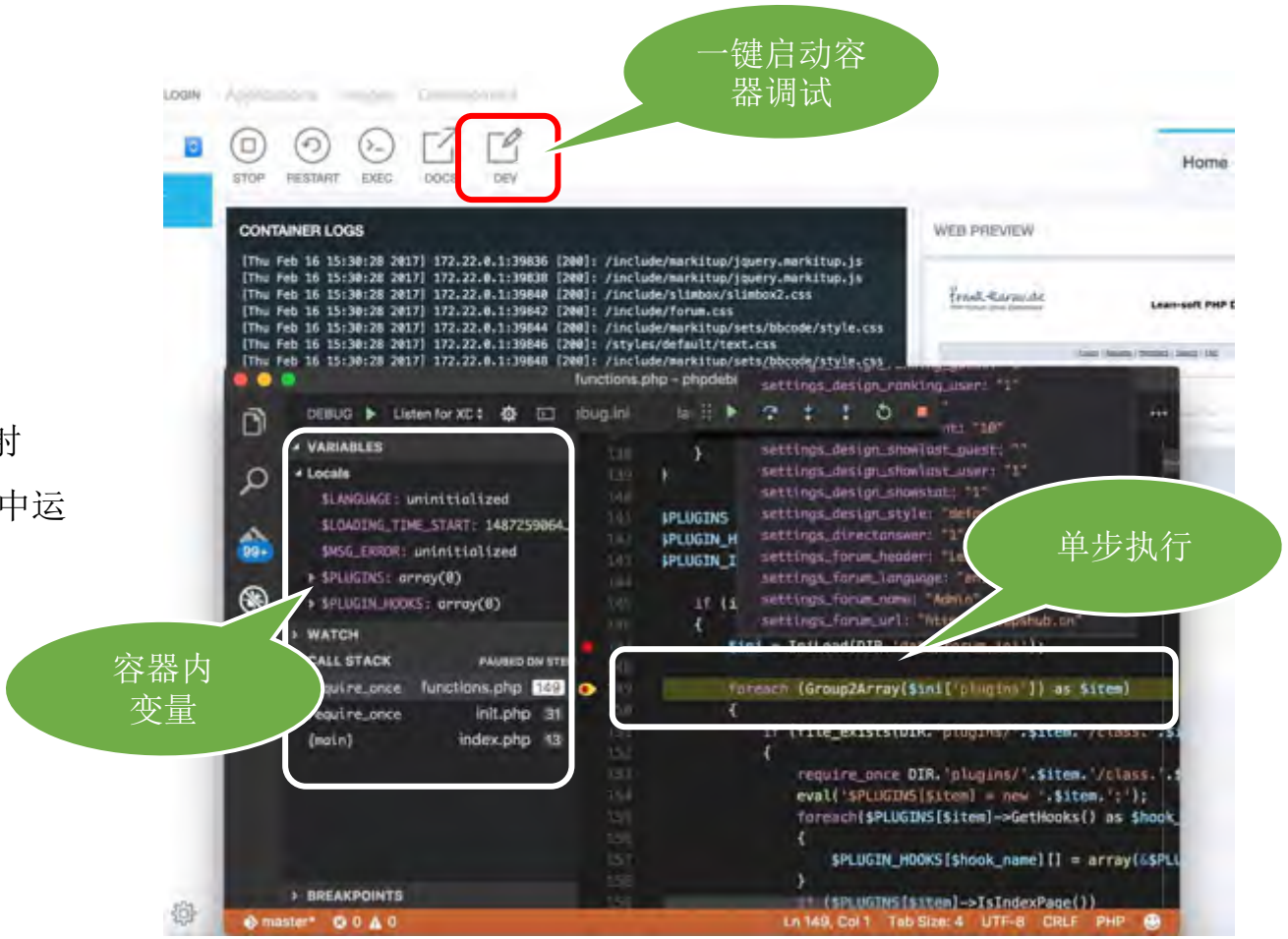
- 创建完全隔离的开发环境容器
- 一键编译，打包镜像，部署镜像并启动调试
- 设置断点，单步调试运行在容器中的代码
- Dev Environment as Code (Infrastructure as Code)
- 环境和代码一通保存，代码版本对应镜像版本
- 快速复制开发环境，测试环境
- Code Ready for Production



扫码阅读文档

IDE 集成单步调试

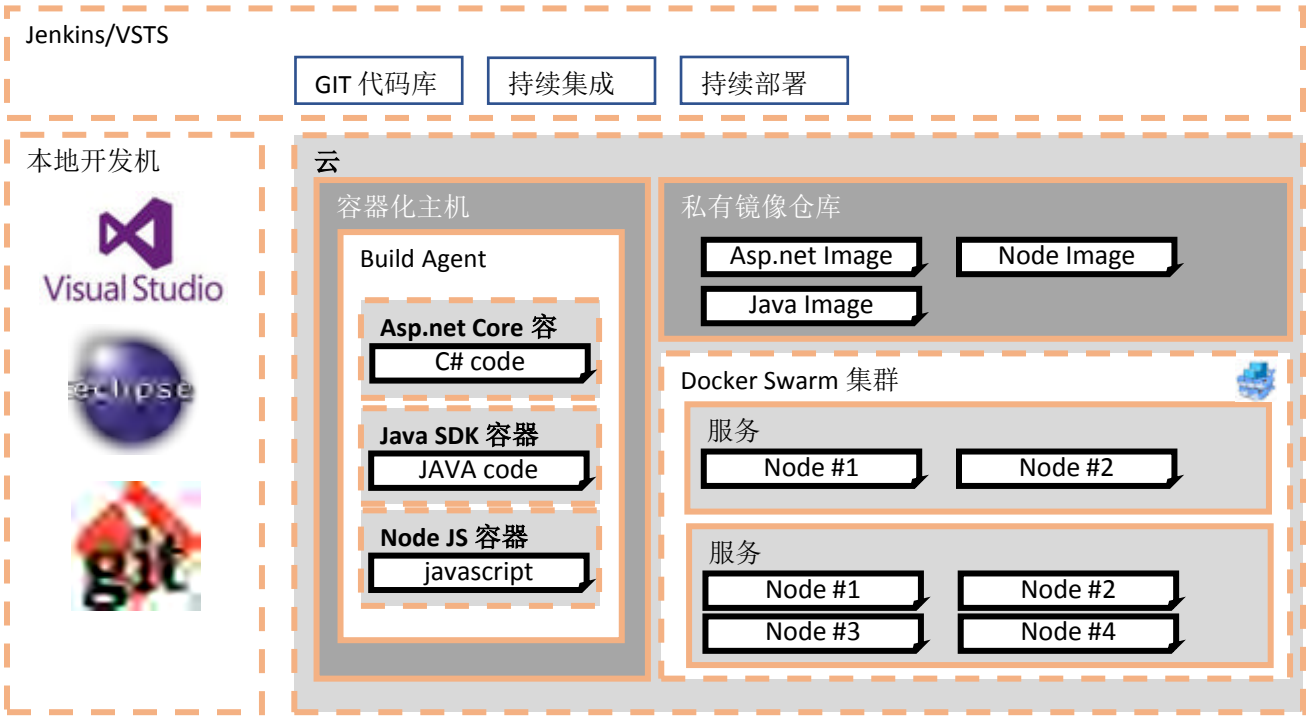
- 一键启动容器进行调试
- 自动启动IDE，完成vol映射
- 在vscode中单步调试容器中运行的应用程序



话题3：容器化DevOps流水线的机遇

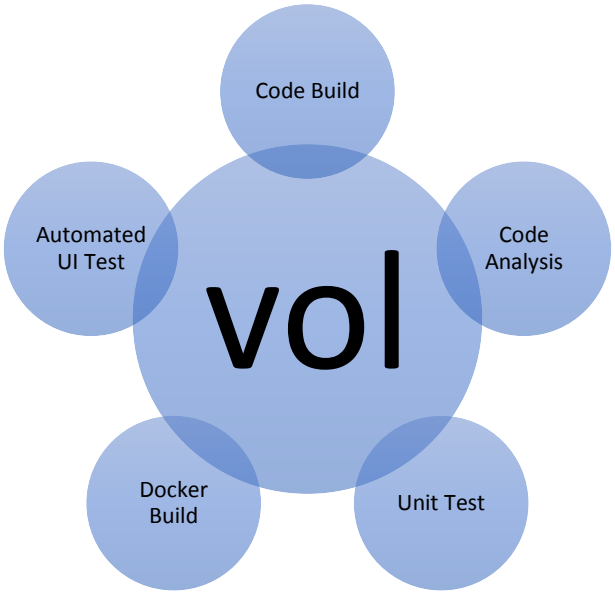
DevOps Pipeline

- 使用Git/Jenkins/VSTS作为持续集成(CI)和持续部署(CD)工具
- 驱动运行Docker主机中的Build Agent完成镜像构建，推送及部署
- 借助Docker Swarm的集群管理能力进行自动扩容



容器化构建

- 使用容器来构建容器
- 围绕vol构建工作容器，按需创建，按需销毁



高效运维社区

GreatOPS Community

GOPS 4月深圳 / 7月北京 / 11月上海

EXIN DevOps Master 认证研修

DevOpsDays 3月北京 / 8月上海

DevOps 企业内训 / 咨询服务

DevOps China 全国巡回技术沙龙

其他量身定制服务项目

商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com

Thanks

荣誉出品

高效运维社区

国际最佳实践管理联盟

DevOpsDays Beijing