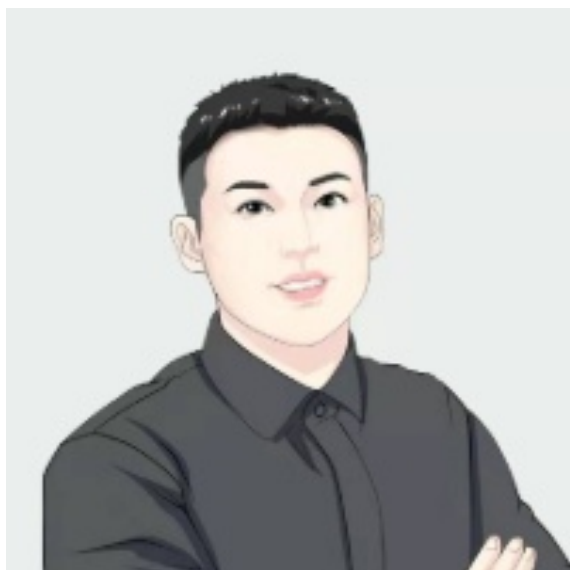


① 微服务CI/CD训练营



讲师简介



泽阳，在DevOps的实践过程中踩过了无数的深坑，积累了丰富的填坑经验。为了让大家顺利的实践DevOps工程，暂将大部分坑的解决方法整理在《Jenkins实践》、《GitlabCI实践》、《Spinnaker实践》课程中。

课程背景

近几年DevOps的发展迅速，持续集成（CI）和持续部署（CD）作为DevOps的核心相信大家都很熟悉了。更多公司开始使用CI/CD服务来解决频繁集成与发布问题。

随着服务架构由单体到微服务架构，传统的VM虚拟机到容器化技术飞速发展。对持续交付工具的需求越来越多。微服务带来了许多优点，例如：将复杂的单体服务根据功能拆分成若干个微服务，每个服务可以独立部署互不影响。微服务的不断增多，在进行集成和部署的工作时通常需要花费大量的协调时间，这使微服务模式下的持续交付变得越来越复杂。

课程内容

- 理论篇： 单体应用到微服务架构所面临的挑战与问题。
- 实践篇： 基于DevOps工具链落地CI/CD。自动化一切可自动化操作。

模块一： 微服务简介

了解微服务的基本特点和CI/CD遇到的一些挑战。掌握微服务的基础知识即可。

模块二： 微服务模式下的需求管理

需求如何管理？ 采用**SCRUM**项目，创建任务故事与代码关联对应的服务模块。

模块三： 微服务模式下的代码管理

代码库如何拆分？ 多个仓库模式和单个仓库。版本控制系统的分支开发策略如何选择？（这部分会与需求管理做集成实践：多服务模式 避免手动创建分支，自动化创建分支，需求与代码关联，自动合并代码。）

课程内容

模块四： 微服务模式下的持续集成

持续集成流水线设计与实现，提交流水线实现（单个git仓库如何根据不同的子目录触发对应的流水线构建？）多模块如何进行**代码扫描**？ 代码扫描结果关联Commit信息控制代码合并。（例如代码质量阈失败，不允许合并代码。）

模块五： 微服务模式下的持续部署

持续部署是在持续集成的基础上，可以使用sed替换yaml使用**kubectl**部署，或者使用**Helm**部署，也可以使用专业的**CD工具**完成交付（包含部署策略）。

模块六： 总结

总结课程所有理论内容，完成整个**CI/CD**框架流程的实现。

学习指南

本次课程我将手把手带领大家完成流水线设计与搭建。这个课程更多是大家的交流与分享。

开营时间： 2021年 02月06日 - 07日

直播时间：

- 02月06日
 - 上午：10:00-12:00 （简介与需求管理）
 - 下午：14:00-16:00 （代码与需求管理）
 - 02月07日
 - 上午：10:00-12:00 （持续集成）
 - 下午：14:00 - 16:00 （持续部署）
-

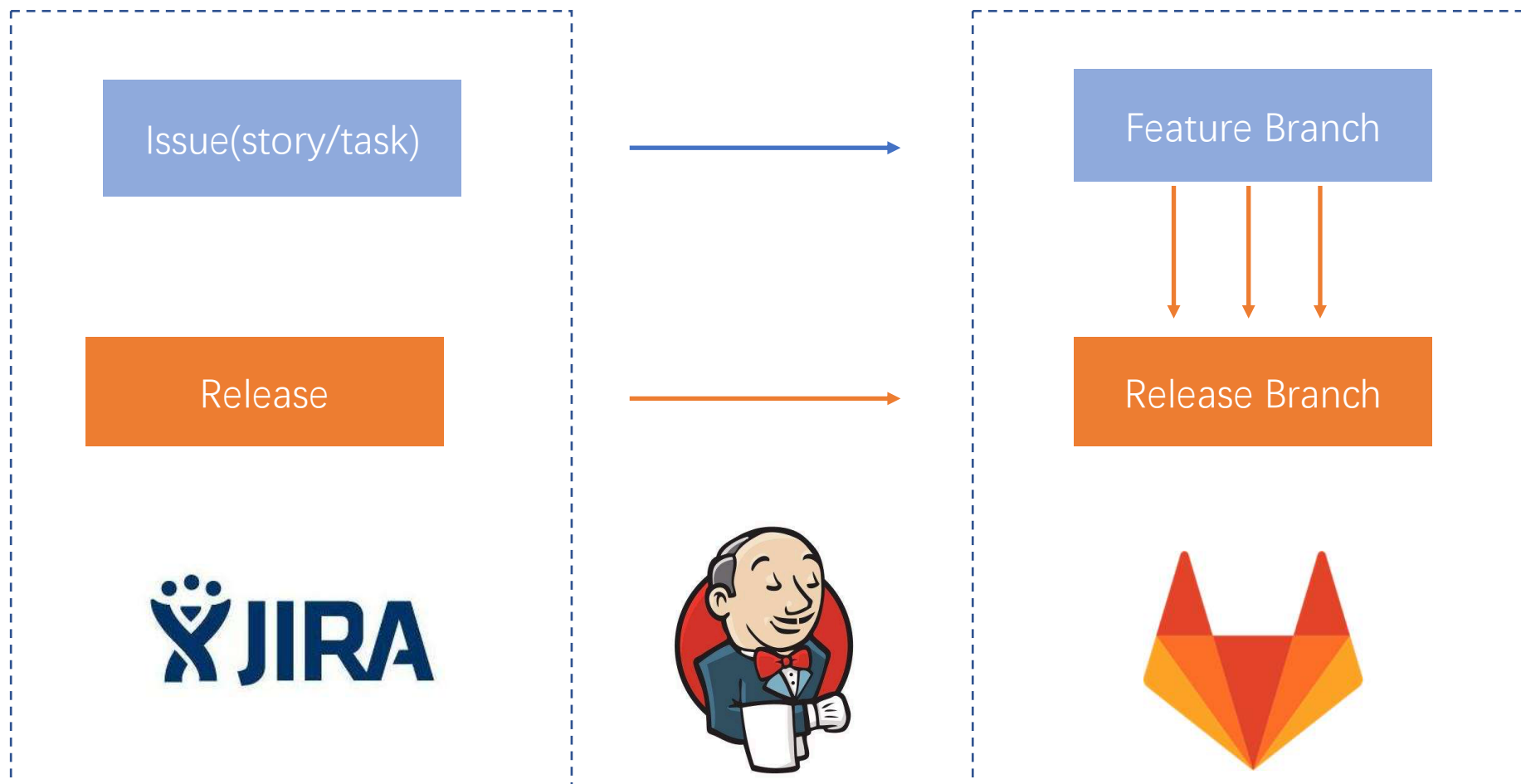
微服务带来的影响

微服务划分的影响，如下：

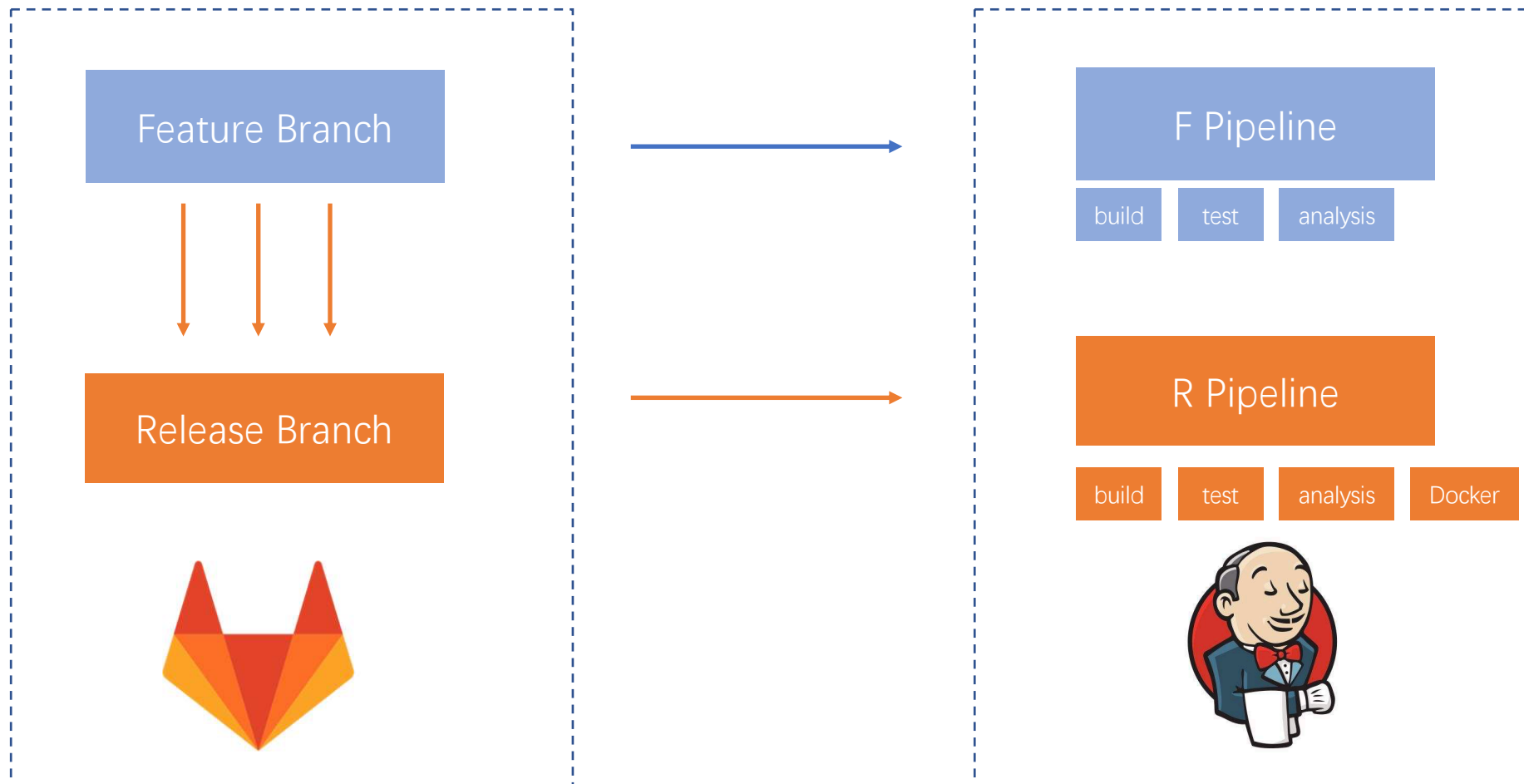
- 需求管理：哪个任务/故事需要关联到哪个服务？手动关联？
- 代码管理：代码库如何划分？一个代码库？多个代码库？ 分支策略？
- 持续集成： CI流水线该如何设计？ 一条？两条？多条？.....
- 持续部署： CD流水线该如何设计？ 一条？两条？多条？.....

在微服务持续集成的过程中，微服务测试是非常重要的。没有充分的单元测试就无法确定服务是否可靠的。当然不能仅仅是单元测试，考虑到复杂性，本课程主要是想分享给大家一个基础的CI/CD交付框架，提供核心的实现，但部分细节问题还需在实际环境中补充完善。

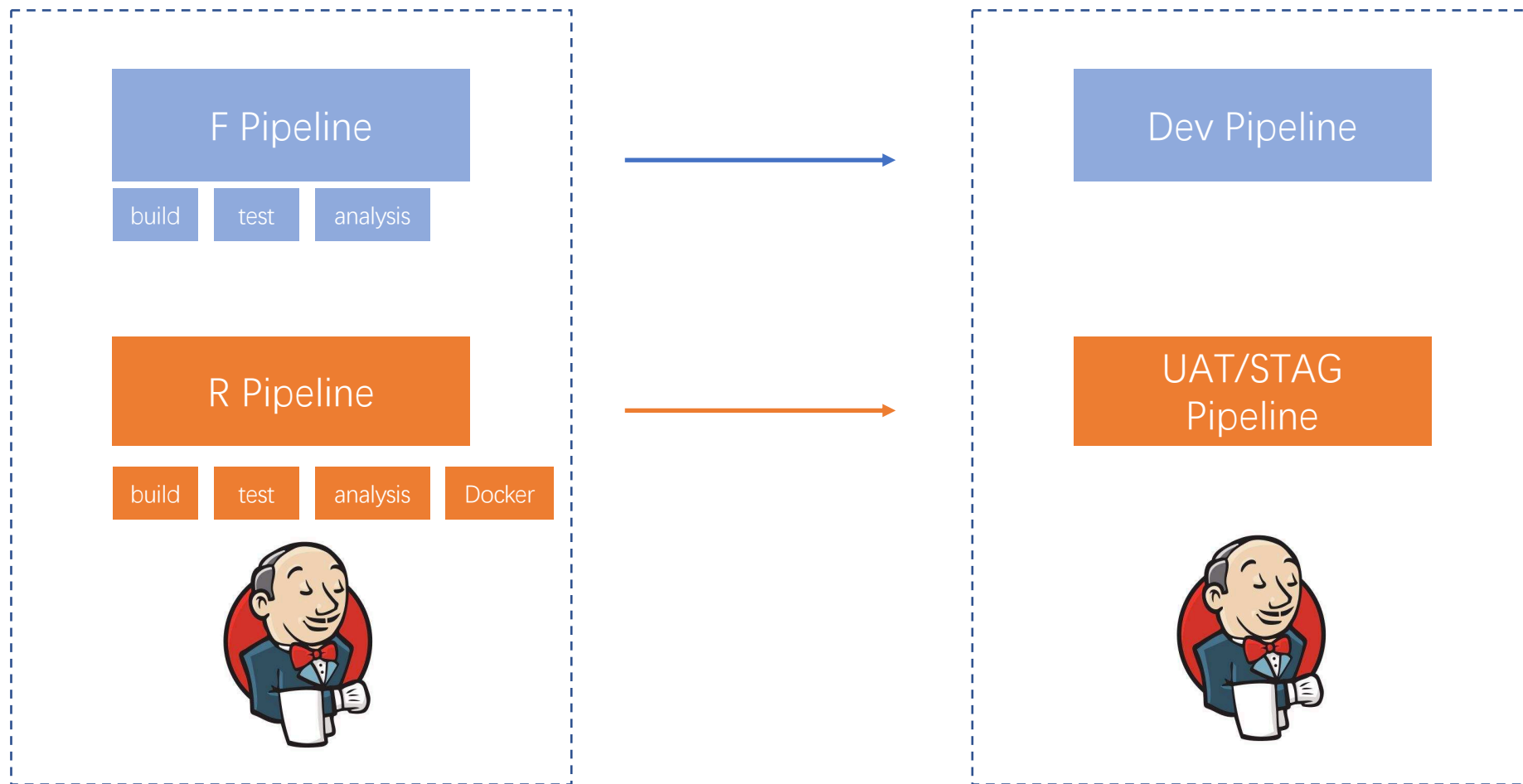
需求与代码集成



代码与持续集成



持续集成与持续部署



准备工作

- 一套DevOps实践环境
 - Jira + GitLab + Jenkins + SonarQube + Helm + Kubernetes
- 有Jenkins Pipeline基础更佳

