



智能网络未来进化  
IA52017 · NANJING

运筹帷幄  
从容自如



IT大咖说

基于容器的持续集成平台

华泰证券 管文琦

- 华泰证券股份有限公司是一家中国领先的综合性证券集团，具有庞大的客户基础、主要财务指标和业务指标在国内排前五
- 主营业务：经纪及财富管理、投资银行、投资及交易、资产管理、海外业务

# 华泰证券互联网战略

十年磨一剑 华泰证券打造券商互联网金融标杆

我们的目标：

# 敏捷交付

持续创新的保证

DEV OPS



## 敏捷交付三大技术支撑



平台化的基础设施



拥抱DevOps 最佳实践



微服务软件架构

## ■ 平台化的基础设施



# 什么是平台化的基础设施？



# 基于DevOps 最佳实践的 交付能力



## DevOps 内涵



鼓励沟通与协作的文化、  
思维方式与组织架构



一系列工程化的最佳实践  
( 持续集成、持续交付流  
程、方法论 )



落实最佳实践的  
工具和平台



要衡量一个团队的DevOps能力如何，只需要问一个问题：  
**上线一次应用有多难？**

## 持续集成、持续交付流水线



多数成功实践DevOps与持续交付的  
团队选择采用 **微服务架构**

## 微服务支持



容器承载



健康检查



服务模板编排



自动扩缩容



负载均衡对接



配置管理



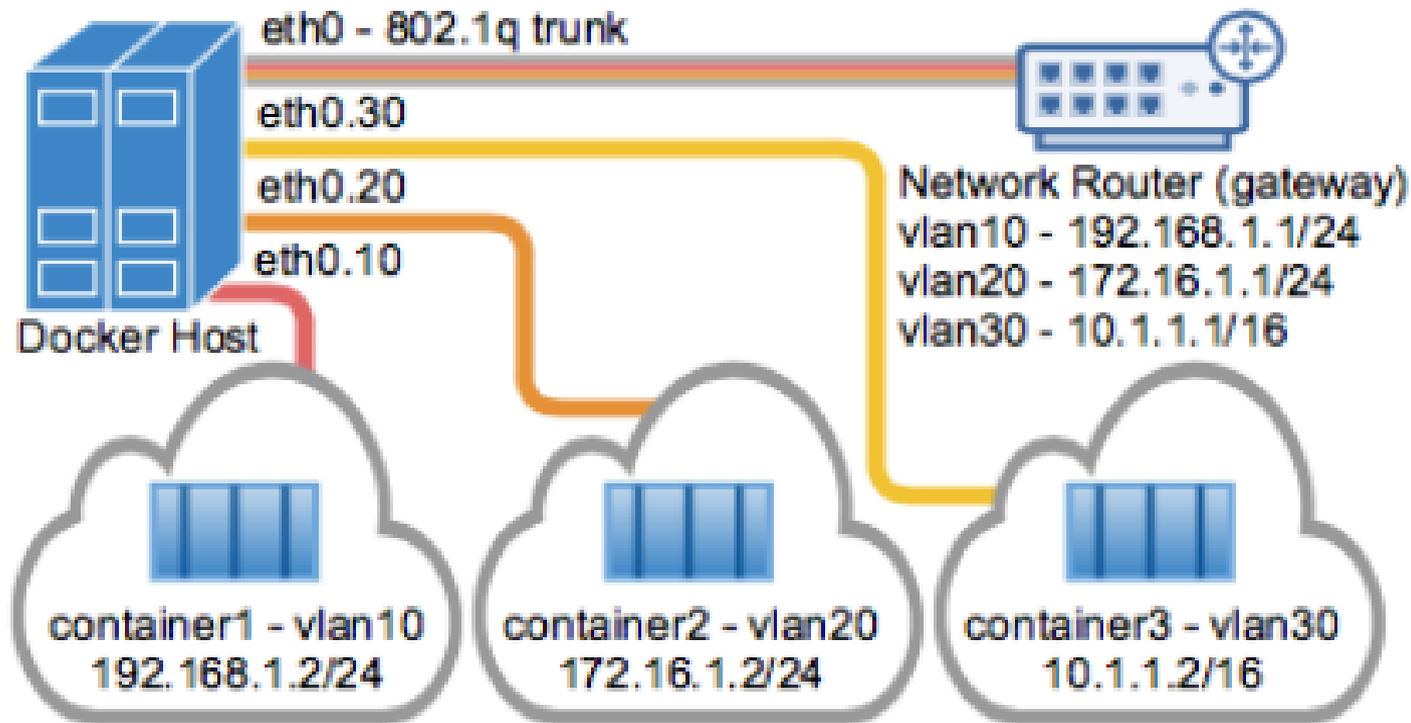
采用微服务架构不要低估细粒度分布式应用治理与运维的复杂度

平台建设的关键点

# 容器云CI平台的落地路径



## 基于MacVLAN的网络架构



## Macvlan的优势

- 1、MacVLAN 网络支持集群中的容器与集群之外的资源互通，比如数据库和各类中间件等
- 2、是underlay 的 MacVLAN 网络能利用底层的 SDN 提供更为稳定和高效的网络服务

# 测试数据的恢复和管理



将容器技术、存储快照技术和数据库技术结合起来，解决单元测试和自动化测试过程中，需要反复恢复数据的痛点。把数据库从“重”变“轻”，从而更好的支撑敏捷的过程。

比如 程序猿写了一个单元测试junit

1、甲客户资金：100元

2、苹果10元1个

甲客户买了1个苹果（一次下单）

Junit测试 程序猿马云往往会写个断言：

```
assertEquals(90, 甲客户的资金, Constant.ZERO);
```

第一次运行单元测试，这个值是正确的，单元测试通过，  
如果第二次提交的，单元测试就无法通过。

现在的情况，我们不得不想其他办法绕开或者克服这个问题

# 我们能不能做到

- 1、每次需要的时候，都能快速拉起一个数据库？
- 2、这个数据库每次被调用的时候，都能恢复到指定时刻的数据？

# 容器共享存储和快照



# 简单维护多个测试环境的版本

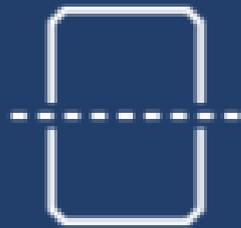
- 存储
- 存储卷
- 快照
- 网络
- 跨云管理
- 资源管理
- 运维中心
- 监控
- 事件
- 日志
- 通知
- 警报
- 应用平台
- 中间件

快照名称(备注)	快照 ID	大小 (GB)	存储卷	状态	创建时间	操作
oracle-20170902	a73f381a	10	oracle	完成	2017-09-29 10:48:44	<a href="#">+</a> <a href="#">-</a>
oracle-20170901	96d92f14	10	oracle	完成	2017-09-29 10:48:27	<a href="#">+</a> <a href="#">-</a>

## 多租户管理



基于角色的权限管理



项目隔离、资源配额

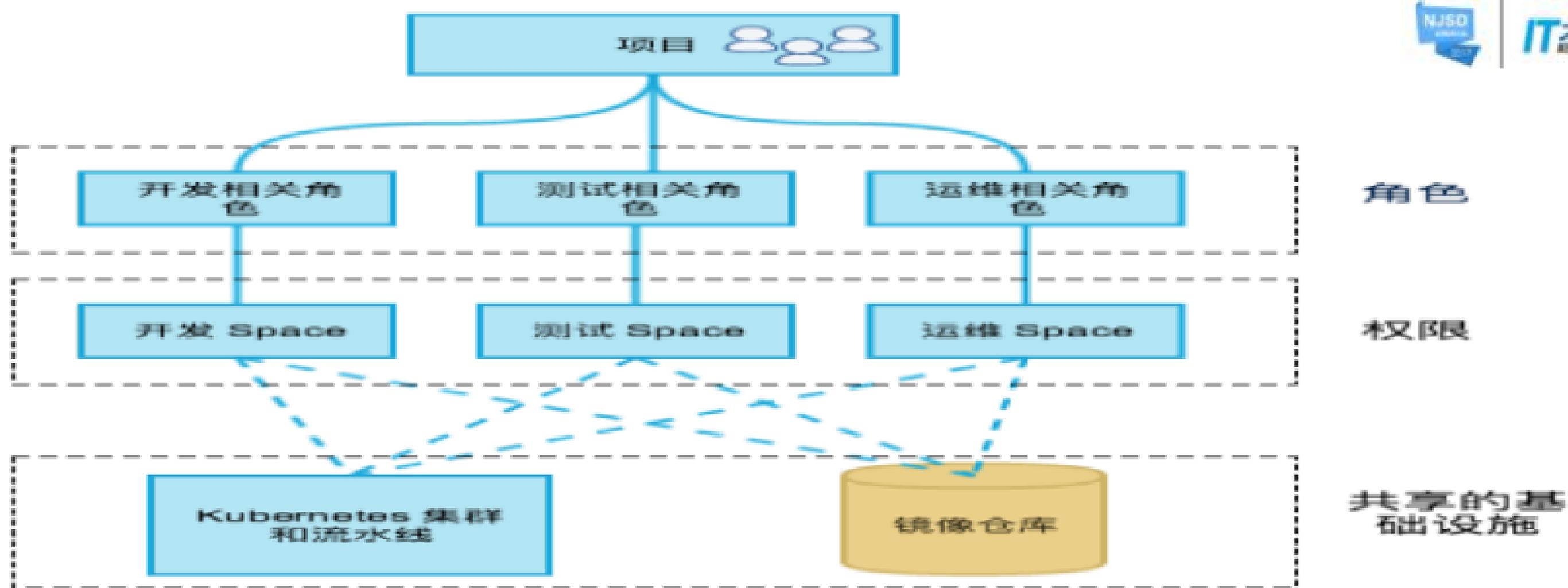


企业内部多租户  
自服务 IT 治理

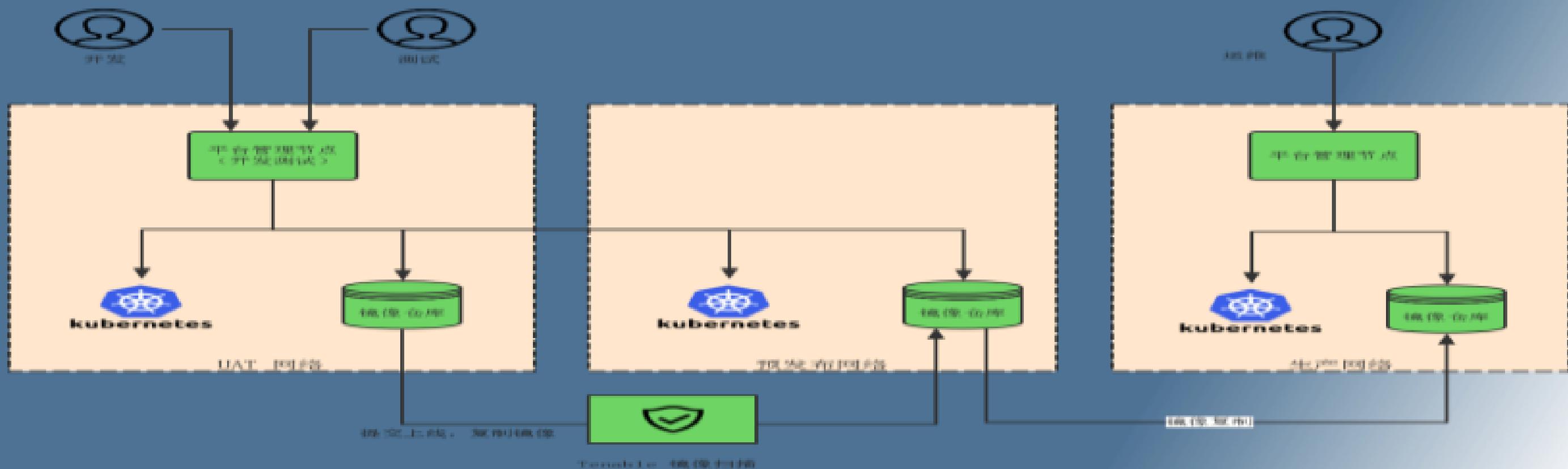


## 多租户

- 有独立的资源的命名空间：资源是包括服务、镜像仓库、流水线，存储资源
- 有独立的用户和角色。每个租户都可以自己定义租户内部的角色，并且有不同的用户
- 每个租户都有自己的资源配额



# 整体架构示意图

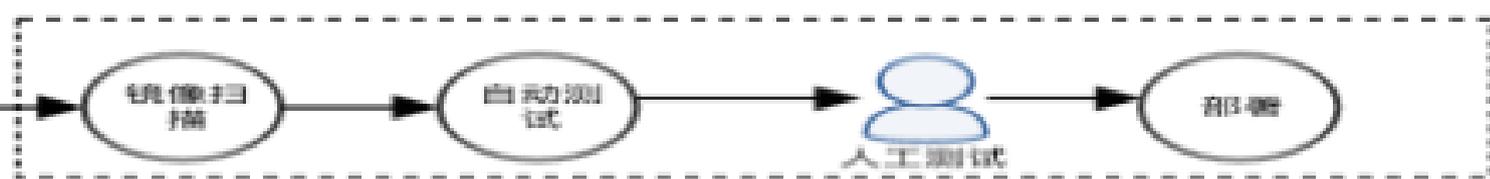
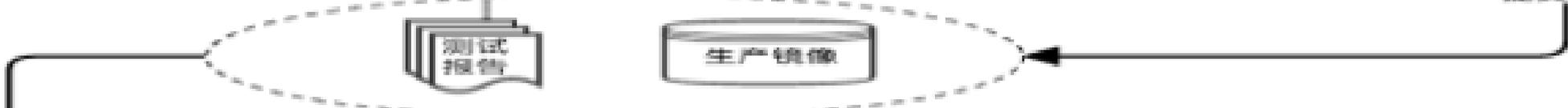




提交测试



提交上线



# 集成相关常用敏捷开发工具



任务可以并发或串行执行

# Junit 集成

配置参数	
测试模式	<ul style="list-style-type: none"> <li>• 单一测试</li> <li>• 批测试</li> </ul>
输出路径	
报告格式	<ul style="list-style-type: none"> <li>• XML</li> <li>• Brief</li> <li>• Plain</li> </ul>
	<ul style="list-style-type: none"> <li>• 目录、文件路径</li> </ul>



Unit Test Results. Designed for use with JUnit and Ant.

Summary

Tests	Failures	Errors	Success rate	Time
3	0	0	100.000%	0.140

Note: All tests are anticipated and checked for with assertions while errors are unanticipated.

Packages

Name	Tests	Errors	Failures	Time(s)	Time Stamp	Host
com.gluon.be	3	0	0	0.140	2010-03-09 15:43:46	192.168.1.10

- 展示单次构建的测试结果
- 展示每个构建项目的测试质量改进趋势

Unit Test Results. Designed for use with JUnit and Ant.

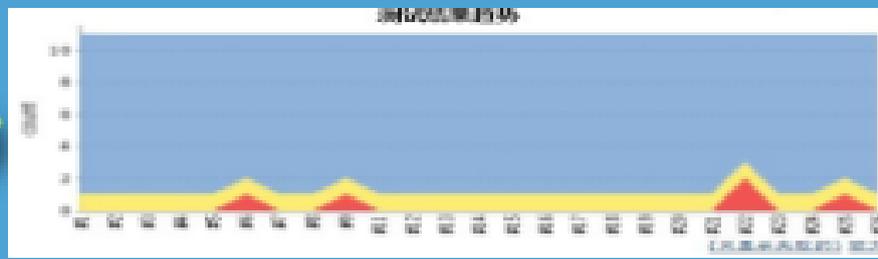
Class: com.gluon.be.ComplexCalculationTest

Name	Tests	Errors	Failures	Time(s)	Time Stamp	Host
ComplexCalculationTest	3	0	0	0.080	2010-03-09 15:43:46	192.168.1.10

Tests

Name	Status	Type	Time(s)
DivisionTest	Success		0.030
MultiplyTest	Failure	expected: <200> but was: <2000>	0.010

Tests: 3, Failures: 0, Assertions: 3, Errors: 0, expected: <200> but was: <2000>  
 at: com.gluon.be.ComplexCalculationTest.multiplyTest() method  
 Source:



# Jacoco 集成

- 展示单次构建的测试覆盖率结果
- 展示每个构建项目的测试覆盖改进趋势

**配置参数**

覆盖率检查规则

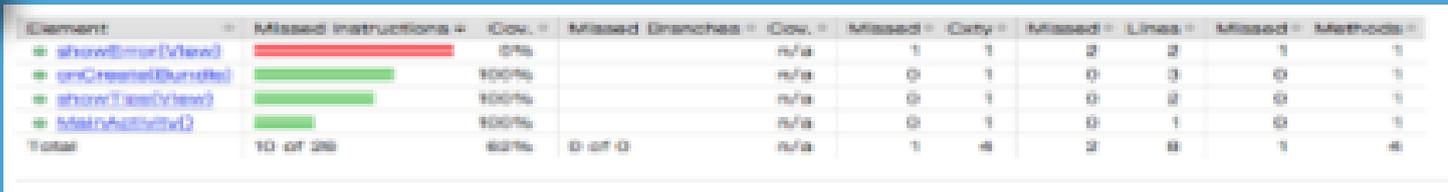
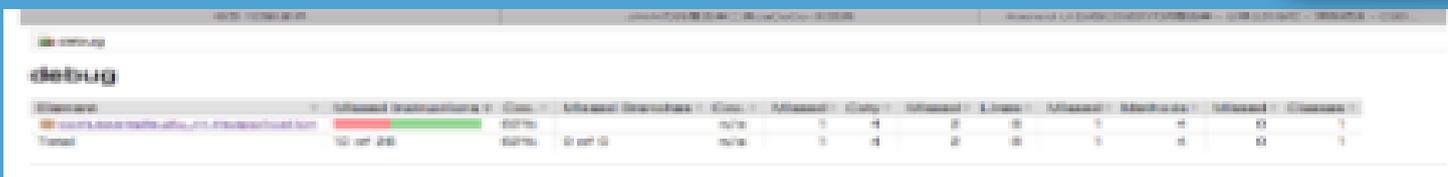
报告格式

- XML
- HTML
- CSV

输出报告路径

- 类文件路径等

Instruction	% Branch	% Complexity	% Line	% Method	% Class
0	0	0	0	0	0
0	0	0	0	0	0

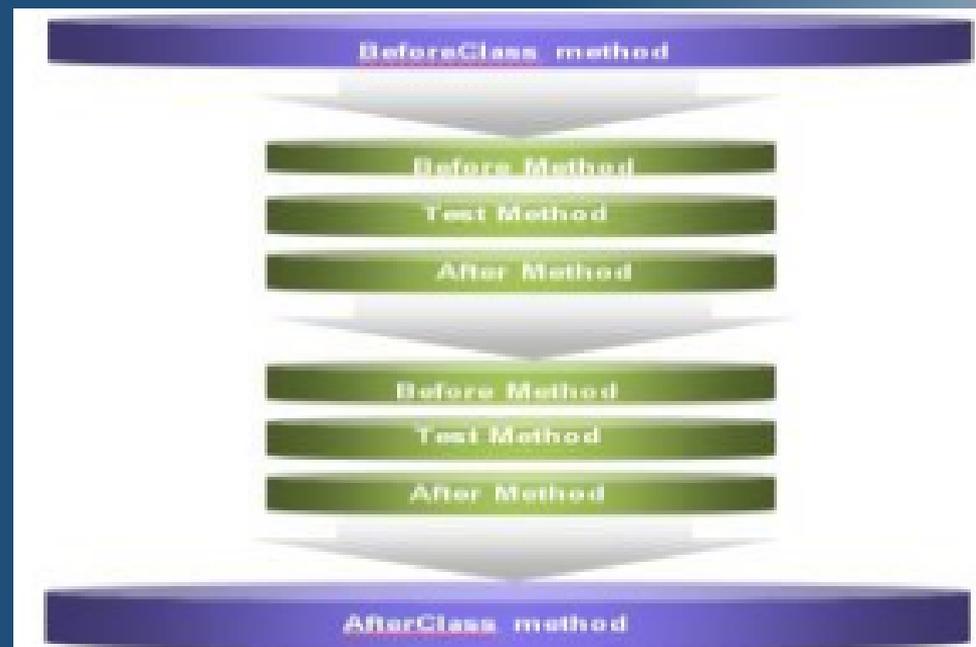


覆盖率

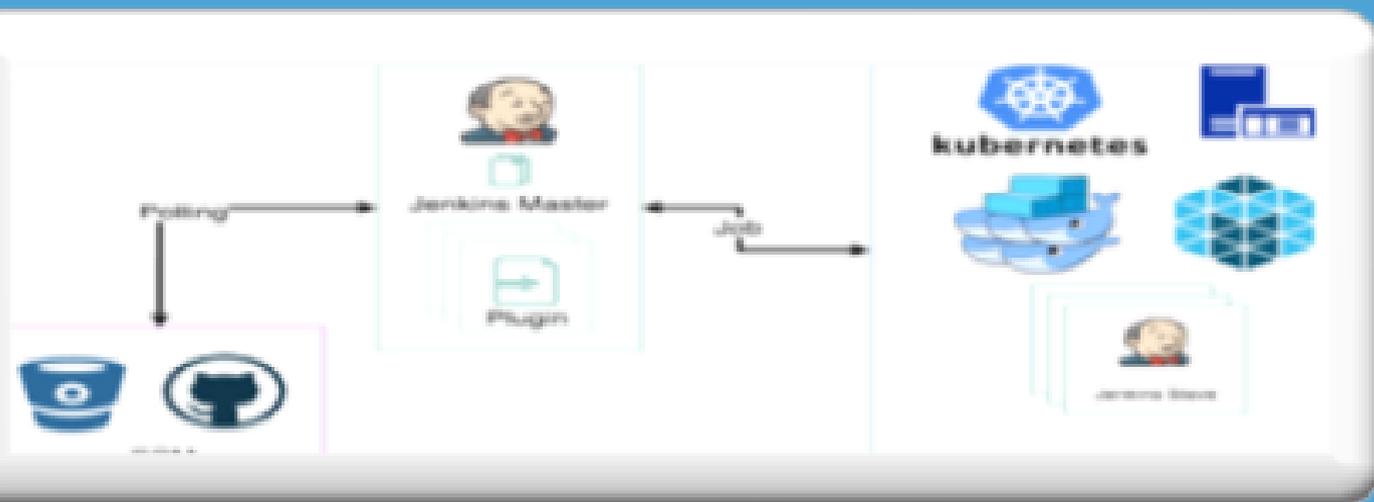
100%



- 通过集成中心整合主流单元测试和覆盖率分析利器
- 便捷设置报告格式、质量阈值
- 图表展示分析报告



## 深度集成Jenkins，合力打造功能完备的CI/CD



- 通过集成中心可部署多个Jenkins实例
- 丰富的Pipeline设计功能，提供场景模板和扩充

# 虚拟机和容器的故事

■ 谢谢

