



2016中国开源年会

China Open Source Conference 2016



开源软件在华为容器 项目测试中的应用

-利用开源软件构建devops工程能力

孙远 华为中央软件研究院



我的经历

美国风河系统公司

- 负责linux build system、analysis tools、workbench测试工作。

华为中央软件研究院

- 带领团队进行容器OS、docker测试工作，并参与项目过程改进。
- 参与开源社区：补充ltp社区user namespace特性测试用例和docker社区中的测试用例。
- 完成《docker进阶与实战》测试章节的编写工作。



内容



- 传统软件开发流程的痛点
- Devops、敏捷开发、容器技术
- 选取开源软件的标准
- Devops与容器技术所涉及的开源软件
- Devops及容器云产业状况
- Devops部署
- 性能测试、安全测试、长稳测试
- 基于微服务的软件产品供应链
- 开源软件问题求助方式与注意事项
- 企业参与开源社区的策略

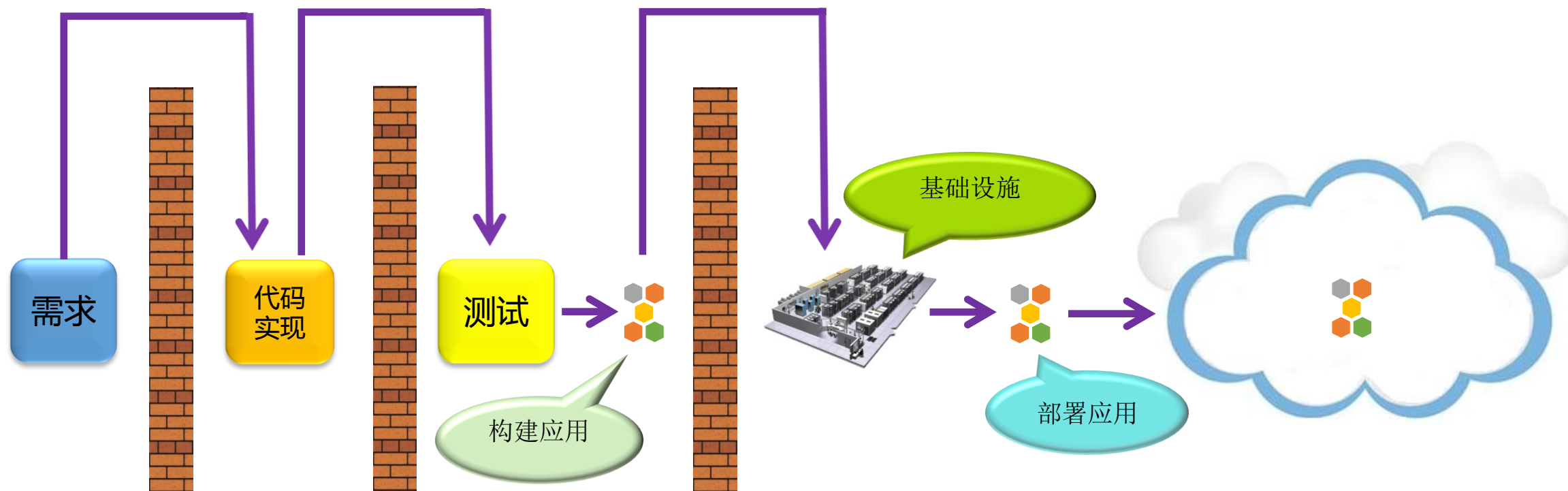


传统软件开发的痛点

- 开发、测试、运维环境不统一；无法准确获取客户的软件环境；可移植性差，不同环境要做非常多的开发和测试。
- 开发在提交代码前未做充分的测试。
- 开发无法复现测试报出的bug，开发与测试之间相互推诿。
- 虚拟机开销大，利用率不高，配置测试环境的时间较长，测试自动化成本高。
- 服务伸缩速度慢，无法应对业务的瞬时峰值。



传统软件开发流程



各团队的痛点



- 开发团队：

我们把代码自验做得那么完善还需要测试干什么？运维团队用的是什么系统？在我们的系统中软件明明是运行正常的，到了运维团队那么怎么就不行了。

- 测试团队：

开发输出的环境配置文档又跳步骤了，而且他们提供的版本无法安装。

- 运维团队：

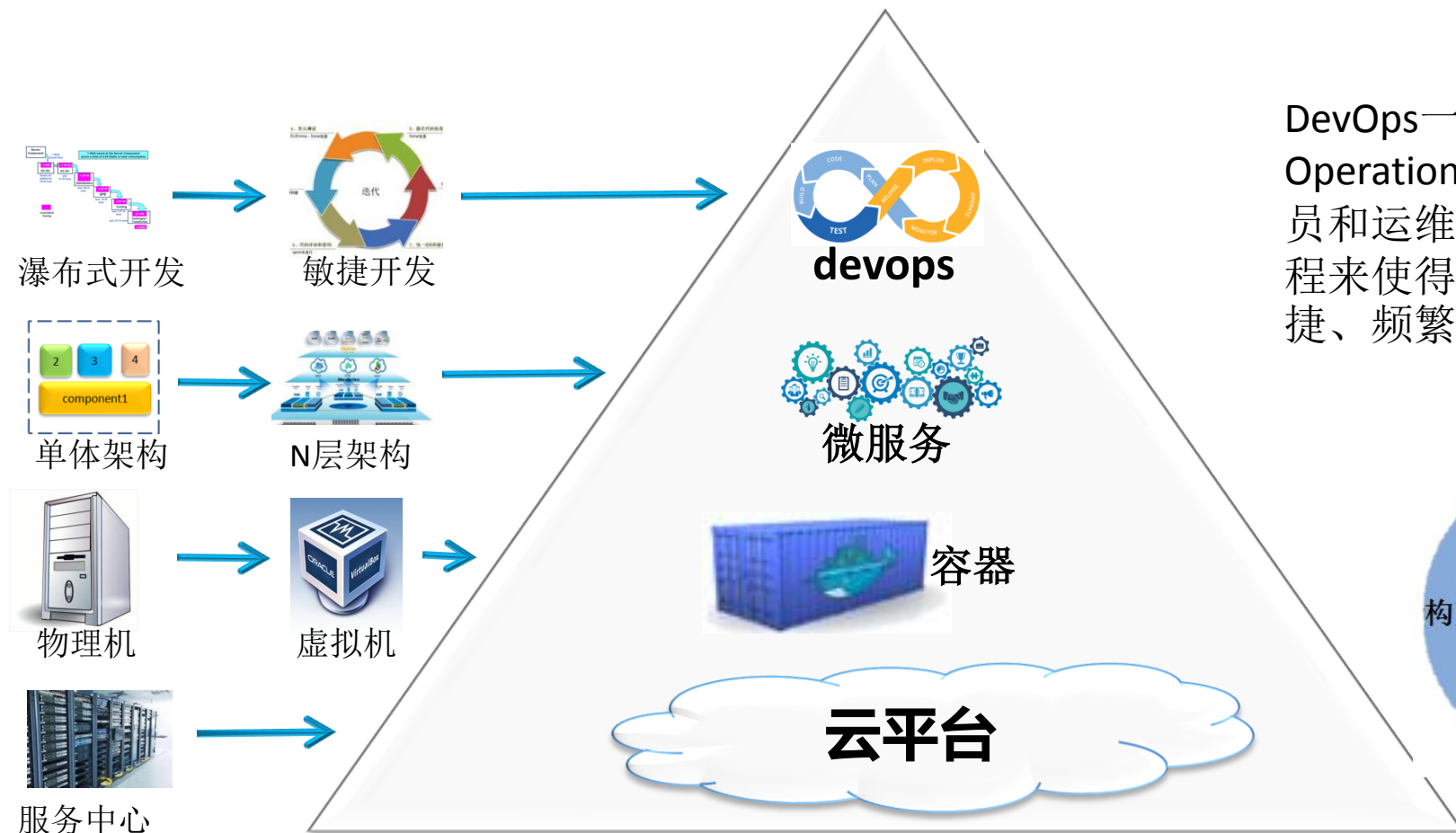
提供给我们的软件没跑多久就挂了，上游提供的软件质量太差了，这已经不是第一次了。



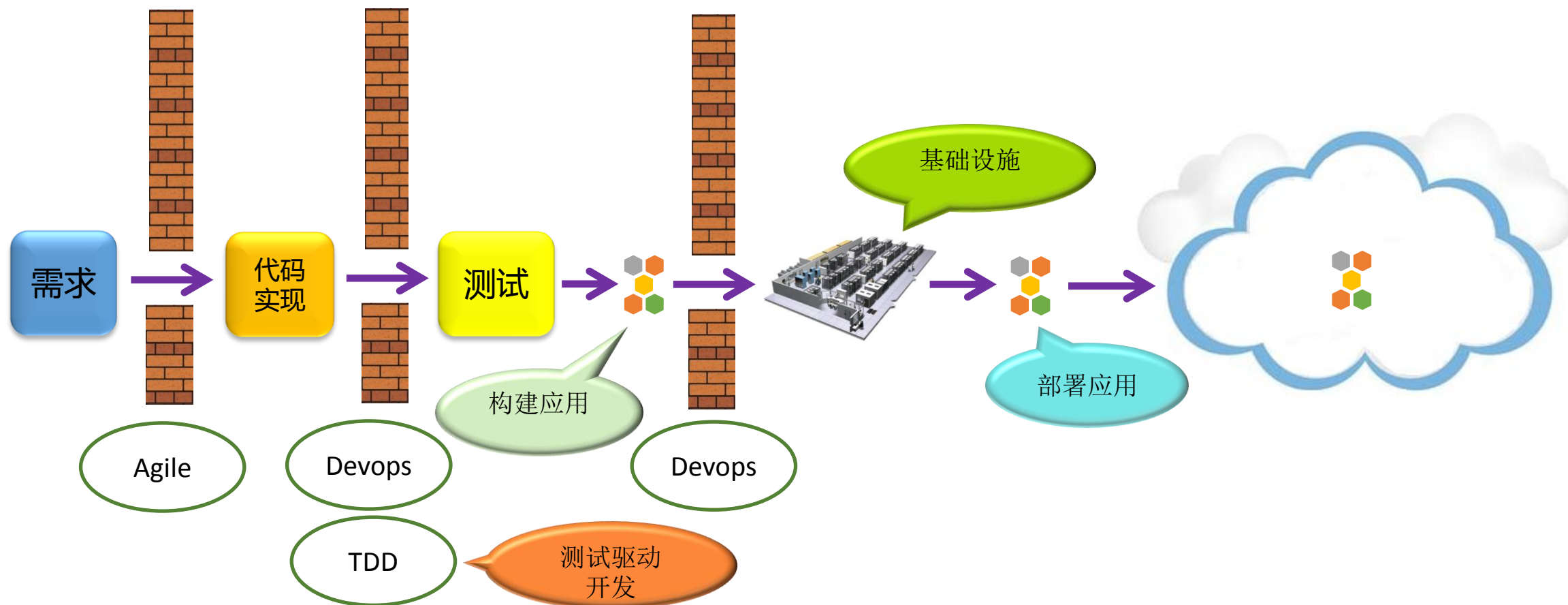
Devops简介



DevOps一词的来自于Development和Operations的组合，突出重视软件开发人员和运维人员的沟通合作，通过自动化流程来使得软件构建、测试、发布更加快捷、频繁和可靠。



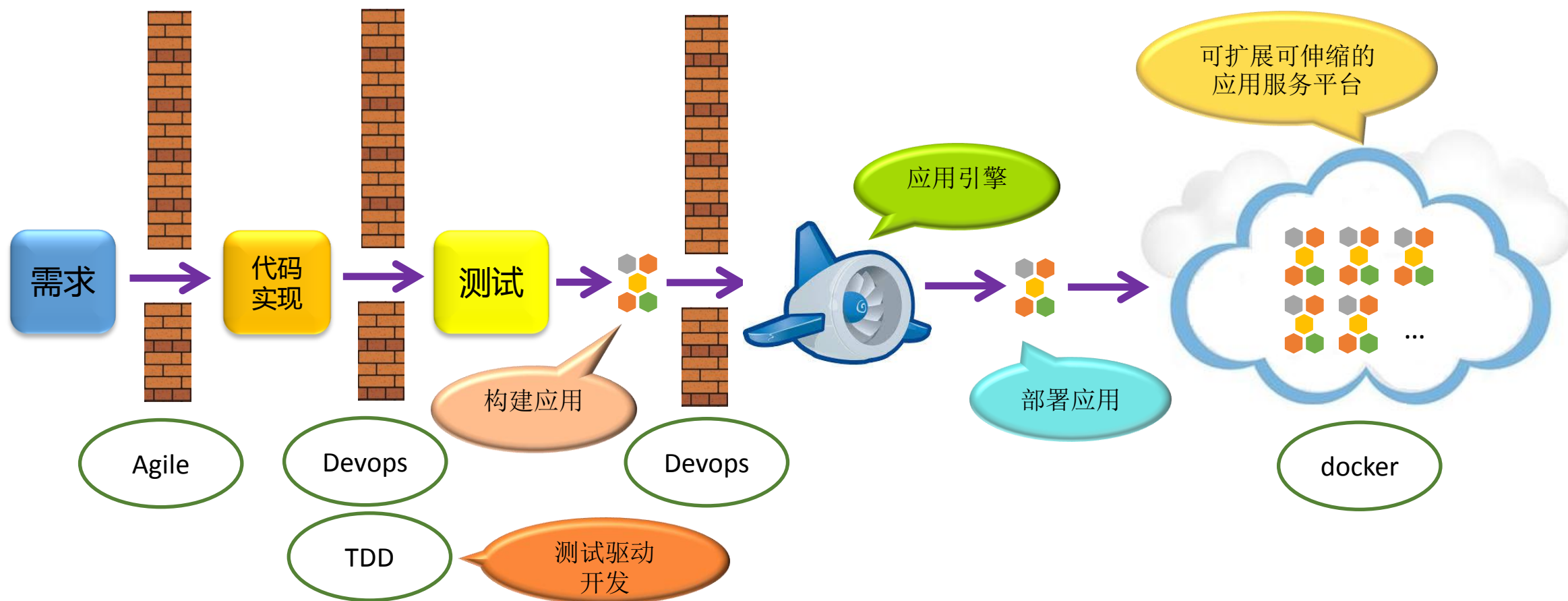
Devops与敏捷开发



当用户访问量激增时



Devops、敏捷开发与容器技术



选取开源软件的标准



标准	描述
开源协议	谨慎使用 采用GPL 协议的开源软件。
软件是否满足业务（包括功能、系统架构、稳定性、扩展性、性能、兼容性、安全性）	略
contributors、commits、Pull Requests、Issues、Fork数值	更多的数值代表着有更多的社区活跃度，有利于社区向良性发展。
完善的文档	丰富的文档、wiki页面、指导书将更好的引导我们理解社区所涉及的项目细节和使用方法。
项目开源贡献者的分布情况	有多个公司参与的开源项目更加可靠，避免项目被某一个公司绑架。
市场占用率及认可程度、成熟度	成熟度高的开源项目软件缺陷少、认可度高，便于商业推广。
项目是否加入开源基金会	加入开源基金会的项目不会被某一团体绑架或者被竞争对手收购。
开源社区接纳新特性的能力	社区接纳贡献者提交新特性的可能性越高，越有利于引导社区向适合公司战略的方向发展。
对于提出问题或软件缺陷响应的速度	响应速度越快，越有利于用户问题得到解决。
是否有成熟的测试用例和验证方案	这有助于更好的保证软件质量。
社区是否有大公司支持，maintainer、committer、contributor区分是否清晰。	有大公司商业支持的项目，服务更加优质，同时也具备进一步的商业合作。

Devops与容器技术所涉及的开源软件



类别	开源软件
安全容器	hyper,clear container,photon...
镜像存储	docker-registry...
容器引擎	docker, rkt...
安全	selinux, seccomp, notary...
网络	ipsec,haproxy...
存储	ceph,swift...
数据库	mysql,mongoDB...
编排工具	mesos,kubernetes,swarm,compose,marathon...
虚拟化平台及工具	docker machine,virtualbox, openstack...
容器操作系统	rancheros,coreos,vmware photon,snappy ubuntu core,redhat atomic host...
开放容器项目标准	runc,image-spec,image-tools,runtime-spec,runtime-tools...
支持容器运行的操作系统特性	kernel

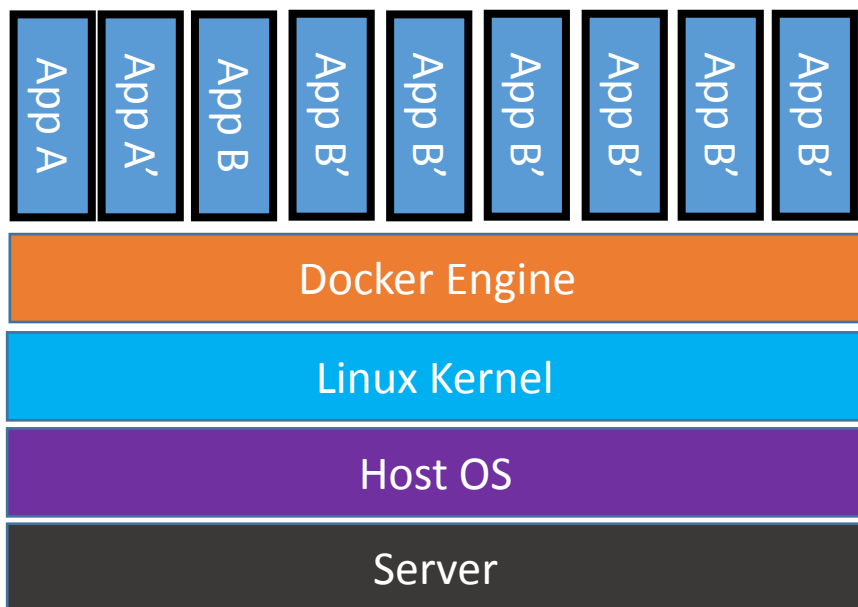
Devops及容器云产业状况



分类	描述	现状
容器商用云平台	亚马逊云、docker云、XX云...	大小企业并存，同质化严重，且竞争激烈，属于群雄逐鹿阶段。在不远的未来，如果企业缺乏资金、核心技术和对生态准确的把握，将被市场淘汰。市场将最终形成寡头。
其它外围工具	安全、存储、监控等	中小企业的机会点
编排工具	mesos,kubernetes,swarm	有实力的公司在角逐，三分天下的局面
容器引擎	docker, rkt...	寡头，技术门槛相对不高，但需要完善生态
容器OS	rancheros,coreos,vmware photon,snappy ubuntu core,redhat atomic host...	寡头，有实力的公司参与
OCI标准项目	runc,image-spec,runtime-spec	标准制定的战场，大公司博弈的战场
底层内核	kernel	技术门槛高，大公司博弈的战场



Docker架构



支撑docker的核心技术

- Cgroups

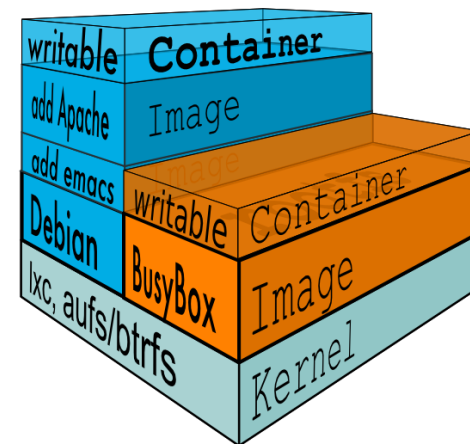
资源限制

- Namespace

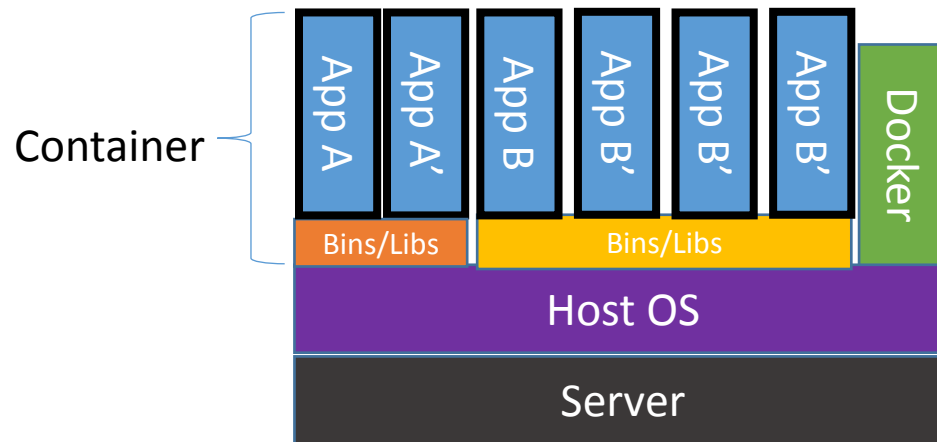
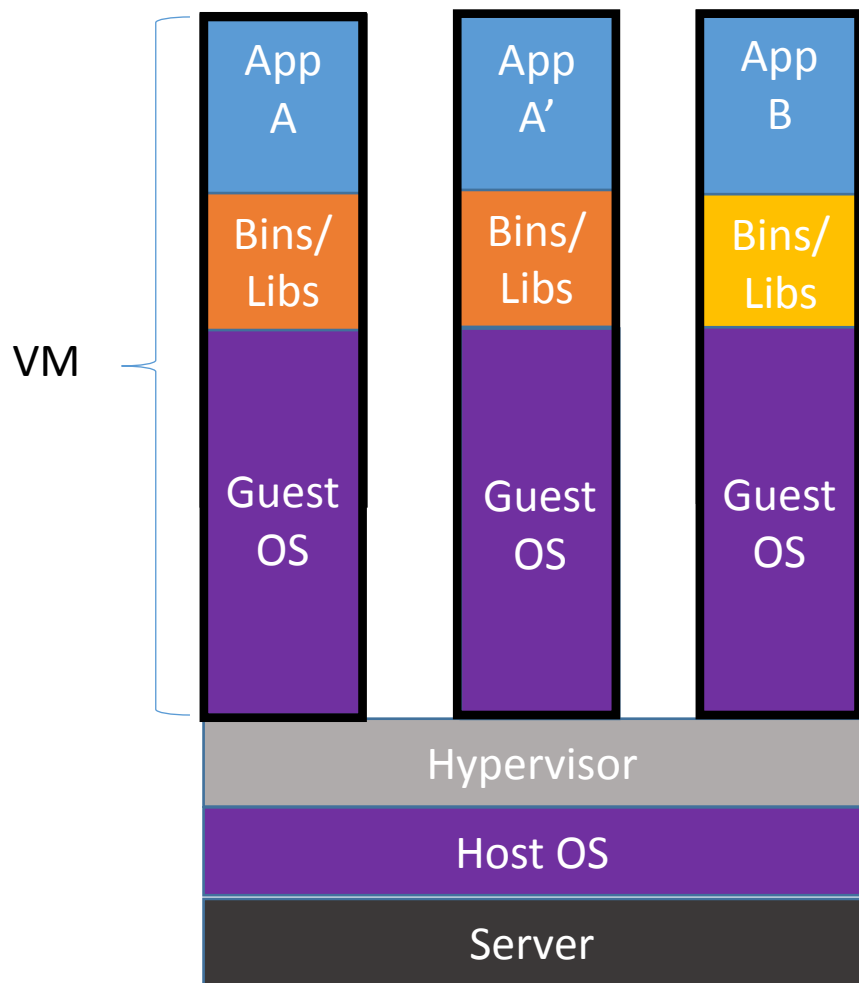
资源隔离

- Union File System

Docker镜像的基石、分层存储



VM vs Docker



指标项:

- 启动时间
- 部署时间
- 系统资源占用
- 环境分享
- 环境释放时间

- 模拟不同arch环境
- 扩展性
- 应用可移植性
- 系统可靠性
- 系统安全性

Goal of Docker



Goal: Build, Ship, and Run Any App, Anywhere.

Build

Compose your application from microservices, without worrying about inconsistencies between development and production environment.

Ship

Store and distribute your docker images in registry services.

Run

Deploy scalable services, securely and reliably on a wide variety of platforms.

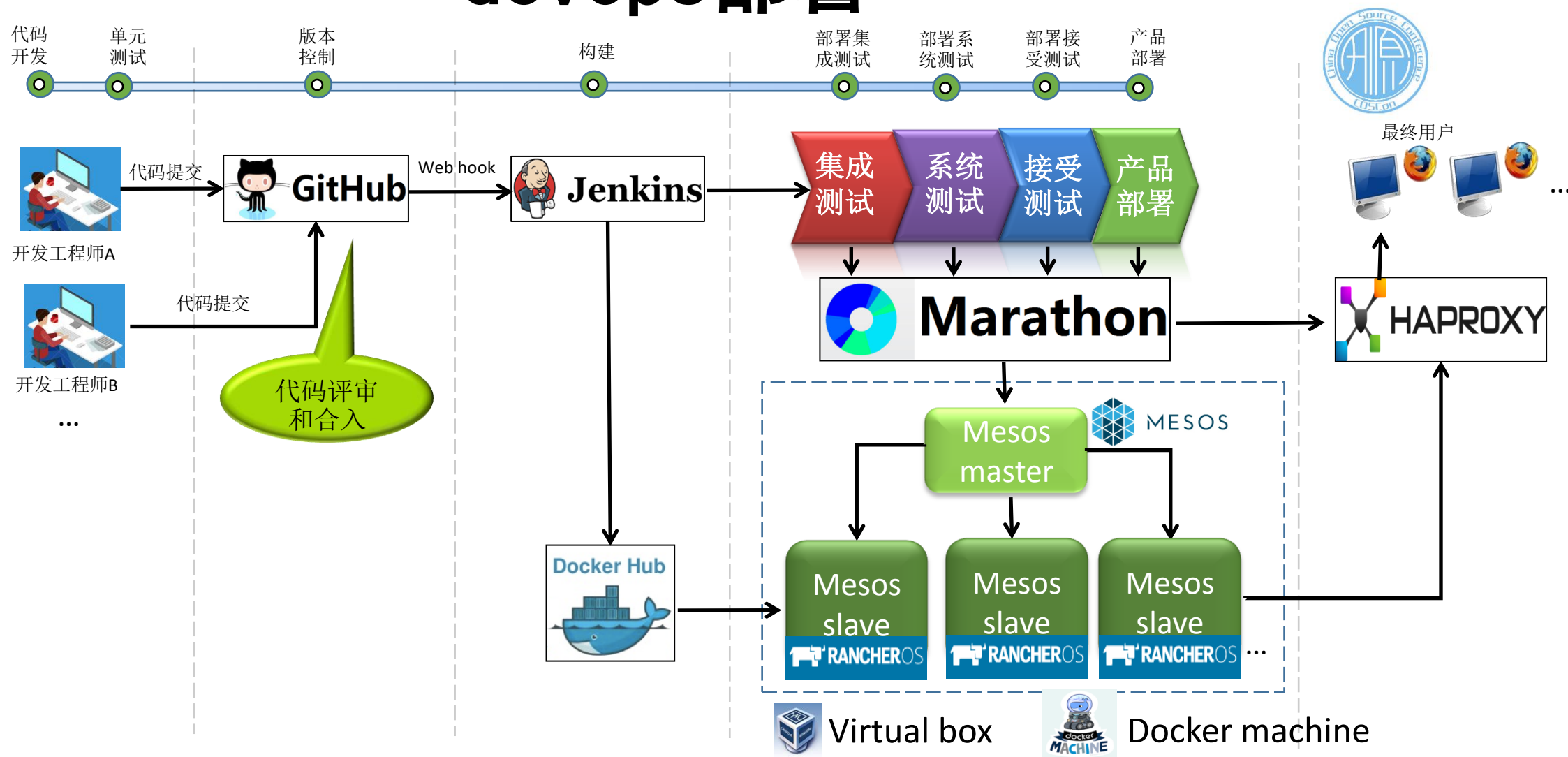
Any App



Anywhere



devops部署

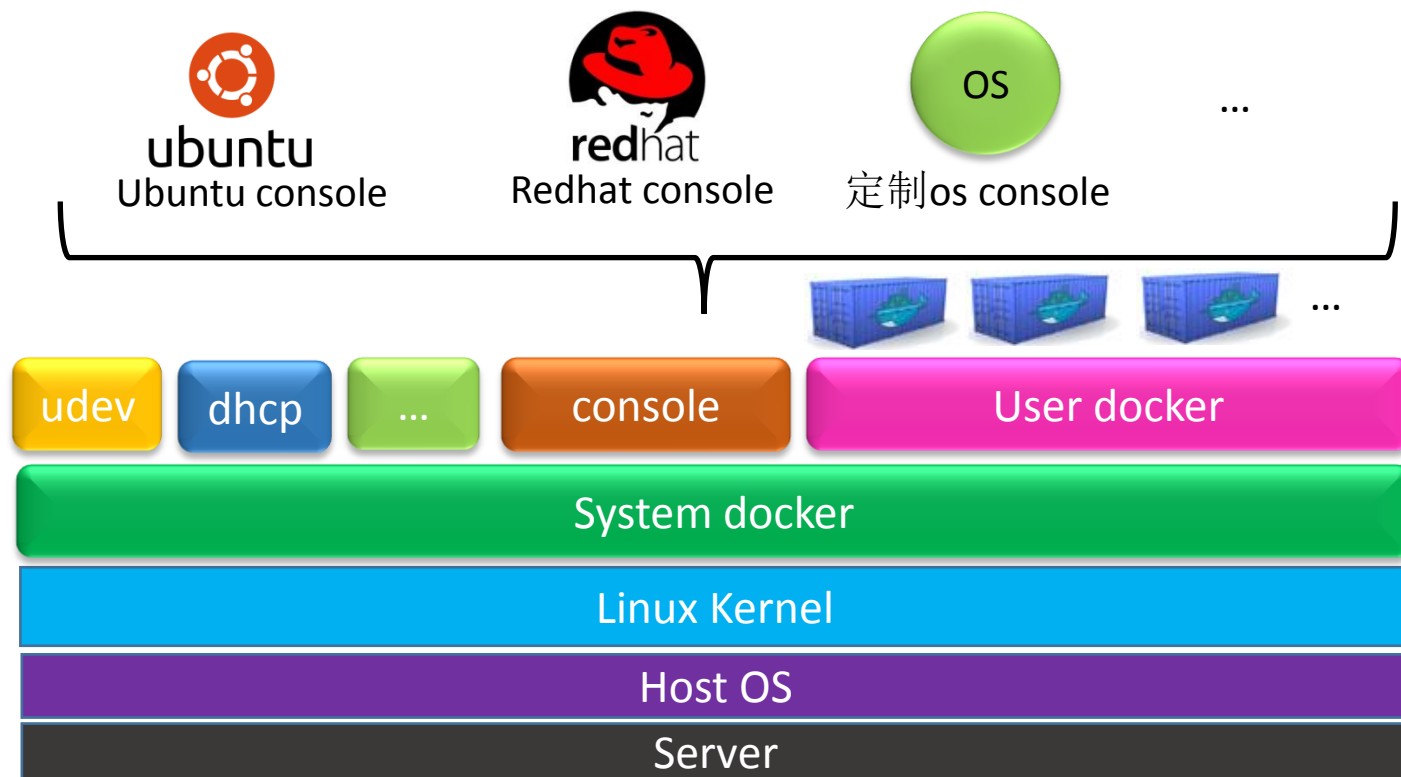


Devops部署所使用的开源软件



软件名称	描述
Github	代码仓库
Jenkins	自动化任务调度CI工具
Docker registry	镜像存储仓库
Marathon	容器编排调度工具
Mesos	容器编排调度工具
Rancheros	轻量级容器OS
Haproxy	负载均衡器
Zabbix	OS监控工具
Docker machine	部署OS工具
Virtualbox	虚拟机平台

Rancheros



File system and tools	8.4MB
User docker (upstream docker)	13.7MB
System docker	4MB
Kernel drivers	11.7MB
Kernel core	4.2MB

- 资源占用少
- 便于系统升级回滚
- 可以自主定制和更换os console

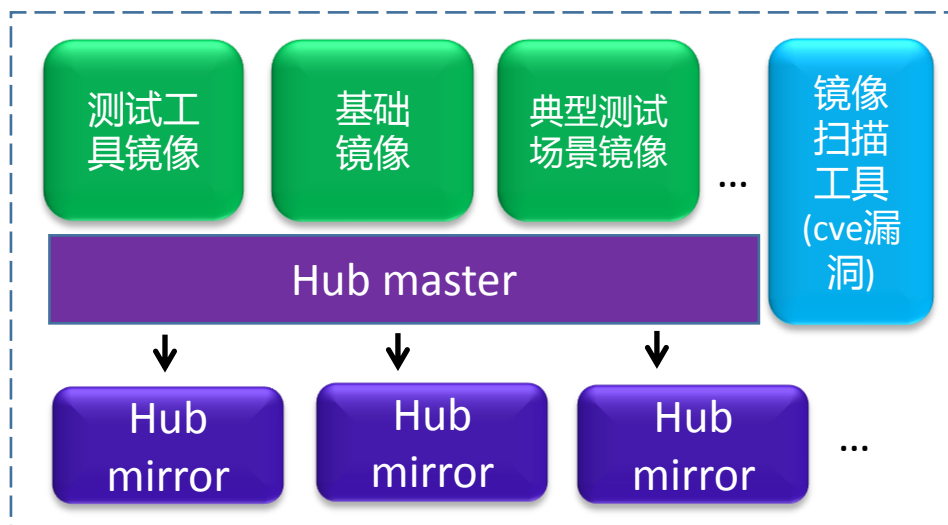
不足的方面需要进行改造

- 代理能力不足
- 启动需要网络支持，无网络情况下无法启动
- 缺少安全加固

改造

- 适配到公司内部网络
- 将默认console镜像内置，使其启动不依赖于网络
- 添加安全加固特性

Docker registry



不足的方面需要进行改造

- 缺少多租户鉴权
- 后端存储驱动不全
- 缺少安全加固
- 无法支持镜像站点

改造

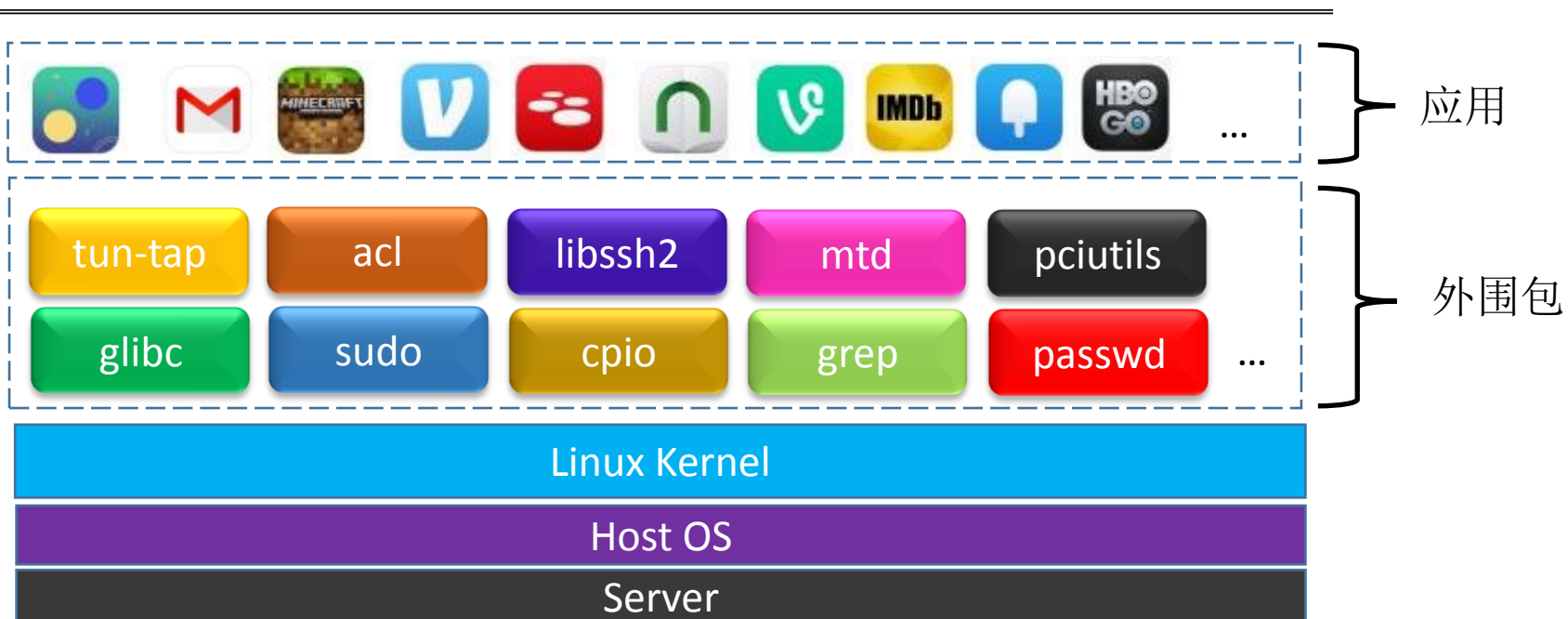
- 添加多租户鉴权
- 补充后端存储
- 添加镜像扫描特性，确保镜像安全。
- 添加镜像站点

Linux外围包测试



测试场景

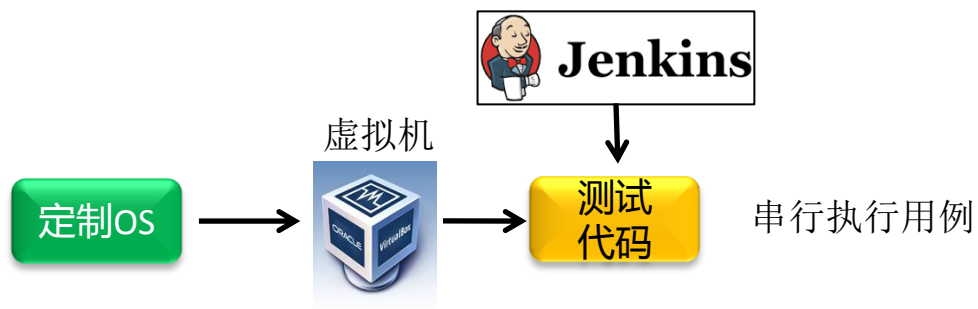
- 在OS启动后执行外围包功能测试(runtime)
- 需要确认外围包功能和依赖
- 覆盖不同类型的Linux OS



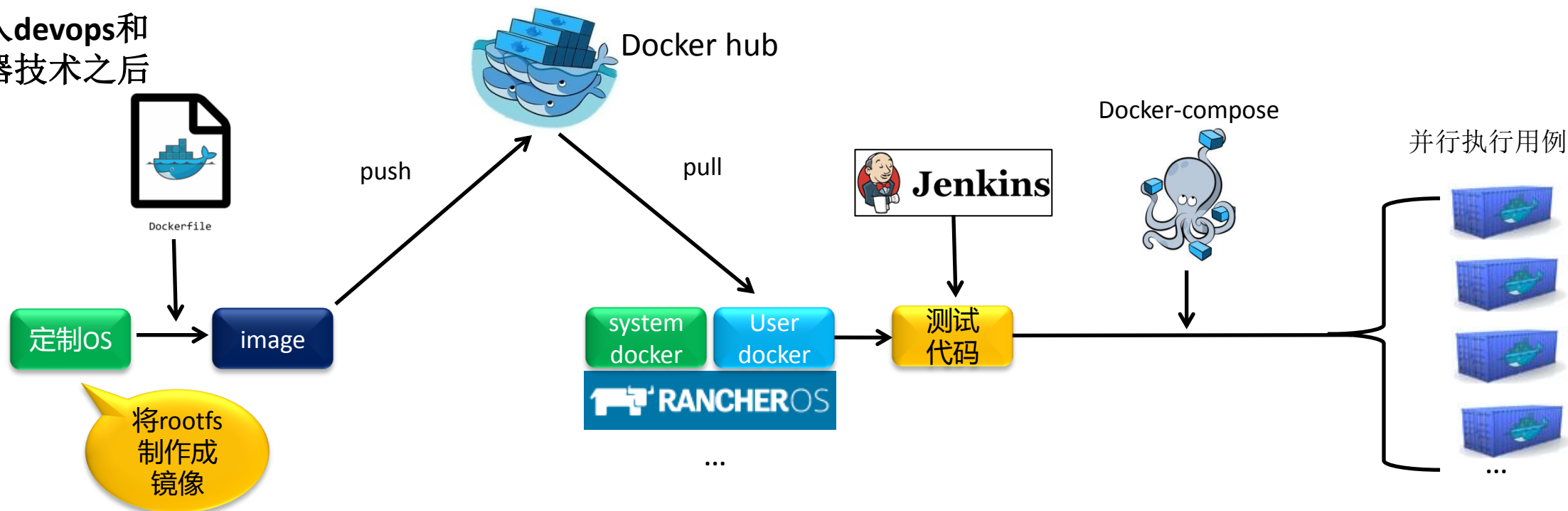
Linux外围包测试



引入devops和
容器技术之前



引入devops和
容器技术之后

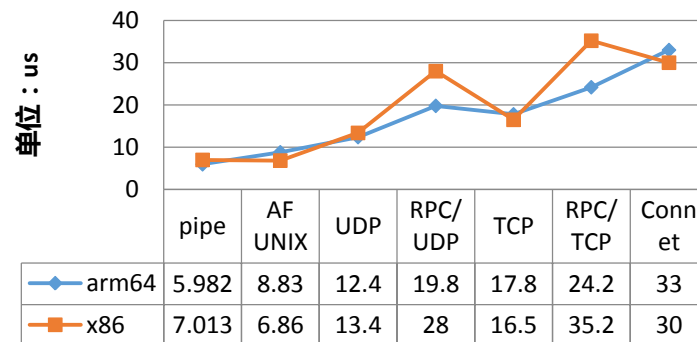
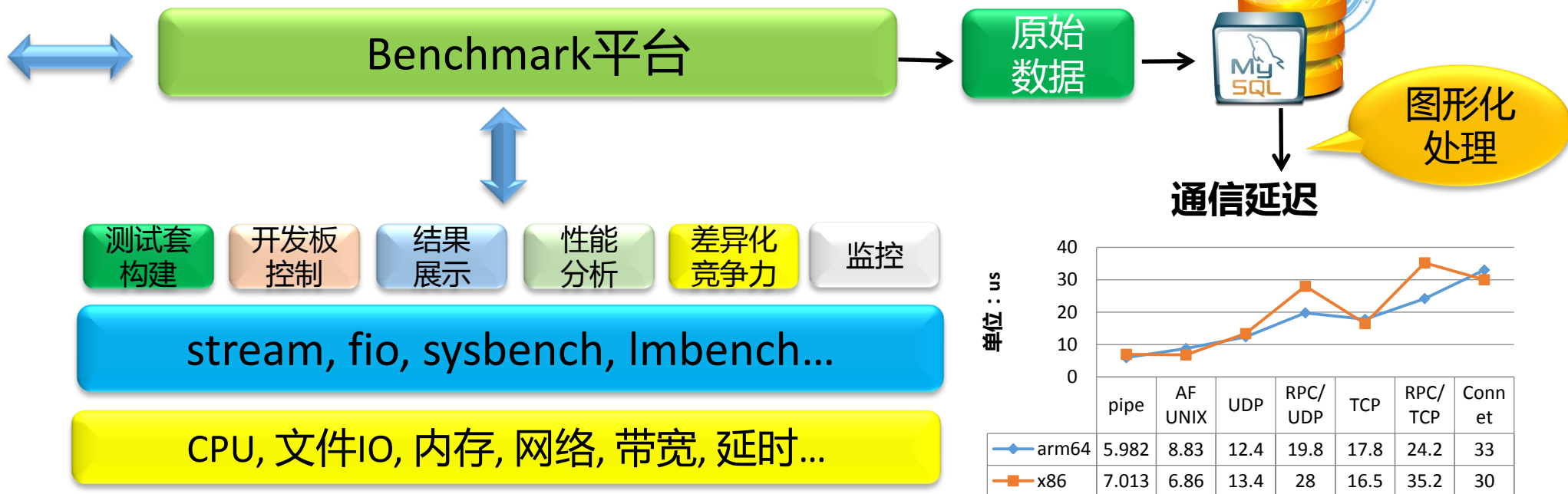


Linux外围包测试



引入devops和容器技术之前	引入devops和容器技术之后
测试串行执行，不易并发。	测试并行执行，提高了cpu利用率，可以更快的完成测试。
无法解决多个外围包的环境污染问题。	容器可快速启动与关闭，每次都是清洁的环境。
外围包编译环境不易统一，导致开发和测试环境不统一。	通过镜像保存编译环境，确保环境统一。
环境释放时清理工作依赖于程序员的技能。	环境释放时清理工作由docker接管，程序员不必写cleanup函数。
测试网络包时需要至少两台主机。	测试网络包时只需要在一台主机中启动两个容器。
开发自验不一定充分，导致低级问题遗留给测试。	提交代码时第一时间触发测试，确保低级问题不遗留给测试。

性能测试与安全测试



安全测试任务:

- 安全威胁分析
- 安全合规测试
- 安全扫描工具

开源安全扫描工具:

- Nmap: 端口扫描
- Nessus: 扫描系统cve漏洞
- ...

压力稳定性测试



压力稳定性测试容器化

- CPU负载

- 按照加压模型，对cpu占有率按照30%~80%

- 内存负载

- 按照加压模型，对内存占有率按照50%~80%进行规划

- IO频率

- 使用fio工具

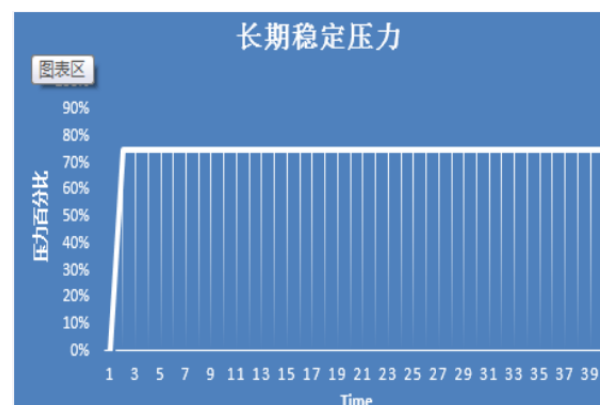
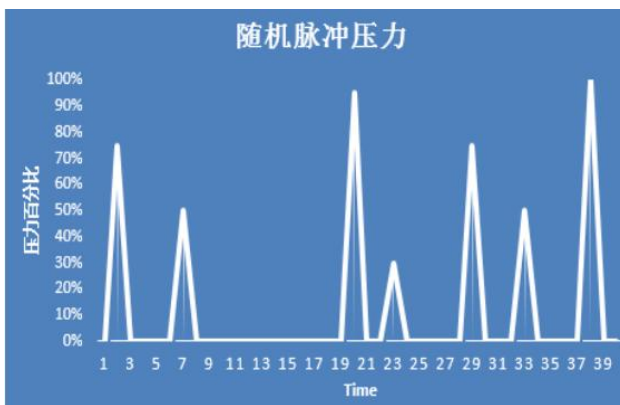
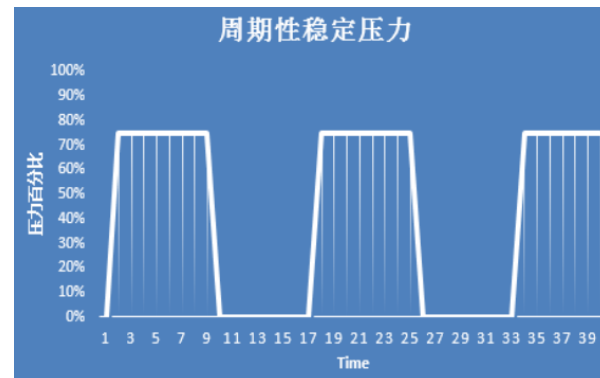
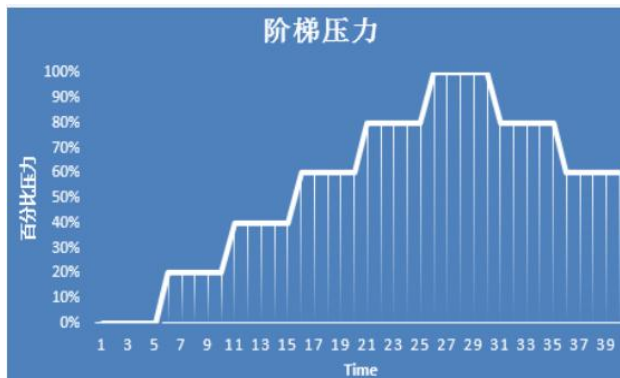


虚拟机

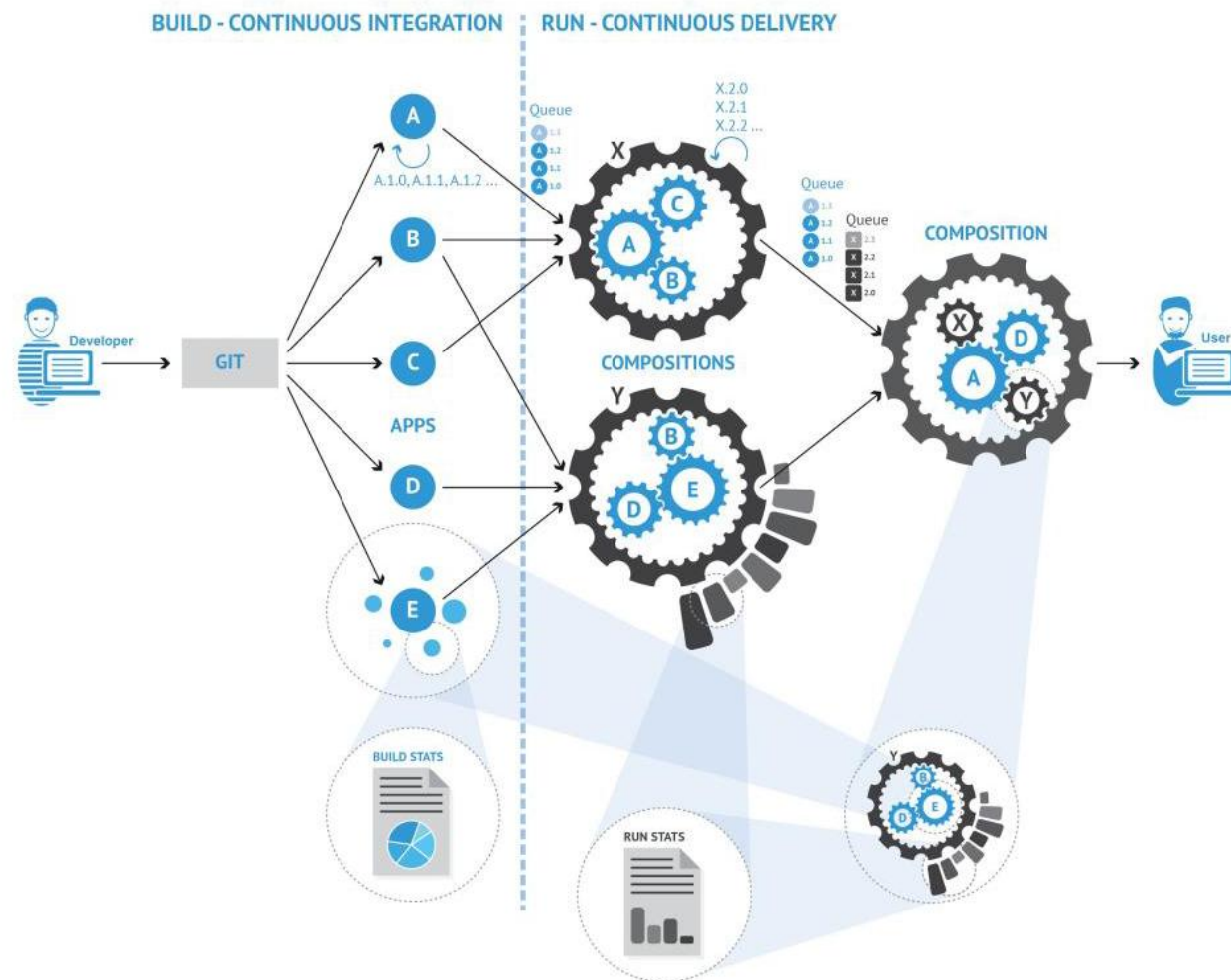


容器

确保测试运行的更加平稳、持续；将测试工具内存泄露等问题带来的负面影响降到最低。



基于微服务的软件产品供应链



下一步计划



- 完善项目安全加固策略
- 创建用户友好的操作界面
- 制定一个基于容器技术的可以改善全流程测试的方案，包括硬件驱动测试、内核测试、中间层测试、应用测试。
- 将测试场景、测试工具、安全工具进行容器化。

开源软件问题求助方式与注意事项



- 通过互联网搜索引擎进行查询。
- 通过专业技术网站或论坛求助。
- 通过社区中issue板块或者发送邮件到社区邮件列表进行提问。
 - 如果是报bug，需要满足社区提交bug的规范。
 - 处理好与maintainer的管理，取得其信任。如果不是原则问题，避免与maintainer进行过分争论。
 - 与maintainer有不同意见时不要挑战其权威，需要用委婉的方式解决分歧。
 - No way. I disagree with you.
 - In general, I agree with you. However, in this case ... Maybe...
 - 避免在社区中泄露公司的项目细节。
 - 不要向社区直接提出低级问题，如配置git的方式。没有把握时可以先在公司内部评审，避免低级问题对公司外部形象造成影响。
- 当问题得不到解决时将问题提交给更多的maintainer或社区贡献者。
- 通过社区irc会议进行求助。
- 在外部会议上与maintainer或者专业人士进行讨论。
- 与开展开源项目的公司展开商业合作。

企业参与开源社区的策略



企业参与开源社区的目的：构建差异化竞争力，实现商业利益最大化。开源不是情怀，不是为社会做贡献，而是实实在在的竞争手段，是竞争白热化以后最常用的竞争手段。

开源不是“把代码贡献出去”的一锤子买卖，每行送出去的代码，都是为了收益。没有背后战略的支撑的开源，不是开源，是开玩笑。

●核心社区

- 参与制定标准，增大社区话语权，引导社区向有利于公司的方向发展，制定有利于公司的规范。
- 做成基金会项目，如OCI项目。
- 防止社区被竞争对手控制，或被某一公司绑架，如容器引擎项目。

●重要社区

- 与外部合作参与社区。
- 独自参与社区，深入了解趋势，为商业化做准备。

●非重要社区

- 跟踪社区发展，如性能测试套。

●边缘社区

- 已用为主，不投入，如jenkins。



引用资料

- 《The Humble Programmer》 Edsger W. Dijkstra
- 《why-docker-website》
- 《docker进阶与实战》
- 《Software Testing》 (2nd Edition) Ron Patton
- 《Testing Cloud Services: How to Test SaaS, PaaS & IaaS (Rocky Nook Computing)》 (1st Edition) Kees Blokland, Jeroen Mengerink, Martin Pol
- 《Using Docker/Software Containers for Automated Testing》 Pini Reznik
- 《DevOps的前世今生》 木环



提问环节



联系我：

孙远 华为中央软件研究院

研究方向：容器技术、docker、linux内核、软件测试、自动化测试

Email: sunyuan3@huawei.com,
yuan.sun82@gmail.com