



云原生在企业内的技术栈治理

联想

xCloud云原生

联想集团：全球供应链 卓越运营



180+
市场



10亿+
全球用户

1.5亿+
智能设备
年出货量



35
全球工厂



#9
Gartner 2022
供应链25强

¥ 12亿+
数字化转型
投资



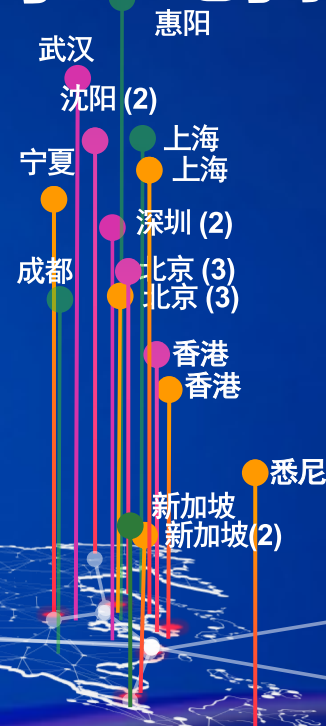
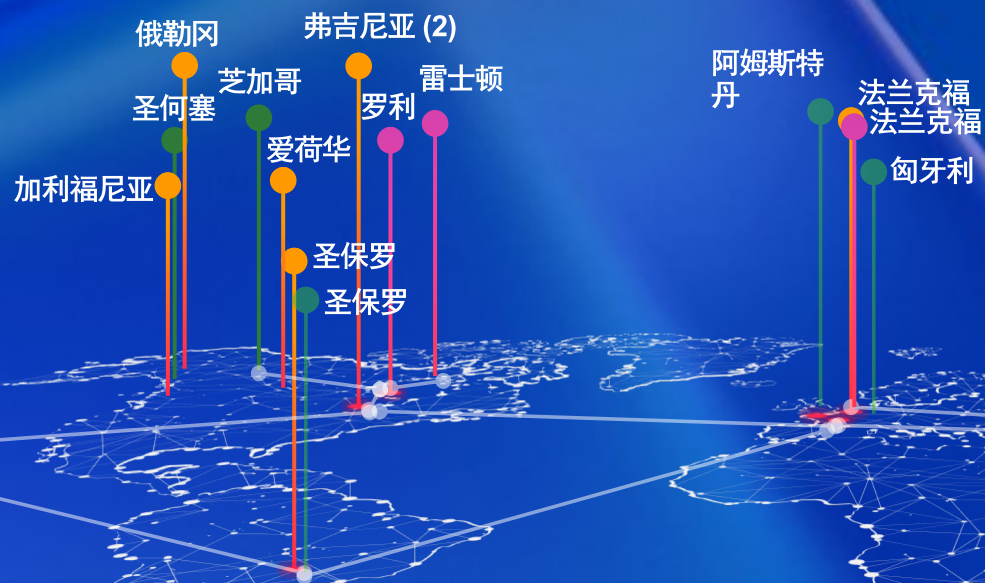
5百万+
订单行/年

2000+
供应商

xCloud云原生团队支撑联想全球IT创新布局



■ 私有云数据中心 ■ 公有云数据中心 ■ 边缘数据中心



中国，日本，美国
三大研发中心

82,000+
全球员工

2700+
应用

280+
全球办公室

12
联想全球工厂



16
公有云专区

21
私有云数据中心及
边缘数据中心分布
在全球17个城市

联想云原生实践全面支持 业务创新

- 统一纳管全球16个公有云专区、21个私有云及边缘数据中心，覆盖联想全球供应链、产研、营销等业务创新；
- 统一内外部技术栈，实现安全IT治理、高效运维，近30人运维团队支持集团9000+研发人员，承载集团1700+业务应用；
- 平均降低30%基础设施成本，服务可用率提升至99.996%，实现平衡安全，成本，弹性和敏捷基础IT架构。

联想多维度业务融合与变革驱动 -- 云原生PaaS平台的演进



缺乏标准化，业务融合难



人员协作管理成本高



沉淀管理经验，业务平滑稳定过渡

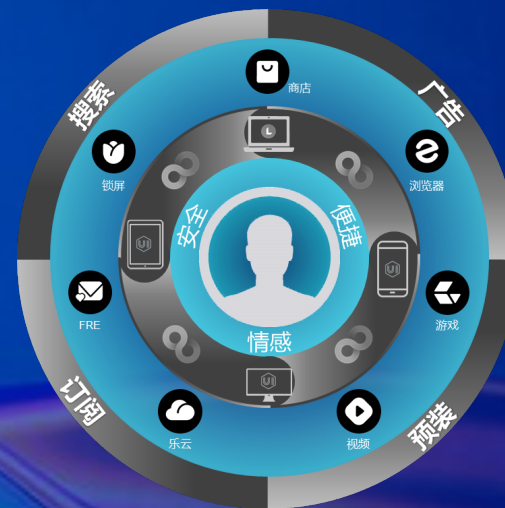


联想CIS（消费者生态互联网）业务背景



消费业务应用场景

联想CIS（消费生态互联网）业务承载联想云游戏、电脑管家、软件商店、浏览器等诸多消费业务场景，包含内容管理、身份认证、推送服务、移动支付等多功能服务。



联想消费互联网服务（CIS）承载联想面向广大消费者的消费业务应用，线上线下已有数百个服务的实例在运行中，在上云前业务应用主要部署在物理机或虚拟机中，通过机房的对外出口将应用发布至互联网中，部分访问量较高的关键应用，单独运行多实例在不同的服务器上。

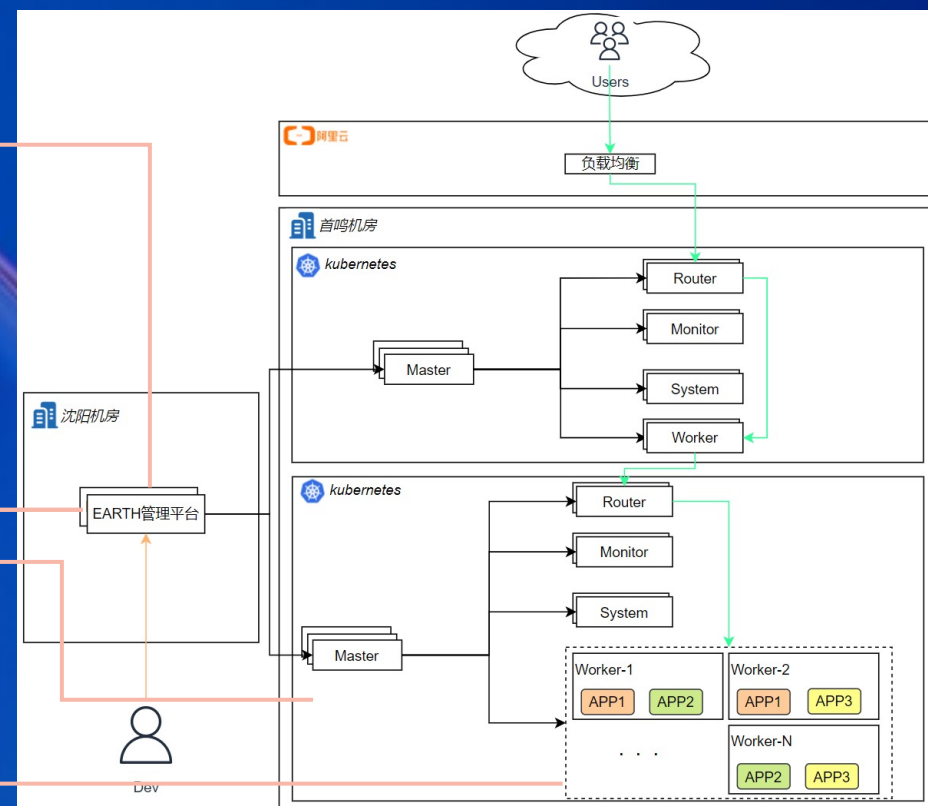
联想CIS（消费者生态互联网）技术架构



- 利用全球负载均衡服务，识别访问CIS的正常流量与突发异常流量，多集群架构，由私有云容器集群承载低谷期流量、公有云容器集群承载高峰期或突发异常流量。
- 私有云注重数据存储安全性与业务承载韧性，通过在双活数据中心部署双活集群，保证数据不丢失、业务不中断。

技术架构

- 1.利用EARTH平台自动扩缩容特性，根据业务负载进行动态横向扩展；
- 2.通过平台集中部署服务，实现新版应用秒级更新；
- 3.通过平台管理应用历史部署版本，新版本出问题后能快速回滚；
- 4.通过内外网集群架构，隐藏业务真实服务器IP地址，有效防止DDOS攻击，提升安全性；
- 5.通过Kubernetes的服务调度策略，提升Worker节点的使用率。



自服务平台，打造全新的高效服务模式



开发团队专注于业务场景研发, 通过容器云平台自服务能力进行应用部署、发布、上线以及应用日常运维

研发团队A

研发团队B

研发团队C

...

研发团队Z

自服务

自服务

自服务

自服务

云原生各平台

云原生平台运维团队

运维团队通过容器云平台以少量人员保障平台稳定运行, 支撑上层应用
同时能够治理所有研发团队使用的技术栈和技术架构

操作系统运维团队

存储运维团队

网络运维团队

服务器运维团队

安全团队

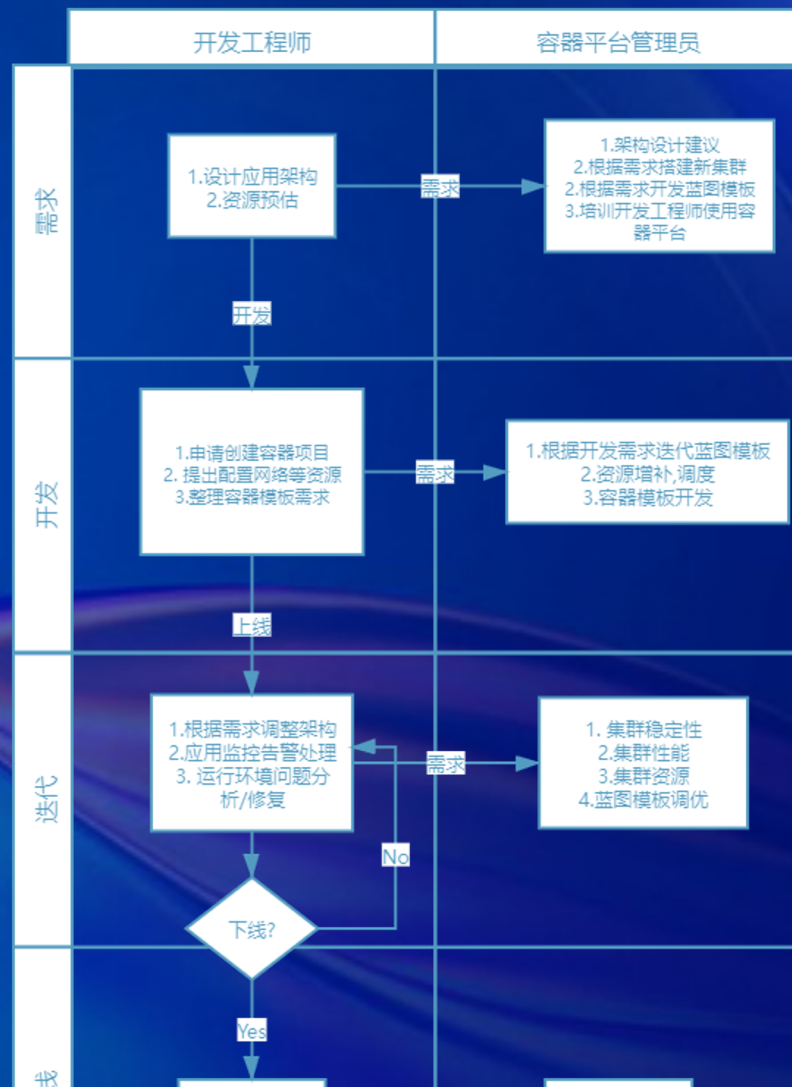
技术支持

容器云平台运维责任模型和流程图

通过容器云平台把控各岗位职责及责任模型，从需求提出、开发、迭代至上线全流程提供架构设计及标准模板，保障了平台、应用的稳定运行及性能提升。



平台运维团队	平台可用性	容器平台的总体可用性, 包括容器控制台、Kubernetes的master节点node节点、容器 Router(7层网络入口)、容器 Builder/Compiler(构建器/代码编译器)、Registry(镜像中心)的可用性, 容器所提供的日志、监控、告警等服务的可用性。
	平台性能	容器的节点资源问题导致的性能问题, 容器 Router导致的网络访问性能问题, 容器集群内容容器到容器访问的性能问题。
	平台安全	容器平台的组件安全问题, 容器平台的服务器的安全问题, 容器平台提供的模板中的安全问题 (如Nginx、Tomcat等), 容器 Router的安全问题。
	应用模板开发	根据开发团队需求开发或更新应用蓝图模板。全局管理模板中的技术栈合规性, 安全性等问题。
	平台容量管理	包括容器平台底层集群和资源的资源监控, 规划, 搭建, 扩容, 缩容。
应用开发团队	应用资源	应用域名申请, 应用防火墙开通, 应用数据库、消息队列、业务日志收集、应用业务数据持久存储等业务的申请及搭建。
	业务和架构	应用业务逻辑开发, 应用逻辑/部署架构设计, 应用技术栈选择, 应用集成关系设计。
	应用性能	代码及应用本身架构引起的性能问题; 分析应用所依赖的数据库、消息队列、集成或调用的其他系统等引起的性能问题。
	应用可用性	代码及应用本身架构引起的可用性; 应用所依赖的数据库、消息队列、集成或调用的其他系统等引起的可用性问题。
	应用bug	应用功能bug, 代码导致的502错误或其他应用报错等非正常响应问题。
	应用安全	应用代码的安全问题; 应用所发布的服务 (页面/API/Socket等) 的安全问题。



降低各研发团队的使用门槛



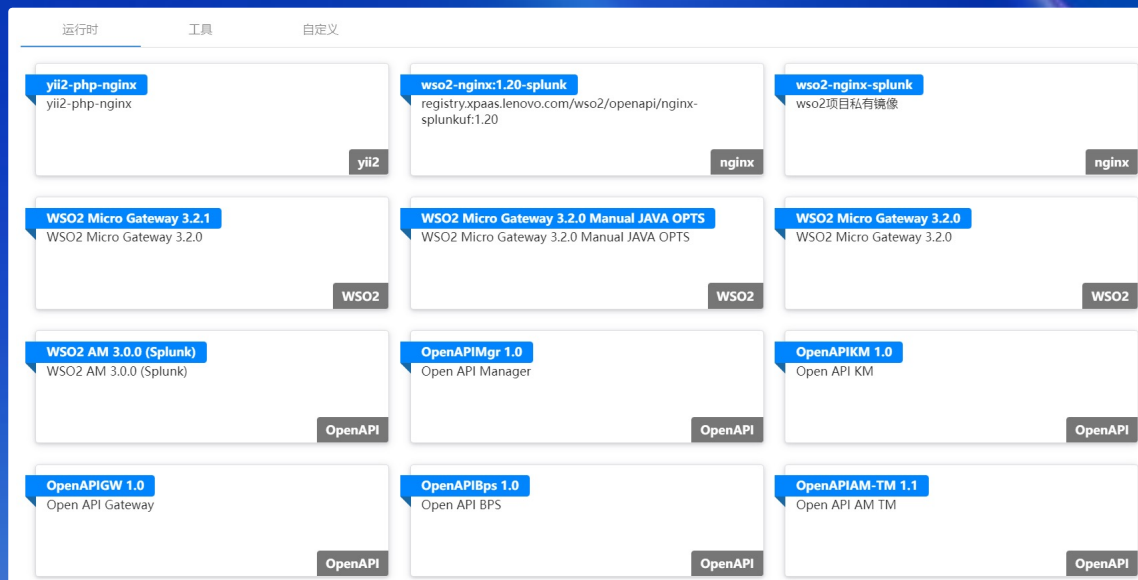
- ✓ 无须掌握容器、微服务及K8S技术
- ✓ 黑屏→白屏，图形化代替命令行
- ✓ 集群搭建可视化，降低难度
- ✓ 集群搭建可视化，降低难度
- ✓ 通过平台支撑应用全生命周期



上平台——配置服务模板



普通服务发布流程



1. 同时支持有状态服务和无状态服务模板，也支持用户自定义服务模板；
2. 有状态服务模板包含多种容器运行时环境（如Nginx、Java、NodeJS、Python等）；
3. 无状态服务模板包含Mongo、Redis、MariaDB等多种容器运行时环境。