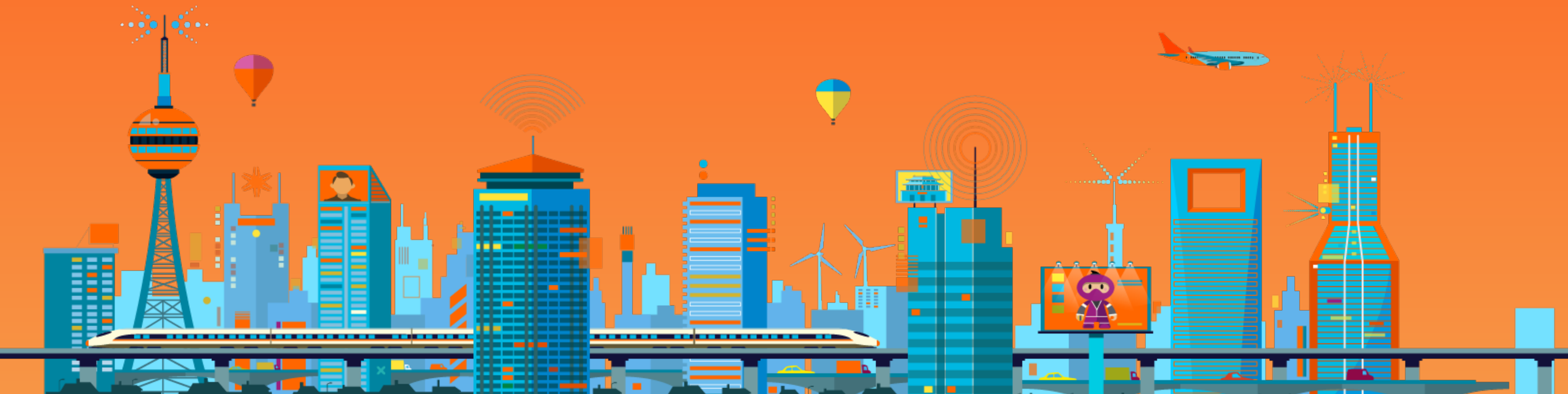
Alibaba Group
阿里巴巴集团

像搭积木一样玩转Docker的持续交付

莫源

zhongwei.lzw@alibaba-inc.com



大纲

用Lego的思维看基于Docker的交付方式

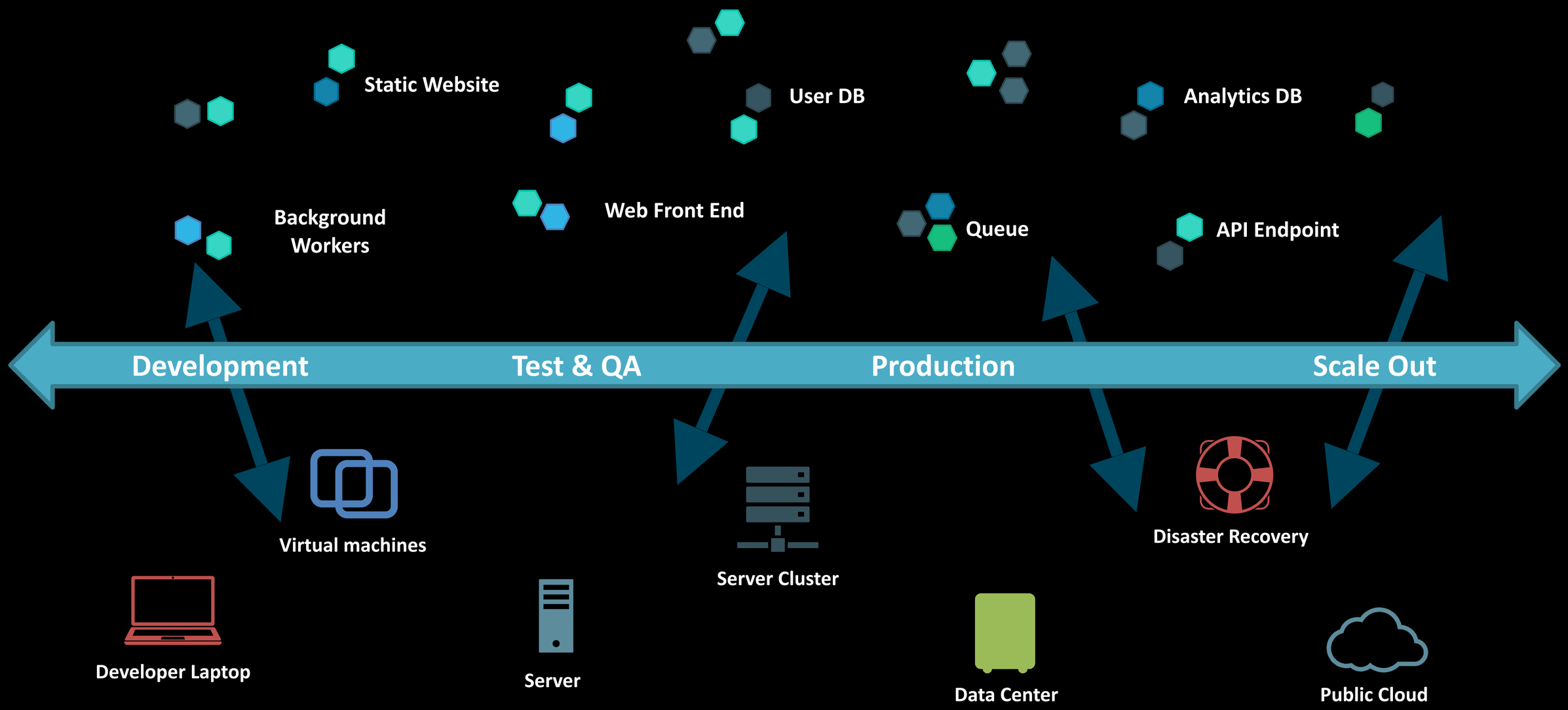
持续交付 + Docker = ContainerOps

阿里云容器服务持续集成模组

从零组装属于自己的持续交付系统

用Lego的思维看基于Docker的交付方式

不同应用，不同环境，不同开发交付运维方式带来的挑战



常见的交付流程

Development **Test & QA** **Production** **Scale Out**

开发人员开发代码并在本地或者测试环境通过

测试人员自动或者手动验证用例

运维人员准备基础环境，以及自动部署的脚本（如果有的话）另外进行日志收集、报警监控、应用扩速容、系统调优。

1.有一套基础的技术栈和环境自动化的流程
2.团队的技术框架尽可能的保持稳定
3.有良好的文档与技术沉淀
4.团队成员相对稳定

当平衡被打破

反思问题

1.交付流程不能良好的自动化或者自动化的成本过高

2.交付缺乏自描述的机制，严重依赖文档或者口传心授

3.不同的角色对于职责的交叉地带缺乏合作

总结成三个词：自动集成难、持续交付难、合作协同难

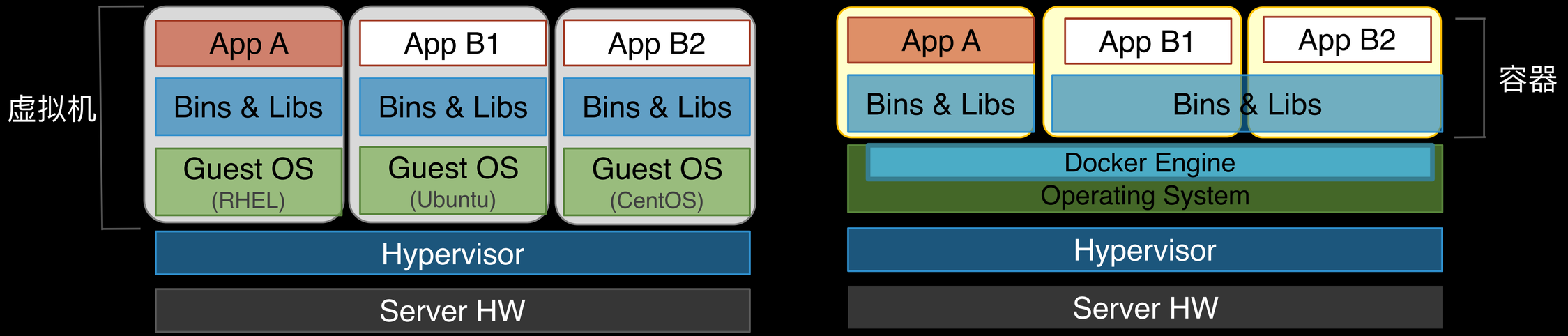
Docker为交付而生

Docker是一种轻量级的操作系统虚拟化方案，为**交付**而生

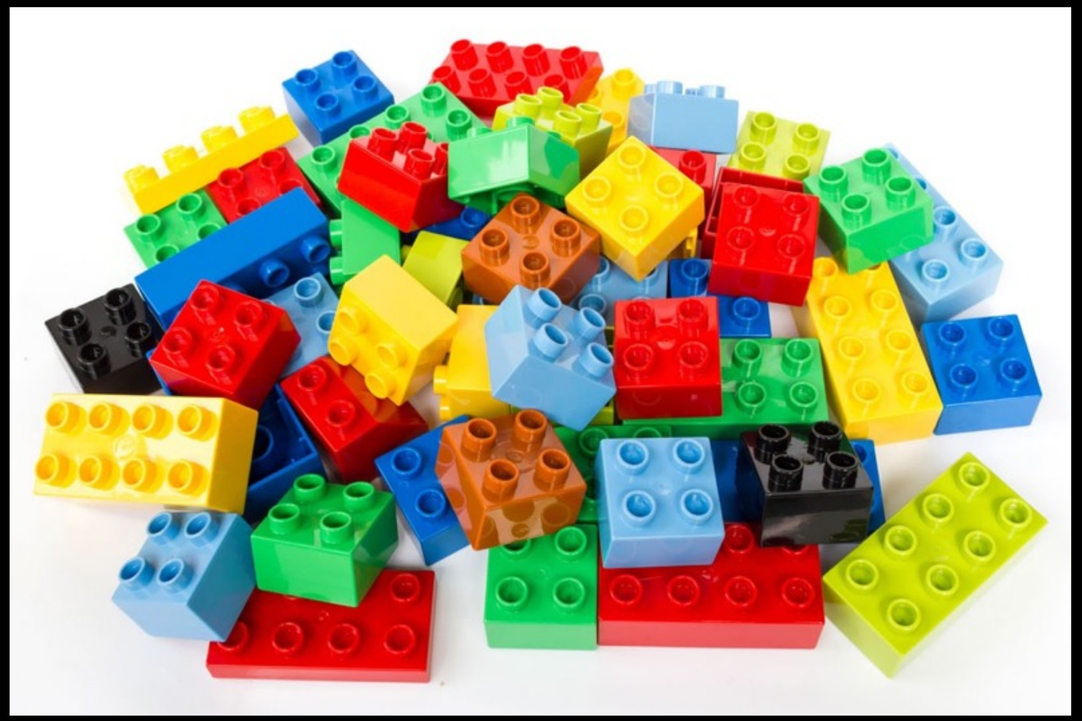
结合Docker容器和虚拟化技术

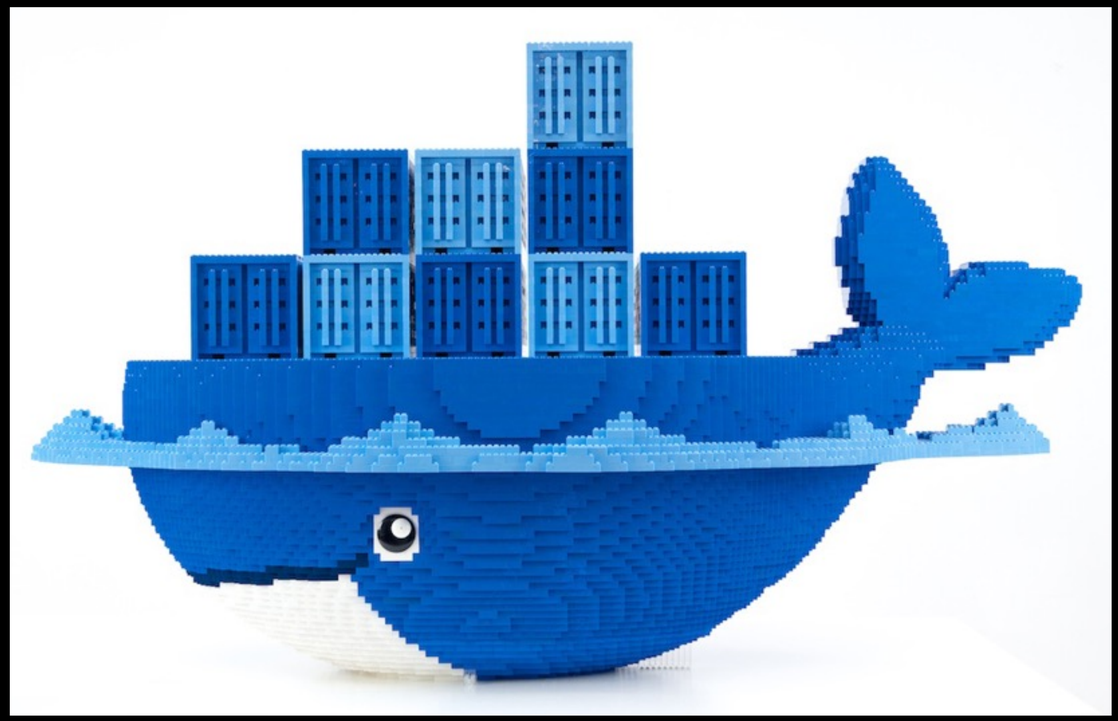
利用虚拟机提供弹性基础架构，更好的安全隔离，动态热迁移

利用容器技术实现简化应用部署、运维；实现弹性应用架构



Lego做了一件伟大的事情 — **标准化模组**





Docker同样做了一件伟大的事情 — **标准化交付（软件定义交付）**

1.复杂的环境变成了标准的**Dockerfile**，不同的环境可以通过同样的方式打包。 **标准化了软件环境。**
2.不同种类的应用可以用**相同的命令或者API**用相同的管理方式，统一管理。 **标准化了软件形态。**
3.不同的应用拓扑关系变成了遵循特定语法的**编排模板**，编排模板可以自描述。**标准化了交付流程。**

持续交付 + Docker = ContainerOps

你若问十个哲学家什么是**『哲学』**
通常你会得到十一种答案（有一种是你自己的）

你若问十个持续集交付道师什么是**『持续交付』**
你恐怕得到的是上百种答案（因为你自己也有好几种）

只有一个哲学问题是严肃的，那就是**生与死**

关于持续集成所需要了解的第一件事情，那就是只有**符合自己业务形态**的持续集成才是你需要的持续集成

PERIODIC TABLE OF DEVOPS TOOLS (V2)

EMBED DOWNLOAD ADD

Os	Open Source
Fr	Free
Fm	Freemium
Pd	Paid
En	Enterprise

SCM	Database Mgmt	Build
CI	Repo Mgmt	Testing
Deployment	Config / Provisioning	Containerization
Cloud / Iaas / Paas	Release Mgmt	Collaboration
BI / Monitoring	Logging	Security

1 Fm Gh Github																				2 Fm Aws Amazon Web
3 Os Gt Git	4 En Dm DBmaestro														5 En Ch Chef	6 En Pu Puppet	7 Os An Ansible	8 Os Sl Salt	9 Os Dk Docker	10 Pd Az Azure
11 Fm Bb Bitbucket	12 Os Lb Liquibase														13 Os Ot Otto	14 En Bl BladeLogic	15 Os Va Vagrant	16 Fr Tf Terraform	17 Os Rk rkt	18 En Gc Google Cloud
19 Os Gl GitLab	20 En Rg Redgate	21 Os Mv Maven	22 Os Gr Gradle	23 Os At ANT	24 Os Fn FitNesse	25 Fr Se Selenium	26 Os Ga Gatling	27 Fr Dh Docker	28 Os Jn Jenkins	29 Pd Ba Bamboo	30 Os Tr Travis CI		31 Pd Gd Deployment Manager	32 Os Sf SmartFrog	33 Os Cn Consul	34 Os Bc Bcfg2	35 Os Mo Mesos	36 En Rs Rackspace
37 Os Sv Subversion	38 En Dt Datical	39 Os Gt Grunt	40 Os Gp Gulp	41 Os Br Broccoli	42 Fr Cu Cucumber	43 Os Cj Cucumber.	44 Fr Qu Qunit	45 Os Npm npm	46 Fm Cs Codeship	47 Pd Vs Visual	48 Fm Cr CircleCI		49 Fr Cp Capistrano	50 Fr Ju JuJu	51 Os Rd Rundeck	52 Os Cf CFEngine	53 Fr Ds Swarm	54 Os Op OpenStack
55 Os Hg Mercurial	56 En Dp Delphix	57 Fr Sb sbt	58 Os Mk Make	59 Os Ck CMake	60 Fr Jt JUnit	61 Fr Jm JMeter	62 Fr Tn TestNG	63 Os Ay Artifactory	64 Fm Tc TeamCity	65 Fm Sh Shippable	66 Os Cc CruiseCont		67 En Ry RapidDeplo	68 Fm Cy CodeDeplo	69 En Oc Octopus Deploy	70 En No CA Nolio	71 Os Kb Kubernetes	72 Fm Hr Heroku
73 En Cw ISPW	74 En Id Idera	75 Os Msb MSBuild	76 Os Rk Rake	77 Fr Pk Packer	78 Os Mc Mocha	79 En Xltv XL	80 Os Jm Jasmine	81 Os Nx Nexus	82 Os Co Continuum	83 Fm Ca Continua	84 Pd So Solano CI		85 En Xld XL Deploy	86 En EB ElasticBox	87 Fm Dp Deploybot	88 En Ud UrbanCode Deploy	89 Os Nm Nomad	90 En Os OpenShift
													91 En Xlr XL	92 En Ur UrbanCode Release	93 En Bm BMC Release	94 En Hp HP Codar	95 En Au Automic	96 En Pl Plutora Release	97 En Sr Serena Release	98 Pd Tfs Team Foundation
													99 Fm Tr Trello	100 Pd Jr Jira	101 Fm Rf HipChat	102 Fm Sl Slack	103 Fm Fd Flowdock	104 Pd Pv Pivotal Tracker	105 En Sn ServiceNow	
													106 Os Ki Kibana	107 Fm Nr New Relic	108 Os Ni Nagios	109 Os Zb Zabbix	110 En Dd Datadog	111 Os El Elasticsear	112 En St StackState	113 En Sp Splunk
													114 Fm Le Logentries	115 Fm Sl Sumo	116 Os Ls Logstash	117 Os Gr Graylog	118 Os Sn Snort	119 Os Tr Tripwire	120 En Ff Fortify	

XebiaLabs
Deliver Faster

Follow @xebialabs

持续交付真正严肃的问题只有三个

问题一：如何**重建**系统　　(How to **recreate** your system?)
翻译为：如何**初始化**环境

问题二：如何安全地**部署**系统　　(How to safely **change** your system?)
翻译为：如何**发布**应用

问题三：部署后的**问题监控与解决**　　(When something has gone **wrong**?)
翻译为：**监控、回滚、扩缩容**

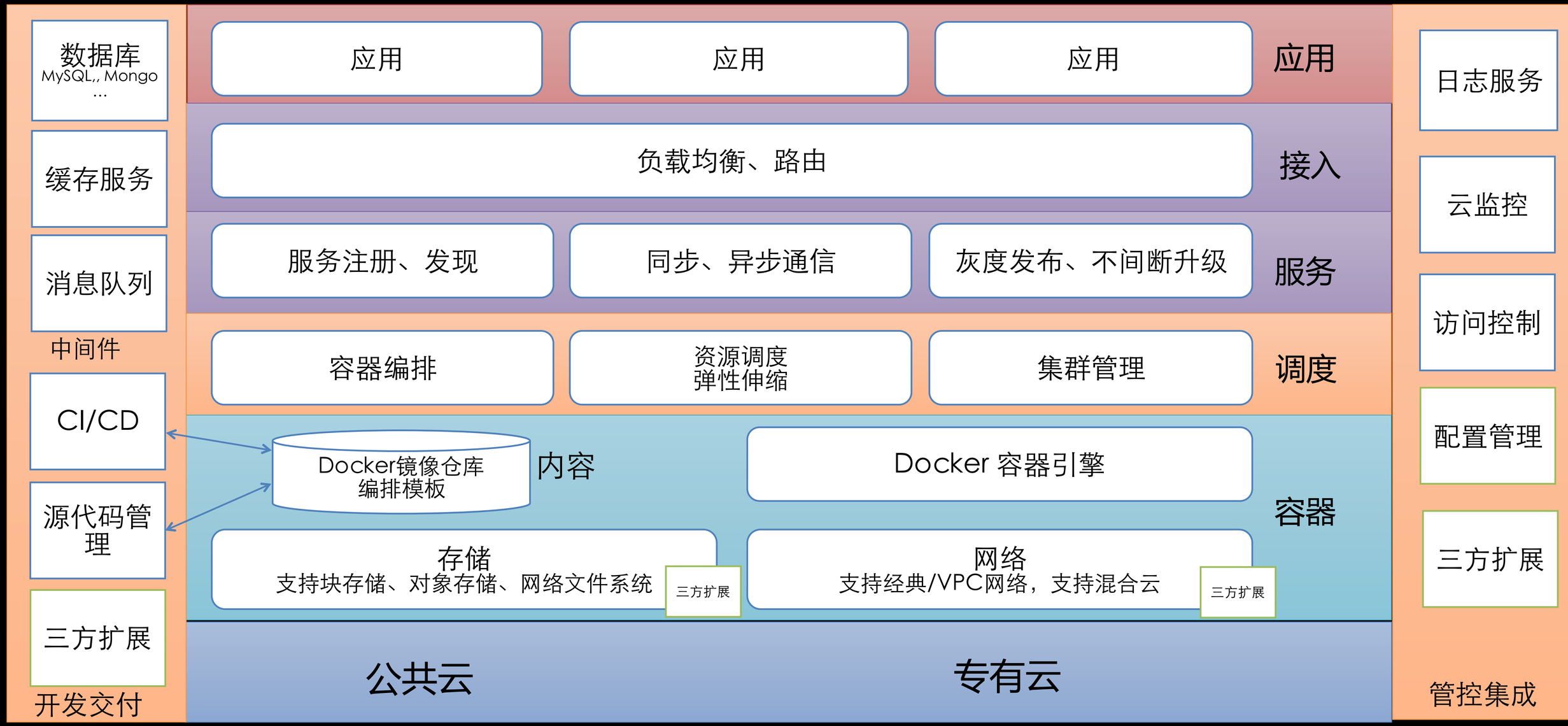
如何用Docker来回答上述的问题

1.环境与软件变成了**Dockerfile**定义的镜像，从镜像可以重建系统
2.相互关联的软件可以通过**编排模板**标准化部署，本地可以运行远程便可交付
3.通过Docker**标准的API**获取容器的状态，出现问题可以通过镜像回滚、更新、扩容

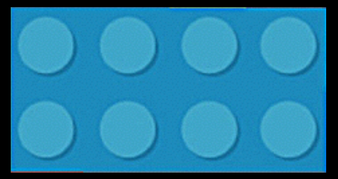
如何用阿里云的容器服务实现持续交付?

阿里云容器服务模组

阿里云容器服务概况

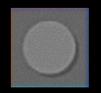

数据库
MySQL,, Mongo
...
缓存服务
消息队列
中间件
CI/CD
源代码管理
三方扩展
开发交付
应用
应用
应用
应用
负载均衡、路由
接入
服务注册、发现
同步、异步通信
灰度发布、不间断升级
服务
容器编排
资源调度
弹性伸缩
集群管理
调度
Docker镜像仓库
编排模板
内容
Docker 容器引擎
存储
支持块存储、对象存储、网络文件系统
三方扩展
网络
支持经典/VPC网络，支持混合云
三方扩展
容器
公共云
专有云
日志服务
云监控
访问控制
配置管理
三方扩展
管控集成

容器Hub持续交付模组



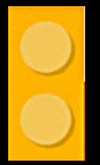
容器Hub

阿里云Code

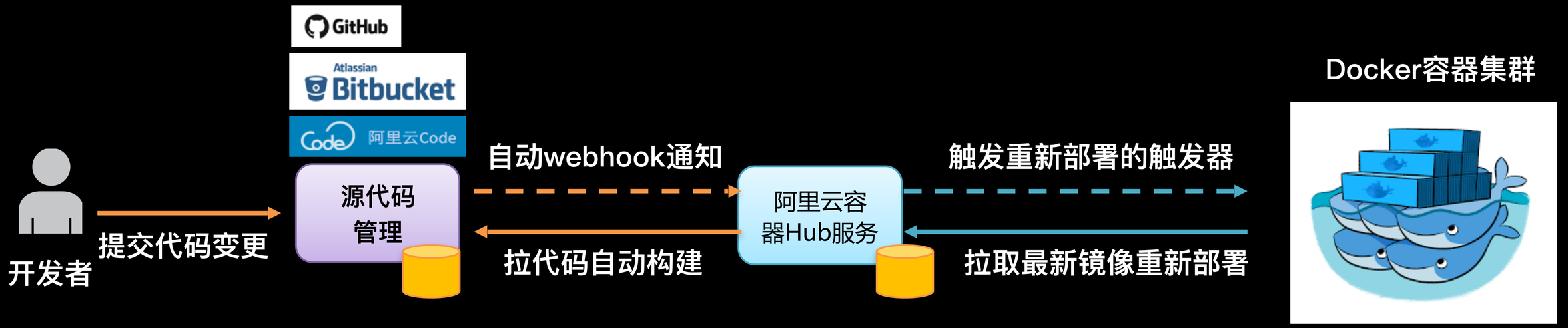


github

bitbucket



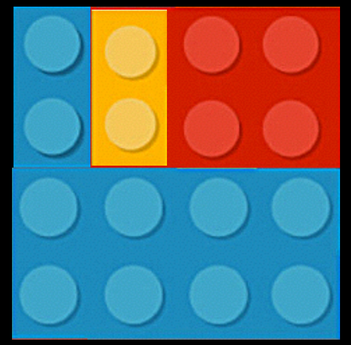
容器服务触发器



步骤：用户提交代码，Hub会根据代码路径下的Dockerfile自动进行镜像构建，在Hub中可以配置重新部署的触发器构建完毕后，会自动触发配置的触发器，进行重新部署。

特性：支持海外构建、支持构建触发器、支持构建缓存、支持私有仓库

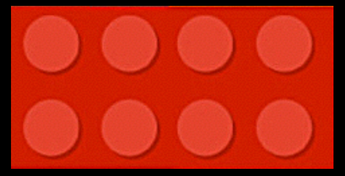
Jenkins持续交付模组



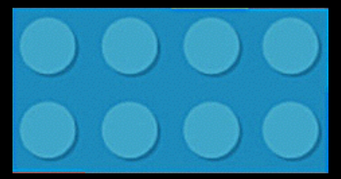
阿里云Code

github

bitbucket

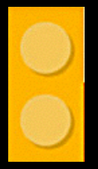


Jenkins Master

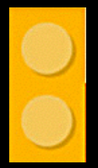


容器Hub

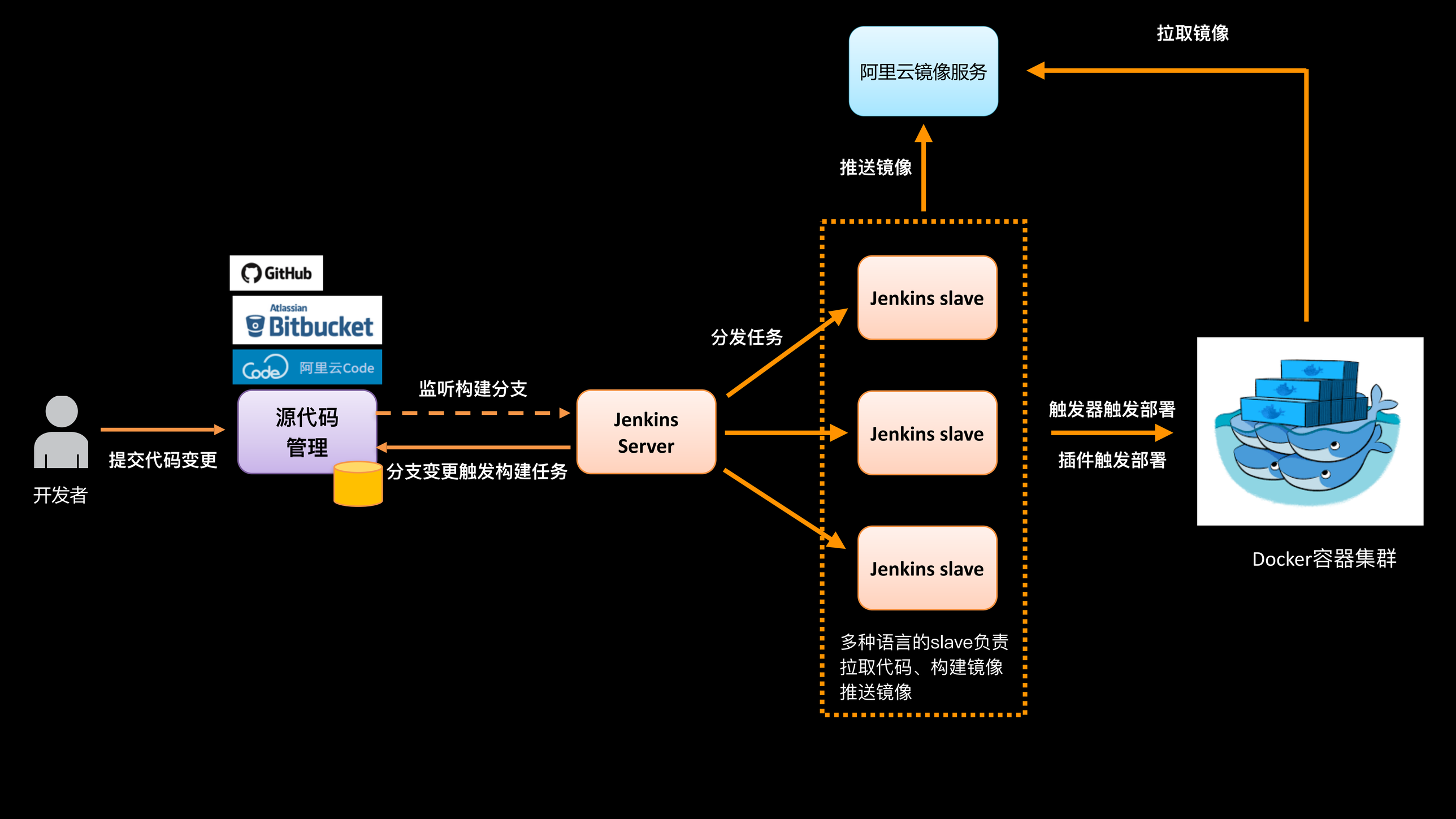
Jenkins slaves



容器服务触发器



Jenkins部署插件


阿里云镜像服务
拉取镜像
推送镜像
GitHub
Atlassian Bitbucket
阿里云Code
源代码管理
监听构建分支
分支变更触发构建任务
提交代码变更
开发者
Jenkins Server
分发任务
Jenkins slave
Jenkins slave
Jenkins slave
多种语言的slave负责
拉取代码、构建镜像
推送镜像
触发器触发部署
插件触发部署
Docker容器集群

slave中的流程

原理：通过Jenkins监听代码分支的的变化，代码产生特定行为的时候触发构建
构建是通过Jenkins各种不同语言的slave实现的，在slave中根据代码路径中的Dockerfile进行镜像的构建，并推送容器Hub。最后触发重新部署，或者使用插件进行重新部署。

特性：灵活、安全、简单，扩展丰富，支持自定义插件的开发，功能非常强大。

遇到的问题与选择

1.Jenkins官网的镜像为什么大多跑不起来

解：Jenkins默认执行的权限为Jenkins的user，挂载的jenkins_home写权限不足
容器服务通过先提权再解权的方式解决镜像运行权限的问题。

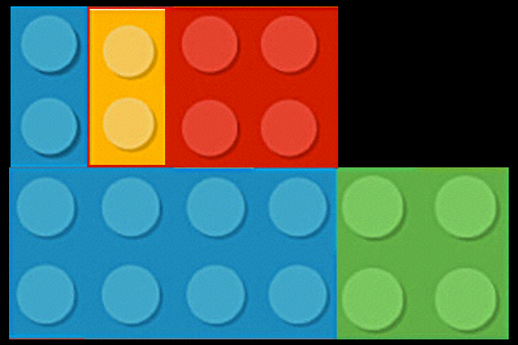
2.Jenkins slave中构建镜像，怎么处理docker in docker

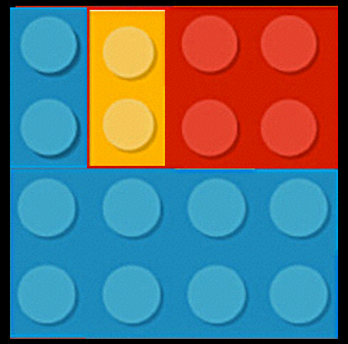
解：可以将/var/run/docker.sock传递进入slave，这样slave中的Docker相当于一个client
而真正的Engine是宿主机的docker，会使用宿主机的资源。

3.Jenkins持续交付中多套环境如何解决

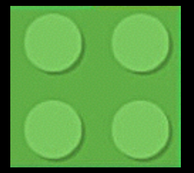
解：可以采用git workflow的方式，实现不同分支不同权限，开发环境、预发环境自动发布
不同的特定分支发布到特定的环境，线上分支采用手动发布，可以使用多种发布策略实现
0宕机发布。

蓝绿发布模组

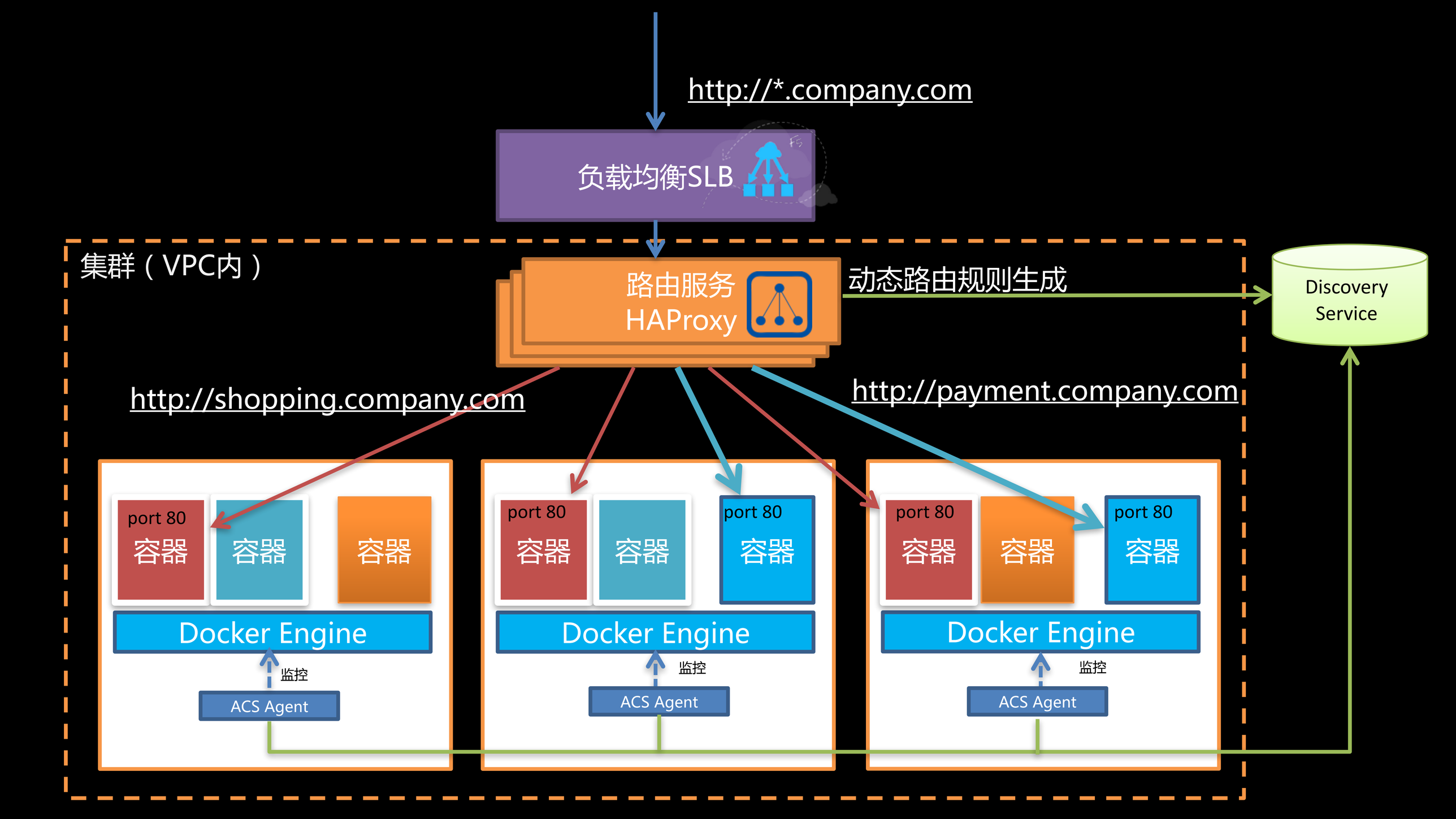




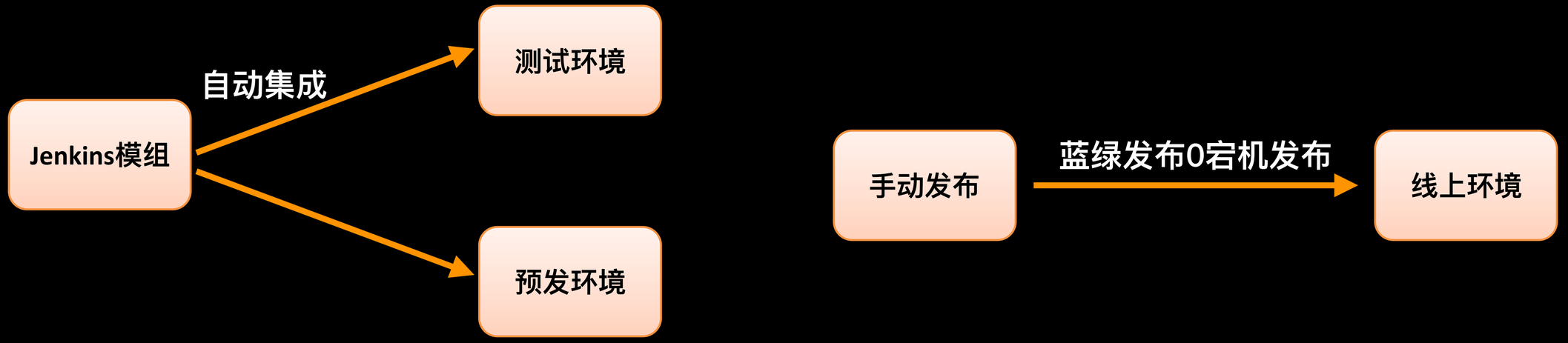
Jenkins持续交付模组



路由服务


http://*.company.com
负载均衡SLB
集群（VPC内）
路由服务
HAProxy
动态路由规则生成
Discovery
Service
http://shopping.company.com
http://payment.company.com
port 80
容器
容器
容器
Docker Engine
监控
ACS Agent
port 80
容器
容器
port 80
容器
Docker Engine
监控
ACS Agent
port 80
容器
容器
port 80
容器
Docker Engine
监控
ACS Agent

蓝绿发布的场景



过程：两个版本并存，通过权重的方式切换流量的上线或者下线
验证无误进行发布确认，老版本容器删除
验证无法通过发布回滚，新版本容器删除

从零开始组装复杂的持续交付系统

FAQ

所有内容的资源地址