

采用开源云原生仓库Harbor实现 安全可靠与可控的容器镜像运维

张海宁 VMware中国研发中心技术总监

2018/06

自我介绍

VMware中国研发先进技术中心技术总监, 负责区块链、云原生应用等方面创新项目

Harbor开源企业级容器Registry项目创始人

Cloud Foundry中国社区最早技术布道师

超级账本Hyperledger Cello项目贡献者

《区块链技术指南》作者之一



公众号：亨利笔记



《区块链技术指南》

议程

1 容器镜像运维基础

2 云原生容器镜像仓库

2.1 开源项目Harbor简介

2.2 安全的容器镜像管理

2.3 可控的容器镜像分发

2.4 可靠的容器镜像运维

3 Kubernetes下的容器镜像运维

议程



1 容器镜像运维基础

2 云原生容器镜像仓库

2.1 开源项目Harbor简介

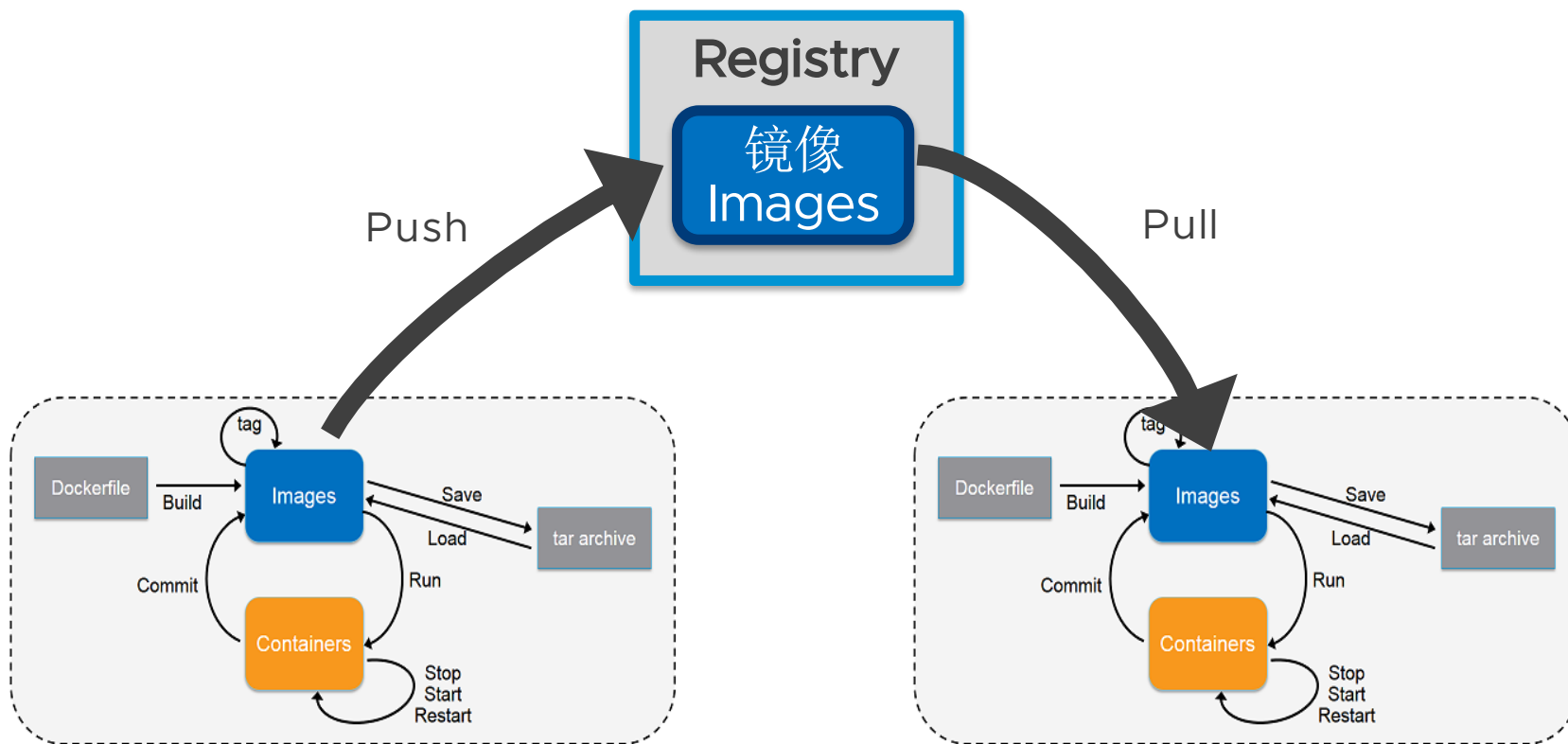
2.2 安全的容器镜像管理

2.3 可控的容器镜像分发

2.4 可靠的容器镜像运维

3 Kubernetes下的容器镜像运维

Registry-镜像管理的关键点



- 镜像存储仓库
- 分发镜像的媒介
- 访问控制和镜像管理较佳节点

议程

1 容器镜像运维基础

2 云原生容器镜像仓库

2.1 开源项目Harbor简介

2.2 安全的容器镜像管理

2.3 可控的容器镜像分发

2.4 可靠的容器镜像运维

3 Kubernetes下的容器镜像运维



一个开源的可信云原生容器镜像仓库项目

vmware.github.io/harbor



由 VMware 中国
团队设计和开发

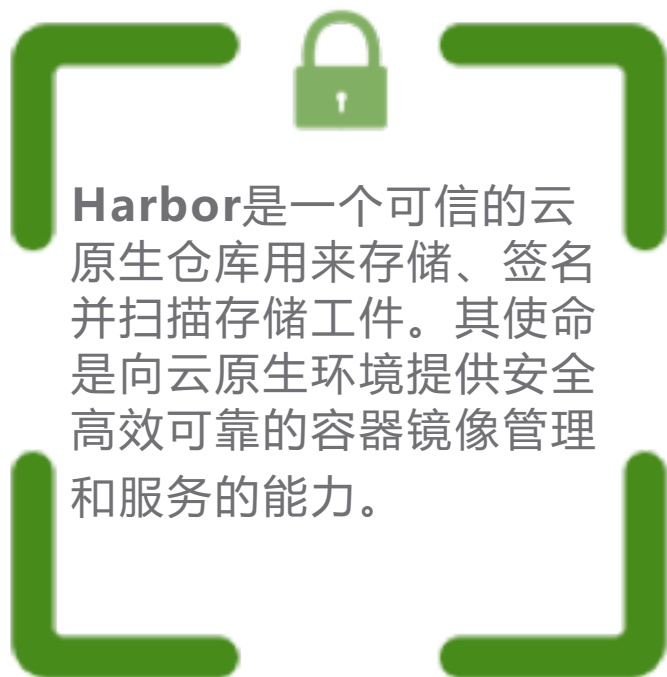


集成到多个企业级
产品中:VIC和PKS



Apache 2.0
使用许可

Harbor聚焦点



Harbor是一个可信的云原生仓库用来存储、签名并扫描存储工件。其使命是向云原生环境提供安全高效可靠的容器镜像管理和服务的能⼒。

产品特征 (Features)

安全性 (Security)

灵活性 (flexibility)

可移植性 (portability)

可控性 (Controllability)

可靠性 (reliability)



可信 (Trust)



合规 (Compliance)

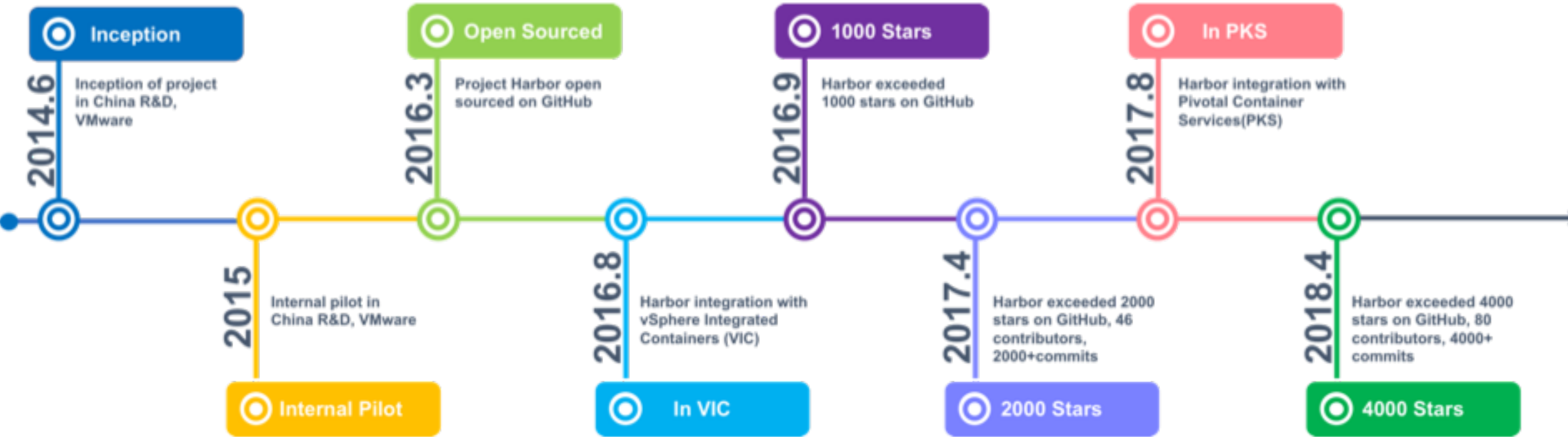


高效 (Performance)



互操作 (Interoperability)

Harbor项目简史



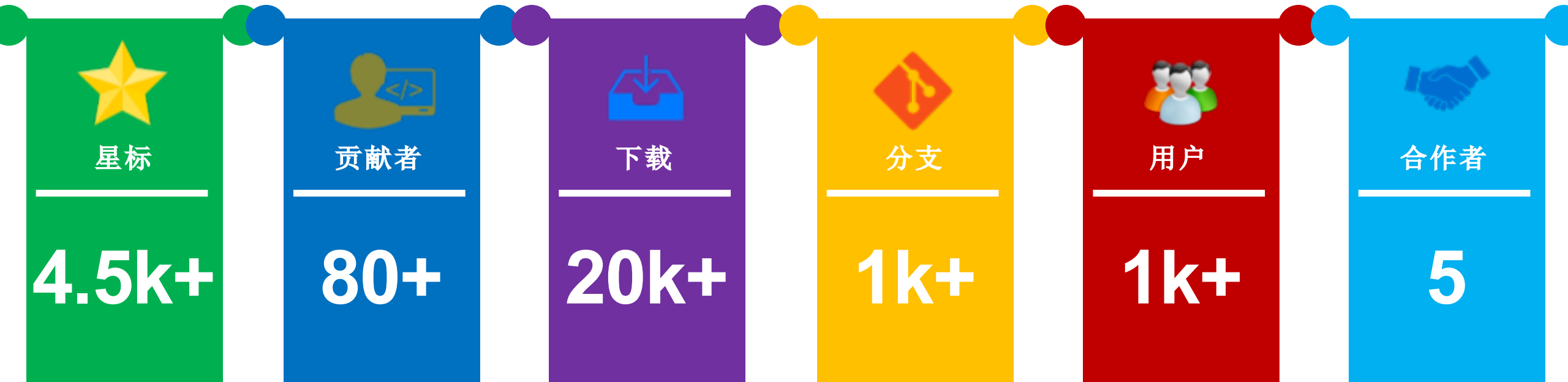
未来更多集成方向



OPEN SERVICE BROKER API



Harbor社区概况



Harbor部分用户



主要特性



GUI支持

基于开源Clarity构建
完备的镜像运管能力
批处理操作支持



Restful API

完善的API支持集成
Swagger API 文档



远程复制

多种过滤器支持
定时,即时和手动触发



高可用

高可用性支持



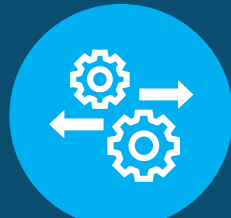
ova@vSphere

ova虚拟机



访问控制

基于角色的访问控制
AD/LDAP 用户集成



分发控制

基于内容信任
基于漏洞扫描
基于RBAC



漏洞扫描

多种漏洞扫描策略
详尽的漏洞扫描报告



内容信任

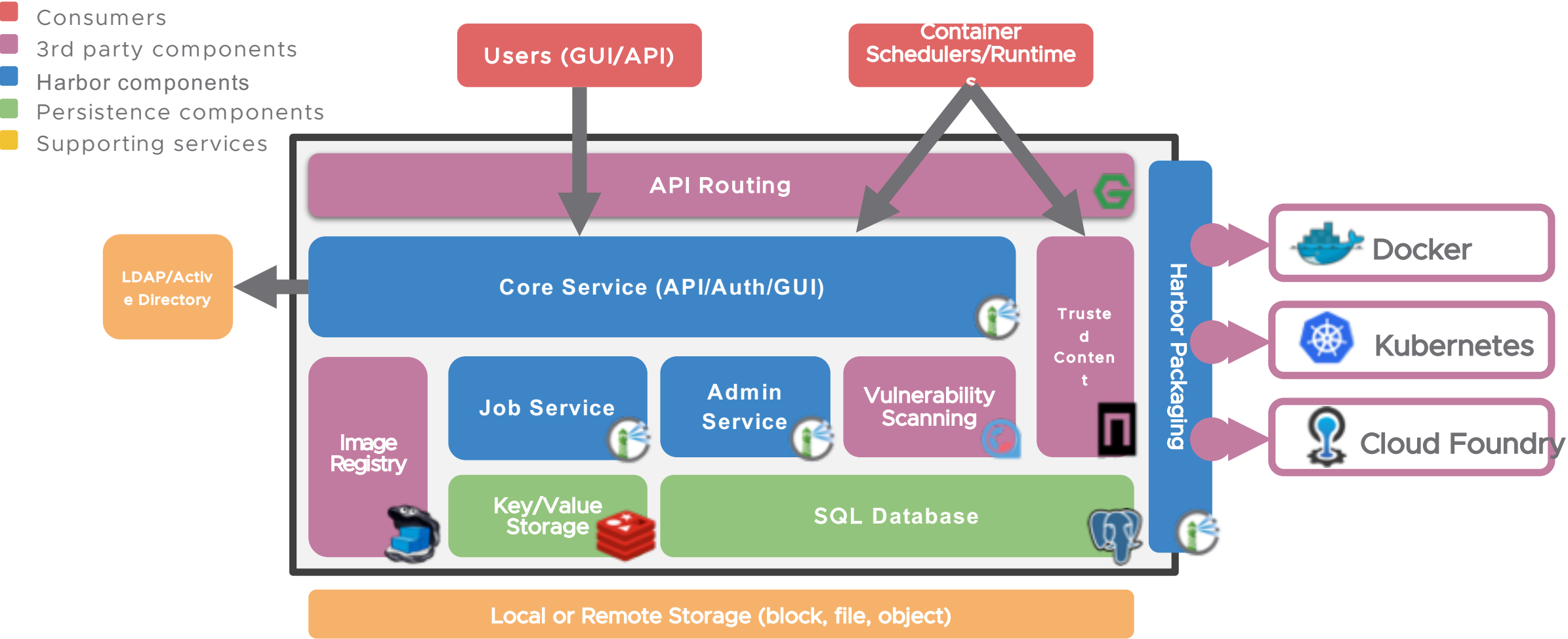
数字签名镜像



审计日志

操作日志记录以审计

Harbor架构



议程

1 容器镜像运维基础

2 云原生容器镜像仓库

2.1 开源项目Harbor简介

2.2 安全的容器镜像管理

2.3 可控的容器镜像分发

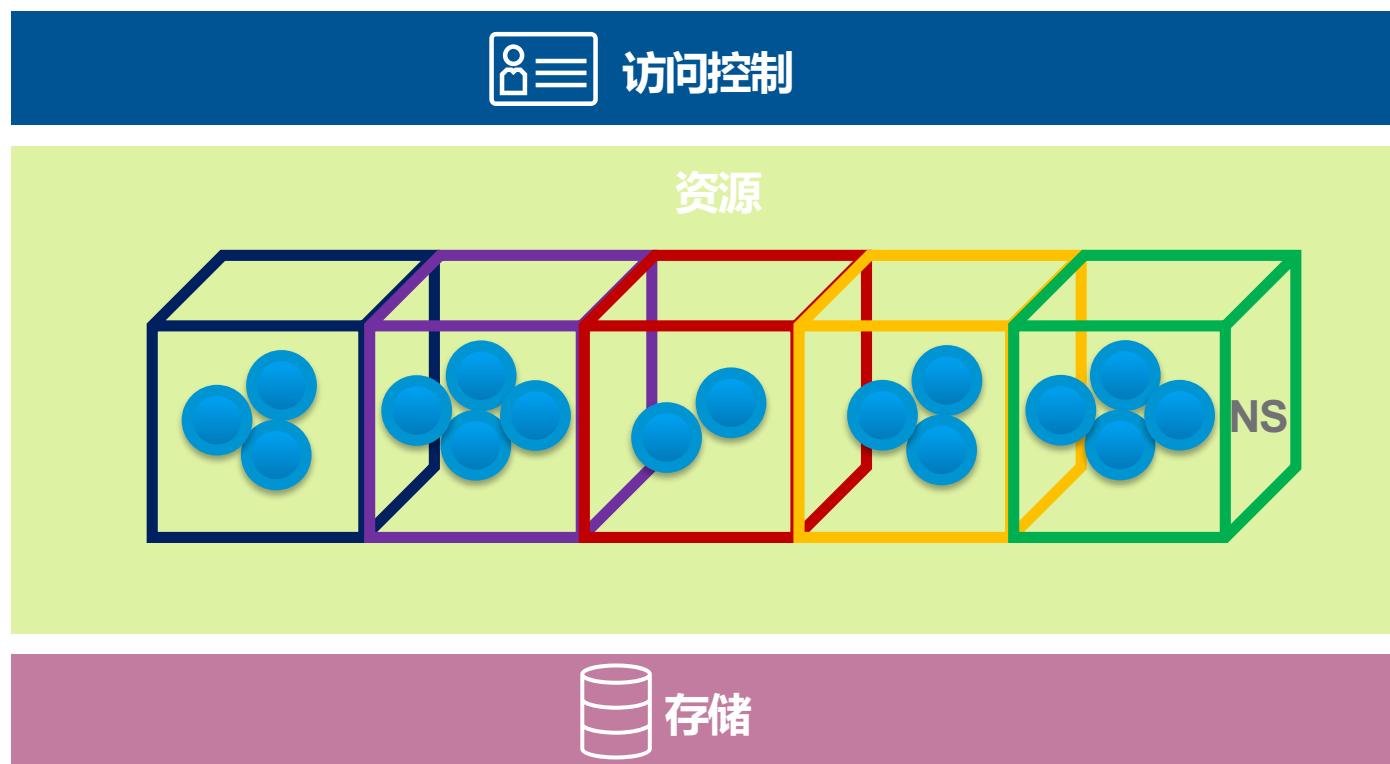
2.4 可靠的容器镜像运维

3 Kubernetes下的容器镜像运维

安全

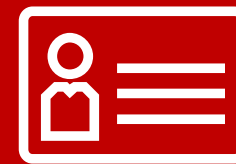
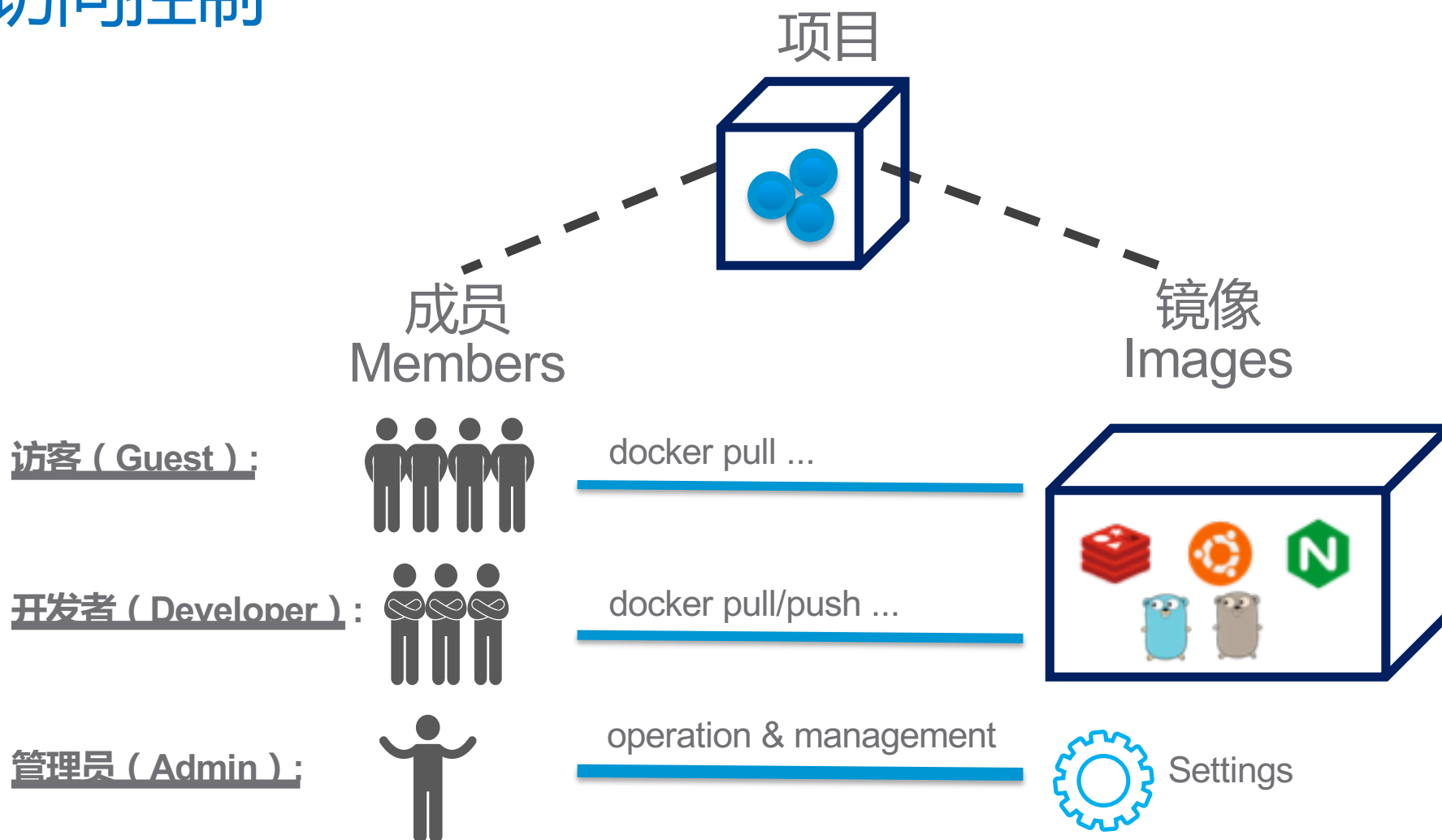


资源隔离



- 基于项目提供独立的NS
- 逻辑隔离，存储共享
- 访问控制的基础
- 为多租户提供可能

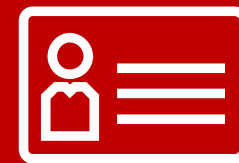
访问控制



访问控制

- 企业用户通常把镜像存放在组织内部
- 不同角色人员应有不同的访问权限
- 不同环境人员的角色不同
- 与已有的LDAP/AD用户系统集成

访问控制



访问控制

< 项目

library

镜像仓库 成员 日志 复制 配置管理

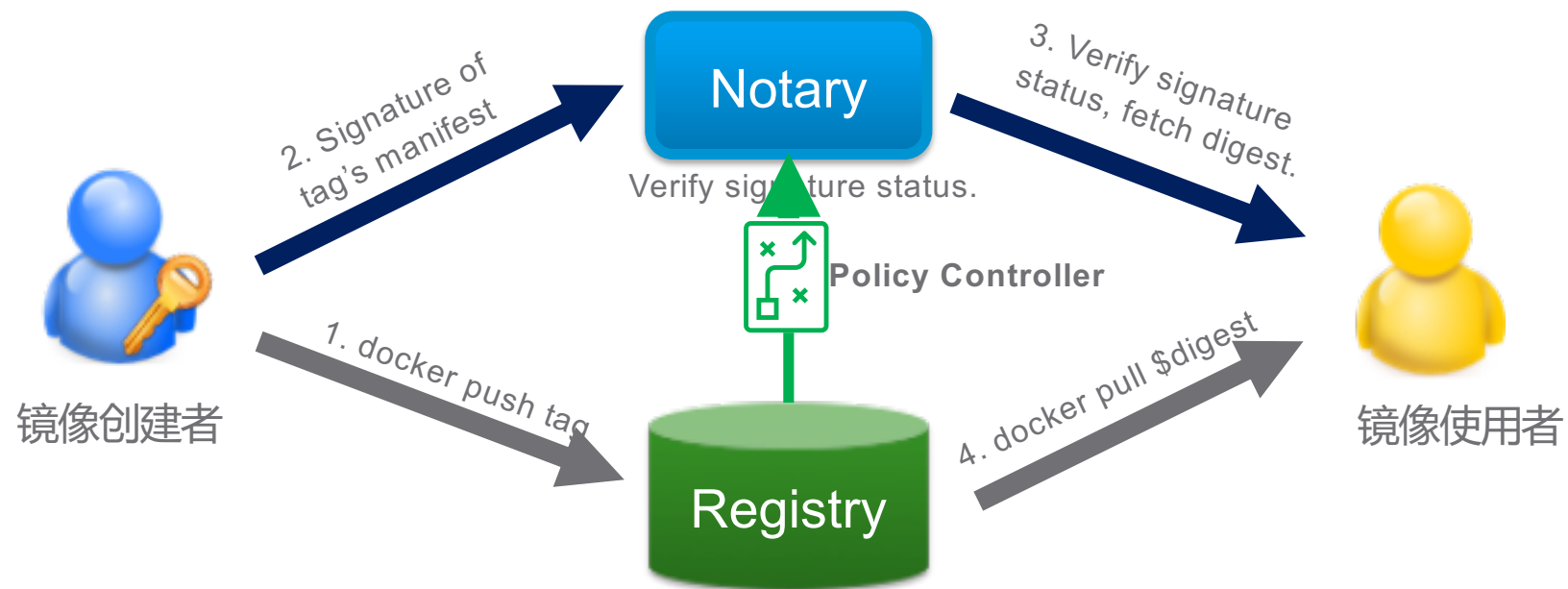
+ 新建成员 设置角色 v 移除成员

<input type="checkbox"/>	姓名	角色
<input type="checkbox"/>	admin@harbor.local	开发人员
<input type="checkbox"/>	wangyan01	访客

记录

- 企业用户通常把镜像存放在组织内部
- 不同角色人员应有不同的访问权限
- 不同环境人员的角色不同
- 与已有的LDAP/AD用户系统集成

内容信任



内容信任

- 发布者对镜像签名
- 下载镜像时使用签名摘要 (Digest)

内容信任



内容信任

- 发布者对镜像签名
- 下载镜像时使用签名摘要 (Digest)

描述信息

镜像

扫描

复制摘要

删除

Q | C

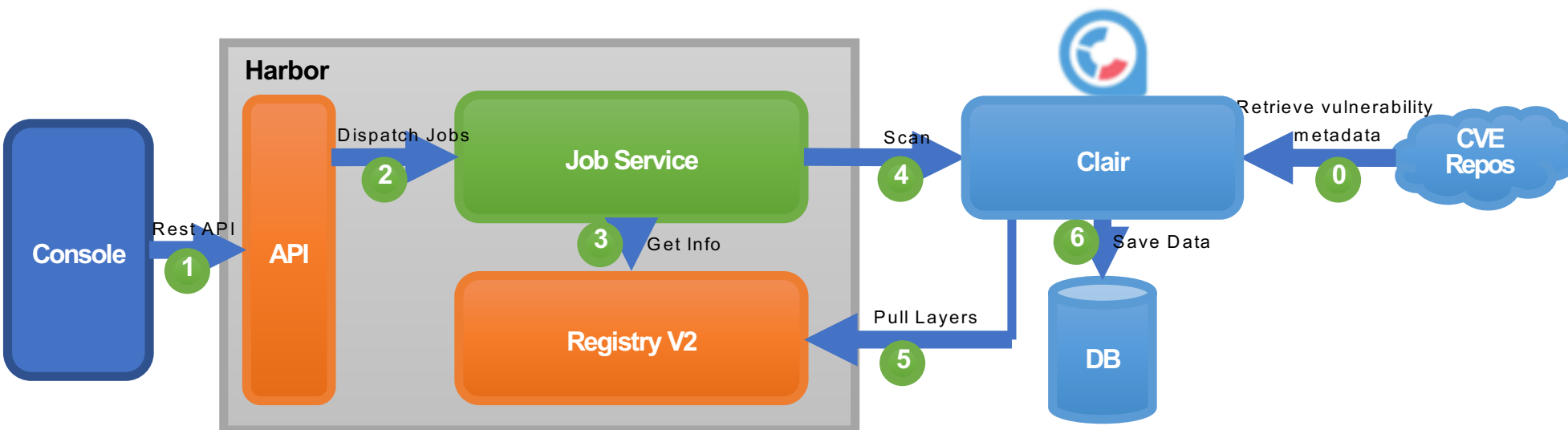
<input type="checkbox"/>	标签	大小	Pull命令	漏洞	已签名	作者	创建时间
<input type="checkbox"/>	v2	2.95KB		<div></div>			2017/11/21 上午8:23
<input type="checkbox"/>	latest	2.95KB		<div></div>			2017/11/21 上午8:23
<input type="checkbox"/>	v1	2.95KB		<div></div>			2017/11/21 上午8:23

漏洞扫描



漏洞扫描

- 对镜像文件做静态分析
- 多种扫描模式支持
 - 定时
 - 事件
 - 手动
- 漏洞数据库定期更新
- 多重漏洞数据来源
 - Debian Security Bug Tracker
 - Ubuntu CVE Tracker
 - Red Hat Security Data
 - Oracle Linux Security Data
 - Alpine SecDB



漏洞扫描



漏洞扫描

project061529900025/tomcat:latest

Author: anonymity
Architecture: amd64
OS: linux
Docker Version: 17.06.2-ce
Scan Completed: Jun 25, 2018

7 high Level Vulnerabilities
11 medium Level Vulnerabilities
13 low Level Vulnerabilities
2 unknown Level Vulnerabilities

SCAN



Vulnerability	Severity	Package	Current version	Fixed in version
▼ CVE-2017-17942	medium	tiff	4.0.8-2+deb9u2	
Description: In LibTIFF 4.0.9, there is a heap-based buffer over-read in the function PackBitsEncode in tif_packbits.c.				
> CVE-2018-5784	medium	tiff	4.0.8-2+deb9u2	
> CVE-2018-8905	medium	tiff	4.0.8-2+deb9u2	
> CVE-2017-17973	negligible	tiff	4.0.8-2+deb9u2	

- 对镜像文件做静态分析
- 多种扫描模式支持
 - 定时
 - 事件
 - 手动
- 漏洞数据库定期更新
- 多重漏洞数据来源
 - Debian Security Bug Tracker
 - Ubuntu CVE Tracker
 - Red Hat Security Data
 - Oracle Linux Security Data
 - Alpine SecDB

议程

1 容器镜像运维基础

2 云原生容器镜像仓库

2.1 开源项目Harbor简介

2.2 安全的容器镜像管理

2.3 可控的容器镜像分发

2.4 可靠的容器镜像运维

3 Kubernetes下的容器镜像运维

可控

基于
访问级别

基于
内容信任

基于
漏洞扫描

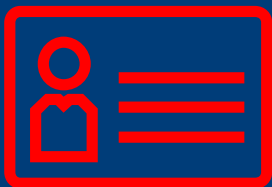
扫描控制



控制策略



公开项目



私有项目



漏洞级别 < 阈值



漏洞级别 ≥ 阈值



已签名



未签名 (阻止)



控制策略

- 设置项目访问级别：公有/私有
- 设置漏洞级别阈值：超过阈值的镜像无法下载
- 设置内容信任：未认证镜像无法下载
- 设置自动扫描：上传即扫描

控制策略



控制策略

< 项目

library

镜像仓库 成员 日志 复制 配置管理

项目仓库

☒ 公开

所有人都可访问公开的项目仓库。

部署安全

☐ 内容信任

仅允许部署通过认证的镜像。

☐ 阻止潜在漏洞镜像

阻止危害级别 较低 以上的镜像运行。

漏洞扫描

☐ 自动扫描镜像

当镜像上传后，自动进行扫描

保存

取消

- 设置项目访问级别：公有/私有
- 设置漏洞级别阈值：超过阈值的镜像无法下载
- 设置内容信任：未认证镜像无法下载
- 设置自动扫描：上传即扫描

议程

1 容器镜像运维基础

2 云原生容器镜像仓库

2.1 开源项目Harbor简介

2.2 安全的容器镜像管理

2.3 可控的容器镜像分发

2.4 可靠的容器镜像运维

3 Kubernetes下的容器镜像运维

可靠

镜像分发

HA部署



一致性：同一个 Dockerfile 始终生成同一个镜像？

```
FROM ubuntu
```

```
RUN apt-get install -y python
```

```
ADD app.jar /myapp/app.jar
```

基础镜像 **ubuntu:latest** 可能在不同构建时间会有差别

即使 **ubuntu:14.04** 也可能会有改变（补丁不同）

apt-get (curl, wget..) 无法保证安装同样的软件包

ADD 依赖构建时候的文件

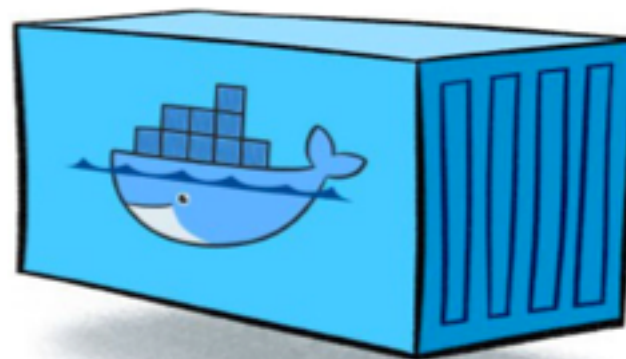
镜像一致性

容器镜像贯穿软件生命周期各个阶段

- 开发
- 测试
- 准生产
- 产线

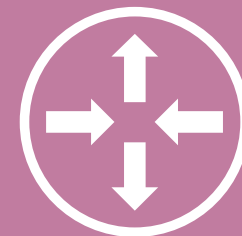
镜像一致性重要性

- 版本控制
- 问题追踪
- 审计



二进制格式

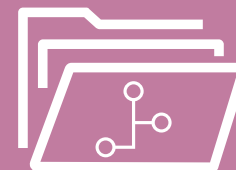
镜像分发



镜像分发

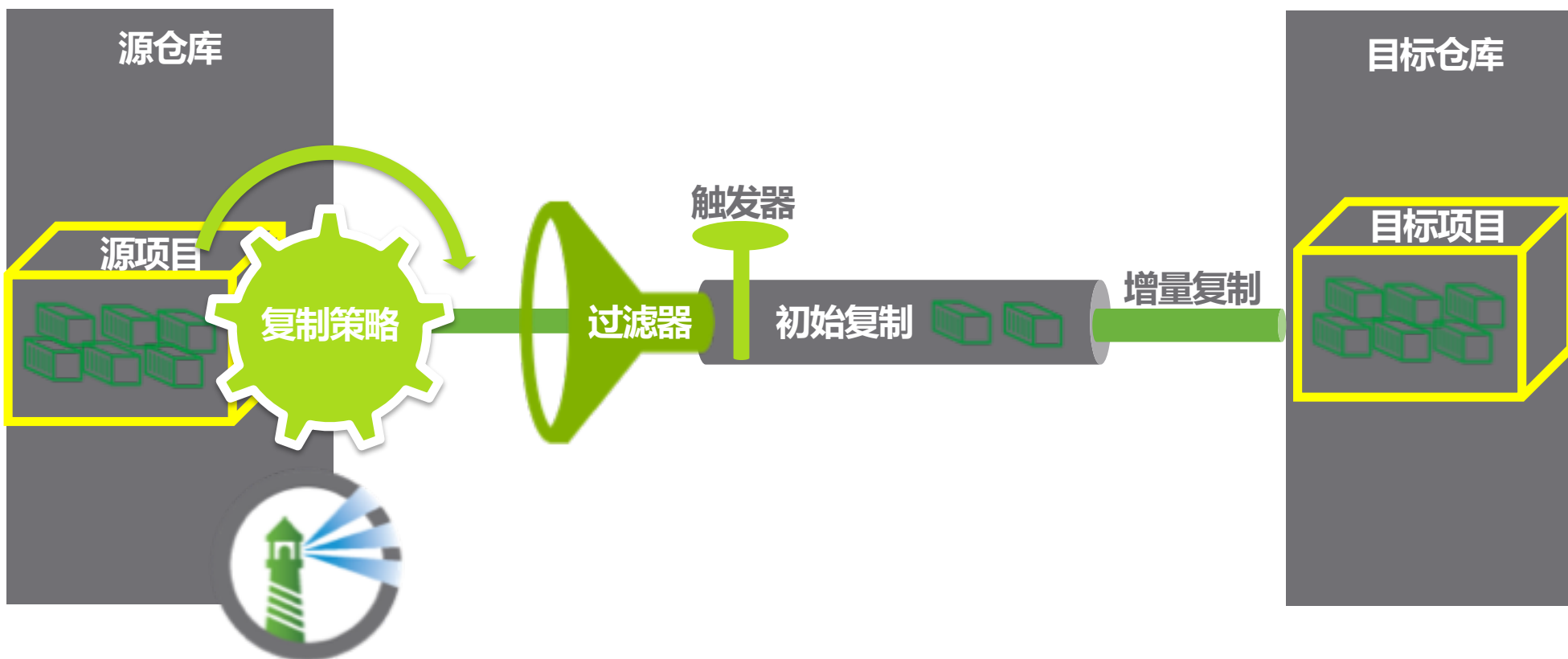
- 多点保持镜像一致
- 镜像备份与恢复
- 就近访问与下载
- 负载分担

镜像复制

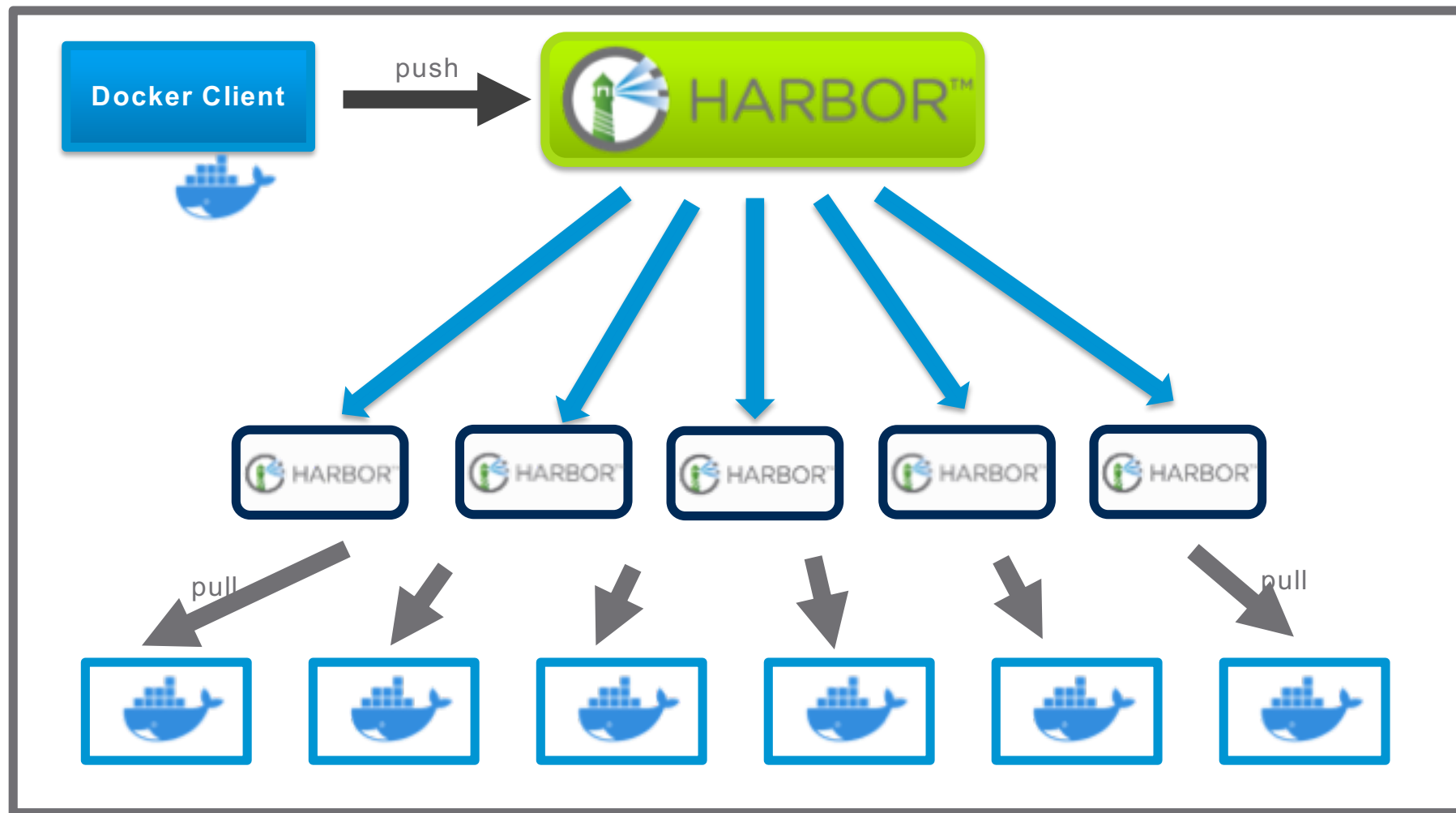


镜像复制

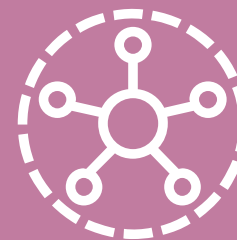
- 基于策略
- 面向项目
- 增量复制
- 支持过滤器
- 多种触发策略



镜像分发-负载分担



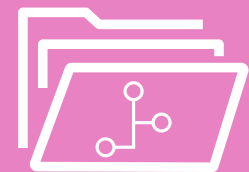
Master – Slave 模式



负载分担

- 容器镜像通常从registry分发
- 在大规模集群场景下，Registry 是镜像分发瓶颈
- 扩展 registry 服务
 - 多实例共享存储
 - 多实例不共享存储

复制策略管理



复制管理

项目

日志

系统管理

用户管理

仓库管理

复制管理

配置管理

复制管理

+ 新建规则

修改

删除

复制

名称	项目	描述	目标名	触发模式
111111	54321	111	test111	Scheduled
123123	library	12333333333	test111	Manual
123123123	library	-	test111	Manual

1 - 3 共计 3 条记录

复制任务

停止任务

高级检索

Q

C

名称	状态	操作	创建时间	更新时间	日志
library/hello-world	retrying	transfer	2018/1/29 下午6:01	2018/1/29 下午7:21	
library/hello-world	stopped	transfer	2018/1/29 下午4:53	2018/1/29 下午4:54	

1 - 2 共计 2 条记录

- 可在系统与项目级别管理复制策略
- 查看复制历史
- 查看复制日志

复制策略构建

< 复制管理

新建规则

名称* my rule

描述
Just a test case

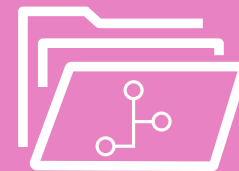
源 项目*

源镜像过滤器
repository test*
tag 1.*

目标*
khans3: http://10.112.122.203
userName:
password: *****
+ 新增

触发模式
手动
☐ 删除本地镜像时同时也删除远程的镜像。
☒ 立即复制现有的镜像。

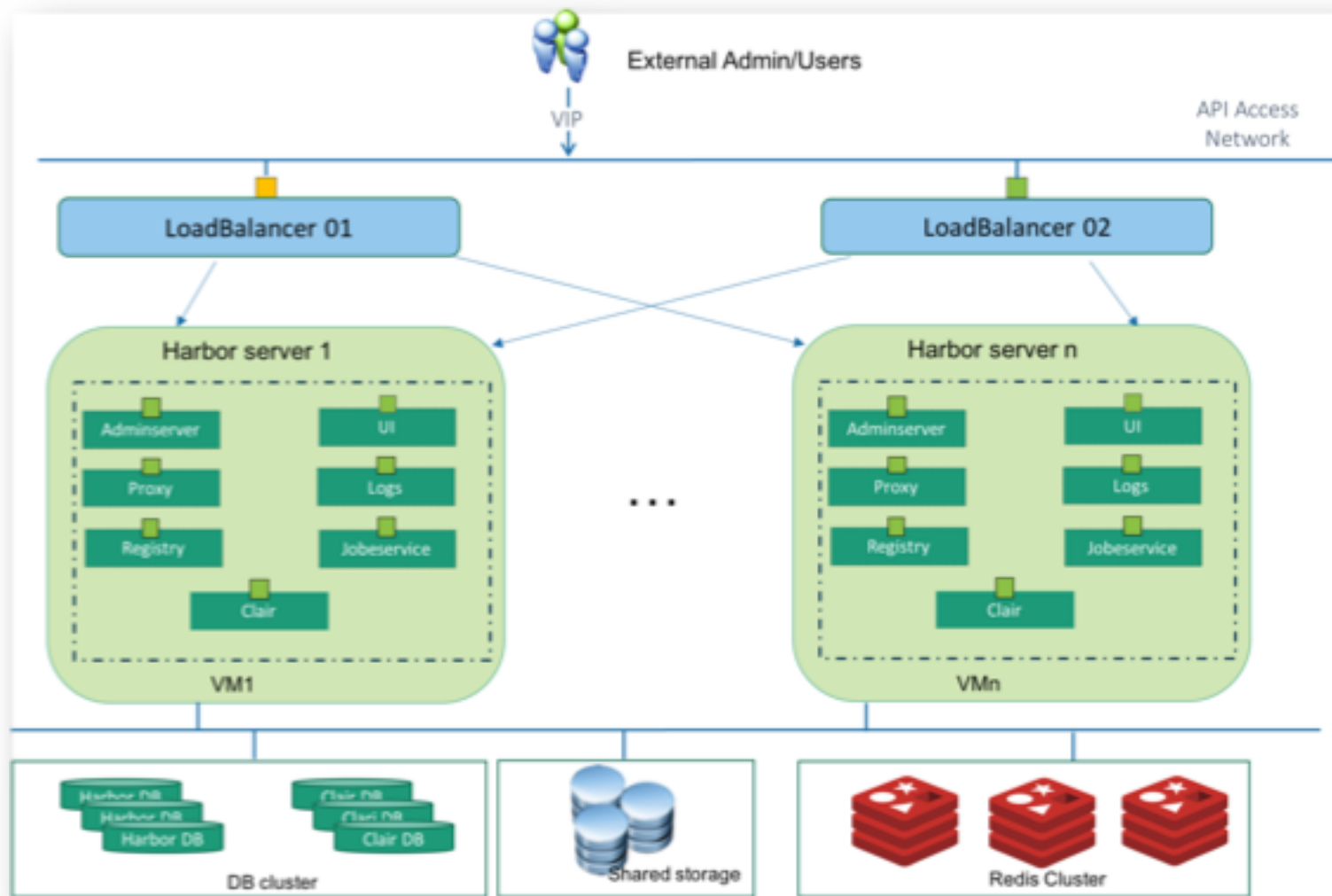
保存 取消



复制管理

- 支持Repo过滤
- 支持Tag过滤
- 多种触发模式
 - 手动
 - 定期
 - 事件

Harbor HA部署



HA部署

- Master-Master部署
- 基于特定平台方案

议程

1 容器镜像运维基础

2 云原生容器镜像仓库

2.1 开源项目Harbor简介

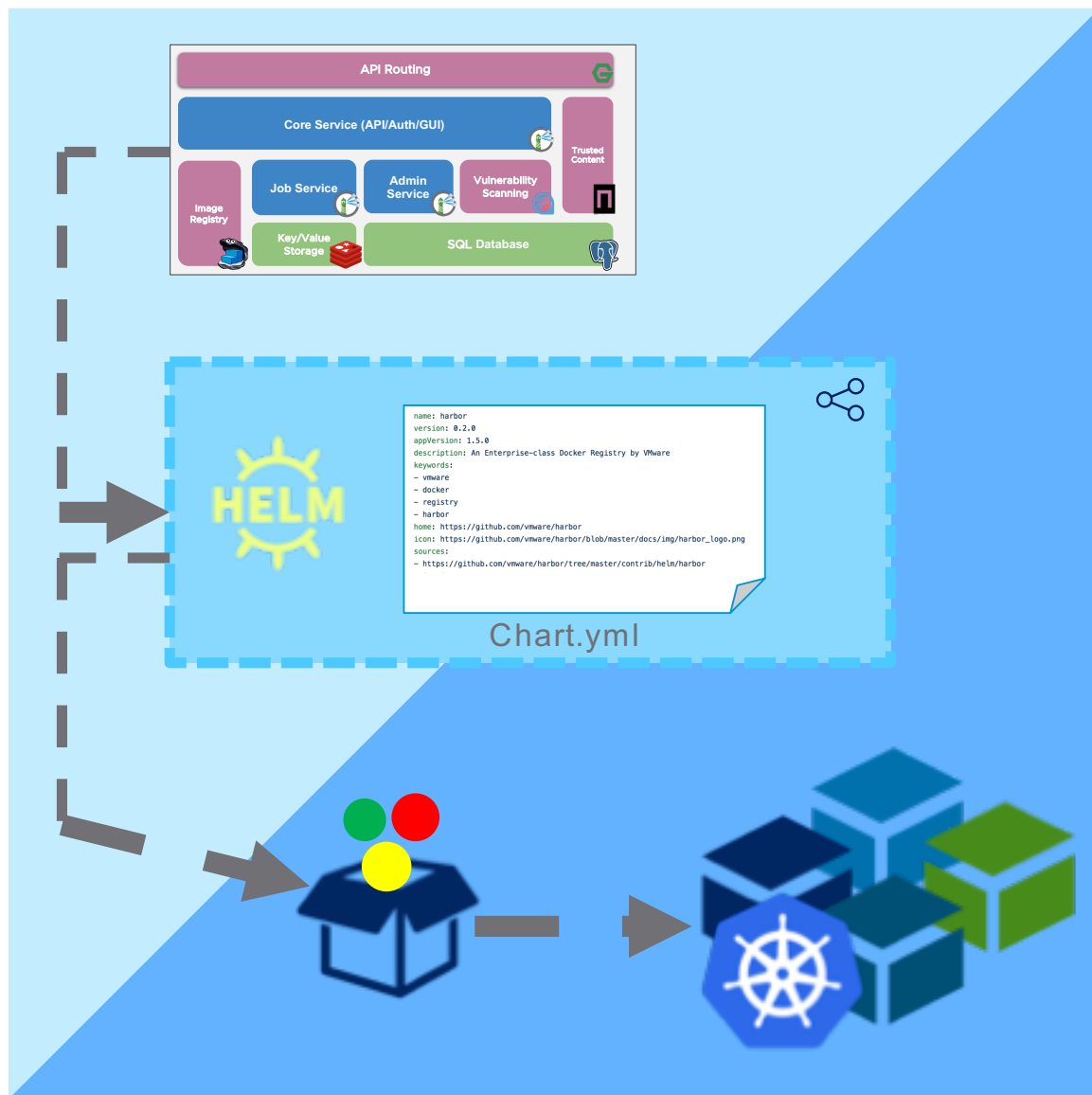
2.2 安全的容器镜像管理

2.3 可控的容器镜像分发

2.4 可靠的容器镜像运维

3 Kubernetes下的容器镜像运维

通过Helm Chart部署Harbor



拥抱Kubernetes



标准化和可重用



云原生应用模式



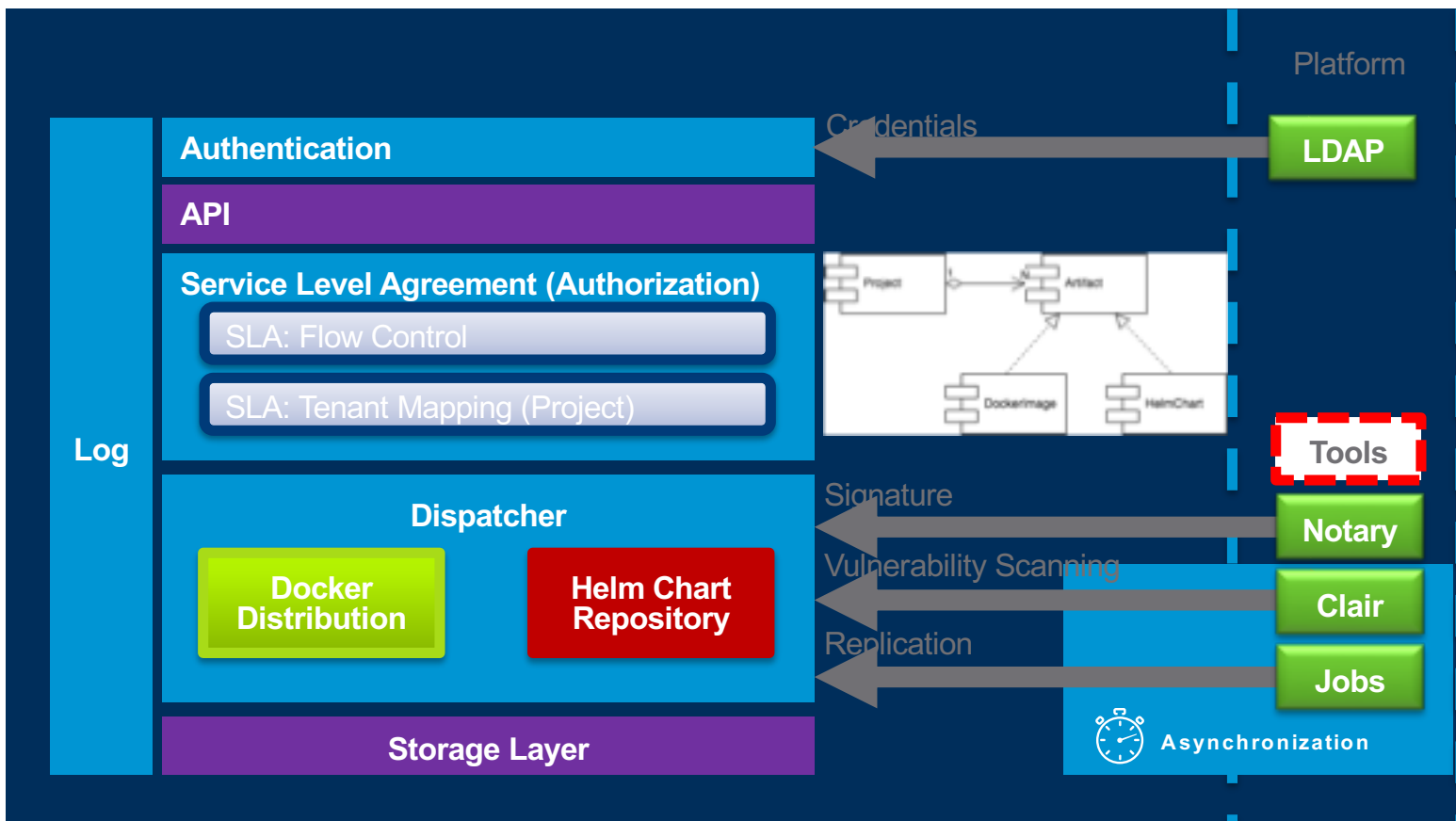
简化部署难度



提升可操作性

Helm Chart仓库支持🕒


- Updated Components
- Planning Components




在容器镜像管理基础之上，Harbor将实现Helm Chart仓库的能力以支持对Chart的管理，并将通过创新方式打通Chart与镜像管理的通道。

- Helm Chart事实上的Kubernetes包管理标准
- 企业级应用与服务编排部署的有效模式
- 简化难度，大幅提升生产效率
- 与容器镜像紧密关联，利用Harbor优势
- 拥抱云原生与Kubernetes


Helm Chart仓库支持 (续)🕒




MANAGE




DOWNLOAD




UPLOAD




DELETE



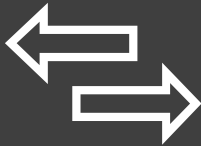
TRUST



VERIFY&SCAN



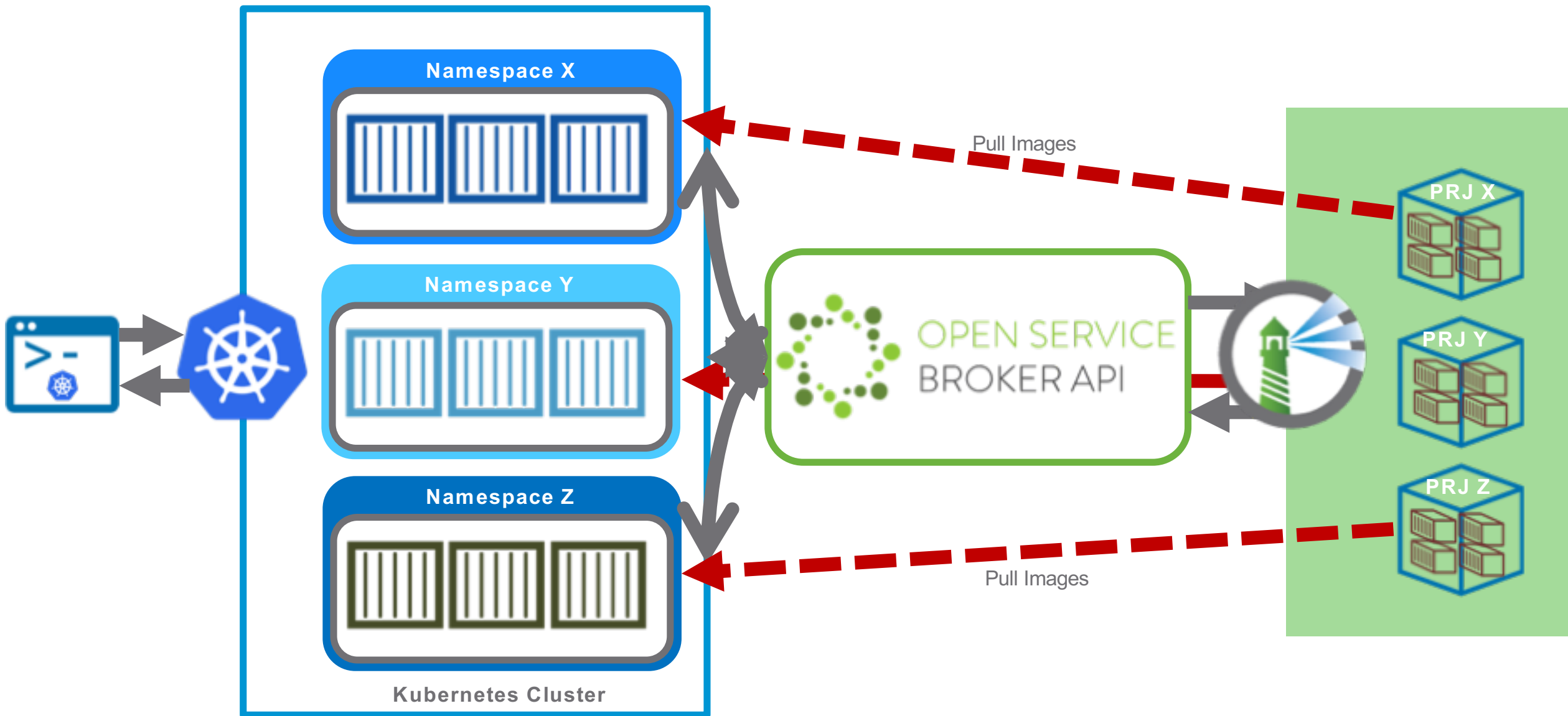
REPLICATE



EXPORT/IMPORT



与Kubernetes更多集成





网站：<https://github.com/vmware/harbor>

Twitter: @Project_harbor

Email组：（ 加入方式参见GitHub ）

harbor-users@googlegroups.com
harbor-dev@googlegroups.com

欢迎参加定期的社区会议（ 视频会议 ）

需要加入Harbor群的朋友，请联系演讲人