

Secure and Efficient Container Image Management for Kubernetes via Harbor

在Kubernetes上采用Harbor高效安全管理镜像

张海宁，VMware中国研发先进技术中心技术总监

杜 宁，Caicloud解决方案与交付副总裁

自我介绍

- VMware中国研发先进技术中心首席架构师、技术总监
- 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之一
- 多年全栈工程师
- 《区块链技术指南》、《软件定义存储》作者之一



公众号：亨利笔记



《区块链技术指南》



《软件定义存储》

主办：



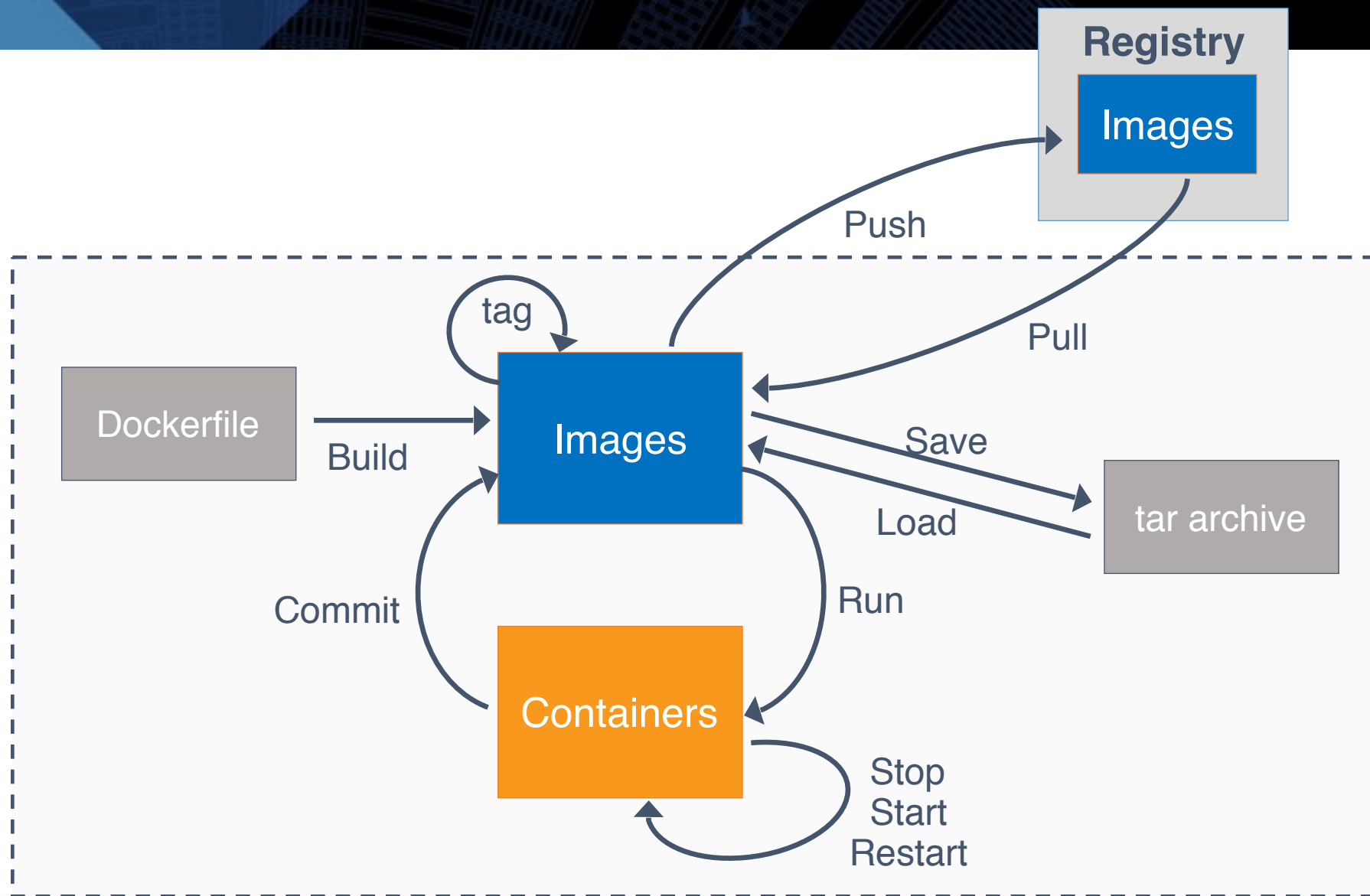
Agenda

- 1 Container Image Basics
- 2 Project Harbor Introduction
- 3 Consistency of Images
- 4 Security
- 5 Image Distribution
- 6 Integration with Kubernetes

Agenda

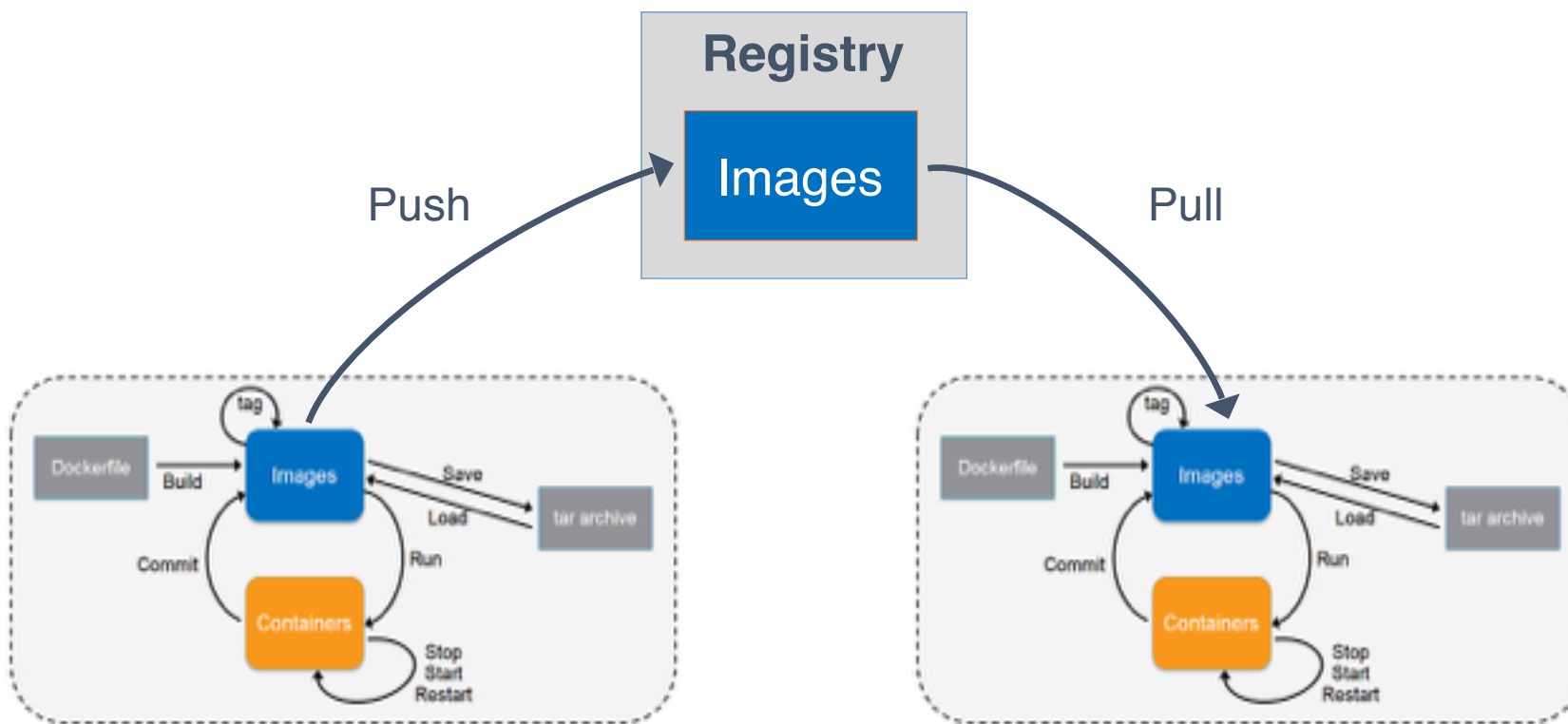
- 1 Container Image Basics**
- 2 Project Harbor Introduction
- 3 Consistency of Images
- 4 Security
- 5 Image Distribution
- 6 Integration with Kubernetes

Lifecycle of Containers and Images



Registry - Key Component to Manage Images

- Repository for storing images
- Intermediary for shipping and distributing images
- Ideal for access control and other image management



Agenda

1 Container Image Basics

2 Project Harbor Introduction

3 Consistency of Images

4 Security

5 Image Distribution

6 Integration with Kubernetes

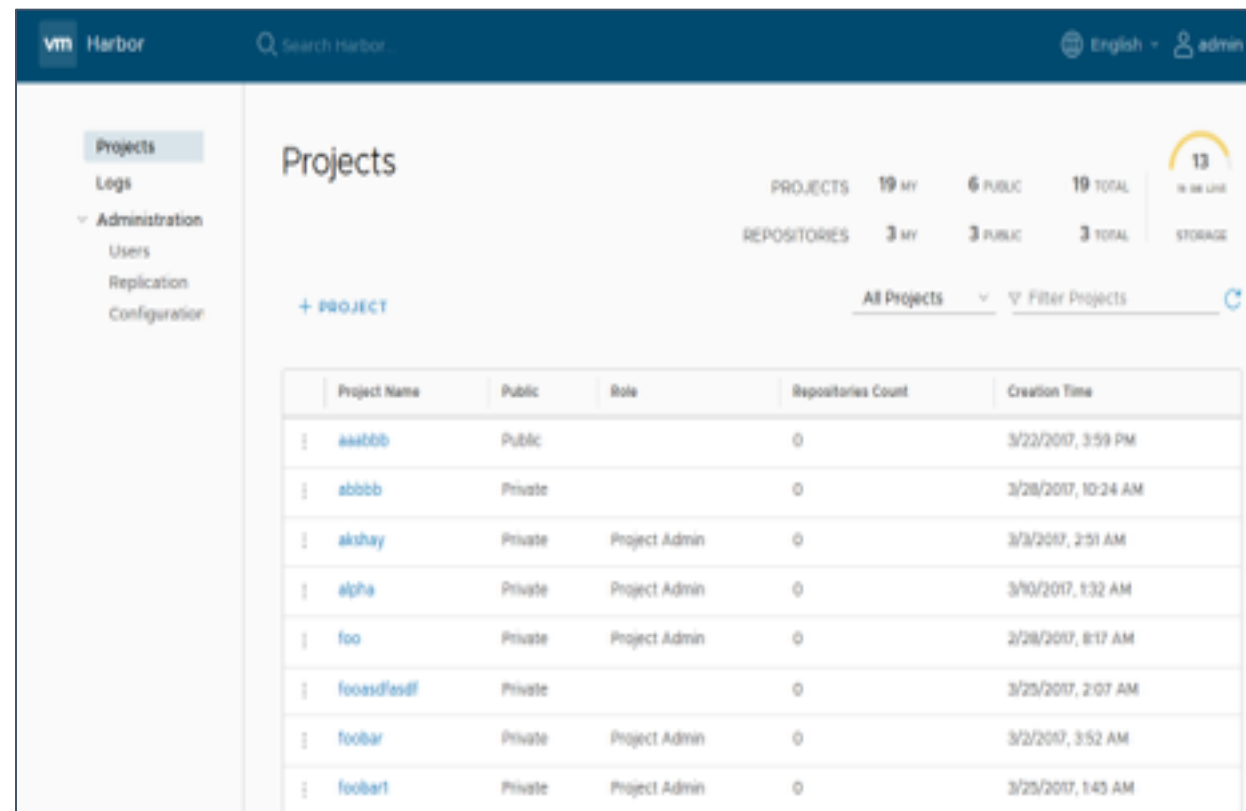
Project Harbor



- An open source enterprise-class registry server.
- Initiated by VMware China, adopted by users worldwide.
- Apache 2 license.
- <https://github.com/vmware/harbor/>

Key Features

- User management & access control
 - RBAC: admin, developer, guest
 - AD/LDAP integration
- Policy based image replication
- Vulnerability Scanning
- Notary
- Integrated with Kubernetes
- Web UI
- Audit and logs
- Restful API for integration
- Lightweight and easy deployment



Project Name	Public	Role	Repositories Count	Creation Time
aaabbb	Public		0	3/22/2017, 3:59 PM
abbbb	Private		0	3/28/2017, 10:24 AM
akshay	Private	Project Admin	0	3/3/2017, 2:51 AM
alpha	Private	Project Admin	0	3/10/2017, 1:32 AM
foo	Private	Project Admin	0	3/28/2017, 8:17 AM
fooadfesdf	Private		0	3/25/2017, 2:07 AM
foobar	Private	Project Admin	0	3/2/2017, 3:52 AM
foobar1	Private	Project Admin	0	3/25/2017, 1:43 AM

Harbor Momentum

20K+

Downloads

2800+

Stars

1000+

Users

800+

Forks

58

Contributors

6

Partners

Harbor Users and Partners



Harbor Architecture

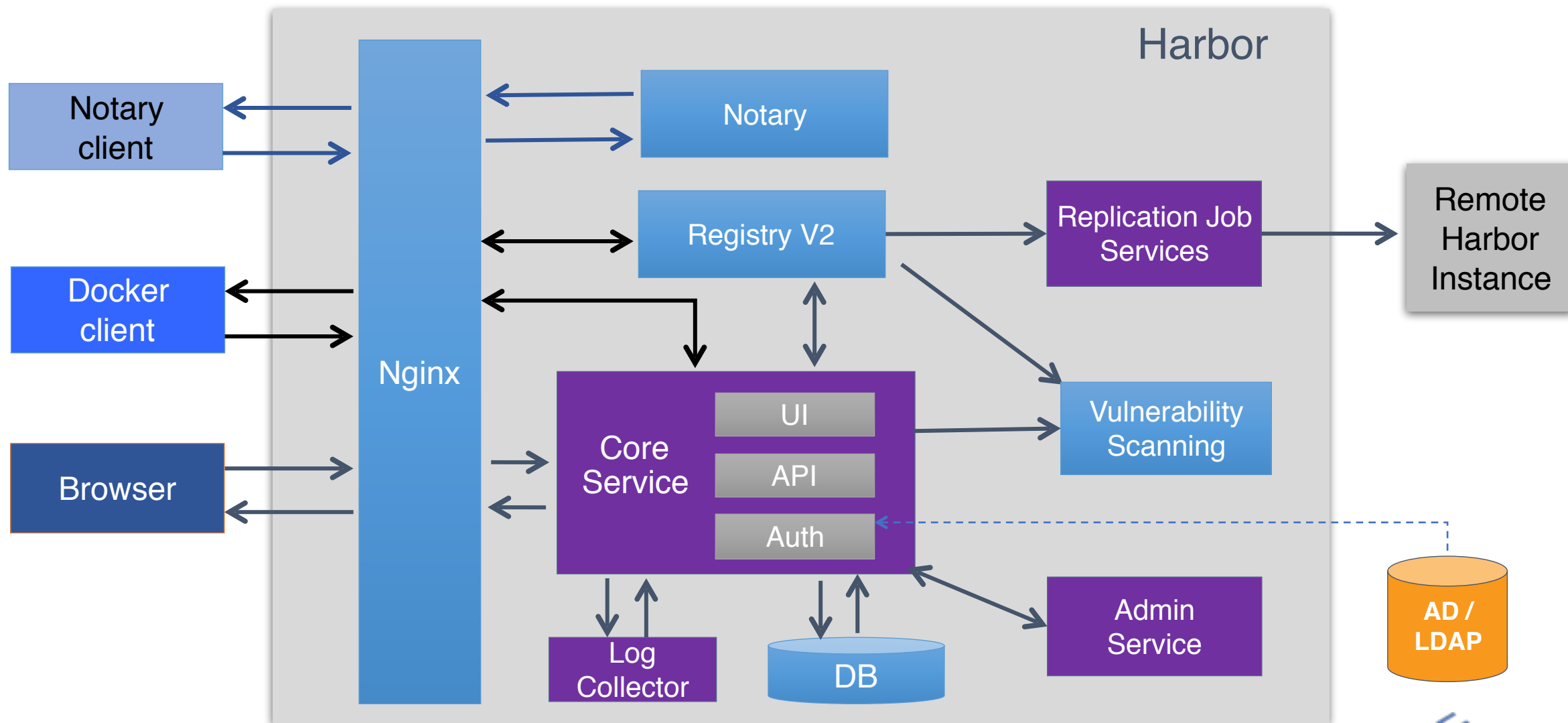
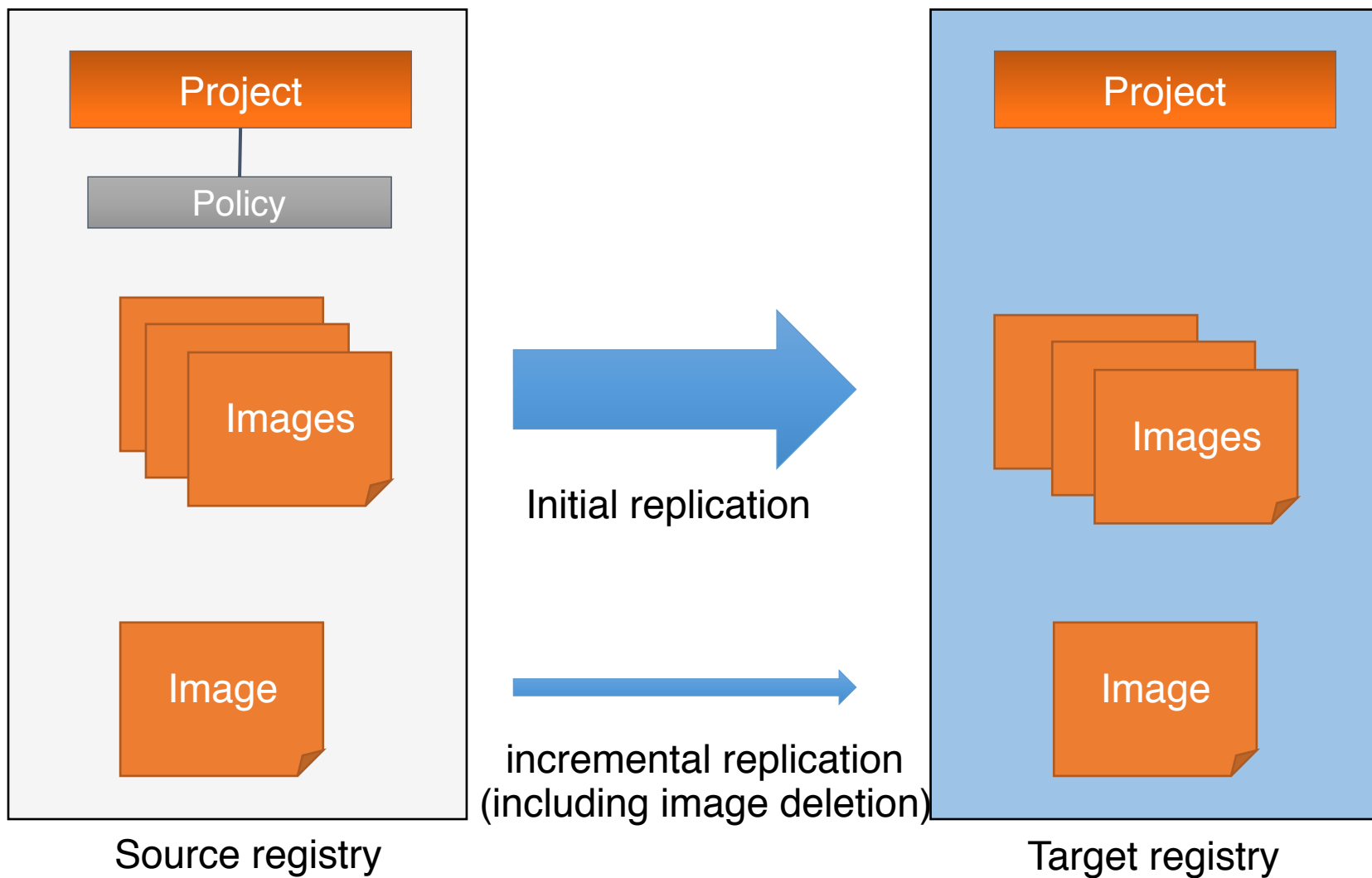


Image Replication (Synchronization)



Agenda

- 1 Container Image Basics
- 2 Project Harbor Introduction
- 3 Consistency of Images**
- 4 Security
- 5 Image Distribution
- 6 Integration with Kubernetes

Consistency of Container Images

- Container images are used throughout the life cycle of software development
 - Dev
 - Test
 - Staging
 - Production
- Consistency must be maintained
 - Version control
 - Issue tracking
 - Troubleshooting
 - Auditing

Same Dockerfile Always Builds Same Image?

Example:

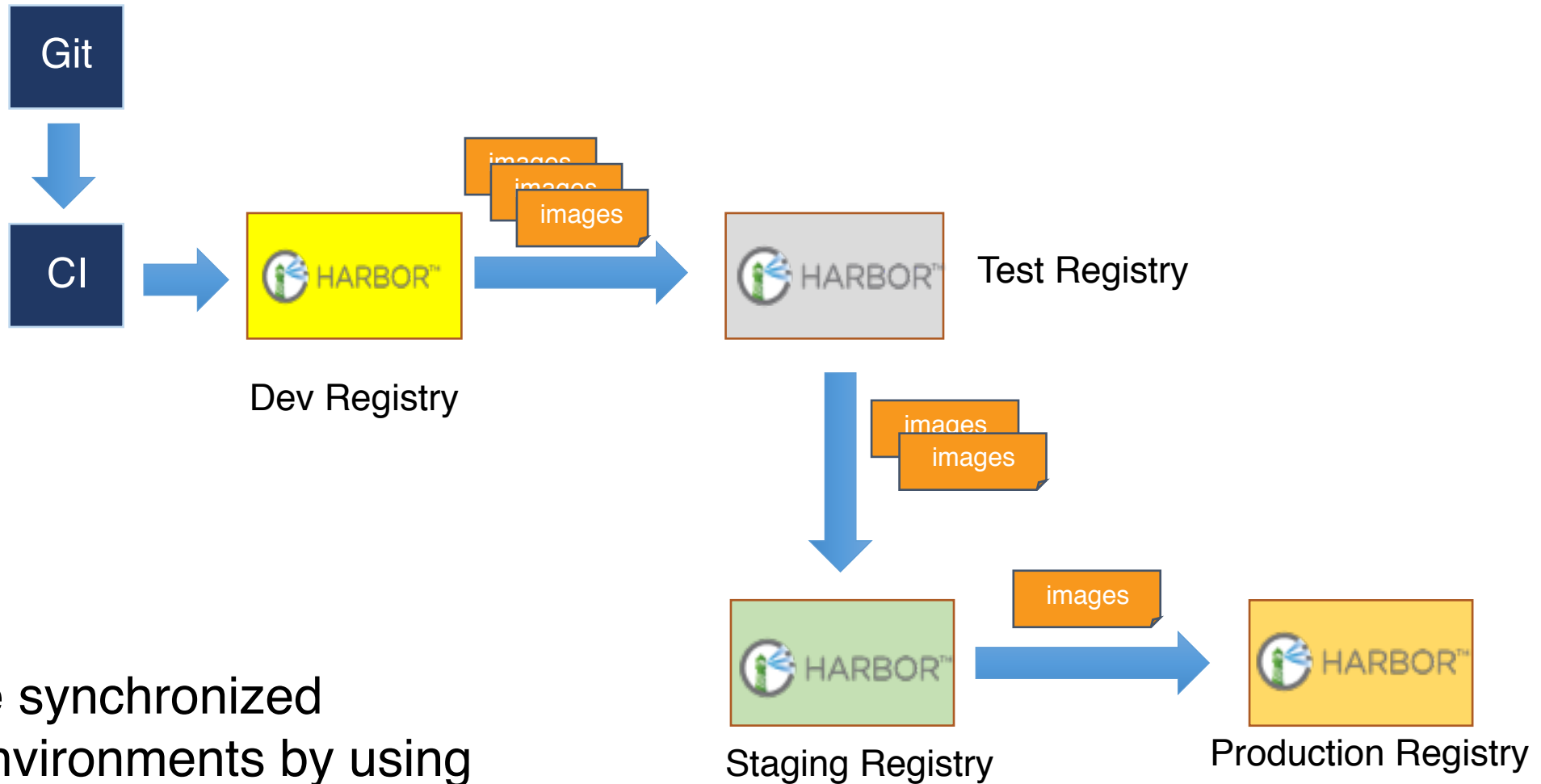
```
FROM ubuntu
```

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

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

- Base image `ubuntu:latest` could be changed between builds
- `ubuntu:16.04` could also be changed due to patching
- `apt-get (curl, wget..)` cannot guarantee always to install the same packages
- ADD depends on the build time environment to add files

Shipping Images in Binary Format for Consistency



Images are synchronized between environments by using Harbor registry.

Agenda

1 Container Image Basics

2 Project Harbor Introduction

3 Consistency of Images

4 Security

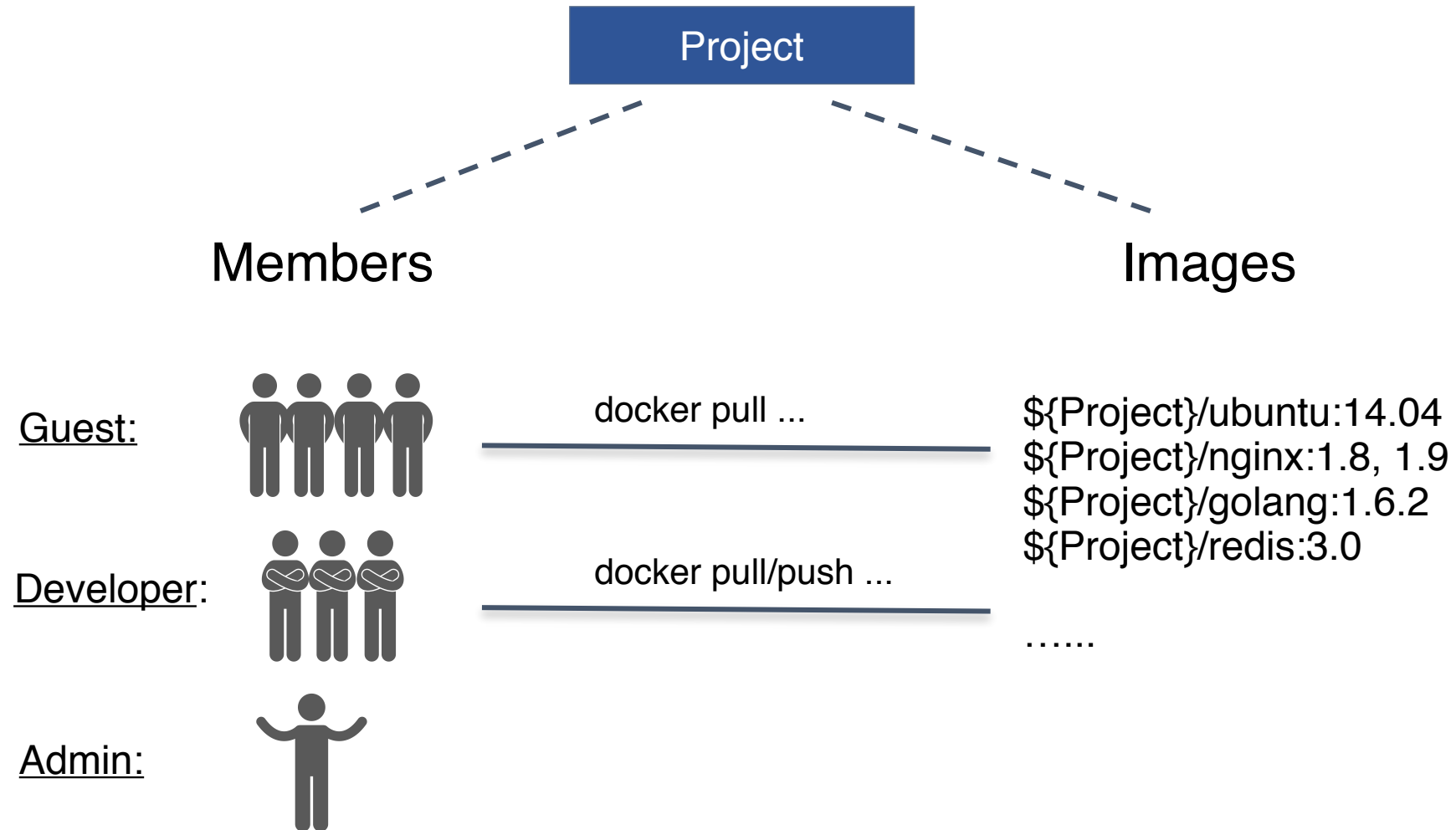
5 Image Distribution

6 Integration with Kubernetes

Access Control to Images

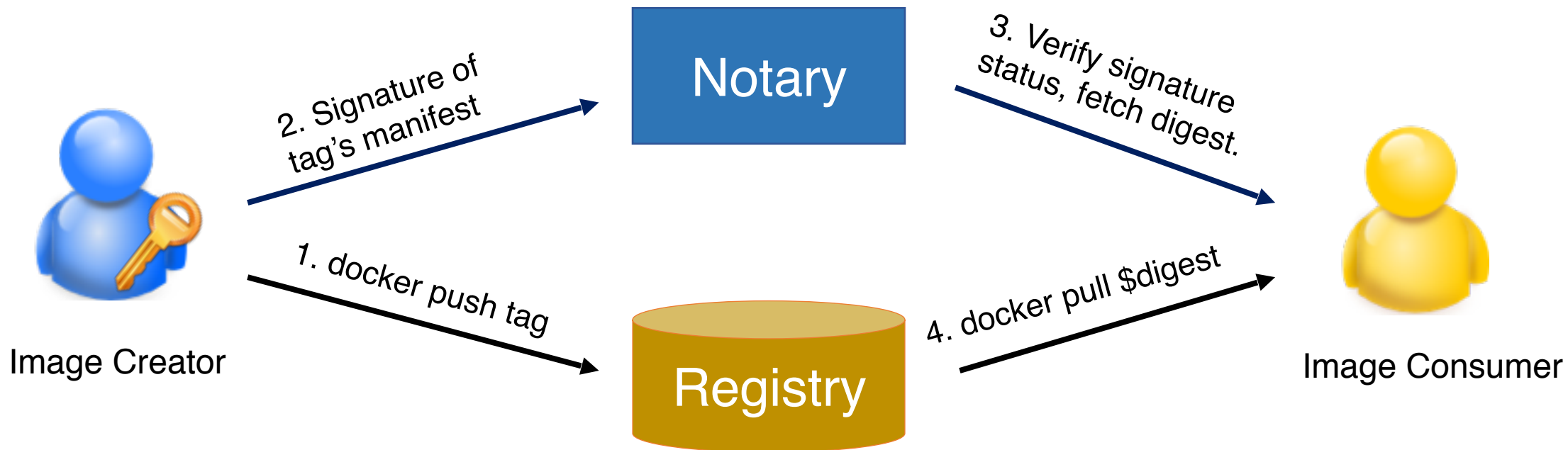
- Organizations often keep images within their own organizations
 - Intellectual property stays in organization
 - Efficiency: LAN vs WAN
- People with different roles should have different access
 - Developer – Read/Write
 - Tester – Read Only
- Different rules should be enforced in different environments
 - Dev/test env – many people can access
 - Production – a limited number of people can access
- Can be integrated with internal user management system
 - LDAP/Active Directory

Example: Role Based Access Control in Harbor



- Enable **content trust** by installing **Notary service**
 - Image is signed by publisher's private key during pushing
 - Image is pulled using digest
- Perform vulnerability scanning
 - Prevent images with vulnerabilities from being pulled
 - Regular scanning based on updated vulnerability database

Content Trust for Image Provenance



- **Static analysis** of vulnerability by inspecting filesystem of container image and indexing features in database.
- **Rescanning** is needed only and only if new detectors are added.
- Update vulnerability data regularly
 - Debian Security Bug Tracker
 - Ubuntu CVE Tracker
 - Red Hat Security Data
 - Oracle Linux Security Data
 - Alpine SecDB

- **Set vulnerability threshold**
- **Prevent images from being pulled** if they exceed threshold
- **Periodic scanning** based on updated vulnerability database

Project Repositories

PUSH IMAGE  

	Name	Tags
>	default-project/redis	1
▼	default-project/ubuntu	1

	Tag	Pull Command	vulnerability	
14.04	docker pull 10.160.247.138/default-project/ubuntu:14.04		✓	1/29/2015, 2:37 AM

1 - 1 of 1 items

>	default-project/demo-busybox	2	15
---	------------------------------	---	----

1 - 3 of 3 items

36 of 126 packages have known vulnerabilities.

- 6 high
- 22 medium
- 8 low
- 90 none

Scan completed time: 08/09/2017 23:03:19

Agenda

1 Container Image Basics

2 Project Harbor Introduction

3 Consistency of Images

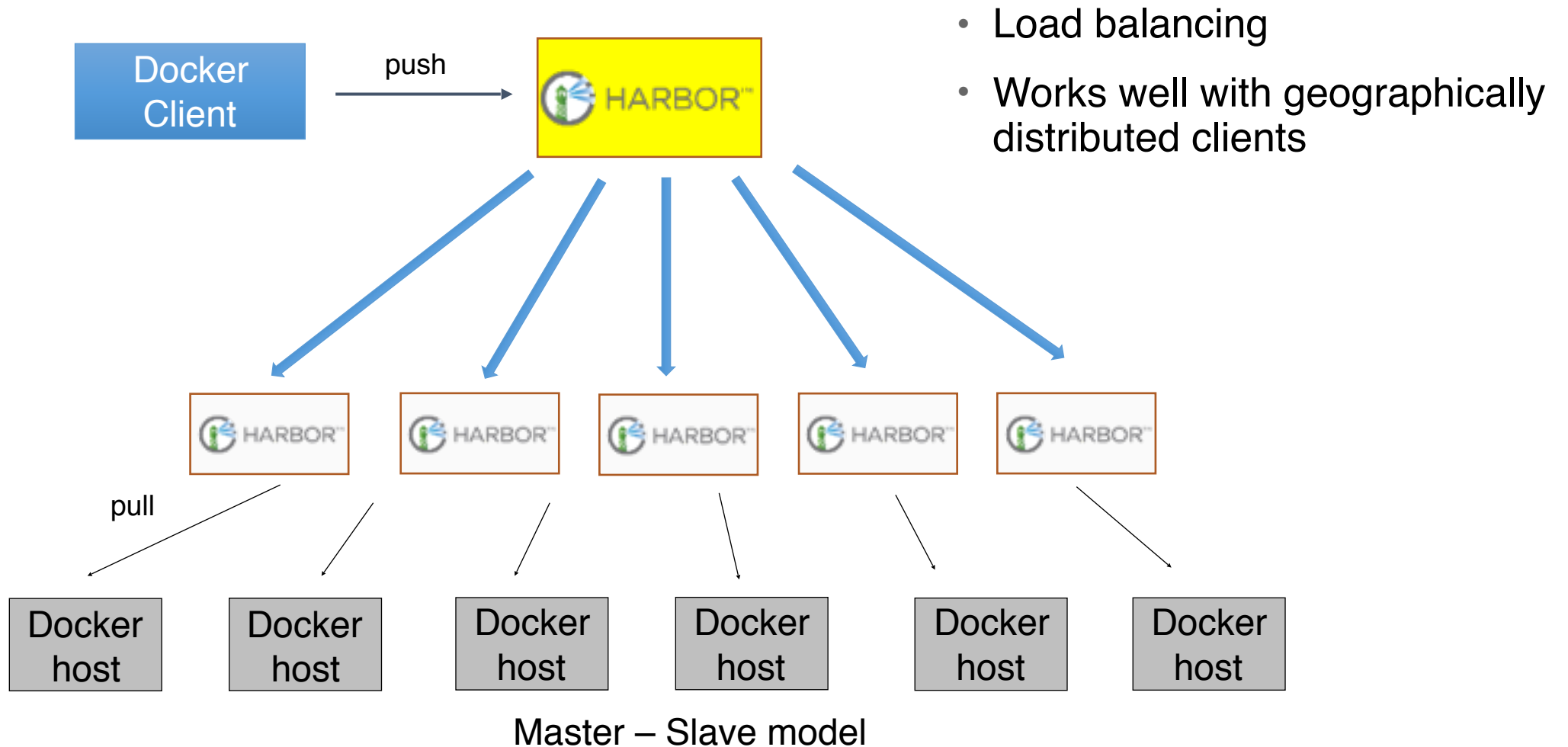
4 Security

5 Image Distribution

6 Integration with Kubernetes

- Container images are usually distributed from a registry.
- Registry becomes the bottleneck for a large cluster of nodes
 - I/O
 - Network
- Scaling out an registry server
 - Multiple instances of registry sharing same storage
 - Multiple instances of independent registry sharing no storage

Image Distribution via Master-Slave Replication



Agenda

1 Container Image Basics

2 Project Harbor Introduction

3 Consistency of Images

4 Security

5 Image Distribution

6 Integration with Kubernetes

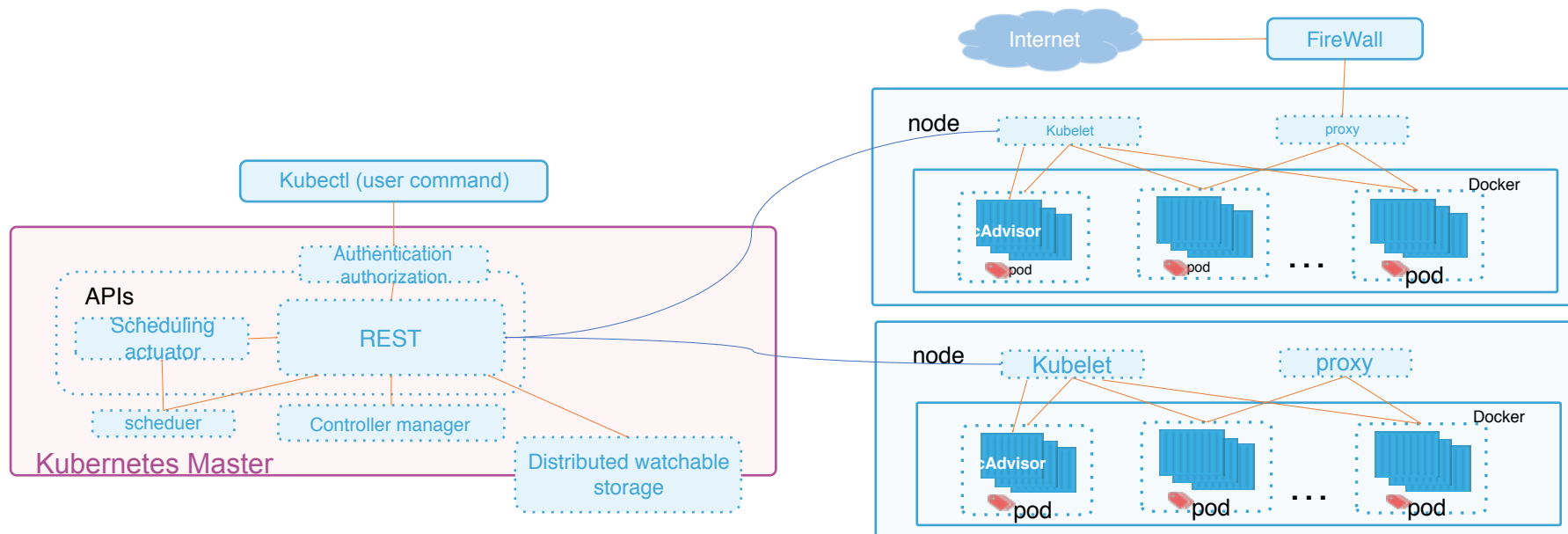
杜 宁

- 才云科技首席解决方案架构师
- Email: tony@caicloud.io
- Cell: 18610255655



Harbor Integration with Kubernetes

- Harbor作为Kubernetes集群的镜像仓库



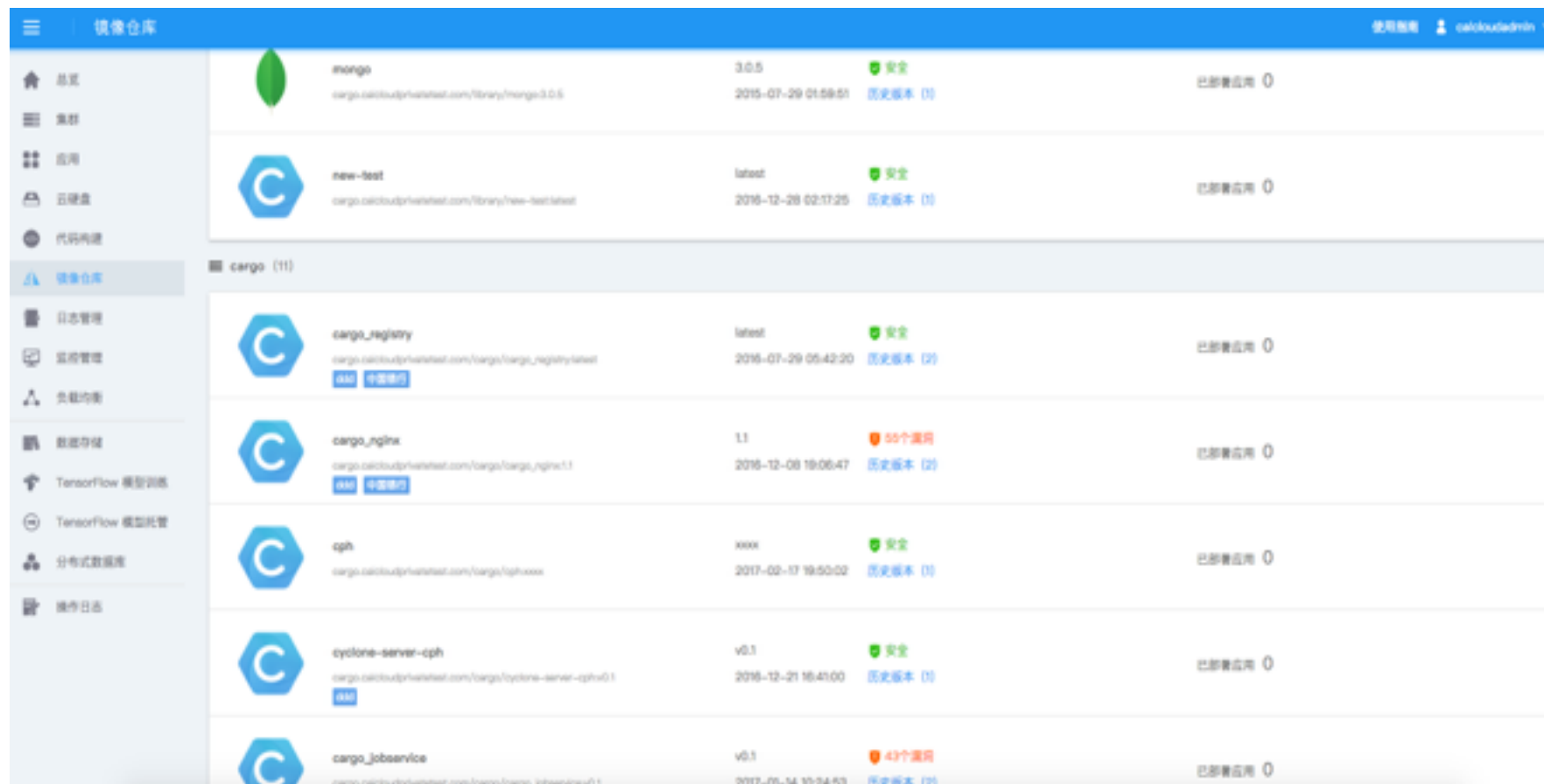
HARBOR™

主办:



Harbor Integration with Kubernetes




- 界面集成
- 租户
- 权限



Harbor Integration with Kubernetes

- 漏洞检测

cargo/cargo_nginx 1.1 漏洞情况	
CVE-2016-2781	Medium
nonpriv session can escape to the parent session by using the TIOCSTI ioctl	
CVE-2016-9069	Medium
heap-use-after-free in nsNode::ReplaceOnwriteBefore	
CVE-2015-8239	Medium
race condition checking digests/checksums in subers	
CVE-2015-6865	Medium
The file_check_mem function in funcs.c in file before 5.23, as used in the Fileinfo component in PHP before 5.5.34, 5.6.x before 5.6.20, and 7.x before 7.0.5, mishandles continuation-level jumps, which allows context-dependent attackers to cause a denial of service (buffer overflow and application crash) or possibly execute arbitrary code via a crafted magic file.	
CVE-2016-7076	Medium
noexec bypass via wordexp()	
CVE-2016-1248	Medium
vim before patch 8.0.0056 does not properly validate values for the 'filetype', 'syntax' and 'keymap' options, which may result in the execution of arbitrary code if a file with a	

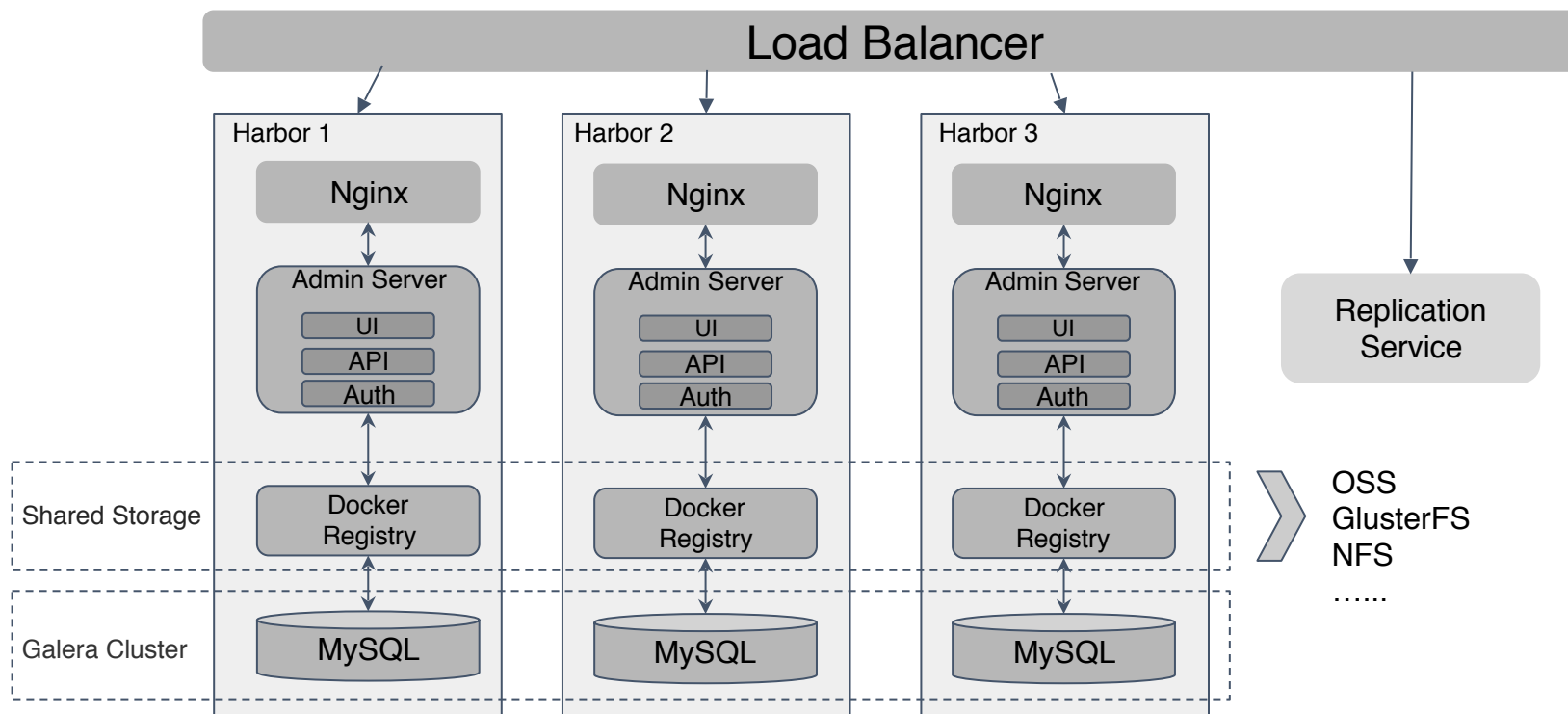
	cargo_ui cargo.caicloudprivatetest.com/cargo/cargo_ui:v1.0-alpha 客户1 标签2	v1.0-alpha 2017-03-09 00:40:31	安全 历史版本 (2)
	cargo_mysql cargo.caicloudprivatetest.com/cargo/cargo_mysql:v1.0-alpha 标签2	v1.0-alpha 2017-02-22 09:22:47	36个漏洞 历史版本 (2)
	cargo_log cargo.caicloudprivatetest.com/cargo/cargo_log:v0.1	v0.1 2017-01-06 10:49:39	56个漏洞 历史版本 (1)

主办:



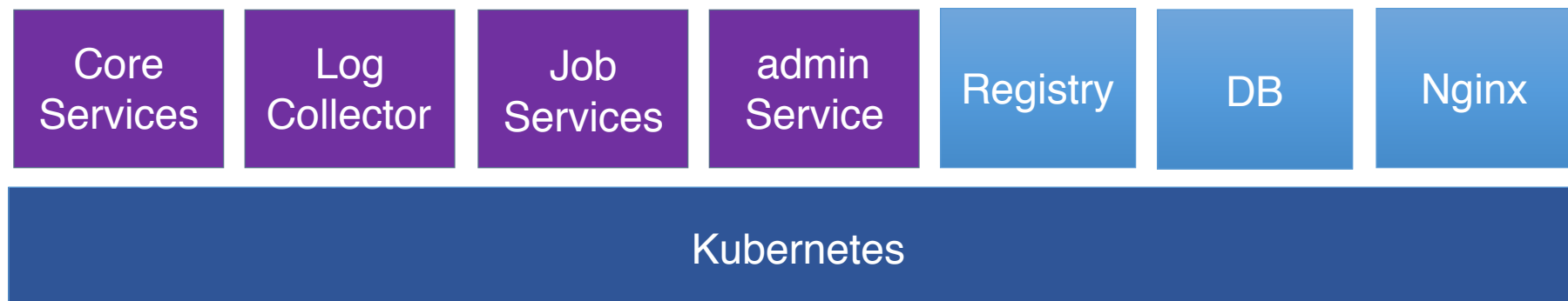
Harbor Integration with Kubernetes

- 高可用性



Harbor Integration with Kubernetes

- 以Pod的形式部署到Kubernetes上





Harbor用户微信群

Thank you for your time

主办：

