VMware {code} Experience Training

Open Source Projects for Enterprise Cloud Native

Ben Corrie, Senior Staff Engineer
Henry Zhang, Chief Architect, R&D, VMware China
Patrick Daigle, Sr. Tech Marketing Architect, VIC
Steven Zou, Staff Engineer, Project Harbor

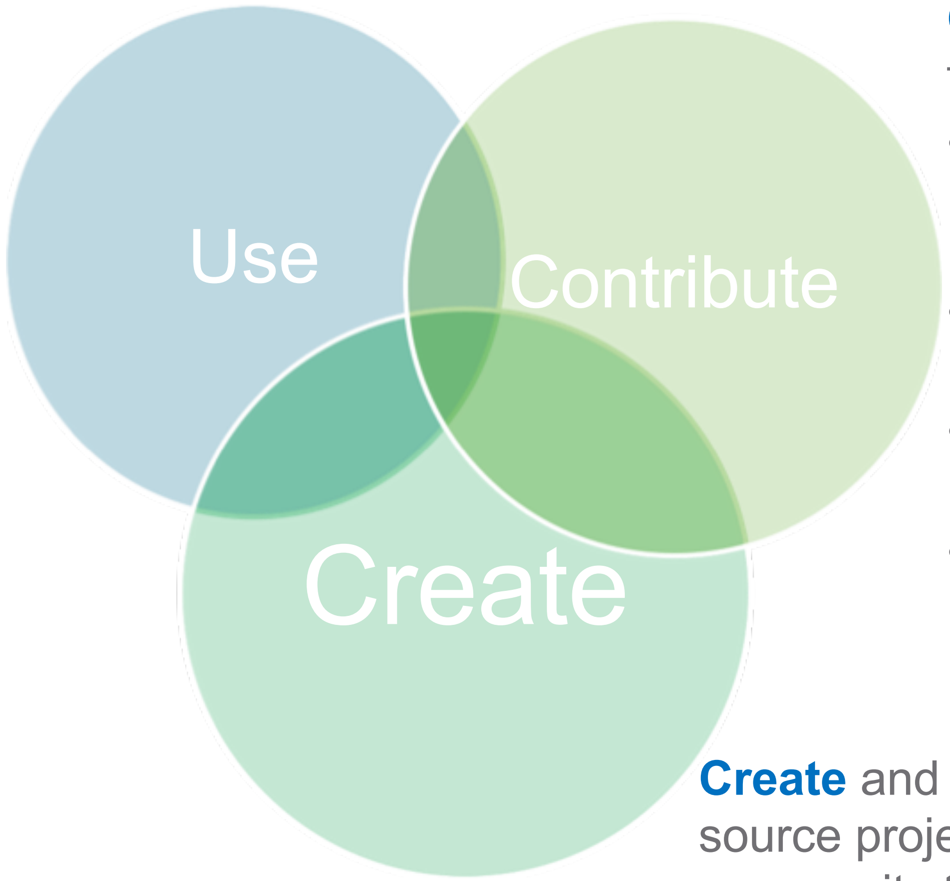
Disclaimer

- This presentation may contain product features that are currently under development.
- This overview of new technology represents no commitment from VMware to deliver these features in any generally available product.
- Features are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.
- Technical feasibility and market demand will affect final delivery.
- Pricing and packaging for any new technologies or features discussed or presented have not been determined.

VMware's Participation in Open Source Community

Use open source code within the enterprise:

- For implementation of industry standards
- To improve software usability
- To accelerate our software development processes



Contribute to upstream projects to add features or fix bugs:

- Linux kernel drivers for our virtual hardware, but also work on key kernel infrastructure
- Open Stack improvement for network, storage, stability
- Kubernetes integration with vSphere as compute and storage provider
- Many smaller contributions and bug fixes to a wide variety of projects

Create and release a new open source project and build new community to support it

- Variety of project – from utilities and glue code to significant, standard-setting projects

vmware®

Leadership in Open Source Cloud Native Projects

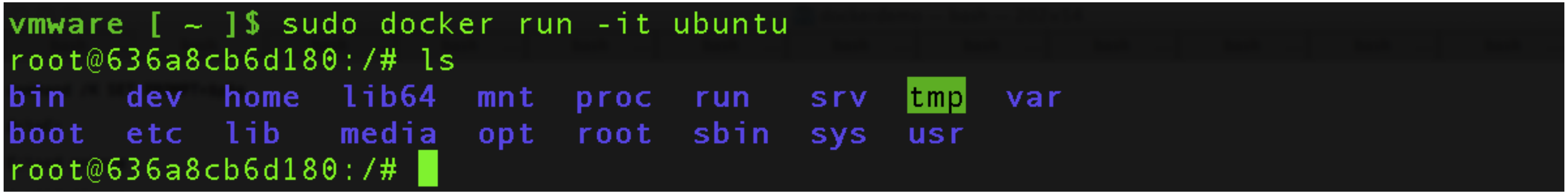
vSphere Integrated Containers Engine

HARBOR™

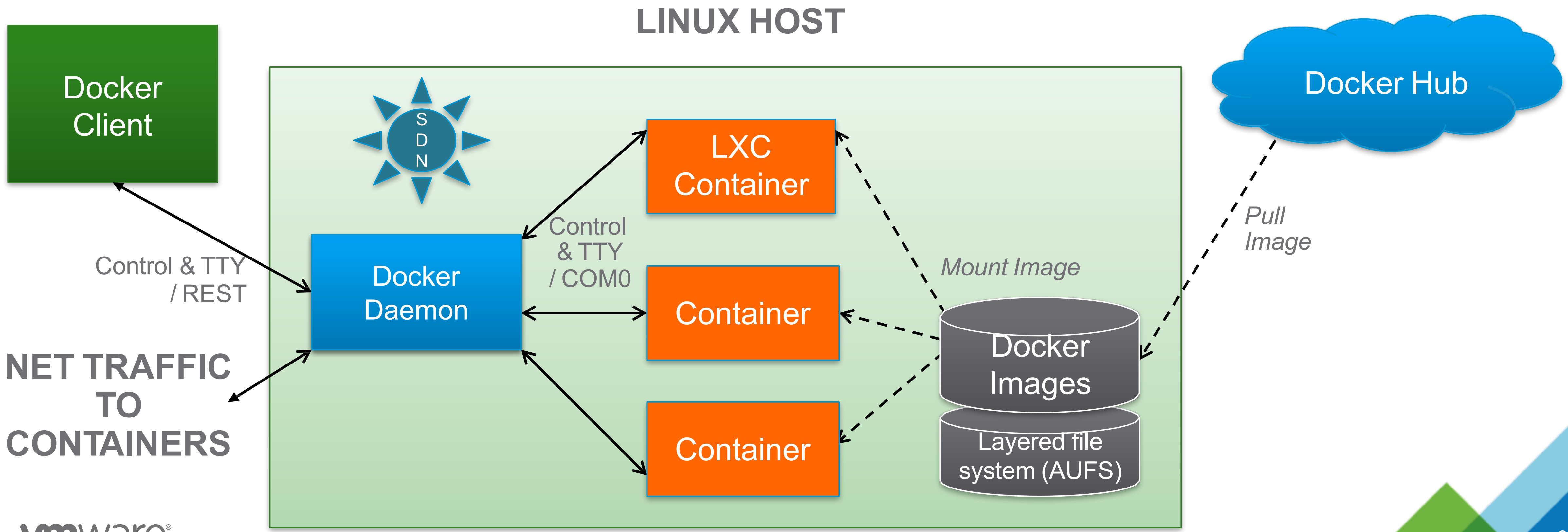
ADMIRAL™

vSphere Integrated Containers

Docker 101



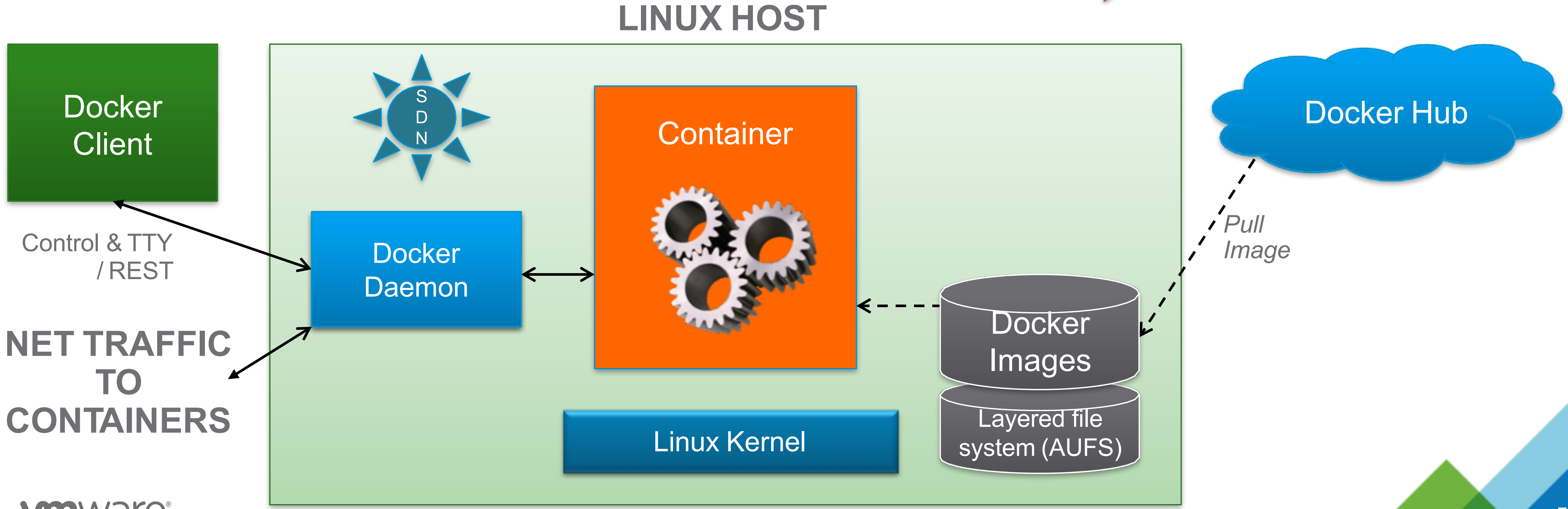
```
vmware [ ~ ]$ sudo docker run -it ubuntu
root@636a8cb6d180:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@636a8cb6d180:/#
```



What is a Container?

1. An executable process
2. Resource constraints / private namespace
3. Binary dependencies: Application, runtime, OS
4. A shared Linux kernel for running the executable
5. Ephemeral and persistent storage layers

STATELESS
PORTABLE
FAST
SECURE

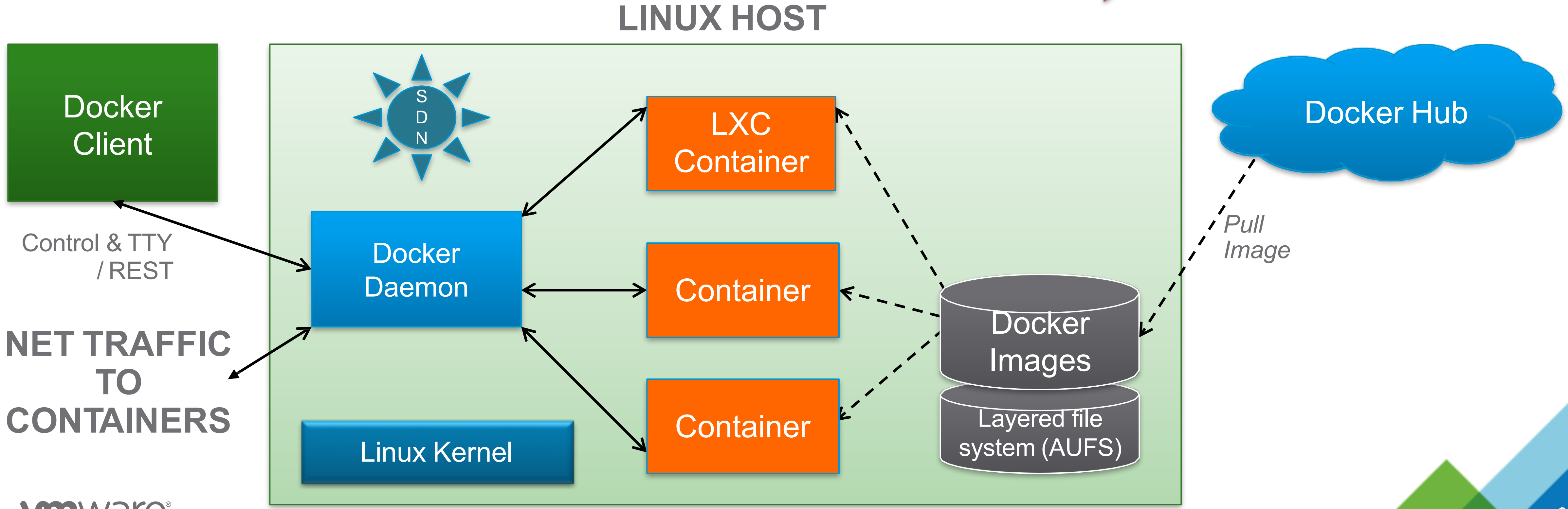


vmware

What is a Container Host?

1. Control plane & lifecycle management for containers
2. Resource scheduling and a container abstraction
3. Infrastructure abstractions: Storage, networking etc
4. A single Linux kernel manages everything
5. A static size and a resource reservation when virtual

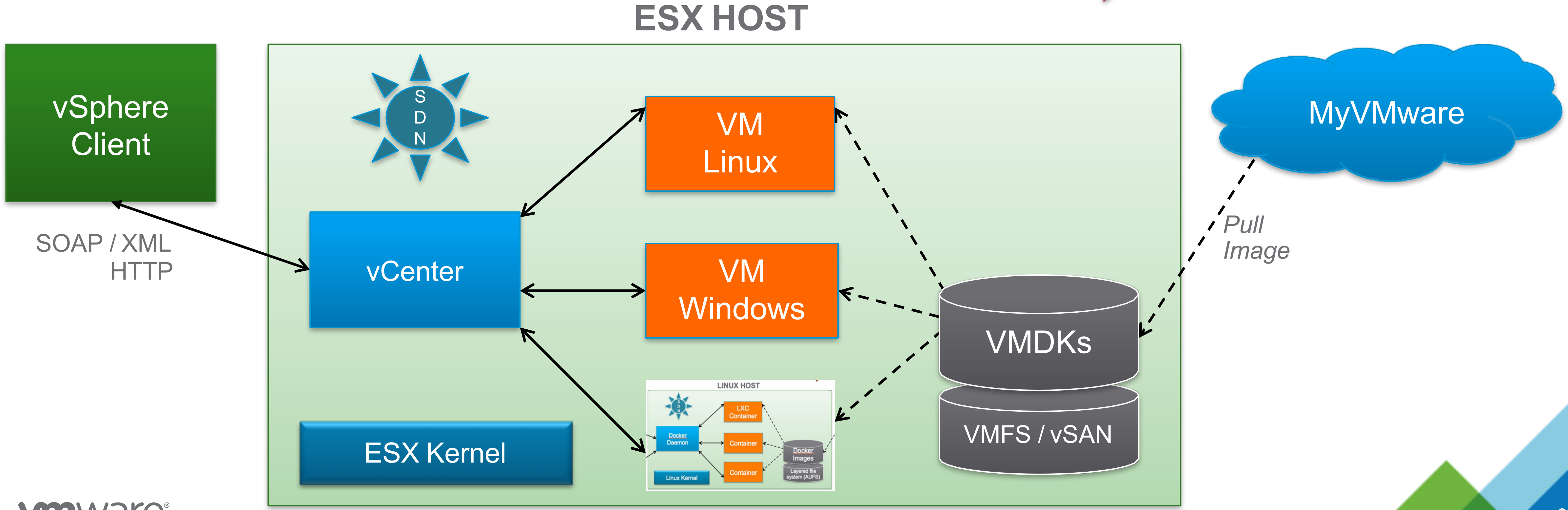
STATEFUL
LONG-RUNNING
SINGLE USER
SINGLE USE



vmware®

What is a Hypervisor?

1. Control plane & lifecycle management for VMs
2. Resource scheduling and a VM abstraction
3. Infrastructure abstractions: Storage, networking etc
4. A hypervisor kernel manages everything except apps
5. A static size, but no resource concerns unless nested



Types of Containers

- ***Long-running***
  - Can be stateless or stateful
  - Eg. Application servers, databases, load-balancers, KV stores etc
  - Typically a need for strong isolation
- ***Transactional***
  - Runs for a period and transforms some data
  - Eg. Runs a build. Processes a web request. Batch processing
  - Should only consume resource when running
- ***Sidecar / helpers***
  - Augments the capabilities of a service or provides a helper function
  - Eg. Logging, monitoring, caching
  - Scales with the service. Potentially hindered by strong isolation

Isolation Domains and Data Persistence

- How do I isolate workloads from each other?
  - Runtime isolation – resource limits, kernel panic, ESX host failure, rack or region failure
  - Network isolation – traffic sniffing, firewalls, encryption, rate limiting
  - Storage isolation – data persistence, backup, networking, RBAC

- Stateful vs Stateless / Cattle vs Pets
  - Different classifications of data. Where should it go?
    - Image state, container state, volume state. What's the difference?
  - Should data lifespan be inherently tied to compute (VM / container)?
  - Without live migration, shared storage and HA, stateless looks attractive

- The question is not "what plumbing do I need?"
  - What *characteristics* or business value do I need for my application?
  - Better question than, "do I need a container or a VM?"

Characteristics: Containers vs VMs? So 2014!

- ***Speed***
  - Start time vs throughput. Benefits transactional containers
  - Hello World vs Tomcat. Less benefit for long-running apps
- ***Efficiency***
  - Less memory consumption? Depends when virtualized
  - Network traffic NAT'd through guest vs straight to vNIC
- ***Portability***
  - Docker image abstraction and API is very portable
  - Subtle issues with kernel versions and patches
- ***Isolation***
  - Shared kernel significantly reduces runtime isolation
- ***Granularity***
  - Containers are great for granular services. Group in a VM as isolation domain.

Container Workflow Efficiencies

- **A Portable Runtime Platform**
  - "Package once, run anywhere"
  - Defacto runtime model for higher level frameworks
- **The Container Model**
  - "Docker is to apt what apt is to tar" – dependency management
  - Snapshotting, image format, Docker Hub, state management (volumes etc)
  - Scripting and automation – express an environment in a text file
  - Predictable initialization state – predictability vs. reconfiguration
- **Continuous Integration**
  - Secure registries, integration with popular tooling
  - Rapid improvements in tooling, monitoring, log integration etc.

vmware®

The Container Stack and Its Challenges

- I want to run containers. Where do I start?
- Should I run containers alongside my other apps or create a silo?
- Should I be considering bare metal?
  - So how many containers should I have in each VM?
  - How big should each container host be? Impact of re-configuration?
  - Which containers can be safely co-located with which other containers?
  - Which Linux distro do I want and how to handle the high patch cadence?
  - Who is responsible for infrastructure admin and how does it integrate?

So What is VIC and How does it help?

- VIC brings all of the container workflow efficiencies to vSphere infrastructure
  - Control vSphere infrastructure from a Docker client without having any vSphere credentials
  - Treat VMs as ephemerally as containers
  - No more OVAs, VMDKs, Templates, Cloning. Push / pull your state from secure registries

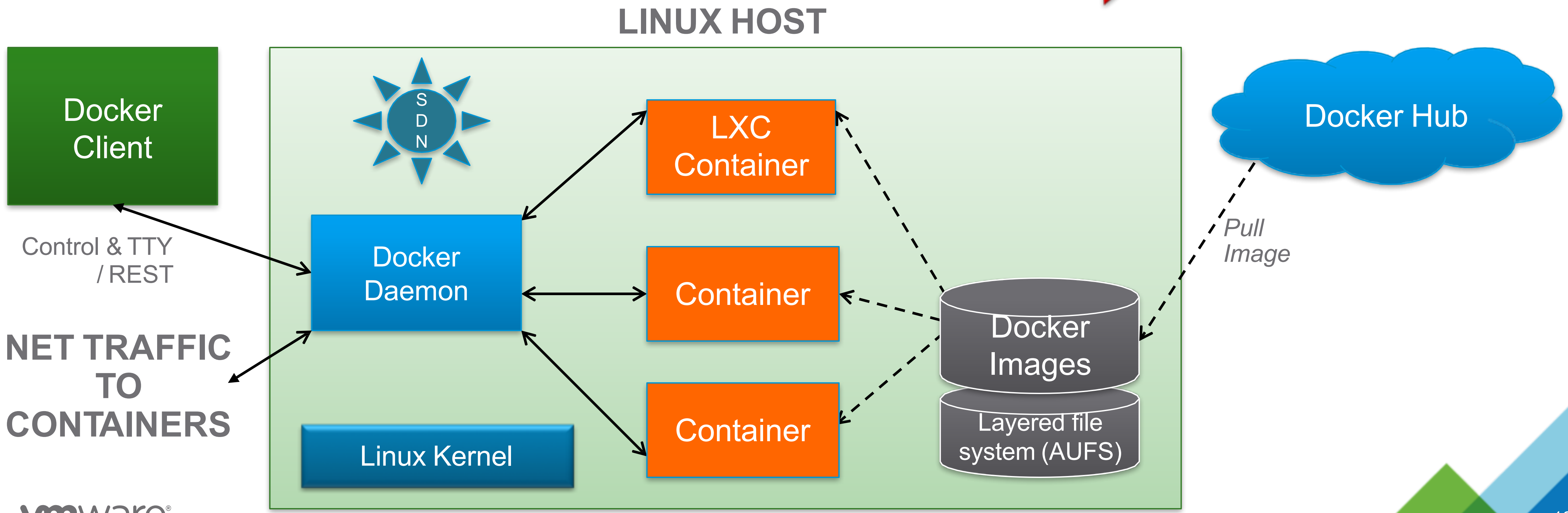
- VIC allows you to translate business value into plumbing
  - Eg. I need to deploy Wordpress with MySQL.
  - Do I want strong isolation between these workloads? Container *as* a VM
  - Do I want strong isolation from other tenants? Container *in* a VM

- VIC helps to draw clear lines between admins and users

Revisit: What is a Container Host?

1. Control plane & lifecycle management for containers
2. Resource scheduling and a container abstraction
3. Infrastructure abstractions: Storage, networking etc
4. A single Linux kernel manages everything
5. A static size and a resource reservation when virtual

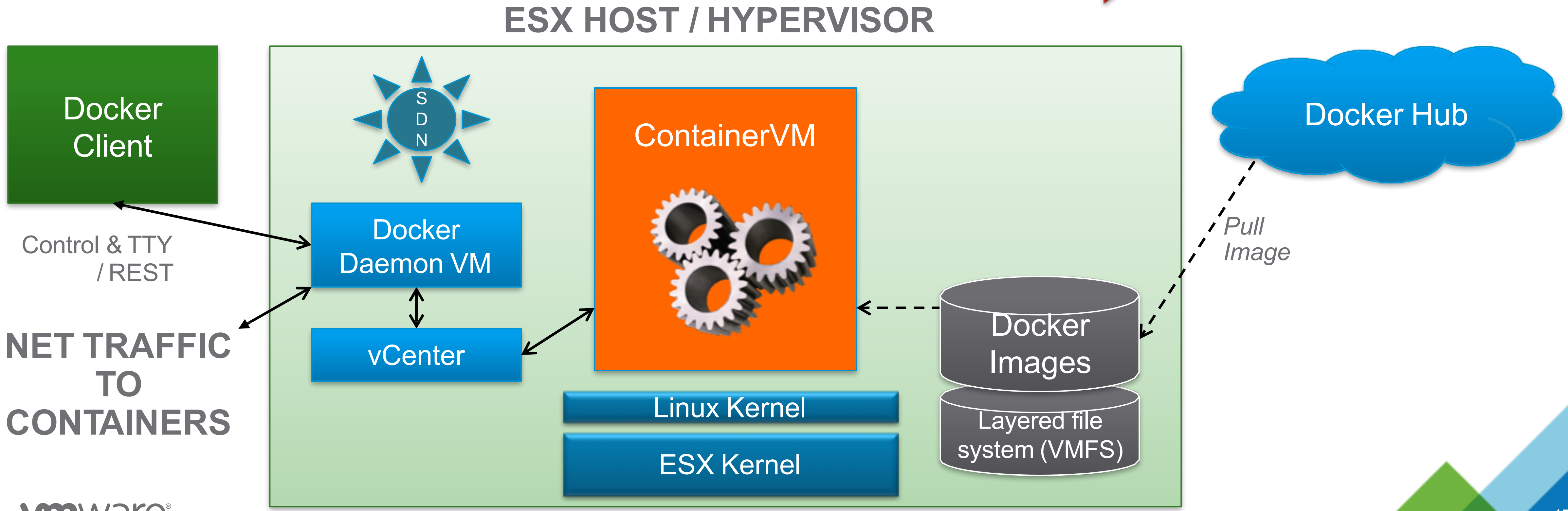
STATEFUL
LONG-RUNNING
SINGLE USER
SINGLE USE



What is a "ContainerVM"?

1. An executable process
2. Resource constraints / private namespace
3. Binary dependencies: Application, runtime, OS
4. A *private* Linux kernel for running the executable
5. Ephemeral and persistent storage layers

STATELESS
PORTABLE
FAST
SECURE

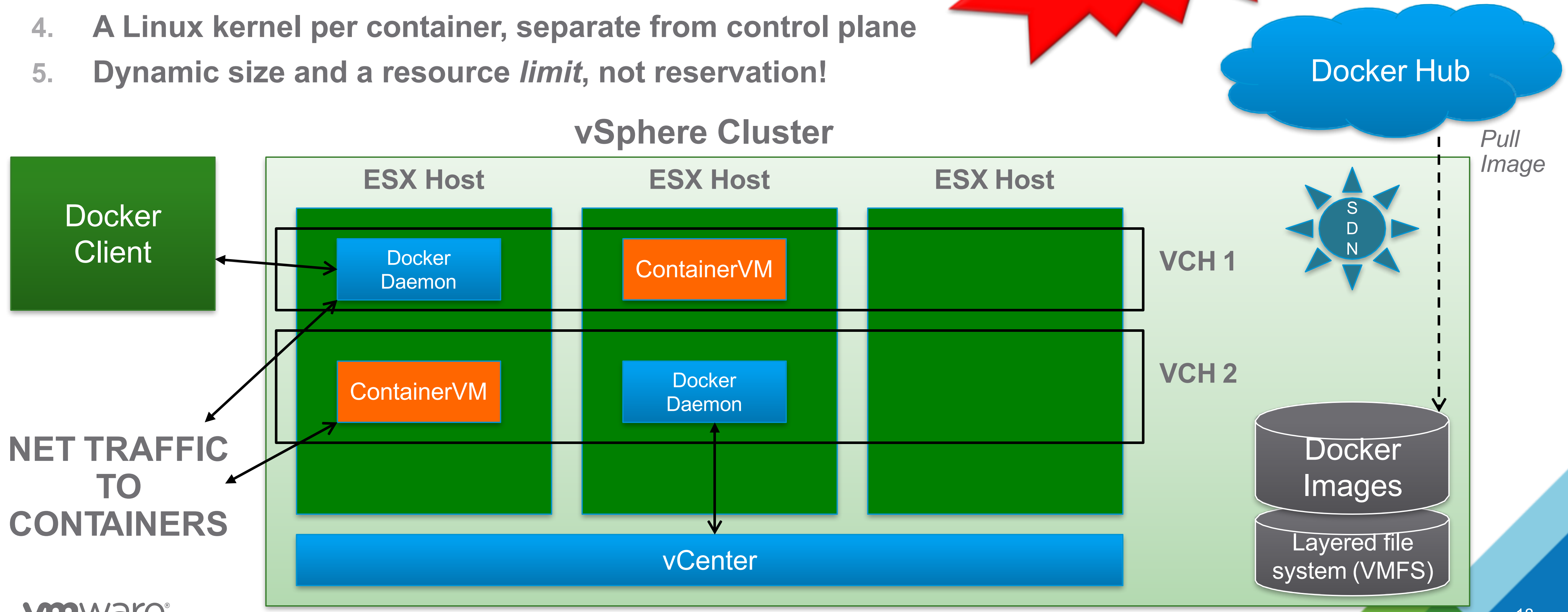


vmware

What is a Virtual Container Host?

1. Control plane & lifecycle management for ContainerVMs
2. Resource scheduling and a container abstraction
3. Infrastructure abstractions: Storage, networking etc
4. A Linux kernel per container, separate from control plane
5. Dynamic size and a resource *limit*, not reservation!

VIRTUAL
SINGLE USER
MULTI USE
MULTI HOST



VIC Roadmap

- VIC 1.0
  - Docker engine and registry (Harbor) shipped in Q4 2016
  - MVP Docker commands. Fundamentals of integration
  - vSAN, vMotion, DRS, NSX distributed port groups
- VIC 1.1
  - Additional Docker capabilities – Eg. compose, exec
  - Critical bug fixes
  - Deeper SDDC integrations – Eg. HA
- VIC 1.2
  - Security enhancements – Image signing, vulnerability scanning
  - Additional Docker capabilities
  - Deeper NSX integration
  - Further SDDC integrations – vVols, SDRS

Questions

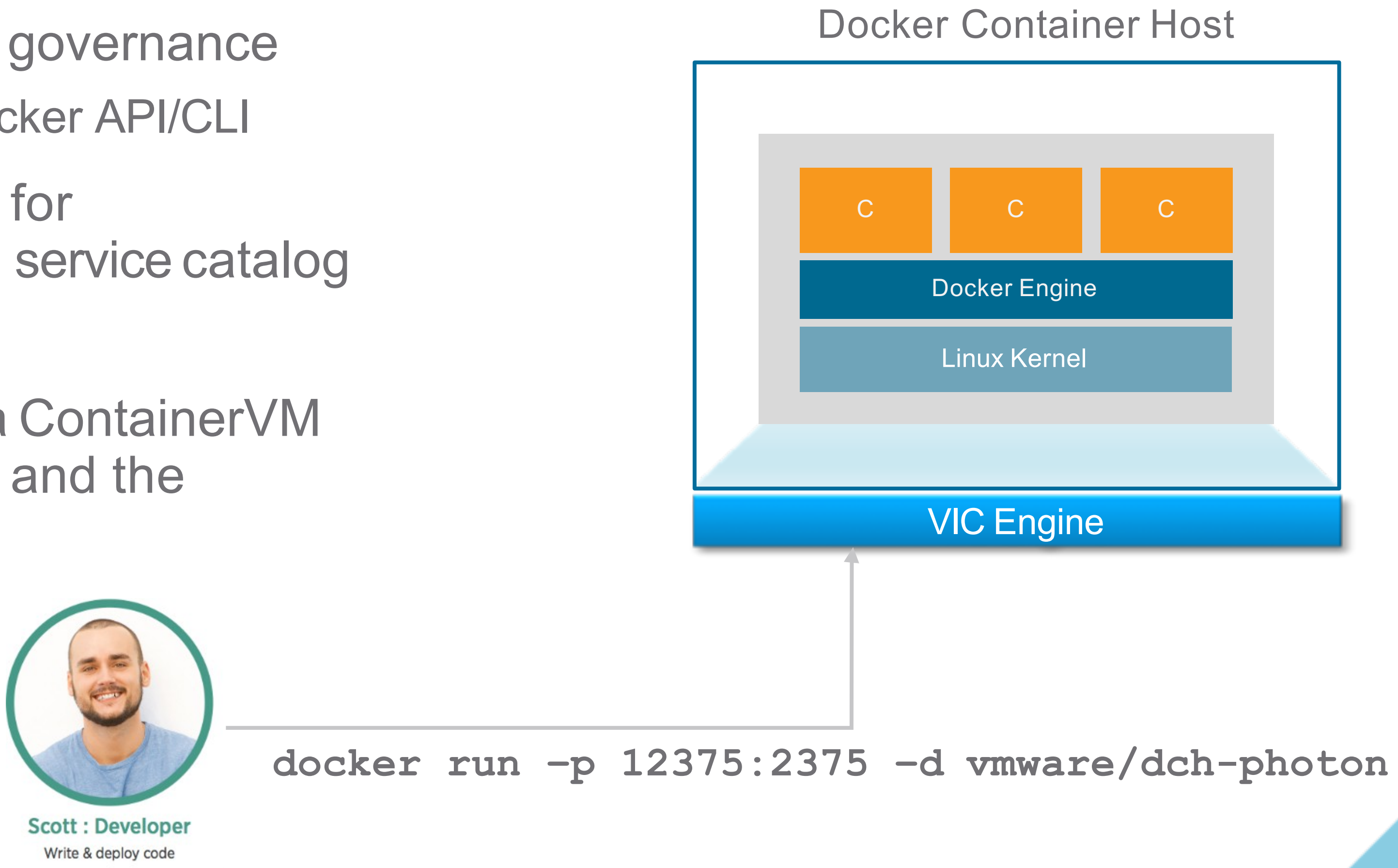
@bensdoings

Github.com/vmware/vic-product

Demo - VIC

New in 1.2 - Developer Sandbox

- Developer self-service with VI Admin governance
  - Developer consumes resources via Docker API/CLI
- Provides developers with self-service for applications not yet in the enterprise service catalog
  - Rapid prototyping
- Run a full-fledged docker engine as a ContainerVM using vSphere Integrated Containers and the Docker API/CLI

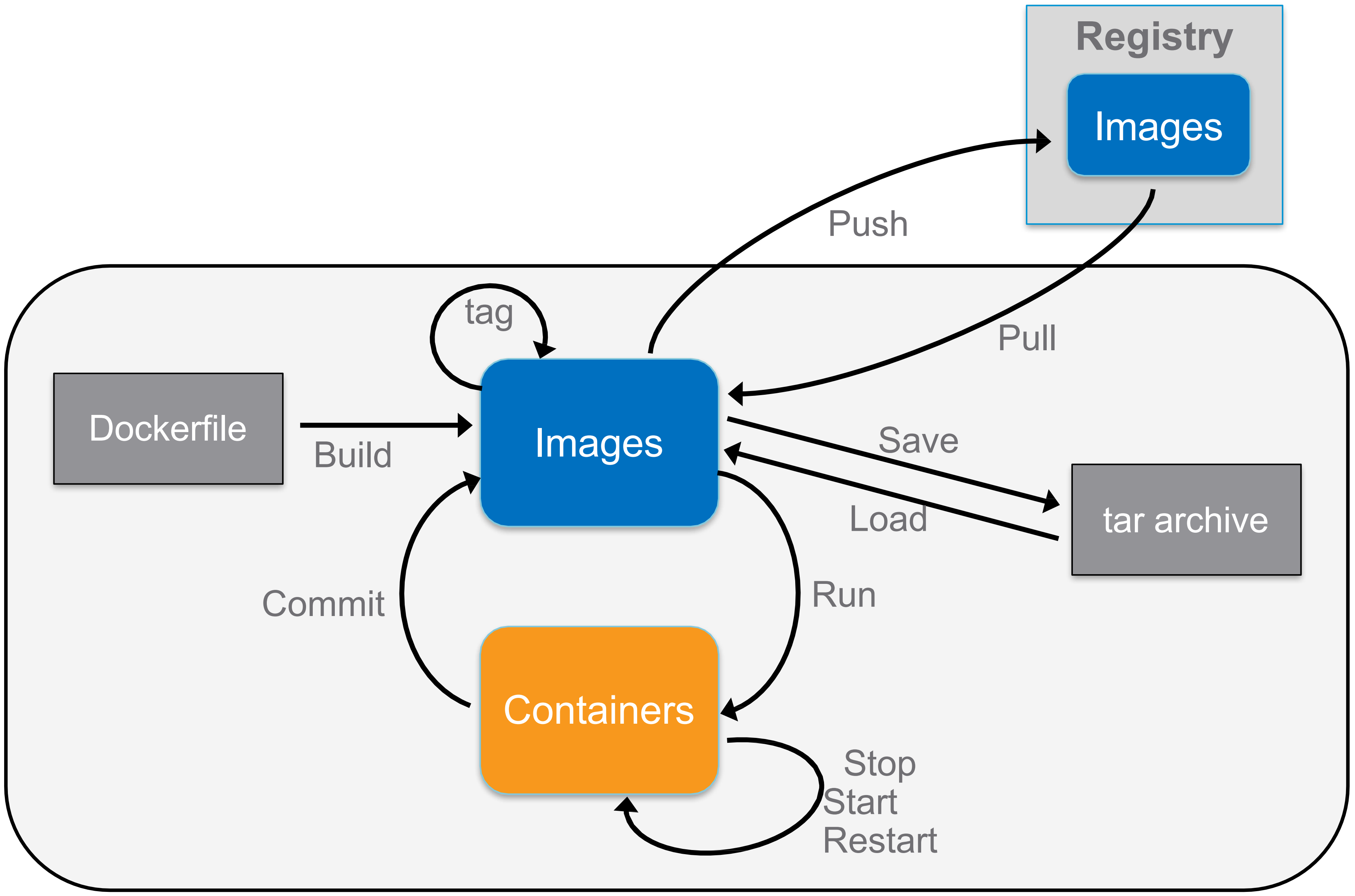


```
docker run -p 12375:2375 -d vmware/dch-photon
```

Project Harbor

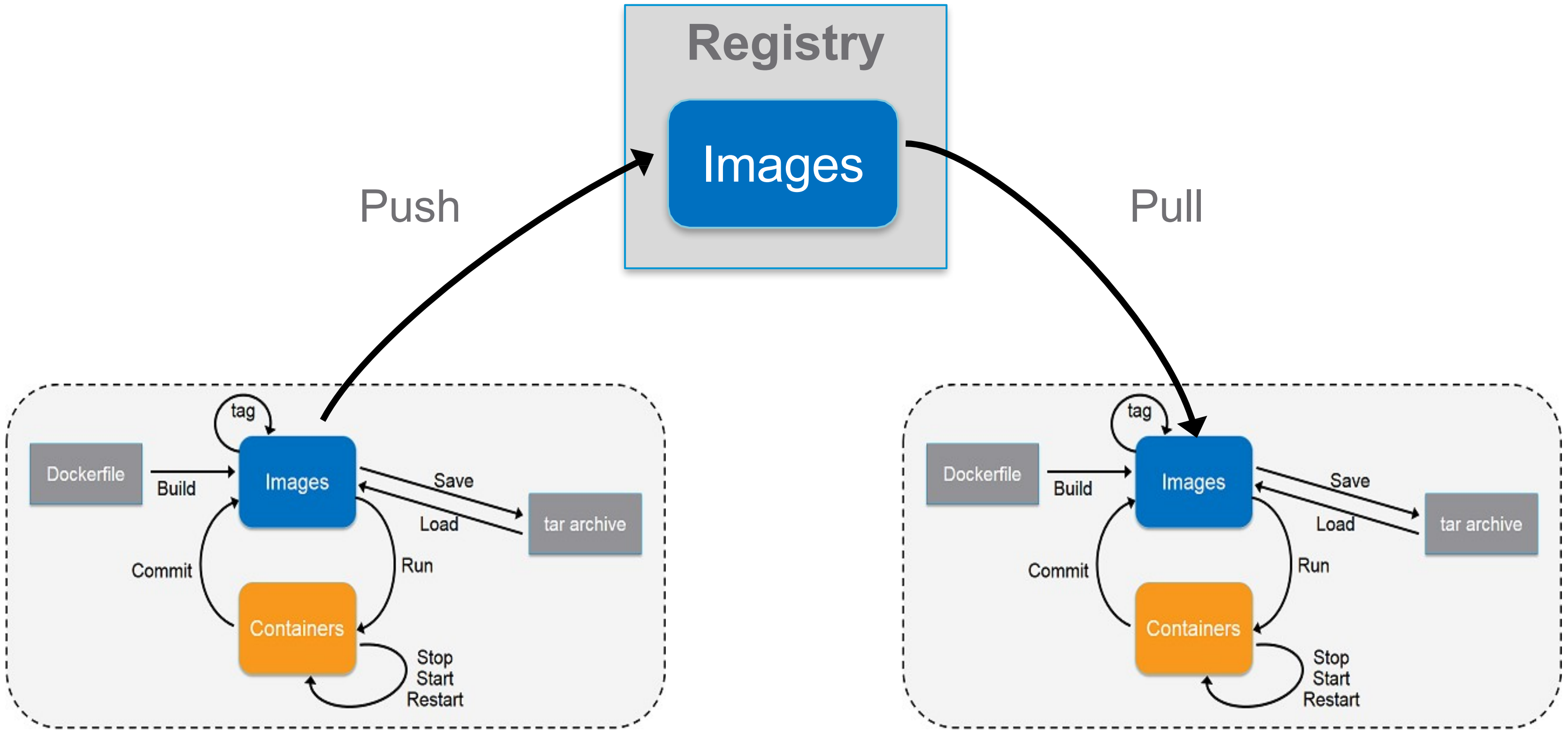
Harbor

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



Harbor

Project Harbor

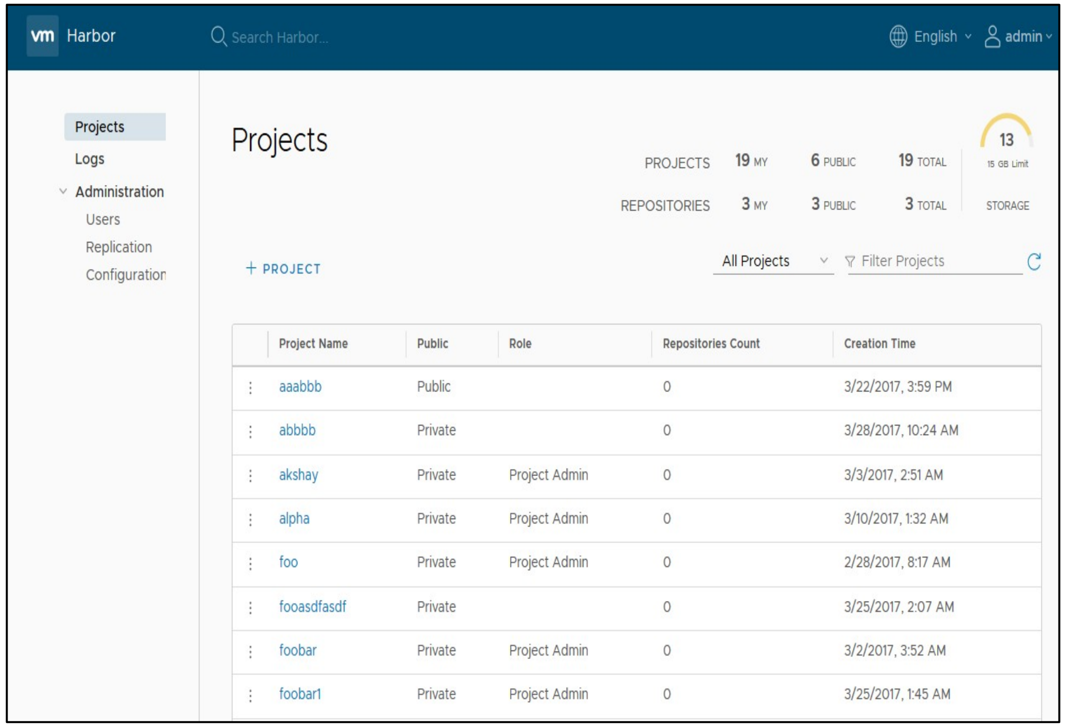
HARBOR™

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

vmware®

Key Features

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



Projects

	MY	PUBLIC	TOTAL
PROJECTS	19	6	19
REPOSITORIES	3	3	3

STORAGE: 13, 15 GB Limit

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	2/28/2017, 8:17 AM
fooasdfasdf	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, 1:45 AM

Users and Developers

- Users

20K+
Downloads

2600+
Stars

300+
Users

- Developers

700+
Forks

55
Contributors

6
Partners

Harbor users and partners

caicloud 才云

RANCHER

数人云 shurenyun.com

品高云 BingoCloud

时速云 tenxcloud.com

OnStar 安吉星

TREND MICRO

AXA 安盛天平

TalkingData

通联数据 DATAYES

iFRE

亿阳信通 BOCO Inter-Telecom

Wise2c TECHNOLOGY

慧博云通 HYDSOFT

SLAMTEC

Open EDUTAINMENT 奥鹏教育

BOER 博耳智能 BOER SMART

BEYONDSOFT beyond your expectations

vmworld

Harbor Architecture

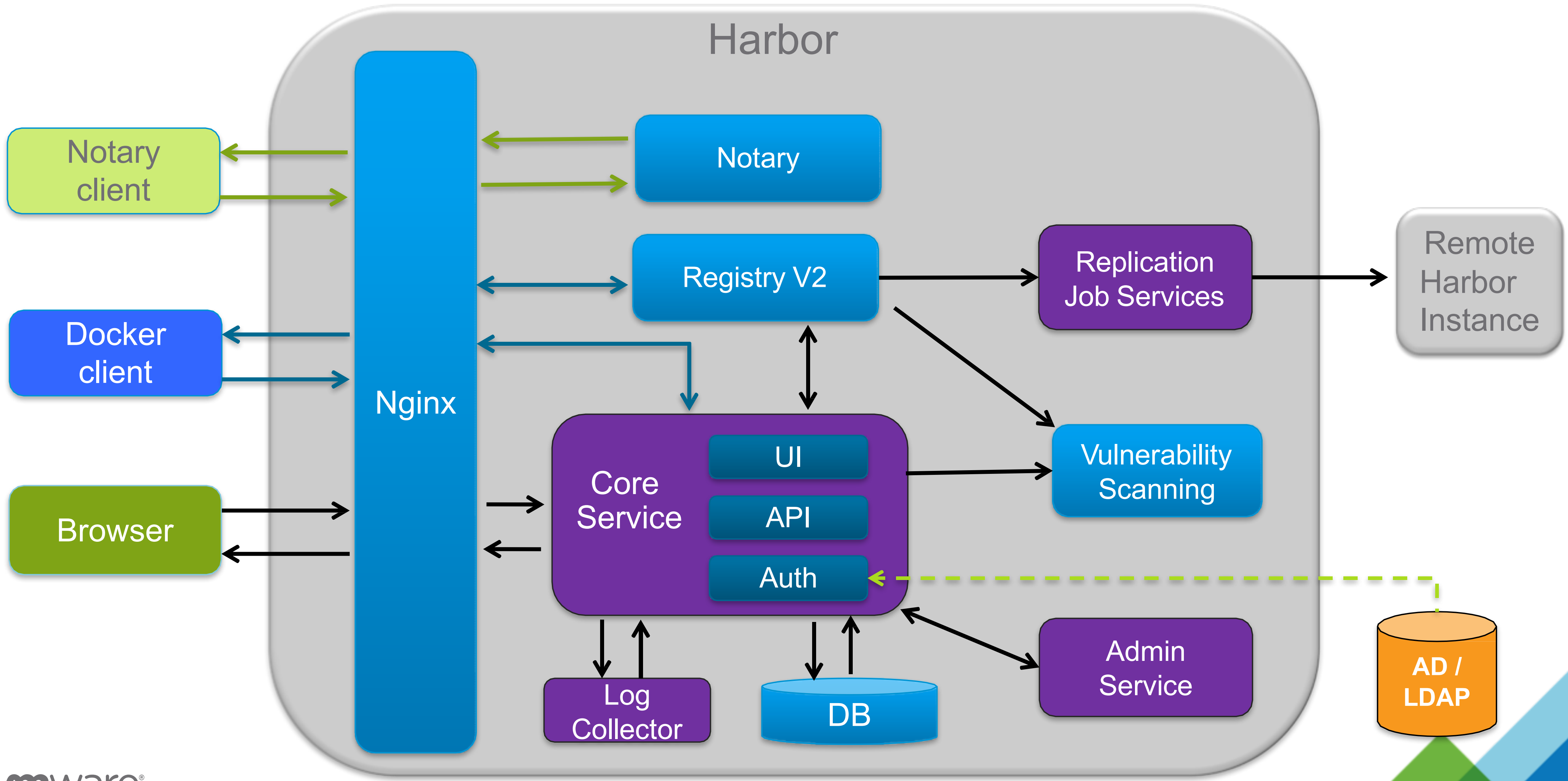
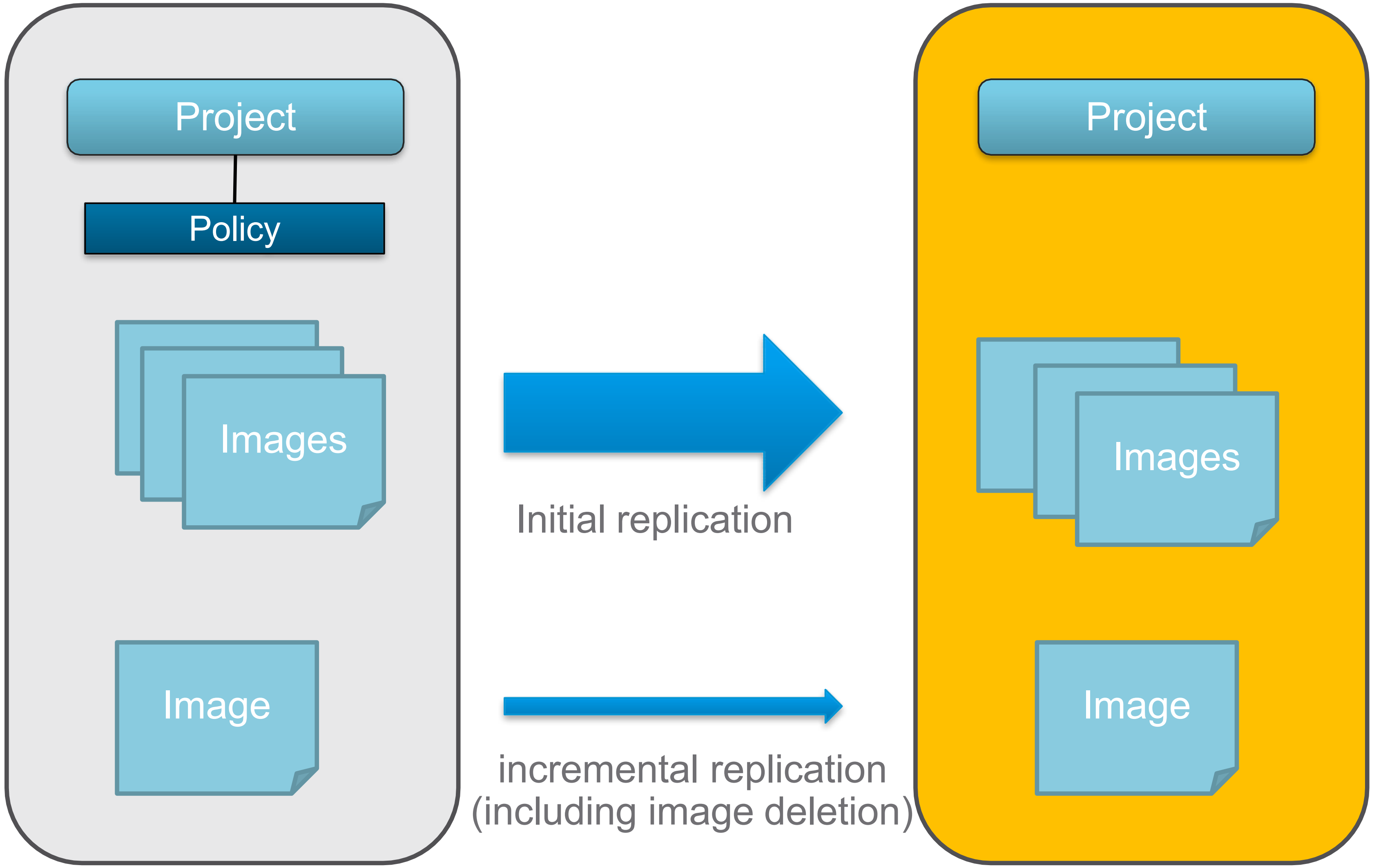


Image replication (synchronization)



Harbor

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

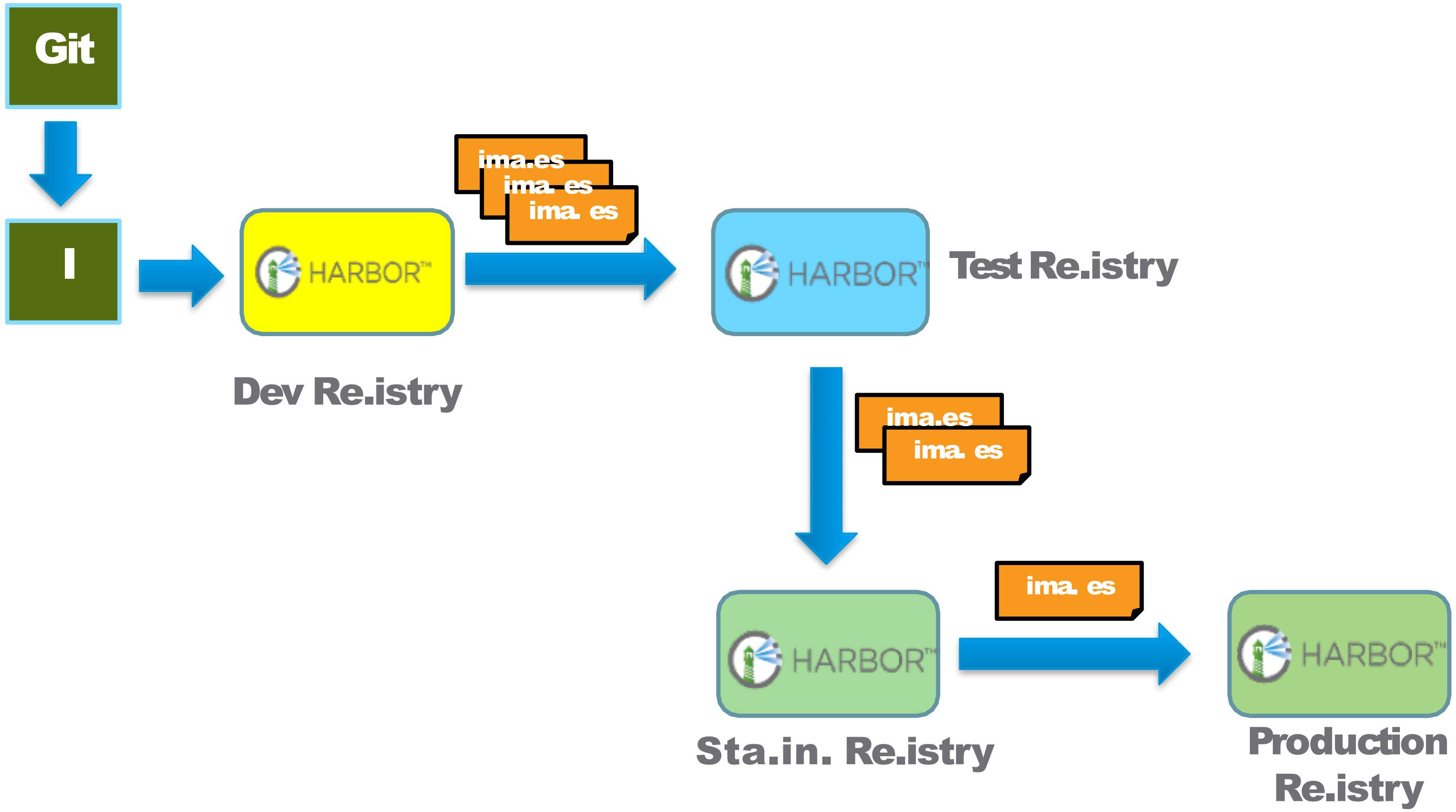
Same Dockerfile Always Builds Same Image?

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:14.04` 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



Ima.es are synchronized between environments
by usin. Harbor re.istry.

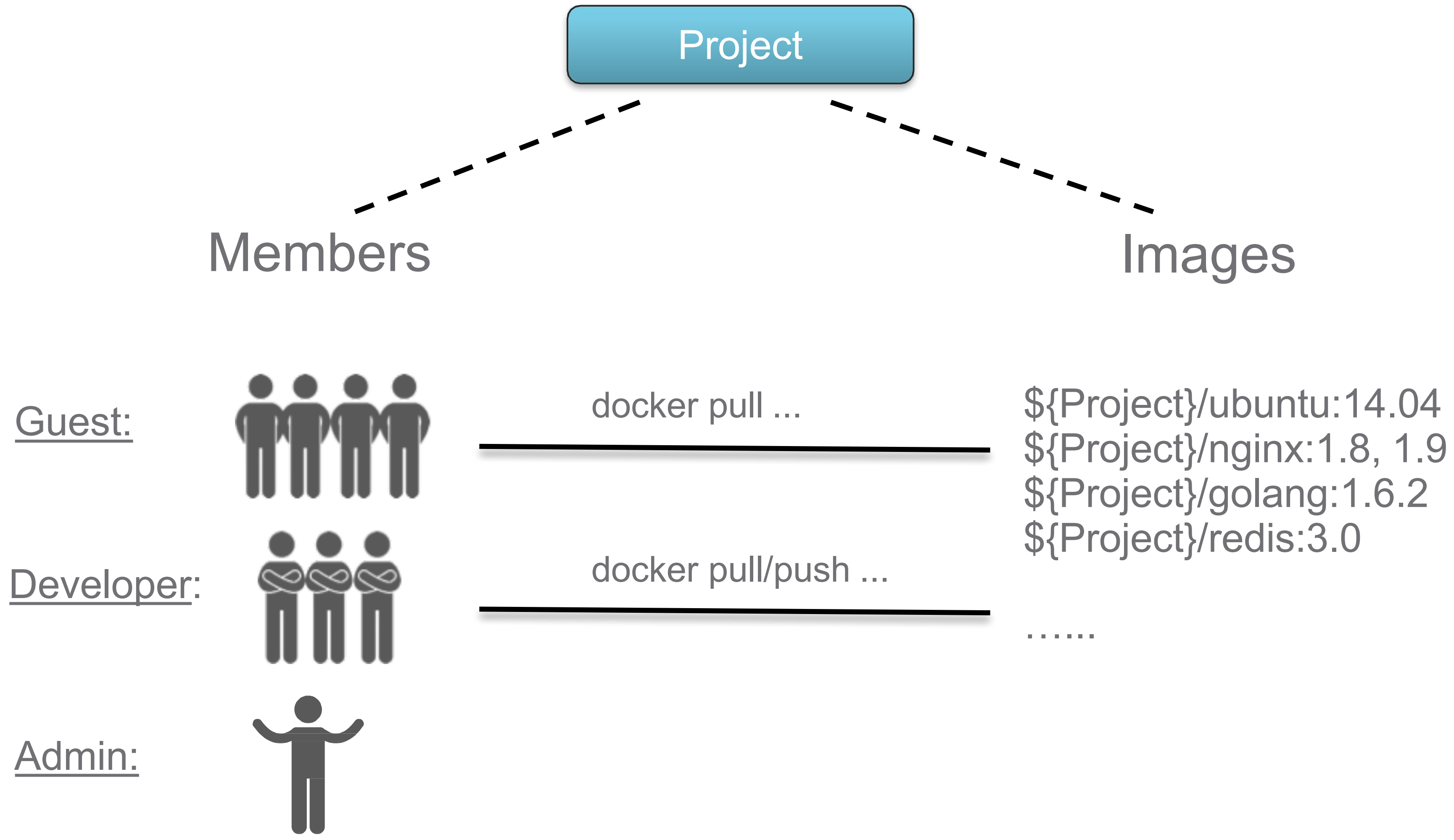
vmware®

Harbbor

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



Other security considerations

- 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
  - Identify images with vulnerabilities during pushing
  - Prevent images with vulnerabilities from being pulled
  - Regular scanning based on updated vulnerability database

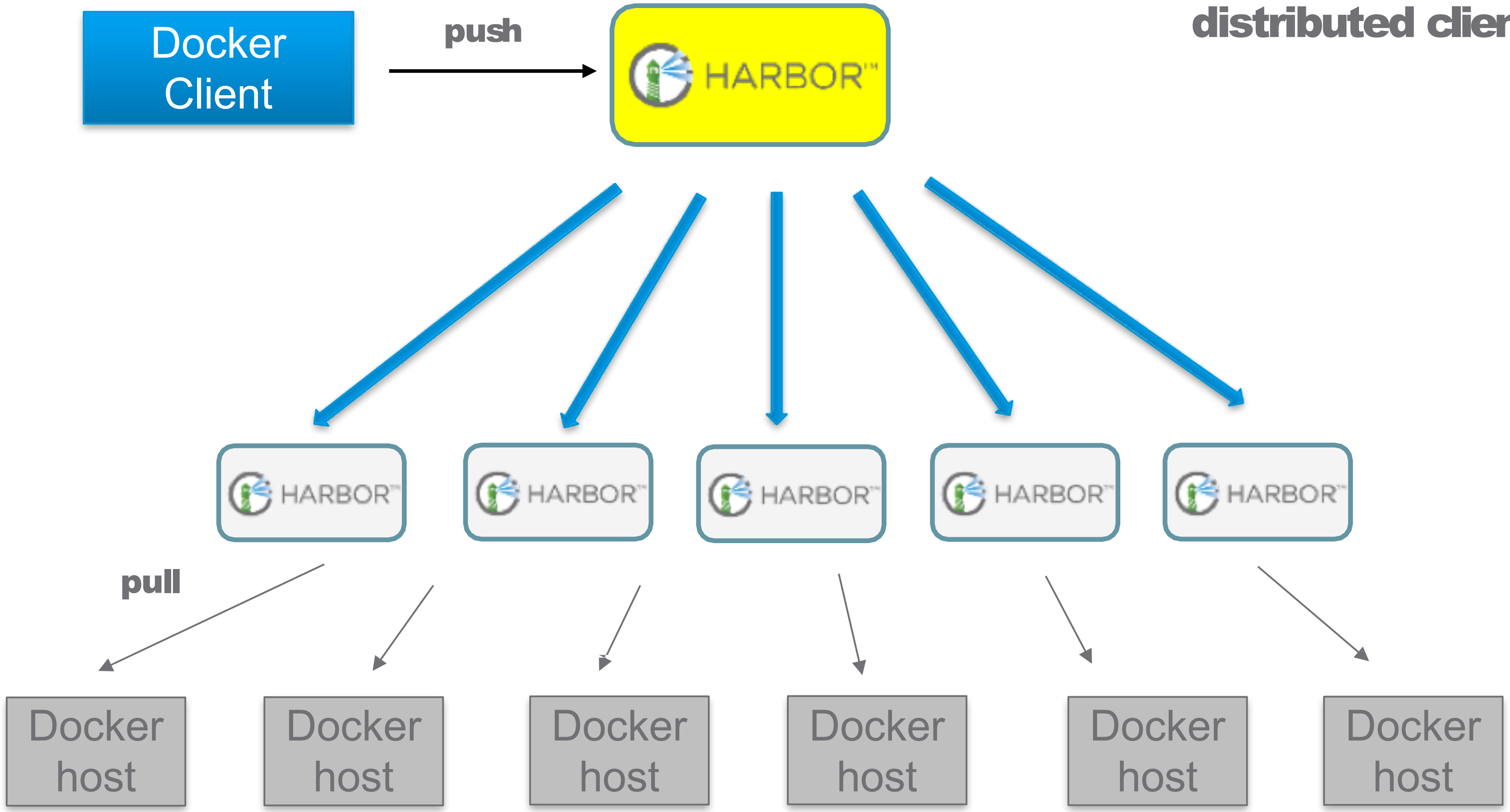
Harbor

Image Distribution

- 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

- Load balancin.
- Works well with .eo.raphically distributed clients



Master – Slave model

Demo: Harbor Registry

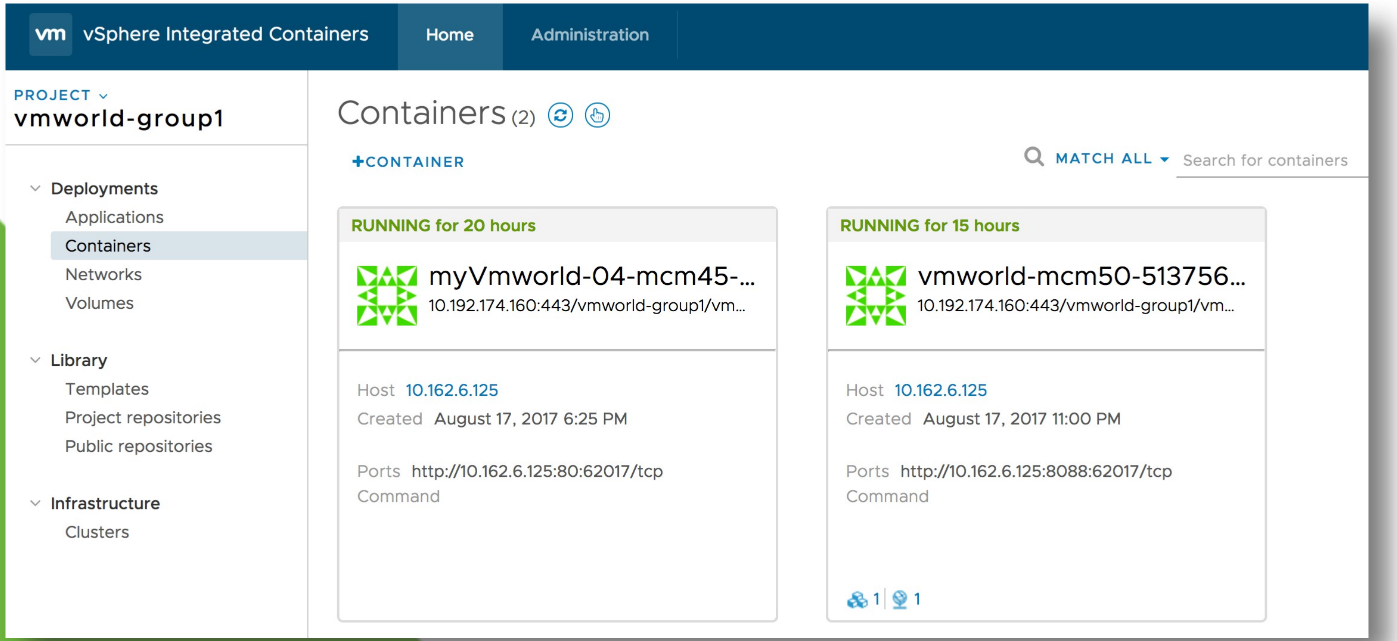
Admiral Overview

What is Admiral?

- A lightweight **Container Management Platform**
- Deploy and manage container based applications.
- Provide placement based on dynamic policy allocation.
- Container Management Layer in both vRA and VIC.
- Works with VIC engine or Docker host, Harbor registry.

Container Deployment

Provision container with the image from the specified registry



vSphere Integrated Containers | Home | Administration

PROJECT
vmworld-group1

Deployments
Applications
Containers
Networks
Volumes

Library
Templates
Project repositories
Public repositories

Infrastructure
Clusters

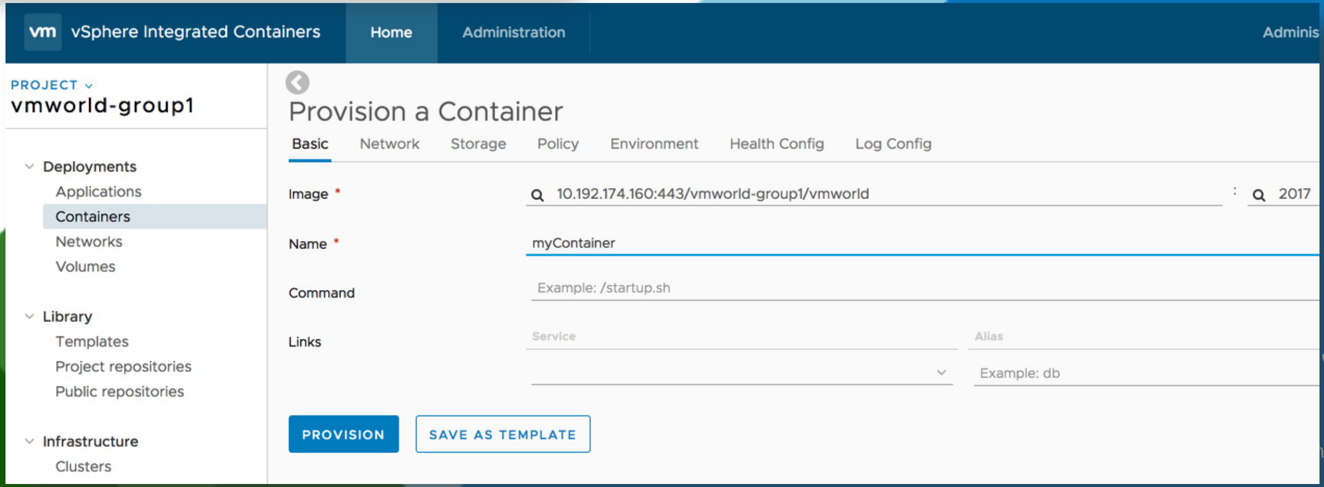
Containers (2)

+CONTAINER

MATCH ALL | Search for containers

RUNNING for 20 hours
myVmworld-04-mcm45-...
10.192.174.160:443/vmworld-group1/vm...
Host 10.162.6.125
Created August 17, 2017 6:25 PM
Ports http://10.162.6.125:80:62017/tcp
Command

RUNNING for 15 hours
vmworld-mcm50-513756...
10.192.174.160:443/vmworld-group1/vm...
Host 10.162.6.125
Created August 17, 2017 11:00 PM
Ports http://10.162.6.125:8088:62017/tcp
Command
1 | 1



vSphere Integrated Containers | Home | Administration | Adminis

PROJECT
vmworld-group1

Deployments
Applications
Containers
Networks
Volumes

Library
Templates
Project repositories
Public repositories

Infrastructure
Clusters

Provision a Container

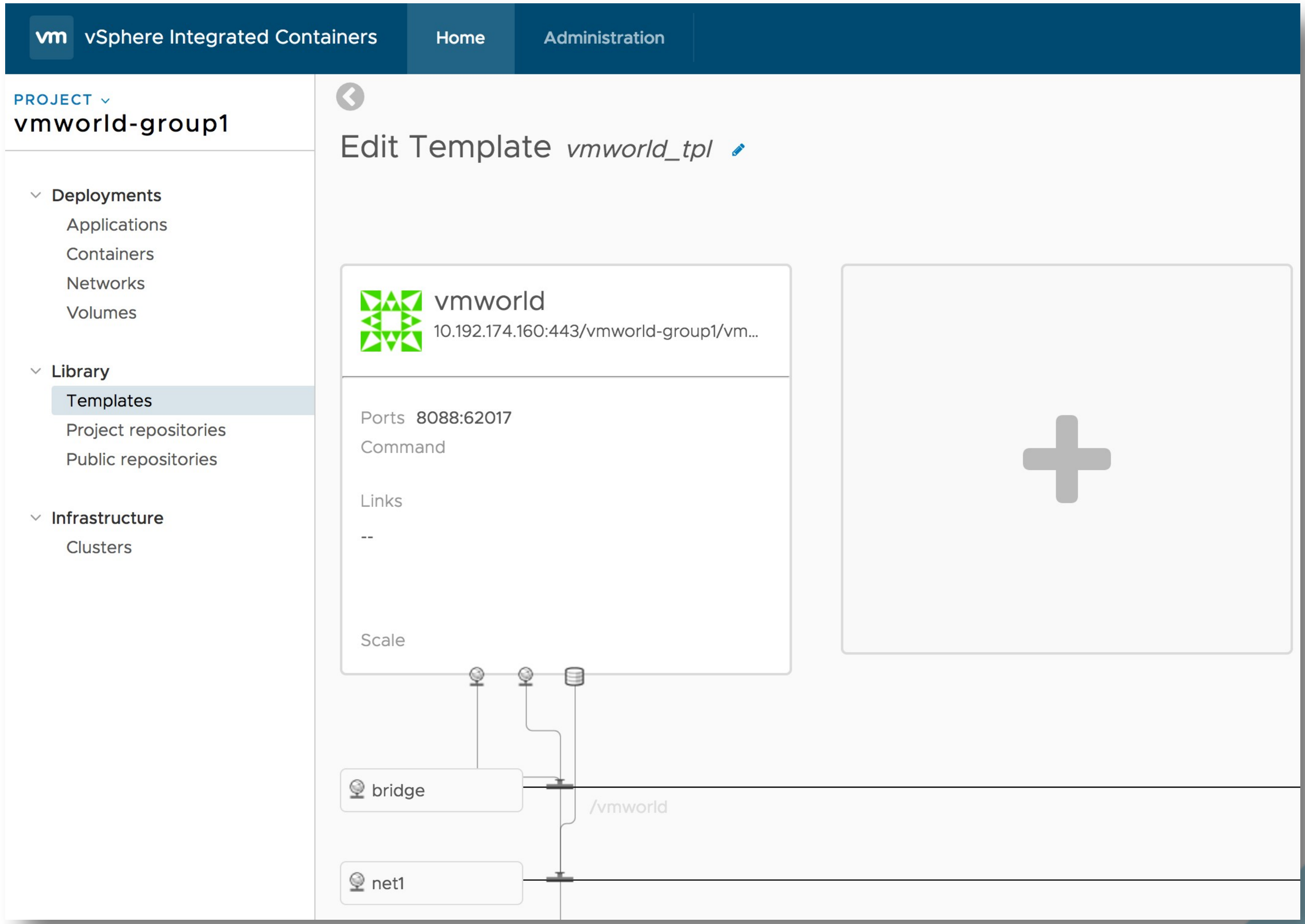
Basic | Network | Storage | Policy | Environment | Health Config | Log Config

Image * 10.192.174.160:443/vmworld-group1/vmworld : 2017
Name * myContainer
Command Example: /startup.sh
Links Service | Alias
Example: db

PROVISION | SAVE AS TEMPLATE

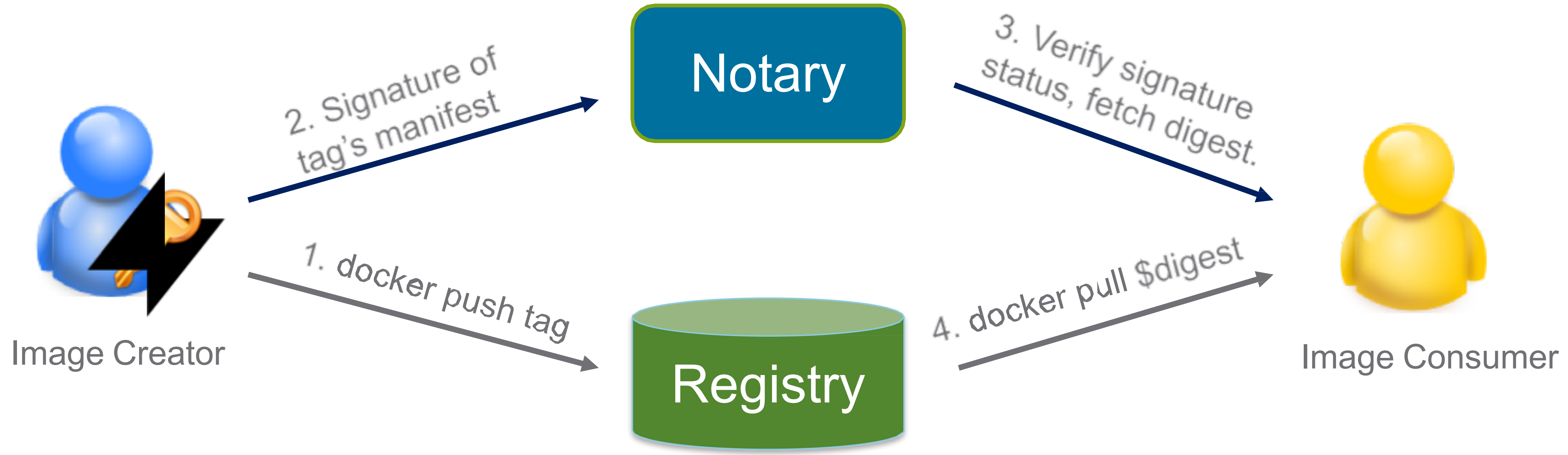
Application Template

- Define your own application structure
  - Containers
  - Networks
  - Volumes
- Provision multiple containers at one time
- Compatible with **docker-compose** yaml file



Practice on VIC engine, Harbor & Admiral

Content trust for image provenance



Vulnerability Scanning

- **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