KubeCon | CloudNativeCon

China 2018

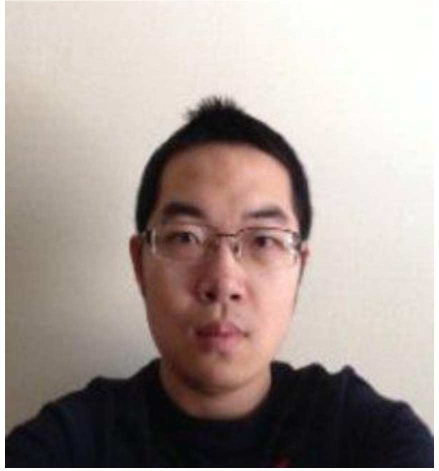
Securing the Perimeter

CFCR/CFAR Chain Of Custody With CI/CD Pipelines

Who We Are...

Keith Strini ...

.... Shaozhen Ding



- Provide operational assistance and guidance
- Build balanced customer product teams delivering “Platform” as a capability within their organization
- Establish and maintain continuous delivery pipelines for deployme nt of Pivotal Cloud Foundry
  and related products in a customer’s infrastructure
- Design and implement continuous integration and continuous deli very processes to deliver
  customer applications to production, fostering a culture of continu ous process improvement

Shift the Mindset

”Assume The Continuous Threat of Compromise and Then Continuously Move The Target ”

From Reactive Courses of Action into Proactive Security Policy

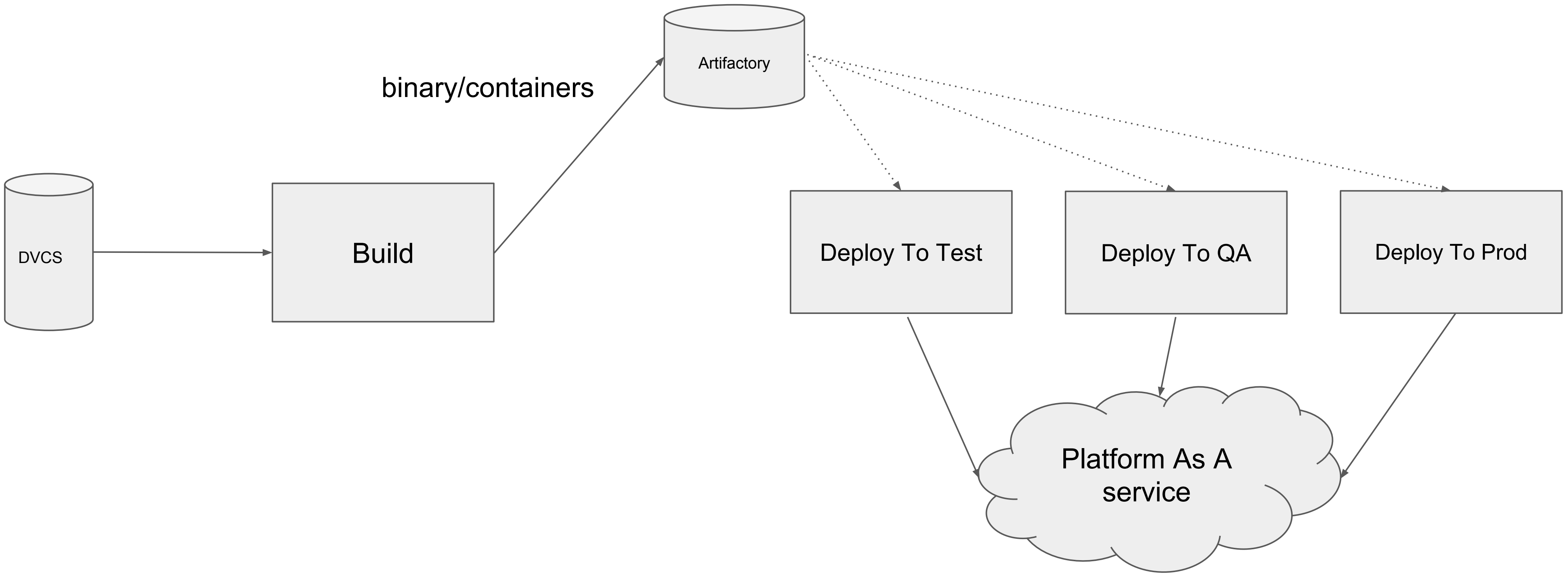
◆ **Recovery Point Objective**

- The recovery point objective (RPO) is the point in time that you wish to recover to.

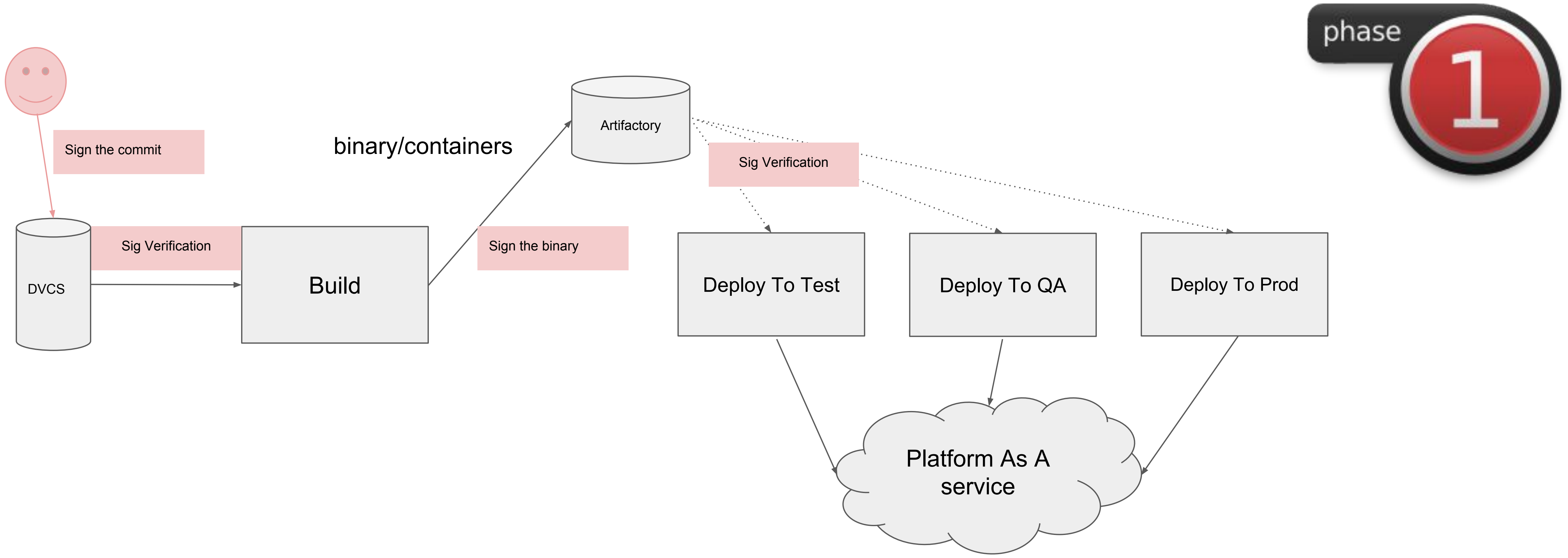
◆ **Recovery Time Objective**

- The recovery time objective (RTO) is how long it takes to recover, taken irrespective of the RPO. That is, after the disaster, how long until you have recovered to the point determined by the RPO.

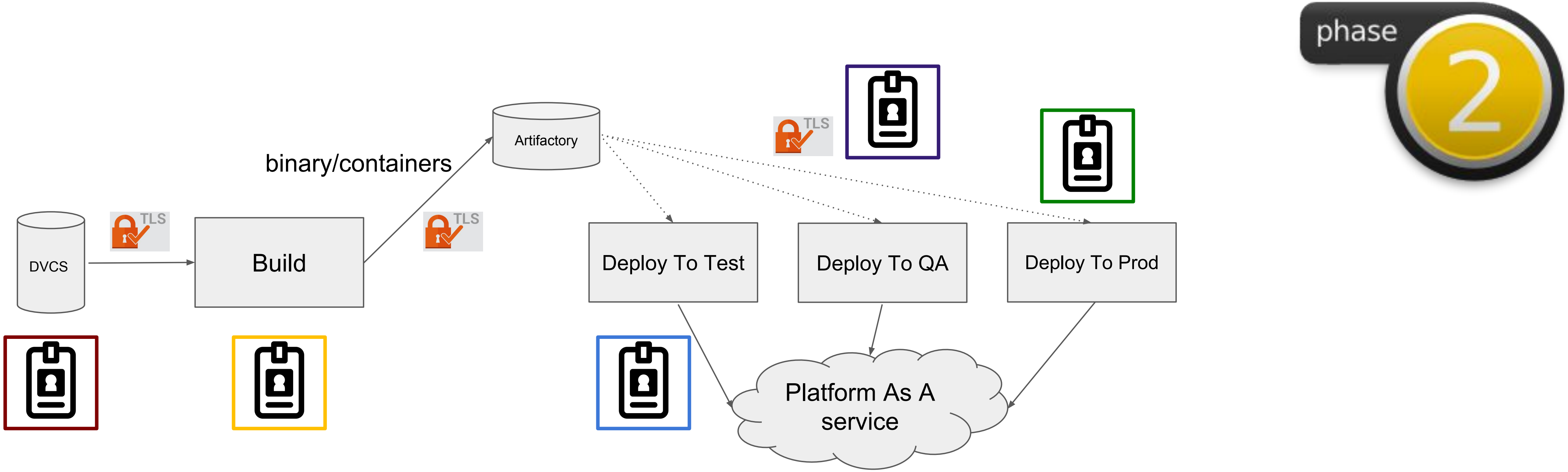
A Standard Delivery Pipeline



Defending the Supply Chain Threat

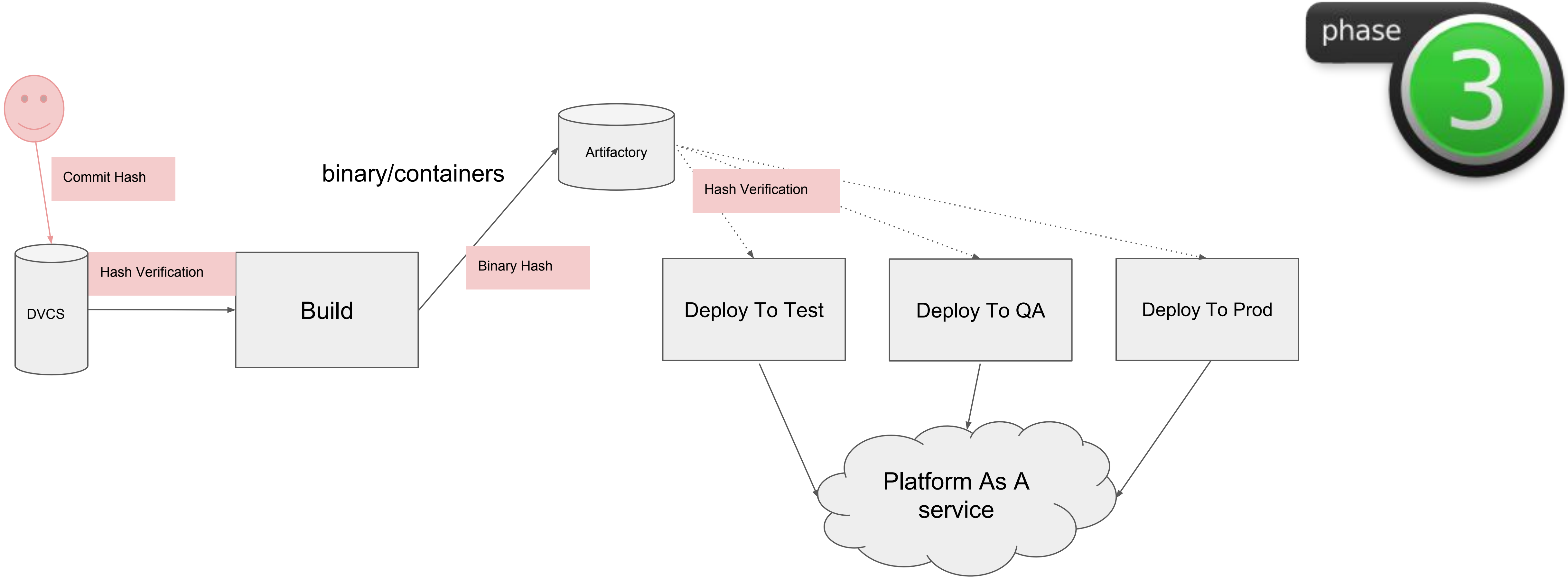


Limiting CI/CD Breakout Exposure

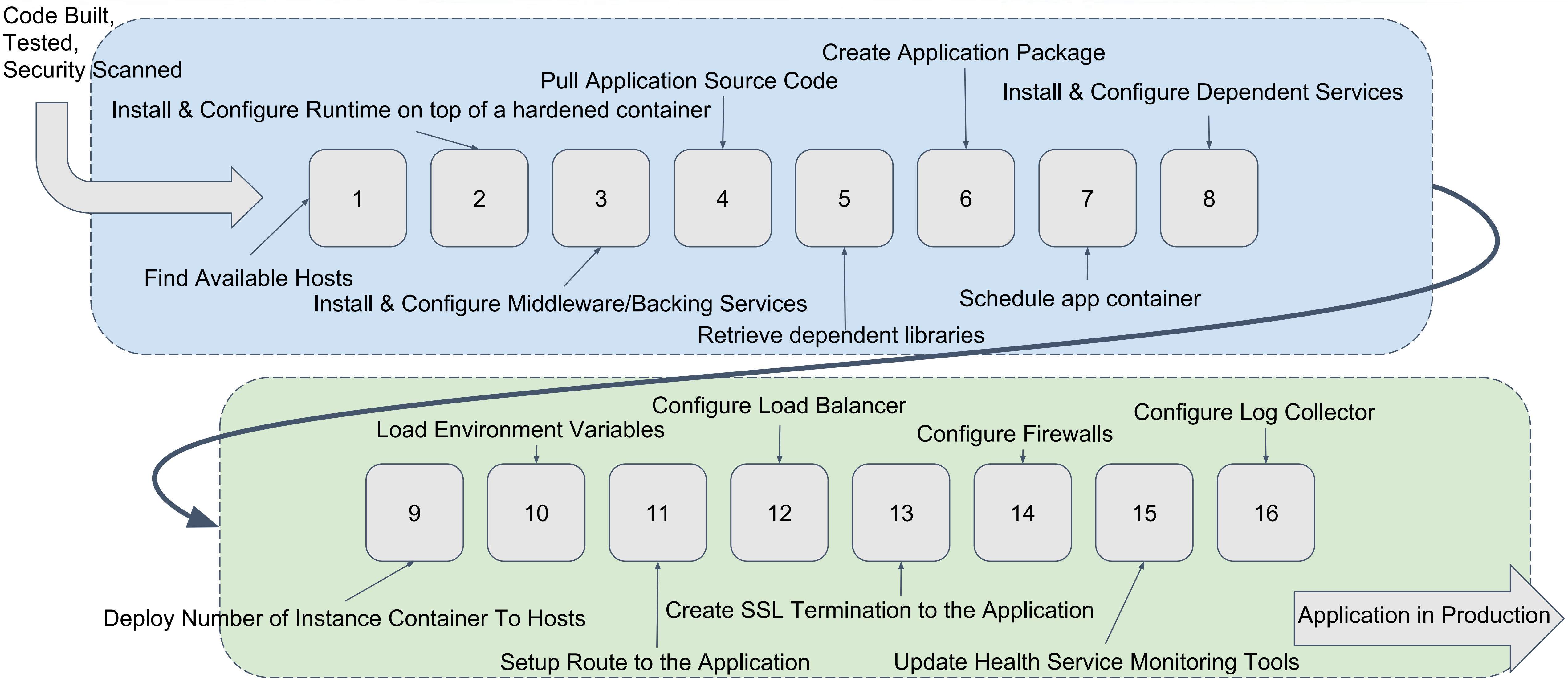


Defending the Application Integrity

KubeCon | CloudNativeCon
China 2018



So What Happens During The Push?


Code Built,
Tested,
Security Scanned
1
2
3
4
5
6
7
8
Find Available Hosts
Install & Configure Runtime on top of a hardened container
Install & Configure Middleware/Backing Services
Pull Application Source Code
Retrieve dependent libraries
Create Application Package
Schedule app container
Install & Configure Dependent Services
9
10
11
12
13
14
15
16
Deploy Number of Instance Container To Hosts
Load Environment Variables
Setup Route to the Application
Configure Load Balancer
Create SSL Termination to the Application
Configure Firewalls
Update Health Service Monitoring Tools
Configure Log Collector
Application in Production

Defending the Continuous Threat

(Privilege Escalation, File System Modification, Container Breakout)

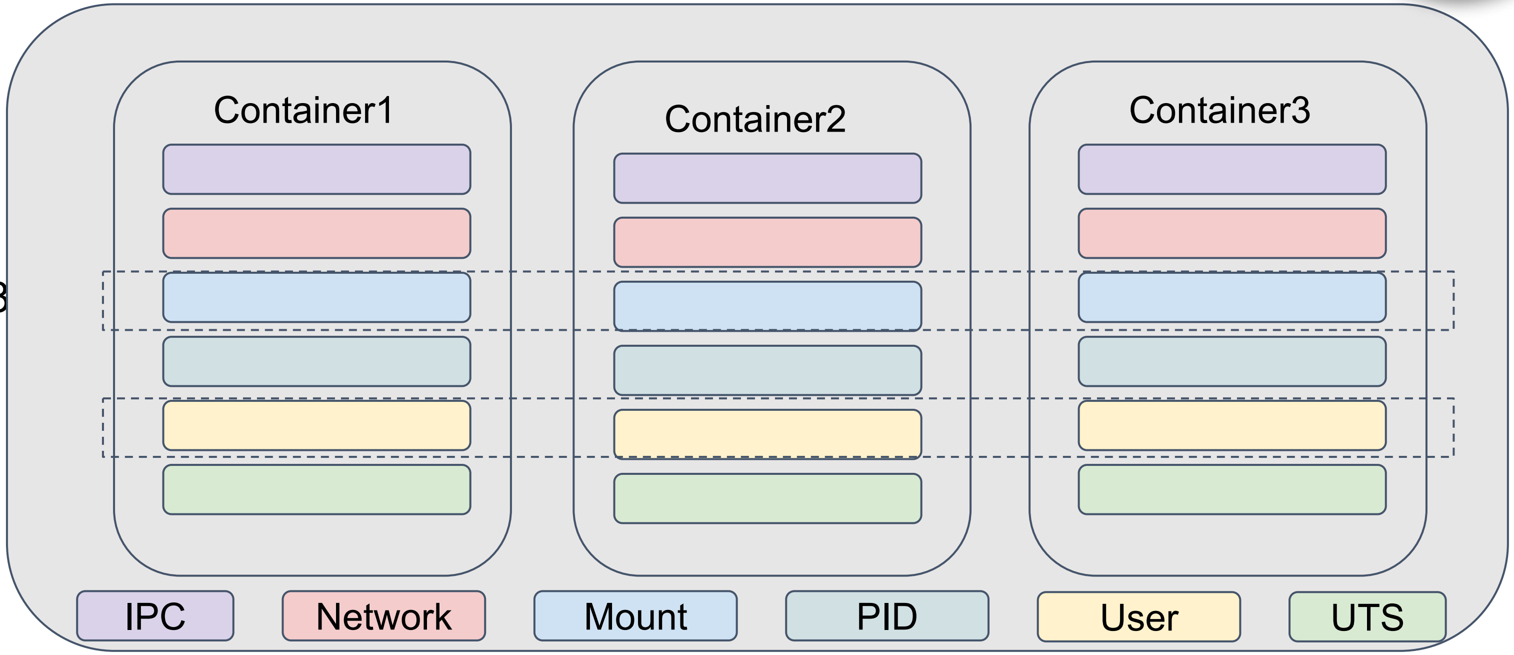
phase 4

User Namespace => Restricted User Scope

Mount Namespace + pivot_root => File System Isolation

Each Container can have its own root filesystem separate from the host

- PCF uses hardened, streamlined Ubuntu stemcell
- VMs use hardened, streamlined cflinuxfs3 rootFS
- PCF uses a combination of OverlayFS and XFS as a filesystem for containers
- The read-only layer in all containers is RootFS
- The application binaries are in a very small read-write layer of the file system



Defending the Continuous Threat

(DoS of Containers, DoS of Service to Host, Kernel Mods, MITM Attacks)

phase 4

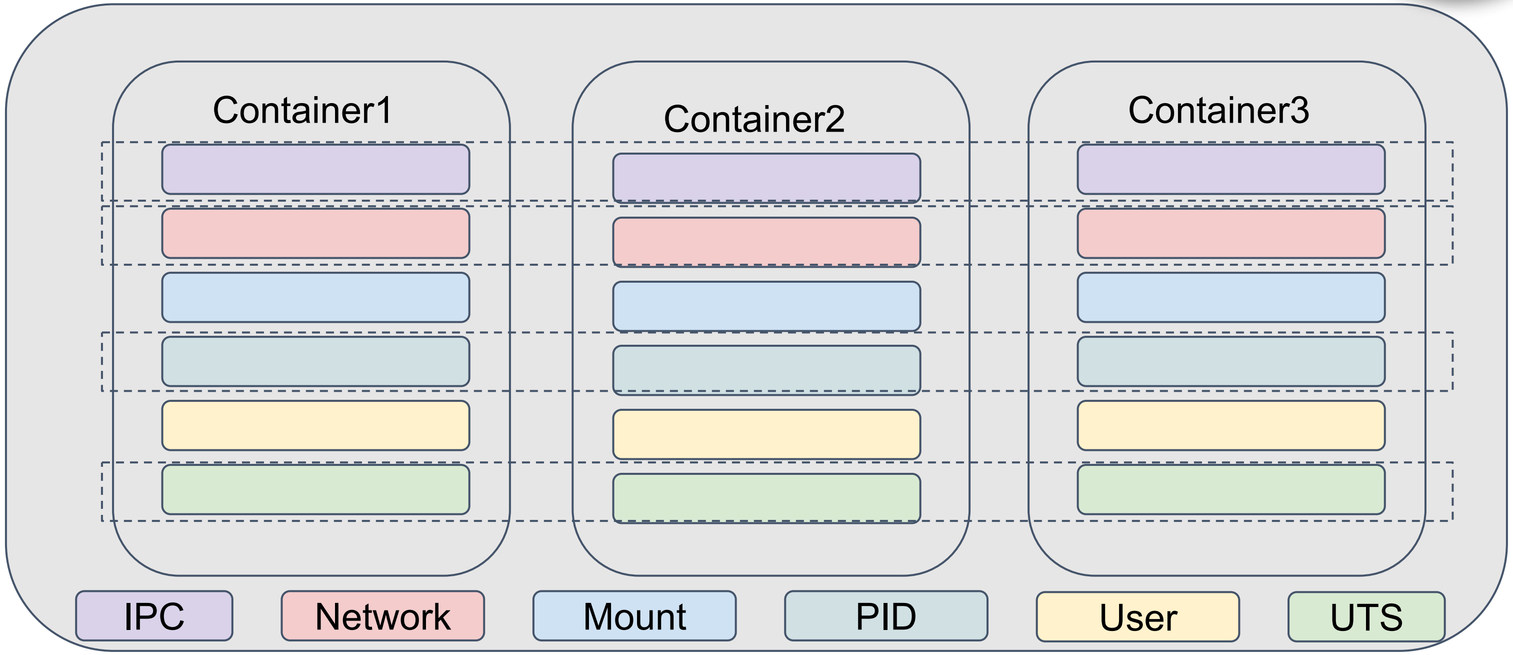
Network Namespace + port virtualization => Network Isolation

CGroups => Resource Isolation (CPU share Capping), Device Access WL

Rootless containers

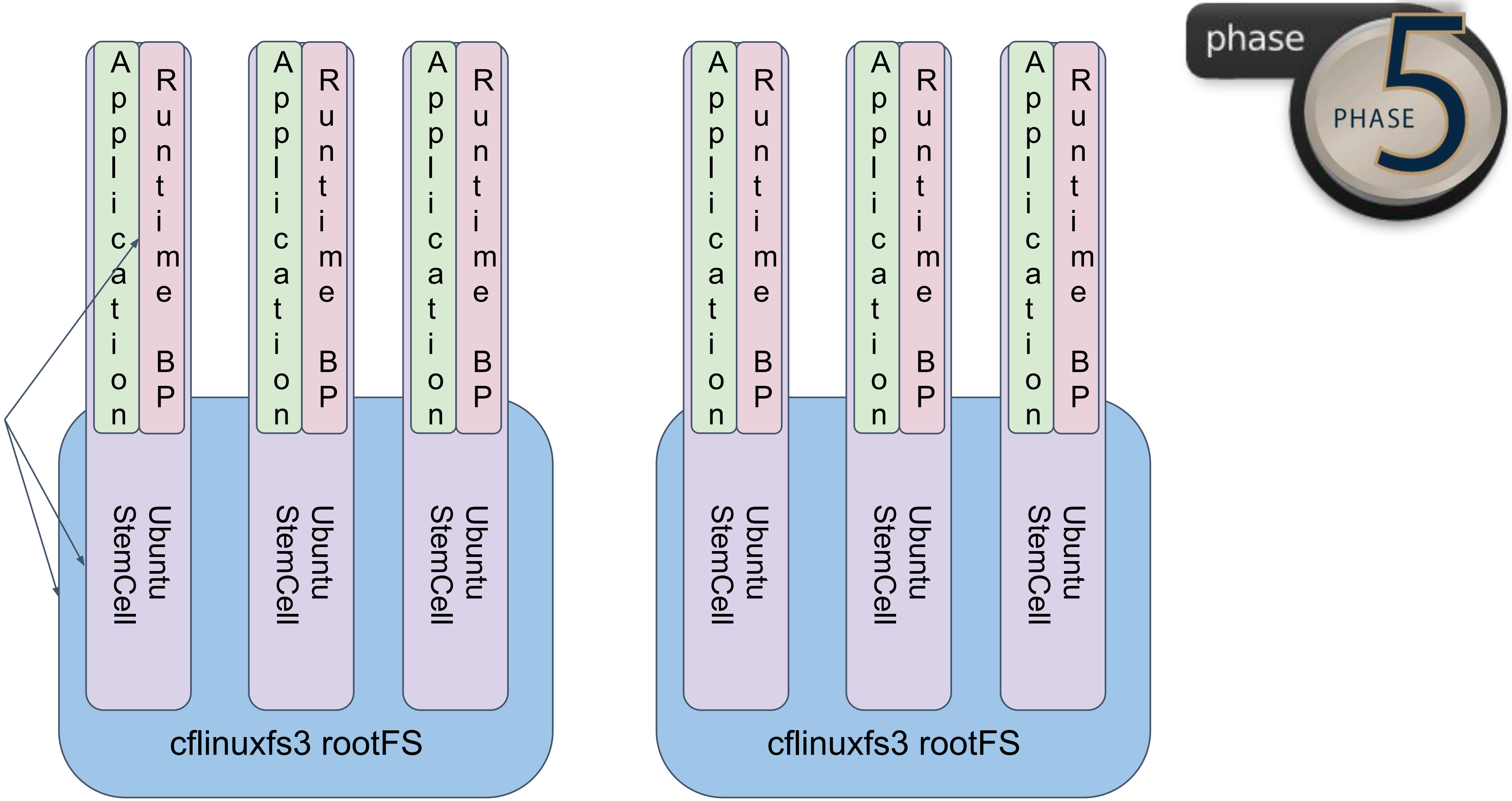
App Armor confines untrusted processes

Seccomp system call filtering



Defending the Continuous Threat

(Malware)



Continuous Zero Downtime CVE Patching

(Repair Vulnerable Components)

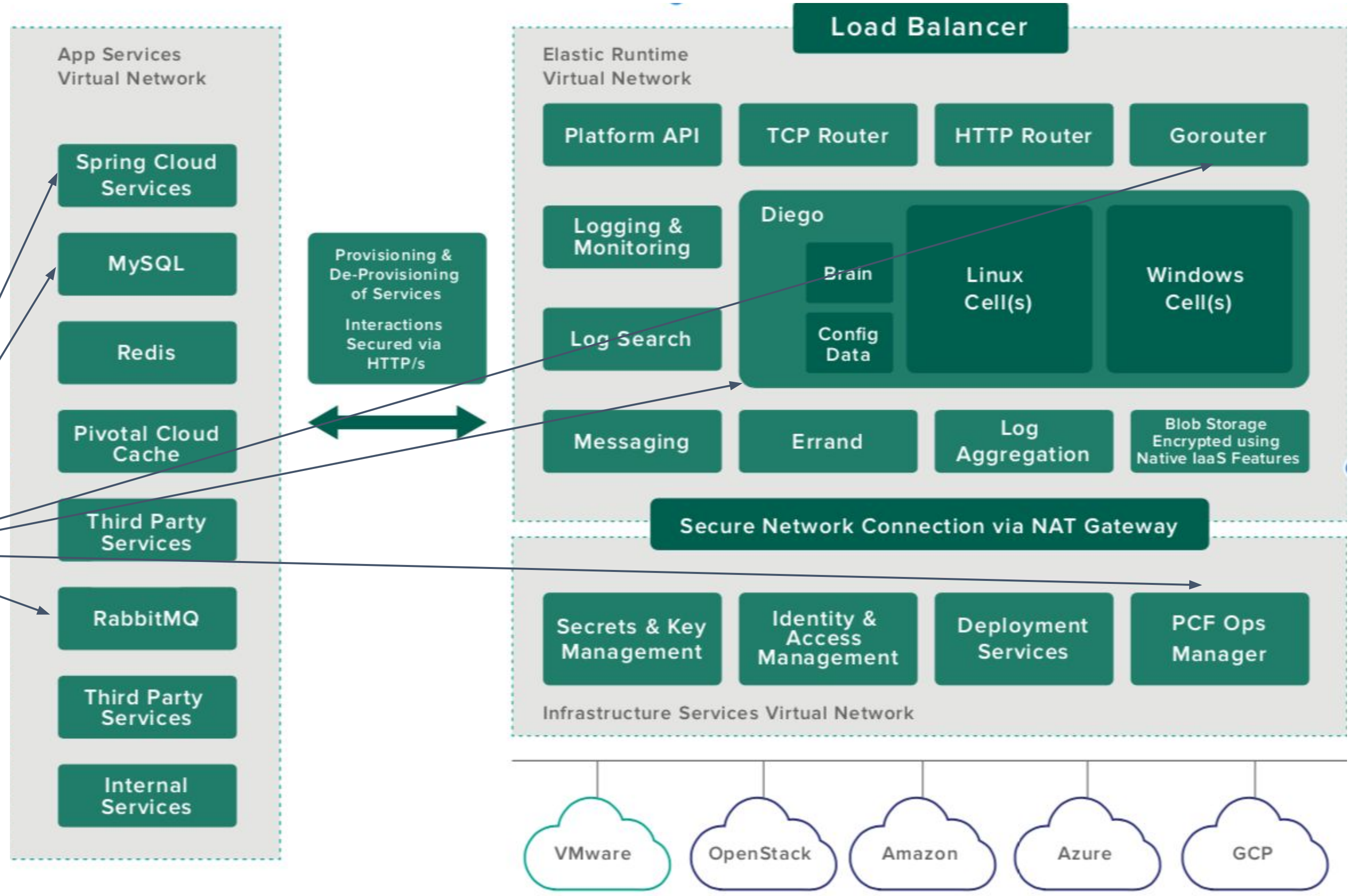
Defending the Continuous Threat

(Advanced Persistent Threat)

phase PHASE 5

Canary Style deployment model plus Infrastructure as code from version control

(Repave all system components)



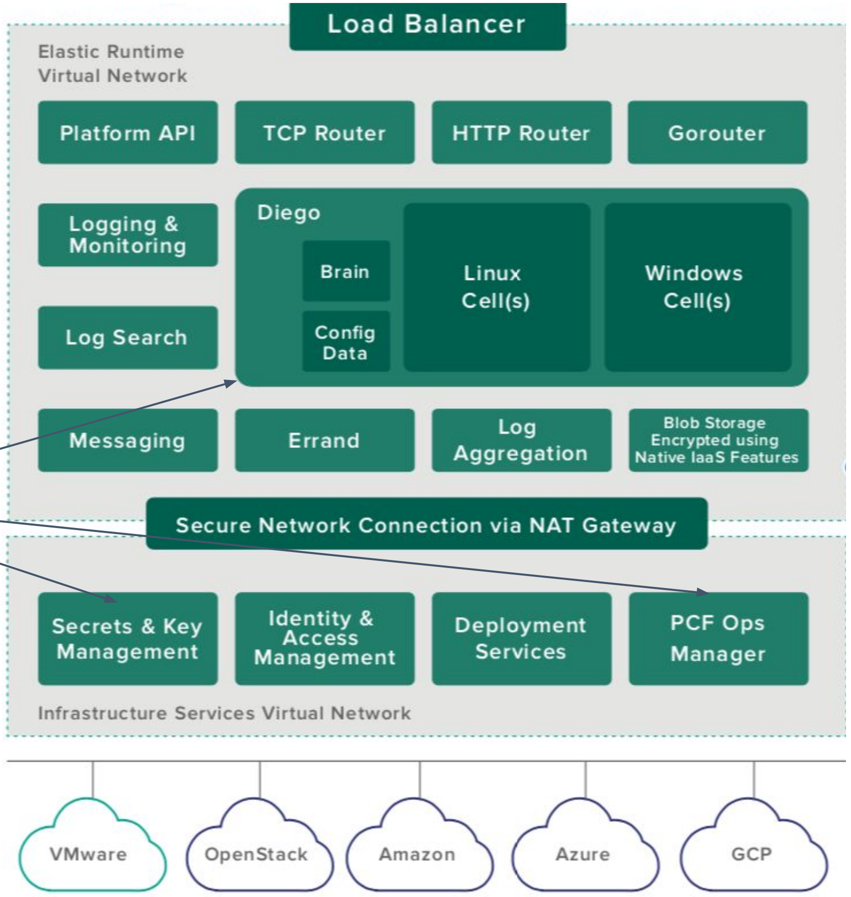
Defending the Continuous Threat

(Leaked Credentials)

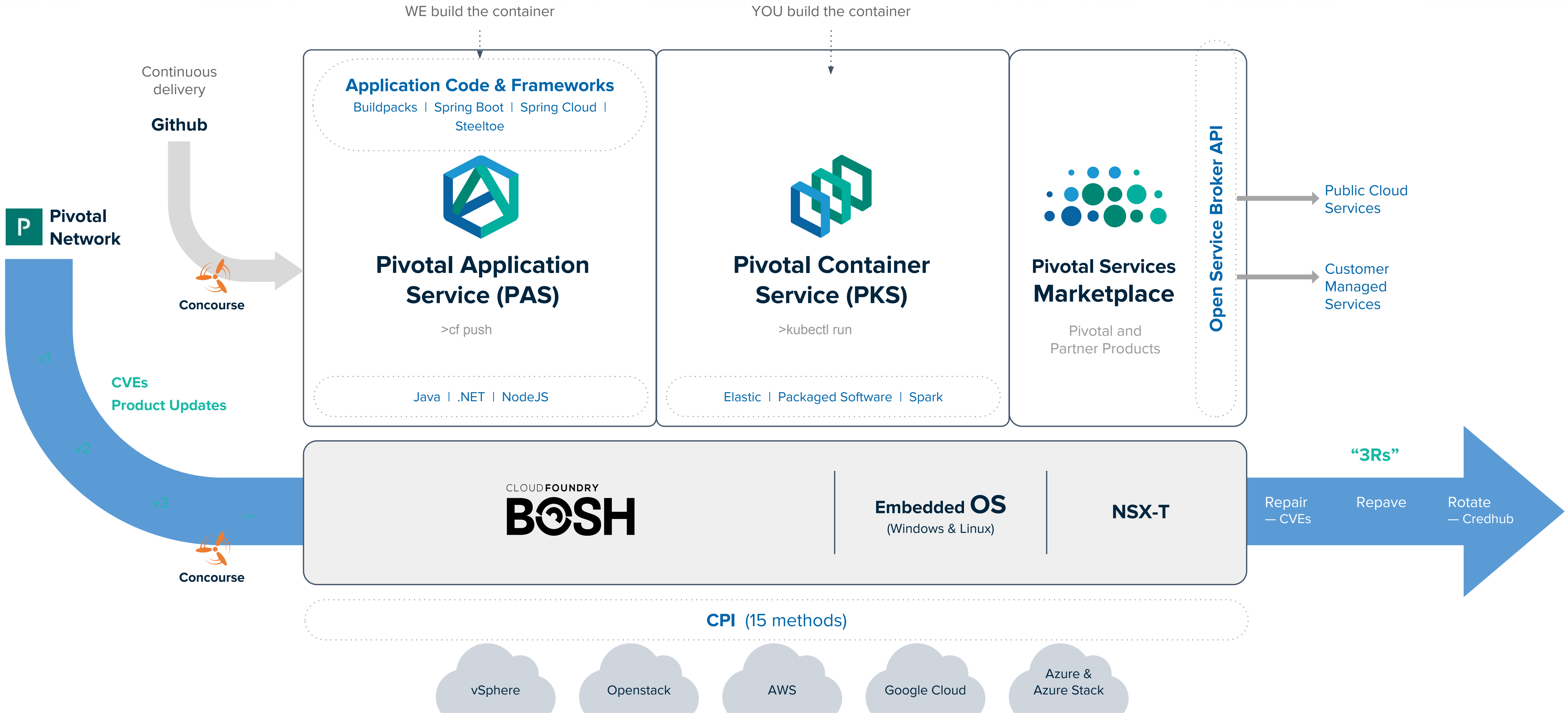
KubeCon | CloudNativeCon
China 2018

phase PHASE 5

Continuous Zero Zero Trust Network Model (Rotation of all system credentials)



Bringing it all together



WE build the container

YOU build the container

Continuous delivery

Github

Pivotal Network

Concourse

v1

CVEs

Product Updates

v2

v3

...

Concourse

Application Code & Frameworks

Buildpacks | Spring Boot | Spring Cloud | Steeltoe

Pivotal Application Service (PAS)

>cf push

Java | .NET | NodeJS

Pivotal Container Service (PKS)

>kubectl run

Elastic | Packaged Software | Spark

Pivotal Services Marketplace

Pivotal and Partner Products

Open Service Broker API

Public Cloud Services

Customer Managed Services

CLOUD FOUNDRY BOSH

Embedded OS (Windows & Linux)

NSX-T

"3Rs"

Repair — CVEs

Repave

Rotate — Credhub

CPI (15 methods)

vSphere

Openstack

AWS

Google Cloud

Azure & Azure Stack

Conclusion - Move The Target

Proactive Security Policy

Phase I - Aggressive Rotation of the issued Developer Keys

Phase II – Rotation of Environment Credentials, End Point IPs, and Dynamic Management of IP WL/ACLs

Phase III - Continuous Verification of the Application Integrity

Phase IV – Continuous Authorization for Runtime Validation

Phase V – Continuous Paving of the Environment, Rotation of the Keys, Renewal of Authorizing Credentials plus Least Privilege Container Authority

KubeCon | CloudNativeCon

China 2018

What Does Bosh Do

CLOUD FOUNDRY
BOSH™

BOSH is an open source tool for release engineering, deployment, lifecycle management, and monitoring of distributed systems such as Kubernetes.

Packaging w/ embedded OS

Server provisioning on any IaaS

Software deployment across availability zones

Health monitoring (server AND processes)

Self-healing w/ Resurrector

Storage management

Rolling upgrades via canaries

Easy scaling of clusters

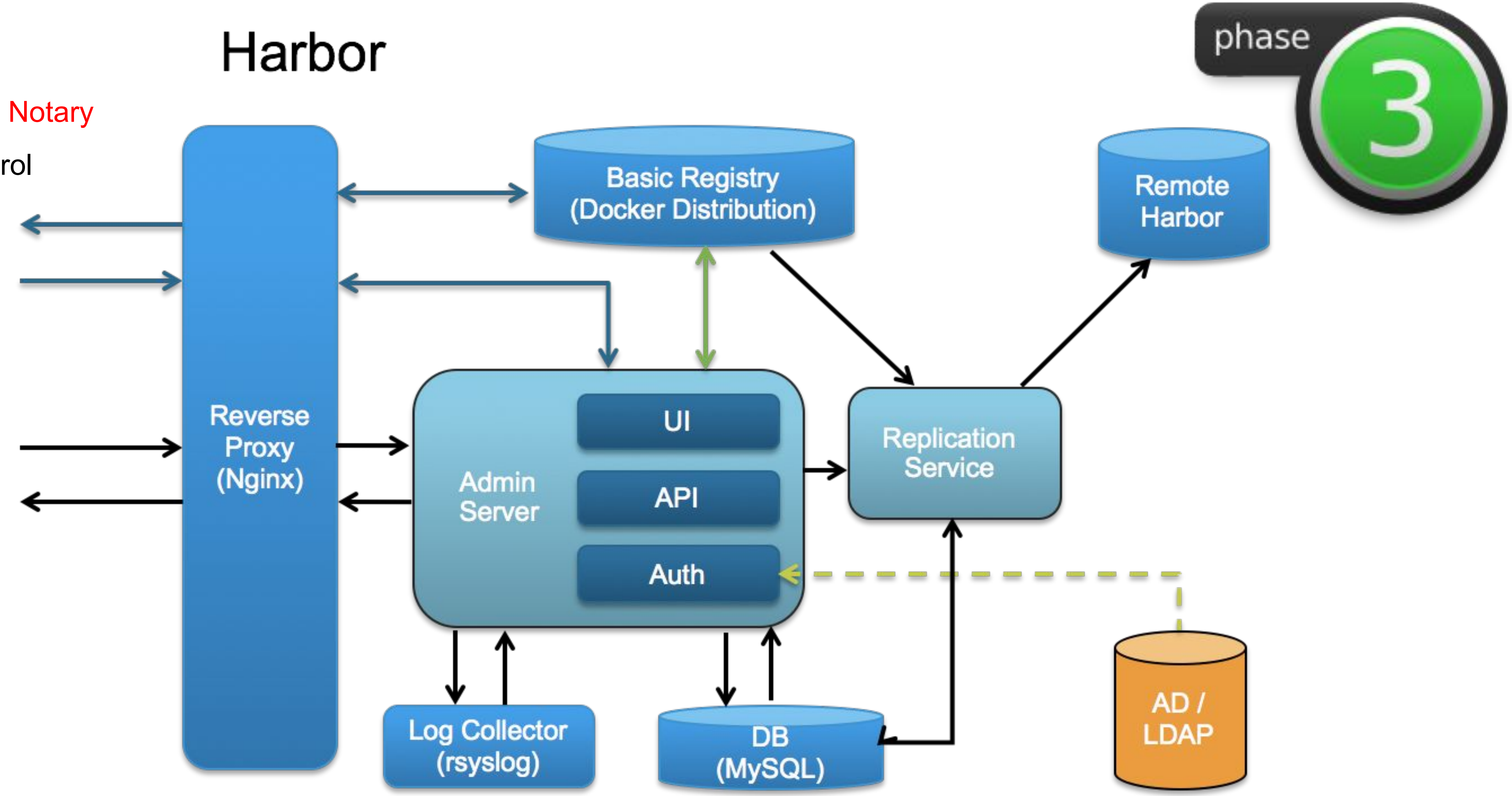
Backup and Restore

Rotating Server Credentials

- Cloud Native Security
  - Repair
  - Repave
  - Rotate

Enterprise Docker Registry - Harbor

- LDAP Integration
- Clair Security Scanning
- Docker Content Trust via Notary
- Role Based Access Control


Harbor
phase 3
Reverse Proxy (Nginx)
Basic Registry (Docker Distribution)
Remote Harbor
Admin Server
UI
API
Auth
Replication Service
Log Collector (rsyslog)
DB (MySQL)
AD / LDAP