SKILUP SM DAYS

by: DevOps INSTITUTE

Testability in the Paradigm of Kubernetes

Essential aspects of Testing in the age of Containers & Kubernetes

Himanshu Patel
Digital Transformation Leader
Aventiv Technologies
NavikHub
HRPatel2000@GMail.com

Agenda

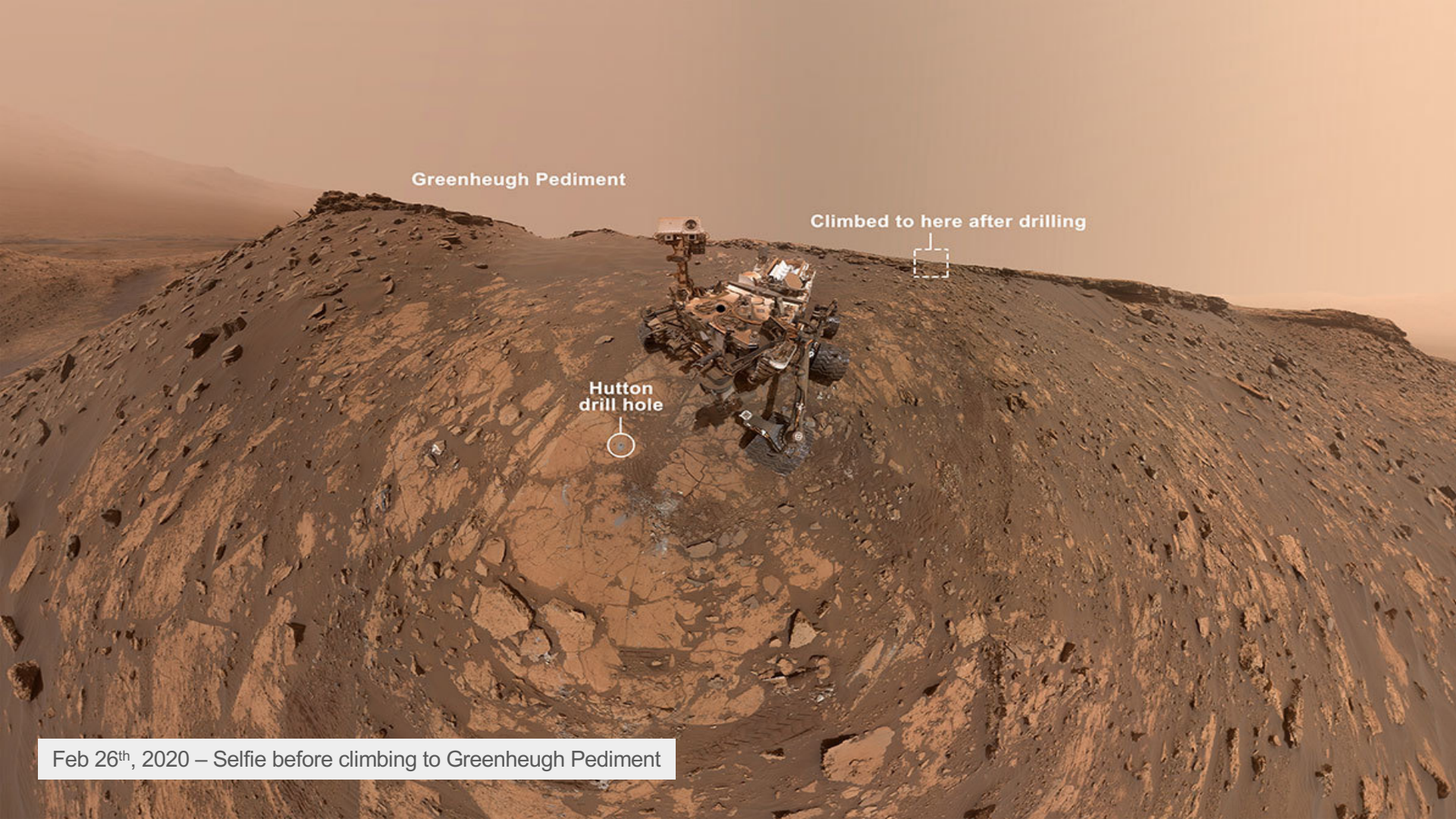
- Context from a Different Angle
- Traditional Lean, Agile, DevOps Values
- Kubernetes Paradigm
- Testability in Kubernetes
- Disposable Environments
- Non-Functional Testing
- Chaos Engineering
- Wrap up!

Himanshu Patel

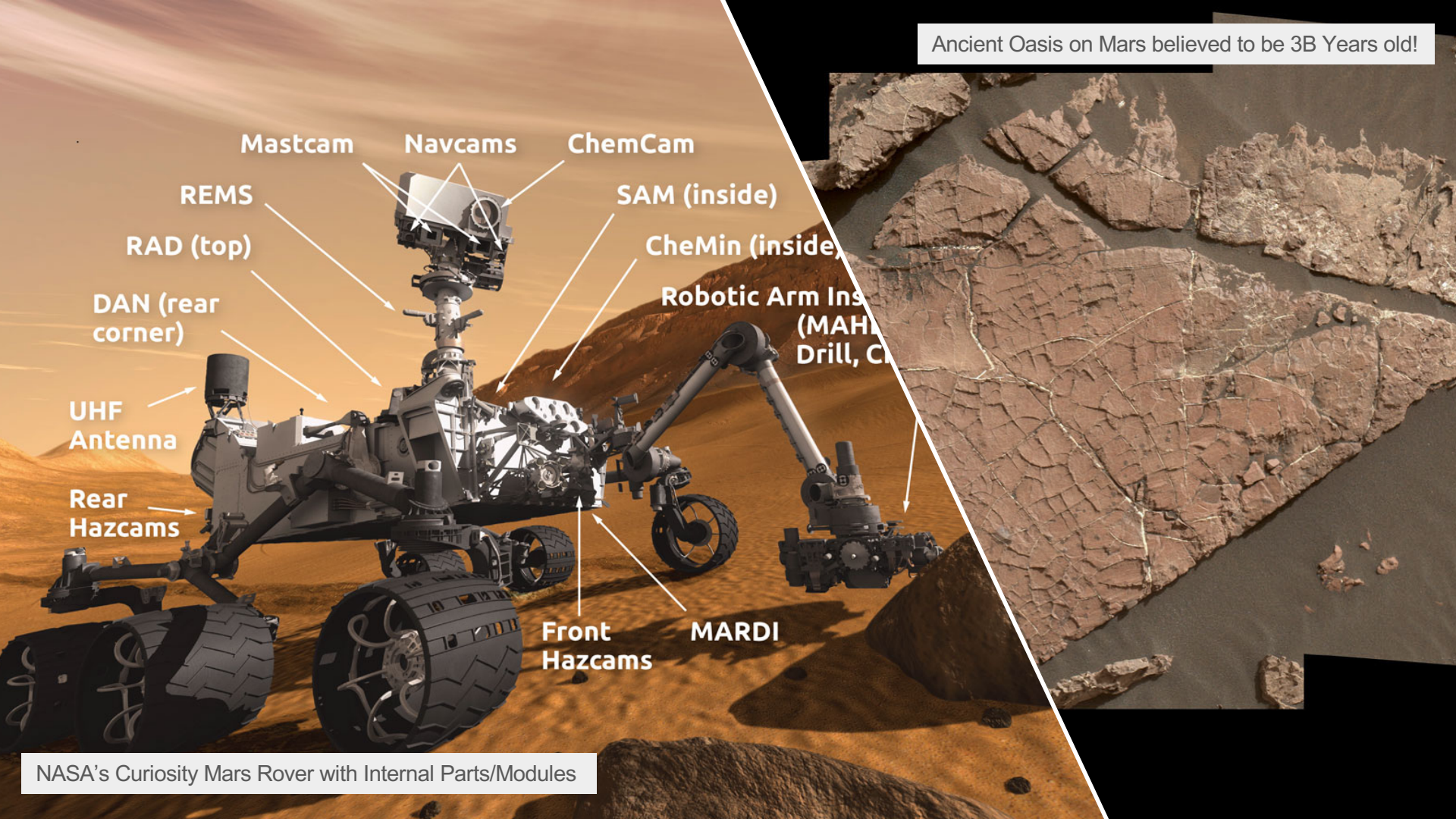
Digital Transformation Leader, DevOps Human & DOI Ambassador

NavikHub

Digital Transformation Leader, Evangelist, Speaker, and Passionate Technologist with a solid breadth and depth, a firm believer in simplistic & sustainable design. Leading Middleware Engineering and DevOps Practices at Aventiv Technologies in Dallas, TX.



Feb 26th, 2020 – Selfie before climbing to Greenheugh Pediment

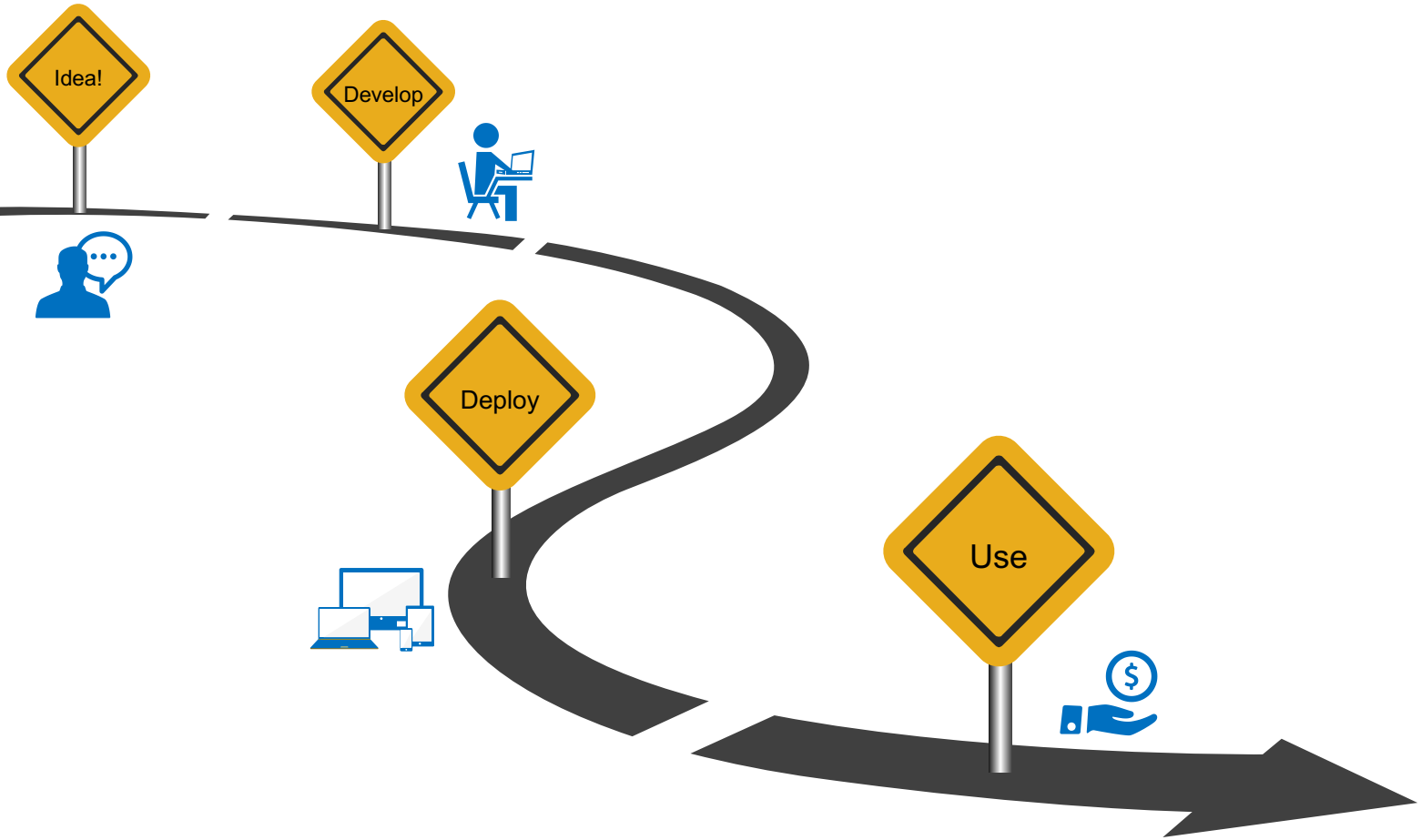


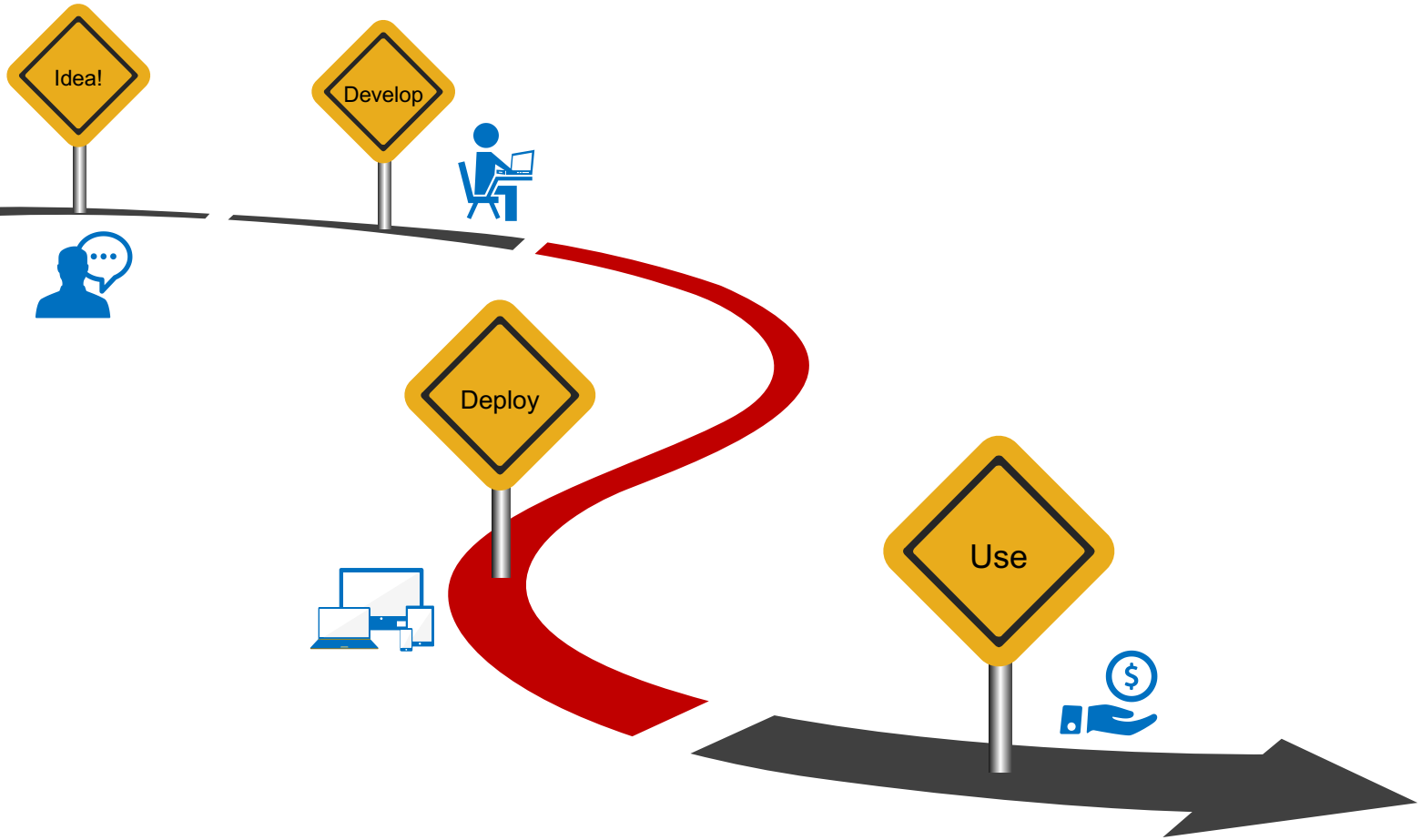
NASA's Curiosity Mars Rover with Internal Parts/Modules

Ancient Oasis on Mars believed to be 3B Years old!

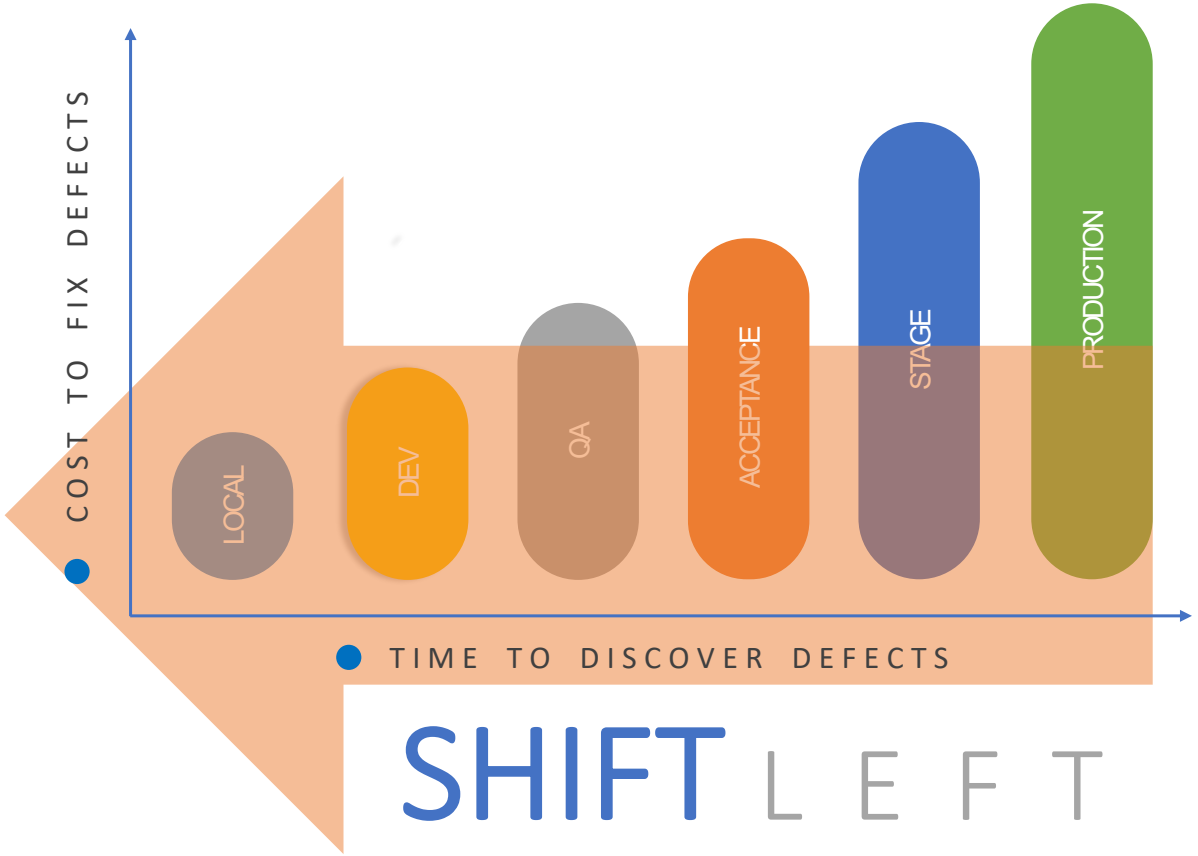
TESTABILITY
IN THE PARADIGM OF CONTAINERS AND KUBERNETES


20K


Idea!
Develop
Deploy
Use

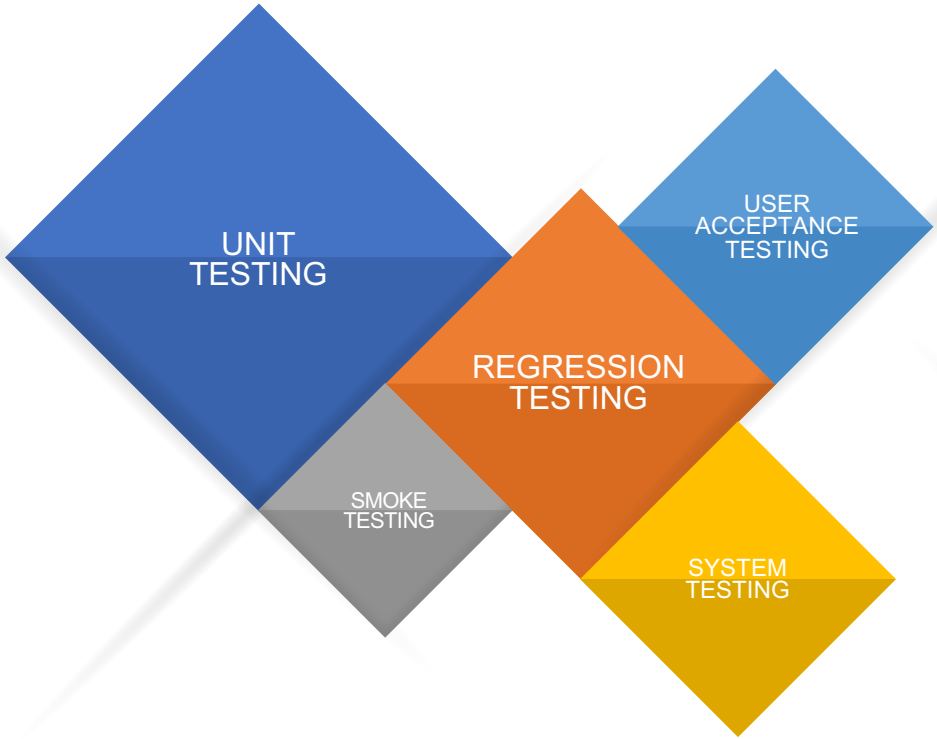

Idea!
Develop
Deploy
Use


Idea!
Develop
Deploy
Use
$



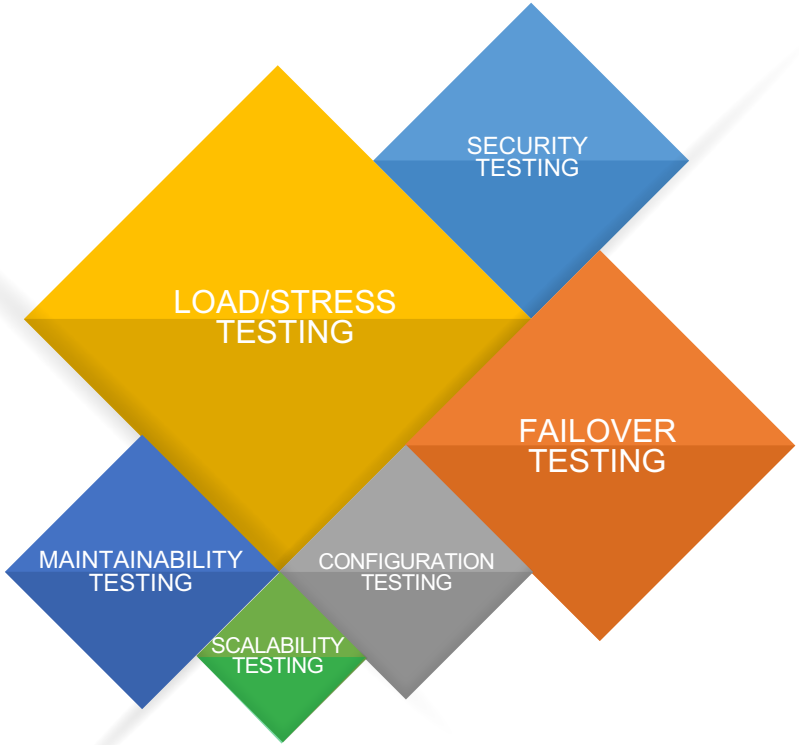
SHIFT LEFT

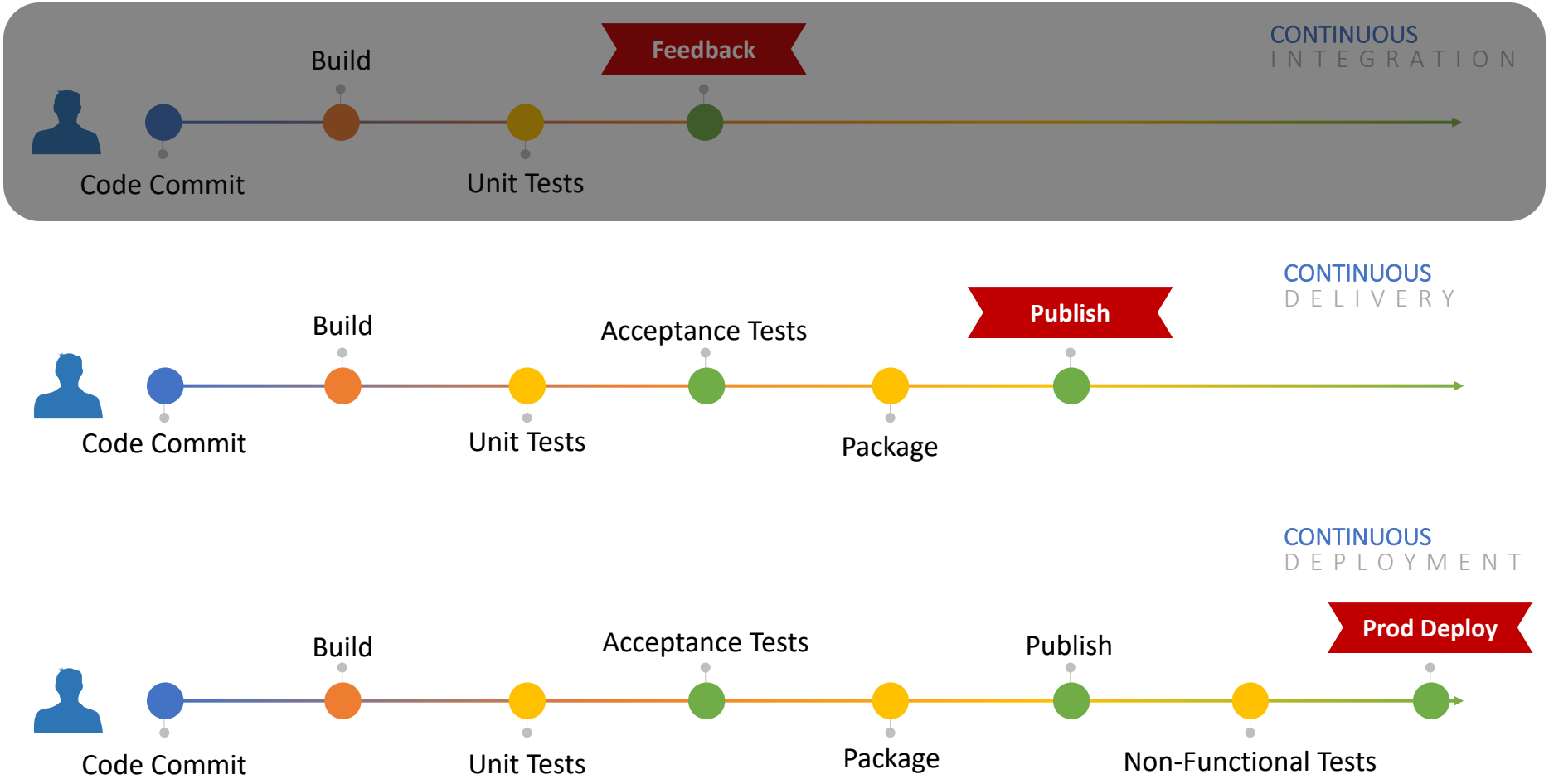
- **EARLY** DETECTION
- **TIME** SAVING
- **COST** SAVING
- **SMOOTH** RELEASES

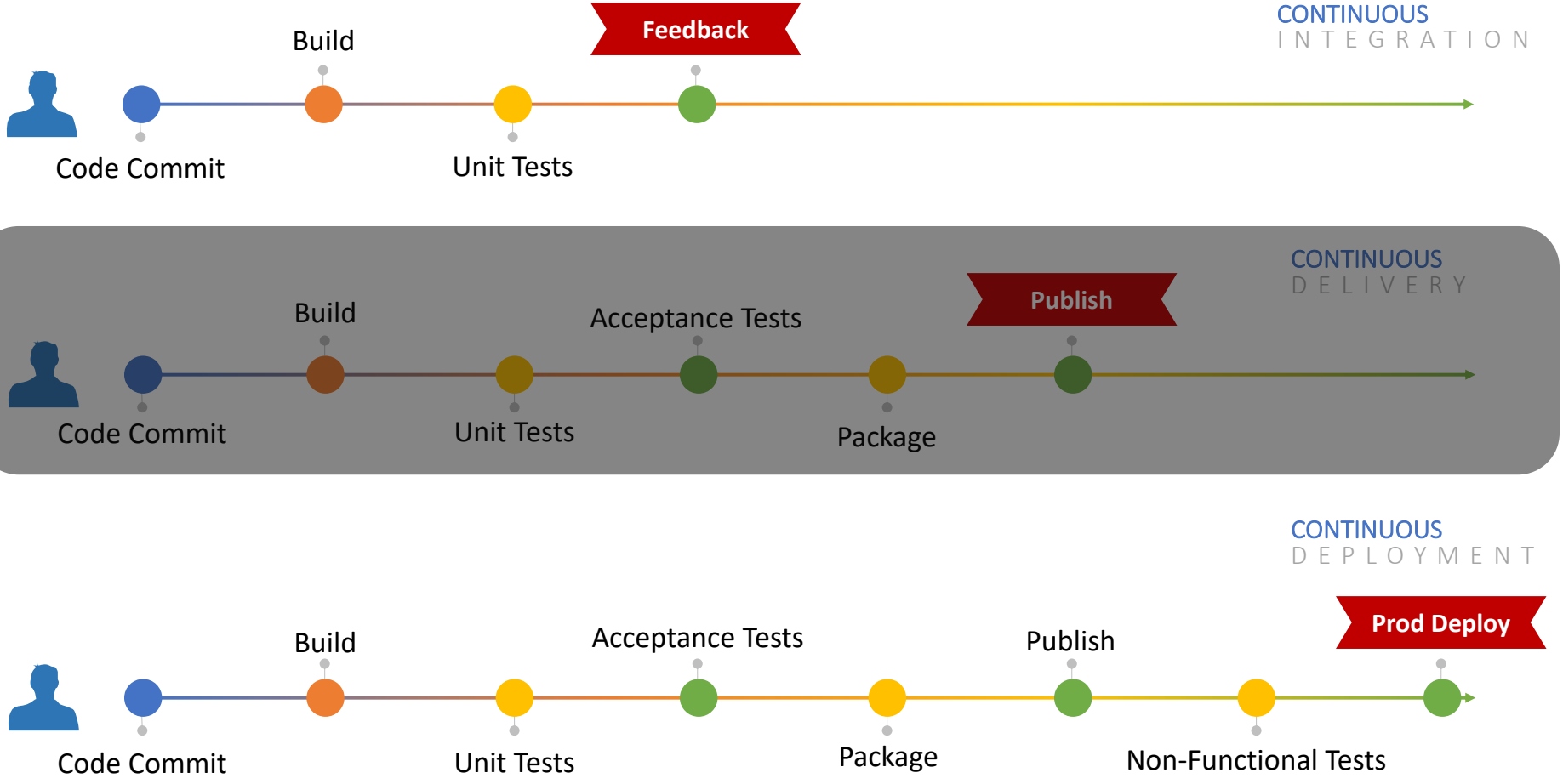


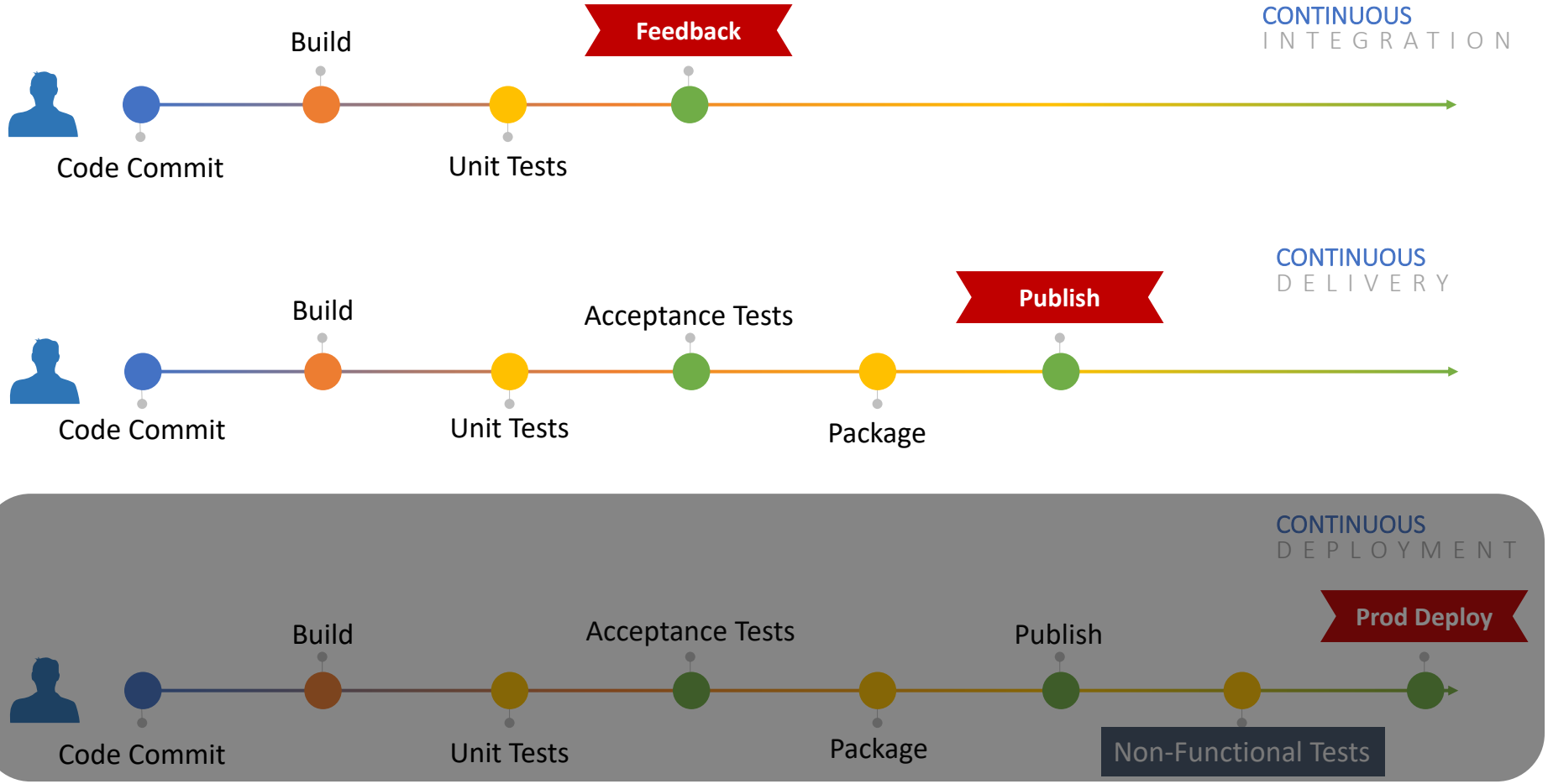
FUNCTIONAL TESTING

NON-FUNCTIONAL TESTING

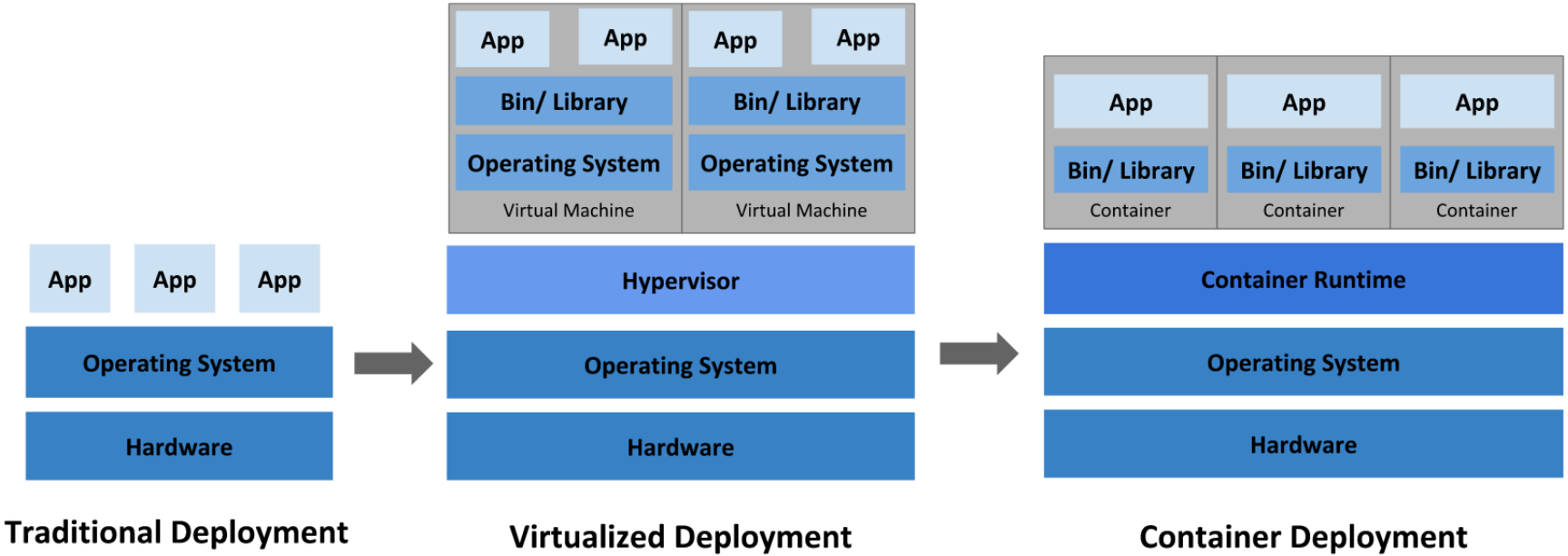


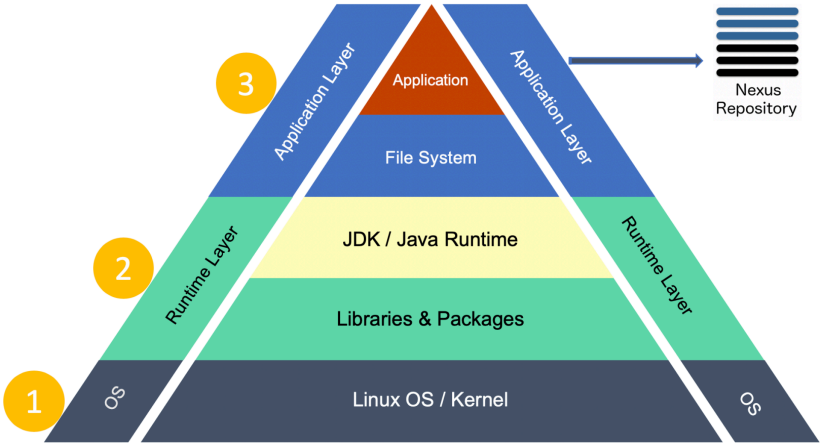

CONTINUOUS
INTEGRATION
Feedback
Build
Code Commit
Unit Tests
CONTINUOUS
DELIVERY
Publish
Build
Acceptance Tests
Code Commit
Unit Tests
Package
CONTINUOUS
DEPLOYMENT
Prod Deploy
Build
Acceptance Tests
Publish
Code Commit
Unit Tests
Package
Non-Functional Tests


CONTINUOUS
INTEGRATION
Feedback
Build
Code Commit
Unit Tests
CONTINUOUS
DELIVERY
Publish
Build
Acceptance Tests
Code Commit
Unit Tests
Package
CONTINUOUS
DEPLOYMENT
Prod Deploy
Build
Acceptance Tests
Publish
Code Commit
Unit Tests
Package
Non-Functional Tests

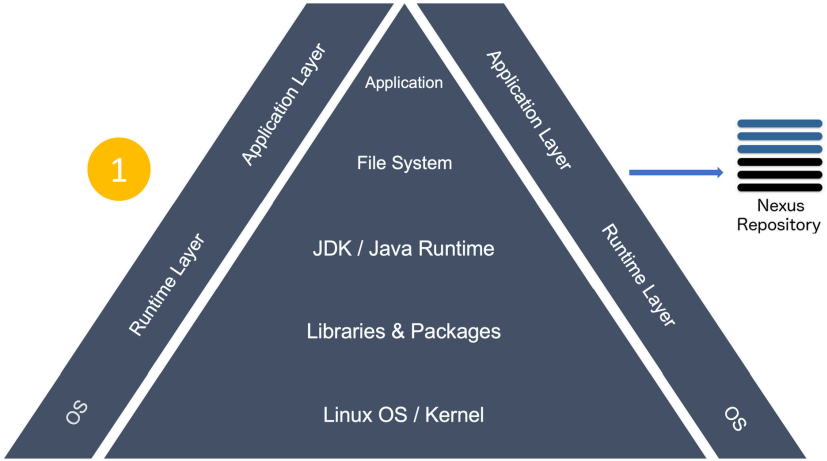

CONTINUOUS
INTEGRATION
Feedback
Build
Code Commit
Unit Tests
CONTINUOUS
DELIVERY
Publish
Build
Acceptance Tests
Code Commit
Unit Tests
Package
CONTINUOUS
DEPLOYMENT
Prod Deploy
Build
Acceptance Tests
Publish
Code Commit
Unit Tests
Package
Non-Functional Tests

EVOLVING DEPLOYMENTS





TRADITIONAL DEPLOYMENT

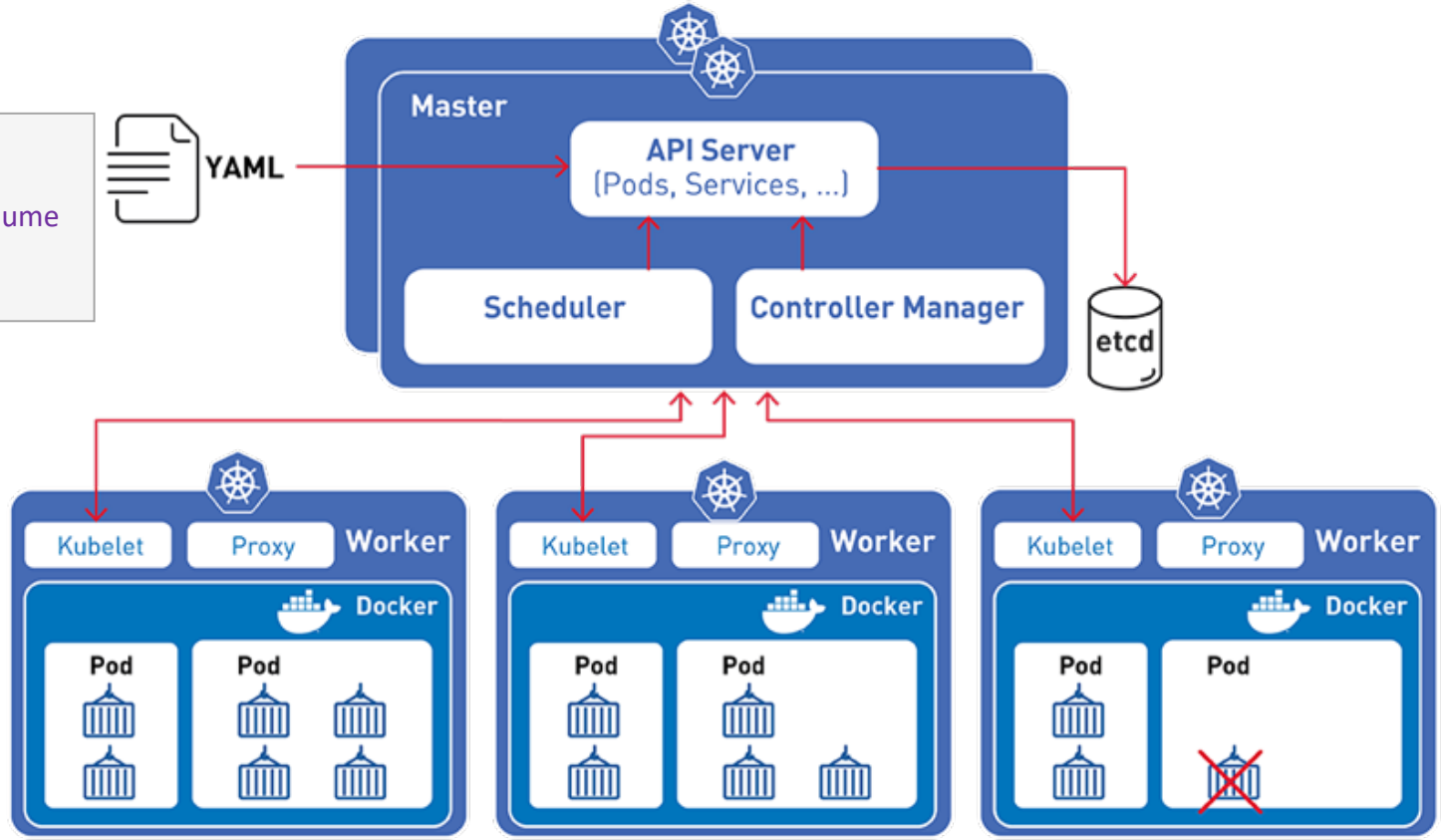


CONTAINERIZED DEPLOYMENT

SKILup DAYS

KUBERNETES ARCHITECTURE

Namespace
Service
Persistent Volume
Deployment
Pod



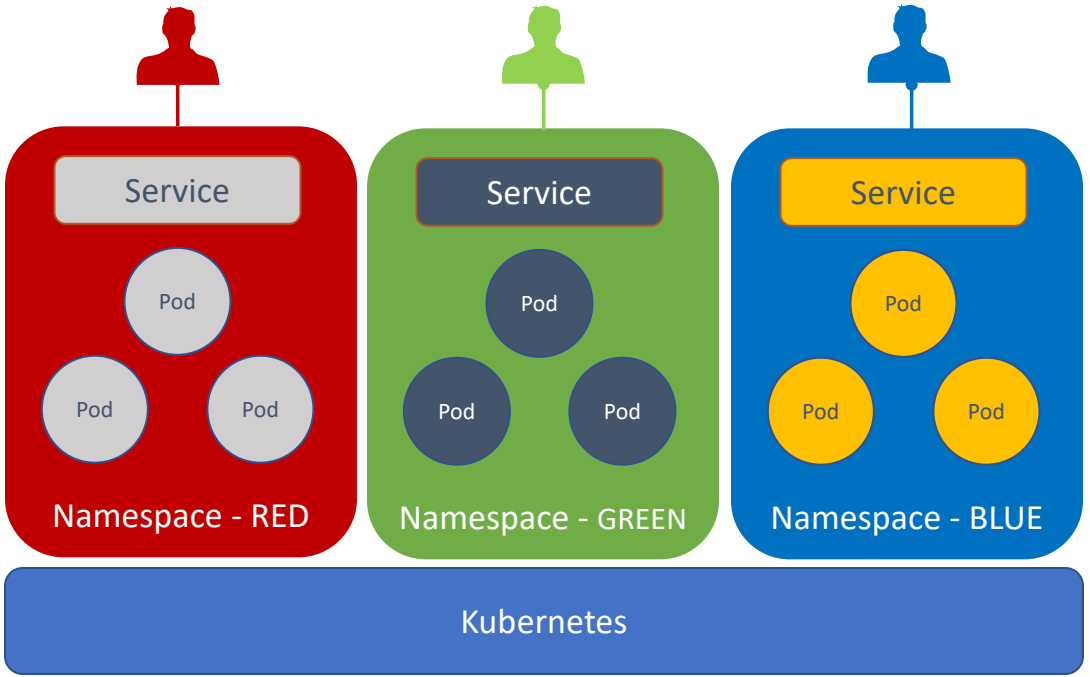
TESTING RESOURCES

TEST CASES / SCENARIOS

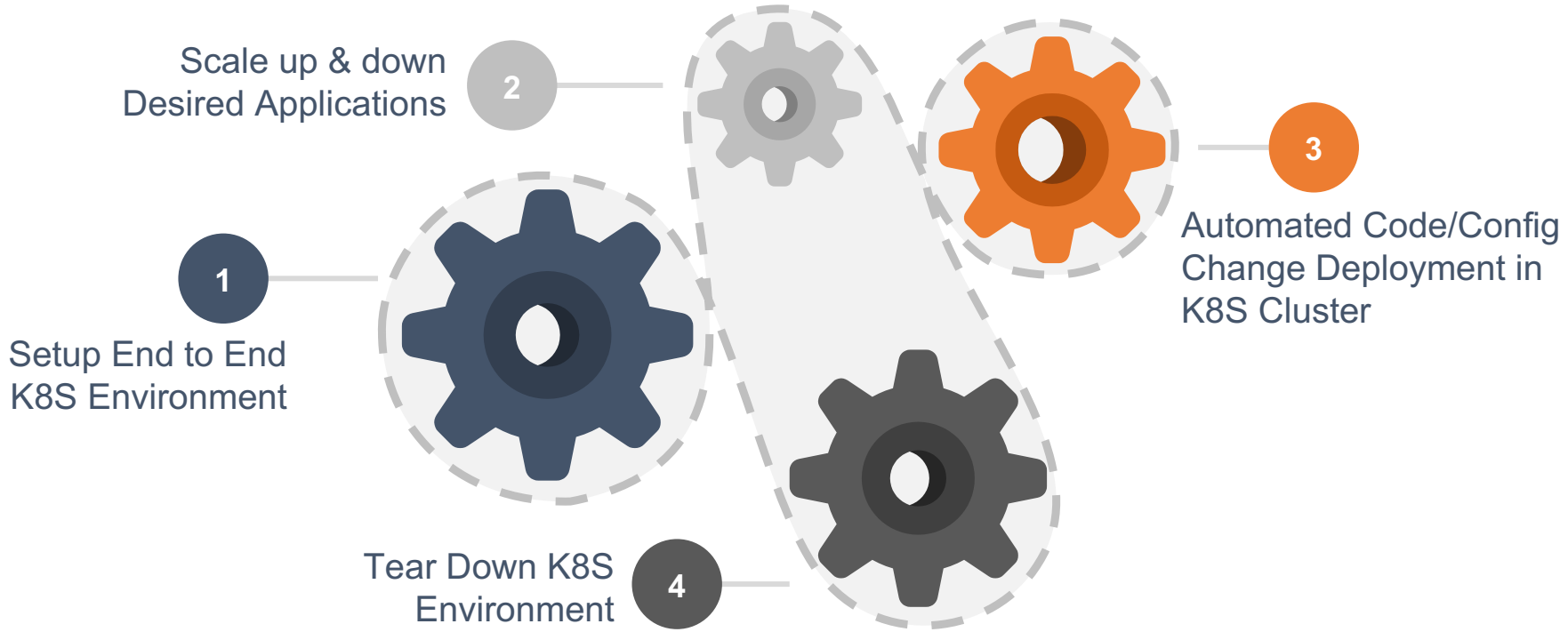
APPLICATION

CONFIGURATION

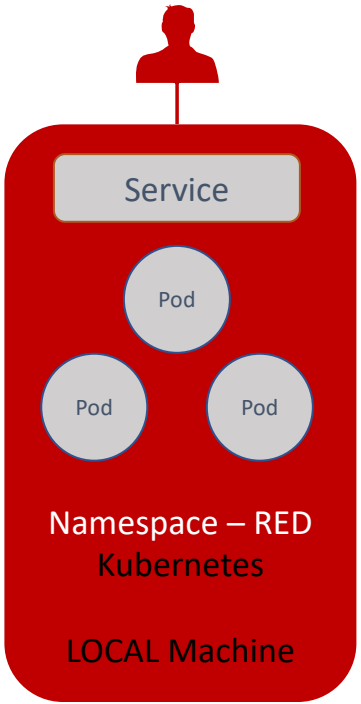
ENVIRONMENT



DEVOPS CI/CD AUTOMATION



SHIFT L E F T



K3S

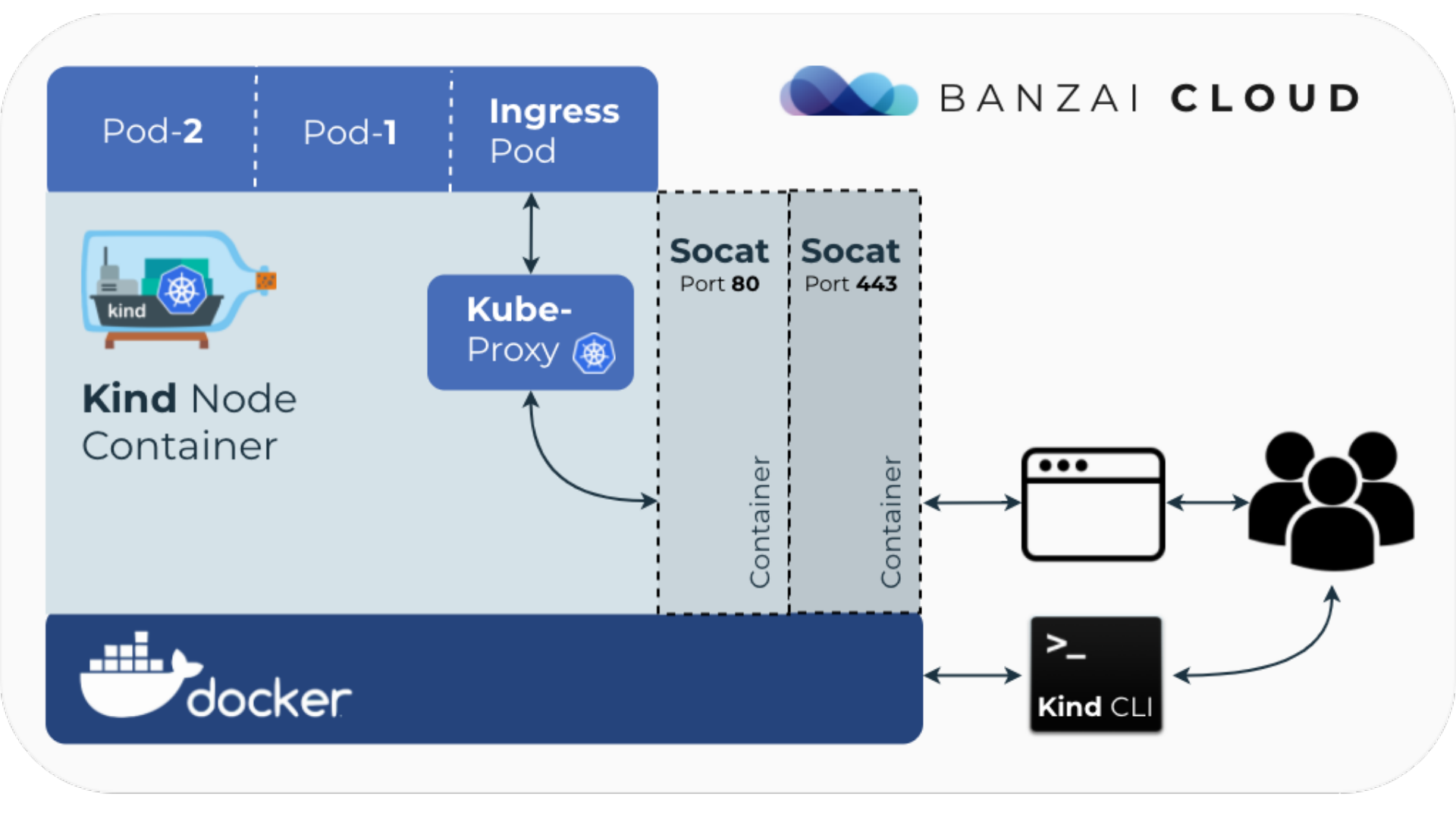
docker

kind

kubernetes

MicroK8s

minikube


Pod-2
Pod-1
Ingress Pod
BANZAI CLOUD
kind
Kube-Proxy
Socat Port 80
Socat Port 443
Container
Container
Kind Node Container
docker
>_
Kind CLI

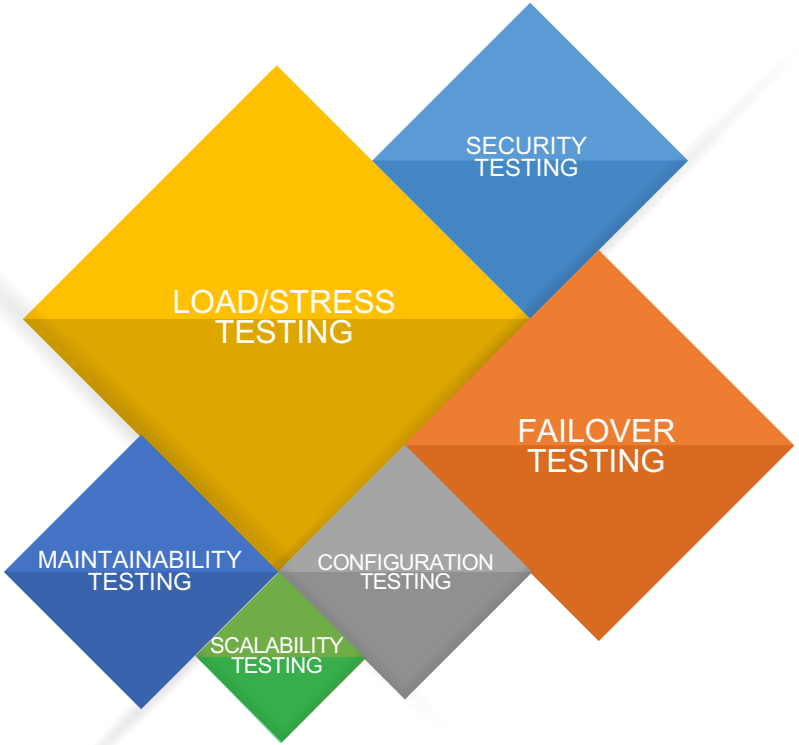
LOCAL SETUP

- What is it?
  - Docker
  - Kind (Kubectl CLI)
  - Docker Registry
  - K8Dash Control Plane
  - CI Server (Jenkins)

- Why K8S Cluster on LOCAL?
  - Multi Cluster Management
  - All K8S Resources Supported
  - Local Docker Registry
  - CLI and Dashboard Access
  - Quick Setup & Teardown
  - SHIFT LEFT

TRY OUT https://github.com/navikco/

NON-FUNCTIONAL TESTING

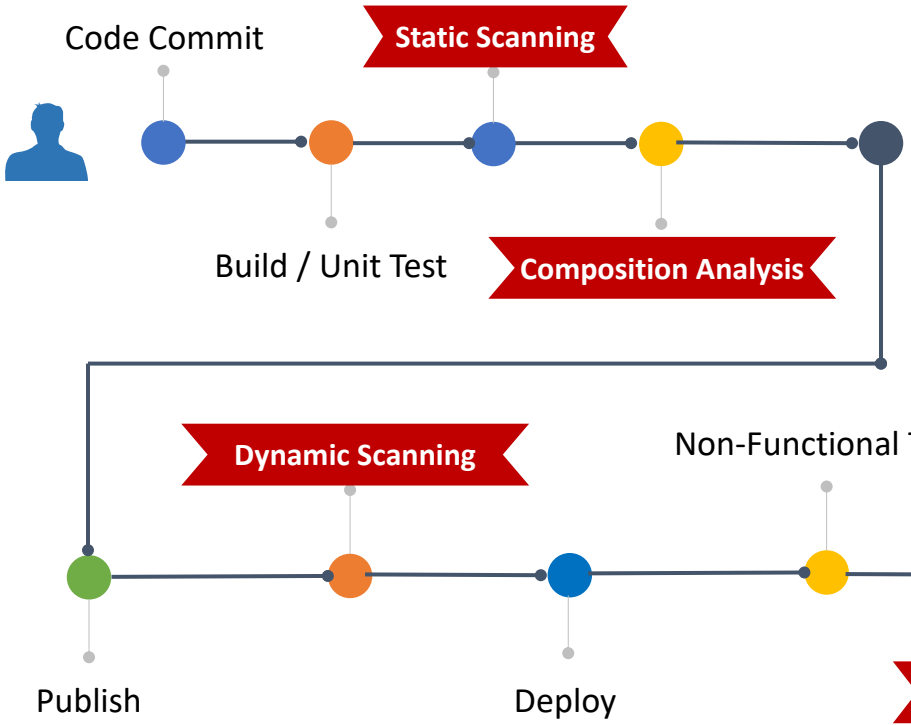


Improved Security & Reduced Risk

High Stability & Site Reliability

Cost & Time Savings

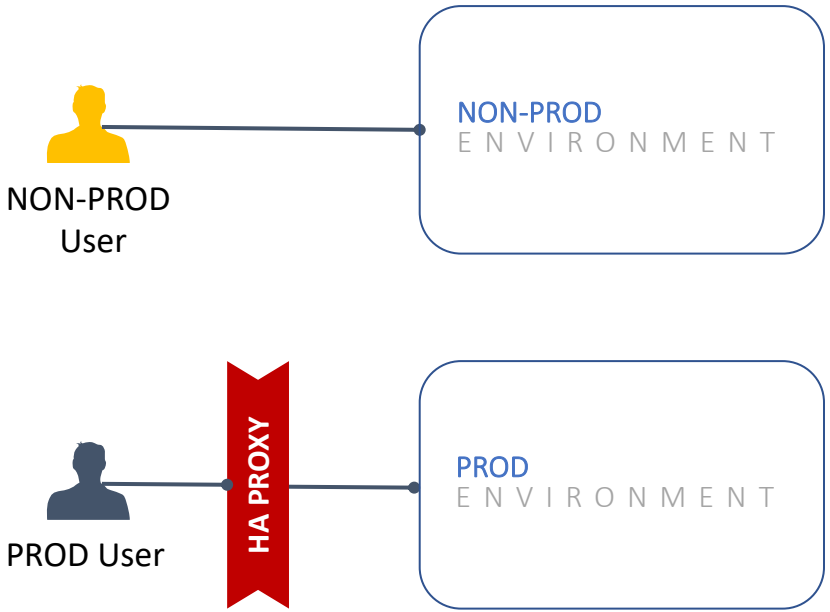
SECURITY TESTING



- Security Testing
  - Rootless Containers
  - Static Scanning
  - Composition Analysis
  - Dynamic Scanning
  - Pen Testing

SECURITY AS PART OF CI/CD DELIVERY PIPELINE

DEALING WITH FAILURES



- Failover Testing
  - Identification, Experimentation, Measurement & Remediation of Failures
  - Identical Non-PROD & PROD Environments

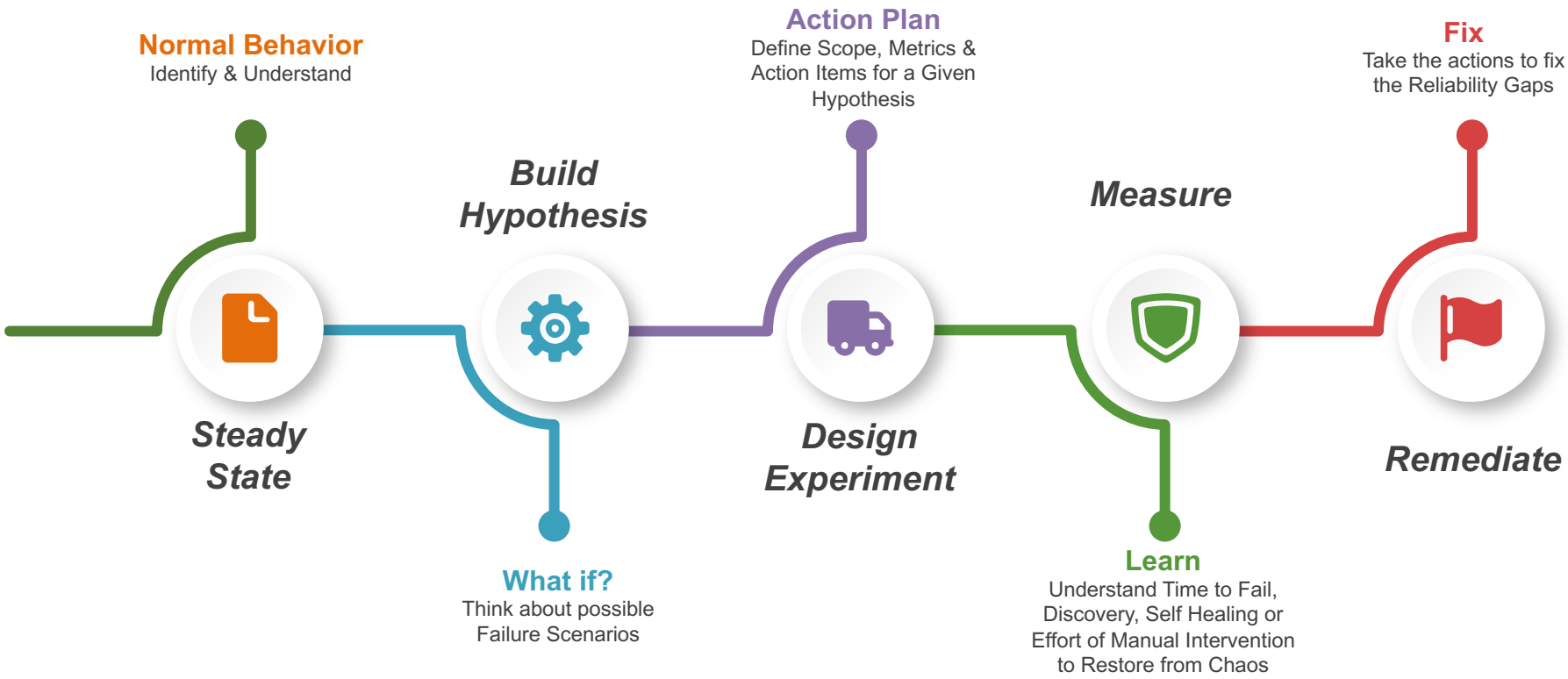
EPHEMERAL ENVIRONMENTS BUILT WITH SAME PRODUCTION MANIFEST

CHAOS ENGINEERING

- Failures are given and everything will eventually fail over time
  - CTO Amazon.com
- Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's capability to withstand turbulent conditions in production
- Bad things will happen to your system, no matter how well designed It is. You cannot become ignorant to it
- Chaos doesn't cause problems. It reveals them.
  - Netflix

CHAOS ENGINEERING IS A PRACTICE TO IMPROVE SYSTEM RESILIENCY & CONFIDENCE

CHAOS ENGINEERING CYCLE



FAILURE INJECTION

Common Failures

- **Host Failure**
- **Block DNS**
- **Resource Attack**
- **Traffic Spikes**
- **Dependency Failure**

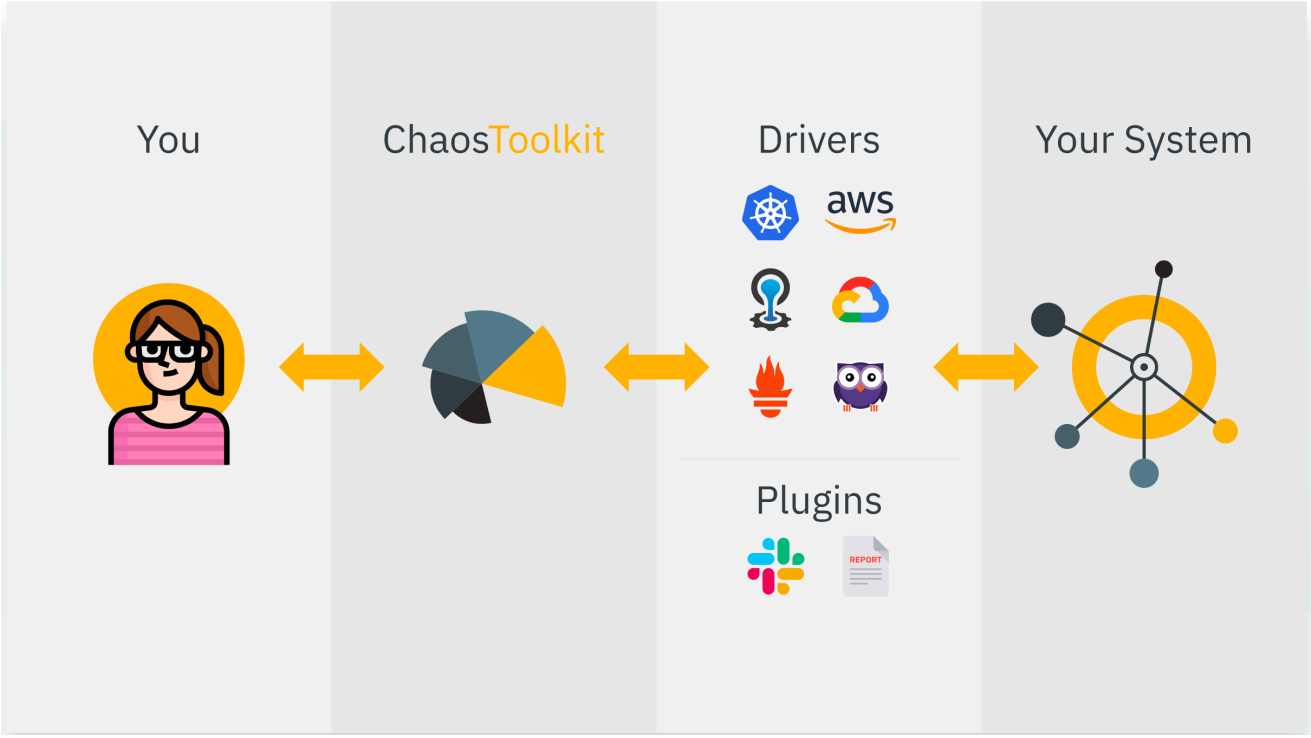
What if?

- Service returns 404, 503?
- Latency Triples?
- Port Inaccessible?
- DB or External System Unavailable?
- Volume Increases Four-Fold?
- IP Tables Wiped out?

Measure

- Time to Detect
- Time to Notification
- Time to Graceful Shutdown
- Time to Partial, Full Auto-Heal or Recovery

CHAOS ENGINEERING TOOLS



- Several Commercial & OSS Options
- Should be Declarative, Extensible & Automated
- Start Small & Build Confidence
- Experiment across Levels,
  - Application
  - Caching
  - Database
  - Network

CHAOS ENGINEERING AS PART OF CI/CD DELIVERY PIPELINE

KUBERNETES TESTABILITY

- Pod Validation
- Rootless Container Validation
- Liveness & Readiness Probs
- Disposable Environments
- Identical Environments for PROD & QA
- Security Testing
- CHAOS ENGINEERING

AUTOMATE N/F TESTS UNDER CI/CD DELIVERY PIPELINE

Oct 11th, 2019 – Another Selfie

THANK YOU!

Meet me in the Network Chat Lounge for questions

SKILUP DAYS