Implementing CI/CD for large scale micro service based programs with Jenkins pipeline

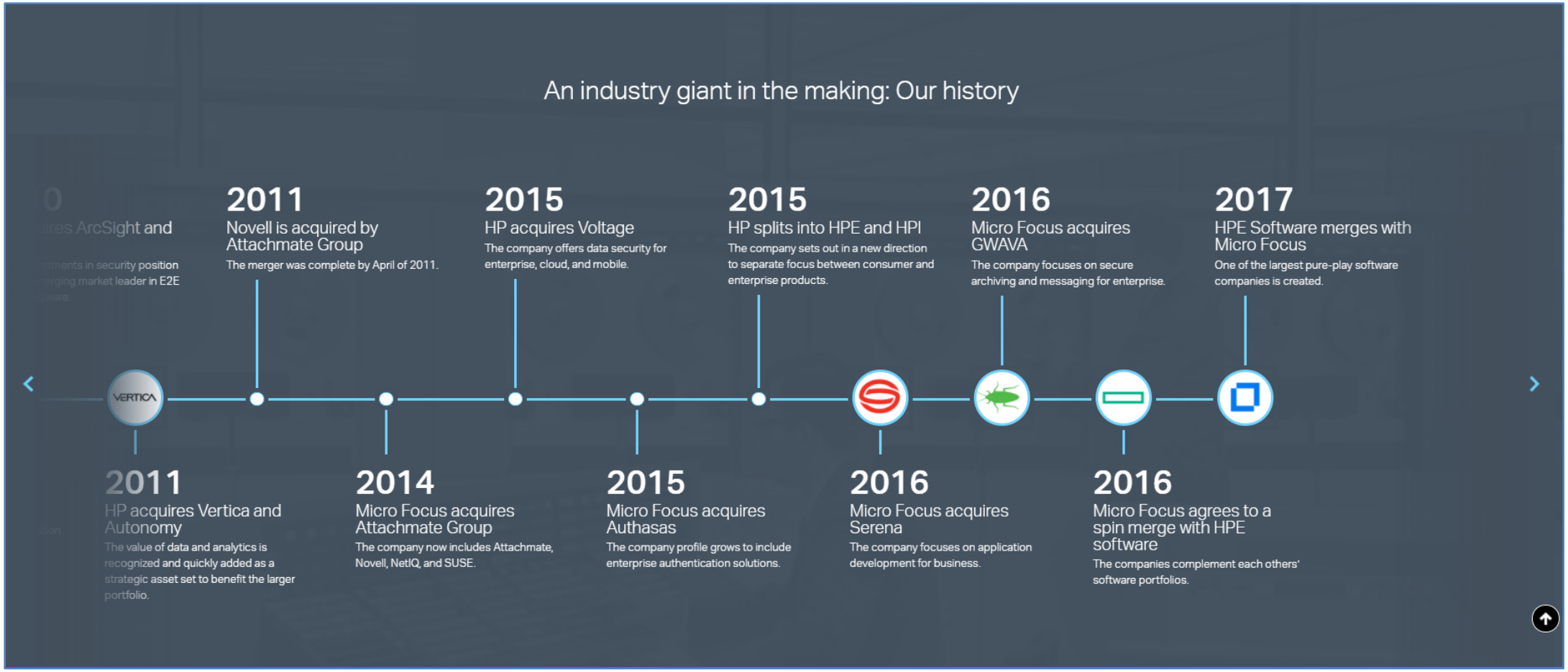
Lin Li (Lily)

2018.06

AGENDA

- Agile and DevOps Transformation
- Why Jenkins Pipeline
- Implementation CI/CD with Jenkins Pipeline
- Q&A

Who are we: since 1939
Build, Operate, & Secure Enterprise Software



We build enterprise-grade scalable software with analytics built in

Hybrid IT

You need to bridge the gap between legacy infrastructure and the digital enterprise. Micro Focus solutions are easy to consume and deploy in any environment, reducing IT costs and time to value. We harden the latest technologies to make them work for you.

DevOps

As a modern DevOps enterprise, you need to accelerate time to market and increase quality. Micro Focus provides an easy-to-use, seamless tool set that scales across the SDLC. From on-premises to cloud and from mainframe to mobile, we address all your DevOps needs.

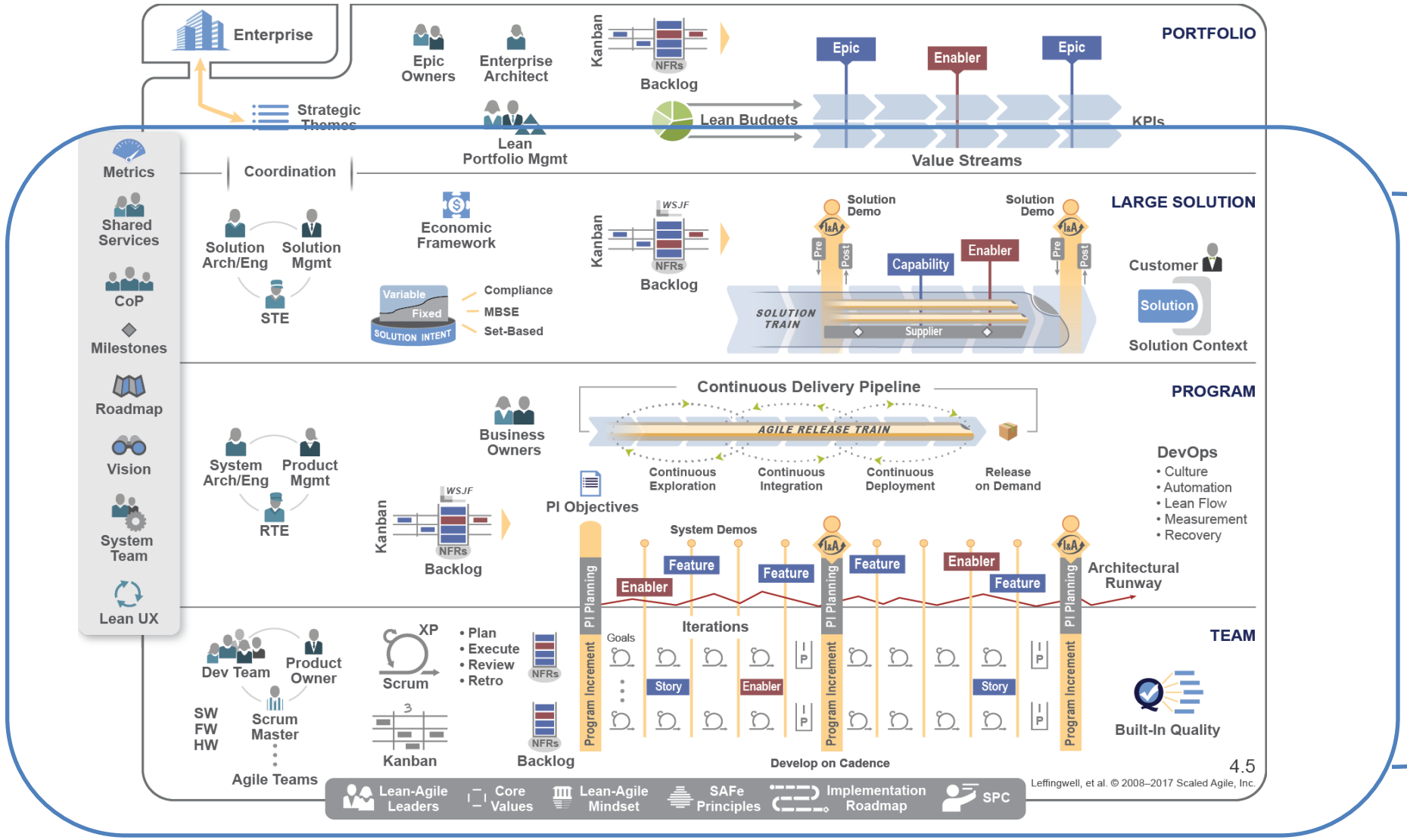
Security & Risk

From compliance issues to the most advanced cyber threats, you need to safeguard your enterprise. Micro Focus protects what you value most: users, data, and applications.

Predictive Analytics

Everyone has data, everyone has analytics, but that's not enough. With predictive and proactive analytics, Micro Focus helps you to not only deliver insights, but drive greater intelligence and productivity across your enterprise.

DevOps transformation alignment with Scaled Agile Framework



Powered by CI/CD

DevOps transformation: impacts the R&D & DevOps practices and operation

Suite Strategy

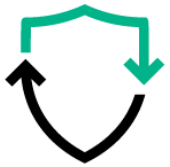
Agile Practices

Containers Microservices

Cross Suite Execution Dependencies

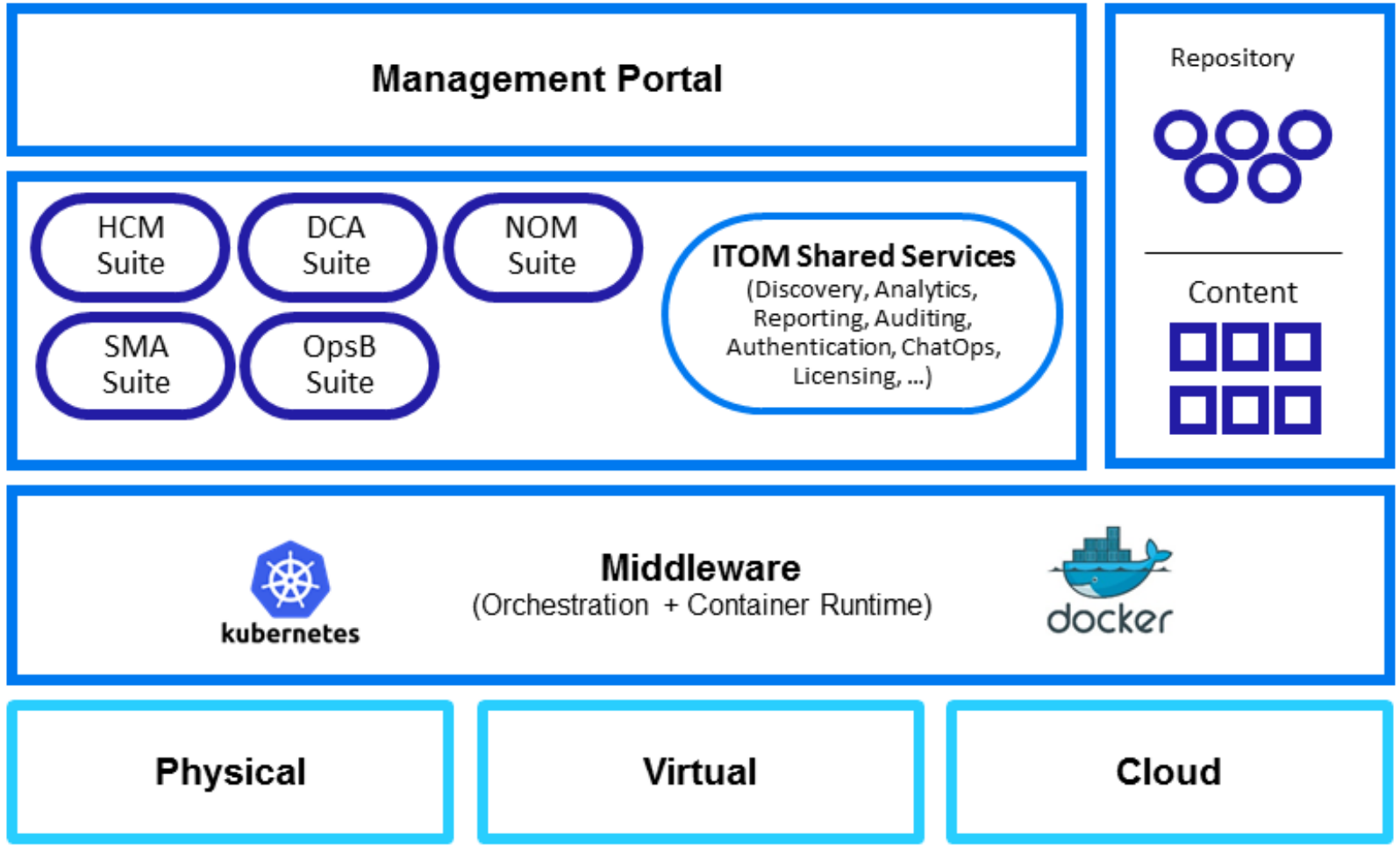
Continuous Integration Delivery & Demo

Standards for Security, Quality & Compliance

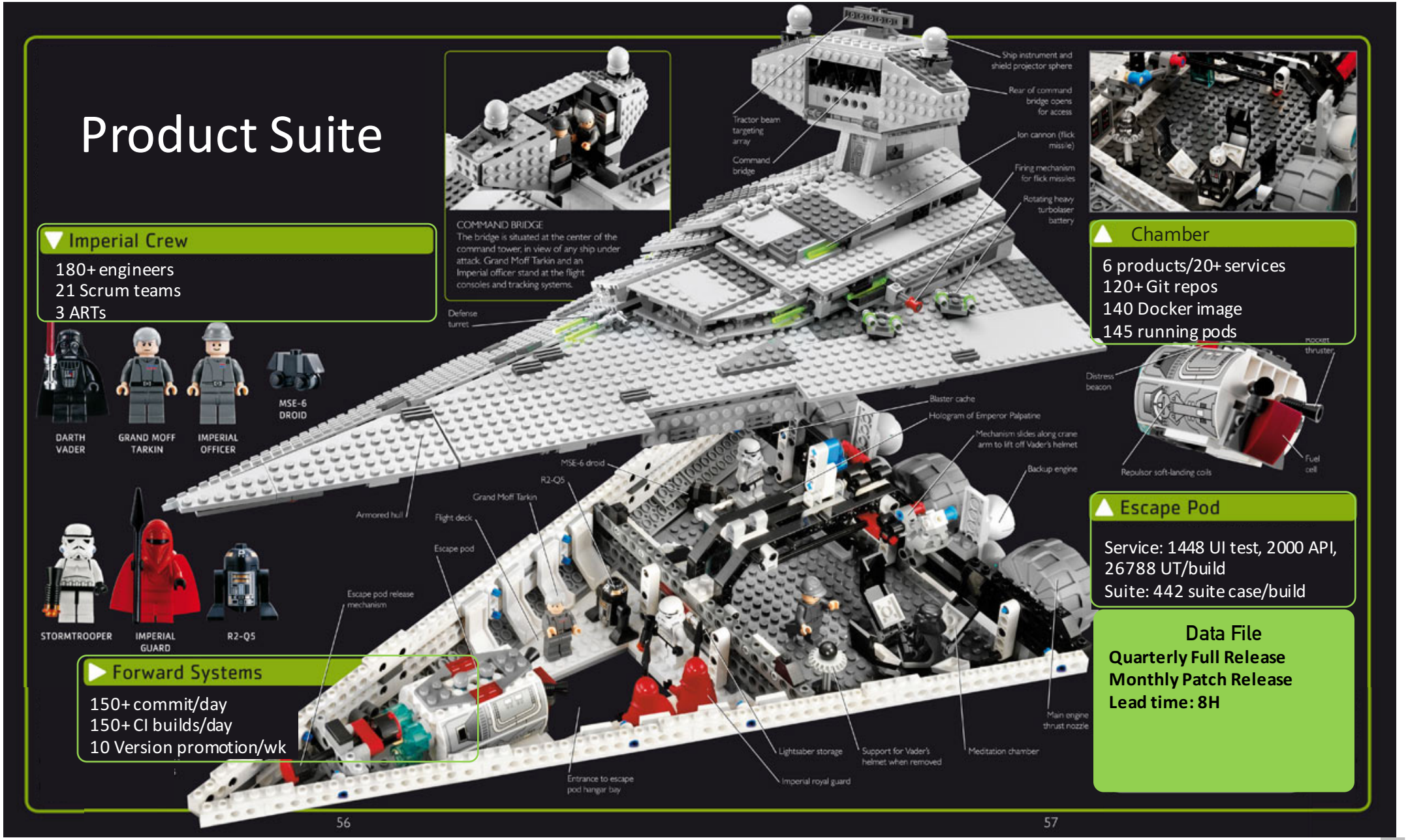


Mission

Deliver out of box solution of micro service based product suites running on top of Kubernetes on premise, public cloud or SaaS to customers and partners



Product Suite



AGENDA

- Agile and DevOps Transformation
- Why Jenkins Pipeline
- Implementation CI/CD with Jenkins Pipeline
- Q&A

Why still Jenkins

Oops!

Travis CI

circleci

GitLab CI

Jenkins penetrates DevOps Practices and Capabilities

Powered by Jenkins

Continuous Planning and Collaboration	Configuration Management	Continuous Integration	Continuous Testing	Continuous Delivery & Deployment	Continuous Improvement
Agile Release Planning	Artifact Warehouse	SCM and Collaboration Frameworks	Code Level Testing	Release Management & Promotion	Quality and Post-Release KPIs
Backlog Management	Configuration Database	Build, Package & Integrate	Build Acceptance Testing	Delivery Sites	CPE & Defect Management
Agile Adoption & Execution KPIs	Artifact API & Catalogue	Infrastructure Provisioning & Workflows	Integration Testing	Demo & SaaS Platforms	Agile DevOps Engineering

Capabilities

Practices

Orchestration of the DevOps toolchain

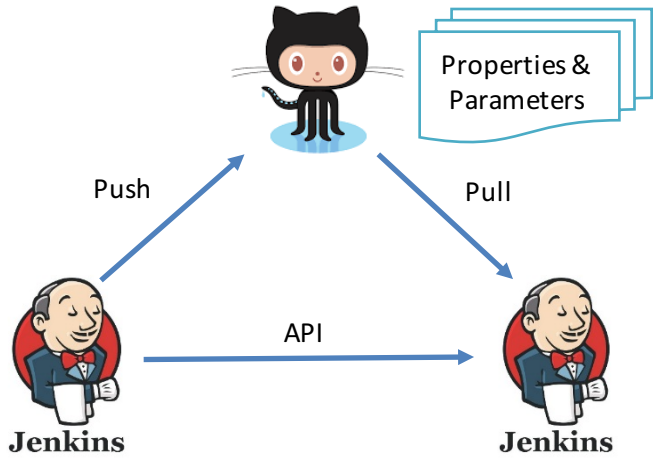
Transform to Jenkins Pipeline

- Scenarios
  - Orchestrate complex process
  - Standardize simple process
- Make everything as code
- Programmer friendly
- Reusable
  - Load from local shared file
  - Shared library
  - Apply from URL
  - Template
- Scripted Pipeline vs Declarative Pipeline

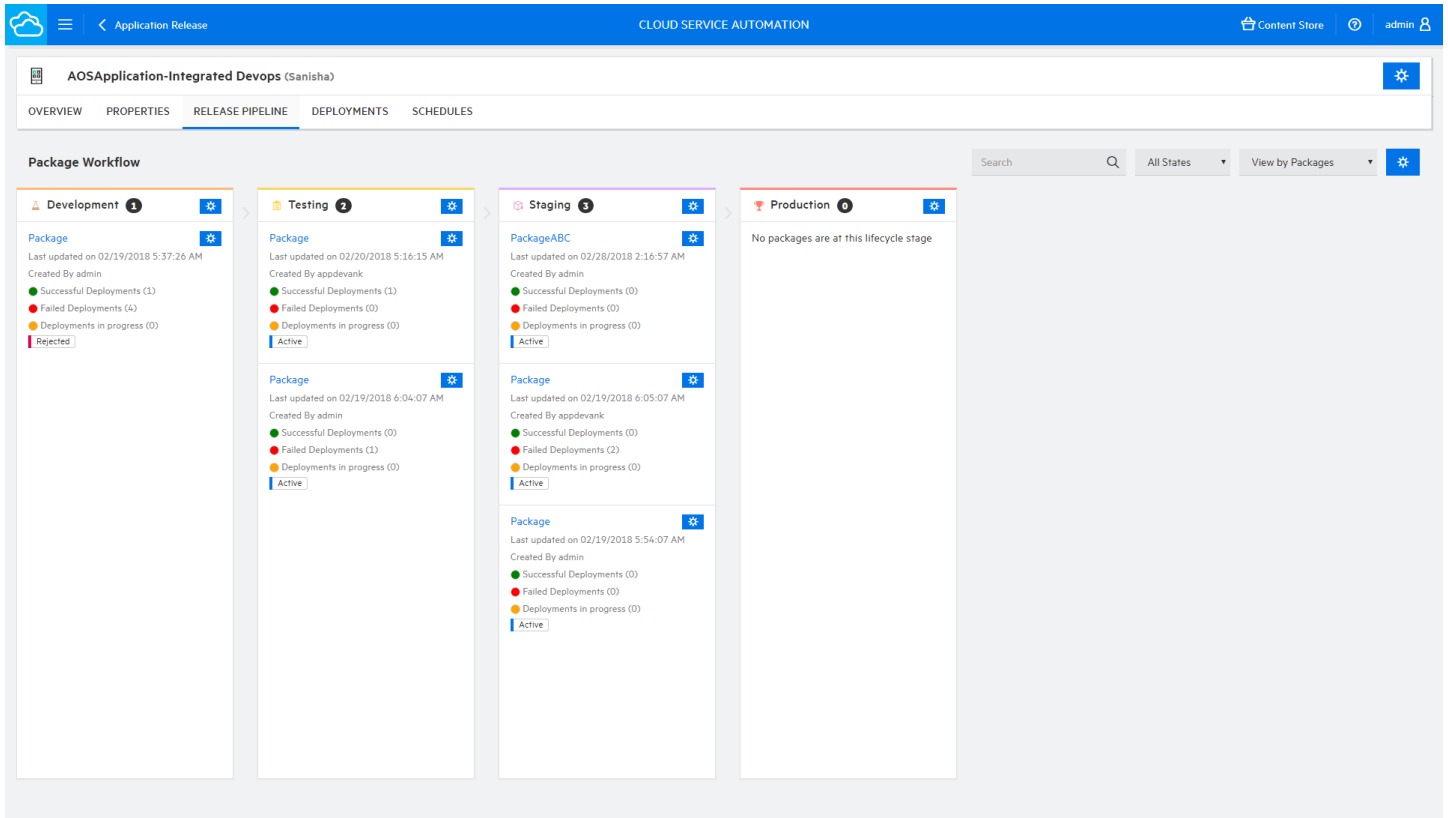
- A standard Git repo structure of service
  - Root directory
    - Dockerfile: define the build environment
    - Jenkinsfile: define the flow
    - deploy-controller: yaml template of Kubernetes
      - manifest.yaml
      - configmap.yaml
      - ingress.yaml
      - rc.yaml
    - Docker: Dockerfile and artifacts
    - Scripts: for CI/CD
    - Config file: for CI/CD
    - Source code

Orchestrating Pipelines in multiple Jenkins masters

Solution 1



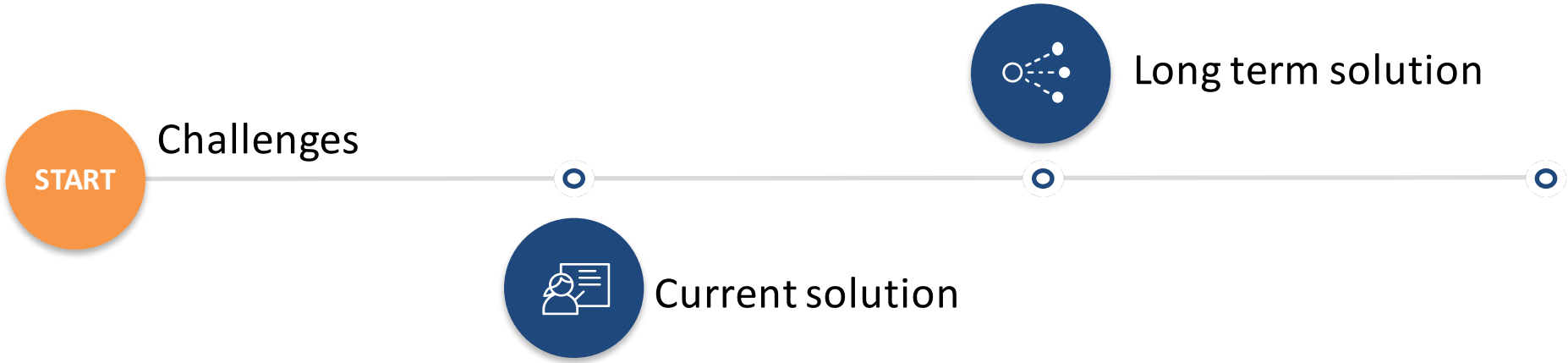
Solution 2



AGENDA

- Agile and DevOps Transformation
- Why Jenkins Pipeline
- Implementation CI/CD with Jenkins Pipeline
- Q&A

Challenges and solutions



Infrastructure

Challenges

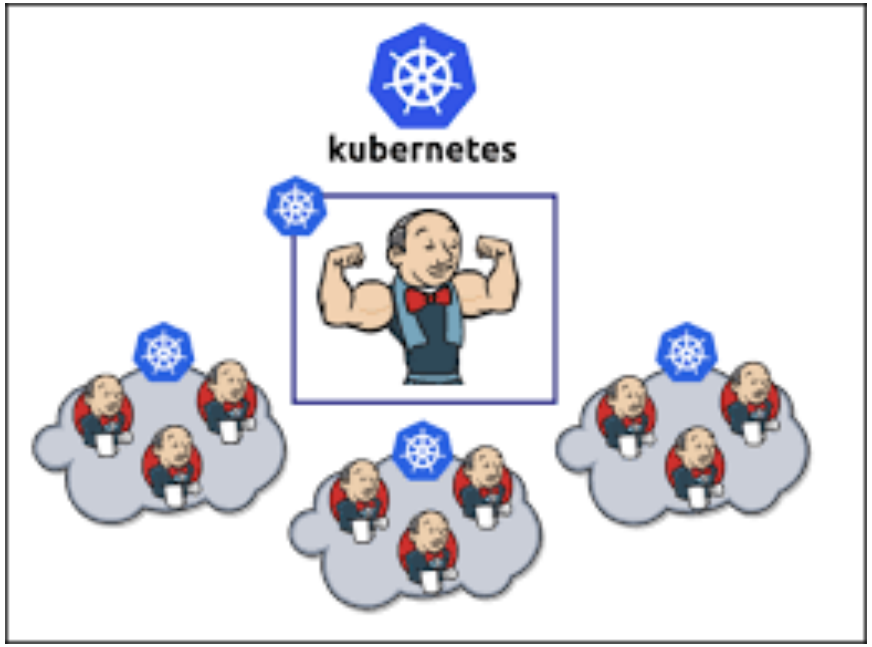
- Jenkins systems owned by different teams
- Duplicate effort for operation
- Simple backup and restore strategy, no rehearsal
- No HA
- Dedicated computing resource
- No scalability

Infrastructure

Current solution

Move to modern style infrastructure

- Step1: Dockerized build environment and tools
- Step2: Dockerized Jenkins agent
- Step3: Running on top of Kubernetes



Infrastructure

Long term solution

- Centralized Jenkins farm on top of Kubernetes
- Cross geography Jenkins setup
  - Network latency
- Shared computing resource
- Auto-Scaling Jenkins with Kubernetes
- Jenkins upgrade strategy
- Self-service way of setting up Jenkins cluster
- HA, IDM enterprise features add-on

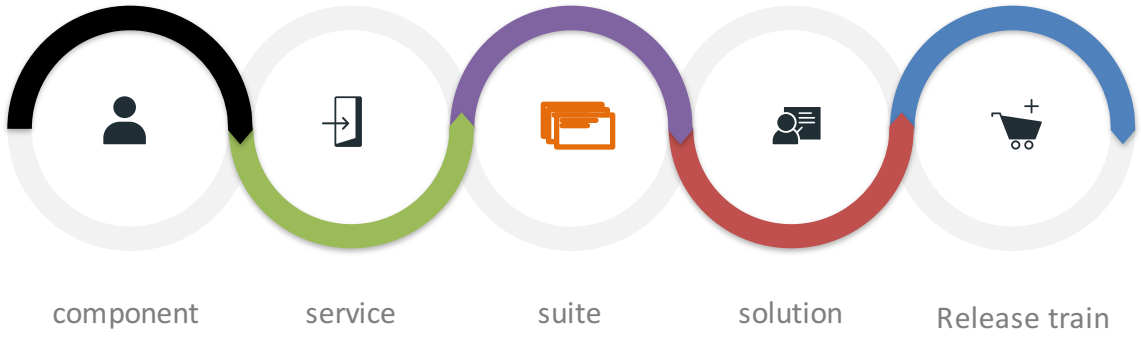
Integration

- 1000+ Git repositories
- 1000+ Jenkins jobs
- Dependencies between services
- Hybrid architecture
- Lead time
- Components -> service -> suite -> solution

Integration

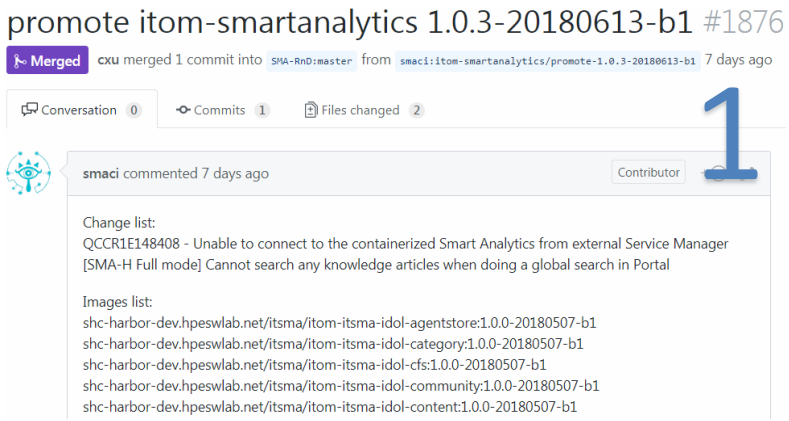
Current solution

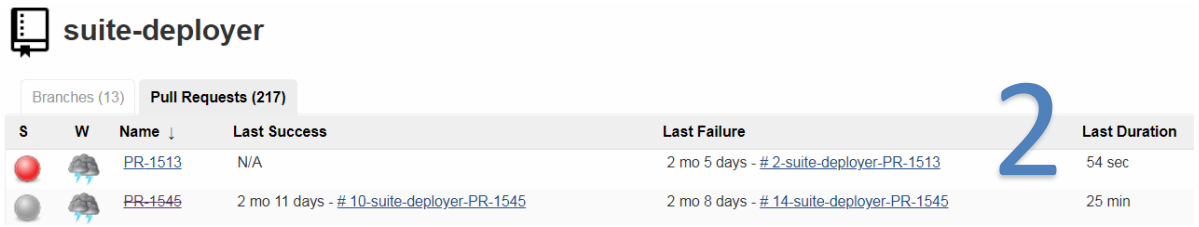
- Multiple level CI/CD pipelines
- Version declare and mapping cross levels
- Promotion and revert process using GitOps style and Jenkins mutibranch pipeline
- Contract based API testing on suite level
- Pipeline as code, define process and rules in code
- Reusable Pipeline, shared function or library
- Use Maven to unify build chain for different technical stacks or compiling tools of each micro services, simplify the logic programming of Jenkins pipeline
- Standard template, structure and Docker base image

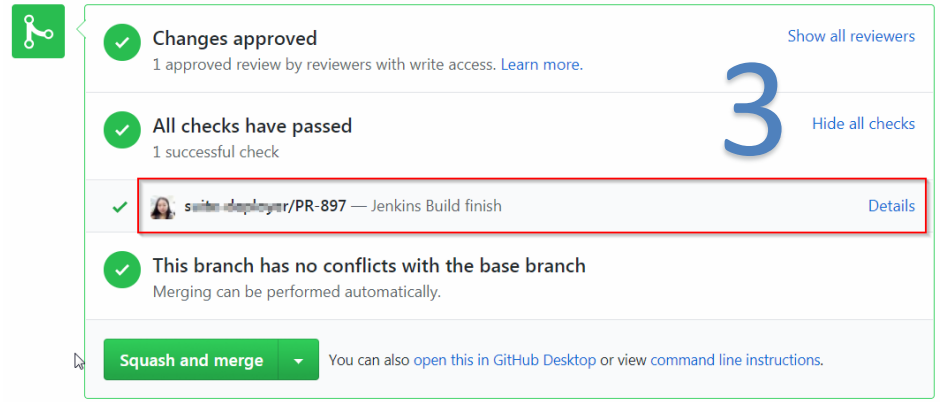


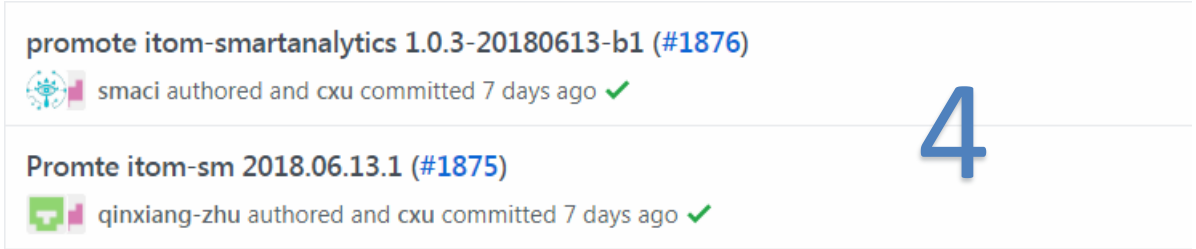
Integration

Promote and revert process using GitOps style and Jenkins mutibranch pipeline









Integration

Long term solution

- Smart trouble shooting for failure
- Graceful roll back mechanism

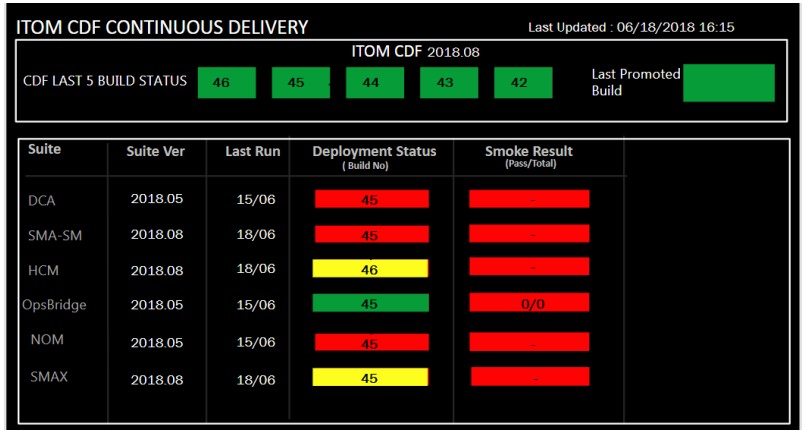
Visibility and traceability

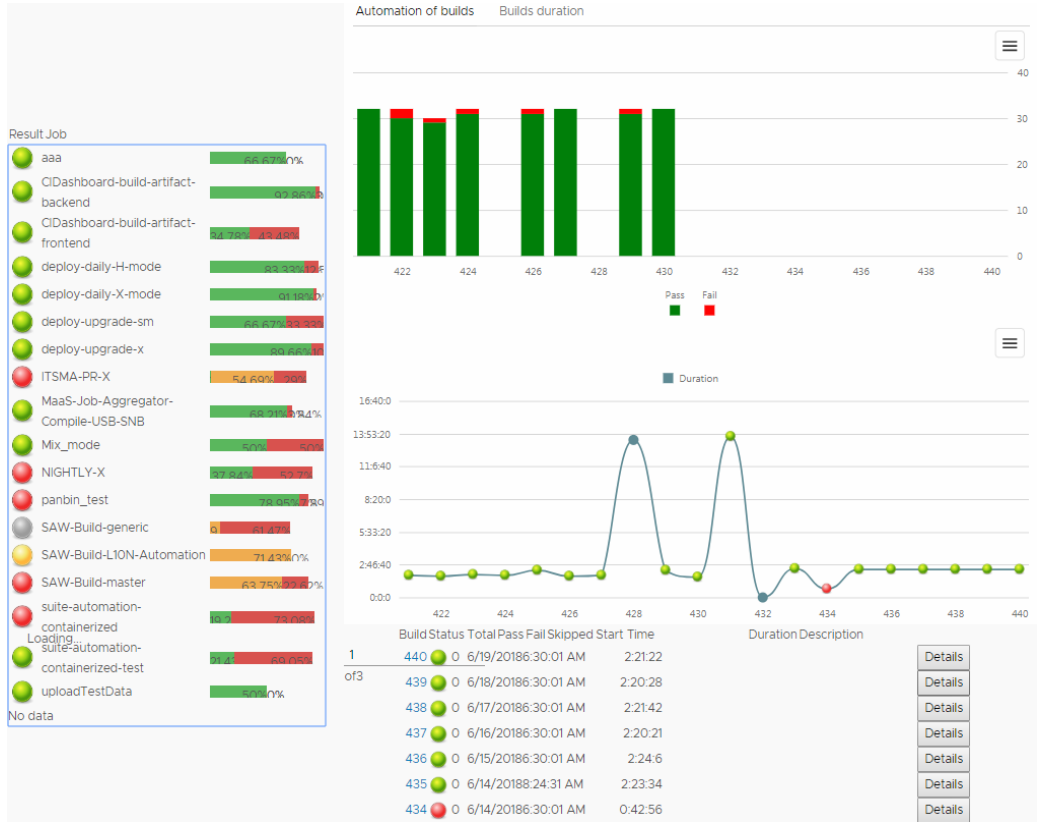
- CI/CD happens distributed
- Not easy to learn the status of pull request, promotion, testing, deployment, security scan etc.

Visibility and traceability

Current solution

- Centralized Dashboard
- Light weight data center
- Jenkins notification plugin
- Request data from GitHub, VMware, kubernetes
- Teams aggregate and pop up data to dashboard





Visibility and traceability

Long term solution

- Powered by Micro focus solution
- Integrate dashboard with Agile and CI/CD system Octane
- Extend generic data collector COSO for CI/CD system
- Vertica for Big data store and measurement
- Auto discover of CI/CD events

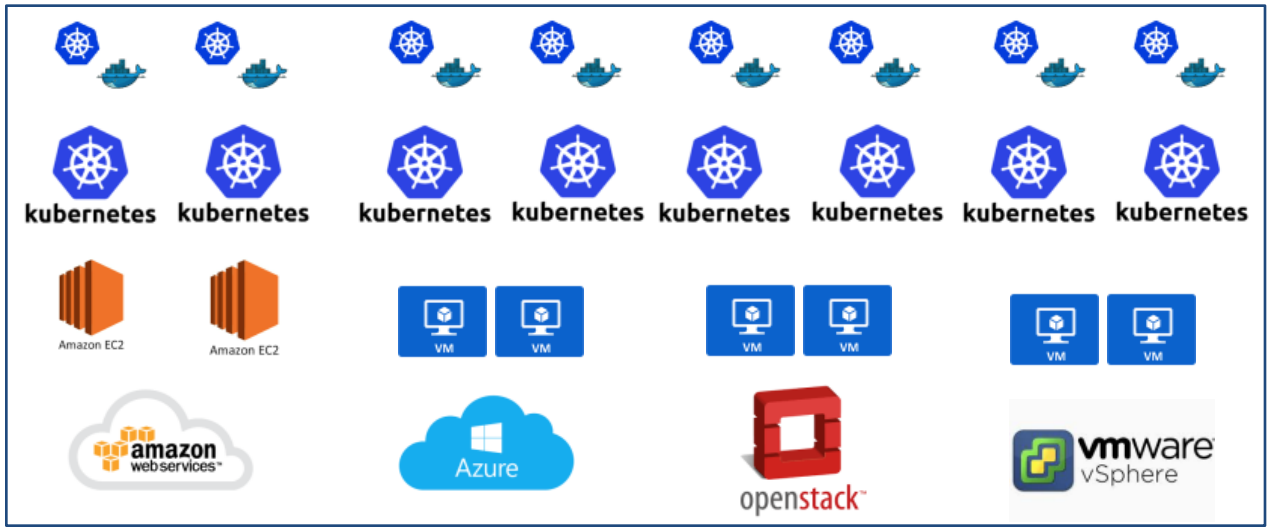
Deployment

- Complexity of environnent: on-premise, SaaS, public cloud
- Complexity of installation mode
- Services have dependances

Deployment

Current solution

- Standardize the VM template which are defined as code
- Unify configuration, environment, and method in each different layers
- Infrastructure provisioning: Terraform, Cloud Formation, Ansible
- Kubernetes cluster: Container Delivery Foundation
- Installation: Deploy kit (Go + Ansible)
- Deployment controller is defined in each service



Deployment

Long term solution

- Replace Ansible solution with CloudSlang and OO
- Integration with service portal Catalog and Offering
- Simulate Day 2 operation

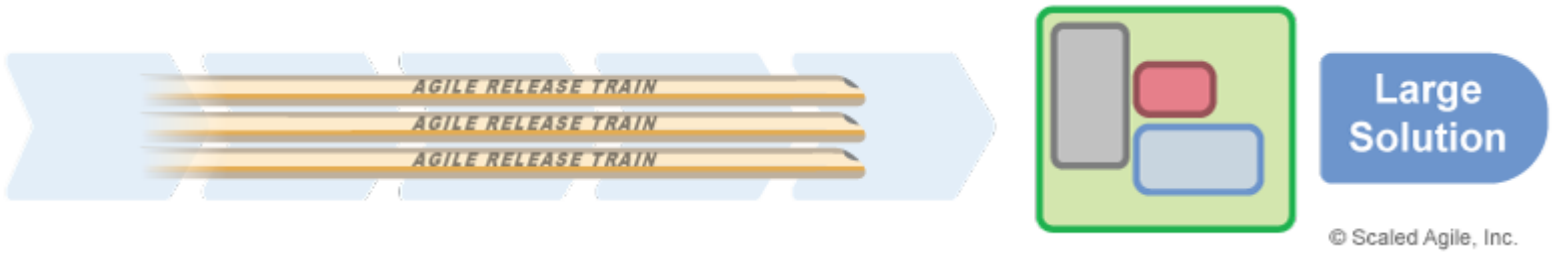
Release automation

- Different release train has different timeline
- Shared service consumed by multiple release trains
- Publish hundreds Docker images on DockerHub within a small time window
- Docker images validation and governance

Release automation

Current solution

- Distribute Docker pushing tasks to multiple instances using Jenkins pipeline parallel functionality
- Provide release automation service running on kubernetes



Deployment

Long term solution

- Integration with approval process
- End to end automation

Common capability as a service

- Overhead of operation

Common capability as a service

Current solution

- Automate everything
- Provide common capability as a service running on Kubernetes cluster or as a Jenkins job
  - VM operation
  - Automation test framework
  - Monitor
  - Security scan
  - 3rd party scan
  - static code analysis
  - code signing

Common capability as a service

Long term solution

- Powered by Micro Focus solution
- Building an interface into ChatOps for any services
- Service portal Catalog Offering

Q&A
Thank You.