



Automated Application Containerization

Journey to Openshift with RHAMT

Dean Peterson
Middleware Specialist Solution Architect
6/10/2019

AGENDA

- Containerization: Why and What?
- Automating the approach
- Next steps



WHAT IS CONTAINERIZATION?

WHAT IS CONTAINERIZATION?

Packaging of a configured application and all its dependencies into a light, portable, cloud-ready sandbox.



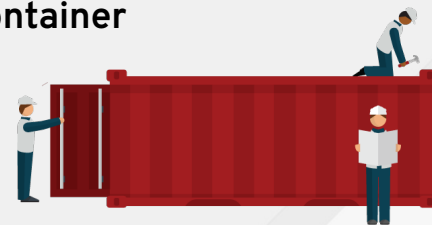
Physical host



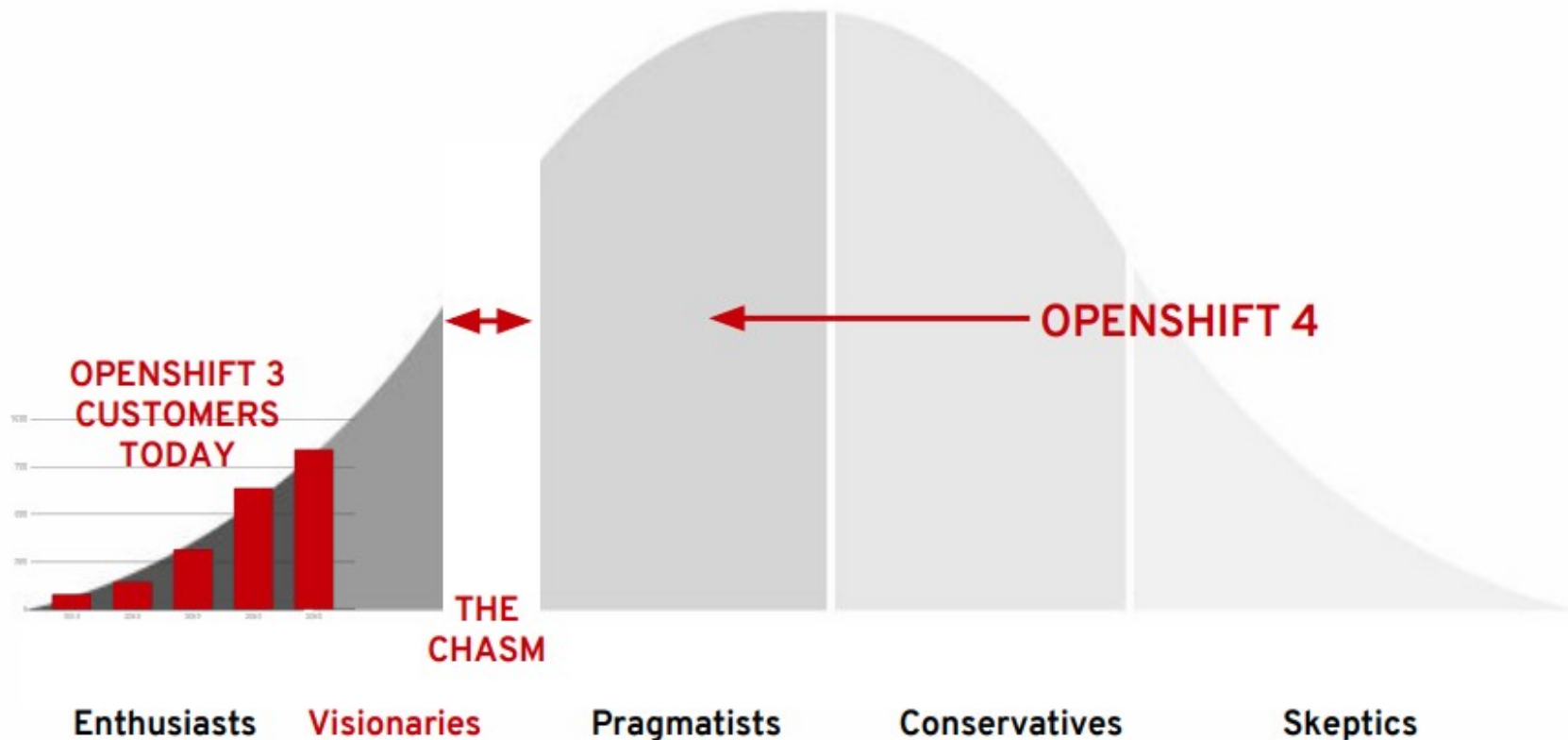
Virtual host



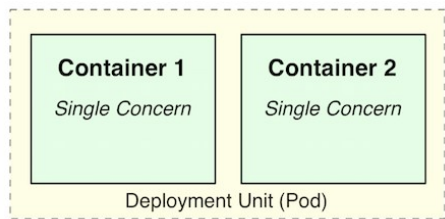
Container



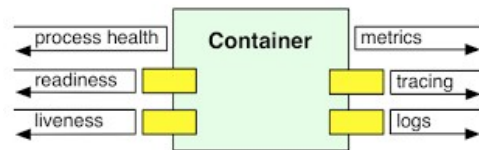
OPENSIFT 4 - PREPARING TO CROSS THE CHASM



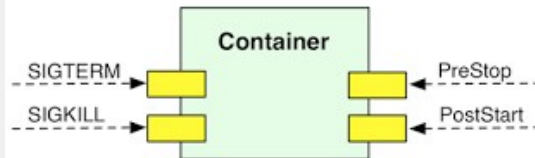
NEW POSSIBILITIES



SINGLE CONCERN



HIGH OBSERVABILITY



LIFECYCLE CONFORMANCE

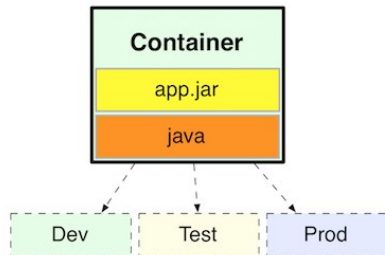
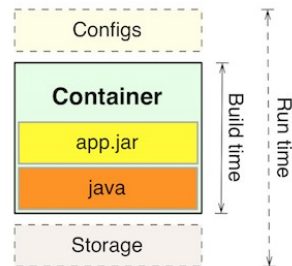


IMAGE IMMUTABILITY

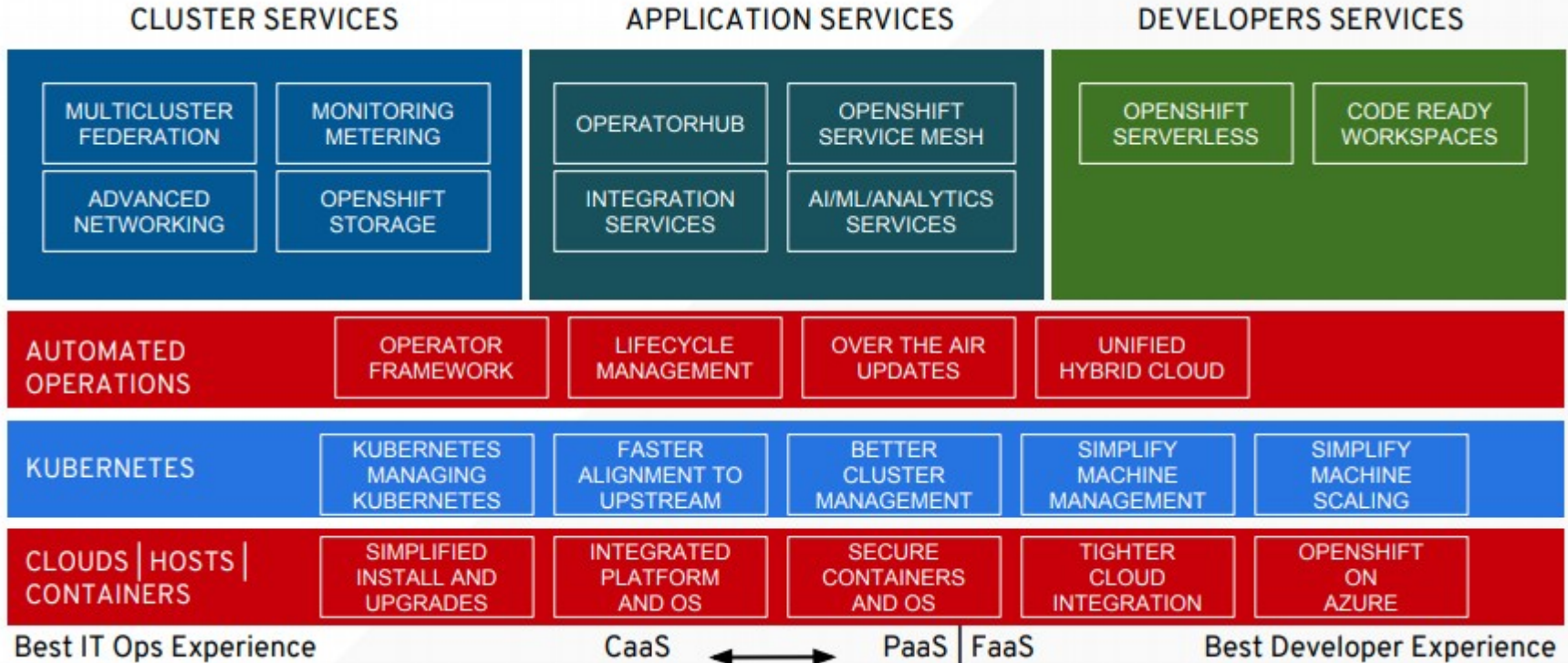


PROCESS DISPOSABILITY

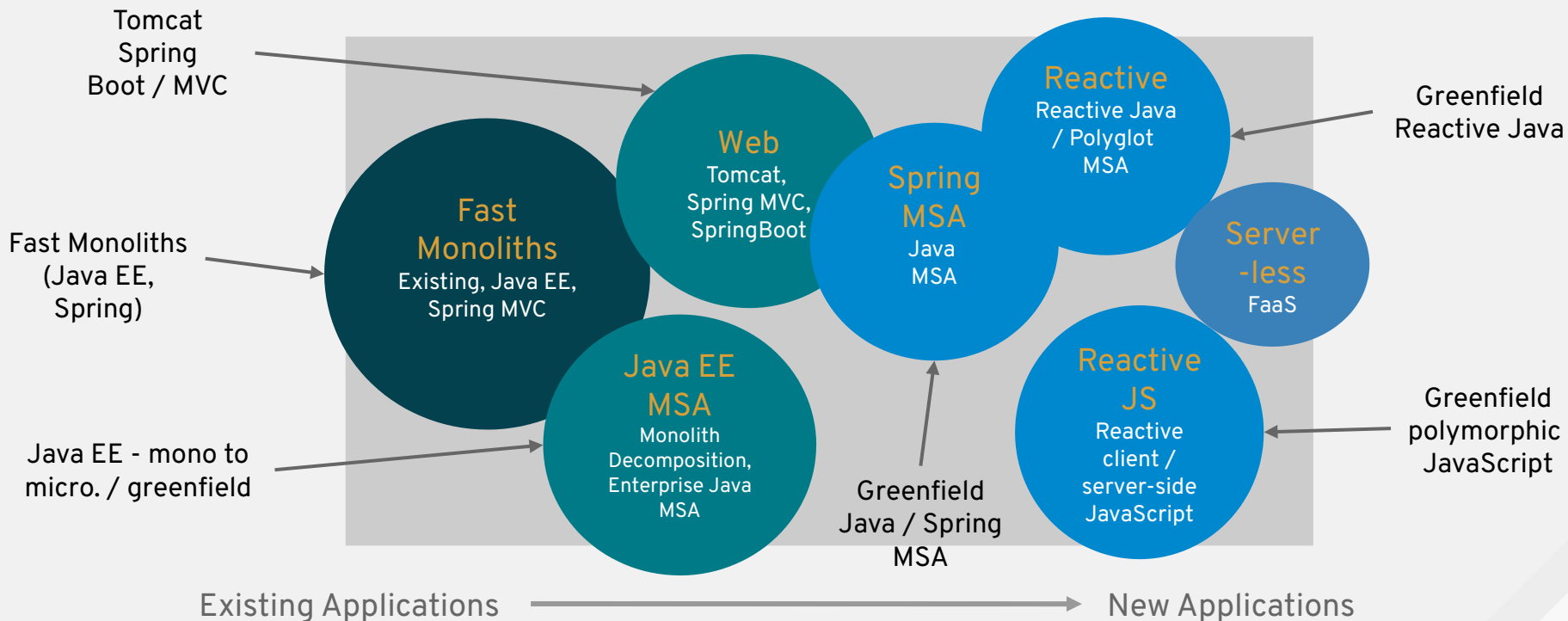


SELF-CONTAINMENT

NEWER POSSIBILITIES



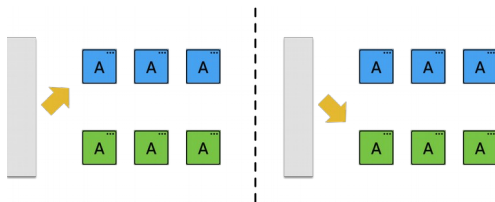
THE SPECTRUM OF ENTERPRISE APPS



NEW WAYS TO DELIVER



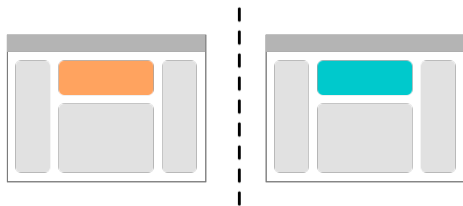
SELF SERVICE



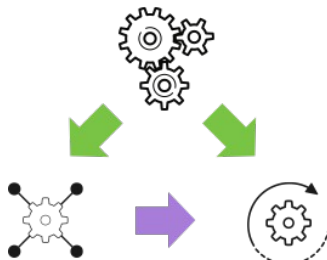
BLUE GREEN DEPLOYMENTS



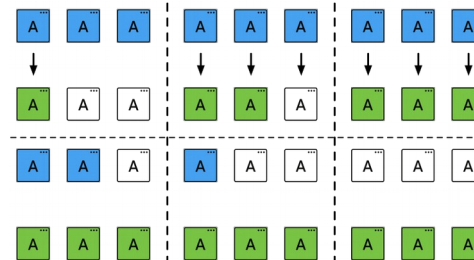
CANARY RELEASES



A/B TESTING

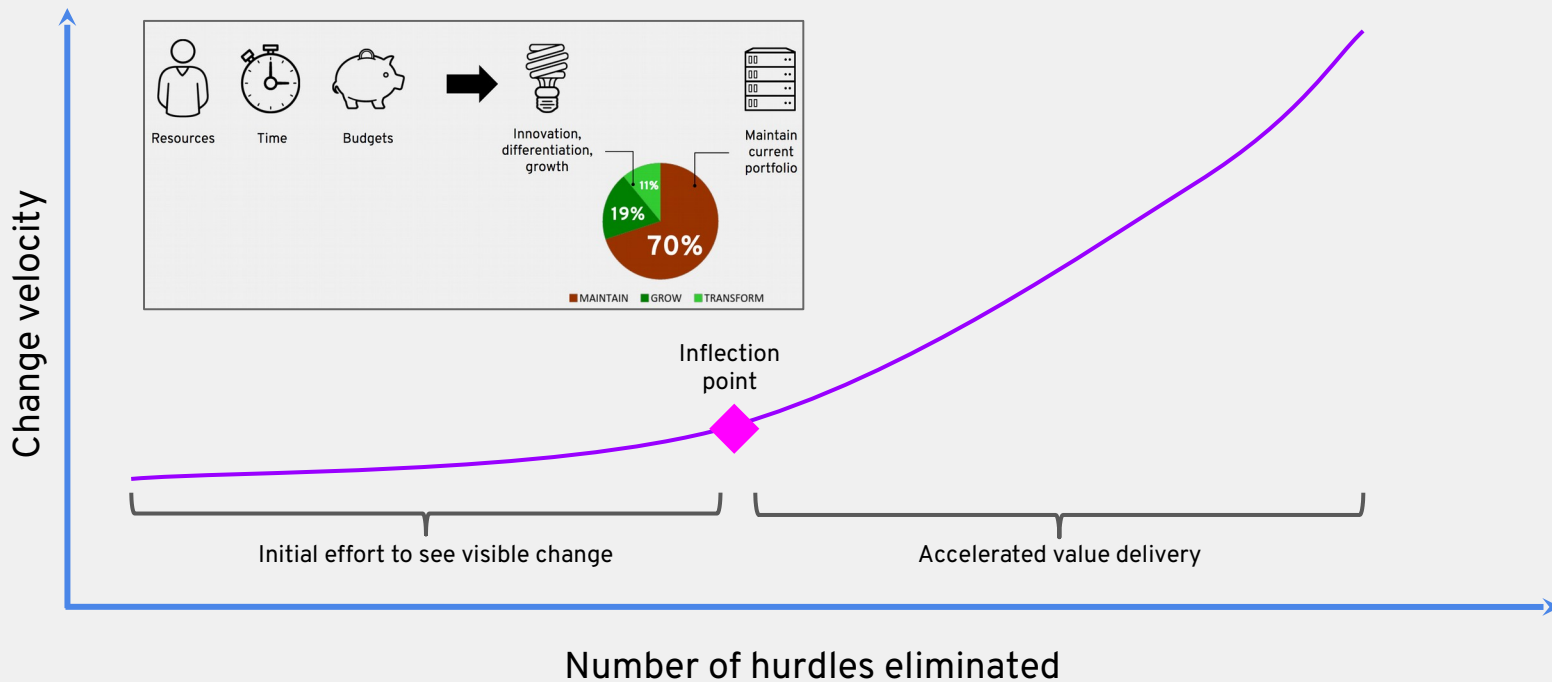


END-TO-END AUTOMATION

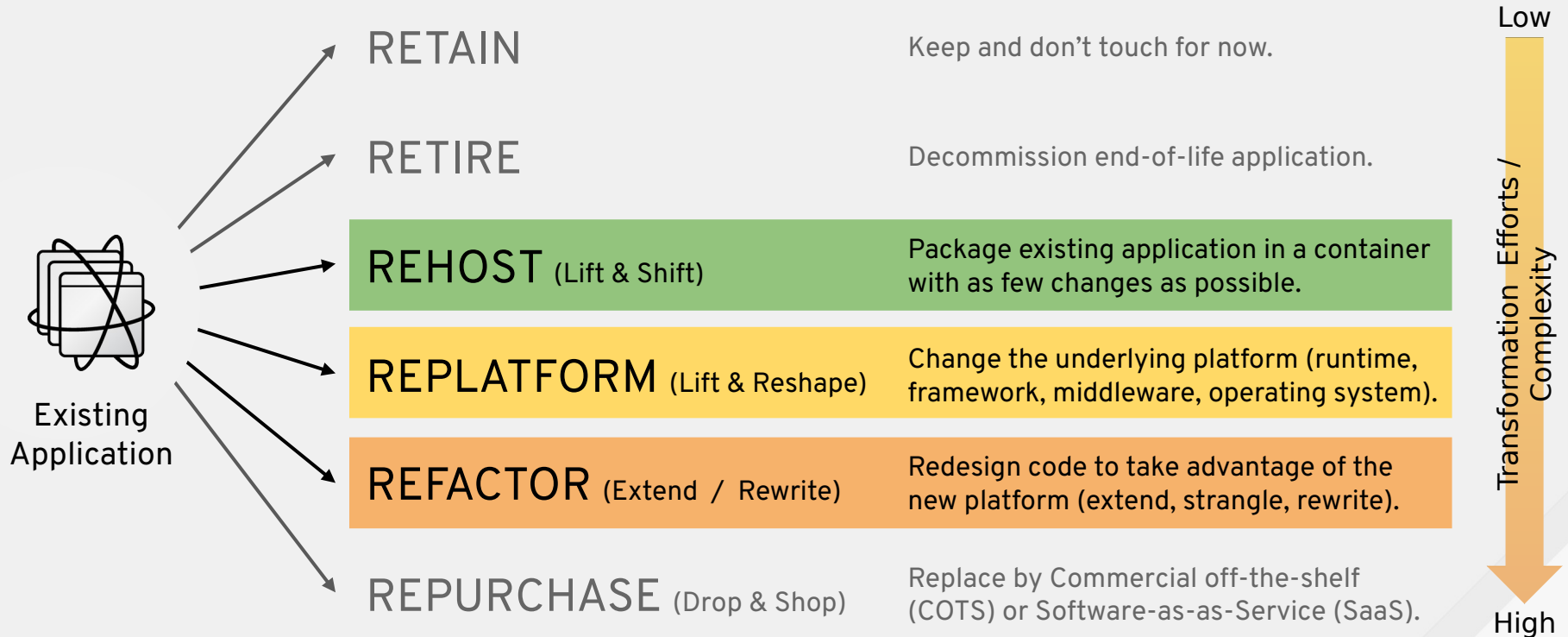


ROLLING UPGRADES

ECONOMICS

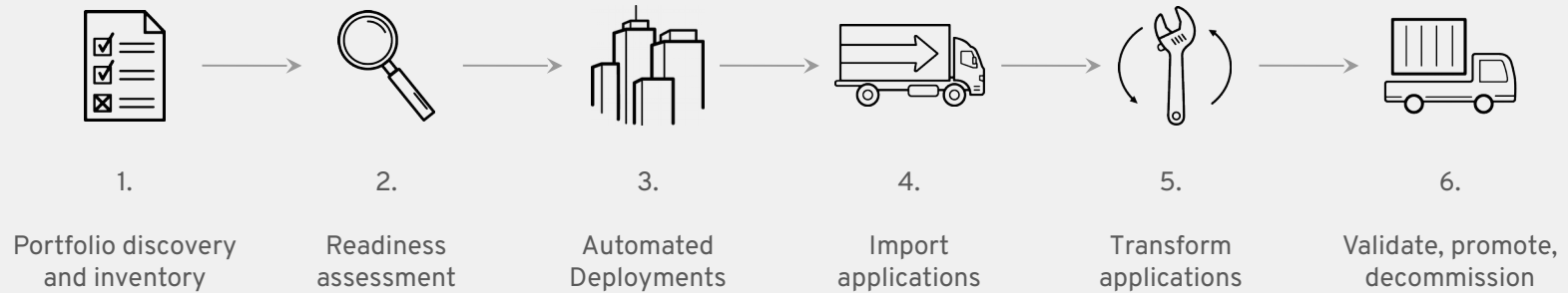


OPTIONS FOR CONTAINERIZATION

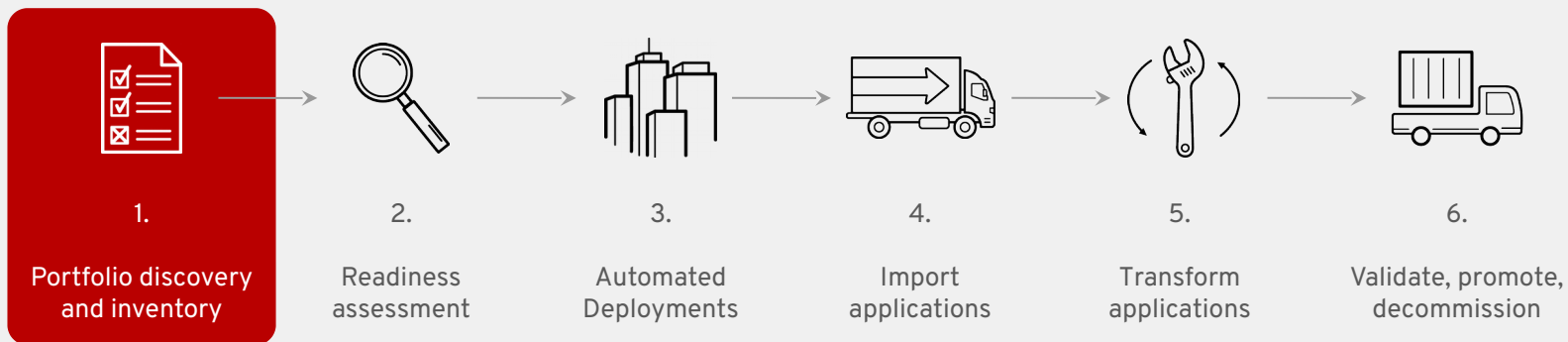


FULL-AUTOMATED APPROACH

CONTAINERIZATION JOURNEY



CONTAINER ADOPTION JOURNEY





1. PORTFOLIO DISCOVERY & INVENTORY

Create an **exhaustive list** (spreadsheet) of all **applications** (plan, governance, estimates)

EXAMPLE

Custom group for similar applications

Owners could be further split between ...
technical lead / business owner / test owner / external provider

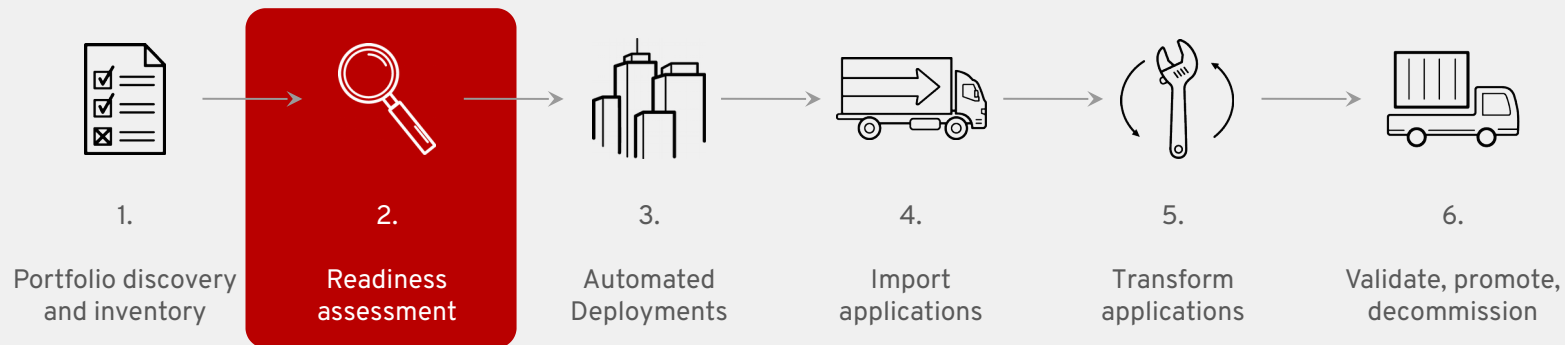
Business domain	Logical application	Artifact name	Relevant version(s)	Application server	Type	Owner	In-scope?	Target release	Complexity	Business criticality	Usage
Core	CoreBanking	bank.ear	5.4_2017.FINAL	WLS 3.4.2	Front-end	Bob Meister	yes	Q2 2021	High	High	High
Core	CoreBanking	bank-backend.jar	5.4_2017.FINAL	WLS 3.4.2	Backend	Bob Meister	yes	Q2 2021	Medium	High	Medium
Batch	JobManager	process-engine.war	12.0.3	WLS 3.4.6	3rd party	Jack Fruit	no	-	Low	High	Low
...

Internal application classification.....

Planning and scope

Relevant for prioritization

CONTAINER ADOPTION JOURNEY



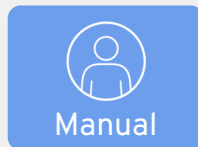


2. READINESS ASSESSMENT

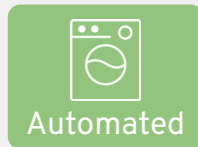
WHAT?

1. Group similar applications based on their business & technical characteristics
2. Assess container-readiness
3. Identify the right transformation based on business value
4. Estimate efforts

HOW?



Conduct interviews, workshops, code & document reviews (all levels).

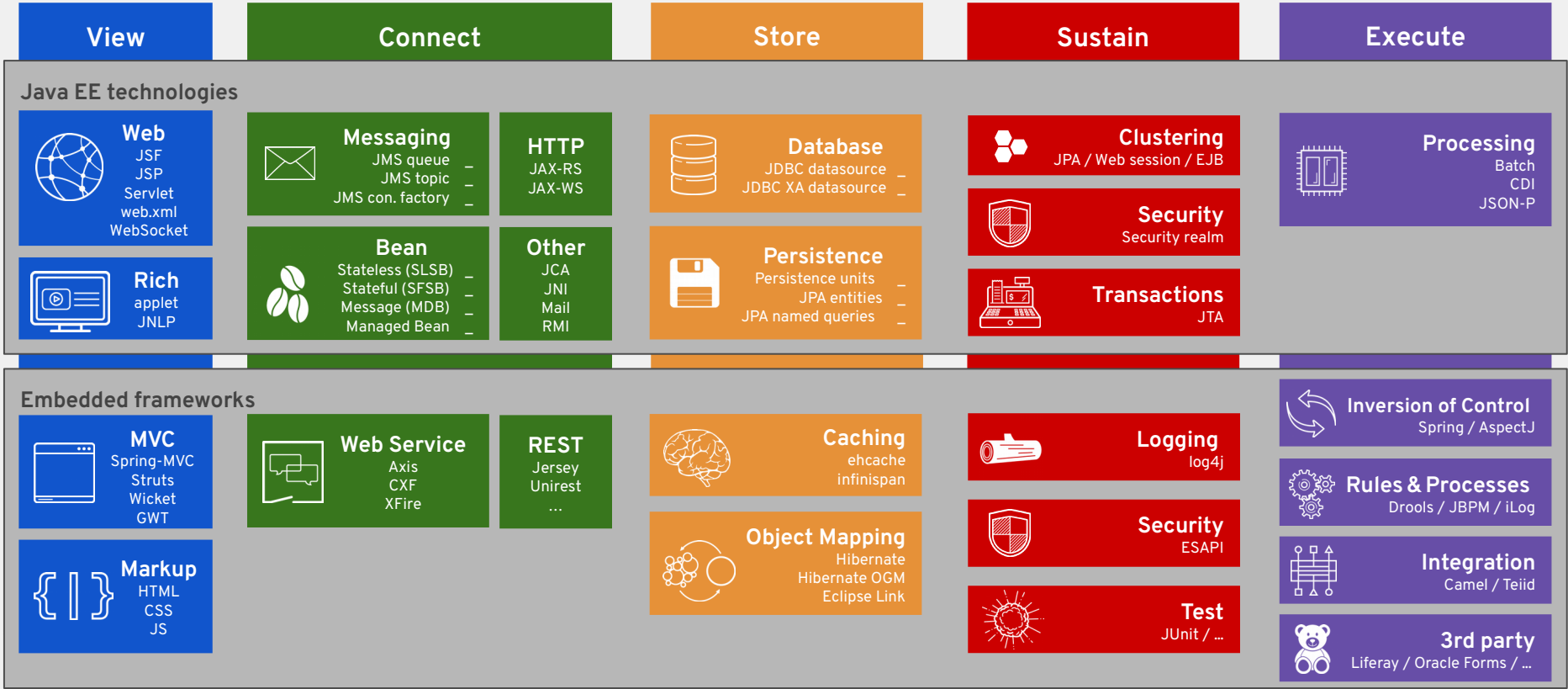


- Questionnaire-based frameworks
- Automated code analysis tool (e.g. RHAMT for Java applications)



2. READINESS ASSESSMENT

GROUPING APPLICATIONS





2. READINESS ASSESSMENT

CONTAINER-READINESS FACTORS



Software architecture

- Monolith vs. Modular
- Coupling (loose vs. tight)
- State management
- Communication protocols
- CPU, IO, storage, connectivity needs
- Compliance, security, availability, resiliency requirements



Dependencies

- OS & hardware
- Licensing and vendor support (3rd party components)



Processes

- Monitoring, alerting, log & configuration management

SCOPING A TRANSFORMATION

- Business criticality, usage, user base
- Application lifecycle (age)
- Expected business value & efforts





2. READINESS ASSESSMENT

TOOLING SUPPORT

RED HAT® APPLICATION MIGRATION TOOLKIT

Catalyze large scale application modernizations and migrations

- Automate code analysis
- Support effort estimation
- Accelerate code migration
- Free & Open Source

Red Hat Application Migration Toolkit

IBM
WebSphere AS

Oracle
WebLogic Server

Java EE
upgrades

JBoss EAP
upgrades

Cloud readiness,
containerization

Pluggable:
your own rules



Command line
interface



Web
console



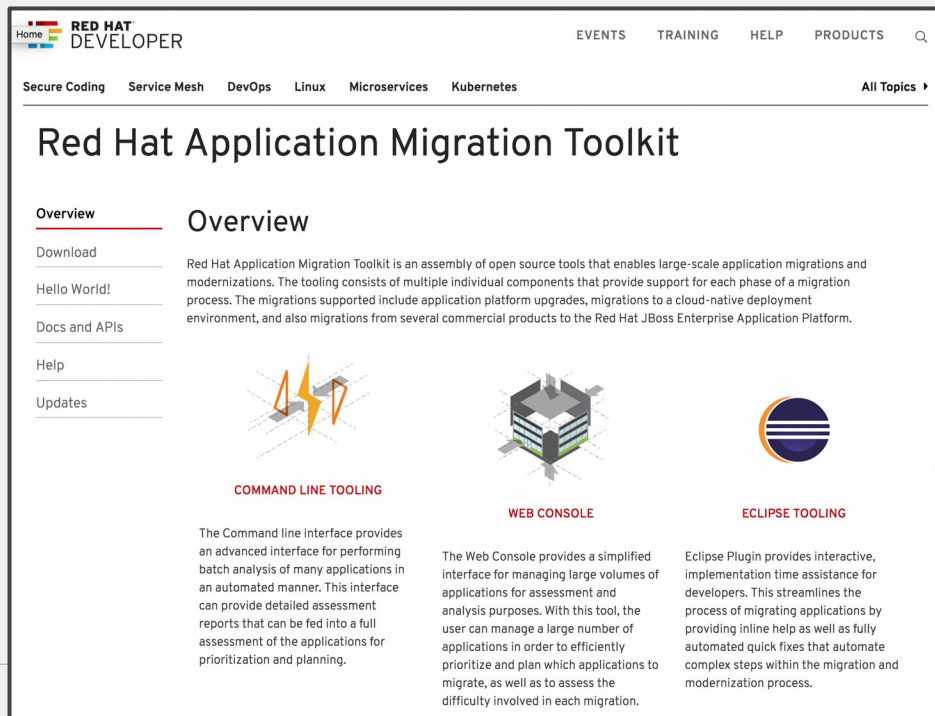
Eclipse
plugin



2. READINESS ASSESSMENT

TOOLING SUPPORT

Download the Toolkit on <https://developers.redhat.com/products/rhamt/overview/>



The screenshot shows the Red Hat Developer website for the Application Migration Toolkit. The page has a navigation bar with links for Home, RED HAT DEVELOPER, EVENTS, TRAINING, HELP, and PRODUCTS. Below the navigation bar is a secondary menu with links for Secure Coding, Service Mesh, DevOps, Linux, Microservices, and Kubernetes, along with an 'All Topics' dropdown. The main heading is 'Red Hat Application Migration Toolkit'. The 'Overview' section is active, with a red arrow pointing to it from the left. The 'Overview' section contains a paragraph describing the toolkit as an assembly of open source tools for large-scale application migrations and modernizations. Below this are three columns, each with an icon and a description: 'COMMAND LINE TOOLING' (lightning bolt icon), 'WEB CONSOLE' (server rack icon), and 'ECLIPSE TOOLING' (Eclipse logo icon).

Overview

Download

Hello World!

Docs and APIs

Help

Updates

Overview

Red Hat Application Migration Toolkit is an assembly of open source tools that enables large-scale application migrations and modernizations. The tooling consists of multiple individual components that provide support for each phase of a migration process. The migrations supported include application platform upgrades, migrations to a cloud-native deployment environment, and also migrations from several commercial products to the Red Hat JBoss Enterprise Application Platform.

COMMAND LINE TOOLING

The Command line interface provides an advanced interface for performing batch analysis of many applications in an automated manner. This interface can provide detailed assessment reports that can be fed into a full assessment of the applications for prioritization and planning.

WEB CONSOLE

The Web Console provides a simplified interface for managing large volumes of applications for assessment and analysis purposes. With this tool, the user can manage a large number of applications in order to efficiently prioritize and plan which applications to migrate, as well as to assess the difficulty involved in each migration.

ECLIPSE TOOLING

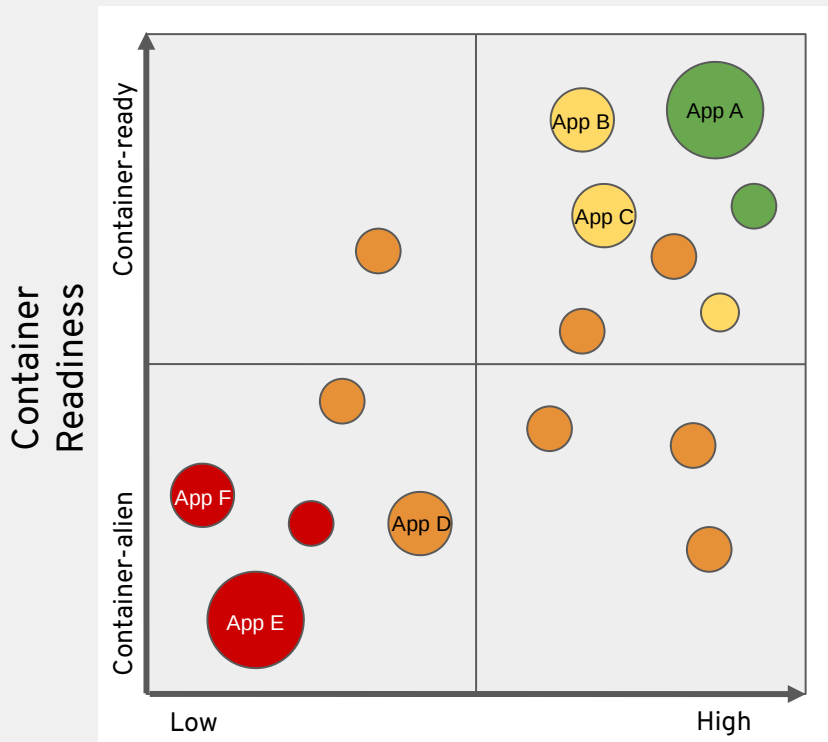
Eclipse Plugin provides interactive, implementation time assistance for developers. This streamlines the process of migrating applications by providing inline help as well as fully automated quick fixes that automate complex steps within the migration and modernization process.

DEMO



2. READINESS ASSESSMENT

RESULTS & PRIORITIZATION



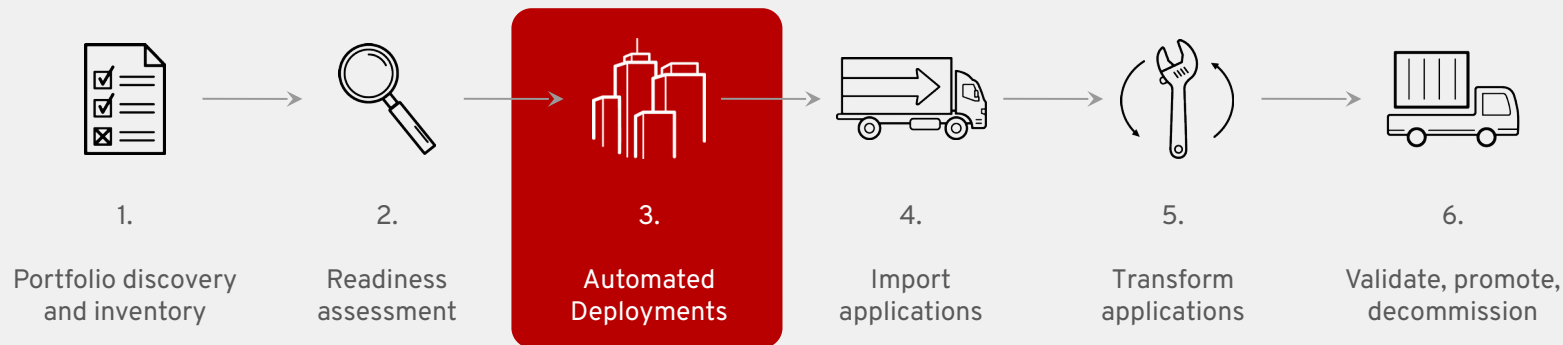
Containerization status

- Done!
- In progress
- Not started
- Out-of-scope!



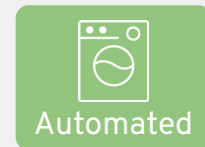
Level-of-Efforts

CONTAINER ADOPTION JOURNEY





3. AUTOMATED DEPLOYMENTS



WHAT?

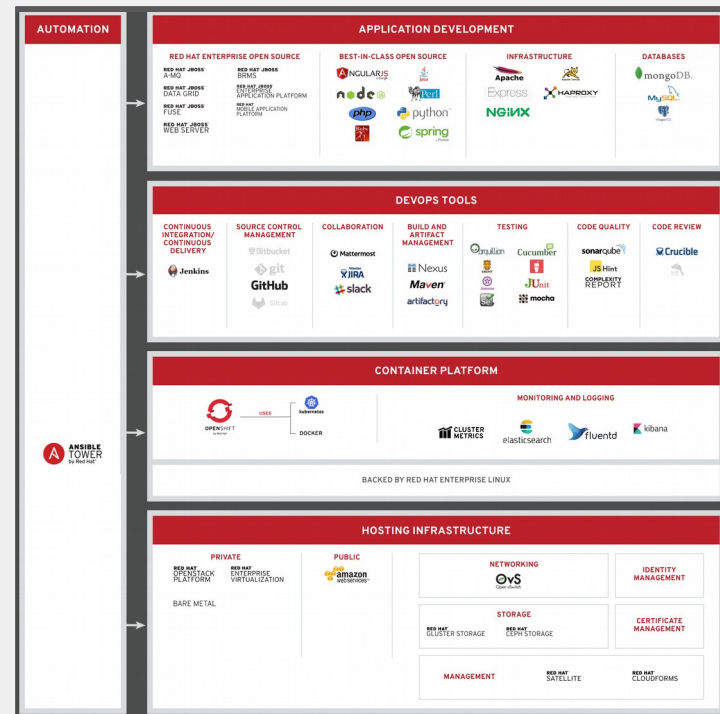
- Design the new platform (MVP, requirements)
- Document architecture decisions.
- Build the full-ecosystem

HOW?

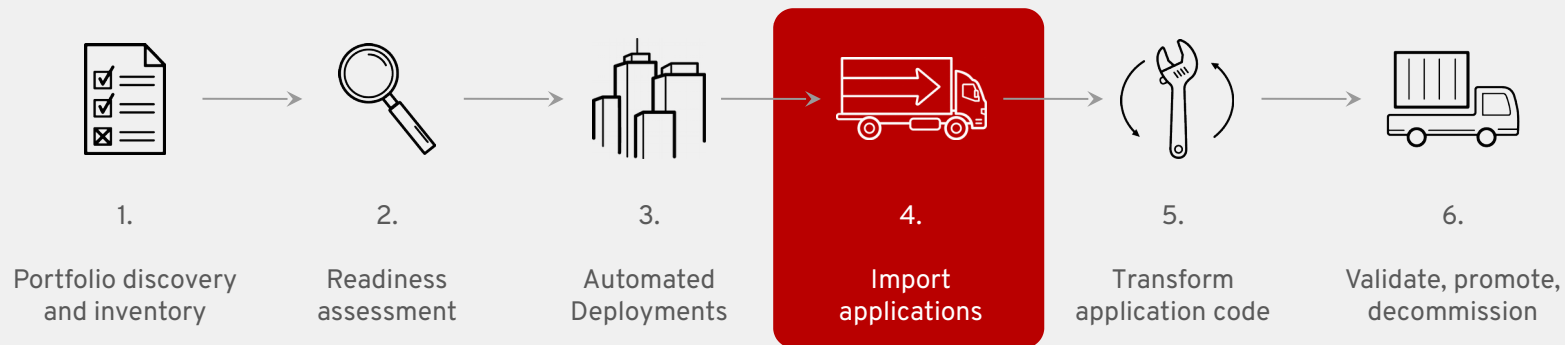
Adapt push-button infrastructure (reference architecture)

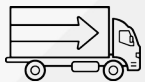
- All needed stack to start building solutions on day one
- Robust IaaS, PaaS, DevOps toolchain & AppDev services
- Fully-automated deployment

<https://www.redhat.com/en/explore/my-open-innovation-lab-stack>



CONTAINER ADOPTION JOURNEY





4. IMPORT APPLICATIONS



WHAT?

Bridge existing and new platforms

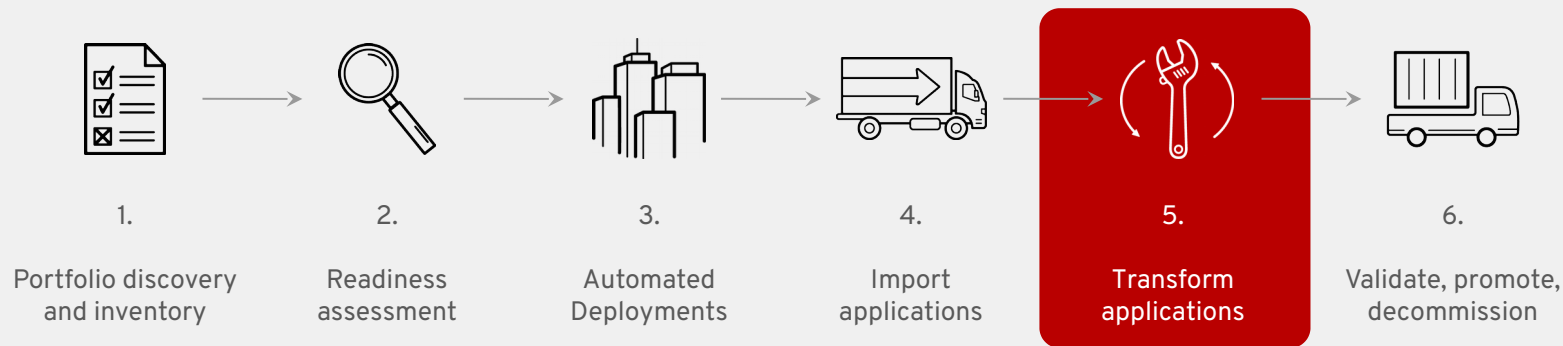
- Import assets of the existing application lifecycle (source code, configuration, repositories, processes)
- Reduce manual migration gaps

HOW?

- Automate import of all applications (no code change)
 - Provide a “deploy-to-new-platform” button to devs
 - Pre-generate templates and configuration
- Enable users (templates, guides)

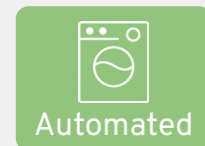
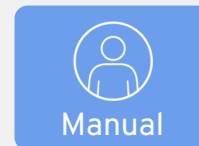


CONTAINER ADOPTION JOURNEY





5. TRANSFORM APPLICATIONS



WHAT?

Update code & configuration to containerize an application according the chosen transformation.

HOW?

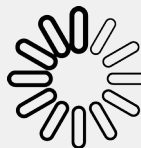
Step-by-step approach with well-defined activities. As few changes as necessary.



1.
Image choice,
initial configuration



2.
Pre-emptive
code changes
(e.g. RHAMT)



3.
Get application
running in a
container

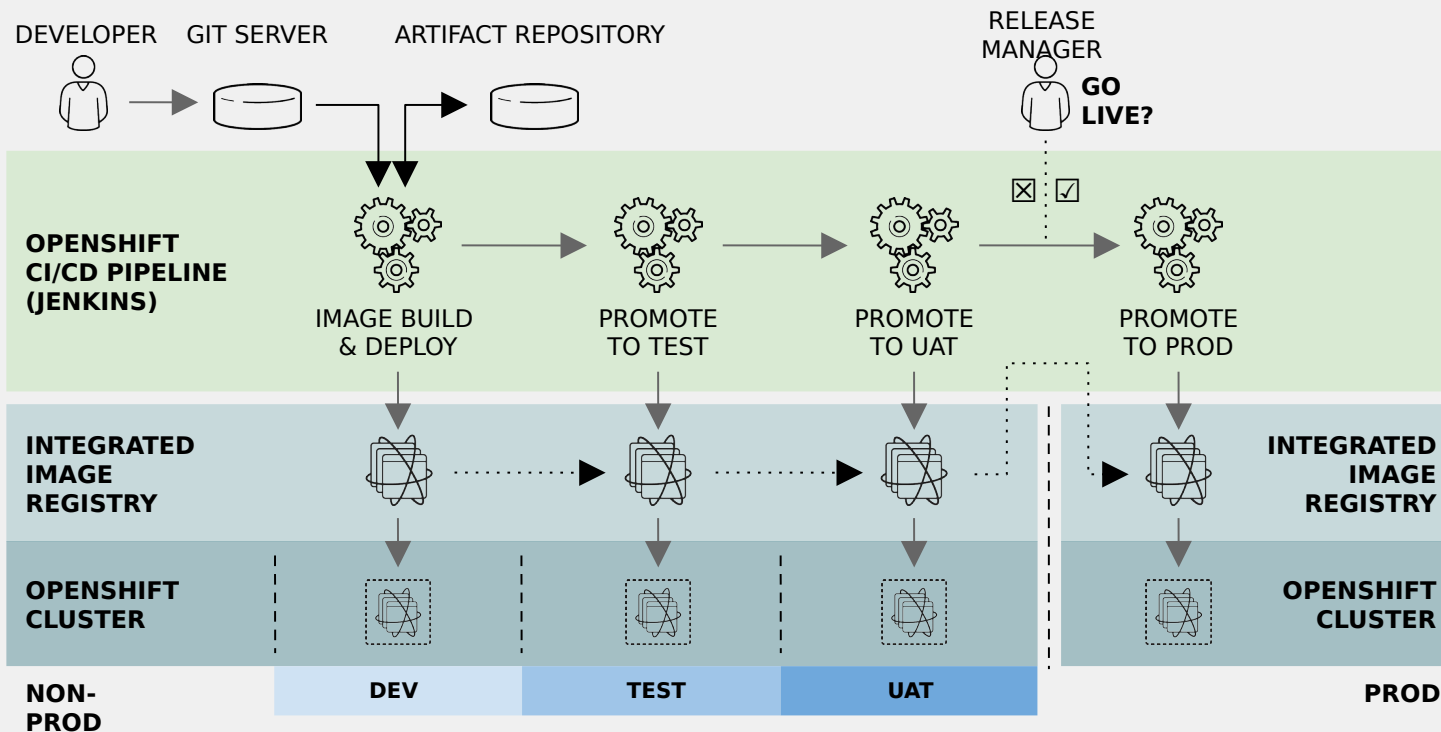


4.
Optional changes
(refactor, strangle)

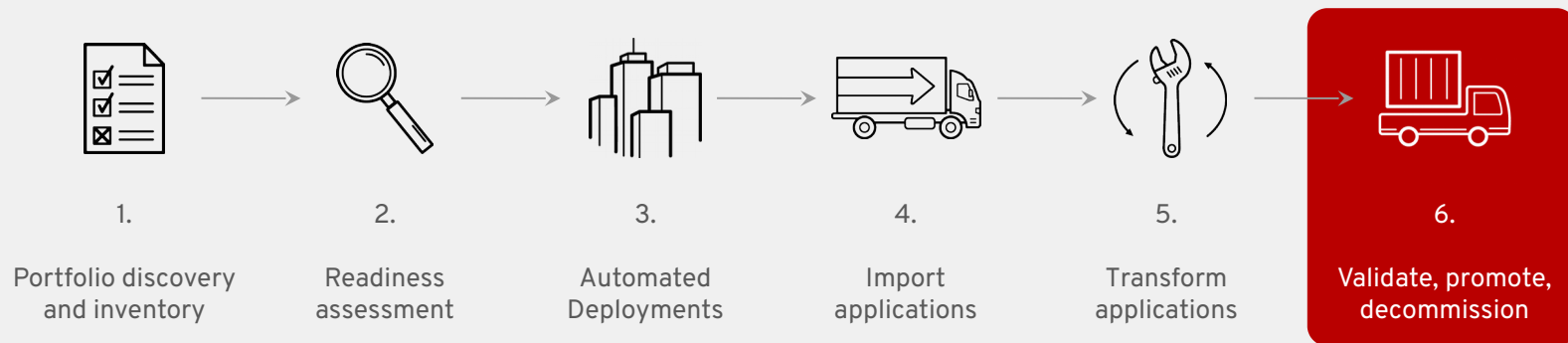


5.
Test & Promote





DEPLOYMENT PIPELINES



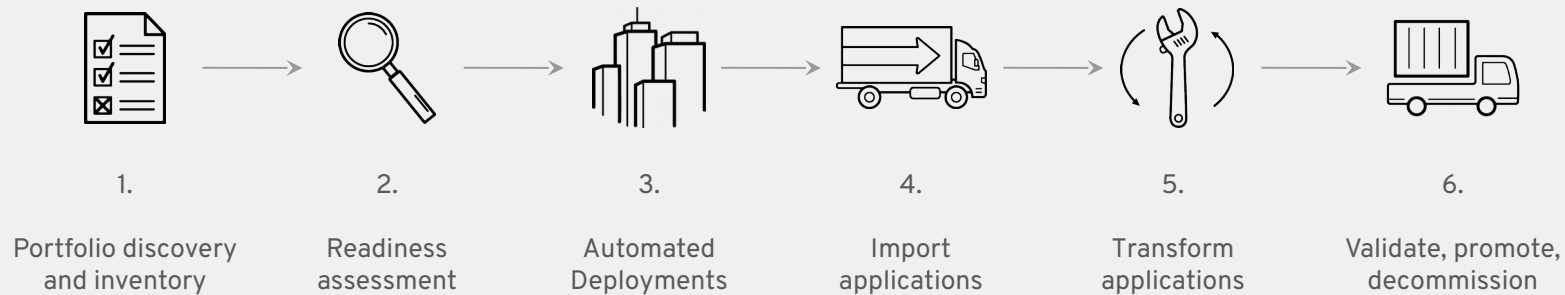
CONTAINER ADOPTION JOURNEY



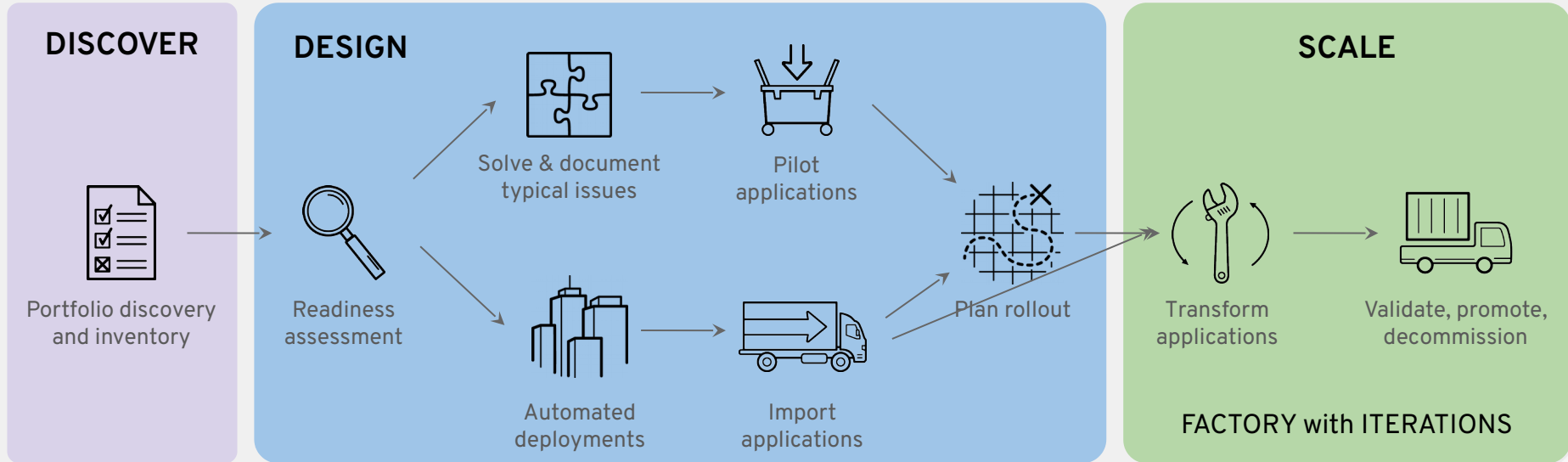
What does a high performing organization look like?

	 ↓ LEAD TIME FOR CHANGE	 ↑ DEPLOYMENT FREQUENCY	 ↓ MEAN TIME TO RECOVERY (MTTR)	 ↓ CHANGE FAILURE RATE*
	Measures of MARKET AGILITY		Measures of RELIABILITY	
WHAT	Time from code committed to deployed to production	Proxy for batch size, how often does an app deploy to production	How long it takes systems to recover from failures in production	Percentage of deployments requiring rollback and/or fixes
WHY	Shorter is better. Enables faster feedback cycles and makes you better able to adjust to the marketplace	Indicator of batch size. Smaller batch size leads to more market agility	Critical to ensure that we aren't speeding up delivery at the expense of negative customer impacts	*Secondary indicator of stability

CONTAINER ADOPTION JOURNEY

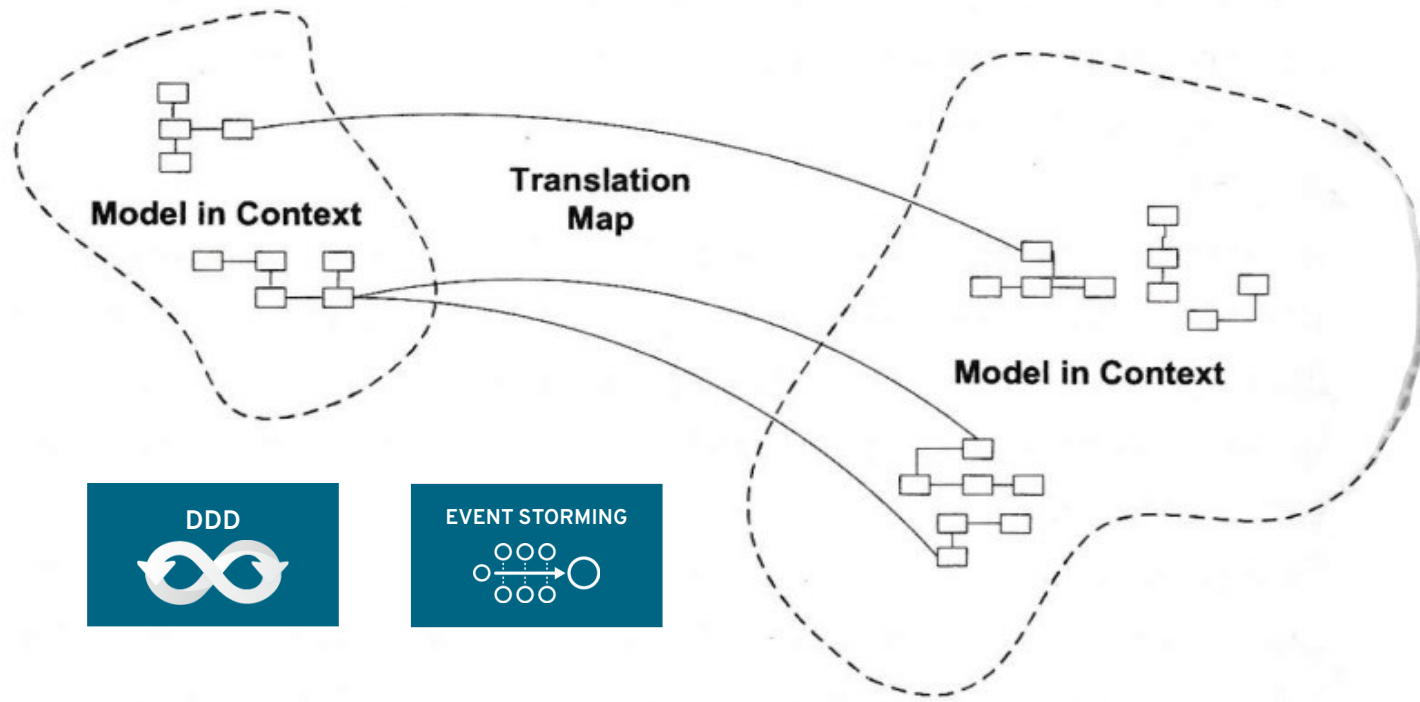


CONTAINER ADOPTION JOURNEY AT SCALE!



NEXT STEPS

CONTEXT MAP



JUMPSTART YOUR MODERNIZATION WITH RED HAT OPEN INNOVATIONS LABS

MODERNIZE TRADITIONAL APPS

- Extend applications
- Optimize applications
- Scale applications
- Expose to orchestration

INNOVATION ACCELERATED

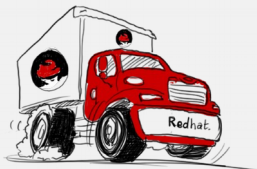
DEVELOP CONTEMPORARY APPS

- Develop on PaaS environment
- Transform how you design and develop apps
- Adopt lean and agile principles
- Master DevOps practices



COLLABORATION

Space to work,
innovate, and discuss



RESIDENCY

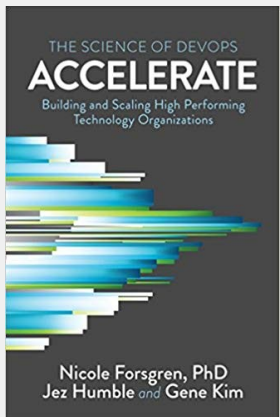
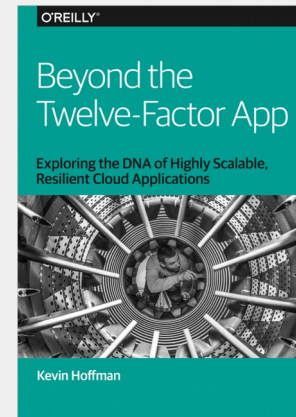
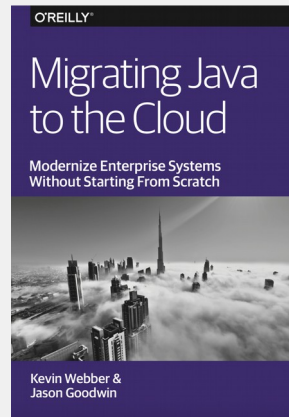
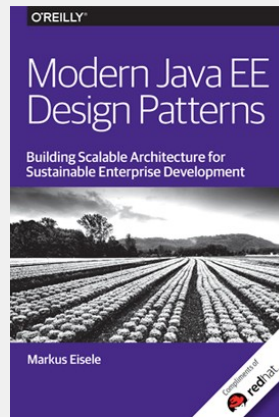
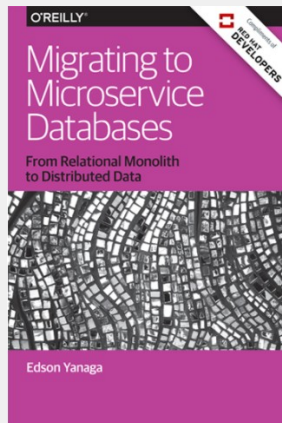
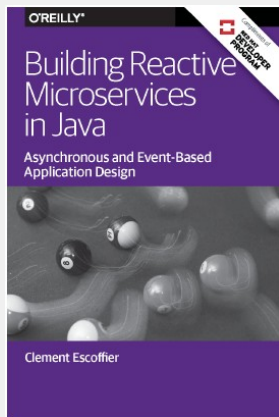
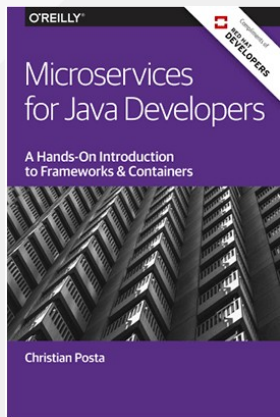
An eight-week accelerated
teaming engagement



COMMUNITY INCUBATION

Communities
supporting innovation

FURTHER READINGS



<https://developers.redhat.com/resources/#!type=book>



THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



[youtube.com/user/
RedHatVideos](https://youtube.com/user/RedHatVideos)