The Silence of the Lambs

Inspecting Source Code and Binaries, in Continuous Delivery Pipelines

Michael Hüttermann
Freelancing DevOps Consultant

http://twitter.com/huettermann
http://huettermann.net
http://code.huettermann.net

Java Champions

Oracle Groundbreaker Ambassador

Academic Research

- Researching DevOps (PhD project)
- Studying DevOps to unveil what DevOps is (for you)
- Running Case Study (some interviews, …)

Context

Objectives

- Learn good practices, derived from real world success stories.
- Learn how to address common project challenges.
- Learn DevOps context and building blocks.
- Learn about tools, and how they can be integrated.
- Learn about microservices, cloud, and Oracle Container Cloud Service.
- Motivate and initially prepare to zoom in yourselves, later on. :-)

Disclaimer

- More than one solution
- Consider individual requirements and use cases
- DevOps: Mix of goals, concepts and tools
- Not an in-depth tool discussion
- Giving colored zoo of appetizers

Agenda

- Setting the stage
- Deriving pipelines (and the cycle time)
- Demoing a roundtrip

Agenda

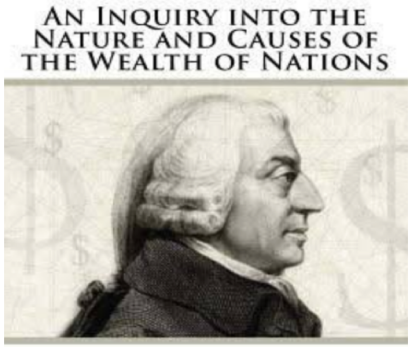
- Setting the stage
- Deriving pipelines (and the cycle time)
- Demoing a roundtrip

QUIZ

First DevOps book?

First DevOps superstar?

Basic conditions
DevOps superstars, and books

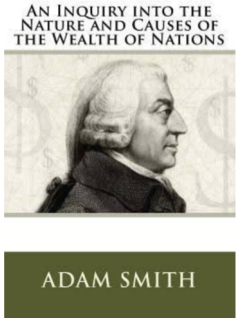


First DevOps book?

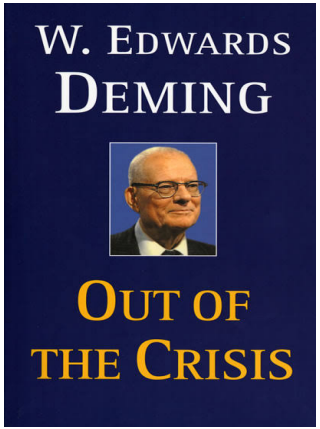
Adam Smith
1723-1790

First DevOps superstar?

Basic conditions
DevOps superstars, and books



Adam Smith
1723-1790

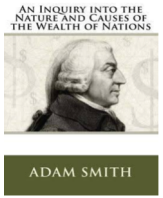


First DevOps book?

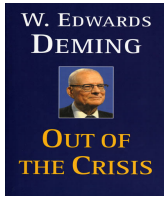
W. Edwards Deming
1900-1993

First DevOps superstar?

Basic conditions
DevOps superstars, and books



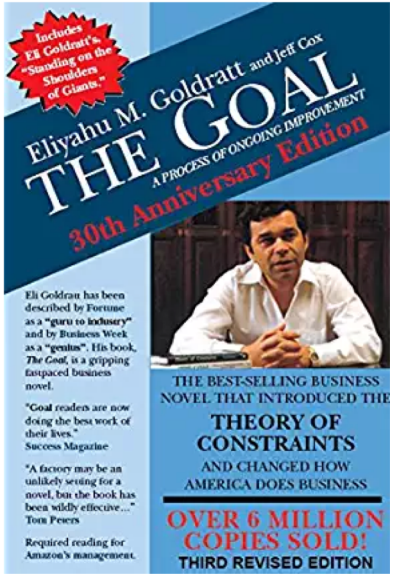
Adam Smith
1723-1790



W. Edwards Deming
1900-1993

First DevOps book/superstar?

First DevOps book?



Eliyahu Moshe Goldratt
1947–2011

First DevOps superstar?

What is DevOps?

DevOps

- Holistic /shared goals
- Holistic /shared processes
- Holistic / shared tools

Cycle Time

- Spanning different functions (yep, it is holistic)
- Measuring the time from start to end of a process
- Creating your own definition helps

  Example: Time from Git Push to availability in production

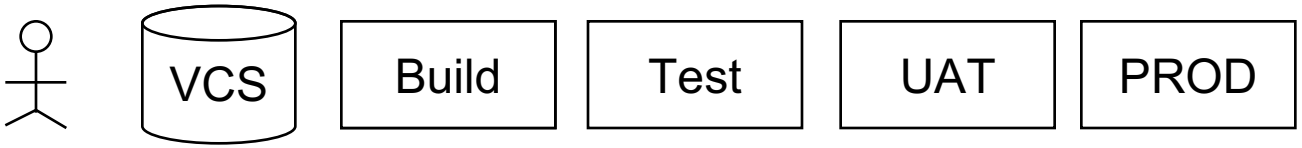
- Managing, with pipelines
- Setting this into context (often a tradeoff)

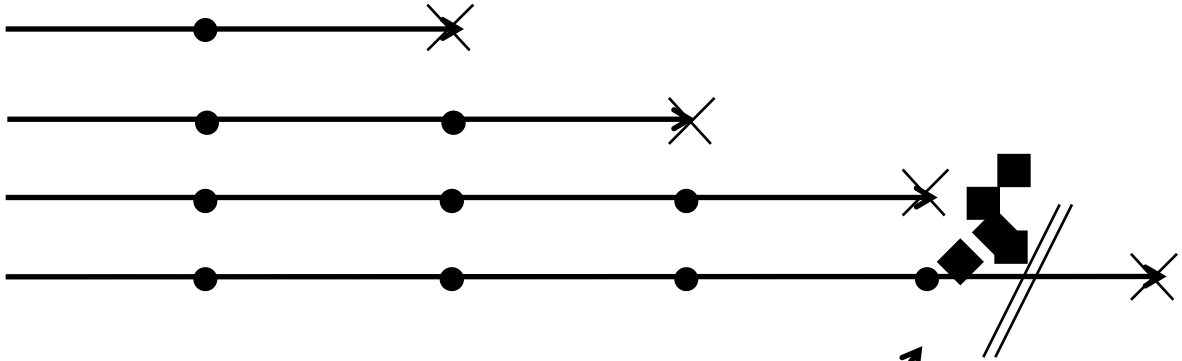
Agenda

- Setting the stage
- Deriving pipelines (and the cycle time)
- Demoing a roundtrip

Continuous Delivery
Start with a value stream map

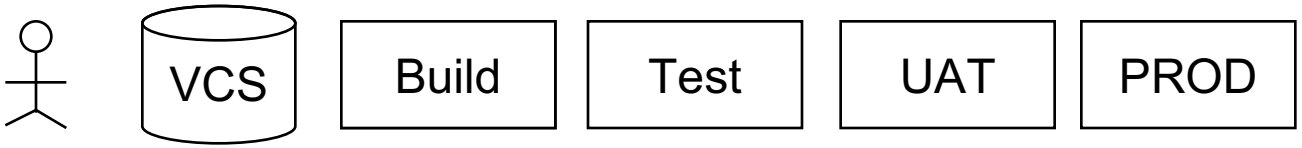
Accelerate Cycle Time, Part I
DevOps: Set of goals, processes, tools

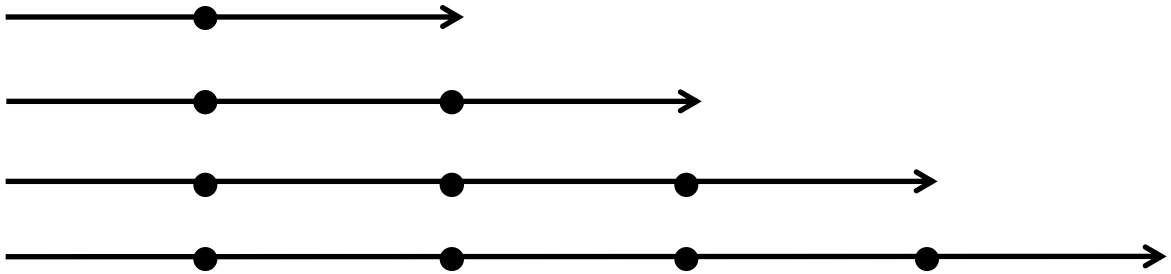


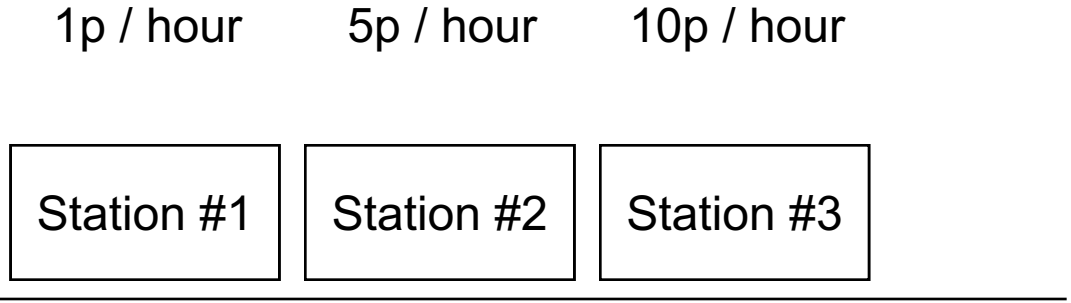


Here (missing) DevOps becomes often obvious!

Accelerate Cycle Time, Part I
DevOps: Set of goals, processes, tools

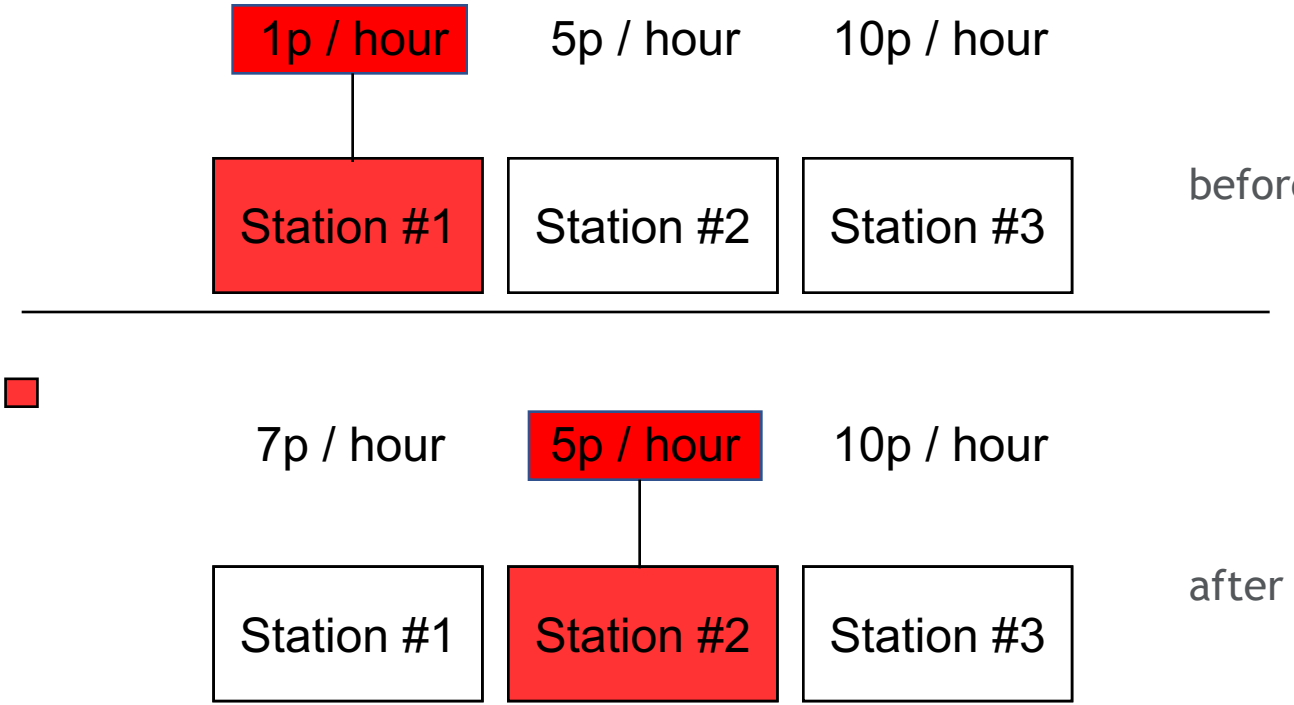






"a chain is no stronger than its weakest link"

Accelerate Cycle Time, Part II
Bottlenecks are normal!



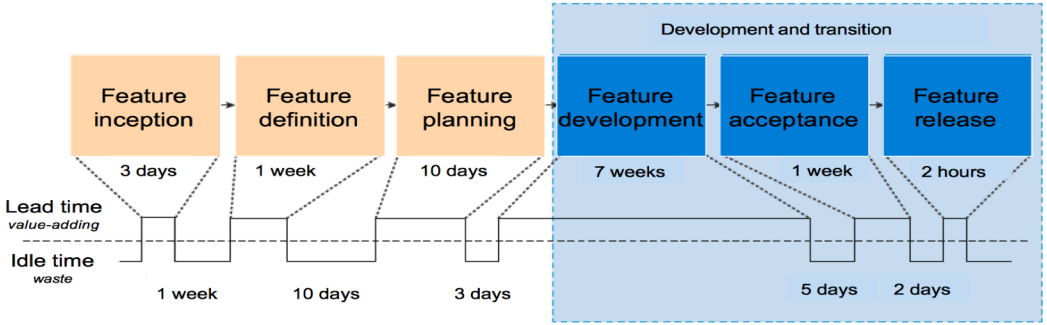
The Theory of Constraints!

Accelerate Cycle Time, Part III
Critical success factors

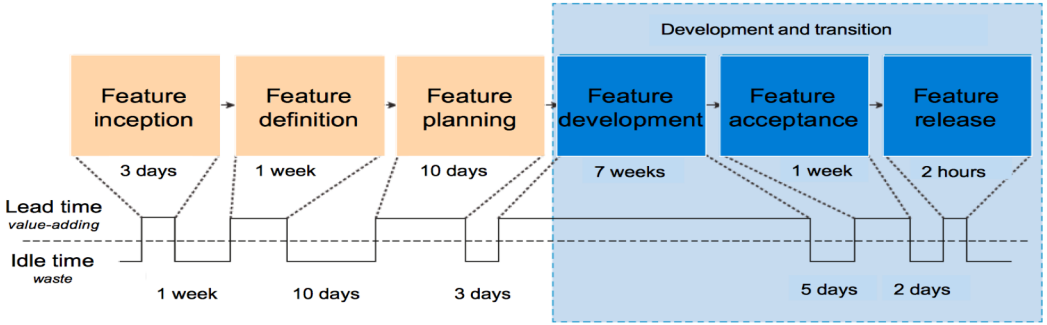
- Form pipelines to be doughnuts, not tubes
- Glue together existing tools, e.g. based on Jenkins
- High quality is a must! (Utilize quality gates)
- Implement high degree of automation[1]
- Form functional+technical consistent releases
- Eliminate local optimization, strive for holistic approaches

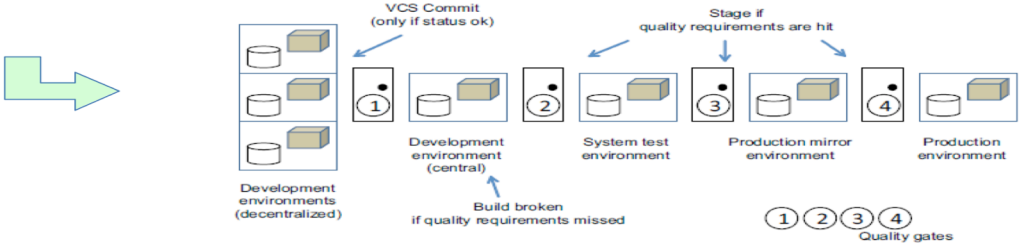
[1] Consider the “Pitfalls of Automation”, see Hüttermann, *DevOps for Developers* (Apress, 2012), pg. 41

Continuous Delivery
Visualize and optimize cycle time

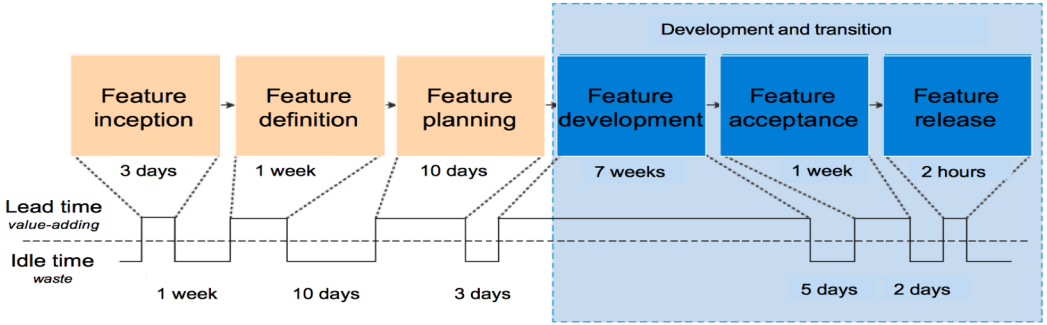


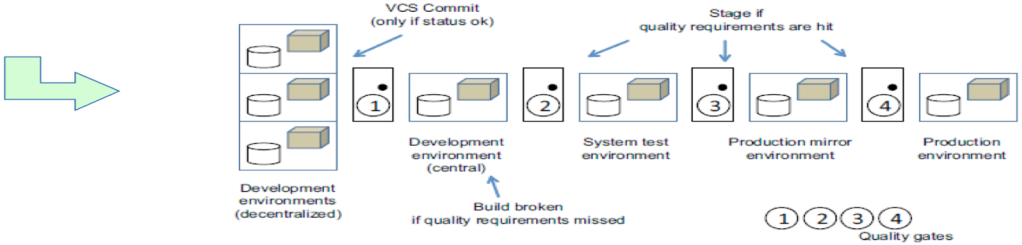
Continuous Delivery
Visualize and optimize cycle time

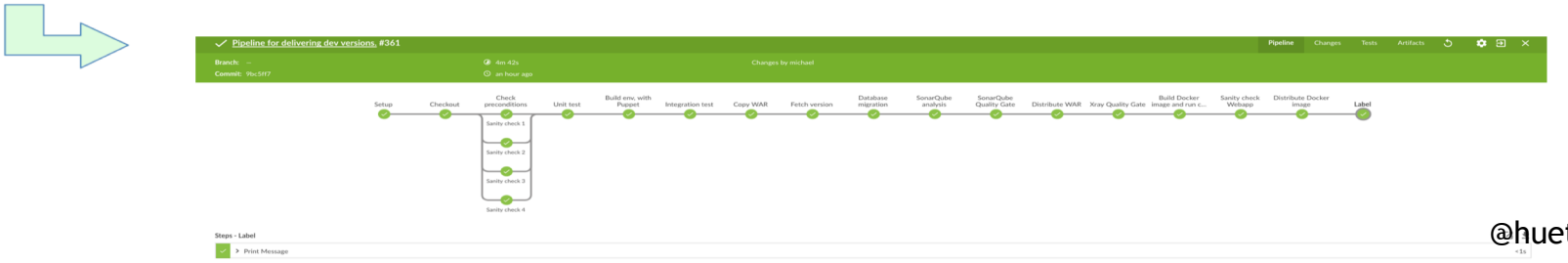




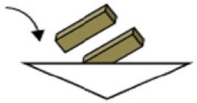
Continuous Delivery
Visualize and optimize cycle time

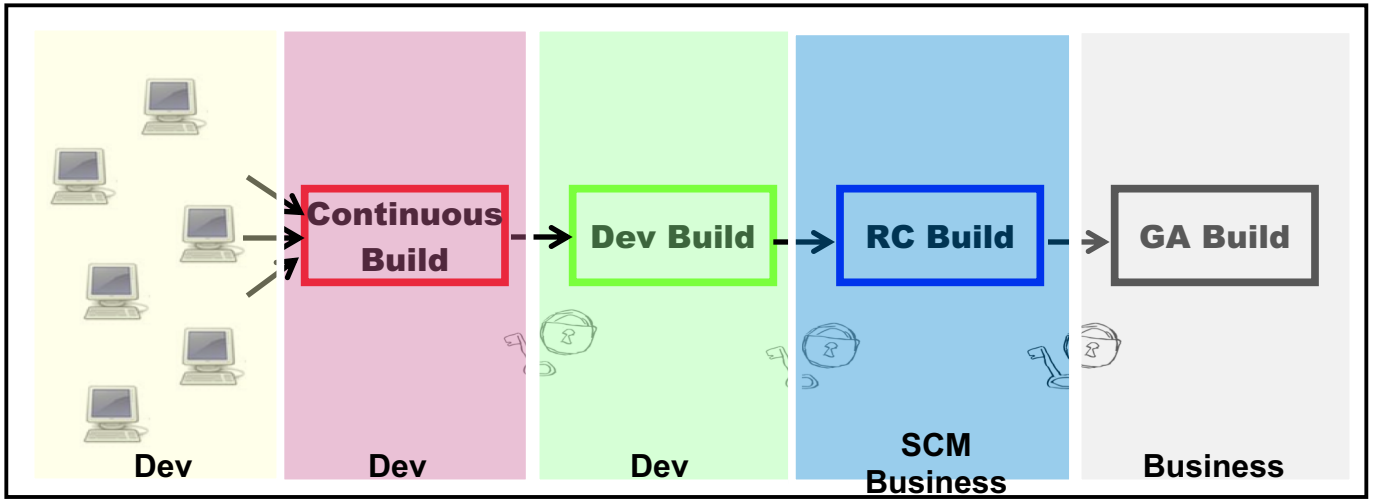


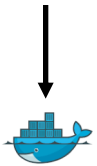


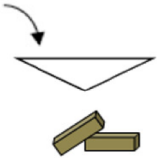


Continuous Delivery
Workflow, and different environments / build types







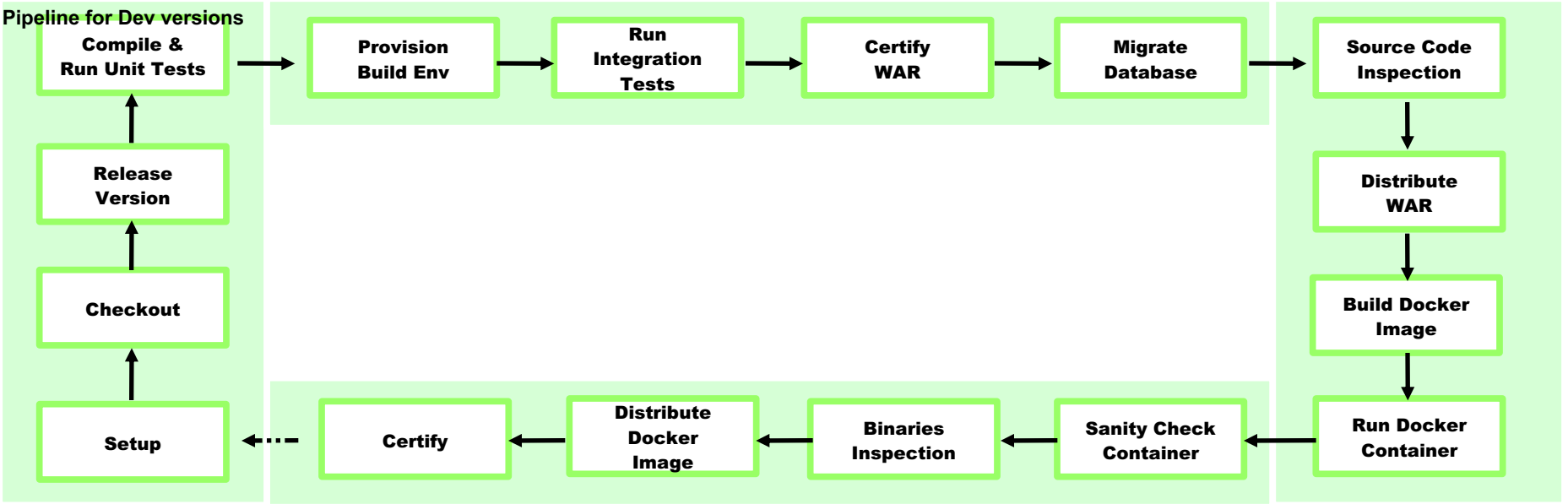



Checkout
Compile
Run
Unit Tests

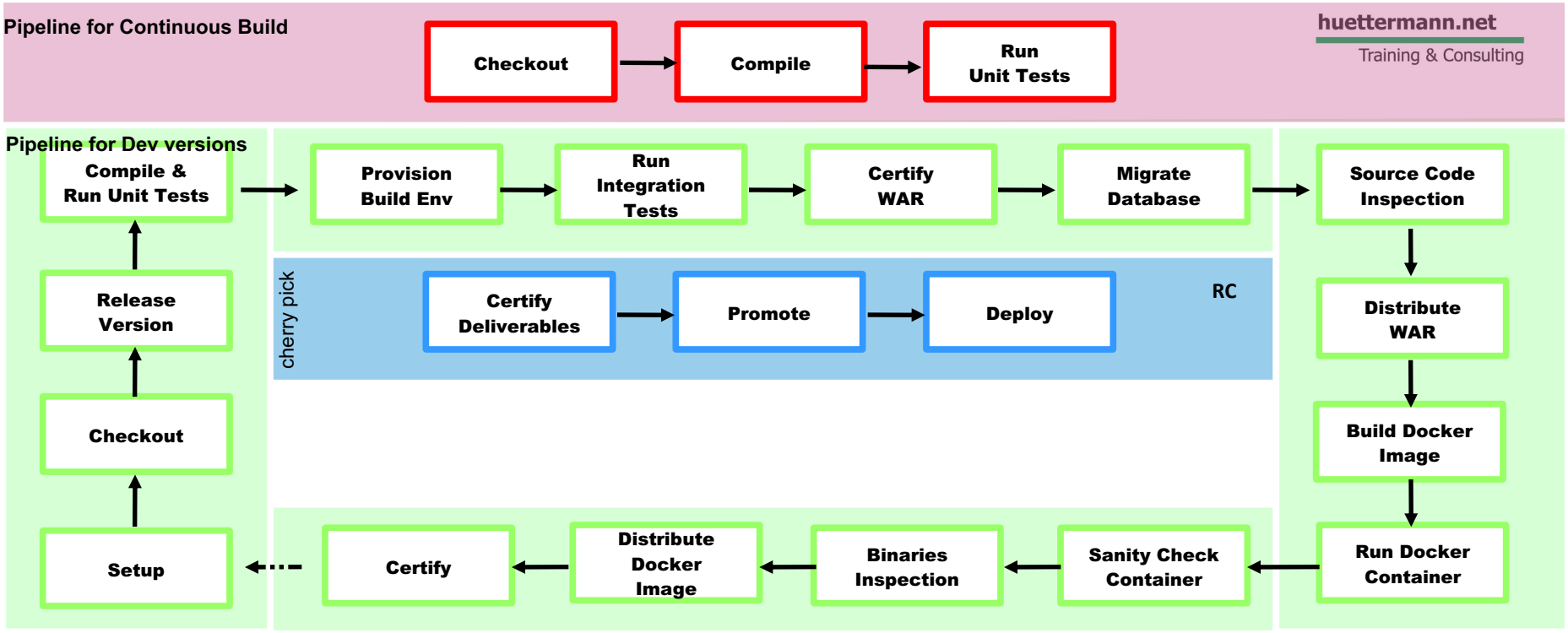
Pipeline for Continuous Build

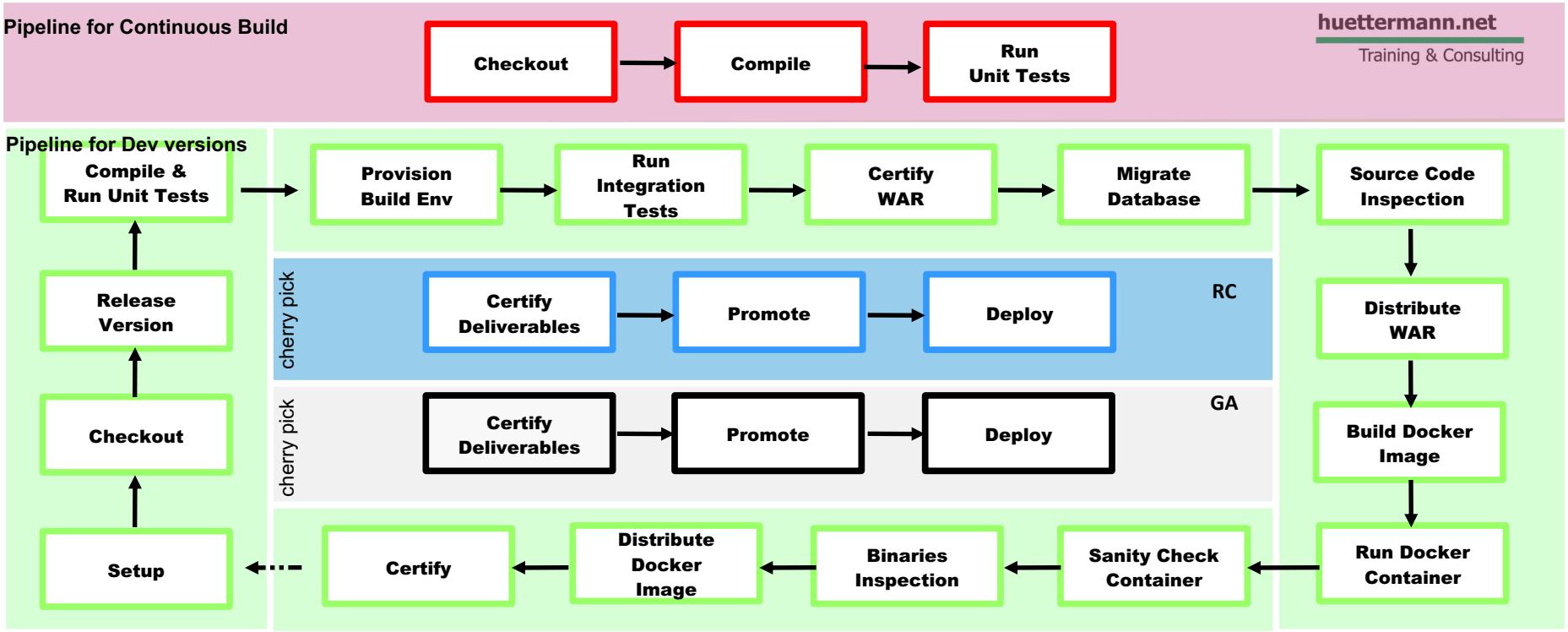
huettermann.net
Training & Consulting

Pipeline for Dev versions



@huettermann


Pipeline for Continuous Build
Checkout
Compile
Run Unit Tests
huettermann.net
Training & Consulting
Pipeline for Dev versions
Compile & Run Unit Tests
Provision Build Env
Run Integration Tests
Certify WAR
Migrate Database
Source Code Inspection
Release Version
cherry pick
Certify Deliverables
Promote
Deploy
RC
Distribute WAR
Checkout
Build Docker Image
Setup
Certify
Distribute Docker Image
Binaries Inspection
Sanity Check Container
Run Docker Container


Pipeline for Continuous Build
Checkout
Compile
Run Unit Tests
Pipeline for Dev versions
Compile & Run Unit Tests
Provision Build Env
Run Integration Tests
Certify WAR
Migrate Database
Source Code Inspection
Release Version
cherry pick
Certify Deliverables
Promote
Deploy
RC
Distribute WAR
Checkout
cherry pick
Certify Deliverables
Promote
Deploy
GA
Build Docker Image
Setup
Certify
Distribute Docker Image
Binaries Inspection
Sanity Check Container
Run Docker Container

Pipeline for Continuous Build

Checkout → Compile → Run Unit Tests

Pipeline for Dev versions

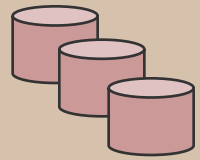
Compile & Run Unit Tests → Provision Build Env → Run Integration Tests → Certify WAR → Migrate Database → Source Code Inspection → Distribute WAR → Build Docker Image → Run Docker Container → Sanity Check Container → Binaries Inspection → Distribute Docker Image → Certify → Setup → Checkout → Release Version → Compile & Run Unit Tests

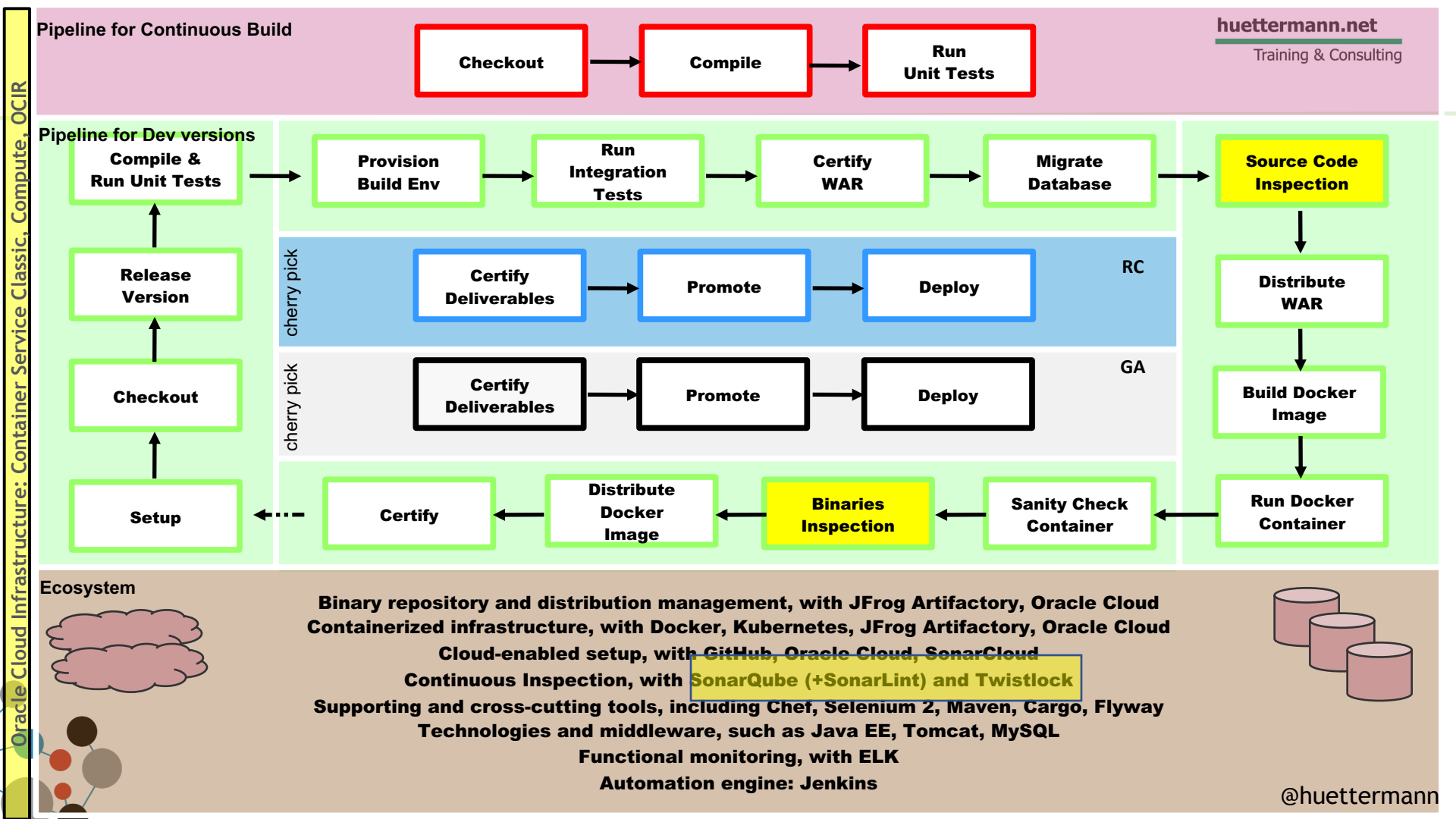
cherry pick (RC): Certify Deliverables → Promote → Deploy

cherry pick (GA): Certify Deliverables → Promote → Deploy

Ecosystem

Binary repository and distribution management, with JFrog Artifactory, Oracle Cloud
Containerized infrastructure, with Docker, Kubernetes, JFrog Artifactory, Oracle Cloud
Cloud-enabled setup, with GitHub, Oracle Cloud, SonarCloud
Continuous Inspection, with SonarQube (+SonarLint) and Twistlock
Supporting and cross-cutting tools, including Chef, Selenium 2, Maven, Cargo, Flyway
Technologies and middleware, such as Java EE, Tomcat, MySQL
Functional monitoring, with ELK
Automation engine: Jenkins




Oracle Cloud Infrastructure: Container Service Classic, Compute, OCIR
Pipeline for Continuous Build
Checkout
Compile
Run Unit Tests
huettermann.net
Training & Consulting
Pipeline for Dev versions
Compile & Run Unit Tests
Provision Build Env
Run Integration Tests
Certify WAR
Migrate Database
Source Code Inspection
Release Version
cherry pick
Certify Deliverables
Promote
Deploy
RC
Distribute WAR
Checkout
cherry pick
Certify Deliverables
Promote
Deploy
GA
Build Docker Image
Setup
Certify
Distribute Docker Image
Binaries Inspection
Sanity Check Container
Run Docker Container
Ecosystem
Binary repository and distribution management, with JFrog Artifactory, Oracle Cloud
Containerized infrastructure, with Docker, Kubernetes, JFrog Artifactory, Oracle Cloud
Cloud-enabled setup, with GitHub, Oracle Cloud, SonarCloud
Continuous Inspection, with SonarQube (+SonarLint) and Twistlock
Supporting and cross-cutting tools, including Chef, Selenium 2, Maven, Cargo, Flyway
Technologies and middleware, such as Java EE, Tomcat, MySQL
Functional monitoring, with ELK
Automation engine: Jenkins
@huettermann

- Identifies issues across languages
- Cloud native, with fine integration points, configurable
- Categories:
  - Vulnerabilities (security, common weakness enumeration CWE)
  - Reliability (bugs)
  - Maintainability (code smells)

- Identifies issues, based on primitives, transitively
- Cloud native, with fine integration points, configurable
- Container security platform, acquired by Palo Alto Networks
- Categories:
  - Vulnerability management (common vulnerabilities and exposures, CVE)
  - Compliance
  - Runtime defense

Pipeline for Continuous Build

Checkout → Compile → Run Unit Tests

Pipeline for Dev versions

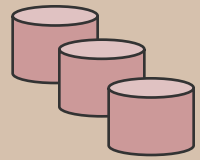
Compile & Run Unit Tests → Provision Build Env → Run Integration Tests → Certify WAR → Migrate Database → Source Code Inspection (1) → Distribute WAR → Build Docker Image → Run Docker Container → Sanity Check Container → Binaries Inspection (2) → Distribute Docker Image → Certify → Setup → Checkout → Release Version → Compile & Run Unit Tests

cherry pick: Certify Deliverables → Promote → Deploy (RC)

cherry pick: Certify Deliverables → Promote → Deploy (GA)

Ecosystem

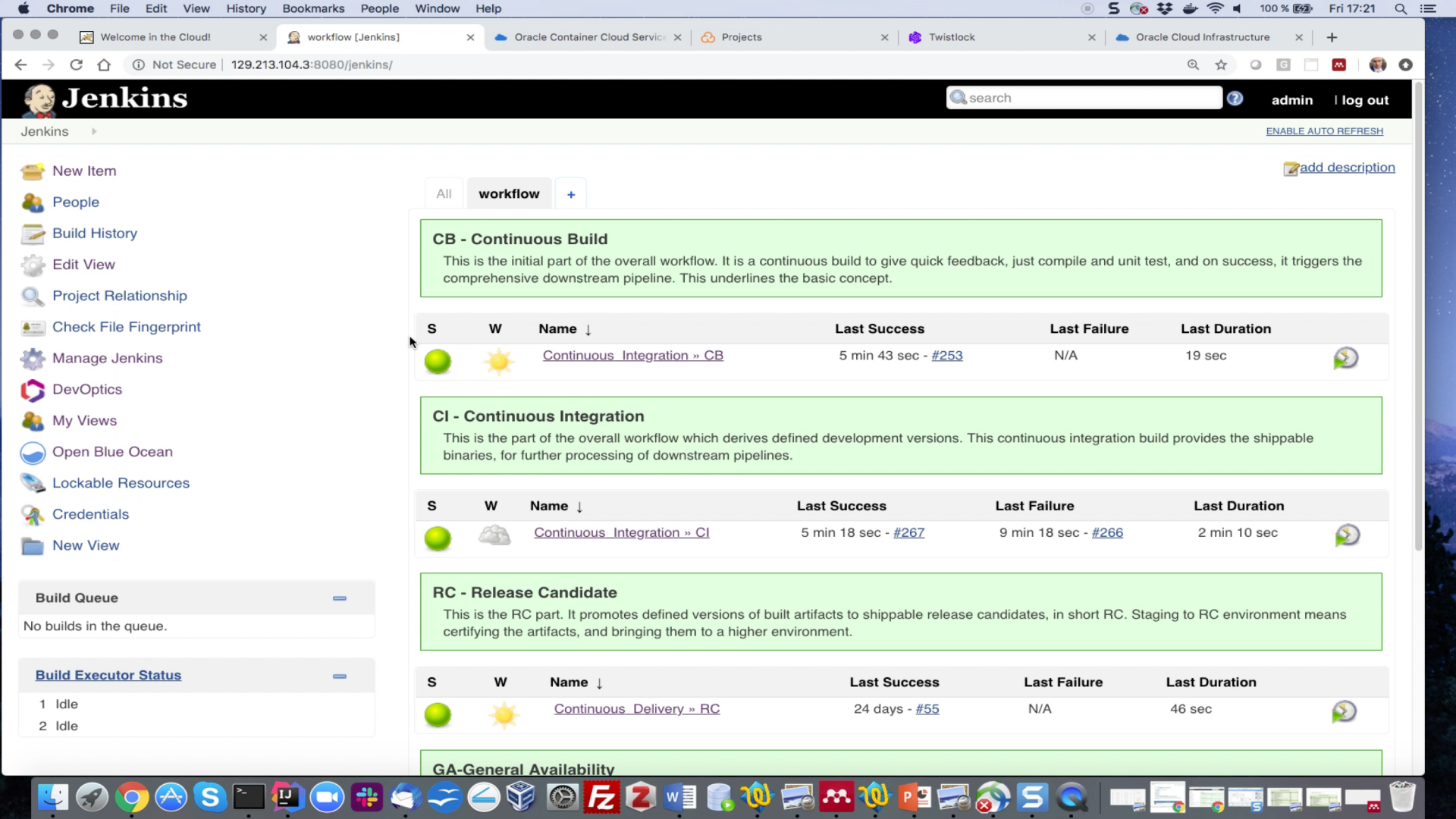
Binary repository and distribution management, with JFrog Artifactory, Oracle Cloud
Containerized infrastructure, with Docker, Kubernetes, JFrog Artifactory, Oracle Cloud
Cloud-enabled setup, with GitHub, Oracle Cloud, SonarCloud
Continuous Inspection, with SonarQube (+SonarLint) and Twistlock
Supporting and cross-cutting tools, including Chef, Selenium 2, Maven, Cargo, Flyway
Technologies and middleware, such as Java EE, Tomcat, MySQL
Functional monitoring, with ELK
Automation engine: Jenkins

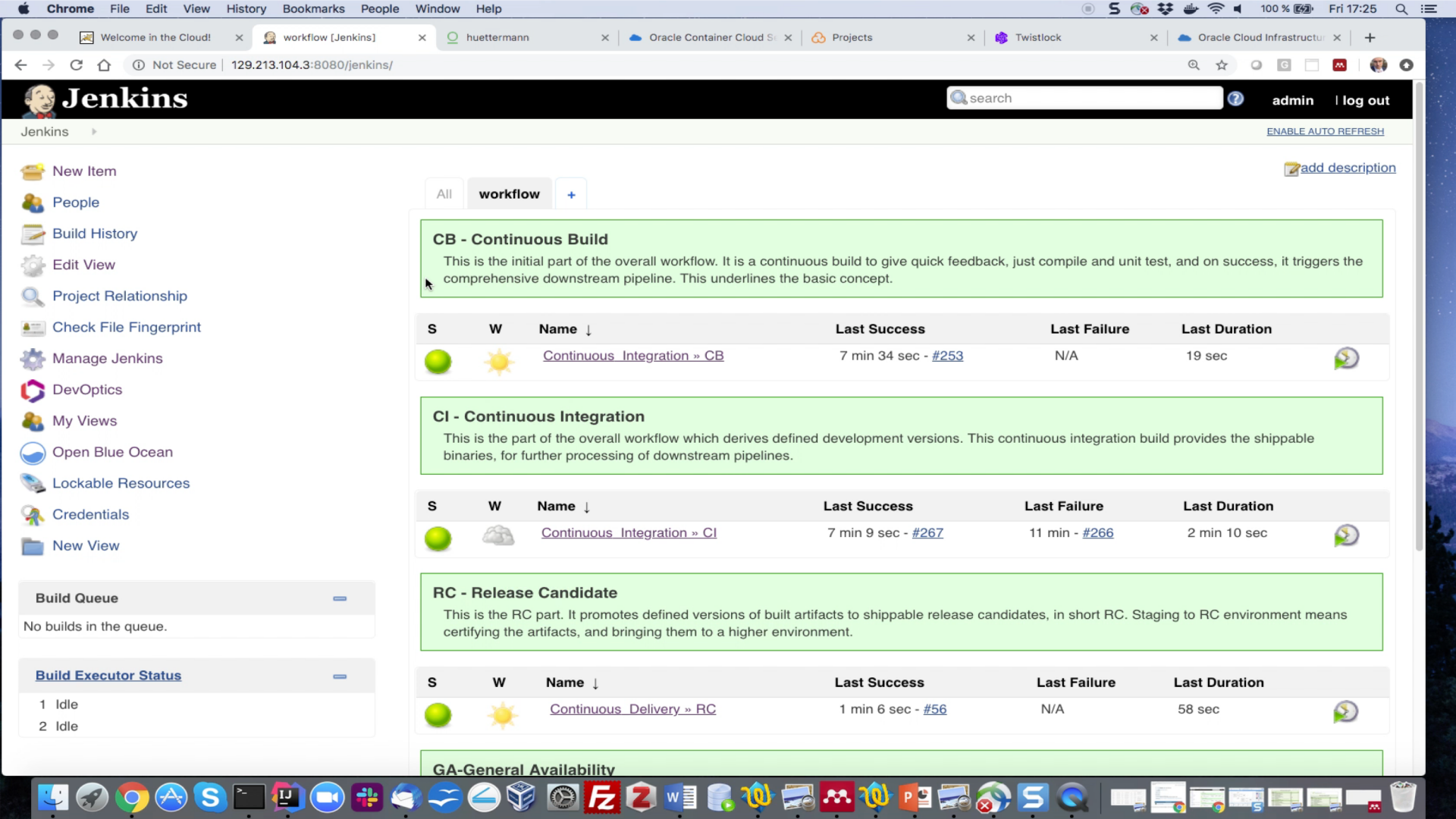


- Setting the stage
- Deriving pipelines (and the cycle time)
- Demoing a roundtrip

Meow app

Hello World ! Greetings from CCAA !!!


Chrome File Edit View History Bookmarks People Window Help
Fri 17:21
Welcome in the Cloud!
workflow [Jenkins]
Oracle Container Cloud Servic
Projects
Twistlock
Oracle Cloud Infrastructure
Not Secure | 129.213.104.3:8080/jenkins/
Jenkins
search
admin
log out
Jenkins
ENABLE AUTO REFRESH
New Item
People
Build History
Edit View
Project Relationship
Check File Fingerprint
Manage Jenkins
DevOptics
My Views
Open Blue Ocean
Lockable Resources
Credentials
New View
add description
All
workflow
+
CB - Continuous Build
This is the initial part of the overall workflow. It is a continuous build to give quick feedback, just compile and unit test, and on success, it triggers the comprehensive downstream pipeline. This underlines the basic concept.
S W Name ↓ Last Success Last Failure Last Duration
Continuous_Integration » CB 5 min 43 sec - #253 N/A 19 sec
CI - Continuous Integration
This is the part of the overall workflow which derives defined development versions. This continuous integration build provides the shippable binaries, for further processing of downstream pipelines.
S W Name ↓ Last Success Last Failure Last Duration
Continuous_Integration » CI 5 min 18 sec - #267 9 min 18 sec - #266 2 min 10 sec
RC - Release Candidate
This is the RC part. It promotes defined versions of built artifacts to shippable release candidates, in short RC. Staging to RC environment means certifying the artifacts, and bringing them to a higher environment.
S W Name ↓ Last Success Last Failure Last Duration
Continuous_Delivery » RC 24 days - #55 N/A 46 sec
GA-General Availability
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle


Chrome File Edit View History Bookmarks People Window Help
Fri 17:25
Welcome in the Cloud!
workflow [Jenkins]
huettermann
Oracle Container Cloud S
Projects
Twistlock
Oracle Cloud Infrastructur
Not Secure | 129.213.104.3:8080/jenkins/
Jenkins
search
admin
log out
Jenkins
ENABLE AUTO REFRESH
New Item
People
Build History
Edit View
Project Relationship
Check File Fingerprint
Manage Jenkins
DevOptics
My Views
Open Blue Ocean
Lockable Resources
Credentials
New View
add description
All
workflow
+
CB - Continuous Build
This is the initial part of the overall workflow. It is a continuous build to give quick feedback, just compile and unit test, and on success, it triggers the comprehensive downstream pipeline. This underlines the basic concept.
S
W
Name ↓
Last Success
Last Failure
Last Duration
Continuous_Integration » CB
7 min 34 sec - #253
N/A
19 sec
CI - Continuous Integration
This is the part of the overall workflow which derives defined development versions. This continuous integration build provides the shippable binaries, for further processing of downstream pipelines.
S
W
Name ↓
Last Success
Last Failure
Last Duration
Continuous_Integration » CI
7 min 9 sec - #267
11 min - #266
2 min 10 sec
RC - Release Candidate
This is the RC part. It promotes defined versions of built artifacts to shippable release candidates, in short RC. Staging to RC environment means certifying the artifacts, and bringing them to a higher environment.
S
W
Name ↓
Last Success
Last Failure
Last Duration
Continuous_Delivery » RC
1 min 6 sec - #56
N/A
58 sec
GA-General Availability
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle

Thank you.