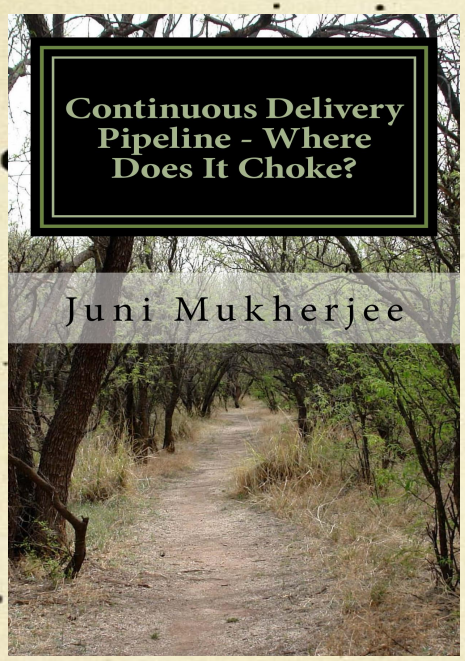
DevSecOps pipeline design patterns that can save the day

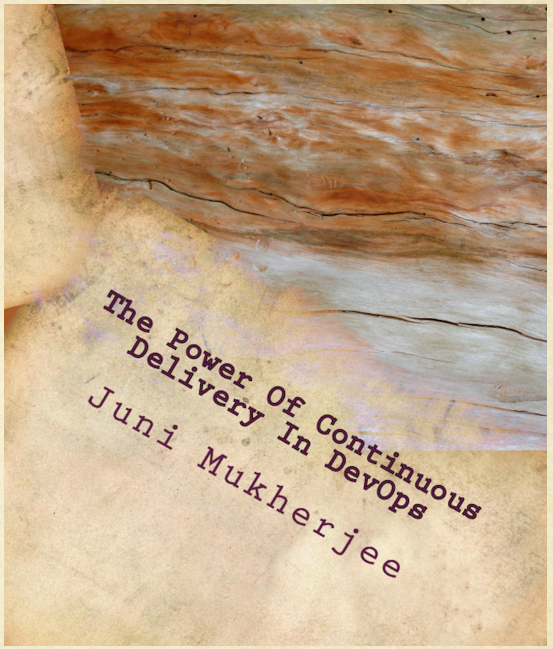
- Juni Mukherjee
- t: @JuniTweets @CloudBees
- w: https://continuity.world

Juni Mukherjee

https://continuity.world/gallery



https://continuity.world/2015-book



https://continuity.world/2017-book

9 savers for pipelines

1. Tool-phobia
2. Declarative vs. scripted pipelines
3. DevSecOps (DevOps)
4. Circuit-breaker pattern
5. Composition anti-pattern
6. Pipeline analytics and insights
7. Hand-off anti-pattern
8. Anti-corruption layer
9. Jenkins X

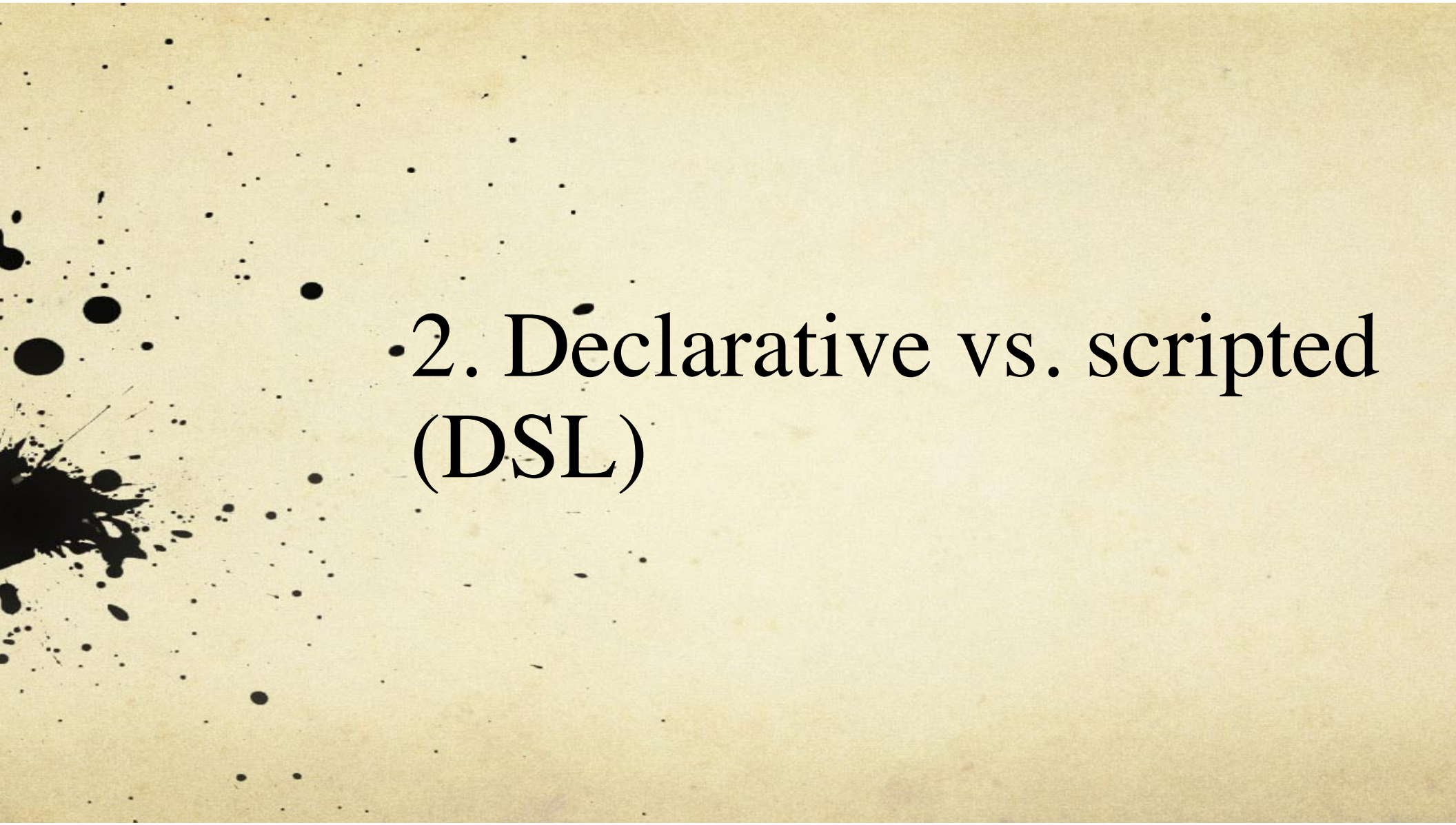
1. Tool-phobia

Continuous Everything …
integration | testing | delivery | deployment | everyone | ..

#	Class/category	Solution (Not meant to be exhaustive…)
1	Orchestrator	Jenkins, GitLab, GoCD, TeamCity, TravisCI, CodeShip, CodePipeline ..
2	Source code repo	GitHub, Bitbucket, CodeCommit, SVN, ..
3	Artifact repo	Artifactory, Nexus, S3, HockeyApp, ..
4	Dashboard	SumoLogic, ..
5	IaaS	AWS, Azure, ..
6	PaaS	CloudFoundry, Heroku, ..
7	Analytics	DevOptics, CloudWatch, New Relic, Dynatrace, Crashlytics, ..
8	Container ecosystem	Docker, CoreOS, Rocket, Swarm, Kubernetes, Mesos, ECS, ..
9	Audit trail	CloudTrail, .. **(never turn it off!)**
10	SAST	Coverity, …
11	DAST	OWASP ZAP, ..
12	Code coverage	Cobertura, JaCoCo, ..

Continuous Everything …
integration | testing | delivery | deployment | everyone | ..

#	Class/category	Solution (Not meant to be exhaustive…)
13	Static code analysis	Sonar, ESLint, Taylor, Lint, ..
14	Functional test	TestNG, Webdriver/Selenium, SauceLabs (Selenium on the cloud), Protractor (Node.js), Appium (Mobile), ..
15	Performance test	JMeter, BlazeMeter (JMeter on the cloud), ..
16	Unit test	JUnit (Java), Jasmine (Node.js), ..
17	Feature Flagging	LaunchDarkly, ..
18	A/B tests	Optimizely, ..
19	Build	Npm (Node.js), Maven(Java), Gradle(Java, Android), ..
20	Database	Liquibase/Datical, Flyway, ..
:::	::::::::::::::::::::::::::	::::::::::::::::::::::::::::::
:::	::::::::::::::::::::::::::	::::::::::::::::::::::::::::::
:::	::::::::::::::::::::::::::	::::::::::::::::::::::::::::::


2. Declarative vs. scripted (DSL)

Declare your pipeline | Maven

```
pipeline {
    agent { docker { image 'maven:3.3.3' } }
    stages {
        stage ('build') {
            steps {
                sh 'mvn --version'
            }
        }
    }
}
```

Declare your pipeline | Node.js

```
pipeline {
    agent { docker { image 'node:6.3' } }
    stages {
        stage ('build') {
            steps {
                sh 'npm --version'
            }
        }
    }
}
```

Declare your pipeline | Python

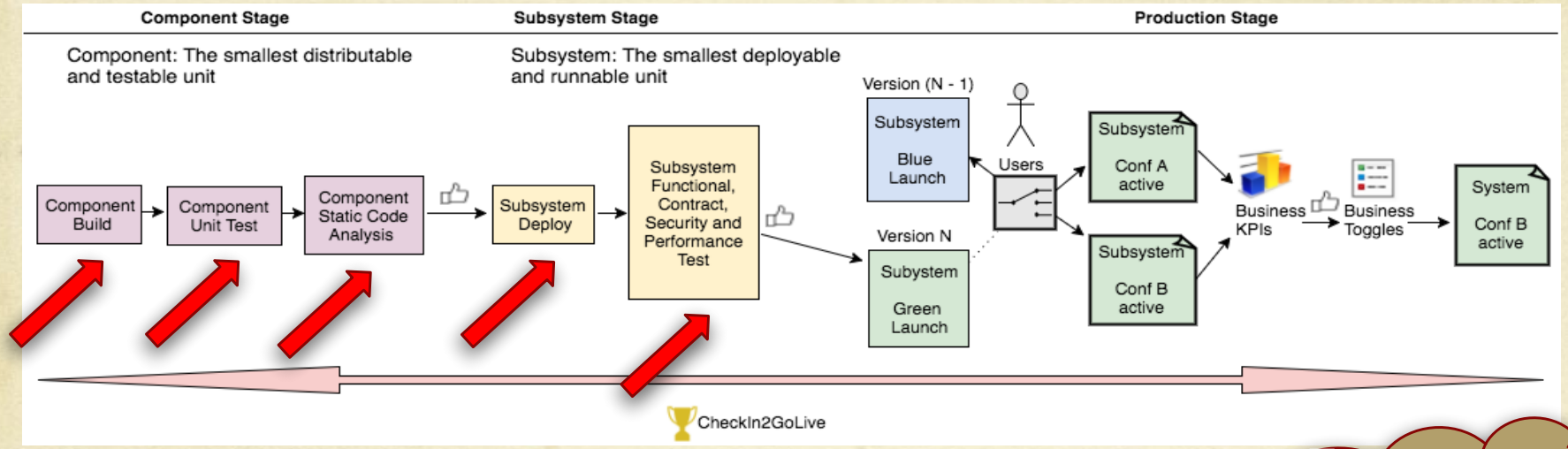
```
pipeline {
    agent { docker { image 'python:3.5.1' } }
    stages {
        stage ('build') {
            steps {
                sh 'python --version'
            }
        }
    }
}
```

3. DevOps or DevSecOps?

(or SecDevOps, BizDevOps, DevQARelOps?)

Juni Mukherjee | @JuniTweets, @CloudBees | https://continuity.world

OSS, Unit Test, SAST, Container, DAST



How do I assess my security posture?

For starters, are security specialists embedded in scrum teams?

Declare multiple stages in a pipeline

```
pipeline {
    agent none
    stages {
        stage('Build') {
            agent { docker 'maven:3-alpine' }
            steps {
                sh 'mvn --version'
            }
        }
        stage('Test') {
            agent { docker 'openjdk:8-jre' }
            steps {
                sh 'java -version'
            }
        }
    }
}
```

Declare multiple steps in a stage

```
pipeline {
    agent any
    stages {
        stage ('Happy Yoda') {
            steps {
                sh 'echo "Fear is the path to the dark side."'
                sh '''
                    echo "Fear leads to anger."
                    ls -la
                '''
            }
        }
    }
}
```

Tech poetry :P!

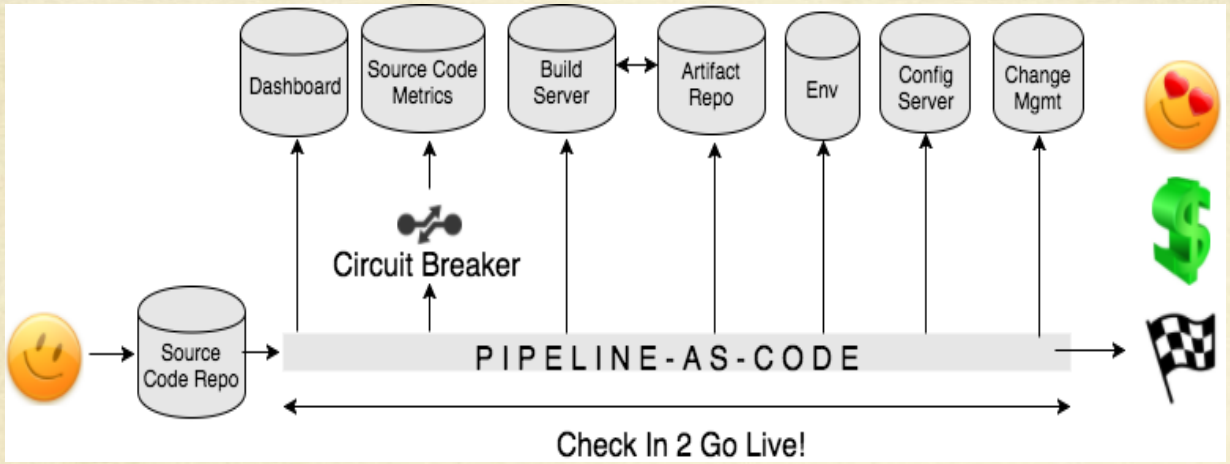
https://www.youtube.com/watch?v=ZZpbfabwpFU

Down the way, where the teams are gay (I mean happy!),
And the sun shines daily on the building top.
I took a trip on my Pipeline ship,
When I came to App Sec(urity), I made a stop.

Oh I am sad to say, I broke the Pipe (line) today,
It won't be up for many a day,
And Stage is down, and heads are turning around,
Explain how it happened in Stand-Up Town!!!

4. Circuit-breaker

Circuit-breaker pattern, (SIIPl*)aaS



Are my vendors' network topologies aligned?

Is my network topology optimized for CD?

Can I do Pipeline-as-conf?

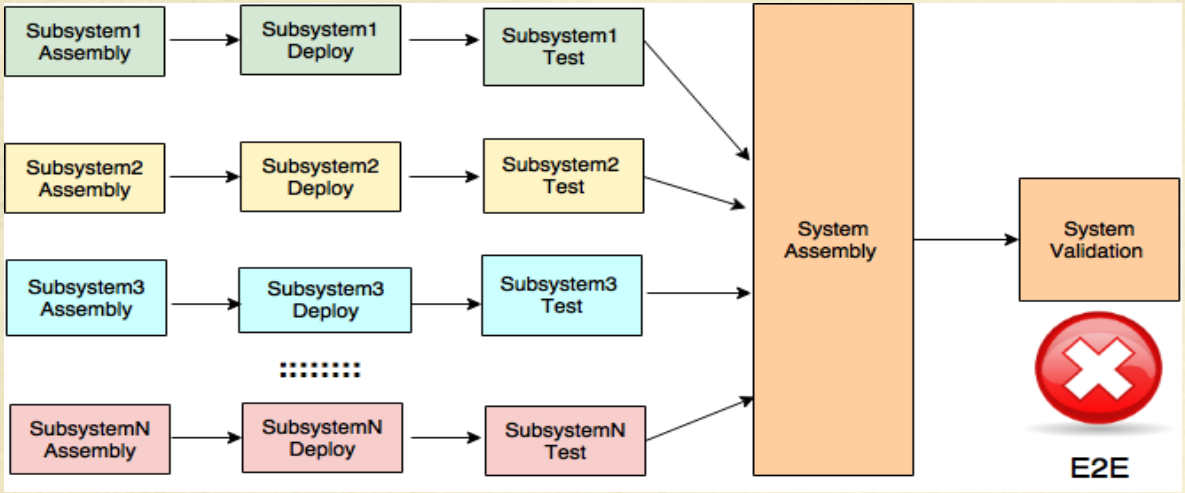
Protect and defend | Audit trail

"Try not. Do or do not. There is no try. --- Yoda"

```
pipeline {
    agent any
    options { timestamps () }
    stages {
        stage('Test') {
            steps {
                retry(3) {
                    sh '<< Connect to *aaS>>'
                }
                timeout(time: 10, unit: 'MINUTES') {
                    sh '<< Run web and mobile functional tests>>'
                }
            }
        }
    }
}
```

5. Composition/Assembly

Composition anti-pattern, Arch Coupling



Do my applications lend themselves to 12-factor?

Do I have monoliths vs. SOA vs. microservices?

How can I avoid a big ball of mud and a big ball of tests?

Do I have RoI to strangulate my whole monolith?

Am I stuffing everything into a container?

6. Pipeline analytics and insights

Biggest bang for the buck

CheckIn2Dev
CheckIn2Stage
CheckIn2GoLive!

Pipeline deploys on Dev
Pipeline deploys on Stage
Pipeline deploys in Production

Pipeline failures due to Unit Tests
Pipeline failures due to Functional Tests
Pipeline failures due to Performance Tests
Pipeline failures due to Security Vulnerabilities
::::::::::
::::::

Code Quality = Fn (
Cyclomatic Complexity,
Code Duplication,
Unit Test Coverage)

Do I provision Dev(1..M), DevInt(1..N), Perf(1..X), …?

Do I know how many environments I have vs. how many I need?

Show me the money!

Business Value Per Sprint

Not # of releases
Not # of tests executed

Stability Index=
Fn (CheckIn2GoLive,
# of Escaped Defects,
Customer Delight,
::::)

Pipeline failures in Dev
Pipeline failures in Stage
Pipeline failures in Production
# of Escaped Defects

Do teams have conflicting goals?

Are my KPIs departmental vs. organizational?

Do I trend speed and quality on the same canvas?

post from your pipeline | 'finally'

```
pipeline {
    agent any
    stages {
        stage('Test') {
            steps {
                sh '<< Carry out your execution >>'
            }
        }
    }
    post {
      // results, notices, logs, metrics, cleanups, ..
    }
}
```

Notify | Audit trail | Analytics

```
post {
    failure {
        mail to: butler@jenkinsworld.com, subject: 'What's for dinner?'
    }
}

post {
    success {
        slackSend channel: '#production-changes-room',
                  color: 'good',
                  message: "The pipeline ${currentBuild.fullDisplayName} completed
                                                    till production."
    }
}
```

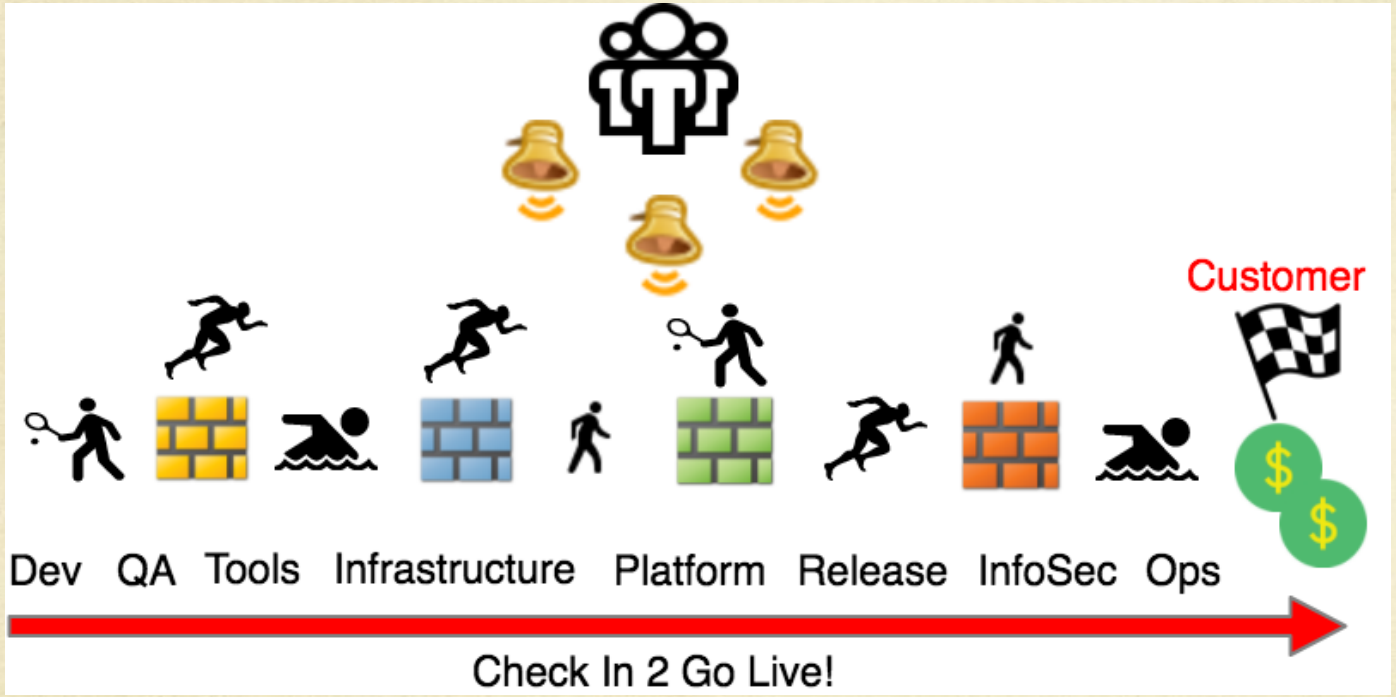
post | Pipeline Analytics & Insights

```
post {
    always {}          // just do it
    success{}          // do nothing
    failure {}         // open a ticket for X
    unstable {}        // open a ticket for Y
    changed {}         // open or resolve
    fixed {}           // resolve an open ticket
    aborted {}         // manually? by who?
    regression {}      // repeat offenders
}
```

7. Hand-off

Juni Mukherjee | @JuniTweets, @CloudBees | https://continuity.world

Hand-off anti-pattern, VSM, Flow, Drag



Automated waste > manual waste, but is still waste.

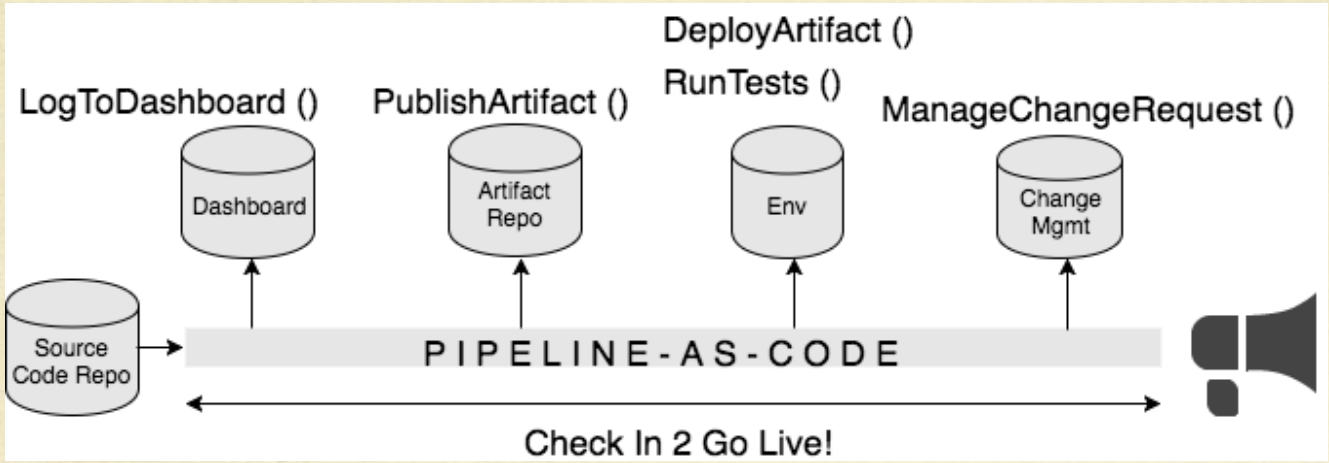
Do I seek or give sign-offs?

parallel | Optimize Time2Market

```
pipeline {
    agent any
    stages {
        stage('Deploy') {
            steps {}
        }
        stage('Run security and performance tests in parallel') {
        failFast true
            parallel {
                stage('SecurityTest') {
                    steps {}
                }
                stage('Performance Test') {
                    steps {}
                }
            }
        }
    }
}
```

8. Anti-corruption layer

Anti-corruption layer (ACL) pattern



Are my assets on-prem vs. cloudy?

Do I have RBAC? SoD.

input | CD vs. CD

```
pipeline {
    agent any
    stages {
        stage (’Ask an authorized human') {
            input {
                message ”Pipeline is no longer a pipe dream! Agree?"
                ok "Yep."
                submitter “Yoda, Butler”
            }
            steps {<< Deploy to production >>}
        }
    }
}
```

Managing secrets | RBAC | SoD

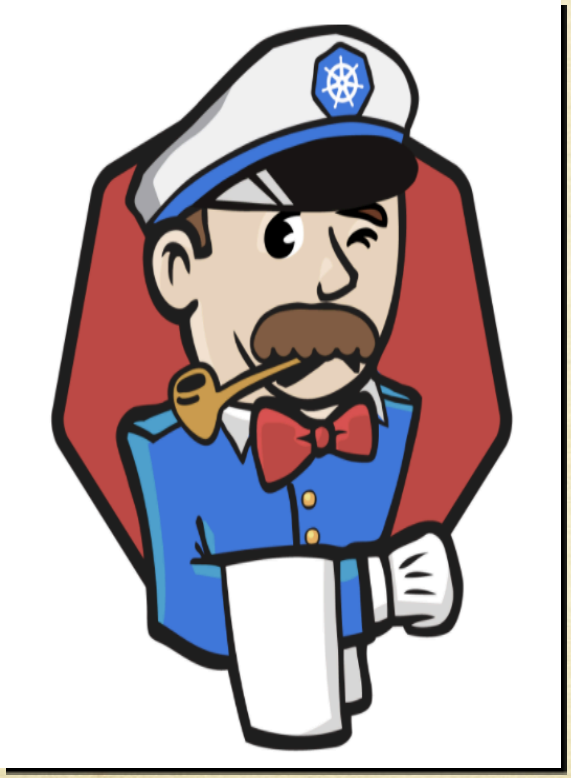
```
pipeline {
  environment {
      AWS_ACCESS_KEY_ID = credentials('jenkins-aws-secret-key-id')
      AWS_SECRET_ACCESS_KEY = credentials('jenkins-aws-secret-access-key')
   }
   stages {
      stage (’Publish artifacts to S3 and deploy to EC2’) {
          steps { echo $AWS_SECRET_ACCESS_KEY }   //***
      }
   }
}
```

Managing credentials

```
pipeline {
stages {
      stage (’Authenticate to APP1 and APP2') {
         steps {
            withCredentials(bindings: [sshUserPrivateKey(credentialsId: 'jenkins-ssh-key-for-
abc',
                  \
                  keyFileVariable: 'SSH_KEY_FOR_APP1’)]){ }
            withCredentials(bindings: [certificate(credentialsId: 'jenkins-certificate-for-xyz', \
                  keystoreVariable: 'CERTIFICATE_FOR_APP2', \
                  passwordVariable: 'XYZ-CERTIFICATE-PASSWORD')]) {}
         }
      }
   }
```

9. Jenkins X

Jenkins X

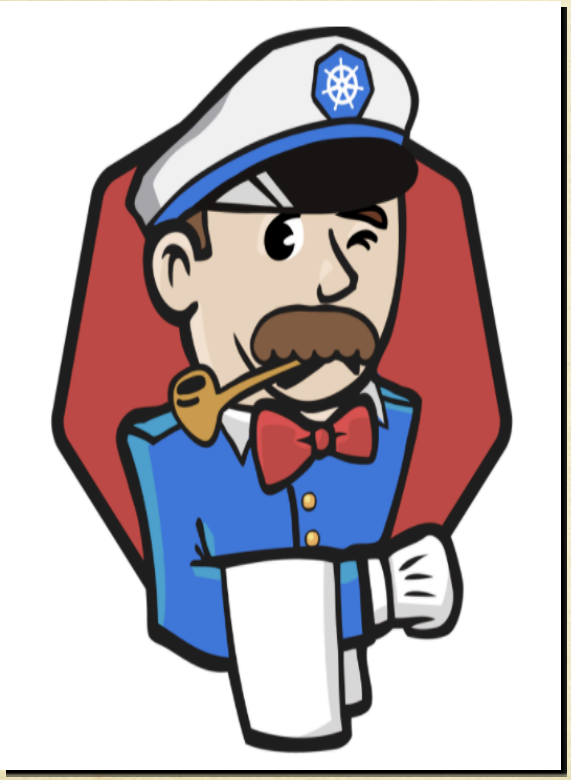


- CI/CD solution
- for modern cloud applications
- on Kubernetes

@jamestrachen

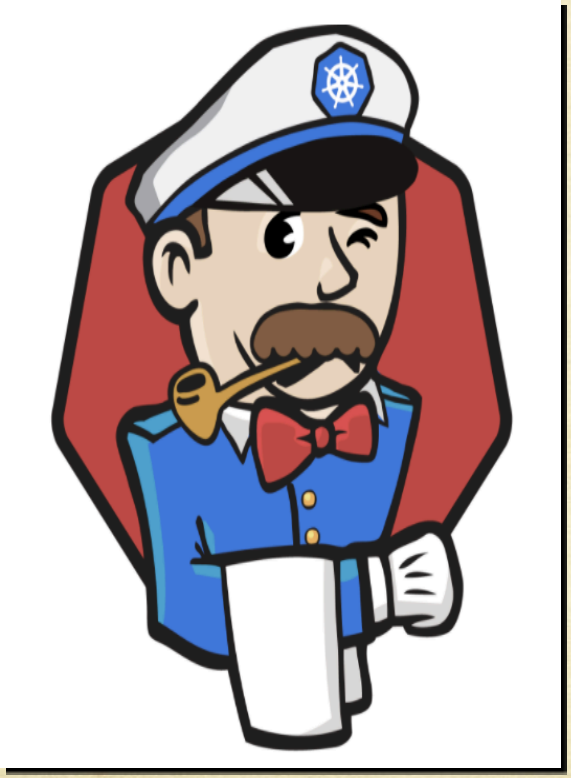
https://jenkins.io/blog/2018/03/19/introducing-jenkins-x/

Jenkins X – OOTB experience!



- Detect type of existing project, or quickstart
- GitHub integration
- Dockerfile to containerize apps
- Managing/configuring Jenkins or authoring Jenkinsfile for pipelines
- Helm charts
- GitOps
  - Deploy to a “staging” environment
  - Manual promotion to ”production”

Jenkins X – Pipe on!



https://github.com/jenkins-x

https://jenkins-x.io/getting-started/

https://jenkins-x.io/commands/jx/

https://jenkins-x.io/community/

https://kubernetes.slack.com/,
#jenkins-x-user|#jenkins-x-dev

Thank you @WeAreDevs ☺️

- Juni Mukherjee
- t: @JuniTweets @CloudBees
- w: https://continuity.world