



DevOps World



Jenkins World

Scaling developer onboarding

in Fortune 50 and smaller organizations
Sam Gleske – Integral Ad Science



integralads.com



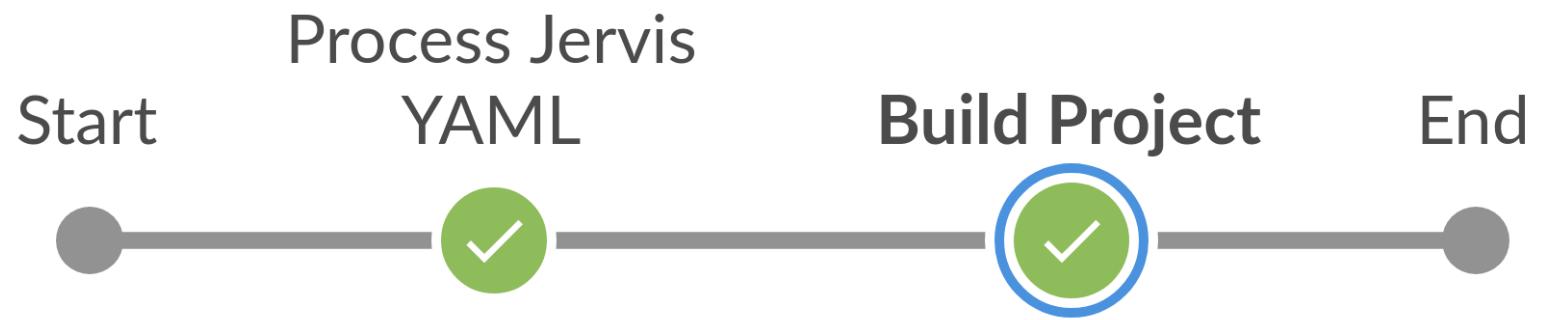
The developer onboarding experience

Eye candy

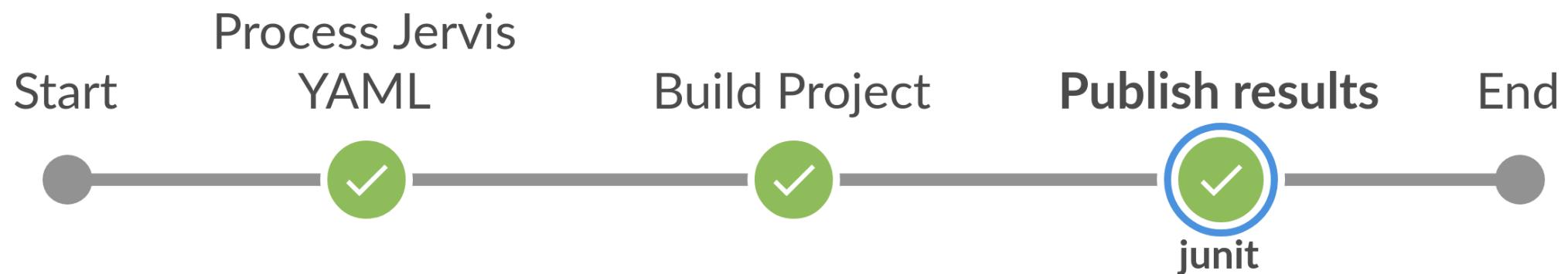
Basics of self service

- 1) Read documentation; add a basic YAML file to repository master branch.
- 2) Visit Jenkins and generate jobs with two button clicks.
- 3) Iterate and improve release process in a pull request.

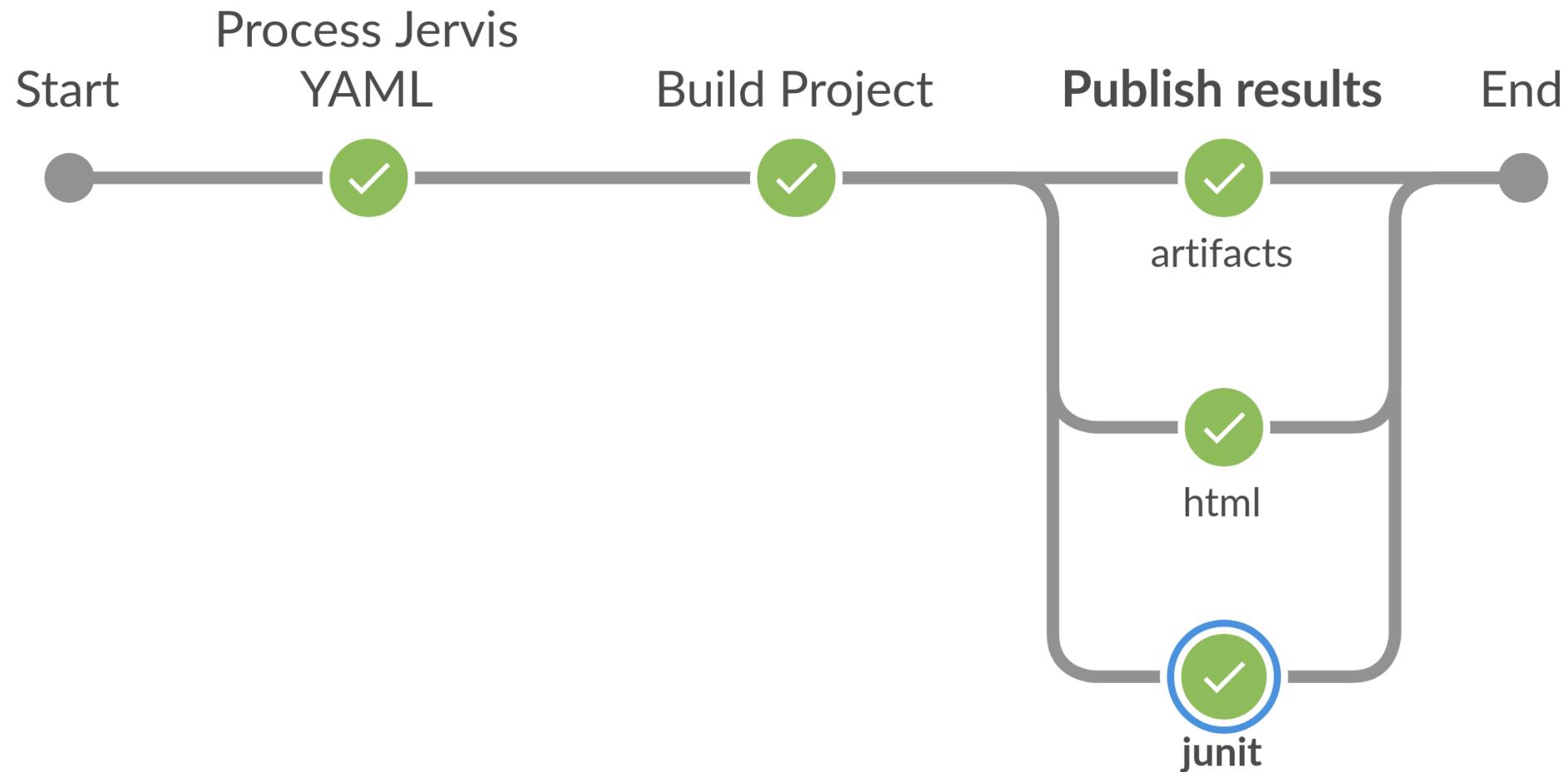
language: java



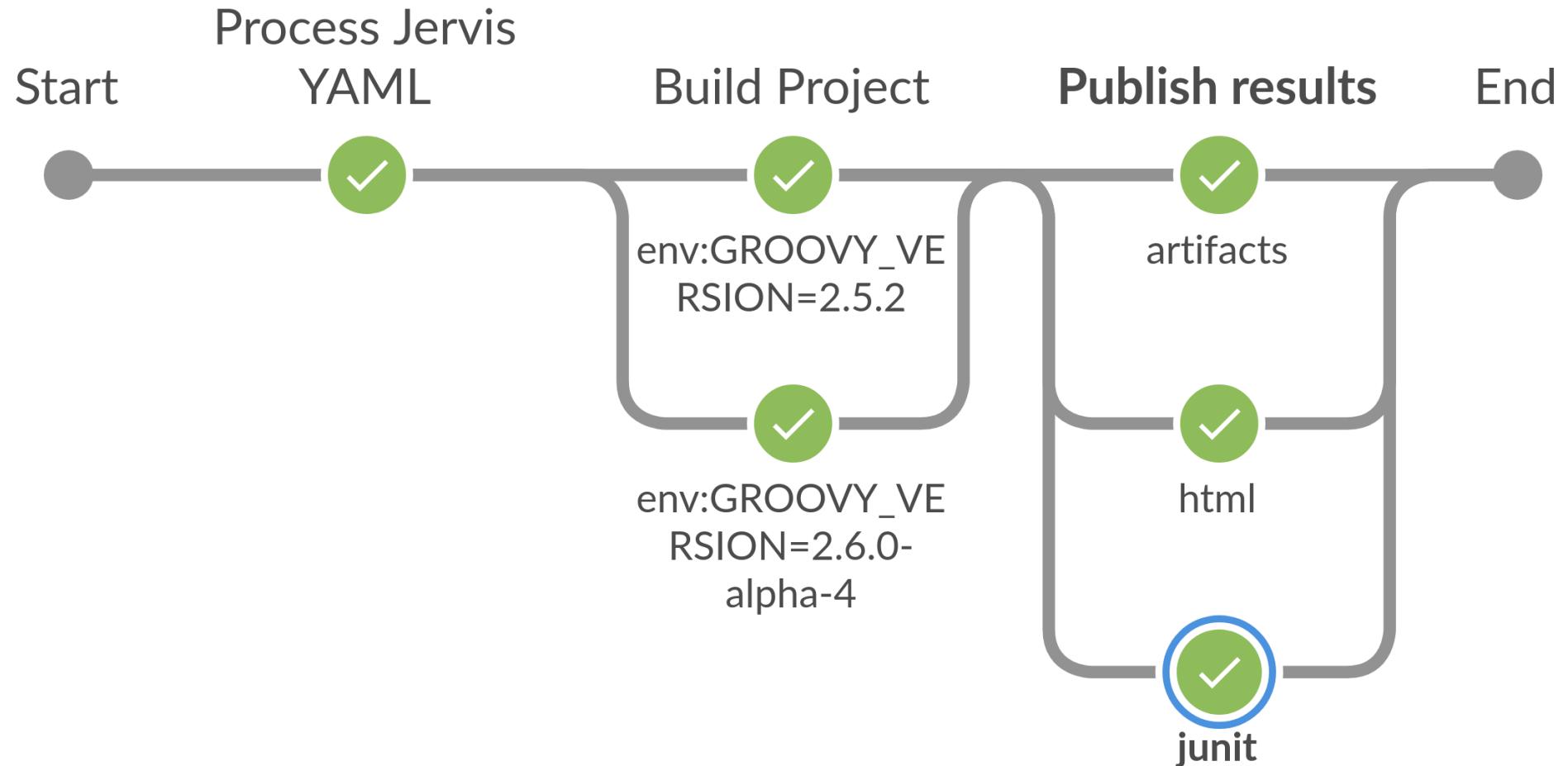
```
language: java
jenkins:
  collect:
    junit: build/test-results/**/*.xml
```



```
language: java
script: ./gradlew clean check groovydoc assemble
jenkins:
  collect:
    artifacts: build/libs/*.jar
    junit: build/test-results/**/*.xml
    html:
      path: build/docs/groovydoc
      reportName: 'groovy-gradle-seed API docs'
```



```
language: java
env:
  - GROOVY_VERSION=2.5.2
  - GROOVY_VERSION=2.6.0-alpha-4
script: ./gradlew clean check groovydoc assemble
jenkins:
  stash:
    - name: artifacts
      matrix_axis:
        env: GROOVY_VERSION=2.5.2
    - name: html
      matrix_axis:
        env: GROOVY_VERSION=2.5.2
    - name: junit
      matrix_axis:
        env: GROOVY_VERSION=2.5.2
collect:
  artifacts: build/libs/*.jar
  junit: build/test-results/**/*.*xml
  html:
    path: build/docs/groovydoc
    reportName: 'groovy-gradle-seed API docs'
```



Platforms

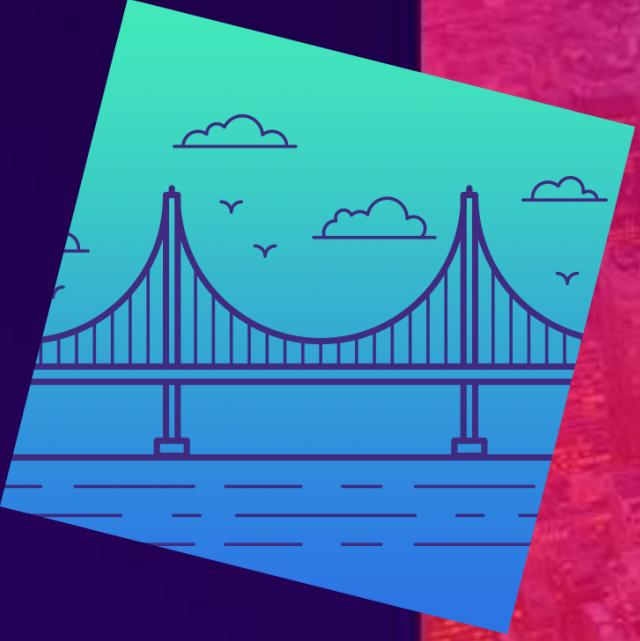
```
{  
    "defaults": {  
        "platform": "docker-in-docker",  
        "os": "alpinejdk8",  
        "stability": "stable",  
        "sudo": "sudo"  
    },  
    "supported_platforms": {  
        "docker-in-docker": {  
            "alpinejdk8": {  
                "language": [  
                    "java",  
                    "shell"  
                ],  
                "toolchain": [  
                    "env",  
                    "jdk"  
                ]  
            }  
        }  
    },  
    "restrictions": {}  
}
```

Toolchains

```
{  
    "toolchains": {  
        "java": ["env", "jdk"],  
        "shell": ["env"]  
    },  
    "env": {  
        "matrix": "advanced",  
        "*": [  
            "export ${jervis_toolchain_1value}"  
        ]  
    },  
    "jdk": {  
        "default_1value": "openjdk8",  
        "friendlyLabel": true,  
        "openjdk8": [  
            "export JAVA_HOME=/usr/lib/jvm/java-1.8-openjdk",  
            "export PATH="${JAVA_HOME}/bin:${PATH}",  
            "java -Xmx32m -version",  
            "javac -J-Xmx32m -version"  
        ]  
    }  
}
```

Lifecycles

```
{  
    "java": {  
        "friendlyName": "Java",  
        "defaultKey": "gradlew",  
        "gradlew": {  
            "fileExistsCondition": "gradlew",  
            "fallbackKey": "gradle",  
            "install": "./gradlew assemble",  
            "script": "./gradlew check"  
        },  
        "gradle": {  
            "fileExistsCondition": "build.gradle",  
            "fallbackKey": "maven",  
            "install": "gradle assemble",  
            "script": "gradle check"  
        },  
        "maven": {  
            "fileExistsCondition": "pom.xml",  
            "fallbackKey": "ant",  
            "install": "mvn install -DskipTests=true",  
            "script": "mvn test"  
        },  
        "ant": {  
            "fileExistsCondition": "build.xml",  
            "script": "ant test"  
        }  
    },  
    "shell": {  
        "friendlyName": "Shell",  
        "defaultKey": "none",  
        "none": {}  
    }  
}
```



Jenkins development announcements

New Jenkins Plugin: SCM Filter Branch PR

Filter by name including PRs destined for this branch (with wildcards)

Branch Include	master	(?)
Branch Exclude		(?)
Tag Include	*	(?)
Tag Exclude		(?)

Filter by name including PRs destined for this branch (with regular expression)

Branch regular expression	\Qmaster\E	(?)
Tag regular expression	.*	(?)

Coming Soon: Pipeline Multibranch With Defaults Sandbox

Build Configuration

Mode

by default Jenkinsfile



Script ID

Jenkinsfile

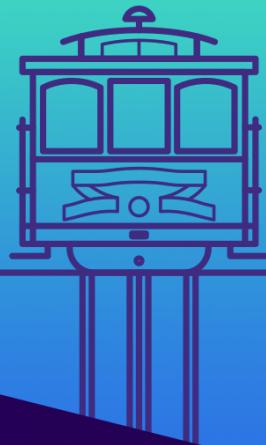


Run default Jenkinsfile within Groovy sandbox



Jervis has passed 1.0 release

- Job DSL library for generating jobs
- 100% multi-branch pipeline build execution
- Shared pipeline library provides **isPRBuild()** and **isTagBuild()** conditionals.
- Provides **IS_PR_BUILD** and **IS_TAG_BUILD** shell variables.
- Extensible
 - Modify Job DSL scripts via Groovy bindings.
 - Shared pipeline library via **hasGlobalVar()** pipeline step.



Development workflow
for delivering software
at scale

Assumptions (or life will be harder)

- Nobody wants to “learn Jenkins”
- Every project uses pinned versions of dependencies.
 - Java-esque (don’t rely on -SNAPSHOT)
 - NodeJS (npm shrinkwrap or just use yarn; if using gulp then same applies)
 - Python (pip freeze)
 - Ruby (Gemfile.lock)
- Cache dependencies from the Internet.
- Always think about idempotency. Not just for infrastructure but for release flows as well. Ask yourself questions:
 - What if someone re-runs this Jenkins pipeline? e.g. Rollbacks or branches.
 - What if the step I’m executing is already my desired state? Check state and skip it.

Slicing the explanation

- Code repository perspective (development flow)
- Events and interactions perspective
- System architecture perspective

Multi-Member Project Workflow (GitHub Flow With Releases)

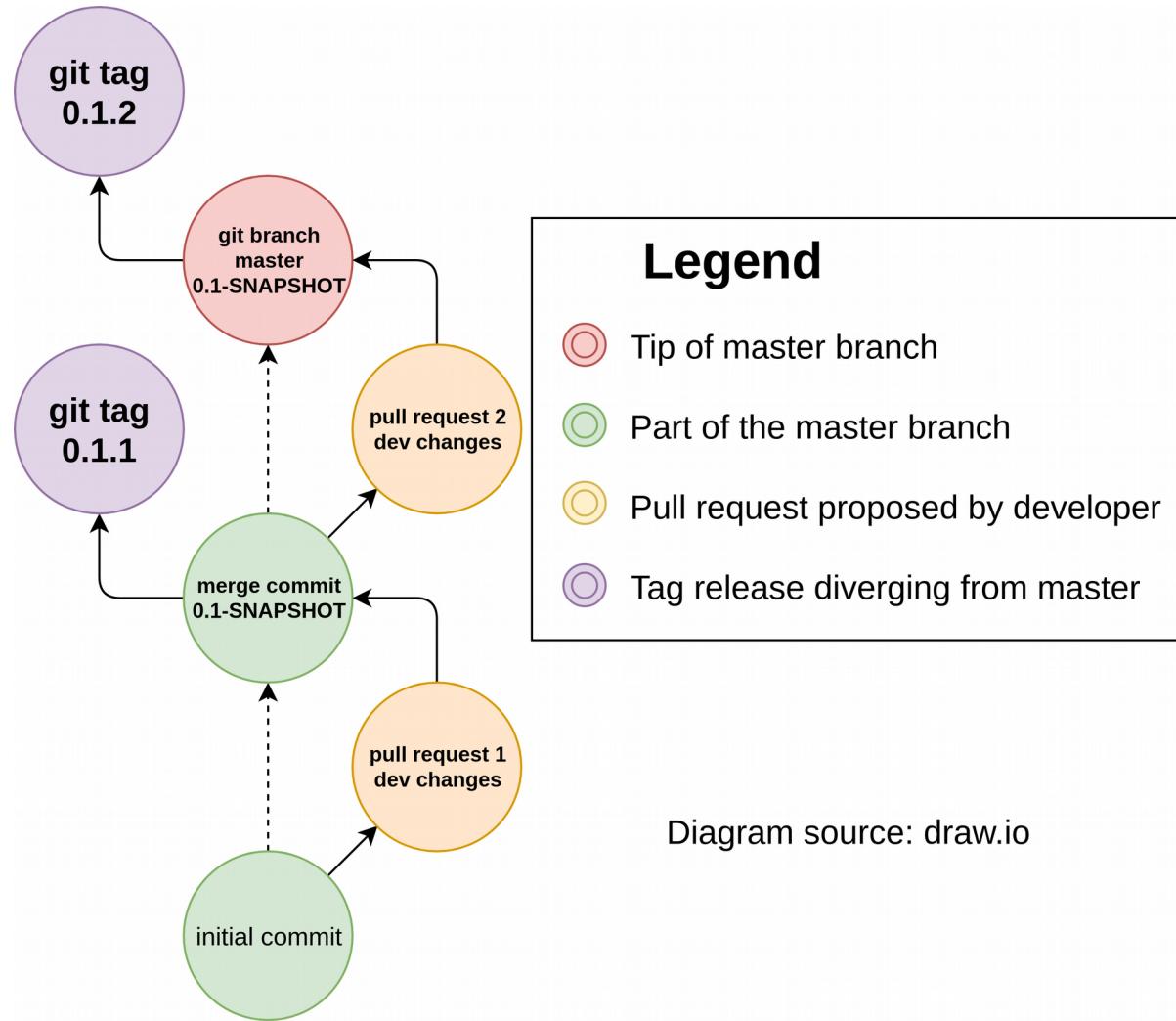
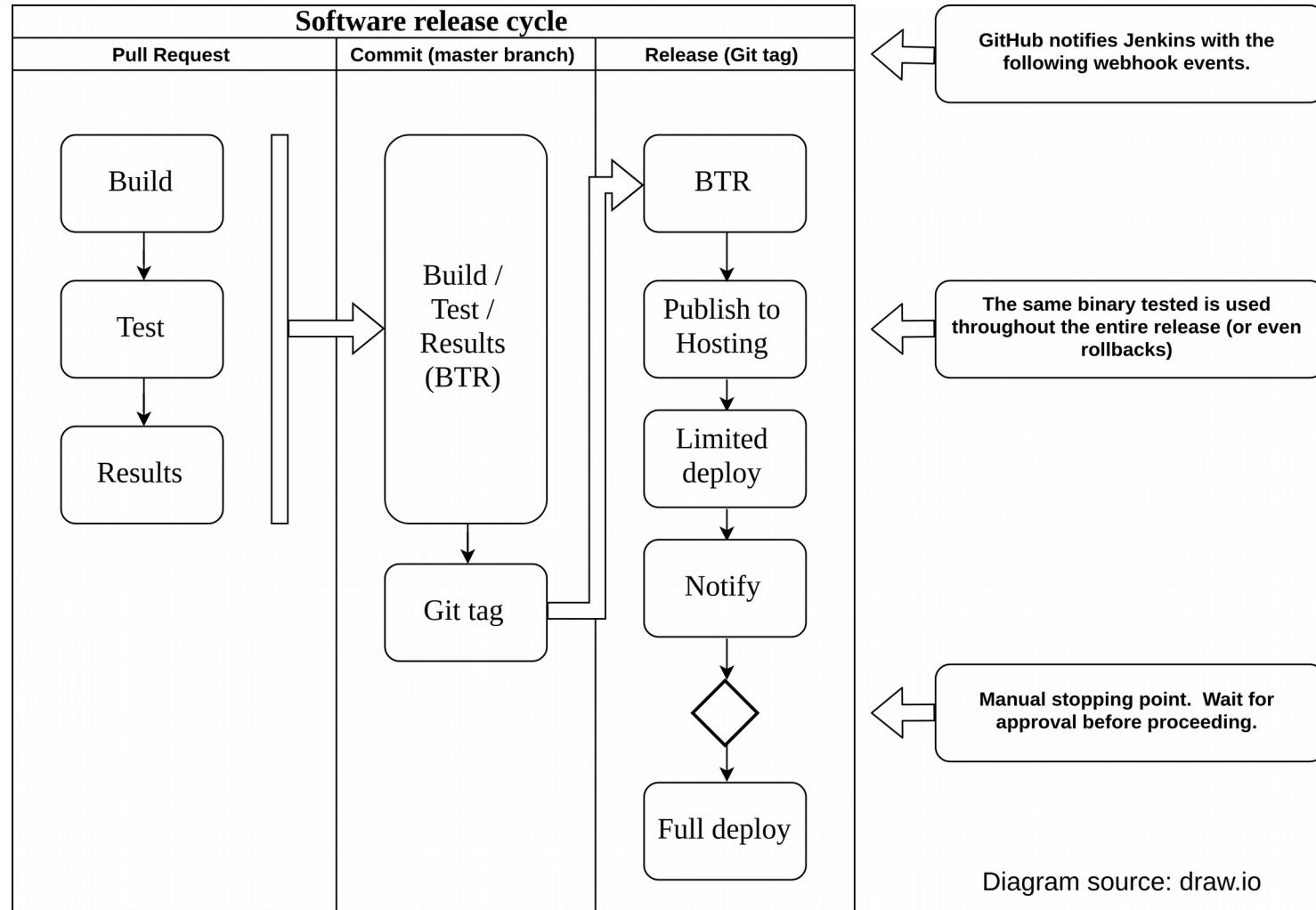


Diagram source: draw.io

Jenkins Multibranch Pipelines Running GitHub Flow



Overview of interactions between Jenkins and GitHub

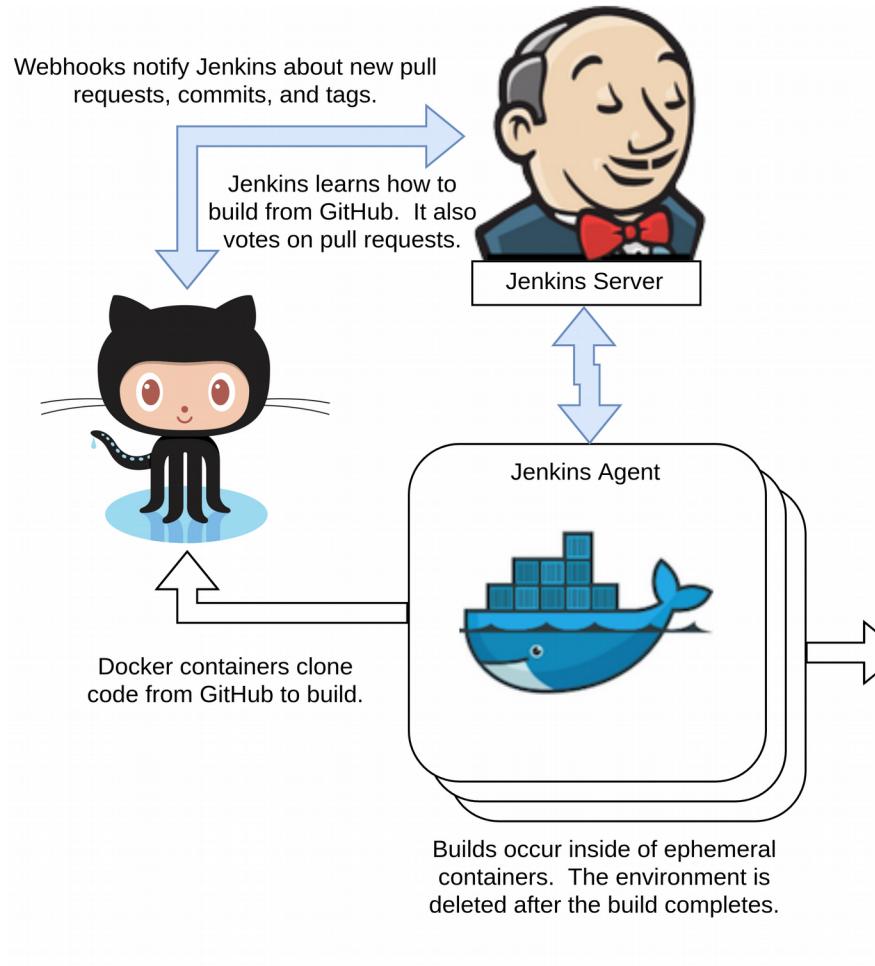
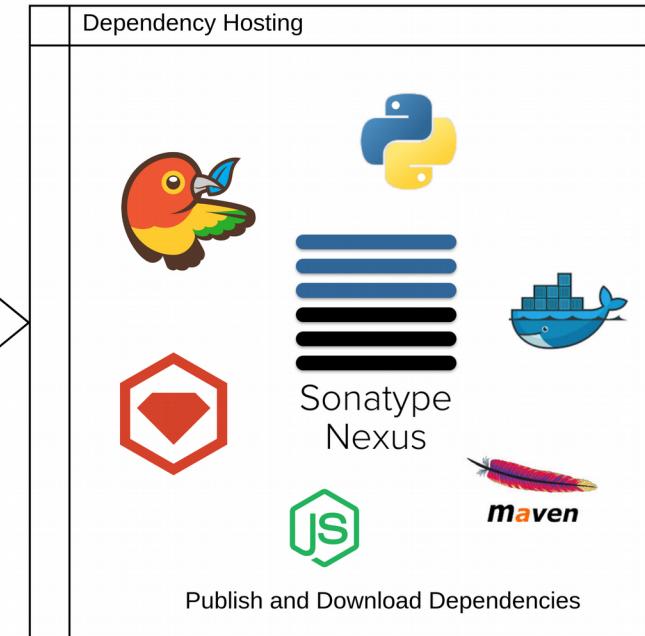


Diagram source: draw.io





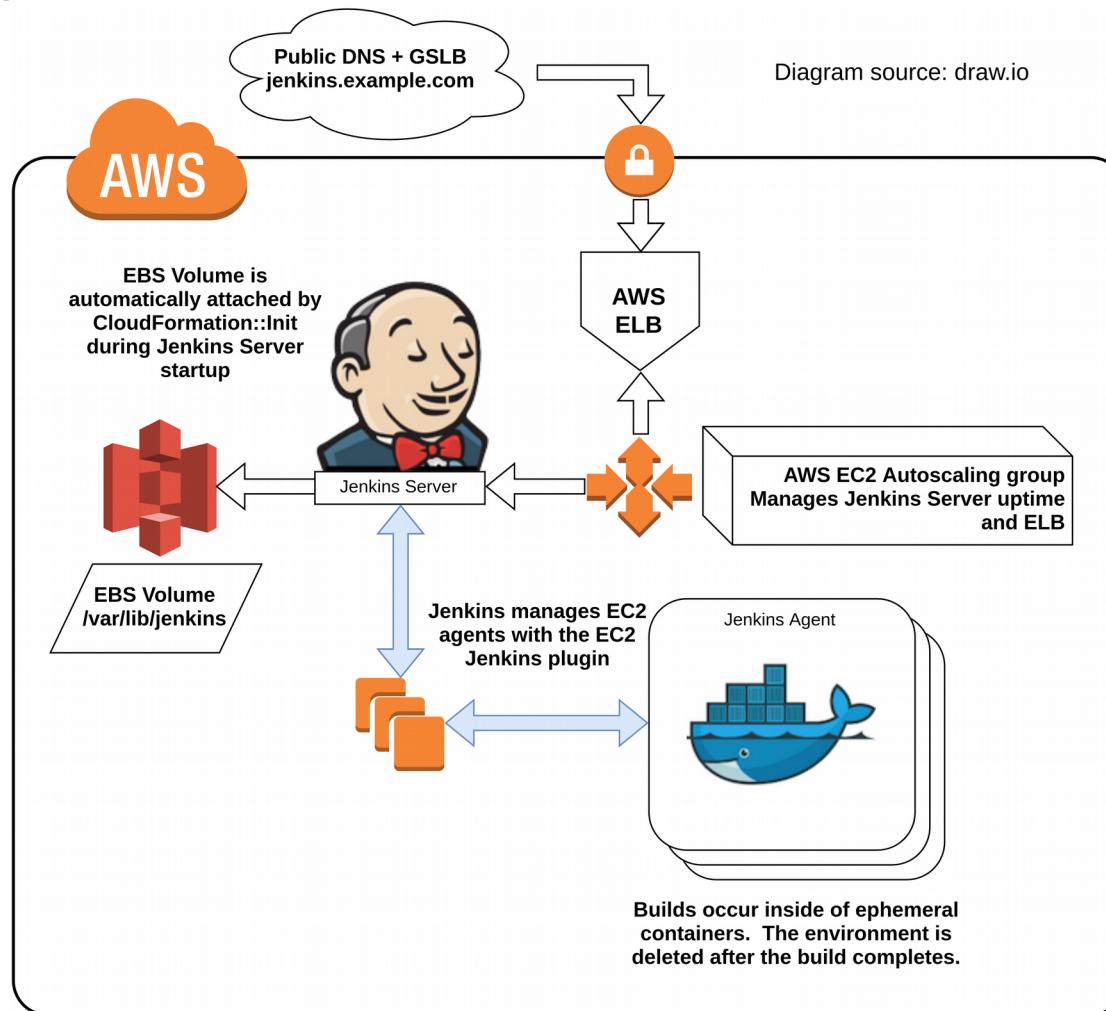
Managing immutable infrastructure

Jenkins as a Service

Code examples

- <https://github.com/samrocketman/jenkins-bootstrap-shared>
- <https://github.com/samrocketman/jenkins-bootstrap-jervis>
- <https://github.com/samrocketman/demo-jenkins-world-2018-jenkins-bootstrap>
- <https://github.com/samrocketman/demo-jenkins-world-2018-groovy-jar>

High availability Jenkins



<https://medium.com/@sebolabs/jenkins-ha-aws-cd55d82057c8>



What do I need to learn?

Let's be real

Where to start learning...

- Jenkins World 2017: Mastering the Jenkins Script Console
- Jenkins World 2017: Pipelines At Scale
- Groovy and Java programming languages.
- Job DSL Scripting (Groovy-based).
- Pipeline DSL Scripting (Groovy-based).
- Learn how to read Java stack traces.
- Familiarize with the Jenkins core code base.
- Explore Jenkins plugin code bases as needed.



Tech Demo

Show and tell a real workflow

Organizational impact

- Less development down time
- Reduced cost of operation
- More auditing and security
- Reduced time to market
- Improved customer experience



<https://openclipart.org/detail/186813/profit-high-five-graph-in-perspective>

DevOps World



Jenkins World

IAS Integral
Ad Science

