



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.

Screenshot of onboarding

JW 2018 Project Onboarding Instructions

Welcome to [Jenkins as a service](#). These first steps are necessary to onboard your project.

Developer's first steps

Creating a project in Jenkins is simple. Follow these three steps:

1. Add a `.jervis.yml` file to your GitHub repository following the [User Guide](#).
2. Execute the `Generate Jenkins Jobs from YAML` onboarding job by clicking  icon.
3. If creating your job succeeds, then visit the [GitHub Organizations View](#) in the tabs above.

Documentation

Visit the [Jervis wiki](#) for comprehensive documentation on how to configure and manage Jenkins as a service.

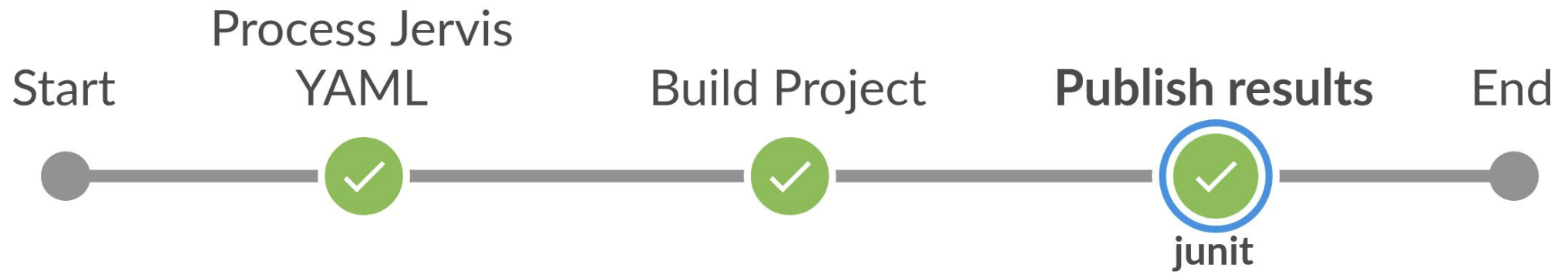
language: java


```
language: java
```

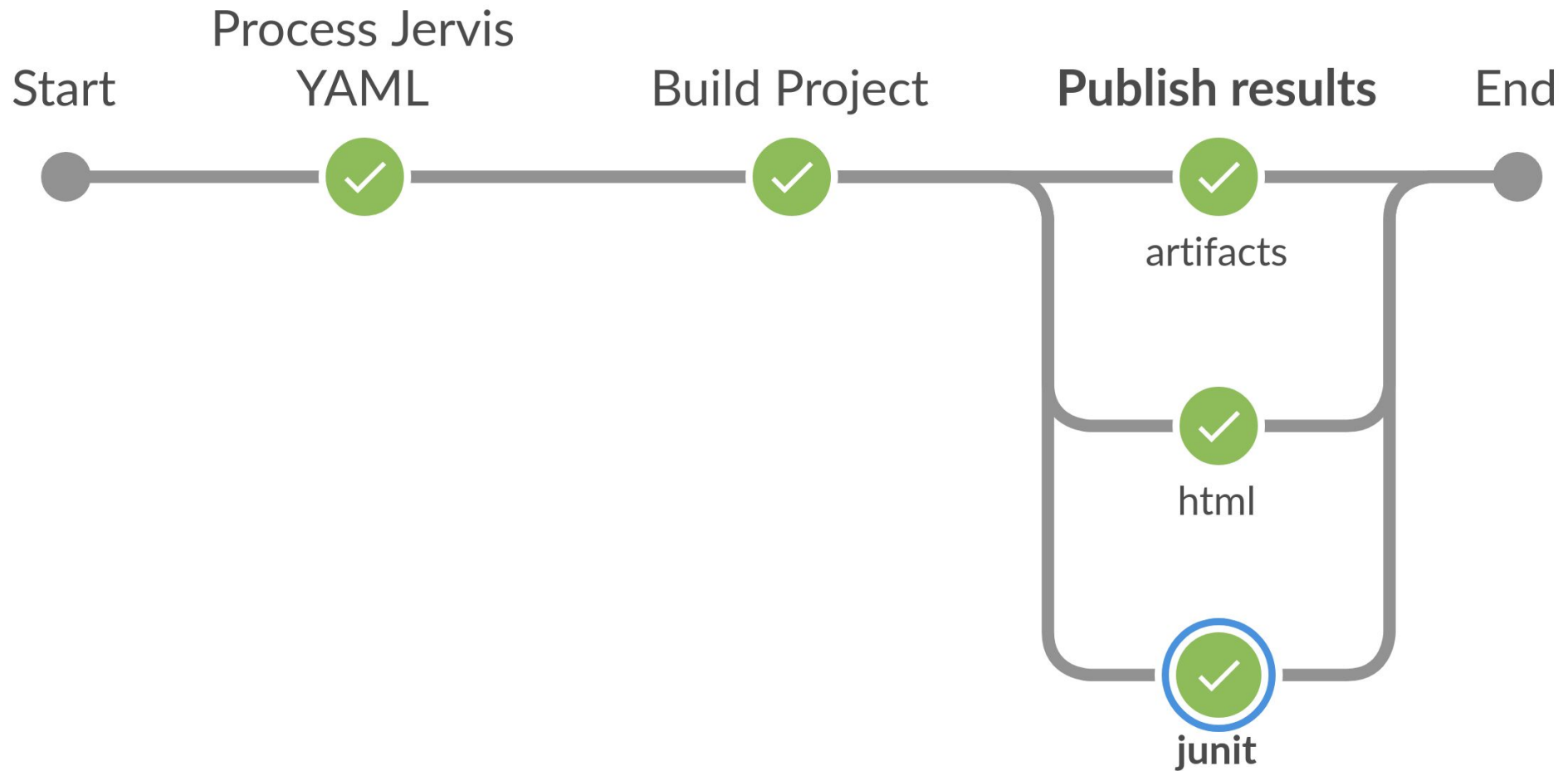
```
jenkins:
```

```
  collect:
```

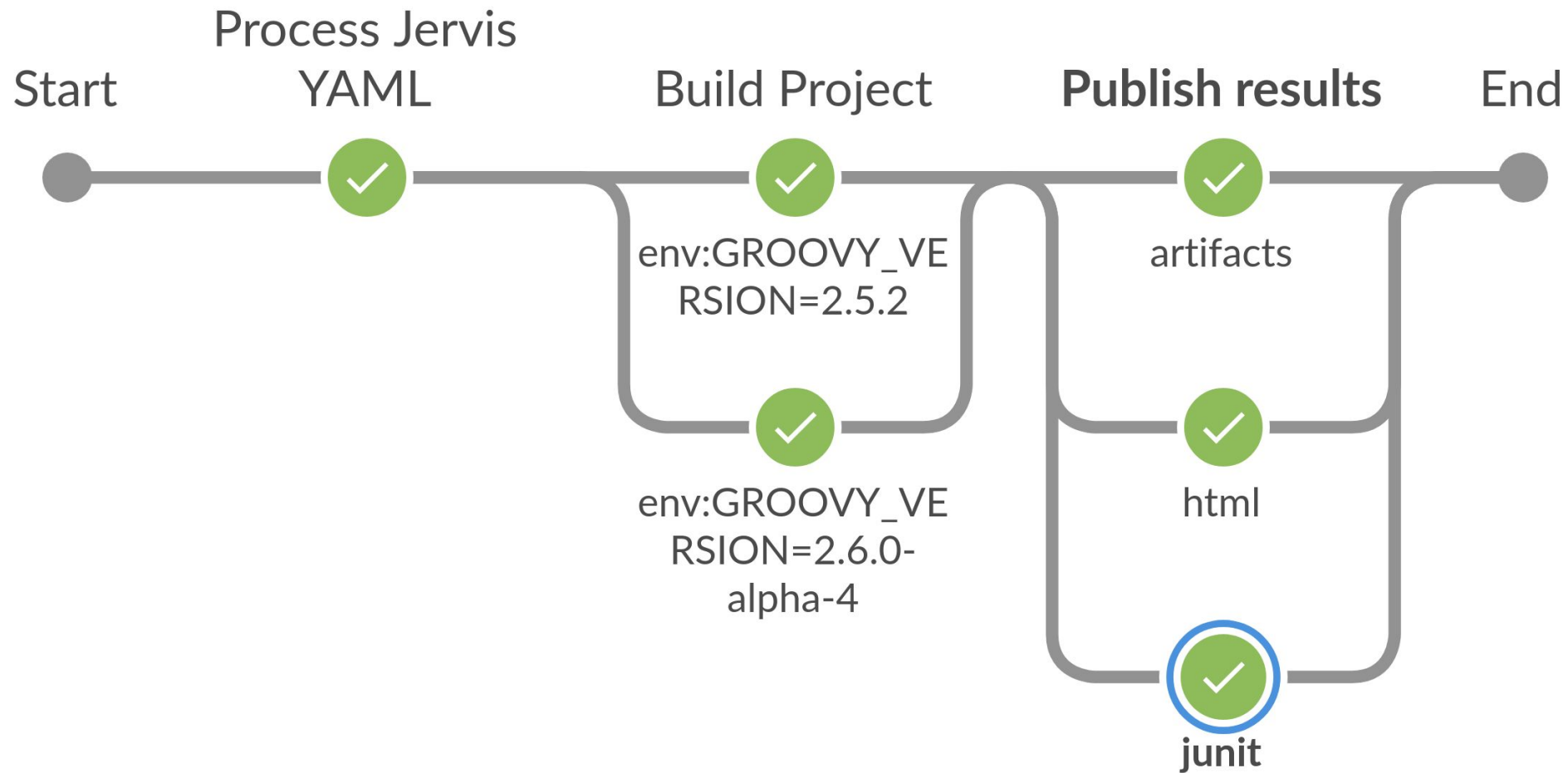
```
    junit: build/test-results/**/*.*xml
```

```
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

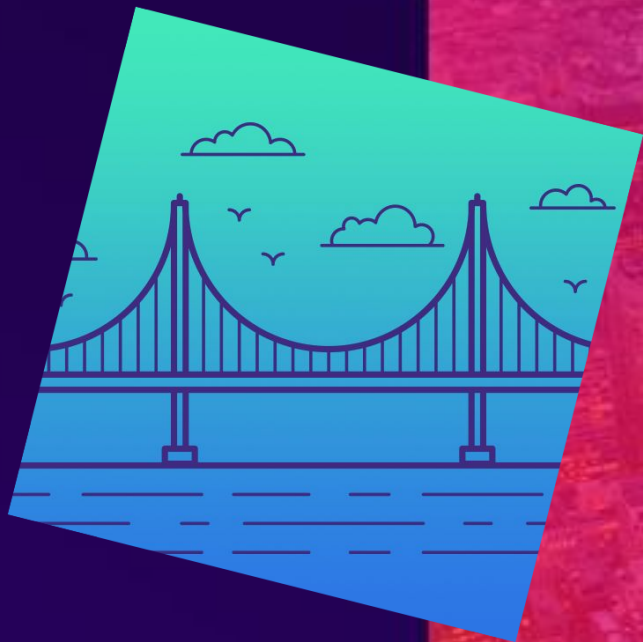
```
{
  "defaults": {
    "platform": "docker-in-docker",
    "os": "alpinejdk8",
    "stability": "stable",
    "sudo": "sudo"
  },
  "supported_platforms": {
    "docker-in-docker": {
      "alpinejdk8": {
        "language": [
          "java",
          "shell"
        ],
        "toolchain": [
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


Toolchains

```
{
  "toolchains": {
    "java": ["env", "jdk"],
    "shell": ["env"]
  },
  "env": {
    "matrix": "advanced",
    "*": [
      "export ${jervis_toolchain_ivalue}"
    ]
  },
  "jdk": {
    "default_ivalue": "openjdk8",
    "friendlyLabel": true,
    "openjdk8": [
      "export JAVA_HOME=\"/usr/lib/jvm/java-1.8-openjdk\"",
      "export PATH=\"${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) X

Branch Include

master

?

Branch Exclude

?

Tag Include

*

?

Tag Exclude

?

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

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 1.0 Is Generally Available

- 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 (and good practices)

- In a large organization, engineers don't want to “learn Jenkins”.
- Features of onboarding should be documented to reflect what engineers really know and want out of a CI and CD system
 - They know where their code is hosted.
 - They know how their project builds.
 - They know where their project will end up.

Assumptions (and good practices) Continued

- Every project uses pinned versions of dependencies.
 - Java-esque (don't rely on -SNAPSHOT)
 - NodeJS (npm shrinkwrap; npm or yarn lock files; if using gulp then same applies)
 - Python (pip freeze)
 - Ruby (Gemfile.lock)
- Cache dependencies from the Internet.
 - Insulation
 - Reliability
 - Archival

Assumptions (and good practices) Continued

- Always think about idempotency.
 - Configuration as code
 - Infrastructure as code
 - Release pipeline as code
- 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.

Evangelizing a project development workflow

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

Multi-Member Project Workflow (GitHub Flow With Releases)

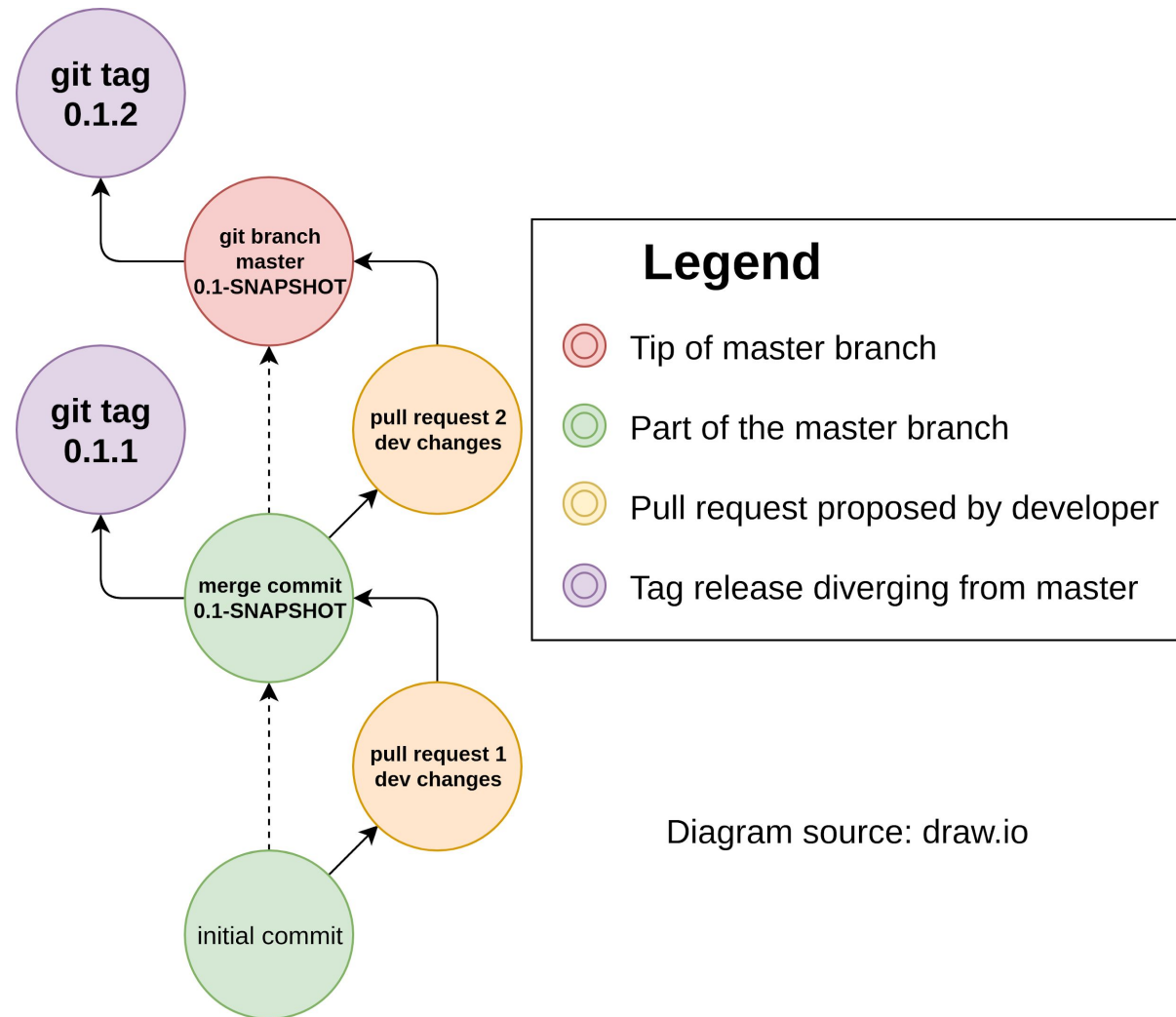
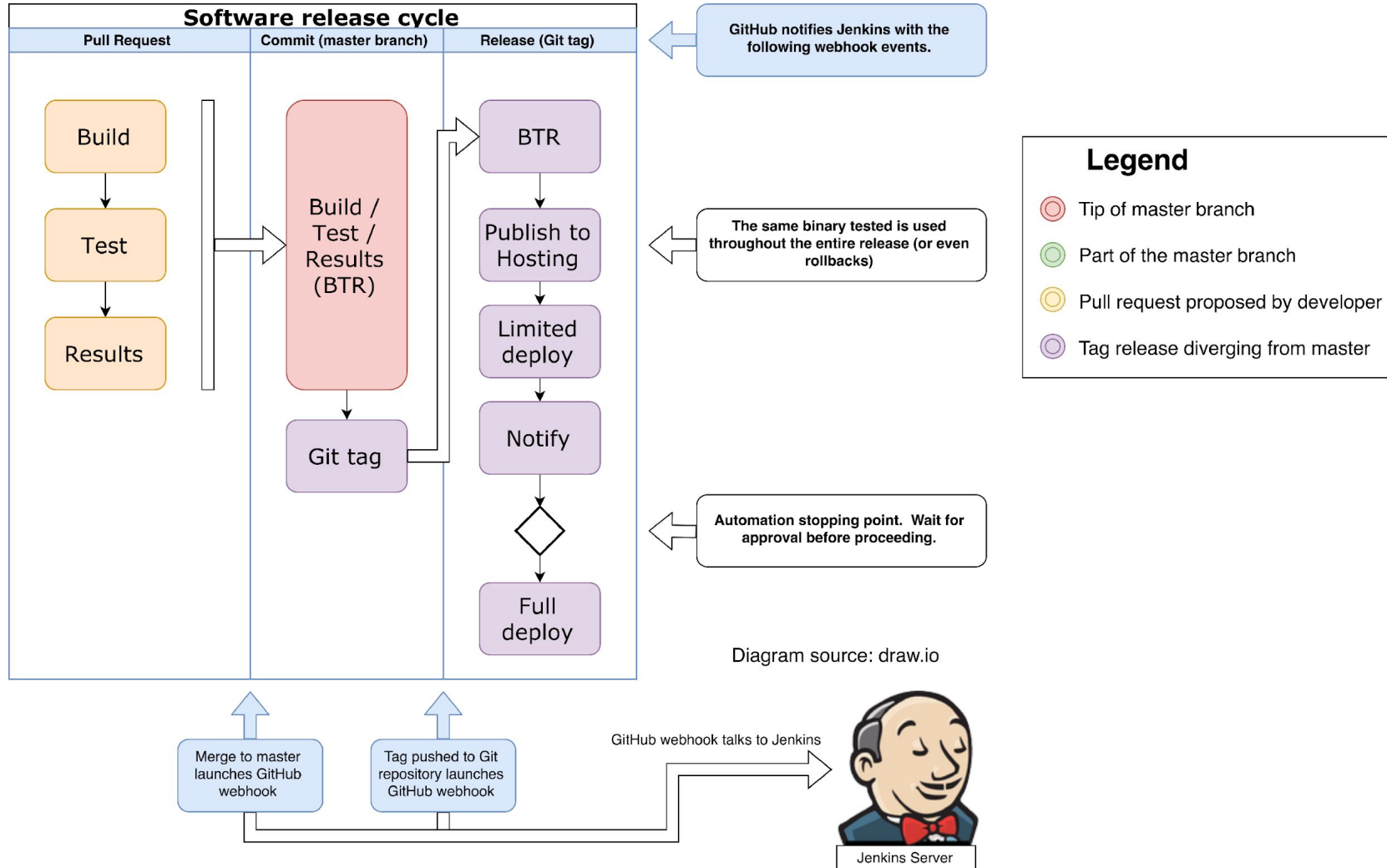
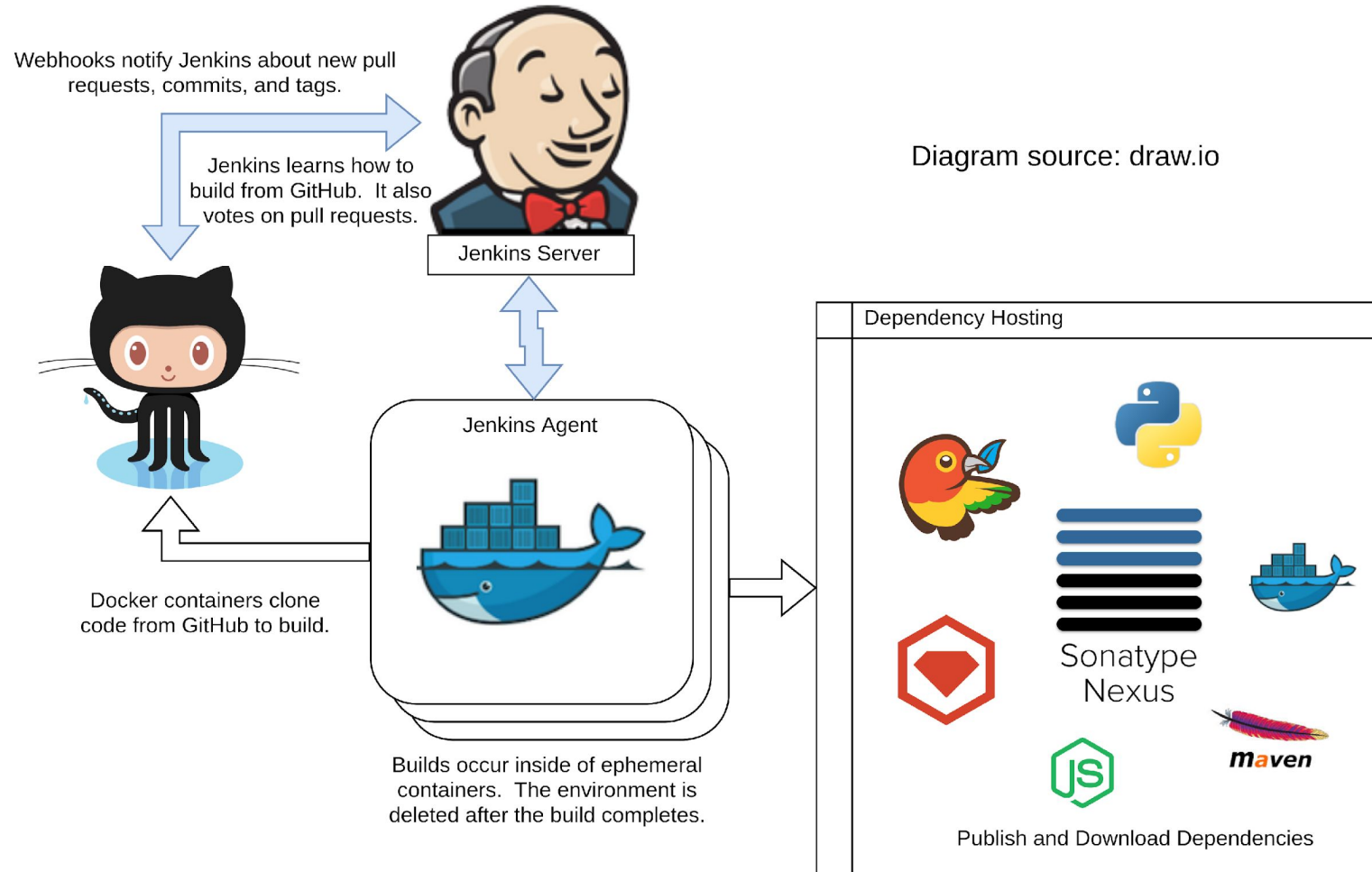


Diagram source: draw.io

Jenkins Multibranch Pipelines Running GitHub Flow



Overview of interactions between Jenkins and GitHub





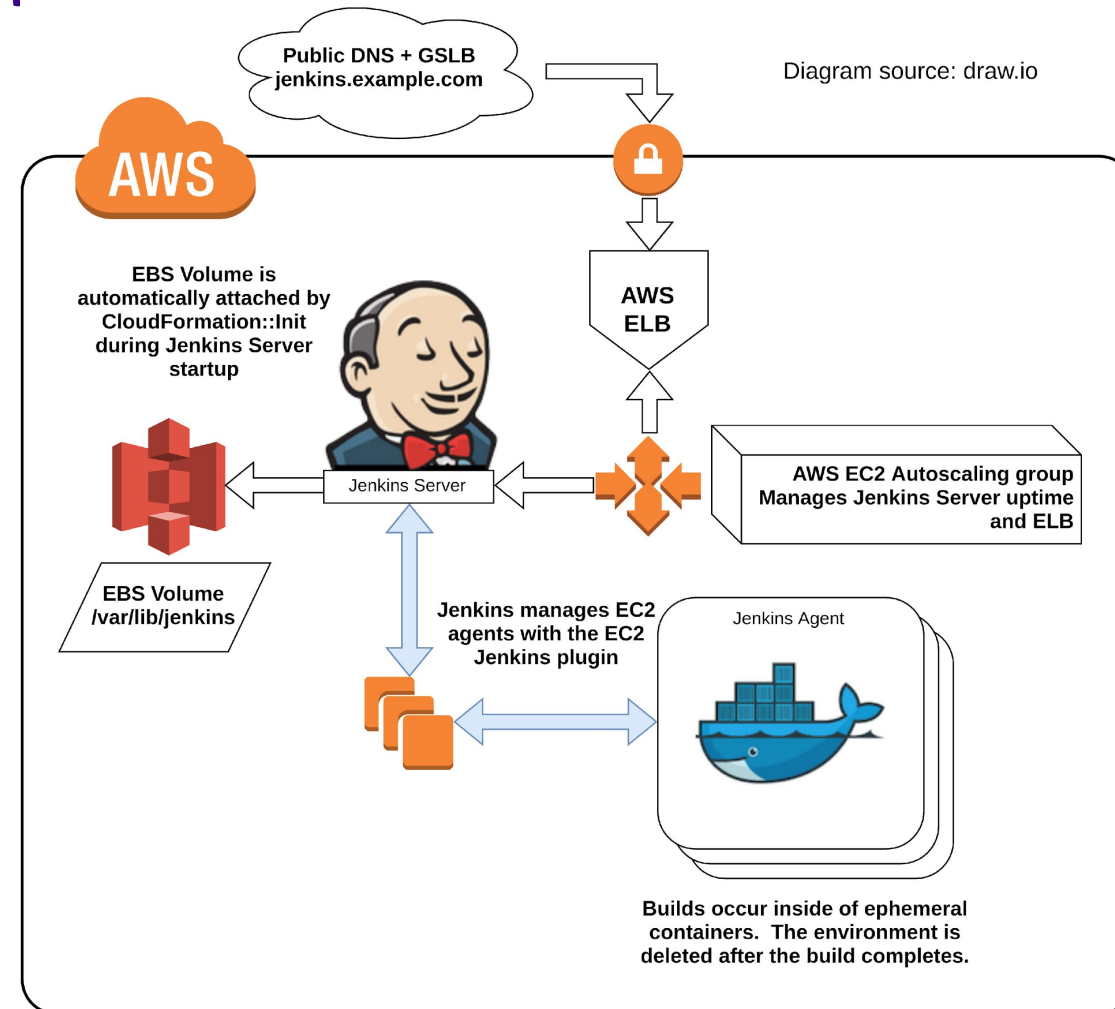
Managing immutable infrastructure

Jenkins as a Service

Code examples

- <https://github.com/samrocketman/jervis>
- <https://github.com/samrocketman/jenkins-bootstrap-shared>
- <https://github.com/samrocketman/jenkins-bootstrap-jervis>
- <https://github.com/samrocketman/docker-jenkins-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

