Under the hood of serverless Jenkins
Jenkinsfile Runner

Oleg Nenashev (@oleg_nenashev)
CloudBees

St. Petersburg, Oct 30, 2019

TL;DR

“You can run Jenkins Pipelines as single-shot container in any environment”

This talk: How to do it?

> whoami

@oleg_nenashev

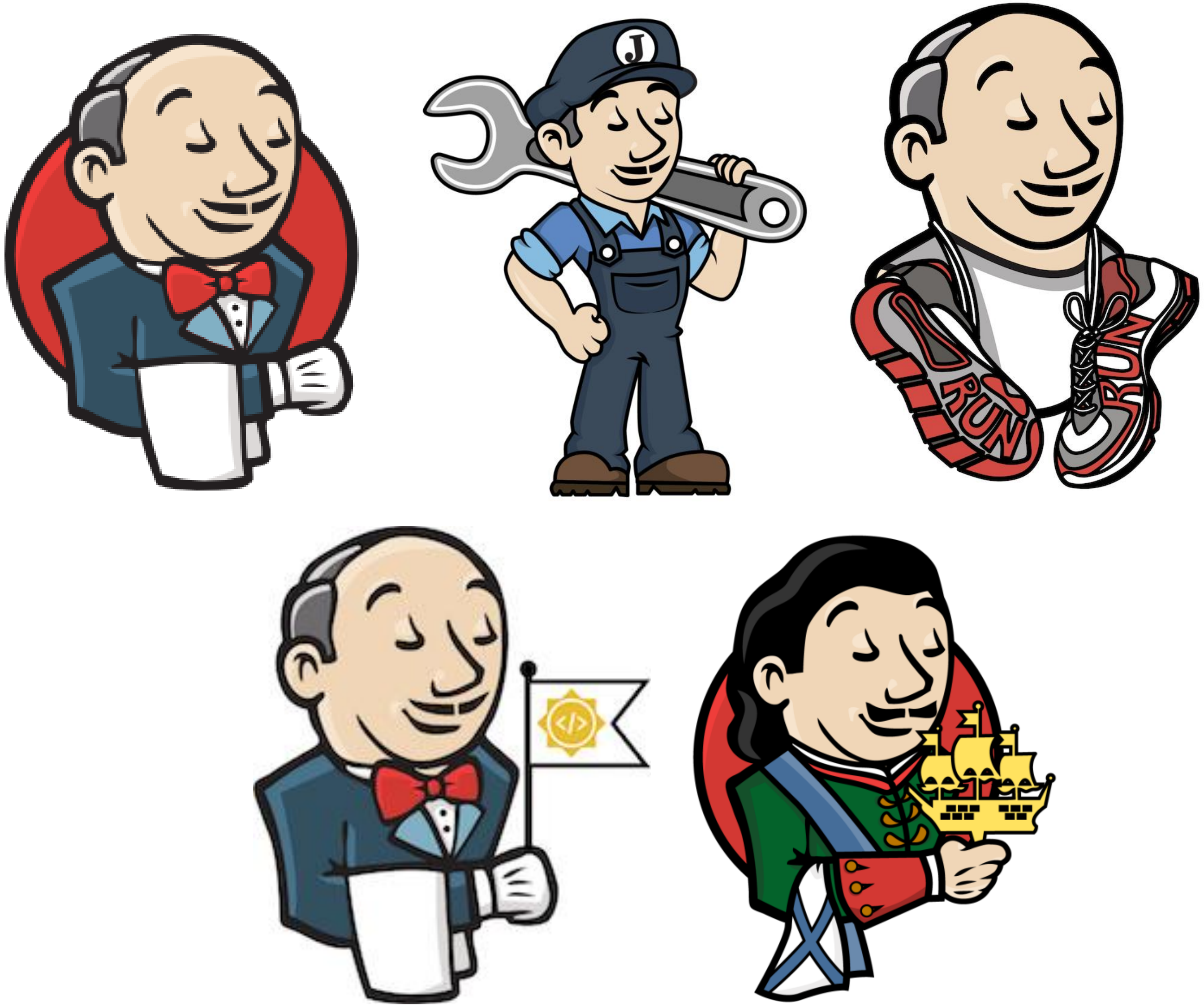
oleg-nenashev

- Based in Neuchatel, Switzerland
- Principal SW Engineer, CloudBees
- Jenkins core maintainer

> whoami -jenkins

@oleg_nenashev

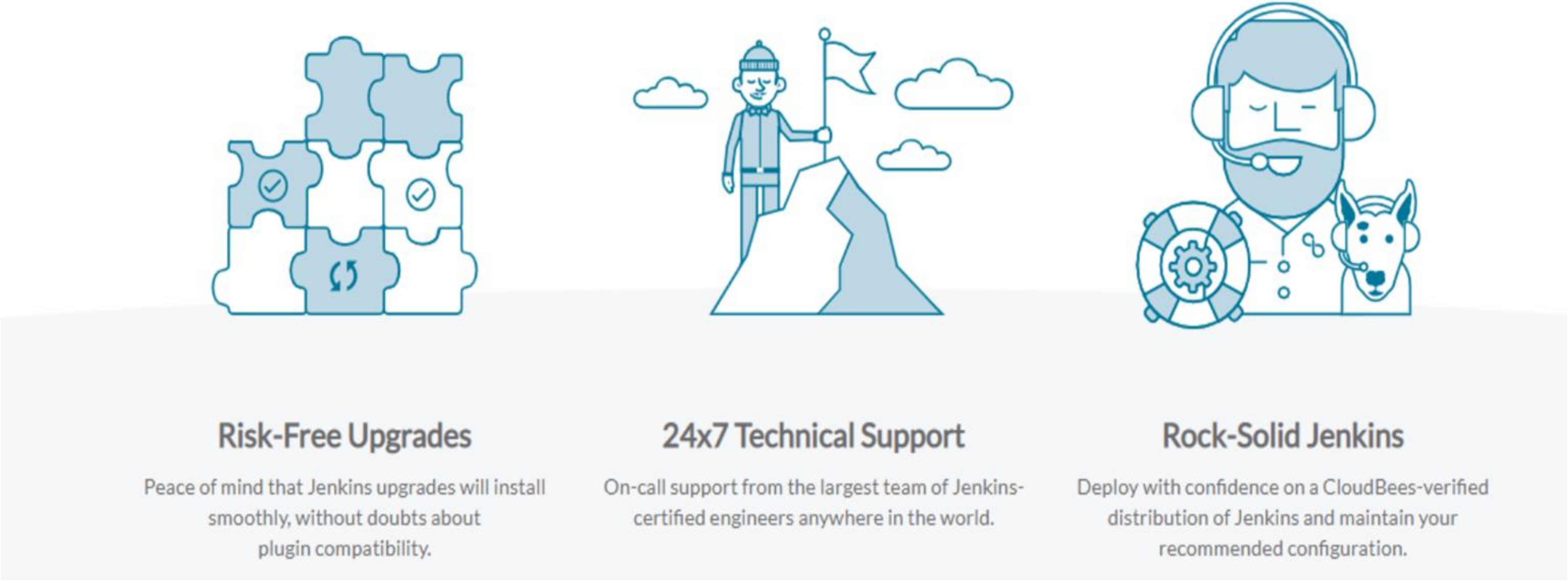
oleg-nenashev



> whoami -cloudbees

https://www.cloudbees.com

- CloudBees Jenkins Distribution
- CloudBees Jenkins Support
- Community, e.g. Java 11 support, Jenkinsfile Runner, JCasC



Risk-Free Upgrades

Peace of mind that Jenkins upgrades will install smoothly, without doubts about plugin compatibility.

24x7 Technical Support

On-call support from the largest team of Jenkins-certified engineers anywhere in the world.

Rock-Solid Jenkins

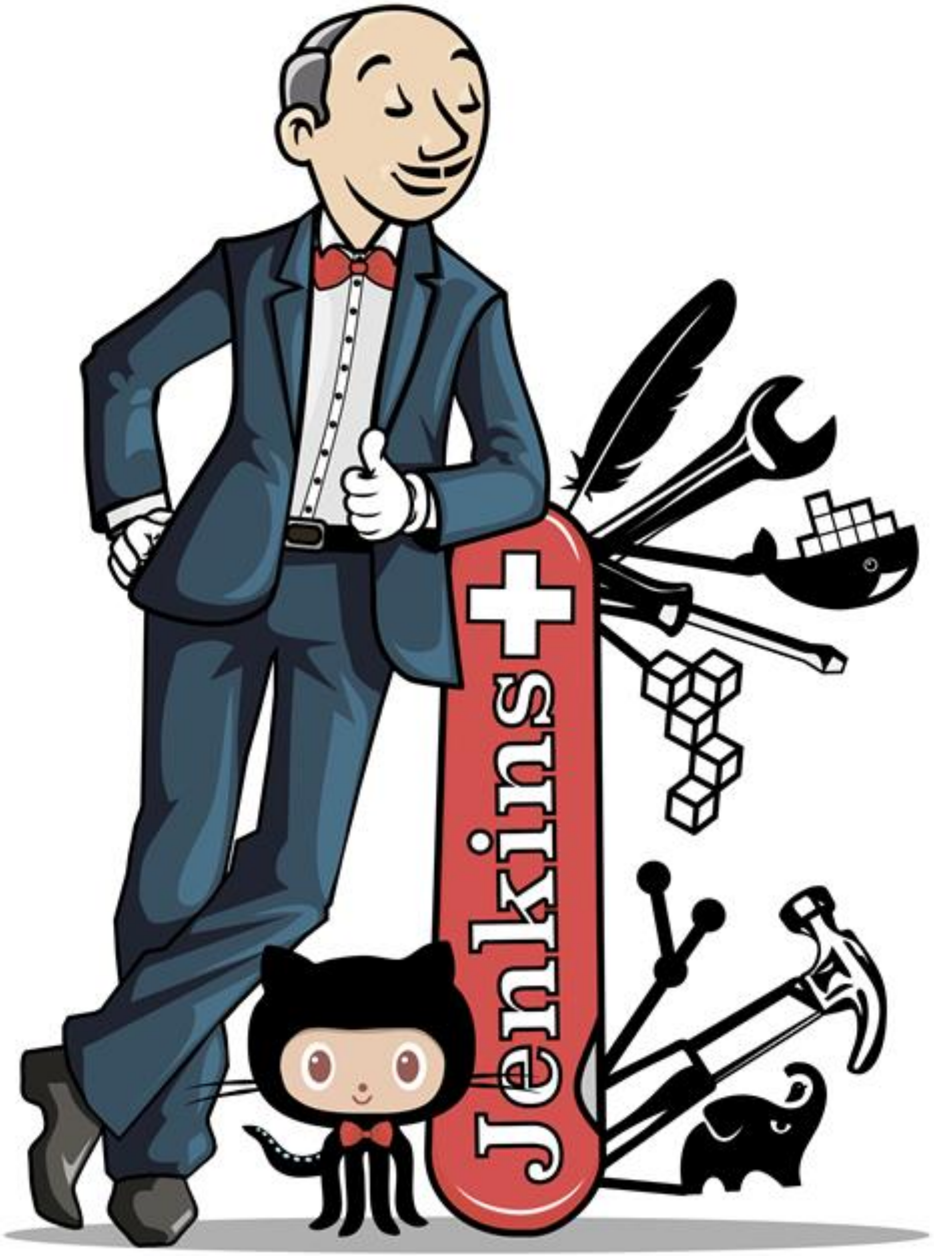
Deploy with confidence on a CloudBees-verified distribution of Jenkins and maintain your recommended configuration.

Disclaimer: Opinions are mine; typos are mine

About you

Who is Mr. Jenkins?

- Automation server/framework
- Open-source
- More than 1700 plugins
- Big community
- Commercial support is available



Jenkins as you may knew it… in 2012

Batteries are not included

Complex configuration

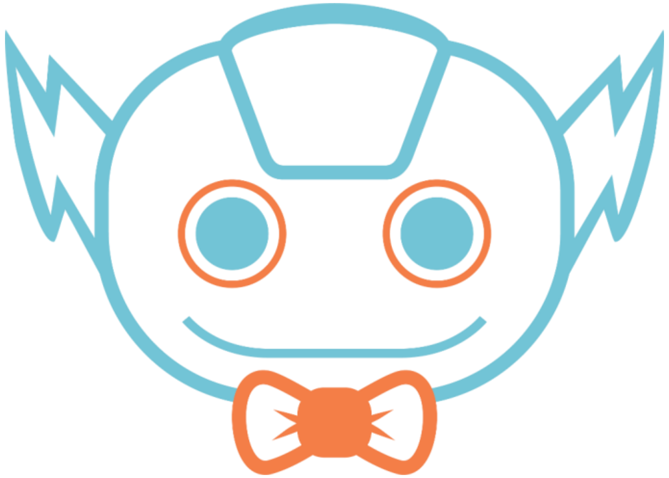
Mega-masters, limited scalability

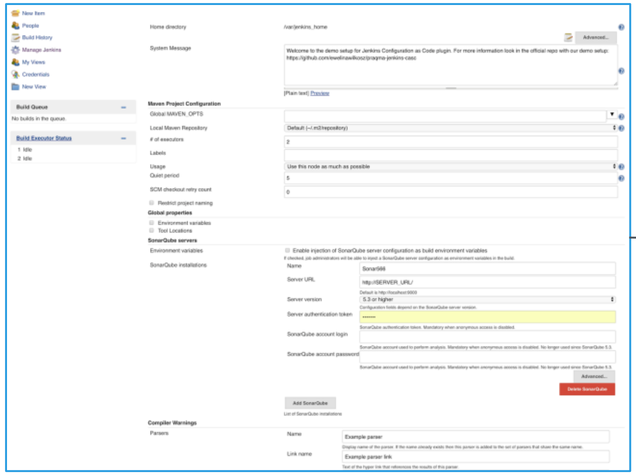
Painful upgrades

Maintenance is required

Jenkins in 2019

- Jenkins Pipeline
- Config-as-Code Plugin
- Modern plugins
- Modern packaging
- Jenkins X





No builds in the queue.

1 Idle
2 Idle

Home directory /var/jenkins_home

System Message
Welcome to the demo setup for Jenkins Configuration as Code plugin. For more information look in the official repo with our demo setup: https://github.com/ewelinawilkosz/praqma-jenkins-casc

[Plain text] Preview

Maven Project Configuration

Global MAVEN_OPTS

Local Maven Repository: Default (~/.m2/repository)

# of executors: 2

Labels

Usage: Use this node as much as possible

Quiet period: 5

SCM checkout retry count: 0

□ Restrict project naming

Global properties

□ Environment variables

□ Tool Locations

SonarQube servers

Environment variables: □ Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name: Sonar566

Server URL: http://SERVER_URL/

Default is http://localhost:9000

Server version: 5.3 or higher

Configuration fields depend on the SonarQube server version.

Server authentication token: ••••••••

SonarQube authentication token. Mandatory when anonymous access is disabled.

SonarQube account login

SonarQube account used to perform analysis. Mandatory when anonymous access is disabled. No longer used since SonarQube 5.3.

SonarQube account password

SonarQube account used to perform analysis. Mandatory when anonymous access is disabled. No longer used since SonarQube 5.3.

List of SonarQube installations

Compiler Warnings

Parsers

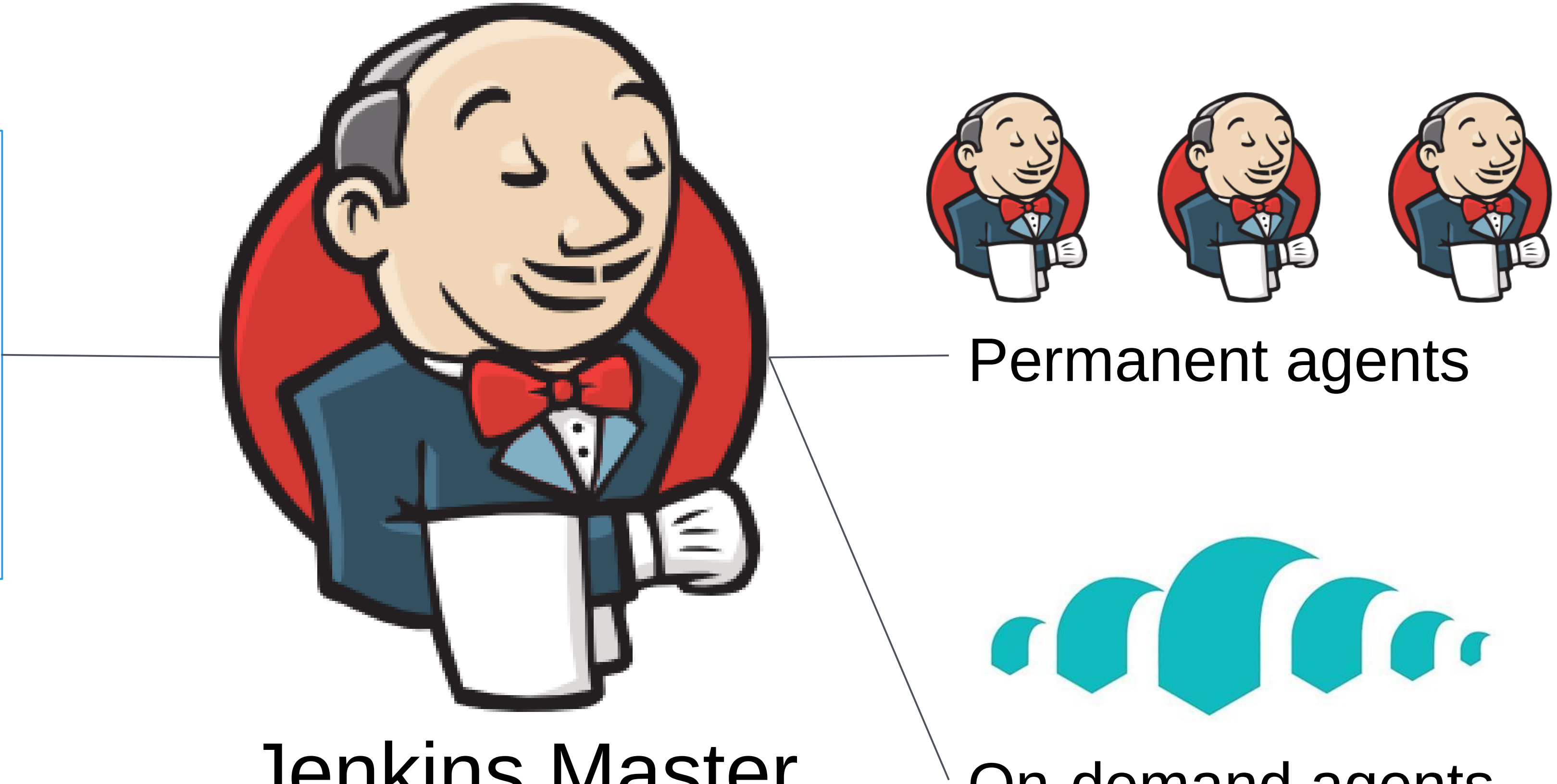
Name: Example parser

Display name of the parser. If the name already exists then this parser is added to the set of parsers that share the same name.

Link name: Example parser link

Text of the hyper link that references the results of this parser.

Web UI

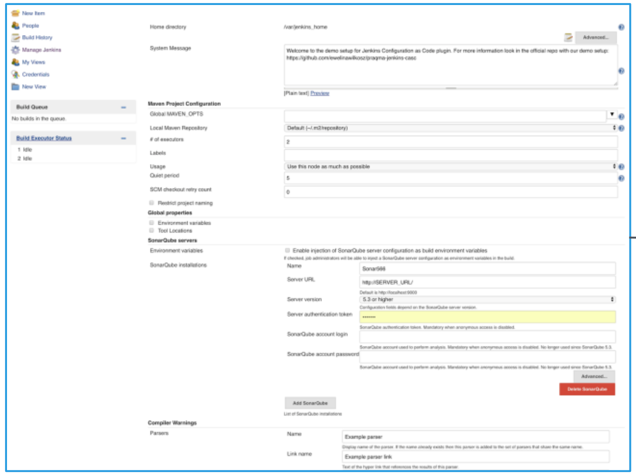


Permanent agents

Jenkins Master

On-demand agents
from cloud providers

Master as a SPoF Macroservice



No builds in the queue.

1 Idle
2 Idle

Home directory: /var/jenkins_home

System Message: Welcome to the demo setup for Jenkins Configuration as Code plugin. For more information look in the official repo with our demo setup: https://github.com/ewelinawilkosz/praqma-jenkins-casc

[Plain text]

Maven Project Configuration

Global MAVEN_OPTS:

Local Maven Repository: Default (~/.m2/repository)

# of executors: 2

Labels:

Usage: Use this node as much as possible

Quiet period: 5

SCM checkout retry count: 0

□ Restrict project naming

Global properties

□ Environment variables

□ Tool Locations

SonarQube servers

Environment variables: □ Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name: Sonar566

Server URL: http://SERVER_URL/

Default is http://localhost:9000

Server version: 5.3 or higher

Configuration fields depend on the SonarQube server version.

Server authentication token: ••••••••

SonarQube authentication token. Mandatory when anonymous access is disabled.

SonarQube account login:

SonarQube account used to perform analysis. Mandatory when anonymous access is disabled. No longer used since SonarQube 5.3.

SonarQube account password:

SonarQube account used to perform analysis. Mandatory when anonymous access is disabled. No longer used since SonarQube 5.3.

List of SonarQube installations

Compiler Warnings

Parsers

Name: Example parser

Display name of the parser. If the name already exists then this parser is added to the set of parsers that share the same name.

Link name: Example parser link

Text of the hyper link that references the results of this parser.

Web UI

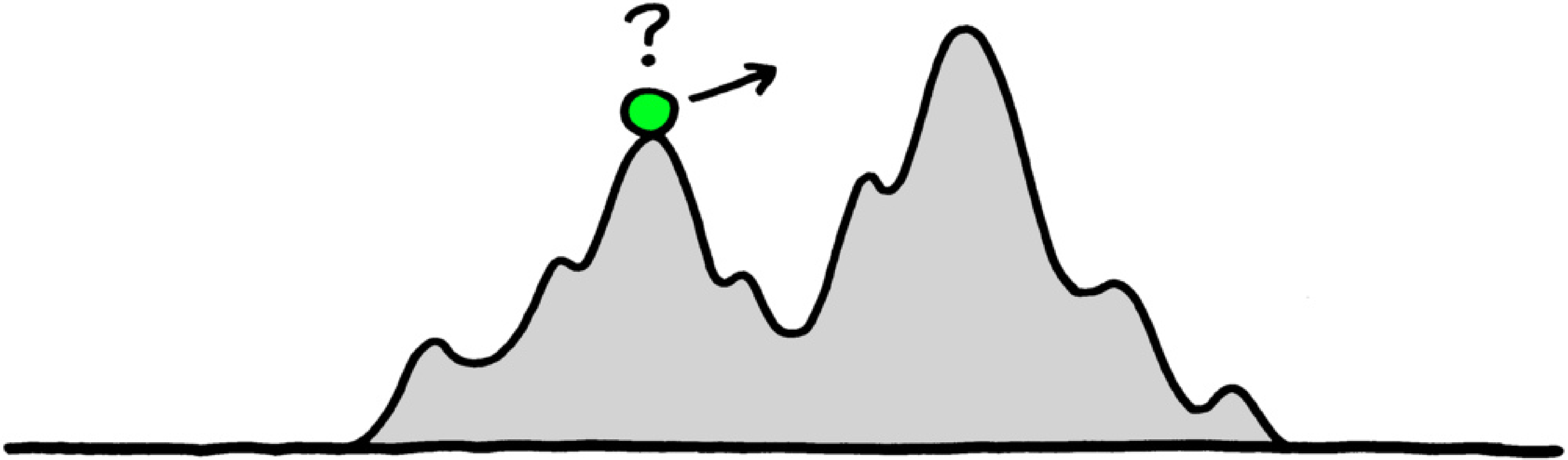
Jenkins
Master

Permanent agents

On-demand agents
from cloud providers

Aug 31, 2018 – Kohsuke Kawaguchi,
“Jenkins: Shifting Gears”
https://jenkins.io/blog/2018/08/31/shifting-gears/

Shifting Gears: Making Changes



Cloud Native Jenkins

“General purpose CI/CD engine that runs on Kubernetes and embraces fundamentally different architecture & extensibility mechanisms”

Kohsuke Kawaguchi, 2018

https://flic.kr/p/75ncBF

Cloud Native Jenkins

- Best service for each need
- Pay per use
- “Infinite” scaling
- Easy to use
- Easy to maintain
- Fast to develop

https://flic.kr/p/75ncBF

Cloud Native Special Interest Group

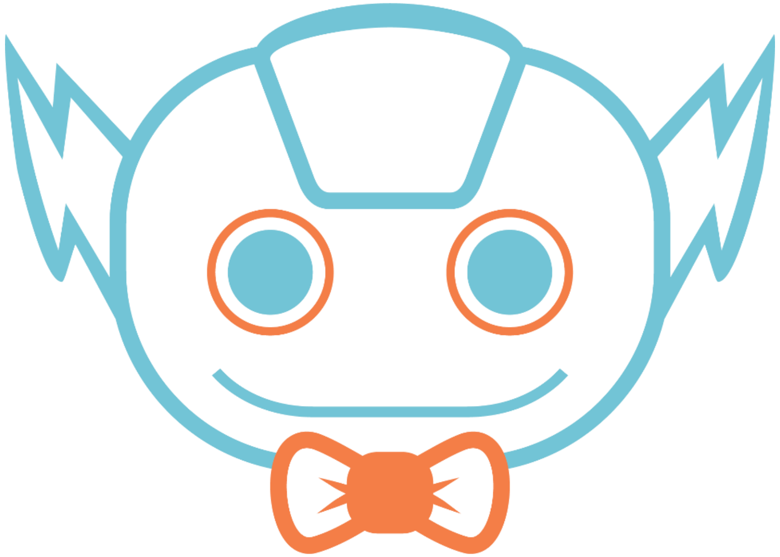
Founded in July 2018

https://jenkins.io/sigs/cloud-native

Goal: Improve Jenkins to run on cloud environments as a "Cloud Native" application

© 2019 CloudBees, Inc. All Rights Reserved.

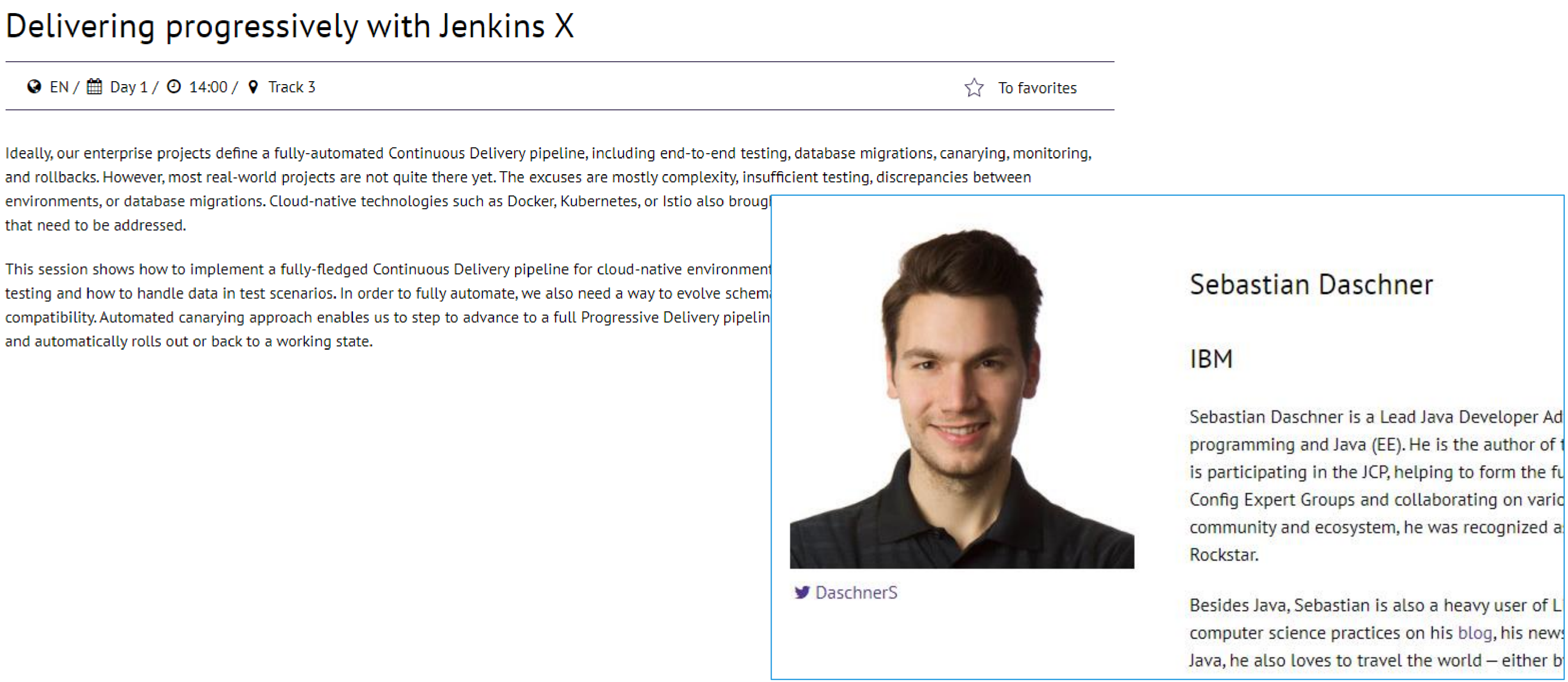
Jenkins X?



JENKINS X

Accelerate Your Continuous Delivery on Kubernetes

https://jenkins-x.io



Delivering progressively with Jenkins X

EN / Day 1 / 14:00 / Track 3

☆ To favorites

Ideally, our enterprise projects define a fully-automated Continuous Delivery pipeline, including end-to-end testing, database migrations, canarying, monitoring, and rollbacks. However, most real-world projects are not quite there yet. The excuses are mostly complexity, insufficient testing, discrepancies between environments, or database migrations. Cloud-native technologies such as Docker, Kubernetes, or Istio also broug
that need to be addressed.

This session shows how to implement a fully-fledged Continuous Delivery pipeline for cloud-native environment
testing and how to handle data in test scenarios. In order to fully automate, we also need a way to evolve schem
compatibility. Automated canarying approach enables us to step to advance to a full Progressive Delivery pipelin
and automatically rolls out or back to a working state.

DaschnerS

Sebastian Daschner

IBM

Sebastian Daschner is a Lead Java Developer Ad
programming and Java (EE). He is the author of t
is participating in the JCP, helping to form the fu
Config Expert Groups and collaborating on vario
community and ecosystem, he was recognized a
Rockstar.

Besides Java, Sebastian is also a heavy user of L
computer science practices on his blog, his news
Java, he also loves to travel the world – either b

Jenkins X 2.x is...

- AWESOME (c)
- Cloud native
- K8s native
- Scalable
- Designed for CD
- Moving fast

Jenkins X 2.x is...

- AWESOME (c)
- Cloud native
- K8s native
- Scalable
- Designed for CD
- Moving fast

Limitations

- Moving too fast?
- Kubernetes-only
- GitOps-only
- Limited SCMs support
- Limited CI capabilities

Jenkins X 2.x is…

- AWESOME (c)
- Cloud native
- K8s native
- Scalable
- Des
- Mo

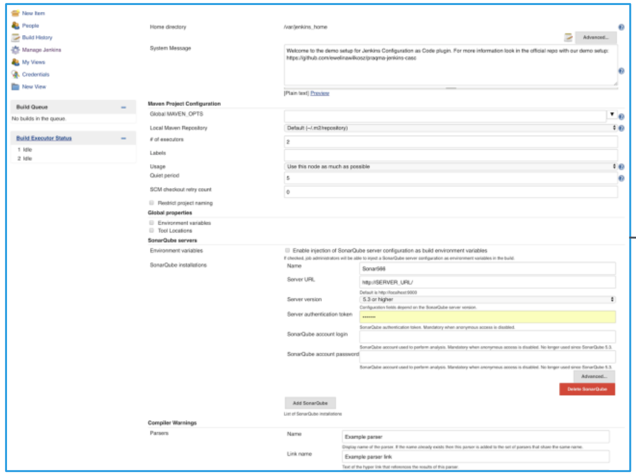
Limitations

- Moving too fast?
- Kubernetes-only
- GitOps-only
- Limited SCMs support

Try out Jenkins X if it fits your environment OR if you plan to go all-in Kubernetes. Jenkins is still around for other cases

Going Back?

Master as a SPoF Macroservice



No builds in the queue.

1 Idle
2 Idle

Home directory /var/jenkins_home

System Message: Welcome to the demo setup for Jenkins Configuration as Code plugin. For more information look in the official repo with our demo setup: https://github.com/ewelinawilkosz/praqma-jenkins-casc

[Plain text] Preview

Maven Project Configuration

Global MAVEN_OPTS

Local Maven Repository: Default (~/.m2/repository)

# of executors: 2

Labels

Usage: Use this node as much as possible

Quiet period: 5

SCM checkout retry count: 0

□ Restrict project naming

Global properties

□ Environment variables

□ Tool Locations

SonarQube servers

Environment variables: □ Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name: Sonar566

Server URL: http://SERVER_URL/

Default is http://localhost:9000

Server version: 5.3 or higher

Configuration fields depend on the SonarQube server version.

Server authentication token: •••••••

SonarQube authentication token. Mandatory when anonymous access is disabled.

SonarQube account login

SonarQube account used to perform analysis. Mandatory when anonymous access is disabled. No longer used since SonarQube 5.3.

SonarQube account password

SonarQube account used to perform analysis. Mandatory when anonymous access is disabled. No longer used since SonarQube 5.3.

List of SonarQube installations

Compiler Warnings

Parsers

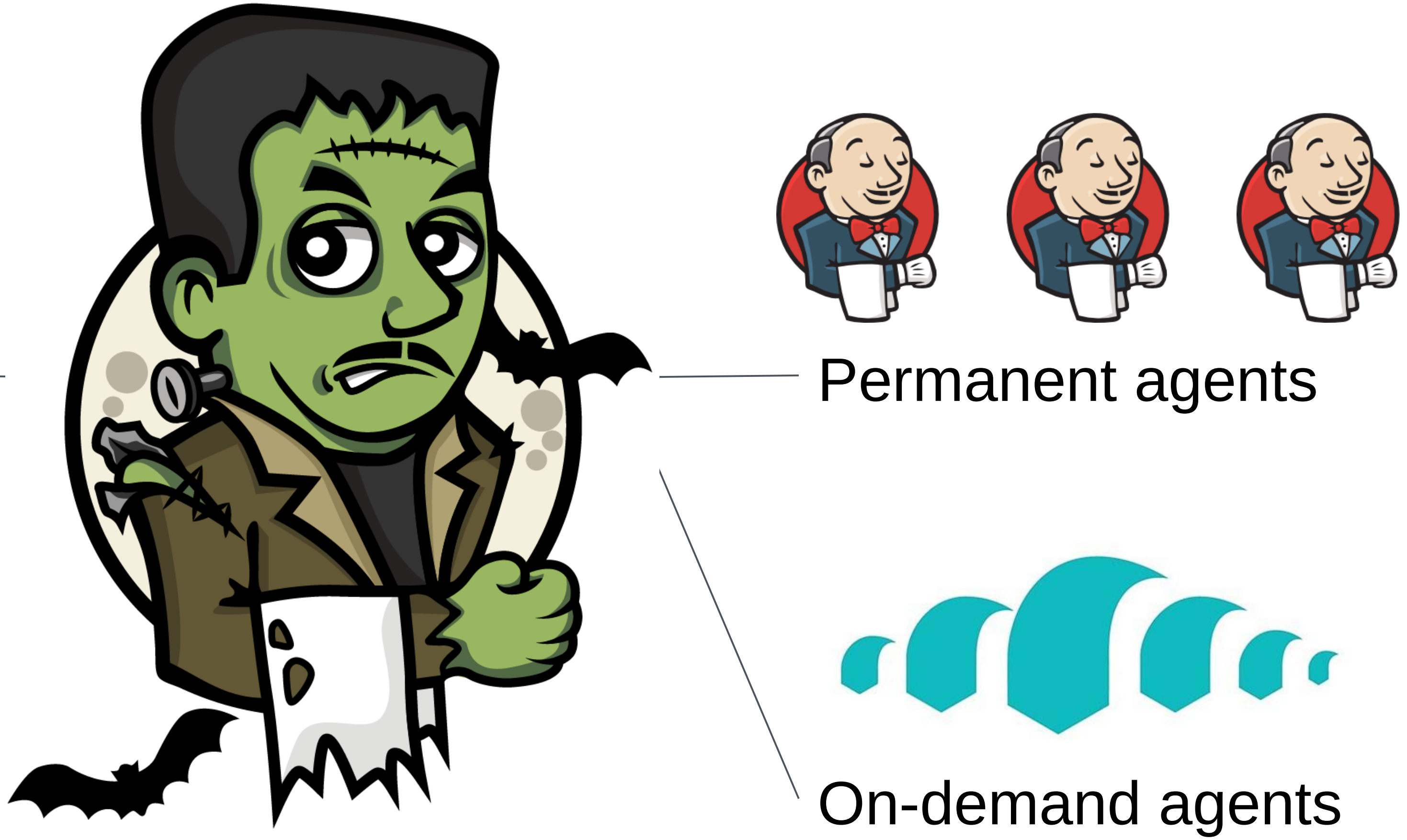
Name: Example parser

Display name of the parser. If the name already exists then this parser is added to the set of parsers that share the same name.

Link name: Example parser link

Text of the hyper link that references the results of this parser.

Web UI



Permanent agents

On-demand agents from cloud providers

Jenkins Master

(┛ಠ_ಠ)┛彡┻━┻

What if Jenkins master was "on-demand"?

RUN
RUN

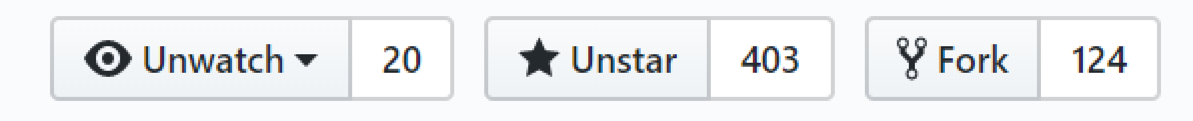
Jenkinsfile Runner

https://github.com/jenkinsci/jenkinsfile-runner

Started by Kohsuke in Mar 2018

30 contributors

Status: Beta (1.0-beta-11)



JFR TL;DR

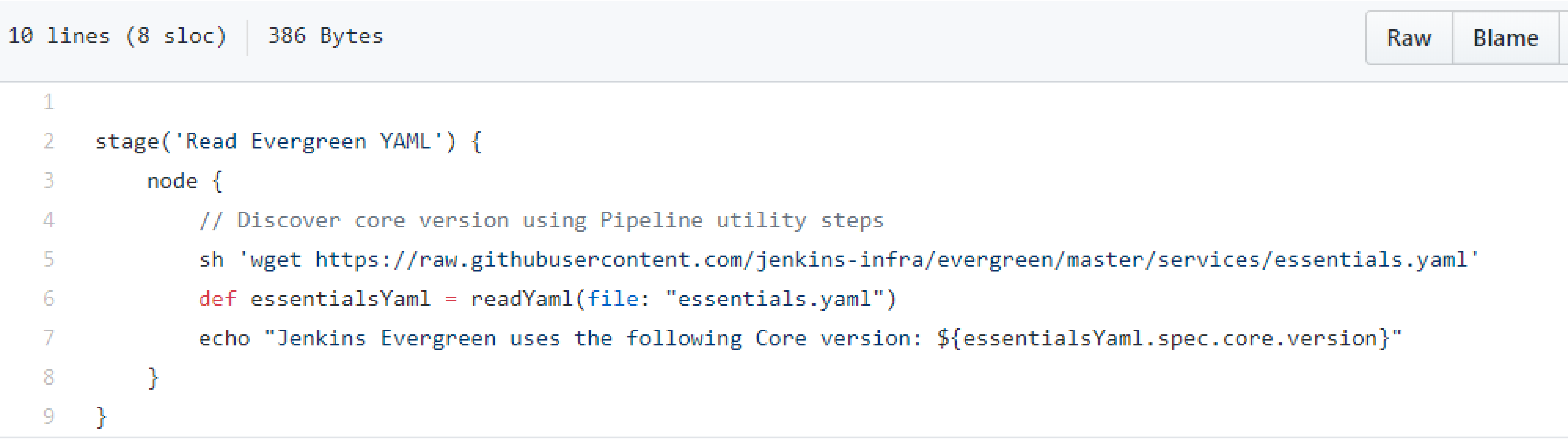
- Runs Pipeline and shuts down
- Thin JAR and fat JAR
- Docker image
- Runs everywhere

https://github.com/jenkinsci/jenkinsfile-runner

Usage

```
docker run --rm \
    -v $(pwd)/Jenkinsfile:/workspace/Jenkinsfile \
     jenkins4eval/jenkinsfile-runner
```

Demo. Our Jenkinsfile



10 lines (8 sloc) | 386 Bytes

```

stage('Read Evergreen YAML') {
    node {
        // Discover core version using Pipeline utility steps
        sh 'wget https://raw.githubusercontent.com/jenkins-infra/evergreen/master/services/essentials.yaml'
        def essentialsYaml = readYaml(file: "essentials.yaml")
        echo "Jenkins Evergreen uses the following Core version: ${essentialsYaml.spec.core.version}"
    }
}
```

Terminalizer

Jenkinsfile Runner is not just CLI

Features

- Startup in few seconds
- No-sandbox mode support
- Java 11 support

Limitations

- Jenkinsfile Runner is not fully compatible
- No Web UI, no way to extract Jenkins reports
- No native support of external storage
- No build throttling, cross-master communication, etc.
- No out-of-the-box sidecar containers support

Jenkinsfile Runner. Usage examples

Jenkins X 1.x, serverless mode

- jenkins-x/jenkins-x-serverless

ci.jenkins.io-runner

- jenkinsci/ci.jenkins.io-runner

“FaaS” packaging

- GitHub Actions: jonico/jenkinsfile-runner-github-actions
- Codeship: oleg-nenashev/codeship-jenkinsfile
- AWS Lambda: carlossg/jenkinsfile-runner-lambda
- Project Fn: carlossg/jenkinsfile-runner-fn

“Packaging”?

Jenkinsfile Runner flavors

Vanilla: Docker/Binary in the official repo

- Basic plugins: Pipeline, JCasC, etc.
- GitHub: [jenkinsci/jenkinsfile-runner]
- DockerHub: [jenkins4eval/jenkinsfile-runner]

Jenkinsfile Runner flavors

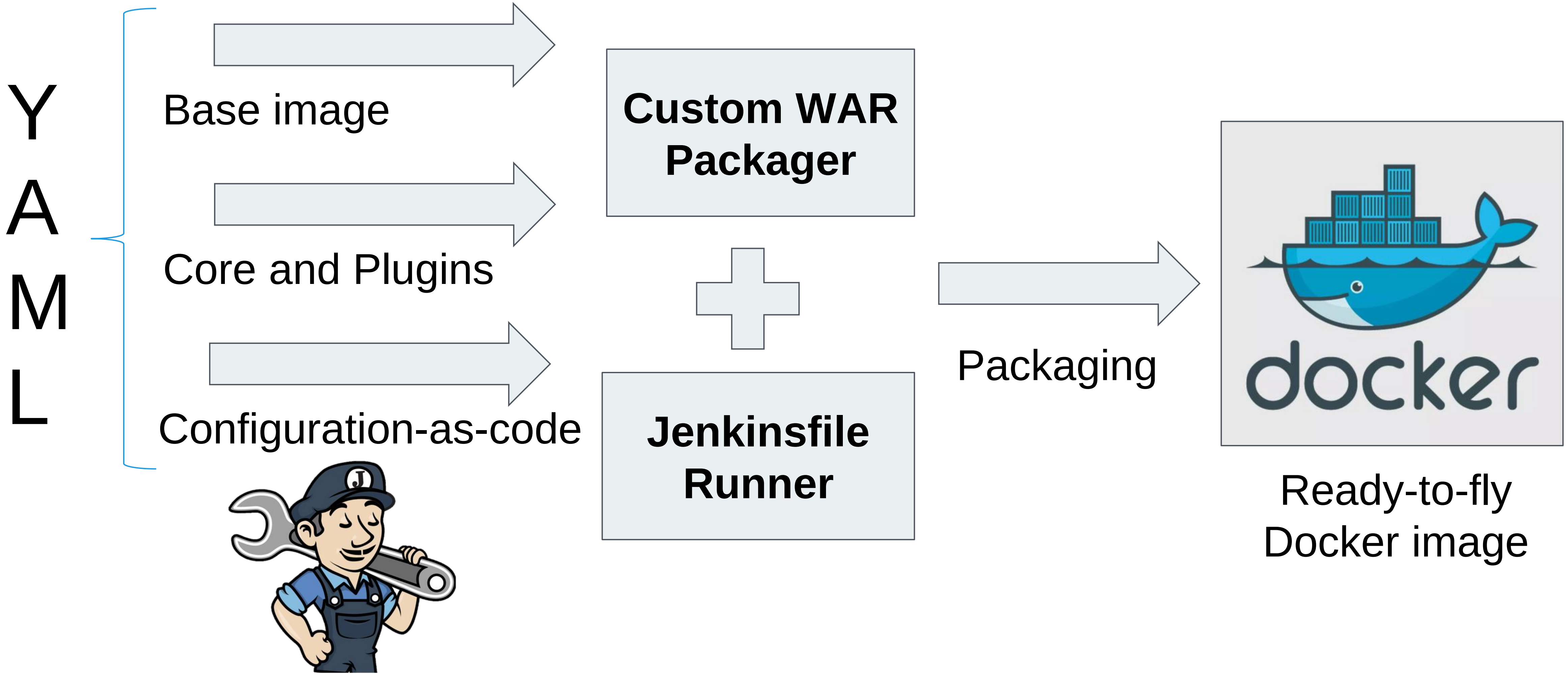
Vanilla: Docker/Binary in the official repo

- Basic plugins: Pipeline, JCasC, etc.
- GitHub: jenkinsci/jenkinsfile-runner
- DockerHub: jenkins4eval/jenkinsfile-runner

Custom JFR packages

- Way to go in 95% of cases
- Ready-to-fly Docker images with plugins and configs
- Extending the Vanilla Docker image
- https://jenkins.io/blog/2018/10/16/custom-war-packager/

Building custom Jenkinsfile Runner images



YAML

Base image

Core and Plugins

Configuration-as-code

Custom WAR Packager

Jenkinsfile Runner

Packaging

Ready-to-fly Docker image

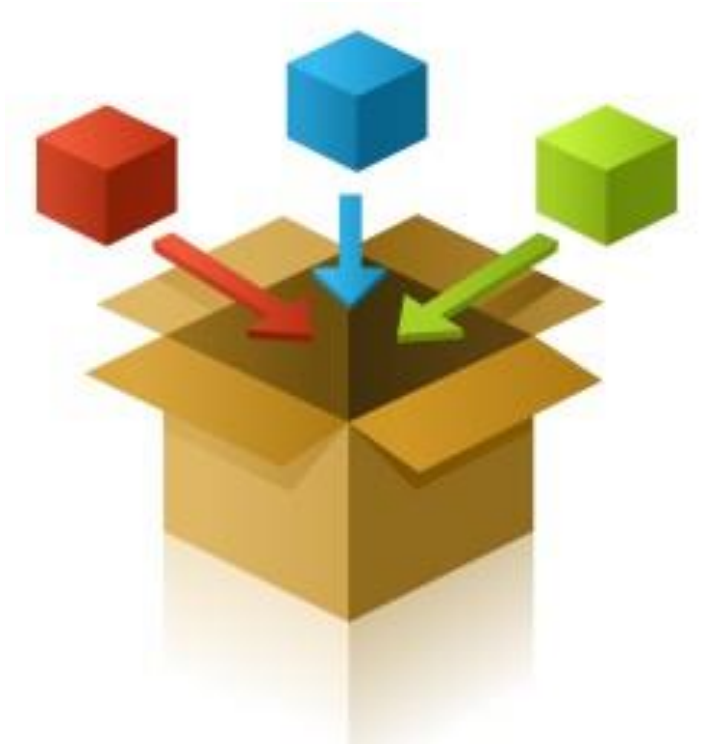
https://jenkins.io/blog/2018/10/16/custom-war-packager/

Custom WAR Packager and Self-configuration

1. Custom WAR Packager is managed by a config YAML
2. Docker image: plugins.txt, Java flags, etc.
3. Groovy Init Scripts
4. AND: Configuration-as-Code Plugin

   https://plugins.jenkins.io/configuration-as-code

   Allows configuring Jenkins from YAML



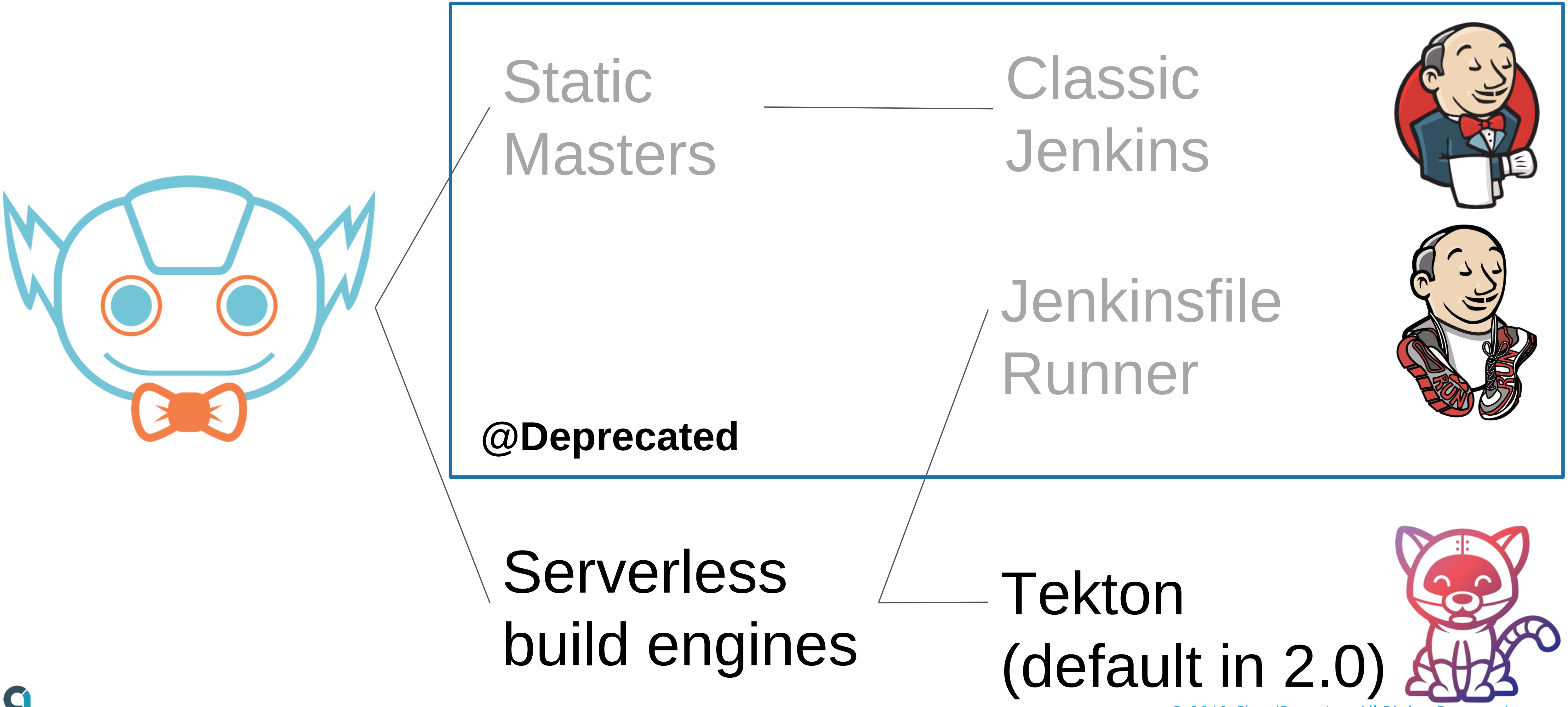
https://github.com/jenkinsci/custom-war-packager

Jenkins X. Serverless Mode (before April 2019)

Serverless Jenkins with Jenkins X

https://medium.com/@jdrawlings/serverless-jenkins-with-jenkins-x-9134cbfe6870

Jenkins X 2.x. Modes and engines



Static Masters

Classic Jenkins

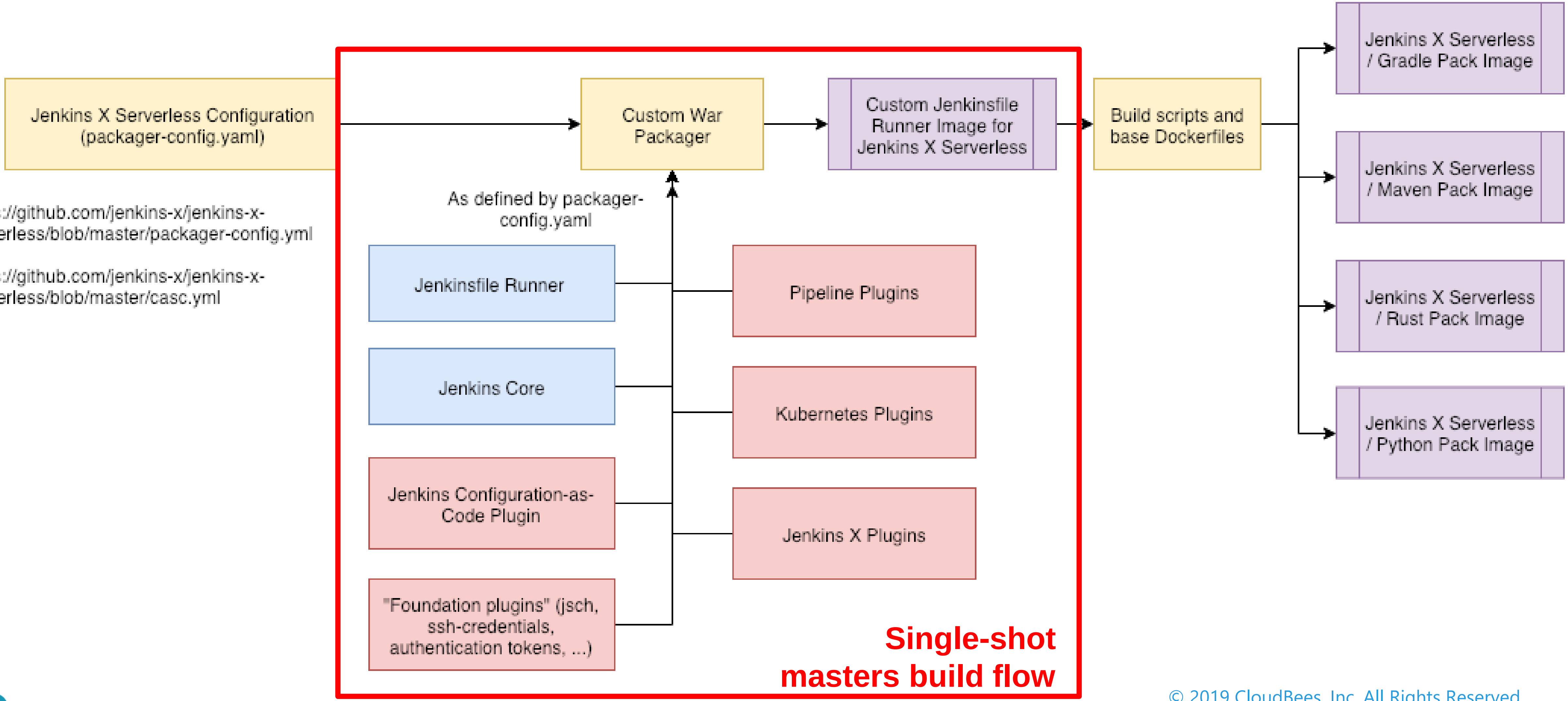
Jenkinsfile Runner

@Deprecated

Serverless build engines

Tekton (default in 2.0)

Jenkins X. Serverless Mode packaging



Jenkins X Serverless Configuration (packager-config.yaml)

Custom War Packager

Custom Jenkinsfile Runner Image for Jenkins X Serverless

Build scripts and base Dockerfiles

Jenkins X Serverless / Gradle Pack Image

Jenkins X Serverless / Maven Pack Image

Jenkins X Serverless / Rust Pack Image

Jenkins X Serverless / Python Pack Image

As defined by packager-config.yaml

Jenkinsfile Runner

Jenkins Core

Jenkins Configuration-as-Code Plugin

"Foundation plugins" (jsch, ssh-credentials, authentication tokens, ...)

Pipeline Plugins

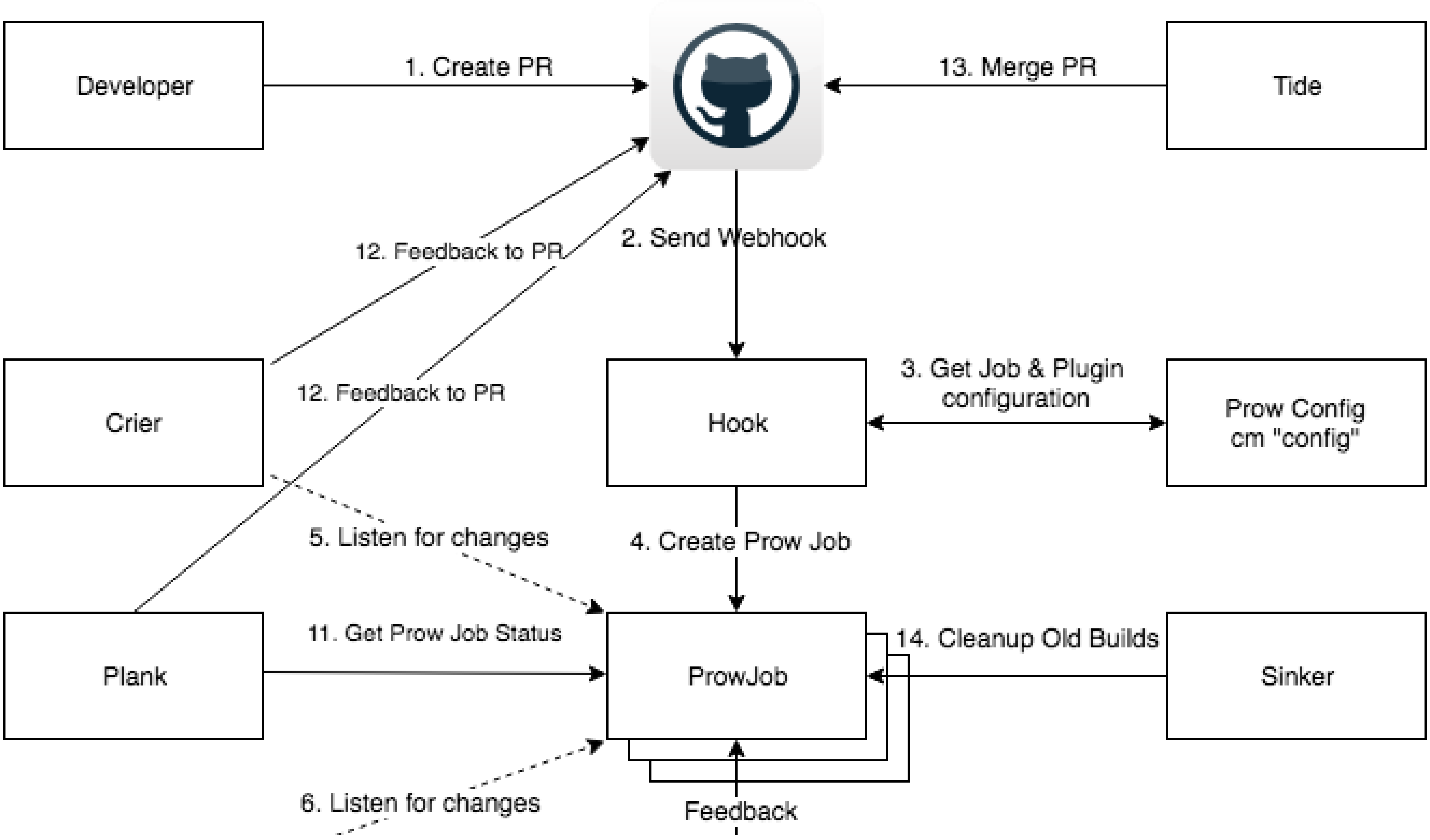
Kubernetes Plugins

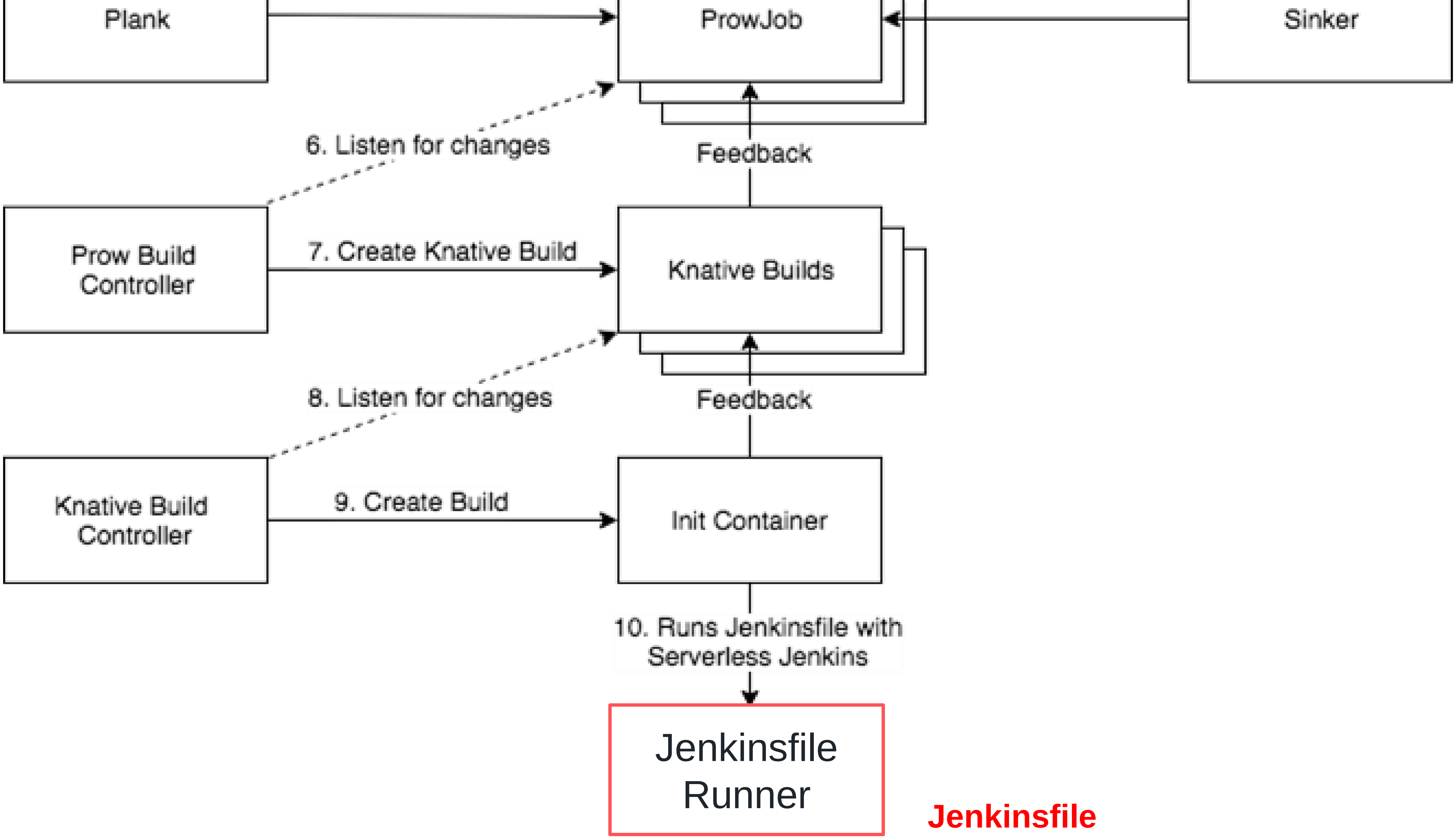
Jenkins X Plugins

Single-shot masters build flow

https://github.com/jenkins-x/jenkins-x-serverless/blob/master/packager-config.yml

https://github.com/jenkins-x/jenkins-x-serverless/blob/master/casc.yml


Developer
1. Create PR
13. Merge PR
Tide
12. Feedback to PR
2. Send Webhook
12. Feedback to PR
Crier
Hook
3. Get Job & Plugin configuration
Prow Config cm "config"
5. Listen for changes
4. Create Prow Job
11. Get Prow Job Status
Plank
ProwJob
14. Cleanup Old Builds
Sinker
6. Listen for changes
Feedback



Plank

ProwJob

Sinker

6. Listen for changes

Feedback

Prow Build Controller

7. Create Knative Build

Knative Builds

8. Listen for changes

Feedback

Knative Build Controller

9. Create Build

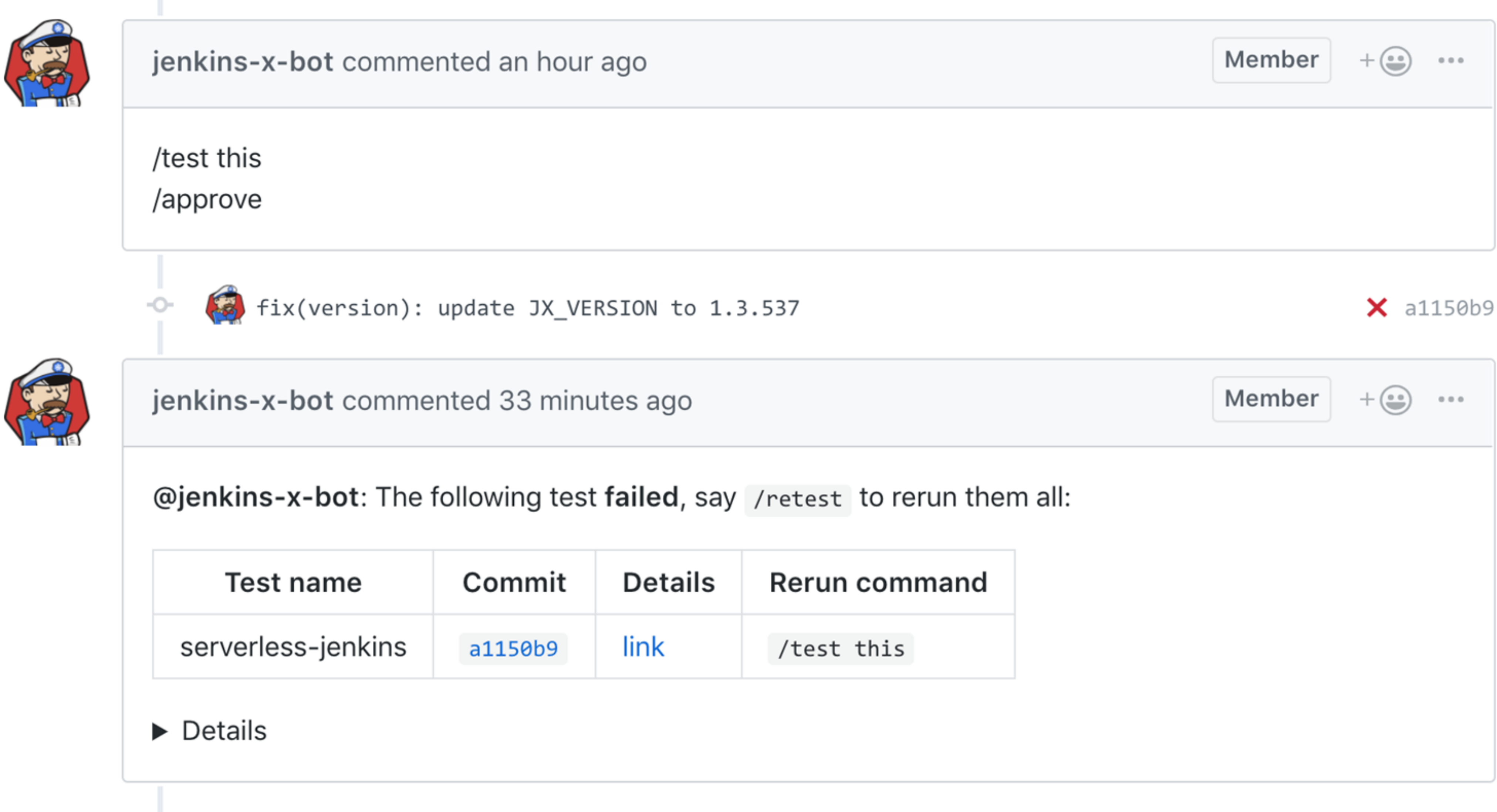
Init Container

10. Runs Jenkinsfile with Serverless Jenkins

Jenkinsfile Runner

Jenkinsfile Runner is Here

CommentOps - communication with users



jenkins-x-bot commented an hour ago

/test this
/approve

fix(version): update JX_VERSION to 1.3.537 ✗ a1150b9

jenkins-x-bot commented 33 minutes ago

@jenkins-x-bot: The following test **failed**, say `/retest` to rerun them all:

Test name	Commit	Details	Rerun command
serverless-jenkins	`a1150b9`	link	`/test this`

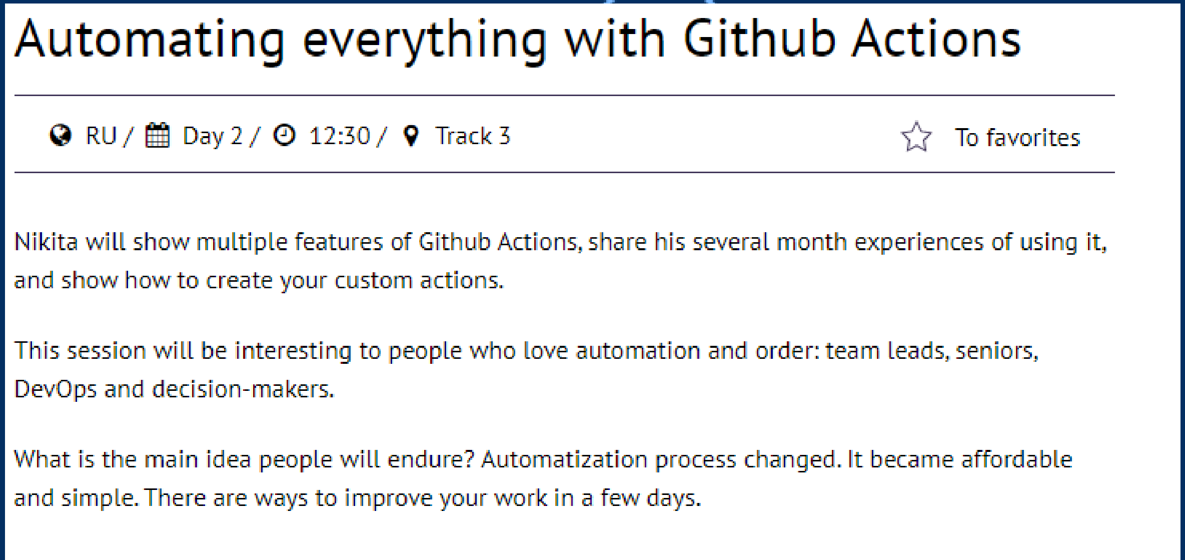
▶ Details

GitHub Actions

… and Jenkinsfile Runner

GitHub Actions

Automate your workflow from idea to production.



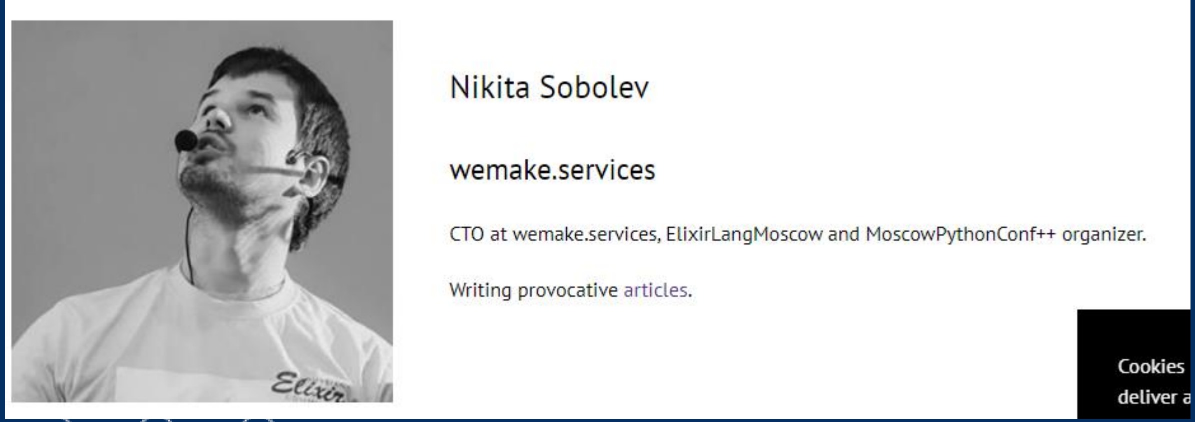
Automating everything with Github Actions

RU / Day 2 / 12:30 / Track 3 ☆ To favorites

Nikita will show multiple features of Github Actions, share his several month experiences of using it, and show how to create your custom actions.

This session will be interesting to people who love automation and order: team leads, seniors, DevOps and decision-makers.

What is the main idea people will endure? Automatization process changed. It became affordable and simple. There are ways to improve your work in a few days.



Nikita Sobolev

wemake.services

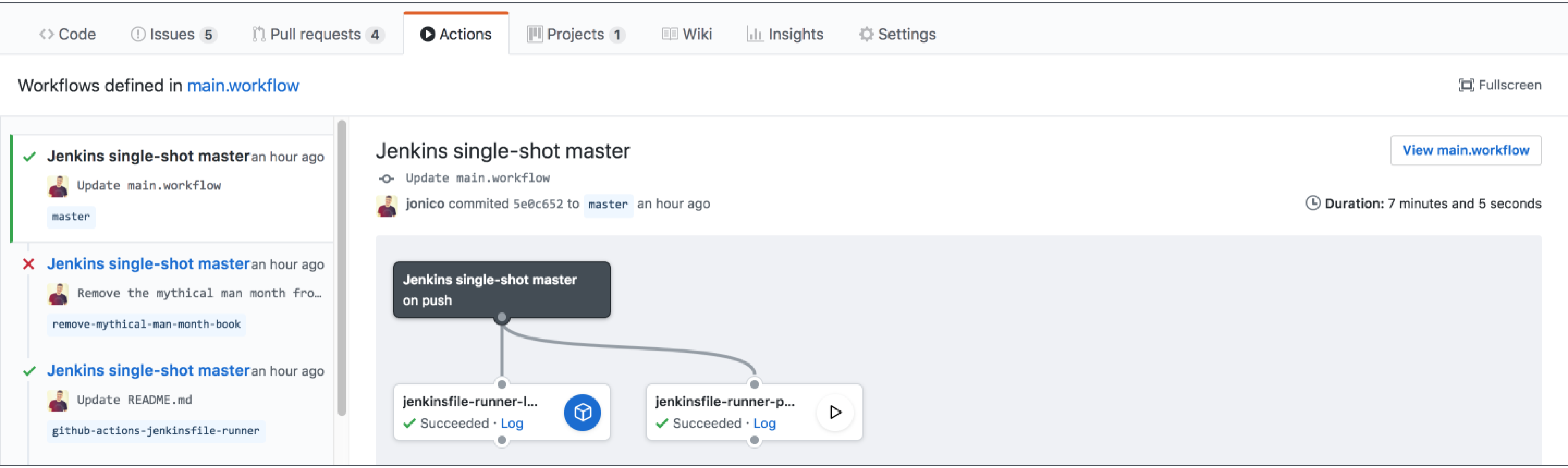
CTO at wemake.services, ElixirLangMoscow and MoscowPythonConf++ organizer.

Writing provocative articles.

Cookies
deliver a

https://github.com/features/actions

GitHub Actions!



Workflows defined in main.workflow

✓ Jenkins single-shot master an hour ago
Update main.workflow
master

✕ Jenkins single-shot master an hour ago
Remove the mythical man month fro…
remove-mythical-man-month-book

✓ Jenkins single-shot master an hour ago
Update README.md
github-actions-jenkinsfile-runner

Jenkins single-shot master

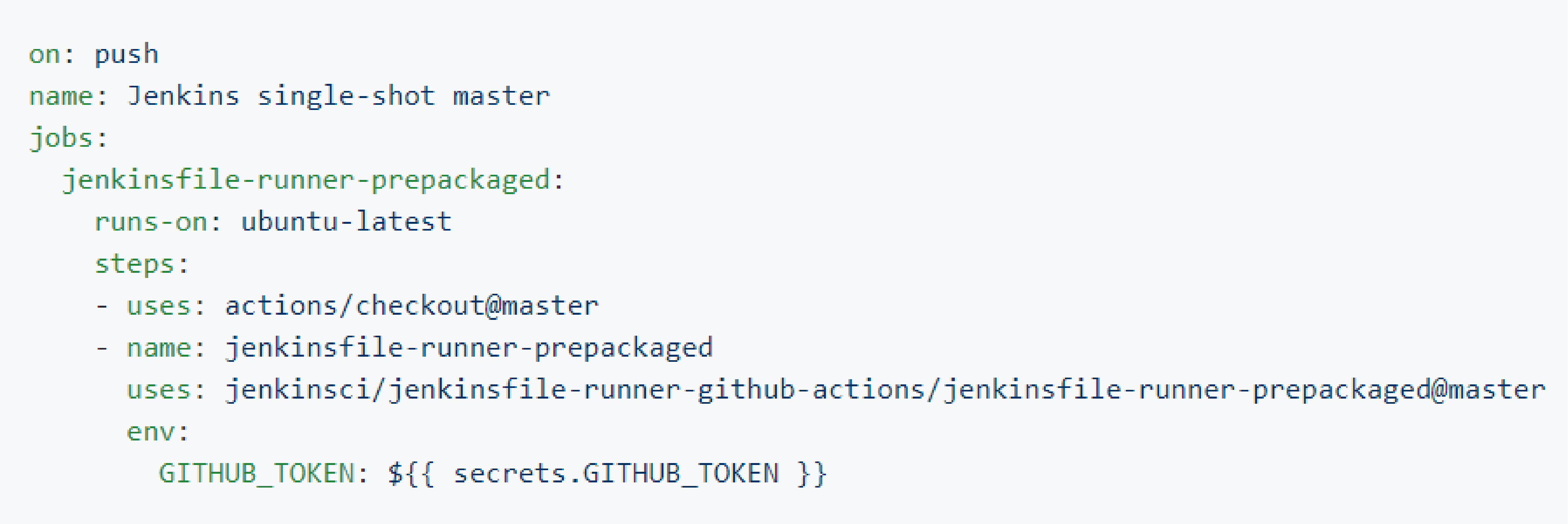
Update main.workflow

jonico commited 5e0c652 to master an hour ago

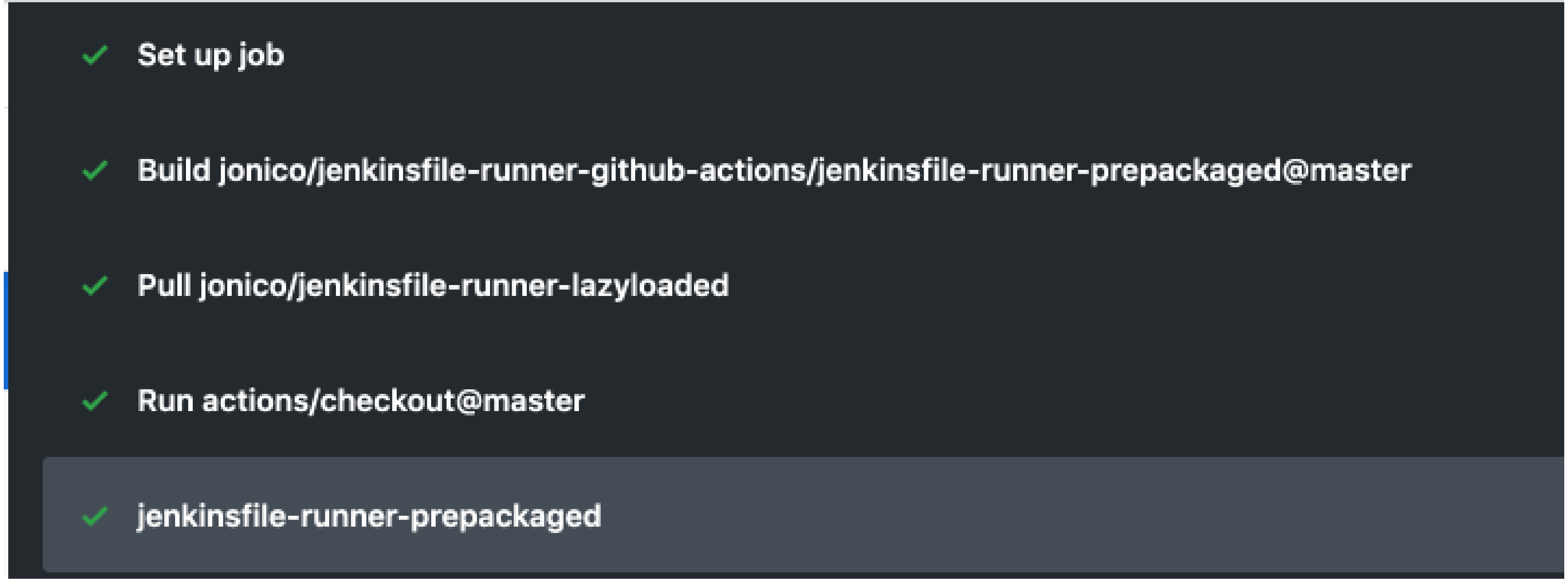
Duration: 7 minutes and 5 seconds

https://github.com/jonico/jenkinsfile-runner-github-actions

Sample definition



```
on: push
name: Jenkins single-shot master
jobs:
  jenkinsfile-runner-prepackaged:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@master
    - name: jenkinsfile-runner-prepackaged
      uses: jenkinsci/jenkinsfile-runner-github-actions/jenkinsfile-runner-prepackaged@master
      env:
        GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```



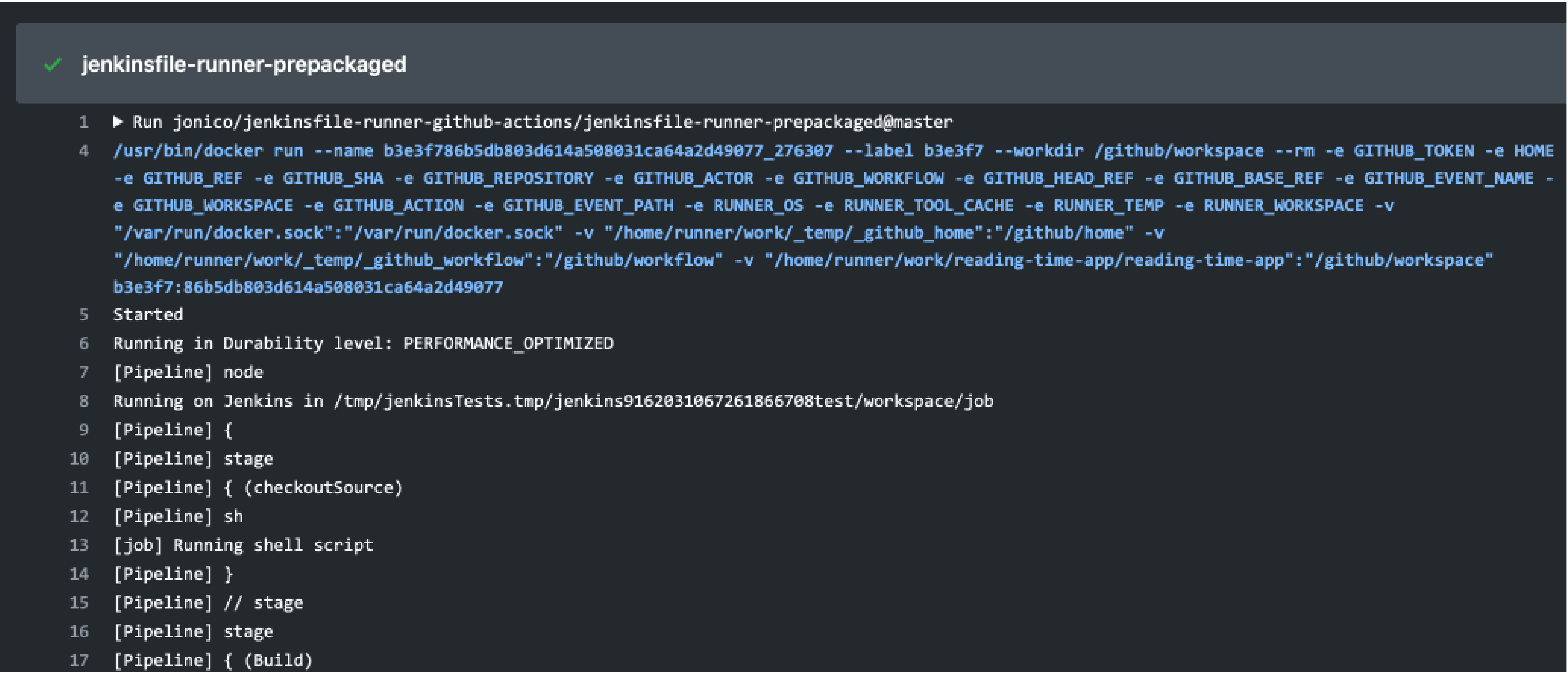
✓ Set up job

✓ Build jonico/jenkinsfile-runner-github-actions/jenkinsfile-runner-prepackaged@master

✓ Pull jonico/jenkinsfile-runner-lazyloaded

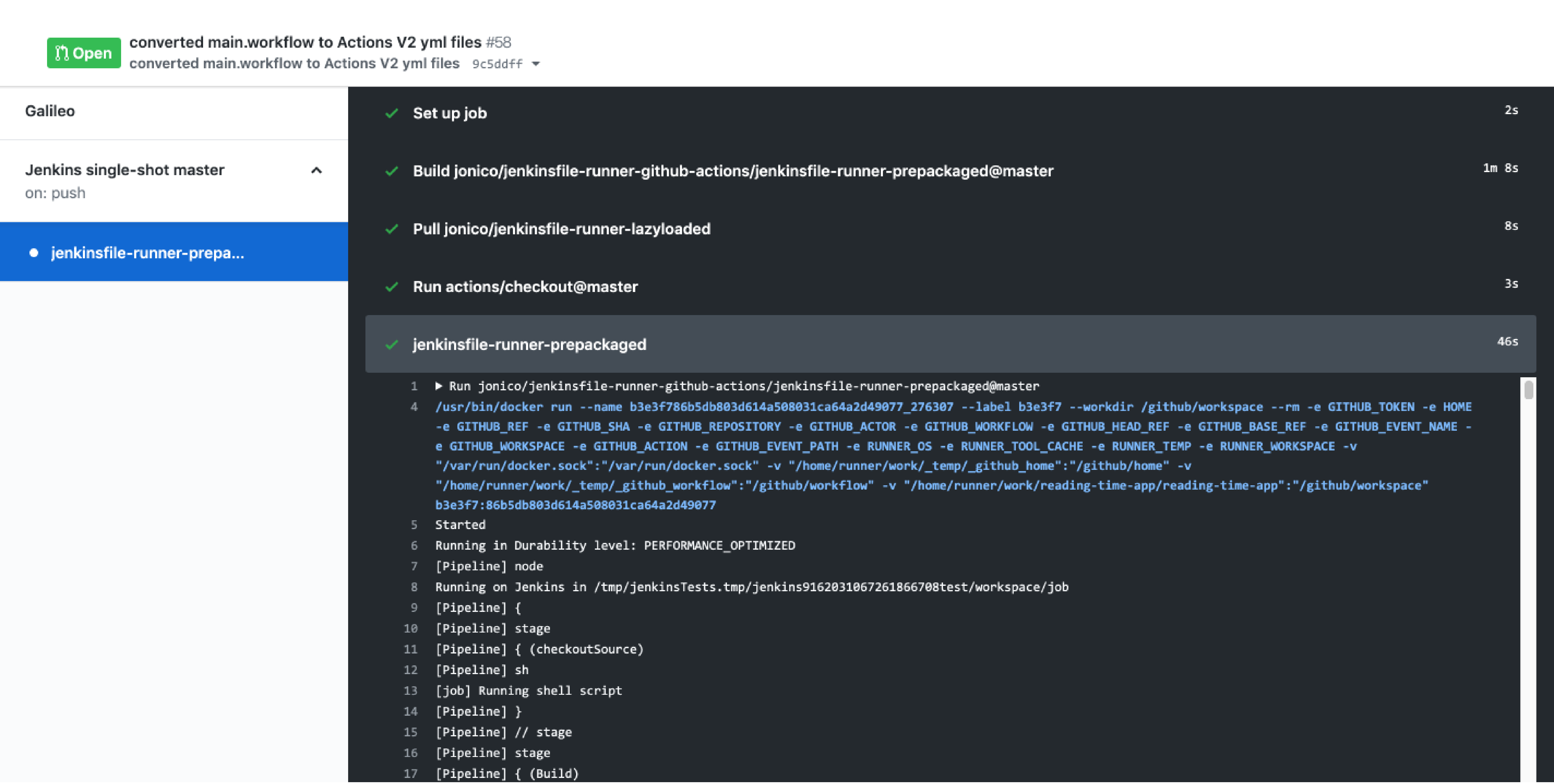
✓ Run actions/checkout@master

✓ jenkinsfile-runner-prepackaged



✓ jenkinsfile-runner-prepackaged

```
▶ Run jonico/jenkinsfile-runner-github-actions/jenkinsfile-runner-prepackaged@master
/usr/bin/docker run --name b3e3f786b5db803d614a508031ca64a2d49077_276307 --label b3e3f7 --workdir /github/workspace --rm -e GITHUB_TOKEN -e HOME
-e GITHUB_REF -e GITHUB_SHA -e GITHUB_REPOSITORY -e GITHUB_ACTOR -e GITHUB_WORKFLOW -e GITHUB_HEAD_REF -e GITHUB_BASE_REF -e GITHUB_EVENT_NAME -
e GITHUB_WORKSPACE -e GITHUB_ACTION -e GITHUB_EVENT_PATH -e RUNNER_OS -e RUNNER_TOOL_CACHE -e RUNNER_TEMP -e RUNNER_WORKSPACE -v
"/var/run/docker.sock":"/var/run/docker.sock" -v "/home/runner/work/_temp/_github_home":"/github/home" -v
"/home/runner/work/_temp/_github_workflow":"/github/workflow" -v "/home/runner/work/reading-time-app/reading-time-app":"/github/workspace"
b3e3f7:86b5db803d614a508031ca64a2d49077
Started
Running in Durability level: PERFORMANCE_OPTIMIZED
[Pipeline] node
Running on Jenkins in /tmp/jenkinsTests.tmp/jenkins9162031067261866708test/workspace/job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (checkoutSource)
[Pipeline] sh
[job] Running shell script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
```



ⁿ Open converted main.workflow to Actions V2 yml files #58
converted main.workflow to Actions V2 yml files 9c5ddff ▾

Galileo

Jenkins single-shot master
on: push

● jenkinsfile-runner-prepa...

✓ Set up job 2s

✓ Build jonico/jenkinsfile-runner-github-actions/jenkinsfile-runner-prepackaged@master 1m 8s

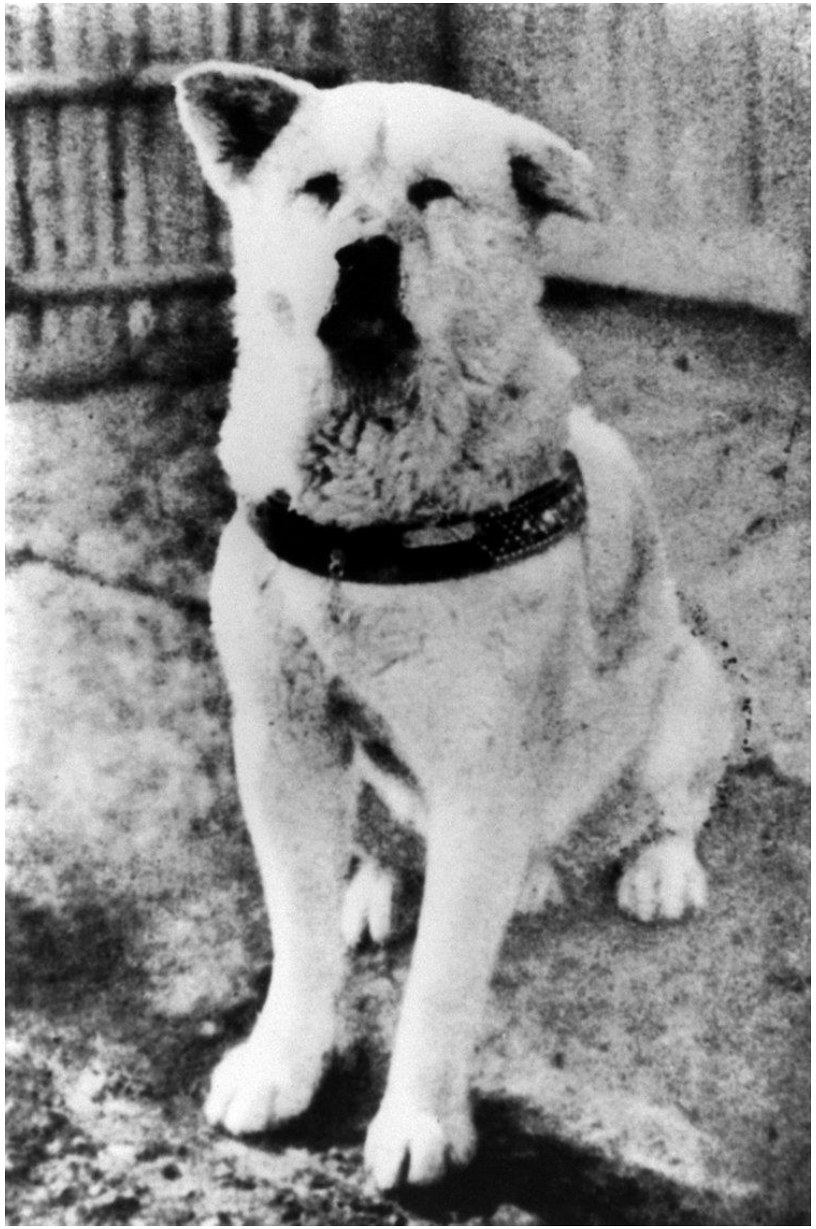
✓ Pull jonico/jenkinsfile-runner-lazyloaded 8s

✓ Run actions/checkout@master 3s

✓ jenkinsfile-runner-prepackaged 46s

```
▶ Run jonico/jenkinsfile-runner-github-actions/jenkinsfile-runner-prepackaged@master
/usr/bin/docker run --name b3e3f786b5db803d614a508031ca64a2d49077_276307 --label b3e3f7 --workdir /github/workspace --rm -e GITHUB_TOKEN -e HOME
-e GITHUB_REF -e GITHUB_SHA -e GITHUB_REPOSITORY -e GITHUB_ACTOR -e GITHUB_WORKFLOW -e GITHUB_HEAD_REF -e GITHUB_BASE_REF -e GITHUB_EVENT_NAME -
e GITHUB_WORKSPACE -e GITHUB_ACTION -e GITHUB_EVENT_PATH -e RUNNER_OS -e RUNNER_TOOL_CACHE -e RUNNER_TEMP -e RUNNER_WORKSPACE -v
"/var/run/docker.sock":"/var/run/docker.sock" -v "/home/runner/work/_temp/_github_home":"/github/home" -v
"/home/runner/work/_temp/_github_workflow":"/github/workflow" -v "/home/runner/work/reading-time-app/reading-time-app":"/github/workspace"
b3e3f7:86b5db803d614a508031ca64a2d49077
Started
Running in Durability level: PERFORMANCE_OPTIMIZED
[Pipeline] node
Running on Jenkins in /tmp/jenkinsTests.tmp/jenkins9162031067261866708test/workspace/job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (checkoutSource)
[Pipeline] sh
[job] Running shell script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
```

GitHub Actions is in Limited Beta



Jenkinsfiles on CodeShip?

Jenkinsfile Runner on CloudBees CodeShip

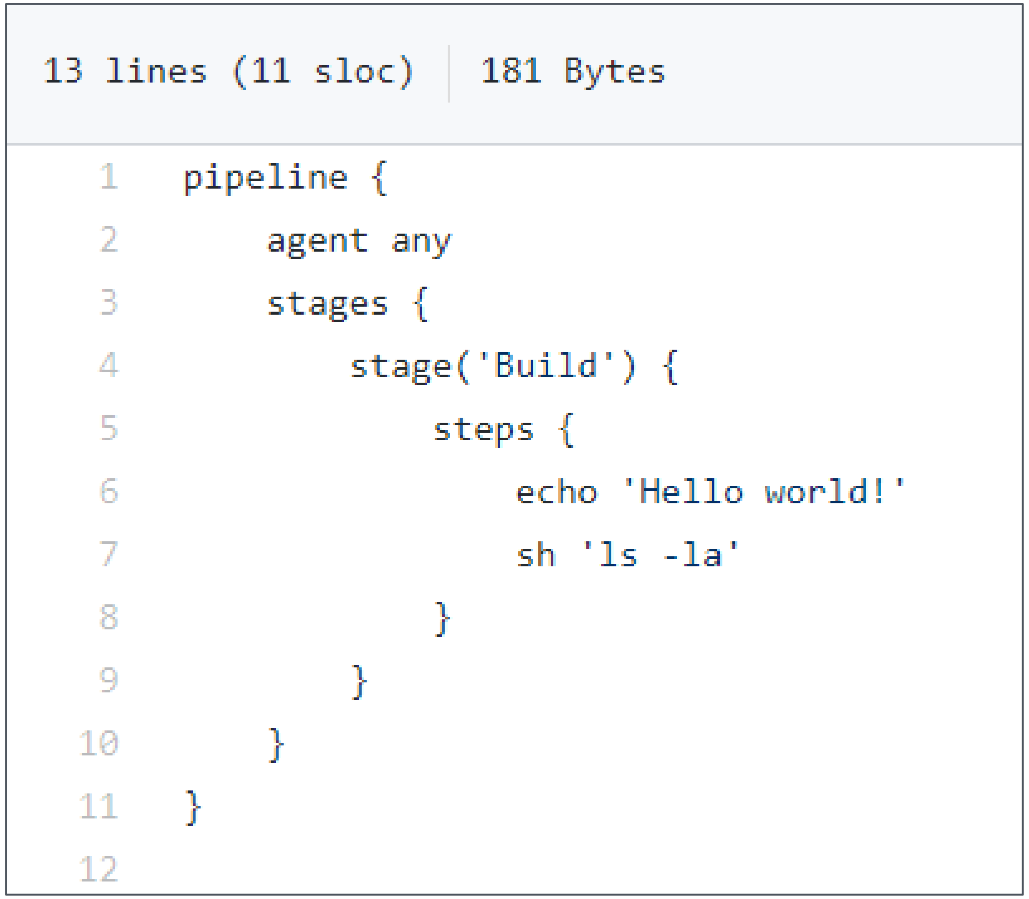
- Jenkins has no good SaaS
- CodeShip Pro - CI SaaS with native Docker support
- Free tiers for open-source projects
- What if we run Jenkins there?

https://codeship.com/features/pro

Demo 1

https://github.com/oleg-nenashev/codeship-jenkinsfile

Jenkinsfile

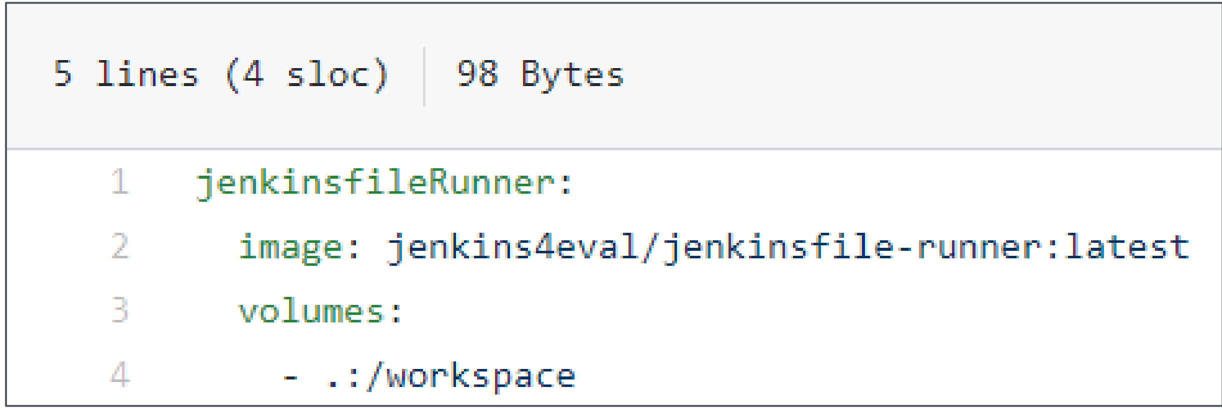


13 lines (11 sloc) | 181 Bytes

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                echo 'Hello world!'
                sh 'ls -la'
            }
        }
    }
}

```

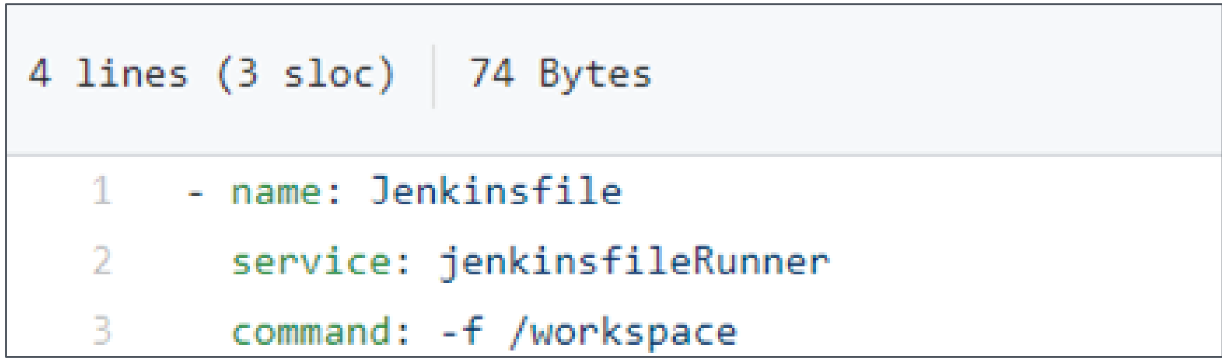
codeship-services.yml



5 lines (4 sloc) | 98 Bytes

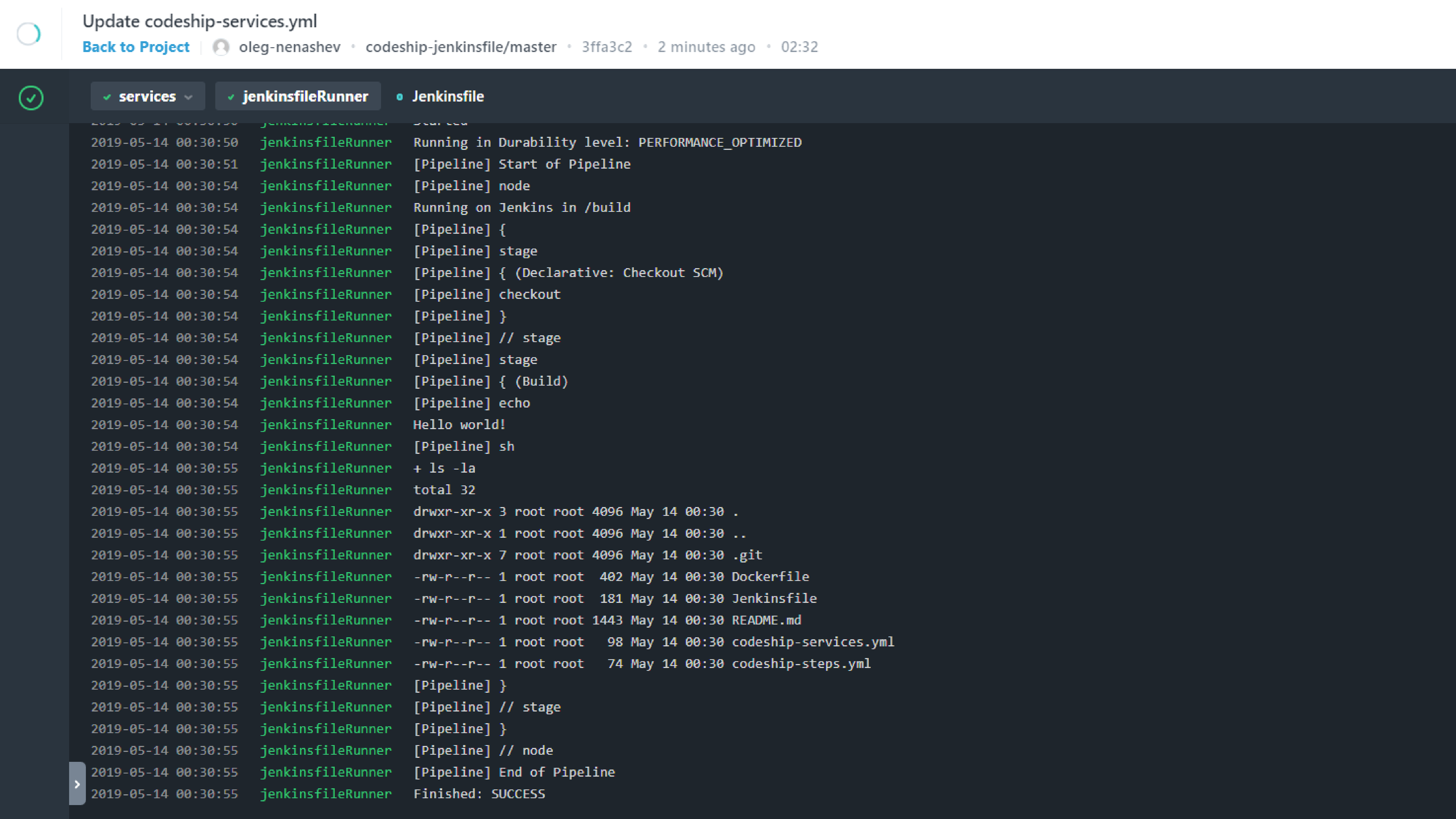
```
jenkinsfileRunner:
  image: jenkins4eval/jenkinsfile-runner:latest
  volumes:
    - .:/workspace
```

codeship-steps.yml



4 lines (3 sloc) | 74 Bytes

```
- name: Jenkinsfile
  service: jenkinsfileRunner
  command: -f /workspace
```



Update codeship-services.yml

Back to Project
oleg-nenashev • codeship-jenkinsfile/master • 3ffa3c2 • 2 minutes ago • 02:32
✓ services
✓ jenkinsfileRunner
Jenkinsfile

```
2019-05-14 00:30:50  jenkinsfileRunner   Started
2019-05-14 00:30:50  jenkinsfileRunner   Running in Durability level: PERFORMANCE_OPTIMIZED
2019-05-14 00:30:51  jenkinsfileRunner   [Pipeline] Start of Pipeline
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] node
2019-05-14 00:30:54  jenkinsfileRunner   Running on Jenkins in /build
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] {
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] stage
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] { (Declarative: Checkout SCM)
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] checkout
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] }
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] // stage
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] stage
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] { (Build)
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] echo
2019-05-14 00:30:54  jenkinsfileRunner   Hello world!
2019-05-14 00:30:54  jenkinsfileRunner   [Pipeline] sh
2019-05-14 00:30:55  jenkinsfileRunner   + ls -la
2019-05-14 00:30:55  jenkinsfileRunner   total 32
2019-05-14 00:30:55  jenkinsfileRunner   drwxr-xr-x 3 root root 4096 May 14 00:30 .
2019-05-14 00:30:55  jenkinsfileRunner   drwxr-xr-x 1 root root 4096 May 14 00:30 ..
2019-05-14 00:30:55  jenkinsfileRunner   drwxr-xr-x 7 root root 4096 May 14 00:30 .git
2019-05-14 00:30:55  jenkinsfileRunner   -rw-r--r-- 1 root root  402 May 14 00:30 Dockerfile
2019-05-14 00:30:55  jenkinsfileRunner   -rw-r--r-- 1 root root  181 May 14 00:30 Jenkinsfile
2019-05-14 00:30:55  jenkinsfileRunner   -rw-r--r-- 1 root root 1443 May 14 00:30 README.md
2019-05-14 00:30:55  jenkinsfileRunner   -rw-r--r-- 1 root root   98 May 14 00:30 codeship-services.yml
2019-05-14 00:30:55  jenkinsfileRunner   -rw-r--r-- 1 root root   74 May 14 00:30 codeship-steps.yml
2019-05-14 00:30:55  jenkinsfileRunner   [Pipeline] }
2019-05-14 00:30:55  jenkinsfileRunner   [Pipeline] // stage
2019-05-14 00:30:55  jenkinsfileRunner   [Pipeline] }
2019-05-14 00:30:55  jenkinsfileRunner   [Pipeline] // node
2019-05-14 00:30:55  jenkinsfileRunner   [Pipeline] End of Pipeline
2019-05-14 00:30:55  jenkinsfileRunner   Finished: SUCCESS
```

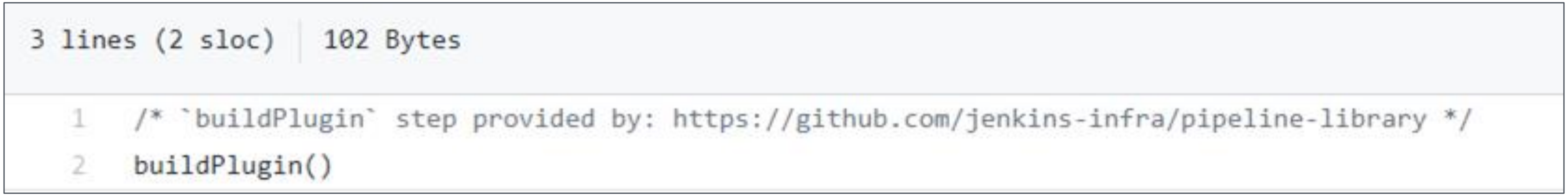
Example 2. Building a Jenkins plugin in CodeShip

https://github.com/oleg-nenashev/mailer-plugin/tree/codeship-demo

- Using ci.jenkins.io-runner
  - https://github.com/jenkinsci/ci.jenkins.io-runner
- Using the standard Jenkins Pipeline library
  - https://github.com/jenkins-infra/pipeline-library
- Using standard Jenkinsfile
  - buildPlugin()

Example 2. Building a Jenkins plugin in CodeShip

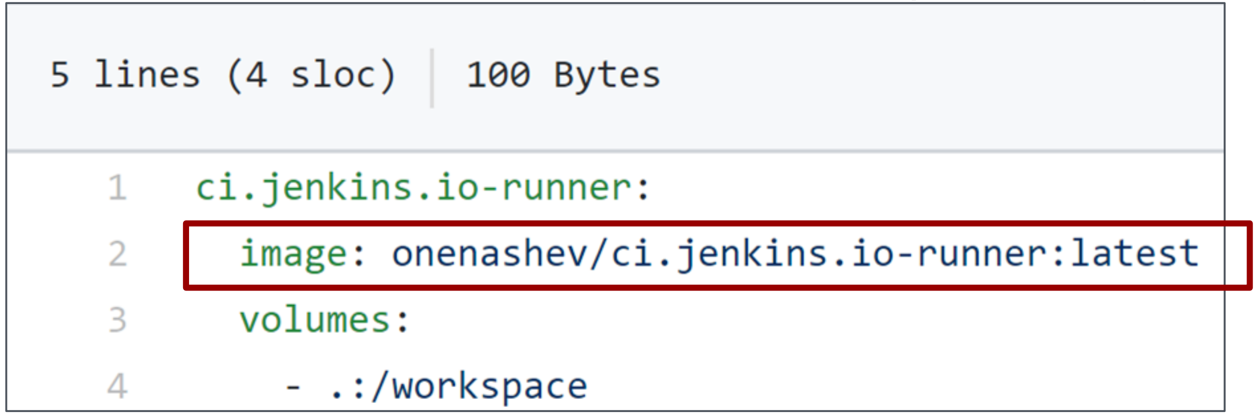
Jenkinsfile



3 lines (2 sloc) | 102 Bytes

```
/* `buildPlugin` step provided by: https://github.com/jenkins-infra/pipeline-library */
buildPlugin()
```

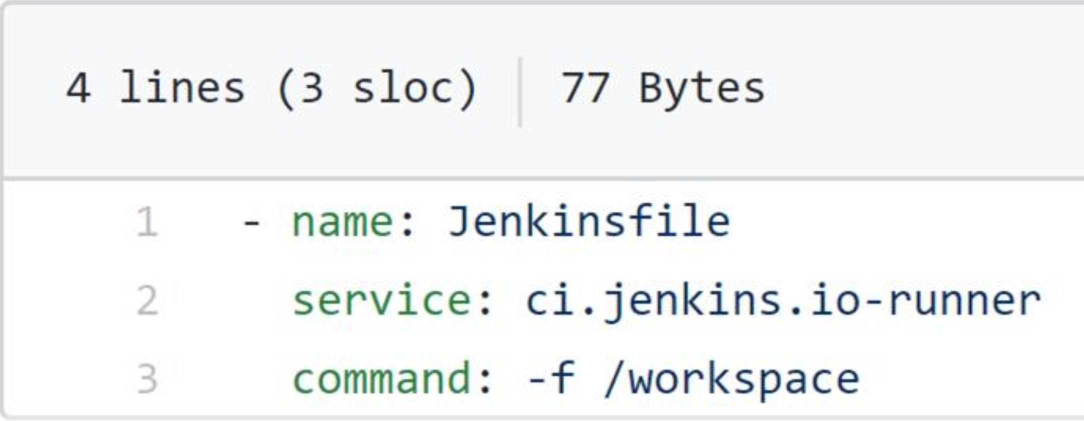
codeship-services.yml



5 lines (4 sloc) | 100 Bytes

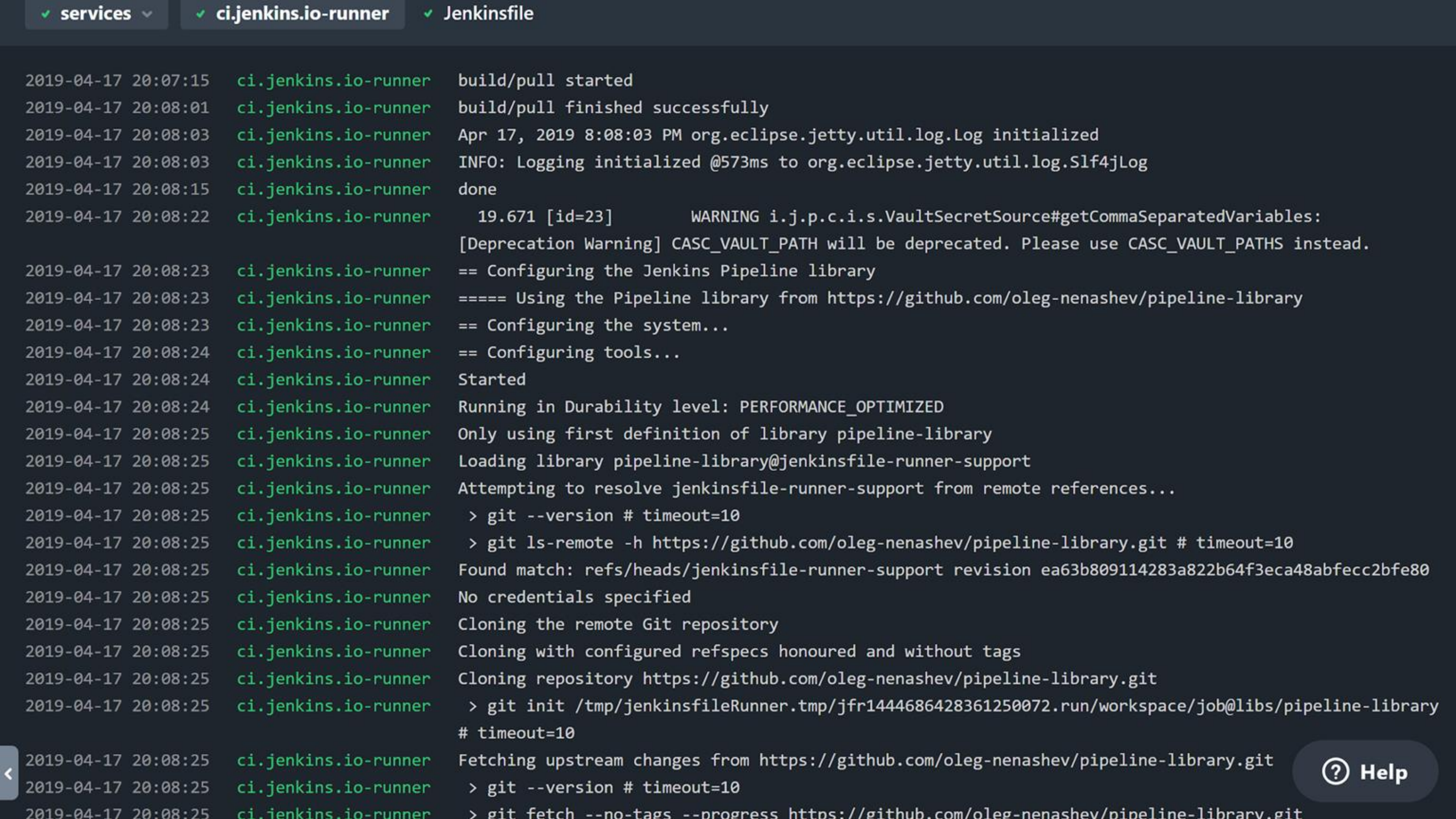
```
ci.jenkins.io-runner:
  image: onenashev/ci.jenkins.io-runner:latest
  volumes:
    - .:/workspace
```

codeship-steps.yml



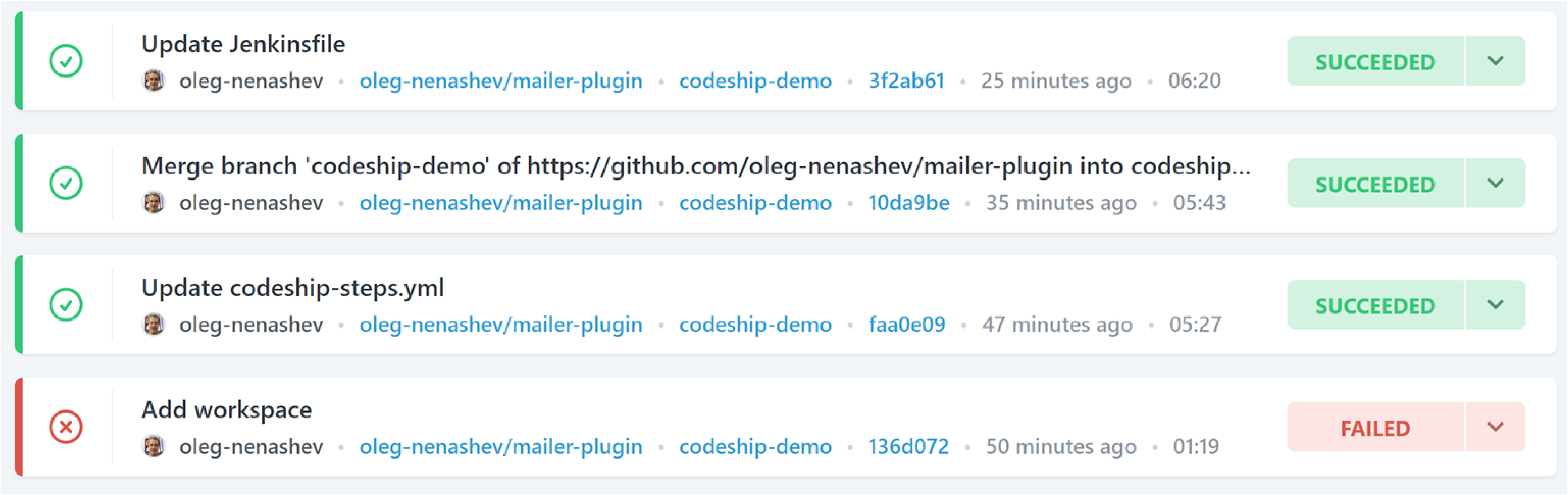
4 lines (3 sloc) | 77 Bytes

```
- name: Jenkinsfile
  service: ci.jenkins.io-runner
  command: -f /workspace
```



```
2019-04-17 20:07:15   ci.jenkins.io-runner   build/pull started
2019-04-17 20:08:01   ci.jenkins.io-runner   build/pull finished successfully
2019-04-17 20:08:03   ci.jenkins.io-runner   Apr 17, 2019 8:08:03 PM org.eclipse.jetty.util.log.Log initialized
2019-04-17 20:08:03   ci.jenkins.io-runner   INFO: Logging initialized @573ms to org.eclipse.jetty.util.log.Slf4jLog
2019-04-17 20:08:15   ci.jenkins.io-runner   done
2019-04-17 20:08:22   ci.jenkins.io-runner     19.671 [id=23]        WARNING i.j.p.c.i.s.VaultSecretSource#getCommaSeparatedVariables:
                                             [Deprecation Warning] CASC_VAULT_PATH will be deprecated. Please use CASC_VAULT_PATHS instead.
2019-04-17 20:08:23   ci.jenkins.io-runner   == Configuring the Jenkins Pipeline library
2019-04-17 20:08:23   ci.jenkins.io-runner   ===== Using the Pipeline library from https://github.com/oleg-nenashev/pipeline-library
2019-04-17 20:08:23   ci.jenkins.io-runner   == Configuring the system...
2019-04-17 20:08:24   ci.jenkins.io-runner   == Configuring tools...
2019-04-17 20:08:24   ci.jenkins.io-runner   Started
2019-04-17 20:08:24   ci.jenkins.io-runner   Running in Durability level: PERFORMANCE_OPTIMIZED
2019-04-17 20:08:25   ci.jenkins.io-runner   Only using first definition of library pipeline-library
2019-04-17 20:08:25   ci.jenkins.io-runner   Loading library pipeline-library@jenkinsfile-runner-support
2019-04-17 20:08:25   ci.jenkins.io-runner   Attempting to resolve jenkinsfile-runner-support from remote references...
2019-04-17 20:08:25   ci.jenkins.io-runner    > git --version # timeout=10
2019-04-17 20:08:25   ci.jenkins.io-runner    > git ls-remote -h https://github.com/oleg-nenashev/pipeline-library.git # timeout=10
2019-04-17 20:08:25   ci.jenkins.io-runner   Found match: refs/heads/jenkinsfile-runner-support revision ea63b809114283a822b64f3eca48abfecc2bfe80
2019-04-17 20:08:25   ci.jenkins.io-runner   No credentials specified
2019-04-17 20:08:25   ci.jenkins.io-runner   Cloning the remote Git repository
2019-04-17 20:08:25   ci.jenkins.io-runner   Cloning with configured refspecs honoured and without tags
2019-04-17 20:08:25   ci.jenkins.io-runner   Cloning repository https://github.com/oleg-nenashev/pipeline-library.git
2019-04-17 20:08:25   ci.jenkins.io-runner    > git init /tmp/jenkinsfileRunner.tmp/jfr1444686428361250072.run/workspace/job@libs/pipeline-library
                                             # timeout=10
2019-04-17 20:08:25   ci.jenkins.io-runner   Fetching upstream changes from https://github.com/oleg-nenashev/pipeline-library.git
2019-04-17 20:08:25   ci.jenkins.io-runner    > git --version # timeout=10
2019-04-17 20:08:25   ci.jenkins.io-runner    > git fetch --no-tags --progress https://github.com/oleg-nenashev/pipeline-library.git
```

CodeShip. Build history



✓ Update Jenkinsfile
oleg-nenashev • oleg-nenashev/mailer-plugin • codeship-demo • 3f2ab61 • 25 minutes ago • 06:20
SUCCEEDED

✓ Merge branch 'codeship-demo' of https://github.com/oleg-nenashev/mailer-plugin into codeship...
oleg-nenashev • oleg-nenashev/mailer-plugin • codeship-demo • 10da9be • 35 minutes ago • 05:43
SUCCEEDED

✓ Update codeship-steps.yml
oleg-nenashev • oleg-nenashev/mailer-plugin • codeship-demo • faa0e09 • 47 minutes ago • 05:27
SUCCEEDED

✗ Add workspace
oleg-nenashev • oleg-nenashev/mailer-plugin • codeship-demo • 136d072 • 50 minutes ago • 01:19
FAILED

We talked about...

- Jenkins X
- GitHub Actions
- CloudBees CodeShip

Last but not least?

Jenkins

Running JFR from Jenkins

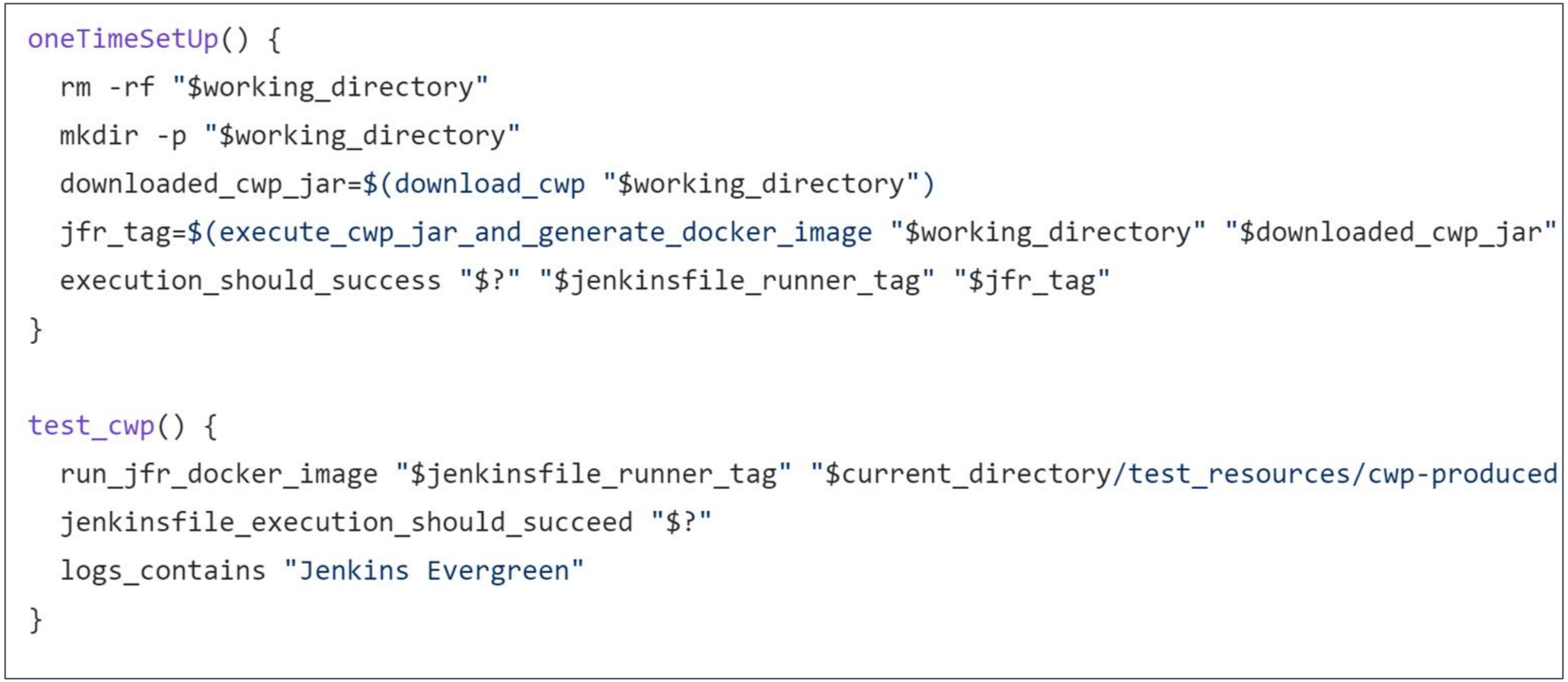
Well, docker run…

TEST ALL THE THINGS
imgflip.com

Jenkinsfile Runner Test Framework

https://github.com/jenkinsci/jenkinsfile-runner-test-framework

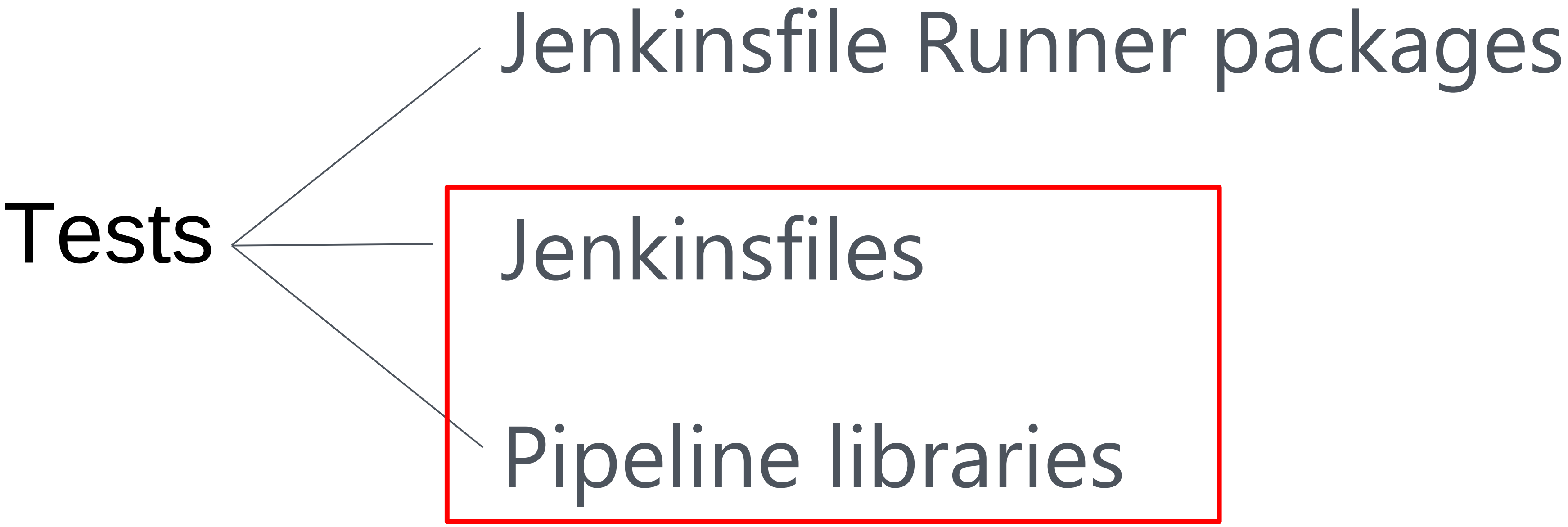
- Custom WAR Packager
- Docker
- shUnit2



```
oneTimeSetUp() {
  rm -rf "$working_directory"
  mkdir -p "$working_directory"
  downloaded_cwp_jar=$(download_cwp "$working_directory")
  jfr_tag=$(execute_cwp_jar_and_generate_docker_image "$working_directory" "$downloaded_cwp_jar"
  execution_should_success "$?" "$jenkinsfile_runner_tag" "$jfr_tag"
}

test_cwp() {
  run_jfr_docker_image "$jenkinsfile_runner_tag" "$current_directory/test_resources/cwp-produced
  jenkinsfile_execution_should_succeed "$?"
  logs_contains "Jenkins Evergreen"
}
```

Jenkinsfile Runner Test Framework



Tests

Jenkinsfile Runner packages

Jenkinsfiles

Pipeline libraries

If Pipeline Unit is not enough…

https://github.com/jenkinsci/JenkinsPipelineUnit

Works for Mock Unit tests

JFR Test framework for integration tests

Future work

- Waiting for user feedback
- Integrating patches
- TBD - Jenkinsfile Runner 1.0
- TBD - Sidecar containers
- Experiments with GraalVM and Quarkus

!=

Takeaways

- Jenkins-as-a-function is real
- You can run your Pipelines everywhere
- There are tools for that!
- Use Jenkins X if applicable (K8s/GitOps)
- Consider Jenkinsfile Runner if not

It is a great time to contribute!

- jenkinsci/jenkinsfile-runner/blob/master/CONTRIBUTING.md

QUESTIONS?

go.cloudbees.com

Contacts:

✉ E-mail: onenashev@cloudbees.com

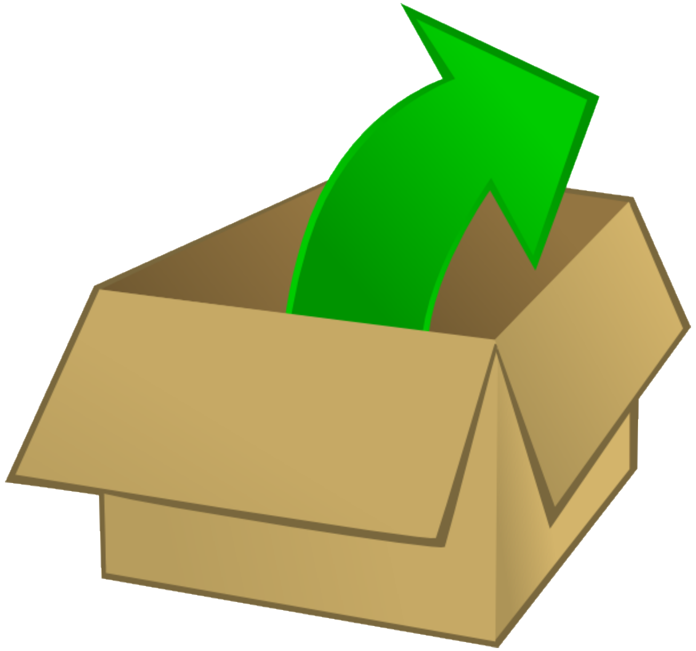
GitHub: oleg-nenashev

Twitter: @oleg_nenashev

+

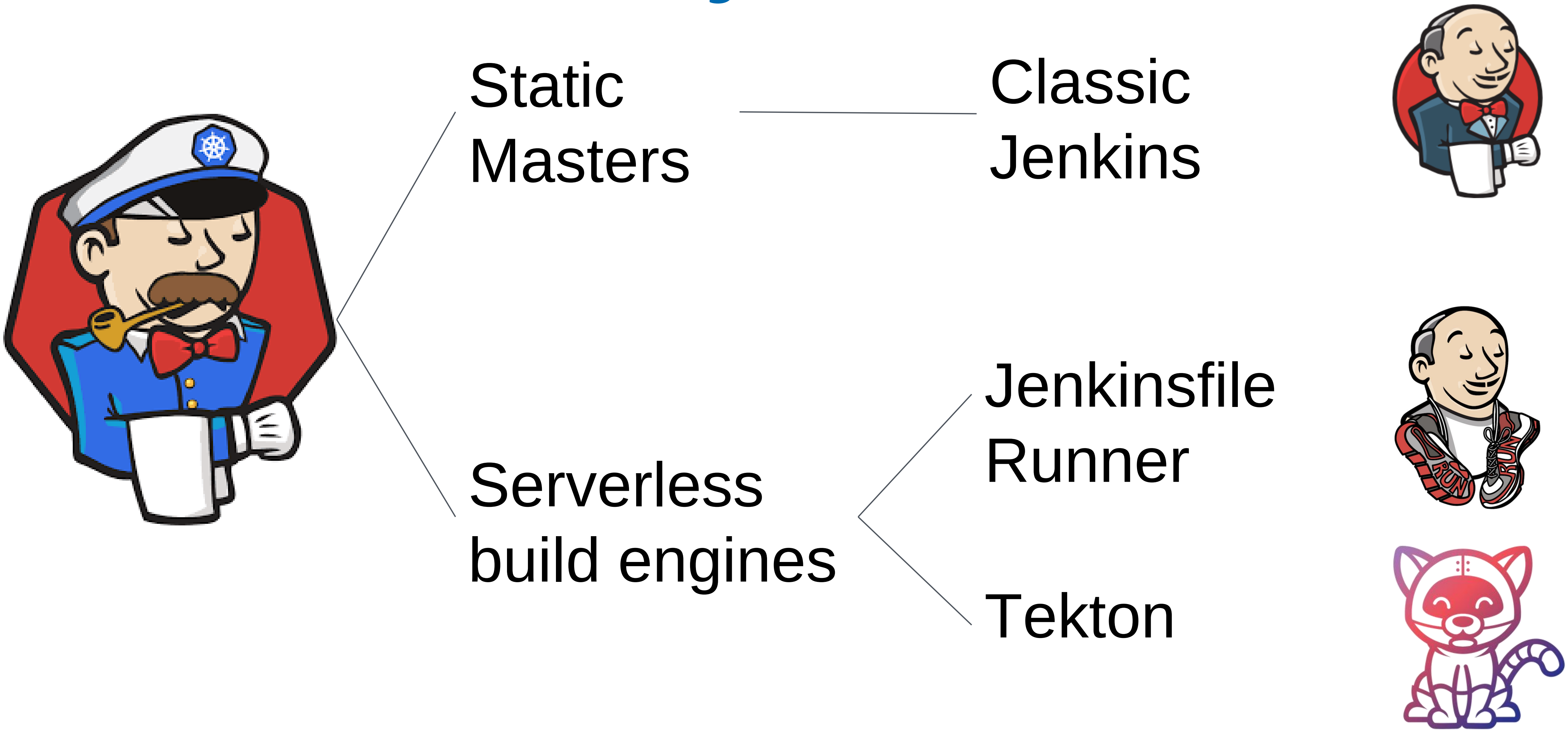
!=

Jenkins X. OOTB experience



- Build packs
- GitOps
- Nexus, chartmuseum, monocular
- Environments: Local, staging, production
- IDE Integration
- Jenkins X Pipelines

Jenkins X 1.x. Modes and engines



Static Masters

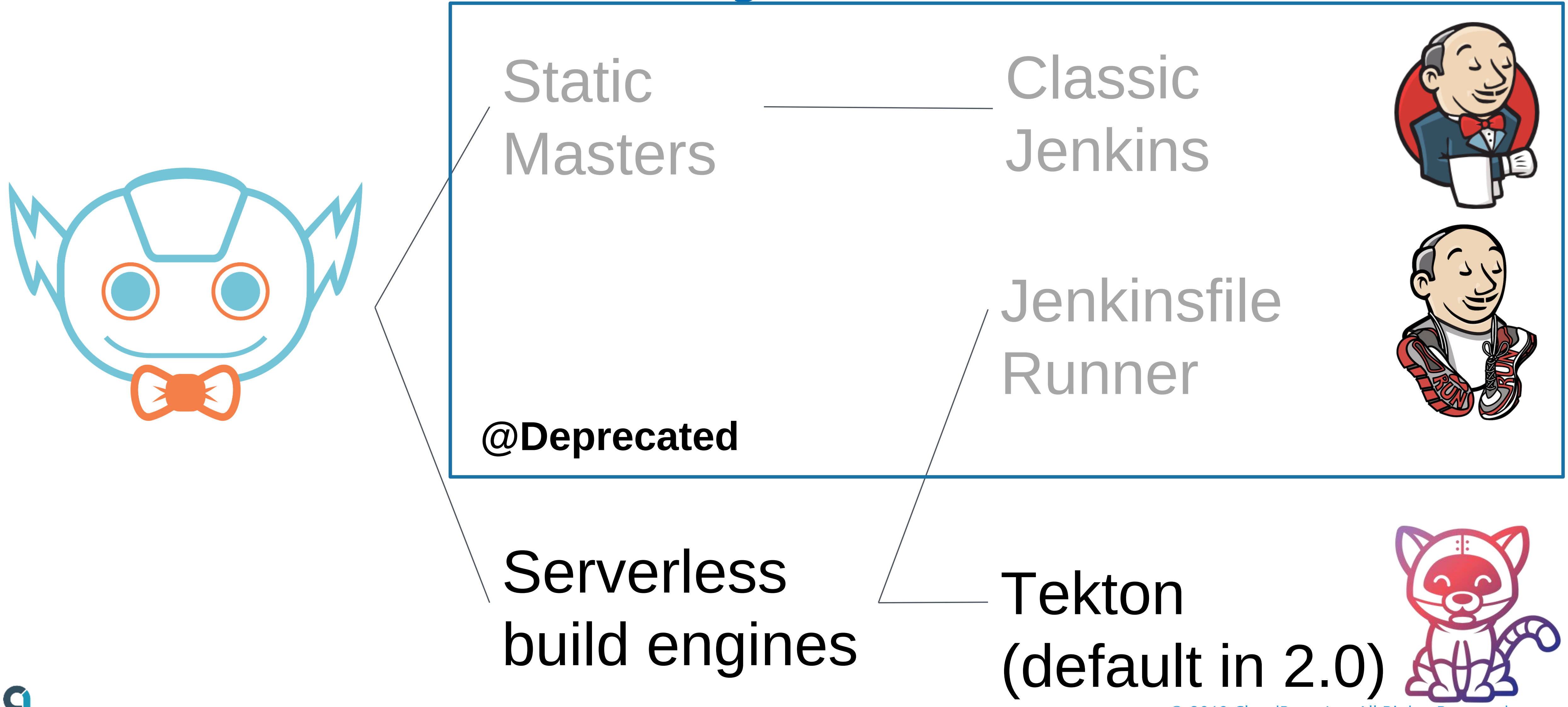
Classic Jenkins

Serverless build engines

Jenkinsfile Runner

Tekton

Jenkins X 2.x. Modes and engines



Static Masters

Classic Jenkins

Jenkinsfile Runner

@Deprecated

Serverless build engines

Tekton (default in 2.0)

cloudbees®

Continuous Delivery Products and Services

Rollout
Feature Flag Management

Core
Unified Software Delivery & Governance

Flow
Adaptive Release Orchestration

CodeShip
CI/CD as a Service

DevOptics
Software Delivery Visibility & Insights

Jenkins

- CloudBees Jenkins Distribution
- CloudBees Jenkins X Distribution

DevOps Excellence

- Accelerator
- Training
- DevOps Consultants

Support

- 24x7 Technical Support
- Assisted Updates
- Customer Success Managers