Jenkinsfile Runner Introduction

Oleg Nenashev (@oleg_nenashev)
CloudBees, Inc.

CDF Interoperability SIG, Jul 09, 2020

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

- Principal SW Engineer, CloudBees
- Jenkins core maintainer & Board Member
- CDF Ambassador

Who is Mr. Jenkins?

- Automation server/framework

- Open-source

- More than 1700 plugins

- Big community

- Commercial support is available

Hudson/Jenkins as you may knew it… in 2011

Batteries are not included

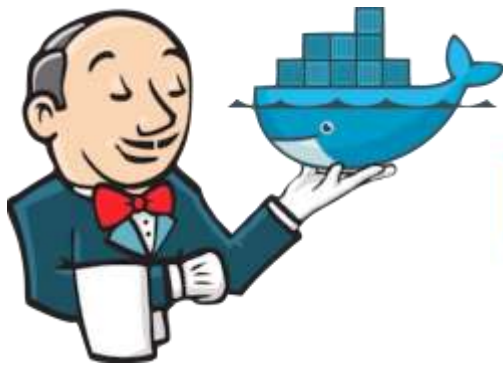Complex configuration

Mega-masters, limited scalability

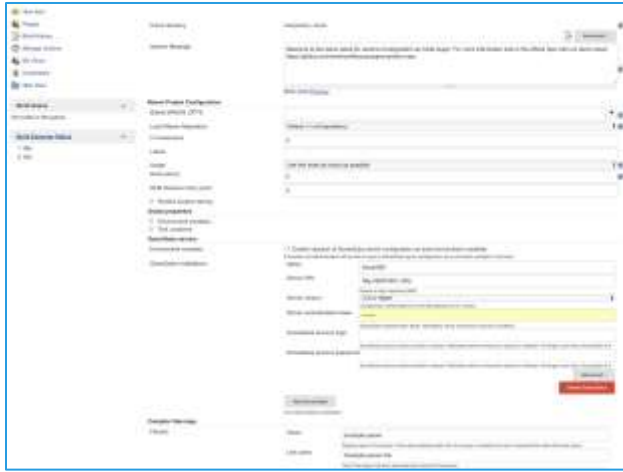Painful upgrades

Maintenance is required

Modern Jenkins

- Pipeline-as-Code
- Configuration-as-Code
- New plugins and integrations
- Modern packaging
- Distributions for public clouds
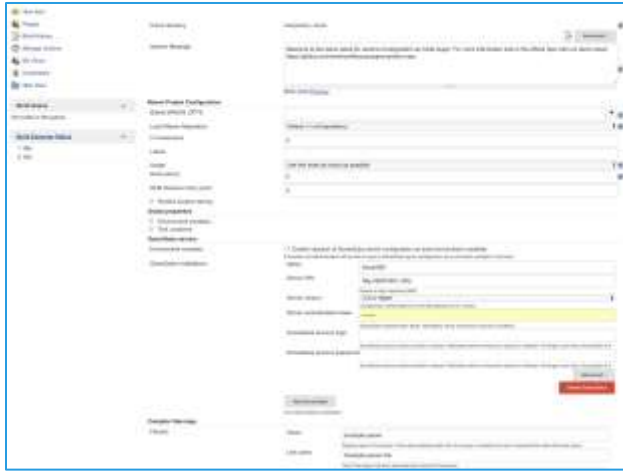
Web UI

Jenkins Master

Permanent agents

On-demand agents
from cloud providers

Master as a SPoF M**a**croservice

Web UI

Permanent agents

On-demand agents
from cloud providers

Jenkins
Master

(┛ ಠ_ಠ)┛ 彡┻━┻

What if Jenkins master was "on-demand"?

Jenkinsfile Runner

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

Started by Kohsuke Kawaguchi in Mar 2018

37 contributors

Status: Beta (1.0-beta-12)

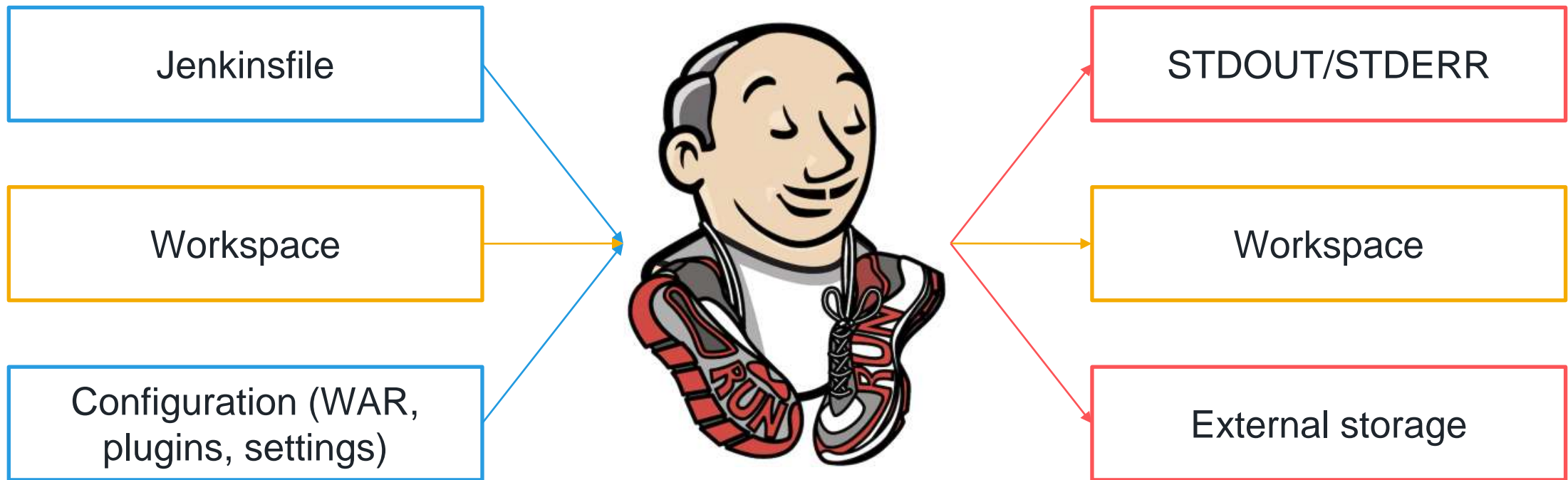| 👁 Unwatch	20	⭐ Unstar	567	🍴 Fork	160

JFR TL;DR

- Runs Pipeline and shuts down

- Runs everywhere

- Fat JAR or a thin JAR, Docker image

- Startup in milliseconds/seconds

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

Jenkinsfile

Workspace

Configuration (WAR, plugins, settings)

STDOUT/STDERR

Workspace

External storage

Usage: jenkinsfile-runner [options] [params]

Options:
-f (--file) FILE   : Path to Jenkinsfile (or directory containing a Jenkinsfile) to run
-w (--jenkins-war) FILE   : Path to the exploded jenkins.war directory
-p (--plugins) FILE   : plugins.txt file or a /plugins installation directory

Advanced:
--runHome FILE   : Path to the empty Jenkins Home directory to use for this run
--runWorkspace FILE   : Path to the workspace to be used within node{} context.
-a (--arg)   : Parameters to be passed to workflow job
-b (--build-number) N   : Build number of the run
-c (--cause) VAL   : Cause of the run
-n (--job-name) VAL   : Name of the job the run belongs to
-ns (--no-sandbox)   : Disable the sandbox environment

Usage in Docker

*docker run --rm *

    *-v $(pwd)/Jenkinsfile:/workspace/Jenkinsfile *

    jenkins4eval/jenkinsfile-runner*

* or another image….

Demo. Our Jenkinsfile

10 lines (8 sloc) | 386 Bytes

Raw | Blame

```
1
2   stage('Read Evergreen YAML') {
3       node {
4           // Discover core version using Pipeline utility steps
5           sh 'wget https://raw.githubusercontent.com/jenkins-infra/evergreen/master/services/essentials.yaml'
6           def essentialsYaml = readYaml(file: "essentials.yaml")
7           echo "Jenkins Evergreen uses the following Core version: ${essentialsYaml.spec.core.version}"
8       }
9   }
```

Live GIF: https://github.com/jenkinsci/jenkinsfile-runner#quick-demo

JFR is **NOT** a CI/CD solution on its own

Web UI

Webhook receivers

Triggers

Log services

Result Storage

Limitations

- Jenkinsfile Runner is not 100% Jenkins

- **Pipeline Execution engine**, not a full CI/CD solution

- No Web UI exposed by default

- No build throttling, cross-master communication, etc.

- Slow agent connection

- No out-of-the-box sidecar containers support

But

Web UI

Webhook receivers

Triggers

Log services

Result Storage
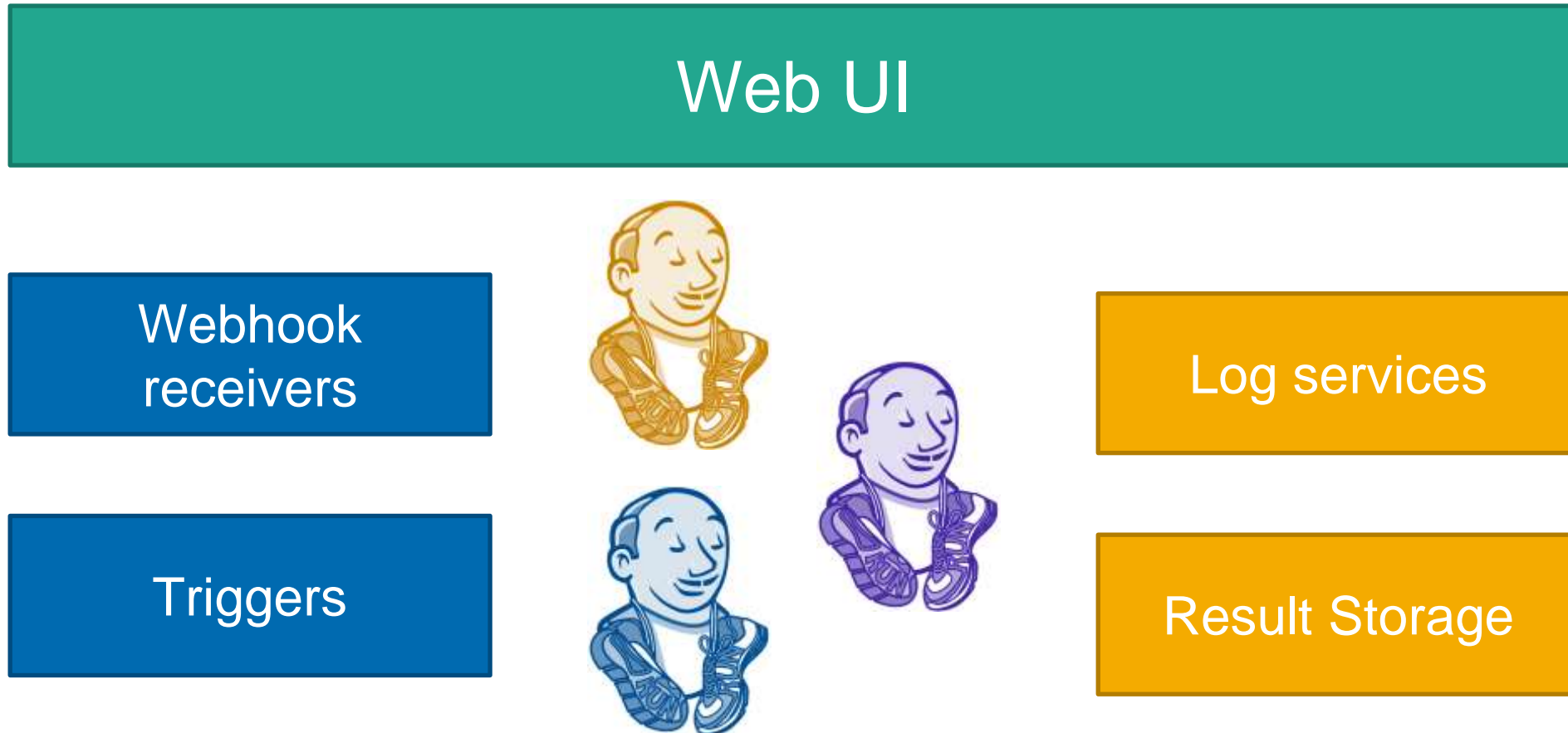
But

Web UI

Webhook receivers

Triggers

Log services

Result Storage

"Packaging"?

Jenkinsfile Runner flavors

Vanilla: Docker/Binary in the official repo

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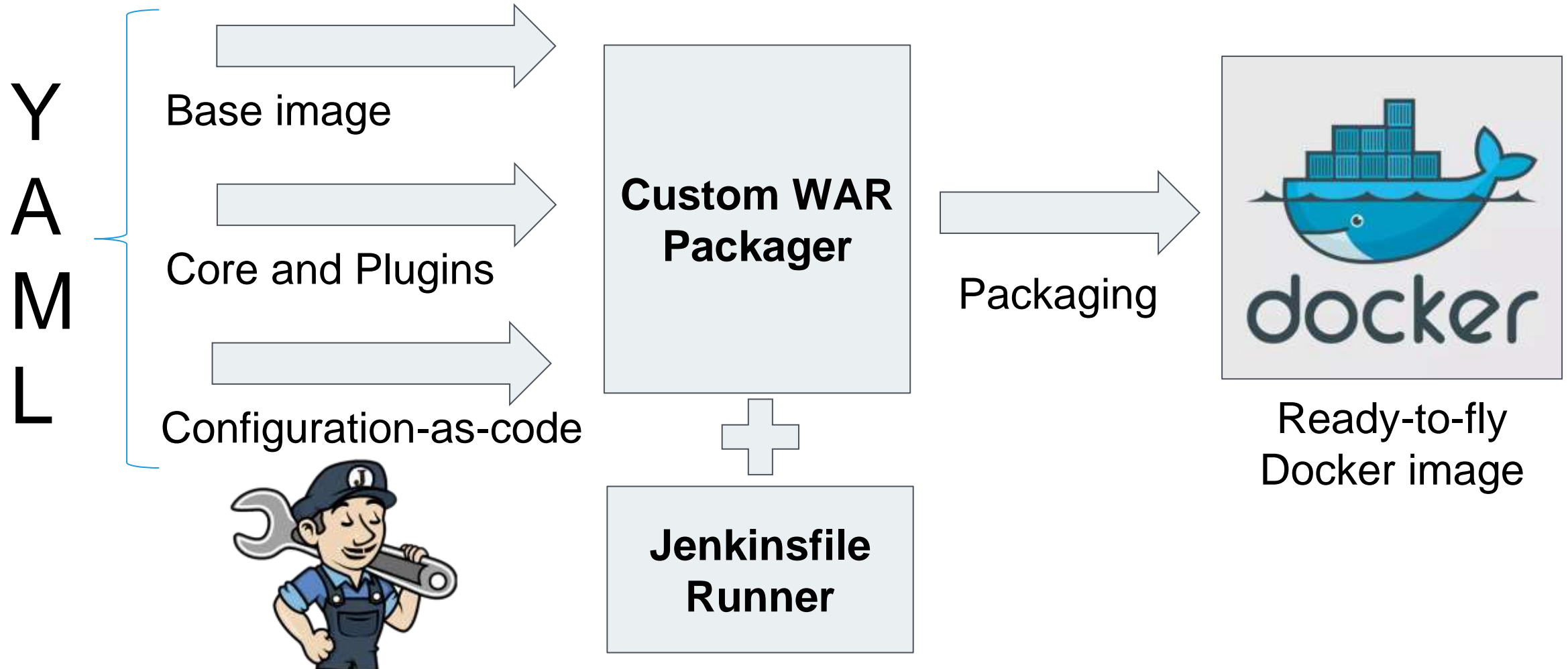- DockerHub: jenkins4eval/jenkinsfile-runner

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

Y
A
M
L

Base image →

Core and Plugins →

Configuration-as-code →

Custom WAR Packager

Packaging →

+

Jenkinsfile Runner

Ready-to-fly Docker image

25

Jenkinsfile Runner. Usage examples

ci.jenkins.io-runner

- jenkinsci/ci.jenkins.io-runner

"FaaS" packaging
- Tekton: https://github.com/SAP/stewardci-jenkinsfilerunner-image
- GitHub Actions: jonico/jenkinsfile-runner-github-actions
- Codeship: oleg-nenashev/codeship-jenkinsfile
- AWS Lambda: carlossg/jenkinsfile-runner-lambda
- Project Fn: carlossg/jenkinsfile-runner-fn

Jenkinsfile Runner. Adoptions

SAP - Steward CI: Jenkinsfile Runner & Tekton

- https://github.com/SAP/stewardci-core
- https://github.com/SAP/stewardci-jenkinsfilerunner-image

Jenkins X - serverless mode in 1.x (EOL, replaced by Tekton and JX Pipeline)

- jenkins-x/jenkins-x-serverless

Gitlab - Jenkinsfile Wrapper

- https://lackastack.gitlab.io/website/posts/gitlabci-jenkinsfile/
- https://gitlab.com/gitlab-org/jfr-container-builder/

Example: CloudBees CodeShip

Running Jenkins Pipelines on CloudBees CodeShip

Jenkinsfile Runner on CloudBees CodeShip

- Jenkins has no popular SaaS offering (now)

- 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

Example. Building a Jenkins plugin in CodeShip

Jenkinsfile

```
3 lines (2 sloc)    102 Bytes

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

codeship-services.yml

```
5 lines (4 sloc)    100 Bytes

1    ci.jenkins.io-runner:
2      image: onenashev/ci.jenkins.io-runner:latest
3      volumes:
4        - .:/workspace
```

codeship-steps.yml

```
4 lines (3 sloc)    77 Bytes

1    - name: Jenkinsfile
2      service: ci.jenkins.io-runner
3      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
```

② Help

CodeShip. Build history

Ping me to talk about...

- Jenkinsfile Runner on Tekton

- Jenkinsfile Runner as a GitHub Action

- Jenkinsfile Runner with Jenkins X 1.x

- Testing Jenkins pipelines with Jenkinsfile Runner

Takeaways

- Jenkinsfile-as-a-function is real

- You can run your Jenkins Pipelines everywhere

- There are tools for that

- Consider Tekton / Jenkins X if applicable (K8s/GitOps)

Next steps

- Waiting for user feedback

- TBD - Jenkinsfile Runner 1.0

- Plan: Release Candidate this summer

Roadmap: Jenkinsfile Runner

Near-term:

- JFR Docker images for developer tools (Maven, Gradle, Python, etc.)
- Startup performance
- K8s-native packaging tools
- OOTB Support for Tekton

Future:

- Native images (Quarkus/GraalVM)
- Support for lightweight sidecar containers

It is a great time to contribute!

- [jenkinsci/jenkinsfile-runner](#)
- Try it out in your projects!
- CONTRIBUTING.md in the repository

QUESTIONS?

Contacts:

✉ E-mail: onenashev@cloudbees.com

⊙ GitHub: oleg-nenashev

🐦 Twitter: @oleg_nenashev