

DOMINO AND CI ANYWHERE

CI Tools For Build Orchestration



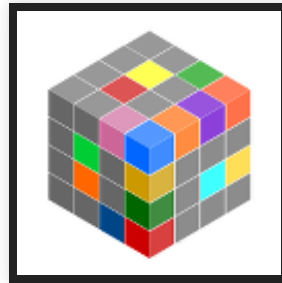
ME

- Github/Twitter : @solvingj
- Email: jerrywiltse@gmail.com



BINCRAFTERS

Open-Source Devops Team




FIND US ONLINE

- <https://opencollective.com/bincrafters>
- <https://bincrafters.github.io>
- <https://twitter.com/bincrafters>
- <https://github.com/bincrafters>
- <https://www.youtube.com/watch?v=v2QiNTvco5E>



PROJECTS ON GITHUB



Bincrafters

A team for OSS developers to collaborate on binary packaging of libraries.

<http://bincrafters.github.io> bincrafters@gmail.com

Repositories 387

People 21


Teams 2

Projects 0



Settings



PACKAGES ON BINTRAY





bincrafters (Bincrafters)


 Contact  Join


Owned Repositories

All (8) Public (8) Private (0) OSS (8) Premium (0)

 public-conan (Conan) 296 packages

 public-deb (Debian) 5 packages

 public-rpm (RPM) 5 packages

 public-choco (NuGet) 5 packages

General

Location

Email bincrafters@gmail.com

Website https://bincrafters.github.io/

Twitter bincrafters

GitHub bincrafters



CI ANYWHERE - CI ABSTRACTION LAYER

- Generalize common CI features
- Decouple selected logic from CI implementation
- Python script framework

CI ANYWHERE - PRIOR ART

- github.com/solvingj/go-github-release-test
- Conan Package Tools
 - Run python build.py anywhere, CI or Local
 - Incubating and Evolving for 2+ years
 - Features for both enterprise and OSS CI's
 - Primitive yet brilliant and invaluable

CI ANYWHERE - MOTIVATION

- Any non-trivial scripting with YML is awful
- OS-Specific scripting languages have issues
- CI's have 90% similar features
- Reproducing a CI job locally is invaluable
- and usually impossible
- Lots of redundant hand-rolled cleanup logic
- Scripts typically lack unit testing

CI ANYWHERE - MOTIVATION

- Choosing and configuring a new CI is painful
- Each CI's is good at some things, bad at others
- Most struggle to support new technology quickly
- Proprietary plugins needed for basic things
- Adding/changing CI becoming more common

CI ANYWHERE - CONAN PACKAGE TOOLS

- <https://github.com/conan-io/conan-package-tools>
- Major part of inspiration
- Perfect example of a generalized implementation
- Standardizes universal boilerplate tasks

CI ANYWHERE - CONAN PACKAGE TOOLS

- Example boilerplate tasks - Build with Docker:
 - Optionally update container with pull
 - Optionally update code in container before run
 - Mounting current working directory
 - sudo both inside and outside docker commands
 - Docker entry script lets user do anything pre-build

CI ANYWHERE - CONAN PACKAGE TOOLS

- Intelligently layers defaults, env vars, script params
- Artifact publishing and credential pre-verification
- Git branch detection and commit parsing
- Shows importance of language-specific features

CI ANYWHERE - CONAN PACKAGE TOOLS

- More accessible extensibility model (code sharing)
- Simply create and share python modules via PIP
- Bincrafters package tools good example
- Compare with creating a custom plugin for CI XYZ
- Jenkins just added "Shared Libraries" feature

TRAVIS CI EXAMPLE

```
before_install: python ci/travis_pipeline.py -step_name before  
install: python ci/travis_pipeline.py -step_name install  
script: python ci/travis_pipeline.py -step_name script  
after_success: python ci/travis_pipeline.py -step_name after_s
```

APPVEYOR EXAMPLE

```
install: python ci/appveyor_pipeline.py -step_name install  
build_script: python ci/appveyor_pipeline.py -step_name build_  
after_build: python ci/appveyor_pipeline.py -step_name after_b  
deploy_script: python ci/appveyor_pipeline.py -step_name deplo
```


CI ANYWHERE - FEATURES

- Commit parsing = CI Command/Control
- "Publish" helpers for all package types
- Generic Credential Management
- Generic Environment Variable Definitions

PROJECT DOMINO

A CI-agnostic Project-Level Dependency System



ABSTRACT CI DEPENDENCIES

- Github is a DAG
- Your Enterprise Codebase is a DAG
- Upstream project build trigger downstream CI
- Like Libraries.io - but trigger jobs instead of email

PRIOR ART

- Libraries.io - Simple, good indicator of demand
- Numerous other recent competitors to Libraries.io
- Fedora - <https://release-monitoring.org/projects>
- Modern CI Systems trigger from "upstream builds"
- All proprietary implementations of a simple concept

DOMINO MOTIVATION

- C/C++ has unique property of ABI compatibility
- Even if API changes, downstream might break
- Bincrafters : 200+ Packages : Interdependent
- Realized our needs are universalish among PM's
 - Apt/Yum/Brew/Conda/Vcpkg/Choco/Onepkg/etc
- Realized it's needed in enterprise as much as OSS
- Great candidate to generalize and open-source

DOMINO USE-CASES

- Downstream dependency libraries
- Downstream package manager repos
- Flexible build policies for downstream consumers
- A CI agnostic dependency definition layer
- For Enterprise
 - Build based on OSS and internal deps
- For OSS
 - Single source of truth for multiple CI:
 - Travis/Appveyor/CircleCI/Etc



TAKEAWAYS

- Bincrafters Looking for New Members + Sponsors
- ... especially if you have C/C++ code
- Also looking for ideas and feedback
- Reach out to me to get involved

THANK YOU

- Github/Twitter : @solvingj
- Email: jerrywiltse@gmail.com

