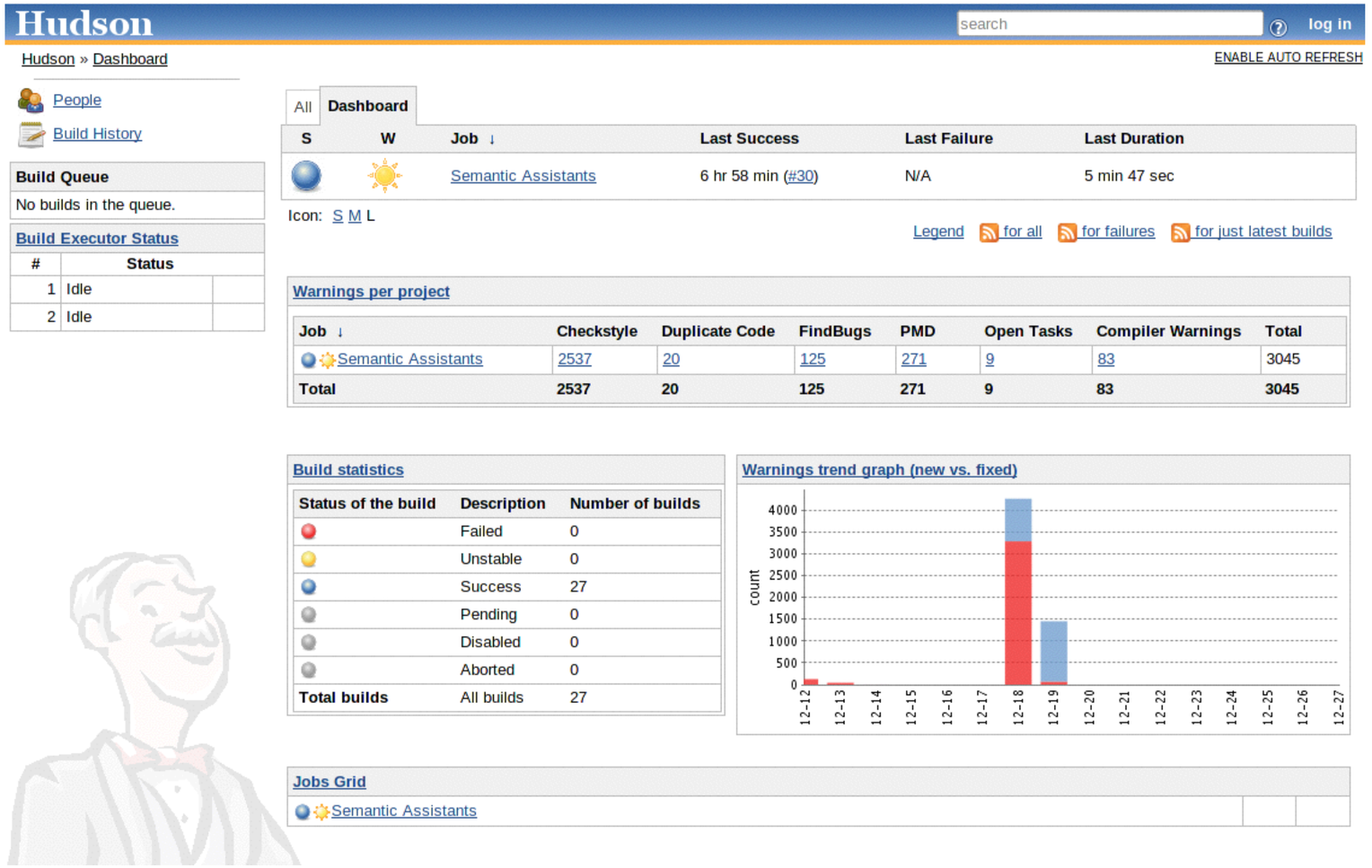
Jenkins CI for MacDevOps

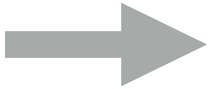
Tim Sutton
Concordia University, Faculty of Fine Arts
Montreal

A story

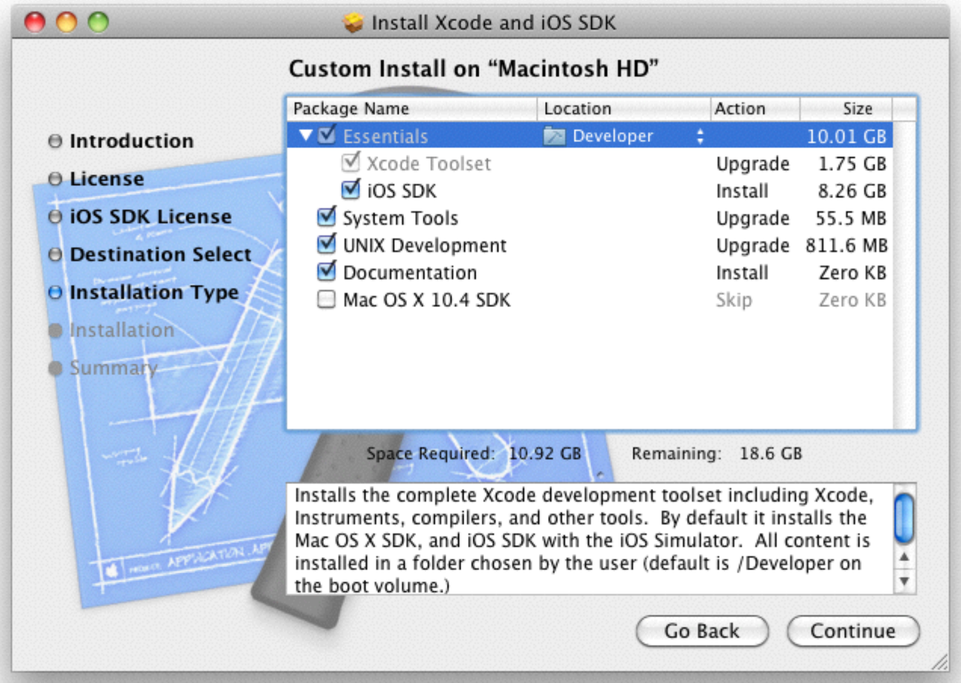
```
svn update
./configure
make
```



Hudson

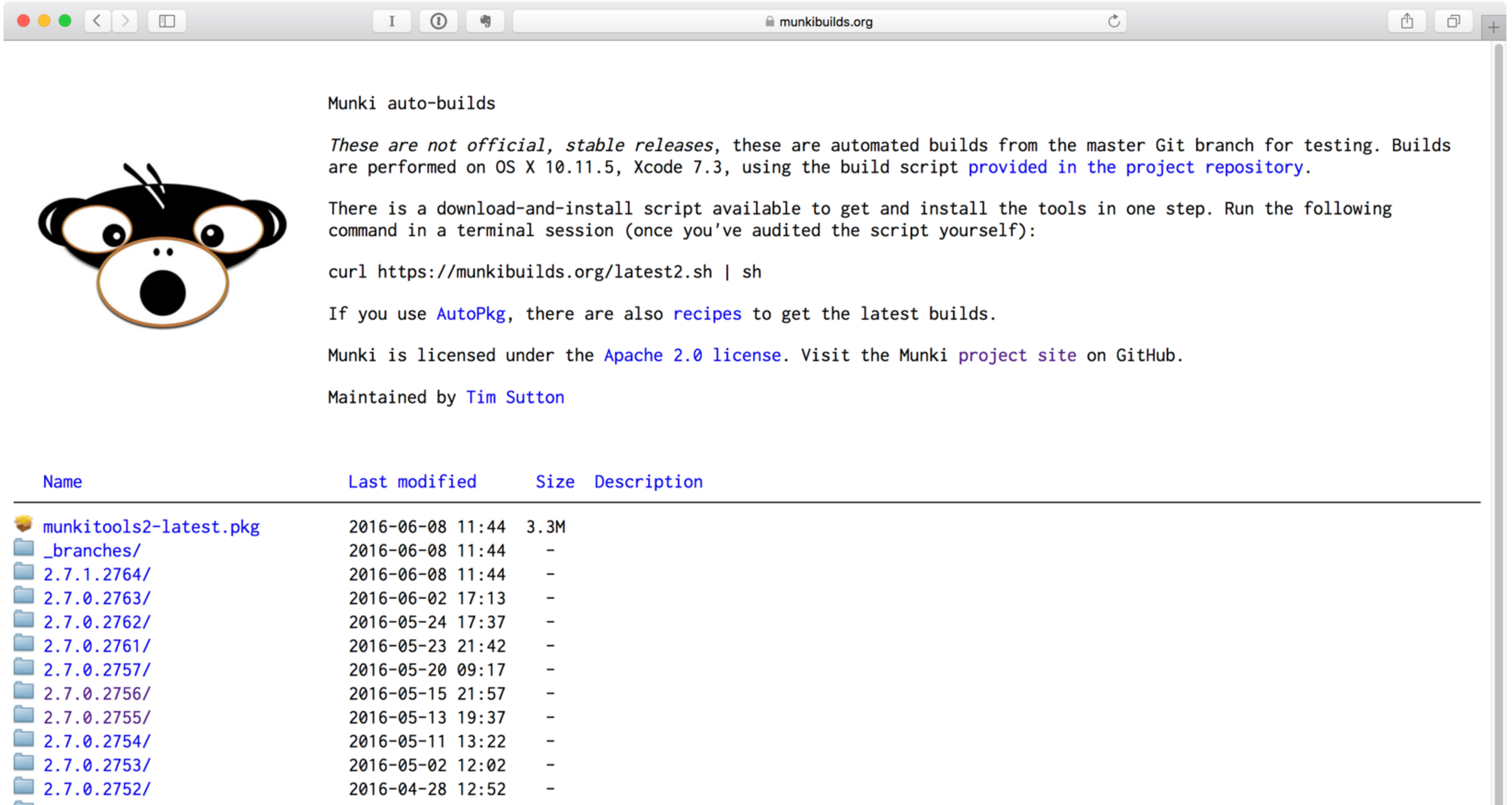


Jenkins




```
git pull
cd code/tools
./make_munki_mpkg.sh
```

munkibuilds.org



Continuous Integration

Continuous Integration

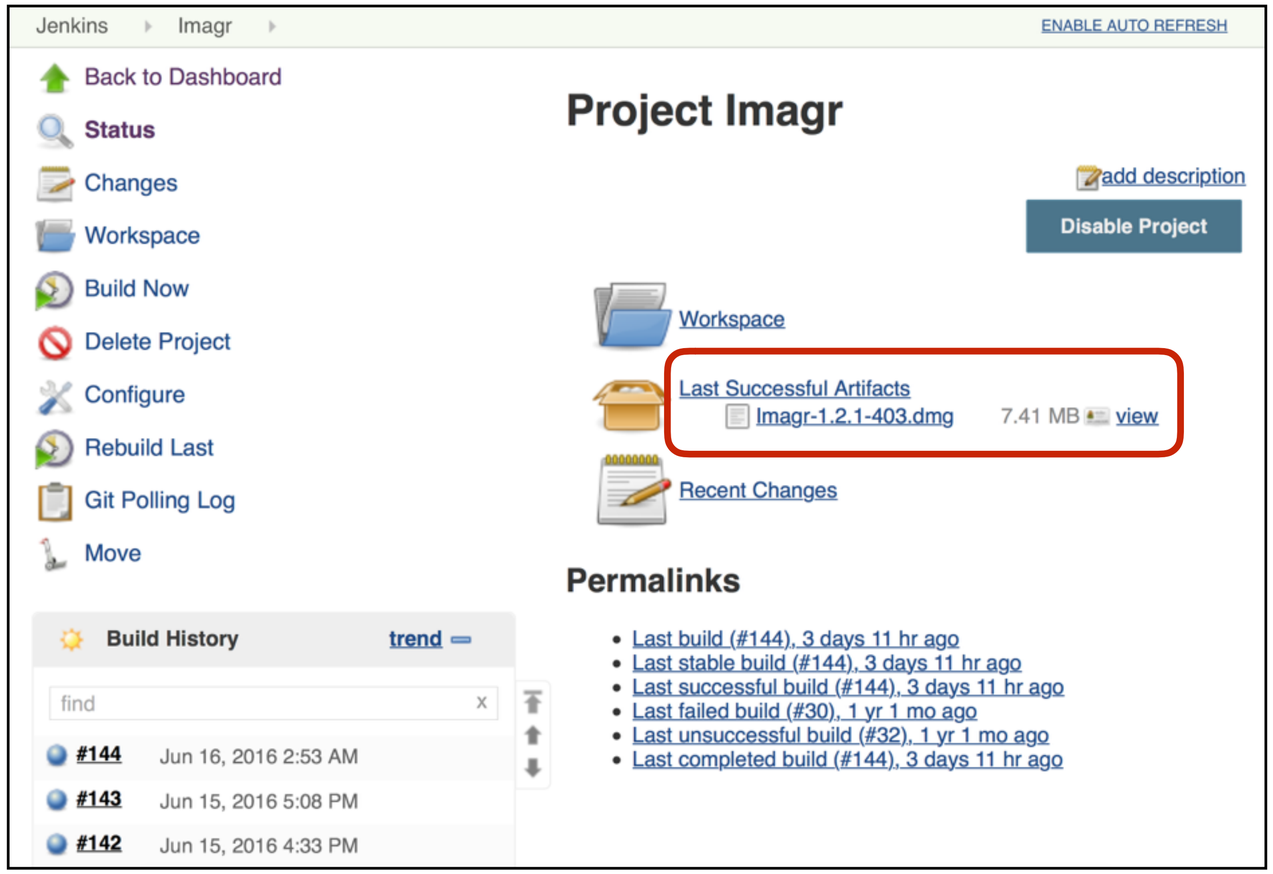
- Software Development
  - Run unit, integration, performance tests
  - Fix code style, handle compiler warnings
  - Define a single trusted build environment (no “works on my machine”)
  - Automate merging or pushing code if conditions pass
  - Orchestrate complex build tasks, shared dependencies

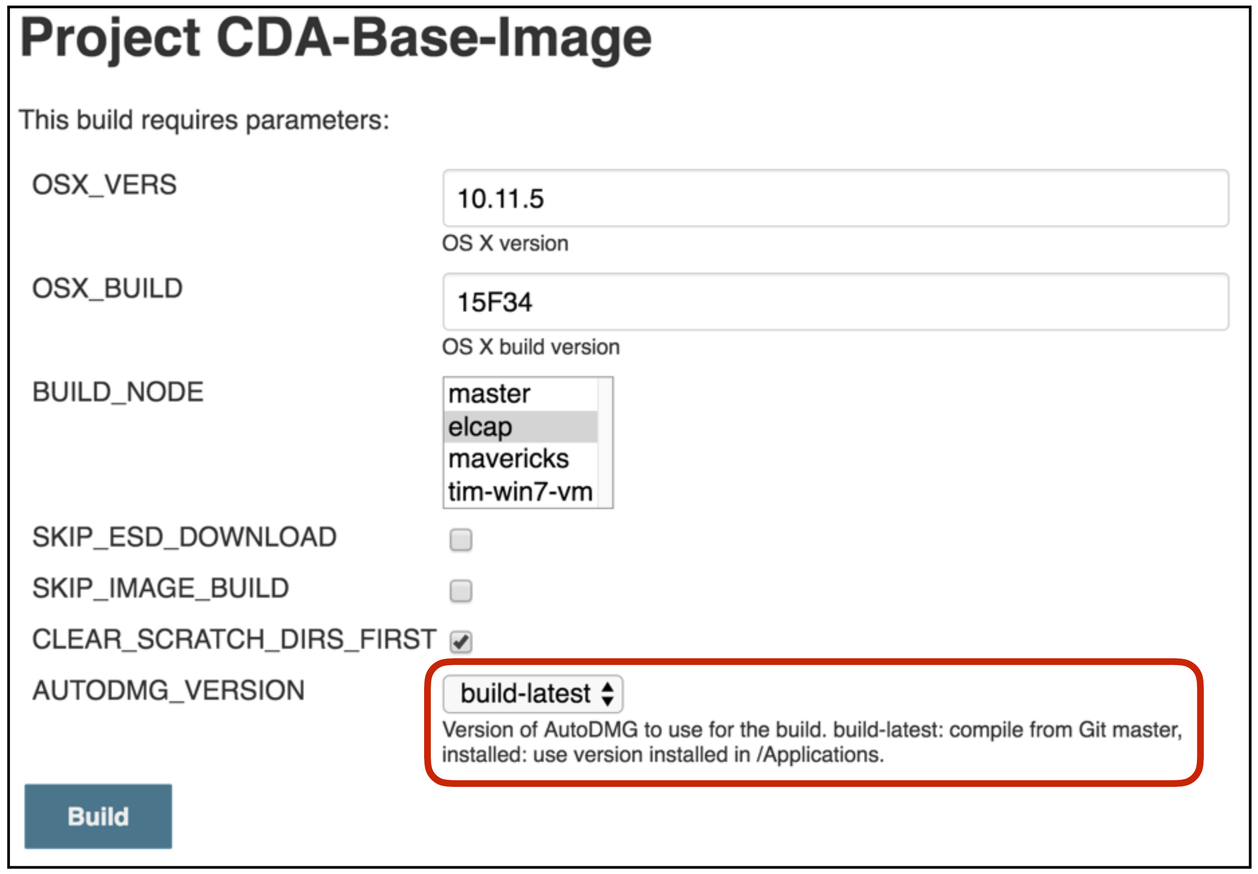
Continuous Integration

- Operations
  - Run tests on configuration management tooling
  - A “build” == any long-running task ending in a product, and which needs a specific environment set up
  - Move “magic” you’ve internalized into shared code
  - Delegate access to others, remove bus factor

For example

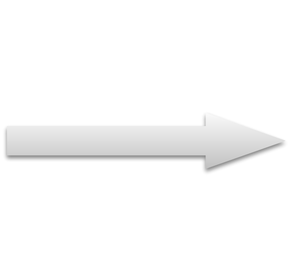
Auto-build your MacAdmin tools



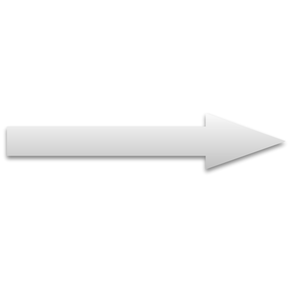


Single input, multiple builds

Munki



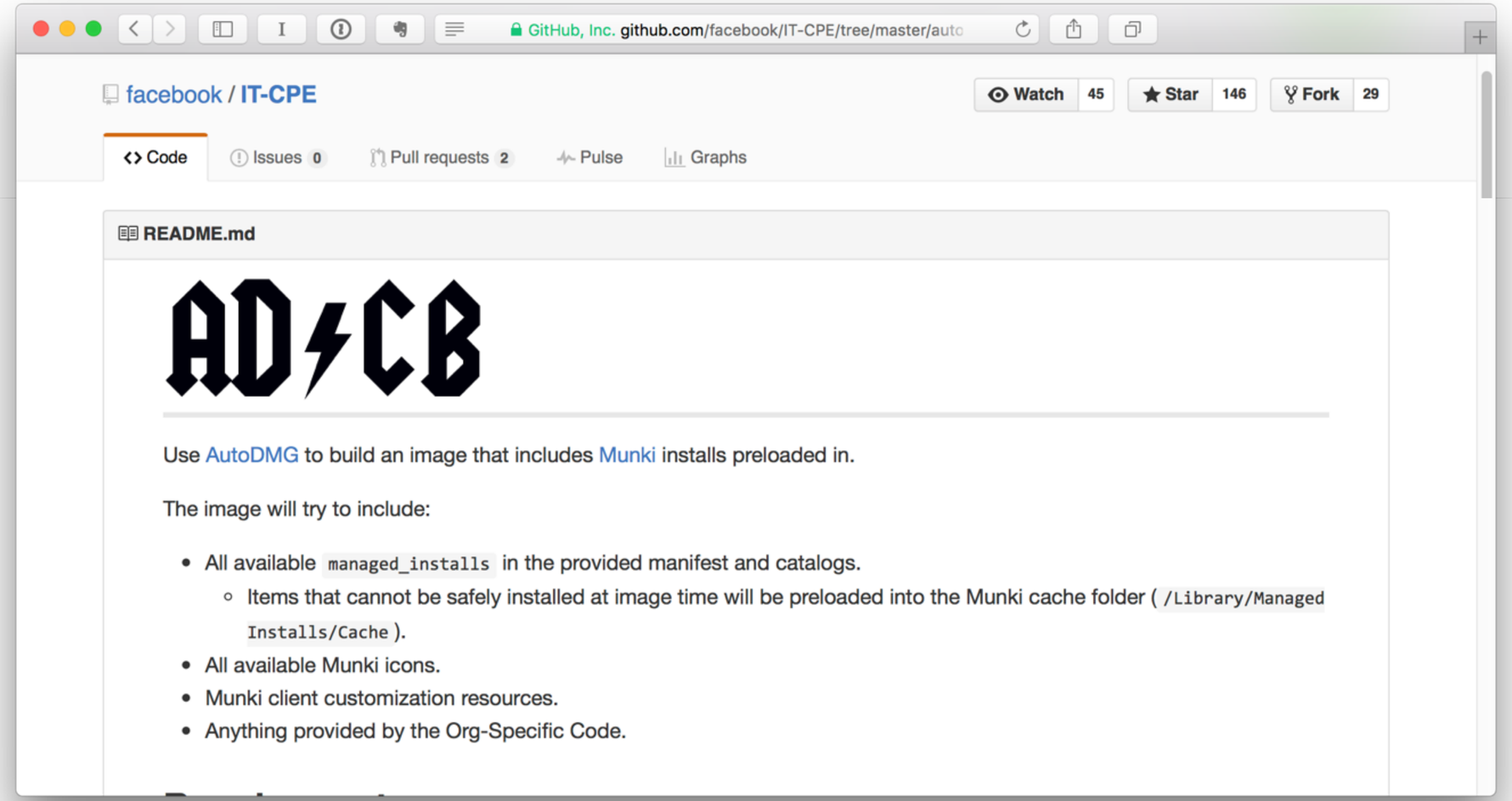
Imagr



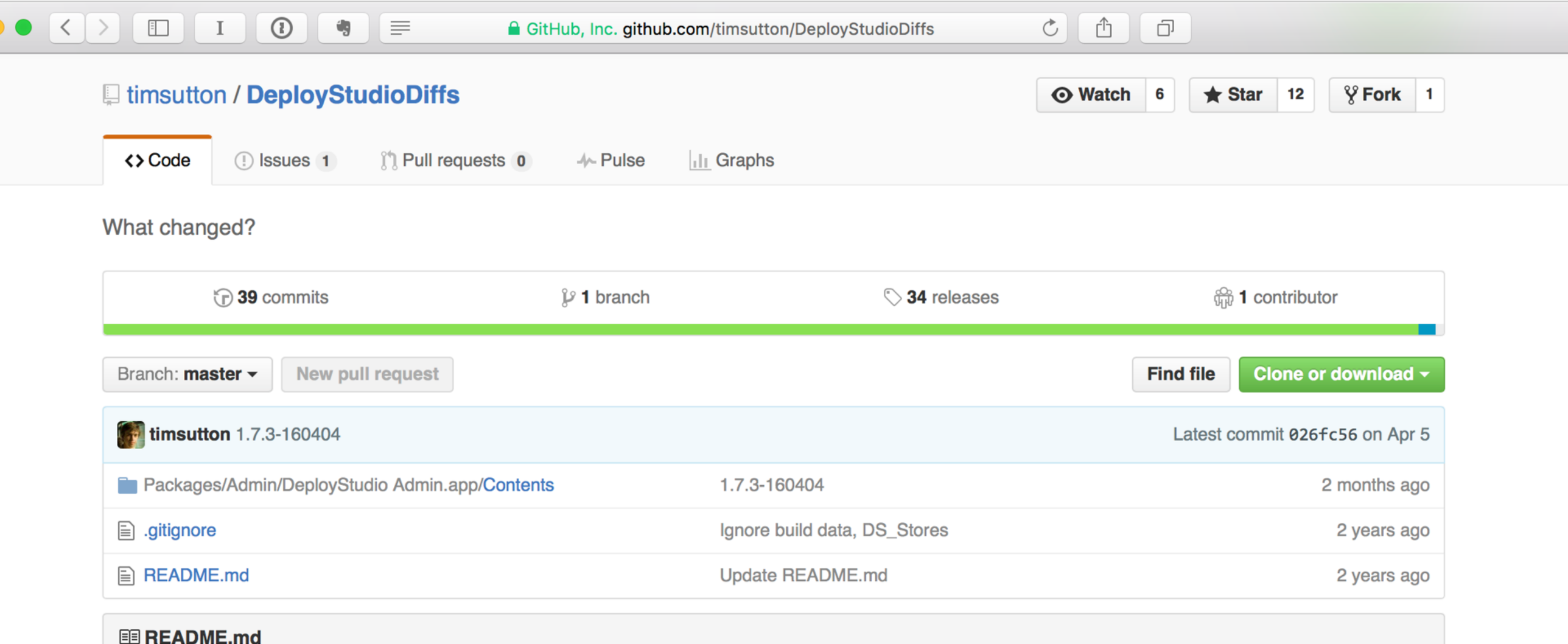
NetBoot /
BSDPy

Single input, multiple builds

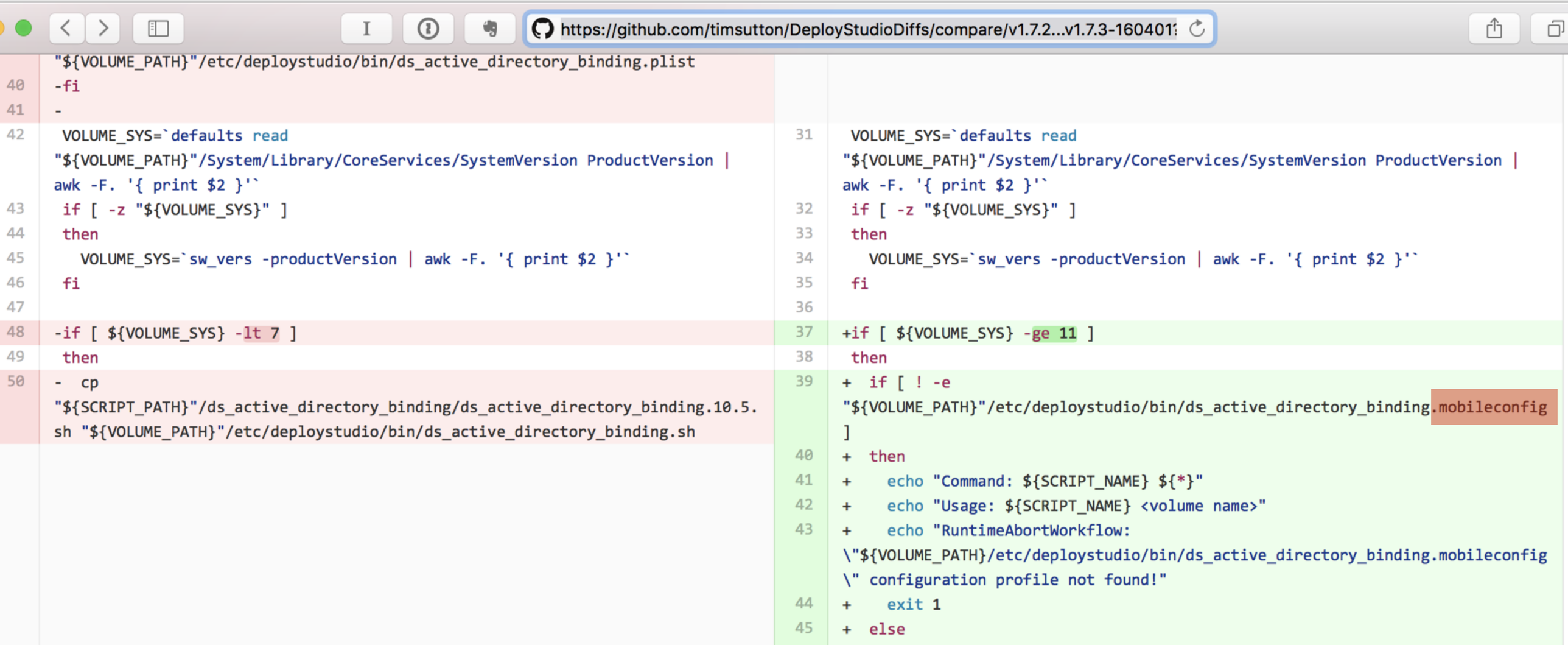
AutoDMG Cache Builder (Nick McSpadden)



Monitor external code



Monitor external code



Monitor external code

http://swcdn.apple.com/content/downloads/47/02/031-63096/dejq3zu7l8qejx7r8xckqzuugus4qfinke/XProtectPlistConfigData.pkg

```
<key>Version</key>
<real>2079</real>
<key>PlugInBlacklist</key>
<dict>
  <key>10</key>
  <dict>
    <key>com.microsoft.SilverlightPlugin</key>
    <dict>
      <key>MinimumPlugInBundleVersion</key>
      <string>5.1.41212.0</string>
      <key>PlugInUpdateAvailable</key>
      <true/>
    </dict>
```

```
<dict>
  <key>MatchFile</key>
  <dict>
    <key>NSURLNameKey</key>
    <string>CleanMyMac</string>
    <key>NSURLTypeIdentifierKey</key>
    <string>public.unix-executable</string>
  </dict>
  <key>MatchType</key>
  <string>Match</string>
  <key>Identity</key>
  <data>8aMuU0OdOtyWejtH+Qcd5sEPzk4=</data>
```

Monitor external code

http://swcdn.apple.com/content/downloads/19/38/031-61264/fn3uieyv462h1xqriahvvn0ri6niv60vb0/MRTConfigData.pkg

```
$ strings MRT

…
OSX.XcodeGhost.A
OSX.Genieo.A
/Applications/InstallMac
/Applications/Genieo.app
~/Library/Safari/Extensions/Omnibar.safariextz
~/Library/Safari/Extensions/GoldenBoy.safariextz
…
```

Iteration

Iteration

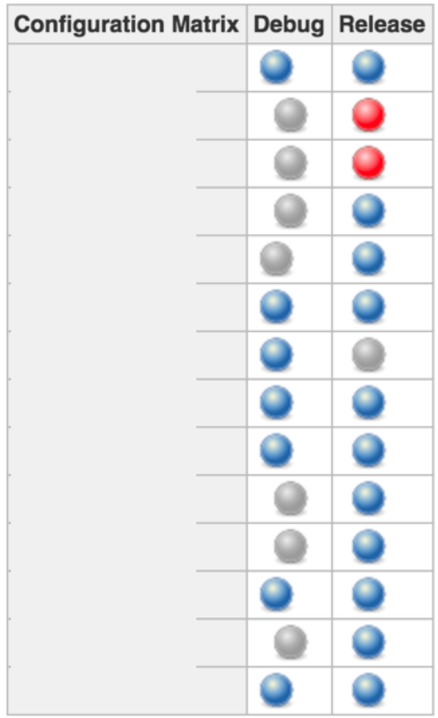
- Start with a manual process that could be fully automated
  - Whether a big or small job, encapsulate it into a single command
- Now make the required environment reproducible
  - Add nodes (or “runners”) to isolate this environment as required
- Isolate the inputs and outputs and turn them into variables
  - Makes the build parameterized, allows for more flexible processing

Iteration

- Don’t let perfect be the enemy of good
- Fail fast (`set -eux`, exit non-zero on unexpected results)
- Entropy is inevitable, exercise workflows regularly
- Job configurations get dense, so keep build steps short..
  - ..and as much in SCM as possible

Patterns and tools

Multiple configurations



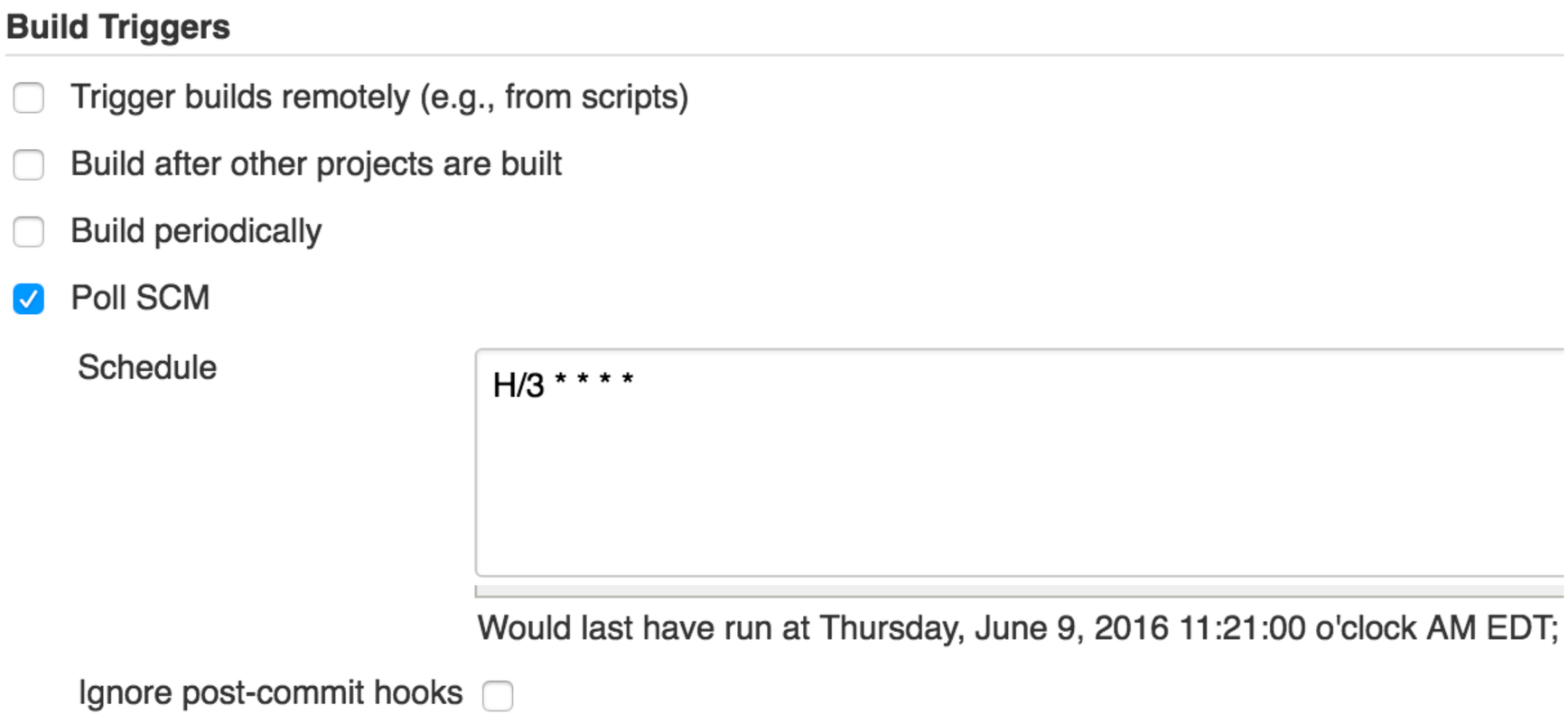
Project osqueryPullRequestBuild

Configurations

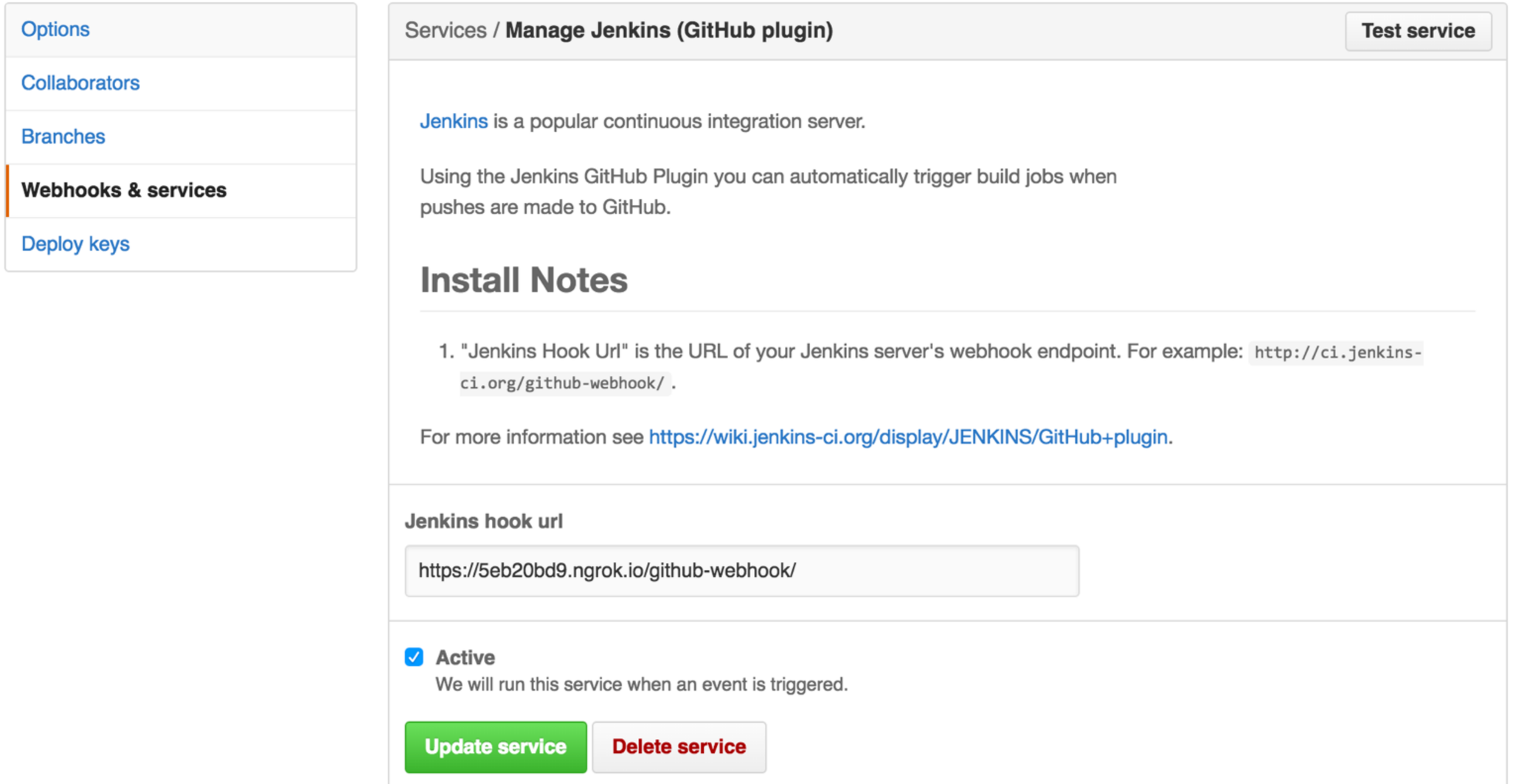
centos6 centos7 osx11 ubuntu12 ubuntu14 ubuntu16

(https://jenkins.osquery.io)

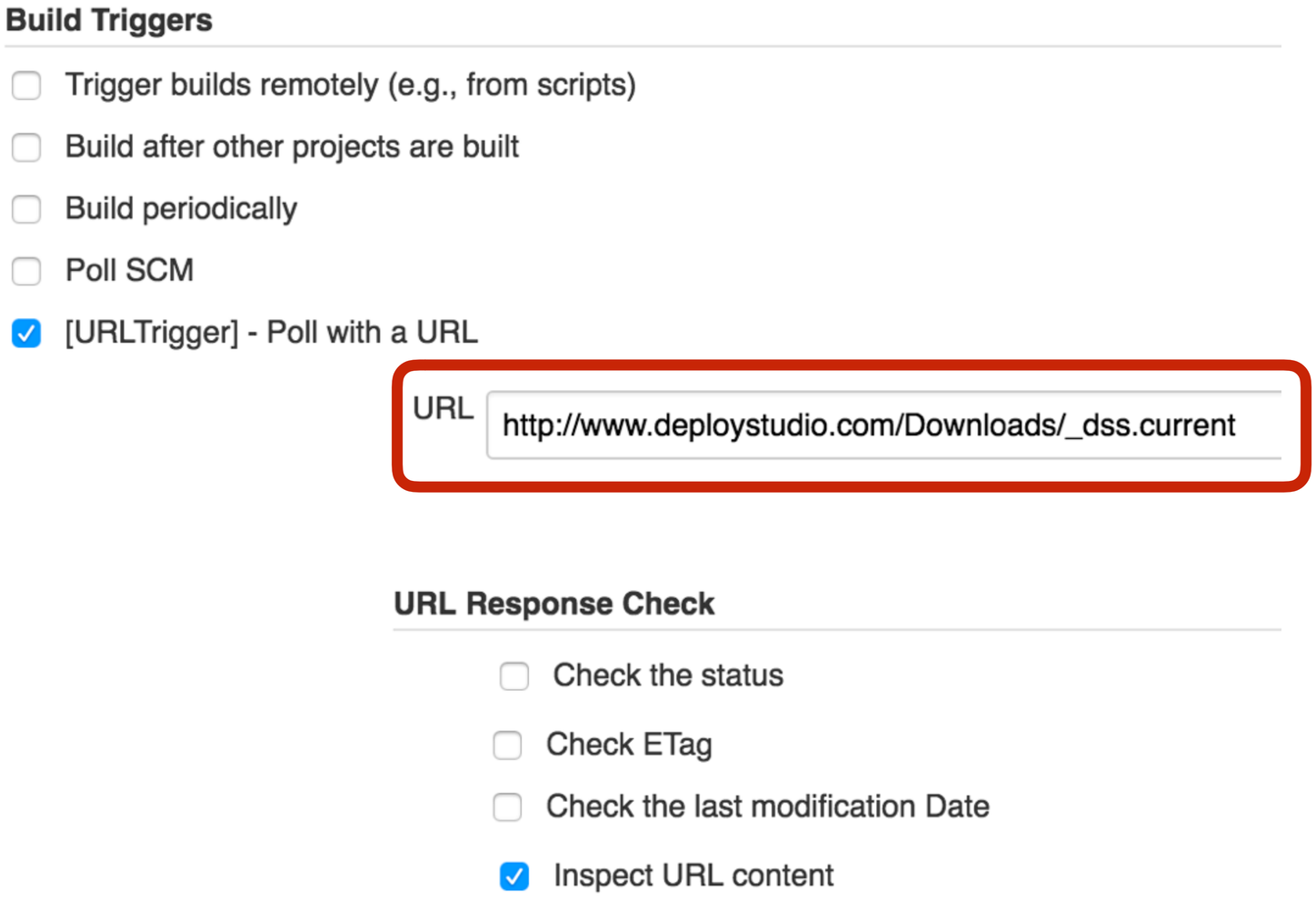
Triggers (SCM polling)



Triggers (GitHub webhook on ‘push’ event)



Triggers (HTTP response)



Triggers (GitHub issue/PR bot)

UniqMartin commented 15 hours ago — Homebrew member

@BrewTestBot test this please

UniqMartin commented 15 hours ago — Homebrew member

From the log of the failed CI build:

```
==> brew install --build-bottle --verbose qt5
[…snip…]
==> ./configure -verbose -prefix /usr/local/Cellar/qt5/5.6.0 -release -separate-debug-info -opensou
[…snip…]
ERROR: -separate-debug-info needs -debug, -debug-and-release, or -force-debug-info
```

I have to admit that I have trouble understanding this given this checked item from the top comment:

> ☑ Have you built your formula locally prior to submission with `brew install <formula>` (where

Slave nodes

- Master -> Slave via SSH
- Slave -> Master via `java -jar slave.jar`
  - (either way, set `-Djava.awt.headless=true`)
  - (either way, slave nodes must be managed statically)
- Jenkins Swarm plugin
- Docker, EC2 dynamic slaves plugins

Dynamic environment variables

```
# Store stuff in properties
cat > build.properties << EOF
"PKGVERSION=$PKGVERSION"
"APPNAME=$APPNAME"
"BUILDDIR=$BUILDDIR"
"BUILDLOG=$BUILDLOG"
"BUILT_PKG_NAME"
"SSH_TARGET_DIR=SSH_TARGET_DIR"
```

See the list of available environment variables

Inject environment variables

Properties File Path: build.properties

Properties Content:

EnvInject Plugin

Send build artifacts over SSH

SSH Publishers

SSH Server

Name: munkibuilds.org

Transfers

Transfer Set

Source files: build/*.pkg, build/*.log, build/*.txt, build/*.html, build/MD5

Remove prefix: build

Remote directory: $SSH_TARGET_DIR

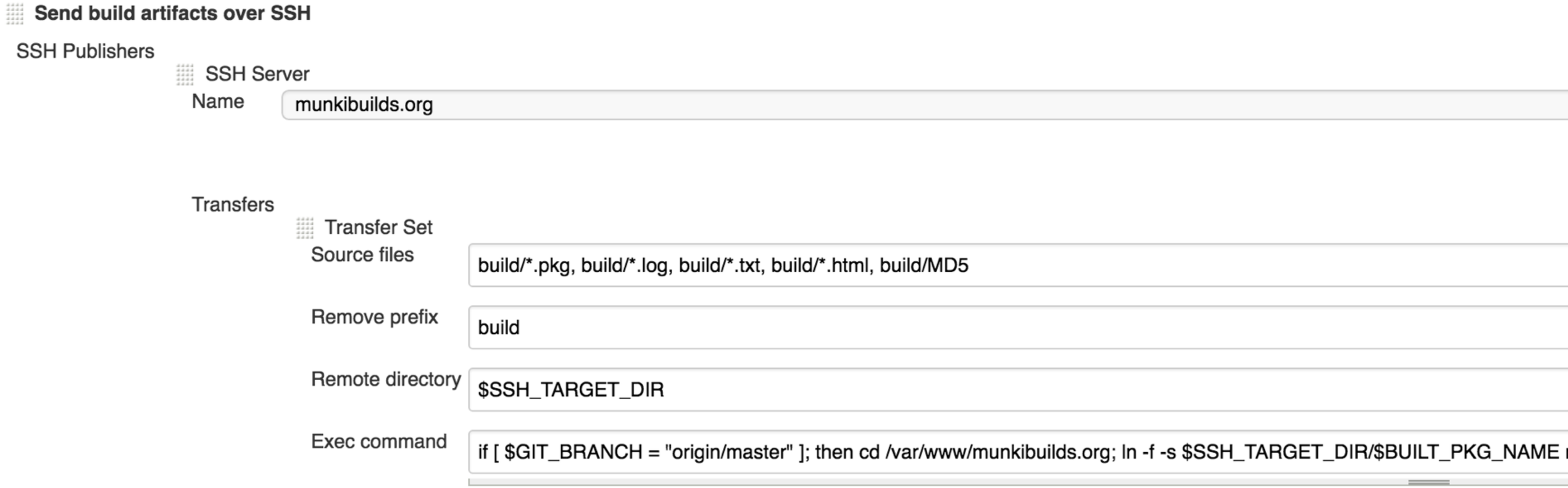
Exec command: if [$GIT_BRANCH = "origin/master"]; then cd /var/www/munkibuilds.org; ln -f

Either Source files, Exec command or both must be supplied

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variab

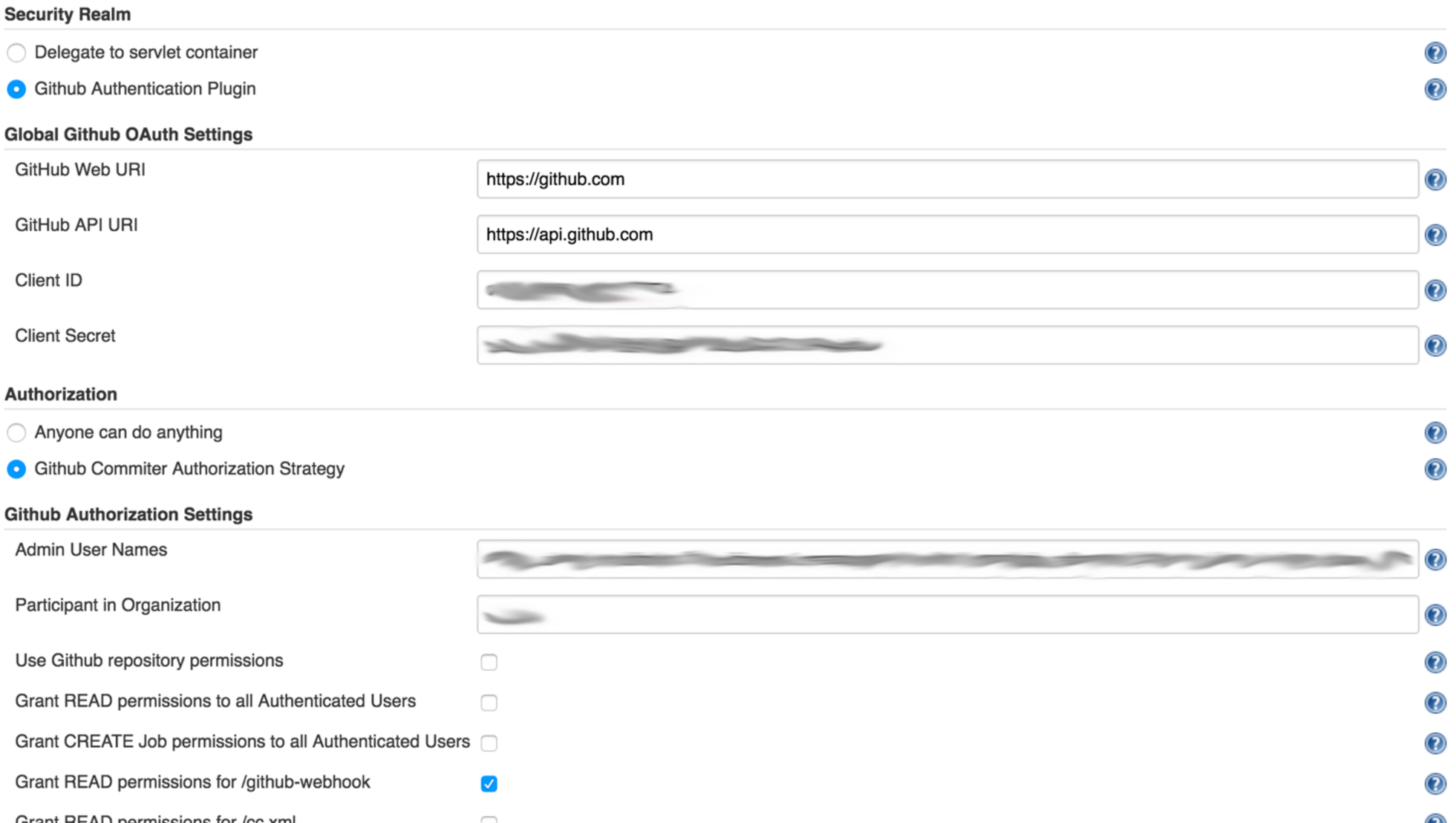
SSH Publisher

Publishing artifacts (SSH)



Publishing artifacts (CIFS)

Auth (GitHub OAuth)



Auth (Active Directory)

Authorization

- ○ Anyone can do anything
- ○ Legacy mode
- ○ Logged-in users can do anything
- ◉ Matrix-based security

User/group	Overall						Credentials					Slave						Job								
	Administer	ConfigureUpdateCenter	Read	RunScripts	UploadPlugins	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Worksp
admins	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☑	☐
tsutton	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑
Anonymous	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Credentials

- Managed centrally, no need to manage on slave nodes
  - SSH keys
  - Service tokens
  - User authentication
  - etc.

Plugins

- .NET Development
- Android Development
- Artifact Uploaders
- Authentication and User Management
- Build Notifiers
- Build Parameters
- Build Reports
- Build Tools
- Build Triggers
- Build Wrappers
- Cloud Providers
- Cluster Management and Distributed Build
- Command Line Interface
- Database
- Deployment
- DevOps
- External Site/Tool Integrations
- Groovy-related
- iOS Development
- Library plugins (for use by other plugins)
- List view columns
- Maven
- Misc ()
- Misc (ca-apm)
- Misc (cmp)
- Misc (file)
- Misc (pipeline)
- Misc (plugin-test)
- Misc (spot)
- Misc (spotinst)
- Misc (textfile)
- Miscellaneous
- Other Post-Build Actions
- Page Decorators
- Python Development
- Ruby Development
- RunConditions for use by the Run Condition plugin
- Scala Development
- Security
- Slave Launchers and Controllers
- Source Code Management
- Source Code Management related
- Testing
- Uncategorized
- User Interface
- Views

Demo

Management, automation

Job DSL Plugin

```
streamFileFromWorkspace("${RECIPE_LIST_FILE}").eachLine {
  def recipeName = it
  job {
    name "${recipeName}"
    logRotator(30, -1, -1, -1)
    label('macpro')

    multiscm {
      git('git://github.com/autopkg/autopkg.git', 'master')
    }

    triggers {
      cron('H H(0-7),H(8-15),H(16-23) * * *')
    }

    steps {
      shell("echo ${recipeName} > recipe.txt")
      shell(readFileFromWorkspace('autopkg-ci/steps/autopkg_run.py'))
    }

   configure { project ->
      def setter = project / publishers / 'hudson.plugins.descriptionsetter.DescriptionSetterPublisher'
```

Job DSL Plugin

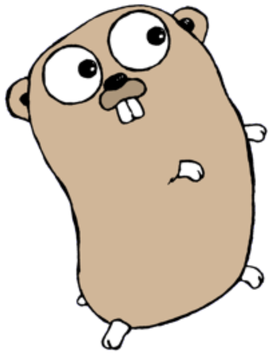
All | autopkg-dsl | autopkg-recipes-all | **autopkg-recipes-munki** | autopkg-recipes-pkg | mu

S	W	Name ↓	Build description
		autopkg-Adium.munki	1.5.7
		autopkg-AdobeAcrobatPro9Update.munki	9.5.5
		autopkg-AdobeAcrobatProXUpdate.munki	10.1.7
		autopkg-AdobeAir.munki	3.8.0.1430
		autopkg-AdobeFlashPlayer.munki	11.8.800.168
		autopkg-AdobeFlashPlayerExtractPackage.munki	11.8.800.94
		autopkg-AdobeFlashPlayerRepackage.munki	11.8.800.168
		autopkg-AdobeReader.munki	11.0.04
		autopkg-BBEdit.munki	10.5.5
		autopkg-Coda2.munki	2.0.11
		autopkg-Cyberduck.munki	4.3.1
		autopkg-Dropbox.munki	2.0.26
		autopkg-Evernote.munki	5.2.1
		autopkg-Facter.munki	1.7.3
		autopkg-Firefox.munki	23.0.1
		autopkg-Flip4Mac-2.munki	2.4.4.2
		autopkg-Flip4Mac-3.munki	3.2.0.16
		autopkg-GoogleChrome.munki	29.0.1547.65
		autopkg-GoogleEarth.munki	7.1
		autopkg-Handbrake.munki	0.9.9
		autopkg-Hiera.munki	1.2.1

Jenkins Job Builder

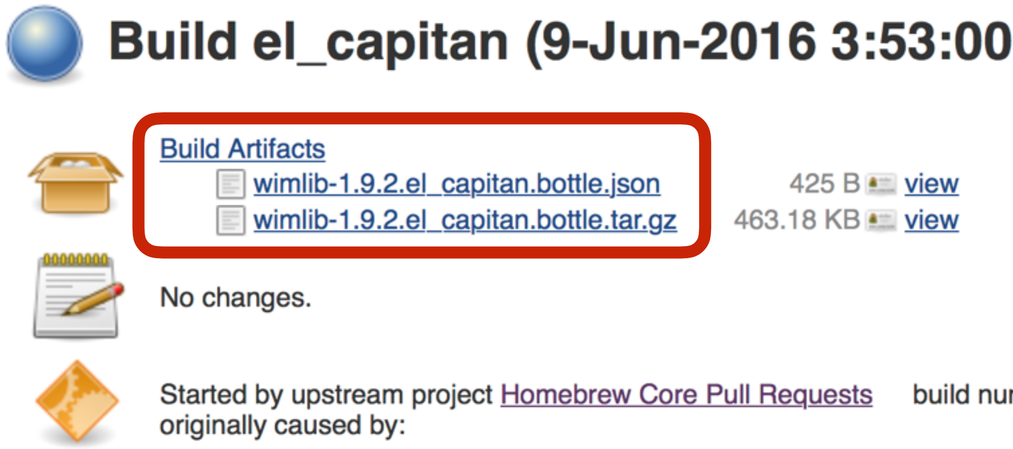
```
- job:
    name: job-name
    project-type: freestyle
    defaults: global
    description: 'Do not edit this job through the web!'
    disabled: false
    display-name: 'Fancy job name'
    concurrent: true
    workspace: /srv/build-area/job-name
    quiet-period: 5
    block-downstream: false
    block-upstream: false
    retry-count: 3
    node: NodeLabel1 || NodeLabel2
    logrotate:
      daysToKeep: 3
      numToKeep: 20
      artifactDaysToKeep: -1
      artifactNumToKeep: -1
```

API Wrappers

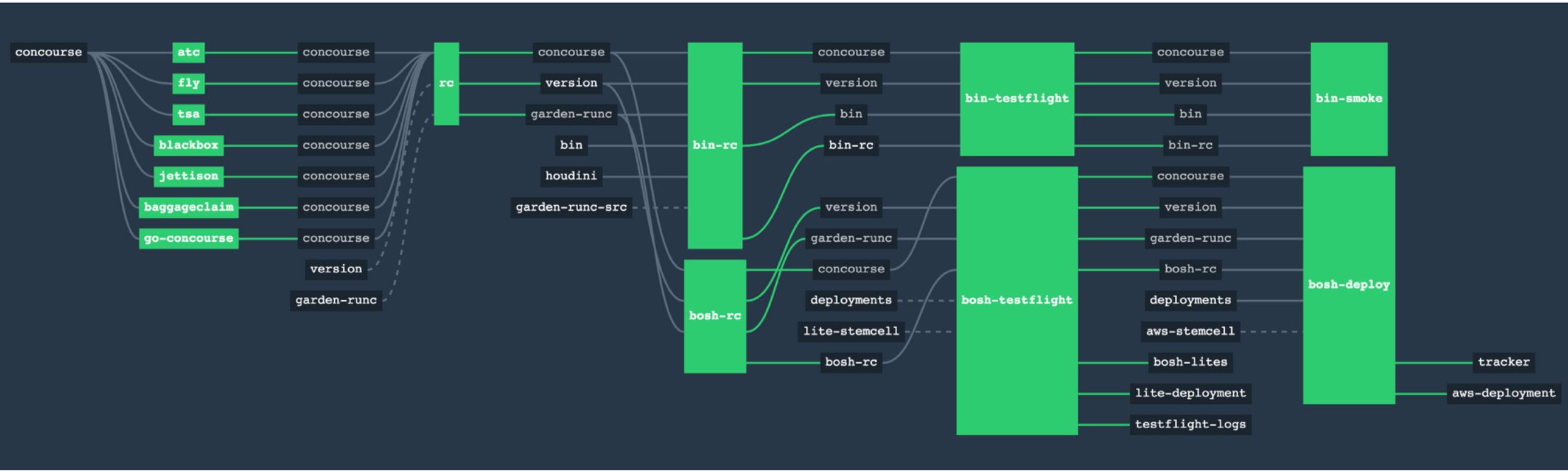


Mature workflows

```
==> git checkout origin/master
==> brew pull --clean https://github.com/Homebrew/homebrew-core/pull/1853
==> brew doctor
==> brew --env
==> brew config
==> brew readall --aliases homebrew/core
==> brew uses wimlib
==> brew fetch --retry makedepend openssl pkg-config
==> brew fetch --retry wimlib --build-bottle --force
==> brew install --only-dependencies --build-bottle --verbose wimlib
==> brew install --build-bottle --verbose wimlib
==> brew audit wimlib
==> brew bottle --verbose --json wimlib
==> brew bottle --merge --write --no-commit ./wimlib-1.9.2.el_capitan.bottle.json
==> brew uninstall --force wimlib
==> brew uninstall --force makedepend pkg-config
==> brew install ./wimlib-1.9.2.el_capitan.bottle.tar.gz
==> brew test wimlib --verbose
==> brew uninstall --force wimlib
==> brew uninstall --force openssl
==> git checkout master -f
==> git reset --hard
==> brew cleanup --prune=7
==> git clean -ffdx
HEAD is now at 11d47e8 boneyard-formula-pr: add new command.
Removing Cellar/
Removing Library/Locks/
Removing etc/openssl/
Recording test results
Archiving artifacts
```

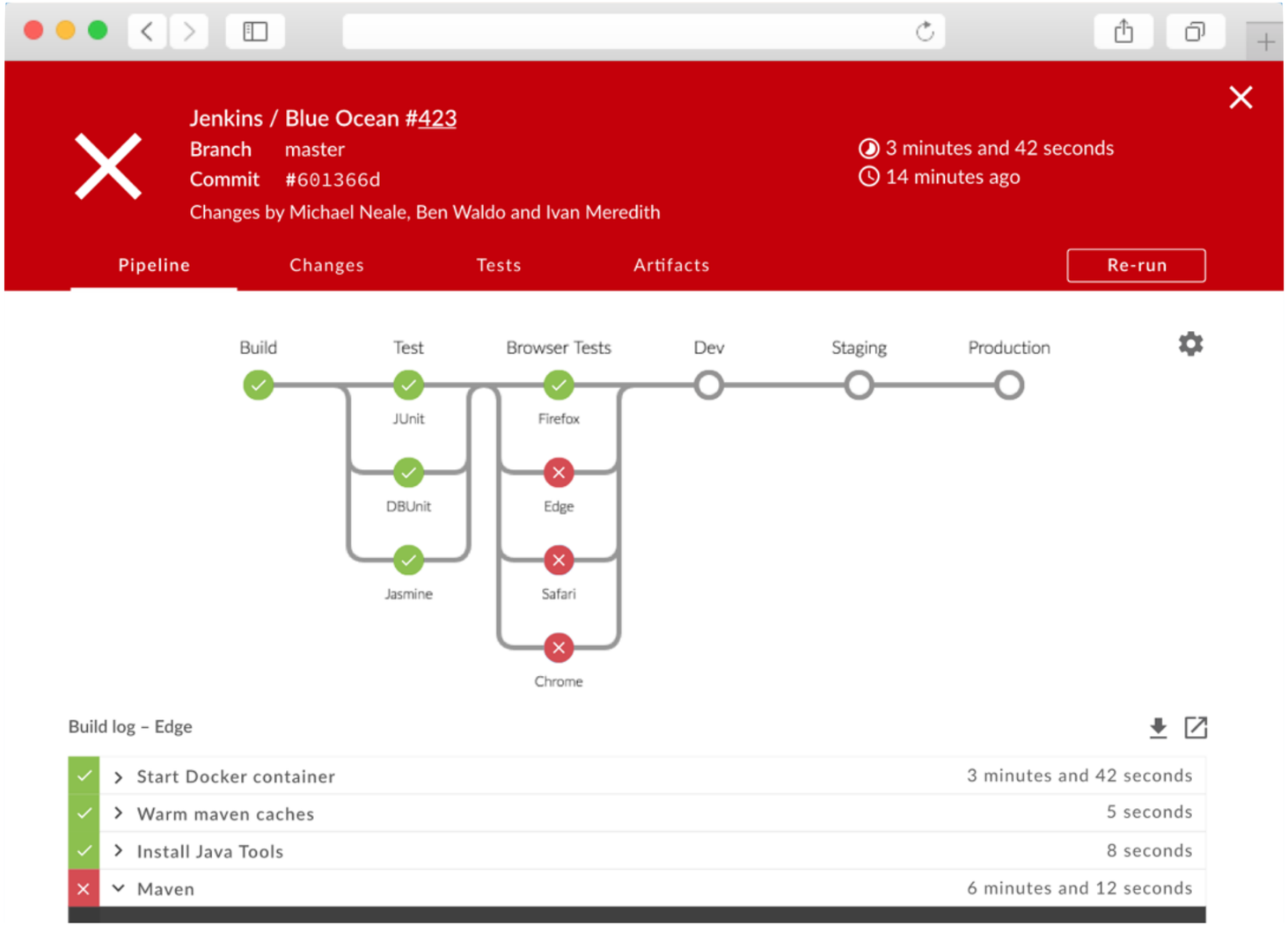


Pipelines

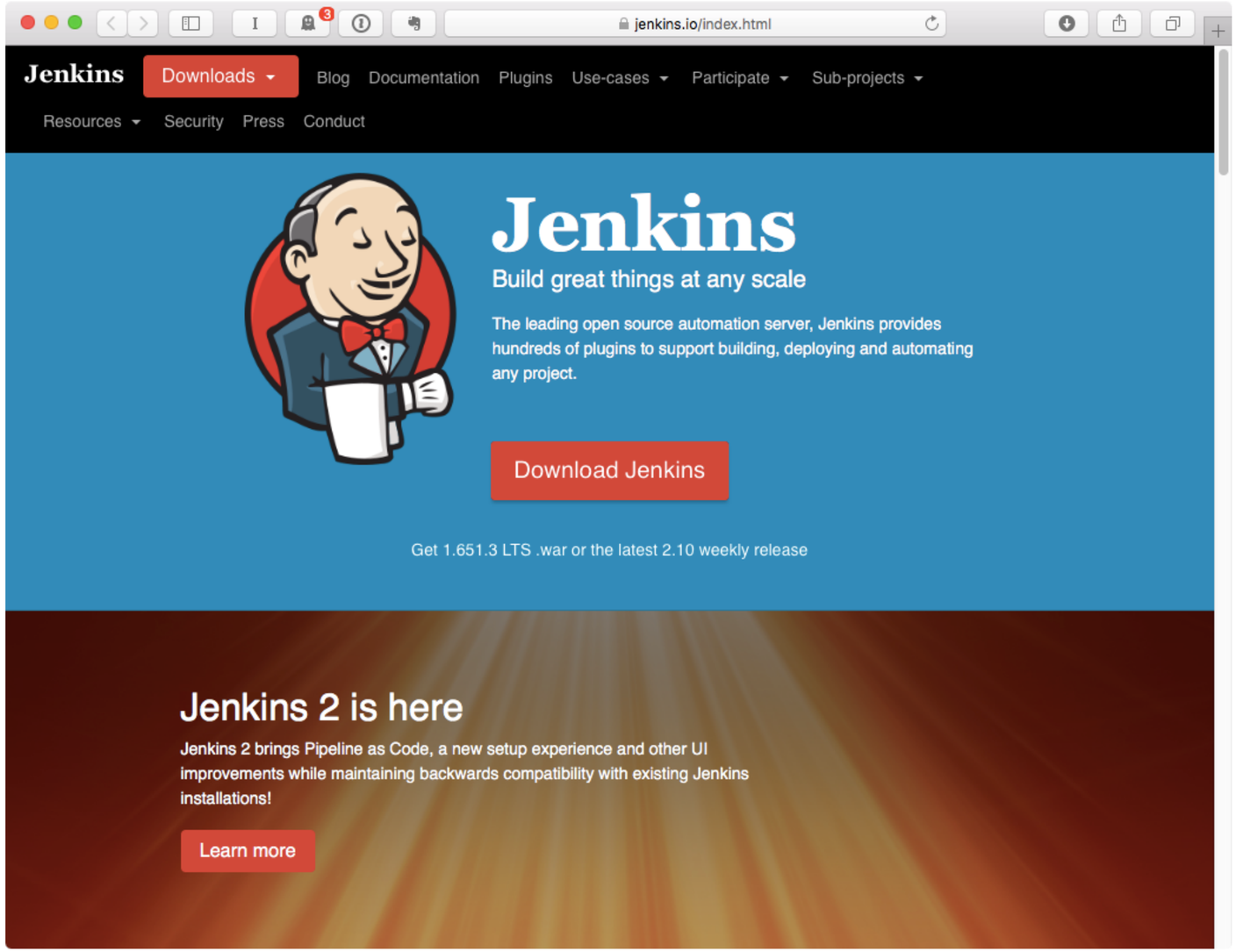


Concourse (Pivotal) pipeline

Jenkins Blue Ocean Project



Jenkins



https://jenkins.io

```
brew install jenkins
```

Other CI platforms

Name	Self-hosted / Enterprise	Cloud-hosted	Bring your own runner
buildbot	✔		✔
Go (Thoughtworks)	✔		✔
Concourse (Pivotal)	✔		✔
Buildkite		✔	✔
Travis-CI	✔	✔	
Appveyor		✔	
GitLab CI	✔	✔	✔
Circle CI		✔	
Bamboo	✔	✔	
BitBucket Pipelines		✔	
buddybuild		✔	

Thank you!

@timsutton

@tvsutton

https://macops.ca/macdevopsyvr-2016