

A Continuous Packaging Pipeline

Maciej Pasternacki <maciej@pasternacki.net>



3ofcoins



@mpasternacki

Web infrastructure problems

- Nginx is too old
- I need Node.js
- I need *foo.jar* on all application servers
- A Java VM would help using it...

Web infrastructure problems

- Nginx is too old
 - I need Node.js
 - I need *foo.jar* on all application servers
 - A Java VM would help using it...
- } Compiling on server
is **WRONG**

PACKAGE



ALL THIS STUFF!



Debian Policy Manual

Goals

1. `git push`

2....

3. Profit

Goals

1. `git push`

2....

3. `apt-get update && \`
`apt-get install foo`

Goals

- Single repository for all packages

Actually, why not chef-repo?

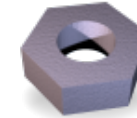
- One directory per package
- Share package definitions between projects
- Run various build systems



git



Vendorificator



Buildbot

Jenkins





git

Vendorificator



Vendor everything

<https://github.com/3ofcoins/vendorificator/>

Vendorificator

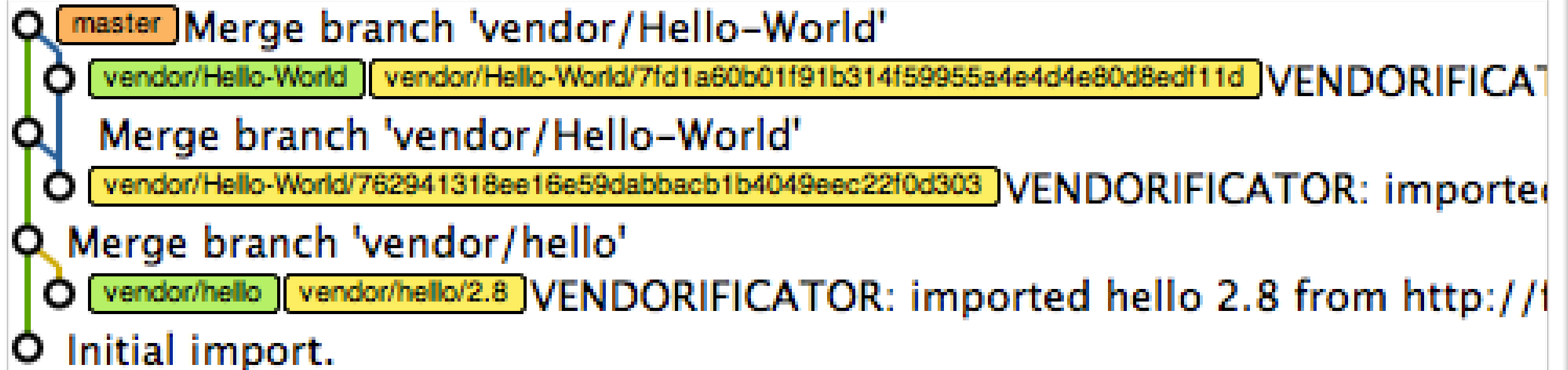
Integrate upstream modules
in a Git repository

```
archive 'hello', :version => '2.8',  
  :url => "http://ftp.gnu.org/gnu/hello/hello-2.8.tar.gz"  
git 'git://github.com/octocat/Hello-World.git'
```

<https://github.com/3ofcoins/vendorificator/>

Vendorificator

```
archive 'hello', :version => '2.8',  
  :url => "http://ftp.gnu.org/gnu/hello/hello-2.8.tar.gz"  
git 'git://github.com/octocat/Hello-World.git'
```




<https://github.com/3ofcoins/vendorificator/>

Vendorificator

- Share package definitions between projects
- Include packages' original sources in the repository

<https://github.com/3ofcoins/vendorificator/>



Continuous
Integration

Rake

A rake with a yellow handle and metal tines is leaning against a tree trunk. The rake is positioned vertically, with the handle extending upwards and the tines pointing downwards. The background shows a grassy area with some fallen leaves and a tree trunk. The word "Rake" is written in large, bright green letters across the top of the image.

<https://github.com/jimweirich/rake>

<http://www.flickr.com/photos/crowderb/287717759/>

Metarake

A Rake extension that:

1. Discovers modules and their build targets
2. Builds modules with unpublished targets
3. Publishes the built targets

<https://github.com/3ofcoins/metarake/>

Metarake

It's used to:

1. Find */Rakefile and their *.deb targets
2. Build packages not in the apt repo
3. Push built packages into the apt repo

<https://github.com/3ofcoins/metarake/>

Metarake

```
require 'metarake'
require 'metarake/publisher/freight'

class PackagesMetaTask < MetaRake::Task
  include MetaRake::Builder::Rake
  include MetaRake::Publisher::Freight

  self.target_filter = /\.deb$/
  self.rake_command = 'fakeroot bundle exec rake'
  self.freight_distro = 'squeeze'
  self.freight_conf_path = 'freight.conf'
end

task :default => PackagesMetaTask.discover!
```

<https://github.com/3ofcoins/metarake/>

Evoker

An add-on to Rake to download and manage external dependencies of a project, patch or modify them as needed, cache them, etc.

<https://github.com/3ofcoins/evoker/>

Evoker

```
tarball('swftools',
        :url => 'http://www.swftools.org/swftools-0.9.1.tar.gz')
patch 'swftools', 'swftools-Makefile_hack.patch', '-p1' if linux?
task 'swftools' => [ :python, 'swftools_pyconfig.py' ] do
  sh <<-EOF
  set -e -x
  cd swftools
  eval `../python/bin/python ../swftools_pyconfig.py`
  ./configure
  make
EOF
end
```

<https://github.com/3ofcoins/evoker/>

Evoker

- Download original sources in build time
- Cache them to prevent redownloads

<https://github.com/3ofcoins/evoker/>

FPM

<https://github.com/jordansissel/fpm/>

<http://www.flickr.com/photos/dominicpics/1406305389/>

FPM

- Create, convert, modify packages
- Sources:
gem (Ruby), pypi (Python), pear (PHP),
npm (Node.js), rpm, deb, **directory**
- Targets:
deb, rpm, solaris, tar, **directory**

<https://github.com/jordansissel/fpm/>

FPM

<https://github.com/jordansissel/fpm/>

FPM

```
$ fpm -s dir -t deb \  
-n jenkins -v 1.396 \  
--prefix /opt/jenkins jenkins.war
```

<https://github.com/jordansissel/fpm/>

FPM

```
$ fpm -s dir -t deb \  
    -n jenkins -v 1.396 \  
    --prefix /opt/jenkins jenkins.war
```

```
$ fpm -s gem -t deb json  
# rubygem-json-1.4.6-1.amd64.deb
```

<https://github.com/jordansissel/fpm/>

FPM

```
$ ./configure --prefix=/usr \  
  && make \  
  && make install DESTDIR=$ROOT  
  
$ fpm -s dir -t deb -n foo -v 0.1 \  
  -d "libssl0.9.8 (> 0)" \  
  -C $ROOT .
```

<https://github.com/jordansissel/fpm/>

Freight

<https://github.com/rcrowley/freight/>

<http://www.flickr.com/photos/mattblaze/4336640682/>

Freight

“A modern take on the Debian archive”

<https://github.com/rcrowley/freight>

Freight

“A modern take on the Debian archive”

```
$ freight add foo_0.0.1-1_all.deb \  
    apt/lucid apt/precise
```

<https://github.com/rcrowley/freight>

Freight

“A modern take on the Debian archive”

```
$ freight add foo_0.0.1-1_all.deb \  
    apt/lucid apt/precise
```

```
$ freight cache apt/lucid apt/precise
```

<https://github.com/rcrowley/freight>



```
apt-get install anarchism
```

<https://we.riseup.net/debian/install-anarchism>

Example

`nodejs/Rakefile`

Example

```
PKG_NAME = 'nodejs'  
PKG_DESCRIPTION = 'Node.js'  
PKG_VERSION = '0.8.17'  
PKG_ITERATION = 'local1'  
PKG_ARCH = 'amd64'  
PKG_DEB_FILE =  
    "#{PKG_NAME}_#{PKG_VERSION}-#{PKG_ITERATION}_#{PKG_ARCH}.deb"  
  
SRCDIR = "node-v#{PKG_VERSION}"  
TARBALL = "#{SRCDIR}.tar.gz"  
URL = "http://nodejs.org/dist/v#{PKG_VERSION}/#{TARBALL}"  
CHECKSUM = '8f070b42ffb84fde9d3ed2f802b08664b94dda327a36bf08...'  
BUILT = "#{SRCDIR}/out/Release/node"  
INSTALL_ROOT = File.expand_path('__install_root__')  
INSTALLED = "#{INSTALL_ROOT}/opt/node/bin/node"
```


Example

```
require 'evoker'  
require 'evoker/local_cache'  
  
tarball = Evoker::cached_wget(  
  URL, :checksum => CHECKSUM)
```

Example

```
file BUILT => tarball do
  rm_rf SRCDIR
  sh "tar -xzf #{tarball}"
  chdir SRCDIR do
    sh "./configure --prefix=/opt/node"
    sh "make"
  end
end

file INSTALLED => BUILT do
  rm_rf INSTALL_ROOT
  chdir(SRCDIR) { sh "make install DESTDIR=#{INSTALL_ROOT}" }
end
```

Example

```
desc "Build the package"
file PKG_DEB_FILE => INSTALLED do
  sh <<EOF
fpm -s dir -t deb -C #{INSTALL_ROOT} \\
  --license 'All rights reserved' --vendor 'Ourselves' \\
  --name #{PKG_NAME} --version #{PKG_VERSION} \\
  --iteration #{PKG_ITERATION} --architecture #{PKG_ARCH} \\
  --description '#{PKG_DESCRIPTION}' \\
  .
EOF
end

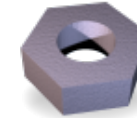
task :default => PKG_DEB_FILE
```



git

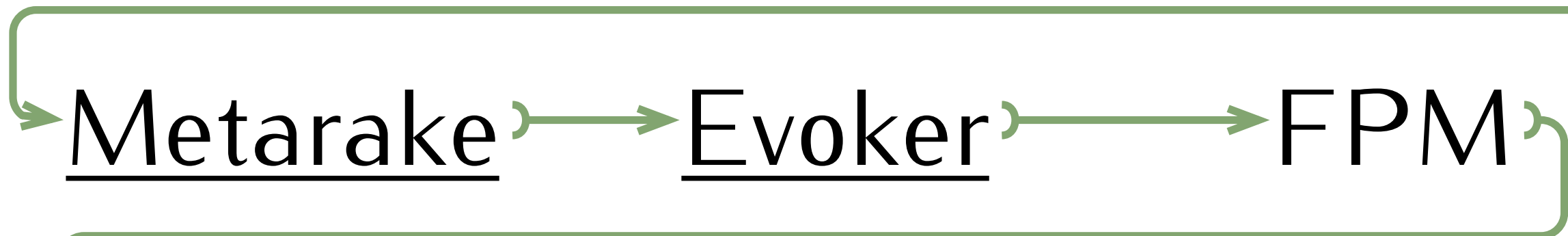


Vendorificator



Buildbot

Jenkins



<https://github.com/...>

...3ofcoins/vendorificator

...jimweirich/rake

...3ofcoins/metarake

...3ofcoins/evoker

...jordansissel/fpm

...rcrowley/freight

} bit.ly/cont-pkg

Maciej Pasternacki <maciej@pasternacki.net>