Groovy, there is a docker in my application pipeline

Kris Buytaert

@krisbuytaert

Kris Buytaert

- I used to be a Dev,
- Then Became an Op
- Chief Trolling Officer and Open Source Consultant @inuits.eu
- Everything is an effing DNS Problem
- Building Clouds since before the bookstore
- Some books, some papers, some blogs
- Evangelizing devops
- Organiser of #devopsdays, #cfgmgmtcamp, #loadays, ….
- Part of the travelling geek circus

What's this devops thing anyhow ?

C(L)AMS

- Culture
- (Lean)
- Automation
- Measurement
- Sharing

Damon Edwards and John Willis

Gene Kim

Nirvana

An “ecosystem” that supports continuous delivery, from infrastructure, data and configuration management to business.

Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers, and operations, delivery teams can get changes released in a matter of hours — sometimes even minutes–no matter what the size of a project or the complexity of its code base.

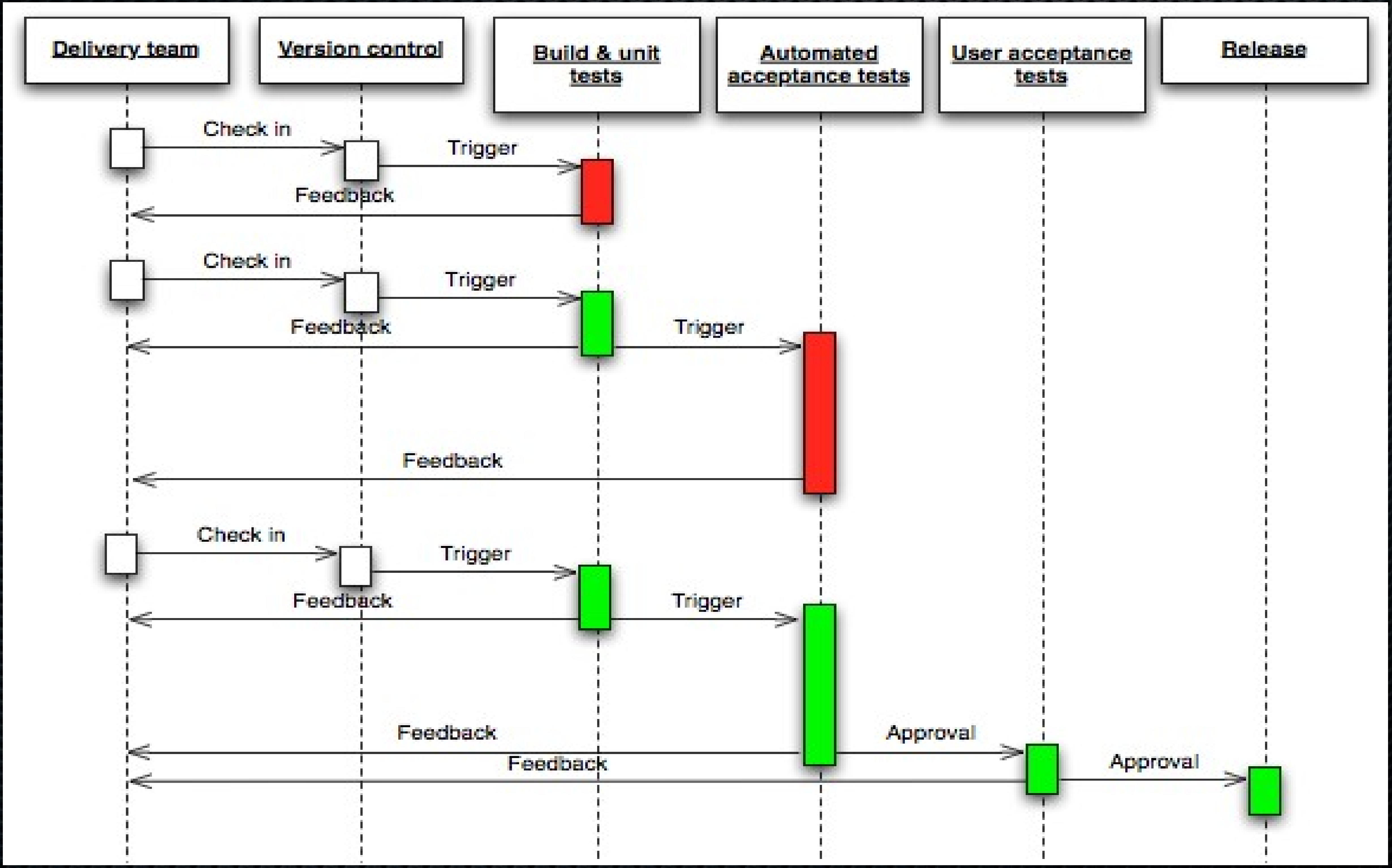
Continuous Delivery , Jez Humble

This talk:

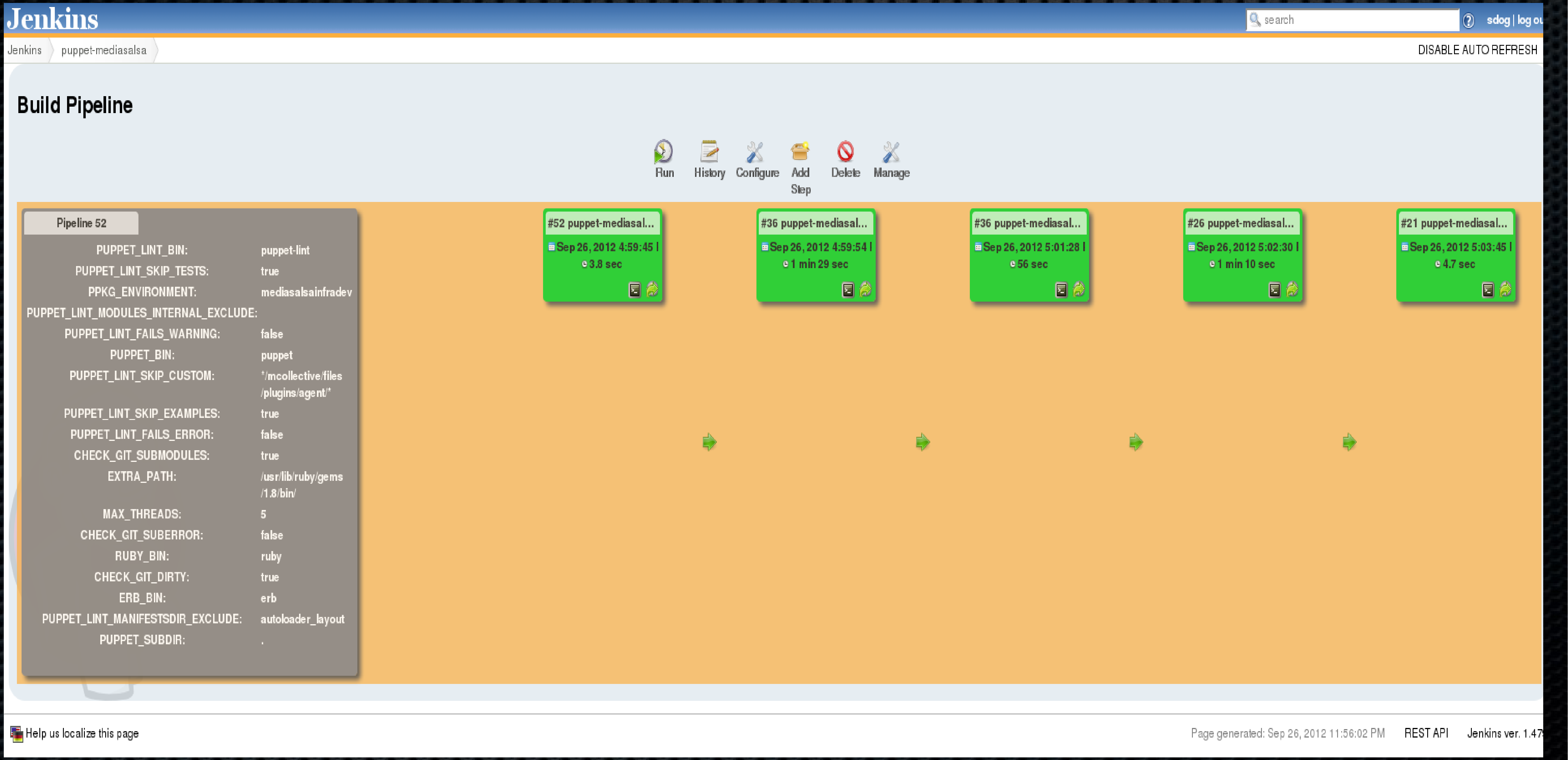
Journey / Early steps of a team that is used to infrastructure as code

Adopting containers step by step.

Build Pipelines



Jenkins Pipeline



Build Pipeline

Pipeline 52

PUPPET_LINT_BIN:	puppet-lint
PUPPET_LINT_SKIP_TESTS:	true
PPKG_ENVIRONMENT:	mediasalsainfradev
PUPPET_LINT_MODULES_INTERNAL_EXCLUDE:	
PUPPET_LINT_FAILS_WARNING:	false
PUPPET_BIN:	puppet
PUPPET_LINT_SKIP_CUSTOM:	*/mcollective/files /plugins/agent/*
PUPPET_LINT_SKIP_EXAMPLES:	true
PUPPET_LINT_FAILS_ERROR:	false
CHECK_GIT_SUBMODULES:	true
EXTRA_PATH:	/usr/lib/ruby/gems /1.8/bin/
MAX_THREADS:	5
CHECK_GIT_SUBERROR:	false
RUBY_BIN:	ruby
CHECK_GIT_DIRTY:	true
ERB_BIN:	erb
PUPPET_LINT_MANIFESTSDIR_EXCLUDE:	autoloader_layout
PUPPET_SUBDIR:	.

#52 puppet-mediasal...
Sep 26, 2012 4:59:45 I
3.8 sec

#36 puppet-mediasal...
Sep 26, 2012 4:59:54 I
1 min 29 sec

#36 puppet-mediasal...
Sep 26, 2012 5:01:28 I
56 sec

#26 puppet-mediasal...
Sep 26, 2012 5:02:30 I
1 min 10 sec

#21 puppet-mediasal...
Sep 26, 2012 5:03:45 I
4.7 sec

Help us localize this page

Page generated: Sep 26, 2012 11:56:02 PM REST API Jenkins ver. 1.47

" Our job as engineers (and ops, dev-ops, QA, support, everyone in the company actually) is to enable the business goals. We strongly feel that in order to do that you must have the ability to deploy code quickly and safely. Even if the business goals are to deploy strongly QA’d code once a month at 3am (it’s not for us, we push all the time), having a reliable and easy deployment should be non-negotiable."

Etsy Blog upon releasing Deployinator

http://codeascraft.etsy.com/2010/05/20/quantum-of-deployment/

We need :

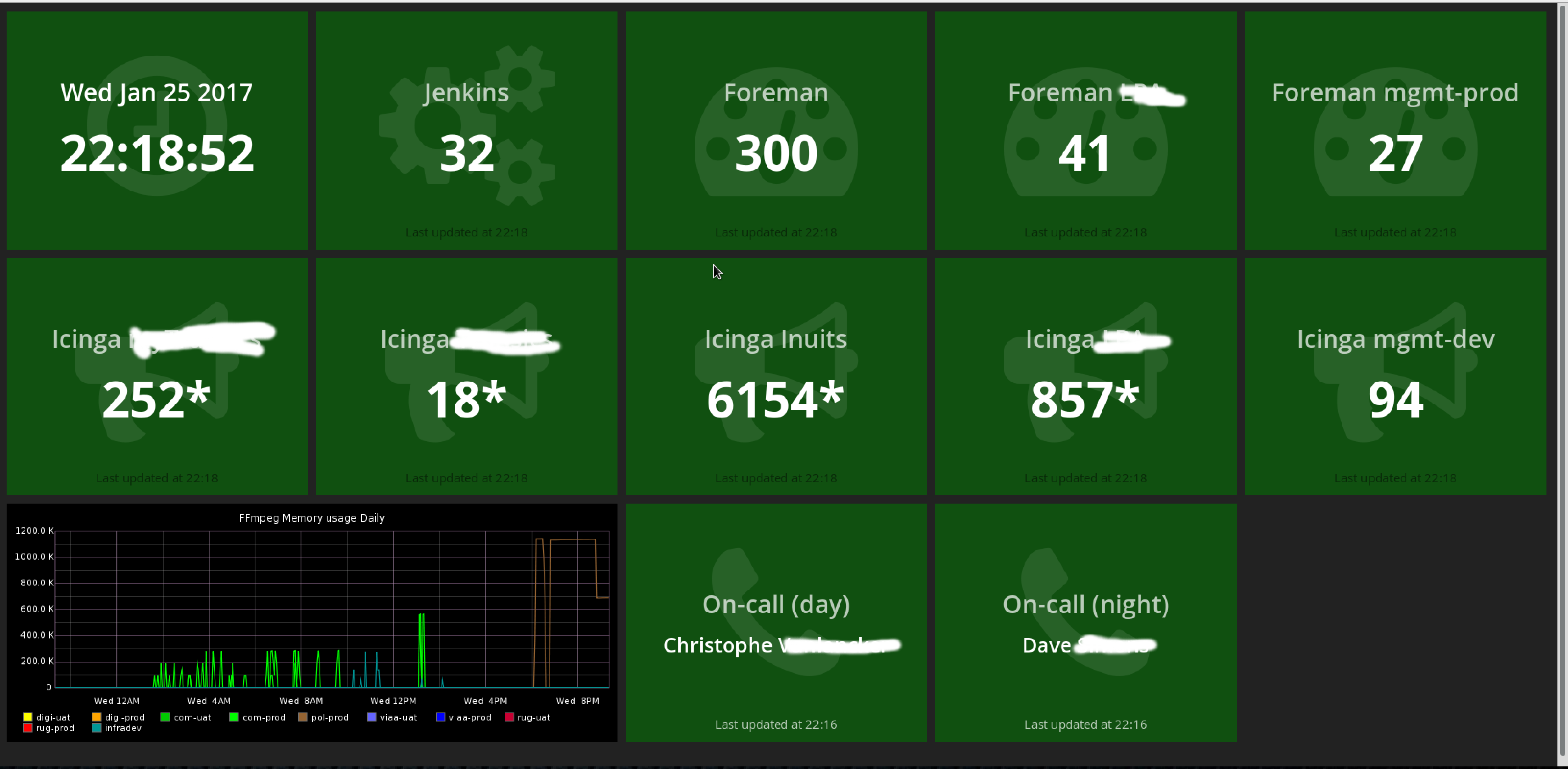
An unmodified artifact from build to deploy.

Same artifact on dev, staging, acceptance, production, shadow, dr ...

Why ops like to package

- Packages give you features
  - Consistency, security, dependencies
- Uniquely identify where files come from
  - Package or cfg-mgmt
- Source repo not always available
  - Firewall / Cloud etc ..
- Weird deployment locations , no easy access
- Little overhead when you automate
- CONFIG does not belong in a package

Example app for today : Dashing



Wed Jan 25 2017

22:18:52

Jenkins

32

Last updated at 22:18

Foreman

300

Last updated at 22:18

Foreman [redacted]

41

Last updated at 22:18

Foreman mgmt-prod

27

Last updated at 22:18

Icinga [redacted]

252*

Last updated at 22:18

Icinga [redacted]

18*

Last updated at 22:18

Icinga Inuits

6154*

Last updated at 22:18

Icinga [redacted]

857*

Last updated at 22:18

Icinga mgmt-dev

94

Last updated at 22:18

FFmpeg Memory usage Daily

On-call (day)

Christophe [redacted]

Last updated at 22:16

On-call (night)

Dave [redacted]

Last updated at 22:16

Dashing is Dead

- No it has been forked
- https://github.com/dashing-io/dashing
- s/dashing/smashing/g;

Dashing {su/ro}cks

The Good

- Lots of existing widgets
- Easy to start
- Simple ruby
- Eventstream for debugging

The Ugly

- Ruby Gem hell
- Widget Deployment from a Gist ?
- No config separation

Deploying Dashing

- gem install dashing
- gem install is the new maven downloading the internet
- Reproducable ?

A typical deployment

- P all software is packaged
  - CentOS mostly
  - RPM generated with fpm
  - Build in Jenkins, uploaded to pulp
- C config is managed by Puppet
- S service is managed by Puppet

Building Ruby/python/node

- We need a chroot
- With the right ruby/python version
- With the right dependencies
- Isolated
- Ruby => rvm
- Ruby 2.1 (dashing is pretty picky about versions)
- What about we try this in a container ?

Pipelines ?

- One to build basic dashing
- One to build and deploy the dashboards, scripts and all other dashing related stuff
  - No hacking In production,
  - Dashboards are production views
  - Dev → prod promotions

Jenkins

- Starting point :
  - Dev jenkins
  - 1 master (no running jobs)
  - Multiple slaves
- Production : different jenkins stack with similar pipelines
- We need to be able to reproduce a pipeline

Building a dashing container step 0

- Empty / standard distro container

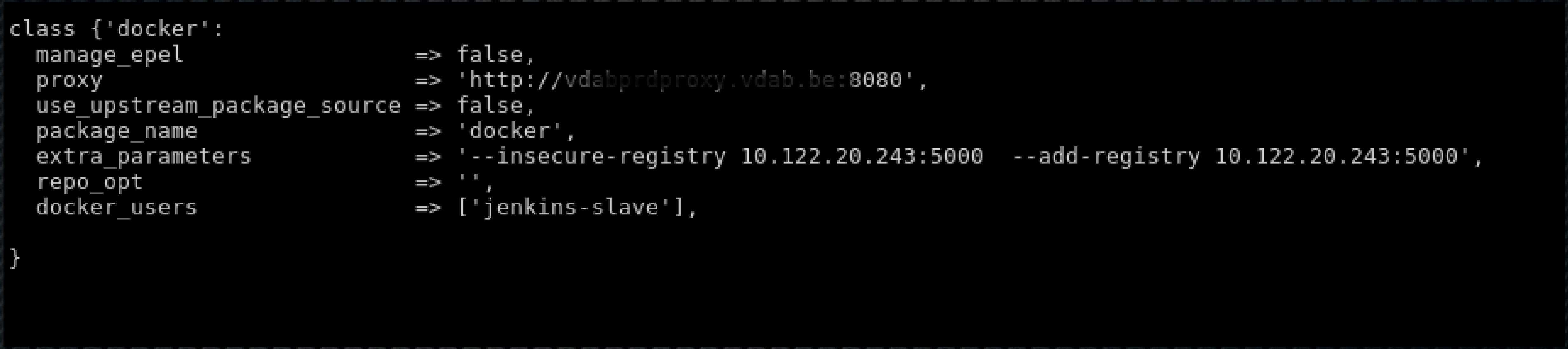
  updates

  add fpm

  epel and build dependencies
- Triggering docker from the cli, no plugin in Jenkins used (coz Bugz)
- $customer environment requires http_proxy

Puppet & Docker

- https://github.com/garethr/garethr-docker



```
class {'docker':
  manage_epel                 => false,
  proxy                       => 'http://vdabprdproxy.vdab.be:8080',
  use_upstream_package_source => false,
  package_name                => 'docker',
  extra_parameters            => '--insecure-registry 10.122.20.243:5000  --add-registry 10.122.20.243:5000',
  repo_opt                    => '',
  docker_users                => ['jenkins-slave'],

}
```

Building a dashing container step 1

- Read rvm installation docs
- frown
- Frown again
- Containers => Yolo
- fpm the whole tree

Building a dashing container step 2

- Take rvm container
- rvm install ruby-2.1
- fpm -s dir -t rpm -n rvm-ruby -v 2.1.8 /usr/local/rvm/rubies/ruby-2.1.8

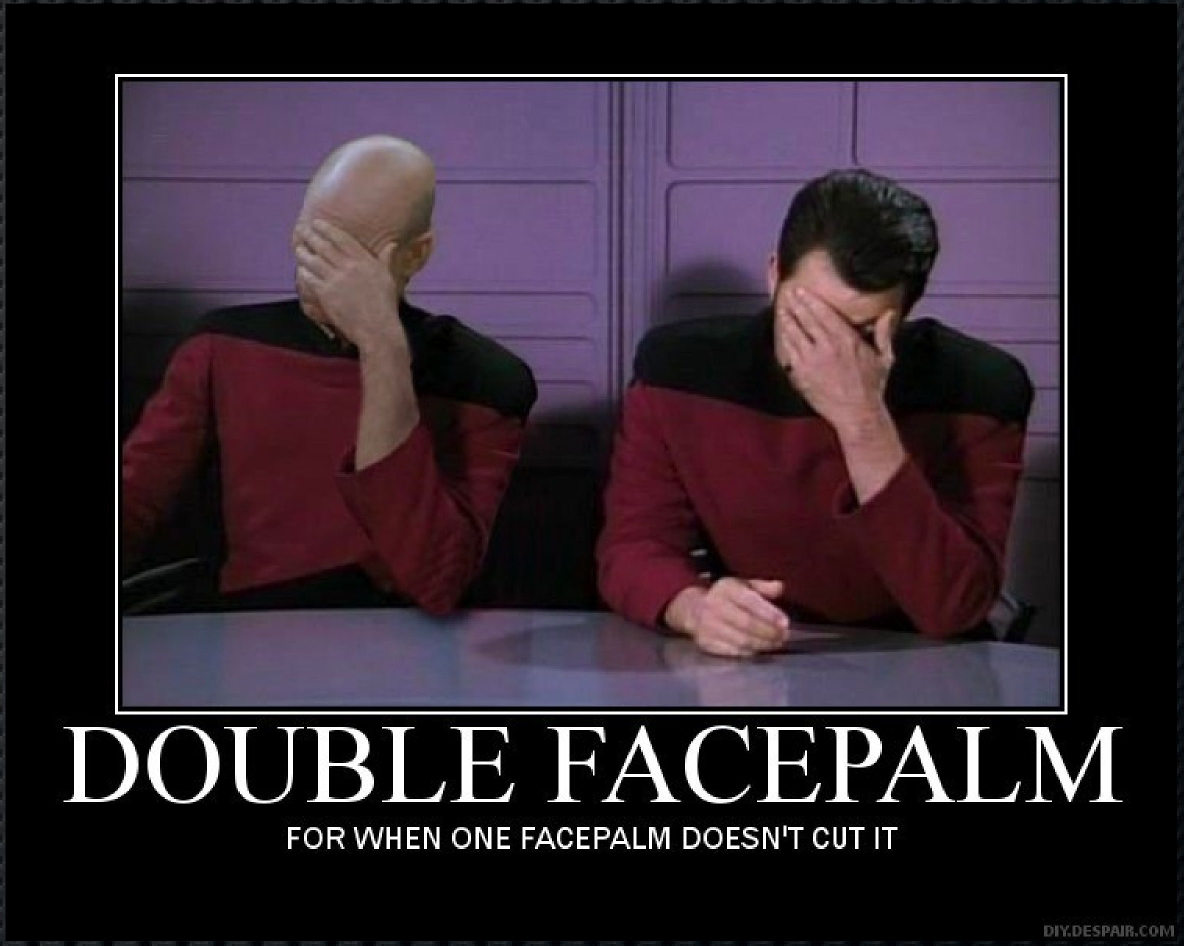
Building a dashing container step 3

- Take ruby-2.1 container
- rvm use 2.1
- gem install bundle
- gem install dashing (fills /usr/local/rvm/gems/ruby-2.1.8 with gems
- mkdir -p /opt/dashing/ && dashing new dashboard
- cd /opt/dashing/dashboard
- bundle install

- Now we have a “reproducable” container which will show an empty default dashboard upon launching
- We also have an artifact which we can redeploy
- We killed most of those layers afterwards

Deploying Dashboard widgets

dashing install GIST_ID



A dashboard

- git repo with
  - Dashboards (html/erb)
  - Jobs
    - Mostly with datasources hardcoded in scripts
    - Not multitenant
  - Widgets
- Pipeline to deploy and test that

Deploying the dashboards

- From dashing container
- Cleanup default dashboards (twitter example)
- Bundle install
  - ruby scripts have dependencies
- Package dependencies
  - actually whole /usr/local/rvm
- Package dashboard

Testing the dashboards

- Not all deploys were working
- New job, required gems are missing
- Testing
  - Build container with most recent dashboard
  - Based on the rpm's
  - docker run -p 0.0.0.0:3030:3030 -d dashing/dashboards
  - wget http://localhost:3030/

Deploying the dashboards

- Deploy 2 rpms on vm's via mcollective
  - dashing-gems
  - dashing-dashboard

  on nodes with profile_dashing
- mco package update   dashing-gems -F environment=svc1prd -C profile_dashing

We need a local docker images repository

- Distributed Jenkins (master + multiple slaves)
- An image build on node X is not available on node Y
- Tests run on other node

```
docker push dashing/dashing
docker push dashing/dashboards
```

We need a local docker images repository

- Pulp ?
  - Read only (August 2016)
  - Good for mirrors
- Nexus / Artifactory
- Docker registry (obsolete, used to be only in a container)
- Docker-distribution : packages available

Docker Incompatibilities

- Search path for images
  - Local first
  - Upstream afterwards

  Docker Inc says NO

  Redhat says Yes
- --build-args   -e
  - Redhat vs Docker implementation differ

Problems solved

- Chrooted package build, no complex mock setups or specfiles
- Internal docker repo allows reuse of build images on other nodes
- Jenkins and docker “integration”

Rinse & Repeat

- Similar patterns for
  - Python , php, etc.
- Test can now run in containers with the correct version
- Tests can be run with multiple versions of php/ phython/ruby etc..

Can you Automate your Pipeline Creation ?

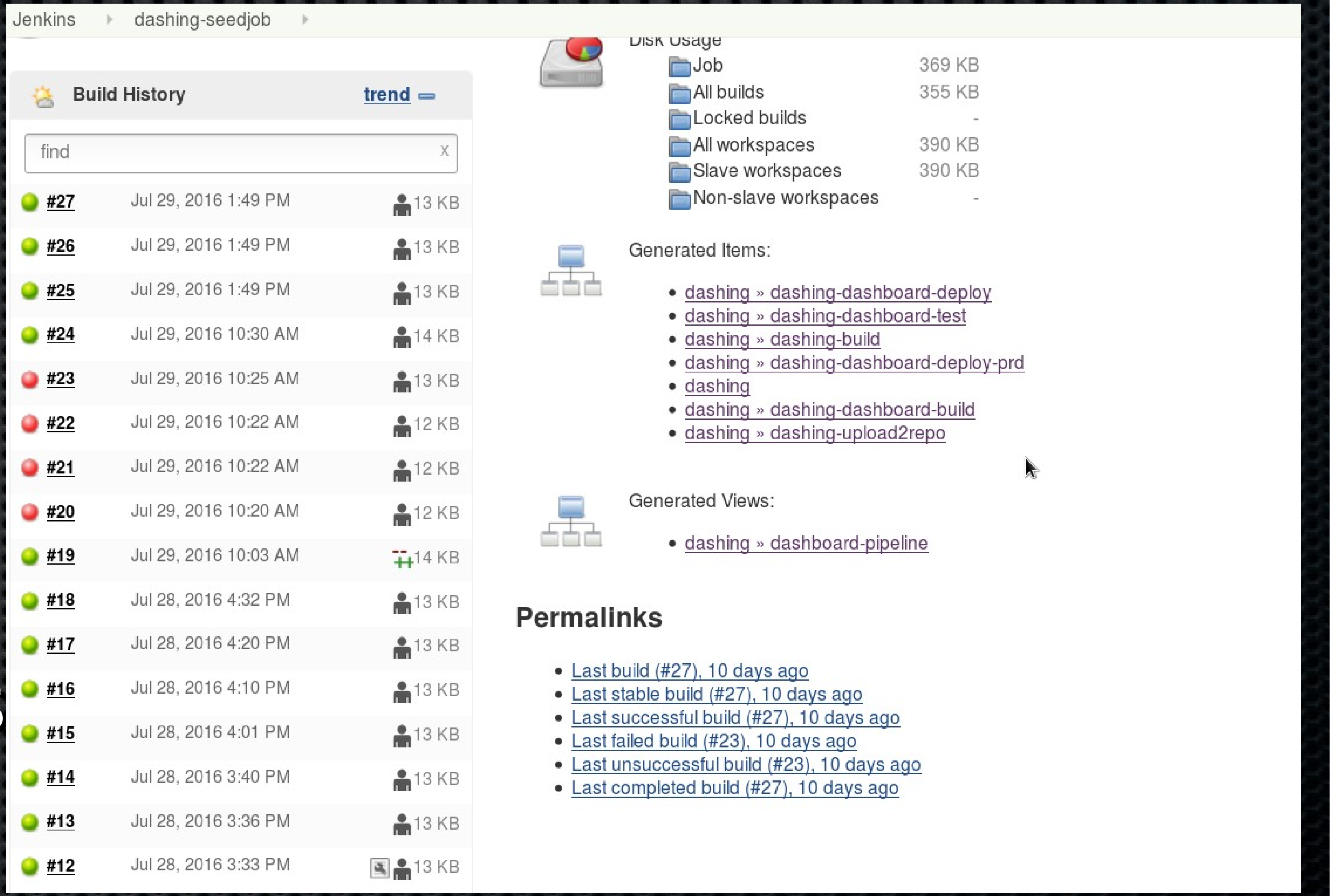
- Pipeline as Code
- Jenkins Job DSL
- Pipeline Plugin

Building the Pipeline

- Dev environment for Jenkins
  - Fully puppetized
- Jobs
  - Jenkins Job DSL Plugin
  - https://wiki.jenkins-ci.org/display/JENKINS/Job+DSL+Plugin
  -

Seedjob

- Groovy
- Git
- Rebuild jobs on commit
- Keep numbers
- Projects in folders



Jenkins › dashing-seedjob ›

Build History trend

find

🟢	#27	Jul 29, 2016 1:49 PM	13 KB
🟢	#26	Jul 29, 2016 1:49 PM	13 KB
🟢	#25	Jul 29, 2016 1:49 PM	13 KB
🟢	#24	Jul 29, 2016 10:30 AM	14 KB
🔴	#23	Jul 29, 2016 10:25 AM	13 KB
🔴	#22	Jul 29, 2016 10:22 AM	12 KB
🔴	#21	Jul 29, 2016 10:22 AM	12 KB
🔴	#20	Jul 29, 2016 10:20 AM	12 KB
🟢	#19	Jul 29, 2016 10:03 AM	14 KB
🟢	#18	Jul 28, 2016 4:32 PM	13 KB
🟢	#17	Jul 28, 2016 4:20 PM	13 KB
🟢	#16	Jul 28, 2016 4:10 PM	13 KB
🟢	#15	Jul 28, 2016 4:01 PM	13 KB
🟢	#14	Jul 28, 2016 3:40 PM	13 KB
🟢	#13	Jul 28, 2016 3:36 PM	13 KB
🟢	#12	Jul 28, 2016 3:33 PM	13 KB

Disk Usage

Job	369 KB
All builds	355 KB
Locked builds	-
All workspaces	390 KB
Slave workspaces	390 KB
Non-slave workspaces	-

Generated Items:

- dashing » dashing-dashboard-deploy
- dashing » dashing-dashboard-test
- dashing » dashing-build
- dashing » dashing-dashboard-deploy-prd
- dashing
- dashing » dashing-dashboard-build
- dashing » dashing-upload2repo

Generated Views:

- dashing » dashboard-pipeline

Permalinks

- Last build (#27), 10 days ago
- Last stable build (#27), 10 days ago
- Last successful build (#27), 10 days ago
- Last failed build (#23), 10 days ago
- Last unsuccessful build (#23), 10 days ago
- Last completed build (#27), 10 days ago

A dsl project

```
def foldername = 'project-x'

folder(“${foldername}”)

job(“${foldername}/XYZ”) {}

job(“${foldername}/ABC”) {}

buildPipelineView(“${foldername}/pipeline”)
        { selectedJob(“${foldername}/XYZ”) }
```

dashing dsl project

```
job("${foldername}/dashing-dashboard-build")

job("${foldername}/dashing-dashboard-test")

job("${foldername}/dashing-upload2repo")

job("${foldername}/dashing-dashboard-deploy")

job("${foldername}/dashing-dashboard-deploy-prd")

job("${foldername}/dashing-build")
```

Job parts

- Logrotator : how long to keep builds
- Scm : git config
- Trigger : when to build
- Label : where to run
- Steps : shell(readFileFromWorkspace('file.sh'))
- publishers

Publisher Parts

- PublishBuild
- CloneWorkspacePublisher
- ArchiveArtifacts
- Downstreamparametrized ...

Pipeline Problems solved

- No more promoted build plugin
  - Manual promote in pipeline
  - Easy visability
- No more clicking around to create / edit pipeline
- One job per task, no reuse of jobs with different parameters
- Centrally managed jobs (git)

Solved problems by Containers

- Multiversion test of application stacks
  - Eg different puppet/ php versions
- Both functional and unit testing in the pipeline
- Non blocking pipeline branches for future versions
- Provide developers with producton alike containers
- Growing container experience with ops folks

Contact

Kris Buytaert Kris.Buytaert@inuits.be

Further Reading
@krisbuytaert
http://www.krisbuytaert.be/blog/
http://www.inuits.be/

Inuits

Essensteenweg 31
Brasschaat
Belgium
891.514.231

+32 475 961221