Continuous Integration in Data Centers
Further 3 Years Later

Michael Prokop
Mika || @mikagrmr
Independent Consultant
Project leader of Grml.org
Debian Developer
Jenkins-debian-glue.org

Source: https://flic.kr/p/pzMqux
Why Continuous Delivery?

Jenkins  Vagrant/Veewee  Puppet

Jenkins-debian-glue.org  Admin Docs (Sphinx)

Custom Grml ISOs (grml-live/grml2iso)
Roadmap

What do we want?
Challenges
How did we get there?
Code Review
Testing Tools
Docker
Infrastructure Tooling
Antipatterns
Pain Points
What do we want?
Expectations

- Independence
- Reliability
- Reproducibility
- Traceability
- Scalability
- Performance
- Predictability
- Innovation
Deployment Pipeline

Source: http://continuousdelivery.com/2010/02/continuous-delivery/
Challenges
Changes since 2013?

More employees
More remote workers
More customers
More projects

Source: https://twitter.com/gabrealness/status/723434940734431237
How did we get there?
“If I had eight hours to chop down a tree, I'd spend six hours sharpening my ax.” – Abraham Lincoln

“We learn geology the morning after the earthquake.” – Ralph Waldo Emerson
Improve

Better documentation
Better communication

Leverage new tools!

More Tests/
More Testing
Faster Release
Cycle
Workflow + Pipeline

Development/Testing

- git commit & git review

Debian builds (+PPA)

- Jenkins verify (-1/+1)

Code Reviewers (-2/-1/0/+1+2)

- Submit to {master,$branch}

- Internal tooling

Available to Customers

$Release (incl. Q/A)

Final Debian build

Release dashboard

$Product

Debian package, Puppet, ...

Internal tooling
Code Review
Gerrit
GitLab
Phabricator
Review Board

Source: https://memegenerator.net/instance/53083760
Advantages of CR

- Share knowledge
- Broadcast progress
- Improve maintainability
- Better code
- Communal ownership
## Gerrit

### Search for status:open

<table>
<thead>
<tr>
<th>Subject</th>
<th>Status</th>
<th>Owner</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
<th>Size</th>
<th>CR</th>
<th>V</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix pep8 import errors</td>
<td></td>
<td>Mohankumar</td>
<td>openstack/networking-sfc</td>
<td>master (bug/1566849)</td>
<td>3:46 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prevent keystone eventlet from starting</td>
<td></td>
<td>Matthew J Black</td>
<td>openstack/puppet-keystone</td>
<td>master (bug/1569390)</td>
<td>3:46 PM</td>
<td></td>
<td></td>
<td></td>
<td>+1</td>
</tr>
<tr>
<td>Convergence: DB api to get all active resources of stack</td>
<td></td>
<td>Rakesh H S</td>
<td>openstack/heat</td>
<td>master (bug/1533170)</td>
<td>3:46 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updates for labels icon and fixes indent positions</td>
<td></td>
<td>Bogdan Dudko</td>
<td>openstack/fuel-ui</td>
<td>master (ui-fix-labels-icon)</td>
<td>3:46 PM</td>
<td></td>
<td></td>
<td></td>
<td>+1</td>
</tr>
<tr>
<td>Verify that the information stored in the database via the fuel_cli</td>
<td></td>
<td>Mikhail Samoylov</td>
<td>openstack/fuel-qa</td>
<td>master (test_db_api_cli)</td>
<td>3:46 PM</td>
<td></td>
<td></td>
<td></td>
<td>+1</td>
</tr>
<tr>
<td>[ubuntu-bootstrap] Fix multipath-reload timeout values</td>
<td></td>
<td>Aleksey Zvyagintsev</td>
<td>openstack/fuel-agent</td>
<td>master (bug/1572210)</td>
<td>3:46 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use reno for release notes</td>
<td></td>
<td>Monty Taylor</td>
<td>openstack/infra-agent</td>
<td>master (add-reno)</td>
<td>3:46 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Added configuration</td>
<td></td>
<td>Karthik S</td>
<td>openstack/networking-ovs-dpdk</td>
<td>master (bug/1461916)</td>
<td>3:46 PM</td>
<td></td>
<td></td>
<td></td>
<td>+1</td>
</tr>
<tr>
<td>Add multipath enhancement to Storwize iSCSI driver</td>
<td></td>
<td>xiaojin</td>
<td>openstack/cinder</td>
<td>master (bp/storwize-iscsi-multipath-enhancement)</td>
<td>3:45 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for both microversion headers</td>
<td></td>
<td>Chris Dent</td>
<td>openstack/cinder</td>
<td>master (cd/handle-new-microversion)</td>
<td>3:45 PM</td>
<td></td>
<td></td>
<td></td>
<td>+1</td>
</tr>
<tr>
<td>Fix Lun ID 0 in HPE 3PAR driver</td>
<td></td>
<td>Jay Mehta</td>
<td>openstack/nova</td>
<td>master (bug/1573298)</td>
<td>3:45 PM</td>
<td></td>
<td></td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>portsecurity_db_common: Access db columns in a consistent way</td>
<td></td>
<td>Ihar Hrachyshka</td>
<td>openstack/cinder</td>
<td>stable/kilo (bug/1509312)</td>
<td>3:45 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generic details display framework</td>
<td></td>
<td>Matt Borland</td>
<td>openstack/horizon</td>
<td>master (bp/angularize-images-table)</td>
<td>3:45 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove translations for vendor specific networking-* projects</td>
<td></td>
<td>Andreas Jaeger</td>
<td>openstack-infra/project-config</td>
<td>master (stadium-redux)</td>
<td>3:45 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Docs: Split Network Services section into multiple files</td>
<td></td>
<td>Pedro Magalhães (pjms)</td>
<td>openstack/openstack-ansible</td>
<td>master (toctree_network_services)</td>
<td>3:45 PM</td>
<td></td>
<td></td>
<td></td>
<td>+1</td>
</tr>
</tbody>
</table>
Git review

playground: zsh - Konsole

mika@heart $ sgit/playground (git)-[master|U] % git diff
diff --git a/README.md b/README.md
index ce8b250..3078d97 100644
--- a/README.md
+++ b/README.md
@@ -1 +1 @@
-Tue Dec 29 13:40:15 CET 2015
+Demo for OSDC 2016

mika@heart $ sgit/playground (git)-[master|U] % git commit -a -m 'MT#0042 git-review'
[master 9385fbe] MT#0042 git-review demo for OSDC
 1 file changed, 1 insertion(+), 1 deletion(-)

mika@heart $ sgit/playground (git)-[master|↑1] % git review
remote: Processing changes: new: 1, refs: 1, done
remote:
remote: New Changes:
remote: http://gerrit.mgm.sipwise.com/5785 MT#0042 git-review demo for OSDC
remote:
To ssh://mprokop@gerrit.mgm.sipwise.com:29418/playground
 * [new branch] HEAD -> refs/publish/master

mika@heart $ sgit/playground (git)-[master|↑1] %
**Deprecate `projectName` setting of CopyArtifact**

Starting with version 1.26 of the CopyArtifact plugin all Jenkins job configurations using this plugin get rewritten on the fly if you run at least one build. This causes serious failures of the CopyArtifact feature as soon as the project name includes axis selections.

If the config.xml of a job includes identical project "and" `projectName` settings (this is what happens with JJB without this patch), like:

```xml
<project>demoJob/axis=$setting</project>
<projectName>demoJob/axis=$setting</projectName>
```

then it fails to run with current version(s) of the Copy Artifacts plugin.

The problem is that while the web interface displays the correct configuration for a fresh job created with JJB (Project Name: "demoJob/axis=$setting" and "Parameter filters" being empty), the "wrong" configuration is actually used (Project Name: "demoJob" and "Parameter filters: "axis=$setting"). The reason is that when you

<table>
<thead>
<tr>
<th>Author</th>
<th>Michael Prokop <a href="mailto:openstack.org@michael-prokop.at">openstack.org@michael-prokop.at</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit</td>
<td>183ccbe7912332afe862ea1ba05264daef0156</td>
</tr>
<tr>
<td>Parent(s)</td>
<td>b3a29bd7669d762214ba654c06ebca4a4d65ac1</td>
</tr>
<tr>
<td>Change-Id</td>
<td>le93316b135dc2b4547922062c7393ea7e5cc4e4c</td>
</tr>
</tbody>
</table>

**Jenkins check (1 rechecks)**
- gate-jenkins-job-builder-docs
- gate-jenkins-job-builder-pep8
- gate-jenkins-job-builder-python

- SUCCESS in 55s
- SUCCESS in 17s
- SUCCESS in 42s
- SUCCESS in 47s
Git review - Download

```
mika@heart ~$ git review -d 5785
Downloading refs/changes/85/5785/1 from gerrit
Switched to branch "review/michael_prokop/5785"
mika@heart ~$ %
```
Webinterface Gertty

Gertty: use gerrit without the web
Good review culture

- Meaningful response cycles
- Be friendly
- Include + integrate newbies/new employees

Source: https://xkcd.com/303/
Best Practices 1

keep reviews as small as feasible
(hint: also makes testing easier)
use Code Review also for **Infrastructure changes**! (IAC)
Best Practices 3

no direct pushes to production branch
Traceability via Branches

Branch/Tag

Environment/Release

“vagrant up $release”
“ID#XXXXX” in commit message to point to according issue/bug number
Challenges

Different time zones

Refactorings

New workflow

Added delays

Culture change
Warning

Laaaaaaaarge change?

“Uff, +2”

Tiny change?

“Please rename $this, quote the variable, the logic could be simplified like…”
*Cough*

No need to double check this change list, if some problems remain the reviewer will catch them.

No need to look at this change list too closely, I'm sure the author knows what he's doing.

Source: http://www.bonkersworld.net/code-reviews/
Resources

• “Expectations, Outcomes, and Challenges of Modern Code Review” [URL]
• “Characteristics of Useful Code Reviews: An Empirical Study at Microsoft” [URL]
• “On Rapid Releases and Software Testing” [URL]
• “Modern Code Reviews in Open-Source Projects: Which Problems Do They Fix?” [URL] and also the other “TestRoots Publications” [URL]
Testing
Tools
Testing Pyramid

- Unit Tests
- Integration Tests
- System
Goss

- [https://github.com/aelsabbahy/goss/](https://github.com/aelsabbahy/goss/)
- Serverspec-like tool for validation
- Golang → one single static binary
- RSpecish, nagios, json, JUnit + TAP output available
- Limited to Linux (so far)
<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addr</td>
<td>is reachable</td>
</tr>
<tr>
<td>command</td>
<td>exit status and outputs</td>
</tr>
<tr>
<td>dns</td>
<td>is resolvable</td>
</tr>
<tr>
<td>file</td>
<td>file exists, owner/perm, content</td>
</tr>
<tr>
<td>group</td>
<td>uid</td>
</tr>
<tr>
<td>package</td>
<td>is listening, listening ip</td>
</tr>
<tr>
<td>port</td>
<td>is running</td>
</tr>
<tr>
<td>process</td>
<td>Is running</td>
</tr>
<tr>
<td>user</td>
<td>uid, home, etc..</td>
</tr>
</tbody>
</table>
mika@osdc ~ % goss autoadd ssh
Adding Group to './goss.yaml':
ssh:
    exists: true
    gid: 103
Adding Package to './goss.yaml':
ssh:
    installed: true
    versions:
    - 1:6.7p1-5+deb8u2
Adding Process to './goss.yaml':
ssh:
    running: true
Adding Service to './goss.yaml':
ssh:
    enabled: true
    running: true
mika@osdc ~ % cat goss.yaml
package:
  ssh:
    installed: true
    versions:
    - 1:6.7p1-5+deb8u2
service:
  ssh:
    enabled: true
    running: true
group:
  ssh:
    exists: true
    gid: 103
process:
  ssh:
    running: true
mika@osdc ~ % goss validate goss.yaml
........
Total Duration: 0.066s
Count: 7, Failed: 0
mika@osdc ~ % goss add port 22
Adding Port to './goss.yaml':
tcp:22:
  listening: true
  ip:
    - 0.0.0.0
mika@osdc ~ % goss validate goss.yaml
........
Total Duration: 0.070s
Count: 9, Failed: 0
mika@osdc ~ % sed -i 's/tcp:22/tcp:23/' goss.yaml
mika@osdc ~ % goss validate --format tap goss.yaml
1..9
ok 1 - Group: ssh: exists: matches expectation: [true]
ok 2 - Group: ssh: gid: matches expectation: [103]
ok 3 - Service: ssh: enabled: matches expectation: [true]
ok 4 - Service: ssh: running: matches expectation: [true]
ok 5 - Process: ssh: running: matches expectation: [true]
not ok 6 - Port: tcp:23: listening: doesn't match, expect: [true] found: [false]
not ok 7 - Port: tcp:23: ip: doesn't match, expect: ["0.0.0.0"] found: [null]
ok 8 - Package: ssh: installed: matches expectation: [true]
ok 9 - Package: ssh: version: matches expectation: ["1:6.7p1-5+deb8u2"]
py.test

- Pytest.org
- Fixtures, Scopes, Monkeypatching
- JUnit + TAP output (hello Jenkins!)
- Plugins
- Example + starting point:
  - https://github.com/vincentbernat/lldpd/tree/master/tests/integration
  - http://www.slideshare.net/VincentBer nat/pytest-all-the-things
Package dependencies

• Dose-distcheck
• Are package dependencies/conflicts satisfiable?
  – dose-debcheck for Debian packages
  – dose-rpmcheck for rpm packages
  – dose-eclipsecheck for OSGi plugins
Puppet Testing

Puppet-lint

RSpec-puppet

Beaker

Puppet testing: see David Schmitt’s talk “Introduction to Testing Puppet Modules” at 2:15pm on Thursday at OSDC
Tests for everything ASAP!

avoid problems during commit/push time +
don't rely on historic/explicit knowledge →
new employees will have a hard time otherwise
Test systems to work on infrastructure without breaking production
Test infrastructure really for *everything*

testing a change for the *test* builds/scripts?
use *test* environment!
Related Resources

- „Advanced Testing with Go“ [URL]
- „System Testing with pytest and docker-py“ [URL]
- Book “How Google Tests Software”

Source: https://twitter.com/francesc/status/718604718294097920
Docker
Obligatory Docker Slide

Source: https://twitter.com/sadserver/status/718455853540487168
Our use cases

- Fast test cycle to avoid long feedback loops in CI/CD pipeline
- Share environment between developers and testing infrastructure
- Developers should be able to control testing infrastructure
Internal docker registry

Docker-registry
Docker-distribution
(implementation of Docker Registry HTTP API V2 for docker 1.6+)

Fast moving
Workflow

Every project ships its own Dockerfile(s), Jenkins identifies changes and **rebuilds** the docker images.

Image lifecycle is **tricky** though (when to build/destroy, naming + tagging conventions,...)
Infrastructure
Tooling
**Puppet related**

- Puppet environments + r10k
- Hiera
- Puppetdb + ansible
- Mcollective
- Puppet codebase from 2.7 to 4.3
- Puppetdashboard → puppetboard
- Own puppet modules → Puppet Forge
Monitoring?

Long running tests failing with ENODISKSPACE

automatic cleanups
(get rid of old artifacts, caches, ...) +
use appropriate monitoring
Monitoring

Metrics + Logging

what is taking long + is worth improving/investigating/....?
(ELK, Graylog, InfluxDB, Grafana,...)
Monitoring Software

Check-mk

Icinga 2 with puppet-icinga2
Vagrant/Packer + grml-debootstrap

Vagrant + Packer by HashiCorp
(Packer = replacement for Veewee, hit Debian/unstable today [URL])

Grml-debootstrap for building Debian (based) base boxes [URL]
Disaster recovery

- Daily fresh installation of nodes
- PIN-protected USB pen drive with sensitive data on it (e.g. hiera)
- iPXE + Grml ISO with netscript=... boot option to deploy
- netboot.xyz

Source: https://www.istorage-uk.com/product/datashur/
Antipatterns
Like a boss

Let's just fix this via SSH

The Definitive Guide

O RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
How to create yet another cronjob

The Definitive Guide

O RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
Manually setting up machines

The Definitive Guide

O RLY? @mikagrm1

Thx @ThePracticalDev + http://dev.to/rly
How to create checklists everyone hates

The Definitive Guide

© RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
Github has better uptime than us anyway!

Making your business depend on others
The Definitive Guide

O RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
Because junit isn't good enough for us!

How to report so no one can parse it
The Definitive Guide

O RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
The computer does the work anyway

How to rebuild stuff like a pro

The Definitive Guide

O RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
Successfully preventing configurability

The Definitive Guide

O RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
Because notifications disturb people!

How to get people to stare at Jenkins web

The Definitive Guide

O RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
Your application is a special snowflake

Excuses for not Writing Unit Tests

The Definitive Guide

O RLY?

@mikagrml

Thx @ThePracticalDev + http://dev.to/rly
Pain Points
Different people/teams have 
different use cases, understandings,...
of certain things (e.g. meaning + usage of
-1/-2/+1/+2 in Gerrit)

Fix via documentation!
Cultural differences

Fix via „Team Handbook“, like the one from Gitlab [URL]
Debian related 1

**Race-free** package info updates
AKA „Hashsum Mismatch Error“

Fix see [URL]
Avoid **Pre-Depends**, they impose harsh constraints on the package manager, consequently make upgrades harder + also cause more problems with piuparts.
The more sophisticated the systems become, the closer the **toolchain** needs stuff Debian has + uses (britney/dak/piuparts/nose/...
Mass changes?

Repository **locking** problem with e.g. reprepro :(
Structuring of hundreds/thousands of Jenkins jobs in `jenkins-job-builder` isn't easy, esp. the more exceptions you have
Unreliable tests?
Run them outside of the production pipeline! Use whitelists/blacklists to reach 100% coverage over time!
Matrix jobs in Jenkins are not always fun, use simple plain **freestyle Jenkins jobs** when possible
Availability

Downtime of Jenkins :( 

Lenient Shutdown
Zuul + Gearman
Jenkins 2.0 with Pipeline
Some architecture decisions are visible only after surviving a new release (Debian: wheezy→jessie)
Full-featured PPAs: bound together with Gerrit topics to share packages from different projects (new features, introducing new components/build-depends,...)
Avoid backwards compatibility nightmare

Release-specific settings into release-specific configurations/scripts/...
Takeaways
Take it home

- Automation is essential
- Configuration Management
- Traceability
- Code Review
- Tests + Testing
- Communication is important
- Bring devs and ops together ("devops")
Wishes || Questions?

@mikagrml
mika @ github
michael-prokop.at/blog/
prokop (at) grml-solutions.com