Continuous Delivery of your infrastructure

Christophe Vanlancker
@Carroarmato0
Christophe Vanlancker

- Internal operations and consulting
- Mentor
- Kris couldn’t make it so I ‘s/Kris/Christophe/g’
Today's Goals

- A reproducible way to deploy and upgrade /etc/puppet
- With 10+ environments in /etc/puppet/environments
- Automatically
- Fast
- Consistent
- Continuously
What's this devops thing anyhow?
C(L)AMS

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

*Damon Edwards and John Willis*

*Gene Kim*
devops (<) continuous delivery
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
Continuous Delivery

- Unit Test: Auto
- Platform Test: Auto
- Deliver to Staging: Auto
- Application Acceptance tests: Manual
- Deploy to Production: Auto
- Post deploy tests: Auto

Continuous Deployment

- Unit Test: Auto
- Platform Test: Auto
- Deliver to Staging: Auto
- Application Acceptance tests: Auto
- Deploy to Production: Auto
- Post deploy tests: Auto
How many times a day?

- 10 @ Flickr
- Deployments used to be pain
- Nobody dared to deploy a site
- Practice makes perfect
- Knowing you can vs constantly doing it
"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/
For years we've tolerated humans to make structural manual changes to the infrastructure our critical applications are running on. 

Whilst at the same time demanding those critical applications to go through rigid test scenarios. 

Who let this happen?
Infrastructure as Code

- Treat configuration automation as code
- Development best practices
  - Model your infrastructure
  - Version your cookbooks / manifests
  - Test your cookbooks / manifests
  - Dev/ test /uat / prod for your infra
- Model your infrastructure
- A working service = automated (Application Code + Infrastructure Code + Security + Monitoring)
Version all the things

No more excuses!

- Source code Application
- Source code Infrastructure
- Builds
- Tests
- Pipelines
- Scripts
- Documentation
- Monitoring scripts
A random project

```
[sdog@mine vagrant-graphite]$ ls
manifests modules README TODO Vagrantfile
[sdog@mine vagrant-graphite]$ tree -dL 2
.
├── manifests
│   └── hosts
└── modules
    ├── apache
    ├── collectd
    ├── graphite
    ├── jmxtrans
    └── logster
    └── statsd
        └── tattle
10 directories
```
Unless you understand Git Submodules

- Basic git,
- No extra tools required

Integrates with other projects too.
(No need for *-librarian etc ..)
Librarian Puppet / r10k

- Insert ugly shell script

- Even with this in place .. people can still hack on the PuppetMaster
Software Release management is not a solved problem
Librarian Puppet

- Hides complexity of submodules
- Easy if you use Forge Modules
  - Does anyone?
  - Do you trust the internet to be around?

- Librarian = Old English for “can't use submodules”
- And hmm... which customer uses which patched version again?
Continuous Integration

Continuous integration (CI) is the practice, in software engineering, of merging all developer working copies with a shared mainline several times a day. It was first named and proposed as part of extreme programming (XP). Its main aim is to prevent integration problems, referred to as "integration hell" (WikiPedia)

Does the app you are deploying still work?
Did you break your puppet / chef code?
Jenkins

- Open Source Continuous Integration Server
- A zillion plugins (400)
- Have developers build stable and deployable code
- Test Infra code
Jenkins Pipeline
What's in your Pipeline ?
A pipeline

- Checkout code
- Syntax
- Style
- Code Coverage
- Tests
- Build
- More Tests
- Package
Syntax and Style

- Initially,
  all code, all the time
- Now,
  only the changed code
- Why not in post Commit Hooks?
Package all the things
Artifacts:

- Tested artifacts that go through a pipeline
  application code,
  Infra code
  metadata
  tests
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
Jordan Sissel is a Hero!
# packaginlove

**Input**
- rpm
- npm
- rubygem
- deb
- python
dir(ectory)

**Output**
- rpm
- deb
- solaris
It's not really packaging

- It's an immutable branch
- It's a tracable release artefact
https://github.com/vStone/jenkins-puppet-scripts

- Tests
- Packages full tree in
  /etc/puppet/environments/$environment/

Introduction

This repo contains various scripts that get used throughout our Jenkins setup.
A pipeline

- Checkout code
- Syntax
- Style
- Code Coverage
- Tests
- Build
- More Tests
- Package
- Upload to Repo
Repository Management

- Pulp
  - Pro: Mirroring Love
  - Con: Mongo, Stability, .deb

- Aptly (deb only)

- Prm (missing snapshot features for .rpm)
Repository Management

Package Flow

<table>
<thead>
<tr>
<th>The Internet</th>
<th>Local Mirror on pulp01</th>
<th>Daily / Development Repos</th>
<th>UAT Repos</th>
<th>Production Repos</th>
</tr>
</thead>
<tbody>
<tr>
<td>CentOS OS</td>
<td>Mirror CentOS OS</td>
<td>Clone Daily CentOS OS</td>
<td>Clone UAT CentOS OS</td>
<td>Mirror Prod CentOS OS</td>
</tr>
<tr>
<td>CentOS Extras</td>
<td>Mirror CentOS Extras</td>
<td>Clone Daily CentOS Extras</td>
<td>Clone UAT CentOS Extras</td>
<td>Mirror Prod CentOS Extras</td>
</tr>
<tr>
<td>CentOS Contrib</td>
<td>Mirror CentOS Contrib</td>
<td>Clone Daily CentOS Contrib</td>
<td>Clone UAT CentOS Contrib</td>
<td>Mirror Prod CentOS Contrib</td>
</tr>
<tr>
<td>CentOS CentOSPlus</td>
<td>Mirror CentOS CentOSPlus</td>
<td>Clone Daily CentOS CentOSPlus</td>
<td>Clone UAT CentOS CentOSPlus</td>
<td>Mirror Prod CentOS CentOSPlus</td>
</tr>
<tr>
<td>CentOS Updates</td>
<td>Mirror CentOS Updates</td>
<td>Clone Daily CentOS Updates</td>
<td>Clone UAT CentOS Updates</td>
<td>Mirror Prod CentOS Updates</td>
</tr>
<tr>
<td>EPEL</td>
<td>Mirror EPEL</td>
<td>Clone DailyUPStream</td>
<td>Clone UAT UPStream</td>
<td>Mirror Prod UPStream</td>
</tr>
<tr>
<td>jPackage</td>
<td>Mirror jPackage</td>
<td>Clone DailyUPCust</td>
<td>Clone UAT DollyUPCust</td>
<td>Mirror Prod DollyUPCust</td>
</tr>
<tr>
<td>Puppetlabs</td>
<td>Mirror Puppetlabs</td>
<td>Clone DailyUPBuild</td>
<td>Clone UAT DollyUPBuild</td>
<td>Mirror Prod DollyUPBuild</td>
</tr>
<tr>
<td>Upstream Repo X</td>
<td>Mirror Upstream Repo X</td>
<td>Clone DailyUPBuild</td>
<td>Clone UAT DollyUPBuild</td>
<td>Mirror Prod DollyUPBuild</td>
</tr>
</tbody>
</table>

Repository Management Diagram

- CentOS OS
- CentOS Extras
- CentOS Contrib
- CentOS CentOSPlus
- CentOS Updates
- EPEL
- jPackage
- Puppetlabs
- Upstream Repo X

Flow:
- Mirror
- Clone
- Upload
- Repackage
A pipeline

- Checkout code
- Syntax
- Style
- Code Coverage
- Tests
- Build
- More Tests
- Package
- Upload to Repo
- Deploy on Test
Repos are SLOW

- Createrepo is slow.
- Pulp is slow

- Bypass repos, upload straight to appropriate PuppetMaster
- Upload to repo for rebootsrapping
A pipeline

- Checkout code
- Syntax
- Style
- Code Coverage
- Tests
- Build
- More Tests
- Package
- Upload to Repo
- Deploy on Test
- Check Puppetruns
- Check Monitoring
Testing = Monitoring

- Deploy a host,
- Add it to the monitoring framework
- Add collection tools
- Add check definitions
- Update the monitoring tool config

FULLY AUTOMATED
A pipeline

- Checkout code
- Syntax
- Style
- Code Coverage
- Tests
- Build
- More Tests
- Package

- Upload to Repo
- Deploy on Test
- Check Puppetruns
- Check Monitoring
- Promote to UAT
Jenkins Promotion

From: jenkins@nuits.eu
To: roxelapluie@nuits.eu, mediaalsa@nuits.eu, caronmatou@nuits.eu
Subject: [Mediaalsa] puppet-mediaalsa-infra-dev-promote-me - Build #461 - Successful!
Date: Thu, 23 Jan 2014 15:08:48 +0100 (CET)

The last changes in the mediaalsa puppet tree have been deployed to infra-dev.
Check the latest changes in https://jenkins.dev.nuits.eu/requests/puppet-mediaalsa/
To promote to UAT or PROD in API, PROD, click here: https://dev.nuits.eu/requests/puppet-mediaalsa/

Changes:
[caronmatou01 Refactor stor01 and stor02 to be in sync and use drbd flag for enabling replication

regards,
Mr Jenkins

Mediaalsa mailing list
Mediaalsa@nuits.eu
https://lists.nuits.eu/msgaaw/listinfo-mediaalsa

![Jenkins UI screenshot]

Jenkins Promotion

Promotions
- ...
Pipelines

- Lots of them
- Similar ones, but not identical ones
- One project = different deployment targets
- People move teams expect same patterns
- Mostly unmaintained
Dirty Clickers

Enter an item name
MyNewProject

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

If you want to create a new item from other existing, you can use this option:

Copy from
pupnp

OK
Complex Pipelines
Scaling Pipelines

- Create a Pipeline,
- For job in Pipeline
  - Create new Job Based on OldJob
- Update One Job
- Never refactor the rest
Generating Jenkins Pipelines

- Template the XML
- Put it in Puppet
  - Worked for stable pipelines
  - Kinda
- XML gets rewritten on the fly
JenkinsJobBuilder

- First Usable Attempt
- Python Based
- Openstack Community
- Limited Functionality
- Little Adoption / Openstack Abandoned it
PipelineDSL

- Jenkinsfile
- Inside a repo
- Groovy
- Limited functionality
- Initially buggy
- Popular for Easy tasks
  - (straight, no splits etc)
Jenkins Job DSL

- Groovy
- Flexible
- Well Supported
- Suitable for more complex Pipelines
Seedjobs

- Groovy
- Git
- Rebuild jobs on commit
- Projects in folders
Larger CI Stacks

- Generate Pipelines / Jobs based on config files,
- Build libraries
  - CheckoutJob
  - DeployJob
  - PackageJob
- Use Groovy / JobDSL to generate PipelineDSL
Testing Multiple Versions

- Initial stage tests code on multiple versions
  - e.g. current puppet version
  - Next puppet version
  - Bleeding Edge version
- Only current version breaks build
- Goal = get all versions green
Problems solved

- One job per task, no reuse of jobs with different parameters
- All Jobs are in sync, update one = update all
- Centrally managed jobs (git)
Stop Clicking, Write Code
Contact

Kris Buytaert
Kris.Buytaert@inuits.be

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

Inuits

Duboistraat 50
2060 Antwerpen
Belgium
891.514.231
+32 475 961221