

Automation w/ Puppet & a Path to Private Hybrid Cloud

by Andrew Ludwar

Lead, UNIX/Linux Infrastructure

Agenda

- Automation
 - What is Puppet?
 - Where does it apply?
 - Gotcha's / best practices
- Path to Private Hybrid Cloud
 - Server provisioning

What is Puppet?

- Automation? - ... in the right context, yes!
- Orchestration? - No!
- Configuration Management? - Yes!
- Fancy scripting? - Yes!
- Marketing fluff? - Couldn't be more true!

What is Puppet?

- Automation... let's define it further
 - Drift management for server/VM “sprawl”
 - Repeatable, reliable provisioning
 - Programmatic approach to infrastructure
- What it's NOT
 - Orchestration language
 - Silver bullet to automation
 - Exists below layer 3 in OSI

OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example	Central Device Protocols
Application (7) Serves as the window for users and application processes to access the network services.	End User layer Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management	User Applications SMTP
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the “Translator” for the network.	Syntax layer encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • Character Set Translation	JPEG/ASCII EBDIC/TIFF/GIF PICT
Session (5) Allows session establishment between processes running on different stations.	Synch & send to ports (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.	Logical Ports RPC/SQL/NFS NetBIOS names
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	TCP Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	FILTERING TCP/SPX/UDP Routers IP/IPX/ICMP
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	Packets (“letter”, contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting	
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	Frames (“envelopes”, contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control	Switch Bridge WAP PPP/SLIP Land Based Layers
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	Physical structure Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts	Hub

What is Puppet?

- Configuration management tool & scripting language
- State Described Language (SDL)
- Client/server architecture
 - puppet client agent --> puppet master server
- Uses SSL certificates; encrypted communication
- OS Platform agnostic (**mostly**, true for Linux, Solaris/Windows not quite as mature)
- Abstracted syntax
 - apache installed = true; instead of
yum install apache && chkconfig apache on && service apache start
apt-get apache, etc.

What is Puppet?

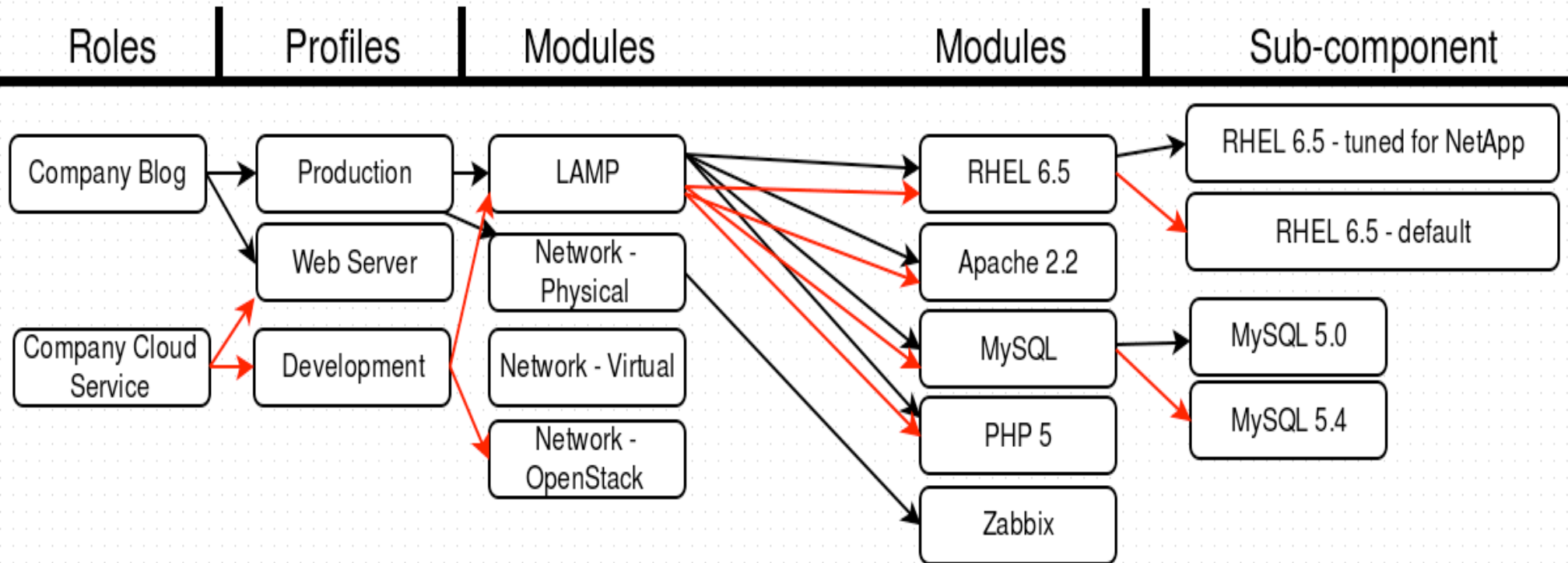
- Reintroduces traditional modular programming concepts into “Puppet Modules” (functions, methods)
- Maintains state of servers based on:
 - Server/OS facts collected into a catalog called Factor/Heira
 - Unique identifiers of hosts (FQDN usually)
 - Node groups and other classification techniques (ENC - External Node Classifier)
 - Hardcoded classification and modules (will haunt you eventually)
- Assumes server is on the network (TCP/IP, not TFTP/BOOTP)
- Agents check in typically every 15 minutes
- Config changes should follow a CI/CD workflow

What is Puppet?

- Enterprise & open source versions for both puppet software and the puppet code written with it
- Written by developers for developers (enterprise consumable?)
- Large reason for the uptake in the “DevOps” culture and processes
- Becoming highly adopted & integrated in the open source ecosystem (& some proprietary)
 - RedHat Satellite 6, RedHat OpenStack, OpenShift, etc.
 - Canonical Landscape, JuJu, MaaS, etc.
 - VMware vCloudAC, Amazon EC2, Google Compute, F5, Cisco, etc.

Managing Puppet

- Roles --> Profiles --> Modules --> Sub-components
 - A node can only have one role at a time (think in terms of the business)
 - A role includes one or more profiles to define that type of server
 - A profile includes and manages modules to define a logical technical stack
 - Modules manage resources and should only be responsible for managing aspects of the component they are written for



What does Puppet mean for me?

- Am I going to have to learn a new scripting language?
 - Potentially, but the syntax is very English & UNIX like
file_permissions { root : root };
apache_installed = true; apache_web_root = '/var/www/html'
 - Bulk of work is classifying systems, identical to kickstart/
jumpstart. You wouldn't necessarily have to write the kickstart
code, but you should be able to kick a server.
- Is automation going to put me out of a job?
 - Not going to happen. Less time is spent on low-value tasks,
and more time being in control of your systems.
 - Automation has been around for years
 - Experience is key

Experience is key...

Ted, the factory manager, was having difficulty in his assembly line. Ted hired his friend Brian, a senior engineer, to see if he could help identify the problem area. Brian walked up to the control panel, watched the line for 10 minutes, pressed one button, and resolved the issue. Ted was thrilled, and told him to send an invoice.

The bill arrived, for \$10,000. Ted, a little shocked, asked for a breakdown. Brian sent another invoice, indicating a \$1 charge for pressing the button, and \$9,999 for knowing what button to push.

What makes sense for your company?

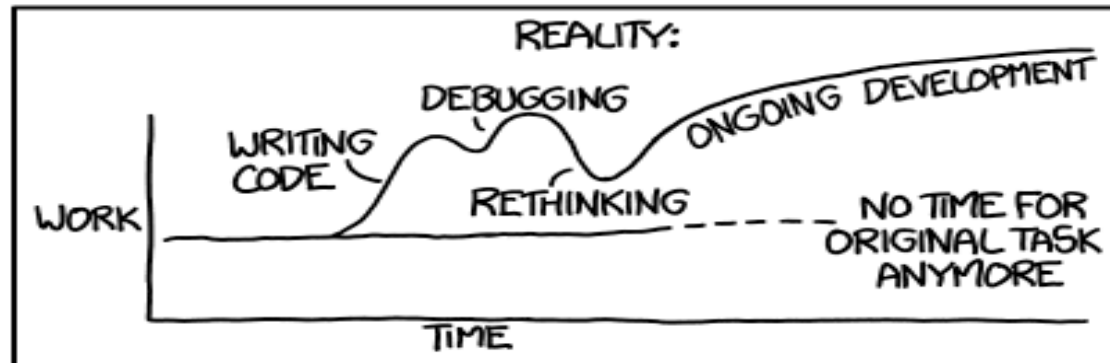
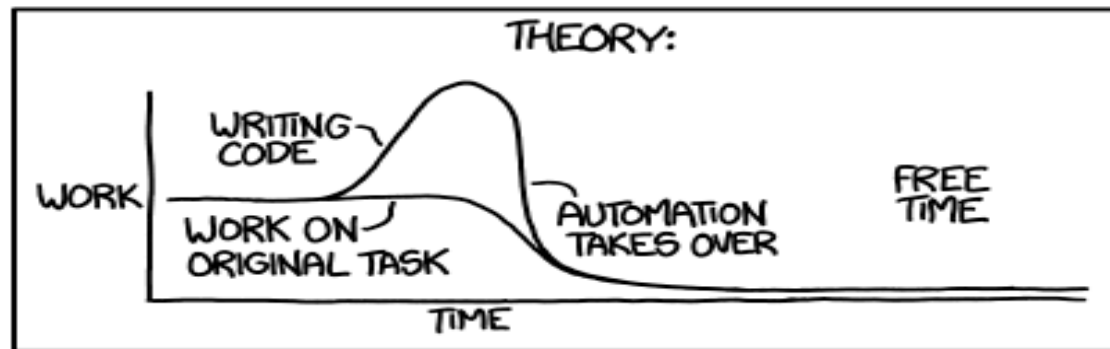
- Puppet enterprise, puppet open source, or leverage Satellite?
- Integration with other tools? (Microsoft System Center?)
- Enterprise tested modules from vendor or roll your own? Hybrid?
- What % of company resources spent being administrators vs developers?
- Code maintenance costs? Future portability?
- Integration with other teams?

What to be careful of?

AUTOMATION

⏪ < PREV RANDOM NEXT > ⏩

"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"



⏪ < PREV RANDOM NEXT > ⏩

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1319/](http://xkcd.com/1319/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/AUTOMATION.PNG](http://imgs.xkcd.com/comics/automation.png)

What to be careful of?

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
	5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
	30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
	1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
	1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
	6 HOURS				2 MONTHS	2 WEEKS	1 DAY
	1 DAY					8 WEEKS	5 DAYS

What to be careful of?

- Indecision comes at a high cost
- Think courageous failures!
- Unconventional thinking



“In a time of rapid change, standing still is the most dangerous course of action.”

What to be careful of?

- Automation scope/feature creep
 - “Puppet can do anything!!”
- Not an orchestration/workflow language
- Traditional thinking vs “DevOps” thinking
 - > Robust first --> Speed first
 - > Speed second --> Robust second



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Unconventional questions from others...

The image features a central white signpost with four dark grey arrows pointing outwards. Each arrow points to a company name and a descriptive sentence. The background is a solid green color.

Company	Description
Uber	The world's largest taxi company, owns no vehicles.
Facebook	The world's most popular media owner, creates no content.
Alibaba	The most valuable retailer, has no inventory.
Airbnb	The world's largest accommodation provider, owns no real estate.

Something interesting is happening.
TOM GOODWIN

wetp@int
creative digital solutions

WetpaintMENA

Path to Private Hybrid Cloud

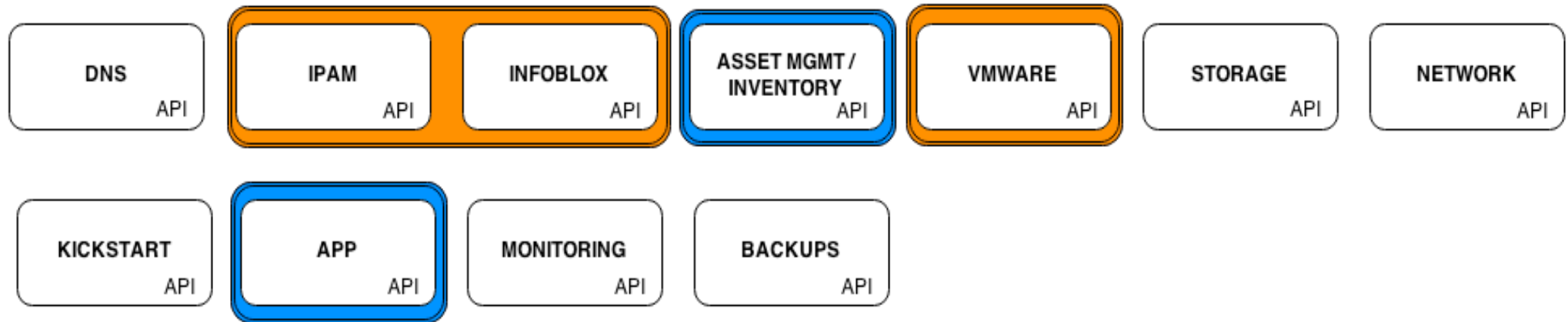
MY REPORT COMES TO THE
CONCLUSION THAT CLOUD TECHNOLOGY
IS OF NO USE TO THIS COMPANY.
I'LL UPLOAD IT TO DROP BOX SO
YOU CAN TAKE A LOOK AT IT.



© D. Fletcher for CloudTweaks.com

Server Provisioning Artifacts:

Traditional IT: (Unresponsive) Avg Completion Time: 10 h / 3.5 weeks 4h / 2 weeks



Responsive IaaS: "Private Cloud"



Avg Completion Time: < 5 mins

Cloud Management Tool: "Single Pane of Glass"



Responsive IaaS: "Public Cloud"



Avg Completion Time: < 5 mins

To Summarize:

- Automation and Orchestration are separate concepts with separate toolsets
- Courageous failures will provide tremendous value
- Unconventional thinking should be front of mind

Additional Resources

- Puppet Architecture - Craig Dunn, PuppetLabs Engineer
 - <http://www.craigdunn.org/2012/05/239/>
 - <http://www.slideshare.net/PuppetLabs/roles-talk>
- Puppet Training Classes
 - [PuppetLabs Fundamentals](#)
 - [PuppetLabs Practitioner \(Advanced\)](#)
- Guide to the Open Cloud
 - <http://www.linuxfoundation.org/publications/linux-foundation/guide-to-the-open-cloud> (PDF)
- RHEV vs VMware Pricing
 - <http://www.redhat.com/en/files/resources/en-rhev-vs-vmware-vsphere-competitive-pricing-review-11717847.pdf>



Thank You