Puppet

9. November 2010

Zusammenfassung

Wie Niklaus lernte, Puppet einzusetzen. Ein Erfahrungsbericht.

Niklaus Giger, Wieshoschet 6, CH-8753 Mollis

1 Ziele

Aufsetzen/Dokumentation so zu gestalten, dass andere Medelexis OC interessiert sind, mitzuarbeiten. Dazu unter GitHub das Projekt elexis-adminer öffnet, wo u.a. auch der aktuelle Stand dieser Datei zu finden ist.

Merkblatt, wie/wo gewisse Sachen von Puppet dokumentiert sind.
Meine Ziele/Anforderungen an Puppet dokumentieren.
Fortschritt/Entscheidungen dokumentieren.
Lyx -X HTML

2 Einstieg

2.1 Warum?

Ich habe ja vor Jahren es mit cfengine versucht und bin davon abgekommen. Warum?

- Aufwand/Ertrag für mein kleines heterogenes (PPC, Intel) Heimnetzwerk zu gross.
- Debian-Konfiguration änderte sich im Laufe der Jahre. Defaults wurden besser, meine alten Rezepte funktionierten nicht mehr und machten gewisse Teile unbrauchbar. Dieses Problem kann nur angegangen werden, indem man die Änderungen jeweils nach einem Upgrade des betroffenen Pakets kritisch hinterfrägt.
- Zentrales Logging der Änderungen/Abläufe wurde nicht richtig implementiert.

- Für gewisse Probleme wäre die Benutzung einer "echten" Script-Sprache praktisch.
- Nicht Idem/Potent. Häufig mit cfengine <timestamp> gearbeitet, damit Dateien nicht dauernd neu erstellt wurden.

2.1.1 Auslöser

2010 habe ich für Elexis-Server und andere mit Hilfe von simple-cdd kundenspezifische Installation von GNU/Debian-Linux durchgeführt, welche von einigen wenigen Text-Dateien getrieben wurden (den profiles von simple-cdd). Aufwand/Ertrag waren gut, sobald man akzeptiert, dass die Testing-Distribution von Debian hin und wieder zu Abbrüchen führt.

Bei Installation von cosre (Mai 2010) und erstem Server (bbuechel, November 2010) festgestellt, dass diese Installation machbar sind, wegen HW-Problemen (z.B. zusätzliche HD-Disk in grub-inkompatibler Reihenfolge) scheitern können. Nachher kommen im Laufe der Zeit neue Anforderungen auf, welche keine Neu-Installation erfordern sollten. Beispiele dafür sind neue Applikationen (z.B. ein Praxis-Wiki, neue Benutzer, Samba).

Deshalb nach einem Tool gesucht, dass automatisch die Konfiguration eines Server à jour halten kann. Nach Alternativen zu cfengine gesucht und Puppet gefunden.

2.2 Einarbeitung

Doku unter http://docs.puppetlabs.com/ quer durchgelesen, um Konzepte zu verstehen.

Für Debian gibt es unter http://www.debian-administration.org/articles/526 eine gute Anleitung, wie man rasch ein kleines Netzwerk mit Puppet startklar macht. Falls man Lenny auf dem Server installiert hat, wird puppet am besten via debian-backports installiert.

Dabei suchte ich v.a. auch nach Vorlagen wie gesamte Sites gemanaged werden können und fand sie z.B. in folgenden Orten

- http://projects.puppetlabs.com/projects/puppet/wiki/Complete_Configuration. Web-Server der alles veröffentlicht, inkl. Beispiele für sich selber und seine Eltern.
- Die ETH-Zürich scheint es zu gebrauchen: http://git.sans.ethz.ch/
- Diverse Module/Applikationen findet man unter http://forge.puppetlabs.com/

Dieses Dokument begonnen.

http://www.debian-administration.org/articles/526

Dort beklagt sich auch jemand, dass puppet viel zuviel Memory braucht (400 MB)

http://projects.puppetlabs.com/projects/1/wiki/Using_Thin_Nginx

2.3 Vorstellungen

github (oder ähnliches) gebrauchen und Projekt/Absicht bei Medelexis-OC & Elexis-Entwickler-Treffen ankündigen.

Beginnen mit autofs-Konfiguration (das kann ich nämlich gebrauchen und ist etwas trickreich). 4hudson (export/uses)

Checkout von git-repository

etckeeper einbauen

Danach Benutzer-Konfiguration mit folgenden Anforderungen:

- Benutzer/Passwort-Kombination von nicht public lesbaren Ort holen. Oder gleiches Default-Passwort für alle Benutzer. Dazu Extlookup benutzen (Siehe Release Notes 2.6.1)
- Yubico-Key verwenden (Optional)
- Beim Aufsetzen Passwort muss sofort geändert werden.

Puppet-Master verwenden (ngiger.dyndns.org zu Beginn, für Demo-Zwecke github)

Report generieren und ablegen.

Wie die SSH-Zertifikate von Server, Client, Benutzer, Firewall (OpenVPN) sicher verwalten. Integration in simple-cdd (z.B. Puppet, OpenSSH-Server, OpenVPN installieren und Zertifikate vom Puppet-Master holen oder auf CD mitbringen). OpenVPN immer/nur auf Verlangen verbunden zu PuppetMaster.

DHCP oder fixe IP/hostname/domainname/gateway verwenden (simple-cdd)

Postgres/MySql Hot-Backup verwenden.

Auf Home-PCs wie gs,mm,transtec diejenigen Pakete/Sachen mit Puppet verwalten, welche nicht mit den Defaults von Debian abgedeckt werden. Wichtige Pakete (etckeeper, git, cups, autofs, rsync) per Default installieren.

http ://rdiff-backup.nongnu.org/ with puppet-module!

Bridge für virtuelle Maschinen offerieren.