



Software-Repositories an zentraler Stelle verwalten

Manager

Wer freie Software einsetzt, hat oft mit mehreren Software-Repositories zu tun, die die gewünschte Software führen. Bei größeren Rechnerlandschaften gilt es dann, die Repository-Informationen auf allen Systemen zu hinterlegen und aktuell zu halten. Pulp hilft alternativ dazu beim zentralen Repository-Management. Thorsten Scherf

Pulp versetzt den Admin in die Lage, viele verschiedene Software-Repositories an einer zentralen Stelle zu konsolidieren. Pulp kann dabei auf unterschiedlichste Quellen zurückgreifen, beispielsweise funktioniert die Abfrage eines regulären Yum-Servers genauso problemfrei wie der Zugriff auf das Red Hat Network oder auf Software-Pakete, die über einen Webserver zur Verfügung stehen. Pulp agiert hierbei nicht als Proxy zwischen den Clients und den verschiedenen Repositories, sondern spiegelt die Pakete lokal und aktualisiert sie auch nach einem festgelegten Intervall. Für ausreichend Platz ist also im Vorfeld zu sorgen.

Pulp kann jedoch auch einzelne Repositories auf externe Server auslagern. Das Ganze nennt sich in der Pulp-Terminologie dann Content-Delivery-Server. Hierbei routet der zentrale Server bestimmte Client-Anfragen einfach an externe Server, die dann die Anfrage beantworten. Somit ist der Aufbau von geografisch verteilten Systemen sehr leicht möglich. Mithilfe von fein granulierten Zugriffsregeln stellt Pulp sicher, dass Clients nur bestimmte Repositories abfragen dürfen. Die Authentifizierung von Benutzern kann dabei sogar für einen LDAP-Server passieren, was also auch den Einsatz in sehr großen Umgebungen sicherstellt.

Im Bereich Reporting kann die Software ebenfalls punkten. So gibt es nicht nur eine Historie, die anzeigt, welches Client-System wann auf welches Repository zugegriffen hat, auch der aktuelle Paketstand der Clients lässt sich darstellen. Somit ist sehr leicht herauszubekommen, wenn ein wichtiges Update auf einem System noch nicht installiert ist. Für experimentierfreudige Admins ist es sicher auch interessant zu erwähnen, dass Pulp eine Rest-API mitbringt, mit der sich viele der anfallenden Aufgaben sehr bequem für Skripte automatisieren lassen. Die komplette Architektur einer Pulp-Umgebung zeigt **Abbildung 1**.

Installation

Für den Einsatz von Pulp stehen derzeit fertige RPM-Pakete für Fedora, Red Hat Enterprise Linux sowie für CentOS zur Verfügung **[2]**. Der Zugriff auf den Source-Code ist über ein entsprechendes Git-Repository **[3]** möglich. Nach dem Download der passenden Yum-Repository Konfigurationsdatei gelingt die Installation des Servers wie gewohnt mittels Yum (**Listing 1**).

Nach der Initialisierung der MongoDB-Datenbank ist der Server dann einsatz-

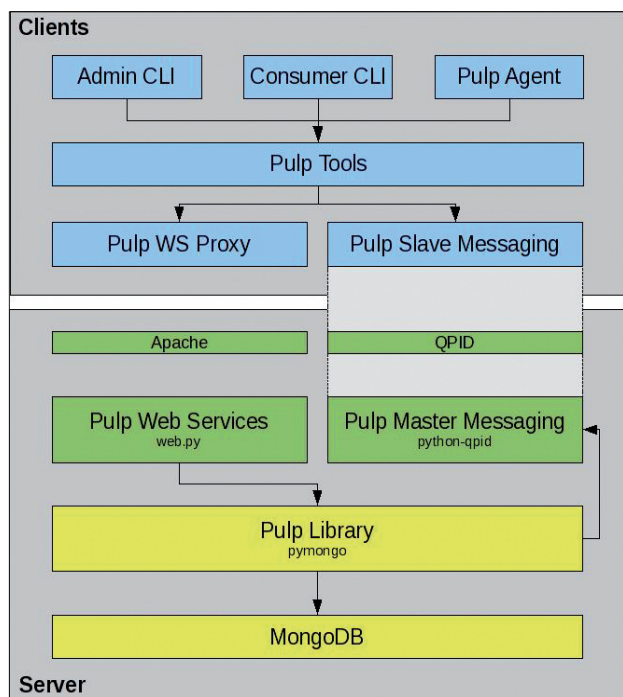


Abbildung 1: Bei Pulp arbeiten viele verschiedene Komponenten zusammen.

bereit. Es ist darauf zu achten, dass in der Konfigurationsdatei »/etc/pulp/pulp.conf« der passende Servername aufgeführt ist und dass Client-Systeme Zugang zu den Ports 5672, 5674 und 443 haben. Die Client-Systeme benötigen ebenfalls Zugang zum bereits erwähnten Pulp-Repository. Am einfachsten ist es, dieses direkt bei der Installation der Systeme mit anzugeben. Findet eine automatisierte Installation der Systeme mittels Kickstart statt, so lässt sich die entsprechende Client-Software gleich mit installieren und konfigurieren. Die Systeme sind dann nach der Installation direkt einsatzbereit und eine manuelle Konfiguration entfällt.

Das Client Paket heißt »pulp-client« und ist im Idealfall auch auf dem Server zu installieren. Dies ist für Admin-Arbeiten notwendig. Wichtig ist, dass die Konfigurationsdatei der Clientsoftware, »/etc/pulp/client.conf« entsprechend angepasst wird, sodass sämtliche Referenzen auf den Server über den FQDN (Fully qualified domainname) stattfinden (Listing 2). Dies ist wichtig, da ansonsten die Zertifikatsverifizierung des Servers fehlschlägt. Im Zertifikat steht der FQDN des Servers. Versucht der Client nun über einen anderen Namen auf diesen zuzugreifen, so beschwert sich die Software über einen falschen Namen. Anschließend ist der

Agent des Clients entsprechend zu starten (»service pulp-agent start«).

```
# pulp-admin -u admin -p admin user ↵
update --username admin --password
Enter new password for user admin:
Re-enter new password for user admin:
Successfully updated [ admin ]
```

Jede administrative Aktion auf dem Server erfordert eine vorherige Authentifizierung. Damit nun nicht bei jedem Befehl der Benutzername und das Passwort abgefragt werden, bietet es sich an, sich einmal auf dem Server anzumelden und die Benutzer-Credentials (in Form eines Benutzer-Zertifikates) unterhalb von »~/pulp« zu speichern. Über das Kommando »auth logout« lassen sich diese wieder ins Nirvana befördern, beispielsweise wenn die Anmeldung mit anderen Credentials notwendig ist:

```
# pulp-admin auth login -u admin -p ↵
password
User credentials successfully stored at ↵
[/home/pulp-user/.pulp]
```

Neue Benutzer lassen sich über das Kommando »user« hinzufügen und auch wieder löschen oder modifizieren. Um dem Benutzer nun bestimmte Rechte zuzuweisen, muss man das Konto über das Kommando »permission grant« entsprechend modifizieren und mit den notwendigen Rechten versehen. Anstatt einzelnen Benutzern Rechte zuzuweisen,

besteht auch die Möglichkeit, diesen bestimmte Rollen zuzuweisen und dann den Rollen die gewünschten Rechte zu geben. So ist das Gruppieren von einzelnen Benutzern mit ähnlichen Rechten einfacher zu handhaben. Bestehende Rollen zeigt Pulp über das Kommando »role show« an, welche Rechte eine Rolle besitzt, erfahren Sie über das Kommando »role info« (Listing 3).

Repositories und Distributionen

Um nun den Server mit Daten zu versorgen, muss der Administrator die gewünschten Repositories synchronisieren. Pulp versteht unter einem Repository die Ansammlung von Software-RPMs, Update-RPMs sowie Installationsdateien (Distributionen) wie beispielsweise Kernel und Initrd. Das Anlegen und anschließende Synchronisieren eines Repositories erfolgt durch den folgenden Kommandozeilenaufwurf:

Listing 1: Installation mit Yum

```
01 # yum install pulp
02 # service pulp-server init
03 # service pulp-server start
04 # chkconfig pulp-server on
05 @KE
```

Listing 2: Servername in der Konfiguration

```
01 # grep ipa1 /etc/pulp/client.conf
02 host = ipa1.virt.tuxgeek.de
03 baseurl = https://ipa1.virt.tuxgeek.de/pulp/repos
04 keyurl = http://ipa1.virt.tuxgeek.de/pulp/gpg
05 ksurl = http://ipa1.virt.tuxgeek.de/pulp/ks
```

Listing 3: Rechte per Rollen zuweisen

```
01 # pulp-admin role list
02 +-----+
03 Available Roles
04 +-----+
05 super-users
06 consumer-users
07
08 # pulp-admin role info --role super-users
09 +-----+
10 Role Information for super-users
11 +-----+
12 Name                super-users
13 Users                admin
14 Permissions:
15 /                   CREATE, READ, UPDATE, DELETE,
EXECUTE
16 @KE
```

```
# pulp-admin repo create --id Fedora14
-x86_64 --feed yum:http://download.fedora
project.org/pub/fedora/linux/releases/14/
Everything/x86_64/os/
Successfully created repository [ example ]
# pulp-admin repo sync --id Fedora14-x86_64
```

Den Status der Synchronisation zeigt Pulp mit dem Kommando »repo status --id Fedora14-x86_64« an. Daneben existieren weitere recht nützliche Kommandos, mit

Listing 4: Distributionen

```
01 pulp-admin distribution list
02 +-----+
03      List of Available Distributions
04 +-----+
05
06 Id          Fedora14-x86_64
07 Description Kickstart Trees for
   Fedora14-x86_64
08 URL         http://ipal.virt.tuxgeek.de/
   pulp/ks/released/fedora/14/x86_64/os
```

Listing 5: Übersicht der Repositories

```
01 # pulp-client repo list
02 +-----+
03      List of Available Repositories
04 +-----+
05
06 Id          Fedora14-x86_64
07 Name        Fedora14-x86_64
08 FeedURL     http://download.fedoraproject.
   org/pub/fedora/linux/releases/14/Everything/x86_64/
   os/
09 FeedType    yum
10 Arch        noarch
11 Sync Schedule 10 0 * * *
12 Packages    10
13 Files        0
14 Distributions None
15 Publish     True
16 Clones       []
17 Groups      None
18 Filters      []
```

Listing 6: Pulp erzeugt eine neue Yum-Datei

```
01 #
02 # Pulp Repositories
03 # Managed by Pulp client
04 #
05 [Fedora14-x86_64]
06 name = Fedora14-x86_64
07 enabled = 1
08 sslverify = 0
09 gpgcheck = 1
10 baseurl = https://ipal.virt.tuxgeek.de/pulp/repos/
   pub/fedora/linux/releases/14/Everything/x86_64/os
11 @KE
```

denen der Admin das Repository anpassen kann. So besteht beispielsweise die Möglichkeit, Filter zu erzeugen, sodass bestimmte Pakete nicht synchronisiert werden. Auch das automatische Aktualisieren eines Repositories ist über die Option »--schedule« möglich. Diese ist entweder direkt beim Anlegen des Repositories mit anzugeben, oder aber nachträglich. Das Kommando »repo update« gestattet es, einzelne Konfigurationseinstellungen nachträglich zu verändern. Eine Übersicht sämtlicher Distributionen zeigt Pulp mittels »distribution list« an (Listing 4). Bei der Installation eines neuen Systems lässt sich die URL zu der gewünschten Distribution dann gleich mit angeben, sodass das Installationsprogramm die notwendigen Kernel-, Initrd- und Installer-Dateien direkt von hier downloaden kann.

Ist auf einem Client bereits das Paket »pulp-client« installiert und entsprechend konfiguriert, muss man ihn im nächsten Schritt beim Pulp-Server registrieren. Das übernimmt der Aufruf von »pulp-client consumer create --id client1«. Dieser Schritt lässt sich natürlich genauso wie die Installation des Pakets automatisieren, beispielsweise als Teil eines Kickstart-Profils. Mittels »pulp-client repo list« erhält der Client dann im Anschluss eine Liste aller zur Verfügung stehenden Repositories des Servers (Listing 5). Die Zeile »Sync Schedule« zeigt hier im Cron-Format an, zu welchem Zeitpunkt das Repository zu aktualisieren ist. Im Beispiel also täglich um 0:10.

Angebunden

Damit der Client nun auch Pakete aus diesem Repository installieren kann, muss der Administrator ihn zuerst an das entsprechende Repository binden. Dafür bietet Pulp den Befehl »pulp-client consumer bind --repo-id Fedora14-x86_64«. Pulp legt hierbei dann unterhalb von »/etc/yum.repos.d« eine neue Yum-Konfigurationsdatei »pulp.conf« an, die dann auf das neue Repository des Servers zeigt (Listing 6). Bindet der Client sich an weitere Repositories des Servers, landen diese alle in der gleichen Yum-Konfigurationsdatei. Ein »yum repolist« auf dem Client bestätigt nun, dass dieser Pakete aus dem soeben konfigurierten

Yum-Repository des Pulp-Servers installieren kann.

Pulp-Admin bietet eine History für jeden Client-Zugriff. Der Aufruf von »pulp-admin consumer info --id ipa2« zeigt sowohl die Konfiguration der einzelnen Repositories wie auch die Installation von einzelnen Paketen hieraus. Sehr nützlich ist auch die Möglichkeit, einzelne Clients zu Gruppen zusammenzufassen. Die Option hierfür lautet »consumergroup create«. Anstatt eine bestimmte Aktion nur auf einem einzelnen Client anzustoßen, führt Pulp beim Einsatz einer Gruppe diese auf sämtlichen Mitgliedern aus.

Soll beispielsweise das Paket »pidgin« auf allen Clients der Gruppe Desktop-Clients installiert werden, würde ein Aufruf von »pulp-admin package install -n pidgin --consumergroupid desktop-clients« genügen. Voraussetzung hierfür ist natürlich, dass sämtliche Mitglieder der Gruppe Zugriff auf das Repository besitzen, in dem sich das zu installierende Paket befindet.

Fazit

Mit Pulp steht ein umfangreiches Management-Tool für Software-Repositories zur Verfügung. Wer keine grafische Oberfläche benötigt, der kann mittels Pulp eine Vielzahl von Client-Systemen verwalten. Sind diese geografisch voneinander entfernt, bietet Pulp mit dem Content-Delivery-Server auch dafür eine Lösung. Die umfangreichen Reporting-Funktionen helfen, den Überblick über die Systeme zu behalten.

Wer Spaß an Automatisierung hat, den wird es freuen, dass Pulp auch eine REST-API anbietet. Die Daten, die Pulp hierüber zurückliefert beziehungsweise erwartet, liegen im JSON-Format [4] vor. Sämtliche modernen Skriptsprachen, wie beispielsweise Perl oder Python, bieten dafür einen Encoder/Decoder an. (ofr) ■

Infos

- [1] Pulp-Projektseite: [\[https://fedorahosted.org/pulp/\]](https://fedorahosted.org/pulp/)
- [2] Download: [\[https://fedorahosted.org/pulp/wiki/UGInstallation\]](https://fedorahosted.org/pulp/wiki/UGInstallation)
- [3] Git-Repository: [\[git://git.fedorahosted.org/git/pulp.git/\]](https://git.fedorahosted.org/git/pulp.git/)
- [4] JSON-Projektseite: [\[http://www.json.org/\]](http://www.json.org/)