



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Informatikdienste
ID Basisdienste

ETH Zürich
Client Delivery
Marcus Möller
Bengt Giger
STB H 20.2
Stampfenbachstrasse 69
8092 Zürich

Telefon +41 44 632 44 65
Telefax +41 44 632 15 20
www.id.ethz.ch

Satellite

Lifecycle Management

Die Gruppe Client Delivery der Informatikdienste stellt Ihnen zur Verwaltung von Red Hat Produkten einen sogenannten Satellite Server zur Verfügung.

Mit dessen Hilfe ist es möglich, den vollständigen Lifecycle eines Red Hat Servers oder Clients abzubilden.

Das vorliegende Dokument beschreibt den Prozess der Installation von Updates, der vom Satellite ausgeht. Zur Vereinfachung wird in dieser Anleitung immer von einem Server-OS ausgegangen. Es ist allerdings genauso möglich Workstations mit den beschriebenen Verfahren zu pflegen.

Login

Sie erreichen den Satellite Server über folgende URL: <https://id-sat-prd.ethz.ch>

Sollten Sie noch nicht über einen Account auf dem Satellite Server verfügen, schicken Sie bitte eine Email an: rhn@id.ethz.ch.

Lifecycle Management

Satellite 6 erlaubt eine genau Festlegung des Software- und Konfigurationsstandes für jeden Host. Einige wichtige Begriffe in diesem Zusammenhang:

Lifecycle Management

- Content View: Zusammenfassung von Paketen, Konfiguration uvm., eingefroren zu einem bestimmten Zeitpunkt (Snapshot). Ausnahme: die Content View *Default Organization View* enthält immer die aktuellsten Pakete.
- Lifecycle: Stufen, welchen eine Content View zugeordnet werden kann. Eine Content View wird in einer Version eingefroren und der ersten Stufe zugeordnet (z. B. Test), dann an die nächste Stufe weitergereicht werden (QS), und schliesslich an die dritte (Produktion).

Ein Host kann an eine der Lifecycle-Stufen gehängt werden und sieht dann Updates einer Version erst dann, wenn sie seine Stufe des Lifecycles erreicht.

Abbildung 1 zeigt für jede Stufe eine eigene Version der Content View: die Entwicklung arbeitet an der Programmversion 1.5b unter der Lifecycle-Version v17, Version v16 ist im Test, v15 in der Qualitätssicherung und die Produktion nutzt v14 mit der Programmversion 1.2:

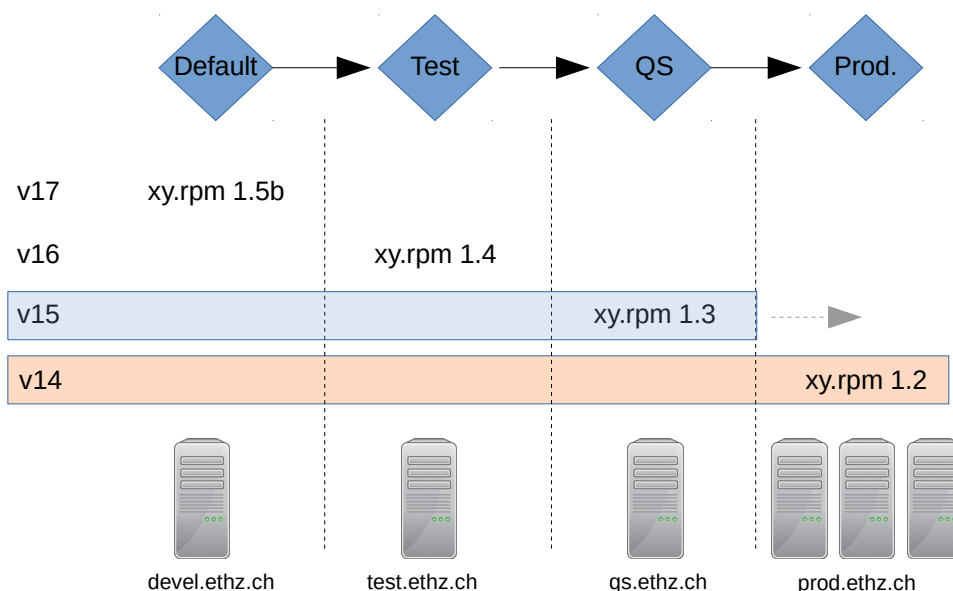


Illustration 1: Ursprünglicher Stand des Lifecycle

Wird nun z. B. die Qualitätssicherung abgeschlossen und die Produktion soll auf die Programmversion 1.2 angehoben werden, wird die Lifecycle-Version v15 der Content View eine Stufe von QS nach *Prod* weitergeschoben:

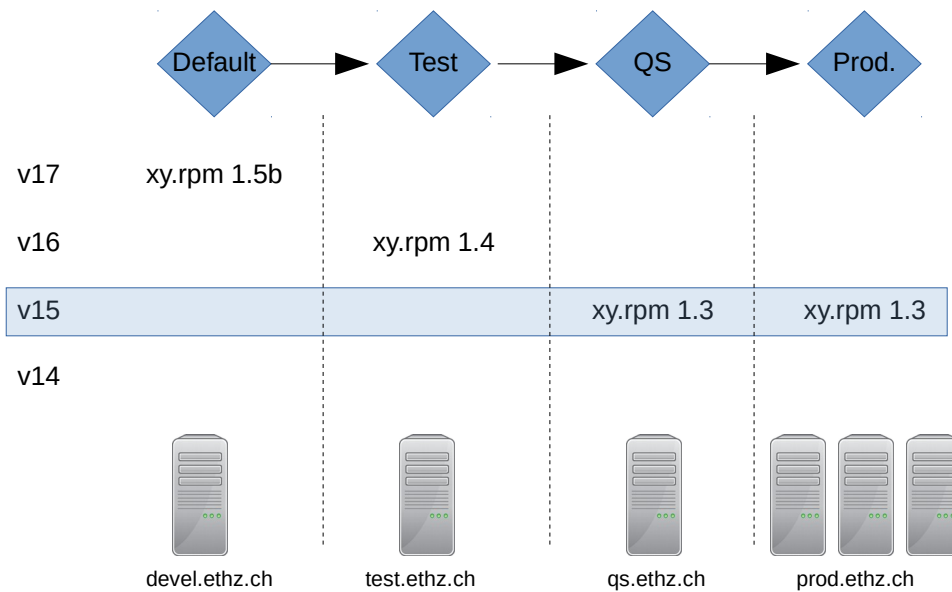


Illustration 2: Lifecycle nach Promotion

Konfiguration

Für die Nutzung muss

1. der Lifecycle definiert,
2. Hosts den Stufen des Lifecycles zugeordnet,
3. eine Content View erstellt,
4. eine Version der Content View erstellt,
5. und die Version durch die Stufen vorgebracht („promoted“) werden

Definition eines Lifecycles

Lifecycles sind unter dem Menu «Content / Lifecycle» zu finden. Mit «New Environment Path» wird ein Formular für die Angaben der ersten Stufe («Environment») geöffnet; ein Lifecycle selber hat keinen Namen. Weitere Stufen können mit «Add New Environment» hinzugefügt werden:

	id-lnx-exam-tst	id-lnx-exam-prd
Content Views	1	1
Content Hosts	1	4

+ Add New Environment

Die Bezeichnungen der Stufen eines Lifecycles und ihre Zahl sind frei wählbar, für das System haben die Namen keine Bedeutung. Wichtig ist das hinter der Lifecycle-Definition stehende

Da in einer Organisation mehrere Lifecycles erstellt werden können, empfiehlt sich in der Praxis eine sprechende Benennung wie im gezeigten Beispiel.

Content View erstellen

Über das Menu «Content / Content Views» und die Schaltfläche «New» wird eine neue Content View erstellt. Nennen Sie sie entsprechend des Schemas „organisation-funktion“, z. B. für die Organisation ID-CD „idcd-bootserver“.

Eine Content View enthält Verweise auf:

- Produkte: welche Produkte sollen in ihren Versionsständen festgehalten werden. Die notwendigen Produkte können über den Tab «Products / Add» hinzugefügt werden.
- Puppet-Module: hier ist ebenfalls ein Tab «Add» vorhanden. Nach der Wahl wird zusätzlich eine Liste der verfügbaren Versionen zur Auswahl angeboten, mit *Latest* als Möglichkeit, immer die aktuellste Version in die Content View zu integrieren.
- Pakete und Paketgruppen: über «Yum Content / Filter» können Pakete und Paketgruppen gewählt werden, die installiert sein müssen.

Darüber hinaus sind weitere Komponenten in einer *Composite Content Views* kombinierbar, welche im nächsten Kapitel besprochen werden und für die praktische Anwendung von zentraler Bedeutung sind.

Auch hier gilt: vergeben Sie klare Namen, wie <gruppe>-<funktion>, wie beispielsweise idcd-examserver.

Content View versionieren

Um im nächsten Schritt einen Host einem Lifecycle-Zustand zuordnen zu können, muss die Content View erst einmal versioniert werden. Verwenden Sie dazu in der eben erstellten Content View die Schaltfläche «Publish New Version». Dadurch wird von jeder in der Content View enthaltenen Komponente (Software-Pakete, Konfigurationsmodule usw.) eine Kopie angelegt. Dies kann je nach Umfang der Content View eine Weile dauern.

Diese Kopie bildet nun den Zustand *Library* dieser Content View, dies ist der aktuellste Stand aller Content Views eines Lifecycles. Ihr zugewiesene Hosts aktualisieren sich bis genau auf diesen Stand.

Content View voranbringen

Eine Version kann nun entlang des Lifecycles weiter geschoben werden. Dazu öffnen Sie dazu die Liste der Content Views über das Menu «Content / Content Views». Das folgende Bild zeigt den folgenden Zustand:

1. *Library*, das Environment für die Entwicklung, steht auf Version 15
2. Test- und Produktionssysteme sind auf Version 13

Version	Status	Environments	Content	Description	Actions
Version 15.0		Library	49045 Packages 7008 Errata (733 ▲ 3063 ✨ 1539 📦) 10 Puppet Modules	Puppet sat_roota...	<input type="button" value="Promote"/> <input type="button" value="Remove"/>
Version 14.0			49011 Packages 7008 Errata (733 ▲ 3063 ✨ 1539 📦) 10 Puppet Modules	Alsa kernel issue ...	<input type="button" value="Promote"/> <input type="button" value="Remove"/>
Version 13.0		id-lnx-exam-tst id-lnx-exam-prd	48709 Packages 6967 Errata (732 ▲ 3028 ✨ 1532 📦) 10 Puppet Modules	Security fixes	<input type="button" value="Promote"/> <input type="button" value="Remove"/>
Version 12.0			48687 Packages	Security: samba, ...	<input type="button" value="Promote"/>

Um nun die Version 15 für das Environment *id-lnx-exam-tst* aktiv zu schalten und für die Testsysteme verfügbar zu machen, klicken Sie in der Zeile für die Version 15 auf *Promote*. Nach Abschluss des Vorgangs steht das Environment *id-lnx-exam-tst* in der obersten Zeile neben *Library*.

Zuordnung eines Hosts / einer Host-Gruppe

Öffnen Sie nun eine Host-Gruppe oder einen einzelnen Host. In der Auswahl «Lifecycle Environment» erscheinen nun die von Ihnen definierten Lifecycles, womit Sie den Host einem bestimmten Zustand zuteilen können. Von nun an wird der Software-Stand dieses Hosts mit demjenigen des Lifecycle Environments verglichen, Updates für diesen Host werden dann angezeigt, wenn das Lifecycle Environment neue Updates anzubieten hat, dh. wenn die

zugehörige Content View aktualisiert wurde. Beachten Sie, dass ein System in einem Lifecycle nur in diesem Fall anzeigt! Auch ein System auf Stand *Library* sieht erst Updates, wenn diese *Library* mit einer neuen Version aktualisiert wurde.

Neuinstallation in einen definierten Zustand

Soll in einer Umgebung mit Lifecycle-Verwaltung ein produktives System neu installiert werden, muss es normalerweise im gleichen Zustand aufgesetzt werden wie die restlichen Systeme.

Erstellen oder bearbeiten Sie hierzu einen Activation Key so, dass Lifecycle Environment und Content View korrekt eingetragen sind. Erstellen oder bearbeiten Sie die Felder für Lifecycle Environment und Content View einer Host Group analog, falls nicht schon vorhanden. Wählen Sie dann den vorbereiteten Activation Key aus; Sie werden feststellen, dass nur passende Keys vorgeschlagen werden, die die gleichen Einstellungen tragen wie die Host Group.

Die installierte Software eines dieser Host Group zugeordneten Systems wird nun nach der Installation den Versionsstand haben, der dem gewählten Lifecycle Environment entspricht.

Strategie für die Praxis: Composite Content View

Für die praktische Anwendung sind monolithische Content Views (CV) nur in speziellen Fällen geeignet. Das Problem: eine CV enthält sämtliche RPM-Pakete und Puppet-Module. Wenn nun eine kleine Änderung z. B. an einem Puppet-Modul eingebracht werden soll, muss dazu eine neue Version der CV erstellt werden. Diese enthält aber auch zwangsweise die aktuellsten Versionen aller RPMs, die nun eine neue Version bilden, die erneut getestet werden muss und möglicherweise inkompatibel mit Anwendungen usw. ist.

Um diese Abhängigkeiten aufzubrechen, gibt es die Composite Content Views (CCV). Diese fassen bestehende Content Views zusammen, ohne selber Produkte oder Konfigurationsmodule zu enthalten. Die eingebundenen CVs werden wie zuvor beschrieben durch Versionierung auf einen definierten Stand gebracht. In der CCV wird für jede eingebundene CV ein Versionsstand ausgewählt, und für die CCV eine übergreifende Version erstellt.

Anwendungsbeispiel

Für ein grundlegendes Beispiel-Setup werden drei CVs erstellt. Es empfiehlt sich, die CVs mit einem Prefix zu benennen, so dass sie später bequem in der Liste der CVs gefiltert werden können.

CV 1 „id-lnx-exam-os“: Basissystem

Diese CV enthält das Produkt für *Red Hat Server 7*. Gegebenenfalls sind auch Produkte wie *Optional* oder *EPEL* hier am richtigen Ort:

Lifecycle Management

id-lnx-exam-os

Publish New Version

Select Action

Content Views > id-lnx-exam-os > Yum Repositories

Details Versions **Yum Content** File Repositories Puppet Modules Docker Content OSTree Content History

Repository Selection

List/Remove Add

Filter... Search Remove Repositories

0 of 1 Selected

<input type="checkbox"/>	Name	Product	Last Sync	Sync State	Content
<input type="checkbox"/>	Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server	Red Hat Enterprise Linux Server	2018-04-06 00:00:35 +0200	Success	18263 Packages 2606 Errata

100 per page Showing 1 - 1 of 1 << < 1 of 1 > >>

CV 2 „id-lnx-exam-katello“: Katello-Agent

Der Katello-Agent ist für die Kommunikation mit dem Satellite-Server notwendig. Seine Version ist abhängig von derjenigen des Satellite, er muss häufiger und unabhängig vom restlichen System aktualisiert werden können. Die CV enthält darum nur das Produkt „Satellite Tools“ in der aktuellsten Version:

id-lnx-exam-katello

Publish New Version

Select Action

Content Views > id-lnx-exam-katello > Yum Repositories

Details Versions **Yum Content** File Repositories Puppet Modules Docker Content OSTree Content History

Repository Selection

List/Remove Add

Filter... Search Remove Repositories

0 of 1 Selected

<input type="checkbox"/>	Name	Product	Last Sync	Sync State	Content
<input type="checkbox"/>	Red Hat Satellite Tools 6.3 for RHEL 7 Server RPMs x86_64	Red Hat Enterprise Linux Server	2018-04-06 00:00:30 +0200	Success	65 Packages 1 Errata

100 per page Showing 1 - 1 of 1 << < 1 of 1 > >>

CV 3 „id-lnx-exam-puppet“: Puppet-Konfigurationsmodule

Diese CV enthält nur die Puppet-Module des Configuration Managements.

CCV: „id-cd-Inx-exam“: Composite View

Eine CCV bietet keine Option um Produkte oder Puppet-Module hinzuzufügen, nur CVs werden angeboten. Achtung: nur publizierte CVs sind sichtbar, es müssen also von den vorgängig vorgestellten CVs erst mindestens eine Version publiziert werden. Ein Propagieren der einzelnen CVs entlang eines Lifecycles ist jedoch nicht notwendig.

id-Inx-exam Publish New Version
Select Action ▾

Content Views > id-Inx-exam > Content Views

Details Versions **Content Views** History Tasks

List/Remove Add

Content Views for id-Inx-exam

Filter... Search ▾ Remove Content Views

0 of Selected

<input type="checkbox"/>	Name	Version	Environment	Description	Content
<input type="checkbox"/>	id-Inx-exam-katello	Latest (Currently 1.0)	Library		1 Repositories 0 Puppet Modules
<input type="checkbox"/>	id-Inx-exam-os	Latest (Currently 1.0)	Library		3 Repositories 0 Puppet Modules
<input type="checkbox"/>	id-Inx-exam-puppet	Latest (Currently 3.0)	Library		0 Repositories 15 Puppet Modules

Die CCV wird nun wie ab Seite 4 beschrieben versioniert, entlang eines Lifecycles propagiert und Hosts/Hostgruppen zugeordnet.

Versionieren von CCVs

Die erste getestete und freigegebene Version des CCVs lässt sich so darstellen:

CCV Version 1		
CV 1	CV 2	CV 3
Vers. 1	Vers. 1	Vers. 1

Erscheint nun eine neue Version des Katello-Agent, und das Update muss unbedingt eingespielt werden, dann wird als erstes eine neue Version des CV 2 (id-lnx-exam-katello) erstellt:

CCV Version 1		
CV 1	CV 2	CV 3
Vers. 1	Vers. 1	Vers. 1
	Vers. 2	

Die übergeordnete CCV ist davon noch nicht betroffen, sie verweist weiterhin auf Version 1 von CV 2. Erst mit einer neuen Version der CCV wird die neue Software für die zugeordneten System sichtbar und kann installiert werden:

CCV Version 2		
CV 1	CV 2	CV 3
Vers. 1	Vers. 1	Vers. 1
	Vers. 2	

Da die Version der CV 1 mit dem Basis-OS unverändert bleibt, kann mit einem *yum update* keine unerwünschte neue Version eines Softwarepakets irrtümlich installiert werden.

Analog kann nun auch die CV mit den Konfigurationsdaten aktualisiert werden, ohne dass davon ein Softwarepaket betroffen ist. Eine weitere Version des CCV sieht nun so aus:

CCV Version 3		
CV 1	CV 2	CV 3
Vers. 1	Vers. 1	Vers. 1
	Vers. 2	Vers. 2

Es ist auch möglich, bei der Erstellung einer neuen CCV-Version einen älteren Stand einer CV zu wählen, um z. B. auf eine vorherige Konfiguration zurück zu gehen:

CCV Version 4		
CV 1	CV 2	CV 3
Vers. 1	Vers. 1	Vers. 1
	Vers. 2	Vers. 2

Bei Software-CVs ist hier Vorsicht geboten, da bereit installierte Pakete mit neueren Versionen nicht zurückgerollt werden und es möglicherweise Probleme mit Abhängigkeiten geben kann.

Strukturierung der CVs

Es bietet sich an, die folgenden Produkte in einzelne CVs einzuteilen:

- Basis-OS
- Katello-Agent
- Konfigurationen (Puppet)
- Software aus eigenen Repositories wie
 - Applikationen
 - TSM Backup-Client
 - Monitoring-Client

Inkrementelles Patchen

Es ist möglich, einzelne neue Patches anzuwenden, ohne dass gleich eine komplett neue Version einer Content View erstellt werden muss – die dann auch andere Errata enthalten würde. Dies ist über den Menüpunkt «Content / Errata» möglich.

Hier sind alle Errata zu sehen, die noch für Hosts in einer Organisation anwendbar sind. Das gewünschte Erratum wird hier ausgewählt:

Errata

All Repositories Search Applicable Installable Apply Errata

1 of 7 Selected

<input type="checkbox"/>	Errata ID	Title	Type	Content Host Counts	Updated
<input type="checkbox"/>	FEDORA-EPEL-2019-fb3b39cbf2	python-beautifulsoup4-4.4.1-1.el7	Product Enhancement Advisory - None	2 Applicable, 2 Installable	7/10/19
<input type="checkbox"/>	FEDORA-EPEL-2019-7a40a554a4	qpid-proton-0.28.0-1.el7	Product Enhancement Advisory - None	19 Applicable, 5 Installable	7/10/19
<input type="checkbox"/>	RHSA-2019:1619	Important: vim security update	Security Advisory - Important	1 Applicable, 0 Installable	6/26/19
<input checked="" type="checkbox"/>	FEDORA-EPEL-2019-f505f7d2fb >	python-epel-rpm-macros-3-25.el7 python-rpm-macros-3-25.el7 python34-3.4.10-2.el7	newpackage - None	6 Applicable, 0 Installable	6/24/19
<input type="checkbox"/>	RHBA-2019:1331	GNOME bug fix and enhancement update	Bug Fix Advisory - None	19 Applicable, 9 Installable	6/4/19

Mit *Apply Errata* werden die Systeme angezeigt, auf die das Erratum angewendet werden kann. Diese Liste kann auch bzgl. der Lifecycle Environments gefiltert werden, womit z. B. alle Produktionssysteme ausgewählt werden können:

Apply to Content Hosts

id-lnx-exam-prd Search Only show content hosts where the errata is currently installable in the host's Lifecycle Environment. Cancel Next

0 of 4 Selected

<input type="checkbox"/>	Name	OS	Environment	Content View
<input type="checkbox"/>	lxam-a-01.ethz.ch	RHEL Server 7.6	id-lnx-exam-prd	id-lnx-exam
<input type="checkbox"/>	lxam-a-02.ethz.ch	RHEL Server 7.6	id-lnx-exam-prd	id-lnx-exam
<input type="checkbox"/>	lxam-b-01.ethz.ch	RHEL Server 7.6	id-lnx-exam-prd	id-lnx-exam
<input type="checkbox"/>	lxam-b-02.ethz.ch	RHEL Server 7.6	id-lnx-exam-prd	id-lnx-exam

100 per page Showing 1 - 4 of 4 << < 1 of 1 > >>

Da der Satellite gemerkt hat, dass diese Systeme eine Content View benutzen und nicht direkt aktualisiert werden können, bietet er im nächsten Schritt an, eine inkrementelle Version der Content View zu erstellen:

Lifecycle Management

Apply Errata

[Errata](#) » [Select Content Host\(s\)](#) » [Confirm](#) 

One or more of the selected Errata are not Installable via your published Content View versions running on the selected hosts. The new Content View Versions (specified below) will be created which will make this Errata Installable in the host's Environment. This new version will replace the current version in your host's Lifecycle Environment. To install these errata immediately on hosts after publishing check the box below.

Content View	Version	Environment	Host Count
▼ id-lnx-exam * Components: id-lnx-exam-os 49.0	68.1 • 49.1	id-lnx-exam-prd	4

* These marked Content View Versions are from Composite Content Views. Their components needing updating are listed underneath.

Apply Errata to Content Hosts immediately after publishing.

Sowohl für die Basis- als auch für die Composite View werden neue Versionen erstellt. Danach kann das Erratum angewendet werden; optional kann die Aktualisierung direkt nach der Erstellung der Content View gestartet werden.