



PUZZLE ITC
changing IT for the better

Tübix 2018

Linux Server- und Desktop- Lifecycle-Management

oder: `${SPERRIGER_TITEL}` v201806101

Lukas Kallies,
Senior System Engineer

 @LuBeka

Agenda

9. Juni 2018

1. Intro
2. Grundlagen
3. Workshop

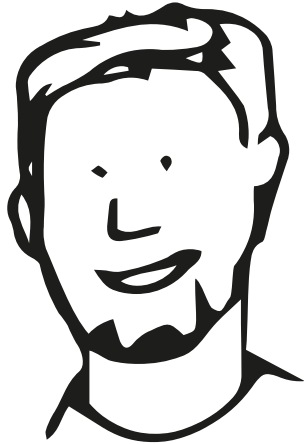
The top corners of the slide feature decorative geometric shapes. On the left, there are overlapping triangles in shades of light blue, teal, and dark blue. On the right, there are overlapping squares and triangles in shades of light blue, teal, and dark blue.

1

Intro

about:Lukas Kallies

Senior System Engineer bei
Puzzle ITC im Bereich Linux,
Lifecycle- und Configuration-
Management



Hochschule Furtwangen (Dipl. Inform. (FH))

science + computing ag

Hochschule Furtwangen (M.Sc.)

HINT AG

Puzzle ITC GmbH

about:Puzzle ITC

Gegründet 1999 in Bern

Software- und Technologie-Unternehmen

Offenen Technologien und Open Source Software



99

Belpstrasse 37
Bern



9

Neugasse 10
Zürich



@puzzleitc

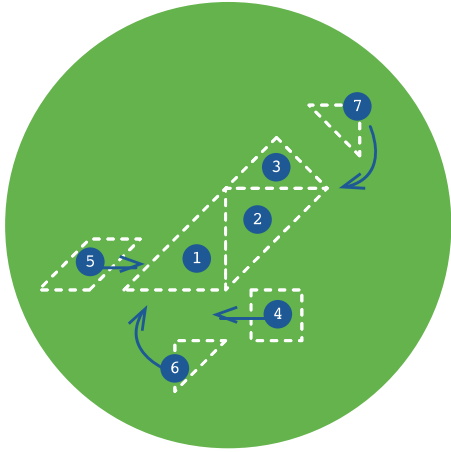


/puzzle



Workshop Mai 2018

about:Lebenszyklusverwaltung



Plan



Build



Run

Lebenszyklusverwaltung (im Kontext)

- Planen
- Entwickeln
- Bereitstellen
- Konfigurieren
- Verwalten
- Zurückbauen



- Hardware, Virtuelle Maschinen und Container
- «Zuhause» und Cloud
- Linux Hosts (Red Hat, SUSE, Debian, Ubuntu, ...)
- Software
- Subscriptions

2

Grundlagen

LCM - Katello Komponenten

Basis



Config Management Plugins



ANSIBLE

Lifecycle Plugins



Produkt

RED HAT
SATELLITE

Begriffe

Product

A₁

Errata

Lifecycle Environment
(LCE)

Activation Key

F₄

Standard Operating
Environment

Subscription

L₁ N₁ V₄

Content View
(CV)

Subscription

- Keine Lizenz
- Bei Red Hat:
 - Keine Versionsbindung
 - Support über mehrere Level
 - Bug Fixes, Sicherheitsupdates, Updates, Upgrade
 - Hard- und Softwarezertifizierung
 - Fachwissen
 - Schutz gegen geistige Eigentumsansprüche (Open Source Assurance Program)

Product

- Gruppierung von Repositories
- Verknüpfung von Subscriptions (Hersteller und eigene)
- Faustregel: pro (Repository) GPG-Key ein Product

Lifecycle Environment (LCE)

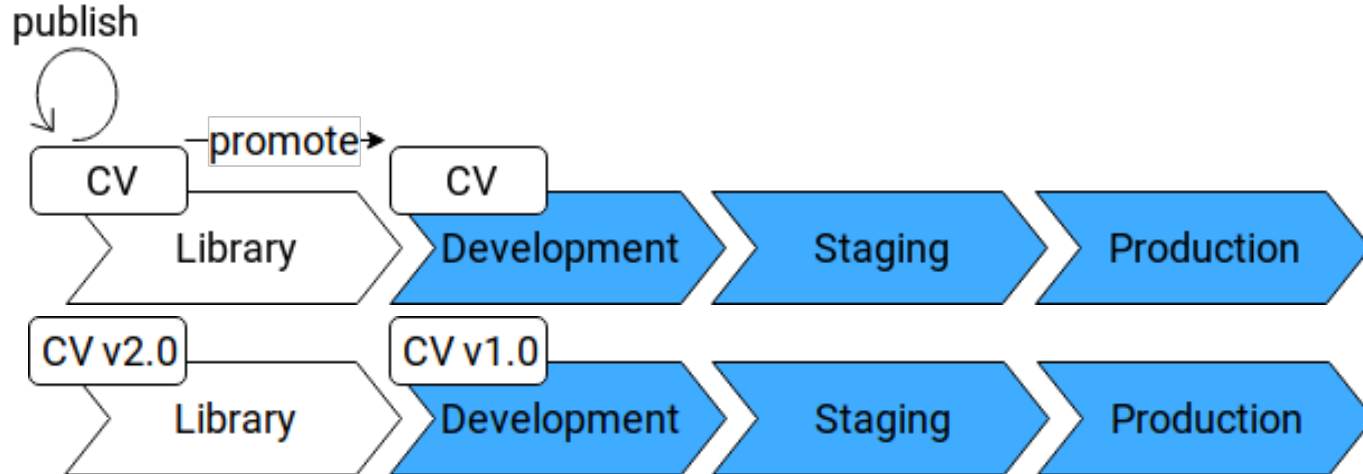
Item: Lifecycle Environment

Items: Lifecycle Environment Path



Content View (CV)

1. Versionierte Sicht auf eine Kollektion von Repositories
2. Ihrerseits kombinierbar in Composite Content Views



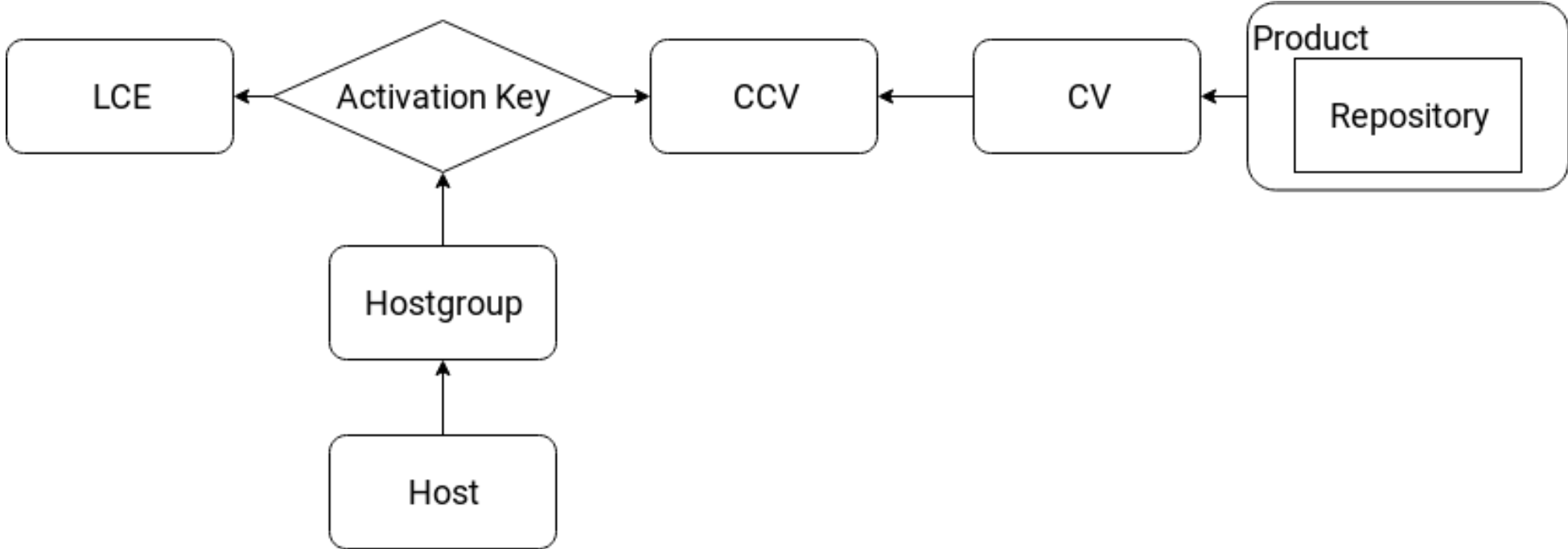
Errata

- «Korrekturverzeichnis»
- Sammlung von Updates, Bug Fixes und Security Patches, die ein spezifisches Problem beheben

Activation Key

- Definieren Eigenschaften eines Hosts bei der Registration
 - Subscriptions
 - Verfügbare Products und Repositories
 - Das Lifecycle Environment und die Content View
 - Eine Host Collection (!=Hostgroup) Mitgliedschaft

Begriffe im Kontext



Standard Operating Environment

Red Hats Vorschlag zur Vereinheitlichung einer Betriebsumgebung mit Satellite 6
u.a.

- Topologie/Organisations-Abbildung
- Naming

SOE Dokument: <https://access.redhat.com/articles/1585273>

Configuration Management

- Puppet
 - foreman-installer (<https://github.com/theforeman/kafo>) ist selbst Puppet-basiert
 - (noch) tiefe Integration von Puppet in Katello
 - Foreman startete als External Node Classifier (ENC) für Puppet (+PXE)
- Ansible
 - Inventory Quelle
 - «Play Ansible roles» Möglichkeit
 - Reporting

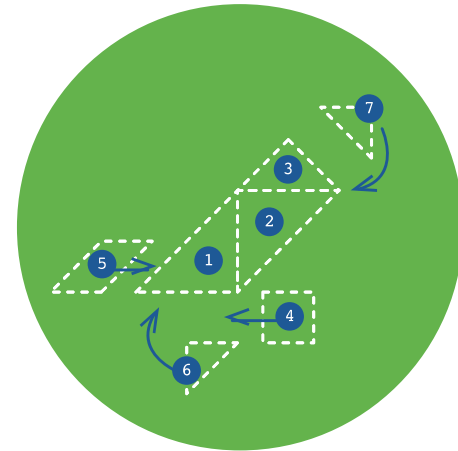


3

Workshop

LukeX.de/workshop/lcm

Unter oben gennanter URL sind Folien, Links und Updates zu finden.



Aufbau

- Wie ist der Workshop aufgebaut?
 - Es werden verschiedene Themen und Aufgaben gestellt
 - Nach der Aufgabenstellung wird ein möglicher Lösungsweg skizziert
 - Teilweise CLI, teilweise UI

Setup

<https://github.com/theforeman/forklift.git> @703e34f

Vagrant 2.0.2 - Ansible 2.5.1 - centos7-katello-3.6 (centos/7
@1804.02)

```
vagrant plugin install vagrant-hostmanager  
LC_ALL=en_US.UTF-8 vagrant up centos7-katello-3.6  
vagrant ssh centos7-katello-3.6 -c 'ip --brief --color addr show'  
  
# foreman-installer -v --scenario katello --disable-system-checks \  
  --foreman-admin-password changeme
```

hammer

```
vagrant ssh centos7-katello-3.6
```

```
mkdir -p ~/.hammer/cli.modules.d
```

```
cat > ~/.hammer/cli.modules.d/foreman.yml << EOF
```

```
:foreman:
```

```
  # Credentials. You'll be asked for the interactively if you leave them  
blank here
```

```
  :username: 'admin'
```

```
  :password: 'changeme'
```

```
  # Check API documentation cache status on each request
```

```
  :refresh_cache: false
```

```
  # API request timeout in seconds, set -1 for infinity
```

```
  :request_timeout: 120
```

```
EOF
```

```
hammer ping; hammer shell
```



Monitor >

Content >

Containers >

Hosts >

Configure >

Infrastructure >

Administer >

Content

Red Hat Subscriptions

Red Hat Repositories

Products

GPG Keys

Sync Plans

Sync Status

Lifecycle

Lifecycle Environments

Content Views

Activation Keys

Content Types

Deb Packages

Docker Tags

Errata

Files

OSTree Branches

Packages

Lifecycle Environment Paths

[+ Create Environment Path](#)

Library	Content Views 2	Products 3	Yum Repositories 3	OSTree Repositories 0	Docker Repositories 1	Packages 10565	Errata 0	Puppet Modules 0
---------	--------------------	---------------	-----------------------	--------------------------	--------------------------	-------------------	-------------	---------------------

[+ Add New Environment](#)

	jakarta-qa	jakarta-prod
Content Views	0	0
Content Hosts	0	0

[+ Add New Environment](#)

	main-dev	main-prod
Content Views	1	1
Content Hosts	1	0

Grundeinstellungen

- Wenn gewünscht: Sprache umstellen (Admin User/My Account/Sprache)
- Grundeinstellungen (Administer Settings)

```
hammer user update --login admin --locale en  
hammer settings set --name login_text --value "Tübix Workshop"
```

Organization und Location

- Erstelle eine Organization und eine Location und setze diese als Standard
- Bonus: Verschachtelte Location

```
hammer organization create --name tuebix
hammer location create --name tue
hammer location list
hammer location create --name sand -parent-id 5
hammer user update --login admin --default-organization tuebix \
  --default-location sand
```

Sync Plan/Product/Repository

- Erstelle ein Product mit synchronisiertem Repository

```
hammer sync-plan create --organization tuebix --name daily \  
  --interval daily --enabled true --sync-date 2018-06-06  
hammer product create --name product-centos --organization tuebix \  
  --sync-plan daily  
hammer repository create --product product-centos --name repo-centos7 \  
  --organization tuebix --content-type yum \  
  --url http://ftp.fau.de/centos/7/os/x86\_64/  
hammer repository synchronize --name repo-centos7 --product product-centos \  
  --organization tuebix
```

- Einschub Red Hat Subscription (access.redhat.com)

Custom Repository

- Erstelle ein Custom Product und Repository für tuebix
- Erstelle ein Custom RPM/DEB/... Paket
 - z.B. mittels <https://github.com/jordansissel/fpm>
- Lade das Paket hoch

```
git clone https://github.com/tuebix/tuebixorg.git
cd tuebixorg
bundle install && bundle exec jekyll build && cd _site
gem install fpm || docker build -t fpm . || ...
fpm -s dir -t rpm -n tuebixorg -v 0.1 --prefix /var/www/html/ \
  --url http://tuebix.org .
hammer repository upload-content --organization tuebix --name repo-tuebix \
  --product product-tuebix --path ./tuebixorg-0.1-1.x86_64.rpm
```

Lifecycle Environment (Path)

- Erstelle zwei Lifecycle Environments in einem Pfad

```
hammer lifecycle-environment create --name main-dev --prior Library \  
--organization tuebix
```

```
hammer lifecycle-environment create --name main-prod --prior main-dev \  
--organization tuebix
```


(Composite) Content View

- Erstelle eine Content View, welches das centos7 Repository enthält
- Erstelle eine Composite Content View, welche die aktuellste Content View beinhaltet
- Publiziere die Content View
- Publiziere die Composite Content View
- Promote die Composite Content View in das erste Lifecycle Environment

Activation Key

- Erstelle einen Activation Key und verknüpfe es mit dem ersten Lifecycle Environment, der Composite Content View und einer centos Subscription

Domain

- Erstelle die Domain tuebix.org
 - Füge alle Locations und
 - alle Organizations hinzu

Hostgroup

- Erstelle eine Hostgroup
- Ordne der Hostgroup folgendes zu
 - Lifecycle Environment und CV
 - Domain
 - min. eine Location
 - Operating System und Partition Table
 - Organization
 - Activation Key

Installation Media

- Erstelle ein Installation Media und weise es einem Operating System hinzu

Host

- Erstelle einen Host mit der eben erstellten Hostgroup
- Füge ein Netzwerk-Interface hinzu
 - min. MAC-Adresse
- Was ändert die Hostgroup?
- Betrachte die Template Preview
 - Provisioning Template (Kickstart)
 - PXELinux (pxelinux.cfg)

Host Collections

- Erstelle eine Host Collection
- Füge beide Hosts hinzu
- Beide?
- Ja.
 - Any Location/Any Organization
- Welche «Actions» sind möglich?

(Cloud) Provider

- Administer/About/Available Providers
- `tfm-rubygem-foreman_{docker,azure,...}`, `foreman-ec2` Pakete vorhanden
- Integration (z.B. bei VMware) mit Konsole, Powerswitch und Provisioning
- Docker Setup
 - https://www.theforeman.org/plugins/foreman_docker/3.x/index.html

Patching/cvmanager

- Publish CV → Update CCV → Publish CCV → Promote CCV ist umständlich
- Daher: <https://github.com/RedHatSatellite/katello-cvmanager>

Config - cvmanager - Commands

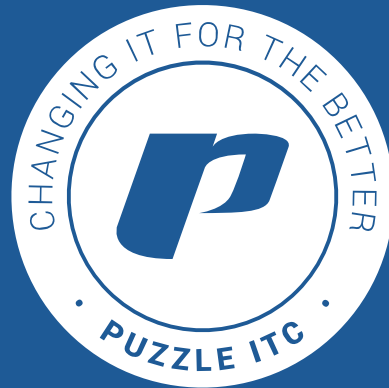
```
---
:settings:
  :user: admin
  :pass: changeme
  :uri: https://localhost
  :timeout: 300
  :org: 4
  :lifecycle: 4
  :keep: 5
:ccv:
  ccv-base:
    cv-os: latest
:publish:
  - cv-os
:promote:
  - ccv-base

sudo yum install rubygem-apipie-bindings

./cvmanager --no-verify-ssl --wait publish
./cvmanager --no-verify-ssl --wait update
./cvmanager --no-verify-ssl --wait promote \
  --to-lifecycle-environment 5
```

Folien, Updates und Informationen
<https://www.lukex.de/workshop/lcm/>

Merci!



 @LuBeKa