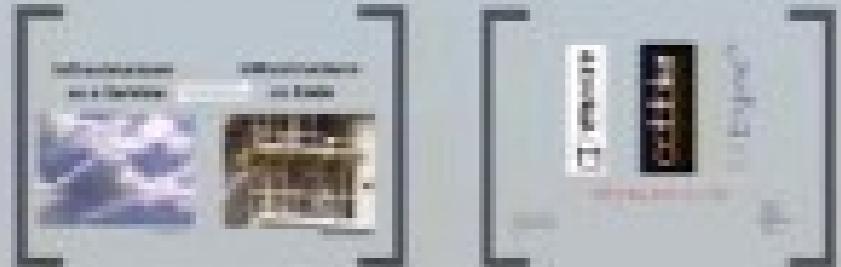


Infrastructure as Code

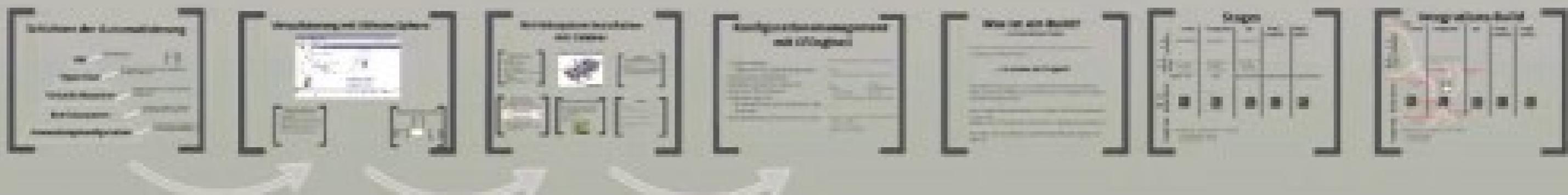


**Virtualisierung und
Automatisierung - wie man sich
ein Rechenzentrum programmiert**

Fazit



Werkzeuge



Ressourcen Planung,
Virtualisierung

PXE Boot, DHCP
Netzwerkinstallation

Service + Stage Datei
Konfiguration

Continuous Delivery

Über uns



Mathias Münch



Jörg Meltzer

Infrastructure as Code

Infrastructure
as a Service



Infrastructure
as Code



CFEngine®

Infrastructure as Code

Infrastructure as a Service

Infrastructure as Code



am 20.07.2018 in Berlin



© 2018, Construction.com Institute



<http://nikhewitt.blogspot.de/>



VMware vSphere 5.5 Update 1
VMware vSphere 5.5 Update 1



CloudBees®
Infrastructure as Code

Infrastructure as Code

Open-Source Code

- Ansible
- Puppet
- Chef
- SaltStack
- Fabric

Closes matches Infrastructure

- AWS CloudFormation
- Terraform
- OpenShift
- Docker Compose

"Infrastructure as code" is how the phrase
differs from other ways to specify systems or to
manage system evolution. "Infrastructure as
code" is pretty clear. Write your infrastructure as
code. Programmable. Readable. Deployable.
John C. Stevens



vmware®

- VMware vSphere 4.1
- VMware vSphere SDK for Perl

cooper

cooper

CFEngine®

```
vmcreate.pl --vmname "testVM" --mac0  
00:50:56:7e:45:1b --tpl L
```

```
cobbler system add  
--name=<x10110.domain.de>  
--profile=<RHEL6>  
--mac=<00:50:56:7e:45:1b>  
--ip=<172.22.113.50>  
--hostname=<x10110.domain.de>
```

```
bundle agent resolver
{
  files:
    "/etc/resolv.conf"
      copy_from => remote(
        "${global.cfmaster}", "${global.filestore}/resolver/resolv.co
nf", "false"),
      classes => if_ok("resolv_conf_ok"),
      perms => mog("644", "root", "root");

  reports:
    resolv_conf_ok::
      "Name resolution configured correctly.";
}
```



• VMware Infrastructure 4.1
• VMware Infrastructure 5.0 (beta)



Oncore Infrastructure

- Oncore Core
- Oncore Automation
- Oncore Metrics
- Oncore System

Devops machine Infrastructure

- Devops Infra
- Automation
- Configuration Management
- Monitoring

Infrastructure as Code

Infrastructure as Code / Infrastructure as Code
Infrastructure as Code is a model that enables organizations to manage infrastructure as code. This means that infrastructure is defined by code, making it more reliable, repeatable, and maintainable. (Source: Wikipedia)

"Infrastructure as code". I love the phrase.
Where devops is a word that is sadly open to so
much (mis)interpretation, "Infrastructure as
code" is pretty clear. Treat your infrastructure as
code. Programmable. Testable. Deployable.
(John E. Vincent)

Ops machen Code

- von Devs lernen
 - Agile Methoden
 - Continuous Delivery
 - Automatisierte Tests

Devs machen Infrastruktur

- von Ops lernen
 - Packaging
 - Konfigurationsmanagement
 - Monitoring

Infrastructure as Code



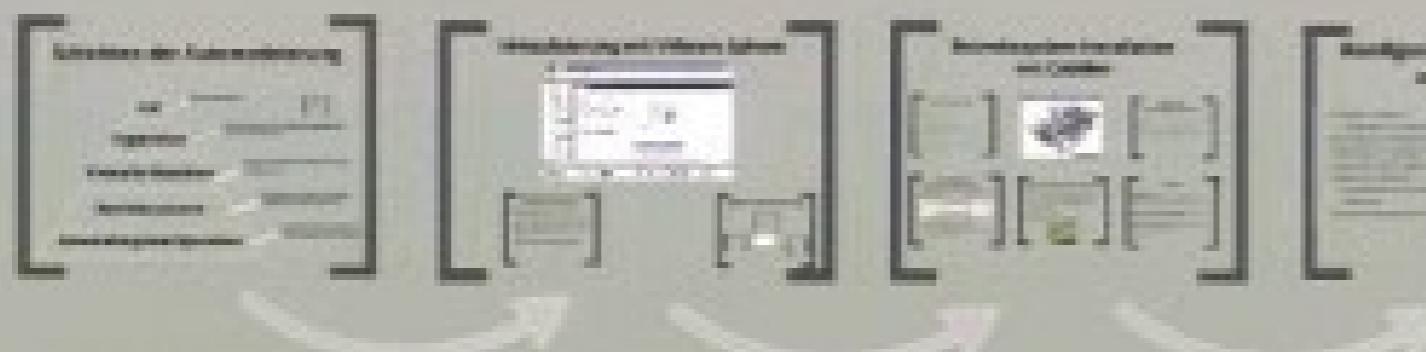
[+]

**Virtualisierung und
Automatisierung - wie man sich
ein Rechenzentrum programmiert**

Fazit



Werkzeuge

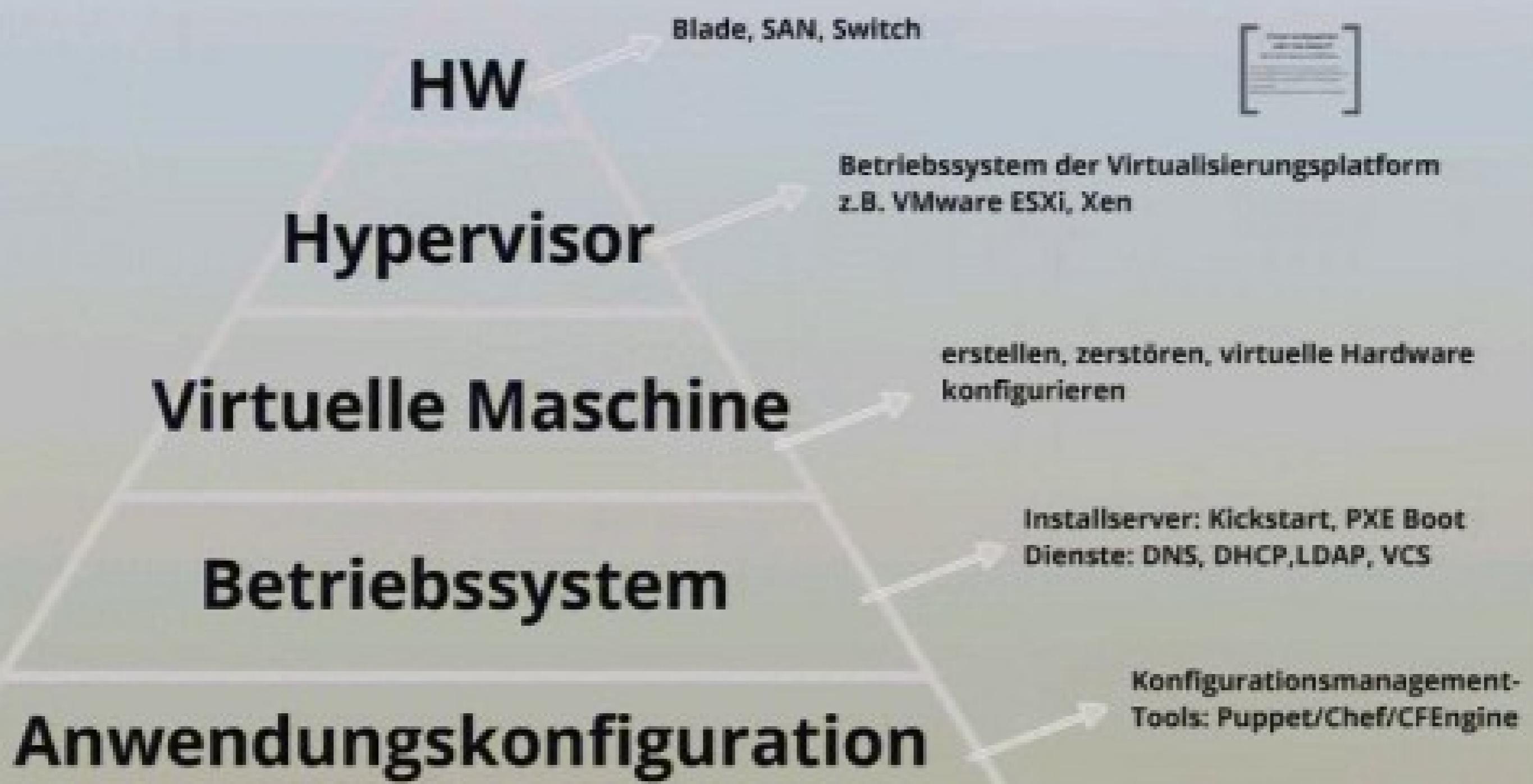


Ressourcen Planung, PXE Boot, DHCP, Service + Stage Datei
Virtualisierung, Netzwerkinstallation, Konfiguration

Continuous Delivery



Schichten der Automatisierung



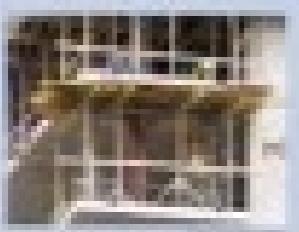
Clonen und patchen oder neu bauen?

Patchen schon damals eine Bad Practice

"'Real Programmers' are reluctant to actually edit a program that is close to working. They find it much easier to just patch the binary object code directly"

(Ed Post, Real Programmers Don't Use PASCAL)

ein Rechenzentrum



CF Eng

cf

automatisierung

zur

Universität Regensburg

und

Wirtschaftswissenschaften

und

Technik

und

Medien

und

Architektur

und

Design

und

Sozialwissenschaften

und

Geographie

und

Geisteswissenschaften

und

Humanmedizin

und

Naturwissenschaften

und

Mathematik

und

Physik

und

Chemie

und

Biochemie

und

Zoologie

und

Botanik

und

Geodäsie

und

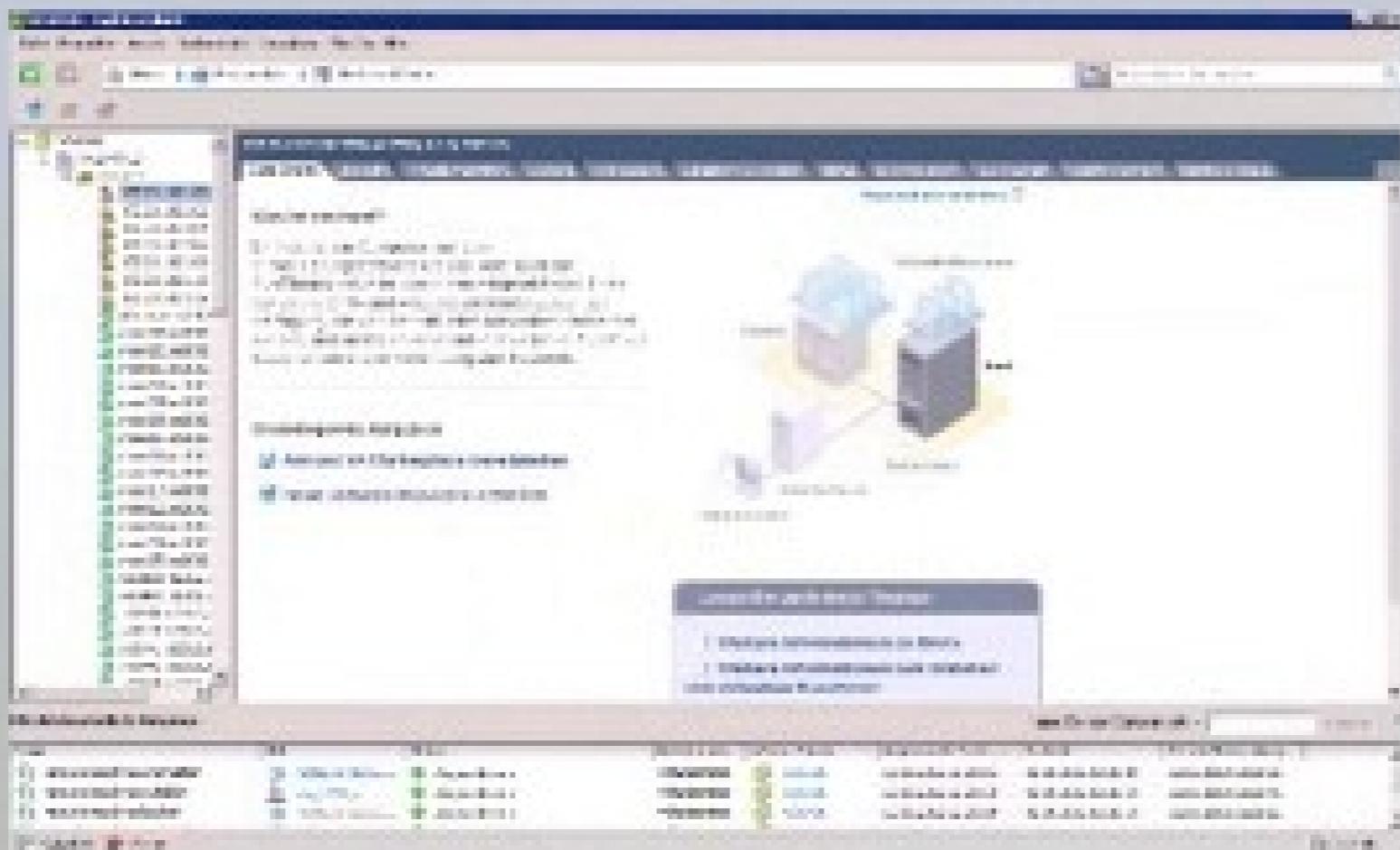
Geoinformatik

und

Geographie

und

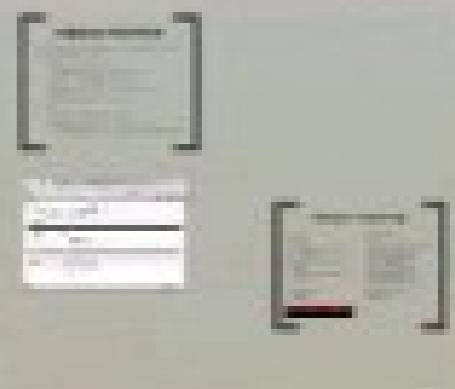
Virtualisierung mit VMware Sphere



Bekannte Features

- Manages ESX/ESXi hosts and virtual machines
- High Availability
- Virtual networking connects the management and virtual hosts via IP, storage, and resource providers
- Virtual machines and hosts exchange data via shared networks
- Resource management organizes the allocation and utilization of resources
- Site Recovery Manager backs up and restores data centers
- Configuration editor to build and edit virtual machines

vSphere Automatisierung



Bekannte Features

- Mehrere OS Installationen passen auf einen Rechner.
- Hardware wird teilbar:
 - Virtualisierung ermöglicht die Betrachtung der Hardware als Pool aus CPU-, Speicher-, Storage- und Netzwerkressourcen.
 - Virtuelle Maschinen sind baubar, solange die Pools reichen.
- Robustheit:
 - Virtuelle Maschinen migrieren bei Überlast auf andere Rechner.
 - Bei Ausfällen starten VMs auf anderen Rechnern.
 - Konfiguration dafür ist leicht einstellbar und änderbar.

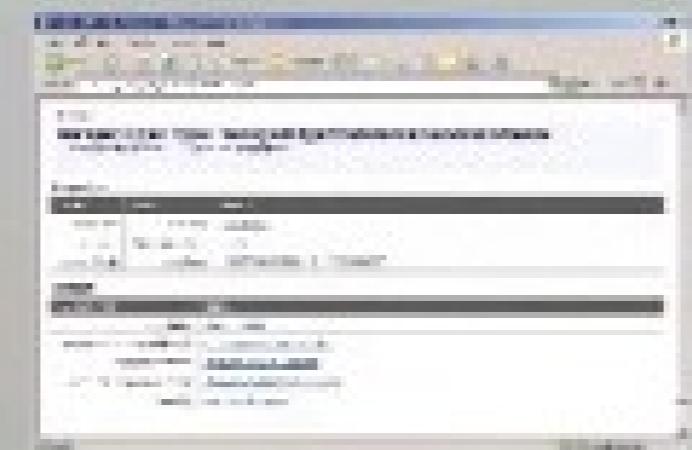
vSphere Automatisierung

vSphere Perl-SDK

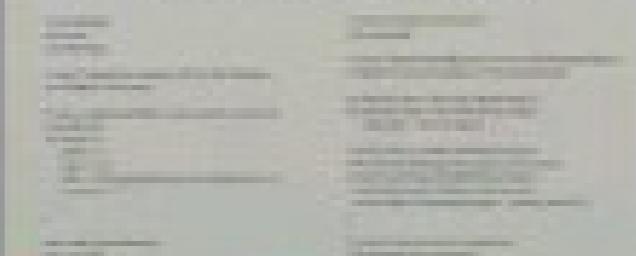
- wichtigste API für vSphere um mit Clienten SSO zu kooperieren
- Schnelle Entwicklung
- Schichtenweise modular
- interne Abstimmung und Dokumentation ist gut
- gute Online API-Dokumentation
- API bietet auch synchrone Operationen
- VMware Community-Support
- schwach
- Codequalität Benutzbarkeit
- POD Documentation in Inklusionstechnik
- keine VMware Shell-Linie zur PowerShell (D)

Virtualisierung erleichtert Automatisierung

- Einzelne Maschinen lassen sich ohne Hardwareaufwand erweitern
- Das VDI Deployment wird programmiert
- Die Virtualisierung erlaubt es automatisch mit Spezialtools VMs weiter zu verteilen ("Clouded" Hosts)



Beispiel - Reporting



Virtualisierung erleichtert Automatisierung

- Zusätzliche Maschinen lassen sich ohne Hardwarekauf erstellen.
- Das VM-Deployment wird programmiert.
- Die Virtualisierungsplatform abstrahiert von Spezialhardware, VMs sehen nur einfache "Standard" Hardware.

vSphere Perl-SDK

- leichtgewichtiger Wrapper um vCenter Soap Api
- Schnelle Entwicklung
 - Beispieldokumentation nutzen
vm{create,destroy,info,control}.pl
 - gute Online API Dokumentation
 - API bietet auch synchrone Operationen
 - VMware Community Skripte
- schlecht
 - Codequalität Beispieldokumentation
 - POD Dokumentation in Modulen fehlt
 - keine VMware Shell analog zur PowerShell CLI

Managed Object Browser - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Stop Home Search Favorites Print Copy Paste Find Help

Address https://<my_esx_server>/vmware.com/mob Go Links

Home

Managed Object Type: ManagedObjectReference:ServiceInstance
Managed Object ID: ServiceInstance

Properties

NAME	TYPE	VALUE
capability	Capability	capability
content	ServiceContent	content
serverClock	dateTime	"2007-11-26T21:49:47.291812Z"

Methods

RETURN TYPE	NAME
dateTime	CurrentTime
HostVmMotionCompatibility[]	QueryVmMotionCompatibility
ServiceContent	RetrieveServiceContent
ProductComponentInfo[]	RetrieveProductComponents
Event[]	ValidateMigration

Done Local intranet

Managed Object Browser - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Stop Refresh Search Favorites Home Stop Refresh Favorites

Address https://<my_esx_server>/vmware.com/mob Go Links

Home

Managed Object Type: ManagedObjectReference:ServiceInstance

Managed Object ID: ServiceInstance

Properties

NAME	TYPE	VALUE
capability	Capability	capability
content	ServiceContent	content
serverClock	dateTime	"2007-11-26T21:49:47.291812Z"

Methods

RETURN TYPE	NAME
dateTime	CurrentTime
HostVmMotionCompatibility[]	QueryVmMotionCompatibility
ServiceContent	RetrieveServiceContent
ProductComponentInfo[]	RetrieveProductComponents
Event[]	ValidateMigration

Done Local intranet

Beispiel - Reporting

```
#!/usr/bin/perl
use strict;
use warnings;

# Step 1: Import the vSphere SDK for Perl Modules.
use VMware::VIRuntime;

# Step 2: (Optional) Define Script-Specific Command-Line Options.
my %opts = (
    entity => {
        type => "-s",
        help => "ManagedObject type: VirtualMachine, etc",
        required => 1,
    },
);

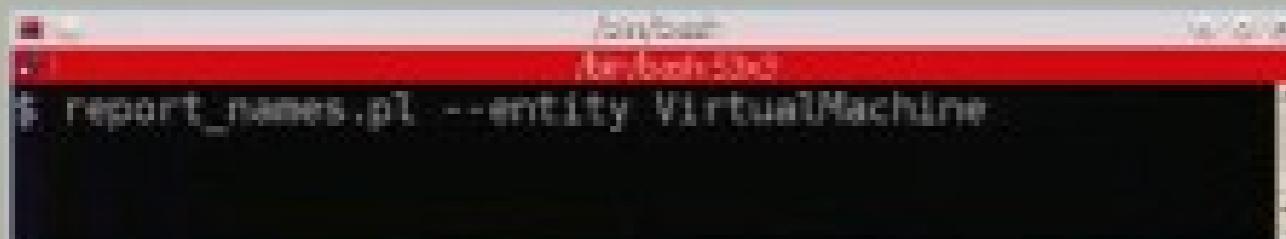
Opts::add_options(%opts);
Opts::parse();
Opts::validate();

# Step 3: Connect to the Server.
Util::connect();

# Step 4: Obtain View Objects of Server-Side Managed Objects.
# Obtain all inventory objects of the specified type
my $entity_type = Opts::get_option('entity');
my $entity_views = Vim::find_entity_views(
    view_type => $entity_type);

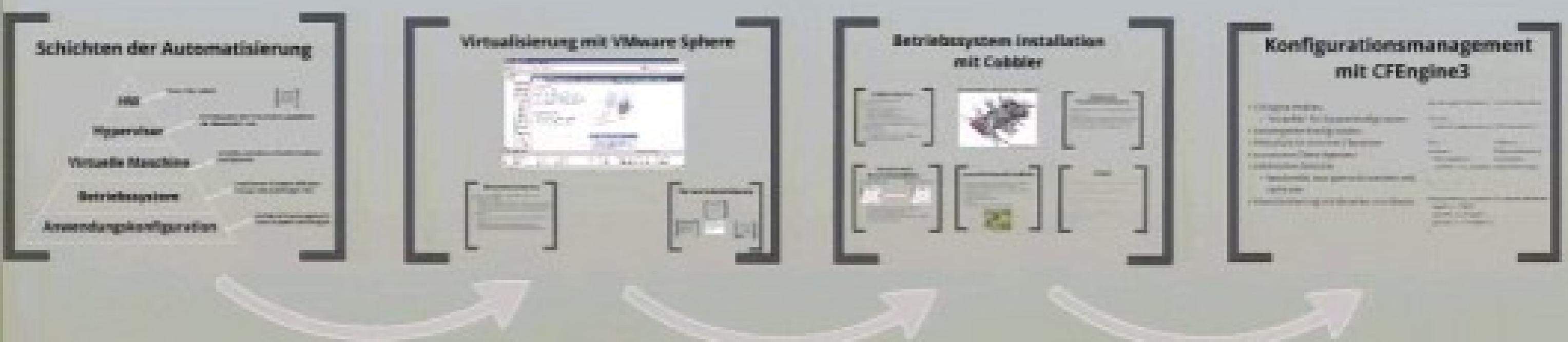
# Step 5: Process Views and Report Results.
# Process the findings and output to the console
foreach my $entity_view (@$entity_views) {
    my $entity_name = $entity_view->name;
    Util::trace(0, "Found $entity_type: $entity_name\n");
}

# Step 6: Close the Server Connection.
# Disconnect from the server
Util::disconnect();
```





Werkzeuge



Ressourcen Planung,
Virtualisierung

PXE Boot, DHCP
Netzwerkinstallation

Service + Stage Datei
Konfiguration

Betriebssystem Installation mit Cobbler

Cobbler Features

- zentraler Installationsserver
- Open Source
- automatisiert OS Installation
 - RHEL, SUSE, Debian
- vereinfacht Handhabung von
 - Prof. Basis, TFTP Server, DHCP, DNS ...
- flexible Konfiguration mit Snippets und Templating
- Web Interface
- Distributionierung
 - Distribution, Profil, System, YUM Groups
 - Python, XML-RPC Schnittstelle

Netzwerkinstallation mit Cobbler

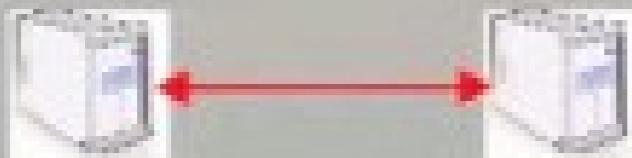


Übergang zum Konfigurationsmanagement

- Versucht Produktivität, nur das Notwendige zu installieren
- Systeme mit Service- und Sludge Informationen auslegen
- Letzter Task der OS Installation ist der Ansatz (auf dem Konfigurationsmanagement Tool)

Herkömmliche Netzwerkinstallation

Install Server neuer Server



- Netzwerkinterface via DHCP belegen
- TFTP Boot Datei auf TFTP Server anlegen
- Copy & Paste Rechner Datei von einem Template
- DHCP POI konfigurieren
- neue Server neu booten
- installieren
- über Reiter in älterer Prozedur abrufen

manuelle Netzwerkinstallation

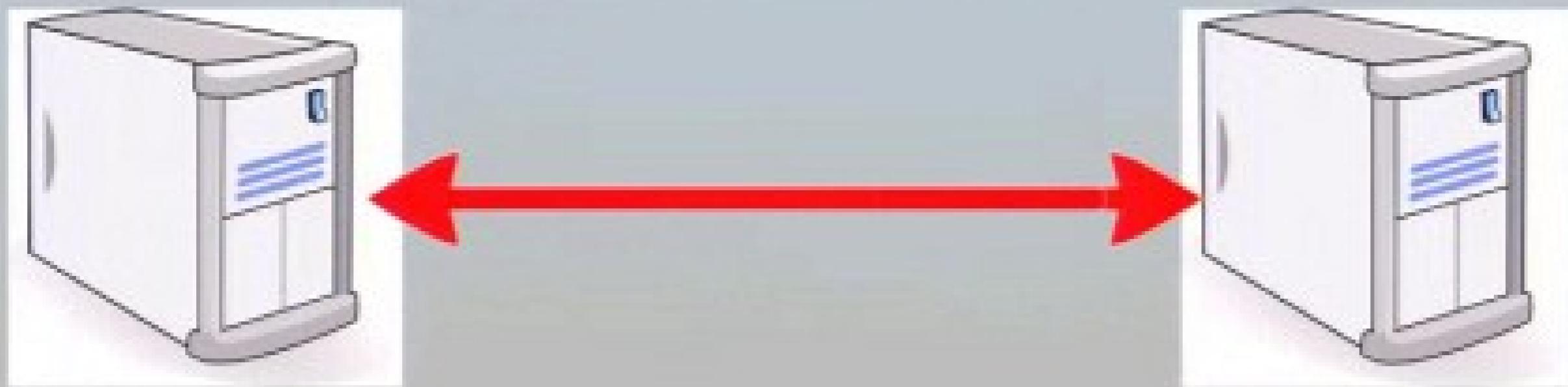
- neue Mac Adresse in Netzwerk erstellen
- IPV4 Boot Datei auf TFTP Server anlegen
- Copy & Paste Rechner Datei von einem Template
- DHCP POI konfigurieren
- neue Server neu booten
- installieren
- über Reiter in älterer Prozedur abrufen



Beispiel

A screenshot of a terminal window displaying configuration management code, likely Puppet or Ansible, showing how to manage network interfaces and services on a system. The code includes commands like 'file', 'service', and 'network' resources with specific parameters and conditions.

Herkömmliche Netzwerkinstallation



- Netzwerkadresse via DHCP beziehen
 - TFTP download Kernel + Initrd
 - Download automatisches Kickstart Bootskript
 - Installation durchführen

manuelle Netzwerkinstallation

- neue Mac Adresse in /etc/dhcpd.conf eintragen
- PXE Boot Datei auf TFTP Server anlegen
- Copy & Paste Kickstart Datei von einem Template
- DHCPD neustarten
- neuen Server einschalten
- warten
- über Fehler in obiger Prozedur ärgern



Netzwerkinstallation mit Cobbler



Cobbler Features

- zentraler Installationsserver
- Open Source
- automatisiert OS Installation
 - RHEL, SuSE, Debian
- vereinfacht Handhabung von
 - PXE Boot, TFTP Server, DHCP, DNS, ...
- Flexible Kickstarts mit Snippets und Templating
- Web Interface
- Orchestrierung
 - Distribution, Profil, System, YUM Repos
- Python, XML-RPC Schnittstelle

Beispiel

```
# import mounted dvd
cobbler import --path=/mnt --name=CentOS-6 --kickstart=/var/lib/cobbler/kickstarts/default.ks
cobbler distro report --name CentOS-6

cobbler profile add --name latest --comment "profile for development hosts"
cobbler repo add --name latest --mirror /var/www/yum/latest --comment "yum repo for development"
...
cobbler profile add --name int --comment "profile for integration hosts"
cobbler repo add --name int --mirror /var/www/yum/int --comment "yum repo for integration"
cobbler system add --name int1 --profile int --interface eth0 --mac ... --ip ...

cobbler profile add --name prod --comment "profile for prod1 hosts"
cobbler repo add --name prod --mirror /var/www/yum/prod --comment "yum repo for prod"
cobbler system add --name prod1 --profile prod

# dhcp, tftp... synchronisieren
cobbler sync
```

Übergang zum Konfigurationsmanagement

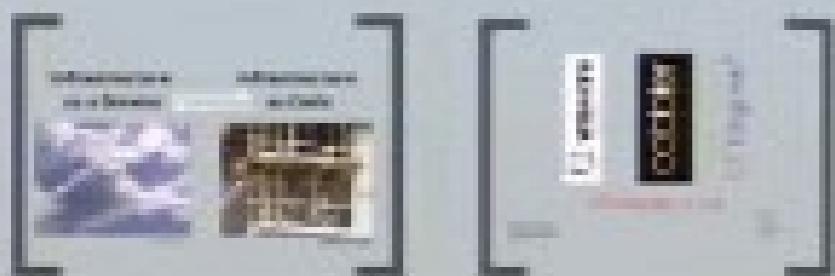
- Vorsicht Featuritis, nur das Nötigste in Kickstart installieren
- System mit Service und Stage Information impfen
- Letzter Task der OS Installation ist der initiale Lauf des Konfigurationsmanagement Tools

Konfigurationsmanagement mit CFEngine3

- CFEngine Policies
 - "Makefile" für Systemkonfiguration
- konvergente Konfiguration
- Policyhub ist dummer Fileserver
- autonome Client Agenten
- deklarative Sprache
 - beschreibt, was gemacht werden soll,
nicht wie
- Modularisierung mit Bundles und Bodys

```
bunde agent shadow { # wie Subroutine
  classes:
    "sollinux" expression => "linux|solaris";
  files:                      # Phase
  sollinux::                  # Einschränkung
    "/etc/shadow"             # promise
    perms => p_shadow;        # promise body
}
body perms p_shadow { # snippet, template
  mode => "640";
  owners => {"root"};
  groups => {"shadow"};
}
```

Infrastructure as Code



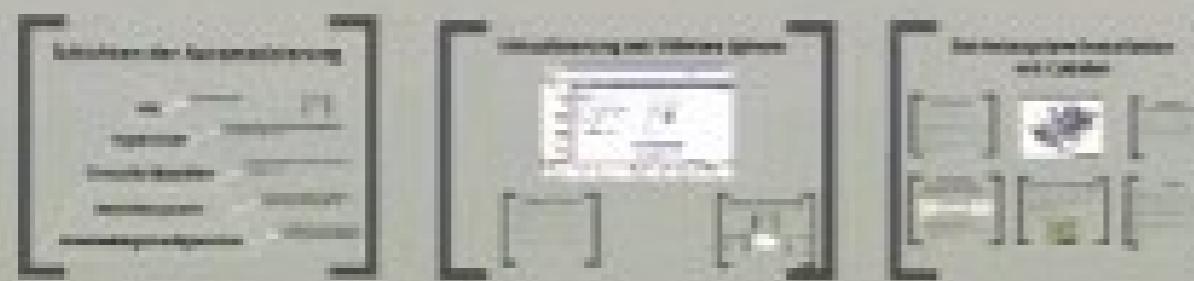
[+]

**Virtualisierung und
Automatisierung - wie man sich
ein Rechenzentrum programmiert**

Fazit



Werkzeuge



Ressourcen Planung,
Virtualisierung

PXE Boot, DHCP
Netzwerkinstallation

Service + Stage Datei
Konfiguration

Continuous Delivery



Was ist ein Build?

... in einem Software Projekt

- Ein Build ist Sequenz von Schritten, die aus Source Code ein deploybares Artefakt erstellt.

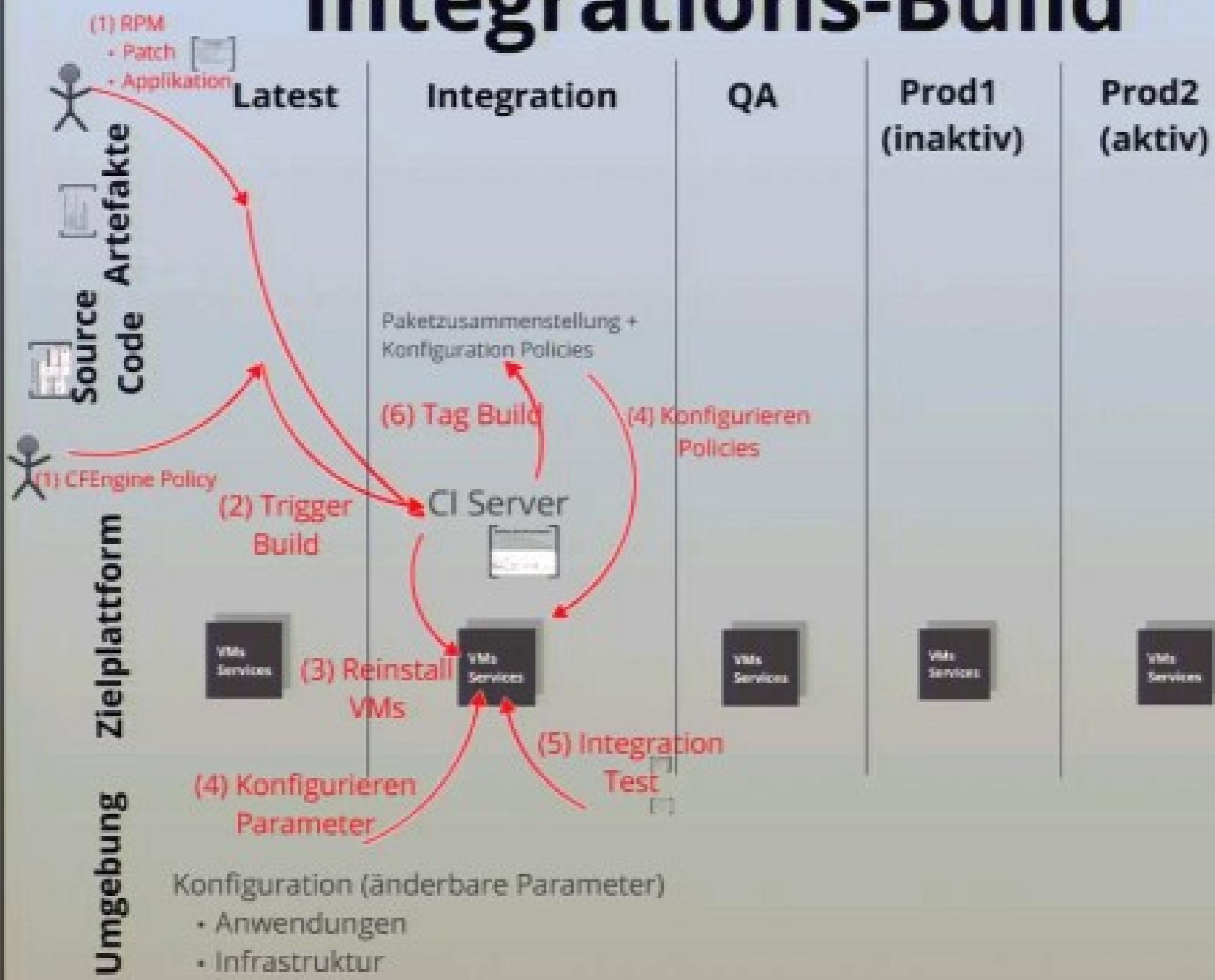
... in einem IaC Projekt?

- Ein Build ist eine Sequenz von Schritten, die für eine bestimmte Paketzusammenstellung und Sammlung von Konfigurationspolicies virtuelle Maschinen baut.
- Der CFEngine Masterserver checkt CFEngine Policies in bestimmter Version aus.
- Cobbler baut YUM Repos mit bestimmter Paketzusammenstellung.
- Ein Skript löst die komplette Neuinstallation für eine Gruppe von VMs aus.

Stages

	Latest	Integration	QA	Prod1 (inaktiv)	Prod2 (aktiv)
Source Artefakte	yum/latest 	yum/int	yum/qa
Code	packages.csv CFEngine masterfiles	packages.csv CFEngine masterfiles	packages.csv CFEngine masterfiles	Integration-Build-42	Integration-Build-42
Version	trunk	r4711	Integration-Build-42	Integration-Build-42	Integration-Build-41
Zielplattform	 VMs Services  VMs Services	App-VMs  VMs Services Infrastruktur-VMs	 VMs Services	 VMs Services	 VMs Services
Umgebung	Konfiguration (änderbare Parameter)				
	<ul style="list-style-type: none"> Anwendungen  Infrastruktur  				

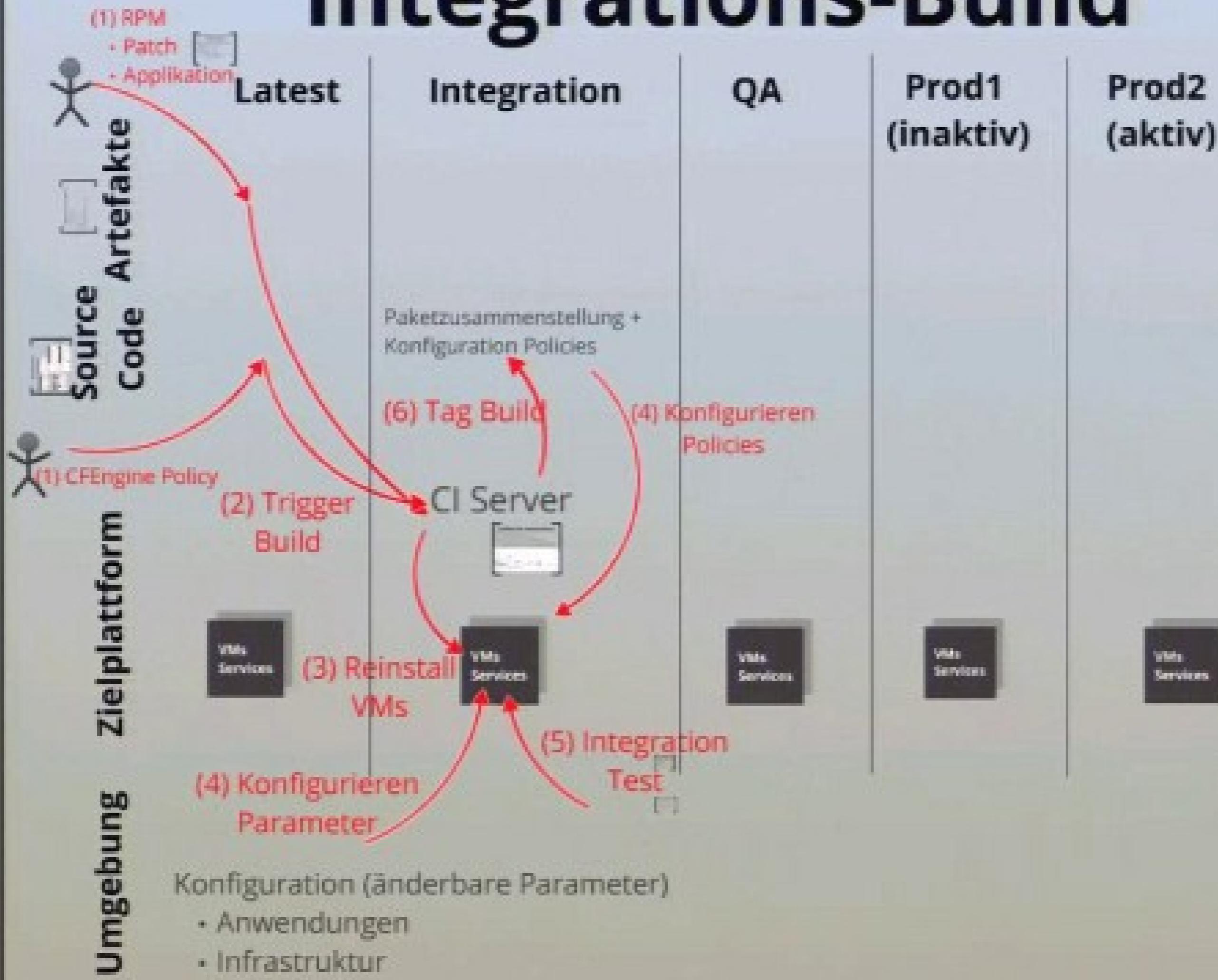
Integrations-Build



Artefakt Versionierung und Paket Auswahl

- Typischer Artefaktname <Name>-<Major>.<Minor>.<Release>
- Strategie bei der Artefaktauswahl für Entwicklungsumgebung
 - immer die höchste Version installieren
 - aktuell bleiben
 - Konfigurationsaufwand minimieren
- Problem bei Bugfixes
 - Lösungsidee Feature Toggling
 - Bugfixes und Features werden gleichzeitig ausgeliefert.
 - Neue Features sind jedoch über die Umgebungskonfiguration initial deaktiviert.

Integrations-Build



RZ Integration und Projekt Integration

- Software Projekte benötigen eigenen Integrationsbuild
- Projekt Integration
 - baut Applikations RPM
 - testet neue Features
- RZ Integration
 - testet die in Produktion aktivierten Features

Jenkins für Infrastruktur

- CI Server bauen Softwareprojekte
 - ... und jetzt auch virtuelle Maschinen
- Jenkins Jobs reinstallieren virtuelle Maschinen und
 - testen, ob die Infrastruktur Maschinen noch baubar sind
 - führen funktionale Tests für fachlichen Anwendungen durch

The screenshot shows the Jenkins web interface. On the left, there's a sidebar with links like 'Dashboard', 'Build History', 'Build Pipeline', 'Freestyle projects', 'Scripted pipelines', 'Job configurations', 'Deployed Builds', 'Recent Builds', and 'Build Processor Status'. A large circular icon with a hand icon is overlaid on the sidebar.

The main area displays a table of Jenkins jobs:

All	Name	Letzter Erfolg	Letzter Fehlerlog	Letzte Dauer	Aktivitäten
1	testcloud	14 Monate (0)	10 Monate (0)	1 Minute 17 Sekunden	
2	testcloud	1 Stunde 1 Minuten (0)	Unerkennbar	0.14 Sekunden	
3	infrastructureMaster	12 Minuten (0)	Unerkennbar	14 Sekunden	
4	infrastructureWorker	1 Stunde 2 Minuten (0)	Unerkennbar	0.14 Sekunden	
5	infrastructureBuilder	6 Tage 21 Stunden (0)	Unerkennbar	40 ms	
6	jenkins-infrastructure-Setup	6 Tage 21 Stunden (0)	Unerkennbar	0.79 Sekunden	
7	integration-Setup	6 Tage 21 Stunden (0)	6 Tage 21 Stunden (0)	0.3 Sekunden	
8	qaTestCloud	6 Tage 21 Stunden (0)	Unerkennbar	0.11 Sekunden	

Below the table, it says 'Sprint: 318 L'. At the bottom, there are buttons for 'Logout', 'Jenkins All Jobs', 'Jenkins New Job', and 'Jenkins Create new job'.

Jenkins für Infrastruktur

- CI Server bauen Softwareprojekte
 - ... und jetzt auch virtuelle Maschinen
- Jenkins Jobs reinstallieren virtuelle Maschinen und
 - testen, ob die Infrastruktur Maschinen noch baubar sind
 - führen funktionale Tests für fachlichen Anwendungen durch

The screenshot shows the Jenkins web interface. On the left, there's a sidebar with icons for Home, Build History, Runlist, Build Queue, Freestyle jobs, Pipeline jobs, and Jenkins itself. Below that is a section for 'Geplante Builds' with a link to 'Neue Builds geplant'. Under 'Build Processes Status', there's a table with two rows: '1 Bereit' and '2 Neue Infrastrukturen: Infrastruktur-Stage #2'. The main area displays a table of Jenkins jobs:

S	W	Name	Letzter Erfolg	Letzter Fehlerlog	Letzte Dauer	Aktivitäten
●	☁️	Ansible	14 Monate (0)	19 Minuten (0)	1 Minute 17 Sekunden	
●	☀️	Ansible2	11 Monate 7 Minuten (0)	Unerkennbar	0.14 Sekunden	
●	☀️	InfrastructureOTMaster	11 Monate 10 Minuten (0)	Unerkennbar	14 Sekunden	
●	☀️	InfrastructureUnit	11 Stunden 7 Minuten (0)	Unerkennbar	0.14 Sekunden	
●	☀️	InfrastructureUnitServer	10 Tage 21 Stunden (0)	Unerkennbar	40 ms	
●	☀️	InfrastructureUnitServer2	8 Tage 21 Stunden (0)	Unerkennbar	0.79 Sekunden	
●	☁️	Infrastructure-Stage	8 Tage 21 Stunden (0)	Unerkennbar	0.3 Sekunden	
●	☀️	QAProject	8 Tage 21 Stunden (0)	Unerkennbar	0.11 Sekunden	

At the bottom, there are links for 'Logout', 'Jenkins All Jobs', 'Jenkins New Failsafe', and 'Jenkins New Failed Jobs'.



Applikation2

Applikation bauen, Unitests, Artefakt in Yum Repository einspielen



InfrastructureCFMaster

CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen



InfrastructureJenkins

Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen



InfrastructureSubversion

SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen



Integration-Infrastructure-Stage

vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen



Integration-Stage

vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnessse), CFEngine Smoke Test



Applikation2

Applikation bauen, Unitests, Artefakt in Yum Repository einspielen



InfrastructureCFMaster

CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen



InfrastructureJenkins

Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen



InfrastructureSubversion

SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen



Integration-Infrastructure-Stage

vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen



Integration-Stage

vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnessse), CFEngine Smoke Test



Applikation2

Applikation bauen, Unitests, Artefakt in Yum Repository einspielen



InfrastructureCFMaster

CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen



InfrastructureJenkins

Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen



InfrastructureSubversion

SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen



Integration-Infrastructure-Stage

vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen



Integration-Stage

vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnessse), CFEngine Smoke Test



Applikation2

Applikation bauen, Unitests, Artefakt in Yum Repository einspielen



InfrastructureCFMaster

CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen



InfrastructureJenkins

Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen



InfrastructureSubversion

SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen



Integration-Infrastructure-Stage

vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen



Integration-Stage

vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnessse), CFEngine Smoke Test



Applikation2

Applikation bauen, Unitests, Artefakt in Yum Repository einspielen



InfrastructureCFMaster

CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen



InfrastructureJenkins

Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen



InfrastructureSubversion

SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen



Integration-Infrastructure-Stage

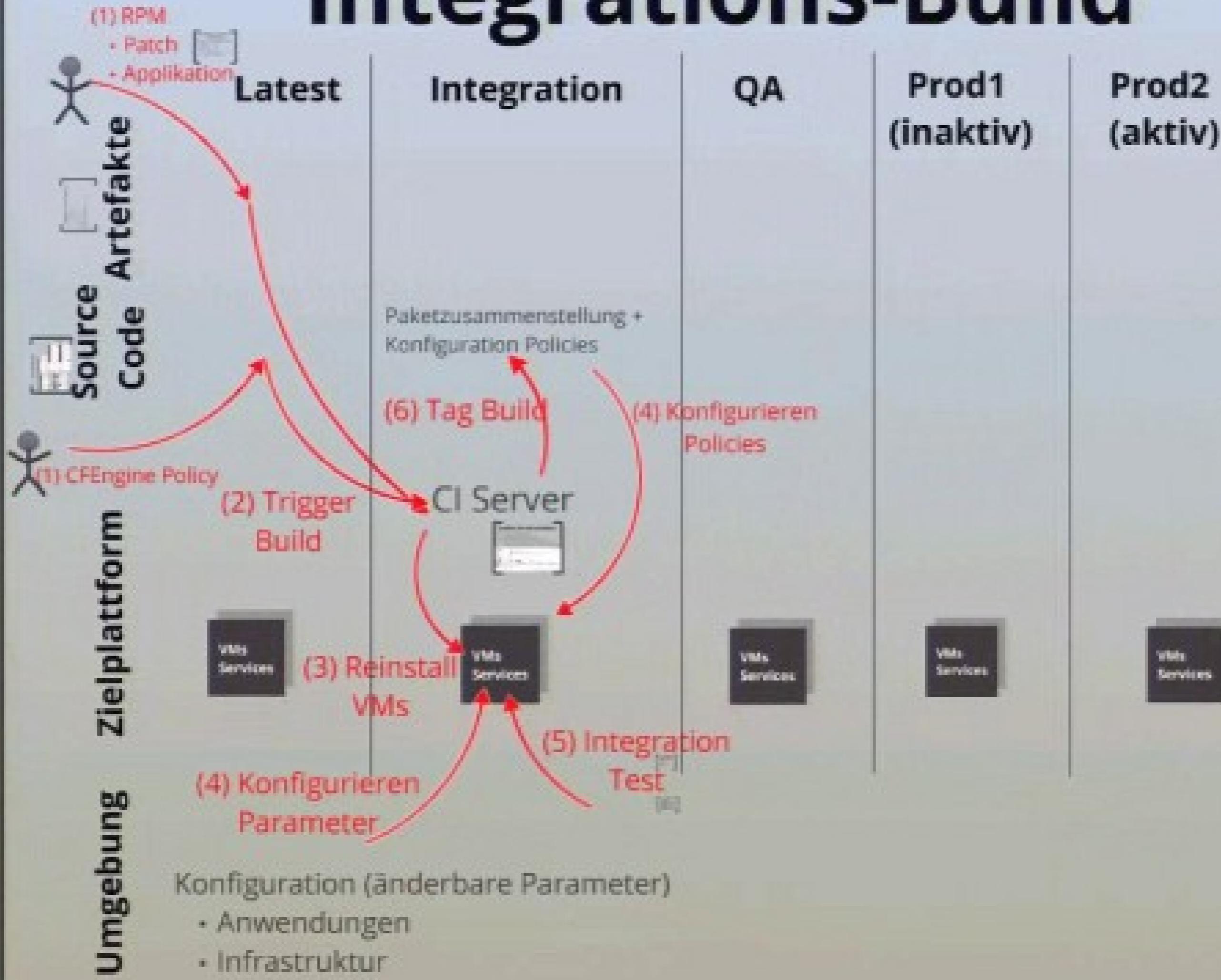
vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen



Integration-Stage

vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnessse), CFEngine Smoke Test

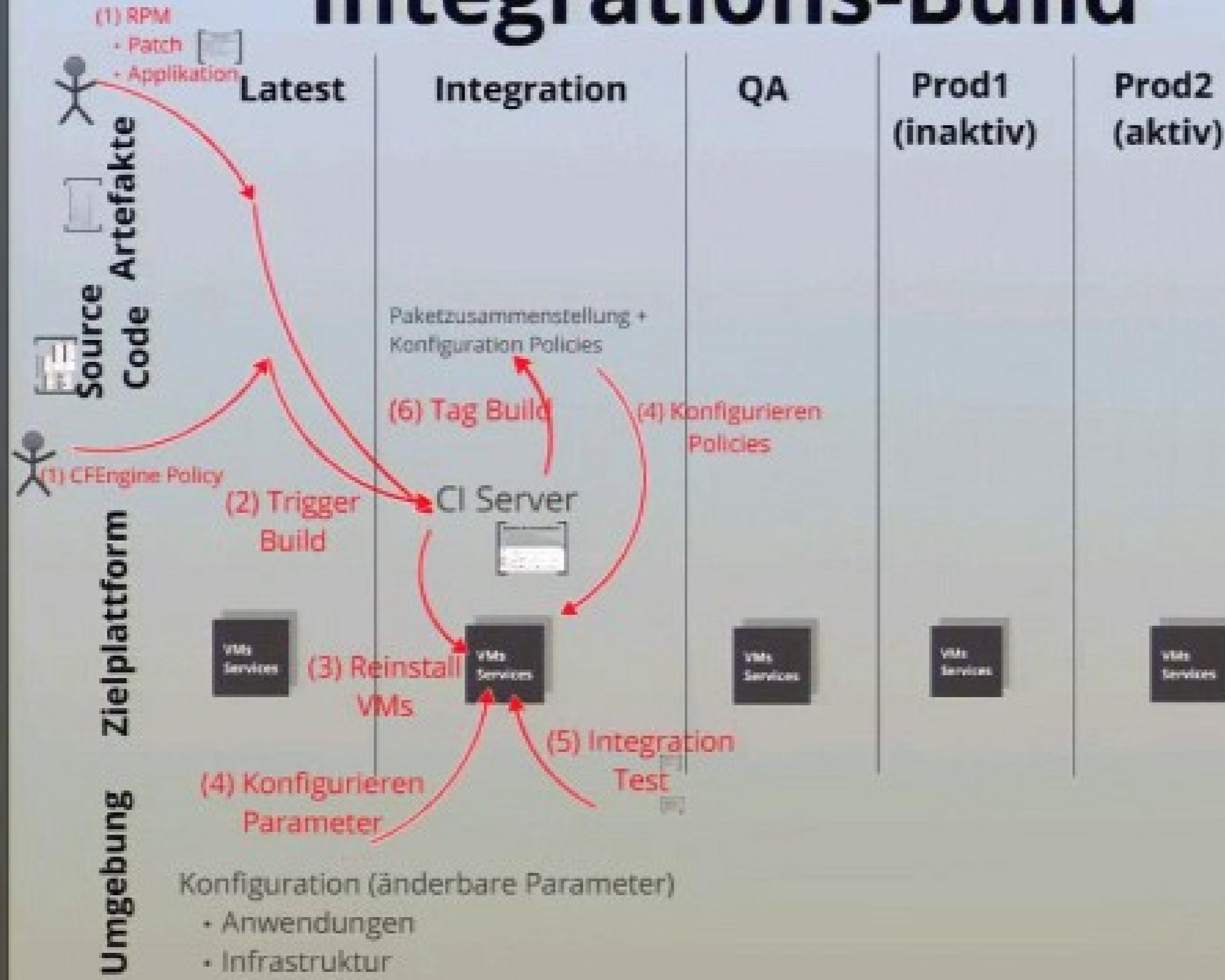
Integrations-Build



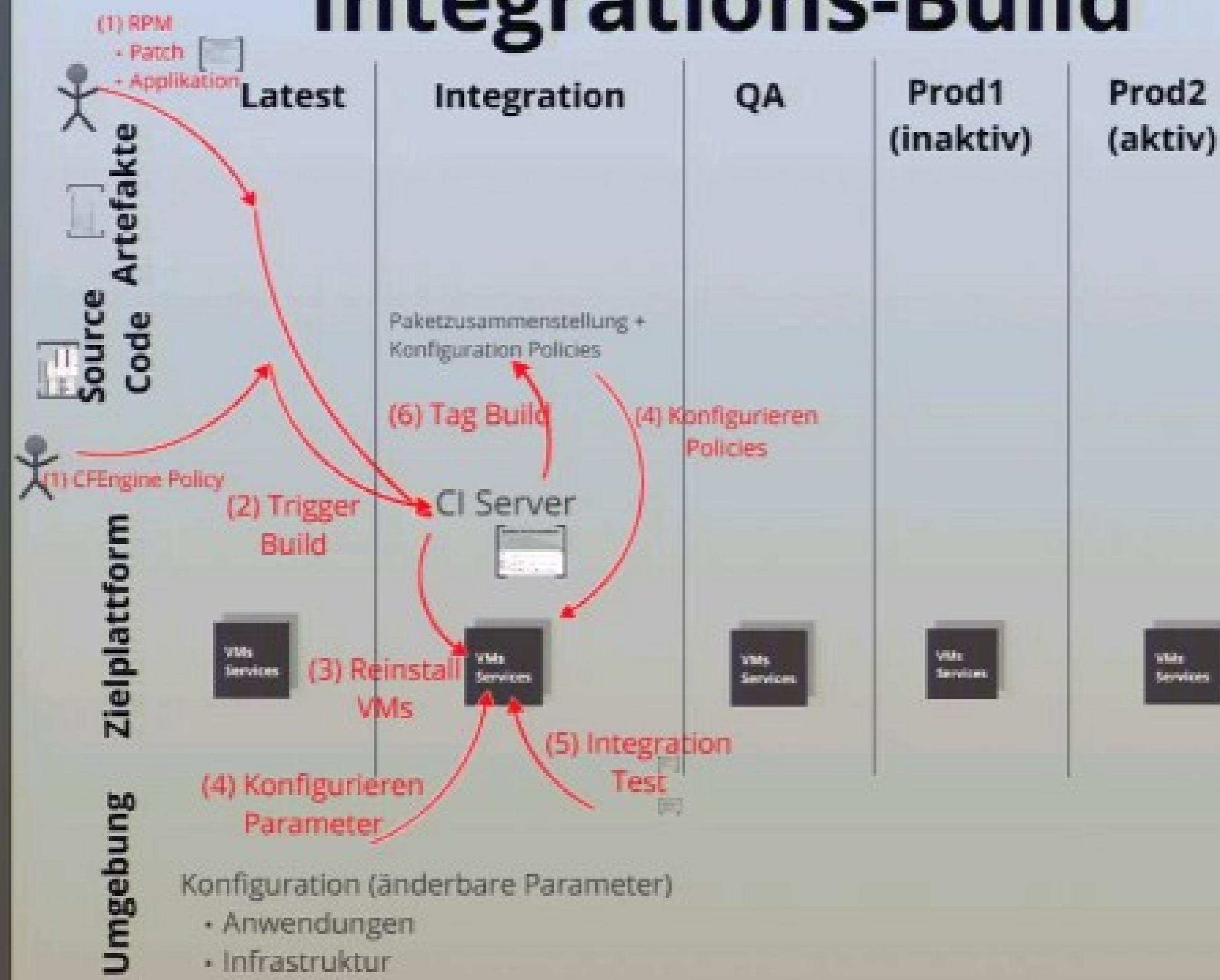
Hartes Staging

- Produktionsinstallation ohne Stress
- Standardisierte Umgebungen
 - in Entwicklungsumgebung Flexibilität auf Anforderung
 - CFEngine Daemon deaktivieren
 - ggf. zusätzliche YUM Repositories nutzen
- Anwendungen müssen Feature Toggling unterstützen
- Oberste Priorität, schnelle Pipeline, häufig stabile Integrationsumgebung
 - möglicher Konflikt
 - Sicherheitspatches täglich in Produktion bringen
 - Anwendung oder CFEngine Policy debuggen

Integrations-Build



Integrations-Build



Das Ganze gibt es wirklich



Infrastructure as Code Projekt

- Kunde: Internetdienstleister, IT Abteilung – 100 MA
- vorher
 - Umgebungen unterschiedlich
 - manuell installiert
 - Hardware schlecht ausgelastet
 - lange Deployment Zyklen
- Aufgabe
 - Rechenzentrum von Grund auf neu bauen
- Team
 - 2 x TNG, 2 x Sysadmins, Tester,
Anwendungsentwickler, Netzwerker
- Zeit: 1 Jahr

Akzeptanz der IaC Umgebung

- Errungenschaften
 - Automatisierung mit klarem Schichtenmodell
 - VMs bauen in 5-7 Minuten
 - großes Vertrauen in Umgebungen
 - tägliche Deployments in Produktion
- Schwierigkeiten
 - Prozesse kontinuierlich überprüfen und ändern
 - Herkömmliches CFEngine gibt wenig Struktur vor
 - Zeit in eigene High Level CFEngine Bibliothek und Dev Umgebung investieren

Das Ganze gibt es wirklich



Infrastructure as Code Projekt

- Kunde: Internetdienstleister, IT Abteilung – 100 MA
- vorher
 - Umgebungen unterschiedlich
 - manuell installiert
 - Hardware schlecht ausgelastet
 - lange Deployment Zyklen
- Aufgabe
 - Rechenzentrum von Grund auf neu bauen
- Team
 - 2 x TNG, 2 x Sysadmins, Tester,
Anwendungsentwickler, Netzwerker
- Zeit: 1 Jahr

Akzeptanz der IaC Umgebung

- Errungenschaften
 - Automatisierung mit klarem Schichtenmodell
 - VMs bauen in 5-7 Minuten
 - großes Vertrauen in Umgebungen
 - tägliche Deployments in Produktion
- Schwierigkeiten
 - Prozesse kontinuierlich überprüfen und ändern
 - Herkömmliches CFEngine gibt wenig Struktur vor
 - Zeit in eigene High Level CFEngine Bibliothek und Dev Umgebung investieren

Infrastructure as Code Projekt

- Kunde: Internetdienstleister, IT Abteilung ~100 MA
- vorher
 - Umgebungen unterschiedlich
 - manuell installiert
 - Hardware schlecht ausgelastet
 - lange Deployment Zyklen
- Aufgabe
 - Rechenzentrum von Grund auf neu bauen
- Team
 - 2 x TNG, 2 x Sysadmins, Tester, Anwendungsentwickler, Netzwerker
- Zeit: 1 Jahr

Akzeptanz der IaC Umgebung

- Errungenschaften
 - Automatisierung mit klarem Schichtenmodell
 - VMs bauen in 5-7 Minuten
 - großes Vertrauen in Umgebungen
 - tägliche Deployments in Produktion
- Schwierigkeiten
 - Prozesse kontinuierlich überprüfen und ändern
 - Herkömmliches CFEngine gibt wenig Struktur vor
 - Zeit in eigene High Level CFEngine Bibliothek und Dev Umgebung investieren

Devops is About CAMS

[] Culture

[] Automation

[] Measurement

Sharing

John Willis

Damon Edwards

Also werden Infrastrukturprojekte jetzt Entwicklungsprojekte - nicht ganz.

- Der Code ist nahe an der wahren Welt.
- Es gibt viele Abhängigkeiten.
- Schreib mal schnell einen Stub für einen Netzwerk-Switch - das geht so einfach nicht.
- Trotz Virtualisierung bleiben viele gemeinsame Ressourcen - Testen ist schwierig wie bei Code mit vielen Singletons.
- Speziell Unit-Tests sind nicht wirklich gelöst.

Tests vs. Monitore - Wo ist der Unterschied?

Tests:

- stellen einen definierten Zustand her,
- sind irgendwann beendet,
- suchen nach Fehlern aufgrund von bekannten, gewollten Veränderungen,
- laufen in spezieller Test- oder CI-Umgebung.

Monitore:

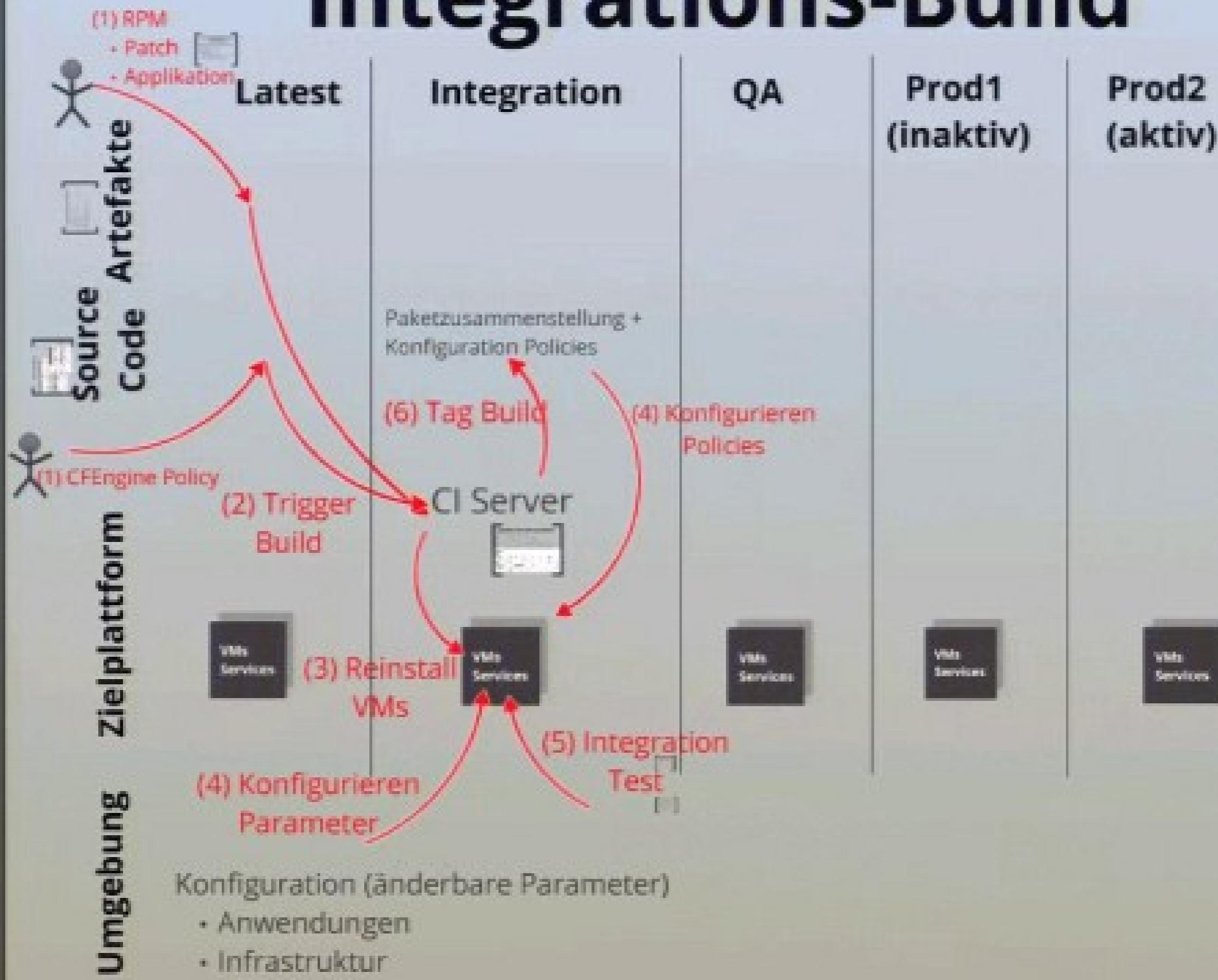
- nehmen die Welt, wie sie ist,
- laufen kontinuierlich,
- finden Störungen, obwohl sich "nichts verändert" hat,
- sind oft Teil einer gemeinsamen Infrastruktur.

Aber:

Beide Verfahren führen Prüfungen/Checks durch und liefern das Ergebnis zurück.

Gemeinsamkeiten nutzen, gemeinsame Testfälle nutzen, z.B. Fitnesse-Tests mit Icinga steuern.

Integrations-Build



Adobe Flash Player 10

Datei Ansicht Steuerung Hilfe

Integrations-Build

Umgang mit Artikeln

Zielplattform

Code Artefakte

Latest Integration QA Prod1 (inaktiv) Prod2 (aktiv)

Integrationsschritte:

- Code Push
- Build
- Deployment
- Integration
- Deployment
- Integration

Trigger: CI Trigger Rule

CI Server

CI Repository Rule

CI Configuration Parameters

Configuration (Unterstützt Parameter):

- Anwendungen
- Infrastruktur

Kontakte

- Links
- Musik
- RZ-Automatisierung
- Suchvorgänge
- Videos
- Öffentlich

prezi Anwendung Änderungsdatum: 14.06.2012 20:50
Größe: 4,32 MB
Erstelltdatum: 15.06.2012 08:45

Suchen

Größe

Ordner	Größe
ordner	4.431 KB
ordner	1 KB
endung	



VLC media player

Microsoft PowerPoint
Übersicht

Übersichts-
Blätter



z2-Automatisierung

Adobe Flash Player 10



14/44



Computer



Big Tech Day



Programme



Internet Explorer

Big Tech Day > RZ-Automatisierung >

Organisieren Ansichten Öffnen Freigaben Brinnen

Linkveritzen	Name	Änderungsdatum	Typ	Größe
Dokumente	data	15.06.2012 08:45	Dateiordner	
Bilder	prezi-app	15.06.2012 08:45	Dateiordner	
Musik	prezi	14.06.2012 20:50	Anwendung	4.431 KB
Weitere...	pwd	14.06.2012 21:35	Datei	1 KB

Ordner

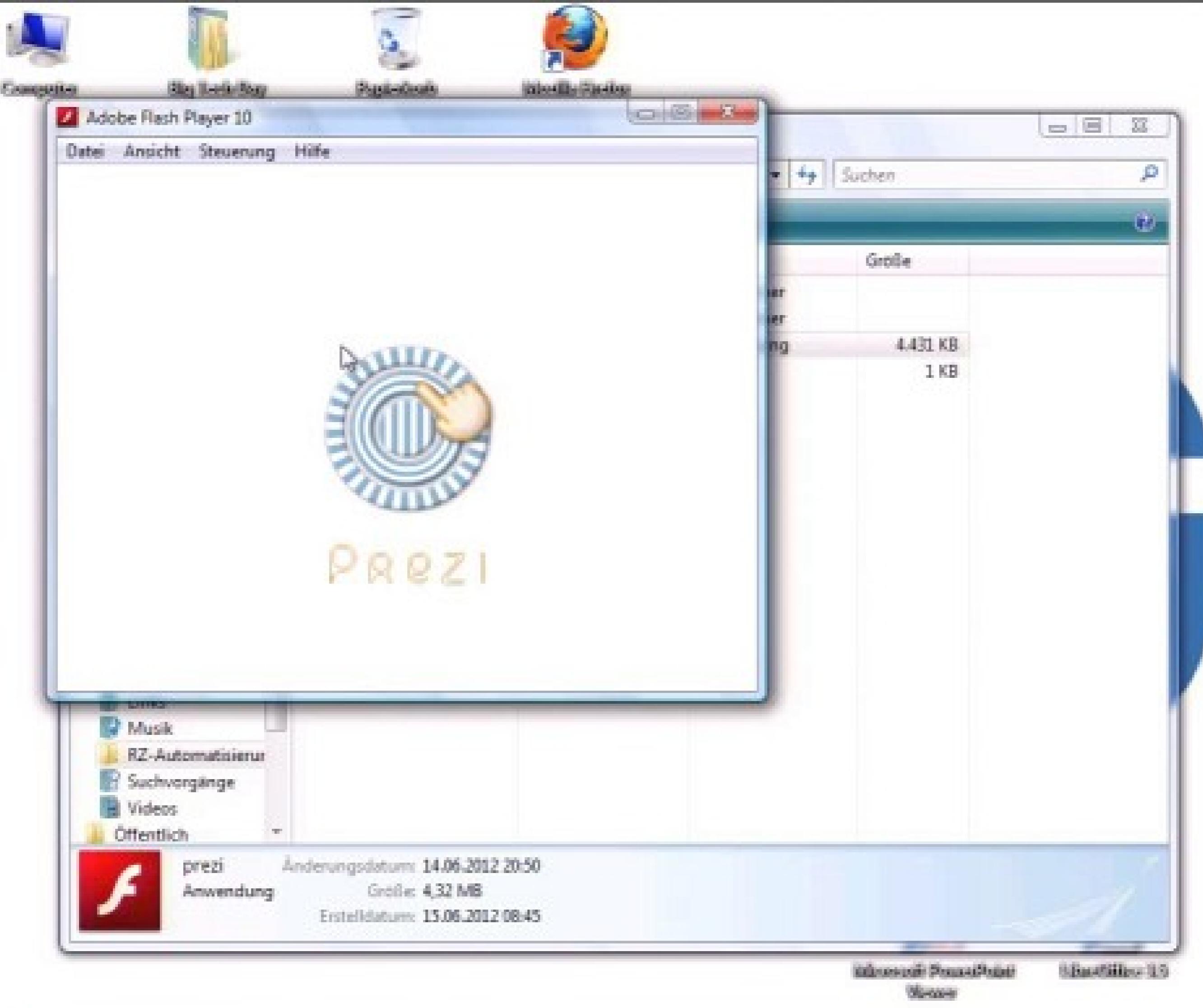
- Big Tech Day
- Bilder
- btd5-bigdata
- Desktop
- Dokumente
- Download
- Favoriten
- Gespeicherte Spiel
- Kontakte
- Links
- Musik
- RZ-Automatisierung
- Suchvorgänge
- Videos
- Öffentlich

prezi Anwendung Änderungsdatum: 14.06.2012 20:50
Größe: 4,32 MB
Erstelldatum: 15.06.2012 08:45

Überarbeitet: Prema Prabhu
Datum: 15.06.2012Überarbeitet: RZ-Automatisierung
Datum: 15.06.2012

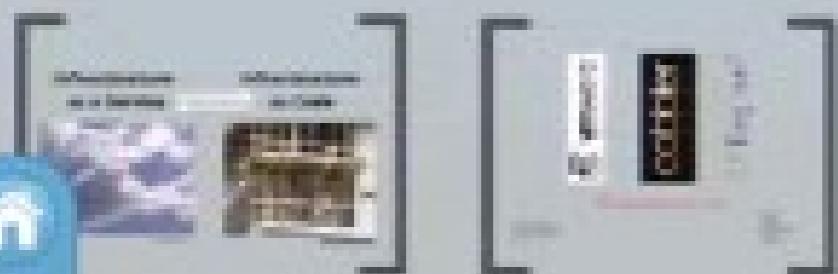
RZ-Automatisierung





VLC media player

Infrastructure as Code



[+]

**Virtualisierung und
Automatisierung - wie man sich
ein Rechenzentrum programmiert**

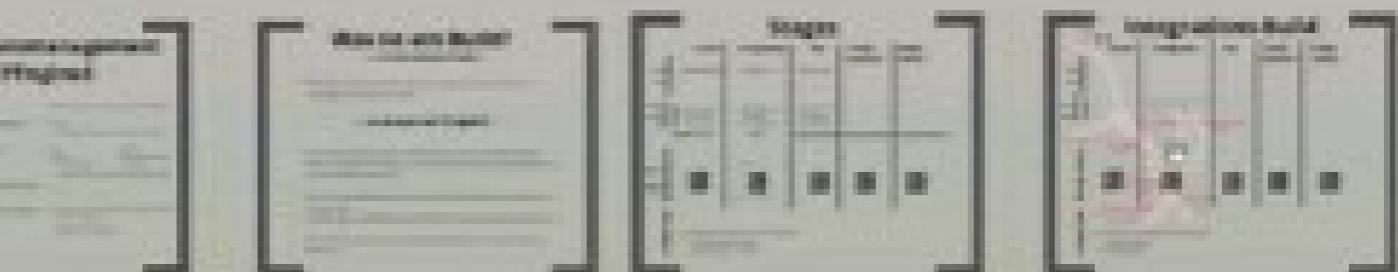
Fazit



Werkzeuge

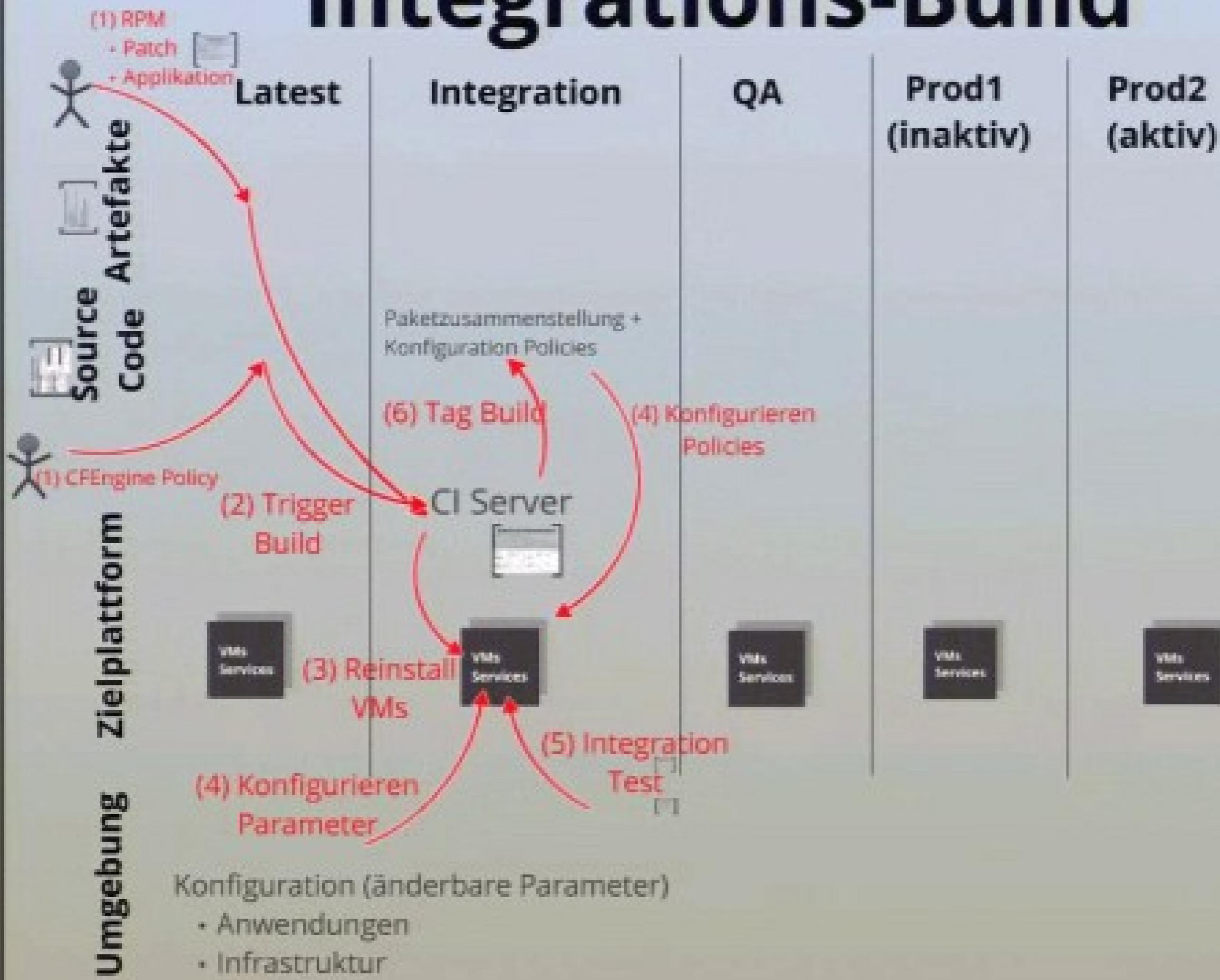


Continuous Delivery



Ressourcen Planung, PXE Boot, DHCP, Service + Stage Datei
Virtualisierung, Netzwerkinstallation, Konfiguration

Integrations-Build





Computer



Big Tech Day



Programme



Internet Explorer

Big Tech Day > RZ-Automatisierung >

Organisieren Ansicht Öffnen Freigeben Brennen

	Name	Änderungsdatum	Typ	Größe
	data	15.06.2012 08:45	Ordner	
	prezi-app	15.06.2012 08:45	Ordner	
	prezi	14.06.2012 20:50	Anwendung	4.431 KB
	pwd	14.06.2012 21:35	Datei	1 KB

Linkfavoriten Dokumente Bilder Musik Weitere: >

Ordner

- Big Tech Day
 - Bilder
 - btd5-bigdata
 - Desktop
 - Dokumente
 - Download
 - Favoriten
 - Gespeicherte Spiel
 - Kontakte
 - Links
 - Musik
 - RZ-Automatisierung
 - Suchvorgänge
 - Videos
- Öffentlich

prezi Anwendung Änderungsdatum: 14.06.2012 20:50
Größe: 4,32 MB
Erstelldatum: 15.06.2012 08:45

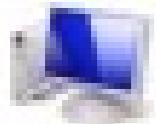
Überarbeitet: Prema Prabhu
Datum: 15.06.2012Überprüft: -
Datum: -

VLC media player



RZ-Automatisierung

DE 14/17



Computer



Big Tech Day



Programme



Java-Programme

Big Tech Day > RZ-Automatisierung

Organisieren Ansicht Öffnen Freigeben Brennen

	Name	Änderungsdatum	Typ	Größe
	data	15.06.2012 08:45	Ordner	
	prezi-app	15.06.2012 08:45	Ordner	
	prezi	14.06.2012 20:50	Anwendung	4.431 KB
	pwd	14.06.2012 21:35	Datei	1 KB

Linkleisten: Dokumente Bilder Musik Weitere: Ordner Big Tech Day Bilder btd5-bigdata Desktop Dokumente Download Favoriten Gespeicherte Spiel Kontakte Links Musik RZ-Automatisierung Suchvorgänge Videos Öffentlich

prezi Anwendung Änderungsdatum: 14.06.2012 20:50 Größe: 4,32 MB Erstellt datum: 15.06.2012 08:45

Überarbeitet: Prema Prabhu
Datum:

Überarbeitet: 15



VLC media player



RZ-Automatisierung

DE



Computer



Big Tech Day



Programme



Firefox

Big Tech Day

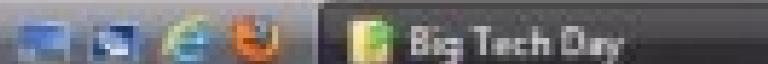
Organisieren Ansichten Explorer Freigeben Brennen

	Name	Änderungsdatum	Größe	Ordnerpfad
	Bilder Dateiordner			btd5-bigdata Dateiordner
	Desktop Dateiordner			Dokumente Dateiordner
	Download Dateiordner			Favoriten Dateiordner
	Gespeicherte Spiele Dateiordner			Kontakte Dateiordner
	Links Dateiordner			Musik Dateiordner
	RZ-Automatisierung Dateiordner			Suchvorgänge Dateiordner
	Videos Dateiordner			BTDS-Scrum-in-Large-Proje... Adobe Acrobat Document 2,33 MB
	BTDS-Scrum-in-Large-Proje... Microsoft PowerPoint Presen... 1.03.14.ppt			e-volo Vertrag 2012 OpenDocument Präsentation mso-mlc

RZ-Automatisierung
Dateiordner
Änderungsdatum: 15.06.2012 08:45

Überarbeitet: Präsentations...
Überarbeitet: 05.06.2012

Überarbeitet: 05.06.2012





Computer



Big Tech Day



Programme



Internet Explorer

Big Tech Day

Suchen

Organisieren Ansichten Explorer Freigeben Brennen

	Name	Änderungsdatum	Größe	Ordnerpfad
Dokumente	Desktop	Desktop	Dateiordner	Dokumente
Bilder	Download	Download	Dateiordner	Favoriten
Musik	Gespeicherte Spiele	Gespeicherte Spiele	Dateiordner	Kontakte
Weitere:	Links	Links	Dateiordner	Musik
Big Tech Day	RZ-Automatisierung	RZ-Automatisierung	Dateiordner	Suchvorgänge
Bilder	Videos	Videos	Dateiordner	BTDS-Scrum-in-Large-Proje...
bed5-bigdata	BTDS-Scrum-in-Large-Proje...	BTDS-Scrum-in-Large-Proje...	Microsoft PowerPoint Pres...	Adobe Acrobat Document
Desktop	TMG Tech Day_CIOs	TMG Tech Day_CIOs	OpenDocument Präsentation	2,33 MB
Dokumente	Cerner_Telefonica Germany...	Cerner_Telefonica Germany...	Adobe Acrobat Document	50,8 MB
Download	PDF	PDF		
Favoriten				
Gespeicherte Spie				
Kontakte				
Links				
Musik				
RZ-Automatisier				
Suchvorgänge				
Videos				
Öffentlich				

RZ-Automatisierung
Dateiordner
Änderungsdatum: 15.06.2012 08:45

Datenbank-PowerPoint
BrowserÜbersicht
Übersicht

VLC media player



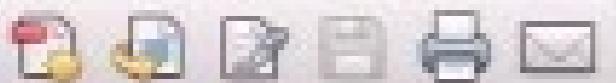
Big Tech Day



DE



14:45



1 / 40

73,7%



Scrum in Large Projects

Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger

Scrum in Large Projects

Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger

Scrum in Large Projects

Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger

Scrum in Large Projects

Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger