Infrastructure as Code in Practice

Überblick und Praxis-Beispiele

SVA

/ whoami

CHRISTIAN STANKOWIC

Senior System Engineer

Geschäftsstelle Mitte

https://cstan.io

Mobil: +49 151 18025982

Tel.: +49 6122 536 0

Fax: +49 6122 536 399

christian.stankowic@sva.de

Borsigstraße 14
65205 Wiesbaden

Ihr Weg zu uns

/ whoami

Katzenbilder, Memes und Bloggen

Enterprise Linux

Monitoring

Virtualisierung

Configuration Management

Automatisierung

/ Agenda

1 / SVA

2 / Motivation und Workflow

3 / Configuration Management

4 / Packer

5 / Terraform

6 / Compliance as Code

SVA

/ Profil und Unternehmensziel

Größter **inhabergeführter System-Integrator** Deutschlands

Starkes Wachstum mit mehr als **1.200 Mitarbeitern** in Deutschland

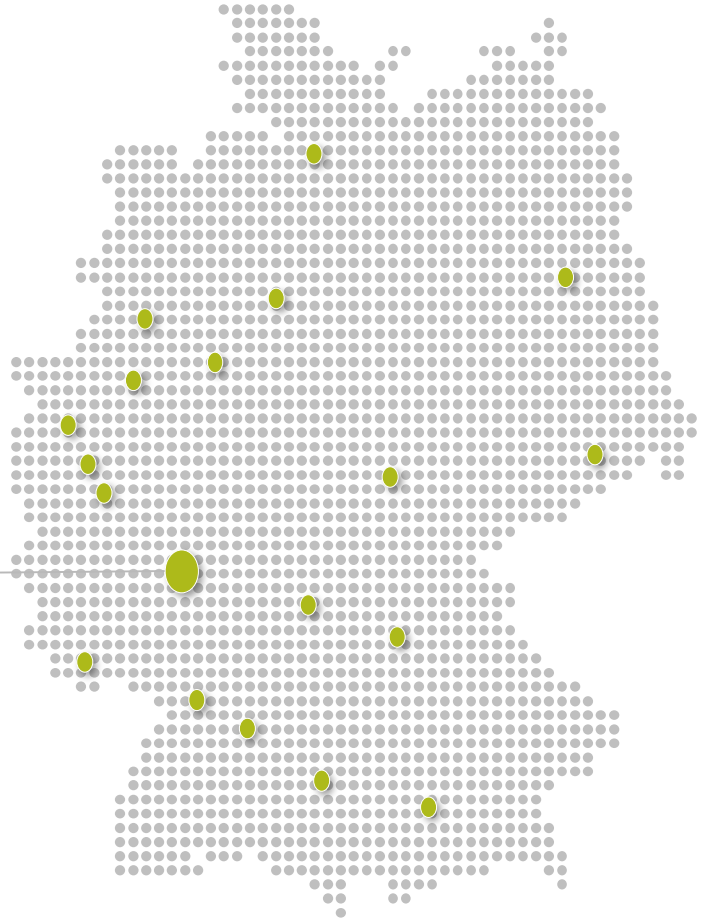
Unternehmerische **Ziele**

Unsere Kunden begleiten wir **langfristig** in allen Prozessschritten.

Über uns

/ Unternehmen

19
Standorte



6 TOP
Branchen

Automotive

Retail

Public

Maschinen & Anlagenbau

Finance & Insurance

Tele-kommunikation

/ Produktportfolio auf einen Blick

Datacenter-Infrastructure

- Storage- & Server-Systeme
- IP-Netzwerk-Infrastruktur
- Software Defined Data Center / Container
- Cloud & Automation
- Hyperconverged

Mainframe

- Services / Consulting
- zHosting
- zBusiness Services
- IBM System z Hardware
- Managed Services

End User Computing

- Application & Desktop (CAD/E) Virtualization
- Virtual Workspace & Mobility
- Application, Information & Device Management
- Unified Endpoint Management

Big Data Analytics & IoT

- Big Data Plattformen
- IT Operations Analytics
- Business Analytics und IoT

SAP

- SAP Infrastruktur
- SAP Technologieberatung
- SAP Solution Manager
- SAP Analytics
- Betriebsunterstützung/ Managed Services

Business Continuity

- Archivierung
- Backup und Recovery
- Disaster Recovery
- High Availability
- Notfallplanung

Service Management

- Software Asset Management
- Enterprise Asset Management
- IT Service Management
- Enterprise Service Management

IT Security

- Information Security und Compliance Consulting
- IT-Security-Architektur und -Integration
- Penetration Testing
- IT-Security Managed Services
- Security Incident Response

SVA Produkte

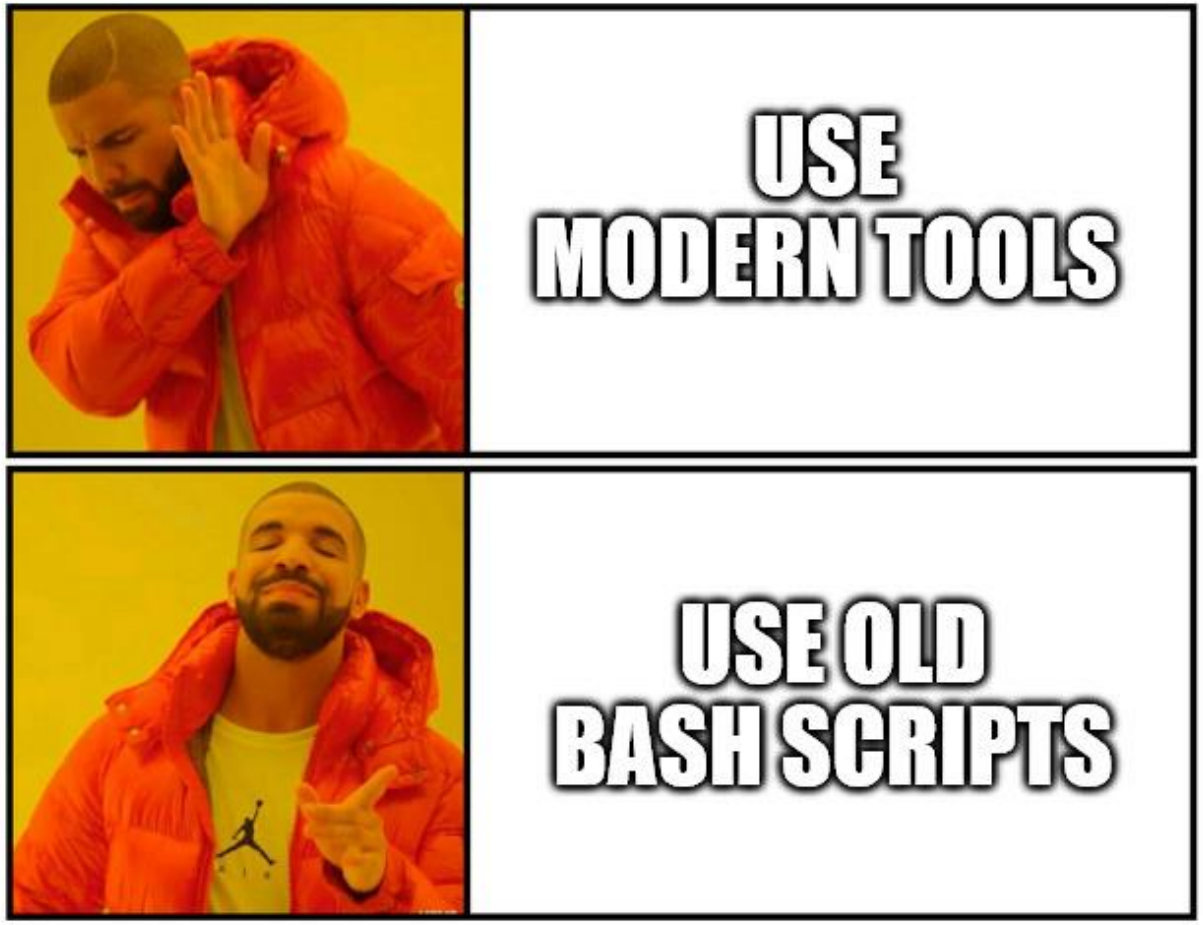
- BVQ
- IDR
- Liberyse
- medPower

Motivation

/ Motivation

- Systeme und Applikationen werden stetig komplexer
- Es muss immer mehr Infrastruktur in kürzerer Zeit gewartet werden
- Manuelle Administration nahezu unmöglich
- Resultate:
  - Entwickler und Administratoren verwachsen immer mehr (*und nennen sich dann Senior DevOps Engineer ☺*)
  - Infrastruktur-Themen bedienen sich Technologien der Software-Entwicklung um Automatisierung voranzutreiben

    ➔ Infrastructure as Code



/ Infrastructure as Code (*IaC*)

- Skripte als Hilfsmittel – quasi „*Vorform*“ von IaC
- Nachteile:
  - Sequentielle Abarbeitung
    - Eventuelle Fehler bei erneuter Ausführung
  - Versionierung
  - Messbarkeit
    - Jedes Soll-Ergebnis muss manuell getestet werden (*if-Abfrage*)
  - Reproduzierbarkeit
    - Eventuelle Fehler auf Systemen mit anderer Shell(*version*)

- Komplexe Infrastruktur-Setups werden in maschinenlesbaren Code abgebildet
  - Dieser definiert den Soll-Zustand
- Code kann versioniert werden (*z.B. über Git*)
- IaC-Software setzt Code in native Ressourcen (*Server, Software, Konfigurationsdateien,…*)
- Nachträgliche Änderungen können kontrolliert auf allen Systemen ausgerollt werden

/ Infrastructure as Code (*IaC*)

```
#!/bin/sh
yum install -y httpd
sed -i "s/OptionXYZ/NoOptionXYZ/g"
/etc/http/httpd.conf
echo "PermitRootLogin yes" >>
/etc/ssh/sshd_config
systemctl restart sshd
```

- Paket wird ungewollt aktualisiert
- Ähnliche Optionen werden ebenfalls ersetzt

- Option wird mehrfach gesetzt und verursacht womöglich Fehler
- Dienst-Status nicht klar definiert (*Neustart eines gestoppten Dienstes könnte fehlschlagen*)

/ Infrastructure as Code (*IaC*)

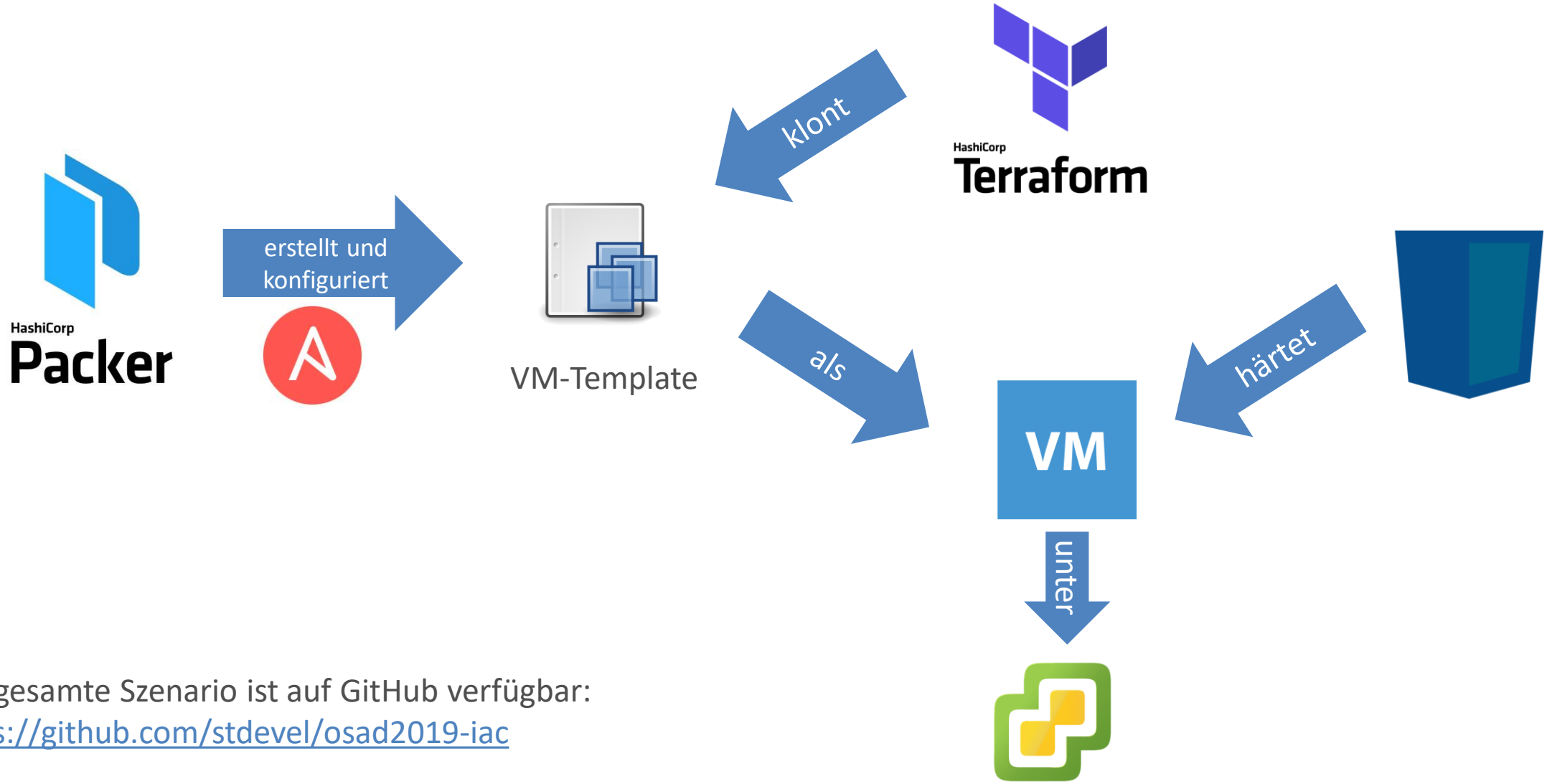
```
package 'httpd'

template '/etc/httpd/httpd.conf' do
  source 'httpd.conf'
done
template '/etc/ssh/sshd_config' do
 source 'sshd_conf'
done

service 'sshd' do
  action [ :enable, :start ]
done
```

← Paket wird nur installiert, falls noch nicht installiert

← Datei wird aus Template erstellt, Makros möglich

← Dienst wird – falls nötig – aktiviert und gestartet

/ Workflow



Das gesamte Szenario ist auf GitHub verfügbar:
https://github.com/stdevel/osad2019-iac

Configuration Management

„Welches Tool soll ich denn wählen, lieber Berater?“

PAULA PINKEPANK, MÖCHTE ANONYM BLEIBEN

SVA

It depends.


NYPD
p

/ Qual der Wahl: Configuration Management

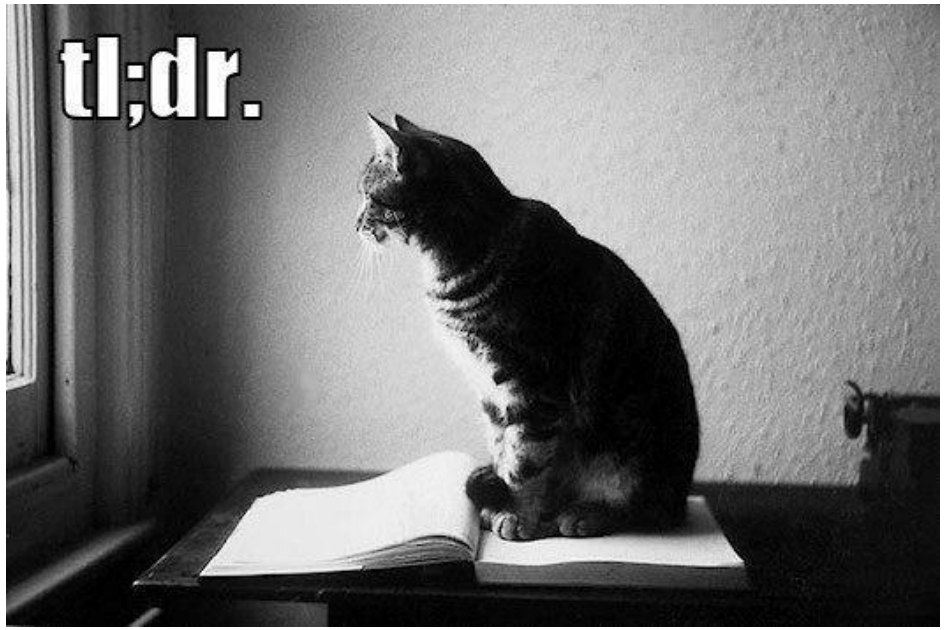
	Ansible	Chef	SaltStack	Puppet
Erscheinungsjahr	2012	2005	2011	2005
Agent	Nein (SSH)	Ja (Client)	Jein (Minion|SSH-only)	Ja (Agent)
Funktionsweise	Push	Pull	Push	Pull
Sprache	YAML und Python (Administratorfreundlich)	Ruby DSL (Entwicklerorientiert)	YAML und Python (Administratorfreundlich)	Puppet DSL (abgewandeltes Ruby)
Ressourcen (2019)	2800 Module	280 Ressourcen	1200 Module	50 Ressourcen
Community (2019)	Ansible Galaxy 1.600 Mitglieder 22.000 Rollen	Chef Supermarket 78.000 Mitglieder 3.800 Cookbooks	Salt Stack Formulas 50 Mitglieder 300 Formulas	Puppetforge 6.300 Module
Client-Support (Auszug)	Alles, was SSH oder WinRM kann	AIX, Debian/Ubuntu, RHEL/CentOS, openSUSE/SLES, Microsoft Windows, Oracle Solaris		
Hochverfügbarkeit	Active/Active	Active/Passive	Active/Active	Active/Passive
Enterprise-Version	Ansible Tower	Chef Infra	SaltStack Enterprise	Puppet Enterprise
Lizenz	GPL	Apache Chef EULA (ab 2020)	Apache	Apache

/ Neues Chef-Lizenzierungsmodell

Im Mai 2019 wurde das Chef EULA verabschiedet:

- *Kommerzielle Nutzung* erfordert nun immer eine Lizenz
  - *„Einsatz im Unternehmen, das direkt oder indirekt auf Gewinn ausgerichtet ist“*
- Nicht-kommerzielle Nutzung nur noch für private und Lernzwecke
  - Experimental Use schließt (*Pre*-)Production aus
- Anpassen der Software ist untersagt
- Teile der Software können als Source Code veröffentlicht werden, in diesem Fall gelten die entsprechenden Lizenzen (*Apache 2.0*)
- Für kommerzielle Kunden aufgrund von Lizenzabkommen keine Veränderung der Lage

- Automatische Zustimmung der Erfassung anonymisierter Daten
- Bisherige Software erreicht **EOL im April 2020**
  - Angebot von Binärpaketen danach derzeit unklar, eventuell manuelle Quellcode-Übersetzung notwendig

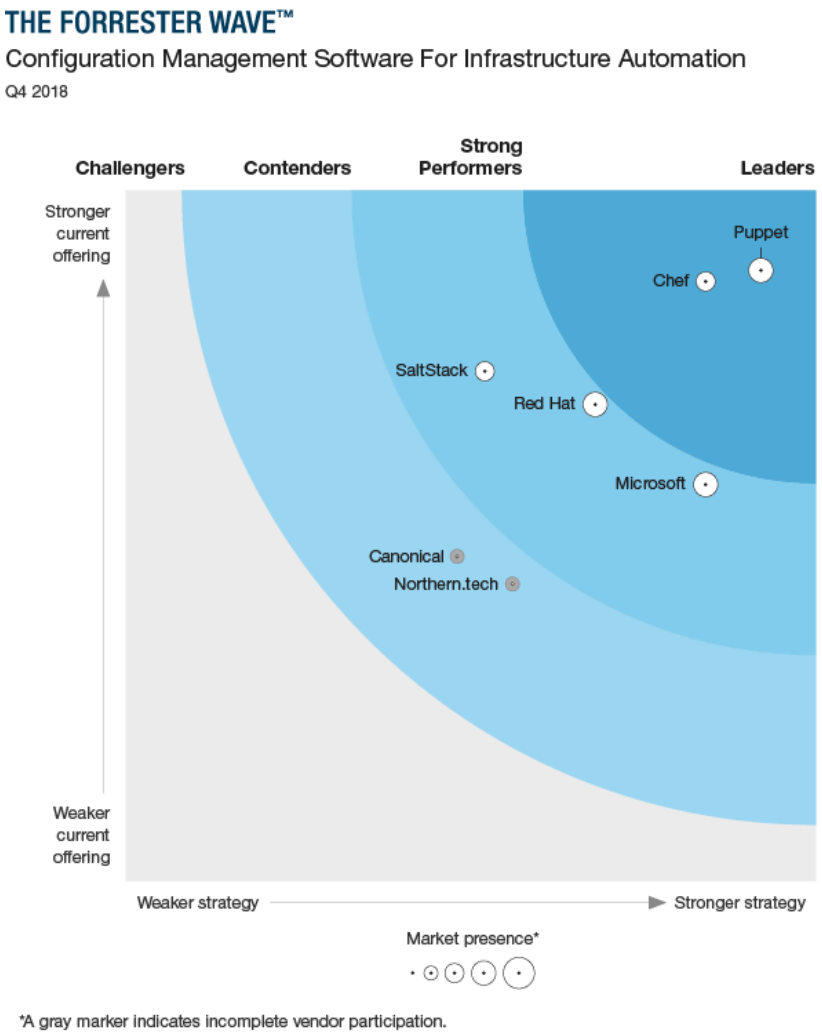


Forrester Marketingstudie 2018

/ Qual der Wahl: Configuration Management

Zusammenfassung:

- Puppet und Chef sind altbekannte Player
- **Puppet** ist aufgrund Stabilität und Governance sehr weit verbreitet, aber **nicht leicht zu erlernen**
- Chef bietet sehr **ausgefeilte Security-Auditingtools** (*InSpec, RSpec*), die auch von anderen Produkten verwendet werden
- SaltStack und Ansible sind neuere Player, die den Markt jedoch stark beeinflussen
- **Ansible** ist aufgrund seiner **Einfachheit** und dem agentenlosen Design sehr beliebt
- SaltStack verfügt über einen Event-Bus und kann auf Infrastruktur-Änderungen in **Echtzeit** reagieren



Quelle: https://reprints.forrester.com/#/assets/2/129/RES143273/reports

Packer

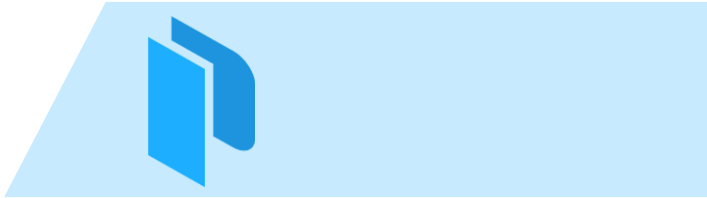
SVA

/ Packer

- Packer ist ein Tool zur automatisierten Erstellung von **Golden Images**
- Geschrieben in Go, open-source: https://packer.io
- Zahlreiche Plugins liefern zusätzliche Funktionalität
- Kompletter Prozess wird in zentraler YAML-Konfigurationsdatei definiert
  - Erstellen der VM
  - Einlegen der CD/DVD
  - Durchführen der Installation
  - Installieren von Updates und Paketen
  - Anwenden von Konfiguration
  - Aufräumen
  - …

HashiCorp
Packer

/ Packer



Erstellen von VMs/Container via:

- VMware
- VirtualBox
- XEN / KVM
- Docker
- ...

Markieren als Template, Upload in Cloud:

- AWS
- GCE
- Docker
- ...

Builder → **Provisioner** → **Post-Processor**

Konfiguration via Skripte oder Configuration Management:

- (*Power*)Shell
- Ansible
- Chef
- Puppet
- Salt

```
{
  "variables" : {
    "cpus": "2",
    "memory": "2048",
    "memory_reserve_all": "false",
    "template_name": "template_centos7",
    "http_proxy": "{{env `http_proxy`}}",
    "https_proxy": "{{env `https_proxy`}}",
    "disk_size": "20480",
    "disk_controller_type": "pvscsi",
    "disk_thin_provisioned": "false",
    "autoinst_cfg": "http/ks.cfg",
    "config_user": "vagrant",
    "config_pass": "vagrant",
    "iso_path": "[SSCFILER] software\\CentOS\\",
    "iso_file": "CentOS-7-x86_64-Minimal-1810.iso",
    "iso_checksum": "38d5d51d9d100fd73df031ffd6bd8b1297ce24660dc8c13a3b8b4534a4bd291c",
    "iso_checksum_type": "md5",
    "vsphere_server": "vc6.labwi.sva.de",
    "vsphere_datacenter": "LABWI",
    "vsphere_username": "DO_NOT_PASTE_CREDENTIALS_IN_THIS_FILE",
    "vsphere_password": "DO_NOT_PASTE_CREDENTIALS_IN_THIS_FILE",
    "vsphere_insecure_connection": "true",
    "vsphere_datastore": "VMFS01-Prod-SWV3",
    "vsphere_cluster": "UCS-Cluster",
    "network": "PoC_Linux",
    "network_card": "vmxnet3",
    "folder": "CStankowic"
  },
```

```
30  "builders": [
31    {
32      "type": "vsphere-iso",
33      "vcenter_server": "{{user `vsphere_server`}}",
34      "datacenter": "{{user `vsphere_datacenter`}}",
35      "username": "{{user `vsphere_username`}}",
36      "password": "{{user `vsphere_password`}}",
37      "insecure_connection": "{{user `vsphere_insecure_connection`}}",
38      "datastore": "{{user `vsphere_datastore`}}",
39      "datacenter": "{{user `vsphere_datacenter`}}",
40      "cluster": "{{user `vsphere_cluster`}}",
41      "vm_name": "{{user `template_name`}}",
42      "network": "{{user `network`}}",
43      "network_card": "{{user `network_card`}}",
44      "CPUs": "{{user `cpus`}}",
45      "cpu_cores": 1,
46      "RAM": "{{user `memory`}}",
47      "RAM_reserve_all": "{{user `memory_reserve_all`}}",
48      "disk_controller_type": "{{user `disk_controller_type`}}",
49      "disk_thin_provisioned": "{{user `disk_thin_provisioned`}}",
50      "floppy_files": [
51        "{{user `autoinst_cfg`}}"
52      ],
53      "boot_command": [
54        "<tab> inst.text inst.ks=hd:fd0:/ks.cfg net.ifnames=0 ip=dhcp<enter><wait>"
55      ],
56      "boot_wait": "10s",
57      "disk_size": "{{user `disk_size`}}",
58      "guest_os_type": "centos7_64Guest",
59      "http_directory": "http",
60      "iso_checksum": "{{user `iso_checksum`}}",
61      "iso_checksum_type": "{{user `iso_checksum_type`}}",
62      "iso_paths": [
63        "{{user `iso_path`}}/{{user `iso_file`}}"
64      ],
65      "shutdown_command": "sleep 30 ; echo 'vagrant' | sudo -S /sbin/poweroff",
66      "shutdown_timeout": "50m",
67      "ssh_password": "vagrant",
```

```
74  "provisioners": [
75    {
76      "type": "shell",
77      "execute_command": "{{ .Vars }} sudo -E -S sh '{{ .Path }}'",
78      "script": "scripts/update.sh"
79    },
80    {
81      "type": "file",
82      "source": "../ansible/simple-webserver.yml",
83      "destination": "/tmp/simple-webserver.yml"
84    },
85    {
86      "type": "ansible-local",
87      "playbook_file": "/tmp/simple-webserver.yml"
88    },
89    {
90      "type": "shell",
91      "execute_command": "{{ .Vars }} sudo -E -S sh '{{ .Path }}'",
92      "script": "scripts/vagrant.sh"
93    },
94    {
95      "type": "shell",
96      "execute_command": "{{ .Vars }} sudo -E -S sh '{{ .Path }}'",
97      "script": "scripts/cleanup.sh"
98    }
99  ]
100 }
101
```

```
$ ./packer validate centos7-x86_64.json
Template validated successfully.
$ ./packer build -var-file=credentials-cstankow.json centos7-x86_64.json
vsphere-iso output will be in this color.

==> vsphere-iso: Creating VM...
==> vsphere-iso: Customizing hardware...
==> vsphere-iso: Mount ISO images...
==> vsphere-iso: Creating floppy disk...
    vsphere-iso: Copying files flatly from floppy_files
    vsphere-iso: Copying file: http/ks.cfg
    vsphere-iso: Done copying files from floppy_files
    vsphere-iso: Collecting paths from floppy_dirs
    vsphere-iso: Resulting paths from floppy_dirs : []
    vsphere-iso: Done copying paths from floppy_dirs
==> vsphere-iso: Uploading created floppy image
==> vsphere-iso: Adding generated Floppy...
==> vsphere-iso: Starting HTTP server on port 8650
==> vsphere-iso: Set boot order temporary...
==> vsphere-iso: Power on VM...
==> vsphere-iso: Waiting 10s for boot...
```

CentOS 7

Install CentOS 7
Test this media & install CentOS 7

Troubleshooting >

```
> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=CentOS\x207\x20x86_64 rd.live
.check quiet inst.text inst.ks=hd:fd0:/k_
```

```
 * shell is available on TTY2
 * when reporting a bug add logs from /tmp as separate text/plain attachments
08:11:02 Not asking for VNC because of an automated install
08:11:02 Not asking for VNC because text mode was explicitly asked for in kickstart
Starting automated install..............
Checking software selection
Generating updated storage configuration
Checking storage configuration...

================================================================================
================================================================================
Installation

 1) [x] Language settings                 2) [x] Time settings
        (English (United States))                (Europe/Berlin timezone)
 3) [x] Installation source               4) [x] Software selection
        (Local media)                            (Custom software selected)
 5) [x] Installation Destination          6) [x] Kdump
        (Custom partitioning selected)           (Kdump is enabled)
 7) [x] Network configuration
        (Wired (eth0) connected)
================================================================================
================================================================================
Progress
Setting up the installation environment
.
Creating disklabel on /dev/sda
.
Creating lvmpv on /dev/sda2
.
Creating ext4 on /dev/mapper/vg_system-lv_tmp
.
Creating ext4 on /dev/mapper/vg_system-lv_root
.
Creating ext4 on /dev/mapper/vg_system-lv_home
_
[anaconda] 1:main* 2:shell  3:log  4:storage-log  5:program-log      Switch tab: Alt+Tab | Help: F1
```

```
==> vsphere-iso: Uploading ../ansible/simple-webserver.yml => /tmp/simple-webserver.yml
==> vsphere-iso: Provisioning with Ansible...
    vsphere-iso: Creating Ansible staging directory...
    vsphere-iso: Creating directory: /tmp/packer-provisioner-ansible-local/5da038cb-65b3-6378-99e6-9f110
    vsphere-iso: Uploading main Playbook file...
    vsphere-iso: Uploading inventory file...
    vsphere-iso: Executing Ansible: cd /tmp/packer-provisioner-ansible-local/5da038cb-65b3-6378-99e6-9f1
CE_COLOR=1 PYTHONUNBUFFERED=1 ansible-playbook /tmp/packer-provisioner-ansible-local/5da038cb-65b3-6378-
-webserver.yml --extra-vars "packer_build_name=vsphere-iso packer_builder_type=vsphere-iso packer_http_a
-o IdentitiesOnly=yes"  -c local -i /tmp/packer-provisioner-ansible-local/5da038cb-65b3-6378-99e6-9f1102
r-ansible-local242111089
    vsphere-iso:
    vsphere-iso: PLAY [Install simple web server] ***********************************************
    vsphere-iso:
    vsphere-iso: TASK [Install webserver] *******************************************************
    vsphere-iso: changed: [127.0.0.1]
    vsphere-iso:
    vsphere-iso: TASK [Enable webserver] ********************************************************
    vsphere-iso: changed: [127.0.0.1]
    vsphere-iso:
    vsphere-iso: TASK [Disable firewall as we also like to live dangerously] ********************
    vsphere-iso: changed: [127.0.0.1]
    vsphere-iso:
    vsphere-iso: PLAY RECAP *********************************************************************
    vsphere-iso: 127.0.0.1                     : ok=3    changed=3    unreachable=0    failed=0    skipped=
d=0
    vsphere-iso:
```

template_centos7_cstankow

AKTIONEN

Übersicht | Überwachen | Konfigurieren | Berechtigungen | Datenspeicher | Netzwerke

Gastbetriebssystem:	CentOS 7 (64-bit)
Kompatibilität:	ESXi 6.7 und höher (VM-Version 14)
VMware Tools:	Wird nicht ausgeführt, Version:10336 (Verwaltet durch Gast)
	Weitere Informationen
DNS-Name:	localhost.localdomain
IP-Adressen:	
Host:	ucs-esx1.labwi.sva.de

VM-Hardware

CPU	2 CPU(s)
Arbeitsspeicher	2048 MB
Festplatte 1	20 GB
Netzwerkadapter 1	PoC_Linux (Getrennt)
CD-/DVD-Laufwerk 1	Getrennt
Grafikkarte	4 MB
VMCI-Gerät	Gerät auf dem PCI-Bus der virtuellen Maschine, das die Kommunikationsschnittstelle der virtuellen Maschine

Hinweise

Hinweise bearbeiten...

Benutzerdefinierte Attribute

Attribut
Backup Status
Last Backup
credentials

Terraform

Deploy me if you can

/ Terraform

- Tool zum Verwalten komplexer Infrastruktur-Landschaften
- Sinnvoll für die zentrale Verwaltung mehrerer unterschiedlicher Plattformen
  - z.B. verschiedene On-Premise Hypervisor + Cloud
- Go, open-source: https://terraform.io

- Definition von Infrastruktur-Ressourcen in der eigenen Markup-Sprache **HCL*** oder JSON
- Sehr große Anzahl verfügbarer Integrationen (*Provider*)
  - Derzeit ca. 100 Erweiterungen
- Integration in gängiges Configuration Management (*Provisioner*)

D.R.Y.
Don‘t repeat yourself

**HashiCorp Configuration Language*

Provider

/ Terraform

HashiCorp Terraform

Cloud
- AWS
- Azure
- Azure Stack
- Google Cloud Platform
- Oracle Cloud Infrastructure
- Oracle Cloud Platform
- Oracle Public Cloud
- VMware NSX-T
- Vmware vSphere
- VMware vCloud Director
- DigitalOcean
- Hetzner Cloud
- Nutanix
- OpenStack
- 1&1

Infrastruktur
- Docker
- Kubernetes
- Helm
- Terraform
- Consul
- Rancher
- Vault
- Chef
- Cobbler

Netzwerk
- Cisco ASA
- FortiOS
- Palo Alto
- F5 BIG-IP
- PowerDNS
- UltraDNS
- DNS
- DNSimple

Datenbanken
- MySQL
- PostgreSQL
- InfluxDB
- Grafana

Monitoring
- Icinga2
- Datadog
- Runscope

Community
- Active Directory
- Artifactory
- Azure DevOps
- CouchDB
- Elasticsearch
- ElephantSQL
- GitHub / GitLab
- IBM Cloud
- Infoblox
- Atlassian Jira
- Apache Kafka
- Kibana
- VMware NSX-V
- oVirt
- Proxmox
- Sensu
- VMware vRealize Automation
- Win DNS
- Microsoft SCVMM

/ Terraform

HashiCorp Terraform

- Eine oder mehrere aufeinander aufbauende **Ressourcen** werden in Konfiguration referenziert
  - z.B. VM ➔ Compute Resource, Netzwerk, Storage,…
  - z.B. Netzwerk-Konfiguration ➔ Hostname, IP, Netzmaske,…
- Um die Wiederverwendung zu Erleichtern gilt es Ressourcenblöcke durch **Variablen** modular zu halten
  - z.B. Standard-Einstellungen, Netzwerk-Konfiguration
- Terraform hat immer Kenntnis über den Soll- und Ist-Stand der zu verwaltenden Ressourcen ➔ `terraform.tfstate`
  - Bei späteren Änderungen wird nicht alles neu konfiguriert
  - In der Zwischenzeit manuell durchgeführte Änderungen werden **überschrieben**

- Terraform-Konfiguration besteht aus:
  - `main.tf` und weiteren `.tf`-Dateien ➔ Ressourcen
  - Variablen in `.tfvars`-Dateien ➔ Konfiguration
  - Konfigurationen lassen sich als **Module** wiederverwenden
- Workflow
  1. Konfiguration validieren:
     `$ terraform validate`
  2. Eventuell benötigte Plugins installieren:
     `$ terraform init`
  3. Simulation / Auswirkungen zeigen:
     `$ terraform plan`
  4. Änderungen übernehmen:
     `$ terraform apply`
  5. Objekte verwerfen:
     `$ terraform destroy`

```
// provider definition
provider "vsphere" {
  vsphere_server = "${var.vsphere_server}"
  user       = "${var.vsphere_user}"
  password   = "${var.vsphere_password}"

  allow_unverified_ssl = true
}



// VM resource definition
resource "vsphere_virtual_machine" "vm" {
  name             = "${var.vm_hostname}"
  resource_pool_id = "${data.vsphere_compute_cluster.cluster.resource_pool_id}"
  datastore_id     = "${data.vsphere_datastore.datastore.id}"

  num_cpus = "${var.vm_cpus}"
  cpu_hot_add_enabled = true
  cpu_hot_remove_enabled = true
  memory   = "${var.vm_memory}"
  memory_hot_add_enabled = true

  guest_id = "${data.vsphere_virtual_machine.template.guest_id}"

  scsi_type = "${data.vsphere_virtual_machine.template.scsi_type}"

  network_interface {
    network_id   = "${data.vsphere_network.network.id}"
    adapter_type = "${data.vsphere_virtual_machine.template.network_interface_types[0]}"
  }
```

```
// data definitions
data "vsphere_datacenter" "dc" {
  name          = "${var.vm_dc}"
}

data "vsphere_datastore" "datastore" {
  name          = "${var.vm_datastore}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}

data "vsphere_compute_cluster" "cluster" {
  name          = "${var.vm_cluster}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}

data "vsphere_network" "network" {
  name          = "${var.user_network}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}

data "vsphere_virtual_machine" "template" {
  name          = "${var.template_name}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}
```

```
λ terraform init
Initializing the backend...

Initializing provider plugins...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.vsphere: version = "~> 1.13"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
λ terraform plan -var-file=demo_vm.tfvars -var-file=credentials-cstankow.tfvars
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.vsphere_datacenter.dc: Refreshing state...
data.vsphere_datastore.datastore: Refreshing state...
data.vsphere_compute_cluster.cluster: Refreshing state...
data.vsphere_network.network: Refreshing state...
data.vsphere_virtual_machine.template: Refreshing state...

------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # vsphere_virtual_machine.vm will be created
  + resource "vsphere_virtual_machine" "vm" {
      + boot_retry_delay                       = 10000
      + change_version                         = (known after apply)
      + cpu_hot_add_enabled                    = true
      + cpu_hot_remove_enabled                 = true
      + cpu_limit                              = -1
      + cpu_share_count                        = (known after apply)
      + cpu_share_level                        = "normal"
```

cmd - terraform apply -var-file=demo_vm.tfvars -var-file=credentials-cstankow.tfvars

```
λ terraform apply -var-file=demo_vm.tfvars -var-file=credentials-cstankow.tfvars
data.vsphere_datacenter.dc: Refreshing state...
data.vsphere_datastore.datastore: Refreshing state...
data.vsphere_virtual_machine.template: Refreshing state...
data.vsphere_compute_cluster.cluster: Refreshing state...
data.vsphere_network.network: Refreshing state...

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # vsphere_virtual_machine.vm will be created
  + resource "vsphere_virtual_machine" "vm" {
      + boot_retry_delay                       = 10000
      + change_version                         = (known after apply)
      + cpu_hot_add_enabled                    = true
      + cpu_hot_remove_enabled                 = true
      + cpu_limit                              = -1
      + cpu_share_count                        = (known after apply)
      + cpu_share_level                        = "normal"
      + datastore_id                           = "datastore-2939"
      + default_ip_address                     = (known after apply)
      + ept_rvi_mode                           = "automatic"
```

```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
```

```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

vsphere_virtual_machine.vm: Creating...
vsphere_virtual_machine.vm: Still creating... [10s elapsed]
vsphere_virtual_machine.vm: Still creating... [20s elapsed]
vsphere_virtual_machine.vm: Still creating... [30s elapsed]
vsphere_virtual_machine.vm: Still creating... [40s elapsed]
vsphere_virtual_machine.vm: Still creating... [50s elapsed]
vsphere_virtual_machine.vm: Still creating... [1m0s elapsed]
vsphere_virtual_machine.vm: Still creating... [1m10s elapsed]
vsphere_virtual_machine.vm: Creation complete after 1m11s [id=423b9db5-2c29-cc8f-406c-f8f2dc2e033d]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Kürzlich bearbeitete Aufgaben Alarme

Name der Aufgabe	Ziel	Status	Details	Initiator	In Warteschlange	Startzeit
Virtuelle Maschine einschalten	stanki-linuxdemo	Abgeschlossen		SVA\cstankow	2 ms	11.10.2019, 13:26:37
Gastbetriebssystem der virtuellen Maschine anpassen	stanki-linuxdemo	Abgeschlossen		SVA\cstankow	3 ms	11.10.2019, 13:26:36
Virtuelle Maschine neu konfigurieren	stanki-linuxdemo	Abgeschlossen		SVA\cstankow	3 ms	11.10.2019, 13:26:36
Virtuelle Maschine klonen	template_centos7	Abgeschlossen		SVA\cstankow	5 ms	11.10.2019, 13:26:01

Compliance as Code

/ Security-Auditing

- Eine regelmäßige Auditierung der Systemlandschaft ist unabdingbar
  - Ständige Software-Updates bringen neue Fehler
  - Veränderungen bei Verschlüsselungs-Best Practices erfordern schnelles Handeln (*Heartbleed*)
- Ein System muss bei der Bereitstellung möglichst sicher sein, nachträgliches Absichern erfolgt i.d.R. nicht
- Das Absichern von Systemen muss weitestgehend automatisiert werden, um eine **praktikable** und **effiziente** Lösung darzustellen
- Verschiedene Open-Source-Tools können hier helfen!

/ Security-Auditing

- OpenSCAP ist eine freie Implementation von **SCAP** (*Security Content Automation Protocol*)
- Protokoll zur automatisierten Vulnerability- und Compliance-Messung und Auswertung
- ➔ Ziel: System-Verwundbarkeit darstellen
- Katalog-Quellen:
  - OpenSCAP-Projekt: https://www.open-scap.org/security-policies/choosing-policy/
  - GitHub-Projekt „*ComplianceAsCode*“: https://github.com/ComplianceAsCode/content
  - National Vulnerability Database des **NIST***: https://nvd.nist.gov/ncp/repository
- Gängige System-Management-Tools wie Foreman und Spacewalk/Uyuni unterstützen das Abfragen von Sicherheitskatalogen gemäß **XCCDF** (*Entensible Configuration Checklist Description Format*)

**National Institute of Standards and Technology*

https://localhost:8443/compliance

Nicht sicher | localhost:8443/compliance/arf_reports/86/show_html

FOREMAN SVA GmbH Berlin Admin User

Monitor
Content
Containers
Hosts
Configure
Infrastructure
Administer

Compliance Reports » client.sva.de

Compliance and Scoring

The target system did not satisfy the conditions of 30 rules! Please review rule results and consider applying remediation.

Rule results

20 passed | 30 failed | 1

Severity of failed rules

1 | 28 medium | 1

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	81.666664	100.000000	81.67%

Rule Overview

client.sva.de

Nicht sicher | localhost:8443/compliance/arf_reports/86

FOREMAN SVA GmbH Berlin Admin User

Monitor
Content
Containers
Hosts
Configure
Infrastructure
Administer

Compliance Reports » client.sva.de

Show log messages:

All messages

Back | Delete | Host details | View full report | Download XML in bzip | Download HTML

Reported at Sep 28, 07:32 PM for policy Demo policy through katello.sva.de

Severity	Message	Resource	Result	Actions
Medium	Disable At Service (atd)	xccdf_org.ssgproject.content_rule_service_atd_disabled	pass	Hosts failing this rule
Medium	Disable Odd Job Daemon (oddjobd)	xccdf_org.ssgproject.content_rule_service_oddjobd_disabled	pass	Hosts failing this rule
Medium	Disable Apache Qpid (qpidd)	xccdf_org.ssgproject.content_rule_service_qpidd_disabled	pass	Hosts failing this rule
Medium	Disable Automatic Bug Reporting Tool (abrtd)	xccdf_org.ssgproject.content_rule_service_abrtd_disabled	pass	Hosts failing this rule
Medium	Disable ntpdate Service (ntpdate)	xccdf_org.ssgproject.content_rule_service_ntpdate_disabled	pass	Hosts failing this rule
Medium	Disable Network Router Discovery Daemon (rdisc)	xccdf_org.ssgproject.content_rule_service_rdisc_disabled	pass	Hosts failing this rule
Low	Ensure /var/log/audit Located On Separate Partition	xccdf_org.ssgproject.content_rule_partition_for_var_log_audit	fail	Hosts failing this rule

	10.2.7

Description	To capture kernel module loading and unloading events, use following lines, setting ARCH to either b32 for 32-bit system, or having two lines for both b32 and b64 in case your system is 64-bit:
	`-w /usr/sbin/insmod -p x -k modules`
	`-w /usr/sbin/rmmod -p x -k modules`
	`-w /usr/sbin/modprobe -p x -k modules`
	
	`-a always,exit -F arch=ARCH -S init_module,finit_module,create_module,delete_module -F key=modules`
	
	The place to add the lines depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the lines to file `/etc/audit/audit.rules`.

Rationale	The addition/removal of kernel modules can be used to alter the behavior of the kernel and potentially introduce malicious code into kernel space. It is important to have an audit trail of modules that have been introduced into the kernel.
Warnings	warning This rule checks for multiple syscalls related to kernel module loading and unloading; it was written with DISA STIG in mind. Other policies should use a separate rule for each syscall that needs to be checked. For example: • `audit_rules_kernel_module_loading_insmod` • `audit_rules_kernel_module_loading_rmmod` • `audit_rules_kernel_module_loading_modprobe`

/ Was ist InSpec?

- Framework für Audits und automatisierte Tests
  - **Compliance**
  - **Security**
  - Unit-Tests
  - …
- **Selbstsprechende Definition** von Anforderungen
- Einfach zu verstehen (*im Vergleich zu XML*)
- Open-Source: https://github.com/inspec/inspec
  - Teil von Chef Automate

Beispiel:

```
describe package('telnetd') do
  it { should_not be_installed }
end

describe sshd_config do
  its('PermitRootLogin') {
    should_not cmp 'yes'
  }
end
```

/ Dev-Sec

- Dev-Sec ist ein Framework zur automatischen Absicherung von Servern und Applikationen:
  - Microsoft Windows / Linux
  - SSH / SSL
  - Docker / Kubernetes
  - Apache / NGINX
  - MySQL / PostgreSQL
- Integriert sich in Puppet, Chef und Ansible
  - Neue und bestehende Systeme können so automatisiert angepasst werden
- Grundlagen für Dev-Sec-Kataloge:
  - CIS-Benchmarks (*Center for Internet Security*)
  - Hersteller Best-Practices und Security Hardening Guides

- Überprüfung der Systeme via InSpec
- Derzeit unterstützt:
  - Ubuntu 14.04, 16.04, 18.04
  - Debian 8, 9
  - Enterprise Linux 6, 7, 8
  - OpenSUSE 42.x / SLES 12.x

- Open-Source: https://dev-sec.io

```
C:\Users\cstankow.SVA\Documents\GitHub\osad2019-iac\dev-sec (master -> origin)
λ inspec exec linux-baseline -t ssh://vagrant:vagrant@localhost:2222 --sudo

     expected: 0
          got: 16

     (compared using ==)

  [PASS]  sysctl-31a: Secure Core Dumps - dump settings
     [PASS]  Kernel Parameter fs.suid_dumpable value should cmp == /(0|2)/
  [SKIP]  sysctl-31b: Secure Core Dumps - dump path
     [SKIP]  Skipped control due to only_if condition.
  [PASS]  sysctl-32: kernel.randomize_va_space
     [PASS]  Kernel Parameter kernel.randomize_va_space value should eq 2
  [PASS]  sysctl-33: CPU No execution Flag or Kernel ExecShield
     [PASS]  /proc/cpuinfo Flags should include NX


Profile Summary: 27 successful controls, 26 control failures, 1 control skipped
Test Summary: 81 successful, 44 failures, 1 skipped
```

```
$ ansible-playbook -i inventory test.yml

PLAY [wrapper playbook for kitchen testing "ansible-os-hardening" with custom vars for testing] ********************************

TASK [Gathering Facts] ***********************************************************************************************************
ok: [localhost]

TASK [ansible-os-hardening : Set OS family dependent variables] ******************************************************************
ok: [localhost]

TASK [ansible-os-hardening : Set OS dependent variables] *************************************************************************
[DEPRECATION WARNING]: Use errors="ignore" instead of skip. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.

TASK [ansible-os-hardening : install auditd package | package-08] ****************************************************************
ok: [localhost]

TASK [ansible-os-hardening : configure auditd | package-08] ******************************************************************
changed: [localhost]

TASK [ansible-os-hardening : create limits.d-directory if it does not exist | sysctl-31a, sysctl-31b] ***************************
ok: [localhost]

TASK [ansible-os-hardening : create additional limits config file -> 10.hardcore.conf | sysctl-31a, sysctl-31b] ****************
changed: [localhost]

TASK [ansible-os-hardening : set 10.hardcore.conf perms to 0400 and root ownership] ******************************************
changed: [localhost]

TASK [ansible-os-hardening : remove 10.hardcore.conf config file] ************************************************************

PLAY RECAP ***********************************************************************************************************************
localhost                  : ok=37   changed=16   unreachable=0    failed=0    skipped=27   rescued=0    ignored=0
```

```
     [PASS]  Kernel Parameter net.ipv6.conf.default.accept_ra_rtr_pref value should eq 0
  [PASS]  sysctl-23: Disable learning Prefix Information from router advertisement
     [PASS]  Kernel Parameter net.ipv6.conf.default.accept_ra_pinfo value should eq 0
  [PASS]  sysctl-24: Disable learning Hop limit from router advertisement
     [PASS]  Kernel Parameter net.ipv6.conf.default.accept_ra_defrtr value should eq 0
  [PASS]  sysctl-25: Disable the system`s acceptance of router advertisement
     [PASS]  Kernel Parameter net.ipv6.conf.all.accept_ra value should eq 0
     [PASS]  Kernel Parameter net.ipv6.conf.default.accept_ra value should eq 0
  [PASS]  sysctl-26: Disable IPv6 autoconfiguration
     [PASS]  Kernel Parameter net.ipv6.conf.default.autoconf value should eq 0
  [PASS]  sysctl-27: Disable neighbor solicitations to send out per address
     [PASS]  Kernel Parameter net.ipv6.conf.default.dad_transmits value should eq 0
  [PASS]  sysctl-28: Assign one global unicast IPv6 addresses to each interface
     [PASS]  Kernel Parameter net.ipv6.conf.default.max_addresses value should eq 1
  [PASS]  sysctl-29: Disable loading kernel modules
     [PASS]  Kernel Parameter kernel.modules_disabled value should eq 0
  [PASS]  sysctl-30: Magic SysRq
     [PASS]  Kernel Parameter kernel.sysrq value should eq 0
  [PASS]  sysctl-31a: Secure Core Dumps - dump settings
     [PASS]  Kernel Parameter fs.suid_dumpable value should cmp == /(0|2)/
  [SKIP]  sysctl-31b: Secure Core Dumps - dump path
     [SKIP]  Skipped control due to only_if condition.
  [PASS]  sysctl-32: kernel.randomize_va_space
     [PASS]  Kernel Parameter kernel.randomize_va_space value should eq 2
  [PASS]  sysctl-33: CPU No execution Flag or Kernel ExecShield
     [PASS]  /proc/cpuinfo Flags should include NX


Profile Summary: 53 successful controls, 0 control failures, 1 control skipped
Test Summary: 125 successful, 0 failures, 1 skipped

C:\Users\cstankow.SVA\Documents\GitHub\osad2019-iac\dev-sec (master -> origin)
λ
```

Any questions?

/ Danke für die Aufmerksamkeit!

CHRISTIAN STANKOWIC

Senior System Engineer

Geschäftsstelle Mitte

https://cstan.io

Mobil: +49 151 18025982

Tel.: +49 6122 536 0

Fax: +49 6122 536 399

christian.stankowic@sva.de

Borsigstraße 14
65205 Wiesbaden