JUG Hamburg, 11.06.2025

Ansible für Entwickler

Konfigurationsmanagement nicht nur für Ops

Sandra Parsick
mail@sandra-parsick.de
@sparsick@mastodon.social

Wer bin ich?

- Sandra Parsick
- Freiberuflicher Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
  - Java Enterprise Anwendungen
  - Agile Methoden
  - Software Craftmanship
  - Automatisierung von Entwicklungsprozessen
- Trainings
- Workshops

mail@sandra-parsick.de

@sparsick@mastodon.social

https://www.sandra-parsick.de

https://ready-for-review.dev

Agenda

1. Ansible – Was ist das?
2. Warum ist es für Entwickler interessant?
3. Einführung in Ansible
4. Wie unterscheidet sich Ansible zur seiner Konkurrenz?
5. Weitere Einsatzszenarien aus Entwicklersicht

Ansible
Was ist das?

Ansible

- Software für
  - Konfigurationsmanagement,
  - Softwareverteilung und
  - Ad-hoc-Kommando-Ausführung

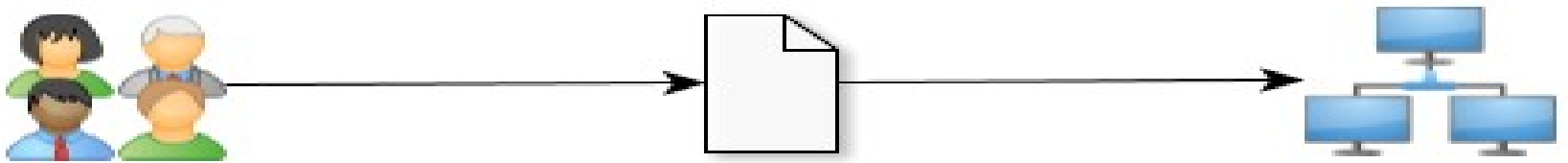
Konfigurationsmanagement (KM)

„Das KM umfasst alle technischen, organisatorischen und beschlussfassenden Maßnahmen und Strukturen, die sich mit der Konfiguration (Spezifikation) eines Produkts befassen.“
https://www.projektmagazin.de/glossarterm/konfigurationsmanagement

Konfigurationsmanagement (KM)

- Softwarekonfiguration
- Hardwarekonfiguration
- Dienstleistungskonfiguration
- Systemkonfiguration

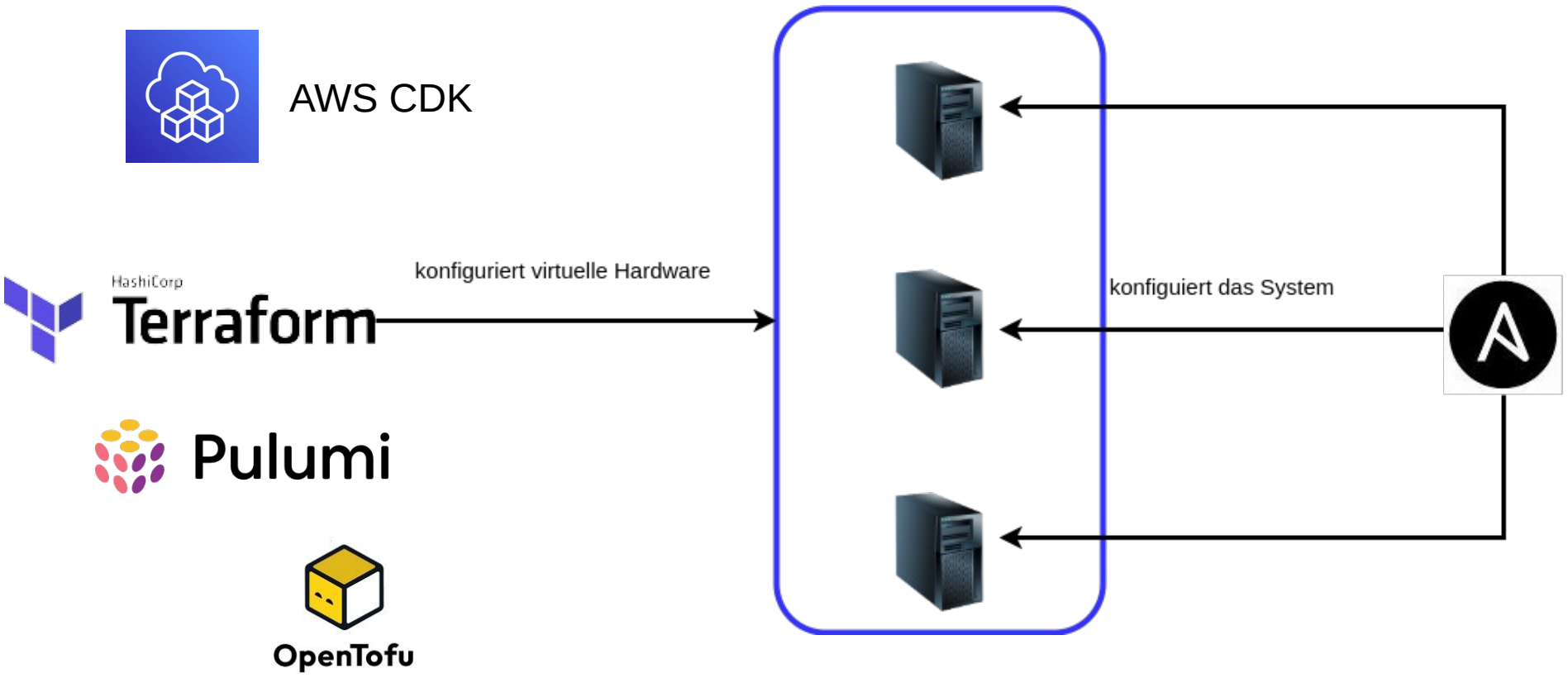
Systemkonfiguration - „Infrastructure As Code“



Systemkonfiguration - „Infrastructure As Code“

Ansible

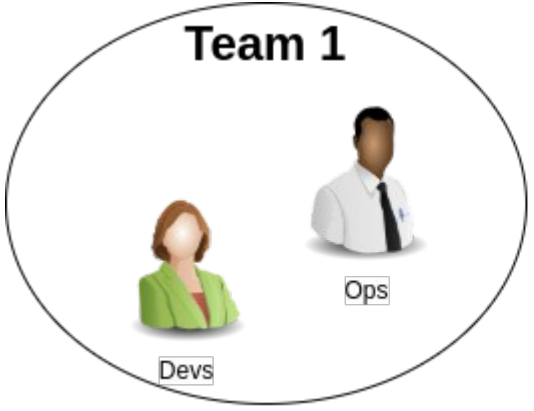
Warum fehlen andere IaC Tools?

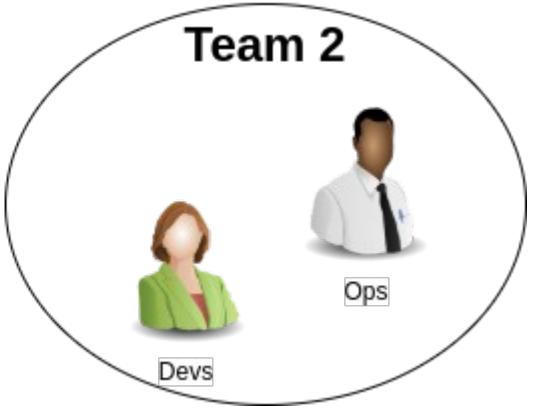


Warum ist es für Entwickler interessant?

Systemkonfiguration für Entwickler

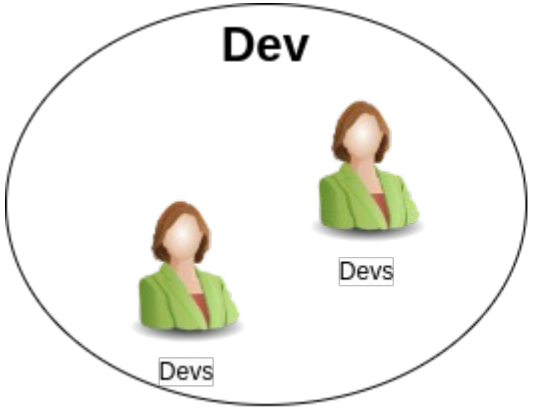
Organisatorische Ausgangslage
Wunsch

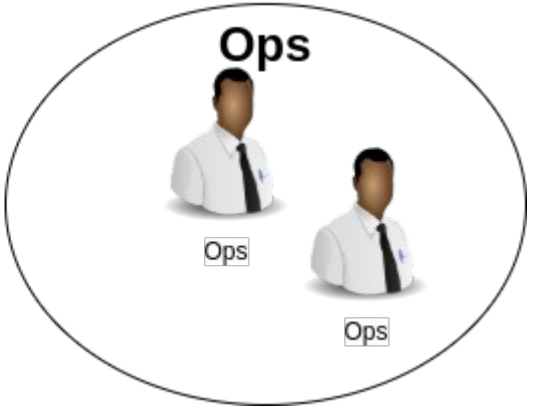




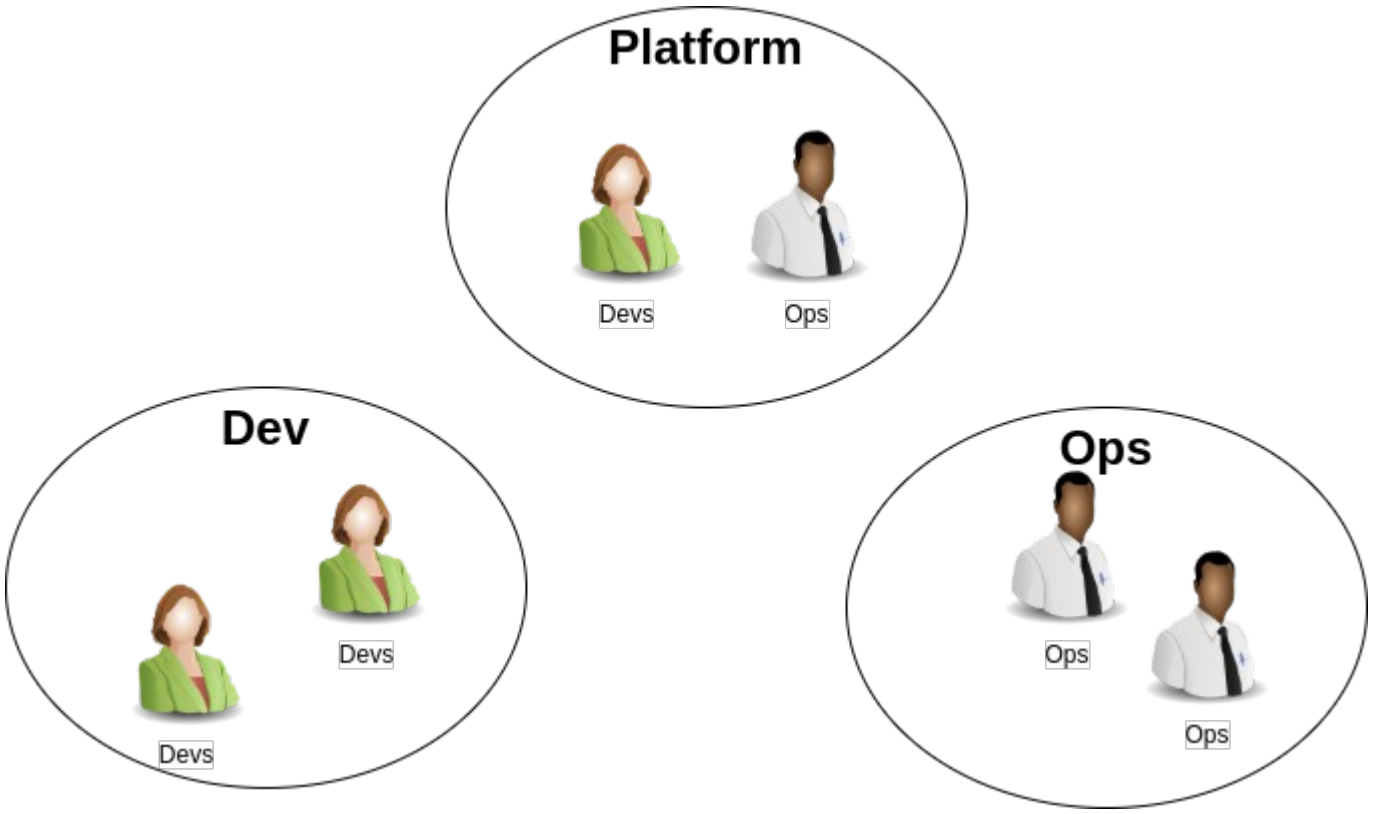
Systemkonfiguration für Entwickler

Organisatorische Ausgangslage
Realität



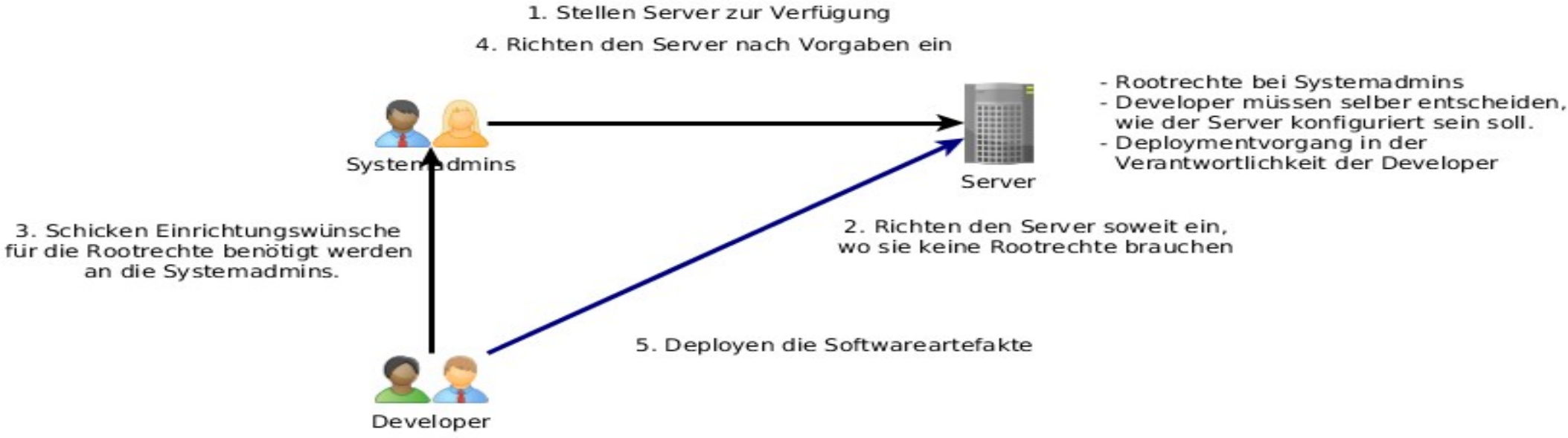


Systemkonfiguration für Entwickler



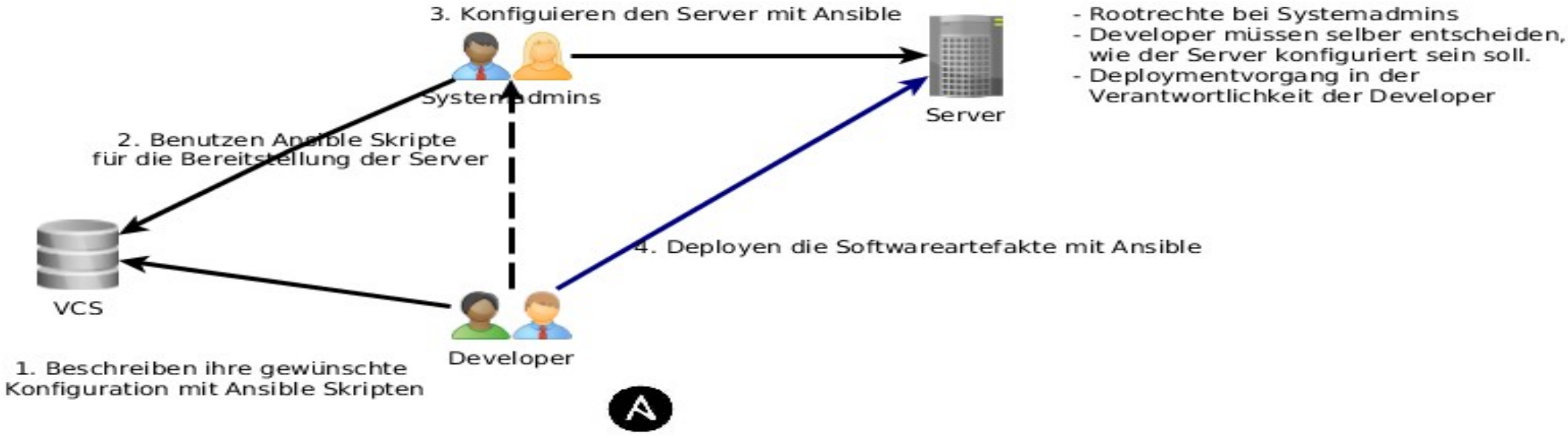
Systemkonfiguration für Entwickler

Prozess zwischen Development und Operation



Systemkonfiguration für Entwickler

Lösungidee mit Ansible

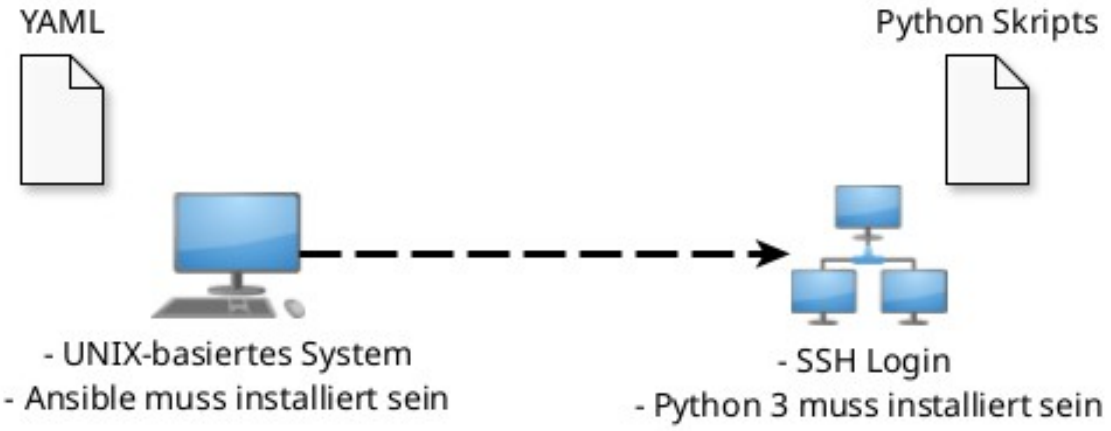


Einführung in Ansible

Ansible

- Software für
  - Konfigurationsmanagement,
  - Softwareverteilung und
  - Ad-hoc-Kommando-Ausführung
- Sprache: Python
- Ansible Skripte: YAML

Funktionsweise

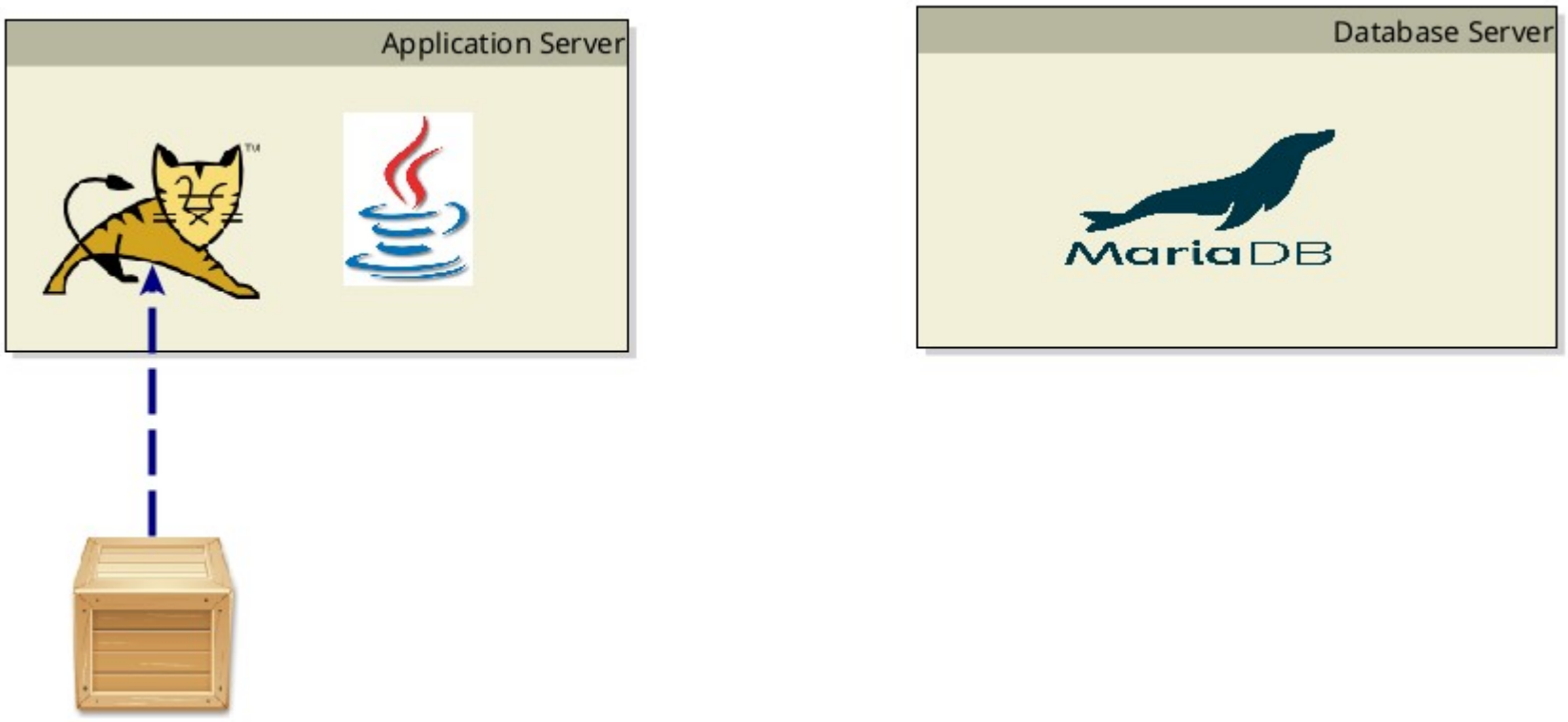


Exkurs: YAML vs JSON

```yaml
InsuranceCompanies:
  Time: Feb 2019
  Top Insurance Companies:
    - 'No': '1'
      Name: Berkshire Hathaway ( BRK.A)
      Market Capitalization: $308 billion
    - 'No': '2'
      Name: China Life Insurance (LFC)
      Market Capitalization: $80  billion
  source: investopedia.com
  url: >-
    https://www.investopedia.com/articles/active-
trading/111314/top-10-insurance-companies-
metrics.asp
```

```json
{
  "InsuranceCompanies": {
    "Time": "Feb 2019",
    "Top Insurance Companies": [
      {
        "No": "1",
        "Name": "Berkshire Hathaway ( BRK.A)",
        "Market Capitalization": "$308 billion"
      },
      {
        "No": "2",
        "Name": "China Life Insurance (LFC)",
        "Market Capitalization": "$80  billion"
      }
    ],
    "source": "investopedia.com",
    "url":
"https://www.investopedia.com/articles/active-
trading/111314/top-10-insurance-companies-
metrics.asp"
  }
}
```

Ansible Beispiel



Setup Application Server Playbook

Setup Application Server

```yaml
- hosts: application_server
  vars:
    tomcat_version: 11.0.8
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      ansible.builtin.apt:
        name: openjdk-21-jdk
        state: present
        update_cache: yes
      become: yes
      become_method: sudo
```

Inventories

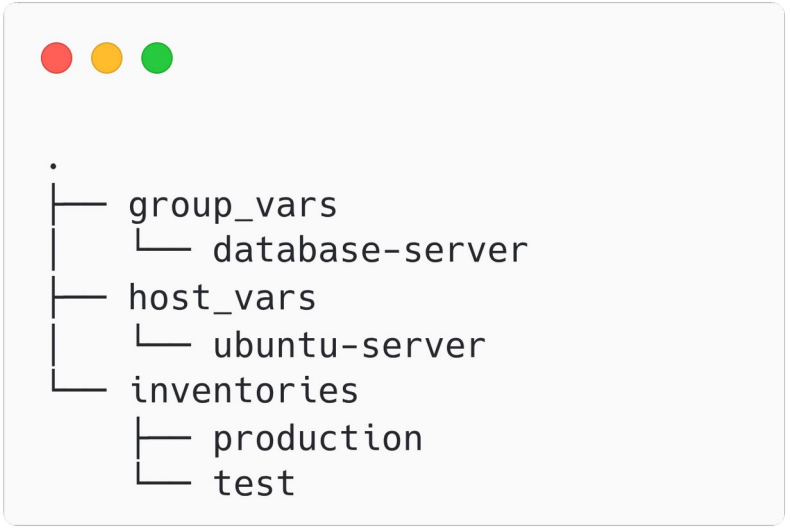
Production

```
[application_server]
192.168.56.10
ubuntu_server db_host=mariadb01

[maria_db_server]
mariadb[01:10]

[oracle_db_server]
db_[a:f].oracle.company.com

[database_server:children]
mariadb_db_server
oracle_db_server

[application_server:vars]
message="Welcome"

[database_server:vars]
message="Hello World!"
```

Test

```
[application_server]
192.168.56.10

[database_server]
192.168.56.10

[all:vars]
ansible_user=vagrant
```

Inventories



```
.
├── group_vars
│   └── database-server
├── host_vars
│   └── ubuntu-server
└── inventories
    ├── production
    └── test
```

```
➜ cat group_vars/database-server
proxy_host: proxy.server
```

Setup Application Server

```yaml
- hosts: application_server
  vars:
    tomcat_version: 11.0.8
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      ansible.builtin.apt:
        name: openjdk-21-jdk
        state: present
        update_cache: yes
      become: yes
      become_method: sudo
```

Module Beispiel

1328 lines (1133 sloc) | 50.5 KB

Raw | Blame

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# Copyright: (c) 2012, Flowroute LLC
# Written by Matthew Williams <matthew@flowroute.com>
# Based on yum module written by Seth Vidal <skvidal at fedoraproject.org>

# GNU General Public License v3.0+ (see COPYING or https://www.gnu.org/licenses/gpl-3.0.txt)

from __future__ import absolute_import, division, print_function
__metaclass__ = type


DOCUMENTATION = '''
---
module: apt
short_description: Manages apt-packages
description:
  - Manages I(apt) packages (such as for Debian/Ubuntu).
version_added: "0.0.2"
options:
  name:
    description:
      - A list of package names, like C(foo), or package specifier with version, like C(foo=1.0).
        Name wildcards (fnmatch) like C(apt*) and version wildcards like C(foo=1.0*) are also supported.
    aliases: [ package, pkg ]
    type: list
```

```
289 '''
290
291 RETURN = '''
292 cache_updated:
293     description: if the cache was updated or not
294     returned: success, in some cases
295     type: bool
296     sample: True
297 cache_update_time:
298     description: time of the last cache update (0 if unknown)
299     returned: success, in some cases
300     type: int
301     sample: 1425828348000
302 stdout:
303     description: output from apt
304     returned: success, when needed
305     type: str
306     sample: "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe following extra packages will be installed:\n  apache2-bin
307 stderr:
308     description: error output from apt
309     returned: success, when needed
310     type: str
311     sample: "AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally t
312 '''  # NOQA
313
314 # added to stave off future warnings about apt api
315 import warnings
316 warnings.filterwarnings('ignore', "apt API not stable yet", FutureWarning)
317
318 import datetime
319 import fnmatch
320 import itertools
321 import os
322 import random
323 import re
324 import shutil
325 import sys
326 import tempfile
327 import time
```

```
            self.backup_dir = None

    def __enter__(self):
        """
        This method will be called when we enter the context, before we call `apt-get …`
        """

        # if policy_rc_d is null then we don't need to modify policy-rc.d
        if self.m.params['policy_rc_d'] is None:
            return

        # if the /usr/sbin/policy-rc.d already exists we back it up
        if self.backup_dir:
            try:
                shutil.move('/usr/sbin/policy-rc.d', self.backup_dir)
            except Exception:
                self.m.fail_json(msg="Fail to move /usr/sbin/policy-rc.d to %s" % self.backup_dir)

        # we write /usr/sbin/policy-rc.d so it always exits with code policy_rc_d
        try:
            with open('/usr/sbin/policy-rc.d', 'w') as policy_rc_d:
                policy_rc_d.write('#!/bin/sh\nexit %d\n' % self.m.params['policy_rc_d'])

            os.chmod('/usr/sbin/policy-rc.d', 0o0755)
        except Exception:
            self.m.fail_json(msg="Failed to create or chmod /usr/sbin/policy-rc.d")

    def __exit__(self, type, value, traceback):
        """
        This method will be called when we enter the context, before we call `apt-get …`
        """

        # if policy_rc_d is null then we don't need to modify policy-rc.d
        if self.m.params['policy_rc_d'] is None:
            return

        if self.backup_dir:
            # if /usr/sbin/policy-rc.d already exists before the call to __enter__
            # we restore it (from the backup done in __enter__)
            try:
                shutil.move(os.path.join(self.backup_dir, 'policy-rc.d'),
                            '/usr/sbin/policy-rc.d')
```

Ansible Modules

Module Index

- All Modules
- Cloud Modules
- Clustering Modules
- Commands Modules
- Crypto Modules
- Database Modules
- Files Modules
- Identity Modules
- Inventory Modules
- Messaging Modules
- Monitoring Modules
- Net Tools Modules
- Network Modules
- Notification Modules
- Packaging Modules
- Remote Management Modules
- Source Control Modules
- Storage Modules
- System Modules
- Utilities Modules
- Web Infrastructure Modules
- Windows Modules

```yaml
- name: Download current Tomcat 11 version
  ansible.builtin.get_url:
    url: "http://archive.apache.org/dist/tomcat/tomcat-11/v{{ tomcat_version }}/bin/{{ tomcat_base_name }}.tar.gz"
    dest: "/tmp"
  delegate_to: localhost

- name: create opt dir
  ansible.builtin.file:
    name: /opt
    mode: "u+rwx,g+rx,a+r"
    owner: vagrant
    group: vagrant
  become: yes

- name: Install Tomcat 11
  ansible.builtin.unarchive:
    src: "/tmp/{{ tomcat_base_name }}.tar.gz"
    dest: "/opt"
    creates: "/opt/{{ tomcat_base_name }}"
    owner: vagrant
    group: vagrant

- name: Set link to tomcat 11
  ansible.builtin.file:
    src: "/opt/{{ tomcat_base_name }}"
    dest: "/opt/tomcat"
    state: link
    force: true
```

```yaml
    - name: setup setenv.sh
      ansible.builtin.template:
        dest: "/opt/{{ tomcat_base_name }}/bin/setenv.sh"
        src: "roles/tomcat11/templates/setenv.sh.j2"
        mode: 755
      when: catalina_opts is defined

    - ansible.builtin.find:
        paths: "/opt/{{ tomcat_base_name }}/bin"
        patterns: "*.sh"
      register: result

    - name: ensure tomcat scripts are executable
      ansible.builtin.file:
        name: "{{ item.path }}"
        mode: "u+rwx,g+rx,a+r"
      loop: '{{ result.files }}'

    - name: install tomcat as service
      ansible.builtin.copy:
        src: roles/tomcat11/files/tomcat.service
        dest: /etc/systemd/system/
      become: yes
```

Templates

- setenv.sh.j2

```
CATALINA_OPTS="{{ catalina_opts }}"
```

Templates - Jinja2

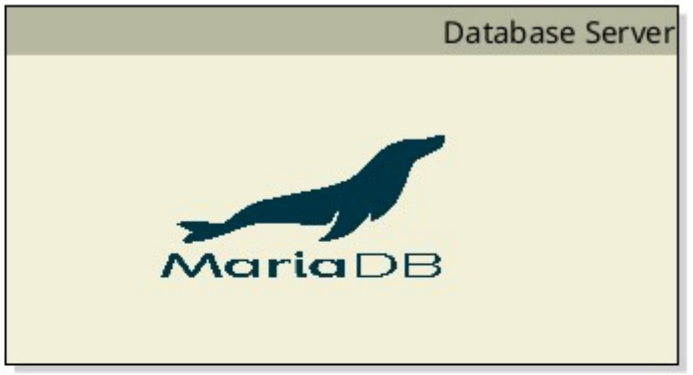
- Templating engine für Python
- Mehr Information unter jinja.palletsprojects.com

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>My Webpage</title>
</head>
<body>
    <ul id="navigation">
    {% for item in navigation %}
        <li><a href="{{ item.href }}">{{ item.caption }}</a></li>
    {% endfor %}
    </ul>

    <h1>My Webpage</h1>
    {{ a_variable }}

    {# a comment #}
</body>
</html>
```

Setup Application Server Playbook

Demo

Setup Database Server Playbook



```yaml
- hosts: database_server
  become: yes
  vars:
    mariadb_root_password: password
  tasks:
    - name: install needed python package
      ansible.builtin.apt:
        name: python3-pymysql
        state: present
        update_cache: yes

    - name: install mariadb-server
      ansible.builtin.apt:
        name: mariadb-server
        state: present

    - name: start mariadb
      ansible.builtin.service:
        name: mariadb
        state: started
```

```yaml
    - name: set bind address
      ansible.builtin.lineinfile:
        dest: /etc/mysql/mariadb.conf.d/50-server.cnf
        line: 'bind-address = 0.0.0.0'
        state: present
        regexp: "^bind-address(.*)"
      notify: restart mariadb

    - name: creates db user dba
      ansible.builtin.mysql_user:
        name: dba
        password: "g3h31m"
        login_user: root
        login_password: password
        priv: "*.*:ALL,GRANT"
        state: present
        host: "%"
        login_unix_socket: /var/run/mysqld/mysqld.sock

  handlers:
     - name: restart mariadb
       ansible.builtin.service:
         name: mariadb
         state: restarted
```

Setup Database Server Playbook

Demo

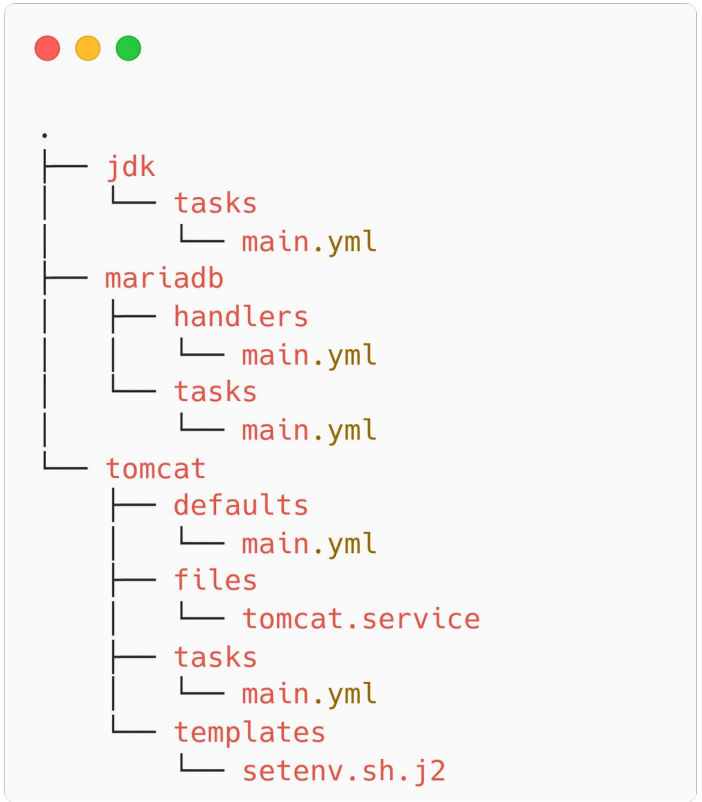
Setup Application Server

```yaml
- hosts: application_server
  vars:
    tomcat_version: 11.0.8
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      ansible.builtin.apt:
        name: openjdk-21-jdk
        state: present
        update_cache: yes
      become: yes
      become_method: sudo
```

```yaml
- name: Download current Tomcat 11 version
  ansible.builtin.get_url:
    url: "http://archive.apache.org/dist/tomcat/tomcat-11/v{{ tomcat_version }}/bin/{{ tomcat_base_name }}.tar.gz"
    dest: "/tmp"
  delegate_to: localhost

- name: create opt dir
  ansible.builtin.file:
    name: /opt
    mode: "u+rwx,g+rx,a+r"
    owner: vagrant
    group: vagrant
  become: yes

- name: Install Tomcat 11
  ansible.builtin.unarchive:
    src: "/tmp/{{ tomcat_base_name }}.tar.gz"
    dest: "/opt"
    creates: "/opt/{{ tomcat_base_name }}"
    owner: vagrant
    group: vagrant

- name: Set link to tomcat 11
  ansible.builtin.file:
    src: "/opt/{{ tomcat_base_name }}"
    dest: "/opt/tomcat"
    state: link
    force: true
```

Roles

```
roles/
    common/
        tasks/
        handlers/
        files/
        templates/
        vars/
        defaults/
        meta/
```



Setup Playbooks mit Roles

- Setup Application Server

```yaml
- hosts: application_server
  roles:
    - jdk
    - { role: tomcat, tomcat_version: 11.0.8 }
```

- Setup Database Server

```yaml
- hosts: database_server
  become: true
  roles:
    - mariadb
```

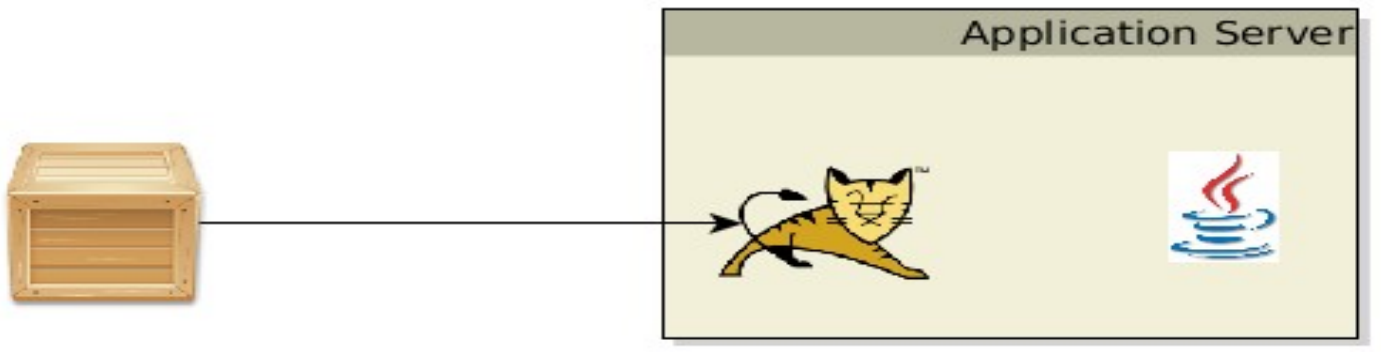
include_role, import_role (seit v2.3)

```yaml
---
- hosts: webservers
  tasks:
    - name: Print a message
      ansible.builtin.debug:
        msg: "this task runs before the example role"

    - name: Include the example role
      include_role:
        name: example

    - name: Import the example role
      import_role:
        name: example

    - name: Print a message
      ansible.builtin.debug:
        msg: "this task runs after the example role"
```

Import Roles vs. Include Roles

Feature	import_role	include_role
Loading Time	Playbook parsing (static)	Playbook execution (dynamic)
Conditional/Loops	Not supported on statement	Supported on statement
Task Order	Fixed at parse time	Dynamic at runtime
Use Case	Predictable, always needed	Conditional, dynamic needed

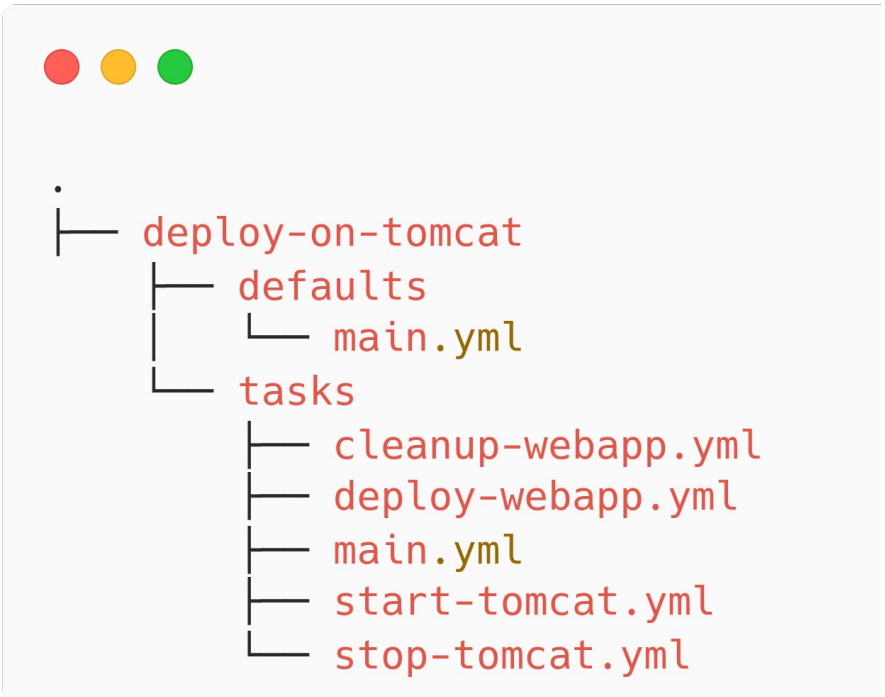
Java Webapplikation Deployment



Deploy Application Playbook

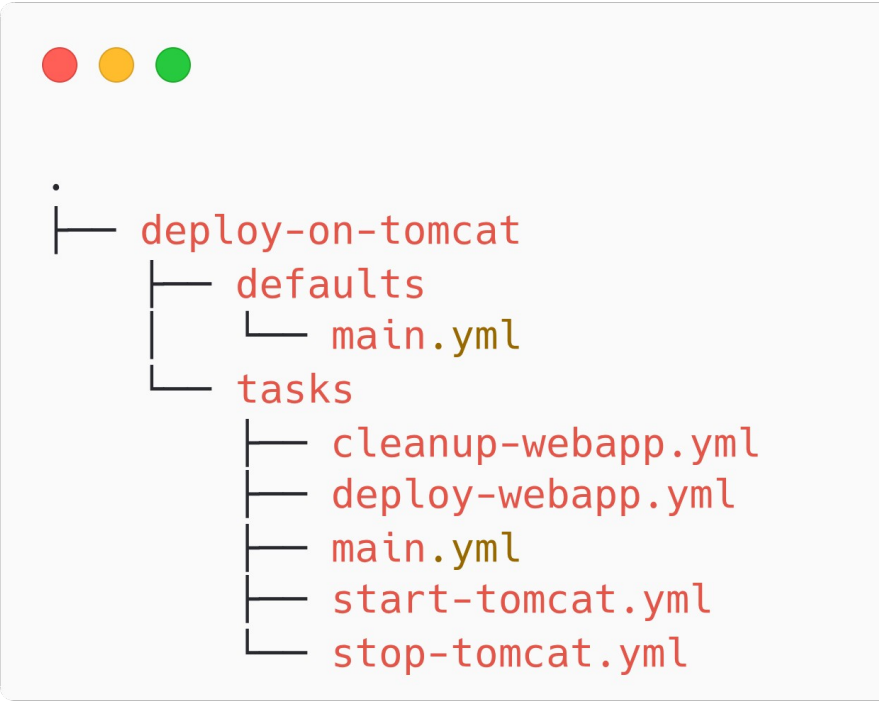
```yaml
- hosts: application_server
  roles:
    - role: deploy-on-tomcat
      vars:
        webapp_source_path: ./demo-app-ansible-deploy-1.0-SNAPSHOT.war
        webapp_target_name: demo
```

deploy-on-tomcat Role




```
# cat tasks/main.yml
- import_tasks: stop-tomcat.yml
- import_tasks: cleanup-webapp.yml
- import_tasks: deploy-webapp.yml
- import_tasks: start-tomcat.yml
```

deploy-on-tomcat Role



```
.
├── deploy-on-tomcat
    ├── defaults
    │   └── main.yml
    └── tasks
        ├── cleanup-webapp.yml
        ├── deploy-webapp.yml
        ├── main.yml
        ├── start-tomcat.yml
        └── stop-tomcat.yml
```

```
# cat tasks/stop-tomcat.yml
- name: stop tomcat
  ansible.builtin.service:
    name: tomcat
    state: stopped
  become: true

- name: wait tomcat shutdown
  ansible.builtin.wait_for:
    port: 8080
    state: stopped
    timeout: 60
```

deploy-on-tomcat Role

```
#cat tasks/cleanup-webapp.yml
- name: cleanup {{ webapp_target_name }}
  ansible.builtin.file:
    name: "{{tomcat_app_base}}/{{ webapp_target_name }}"
    state: absent
```

deploy-on-tomcat Role

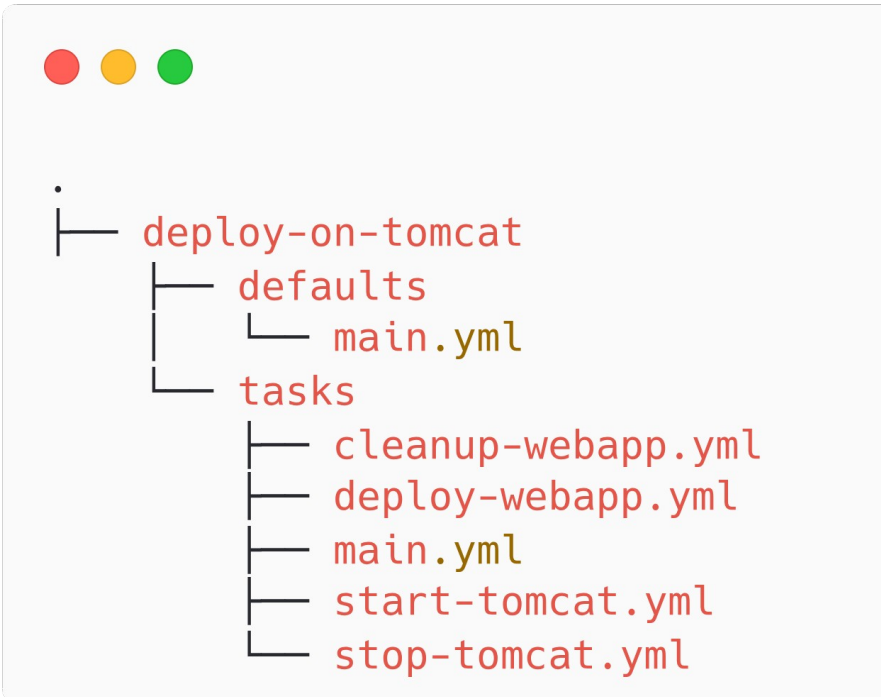
```
# cat tasks/deploy-webapp.yml
- name: delete previous backup
  ansible.builtin.file:
    path: "{{ tomcat_app_base }}/{{ webapp_target_name }}.war.previous"
    state: absent

- name: create new backup
  ansible.builtin.command: mv {{ tomcat_app_base }}/{{ webapp_target_name }}.war
{{ tomcat_app_base }}/{{ webapp_target_name }}.war.previous
  ignore_errors: true

- name: copy webapp {{ webapp_source_path }} to {{ webapp_target_name }}
  ansible.builtin.copy:
    src: "{{ webapp_source_path }}"
    dest: "{{ tomcat_app_base }}/{{ webapp_target_name }}.war"
    mode: "660"
```

deploy-on-tomcat Role

```
.
├── deploy-on-tomcat
    ├── defaults
    │   └── main.yml
    └── tasks
        ├── cleanup-webapp.yml
        ├── deploy-webapp.yml
        ├── main.yml
        ├── start-tomcat.yml
        └── stop-tomcat.yml
```

```
# cat tasks/start-tomcat.yml
- name: start tomcat
  ansible.builtin.service:
    name: tomcat
    enabled: yes
    state: started
  become: true

- name: wait for tomcat to start
  ansible.builtin.wait_for:
    port: 8080
    timeout: 60
```

deploy-on-tomcat Role



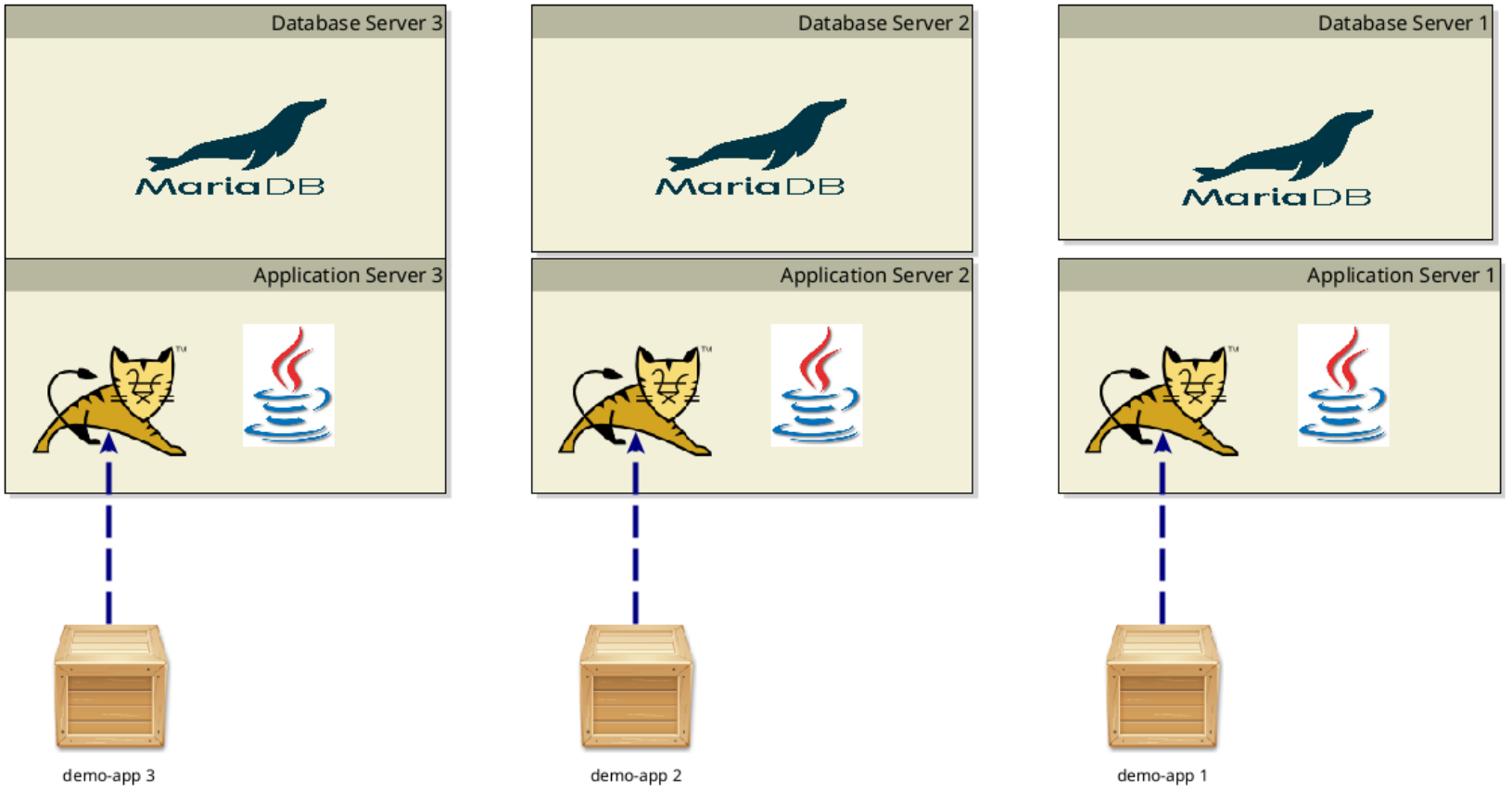
```
.
├── deploy-on-tomcat
    ├── defaults
    │   └── main.yml
    └── tasks
        ├── cleanup-webapp.yml
        ├── deploy-webapp.yml
        ├── main.yml
        ├── start-tomcat.yml
        └── stop-tomcat.yml
```

```
# cat defaults/main.yml
tomcat_app_base: /opt/tomcat/webapps
```

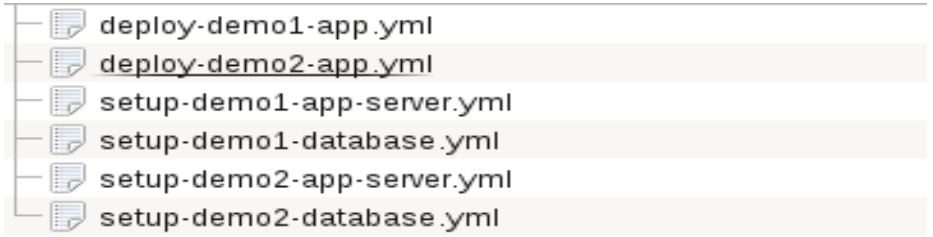
Deploy Application Playbook

Demo

Warum Roles?


Database Server 3
MariaDB
Application Server 3
Database Server 2
MariaDB
Application Server 2
Database Server 1
MariaDB
Application Server 1
demo-app 3
demo-app 2
demo-app 1

Warum Roles?

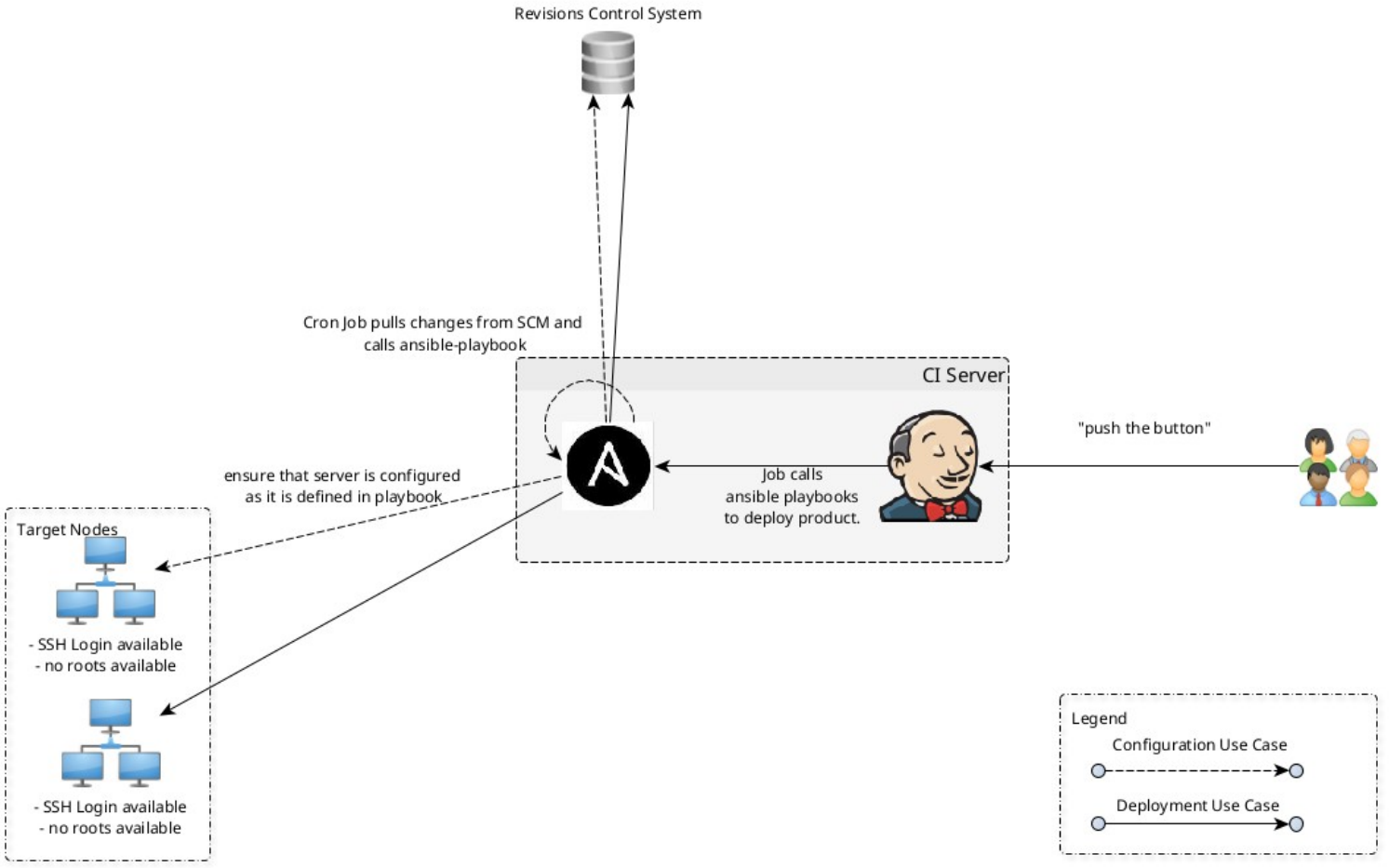

deploy-demo1-app.yml
deploy-demo2-app.yml
setup-demo1-app-server.yml
setup-demo1-database.yml
setup-demo2-app-server.yml
setup-demo2-database.yml

Warum Roles?

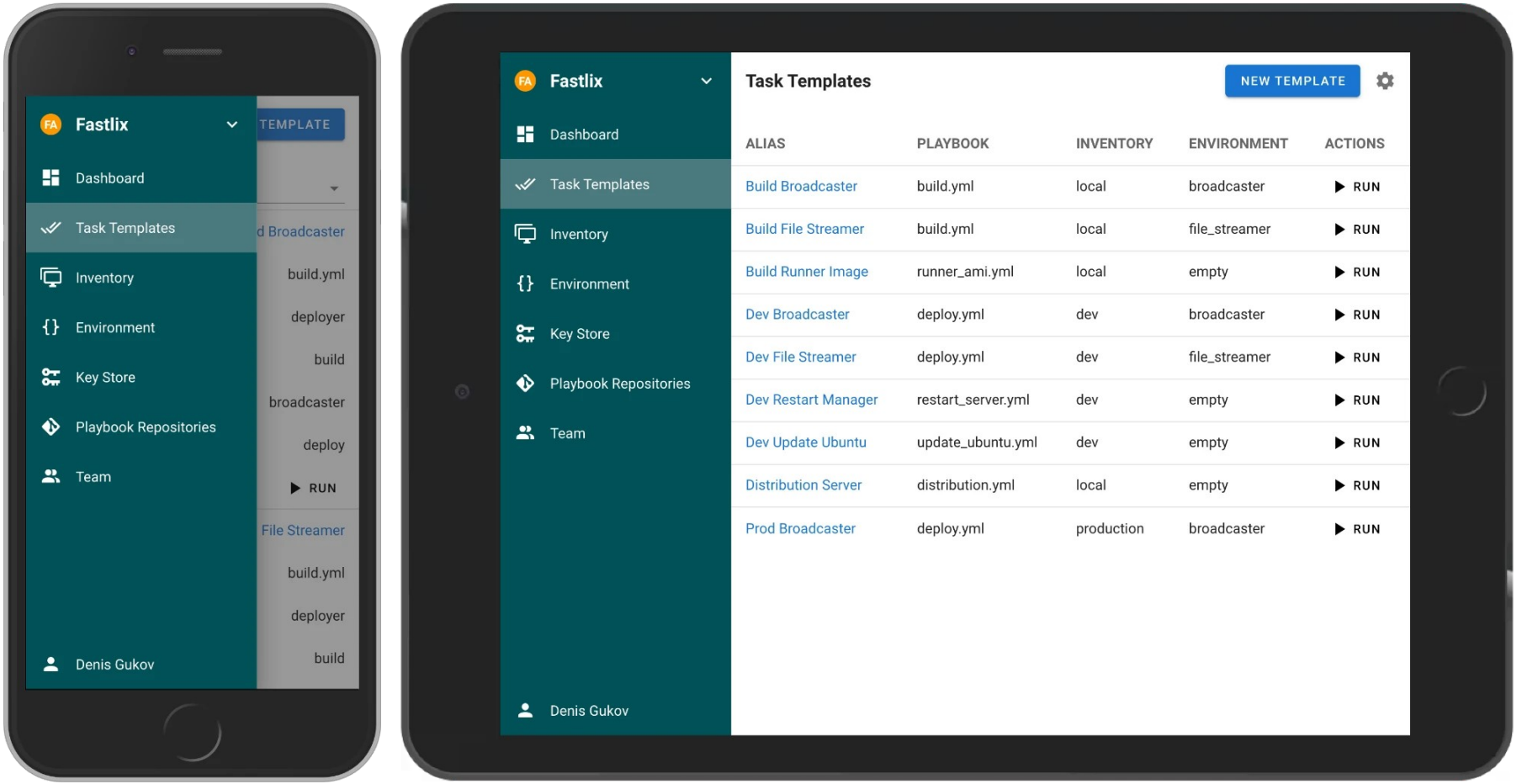
```yaml
- hosts: application_server
  roles:
    - role: deploy-on-tomcat
      vars:
        webapp_source_path: ./demo-app-ansible-deploy-1.0-SNAPSHOT.war
        webapp_target_name: demo1
```

```yaml
- hosts: application_server
  roles:
    - role: deploy-on-tomcat
      vars:
        webapp_source_path: ./demo-app-ansible-deploy-1.0-SNAPSHOT.war
        webapp_target_name: demo2
```

Ansible Infrastruktur



Semaphore UI



IDE - Support

- Eclipse
  - Plugin: YAML Editor
- Intellji IDEA
  - Plugins:
    - Ansible
    - YAML/Ansible Support
- VSCode
  - Extensions: Ansible by Red Hat
- Netbeans

Weitere Features

- Vault – Verschlüsselung
- Facts
- Dynamische Inventories
- Collections
- Playbook Debugger
- Module für Kubernetes, Docker, Cloud allgemein
- Networking Support

Wie werden Ansible Skripte getestet?

- `ansible-playbook --check`
- `ansible-playbook --syntax-check`
- ansible-lint
- Jenkins + Vagrant

Goss

ansible-lint

```
ansible-talk/ansible on  master [!] via A v2.18.4 …
➜ ansible-lint setup-db.yml
WARNING  Listing 10 violation(s) that are fatal
name[play]: All plays should be named.
setup-db.yml:1

yaml[truthy]: Truthy value should be one of [false, true]
setup-db.yml:2

name[casing]: All names should start with an uppercase letter.
setup-db.yml:6:13 Task/Handler: install needed python package

yaml[truthy]: Truthy value should be one of [false, true]
setup-db.yml:10

name[casing]: All names should start with an uppercase letter.
setup-db.yml:12:13 Task/Handler: install mariadb-server

name[casing]: All names should start with an uppercase letter.
setup-db.yml:17:13 Task/Handler: start mariadb

name[casing]: All names should start with an uppercase letter.
setup-db.yml:22:13 Task/Handler: set bind address

name[casing]: All names should start with an uppercase letter.
setup-db.yml:30:13 Task/Handler: creates db user dba
```

ansible-lint

```
Read documentation for instructions on how to ignore specific rule violations.

# Rule Violation Summary

  1 name profile:basic tags:idiom
  1 yaml profile:basic tags:formatting,yaml
  2 yaml profile:basic tags:formatting,yaml
  6 name profile:basic tags:idiom

Failed: 10 failure(s), 0 warning(s) on 1 files. Last profile that met the validation criteria was 'min'.
```

testinfra

```python
def test_openjdk_is_installed(host):
    openjdk = host.package("openjdk-21-jdk")
    assert openjdk.is_installed

def test_tomcat_service_exists(host):
    assert host.file("/etc/systemd/system/tomcat.service").exists

def test_tomcat_folder_exists(host):
    assert host.file("/opt/tomcat").exists
```

```python
def test_mariadb_is_installed(host):
    mariadb = host.package("mariadb-server")
    assert mariadb.is_installed

def test_mariadb_service_is_running(host):
    mariadb = host.service("mariadb")
    assert mariadb.is_enabled
    assert mariadb.is_running

def test_mariadb_config_parameter_exists(host):
    mariadb_conf = host.file("/etc/mysql/mariadb.conf.d/50-server.cnf")
    assert mariadb_conf.contains("bind-address = 0.0.0.0")
```

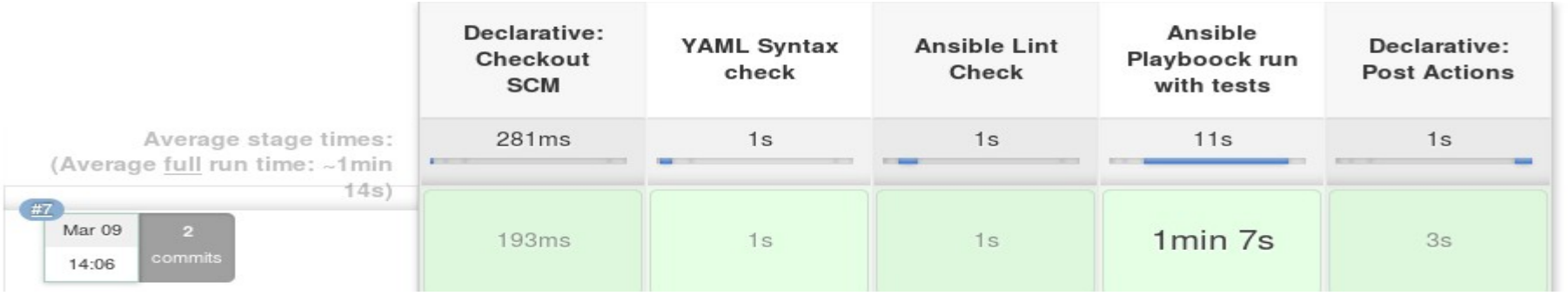
testinfra

```
ansible-talk/ansible on  master [x!?] via Ⓐ v2.18.4 …
➜ py.test --connection=ansible --ansible-inventory inventories/test -v tests/*.py
═══════════════════════════════ test session starts ═══════════════════════════════
platform linux -- Python 3.12.3, pytest-8.4.0, pluggy-1.6.0 -- /home/sparsick/.local/share/pipx/venvs/testinfra/
bin/python
cachedir: .pytest_cache
rootdir: /home/sparsick/dev/workspace/ansible-talk/ansible
plugins: testinfra-6.0.0, testinfra-10.2.2
collected 6 items

tests/test_app.py::test_openjdk_is_installed[ansible://192.168.56.10] PASSED                [ 16%]
tests/test_app.py::test_tomcat_service_exists[ansible://192.168.56.10] PASSED               [ 33%]
tests/test_app.py::test_tomcat_folder_exists[ansible://192.168.56.10] PASSED                [ 50%]
tests/test_db.py::test_mariadb_is_installed[ansible://192.168.56.10] PASSED                 [ 66%]
tests/test_db.py::test_mariadb_service_is_running[ansible://192.168.56.10] PASSED           [ 83%]
tests/test_db.py::test_mariadb_config_parameter_exists[ansible://192.168.56.10] PASSED      [100%]

═══════════════════════════════ 6 passed in 5.36s ═══════════════════════════════
```

Ansible QA Jenkins Pipeline

```
pipeline {
    agent any
        stages {
            stage('YAML Syntax check') {
                steps {
                    ansiblePlaybook inventory: 'inventories/test', extras: '--syntax-check', playbook: 'setup-app.yml'
                    ansiblePlaybook inventory: 'inventories/test', extras: '--syntax-check', playbook: 'setup-db.yml'
                }
            }
            stage('Ansible Lint Check') {
                steps {
                    sh 'ansible-lint *.yml'
                }
            }
            stage('Ansible Playbook run with tests') {
                steps {
                    sh 'cd ..; vagrant up'
                    ansiblePlaybook inventory: 'inventories/test', playbook: 'setup-app.yml'
                    ansiblePlaybook inventory: 'inventories/test', playbook: 'setup-db.yml'
                    sh 'py.test --connection=ansible --ansible-inventory inventories/test -v tests/*.py'
                }
            }
        }
    post {
        always {
            sh 'cd ..; vagrant group destroy -f'
        }
    }
}
```

Ansible QA Jenkins Pipeline



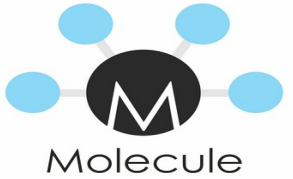
Ansible QA Jenkins Pipeline

Ansible Playboock run with tests - 1m 8s

✓	> vagrant up centos-docker — Shell Script	2s
✓	> Invoke an ansible playbook	58s
✓	> py.test --connection=ansible --ansible-inventory inventories/localhost_d... — Shell Script	8s
✓	> vagrant destroy centos-docker -f — Shell Script	3s

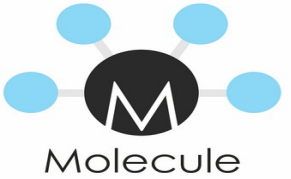
Geht es nicht einfacher?

Molecule



- Spezialisiert für Ansible Roles
- Wrapper um andere Werkzeuge, um komplette Test Szenarien aufzubauen
  - Driver Provider: Docker, Podman, Vagrant, Azure, EC2, GCE, OpenStack
  - Lint Provider: yamllint (default), ansible-lint
  - Verifier framework: TestInfra, Ansible

Molecule




```
.
└── tomcat
    ├── defaults
    │   └── main.yml
    ├── handlers
    │   └── main.yml
    ├── meta
    │   └── main.yml
    ├── molecule
    │   └── default
    │       ├── Dockerfile.j2
    │       ├── molecule.yml
    │       ├── playbook.yml
    │       └── tests
    │           ├── test_default.py
    │           └── test_default.pyc
    ├── README.md
    ├── tasks
    │   └── main.yml
    └── vars
        └── main.yml
```

molecule test

Wie unterscheidet sich Ansible zu seiner Konkurrenz?

Vergleich

- Orchestrierung über SSH
- Benötigt keine Rootrechte auf Zielsystem
- Konfigurationsmgmt + Applikationsdeployment
- Monitoringtool nur in der Enterprise Variante
- Skripte mehr imperativ
- Windows-Support rudimentär
- Skripte OS- bzw. Distributions-spezifisch

- Client-Server Architektur
- Für komfortables Arbeiten benötigt es Rootrechte
- Konfigurationsmgmt

- Monitoringtools Open Source
- Skripte mehr deklarativ
- Windows-Support

- Skripte können OS-unspezifisch sein

Vergleich

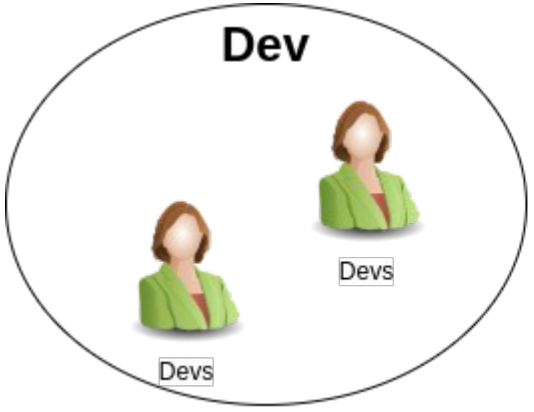

```yaml
- hosts: 192.168.56.10
  become: true
  become_method: sudo
  tasks:
    - name: Ensure Python modules
      ansible.builtin.apt:
        name: python-apt

    - name: add nodejs ppa
      ansible.builtin.apt_repository:
        repo: ppa:chris-lea/node.js

    - name: install nodejs package
      ansible.builtin.apt:
        name: nodejs
```

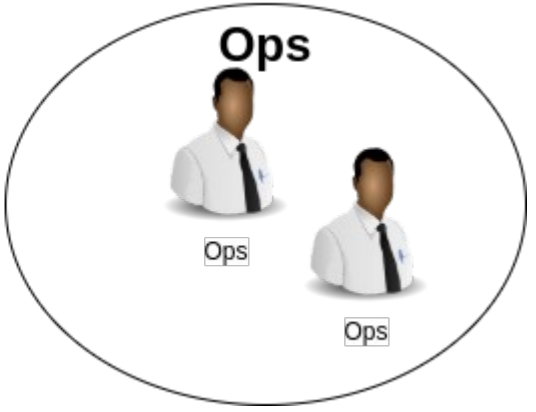
```
class nodejs {
  class { 'apt':  }

  exec { 'apt-get-update':
    command     => '/usr/bin/apt-get update',
  }

  package {'software-properties-common' :
    ensure=> installed,
    require => Exec['apt-get-update'],
  }

  apt::ppa {'ppa:chris-lea/node.js' :}

  package { 'nodejs' :
    ensure => installed,
    require => Apt::Ppa ['ppa:chris-
lea/node.js'],

  }
}
```

Weitere Einsatzszenarien aus Entwicklersicht

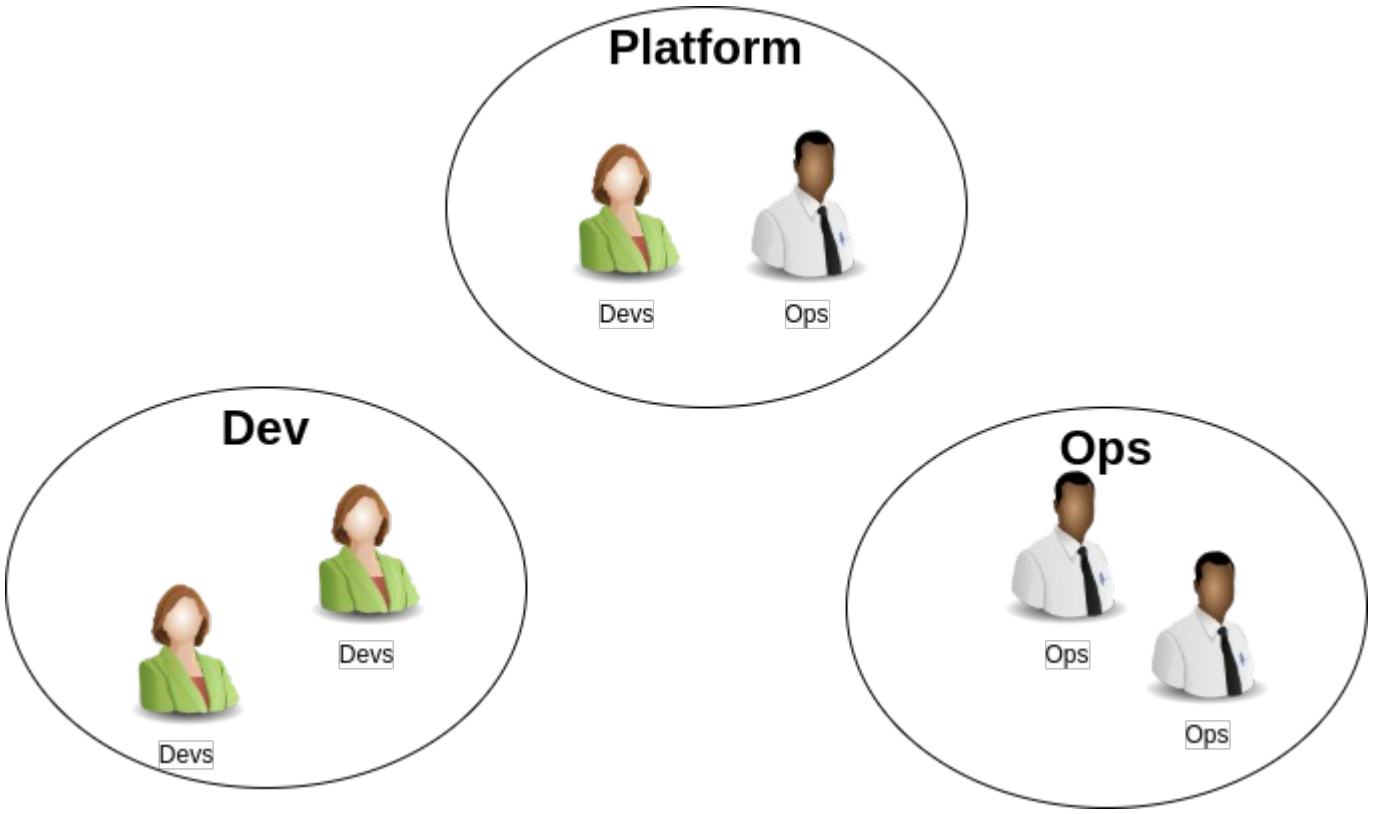
Systemkonfiguration für Entwickler

Organisatorische Ausgangslage
Realität



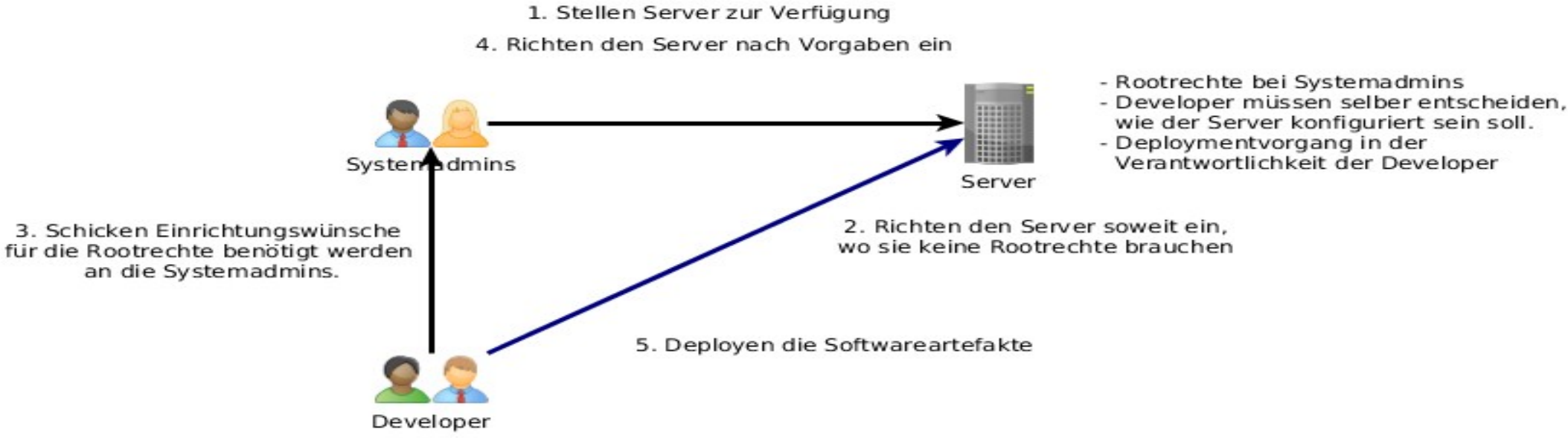


Systemkonfiguration für Entwickler



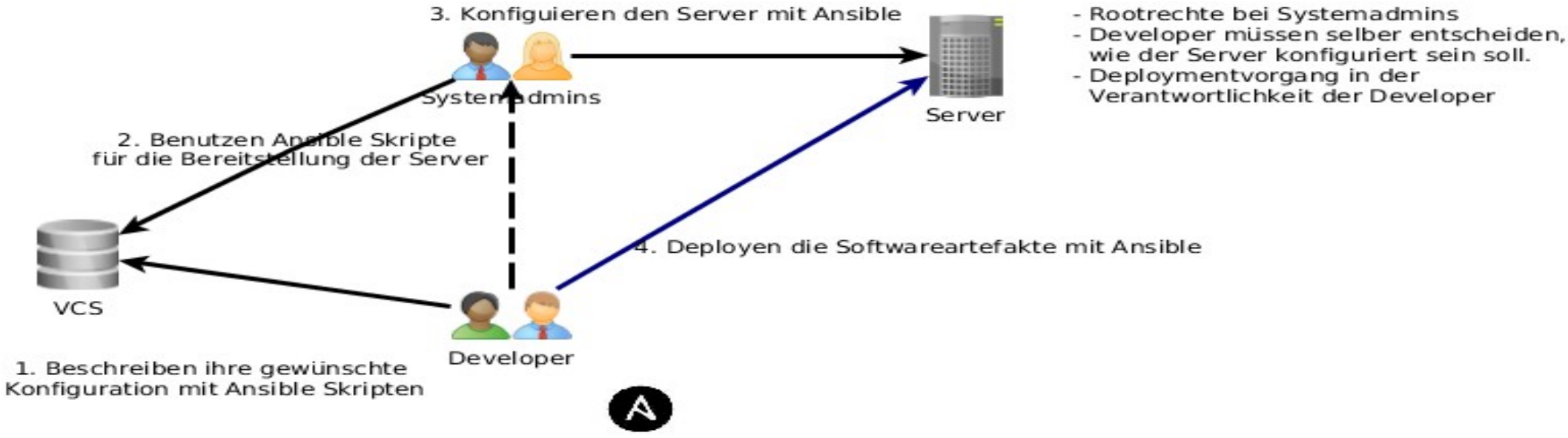
Systemkonfiguration für Entwickler

Prozess zwischen Development und Operation



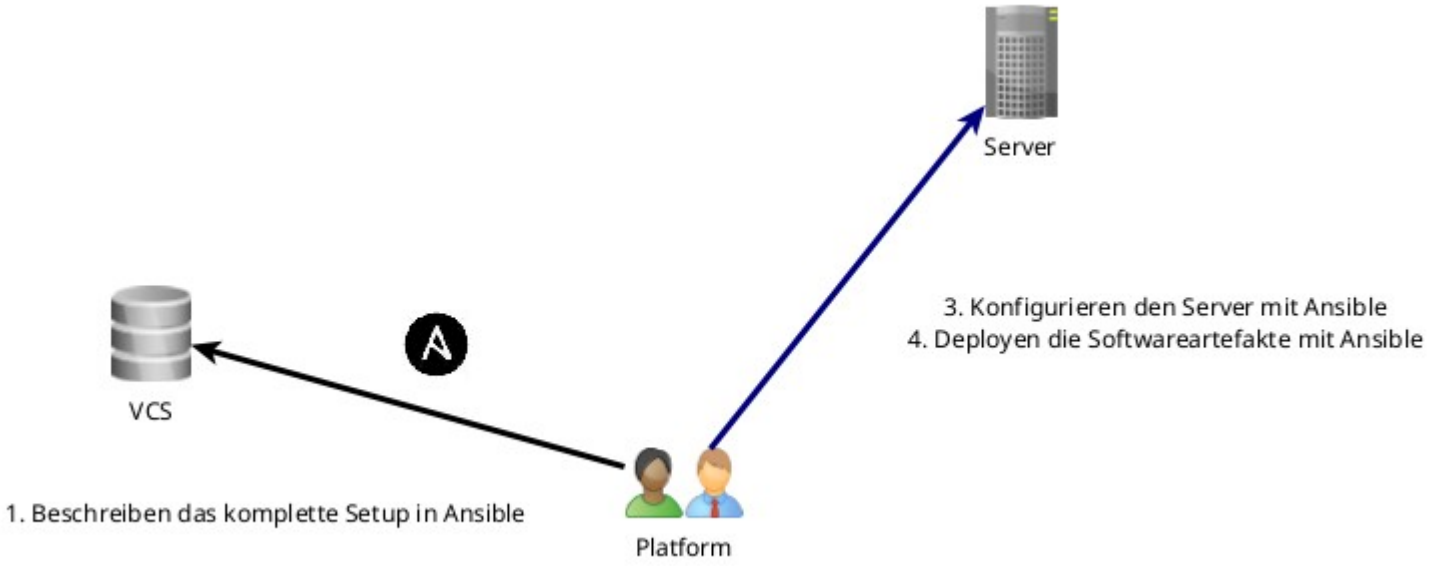
Systemkonfiguration für Entwickler

Lösungidee mit Ansible



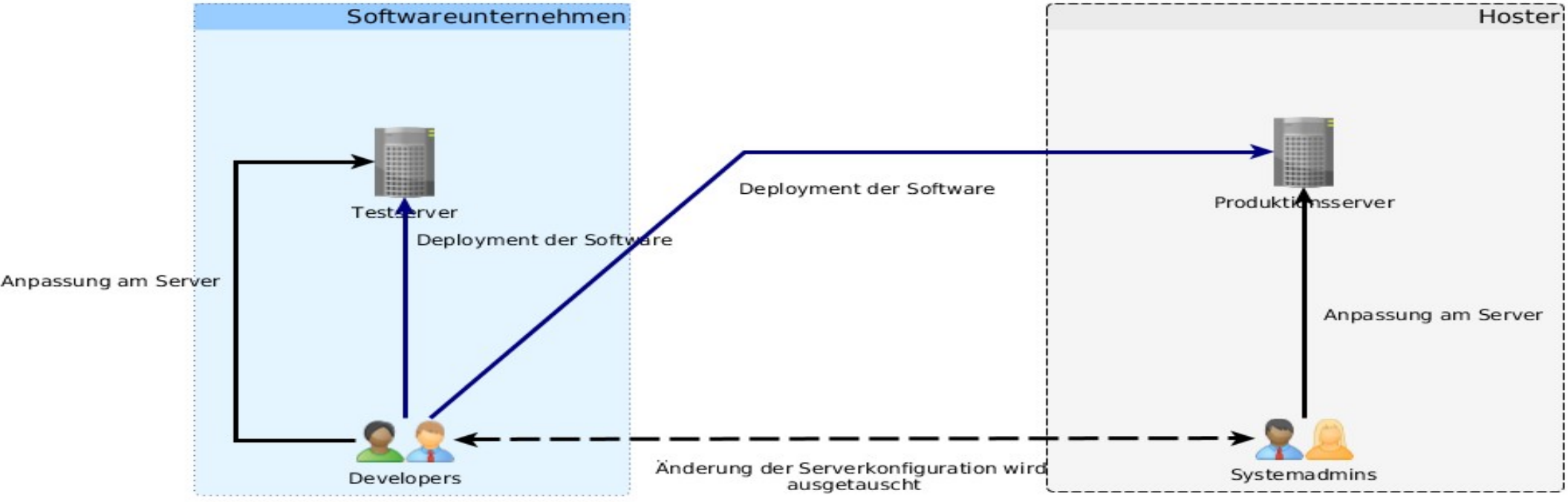
Systemkonfiguration für Entwickler

Lösungidee mit Ansible



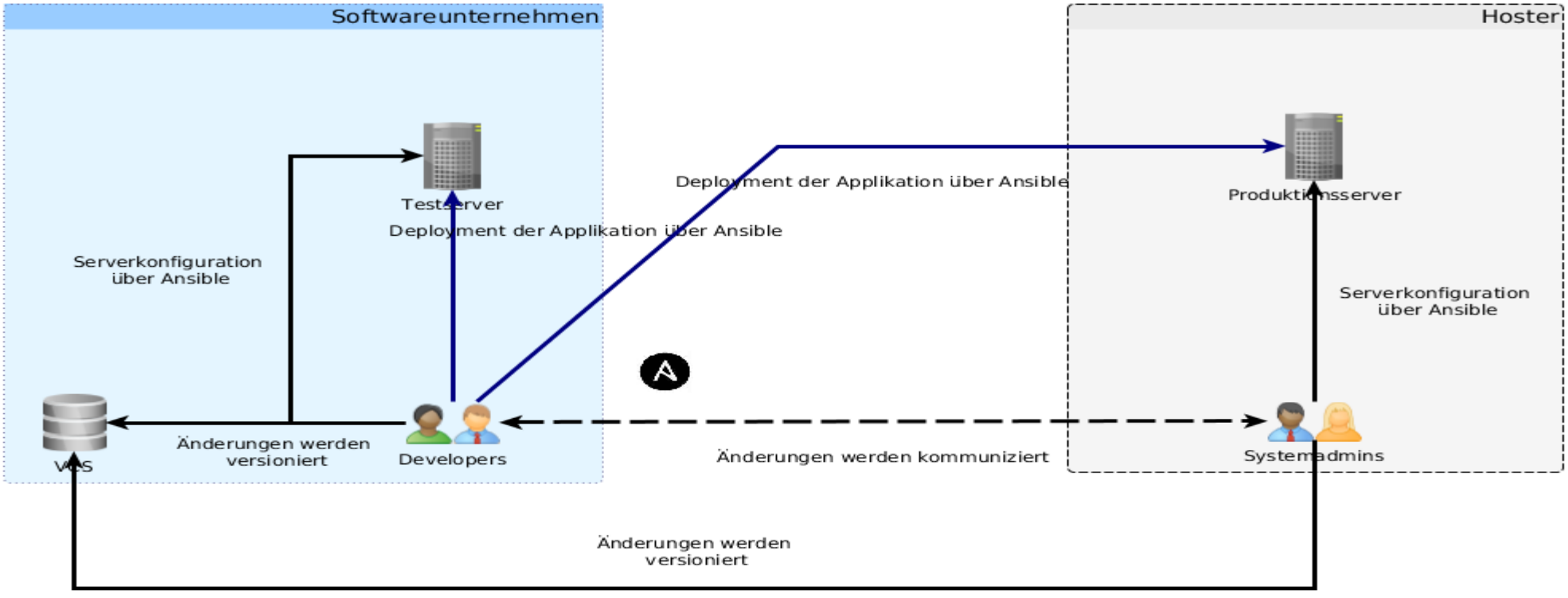
Systemkonfiguration für Entwickler

Produktionsserver sind beim externen Hoster



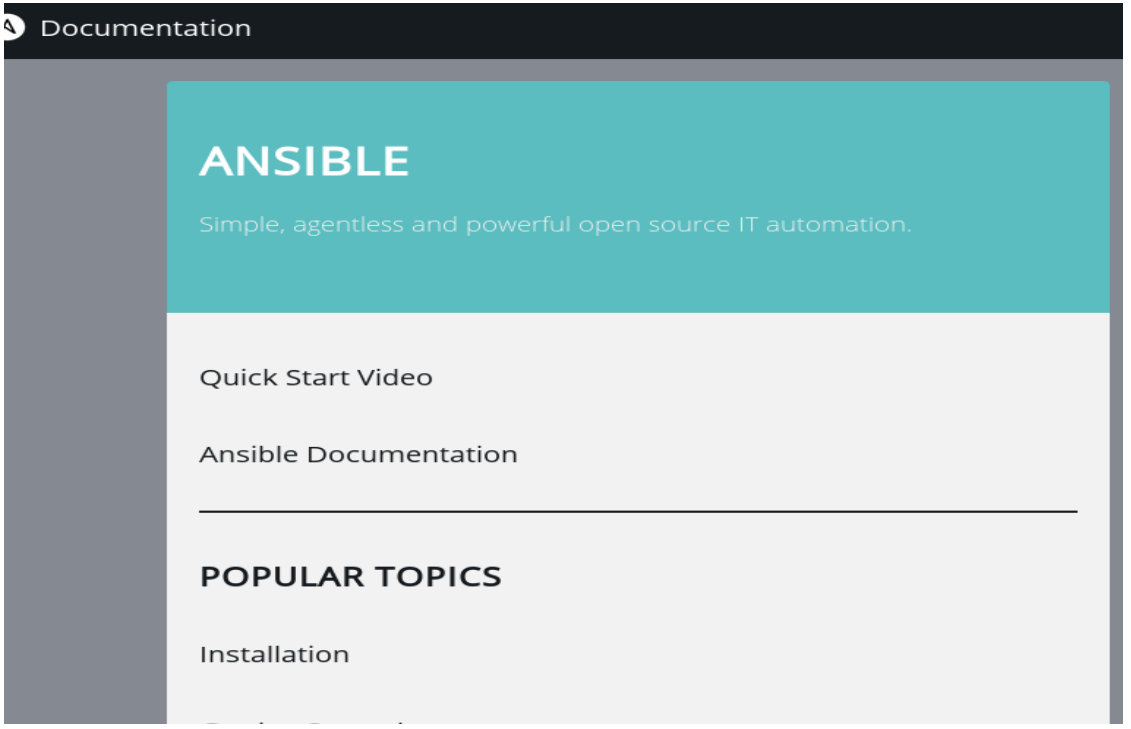
Systemkonfiguration für Entwickler

Lösungsidee



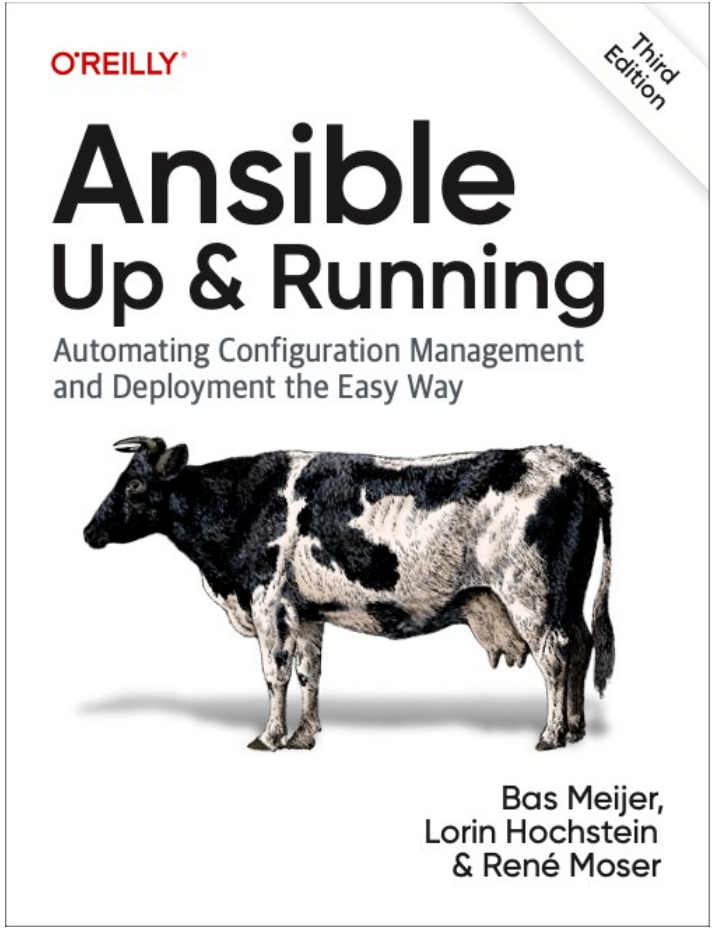
- Hoster verantwortlich für die Systemkonfiguration
- Softwareunternehmen verantwortlich für das Deployment
- Synchronisation zwischen Testserver und Produktionsserver wird vereinfacht

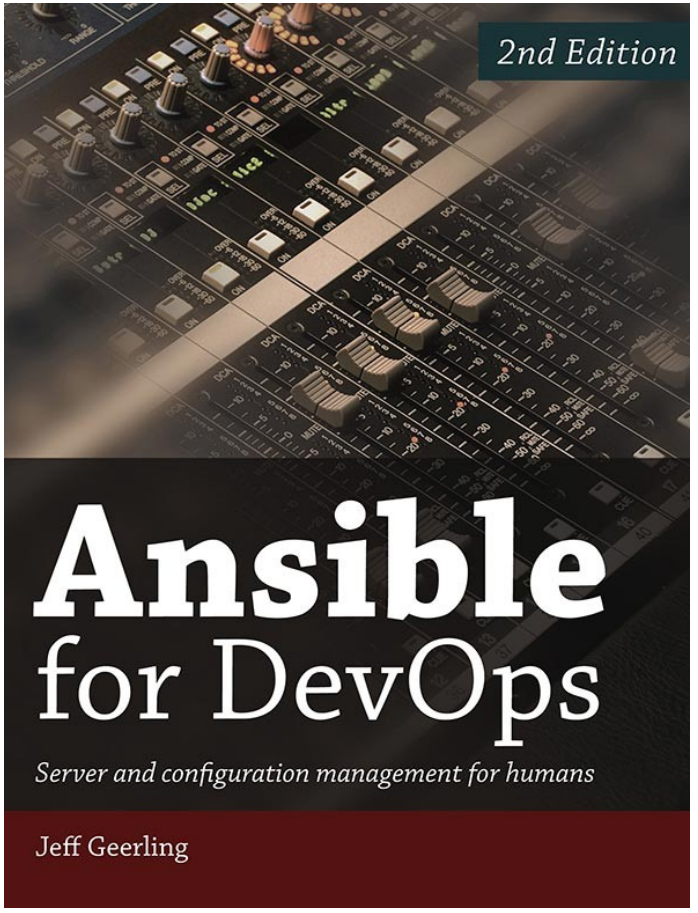
Weitere Informationen



http://docs.ansible.com/

Weitere Informationen





Fragen?

@sparsick@mastodon.social
mail@sandra-parsick.de
https://github.com/sparsick/ansible-talk.git