rheinjug 18.02.2016

Ansible für Entwickler

Sandra Parsick

info@sandra-parsick.de
@SandraParsick

Zur meiner Person

- Freiberufliche Softwareentwickler und Consultant im Java-Umfeld

- Schwerpunkte:

  - Java Enterprise Anwendungen

  - Agile Methoden

  - Software Craftmanship

  - Automatisierung von Entwicklungsprozessen
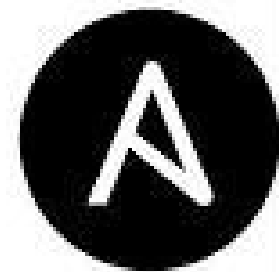
- Softwerkskammer Dortmund

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."
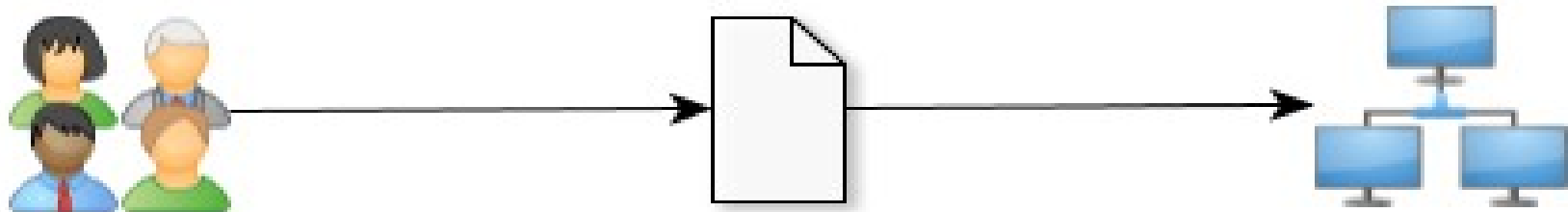*https://www.projektmagazin.de/glossarterm/konfigurationsmanagement*

Konfigurationsmanagement (KM)

- Softwarekonfiguration
- Hardwarekonfiguration
- Dienstleistungskonfiguration
- Systemkonfiguration

Systemkonfiguration
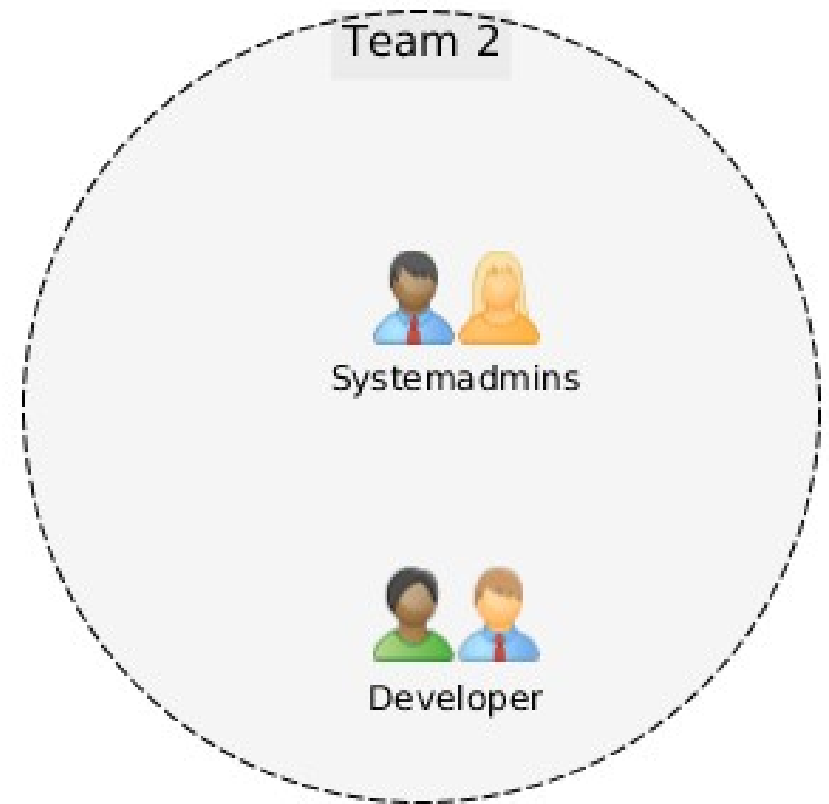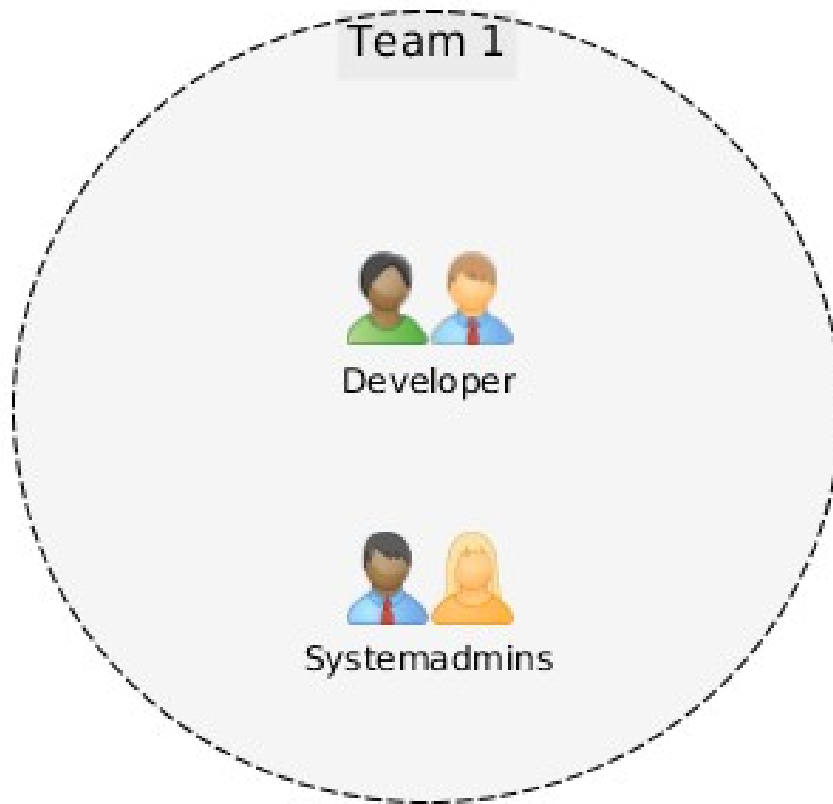# - „Infrastructure As Code"

Systemkonfiguration
- „Infrastructure As Code"
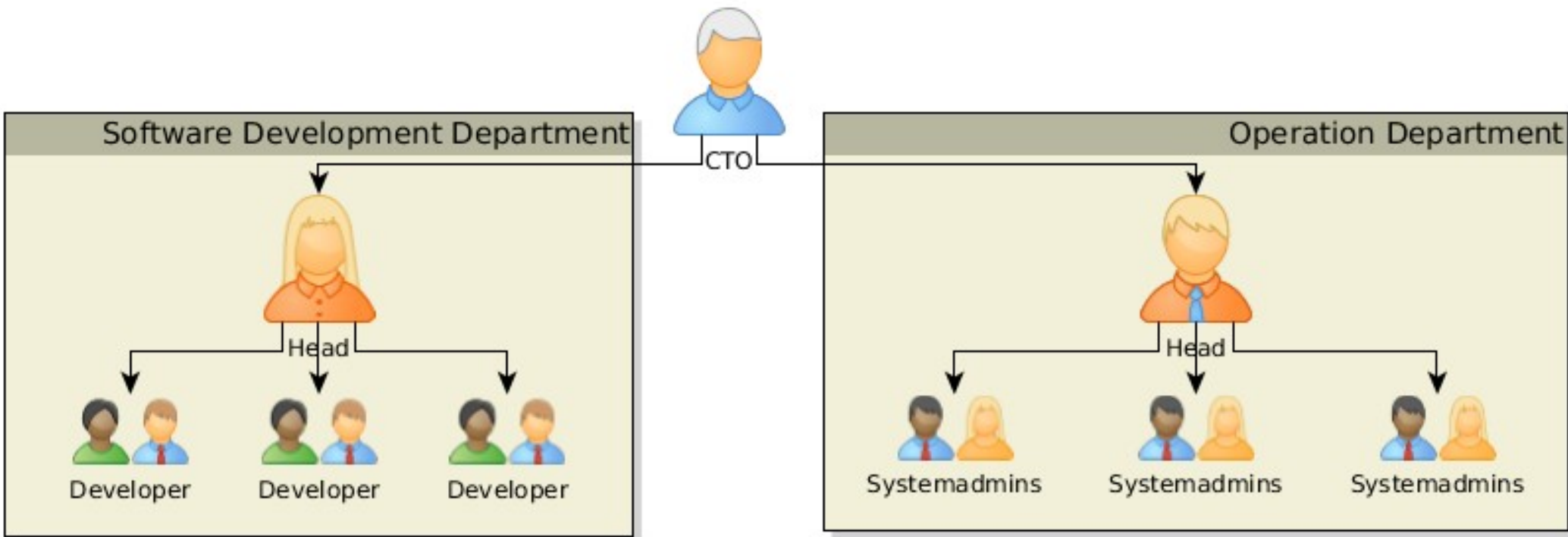
Warum ist es für Entwickler interessant?

Systemkonfiguration für Entwickler

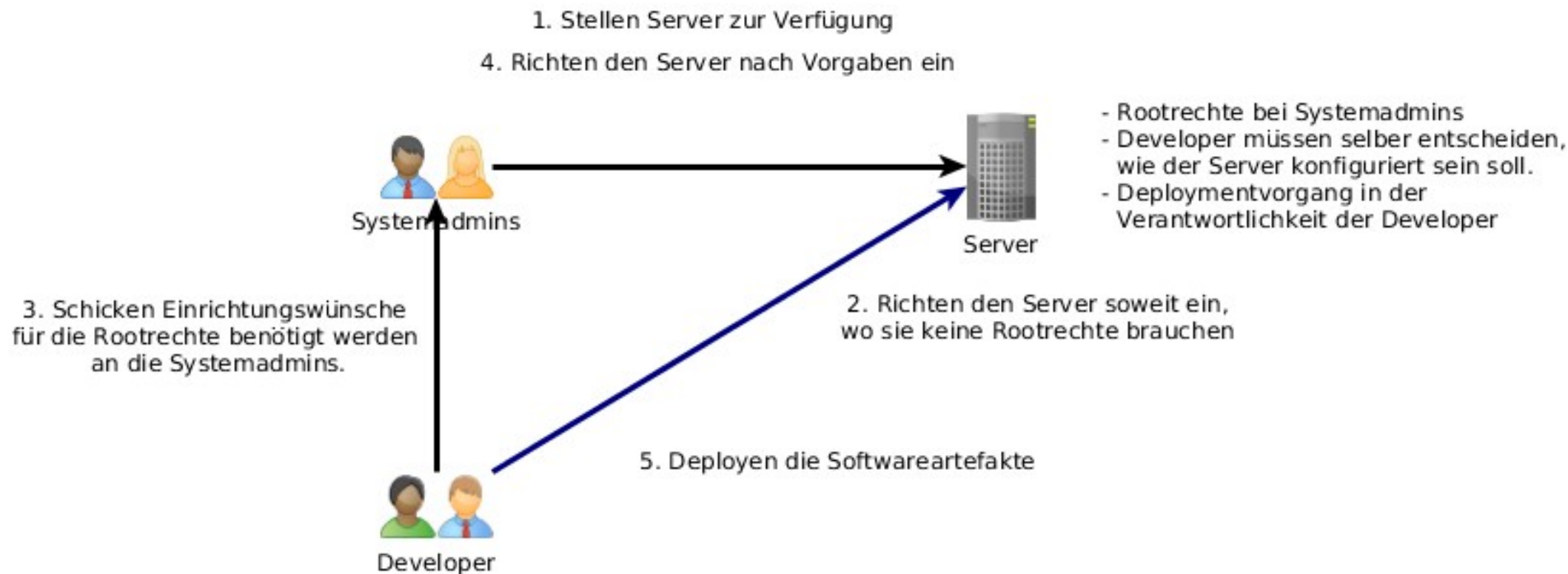## Organisatorische Ausgangslage
Wunsch

Systemkonfiguration für Entwickler
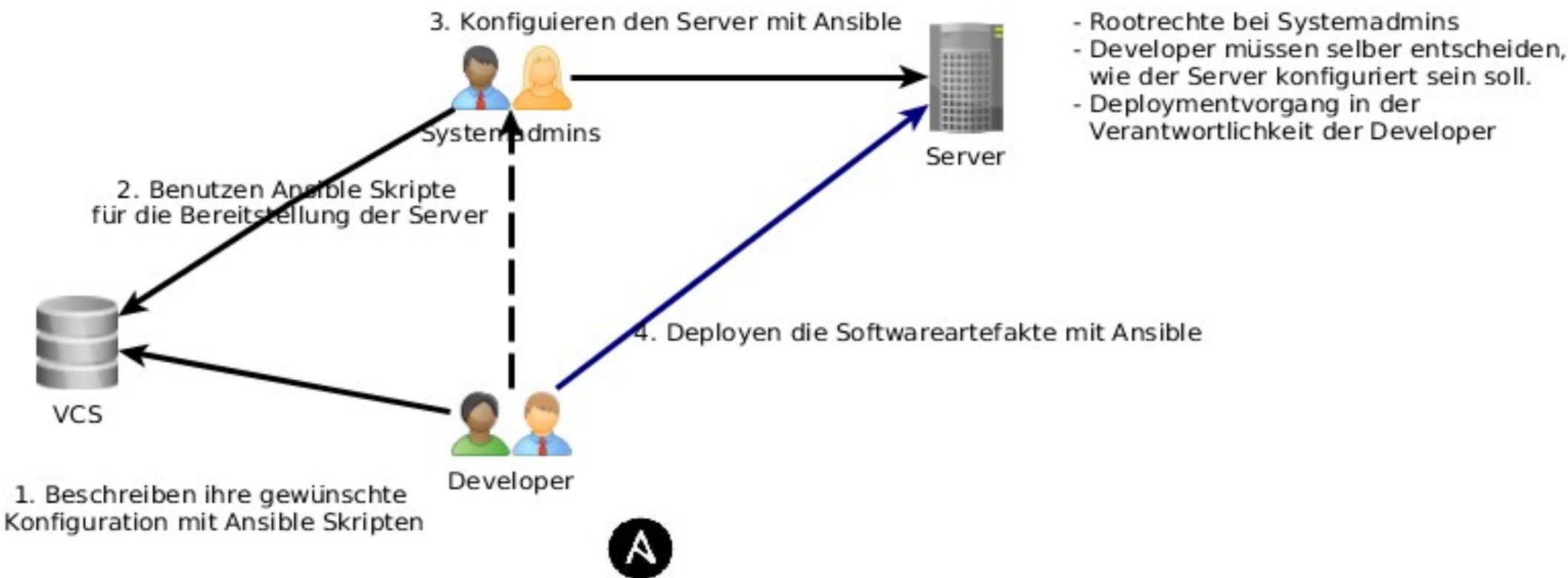
Organisatorische Ausgangslage
Realität

Systemkonfiguration für Entwickler

Prozess zwischen Development und Operation
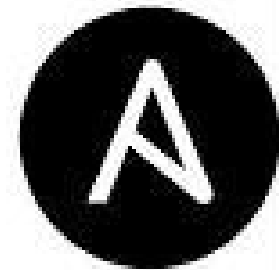
Systemkonfiguration für Entwickler
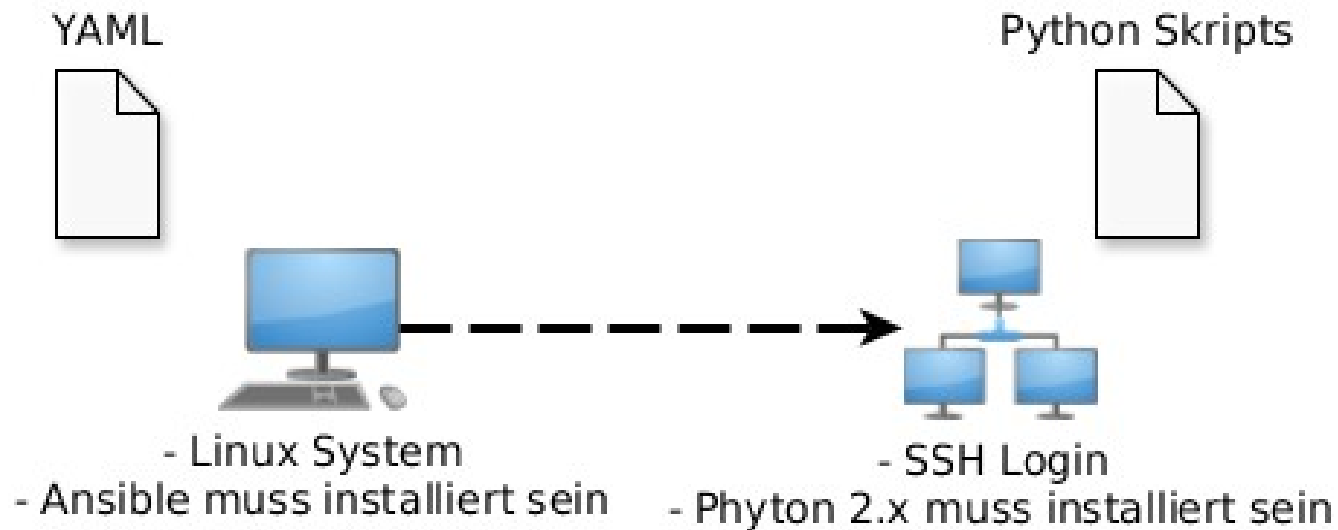
Lösungidee mit Ansible

Einführung in Ansible

Ansible

- Software für

  – Konfigurationsmanagement,

  – Softwareverteilung und

  – Ad-hoc-Kommando-Ausführung

- Sprache: Python

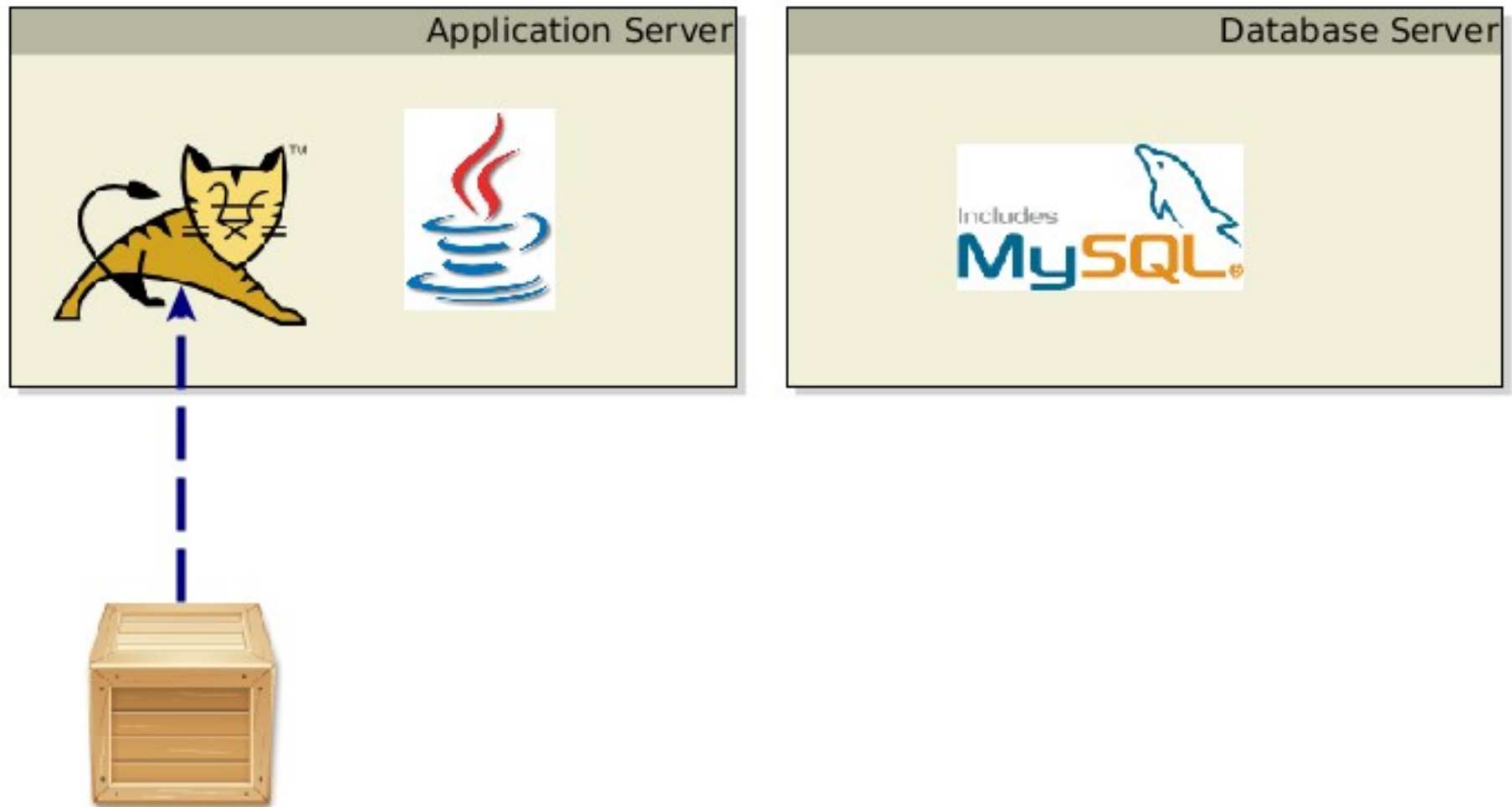- Ansible Skripte: YAML

Funktionsweise

Exkurs: YAML

YAML

```yaml
---
foo: "bar"
baz:
  - "qux"
  - "quxx"
corge: null
grault: 1
garply: true
waldo: "false"
fred: "undefined"
emptyArray: []
emptyObject: {}
emptyString: ""
```
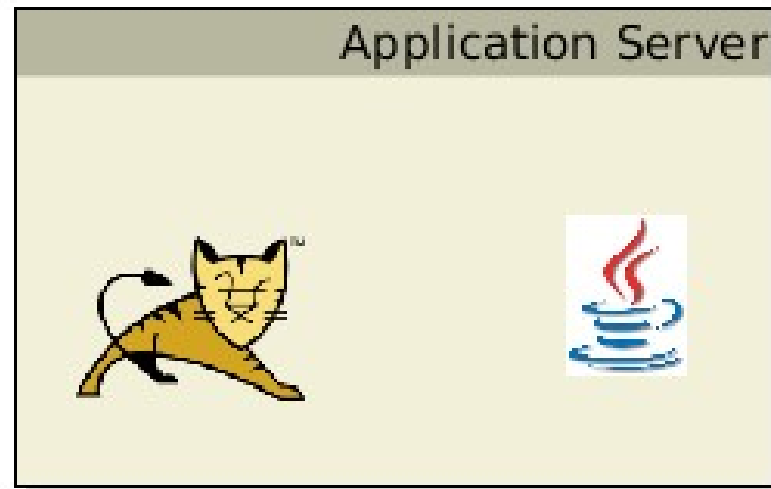
JSON

```json
{
  "foo": "bar",
  "baz": [
    "qux",
    "quxx"
  ],
  "corge": null,
  "grault": 1,
  "garply": true,
  "waldo": "false",
  "fred": "undefined",
  "emptyArray": [],
  "emptyObject": {},
  "emptyString": ""
}
```

Ansible Beispiel

Setup Application Server Playbook

```yaml
- hosts: application-server
  vars:
    tomcat_version: 8.0.24
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      apt: name=openjdk-7-jdk state=present
      become: yes
      become_method: sudo

    - name: Download current Tomcat 8 version
      local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version
        }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp

    - name:
      file: name=/opt mode=777
      become: yes
      become_method: sudo

    - name: Install Tomcat 8
      unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name
        }} owner=vagrant group=vagrant

    - name: Set link to tomcat 8
      file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat  state=link force=yes

    - name: setup setenv.sh
      template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh"
        src="roles/tomcat8/templates/setenv.sh.j2" mode=755
      when: catalina_opts is defined

    - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
      register: tomcat_scripts
      ignore_errors: yes
```
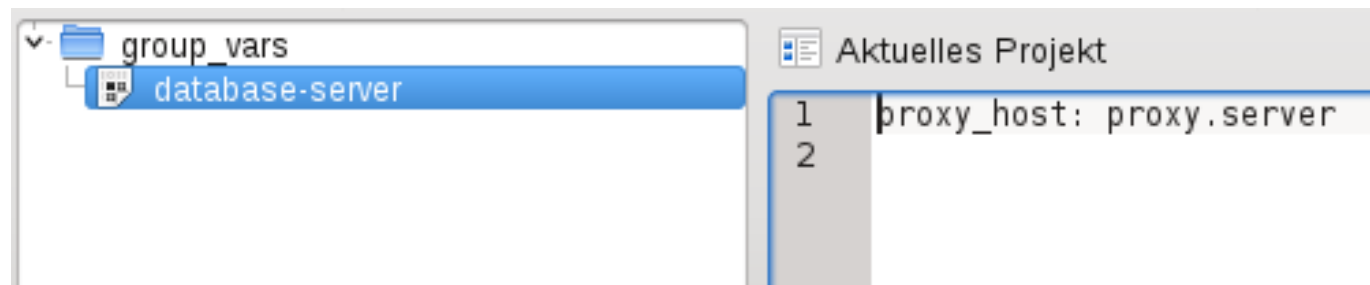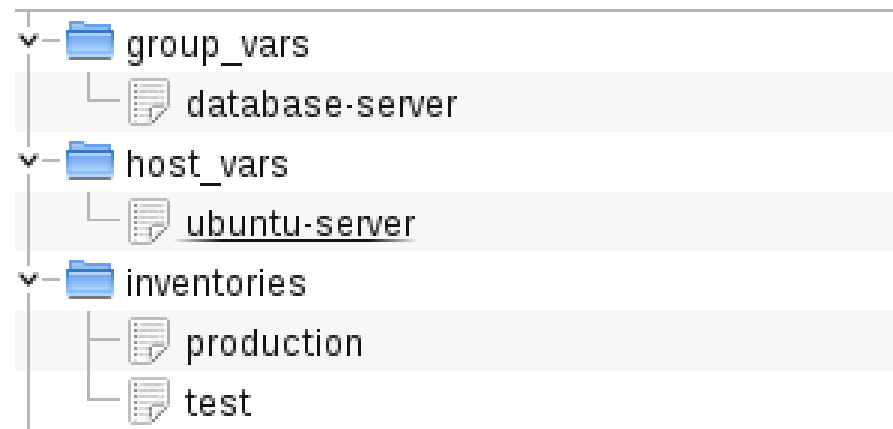
Inventories

Production

```
1   [application-server]
2   192.168.33.10
3   ubuntu-server db_host=mysql01
4
5   [mysql-db-server]
6   mysql[01:10]
7
8   [oracle-db-server]
9   db-[a:f].oracle.company.com
10
11  [database-server:children]
12  mysql-db-server
13  oracle-db-server
14  |
15  [application-server:vars]
16  message="Welcome"
17
18  [database-server:vars]
19  message="Hello World!"
```

Test

```
1   [application-server]
2   192.168.33.10|
3
4   [database-server]
5   192.168.33.10
6
```

Inventories

```yaml
12
13      - name: Download current Tomcat 8 version
14        local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version
          }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp
15
16      - name:
17        file: name=/opt mode=777
18        become: yes
19        become_method: sudo
20
21      - name: Install Tomcat 8
22        unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name
          }} owner=vagrant group=vagrant
23
24      - name: Set link to tomcat 8
25        file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat  state=link force=yes
26
27      - name: setup setenv.sh
28        template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh"
          src="roles/tomcat8/templates/setenv.sh.j2" mode=755
29        when: catalina_opts is defined
30
31      - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
32        register: tomcat_scripts
33        ignore_errors: yes
34
35      - name: ensure tomcat scripts are executable
36        file: name={{item}} mode=755
37        with_items: tomcat_scripts.stdout_lines
38
39      - name: install init.d script for tomcat
40        copy: src=roles/tomcat8/files/init.d/tomcat dest=/etc/init.d/tomcat owner=vagrant
          group=vagrant mode=755
41        become: yes
42        become_method: sudo
```

Templates

- setenv.sh.j2

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

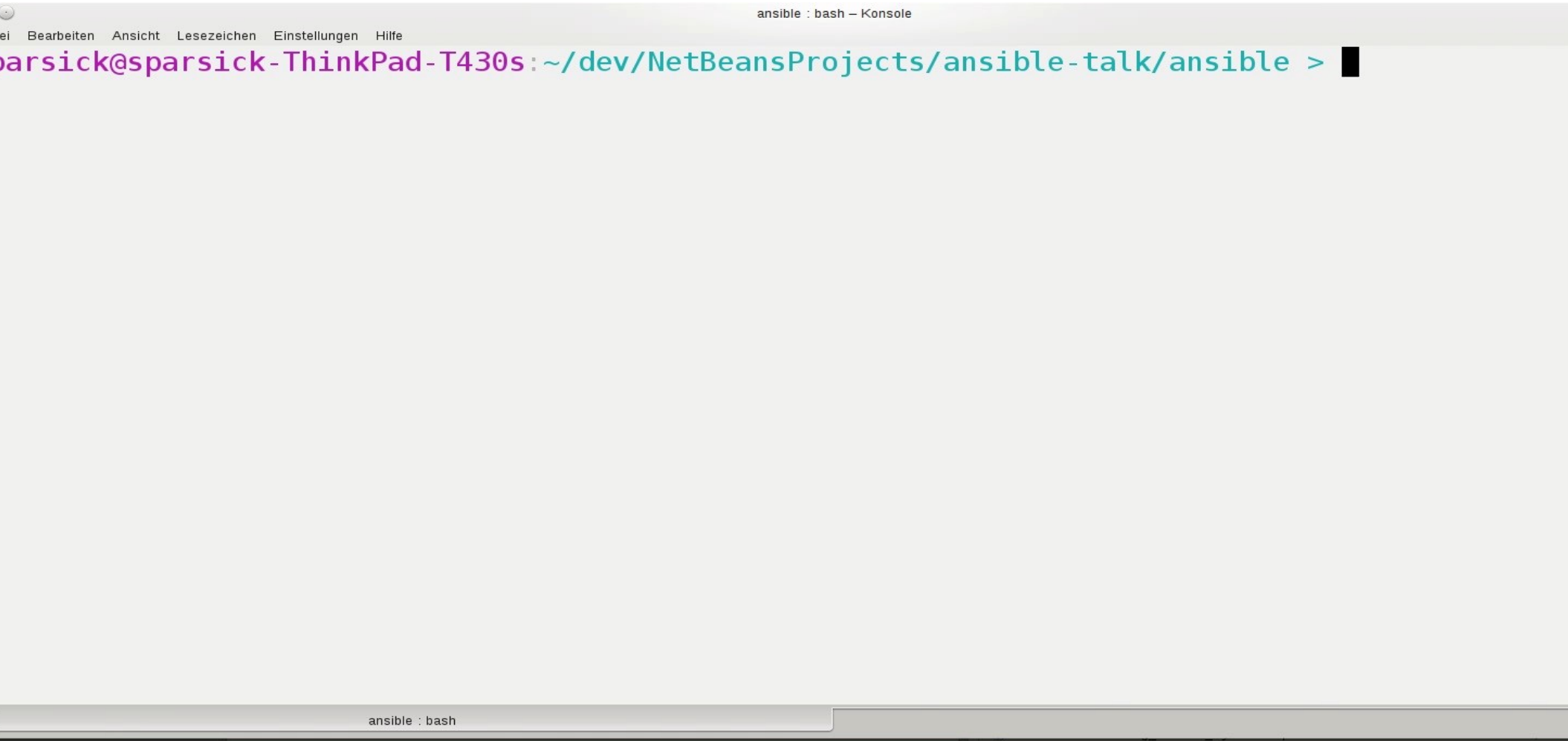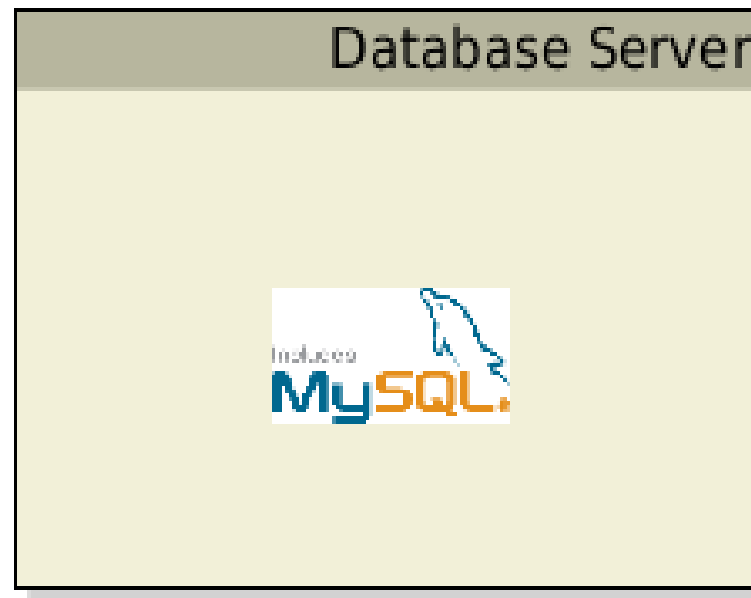# Templates - Jinja2

- Templating engine für Python

```
<title>{% block title %}{% endblock %}</title>
<ul>
{% for user in users %}
  <li><a href="{{ user.url }}">{{ user.username }}</a></li>
{% endfor %}
</ul>
```

- Mehr Information unter
  http://jinja.pocoo.org/docs/dev/

Setup Application Server Playbook

Setup Database Server Playbook

```yaml
  - hosts: database-server
    become: yes
    become_method: sudo

    tasks:
      - name: install mysql db
        apt: name=mysql-server state=present

      - name: installs python-mysqldb
        apt: name=python-mysqldb state=present

      - name: start mysql
        service: name=mysql state=started

      - name: set bind address
        lineinfile: dest=/etc/mysql/my.cnf
          line='bind-address = 0.0.0.0'
          state=present
          regexp=^bind-address(.*)
        notify: restart mysql

      - name: creates db user dba
        mysql_user: name=dba password=g3h3lm priv=*.*:ALL,GRANT state=present host=%

    handlers:
      - name: restart mysql
        service: name=mysql state=restarted
```

Setup Database Server Playbook

Setup Database Server Playbook

```yaml
- hosts: application-server
  vars:
    tomcat_version: 8.0.24
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      apt: name=openjdk-7-jdk state=present
      become: yes
      become_method: sudo

    - name: Download current Tomcat 8 version
      local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version
        }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp

    - name:
      file: name=/opt mode=777
      become: yes
      become_method: sudo

    - name: Install Tomcat 8
      unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name
        }} owner=vagrant group=vagrant

    - name: Set link to tomcat 8
      file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat  state=link force=yes

    - name: setup setenv.sh
      template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh"
        src="roles/tomcat8/templates/setenv.sh.j2" mode=755
      when: catalina_opts is defined

    - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
      register: tomcat_scripts
      ignore_errors: yes
```
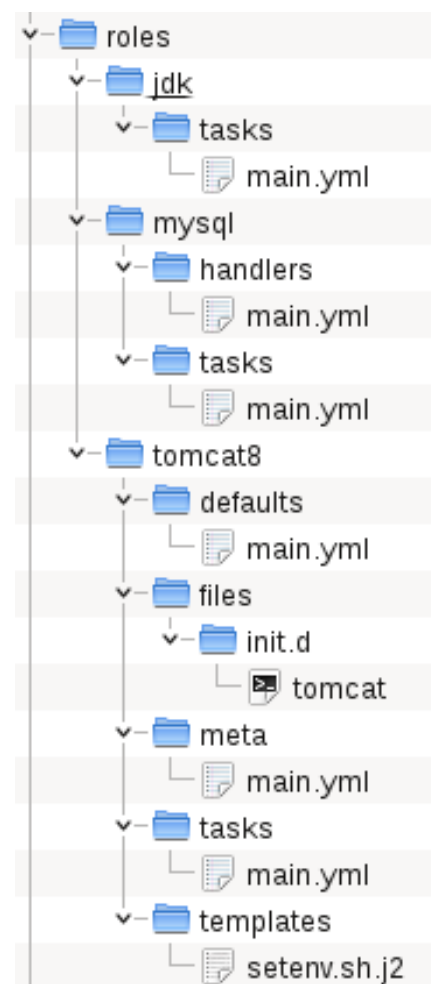
```yaml
12
13      - name: Download current Tomcat 8 version
14        local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version
          }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp
15
16      - name:
17        file: name=/opt mode=777
18        become: yes
19        become_method: sudo
20
21      - name: Install Tomcat 8
22        unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name
          }} owner=vagrant group=vagrant
23
24      - name: Set link to tomcat 8
25        file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat  state=link force=yes
26
27      - name: setup setenv.sh
28        template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh"
          src="roles/tomcat8/templates/setenv.sh.j2" mode=755
29        when: catalina_opts is defined
30
31      - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
32        register: tomcat_scripts
33        ignore_errors: yes
34
35      - name: ensure tomcat scripts are executable
36        file: name={{item}} mode=755
37        with_items: tomcat_scripts.stdout_lines
38
39      - name: install init.d script for tomcat
40        copy: src=roles/tomcat8/files/init.d/tomcat dest=/etc/init.d/tomcat owner=vagrant
          group=vagrant mode=755
41        become: yes
42        become_method: sudo
```

```yaml
1   - hosts: database-server
2     become: yes
3     become_method: sudo
4
5     tasks:
6       - name: install mysql db
7         apt: name=mysql-server state=present
8
9       - name: installs python-mysqldb
10        apt: name=python-mysqldb state=present
11
12      - name: start mysql
13        service: name=mysql state=started
14
15      - name: set bind address
16        lineinfile: dest=/etc/mysql/my.cnf
17          line='bind-address = 0.0.0.0'
18          state=present
19          regexp=^bind-address(.*)
20        notify: restart mysql
21
22      - name: creates db user dba
23        mysql_user: name=dba password=g3h3lm priv=*.*:ALL,GRANT state=present host=%
24
25    handlers:
26      - name: restart mysql
27        service: name=mysql state=restarted
28
```

Roles

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

```
roles
    jdk
        tasks
            main.yml
    mysql
        handlers
            main.yml
        tasks
            main.yml
    tomcat8
        defaults
            main.yml
        files
            init.d
                tomcat
        meta
            main.yml
        tasks
            main.yml
        templates
            setenv.sh.j2
```

Setup Playbooks mit Roles

- Setup Application Server

```
1   - hosts: application-server
2     roles:
3       - jdk
4       - { role: tomcat8, tomcat_version: 8.0.30 }
5
6
7
```

- Setup Database Server

```
1   - hosts: database-server
2     roles:
3       - mysql
4
5
6
```

Java Webapplikation Deployment
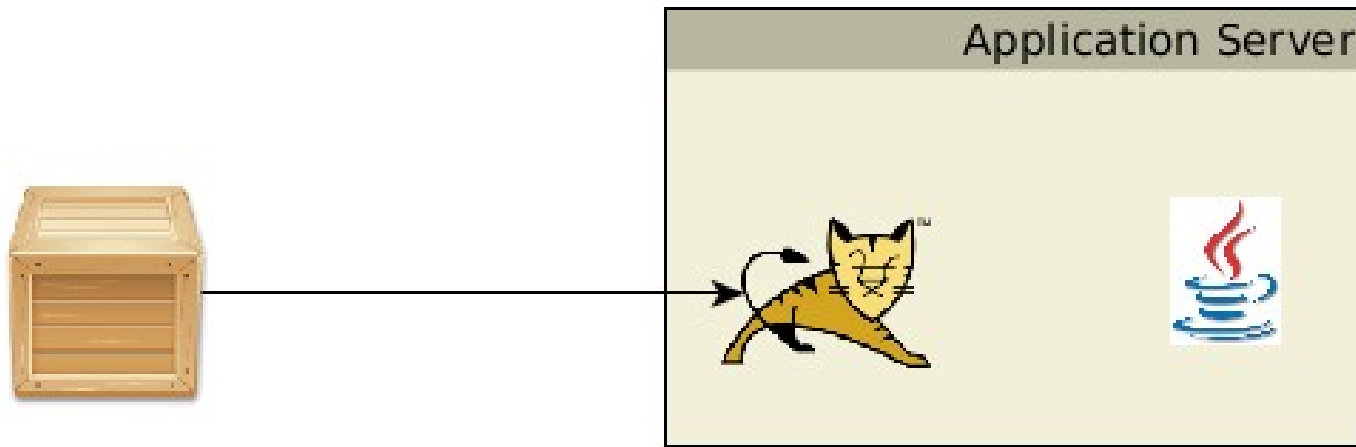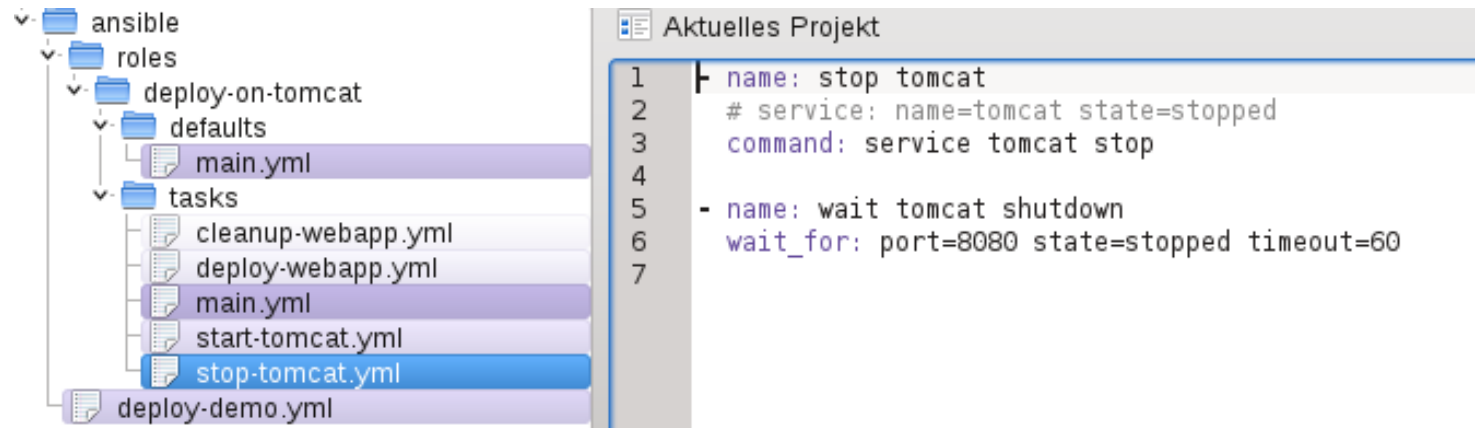
Deploy Application Playbook

```
1   - hosts: application-server
2     roles:
3       - {role: deploy-on-tomcat, webapp_source_path: ./demo-app-ansible-deploy-1.0-
        SNAPSHOT.war, webapp_target_name: demo }
```

deploy-on-tomcat Role

deploy-on-tomcat Role

deploy-on-tomcat Role

deploy-on-tomcat Role

# deploy-on-tomcat Role

deploy-on-tomcat Role

Deploy Application Playbook

Warum Roles?
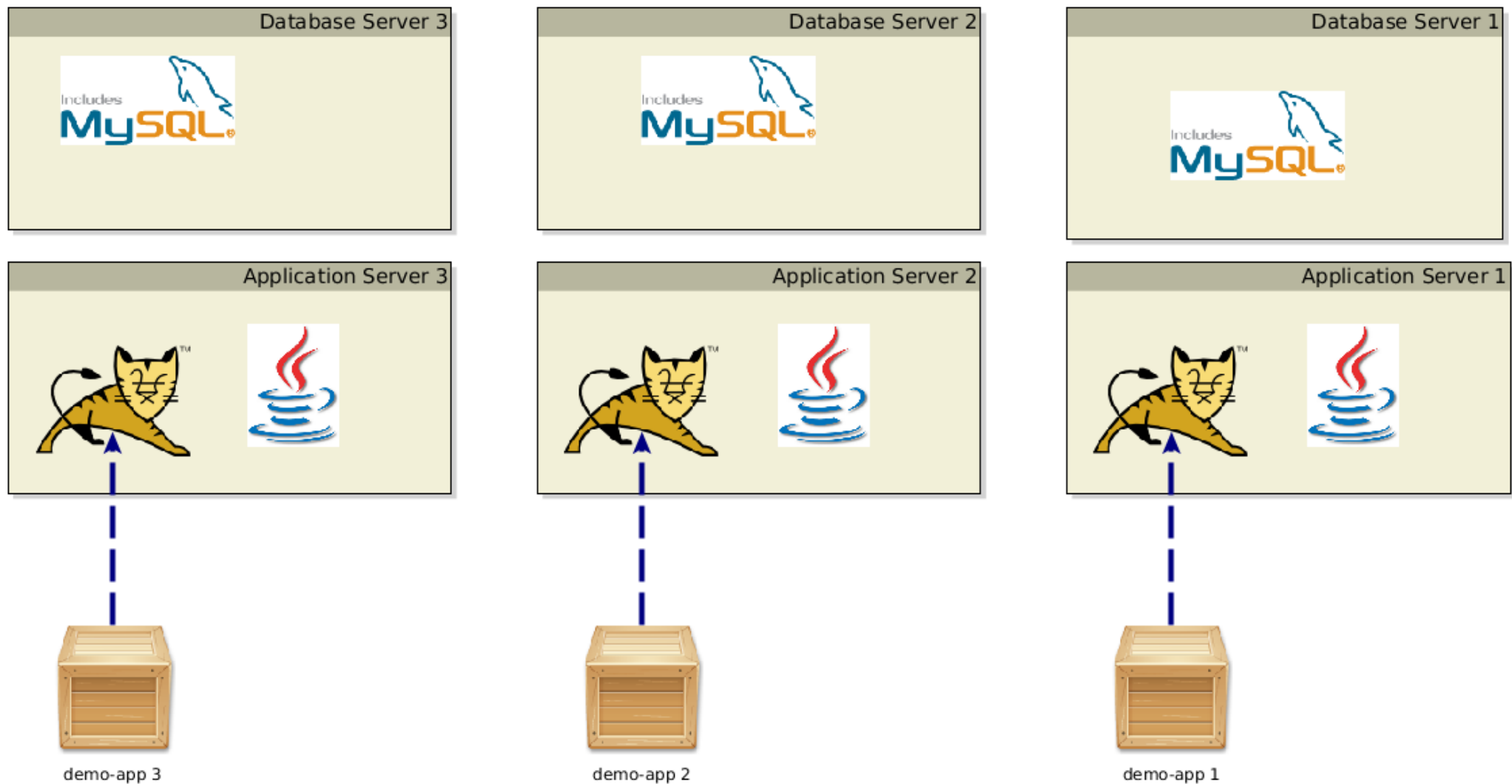
Warum Roles?

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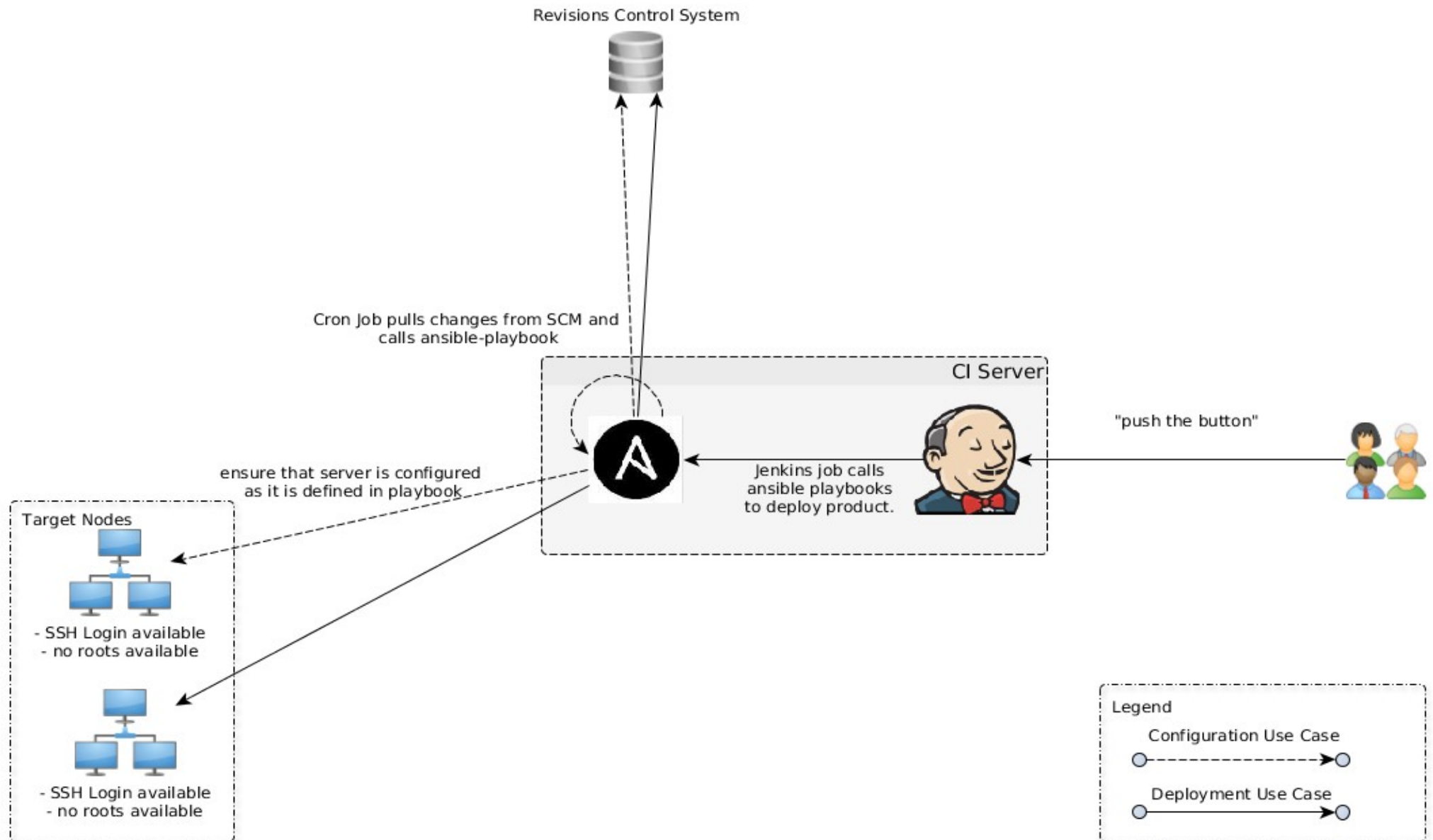setup-demo2-database.yml

Warum Roles?

```
1   - hosts: demo1-application-server
2     roles:
3       - {role: deploy-on-tomcat, webapp_source_path: ./demo1-1.0-
        SNAPSHOT.war, webapp_target_name: demo1 }
4   |
```

```
1   - hosts: demo2-application-server
2     roles:
3       - {role: deploy-on-tomcat, webapp_source_path: ./demo2-1.0-
        SNAPSHOT.war, webapp_target_name: demo2 }
4
5   |
```

Ansible Infrastruktur

Revisions Control System

Cron Job pulls changes from SCM and
calls ansible-playbook

CI Server

"push the button"

ensure that server is configured
as it is defined in playbook

Jenkins job calls
ansible playbooks
to deploy product.

Target Nodes

- SSH Login available
- no roots available

- SSH Login available
- no roots available

Legend

Configuration Use Case

Deployment Use Case

Ansible Tower

Wie werden Ansible Skripte getestet?

- `--check`

- `--syntax-check`

- Jenkins + Vagrant

- Rspec tests

ServerSpec Tests

```
1  require 'spec_helper'
2
3  describe package('mysql-server') do
4    it { should be_installed }
5  end
6
7  describe service('mysql') do
8    it { should be_enabled  }
9    it { should be_running  }
10 end
11
12 describe 'MySQL config parameters' do
13   context mysql_config('bind-address') do
14     its(:value) { should eq '0.0.0.0' }
15   end
16 end
17
```

```
1  require 'spec_helper'
2
3  describe package('openjdk-7-jdk') do
4    it { should be_installed }
5  end
6
7  describe command('ls /etc/init.d/tomcat') do
8    its(:exit_status) { should eq 0 }
9  end
10
11 describe command('ls /opt/tomcat') do
12   its(:exit_status) { should eq 0 }
13 end
14
```

ServerSpec Tests

Wie unterscheidet sich Ansible zu seiner Konkurrenz?

Vergleich

Ansible

- 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. Distrobutions-spezifisch

Puppet

- Client-Server Architektur

- Für komfortables Arbeiten benötigt es Rootrechte

- Konfigurationsmgmt

- Monitoringtools Open Source

- Skripte mehr deklarativ

- Windows-Support

- Skripte können OS-unspezifisch sein

Vergleich

Puppet

```
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'],

  }
```
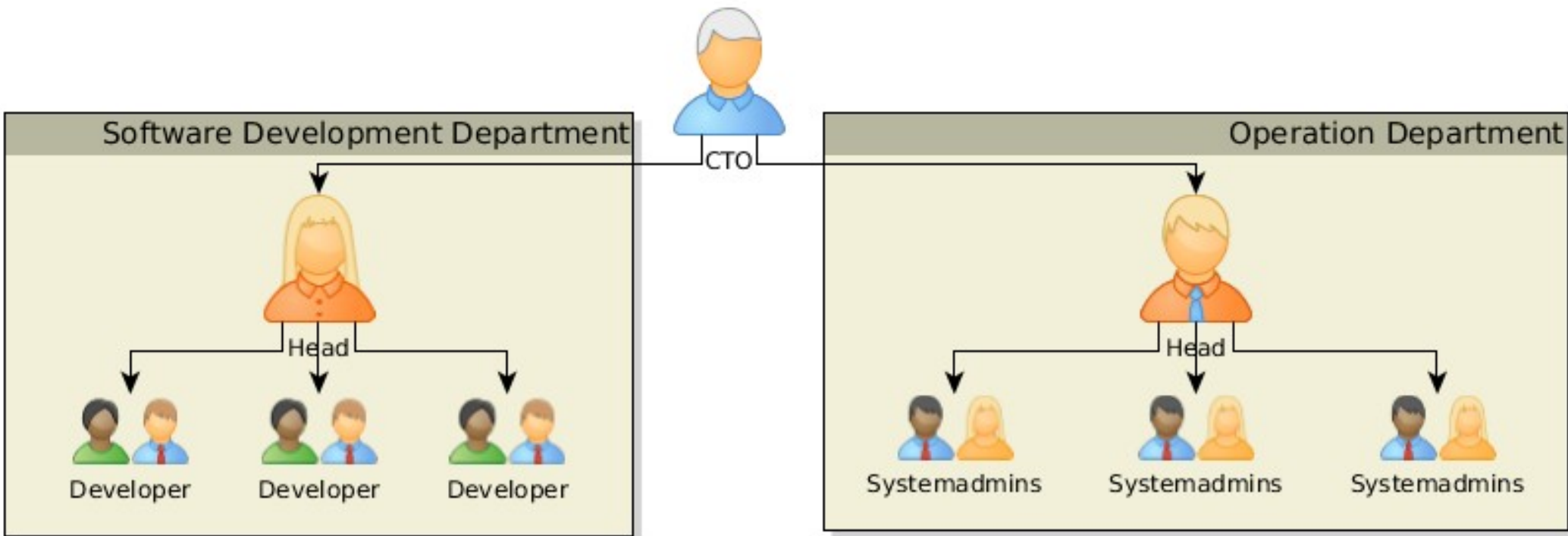
Ansible

```
- name: add nodejs ppa
  apt_repository: repo='ppa:chris-lea/node.js'

- name: install nodejs package
  apt: name=nodejs update-cache=yes
```

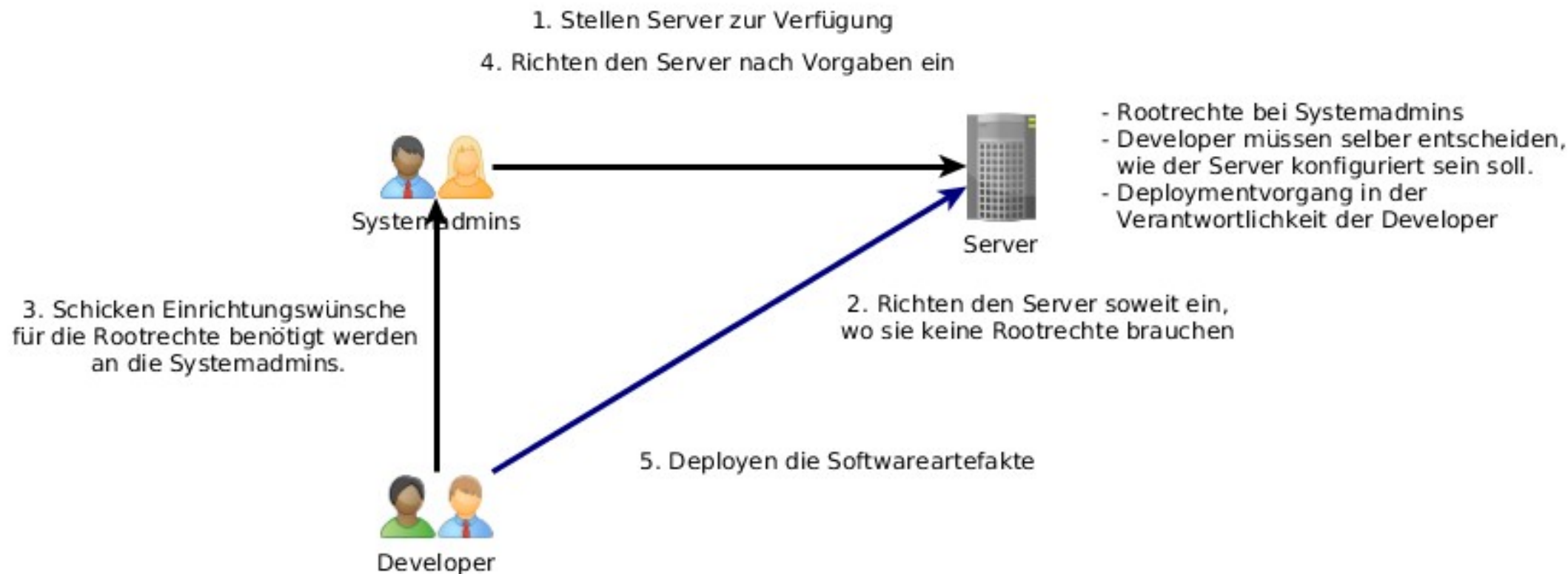# Weitere Einsatzszenarien aus Entwicklersicht

Systemkonfiguration für Entwickler
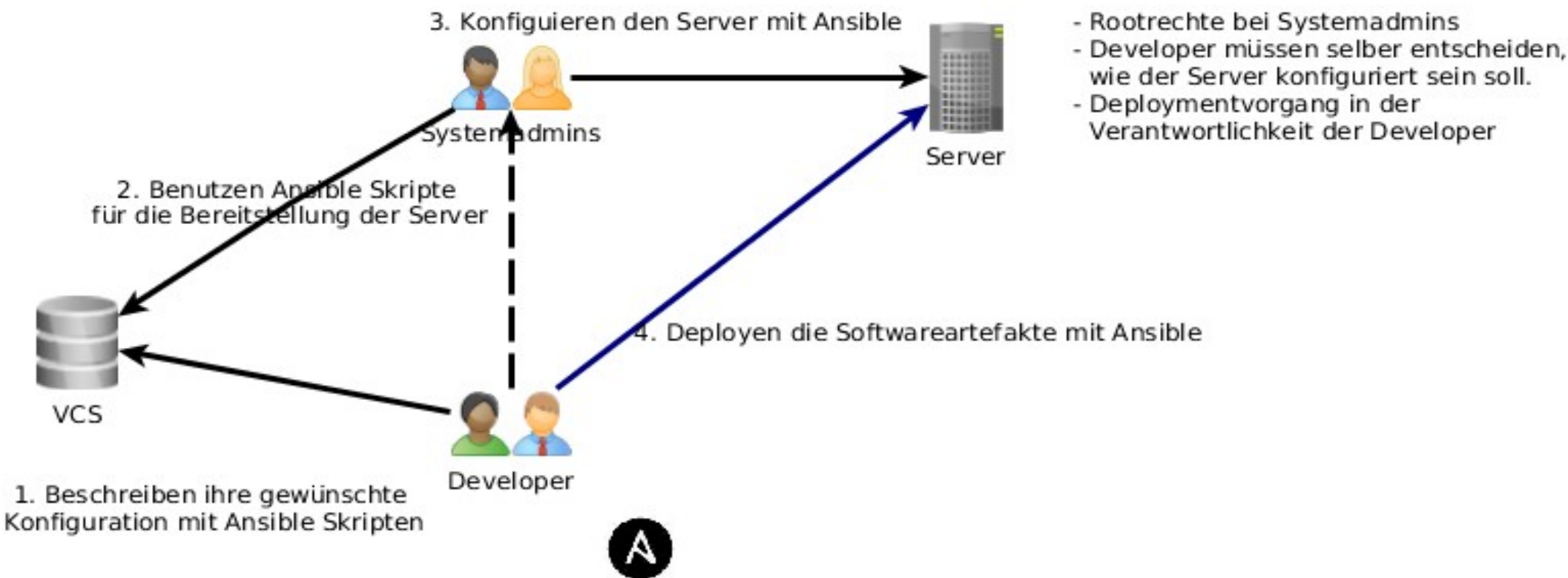
Organisatorische Ausgangslage
Realität

Systemkonfiguration für Entwickler
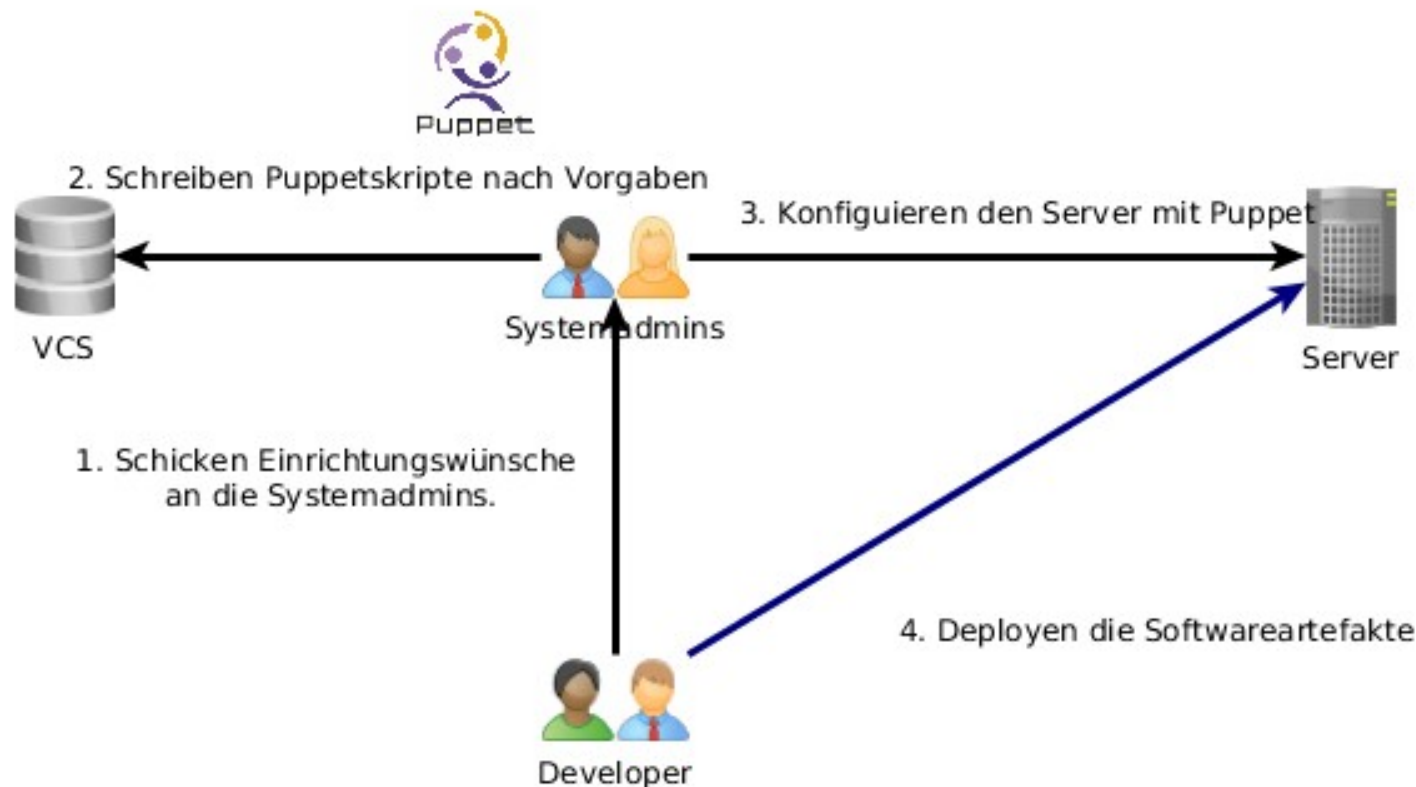
Prozess zwischen Development und Operation

Systemkonfiguration für Entwickler

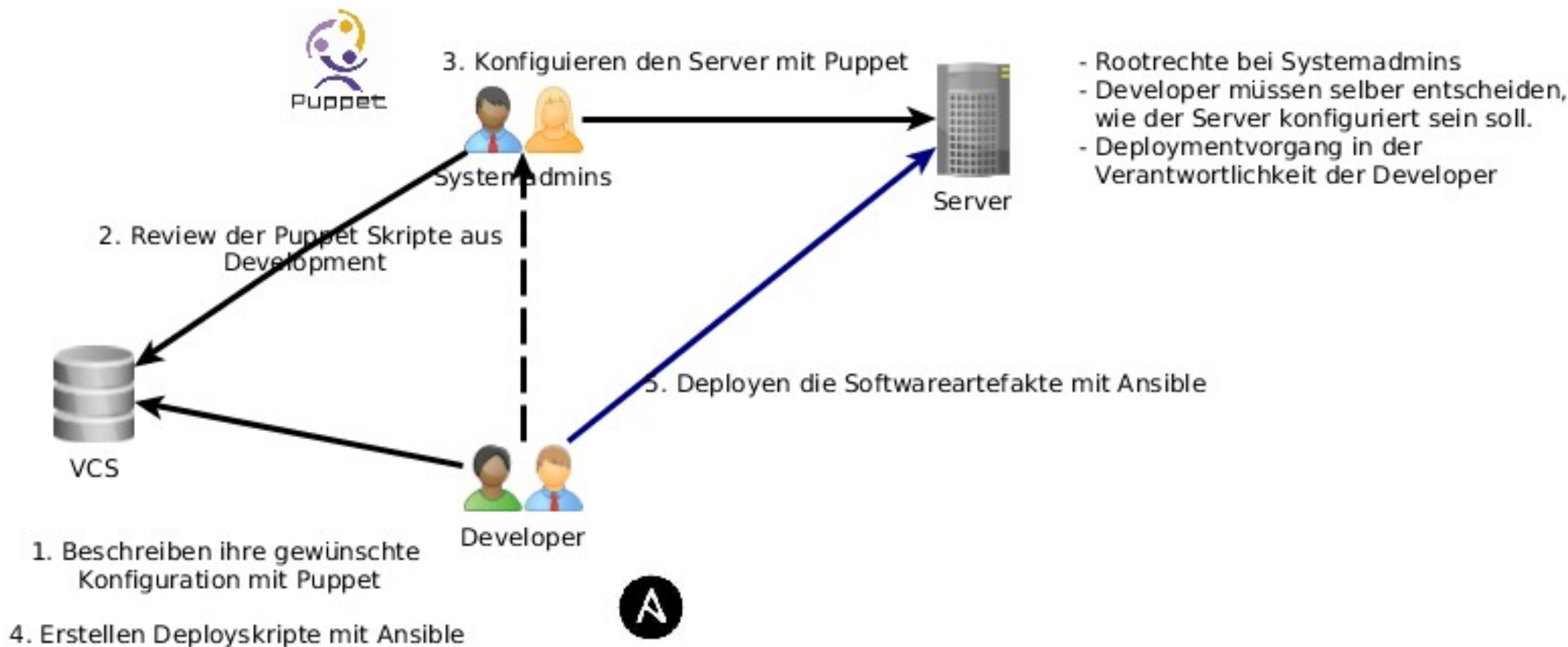## Lösungidee mit Ansible

Systemkonfiguration für Entwickler

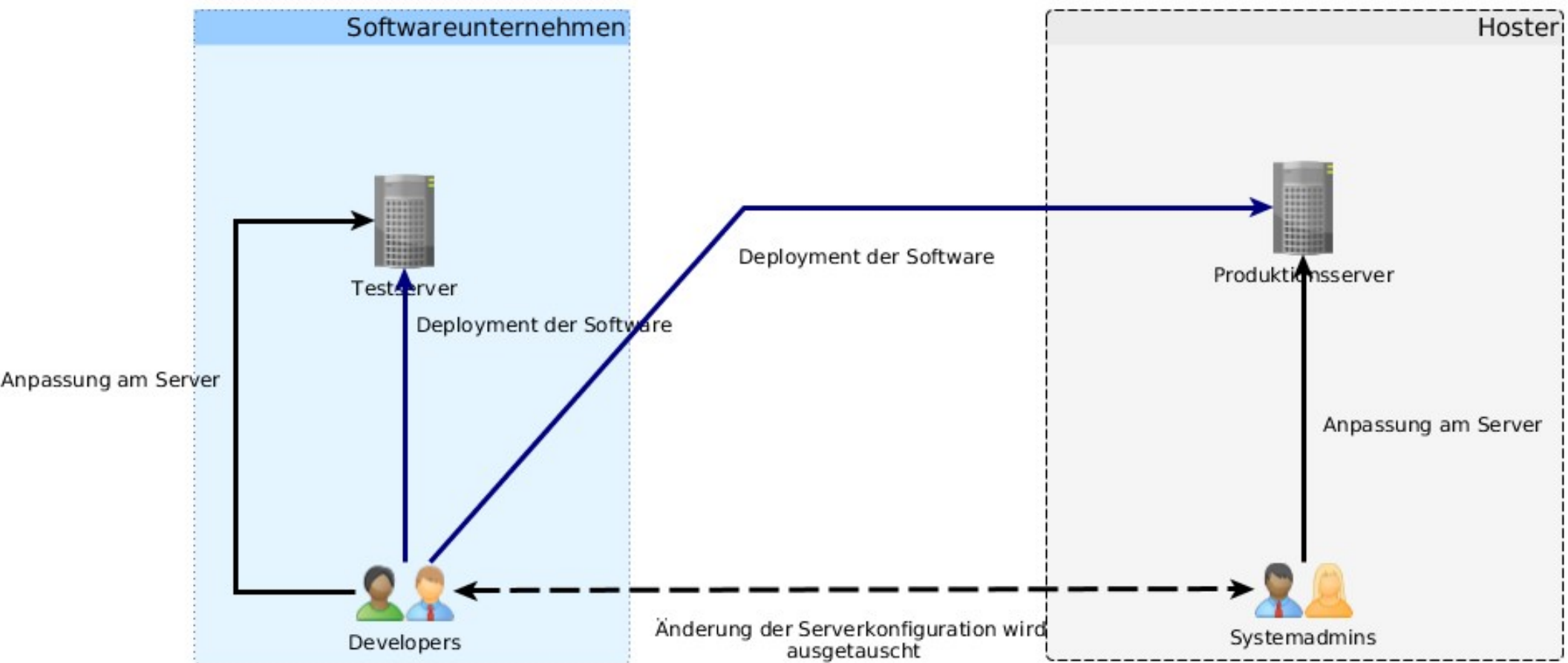## Variante - Prozess zwischen Development und Operation
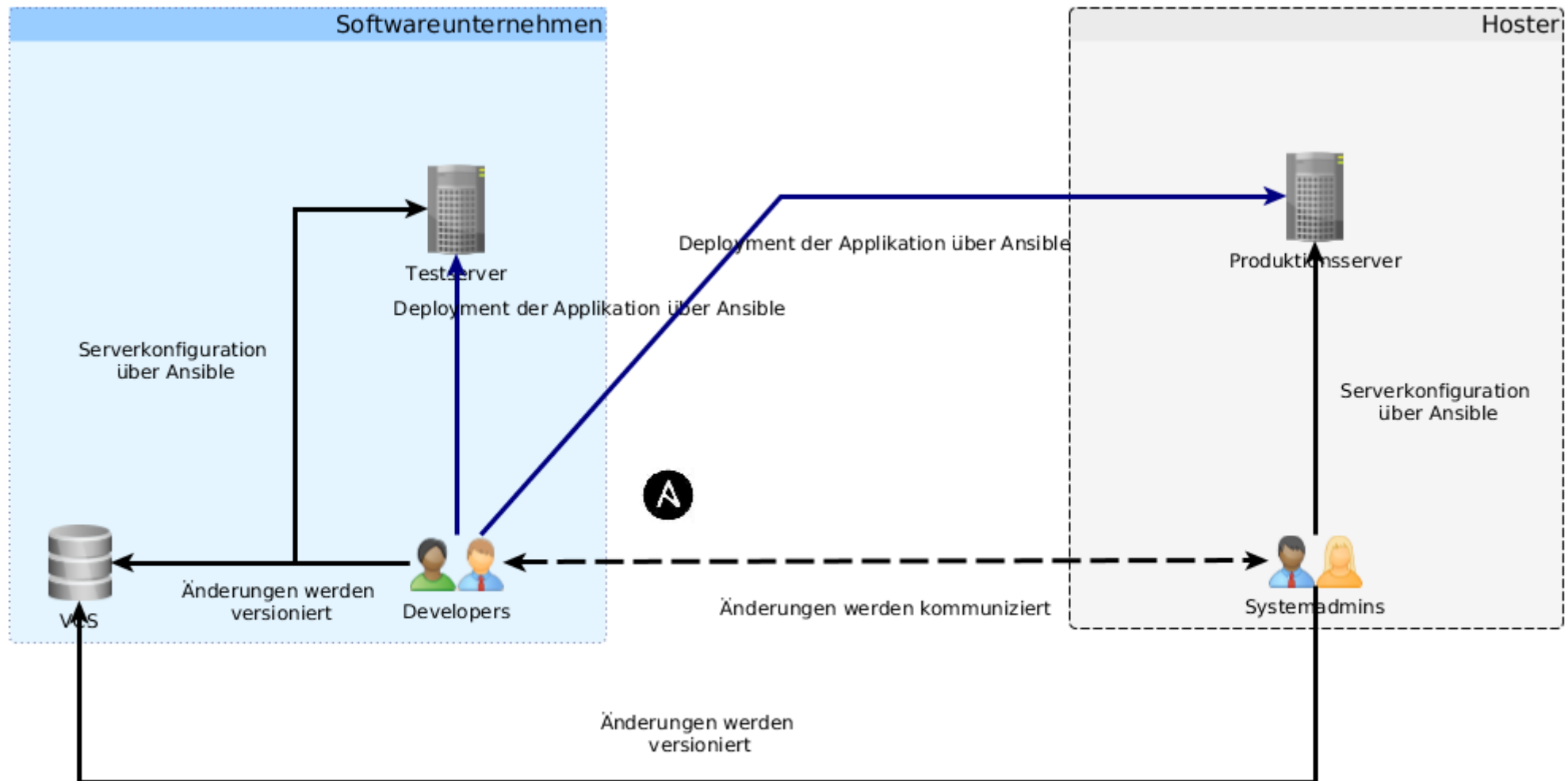
Systemkonfiguration für Entwickler

Lösungsvariante

Systemkonfiguration für Entwickler

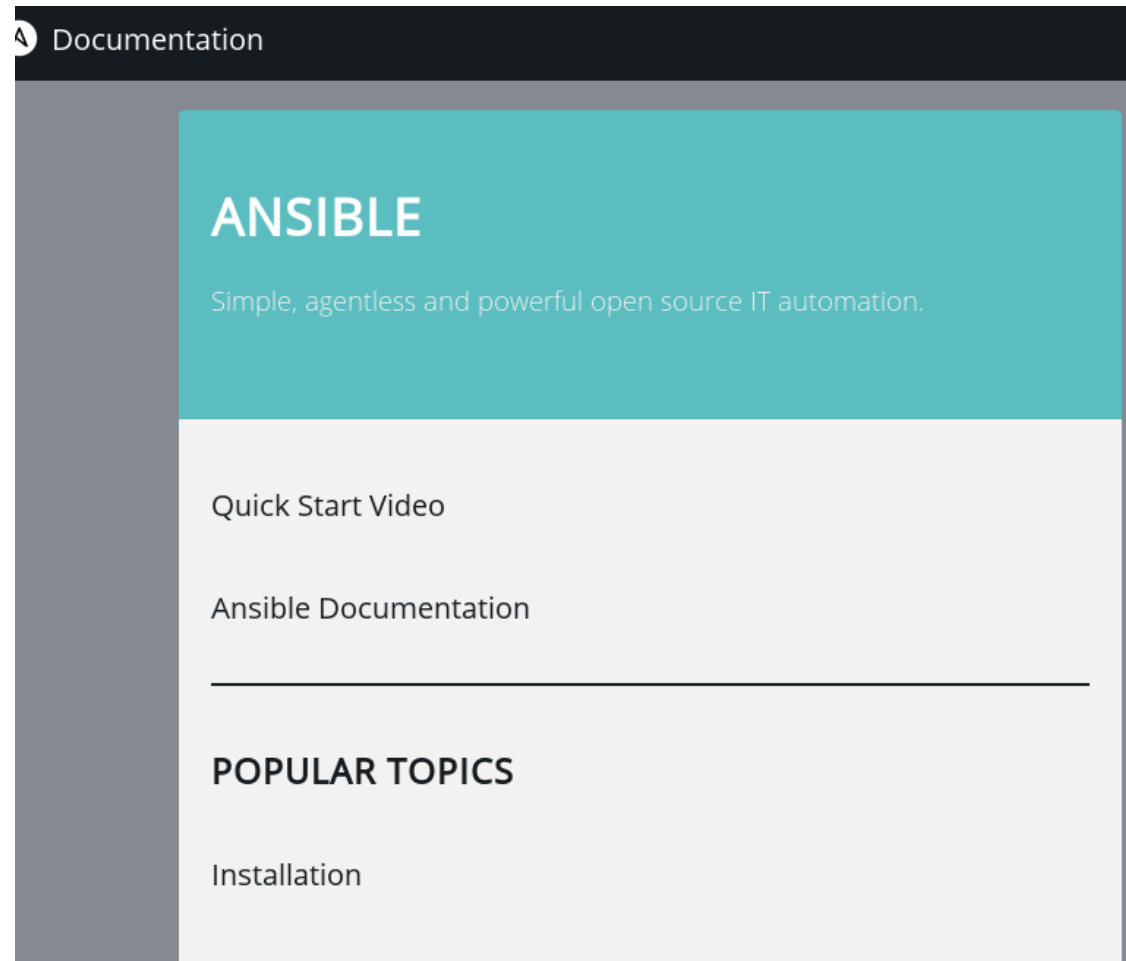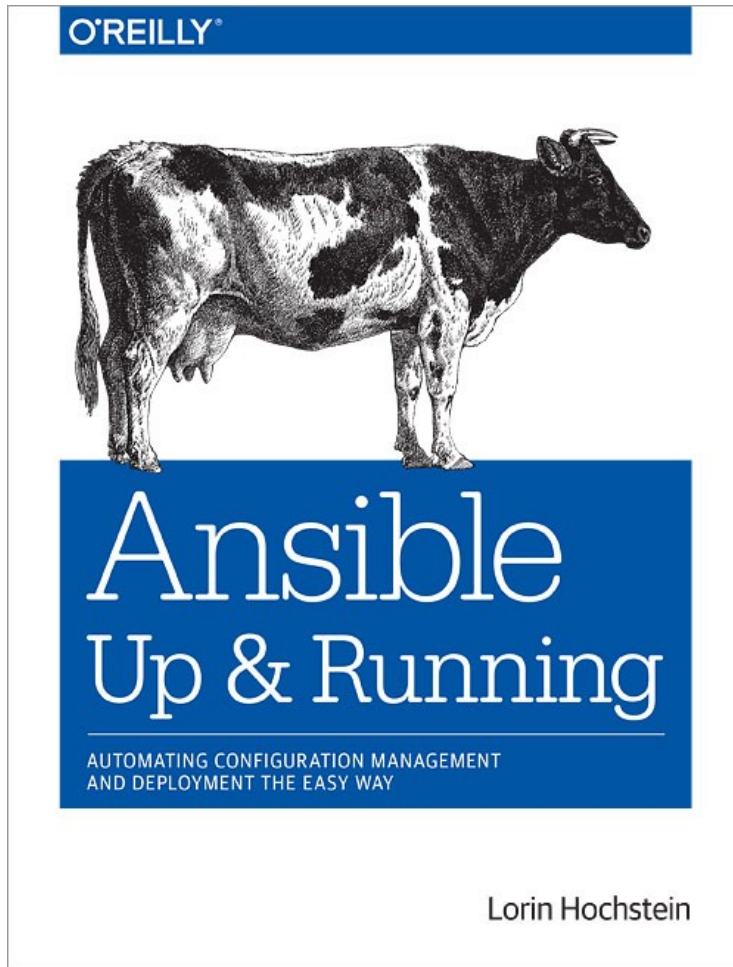## Produktionsserver sind beim externen Hoster

Systemkonfiguration für Entwickler

Lösungsidee

Softwareunternehmen

Hoster

Testserver

Produktionsserver

Deployment der Applikation über Ansible

Deployment der Applikation über Ansible

Serverkonfiguration über Ansible

Serverkonfiguration über Ansible

Änderungen werden versioniert

Developers

Änderungen werden kommuniziert

Systemadmins

VCS

Änderungen werden versioniert

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

Weitere Informationen

http://docs.ansible.com/

Fragen?

@SandraParsick
info@sandra-parsick.de