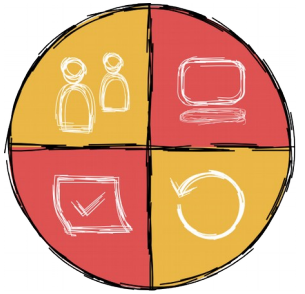
DevOps Frankfurt, 17/11/2016

# Ansible für Devs – Konfigurationsmanagment nicht nur für Ops

Sandra Parsick
mail@sandra-parsick.de
@SandraParsick

# Zur meiner Person

- Sandra Parsick
- Freiberuflicher Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
  - Java Enterprise Anwendungen
  - Agile Methoden
  - Software Craftmanship
  - Automatisierung von Entwicklungsprozessen
- Softwerkskammer Ruhrgebiet
- Twitter: @SandraParsick
- Xing: https://www.xing.com/profile/SandraParsick
- Blog: http://blog.sandra-parsick.de
- E-Mail: mail@sandra-parsick.de



# 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

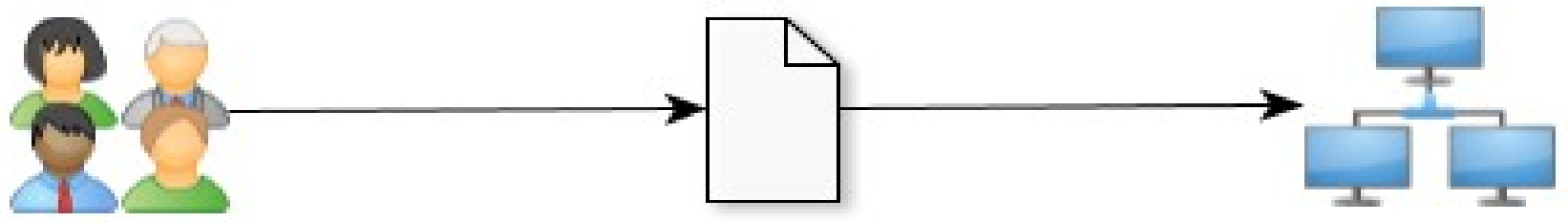
# Konfigurationsmanagement (KM)

*„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“



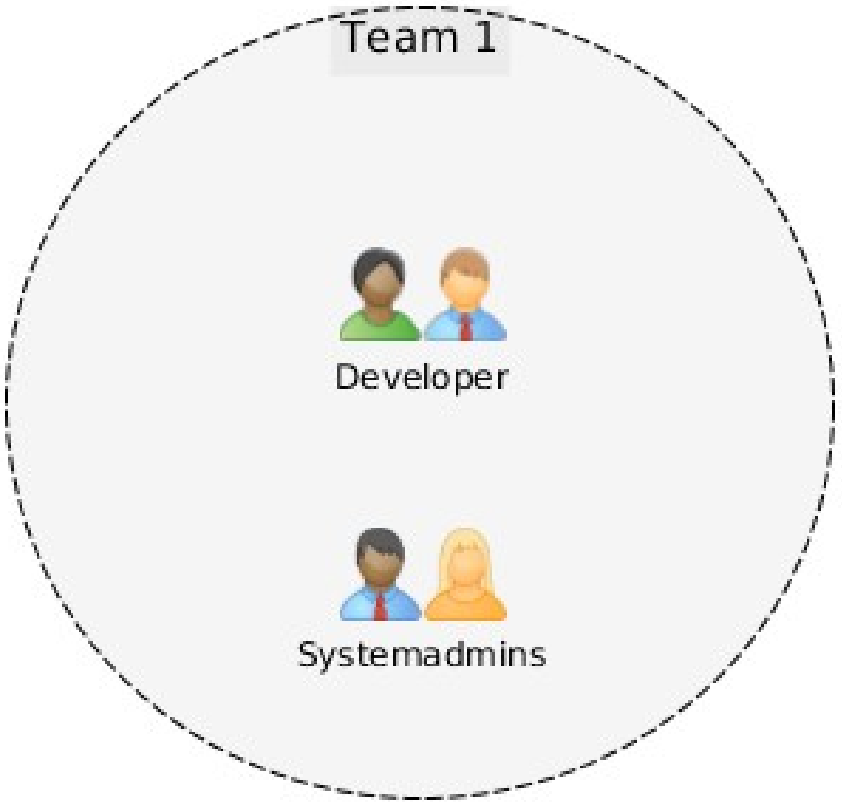


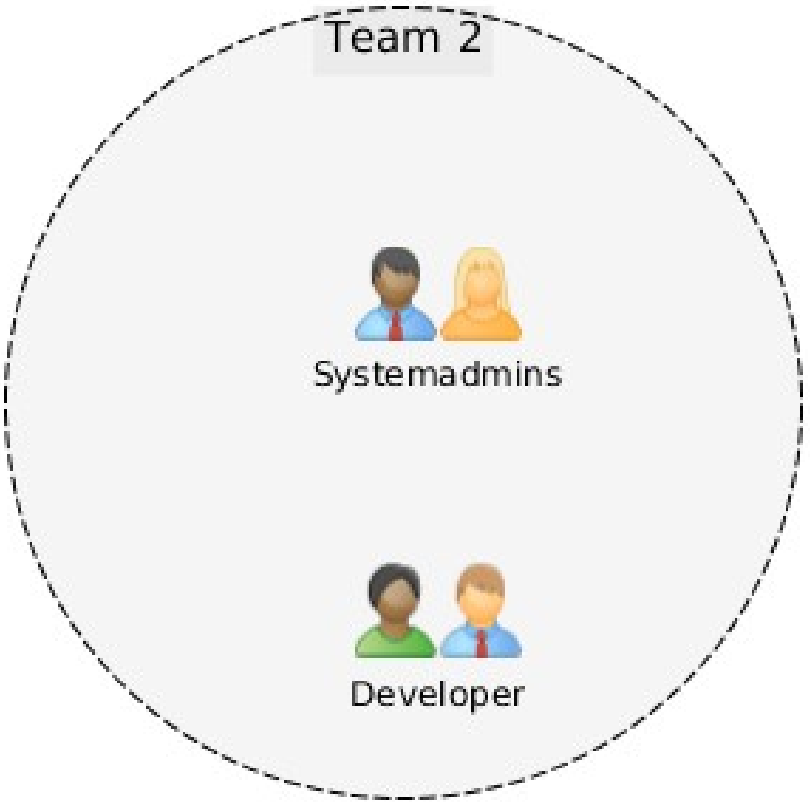
Ansible

# Warum ist es für Entwickler interessant?

# Systemkonfiguration für Entwickler

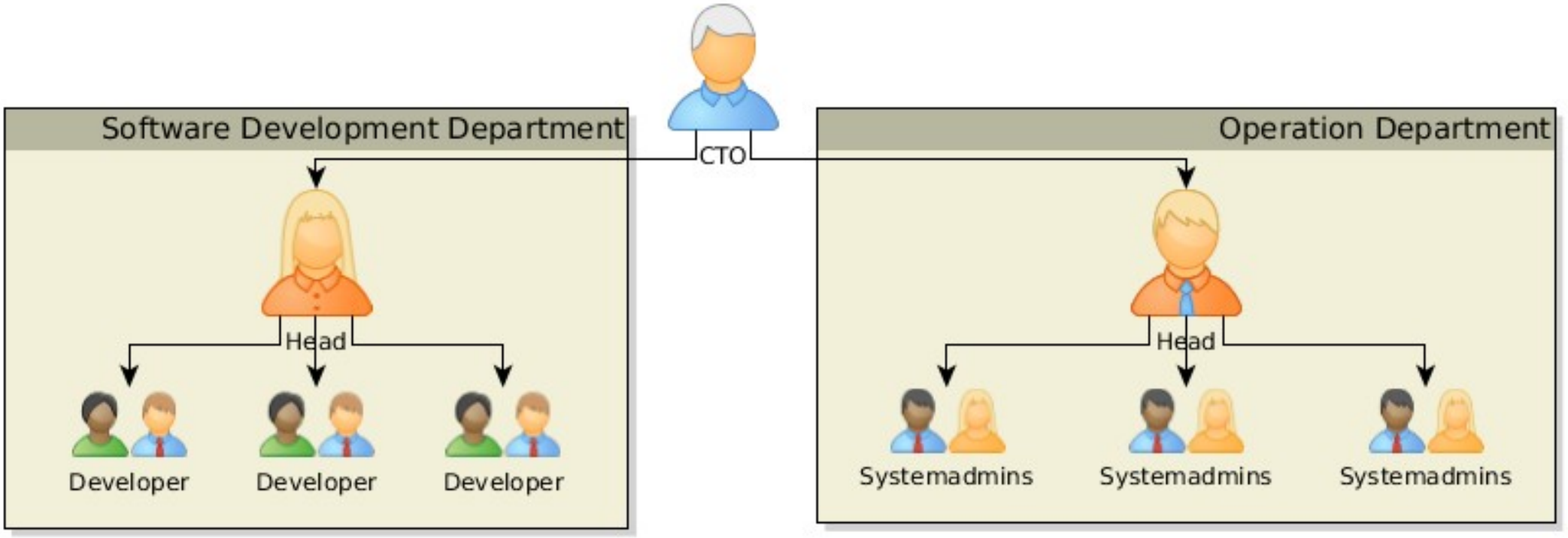
## Organisatorische Ausgangslage
## Wunsch



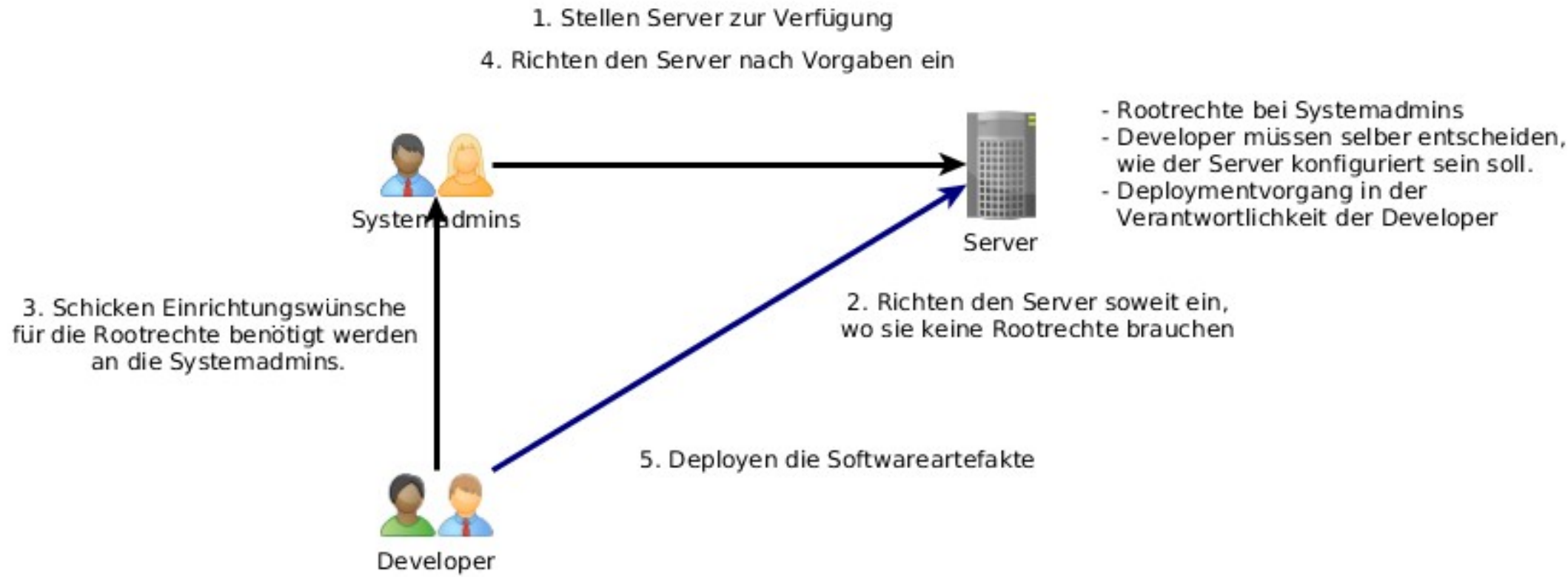
# Systemkonfiguration für Entwickler

## Organisatorische Ausgangslage
## Realität



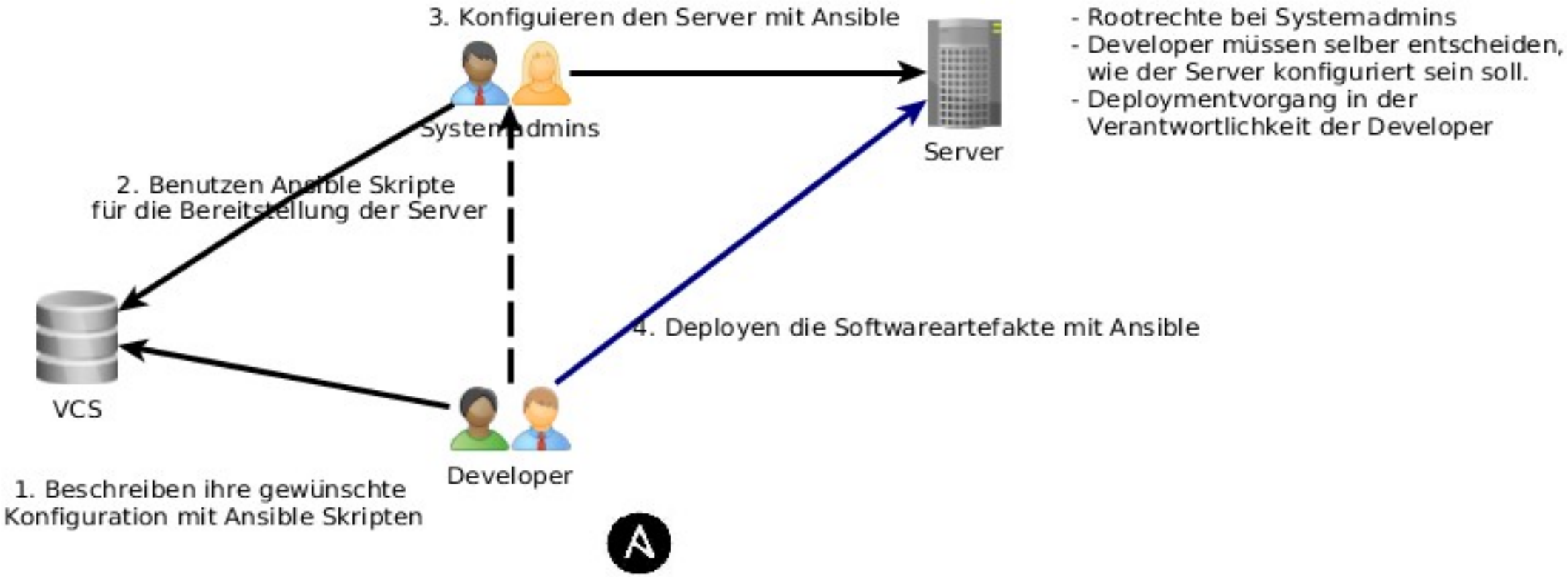
# Systemkonfiguration für Entwickler

## Prozess zwischen Development und Operation



# Systemkonfiguration für Entwickler

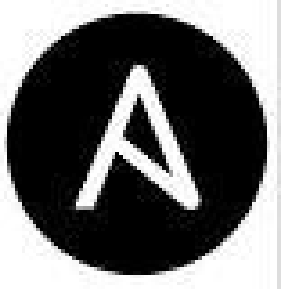
## Lösungidee mit Ansible



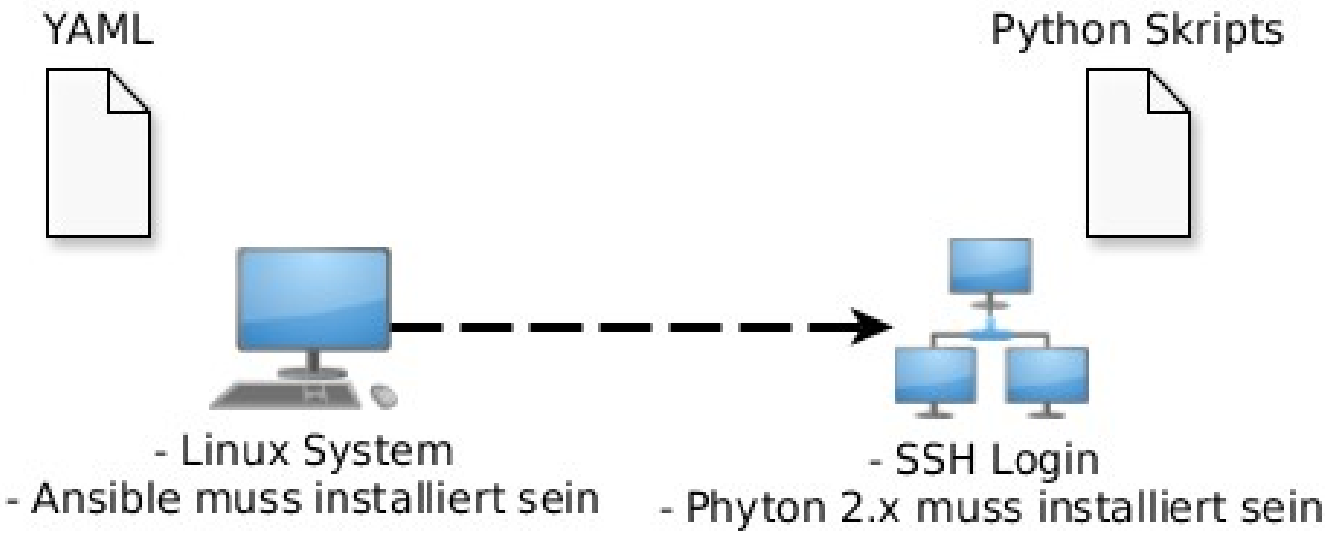
# Einführung in Ansible

# Ansible

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



# Funktionsweise



# Exkurs: YAML

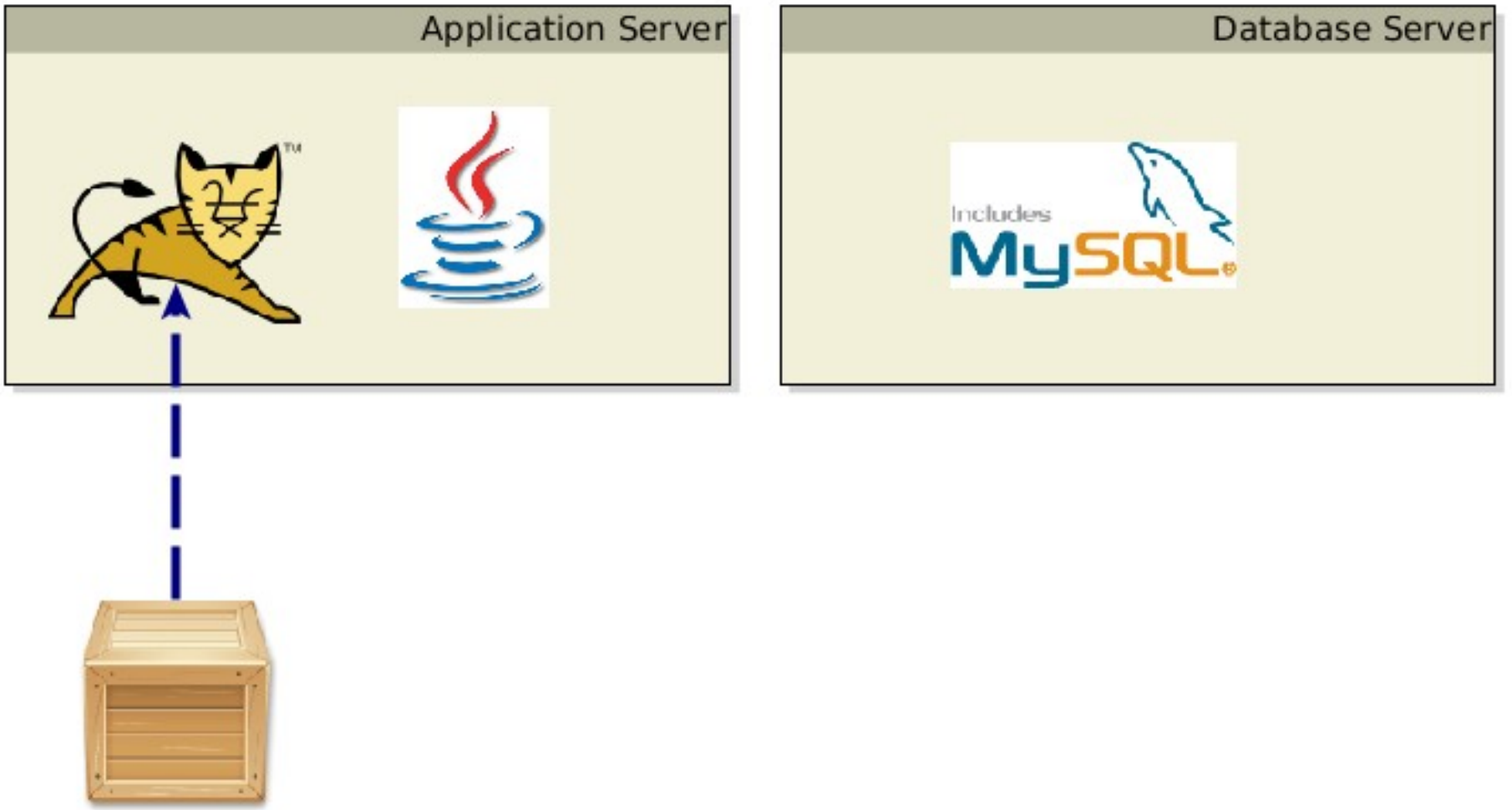
**YAML**

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

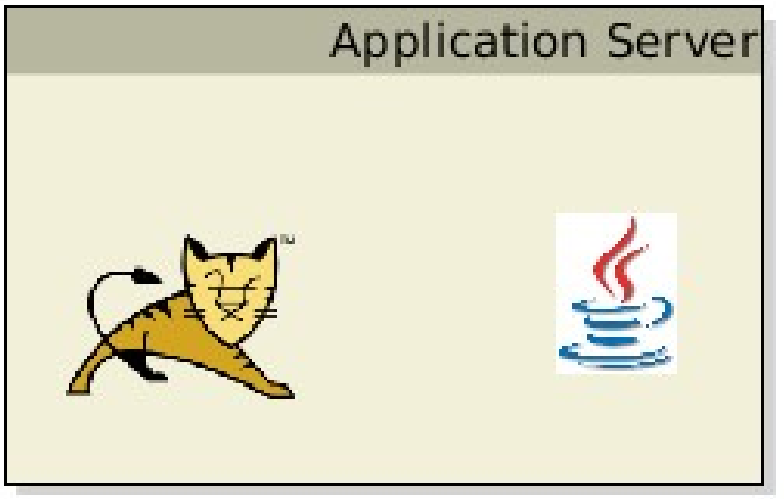
**JSON**

```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: 8.5.8
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      apt: name=openjdk-8-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

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

    - name: ensure tomcat scripts are executable
      file: name={{item.path}} mode=755
```

# Inventories

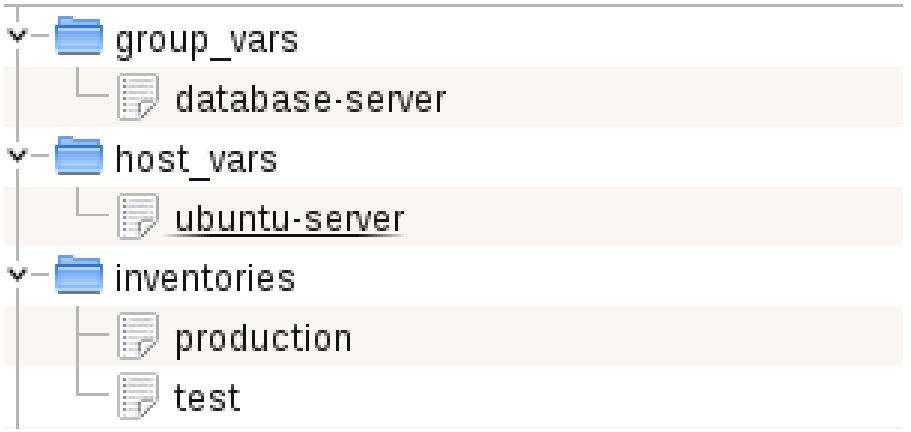
## Production

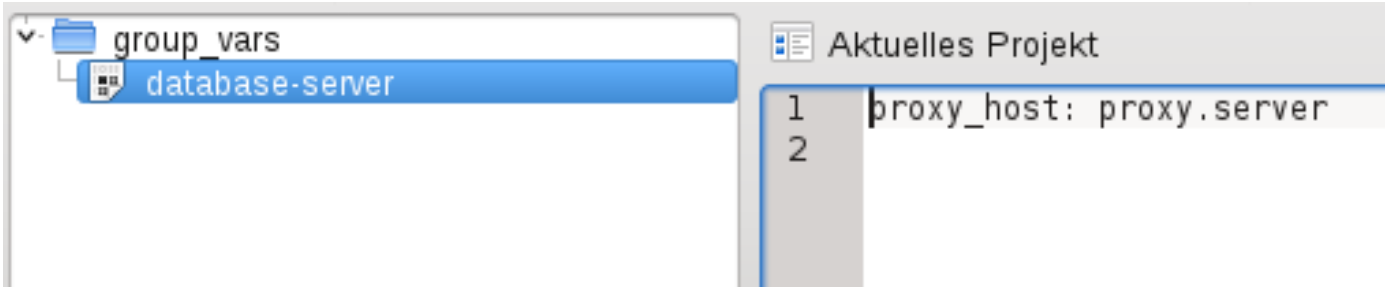
```
[application-server]
192.168.33.10
ubuntu-server db_host=mysql01

[mysql-db-server]
mysql[01:10]

[oracle-db-server]
db-[a:f].oracle.company.com

[database-server:children]
mysql-db-server
oracle-db-server

[application-server:vars]
message="Welcome"

[database-server:vars]
message="Hello World!"
```

## Test

```
[application-server]
192.168.33.10

[database-server]
192.168.33.10

```

# Inventories



```
    apt: name=openjdk-8-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

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

  - name: ensure tomcat scripts are executable
    file: name={{item.path}} mode=755
    with_items: '{{ result.files }}'

  - name: install init.d script for tomcat
    copy: src=roles/tomcat8/files/init.d/tomcat dest=/etc/init.d/tomcat owner=vagrant
    group=vagrant mode=755
    become: yes
    become_method: sudo
```

# Templates

- setenv.sh.j2

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

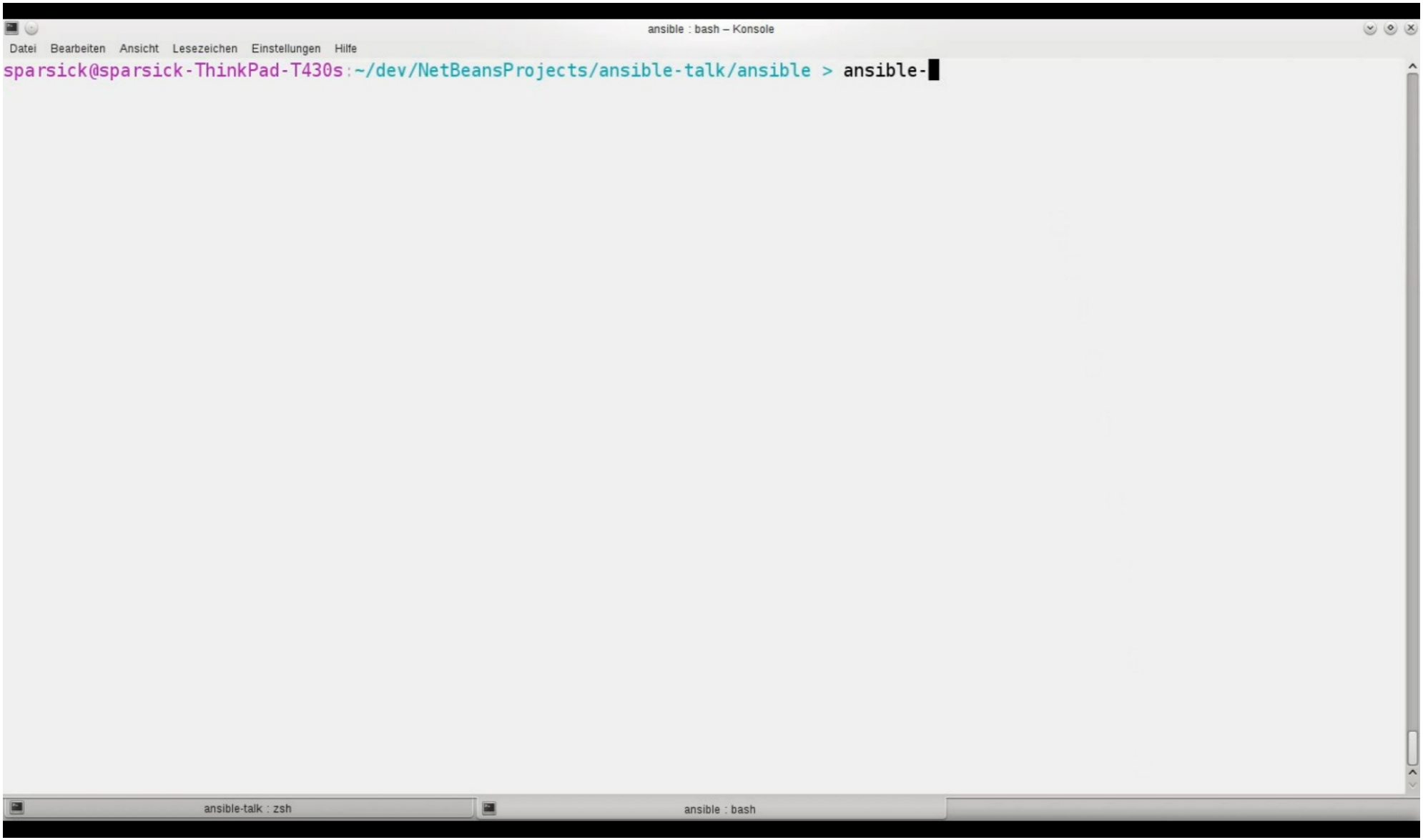
# Templates - Jinja2

- Templating engine für Python

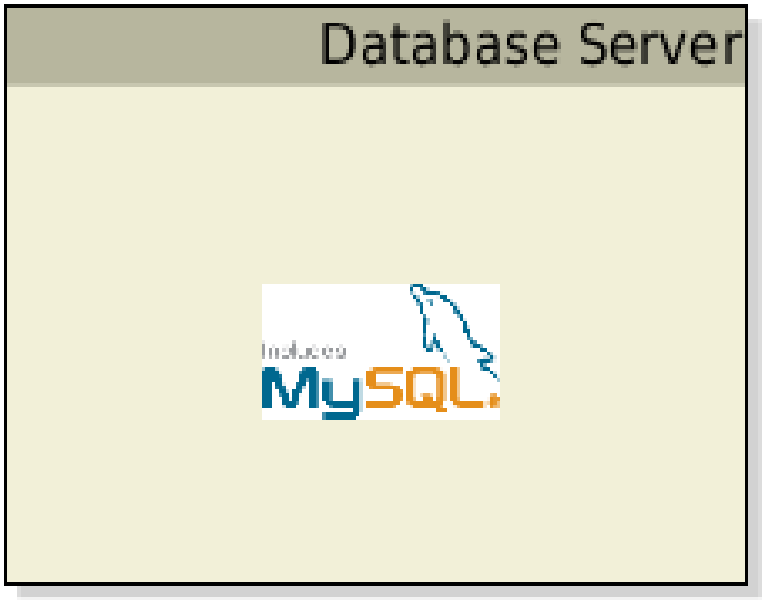
```
<title>{% block title %}{% endblock %}</title>
<ul>
{% for user in users %}
  <li><a href="{{ user.url }}">{{ user.username }}</a></li>
{% endfor %}
</ul>
```

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

# Setup Application Server Playbook

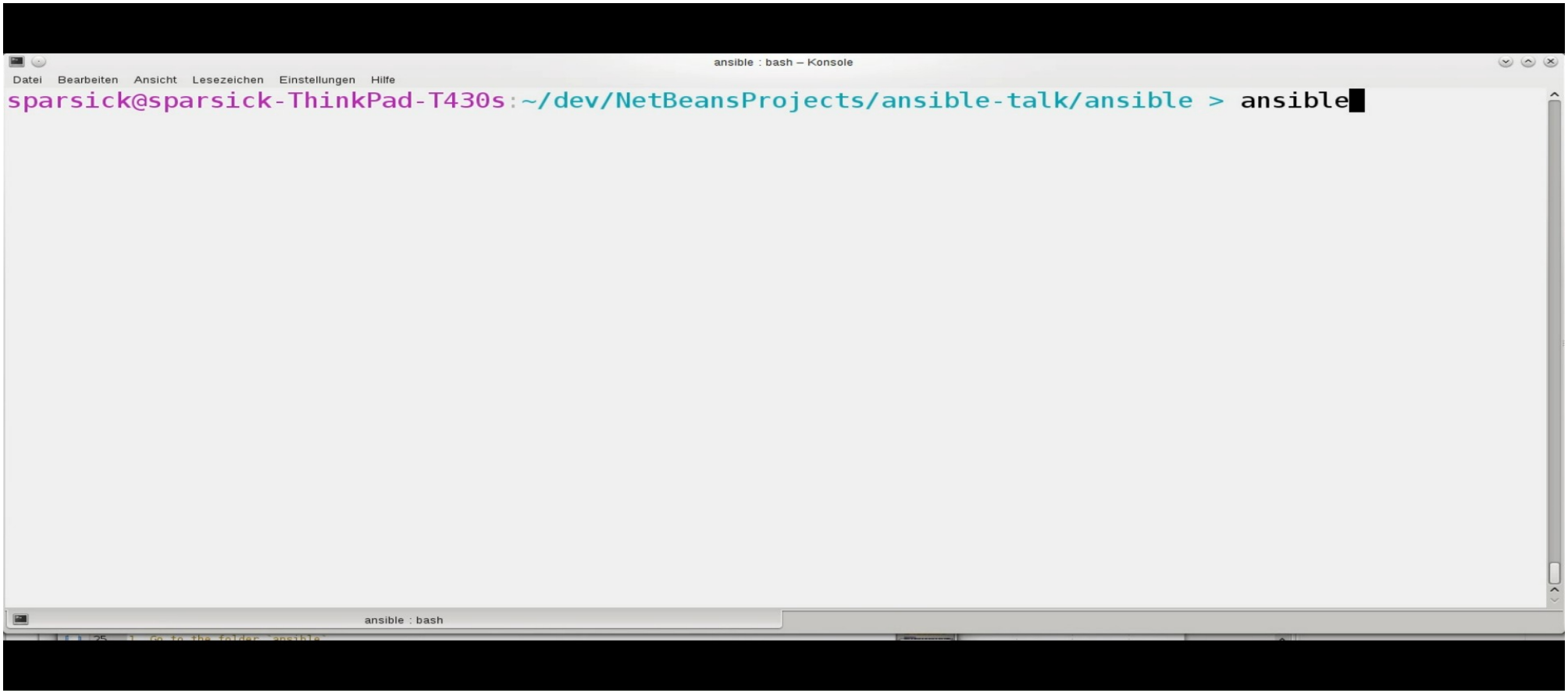


# Setup Database Server Playbook

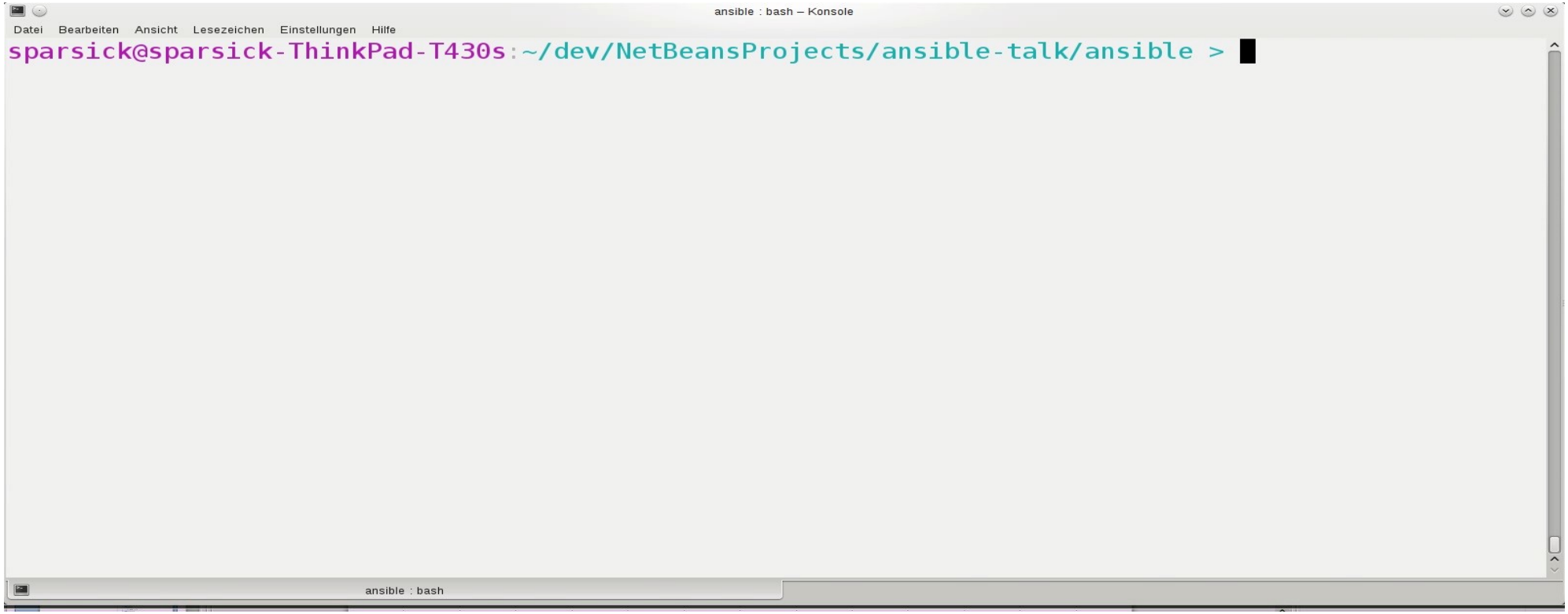


```
1  - hosts: database-server
2    become: yes
3    become_method: sudo
4
5    tasks:
6      - name: install mysql db
7        apt: name=mysql-server state=present
8
9      - name: installs python-mysqldb
10       apt: name=python-mysqldb state=present
11
12     - name: start mysql
13       service: name=mysql state=started
14
15     - name: set bind address
16       lineinfile: dest=/etc/mysql/mysql.conf.d/mysqld.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=g3h31m priv=*.*:ALL,GRANT state=present host=%
24
25   handlers:
26      - name: restart mysql
27        service: name=mysql state=restarted
```

# Setup Database Server Playbook



# Setup Database Server Playbook



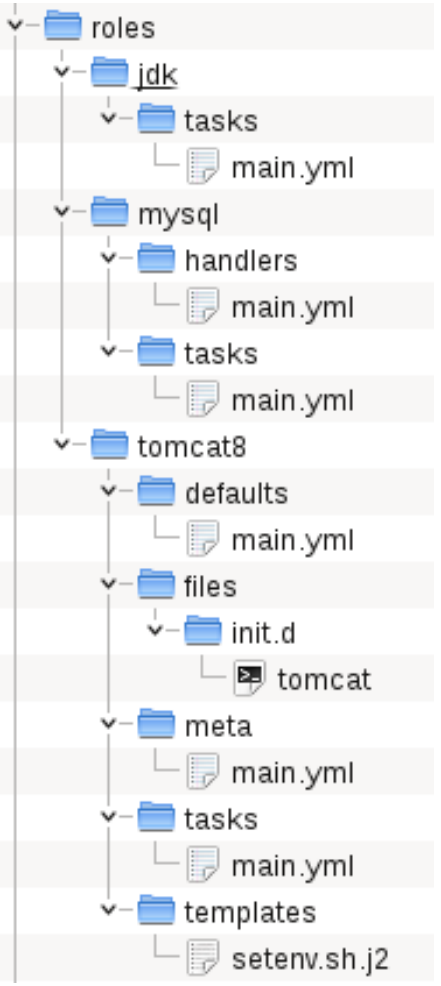
```yaml
- hosts: application-server
  vars:
    tomcat_version: 8.5.8
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      apt: name=openjdk-8-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

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

    - name: ensure tomcat scripts are executable
      file: name={{item.path}} mode=755
```

```
    apt: name=openjdk-8-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

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

  - name: ensure tomcat scripts are executable
    file: name={{item.path}} mode=755
    with_items: '{{ result.files }}'

  - name: install init.d script for tomcat
    copy: src=roles/tomcat8/files/init.d/tomcat dest=/etc/init.d/tomcat owner=vagrant
    group=vagrant mode=755
    become: yes
    become_method: sudo
```

```
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/mysql.conf.d/mysqld.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=g3h31m priv=*.*:ALL,GRANT state=present host=%
24
25   handlers:
26      - name: restart mysql
27        service: name=mysql state=restarted
```

# Roles

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



# Setup Playbooks mit Roles

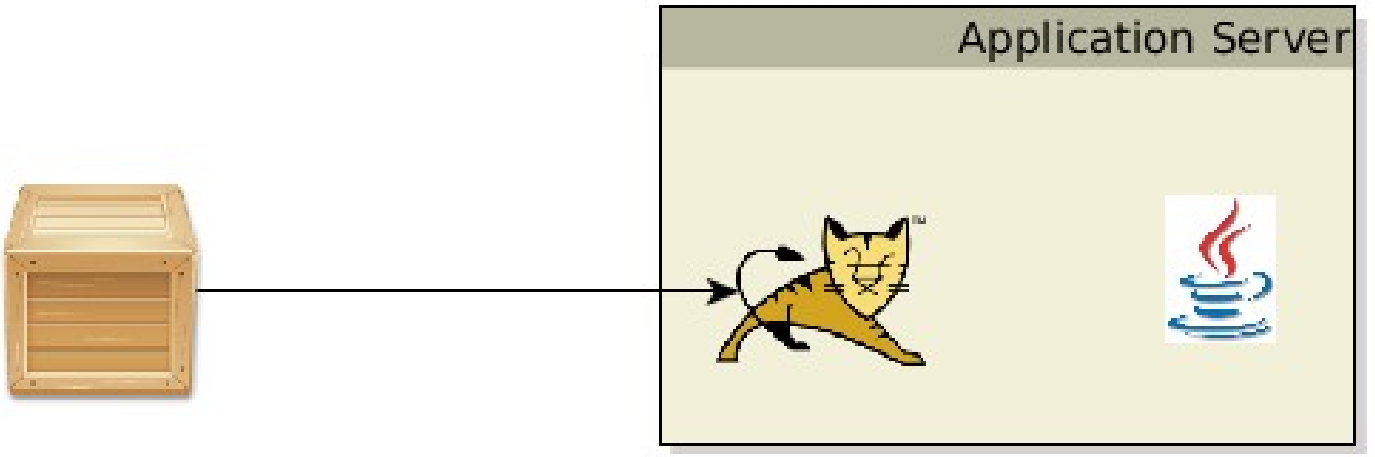
- Setup Application Server

```
1  - hosts: application-server
2    roles:
3      - jdk
4      - { role: tomcat8, tomcat_version: 8.5.8 }
5
```

- Setup Database Server

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

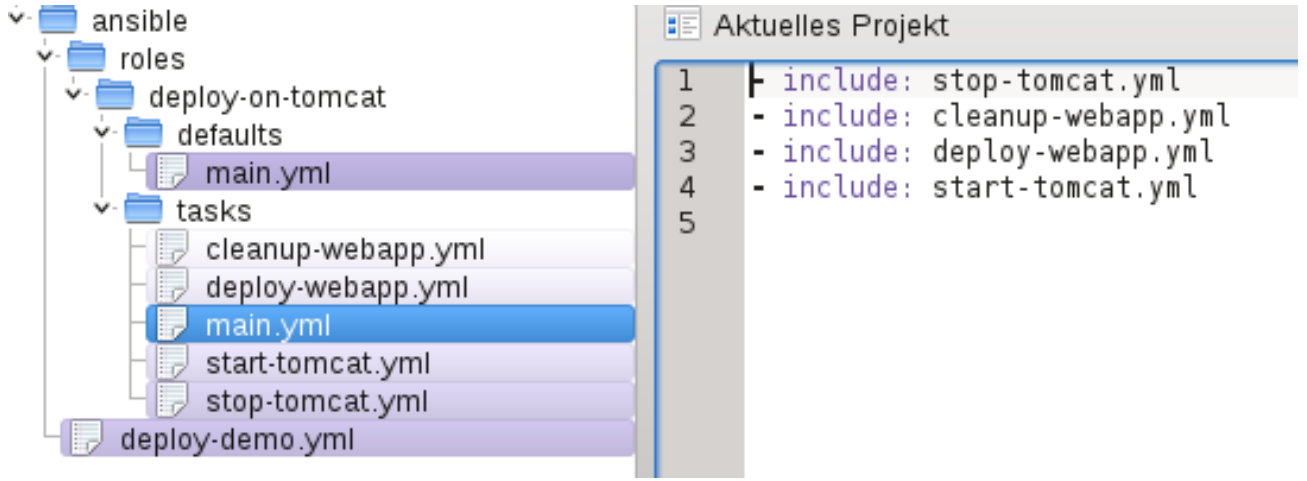
# Java Webapplikation Deployment



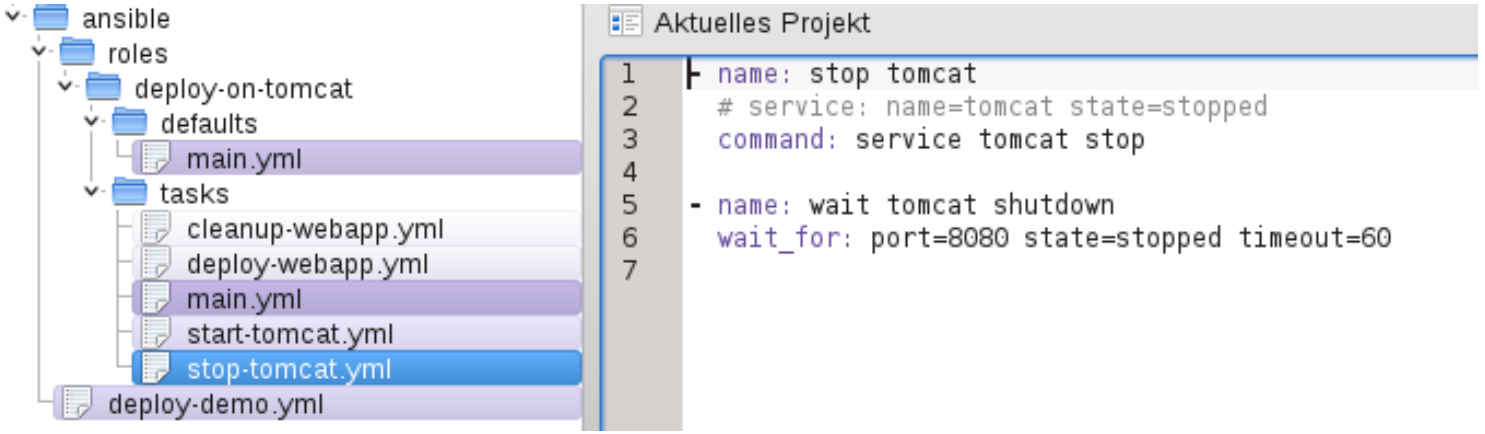
# Deploy Application Playbook

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

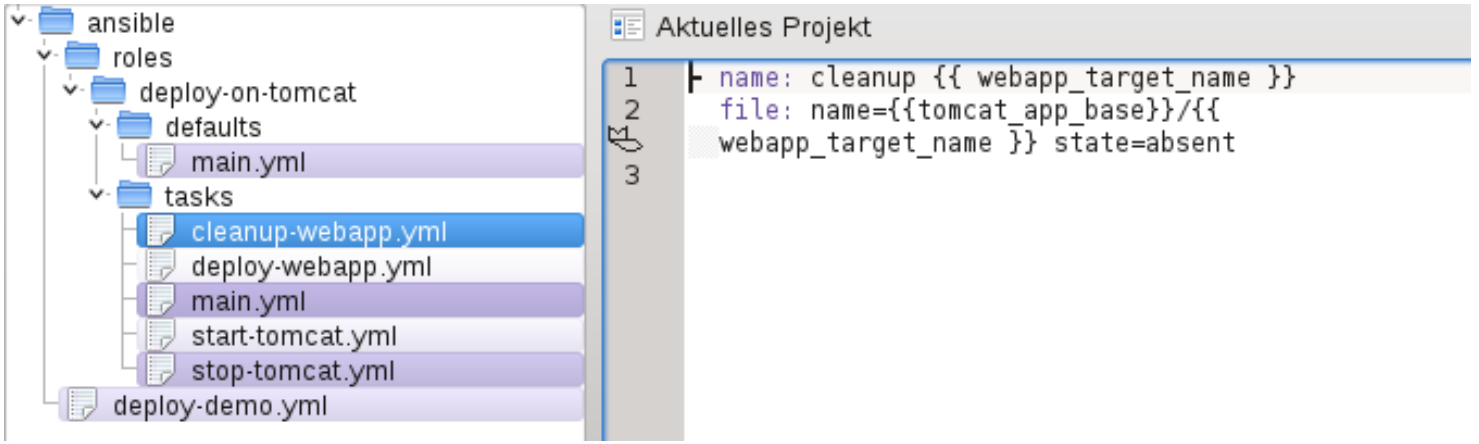
# deploy-on-tomcat Role



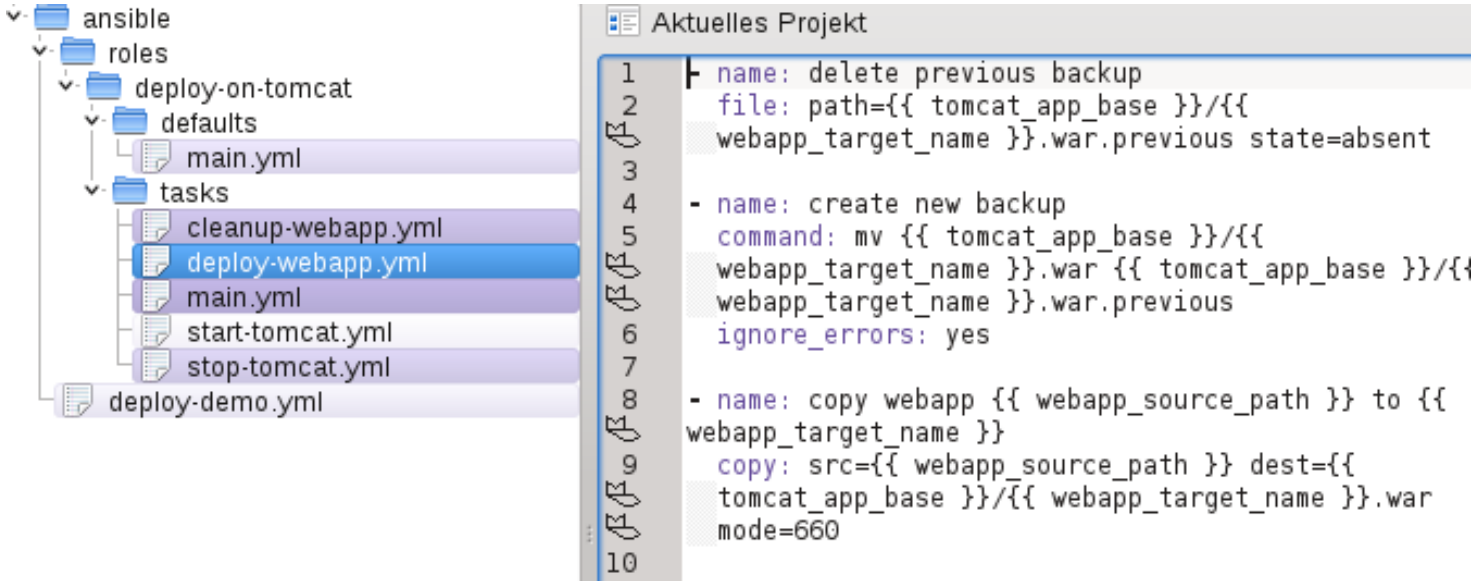
# deploy-on-tomcat Role



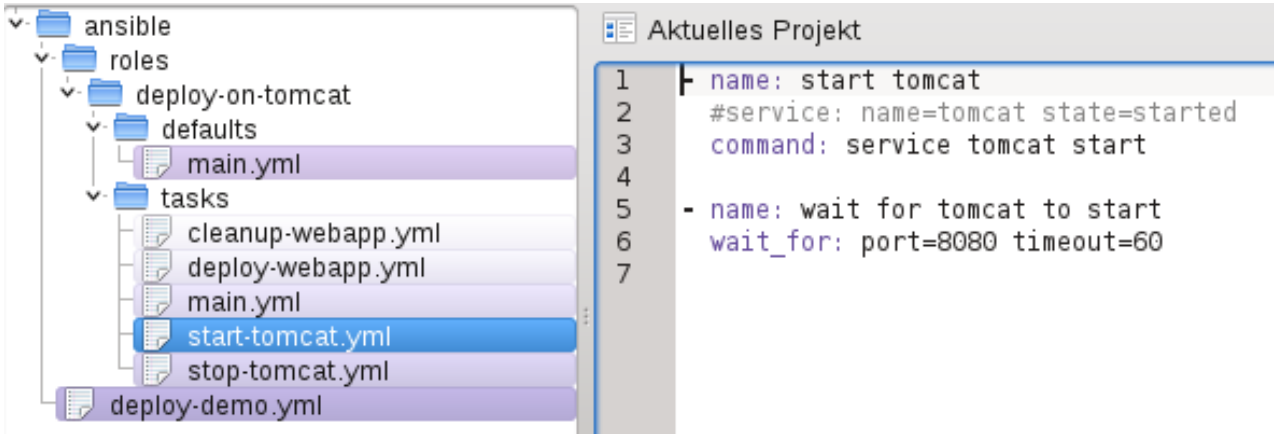
# deploy-on-tomcat Role



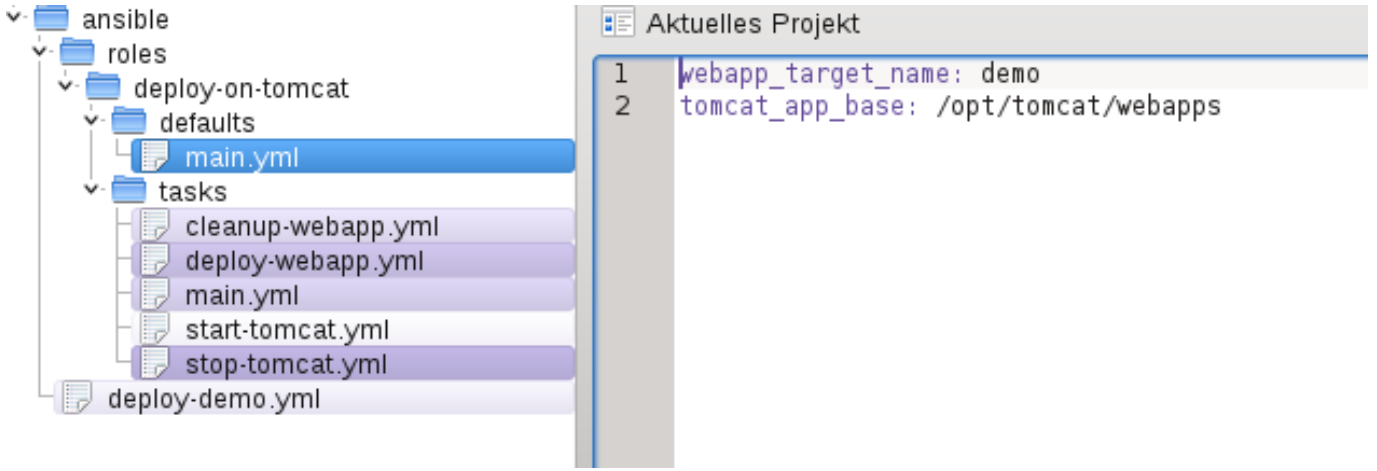
# deploy-on-tomcat Role



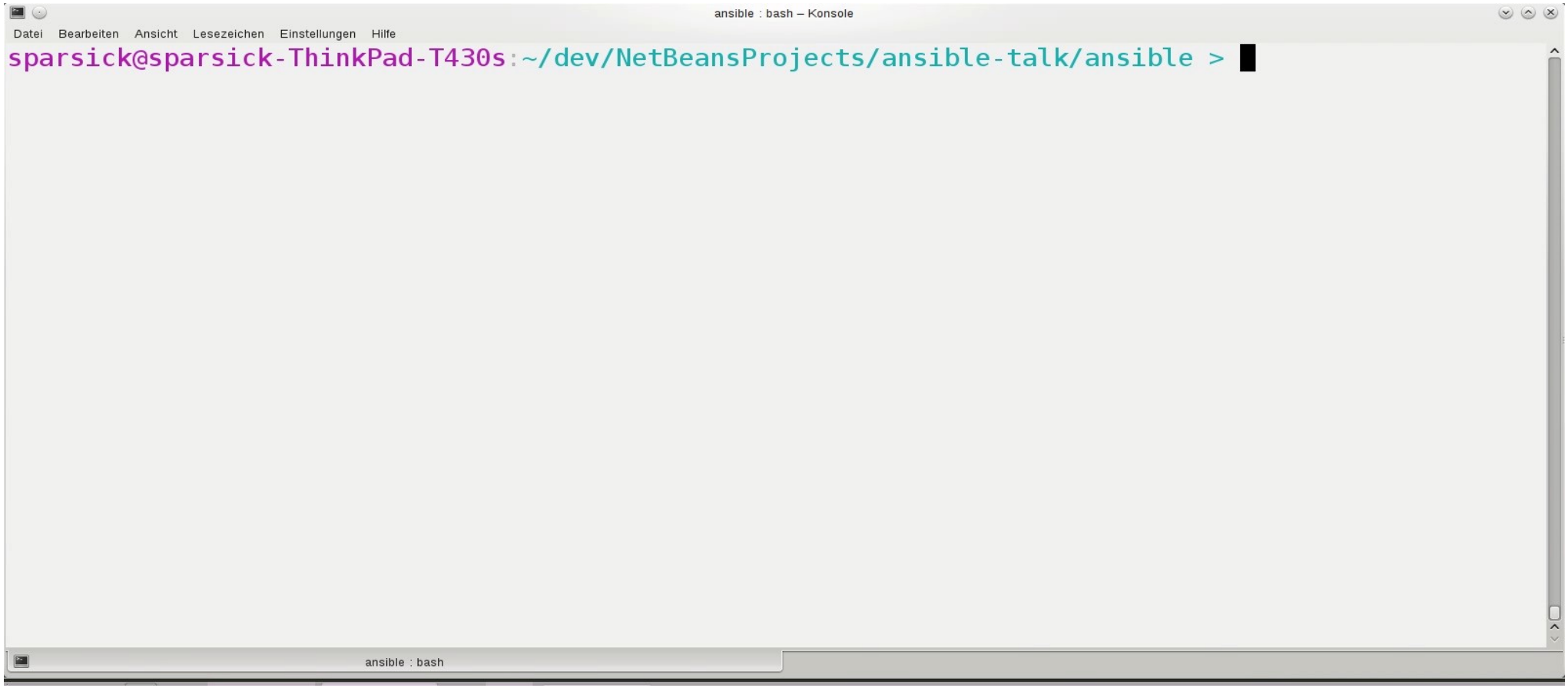
# deploy-on-tomcat Role



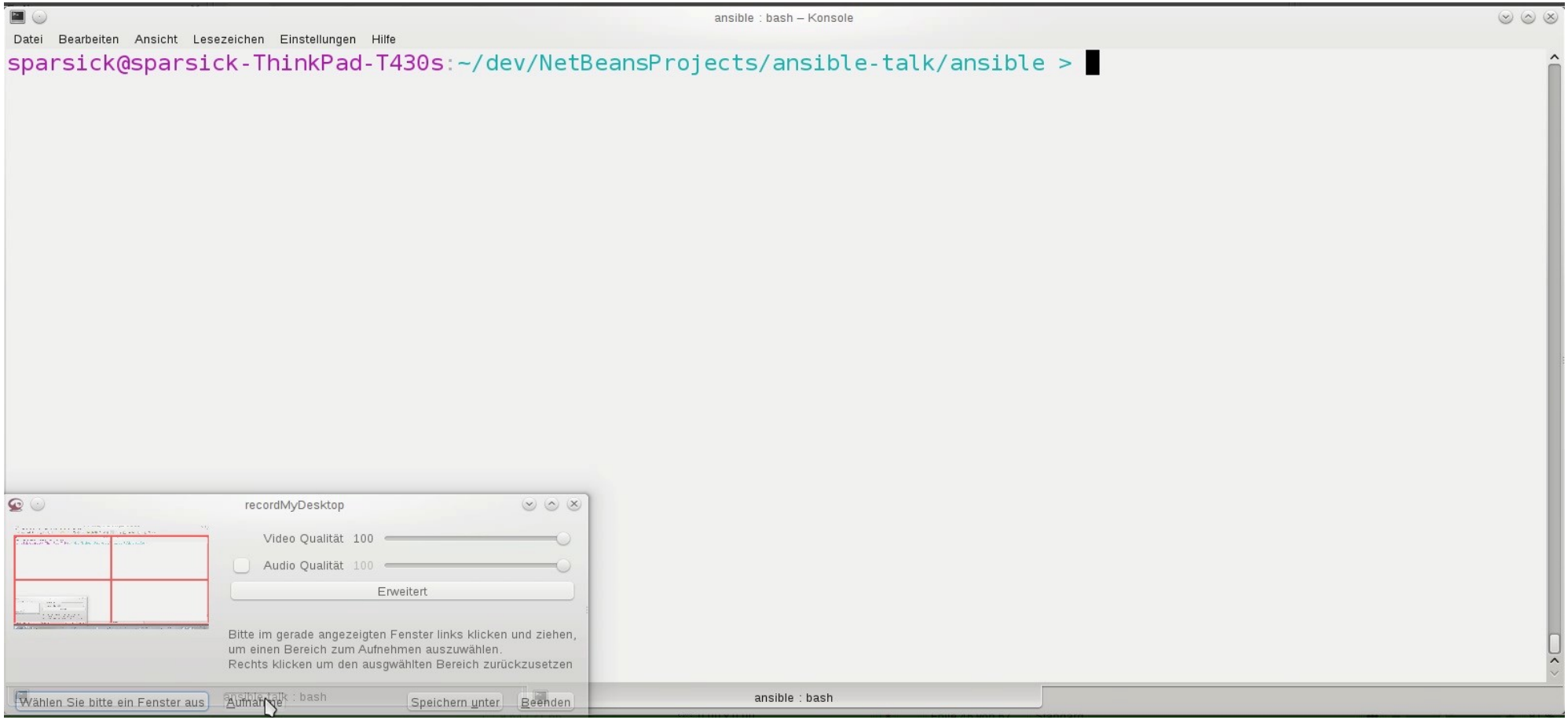
# deploy-on-tomcat Role



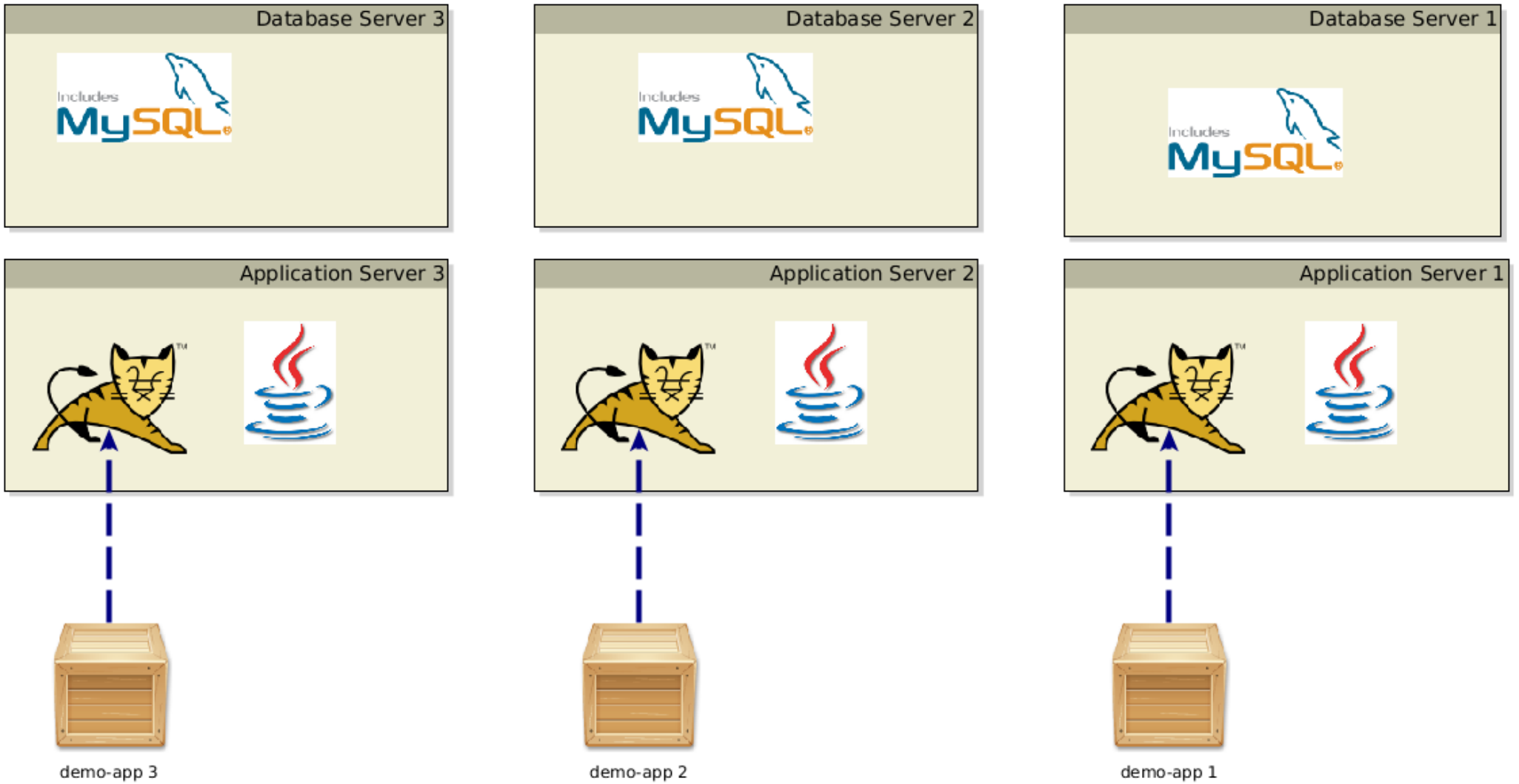
# Deploy Application Playbook



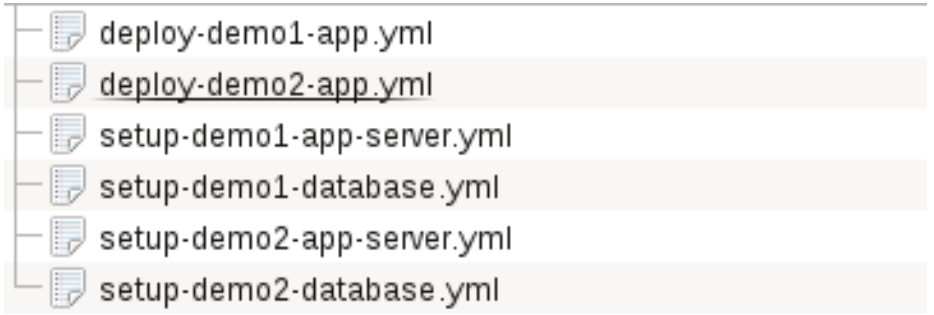
# Ad-hoc-Kommando



# Warum Roles?


Database Server 3
Includes MySQL
Database Server 2
Includes MySQL
Database Server 1
Includes MySQL
Application Server 3
Application Server 2
Application Server 1
demo-app 3
demo-app 2
demo-app 1

# Warum Roles?

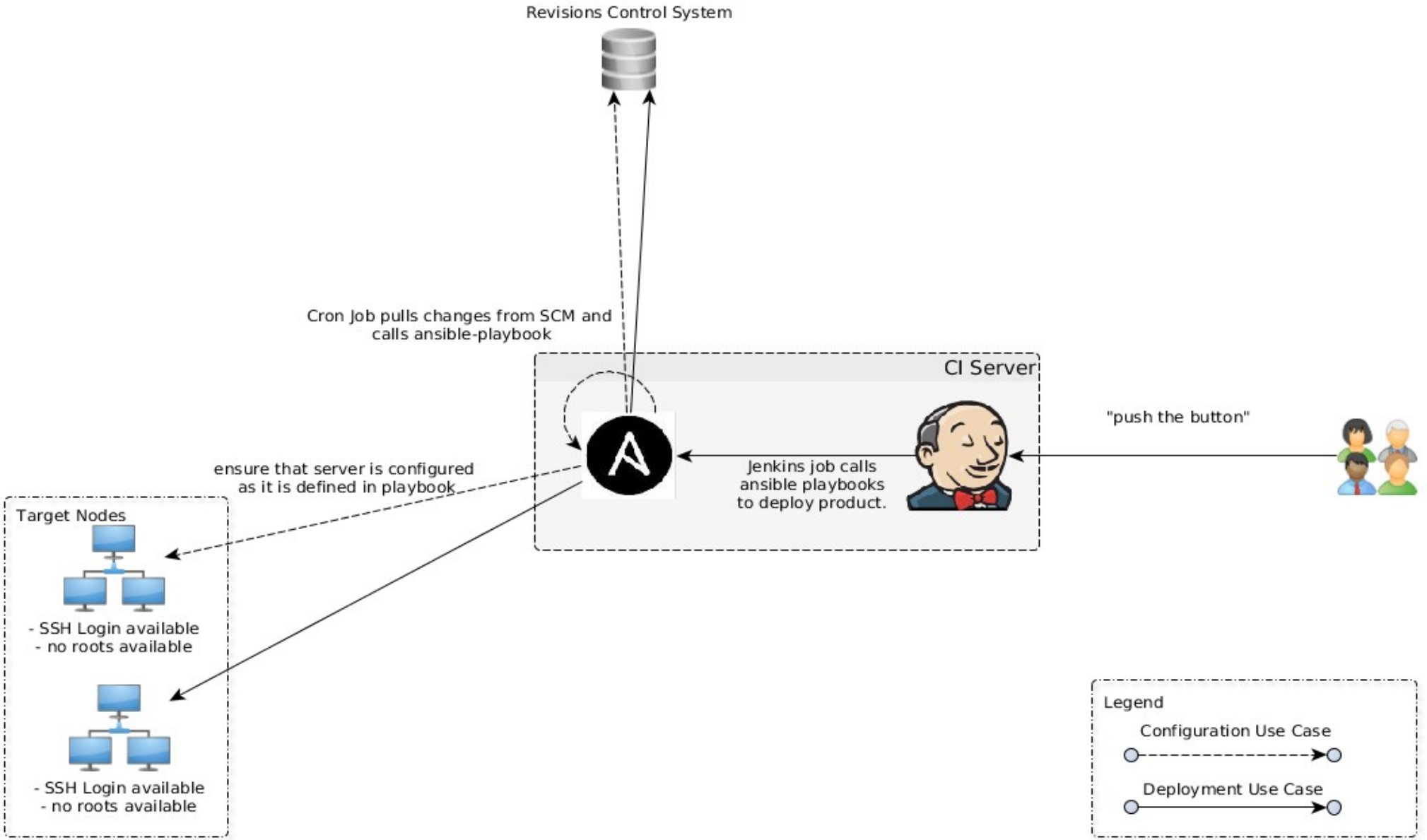

deploy-demo1-app.yml
deploy-demo2-app.yml
setup-demo1-app-server.yml
setup-demo1-database.yml
setup-demo2-app-server.yml
setup-demo2-database.yml

# Warum Roles?

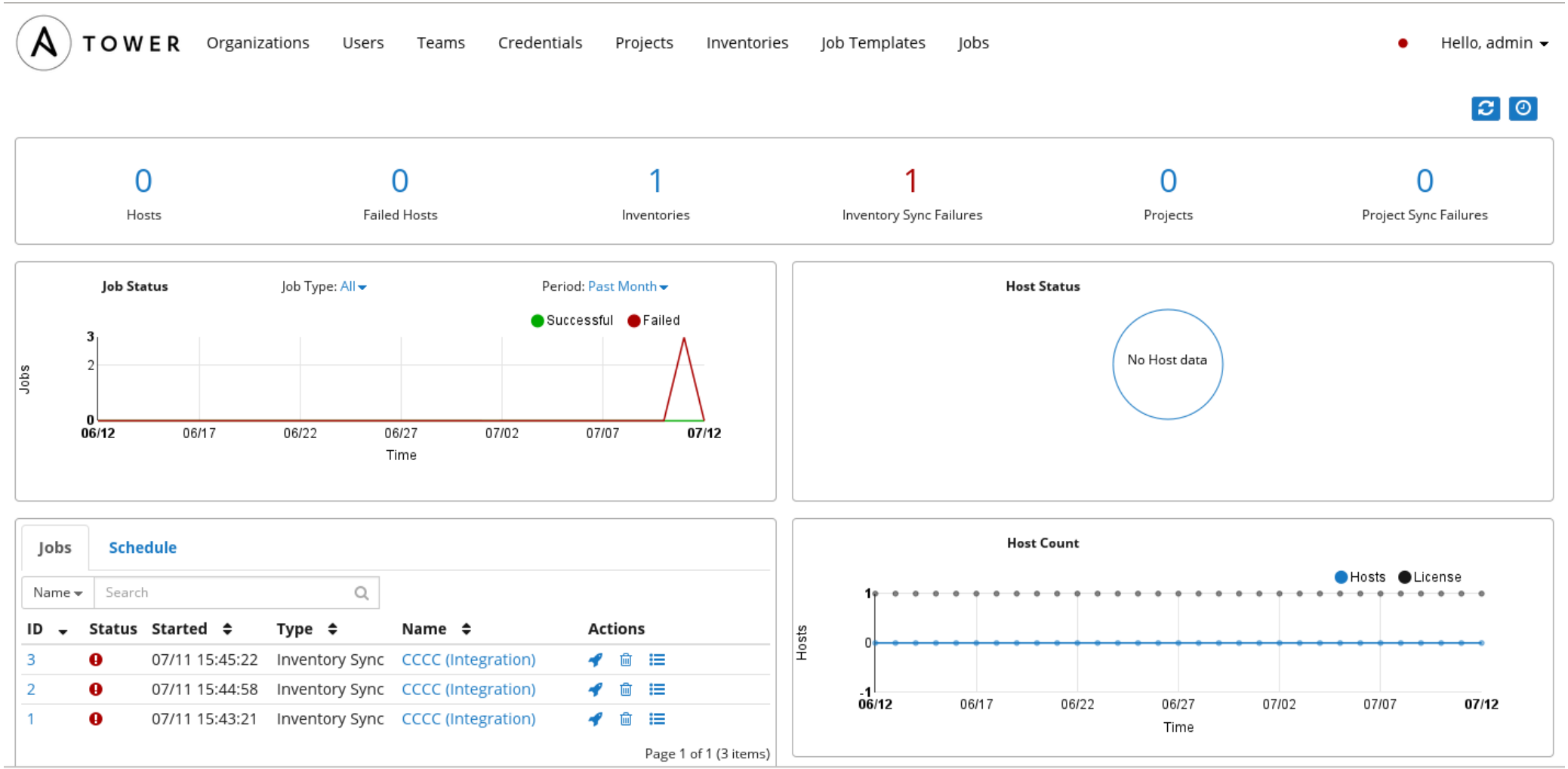
```
1  - hosts: demo1-application-server
2    roles:
3      - {role: deploy-on-tomcat, webapp_source_path: ./demo1-1.0-
       SNAPSHOT.war, webapp_target_name: demo1 }
4
```

```
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



# Ansible Tower



# Weitere Features

- Vault – Verschlüsselung
- Facts
- Dynamische Inventories
- Playbook Debugger
- Module für Docker
- Ansible Container
- Networking Support

# Wie werden Ansible Skripte getestet?

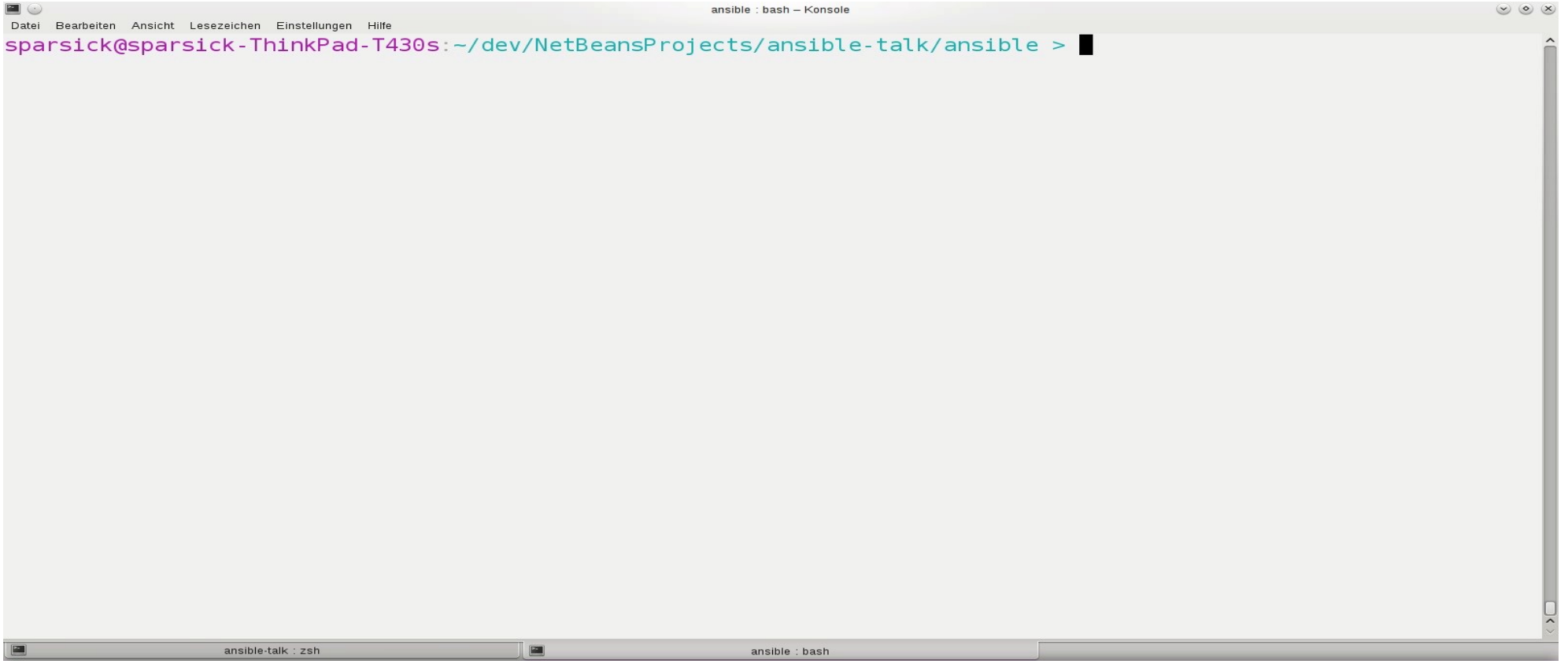
- `ansible-playbook --check`
- `ansible-playbook --syntax-check`
- Jenkins + Vagrant
- Rspec tests

# ServerSpec Tests

```
require 'spec_helper'

describe package('mysql-server') do
  it { should be_installed }
end

describe service('mysql') do
  it { should be_enabled   }
  it { should be_running   }
end

describe 'MySQL config parameters' do
  context mysql_config('bind-address') do
    its(:value) { should eq '0.0.0.0' }
  end
end

```

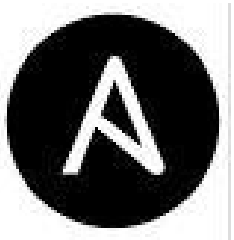
```
require 'spec_helper'

describe package('openjdk-8-jdk') do
  it { should be_installed }
end

describe command('ls /etc/init.d/tomcat') do
  its(:exit_status) { should eq 0 }
end

describe command('ls /opt/tomcat') do
  its(:exit_status) { should eq 0 }
end
```

# ServerSpec Tests



# 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

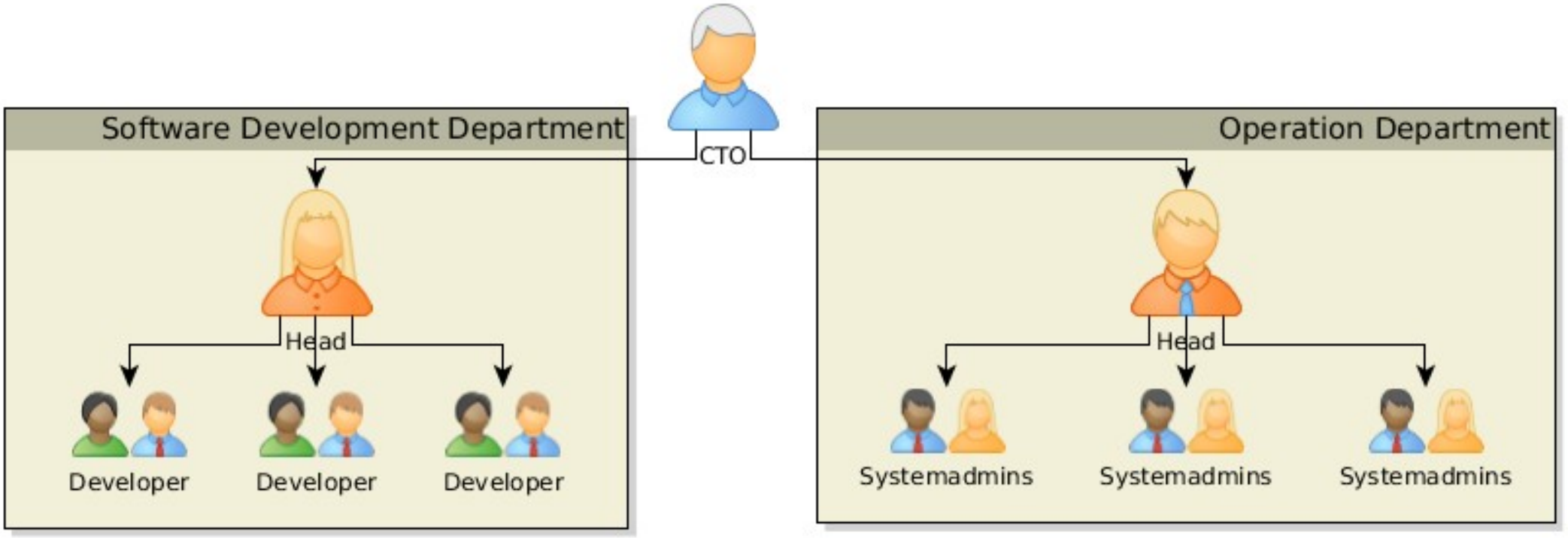

```
- name: add nodejs ppa
  apt_repository: repo='ppa:chris-lea/node.js'

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

```
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

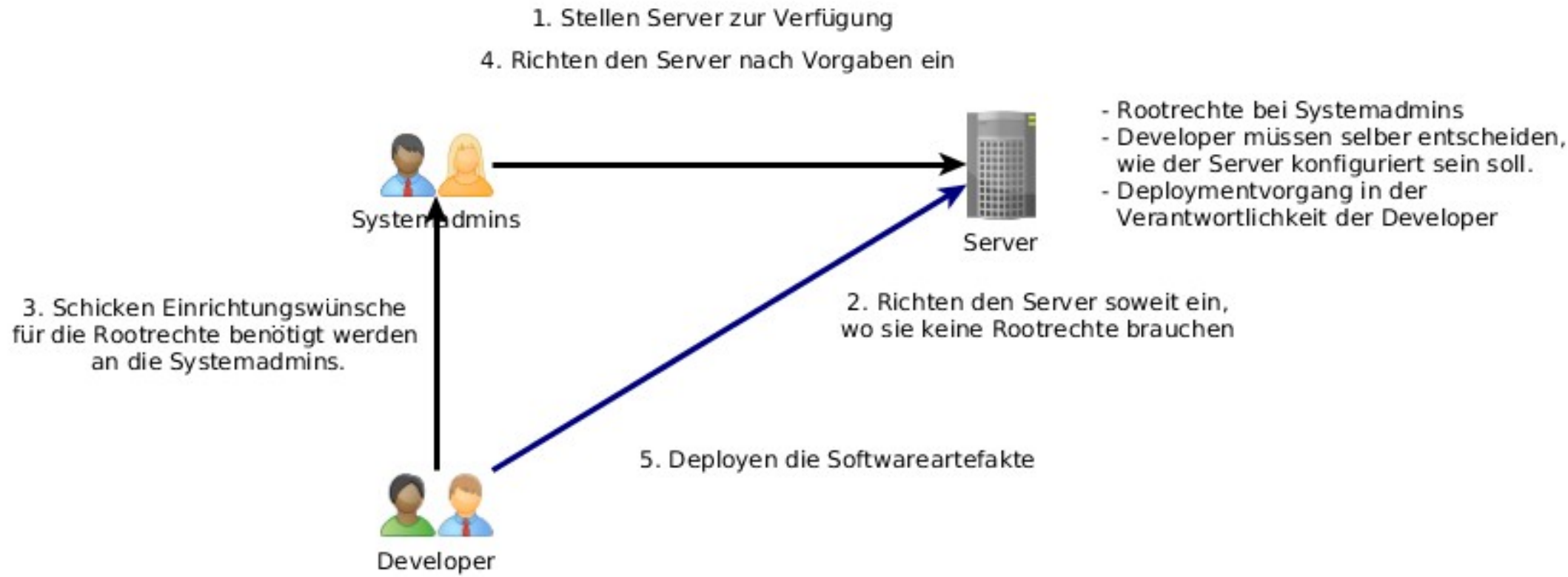
# Systemkonfiguration für Entwickler

## Organisatorische Ausgangslage
## Realität



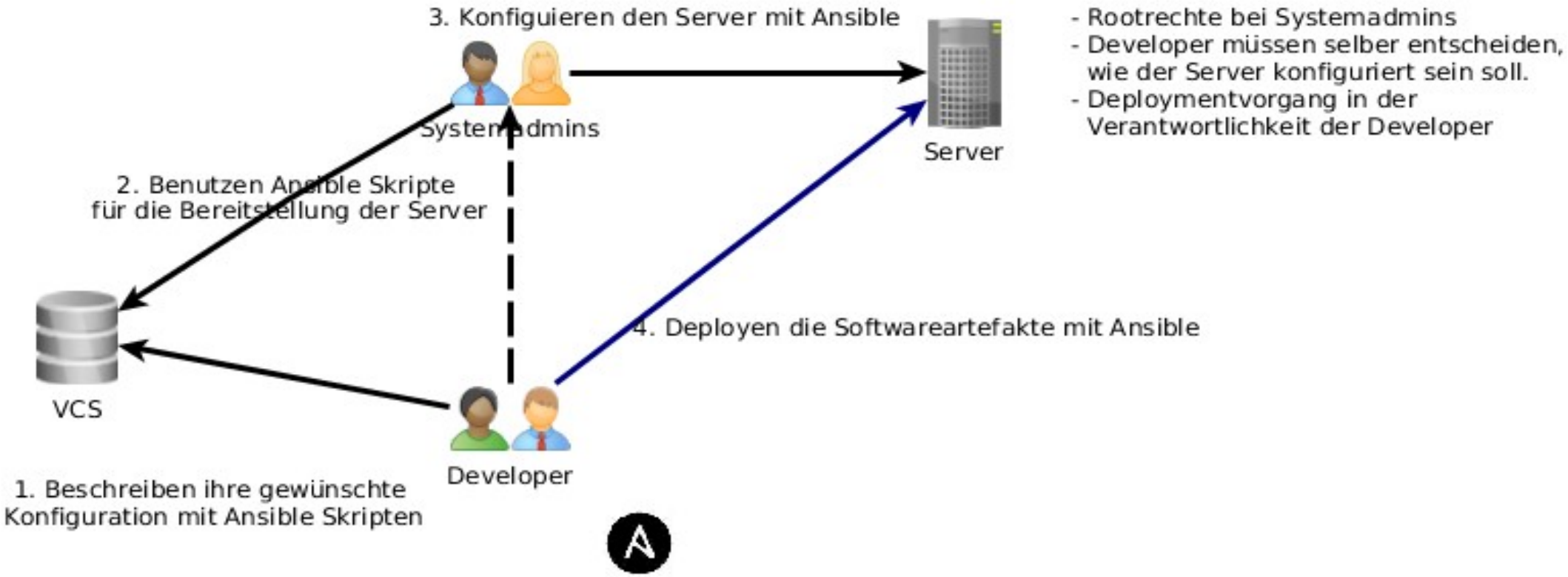
# Systemkonfiguration für Entwickler

## Prozess zwischen Development und Operation



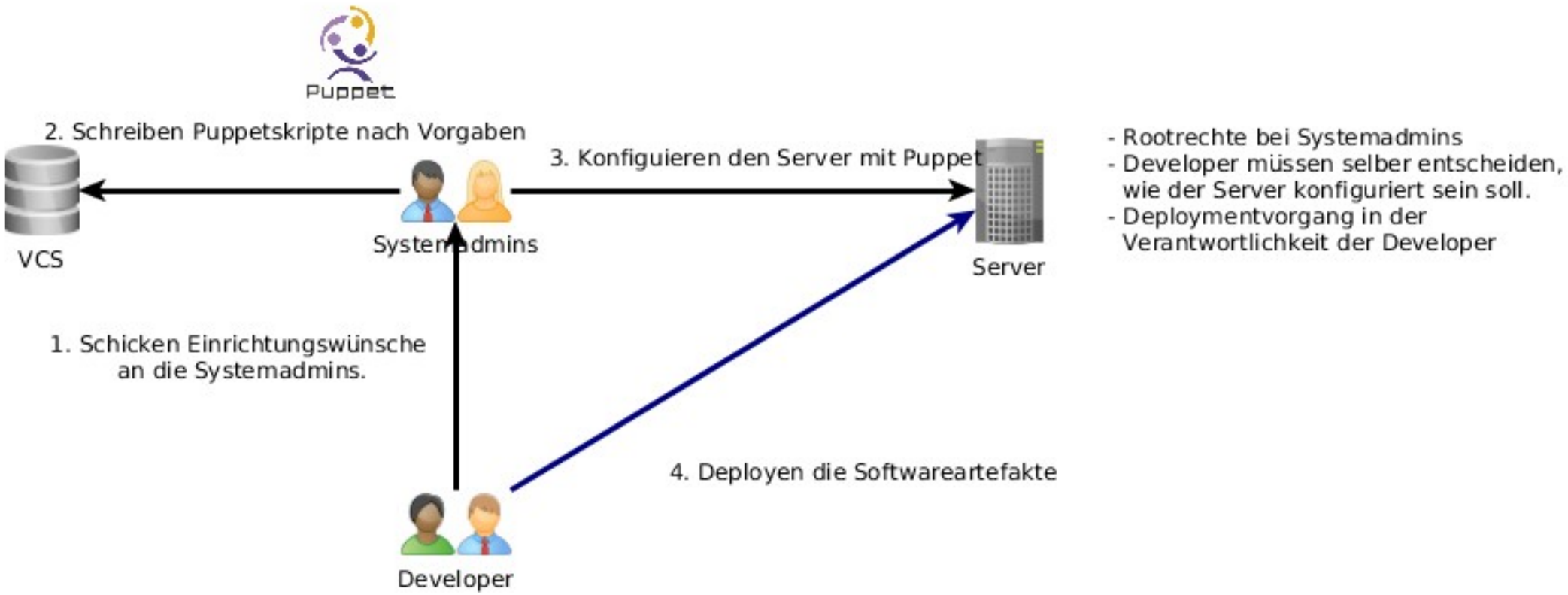
# Systemkonfiguration für Entwickler

## Lösungidee mit Ansible



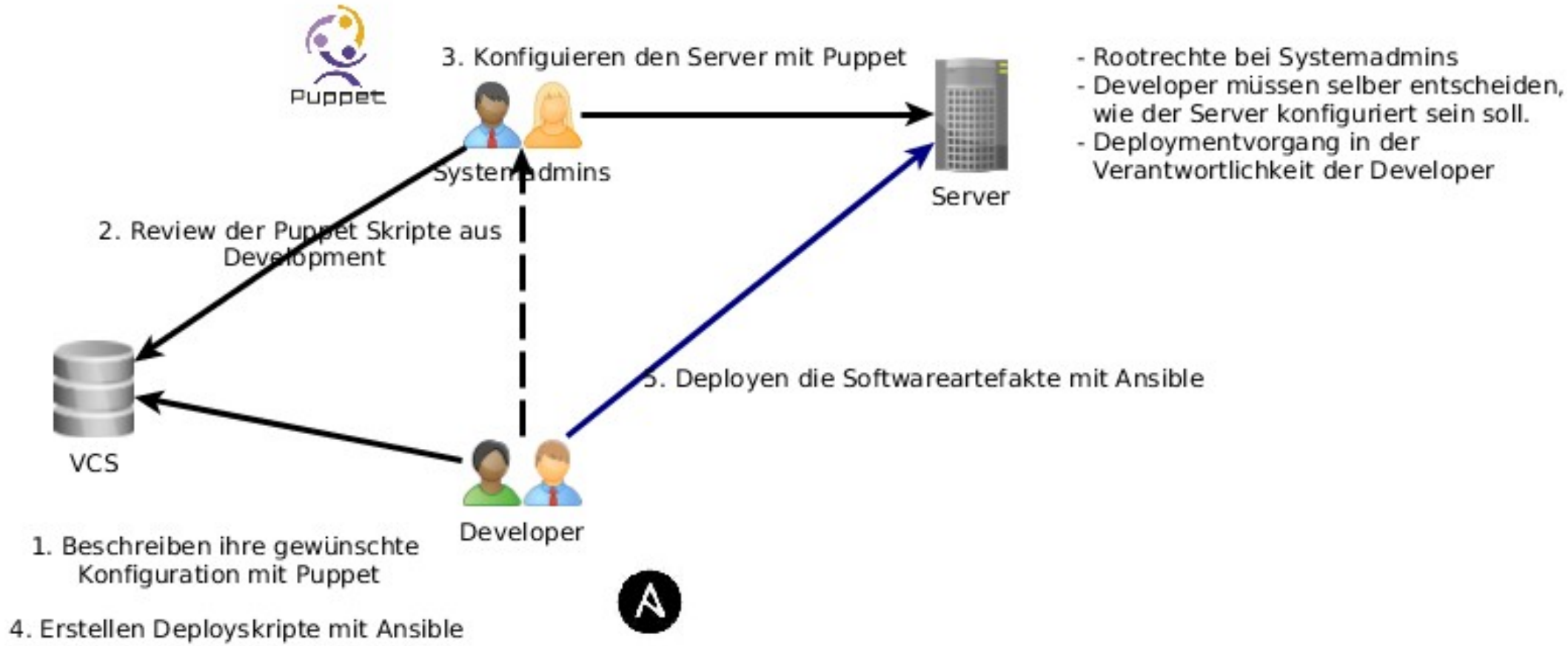
# Systemkonfiguration für Entwickler

## Variante - Prozess zwischen Development und Operation



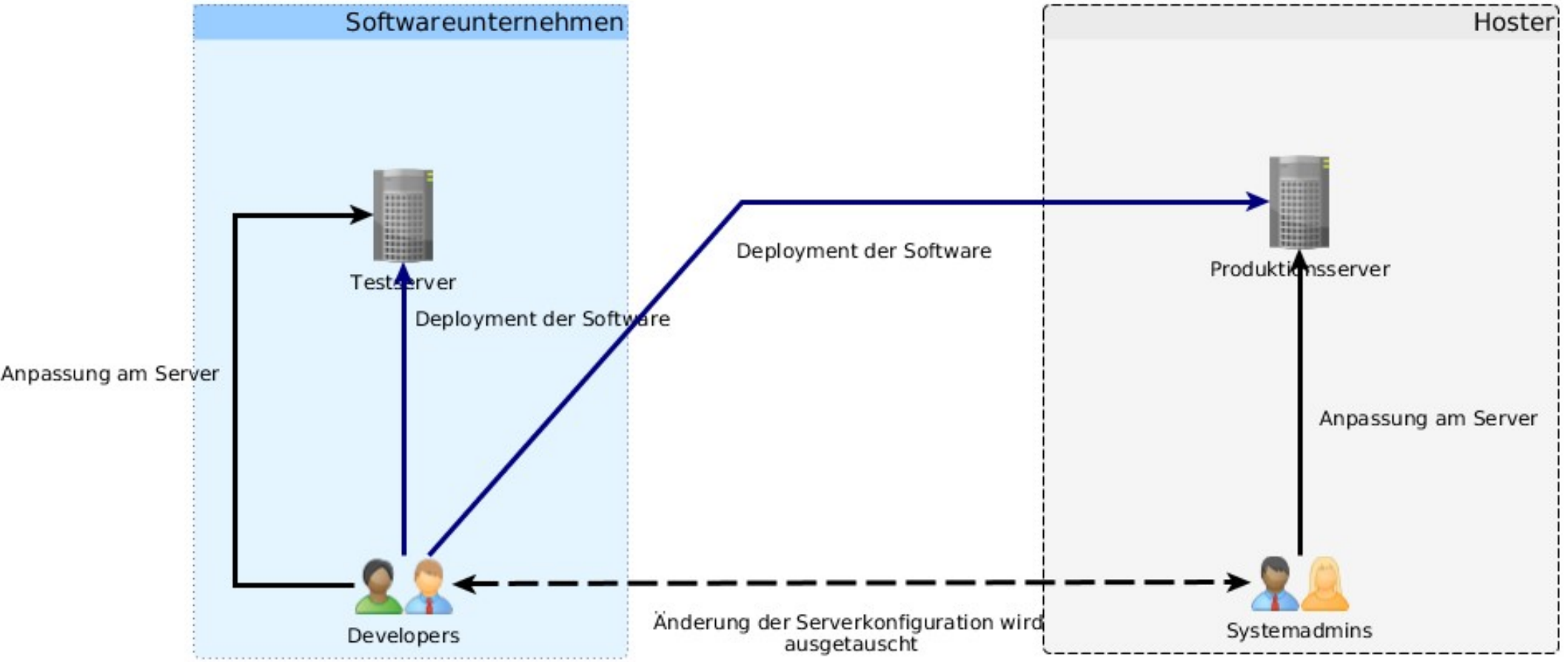
# Systemkonfiguration für Entwickler

## Lösungsvariante



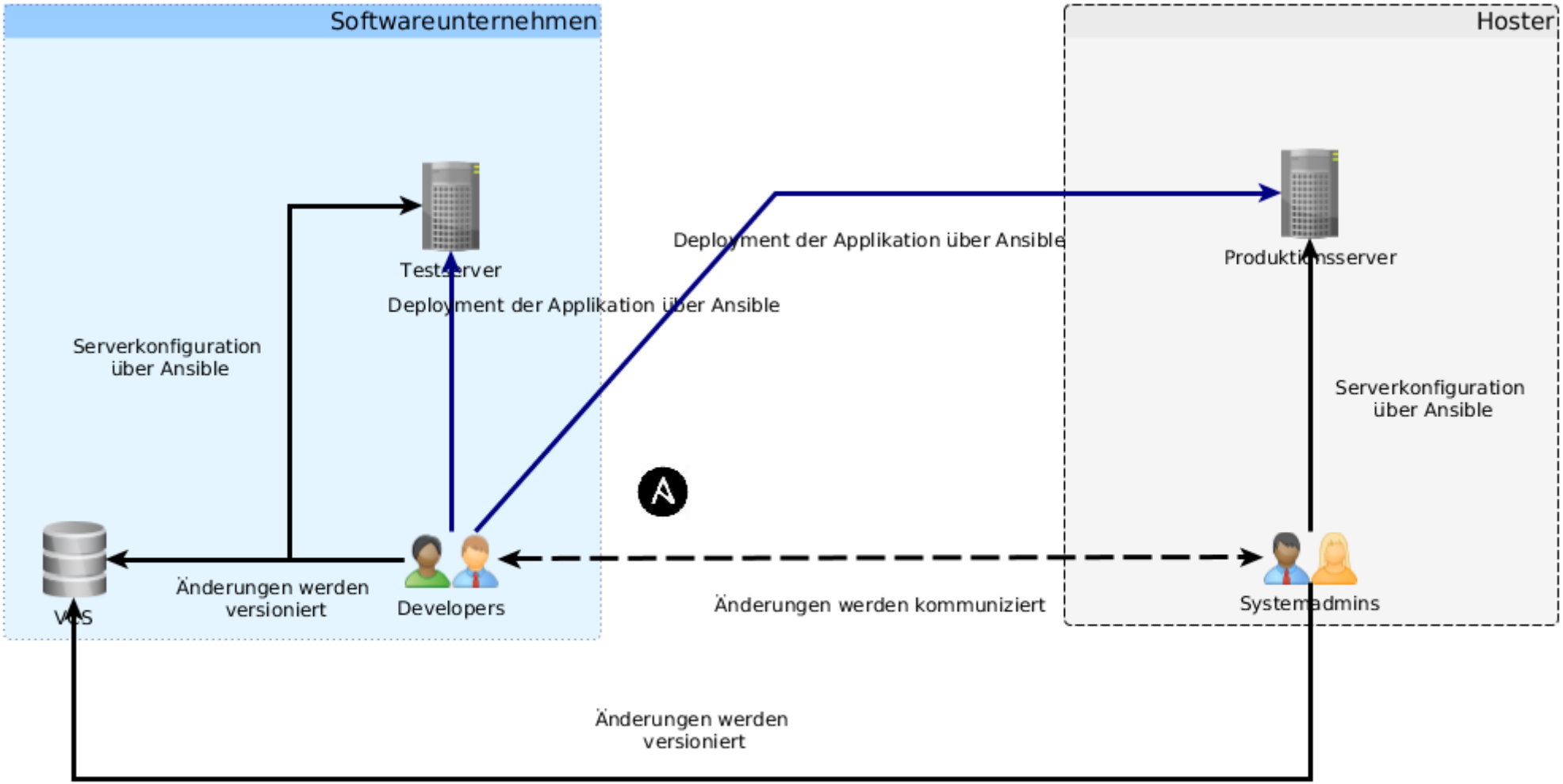
# Systemkonfiguration für Entwickler

## Produktionsserver sind beim externen Hoster



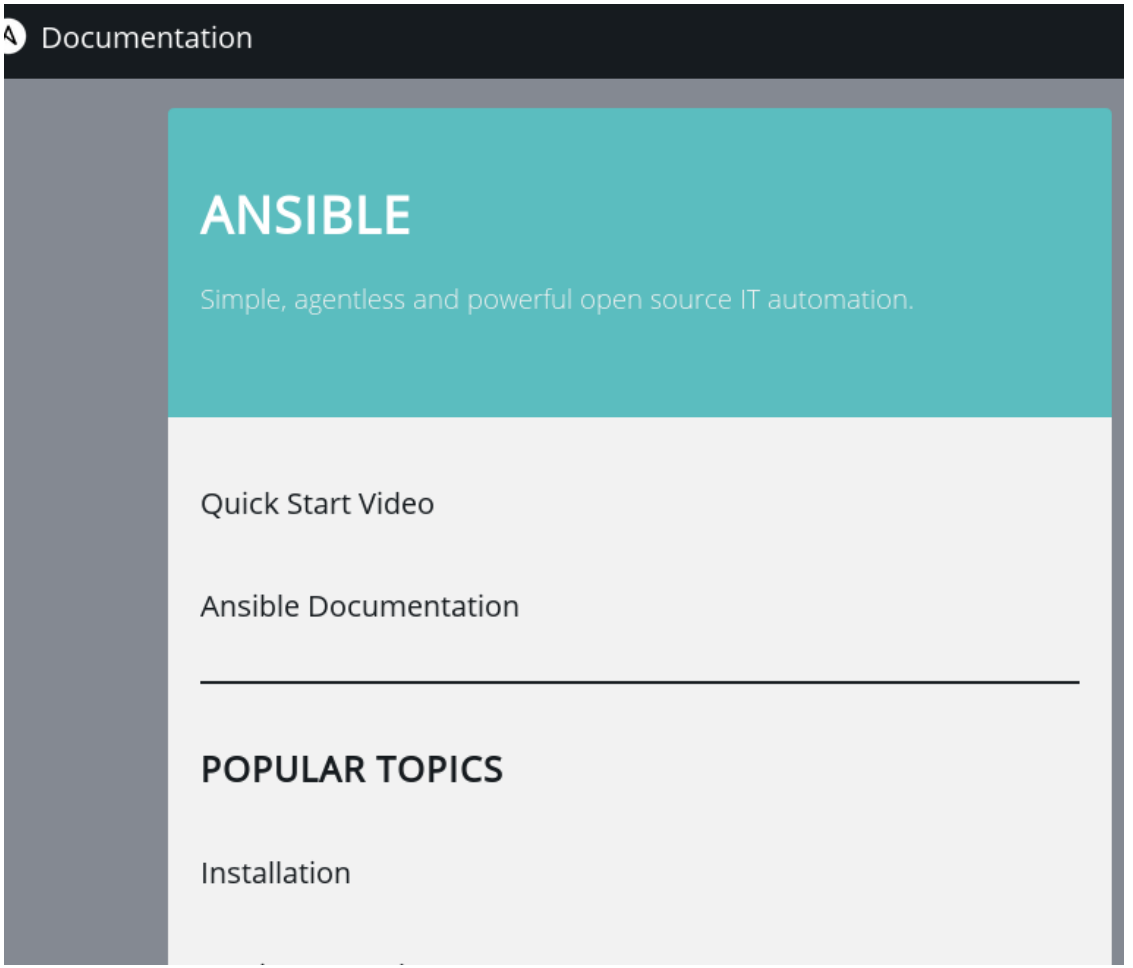
# Systemkonfiguration für Entwickler

## Lösungsidee



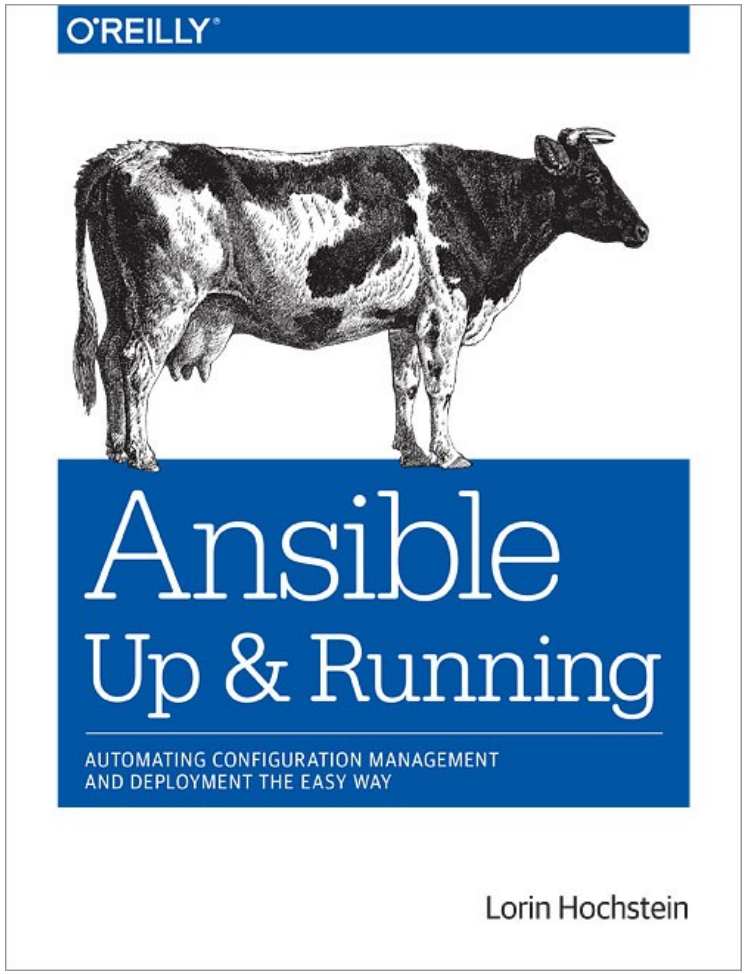

- Hoster verantwortlich für die Systemkonfiguration
- Softwareunternehmen verantwortlich für das Deployment
- Synchronisation zwischen Testserver und Produktionsserver wird vereinfacht

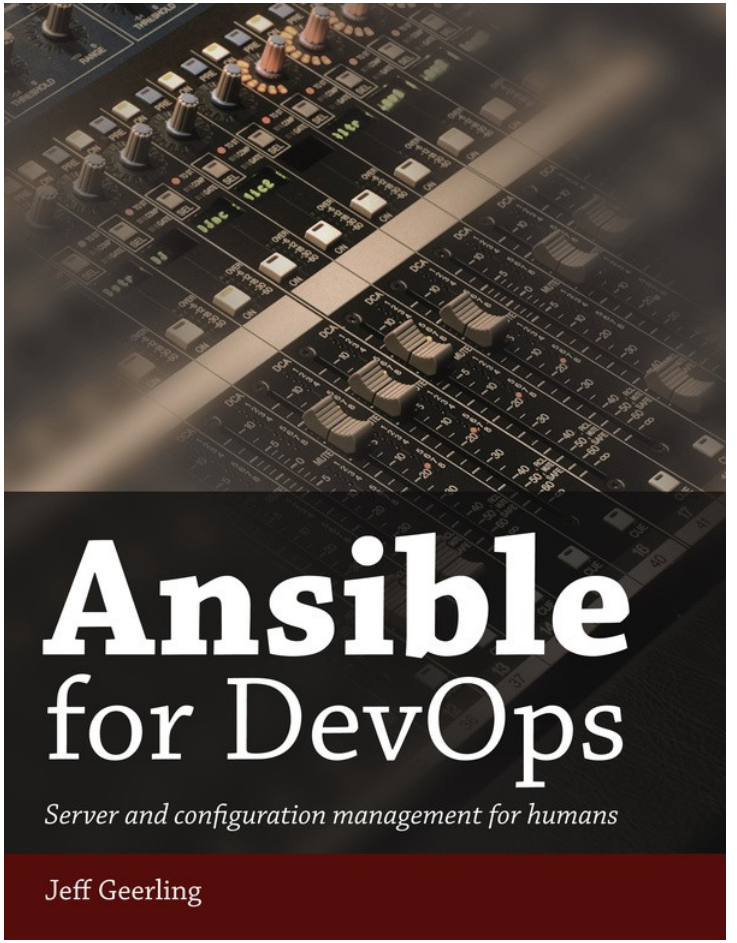
# Weitere Informationen




http://docs.ansible.com/

# Weitere Informationen



# Weitere Informationen


http://bit.ly/2cZ0IrZ

# Fragen?

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