JUG Schweiz Luzern, 15.10.2019

# Ansible für Entwickler

## Konfigurationsmanagement nicht nur für Ops

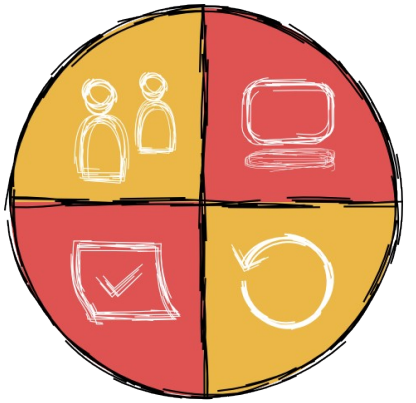
Sandra Parsick
mail@sandra-parsick.de
@SandraParsick

# Zu meiner Person

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

- Twitter: @SandraParsick
- Blog: http://blog.sandra-parsick.de
- E-Mail: mail@sandra-parsick.de

Oracle Groundbreakers



# 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

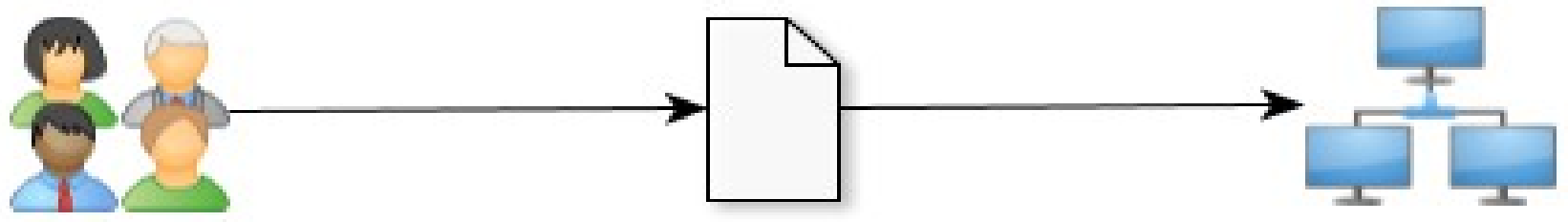
# Konfigurationsmanagement (KM)

*„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“



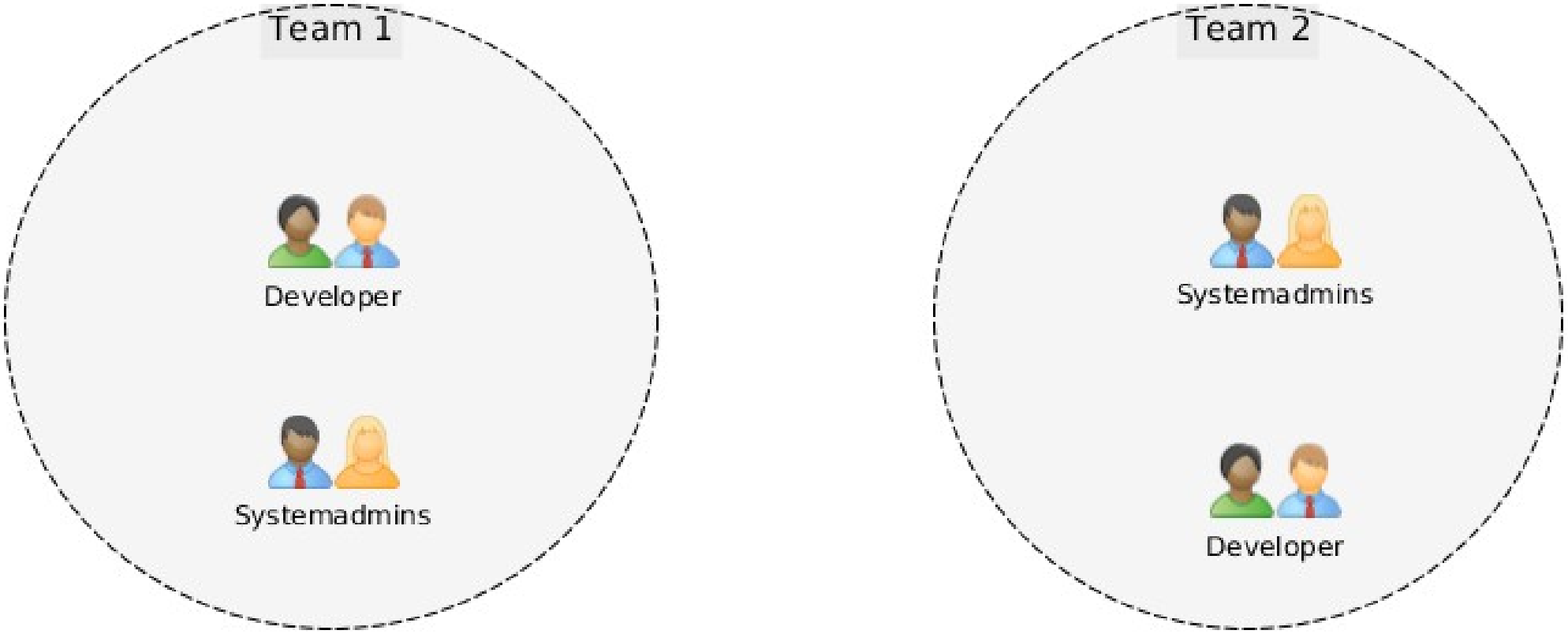


Ansible

# Warum ist es für Entwickler interessant?

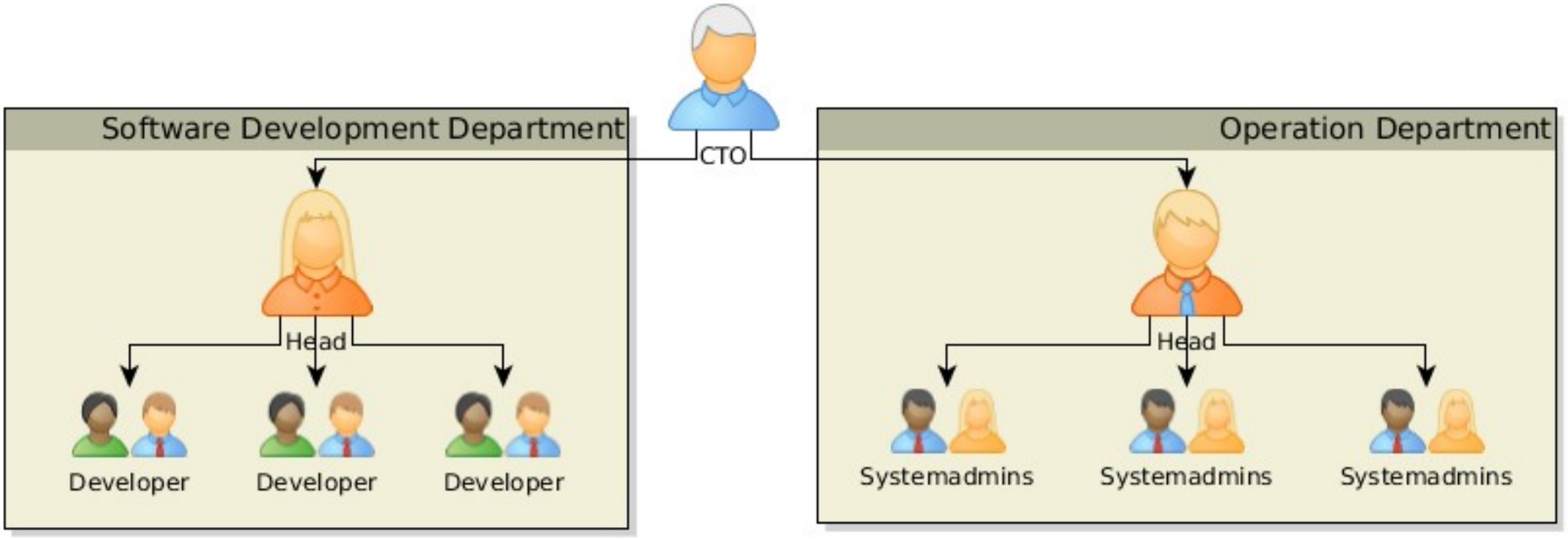
# Systemkonfiguration für Entwickler

## Organisatorische Ausgangslage
## Wunsch



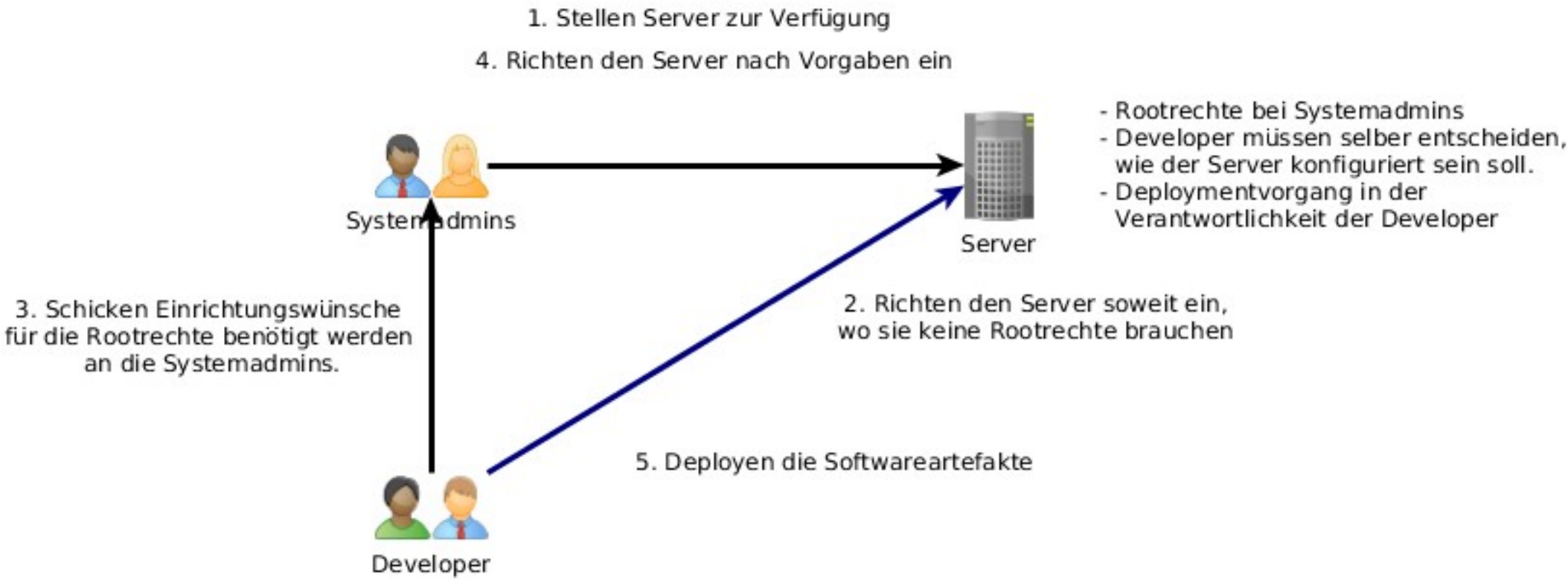
# Systemkonfiguration für Entwickler

## Organisatorische Ausgangslage
## Realität



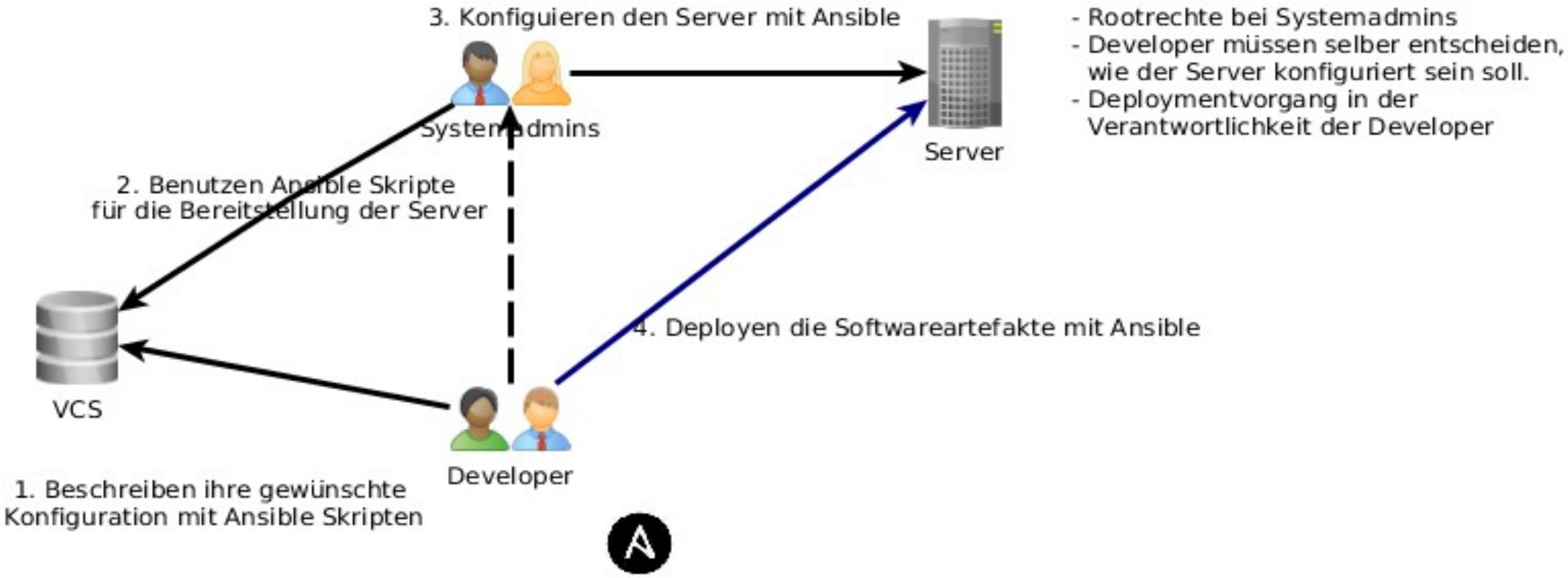
# Systemkonfiguration für Entwickler

## Prozess zwischen Development und Operation



# Systemkonfiguration für Entwickler

## Lösungidee mit Ansible

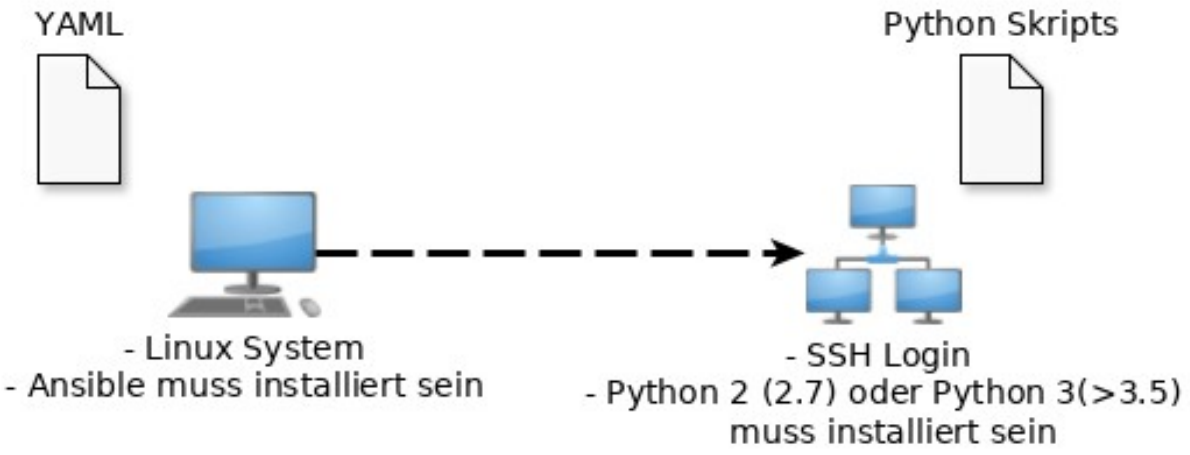


# Einführung in Ansible

# Ansible

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

# Funktionsweise



# Exkurs: YAML

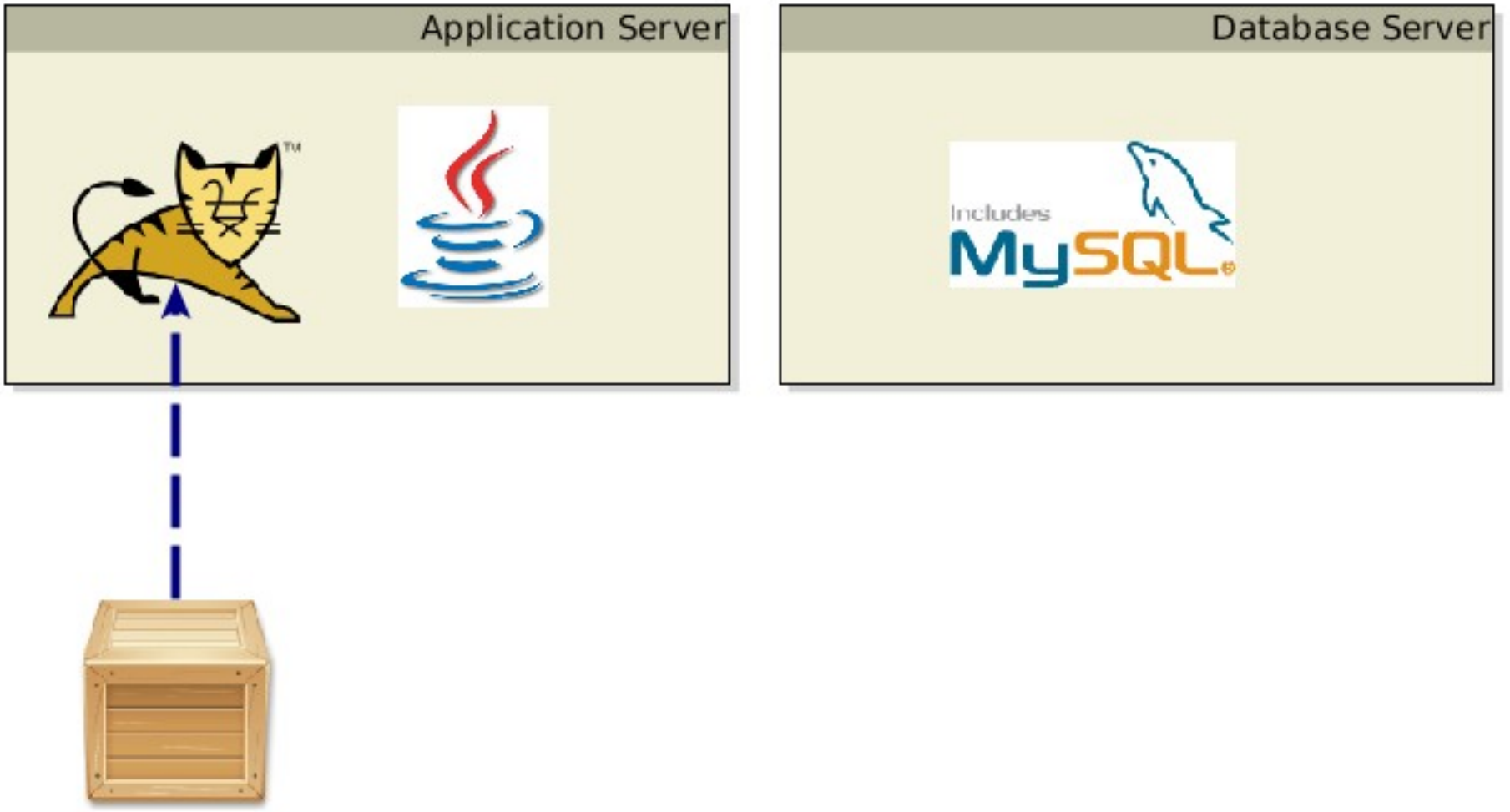
**YAML**

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

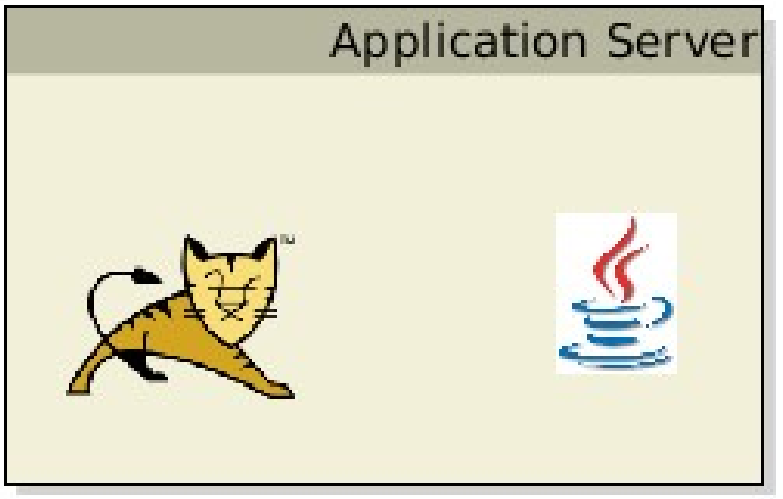
**JSON**

```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



```
- hosts: application_server
  vars:
    tomcat_version: 8.5.46
    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
```

# Inventories

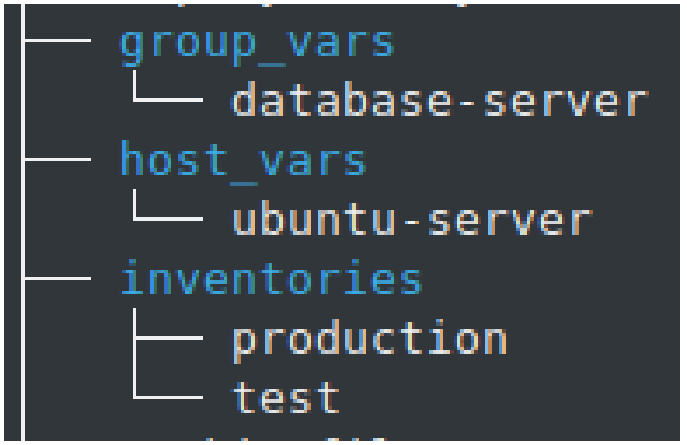
## Production

```
[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

[all:vars]
ansible_user=vagrant
```

# Inventories




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

```
- hosts: application_server
  vars:
    tomcat_version: 8.5.46
    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
```

# Ansible Modules

## Module Index

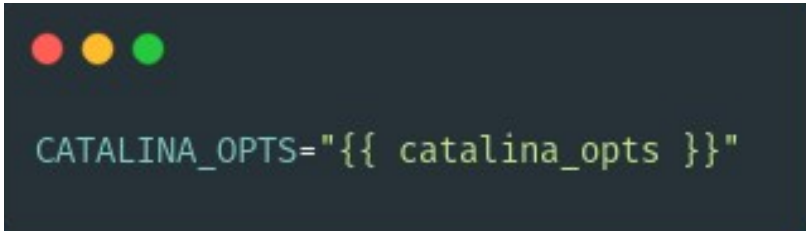
- All Modules
- Cloud Modules
- Clustering Modules
- Commands Modules
- Crypto Modules
- Database Modules
- Files Modules
- Identity Modules
- Inventory Modules
- Messaging Modules
- Monitoring Modules
- Net Tools Modules
- Network Modules
- Notification Modules
- Packaging Modules
- Remote Management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

```
- hosts: application_server
  vars:
    tomcat_version: 8.5.46
    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
```

```
    - 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 tomcat as service
      copy: src=roles/tomcat8/files/tomcat.service dest=/etc/systemd/system/
      become: yes
      become_method: sudo
```

# Templates

- setenv.sh.j2



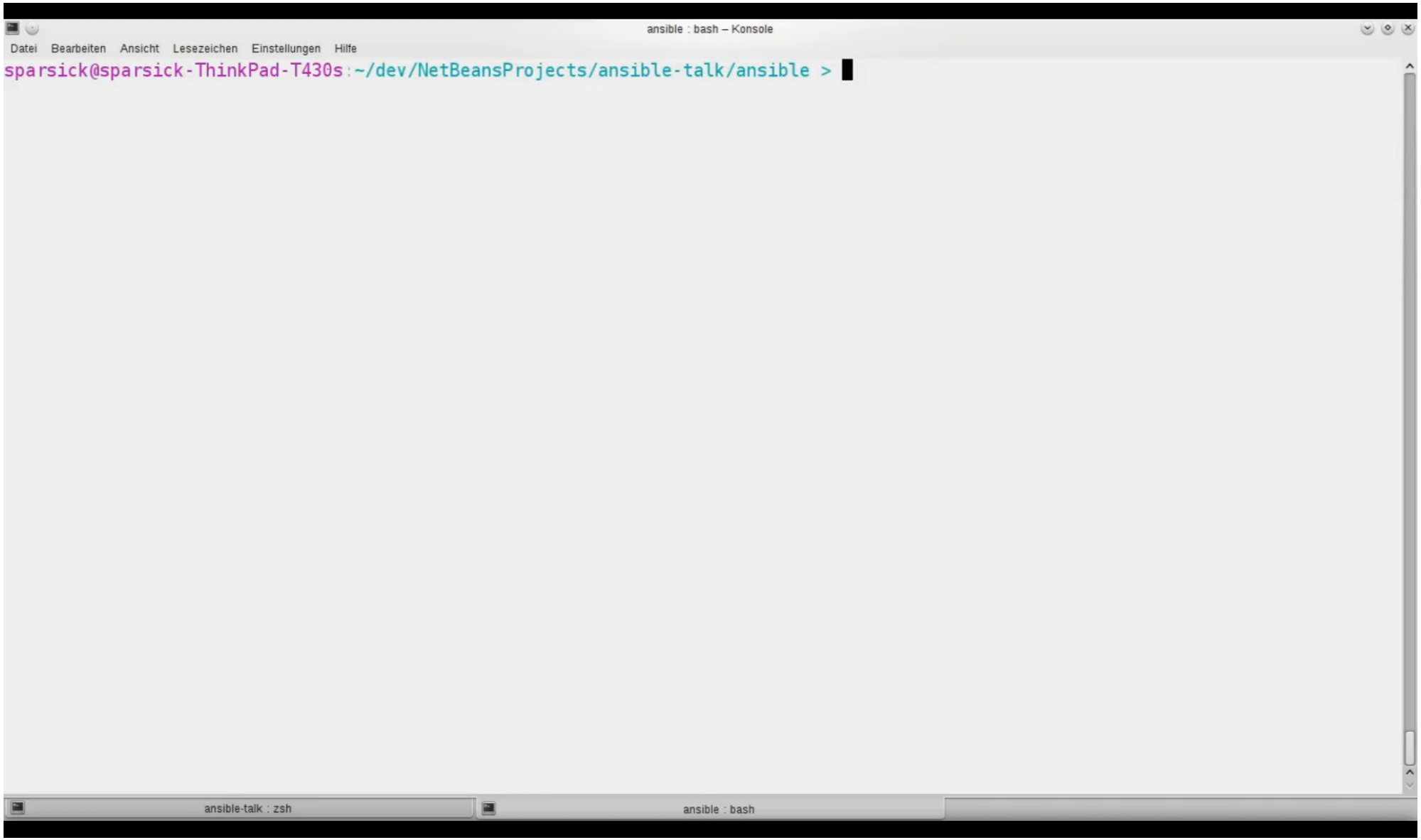
# Templates - Jinja2

- Templating engine für Python

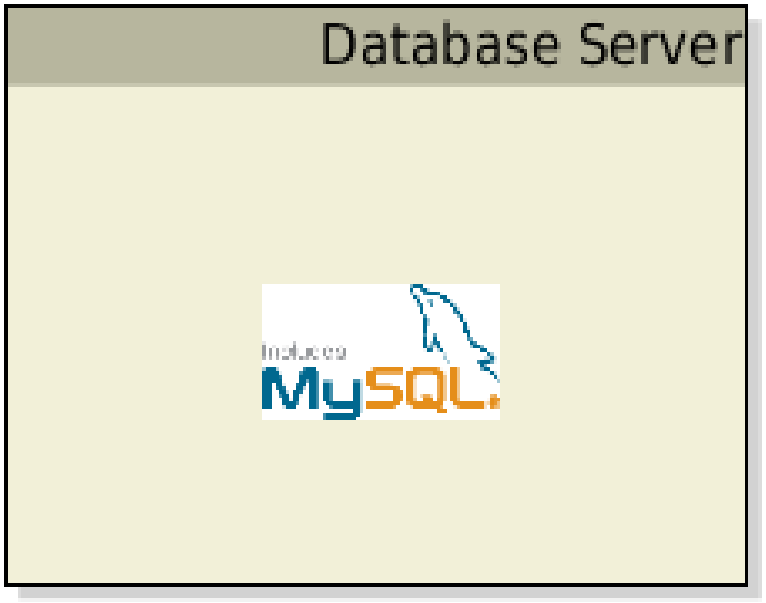
```
<title>{% block title %}{% endblock %}</title>
<ul>
{% for user in users %}
  <li><a href="{{ user.url }}">{{ user.username }}</a></li>
{% endfor %}
</ul>
```

- Mehr Information unter https://jinja.palletsprojects.com

# Setup Application Server Playbook



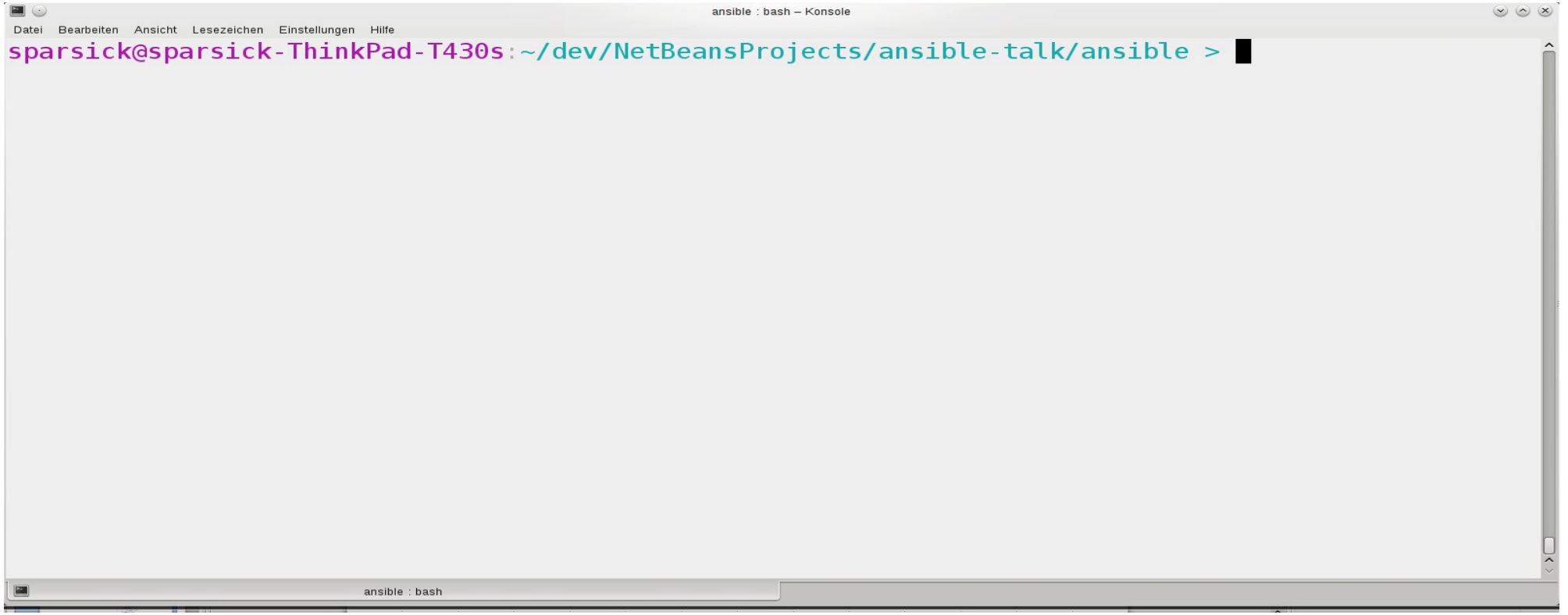
# Setup Database Server Playbook



```yaml
- hosts: database_server
  become: yes
  become_method: sudo

  tasks:
    - name: install needed python package
      apt:
        name: ['python-mysqldb', 'python-apt']
        state: present

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

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

    - name: set bind address
      lineinfile: dest=/etc/mysql/mysql.conf.d/mysqld.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=g3h31m priv=*.*:ALL,GRANT state=present host=%

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

# Setup Database Server Playbook

# Setup Database Server Playbook

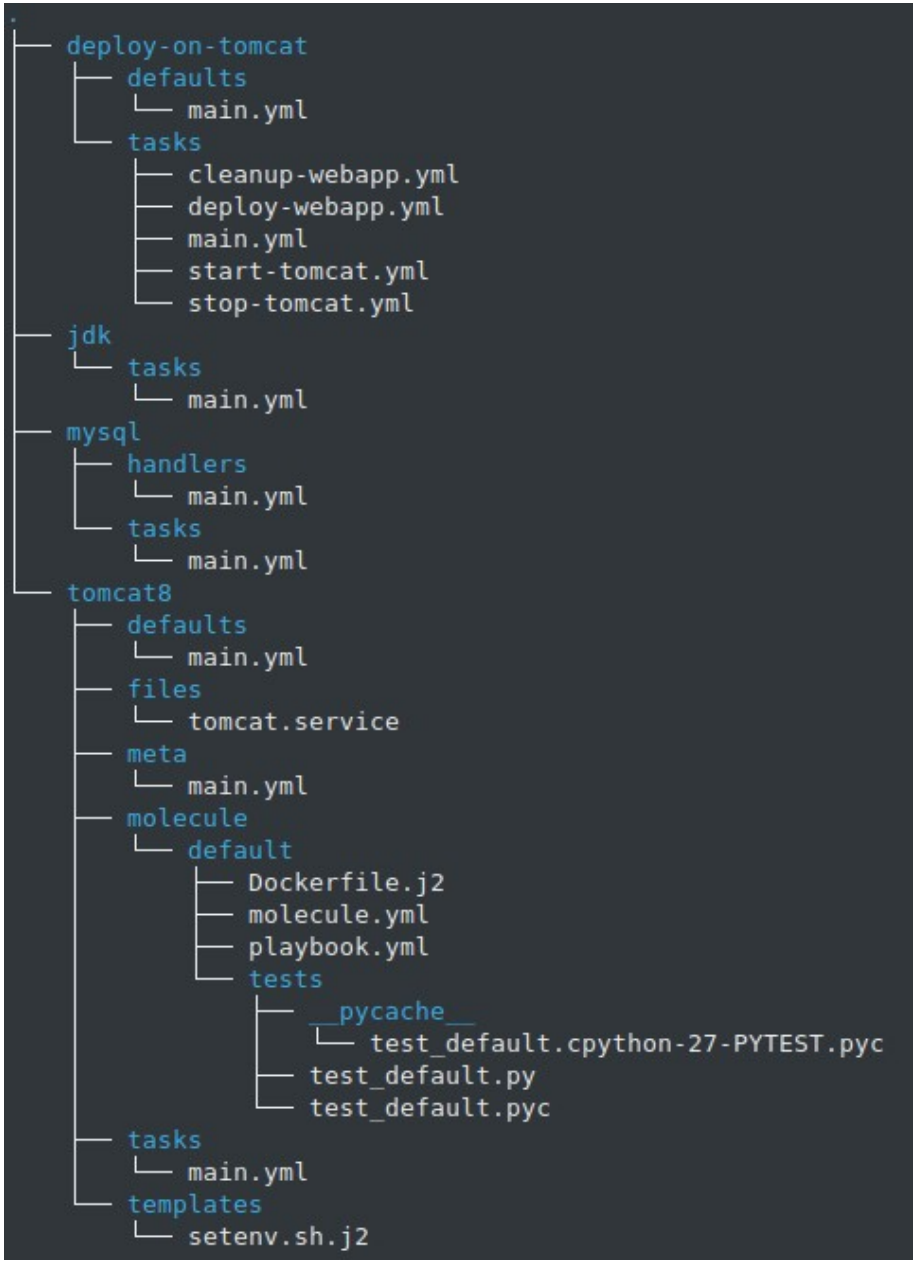


```
- hosts: application_server
  vars:
    tomcat_version: 8.5.46
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  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
```

```yaml
- hosts: database_server
  become: yes
  become_method: sudo

  tasks:
    - name: install needed python package
      apt:
        name: ['python-mysqldb', 'python-apt']
        state: present

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

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

    - name: set bind address
      lineinfile: dest=/etc/mysql/mysql.conf.d/mysqld.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=g3h31m priv=*.*:ALL,GRANT state=present host=%

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

# Roles

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



# Setup Playbooks mit Roles

- Setup Application Server

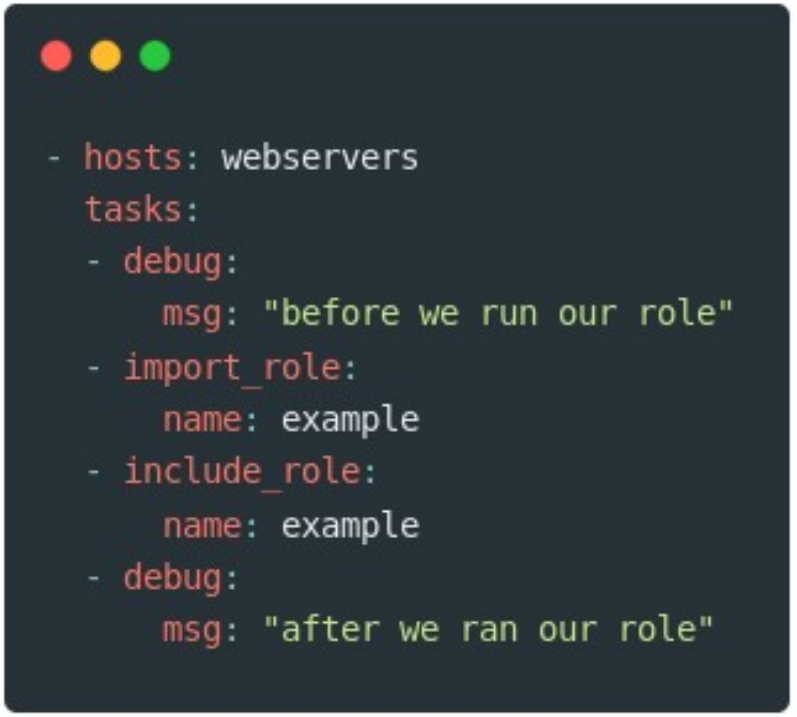
```yaml
- hosts: application_server
  roles:
    - jdk
    - { role: tomcat8, tomcat_version: 8.5.46 }
```

- Setup Database Server

```yaml
- hosts: database_server
  roles:
    - mysql
```

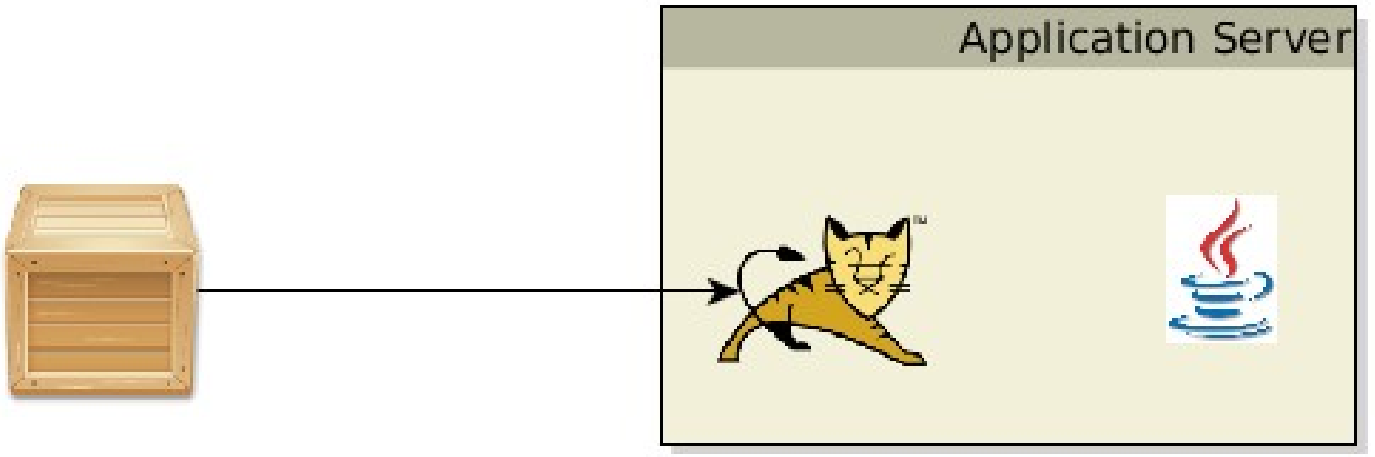
# include_role, import_role (seit v2.3)

---



```yaml
- hosts: webservers
  tasks:
  - debug:
      msg: "before we run our role"
  - import_role:
      name: example
  - include_role:
      name: example
  - debug:
      msg: "after we ran our role"
```

# Java Webapplikation Deployment



# Deploy Application Playbook

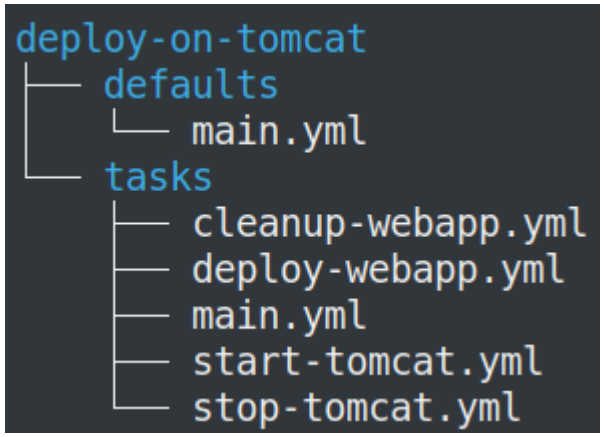
```
- hosts: application_server
  roles:
    - {role: deploy-on-tomcat, webapp_source_path: ./demo-app-ansible-deploy-1.0-SNAPSHOT.war,
webapp_target_name: demo }
```

# deploy-on-tomcat Role

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

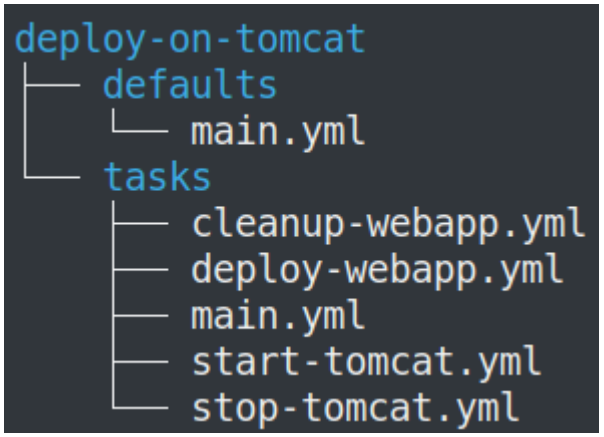
```
# cat tasks/main.yml
- import_tasks: stop-tomcat.yml
- import_tasks: cleanup-webapp.yml
- import_tasks: deploy-webapp.yml
- import_tasks: start-tomcat.yml
```

# deploy-on-tomcat Role



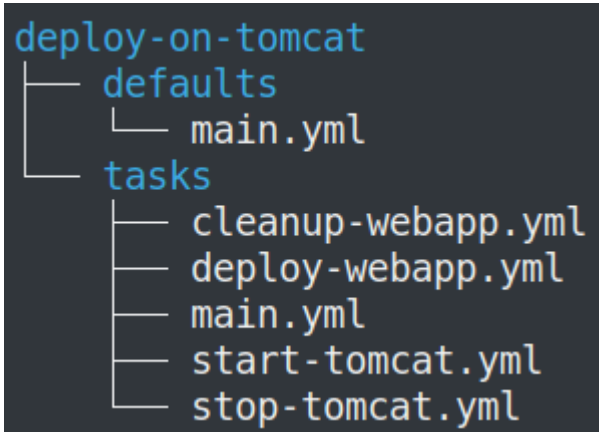

```
# cat tasks/stop-tomcat.yml
- name: stop tomcat
  service: name=tomcat state=stopped
  become: true

- name: wait tomcat shutdown
  wait_for: port=8080 state=stopped timeout=60
```

# deploy-on-tomcat Role



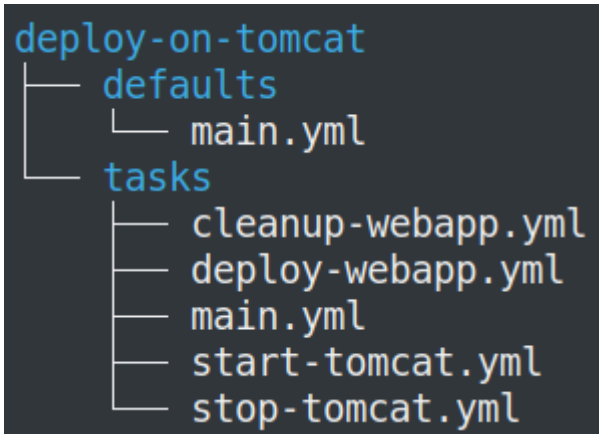

```
# cat tasks/cleanup-webapp.yml
- name: cleanup {{ webapp_target_name }}
  file: name={{tomcat_app_base}}/{{ webapp_target_name }} state=absent
```

# deploy-on-tomcat Role




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

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

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

# deploy-on-tomcat Role



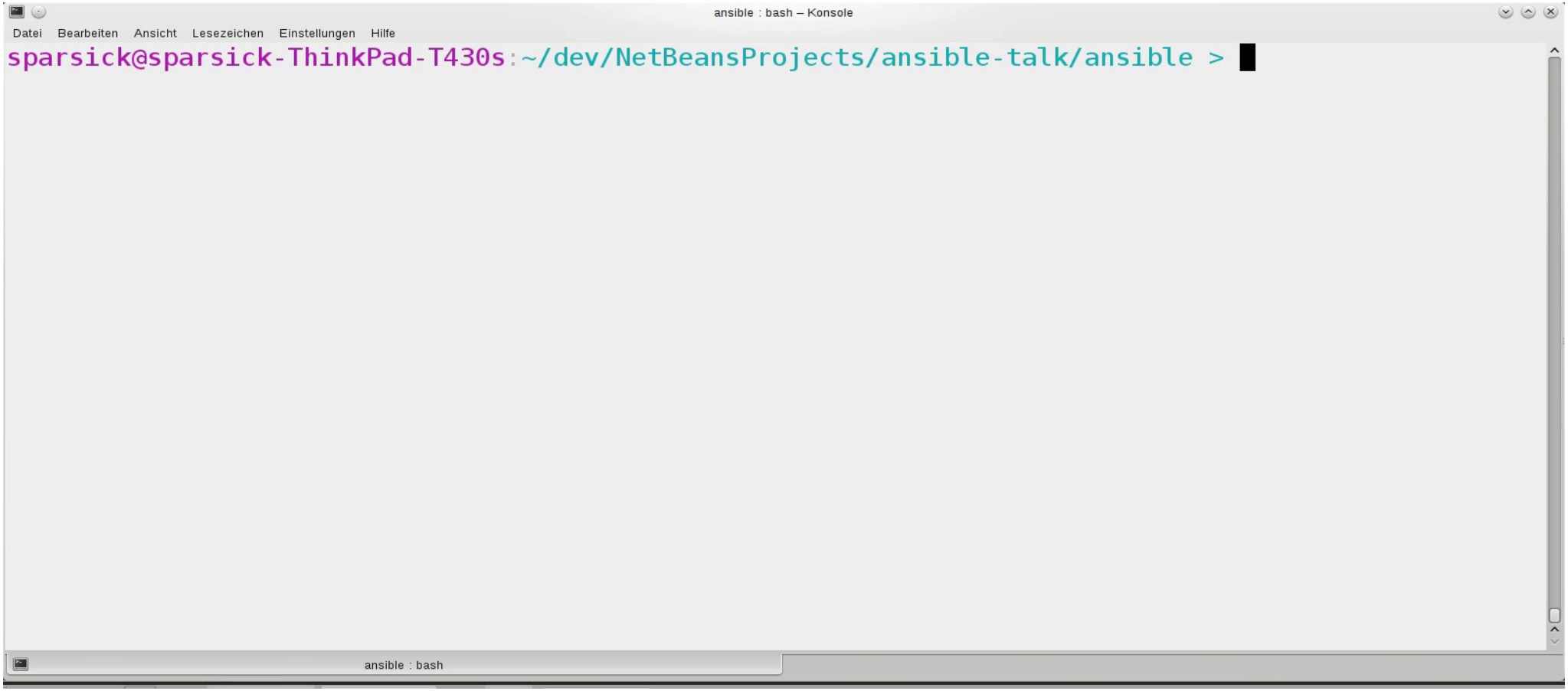

```
# cat tasks/start-tomcat.yml
- name: start tomcat
  service: name=tomcat enabled=yes state=started
  become: true

- name: wait for tomcat to start
  wait_for: port=8080 timeout=60
```

# deploy-on-tomcat Role

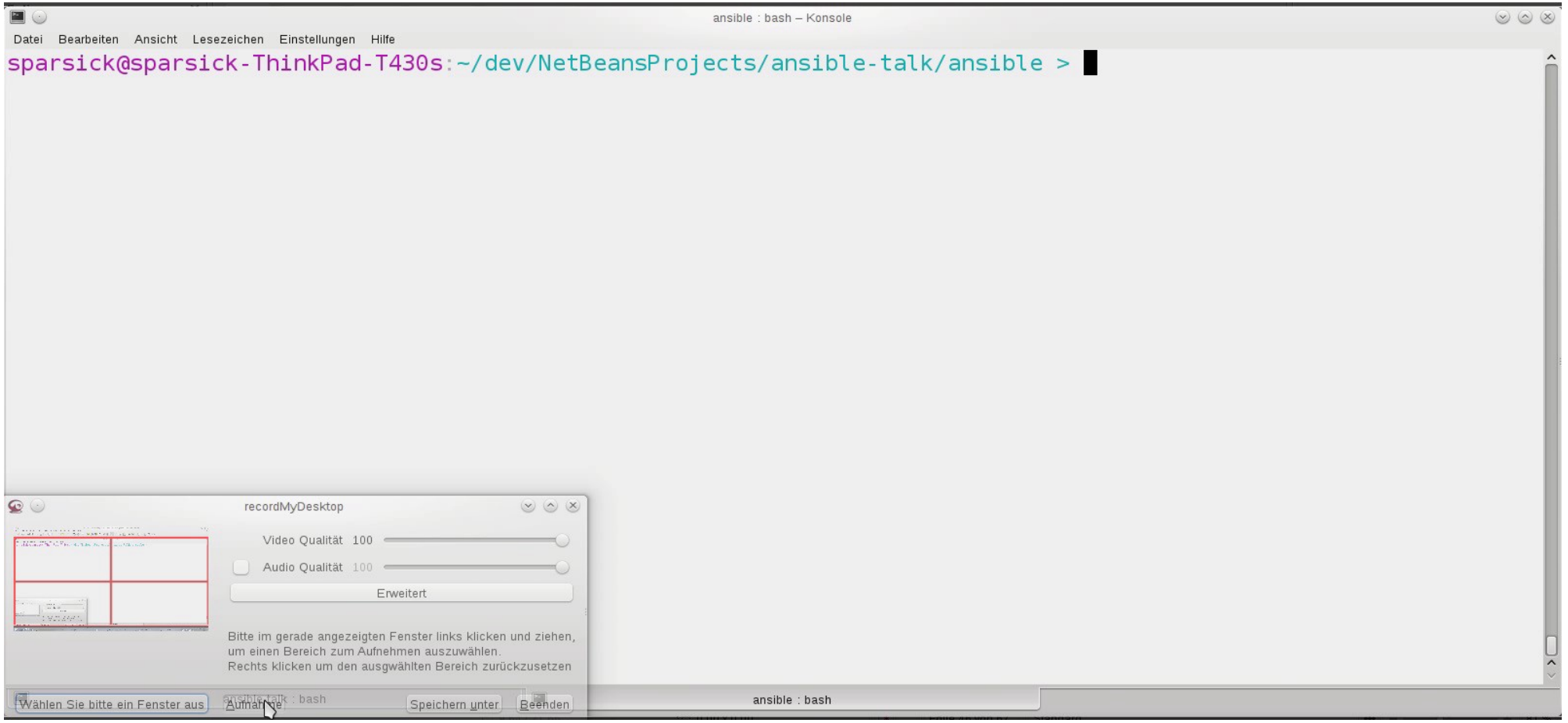
```
deploy-on-tomcat
├── defaults
│   └── main.yml
└── tasks
    ├── cleanup-webapp.yml
    ├── deploy-webapp.yml
    ├── main.yml
    ├── start-tomcat.yml
    └── stop-tomcat.yml
```

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

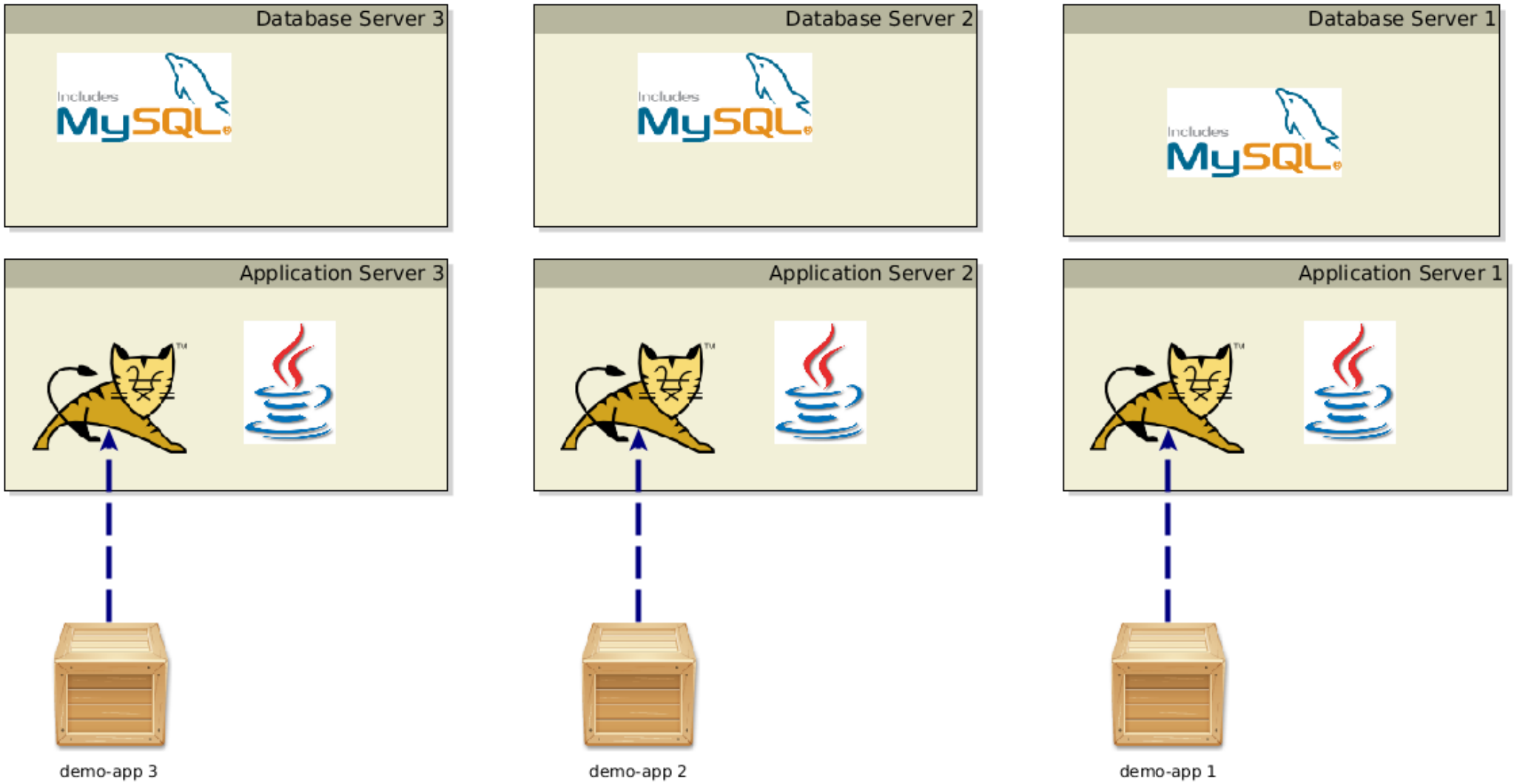
# Deploy Application Playbook



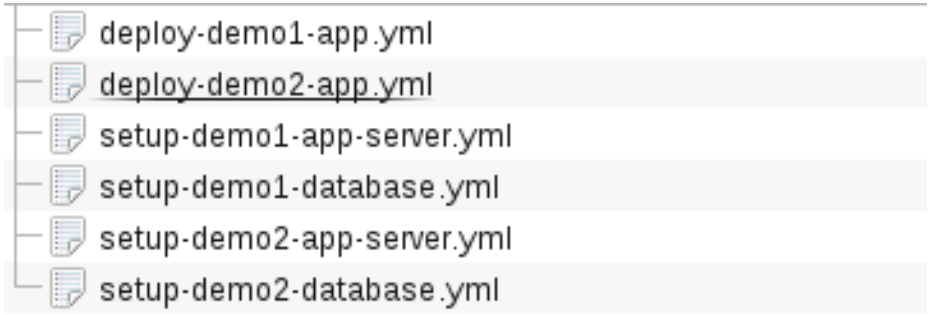
# Ad-hoc-Kommando



# 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?

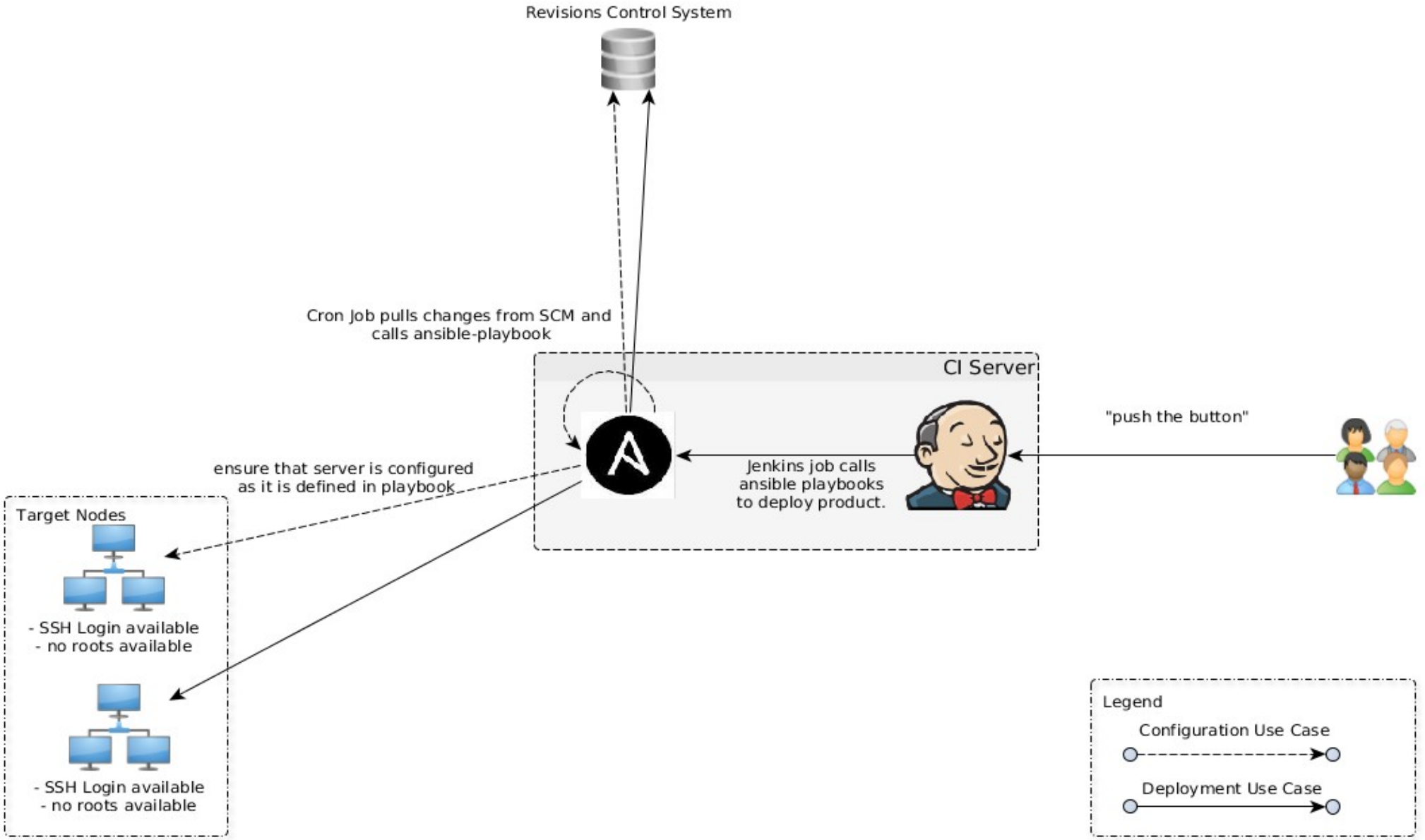

deploy-demo1-app.yml
deploy-demo2-app.yml
setup-demo1-app-server.yml
setup-demo1-database.yml
setup-demo2-app-server.yml
setup-demo2-database.yml

# Warum Roles?

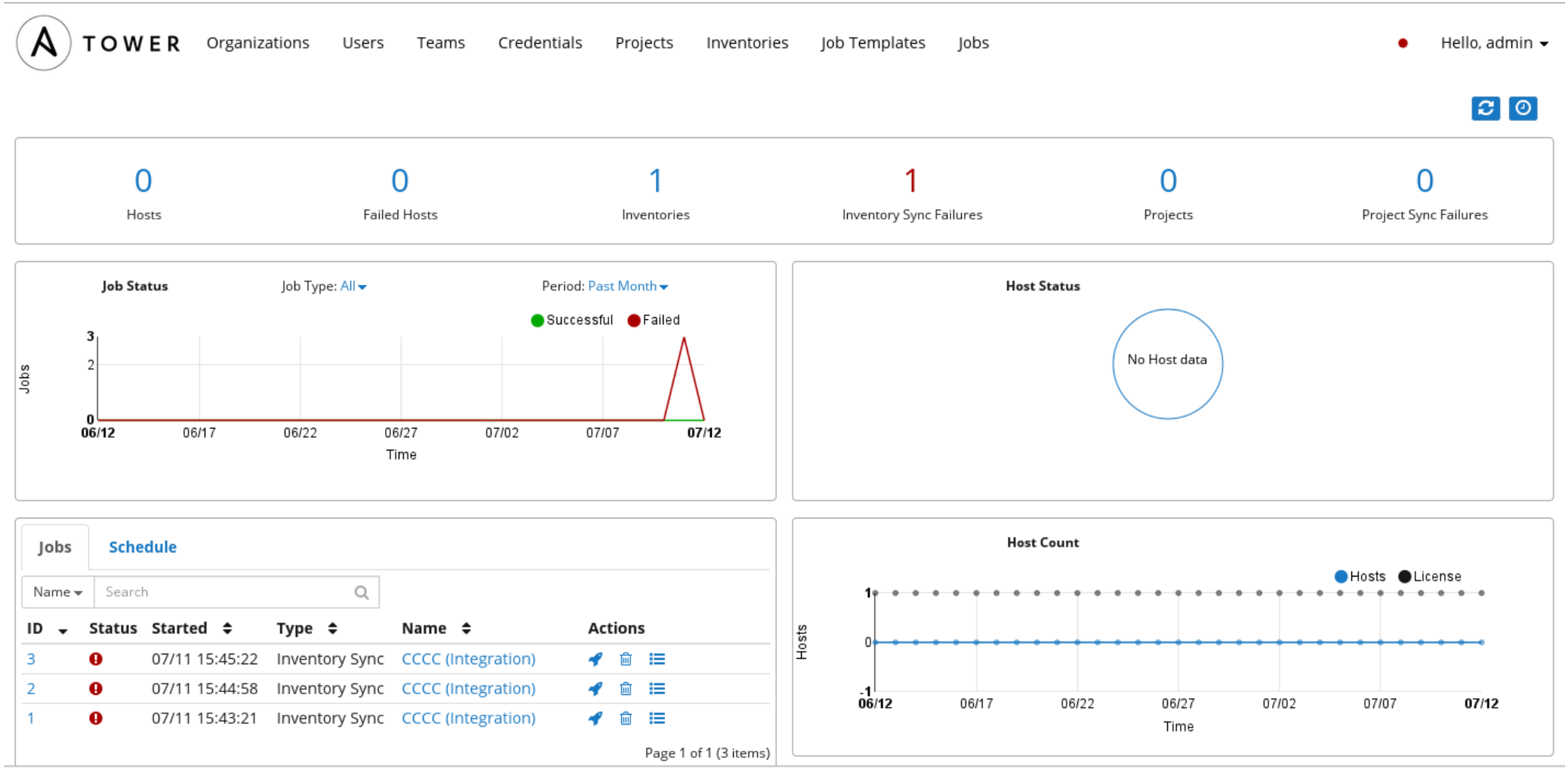
```
- hosts: application_server
  roles:
    - {role: deploy-on-tomcat, webapp_source_path: ./demo1-app-ansible-deploy-1.0-SNAPSHOT.war,
webapp_target_name: demo1 }
```

```
- hosts: application_server
  roles:
    - {role: deploy-on-tomcat, webapp_source_path: ./demo2-app-ansible-deploy-1.0-SNAPSHOT.war,
webapp_target_name: demo2 }
```

# Ansible Infrastruktur



# Ansible AWX / Tower

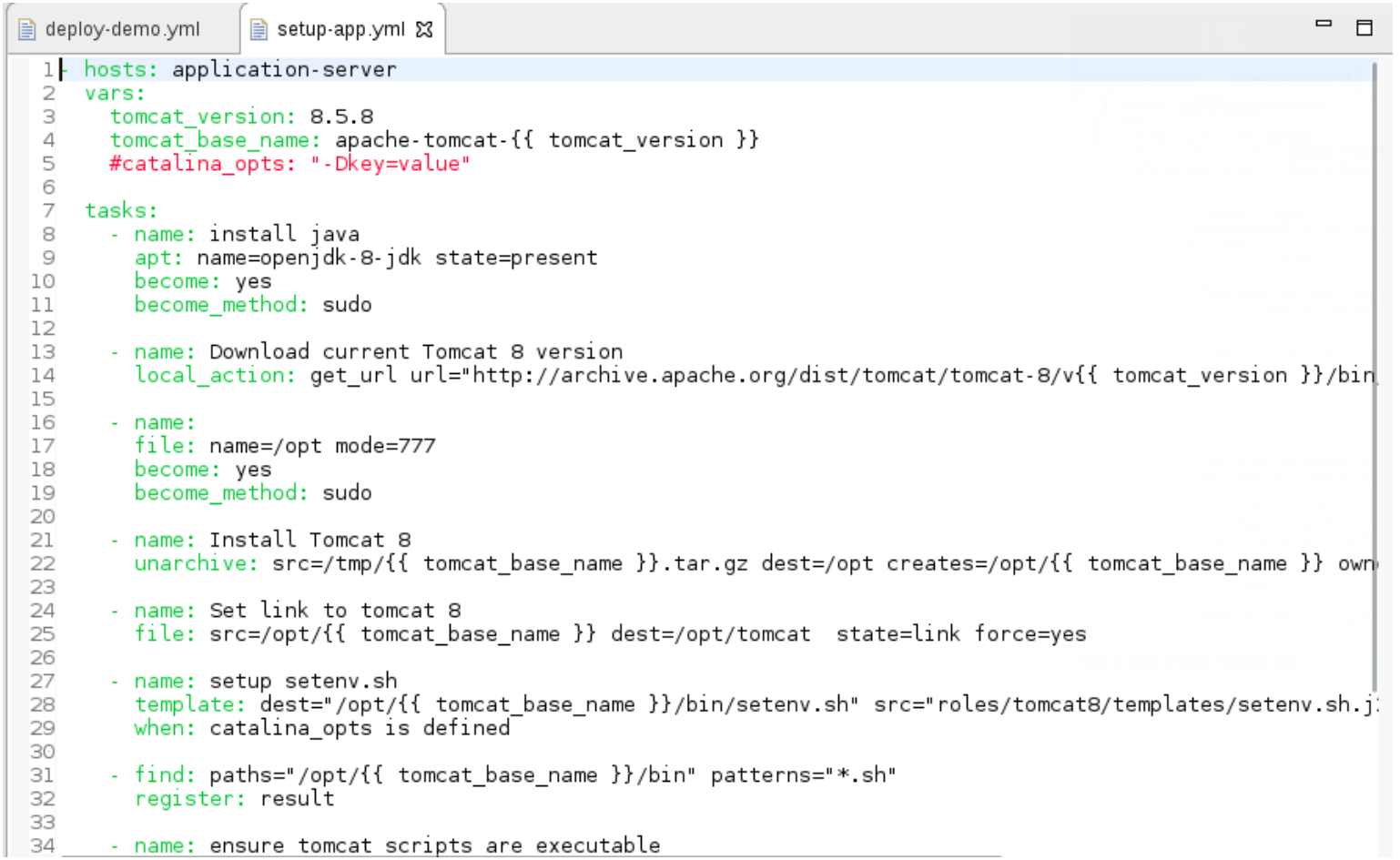


# IDE - Support

- Eclipse
- Intellji IDEA
- Netbeans
- Weitere Informationen unter https://jaxenter.de/ansible-intellij-netbeans-eclipse-51695

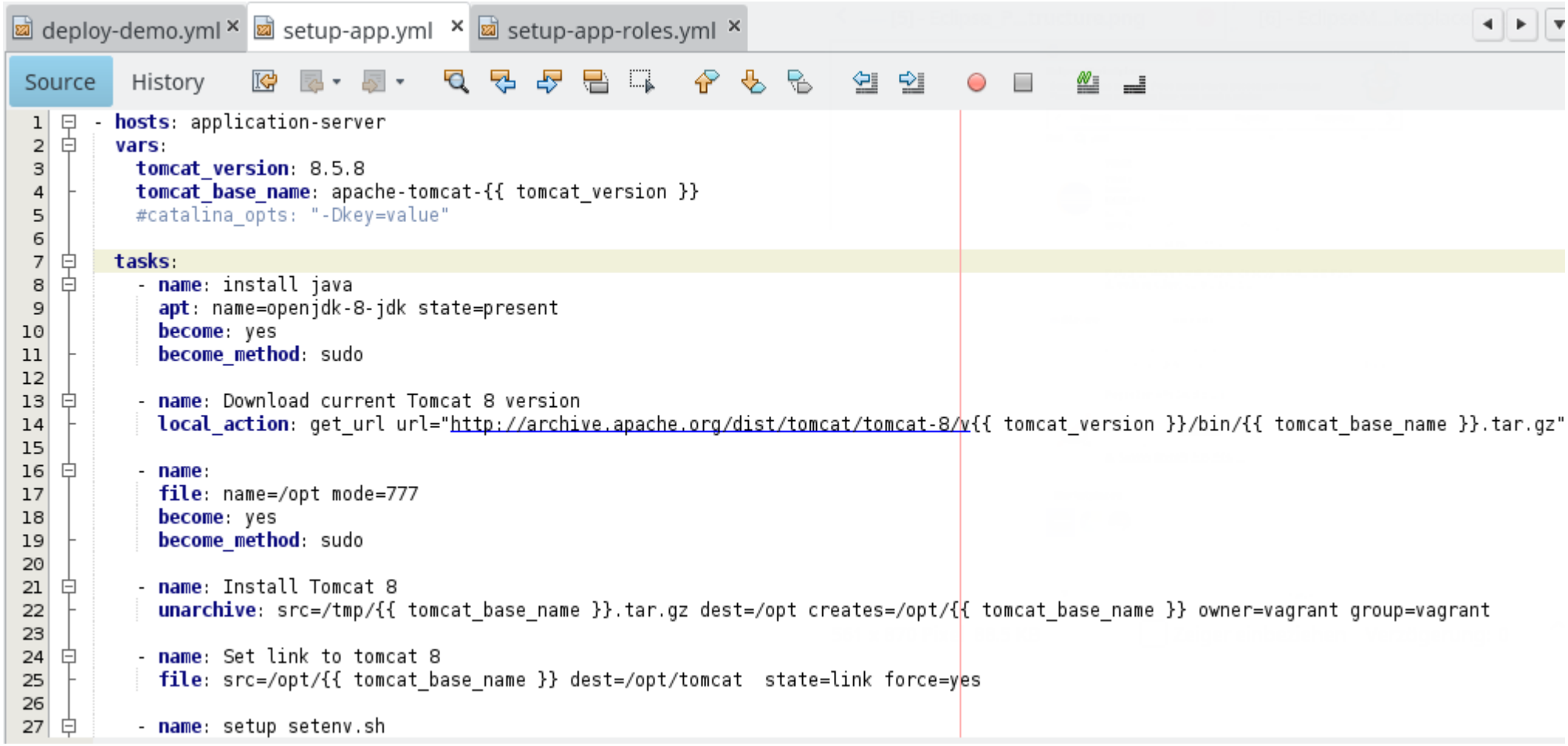
# IDE-Support - Eclipse

- YAML Support über Plugin *yEdit*



# IDE-Support - Netbeans

- YAML Support

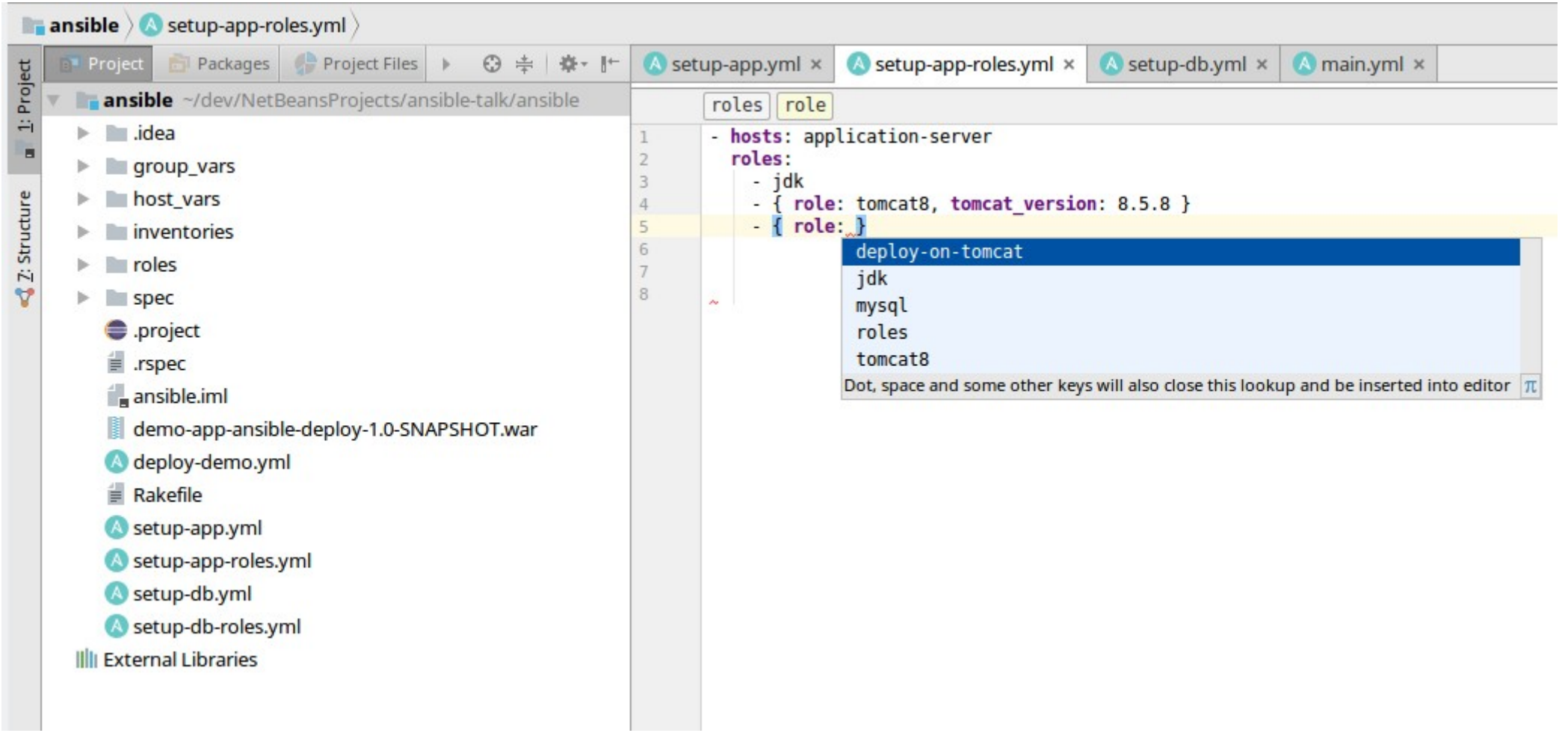


# IDE-Support - IntelliJ IDEA

- YAML Support von Haus aus
- Extra Plugin für Ansible (*YAML/Ansible support)*

# IDE-Support - IntelliJ IDEA

- Autovervollständigung für Roles

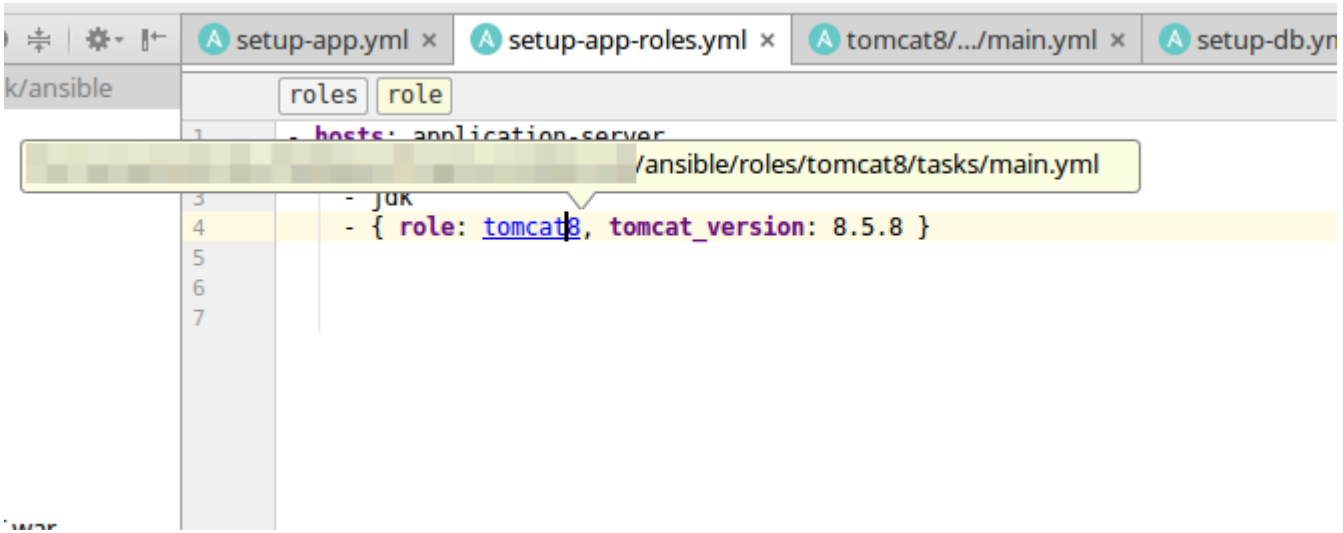


# IDE-Support - IntelliJ IDEA

- Anzeige der Dokumentation für Ansible Module

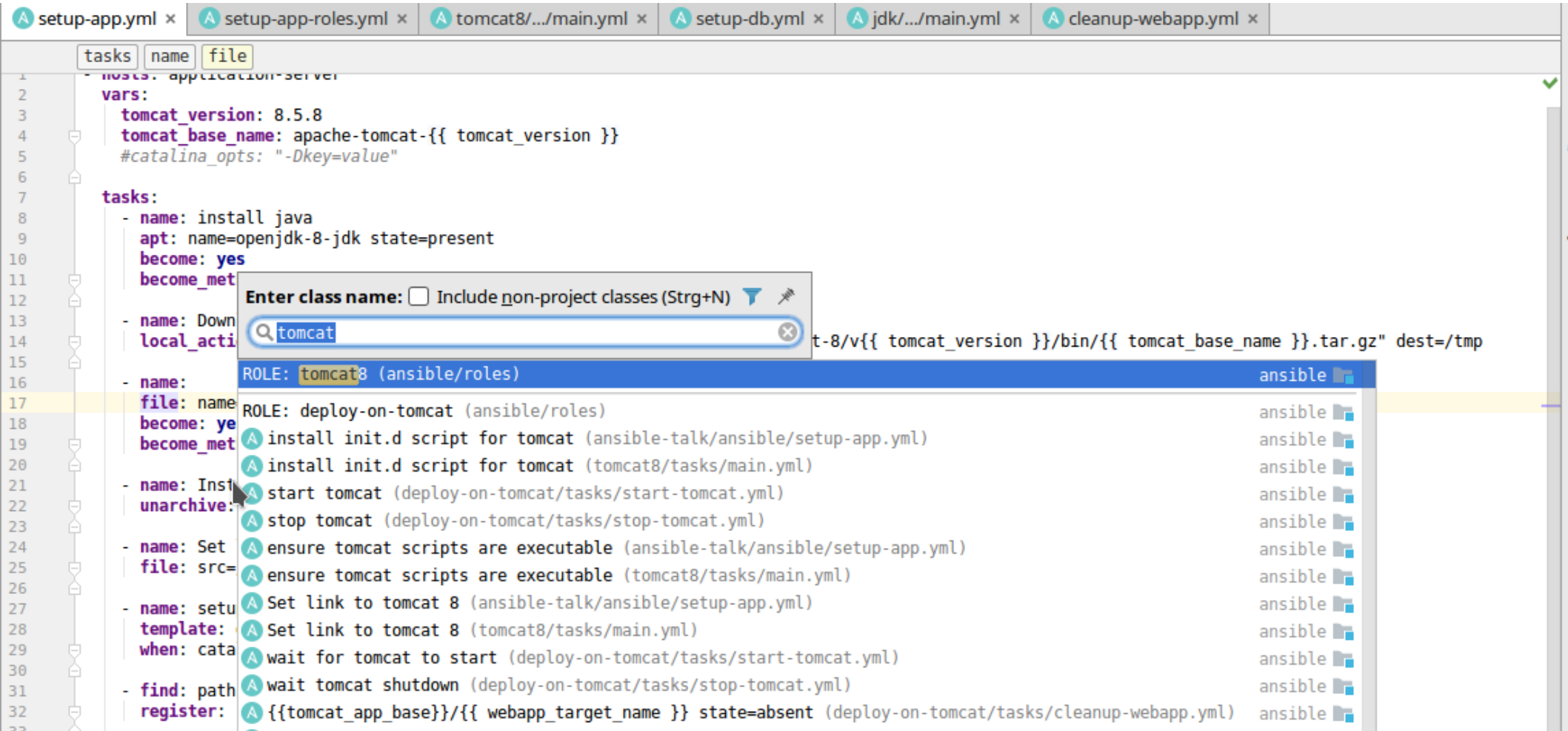
# IDE-Support - IntelliJ IDEA

- Direkte Navigation zu der Rollendefinition



# IDE-Support - IntelliJ IDEA

- Über *Navigate* Funktionalität direkt zu Rollen, Variablen und Task-Namen springen



# 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`
- ansible-lint
- Jenkins + Vagrant



Goss

# ansible-lint

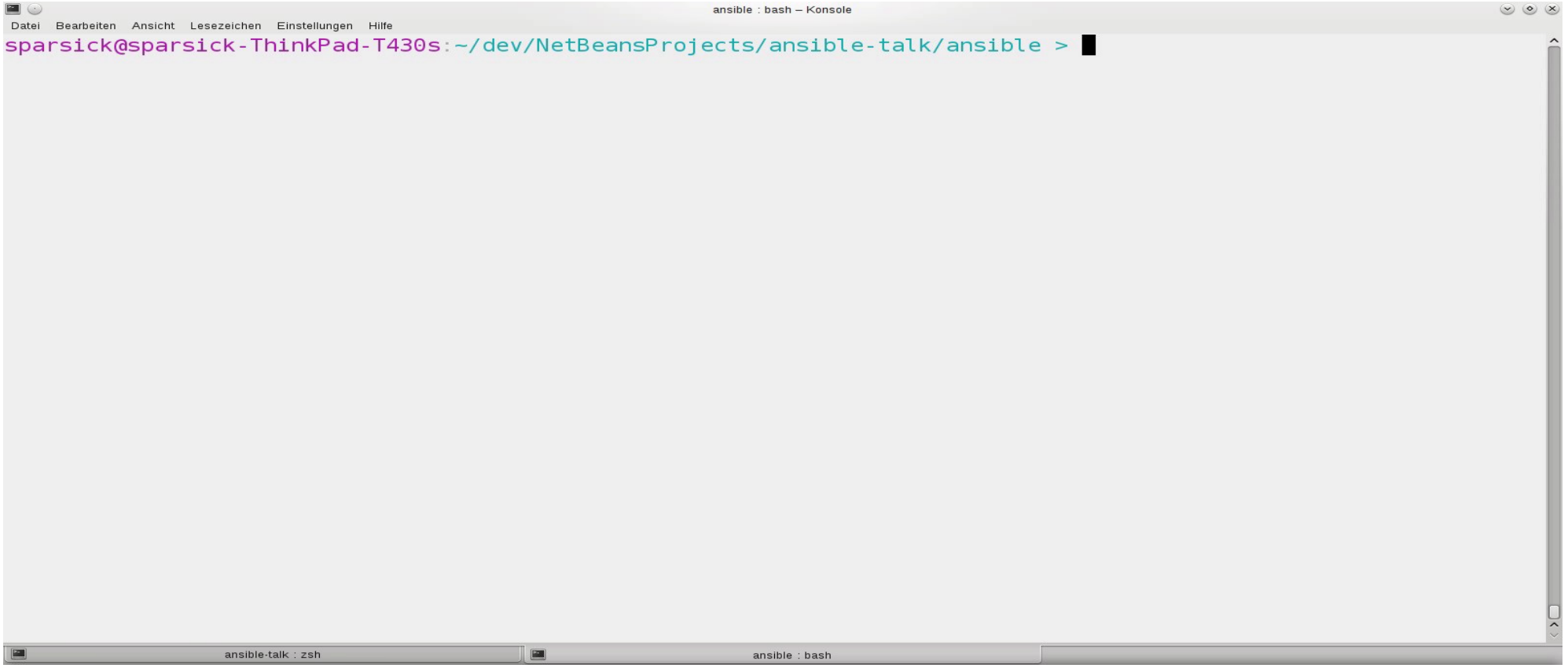
```
sparsick@sparsick-ThinkPad-T460s ~/dev/NetBeansProjects/ansible-talk/ansible master ansible-lint *.yml
[ANSIBLE0012] Commands should not change things if nothing needs doing
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/deploy-on-tomcat/tasks/deploy-webapp.yml:4
Task/Handler: create new backup

[ANSIBLE0002] Trailing whitespace
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/deploy-on-tomcat/tasks/deploy-webapp.yml:7


[ANSIBLE0009] Octal file permissions must contain leading zero
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/deploy-on-tomcat/tasks/deploy-webapp.yml:8
Task/Handler: copy webapp {{ webapp_source_path }} to {{ webapp_target_name }}

[ANSIBLE0002] Trailing whitespace
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/mysql/handlers/main.yml:4
  service: name=mysql state=restarted

[ANSIBLE0002] Trailing whitespace
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/mysql/tasks/main.yml:28
  mysql_user: name=dba password=g3h31m priv=*.*:ALL,GRANT state=present host=%

[ANSIBLE0009] Octal file permissions must contain leading zero
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/tomcat8/tasks/main.yml:4
Task/Handler: file name=/opt mode=777

[ANSIBLE0009] Octal file permissions must contain leading zero
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/tomcat8/tasks/main.yml:15
Task/Handler: setup setenv.sh

[ANSIBLE0011] All tasks should be named
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/tomcat8/tasks/main.yml:19
Task/Handler: find patterns=*.sh paths=/opt/{{ tomcat_base_name }}/bin

[ANSIBLE0009] Octal file permissions must contain leading zero
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/tomcat8/tasks/main.yml:22
Task/Handler: ensure tomcat scripts are executable
```

# ServerSpec Tests

```
require 'spec_helper'

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

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

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

```
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
```

# ServerSpec Tests

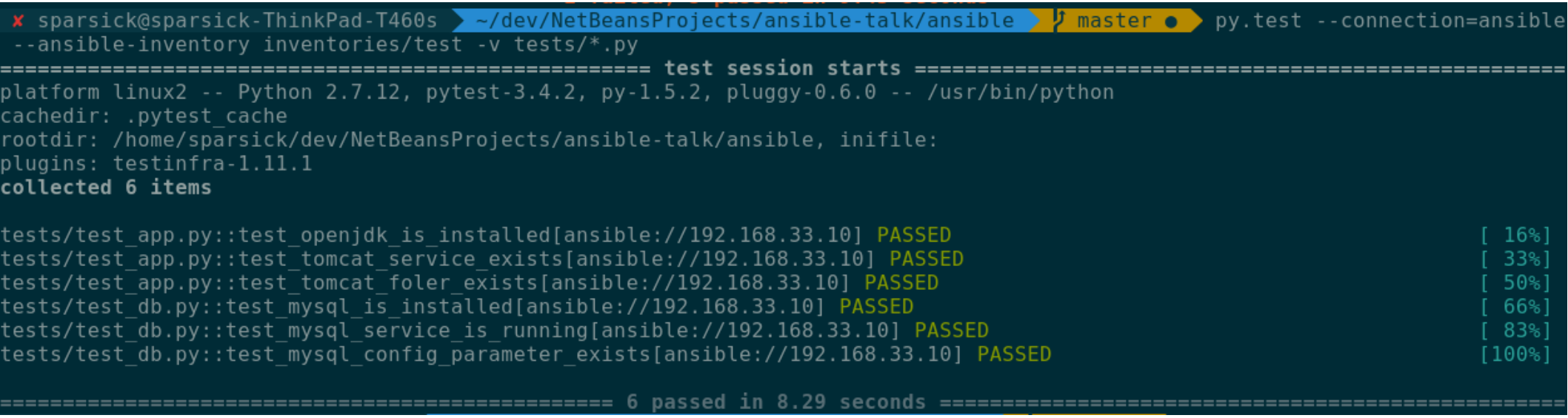


# testinfra

```python
def test_openjdk_is_installed(host):
    openjdk = host.package("openjdk-8-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_mysql_is_installed(host):
    mysql = host.package("mysql-server")
    assert mysql.is_installed

def test_mysql_service_is_running(host):
    mysql = host.service("mysql")
    assert mysql.is_enabled
    assert mysql.is_running

def test_mysql_config_parameter_exists(host):
    mysql_conf = host.file("/etc/mysql/mysql.conf.d/mysqld.cnf")
    assert mysql_conf.contains("bind-address = 0.0.0.0")
```

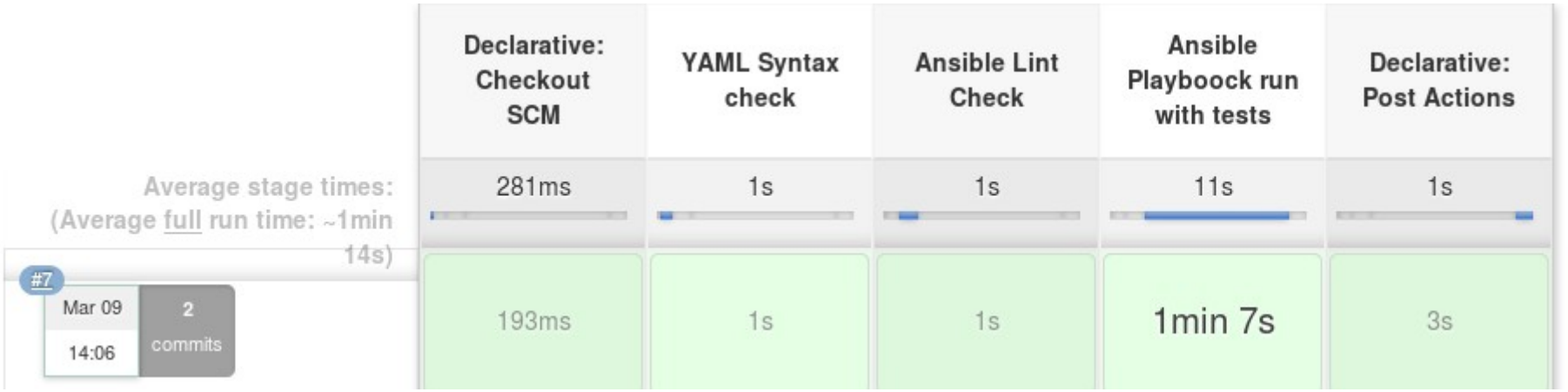
# testinfra


✘ sparsick@sparsick-ThinkPad-T460s ~/dev/NetBeansProjects/ansible-talk/ansible master ● py.test --connection=ansible
--ansible-inventory inventories/test -v tests/*.py
test session starts
platform linux2 -- Python 2.7.12, pytest-3.4.2, py-1.5.2, pluggy-0.6.0 -- /usr/bin/python
cachedir: .pytest_cache
rootdir: /home/sparsick/dev/NetBeansProjects/ansible-talk/ansible, inifile:
plugins: testinfra-1.11.1
collected 6 items
tests/test_app.py::test_openjdk_is_installed[ansible://192.168.33.10] PASSED [ 16%]
tests/test_app.py::test_tomcat_service_exists[ansible://192.168.33.10] PASSED [ 33%]
tests/test_app.py::test_tomcat_foler_exists[ansible://192.168.33.10] PASSED [ 50%]
tests/test_db.py::test_mysql_is_installed[ansible://192.168.33.10] PASSED [ 66%]
tests/test_db.py::test_mysql_service_is_running[ansible://192.168.33.10] PASSED [ 83%]
tests/test_db.py::test_mysql_config_parameter_exists[ansible://192.168.33.10] PASSED [100%]
6 passed in 8.29 seconds

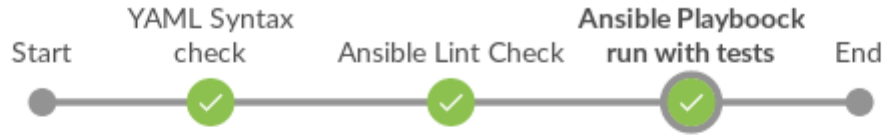
# Ansible QA Jenkins Pipeline

```
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

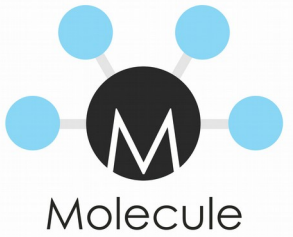



**Ansible Playboock run with tests - 1m 8s**

| | | |
|---|---|---|
| ✓ | 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 |

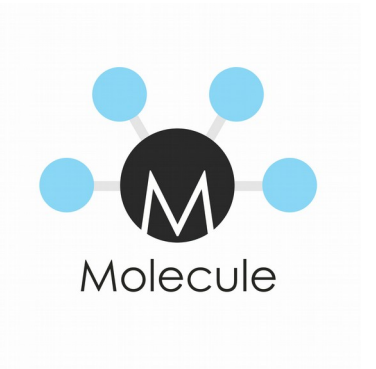
Geht es nicht einfacher?

# Molecule



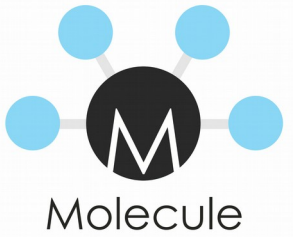

- Spezialisiert für Ansible Roles
- Wrapper um andere Werkzeuge, um komplette Test Szenarien aufzubauen
  - Driver Provider: Docker (default), Vagrant, Azure, EC2
  - Lint Provider: yamllint (default), ansible-lint, flake8 (for test code)
  - Verifier framework: TestInfra (default)

# Molecule



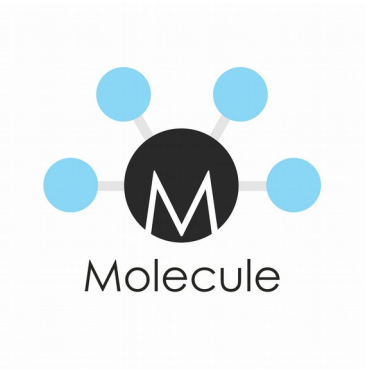

```
.
└── tomcat
    ├── defaults
    │   └── main.yml
    ├── handlers
    │   └── main.yml
    ├── meta
    │   └── main.yml
    ├── molecule
    │   └── default
    │       ├── Dockerfile.j2
    │       ├── molecule.yml
    │       ├── playbook.yml
    │       └── tests
    │           ├── test_default.py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

# Molecule



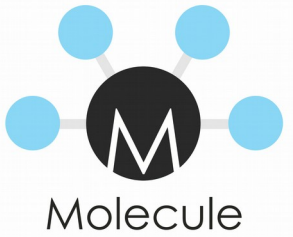

```
roles (master*) » cat tomcat/molecule/default/molecule.yml
---
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: instance
    image: ubuntu:18.04
provisioner:
  name: ansible
  lint:
    name: ansible-lint
verifier:
  name: testinfra
  lint:
    name: flake8
```

# Molecule




```
--> Test matrix

└── default
    ├── lint
    ├── cleanup
    ├── destroy
    ├── dependency
    ├── syntax
    ├── create
    ├── prepare
    ├── converge
    ├── idempotence
    ├── side_effect
    ├── verify
    ├── cleanup
    └── destroy
```

# Molecule




tomcat : zsh — Konsole

Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe

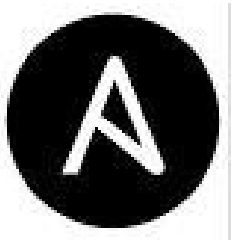
```
sparsick@sparsick-ThinkPad-T460s: /home/sparsick/dev/workspace/infra-testing-talk/infrastructure-as-code-testing/ansible/roles/tomcat git:(master) ✗
→
```

site-packages : zsh | bin : zsh | tomcat : zsh | tomcat : zsh

Entdecken | Testen v... | dev — D... | Thievery... | diskstati... | apps — ... | TOSHIB... | Server ni... | Downloa... | tomcat : ... | ci-server... | System... | Sitzung -... | SimpleSc... | Welcom... | Notes - Z... | 08:35

# 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

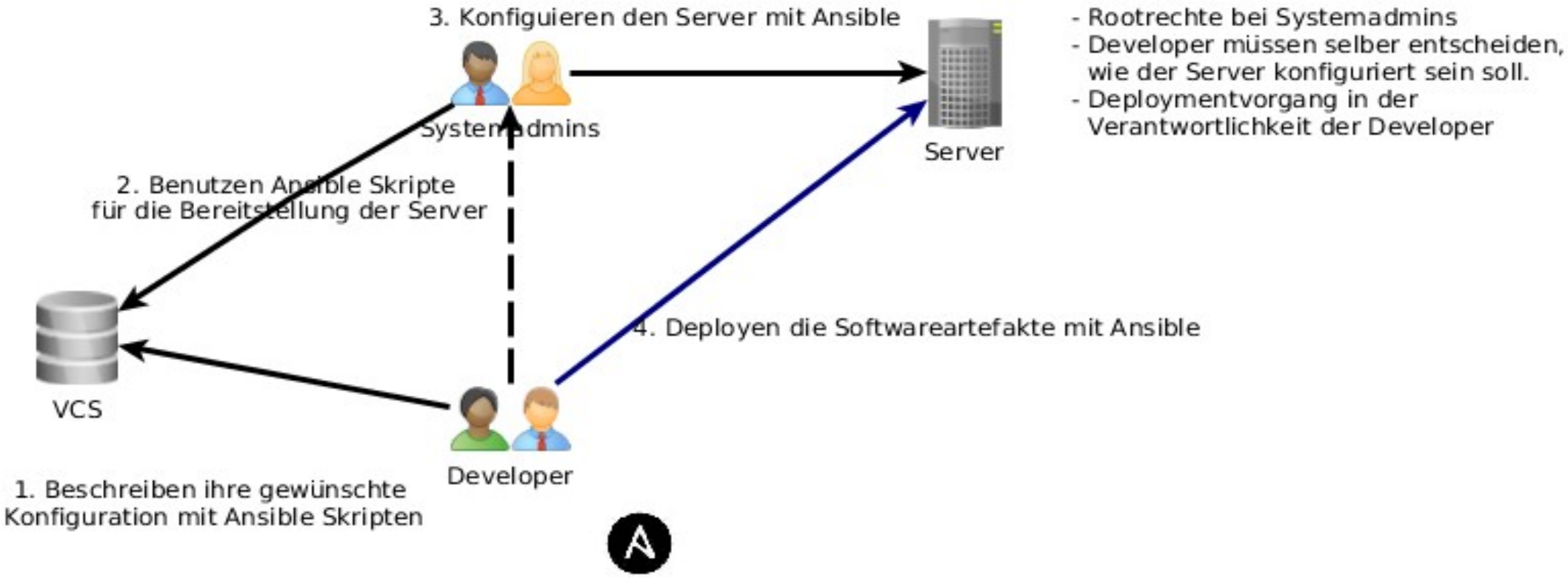

```yaml
- hosts: 192.168.33.10
  become: yes
  become_method: sudo
  tasks:
    - name: Ensure Python modules
      apt:
        name: python-apt

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

    - name: install nodejs package
      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

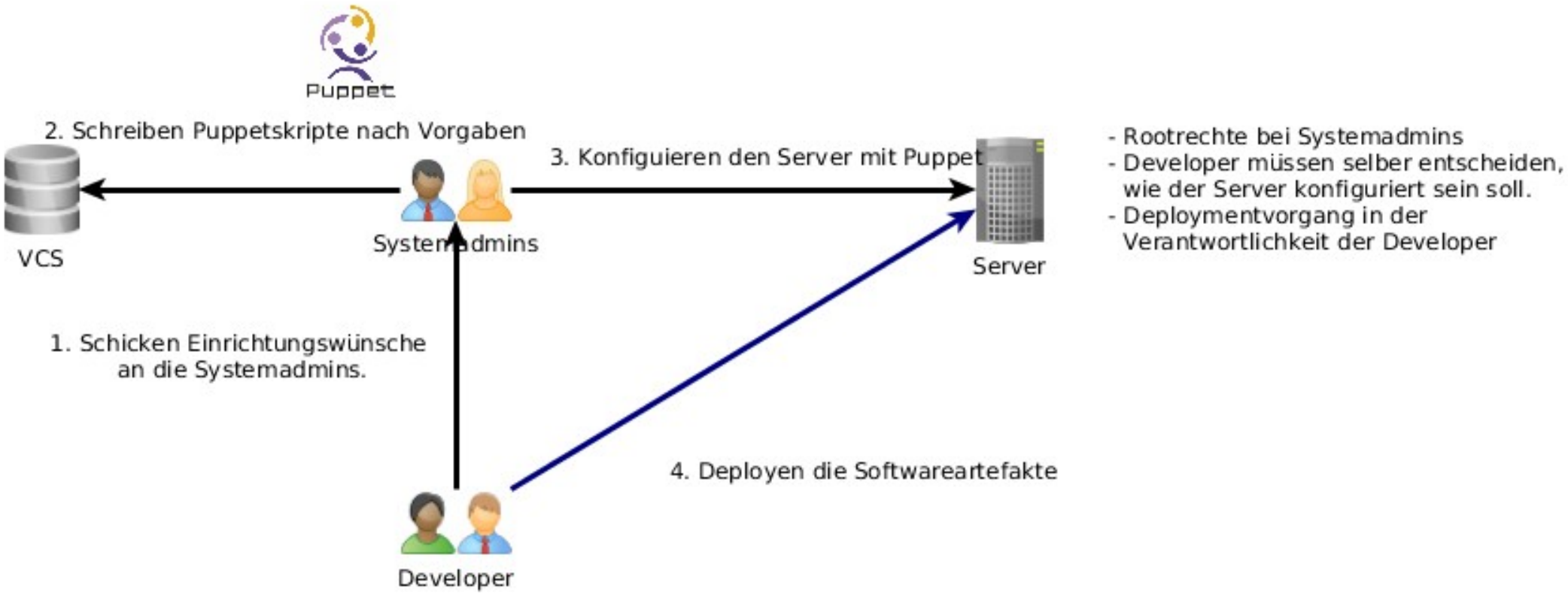
# Systemkonfiguration für Entwickler

## Lösungidee mit Ansible



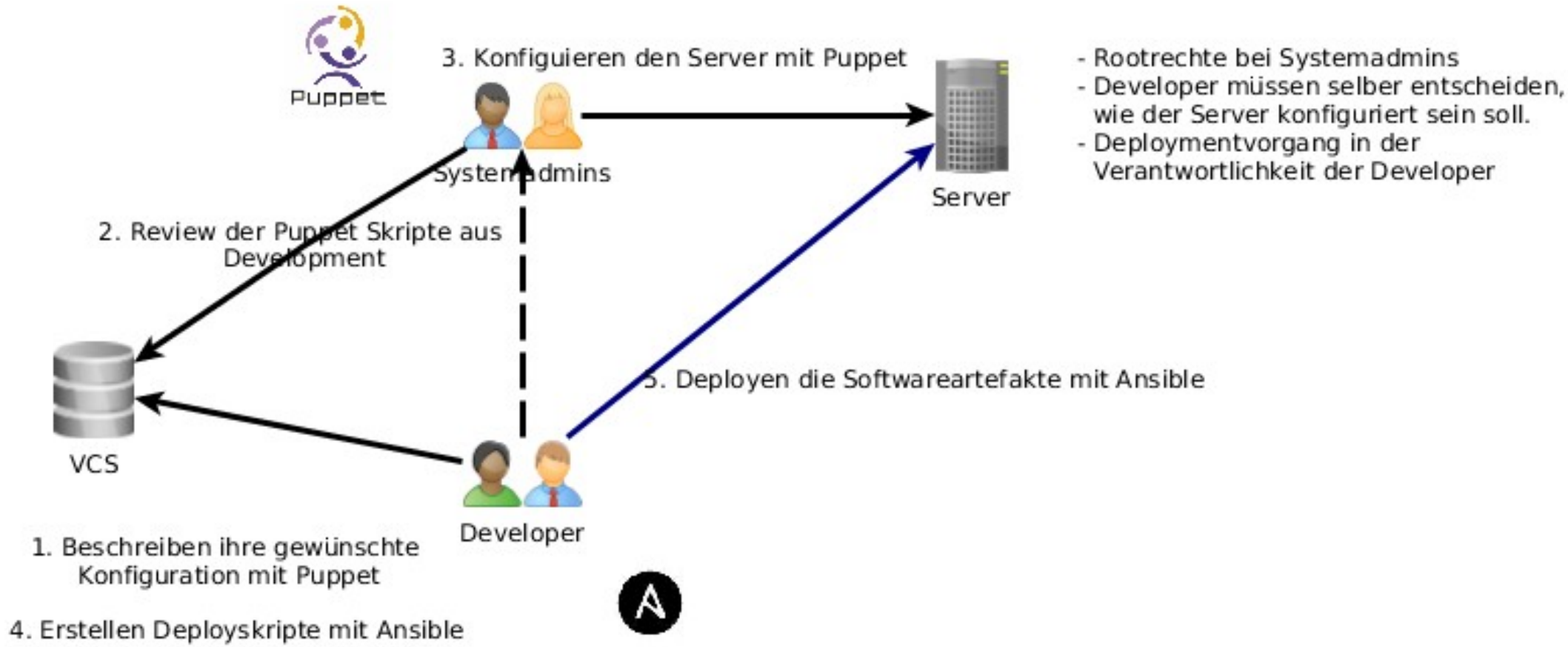
# Systemkonfiguration für Entwickler

## Variante - Prozess zwischen Development und Operation



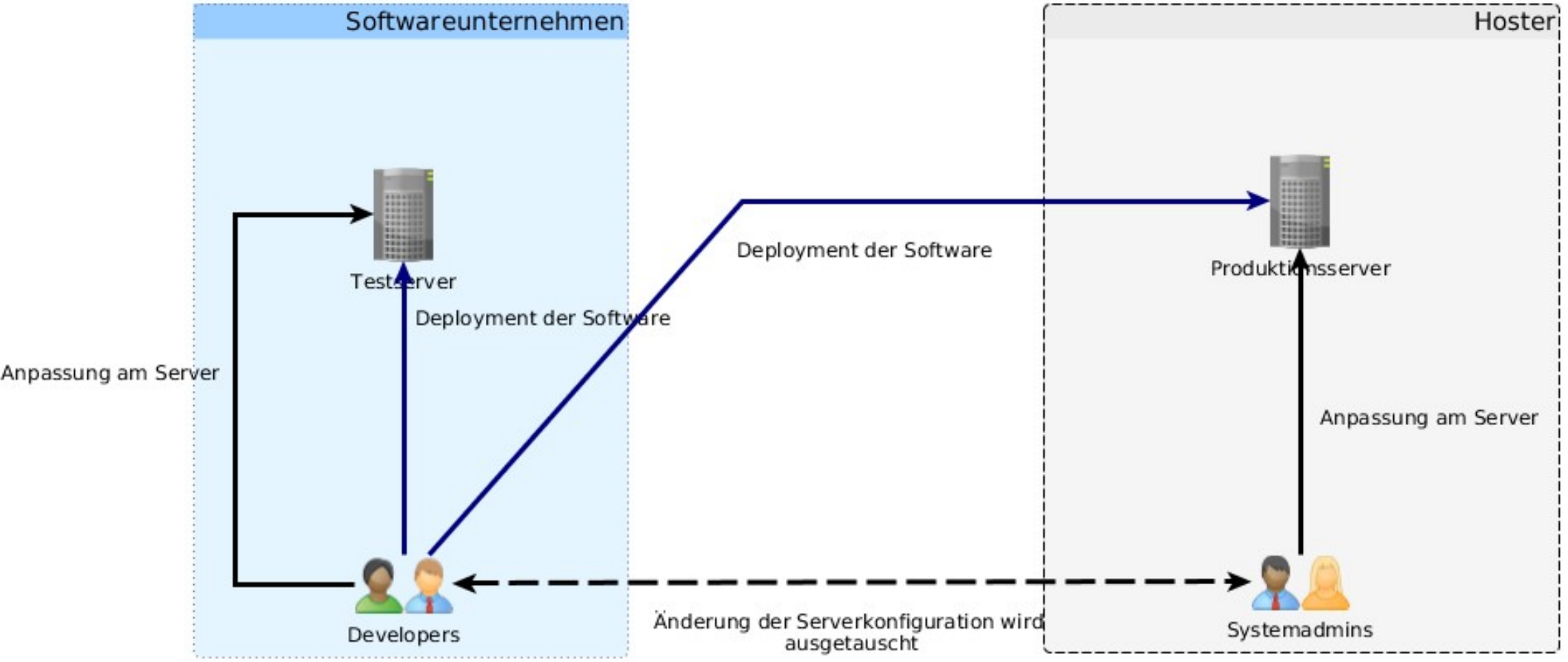
# Systemkonfiguration für Entwickler

## Lösungsvariante



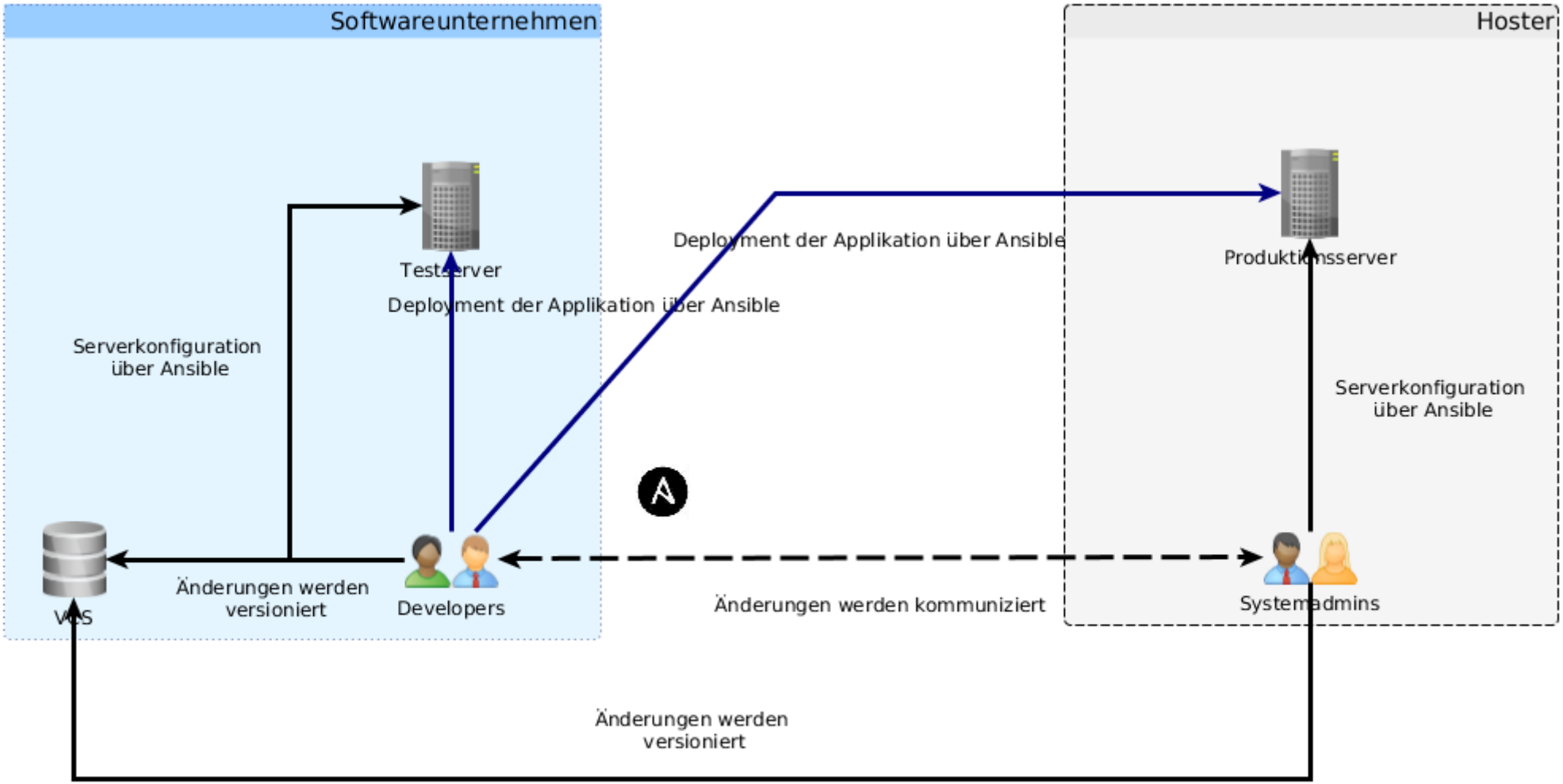
# Systemkonfiguration für Entwickler

## Produktionsserver sind beim externen Hoster



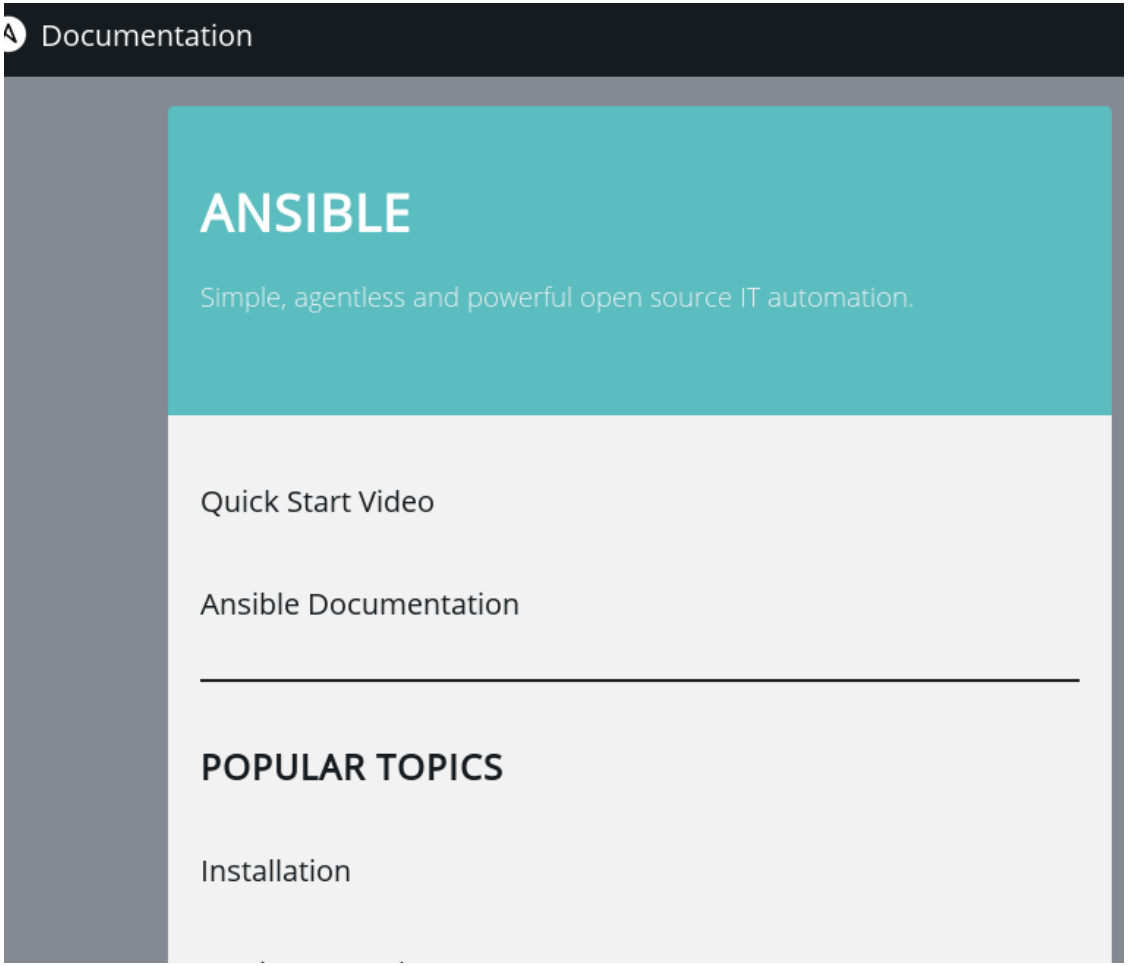
# Systemkonfiguration für Entwickler

## Lösungsidee



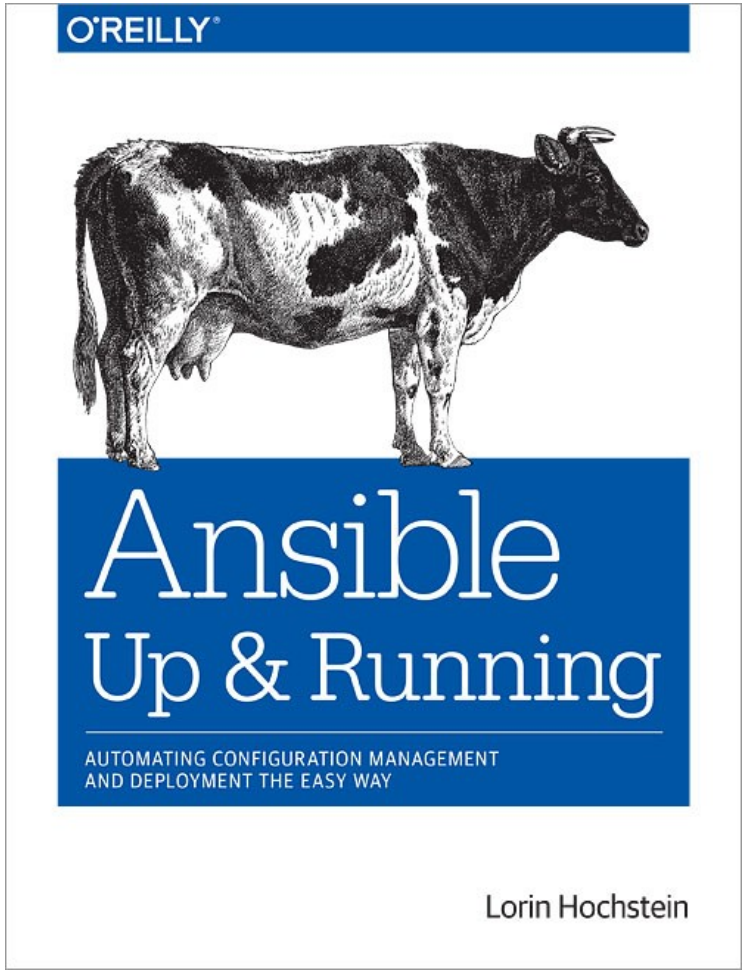

- Hoster verantwortlich für die Systemkonfiguration
- Softwareunternehmen verantwortlich für das Deployment
- Synchronisation zwischen Testserver und Produktionsserver wird vereinfacht

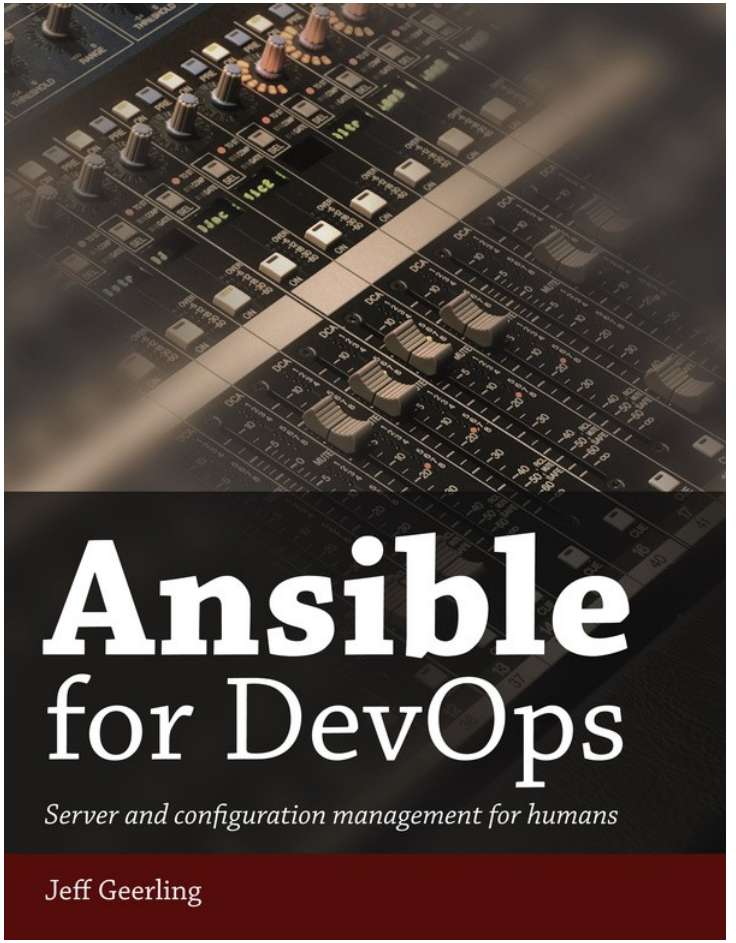
# Weitere Informationen




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