

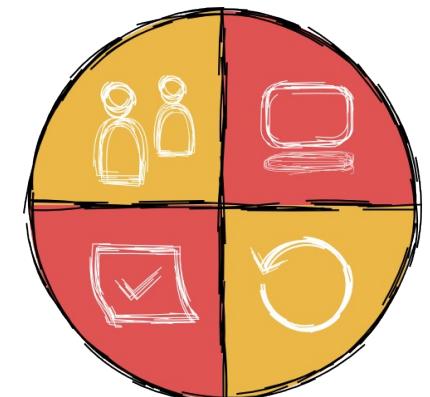
JUG Schweiz Luzern, 15.10.2019

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



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.“

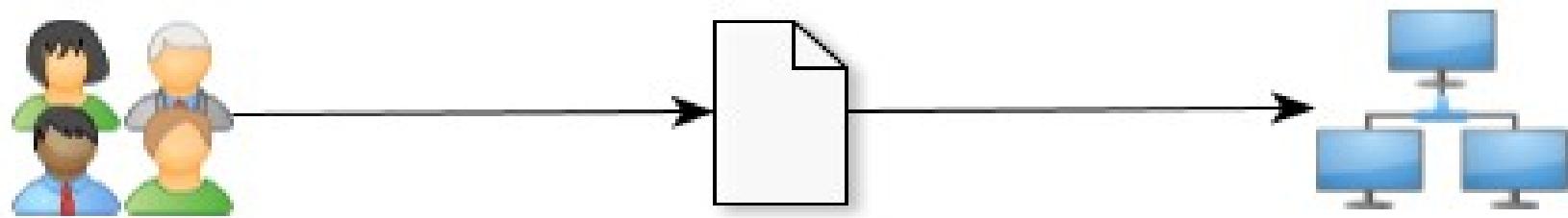
<https://www.projektmagazin.de/glossarterm/konfigurationsmanagement>

Konfigurationsmanagement (KM)

- Softwarekonfiguration
- Hardwarekonfiguration
- Dienstleistungskonfiguration
- Systemkonfiguration

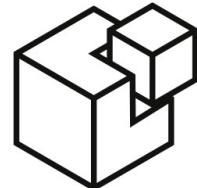
Systemkonfiguration

- „Infrastructure As Code“



Systemkonfiguration

- „Infrastructure As Code“



SALTSTACK

CFEngine



CHEF™



Puppet

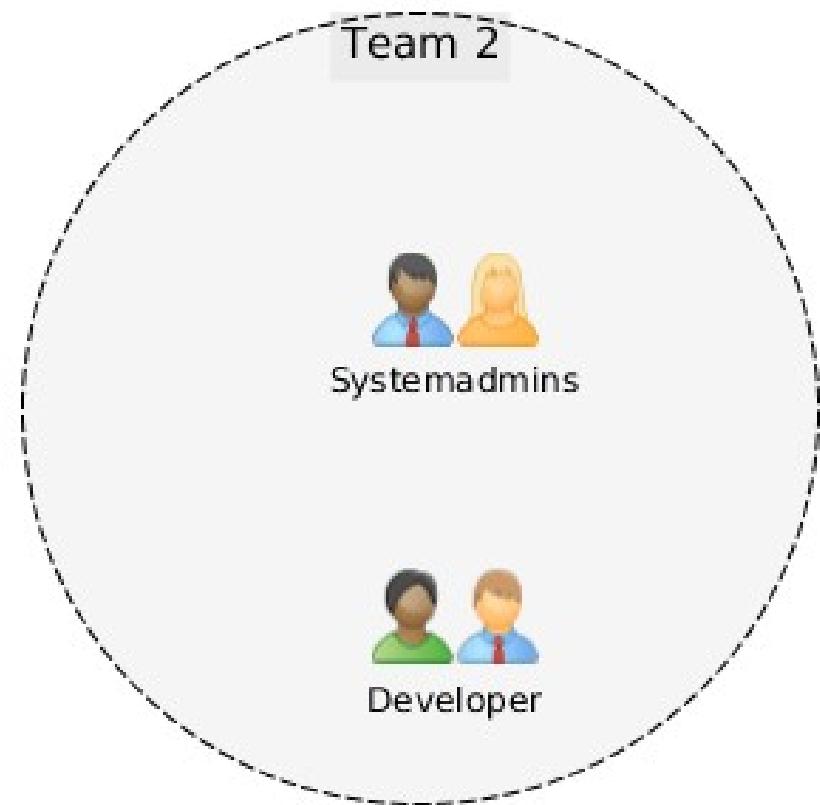
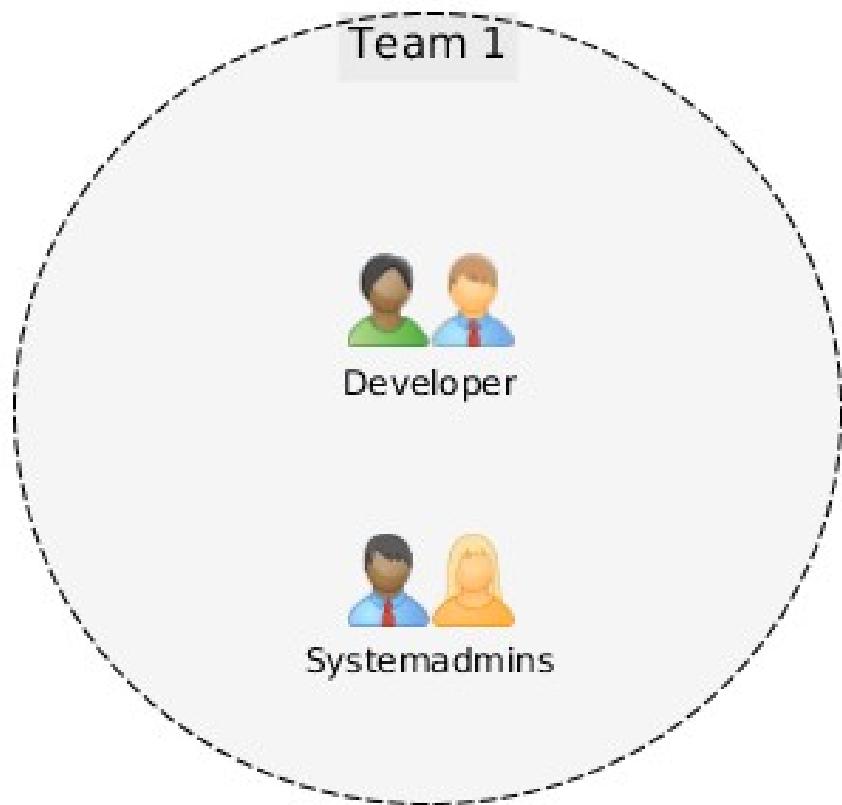


Ansible

Warum ist es für Entwickler interessant?

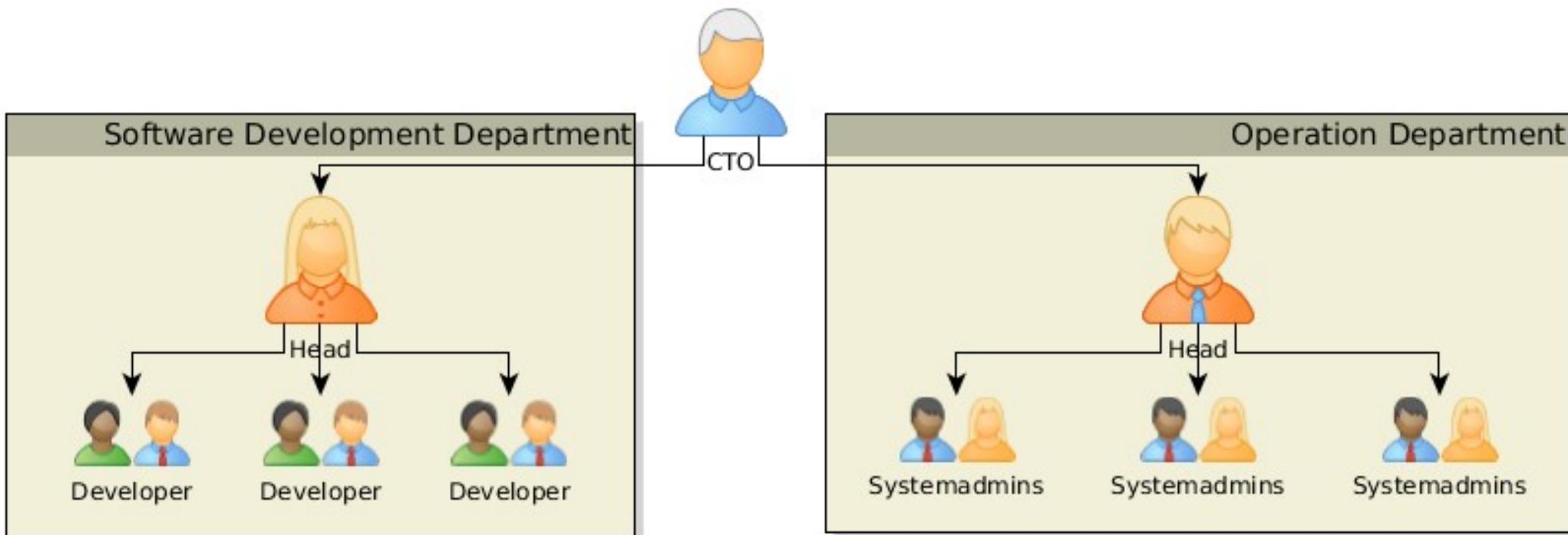
Systemkonfiguration für Entwickler

Organisatorische Ausgangslage
Wunsch



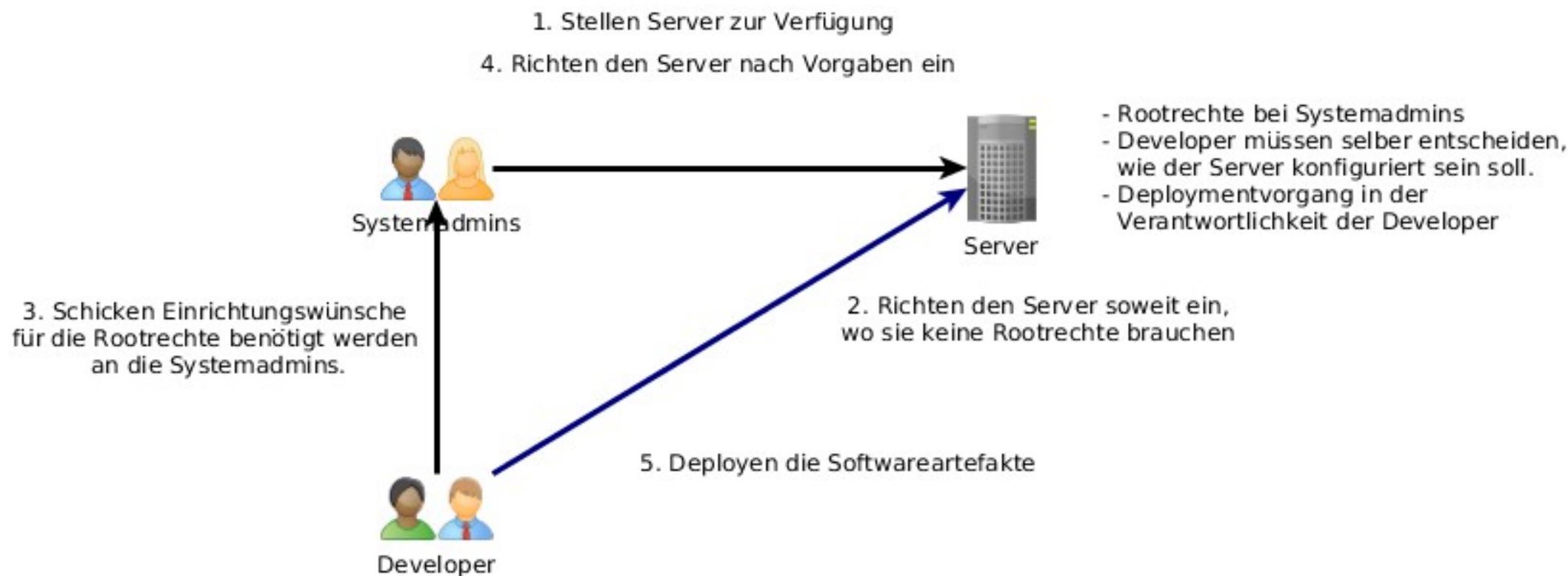
Systemkonfiguration für Entwickler

Organisatorische Ausgangslage
Realität



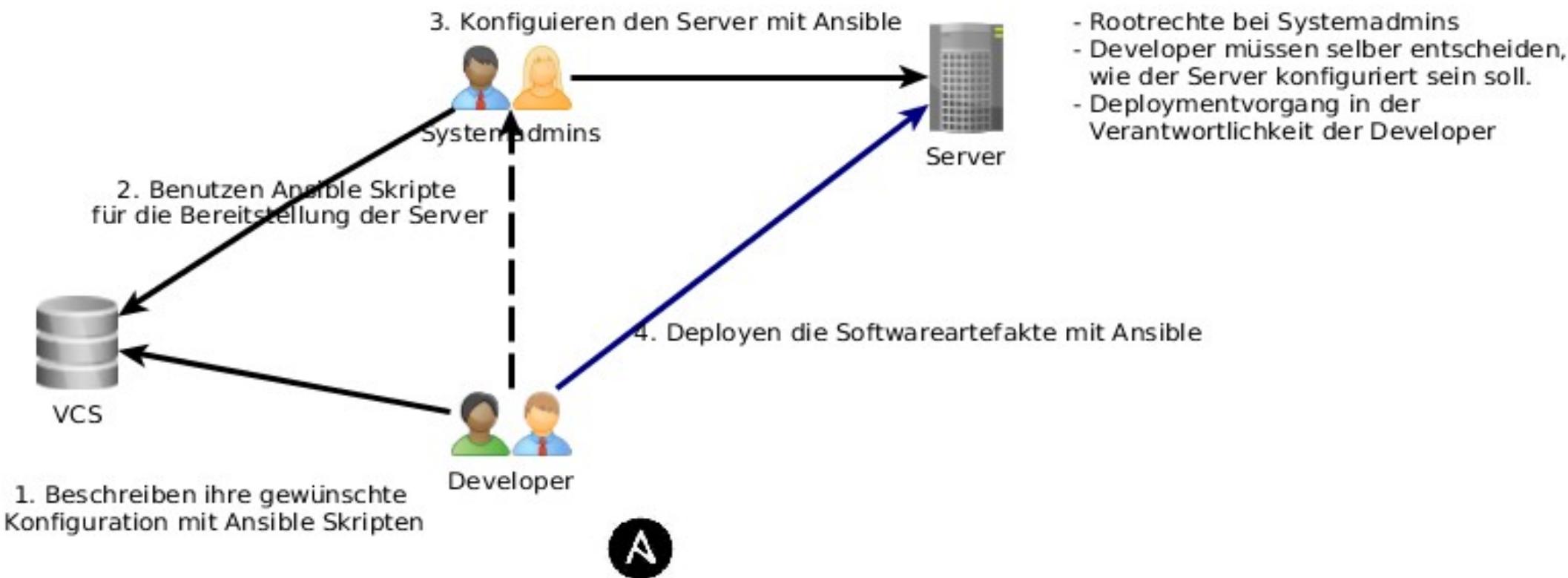
Systemkonfiguration für Entwickler

Prozess zwischen Development und Operation



Systemkonfiguration für Entwickler

Lösungidee mit Ansible



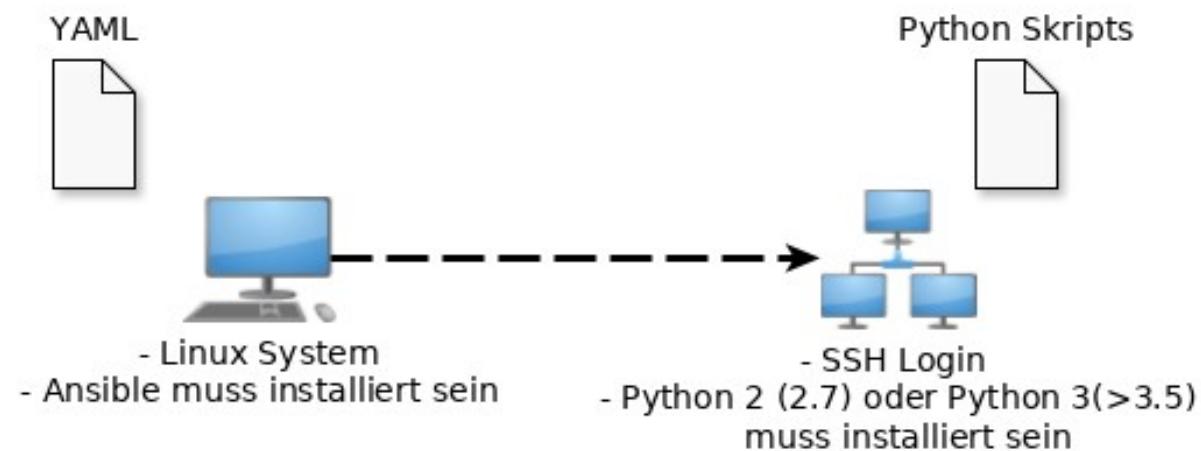
Einführung in Ansible

Ansible

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



Funktionsweise



Exkurs: YAML

YAML

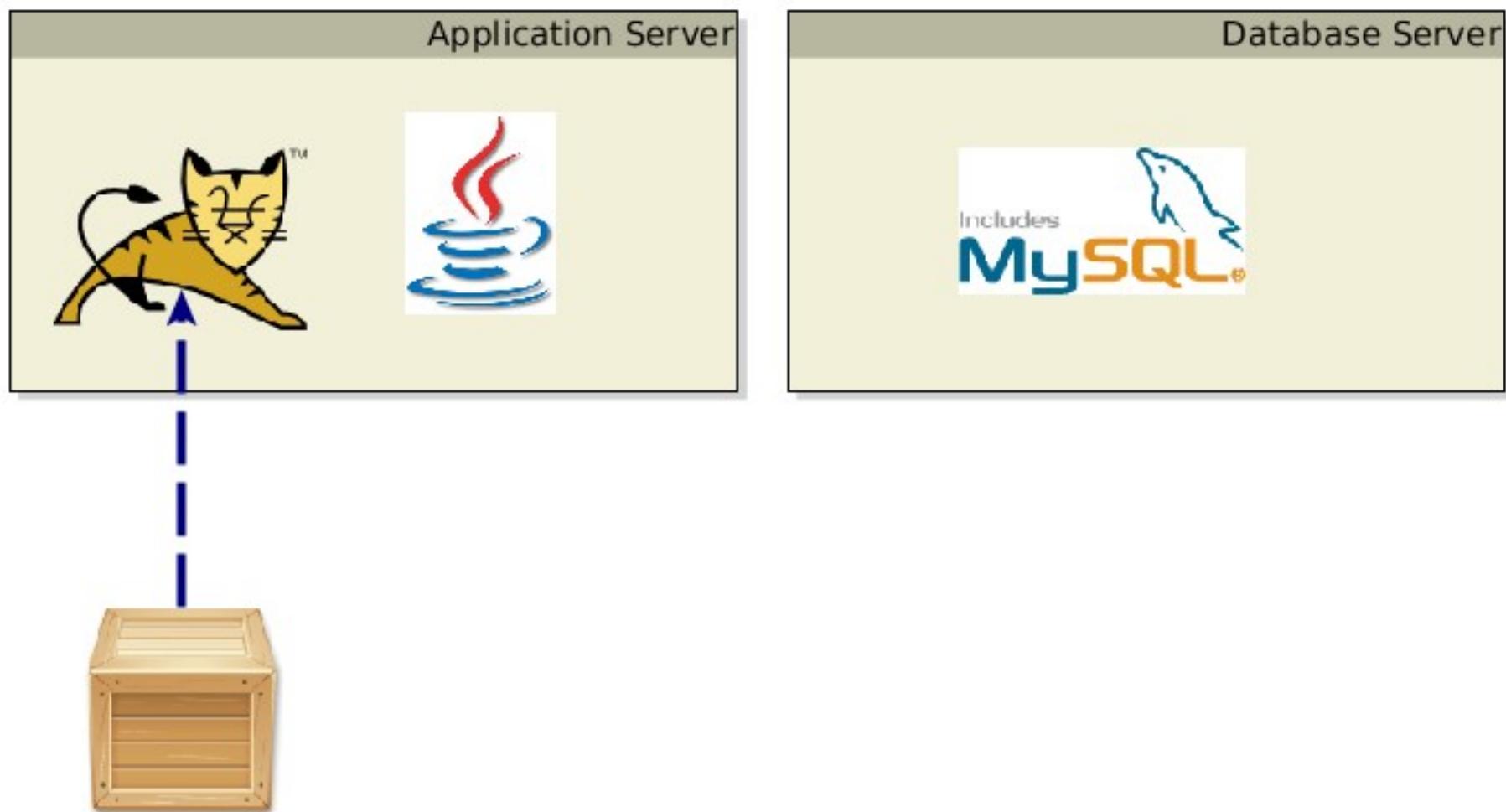
```
---
```

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

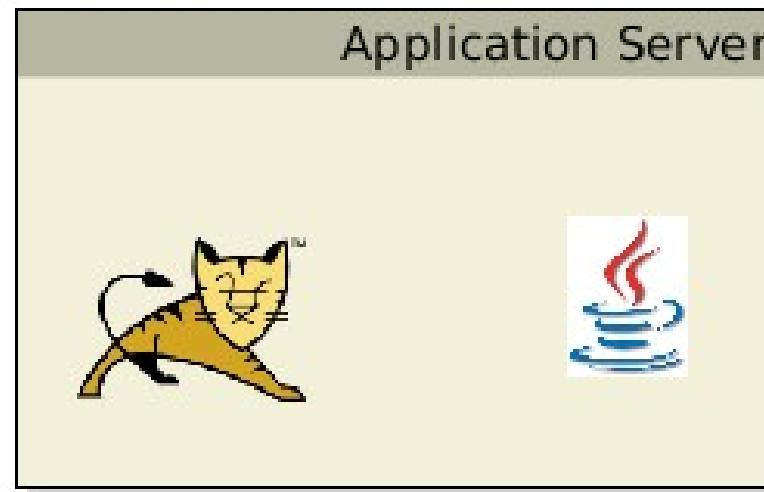
JSON

```
{
  "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

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



```
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

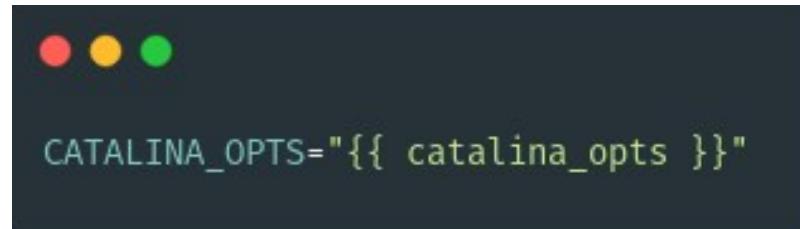
- find: paths="/opt/{{ tomcat_base_name }}/bin" patterns="*.sh"
  register: result

- 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`



A screenshot of a terminal window with a dark background. In the top left corner, there are three small colored dots: red, yellow, and green. Below them, the text `CATALINA_OPTS="{{ catalina_opts }}"` is displayed in white.

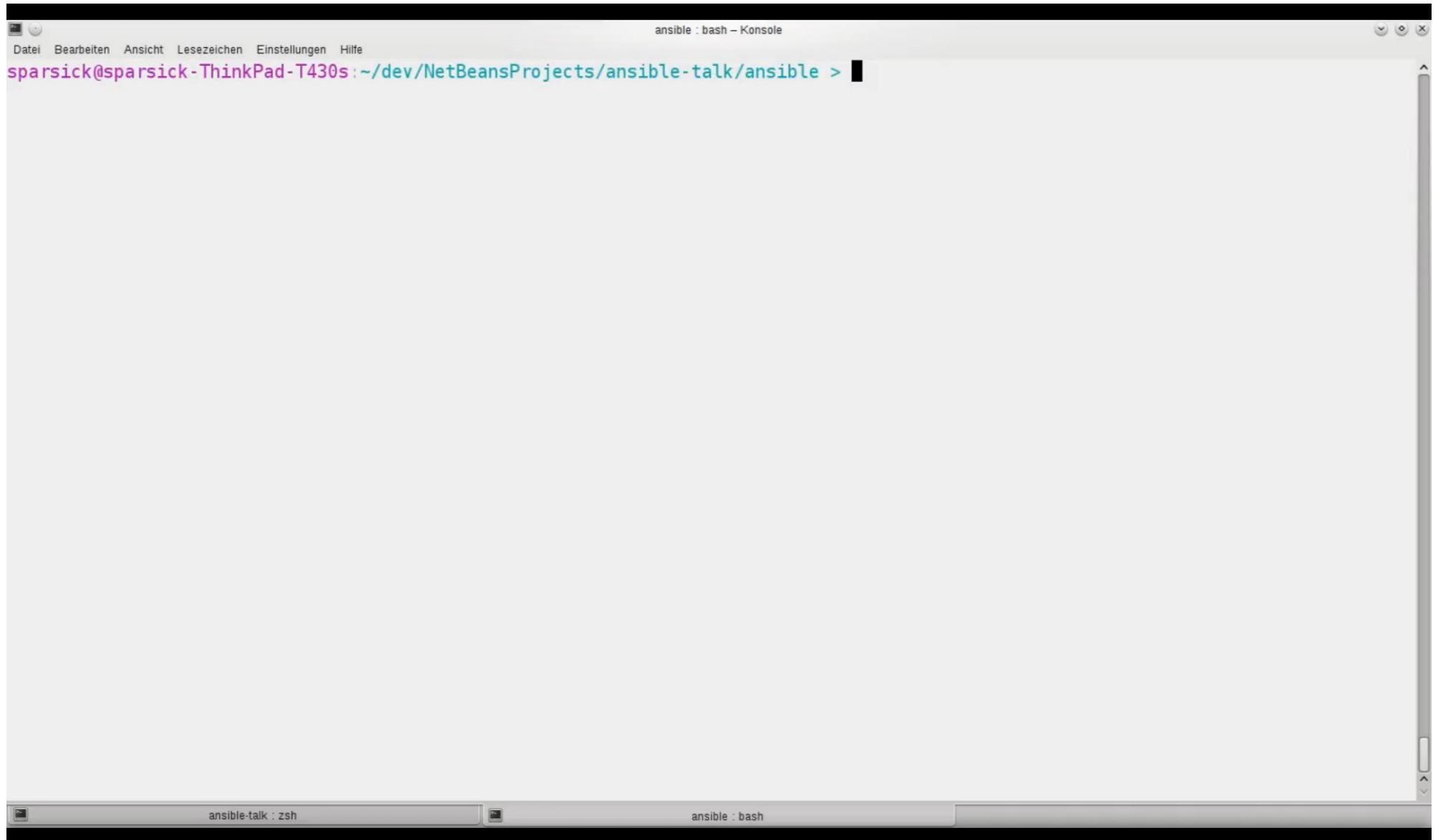
Templates - Jinja2

- 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

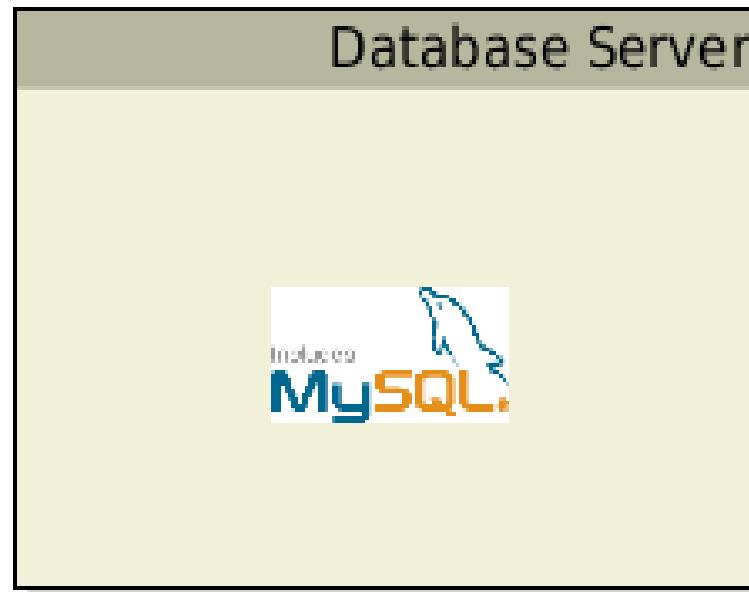


The screenshot shows a Linux desktop environment with two terminal windows open. The top terminal window is titled "ansible : bash – Konsole" and displays a command prompt for the user "sparsick" on a machine named "sparsick-ThinkPad-T430s". The bottom terminal window is titled "ansible-talk : zsh" and also displays a command prompt for the same user and machine. Both terminals show a blank white screen.

```
ansible : bash – Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible > 

ansible-talk : zsh
sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >
```

Setup Database Server Playbook



```
- 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

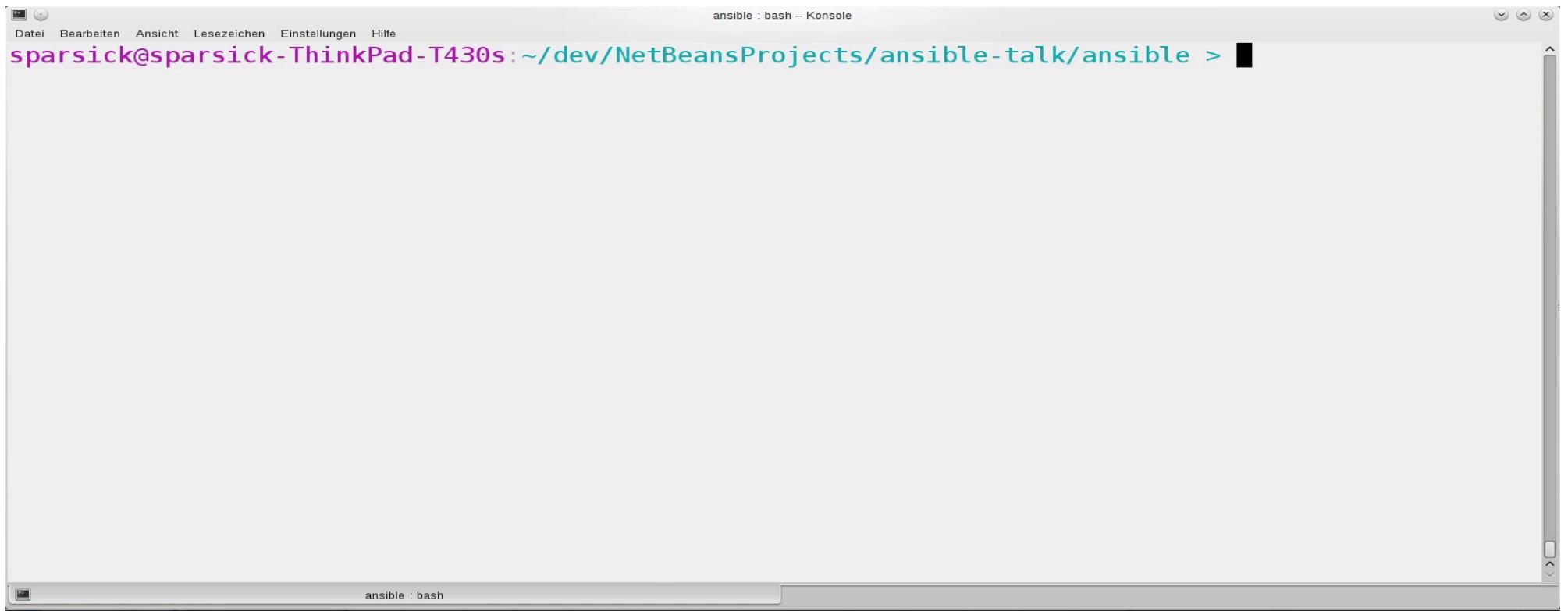
    - name: set bind address
      lineinfile: dest=/etc/mysql/mysql.conf.d/mysqld.cnf
        line='bind-address = 0.0.0.0'
        state=present
        regexp='^bind-address(.*)
      notify: restart mysql

    - 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



The image shows a screenshot of a terminal window titled "ansible : bash – Konsole". The window is part of a desktop environment, indicated by the menu bar at the top which includes "Datei", "Bearbeiten", "Ansicht", "Lesezeichen", "Einstellungen", and "Hilfe". The main area of the terminal is currently empty, showing only the command prompt "sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >". The bottom of the window displays the title bar again with the text "ansible : bash".



```
- 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
```

```
- 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/
```

```
.   deploy-on-tomcat
      ├── defaults
      │   └── main.yml
      ├── tasks
      │   ├── cleanup-webapp.yml
      │   ├── deploy-webapp.yml
      │   ├── main.yml
      │   ├── start-tomcat.yml
      │   └── stop-tomcat.yml
      └── jdk
          └── tasks
              └── main.yml
  mysql
  └── tomcat8
      ├── defaults
      │   └── main.yml
      ├── files
      │   └── tomcat.service
      ├── meta
      │   └── main.yml
      ├── molecule
      │   └── default
      │       ├── Dockerfile.j2
      │       ├── molecule.yml
      │       └── playbook.yml
      └── tests
          └── __pycache__
              ├── test_default.cpython-27-PYTEST.pyc
              ├── test_default.py
              └── test_default.pyc
  └── tasks
      └── main.yml
  templates
  └── setenv.sh.j2
```

Setup Playbooks mit Roles

- Setup Application Server

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

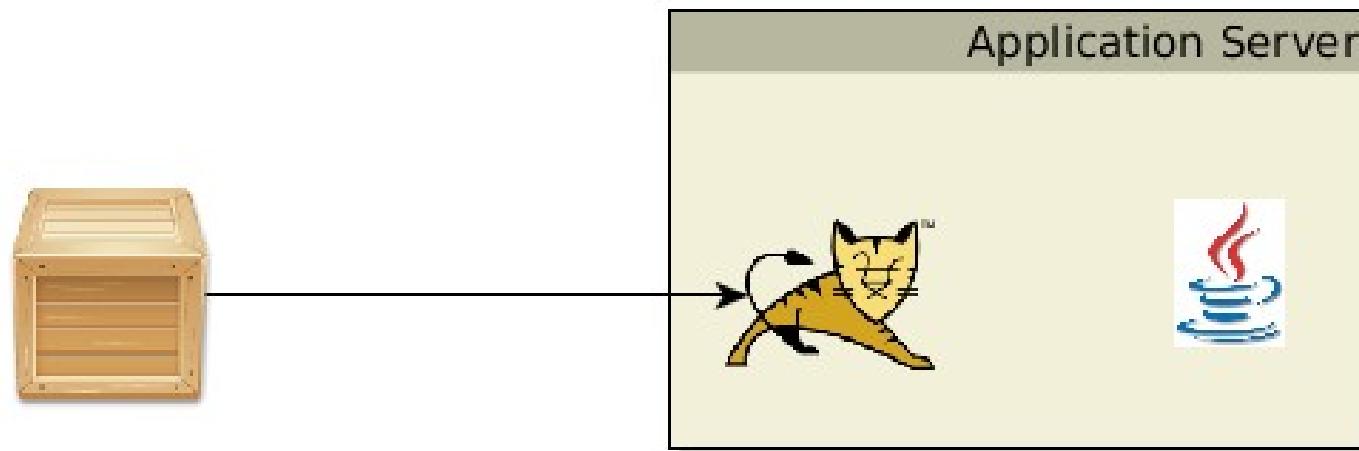
- Setup Database Server

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

include_role, import_role (seit v2.3)

```
- 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

```
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
  service: name=tomcat state=stopped
  become: true

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

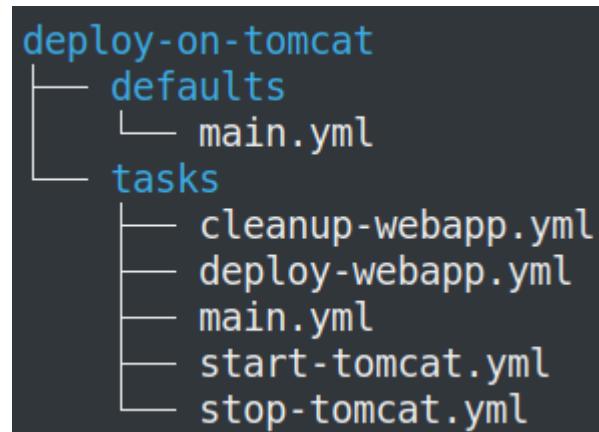
deploy-on-tomcat Role

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



```
# cat 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

```
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
  service: name=tomcat enabled=yes state=started
  become: true

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

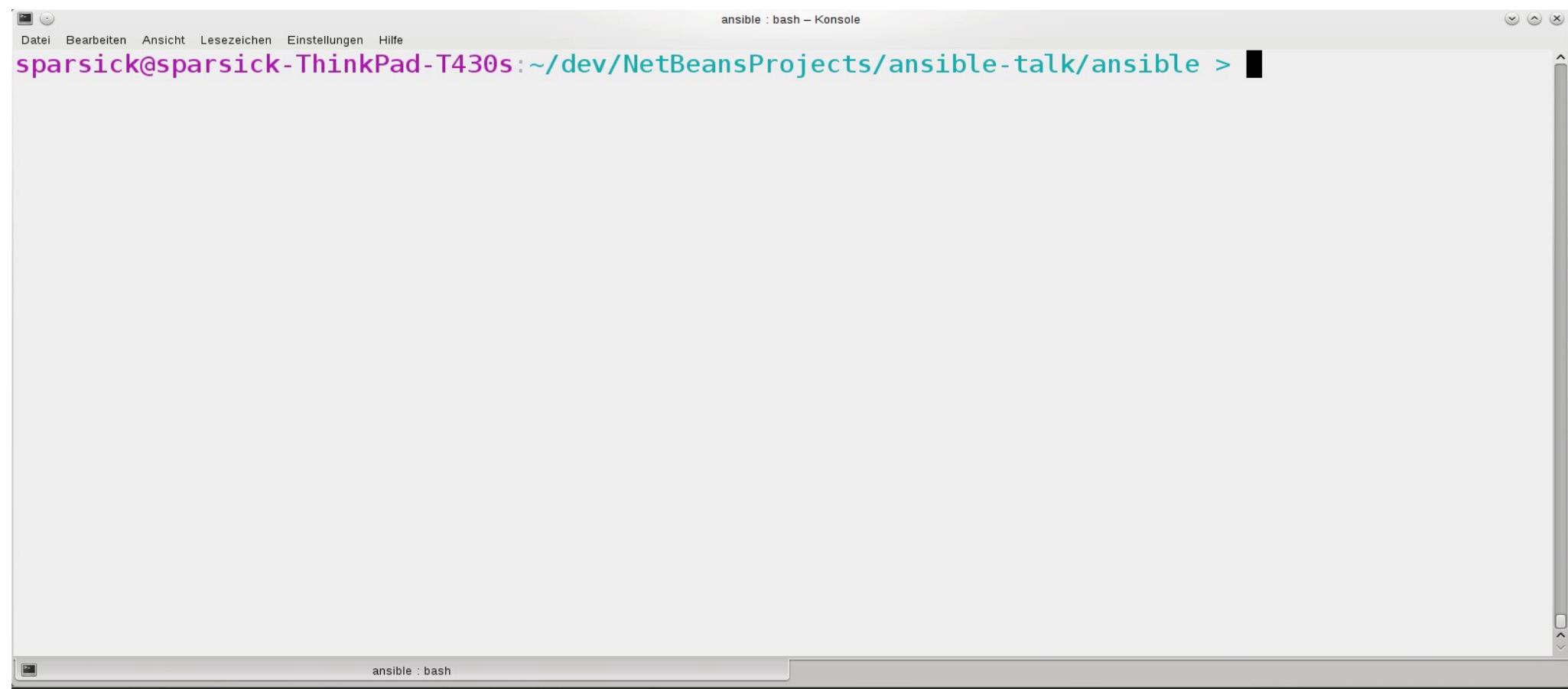
deploy-on-tomcat Role

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

```
● ● ●

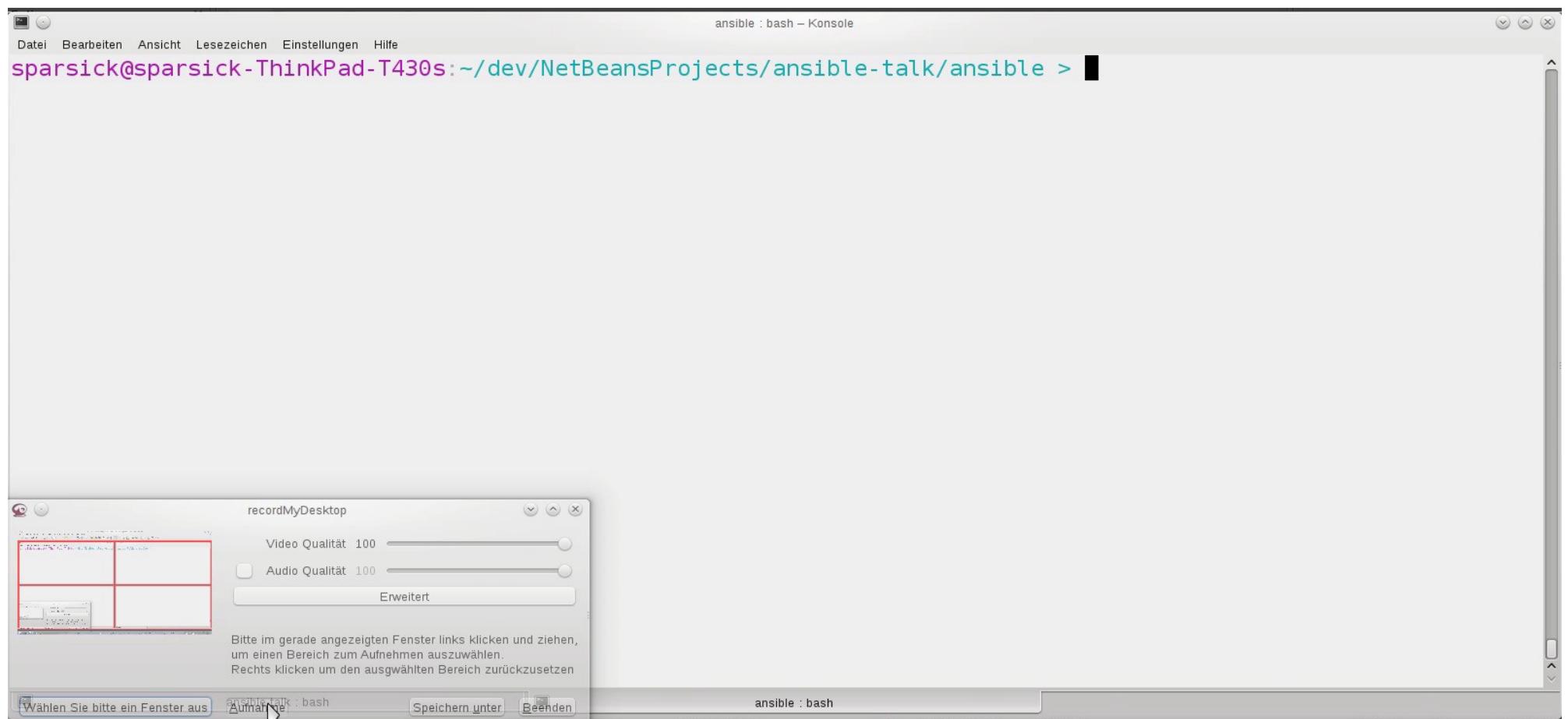
# cat defaults/main.yml
tomcat_app_base: /opt/tomcat/webapps%
```

Deploy Application Playbook

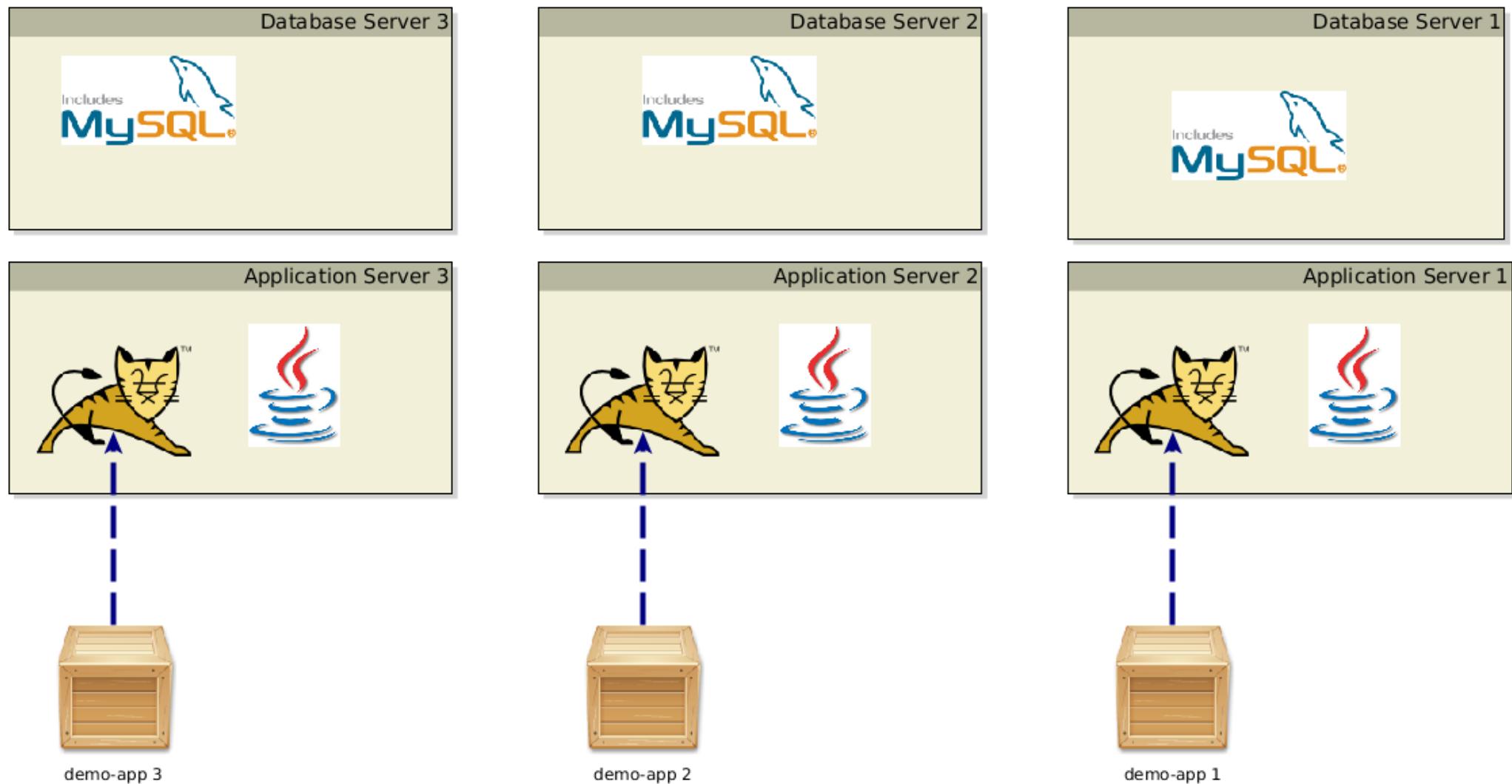


A screenshot of a terminal window titled "ansible : bash – Konsole". The window has a standard OS X interface with a menu bar at the top. The menu bar includes "Datei", "Bearbeiten", "Ansicht", "Lesezeichen", "Einstellungen", and "Hilfe". Below the menu bar, the title bar displays the path "sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >". The main area of the terminal is empty, showing only the command prompt. At the bottom of the window, there is a status bar with the text "ansible : bash".

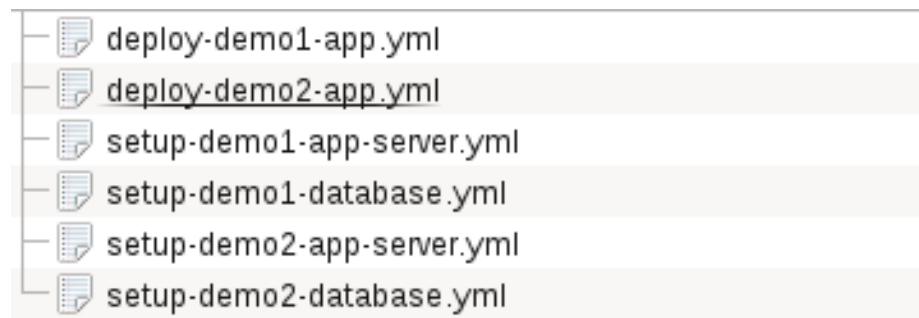
Ad-hoc-Kommando



Warum Roles?



Warum Roles?



Warum Roles?

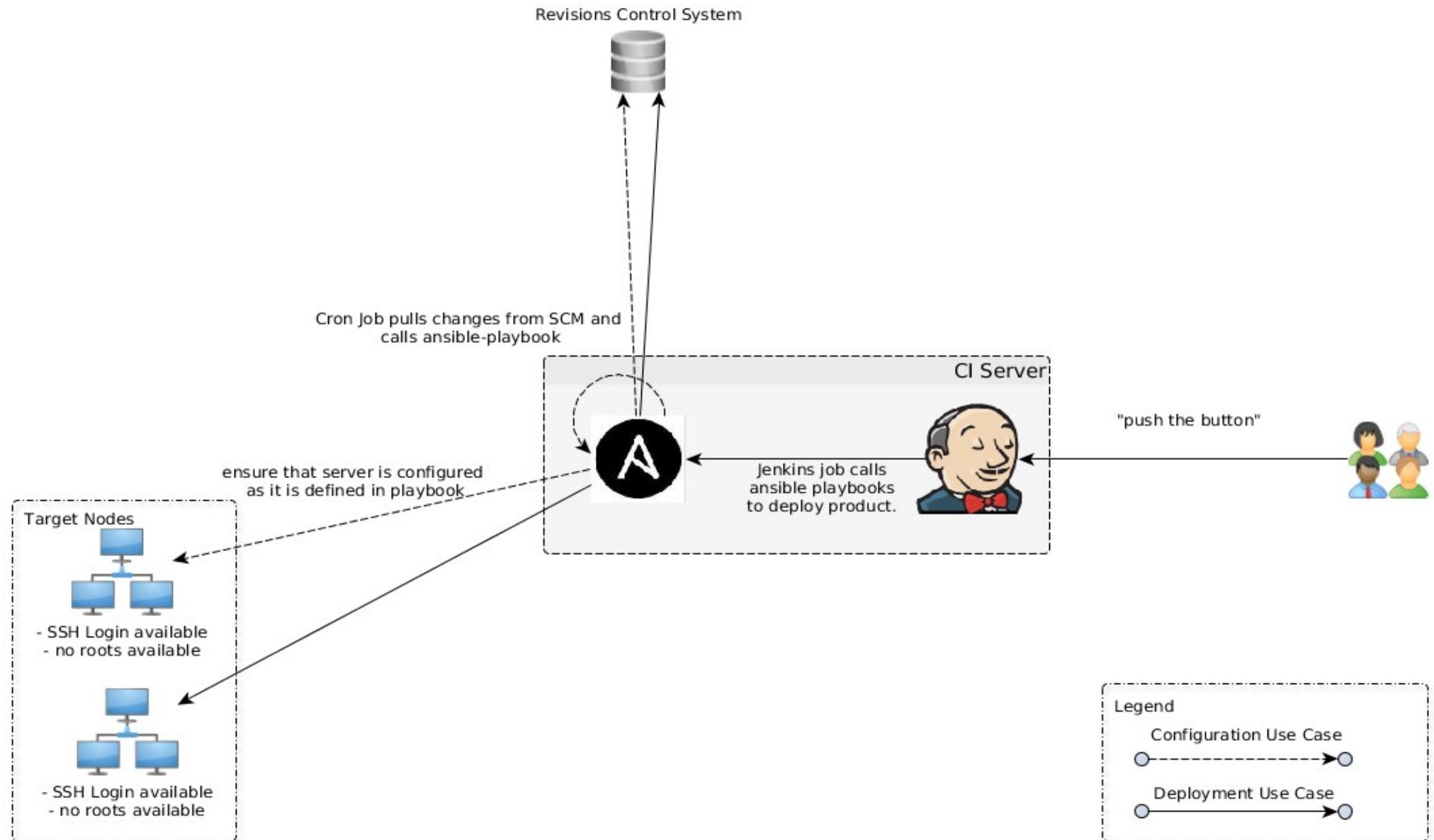


```
- hosts: application_server
  roles:
    - {role: deploy-on-tomcat, webapp_source_path: ./demo1-app-ansible-deploy-1.0-SNAPSHOT.war,
      webapp_target_name: demo1 }
```



```
- 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

A TOWER Organizations Users Teams Credentials Projects Inventories Job Templates Jobs Hello, admin ▾

Hosts Failed Hosts Inventories Inventory Sync Failures Projects Project Sync Failures

Job Status Job Type: All Period: Past Month

Successful Failed

Jobs Time

No Host data

Jobs Schedule

Name Search

| ID | Status | Started | Type | Name | Actions |
|----|--------|----------------|----------------|--------------------|---------|
| 3 | ! | 07/11 15:45:22 | Inventory Sync | CCCC (Integration) | 🔗 🗑️ ⚙️ |
| 2 | ! | 07/11 15:44:58 | Inventory Sync | CCCC (Integration) | 🔗 🗑️ ⚙️ |
| 1 | ! | 07/11 15:43:21 | Inventory Sync | CCCC (Integration) | 🔗 🗑️ ⚙️ |

Page 1 of 1 (3 items)

Host Count

Hosts License

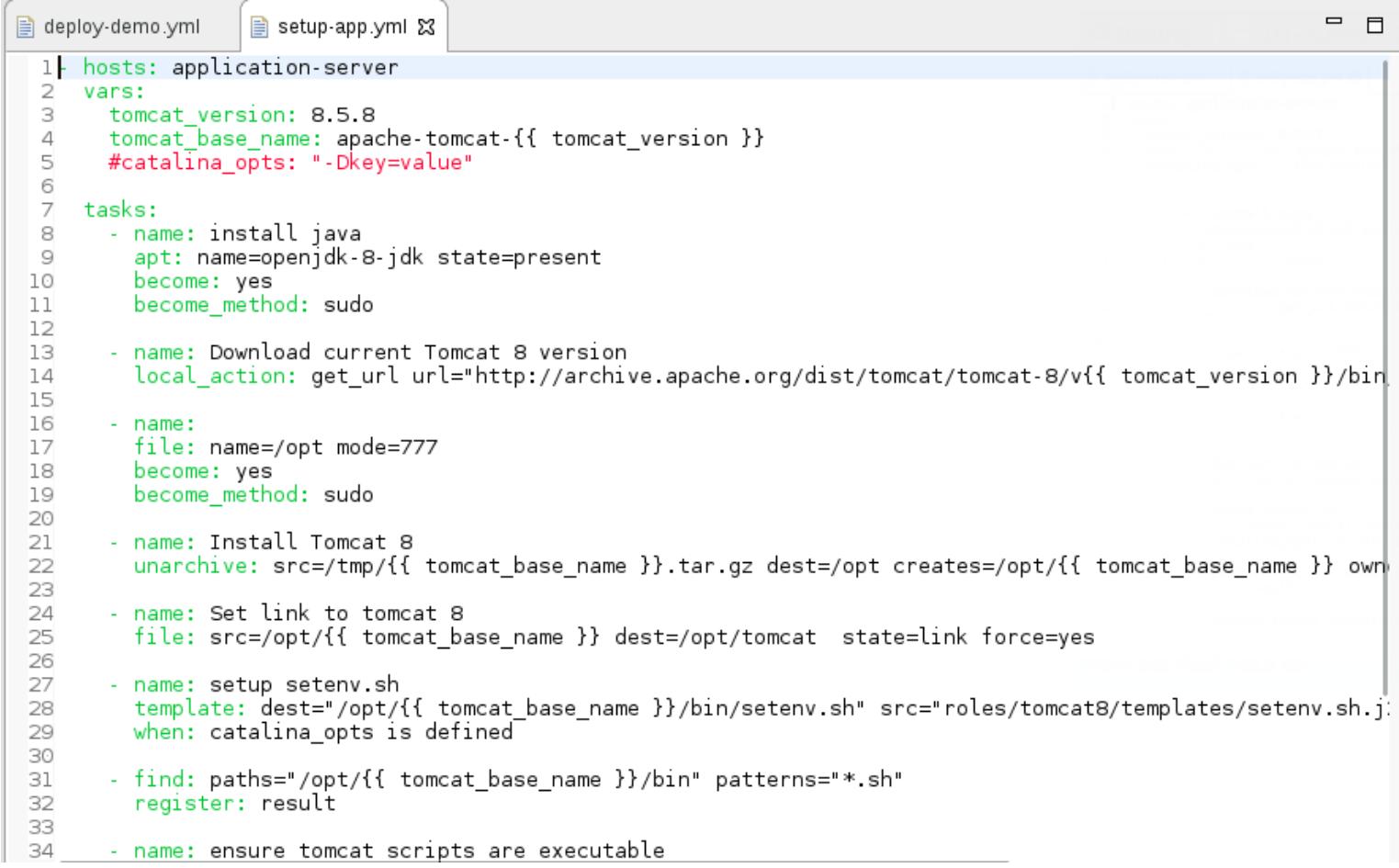
Time

IDE - Support

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

IDE-Support - Eclipse

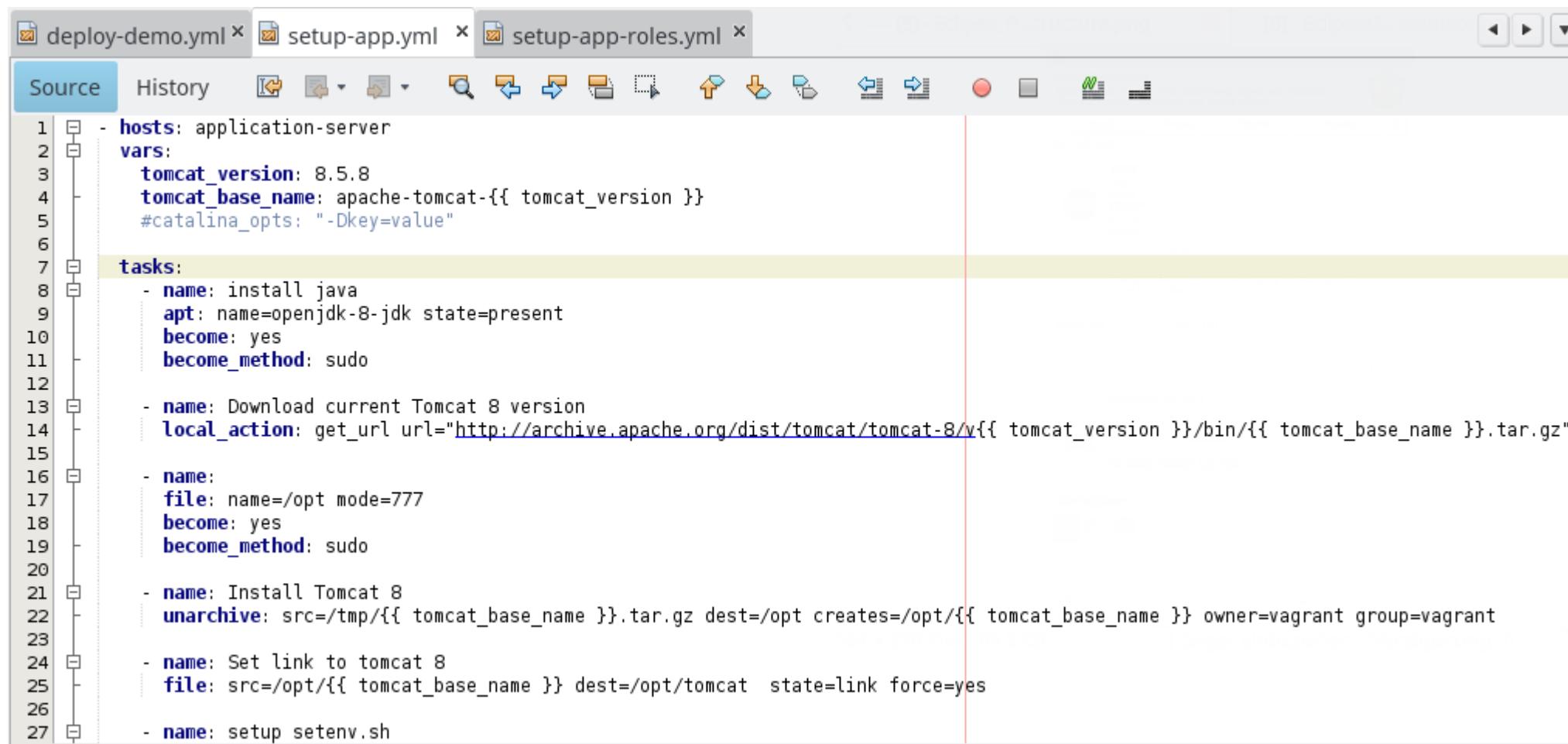
- YAML Support über Plugin *yEdit*



```
1 hosts: application-server
2 vars:
3   tomcat_version: 8.5.8
4   tomcat_base_name: apache-tomcat-{{ tomcat_version }}
5   #catalina_opts: "-Dkey=value"
6
7 tasks:
8   - name: install java
9     apt: name=openjdk-8-jdk state=present
10    become: yes
11    become_method: sudo
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"
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/{{ tomcat_base_name }} owner=www-data group=www-data
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"
29     when: catalina_opts is defined
30
31   - find: paths="/opt/{{ tomcat_base_name }}/bin" patterns="*.sh"
32     register: result
33
34   - name: ensure tomcat scripts are executable
```

IDE-Support - Netbeans

- YAML Support



The screenshot shows the Netbeans IDE interface with three tabs open at the top: "deploy-demo.yml", "setup-app.yml", and "setup-app-roles.yml". The "Source" tab is selected, displaying a block of YAML code. The code defines a host configuration and a series of tasks for installing Java and Tomcat 8.

```
1 hosts: application-server
2   vars:
3     tomcat_version: 8.5.8
4     tomcat_base_name: apache-tomcat-{{ tomcat_version }}
5     #catalina_opts: "-Dkey=value"
6
7 tasks:
8   - name: install java
9     apt: name=openjdk-8-jdk state=present
10    become: yes
11    become_method: sudo
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"
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
```

IDE-Support - IntelliJ IDEA

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

IDE-Support - IntelliJ IDEA

- Autovervollständigung für Roles

The screenshot shows the IntelliJ IDEA interface with the 'ansible' project open. The 'setup-app-roles.yml' file is the active editor. The code completion feature is demonstrating autocompletion for roles. The cursor is at line 5, position 1, where the opening brace of a role definition is highlighted. A completion dropdown menu is open, listing several options: 'deploy-on-tomcat', 'jdk', 'mysql', 'roles', and 'tomcat8'. The 'roles' option is currently selected. The status bar at the bottom right indicates: 'Dot, space and some other keys will also close this lookup and be inserted into editor'.

```
- hosts: application-server
  roles:
    - jdk
    - { role: tomcat8, tomcat_version: 8.5.8 }
    - { role: }
```

IDE-Support - IntelliJ IDEA

- Anzeige der Dokumentation für Ansible Module

The screenshot shows the IntelliJ IDEA interface with several tabs at the top: setup-app.yml, setup-app-roles.yml, setup-db.yml, and main.yml. The main editor window displays an Ansible playbook. A specific line of code, `file: name=/opt mode=777`, is selected and highlighted in yellow. To the right of the editor, a documentation pane titled "Documentation for file" is open, showing the Ansible module documentation for "file". The documentation includes sections like Synopsis, Options, Examples, and Notes. The "Synopsis" section is currently expanded, describing the module's function in setting attributes for files, symlinks, and directories.

```
1 hosts: application-server
2 vars:
3   tomcat_version: 8.5.8
4   tomcat_base_name: apache-tomcat-{{ tomcat_version }}
5   #catalina_opts: "-Dkey=value"
6
7 tasks:
8   - name: install java
9     apt: name=openjdk-8-jdk state=present
10    become: yes
11    become_method: sudo
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
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
```

Documentation for file

file - Sets attributes of files

- Synopsis
- Options
- Examples
- Notes

file Synopsis

re

- Sets attributes of files, symlinks, and directories, or removes files/symlinks/directories. Many other modules support the same options as the file module -

IDE-Support - IntelliJ IDEA

- Direkte Navigation zu der Rollendefinition

The screenshot shows the IntelliJ IDEA interface with several tabs at the top: "setup-app.yml", "setup-app-roles.yml", "tomcat8/.../main.yml", and "setup-db.yml". Below the tabs, there's a navigation bar with "roles" and "role" buttons, and a dropdown menu showing "hosts" and "application-server". A tooltip is displayed over the code editor, pointing to the path "/ansible/roles/tomcat8/tasks/main.yml". The code editor itself contains the following YAML snippet:

```
3   - JUK
4   - { role: tomcat8, tomcat_version: 8.5.8 }
5
6
7
```

IDE-Support - IntelliJ IDEA

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

The screenshot shows the IntelliJ IDEA interface with several tabs at the top: setup-app.yml, setup-app-roles.yml, tomcat8/.../main.yml, setup-db.yml, jdk/.../main.yml, and cleanup-webapp.yml. The main editor area displays an Ansible YAML file. A search dialog is open, showing the text "tomcat" in the search field. Below the search field, a list of results is displayed, each with a preview icon, the role name, and the file path. The results include:

- ROLE: tomcat8 (ansible/roles)
- ROLE: deploy-on-tomcat (ansible/roles)
- A install init.d script for tomcat (ansible-talk/ansible/setup-app.yml)
- A install init.d script for tomcat (tomcat8/tasks/main.yml)
- A start tomcat (deploy-on-tomcat/tasks/start-tomcat.yml)
- A stop tomcat (deploy-on-tomcat/tasks/stop-tomcat.yml)
- A ensure tomcat scripts are executable (ansible-talk/ansible/setup-app.yml)
- A ensure tomcat scripts are executable (tomcat8/tasks/main.yml)
- A Set link to tomcat 8 (ansible-talk/ansible/setup-app.yml)
- A Set link to tomcat 8 (tomcat8/tasks/main.yml)
- A wait for tomcat to start (deploy-on-tomcat/tasks/start-tomcat.yml)
- A wait tomcat shutdown (deploy-on-tomcat/tasks/stop-tomcat.yml)
- A {{tomcat_app_base}}/{{ webapp_target_name }} state=absent (deploy-on-tomcat/tasks/cleanup-webapp.yml)

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



The screenshot shows a terminal window titled "ansible : bash – Konsole". The window has a menu bar with German labels: Datei, Bearbeiten, Ansicht, Lesezeichen, Einstellungen, Hilfe. The main area of the terminal displays the command "sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >". Below the terminal window, there is another window titled "ansible-talk : zsh".

```
ansible : bash – Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >
```

```
ansible-talk : zsh
```

```
ansible : bash
```

testinfra



```
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
```



```
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")
```

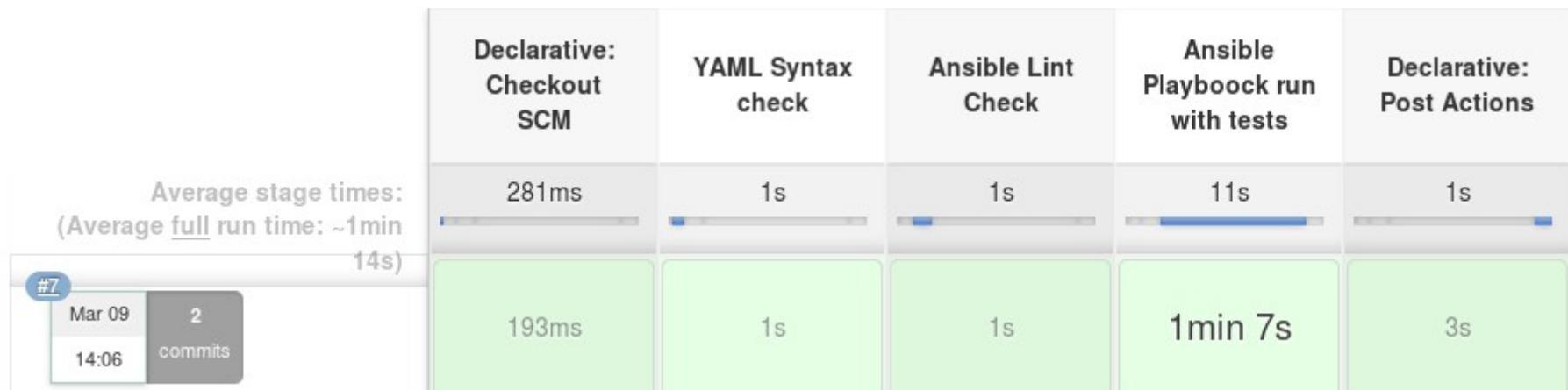
testinfra

```
x 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, ini file:  
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 =====
```

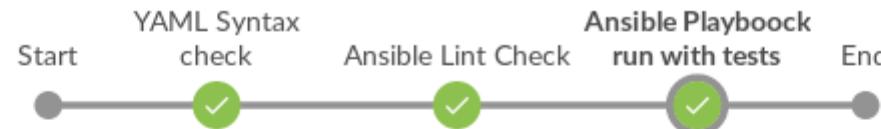
Ansible QA Jenkins Pipeline

```
1 pipeline {
2     agent any
3     stages {
4         stage('YAML Syntax check') {
5             steps {
6                 ansiblePlaybook inventory: 'inventories/test', extras: '--syntax-check', playbook: 'setup-app.yml'
7                 ansiblePlaybook inventory: 'inventories/test', extras: '--syntax-check', playbook: 'setup-db.yml'
8             }
9         }
10        stage('Ansible Lint Check') {
11            steps {
12                sh 'ansible-lint *.yml'
13            }
14        }
15        stage('Ansible Playbook run with tests') {
16            steps {
17                sh 'cd ..; vagrant up'
18                ansiblePlaybook inventory: 'inventories/test', playbook: 'setup-app.yml'
19                ansiblePlaybook inventory: 'inventories/test', playbook: 'setup-db.yml'
20                sh 'py.test --connection=ansible --ansible-inventory inventories/test -v tests/*.py'
21            }
22        }
23    }
24    post {
25        always {
26            sh 'cd ..; vagrant group destroy -f'
27        }
28    }
29 }
```

Ansible QA Jenkins Pipeline



Ansible QA Jenkins Pipeline



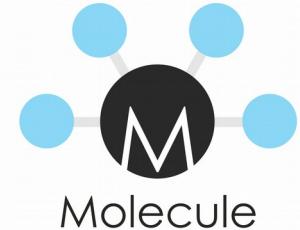
Ansible Playbook run with tests - 1m 8s



| | | |
|---|--|-----|
| ✓ | > 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 |

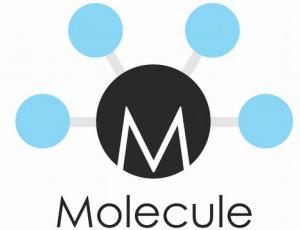
Geht es nicht einfacher?

Molecule



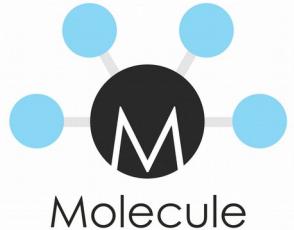
- Spezialisiert für Ansible Roles
- Wrapper um andere Werkzeuge, um komplexe Test Szenarien aufzubauen
 - Driver Provider: Docker (default), Vagrant, Azure, EC2
 - Lint Provider: yamllint (default), ansible-lint, flake8 (for test code)
 - Verifier framework: TestInfra (default)

Molecule



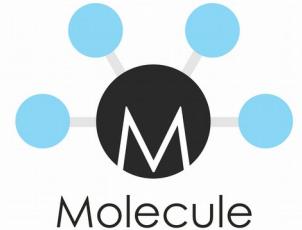
```
.  
└── tomcat  
    ├── defaults  
    │   └── main.yml  
    ├── handlers  
    │   └── main.yml  
    ├── meta  
    │   └── main.yml  
    ├── molecule  
    │   └── default  
    │       ├── Dockerfile.j2  
    │       ├── molecule.yml  
    │       ├── playbook.yml  
    │       └── tests 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

Molecule



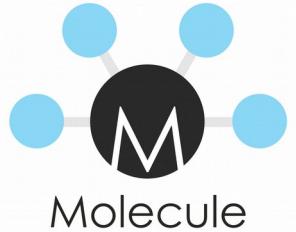
```
roles (master*) » cat tomcat/molecule/default/molecule.yml
---
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: instance
    image: ubuntu:18.04
provisioner:
  name: ansible
  lint:
    name: ansible-lint
verifier:
  name: testinfra
  lint:
    name: flake8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

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

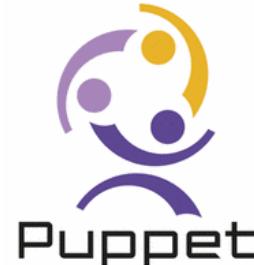
```
sparsick@sparsick-ThinkPad-T460s: /home/sparsick/dev/workspace/infra-testing-talk/infrastructure-as-code-testing/ansible/roles/tomcat git:(master) ✘
```

The screenshot shows a terminal window titled "tomcat : zsh — Konsole". The window has a standard menu bar with German labels: Datei, Bearbeiten, Ansicht, Lesezeichen, Einstellungen, and Hilfe. The main area of the terminal displays a command-line session. The prompt is "sparsick@sparsick-ThinkPad-T460s: /home/sparsick/dev/workspace/infra-testing-talk/infrastructure-as-code-testing/ansible/roles/tomcat git:(master) ✘". The terminal is part of a desktop environment, as evidenced by the taskbar at the bottom which includes icons for various applications like a browser, file manager, and system tools, along with a system tray showing network and battery status.

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



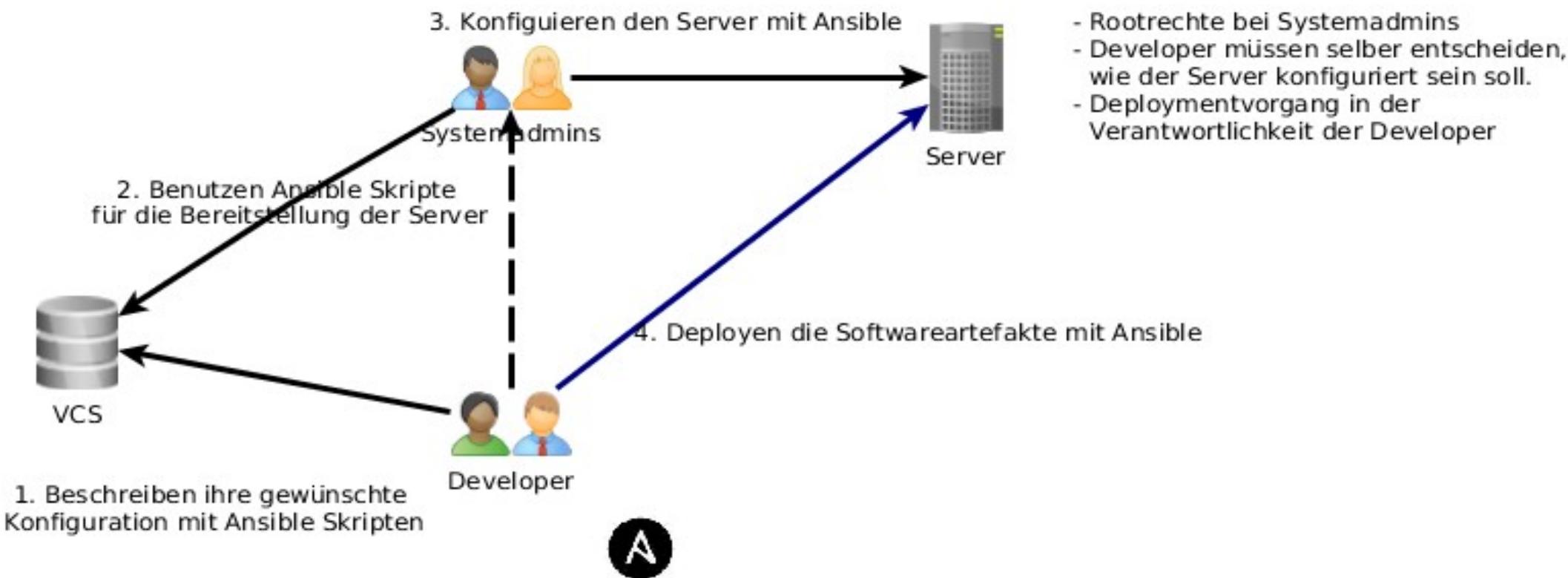
```
1 - hosts: 192.168.33.10
2   become: yes
3   become_method: sudo
4   tasks:
5     - name: Ensure Python modules
6       apt:
7         name: python-apt
8
9     - name: add nodejs ppa
10    apt_repository:
11      repo: ppa:chris-lea/node.js
12
13    - name: install nodejs package
14      apt:
15        name: nodejs
16
```

```
13  class nodejs {
14
15    class { 'apt':
16      }
17
18    exec { 'apt-get-update':
19      command    => '/usr/bin/apt-get update',
20      }
21
22    package {'software-properties-common' :
23      ensure=> installed,
24      require => Exec['apt-get-update'],
25      }
26
27    apt::ppa {'ppa:chris-lea/node.js' :}
28
29    package { 'nodejs' :
30      ensure => installed,
31      require => Apt::Ppa ['ppa:chris-lea/node.js'],
32      }
33
34 }
```

Weitere Einsatzszenarien aus Entwicklersicht

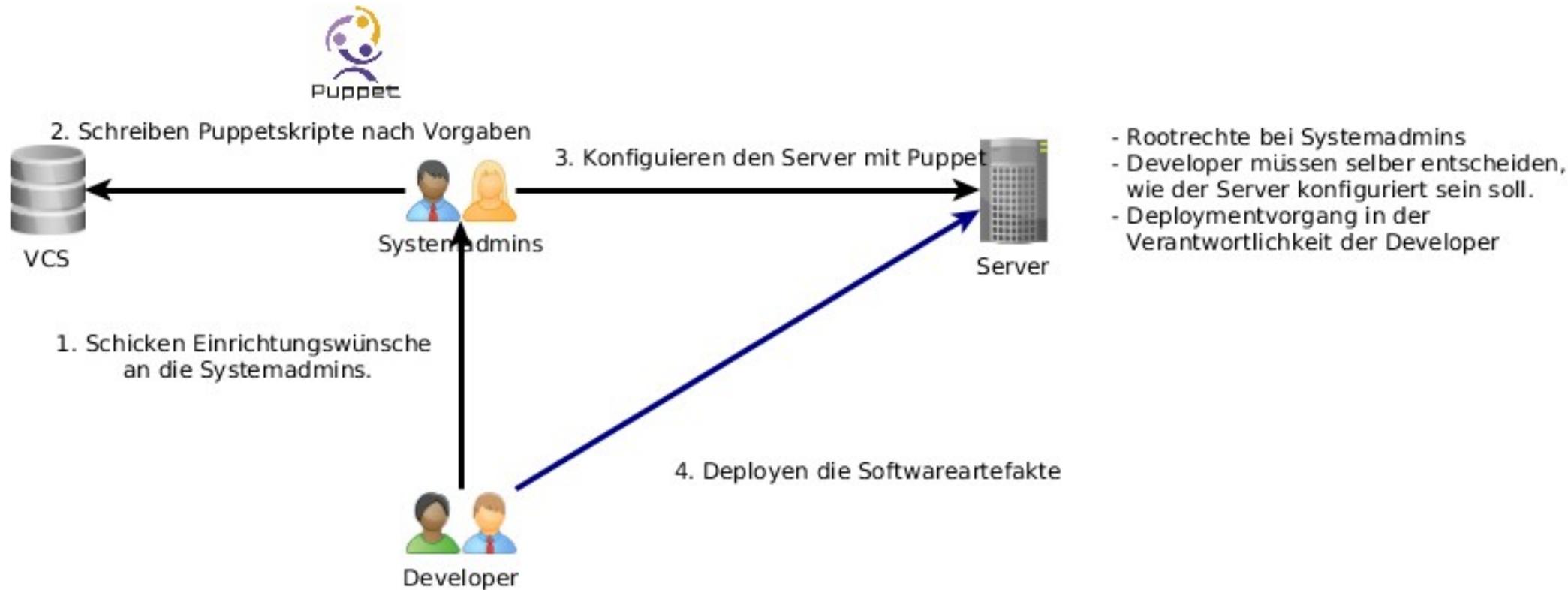
Systemkonfiguration für Entwickler

Lösungidee mit Ansible



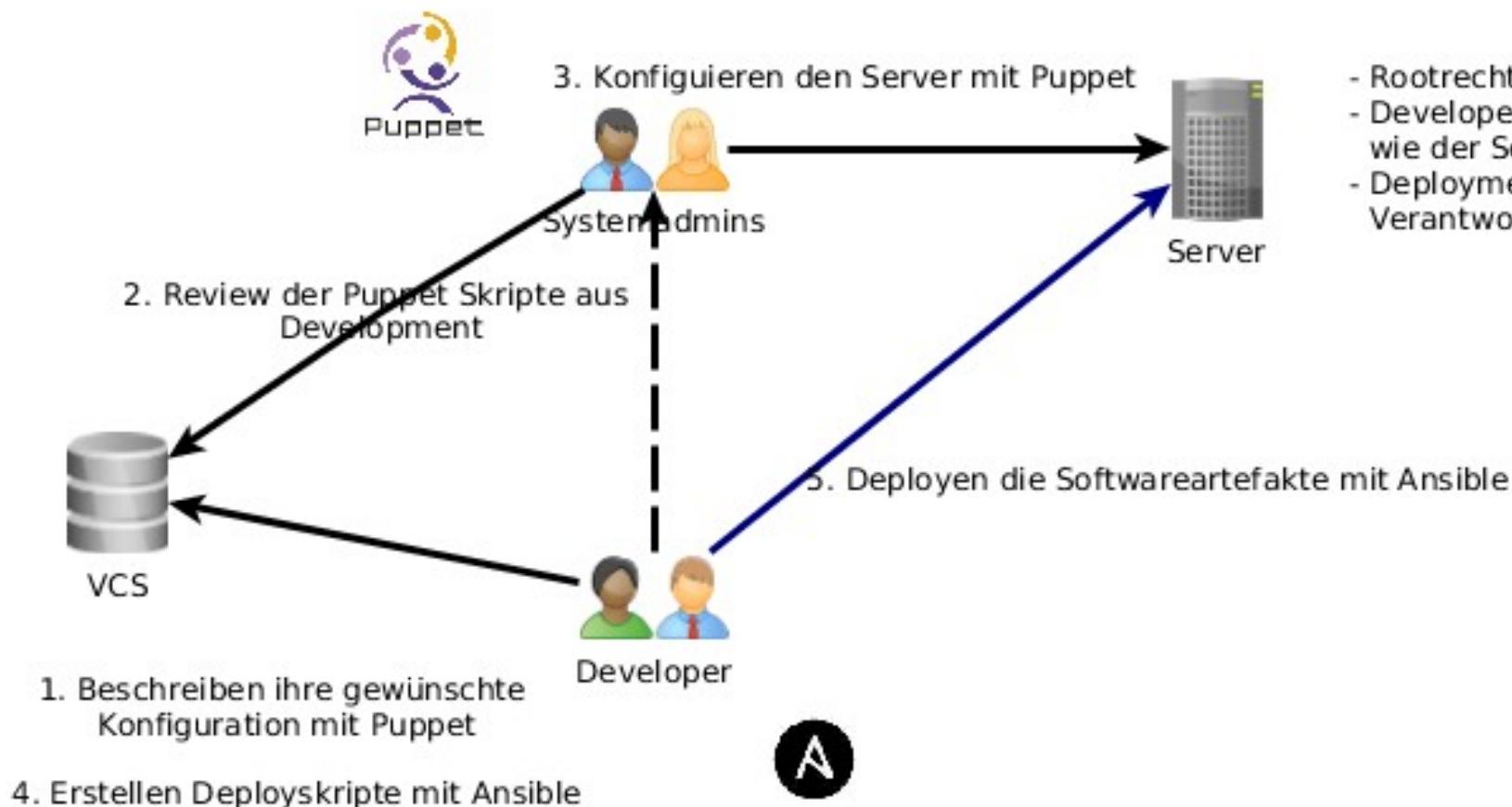
Systemkonfiguration für Entwickler

Variante - Prozess zwischen Development und Operation



Systemkonfiguration für Entwickler

Lösungsvariante

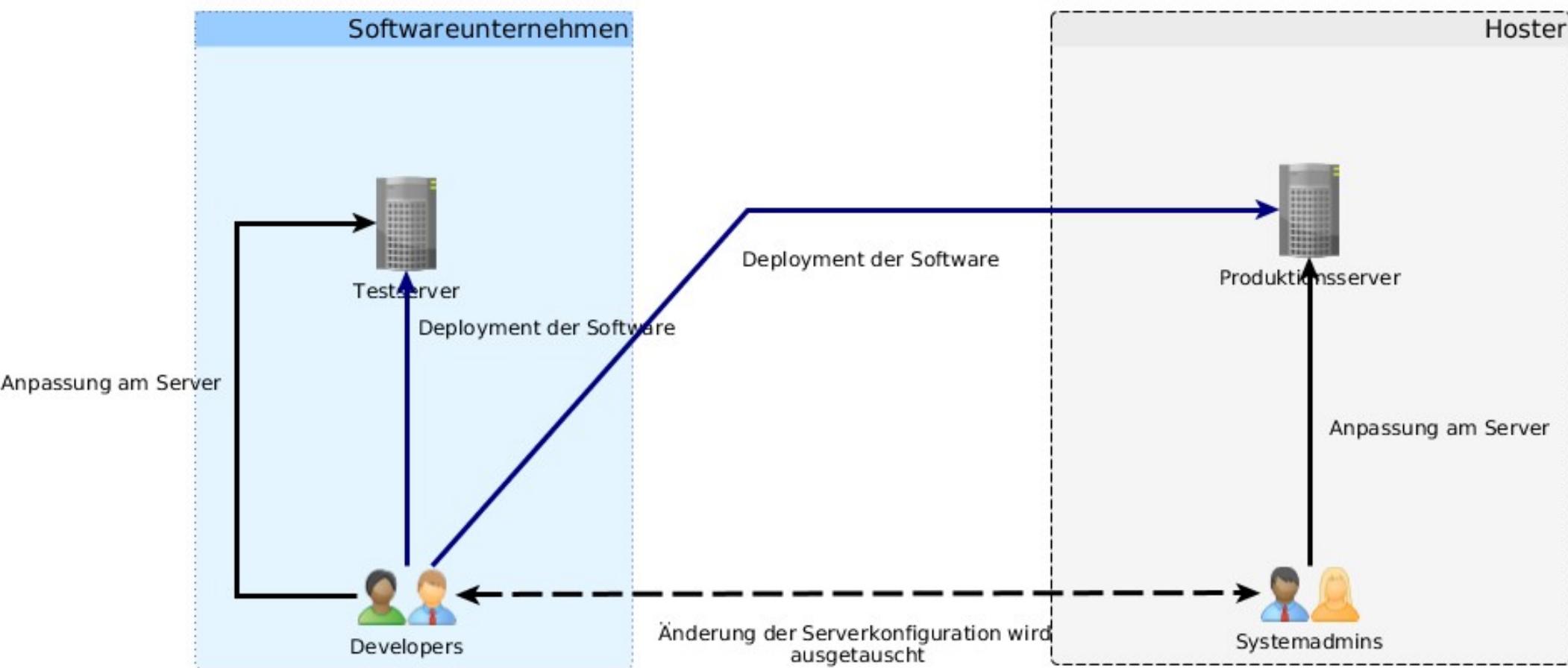


- Rootrechte bei Systemadmins
- Developer müssen selber entscheiden, wie der Server konfiguriert sein soll.
- Deploymentvorgang in der Verantwortlichkeit der Developer

A

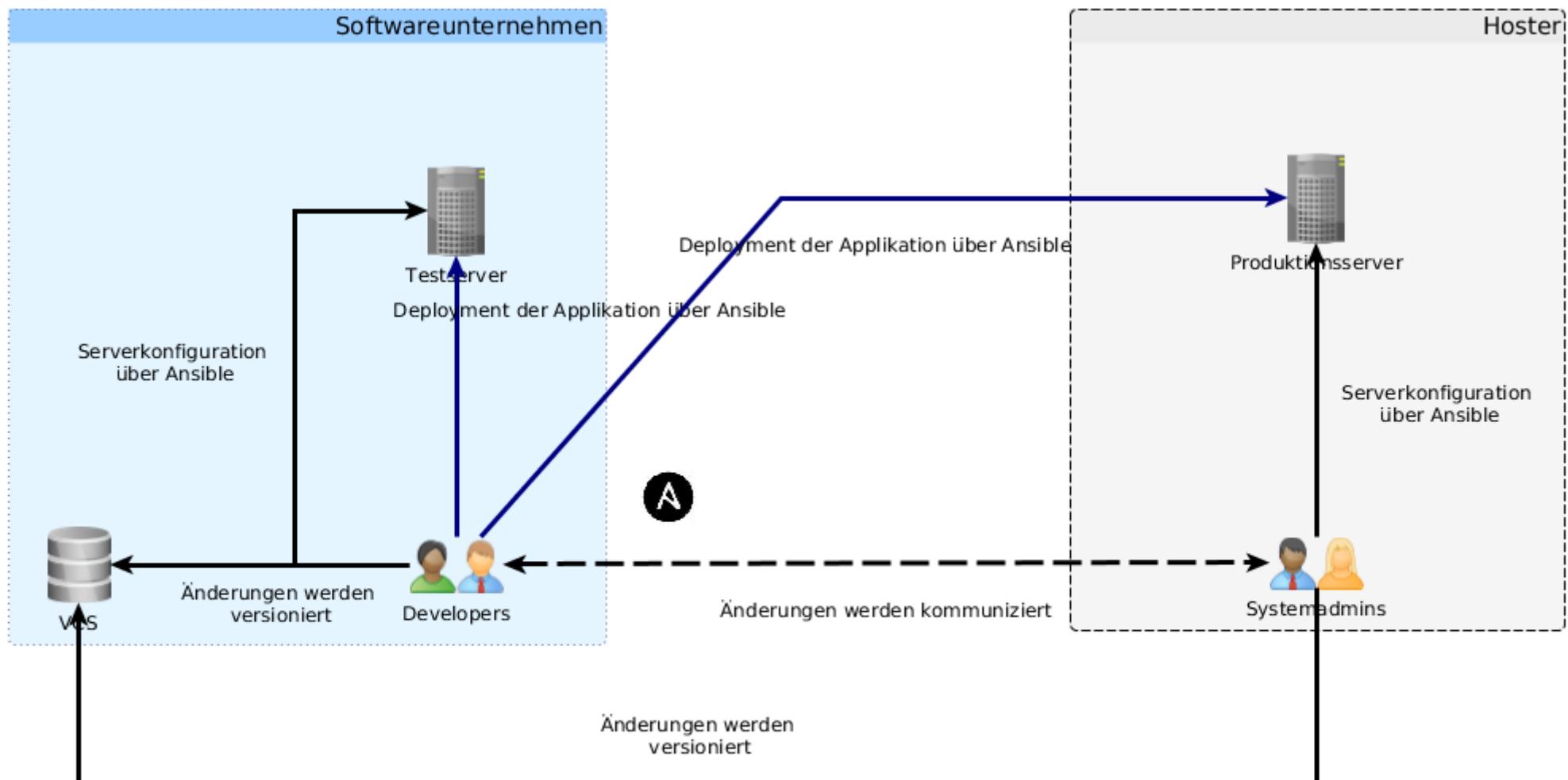
Systemkonfiguration für Entwickler

Produktionsserver sind beim externen Hoster



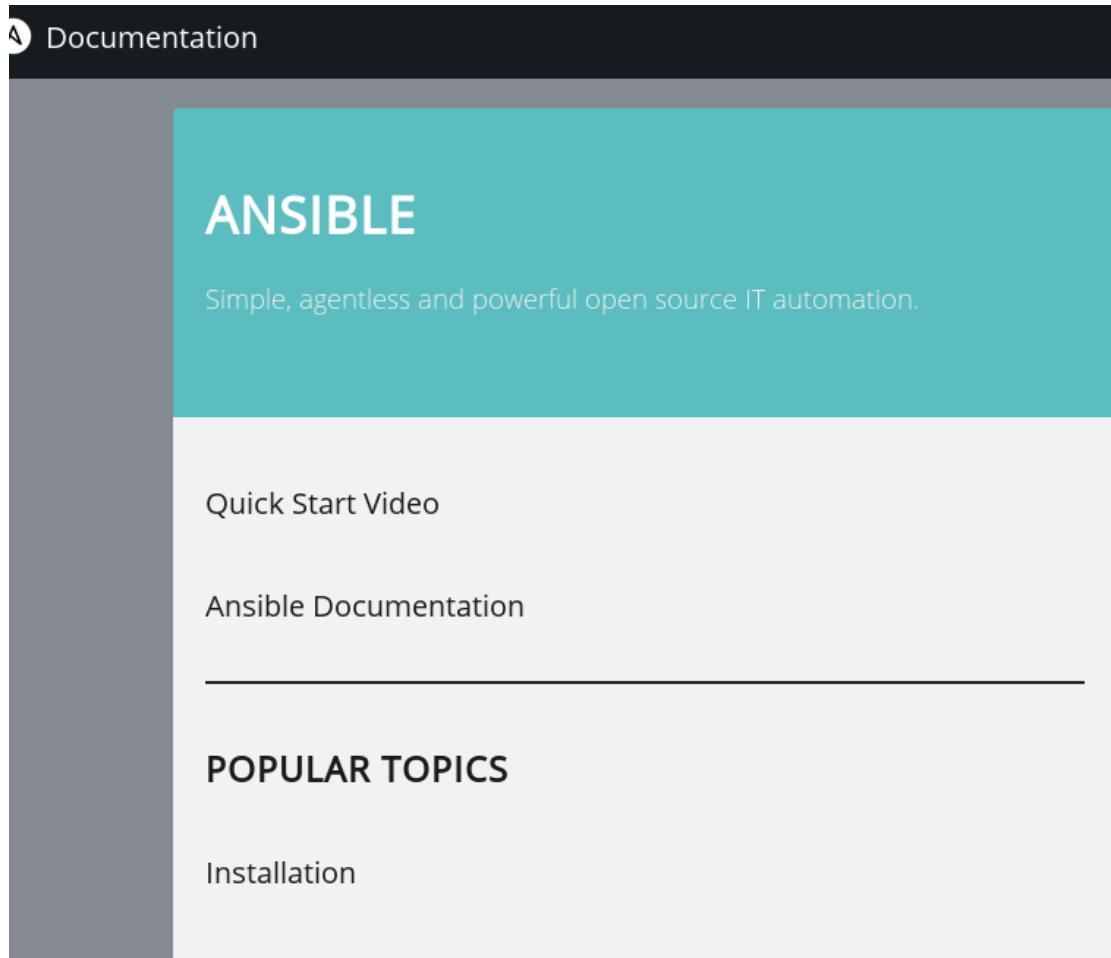
Systemkonfiguration für Entwickler

Lösungsidee



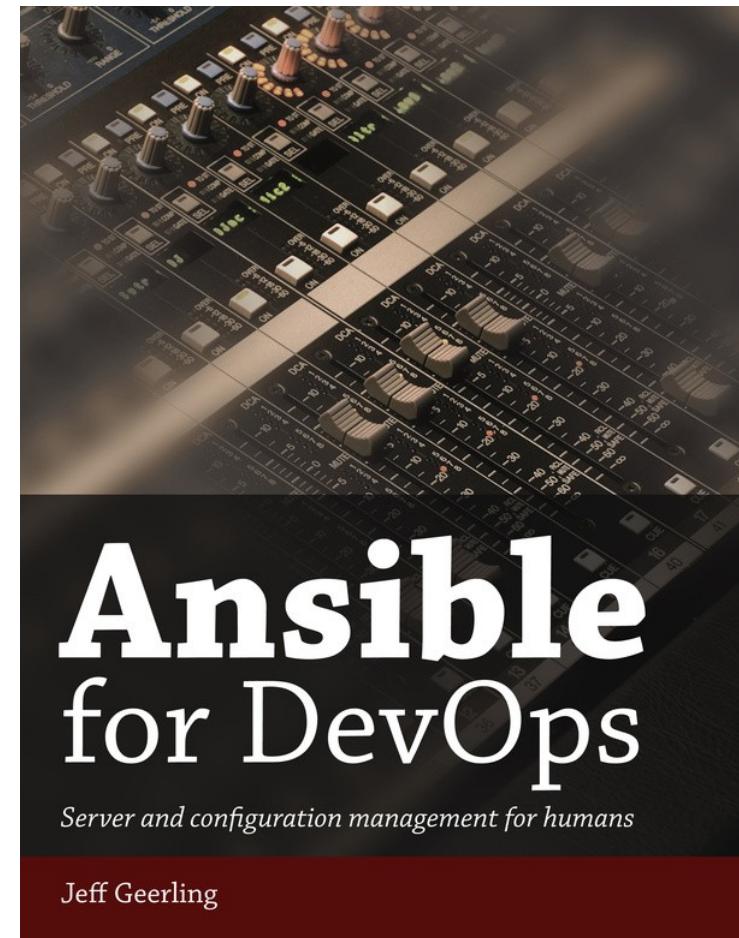
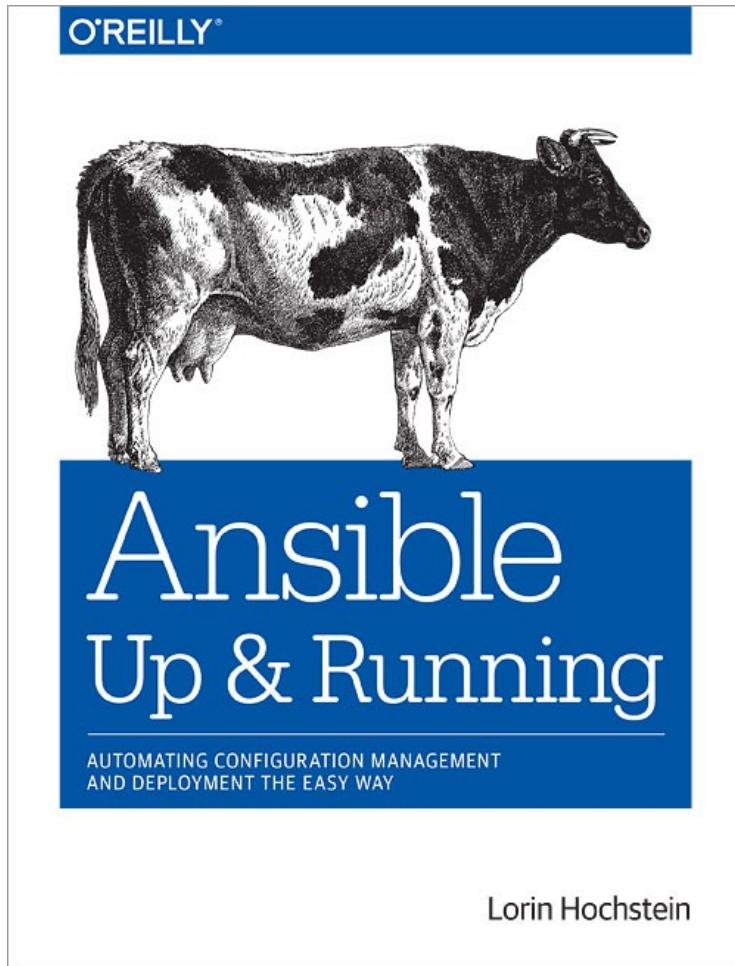
- Hoster verantwortlich für die Systemkonfiguration
- Softwareunternehmen verantwortlich für das Deployment
- Synchronisation zwischen Testserver und Produktionsserver wird vereinfacht

Weitere Informationen



<http://docs.ansible.com/>

Weitere Informationen



Weitere Informationen



The image shows the front cover of the Java aktuell magazine for the winter issue of 2016. The title "Java aktuell" is prominently displayed at the top in large blue letters. Below it, the subtitle "Praxis. Wissen. Networking. Das Magazin für Entwickler Aus der Community – für die Community" is written. The date "04-2016 | Winter | www.ijug.eu" is at the top right. A central graphic features several overlapping arrows in red, yellow, green, and blue, pointing upwards and to the right. To the left of the arrows, there's a cluster of hexagonal icons representing various software development and networking concepts like email, locks, databases, and gear wheels. The word "JAVA" is partially visible within one of these icons. At the bottom left, there's a small barcode and some additional text: "D: 4,90 EUR A: 5,60 EUR CH: 9,80 CHF B: 6,10 CHF ISSN 2191-4977". On the right side, there's a preview of an article titled "Ansible – warum Konfigurationsmanagement auch für Entwickler interessant sein kann" by Sandra Parsick.



<http://bit.ly/2cZ0lrZ>



JUnit 5
Das nächste große
Release steht vor
der Tür

Ansible
Konfigurationsma-
nagement auch für
Entwickler

Spring Boot Starter
Komfortable Modula-
risierung und Konfi-
guration



Fragen?

@SandraParsick

mail@sandra-parsick.de

<https://github.com/sparsick/ansible-talk.git>