

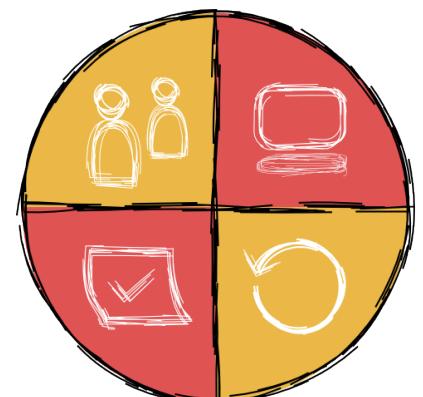
# Cloud Native Night, 24.04.2018

## Ansible für Entwickler Konfigurationsmanagement nicht nur für Ops

Sandra Parsick  
[mail@sandra-parsick.de](mailto: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.“*

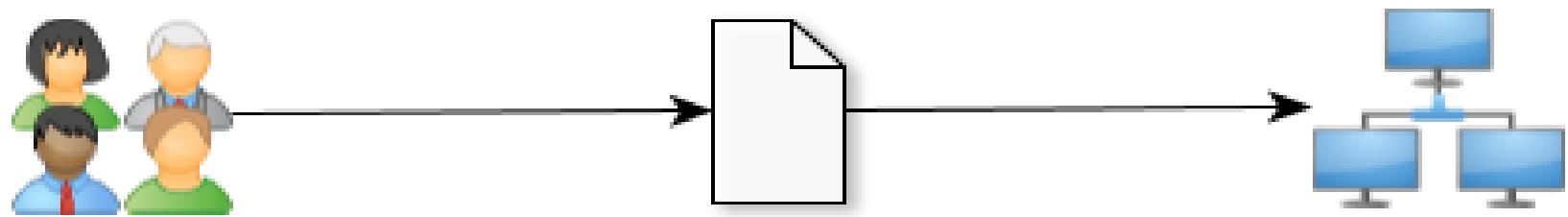
*<https://www.projektmagazin.de/glossarterm/konfigurationsmanagement>*

# Konfigurationsmanagement (KM)

- Softwarekonfiguration
- Hardwarekonfiguration
- Dienstleistungskonfiguration
- Systemkonfiguration

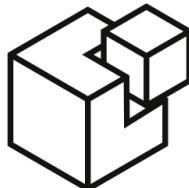
# Systemkonfiguration

- „Infrastructure As Code“



# Systemkonfiguration

## - „Infrastructure As Code“



SALTSTACK

CFEngine



CHEF™



Puppet

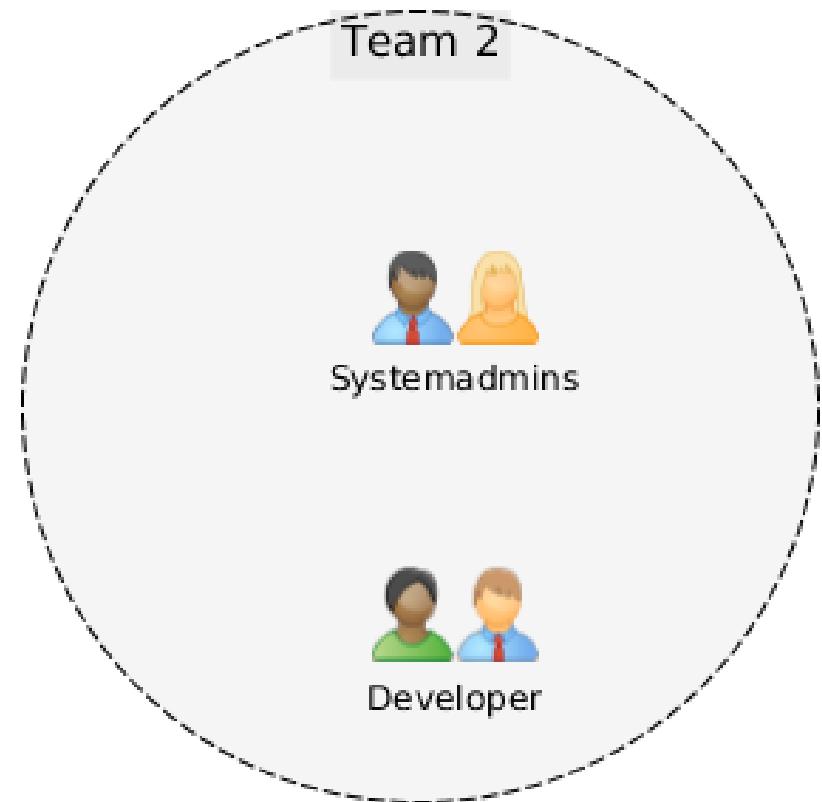
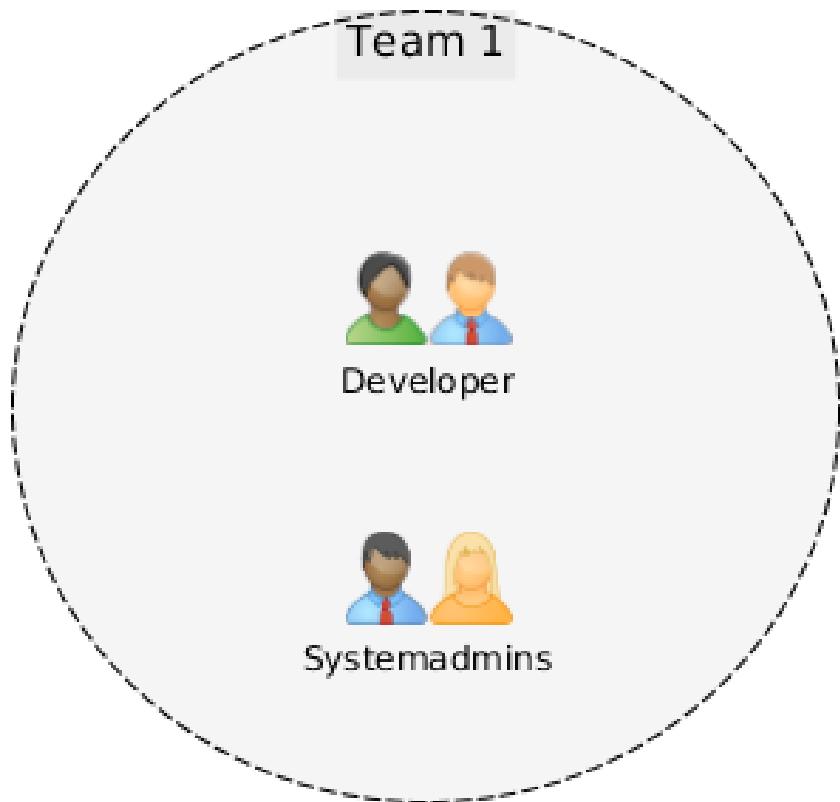


Ansible

Warum ist es für Entwickler interessant?

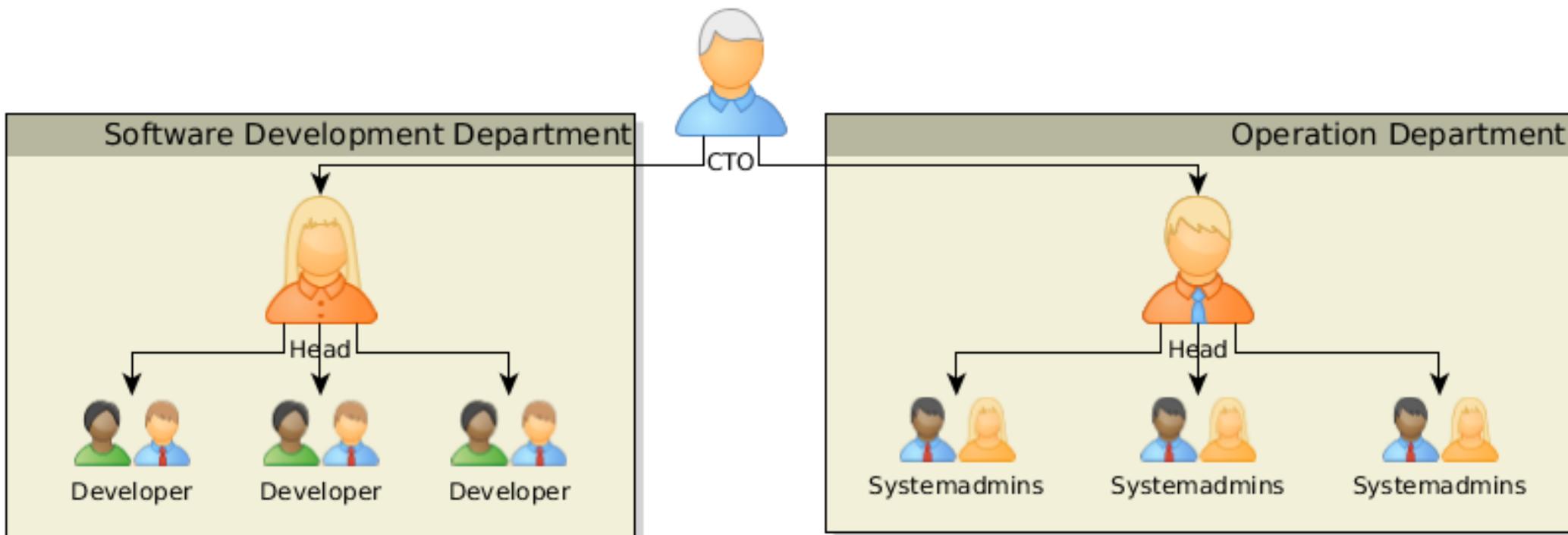
# Systemkonfiguration für Entwickler

Organisatorische Ausgangslage  
Wunsch



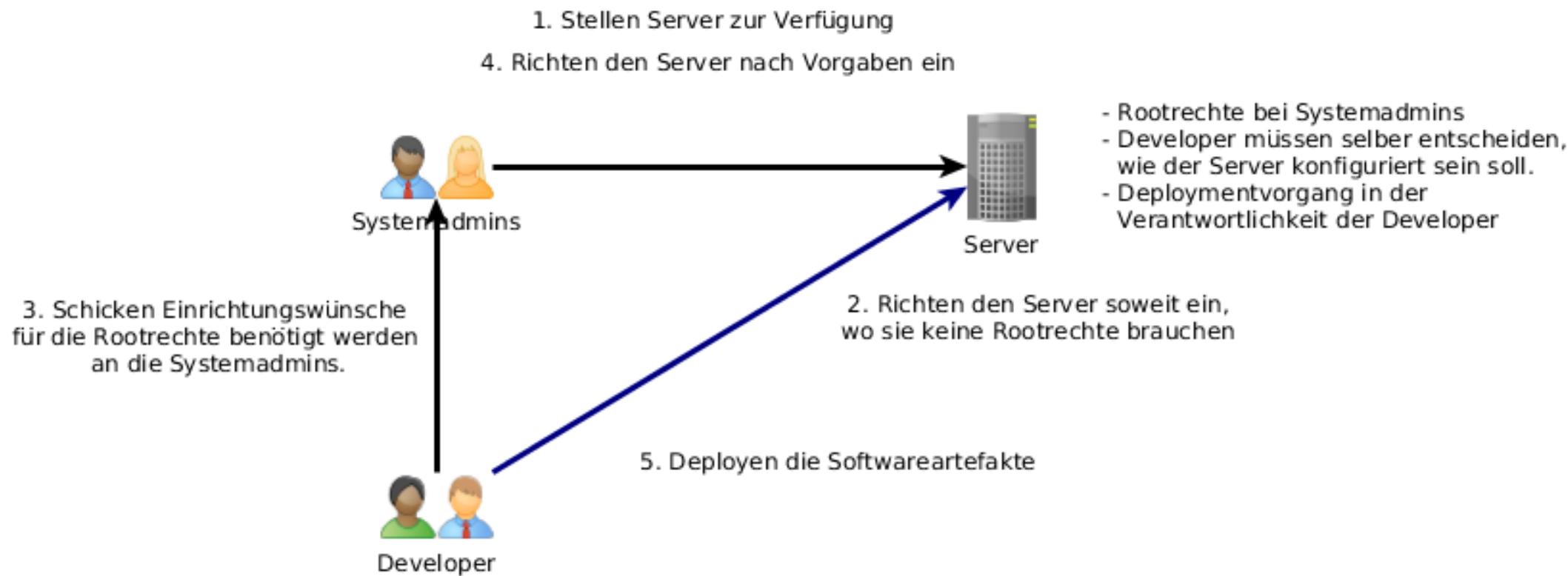
# Systemkonfiguration für Entwickler

Organisatorische Ausgangslage  
Realität



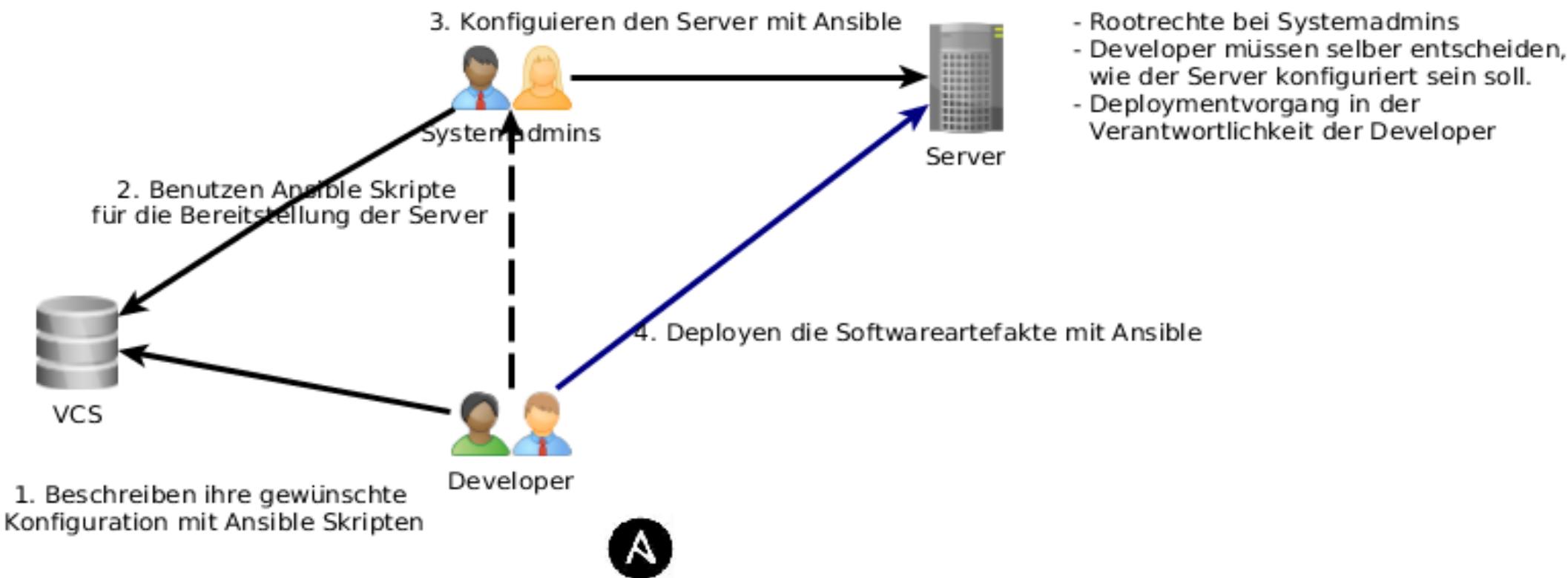
# Systemkonfiguration für Entwickler

## Prozess zwischen Development und Operation



# Systemkonfiguration für Entwickler

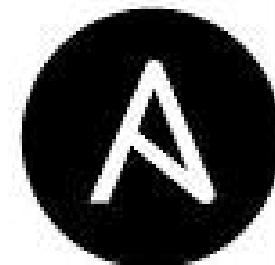
## Lösungidee mit Ansible



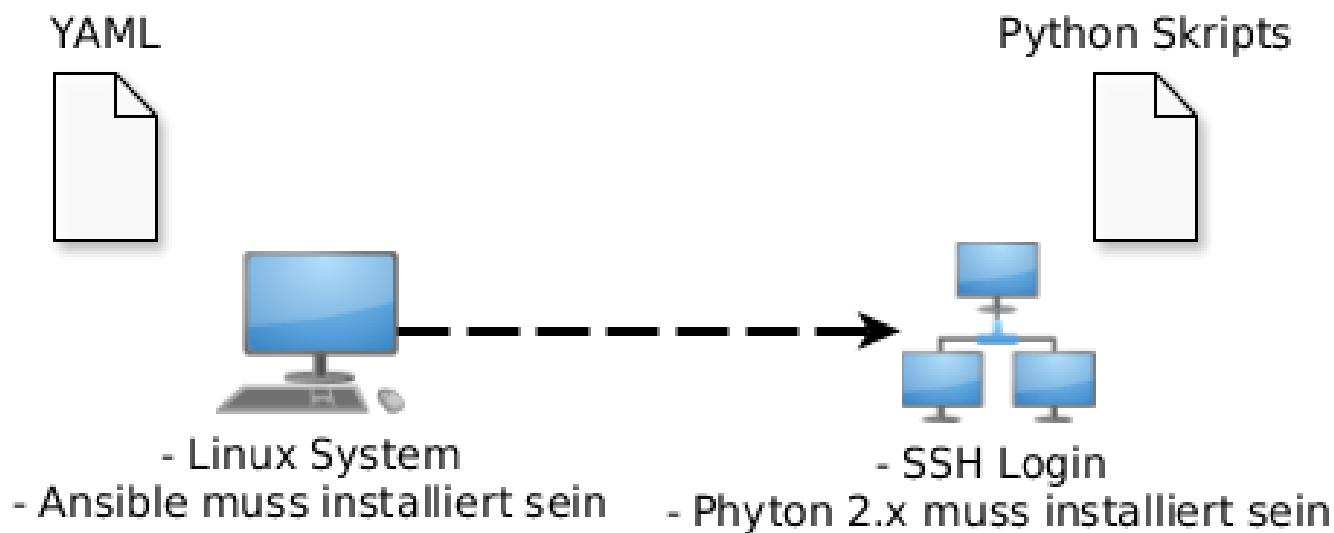
# Einführung in Ansible

# Ansible

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



# Funktionsweise



# Exkurs: YAML

## YAML

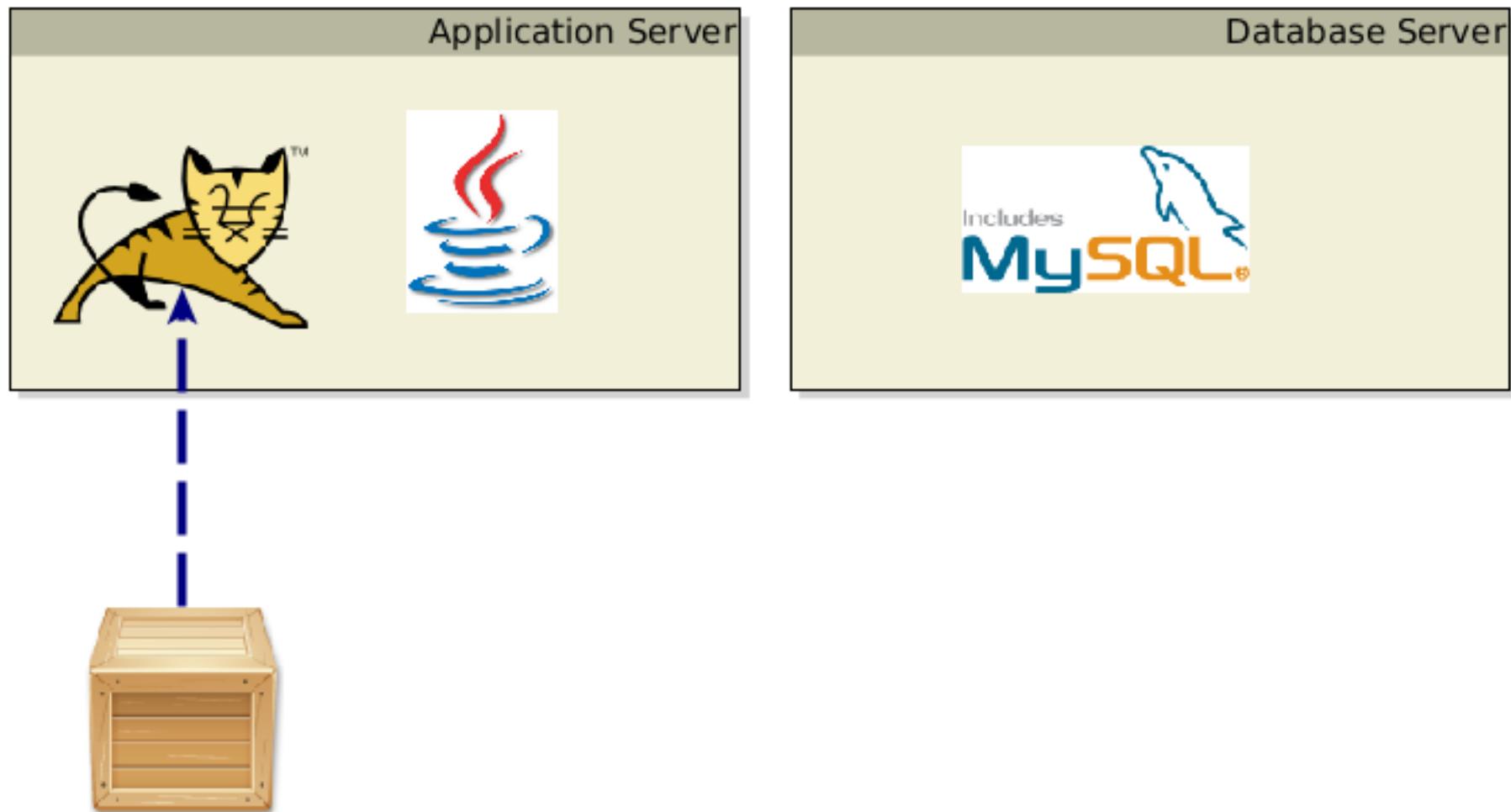
```
---
```

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

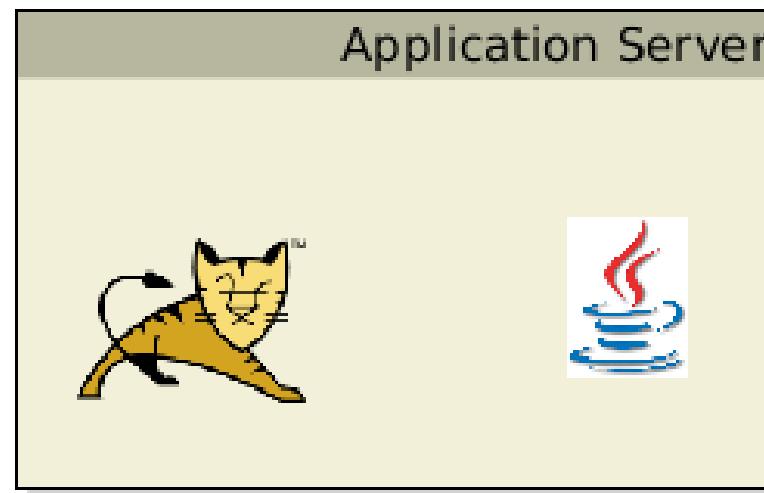
## JSON

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

# Ansible Beispiel



# Setup Application Server Playbook



```
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     become: yes
19     become_method: sudo
20
21   - name: Install Tomcat 8
22     unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name }} owner=vagrant group=vagrant
23
24   - name: Set link to tomcat 8
25     file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat state=link force=yes
26
27   - name: setup setenv.sh
28     template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh" src="roles/tomcat8/templates/setenv.sh.j2" mode=755
29     when: catalina_opts is defined
30
31   - find: paths="/opt/{{ tomcat_base_name }}/bin" patterns="*.sh"
32     register: result
33
34   - name: ensure tomcat scripts are executable
35     file: name={{item.path}} mode=755
36     with_items: '{{ result.files }}'
37
38   - name: install tomcat as service
39     copy: src=roles/tomcat8/files/tomcat.service dest=/etc/systemd/system/
40     become: yes
41     become_method: sudo
42
```

# Inventories

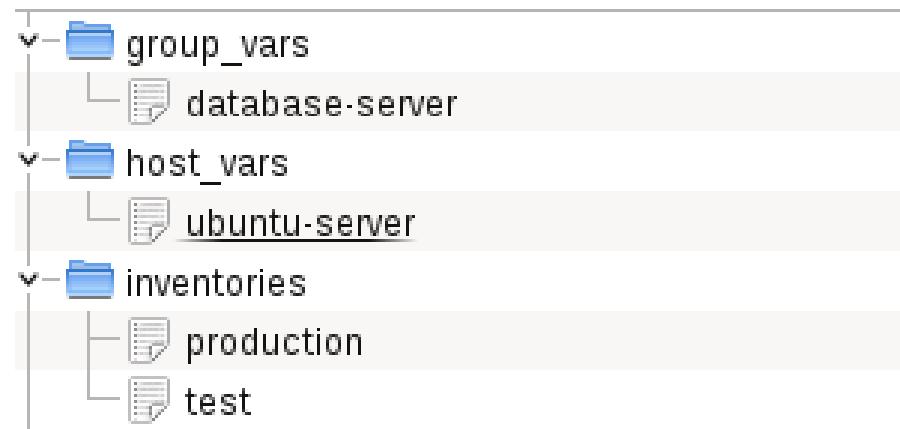
## Production

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

## Test

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

# Inventories



A screenshot of a code editor showing a YAML configuration file under the `group_vars` directory:

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

The file contains the following content:

```
proxy_host: proxy.server
```

```
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 {{ tomcat_base_name }}.tar.gz" dest=/tmp
16
17   - name:
18     file: name=/opt mode=777
19     become: yes
20     become_method: sudo
21
22   - name: Install Tomcat 8
23     unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name }} owner=vagrant group=vagrant
24
25   - name: Set link to tomcat 8
26     file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat state=link force=yes
27
28   - name: setup setenv.sh
29     template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh" src="roles/tomcat8/templates/
30     setenv.sh.j2" mode=755
31     when: catalina_opts is defined
32
33   - find: paths="/opt/{{ tomcat_base_name }}/bin" patterns="*.sh"
34     register: result
35
36   - name: ensure tomcat scripts are executable
37     file: name={{item.path}} mode=755
38     with_items: '{{ result.files }}'
39
40   - name: install tomcat as service
41     copy: src=roles/tomcat8/files/tomcat.service dest=/etc/systemd/system/
42     become: yes
43     become_method: sudo
```

# 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](#)

```
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 {{ tomcat_base_name }}.tar.gz" dest=/tmp
16
17   - name:
18     file: name=/opt mode=777
19     become: yes
20     become_method: sudo
21
22   - name: Install Tomcat 8
23     unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name }} owner=vagrant group=vagrant
24
25   - name: Set link to tomcat 8
26     file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat state=link force=yes
27
28   - name: setup setenv.sh
29     template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh" src="roles/tomcat8/templates/
30     setenv.sh.j2" mode=755
31     when: catalina_opts is defined
32
33   - find: paths="/opt/{{ tomcat_base_name }}/bin" patterns="*.sh"
34     register: result
35
36   - name: ensure tomcat scripts are executable
37     file: name={{item.path}} mode=755
38     with_items: '{{ result.files }}'
39
40   - name: install tomcat as service
41     copy: src=roles/tomcat8/files/tomcat.service dest=/etc/systemd/system/
42     become: yes
        become_method: sudo
```

# Templates

- setenv.sh.j2

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

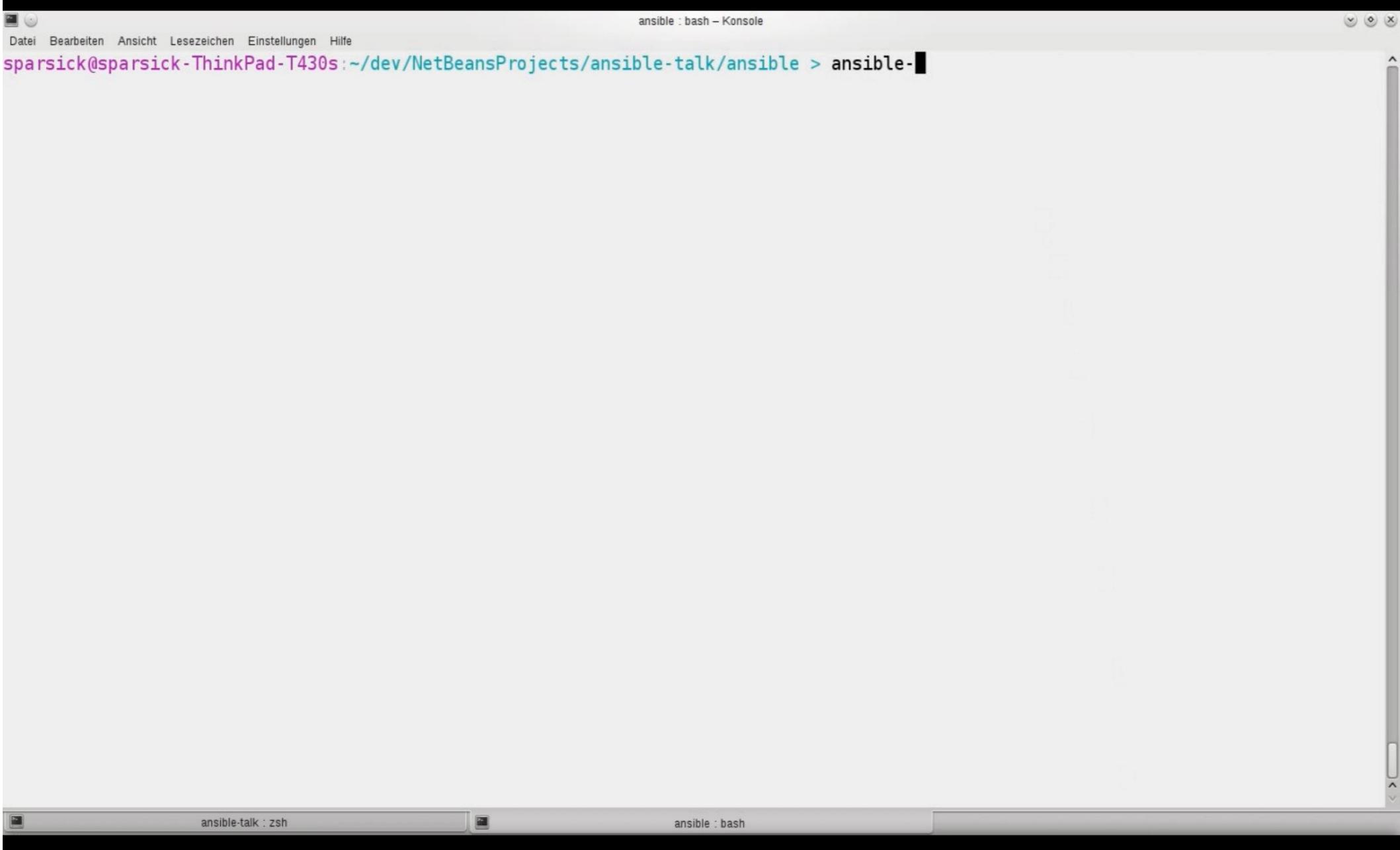
# Templates - Jinja2

- 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

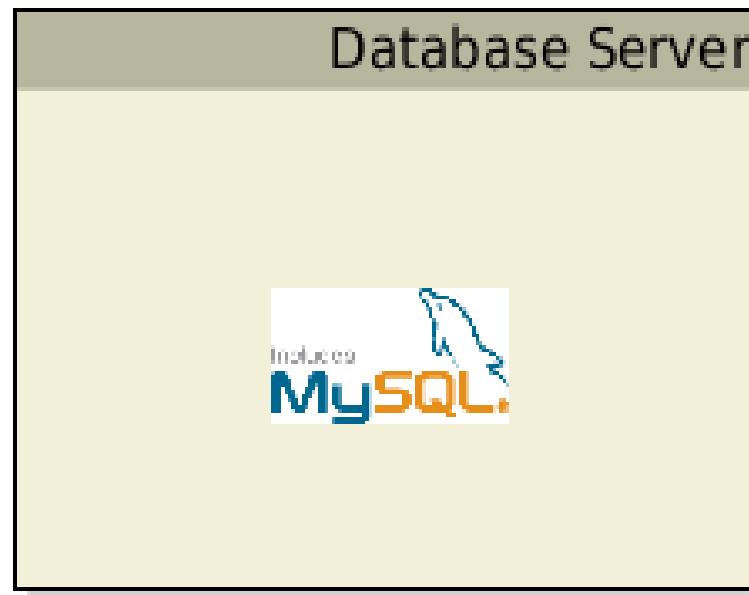


A screenshot of a terminal window titled "ansible : bash – Konsole". The window shows a command line interface with the following text:

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

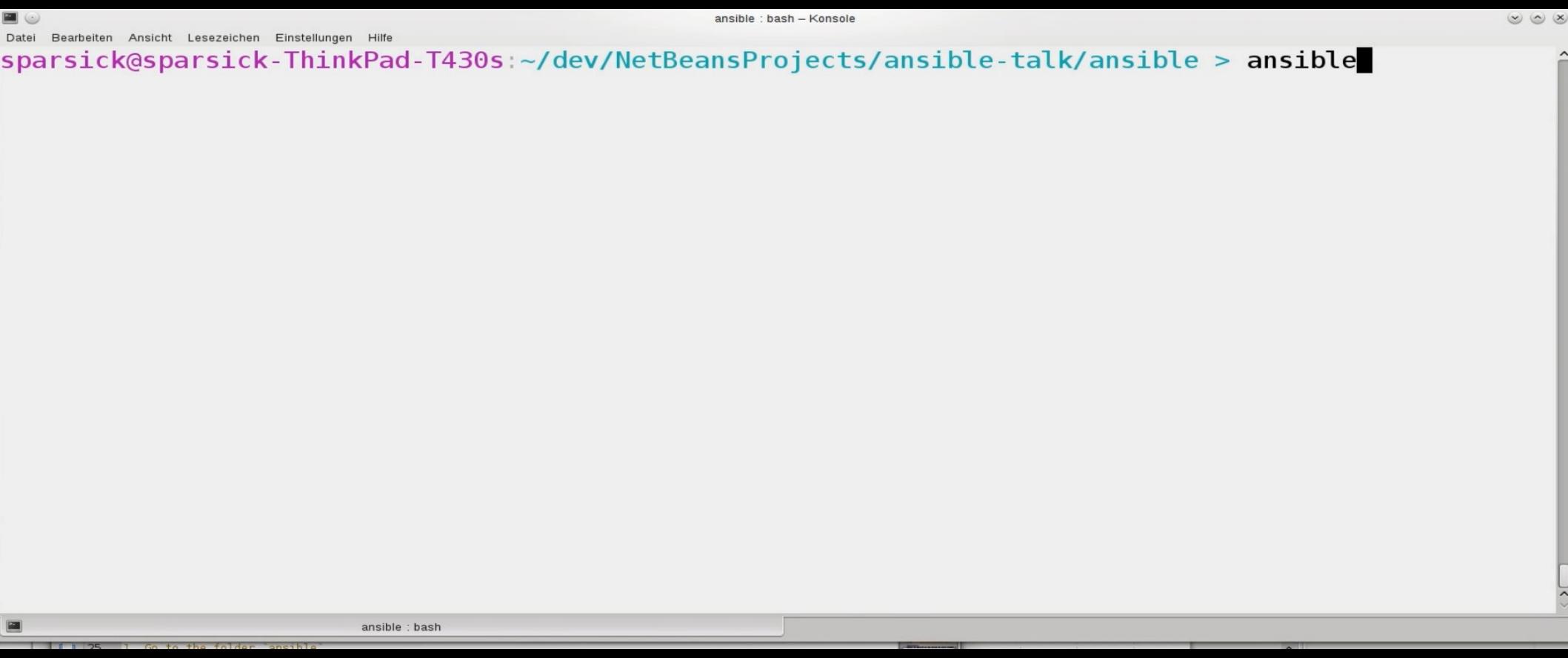
The terminal window has a menu bar with German labels: Datei, Bearbeiten, Ansicht, Lesezeichen, Einstellungen, Hilfe. At the bottom, there are two tabs: "ansible-talk : zsh" and "ansible : bash".

# 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=g3h3l1m priv=*.*:ALL,GRANT state=present host=%
24
25   handlers:
26     - name: restart mysql
27       service: name=mysql state=restarted
```

# Setup Database Server Playbook

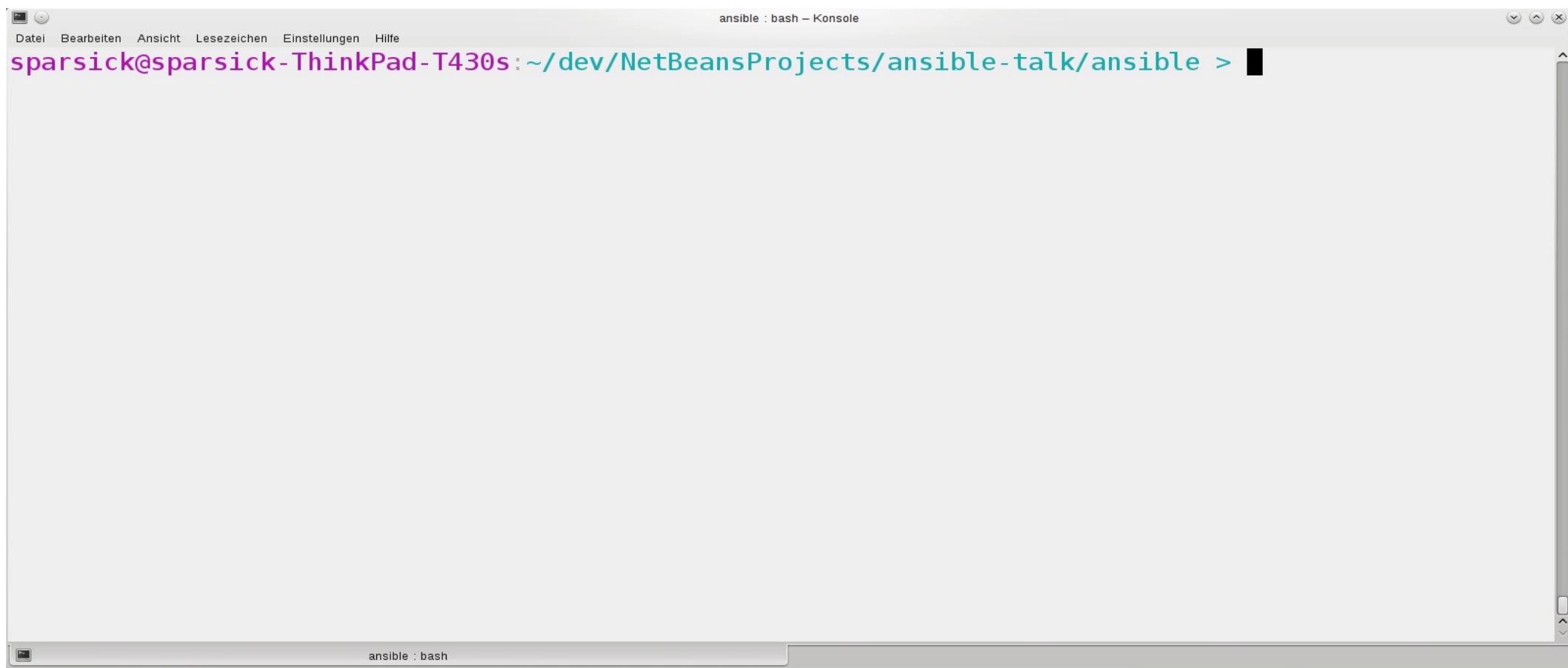


The screenshot shows a NetBeans IDE interface with a terminal window titled "ansible : bash – Konsole". The window contains the following text:

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

The terminal window has a standard Mac OS X style title bar with icons for close, minimize, and maximize. The main area of the window is white, and the text is black. The bottom of the window shows the NetBeans menu bar with items like Datei, Bearbeiten, Ansicht, Lesezeichen, Einstellungen, Hilfe, and a toolbar with several icons.

# Setup Database Server Playbook



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. Below the menu is a command line interface. The prompt is "sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >". The terminal window is set against a light gray background.

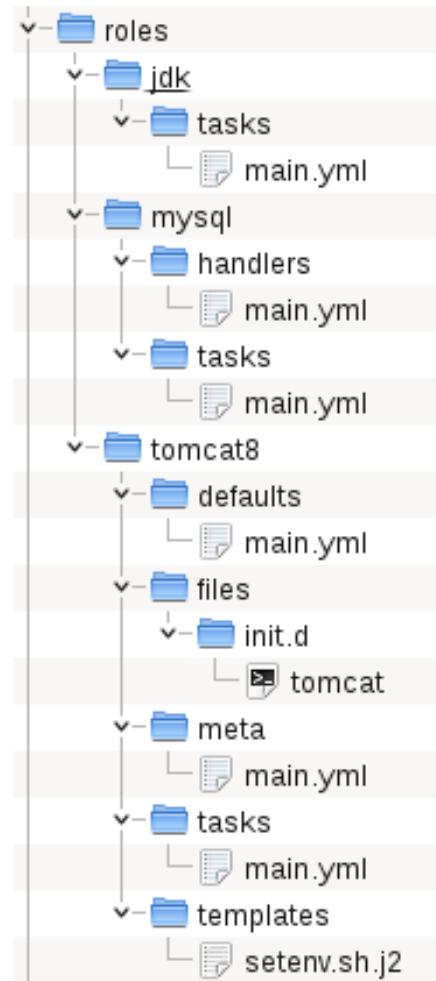
```
ansible : bash – Konsole
Datei  Bearbeiten  Ansicht  Lesezeichen  Einstellungen  Hilfe
sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >
```

```
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     become: yes
19     become_method: sudo
20
21   - name: Install Tomcat 8
22     unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name }} owner=vagrant group=vagrant
23
24   - name: Set link to tomcat 8
25     file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat state=link force=yes
26
27   - name: setup setenv.sh
28     template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh" src="roles/tomcat8/templates/setenv.sh.j2" mode=755
29     when: catalina_opts is defined
30
31   - find: paths="/opt/{{ tomcat_base_name }}/bin" patterns="*.sh"
32     register: result
33
34   - name: ensure tomcat scripts are executable
35     file: name={{item.path}} mode=755
36     with_items: '{{ result.files }}'
37
38   - name: install tomcat as service
39     copy: src=roles/tomcat8/files/tomcat.service dest=/etc/systemd/system/
40     become: yes
41     become_method: sudo
42
```

```
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=g3h3l1m 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
```

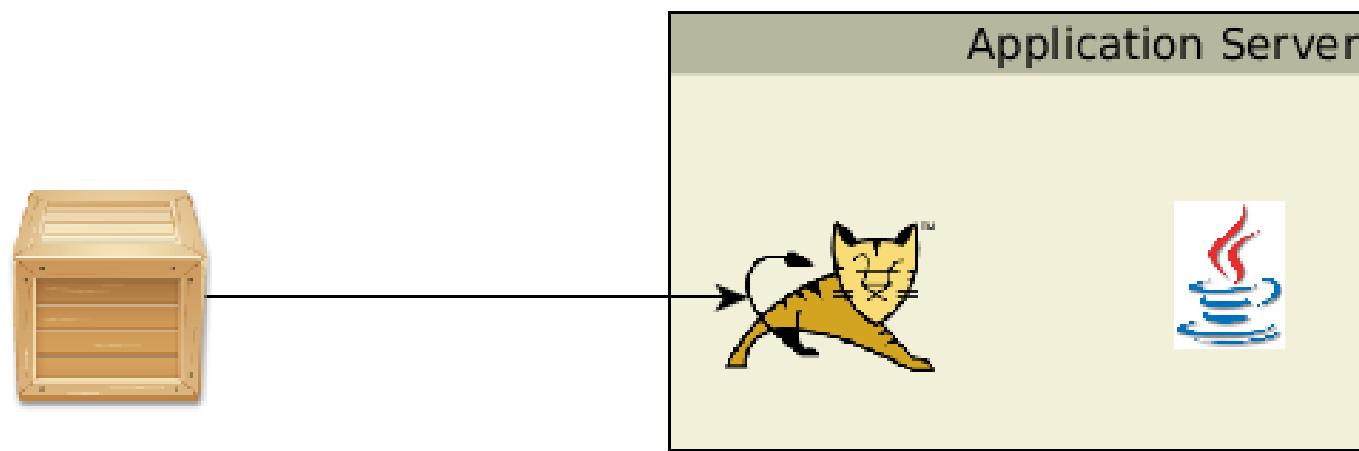
- Setup Database Server

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

# include\_role, import\_role (seit v2.3)

```
1 - hosts: webservers
2   tasks:
3     - include_role: # dynamic|
4       name: foo_app_instance
5     vars:
6       dir: '/opt/a'
7       app_port: 5000
8     - import_role: # static
9       name: foo
10
```

# 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

The image shows a file tree on the left and a code editor on the right.

**File Tree:**

- ansible/
  - roles/
    - deploy-on-tomcat/
      - defaults/main.yml
      - tasks/
        - cleanup-webapp.yml
        - deploy-webapp.yml
        - main.yml
        - start-tomcat.yml
        - stop-tomcat.yml
      - deploy-demo.yml

**Aktuelles Projekt (Current Project) View:**

```
1   include: stop-tomcat.yml
2   - include: cleanup-webapp.yml
3   - include: deploy-webapp.yml
4   - include: start-tomcat.yml
5
```

# deploy-on-tomcat Role

The screenshot shows the file structure and content of the `deploy-on-tomcat` role within an `ansible` project. The tree view on the left lists the following files and folders:

- `ansible`
- `roles`
  - `deploy-on-tomcat`
    - `defaults`
    - `main.yml` (selected)
    - `tasks`
      - `cleanup-webapp.yml`
      - `deploy-webapp.yml`
      - `main.yml` (selected)
      - `start-tomcat.yml`
      - `stop-tomcat.yml` (selected)
  - `deploy-demo.yml`

The right panel displays the YAML code for the selected `main.yml` file under `tasks`:

```
Aktuelles Projekt
1   - name: stop tomcat
2     # service: name=tomcat state=stopped
3     command: service tomcat stop
4
5   - name: wait tomcat shutdown
6     wait_for: port=8080 state=stopped timeout=60
7
```

# deploy-on-tomcat Role

The screenshot shows the file structure and content of the `deploy-on-tomcat` role within an Ansible project. The left pane displays the directory tree:

- `ansible`
- `roles`
- `deploy-on-tomcat`
  - `defaults`
  - `main.yml` (selected)
  - `tasks`
    - `cleanup-webapp.yml` (selected)
    - `deploy-webapp.yml`
    - `main.yml`
    - `start-tomcat.yml`
    - `stop-tomcat.yml`
  - `deploy-demo.yml`

The right pane shows the code content for the selected file, `cleanup-webapp.yml`:

```
1   name: cleanup {{ webapp_target_name }}
2     file: name={{tomcat_app_base}}/{{{
3       webapp_target_name }} state=absent
```

# deploy-on-tomcat Role

The image shows a file explorer on the left and a code editor on the right.

**File Explorer:**

- ansible
- roles
- deploy-on-tomcat
- defaults
- main.yml
- tasks
- cleanup-webapp.yml
- deploy-webapp.yml
- main.yml
- start-tomcat.yml
- stop-tomcat.yml
- deploy-demo.yml

**Code Editor (Aktuelles Projekt):**

```
1   - name: delete previous backup
2     file: path={{ tomcat_app_base }}/{{ webapp_target_name }}.war.previous state=absent
3
4   - name: create new backup
5     command: mv {{ tomcat_app_base }}/{{ webapp_target_name }}.war {{ tomcat_app_base }}/{{ webapp_target_name }}.war.previous
6     ignore_errors: yes
7
8   - name: copy webapp {{ webapp_source_path }} to {{ webapp_target_name }}
9     copy: src={{ webapp_source_path }} dest={{ tomcat_app_base }}/{{ webapp_target_name }}.war mode=660
10
```

# deploy-on-tomcat Role

The screenshot shows the file structure and content of an Ansible role named 'deploy-on-tomcat'. The structure is as follows:

- ansible/roles/deploy-on-tomcat/
  - defaults/main.yml
  - tasks/cleanup-webapp.yml
  - tasks/deploy-webapp.yml
  - tasks/main.yml
  - tasks/start-tomcat.yml (highlighted in blue)
  - tasks/stop-tomcat.yml
- deploy-demo.yml

The right pane displays the content of the 'start-tomcat.yml' file:

```
1   - name: start tomcat
2     #service: name=tomcat state=started
3     command: service tomcat start
4
5   - name: wait for tomcat to start
6     wait_for: port=8080 timeout=60
7
```

# deploy-on-tomcat Role

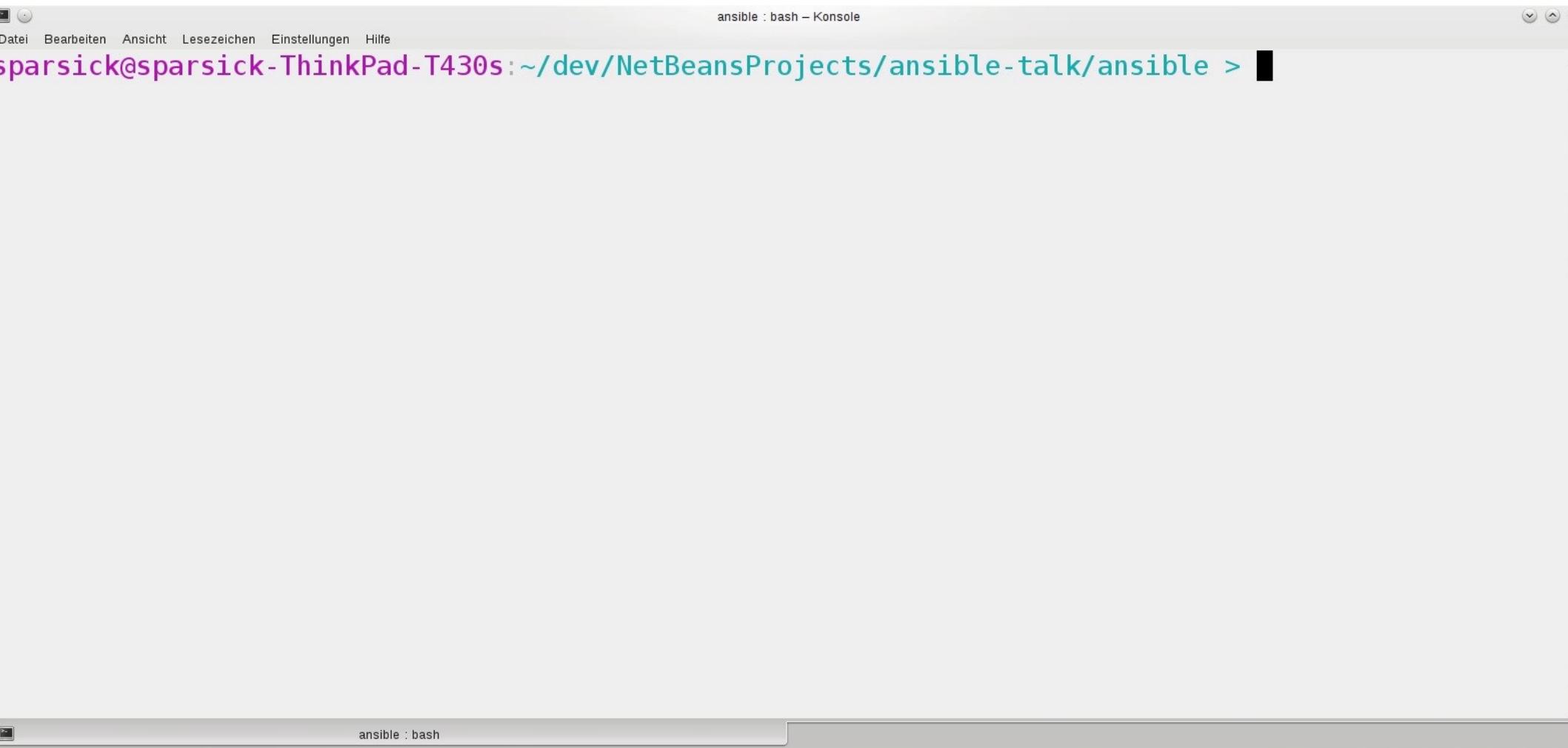
The screenshot shows the file structure of an Ansible role named 'deploy-on-tomcat'. The structure is as follows:

- ansible/roles/deploy-on-tomcat/
  - defaults/main.yml
  - tasks/cleanup-webapp.yml
  - tasks/deploy-webapp.yml
  - tasks/main.yml
  - tasks/start-tomcat.yml
  - tasks/stop-tomcat.yml
  - deploy-demo.yml

On the right side, there is a preview window titled "Aktuelles Projekt" (Current Project) containing the following configuration:

```
1 webapp_target_name: demo
2 tomcat_app_base: /opt/tomcat/webapps
```

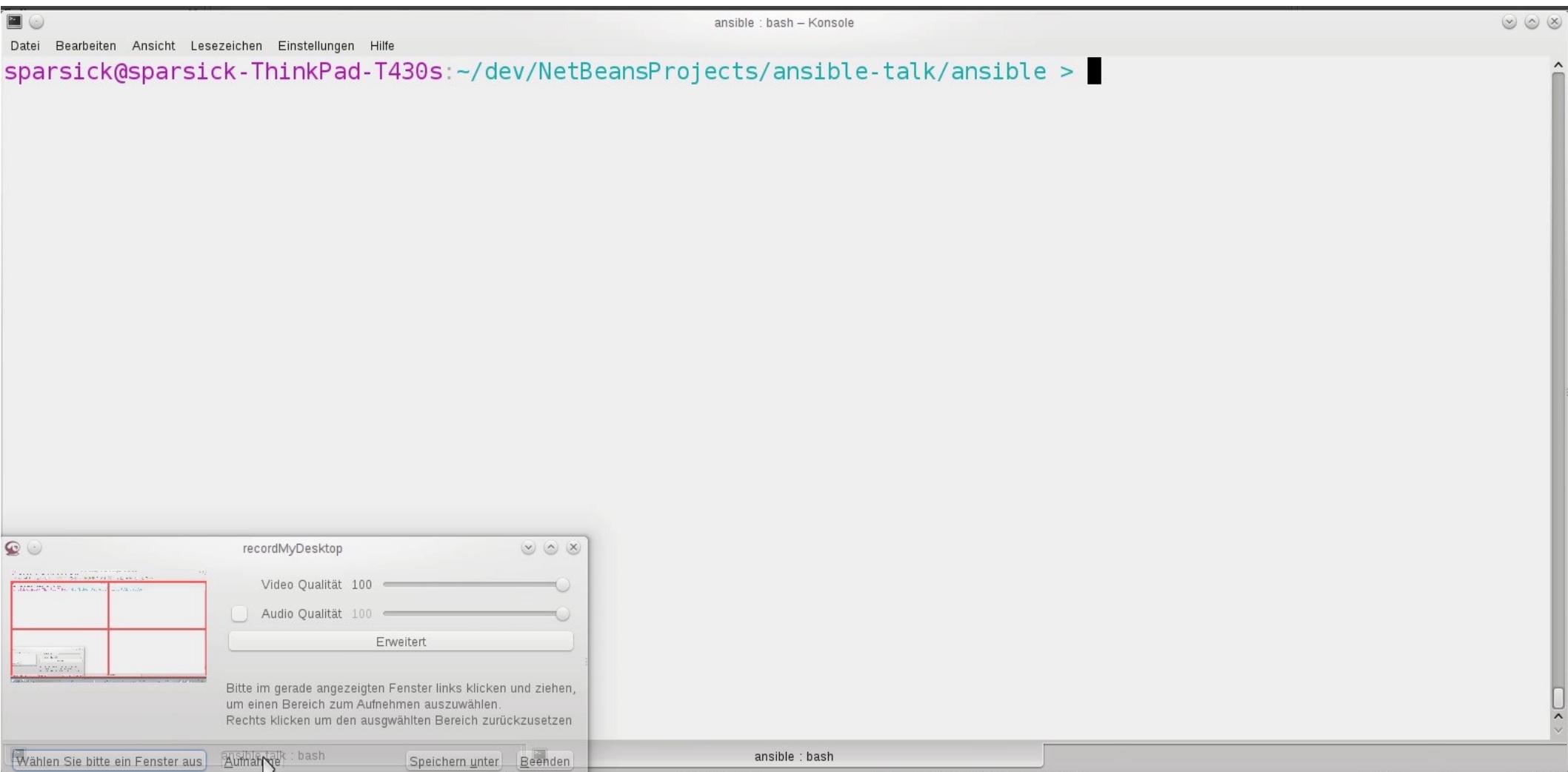
# Deploy Application Playbook



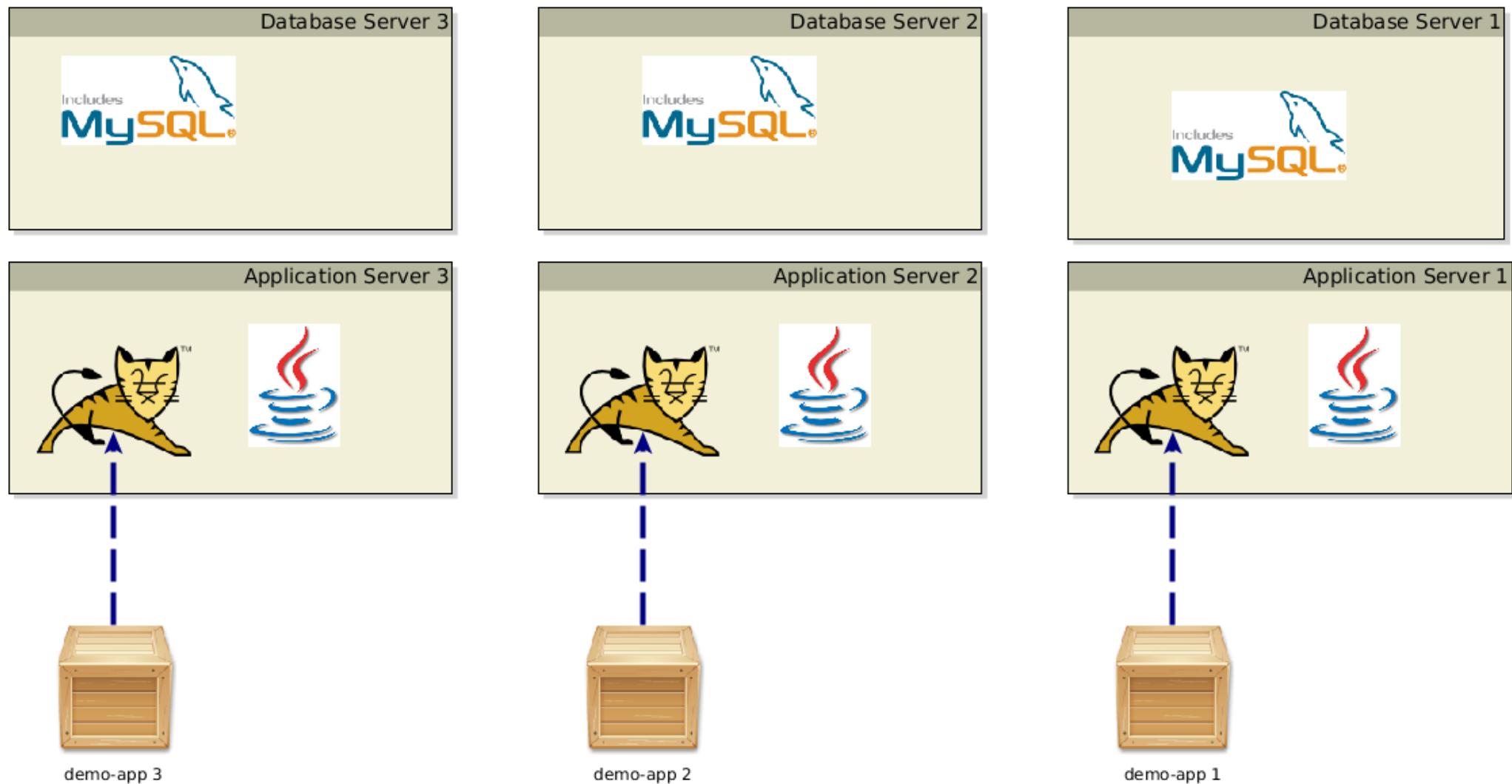
The screenshot shows a terminal window titled "ansible : bash – Konsole". The window has a standard OS X style with a menu bar at the top. The menu bar includes "Datei", "Bearbeiten", "Ansicht", "Lesezeichen", "Einstellungen", and "Hilfe". Below the menu bar, the title bar displays the window title. The main area of the terminal is a command line interface. The prompt shows the user's name "sparsick" followed by the host name "sparsick@sparsick-ThinkPad-T430s", the directory path "("~/dev/NetBeansProjects/ansible-talk/ansible)", and a greater than sign ">". A black vertical bar is positioned to the right of the prompt, likely indicating where the cursor is located.

```
ansible : bash – Konsole
sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >
```

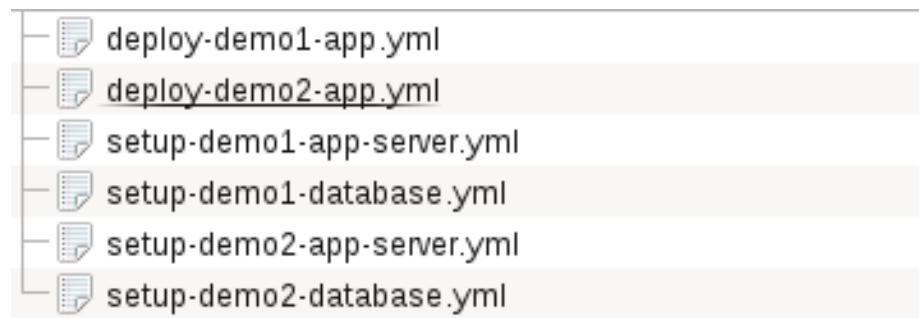
# Ad-hoc-Kommando



# Warum Roles?



# Warum Roles?

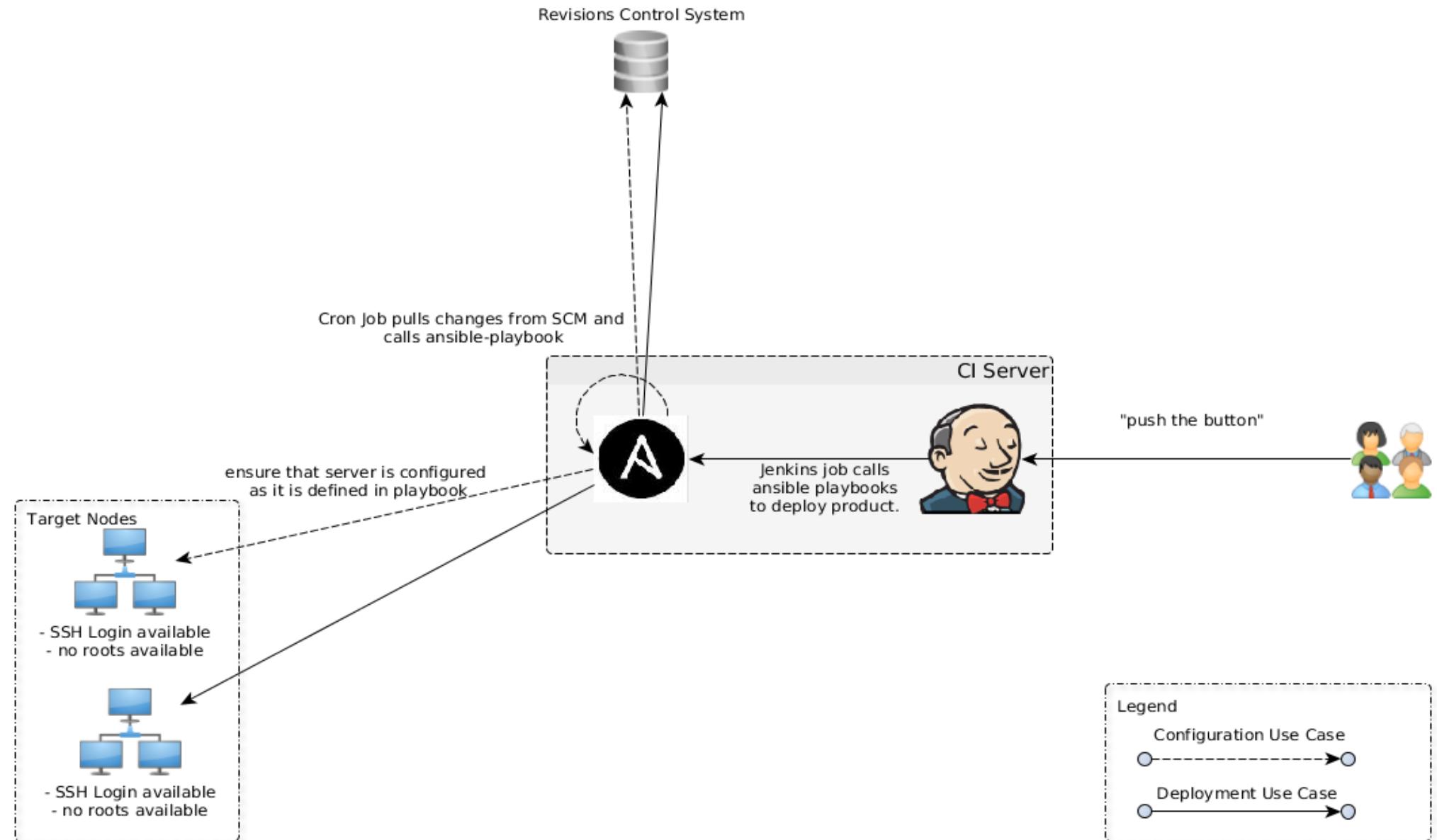


# Warum Roles?

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

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

Hosts License

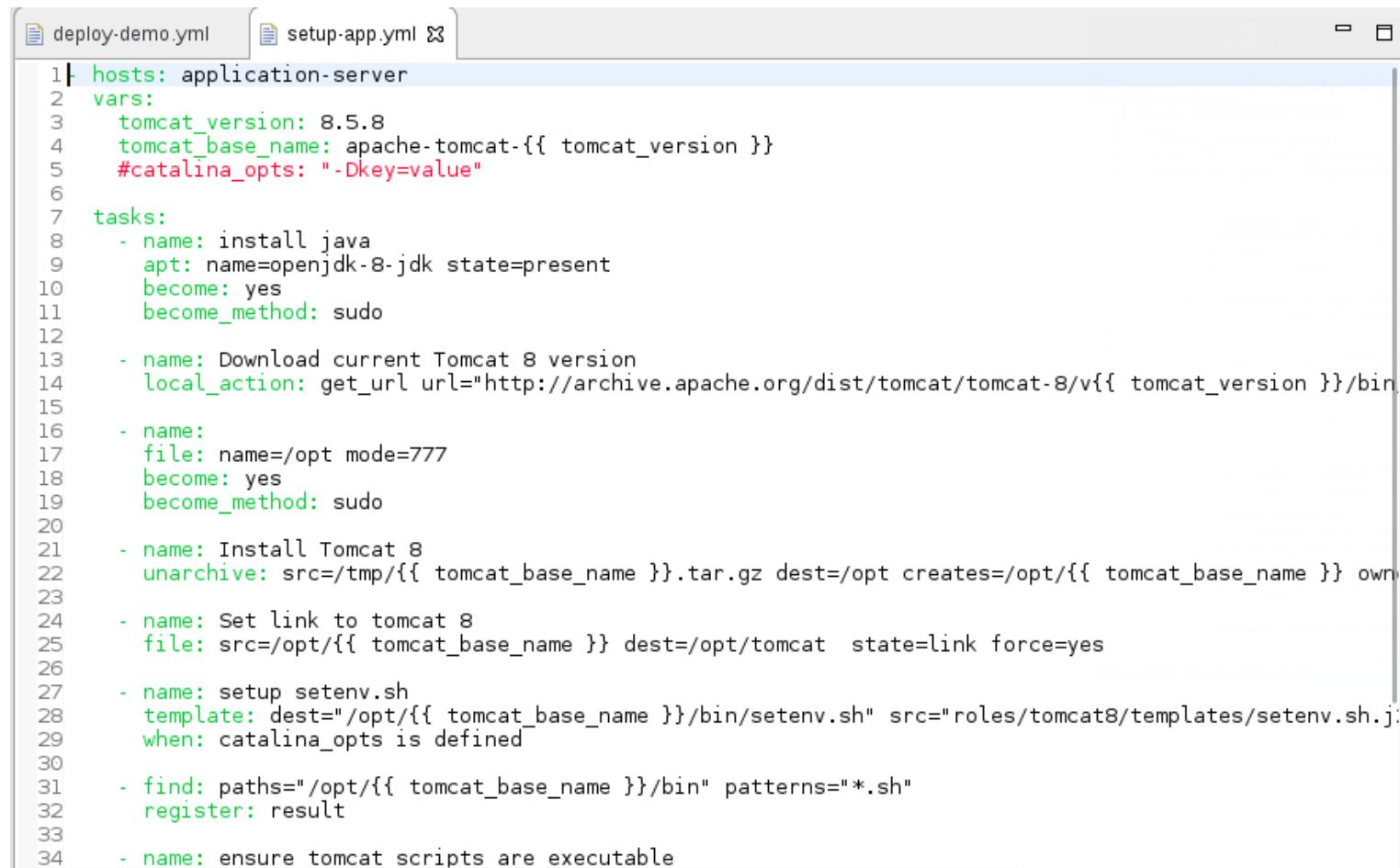
Time

# IDE - Support

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

# IDE-Support - Eclipse

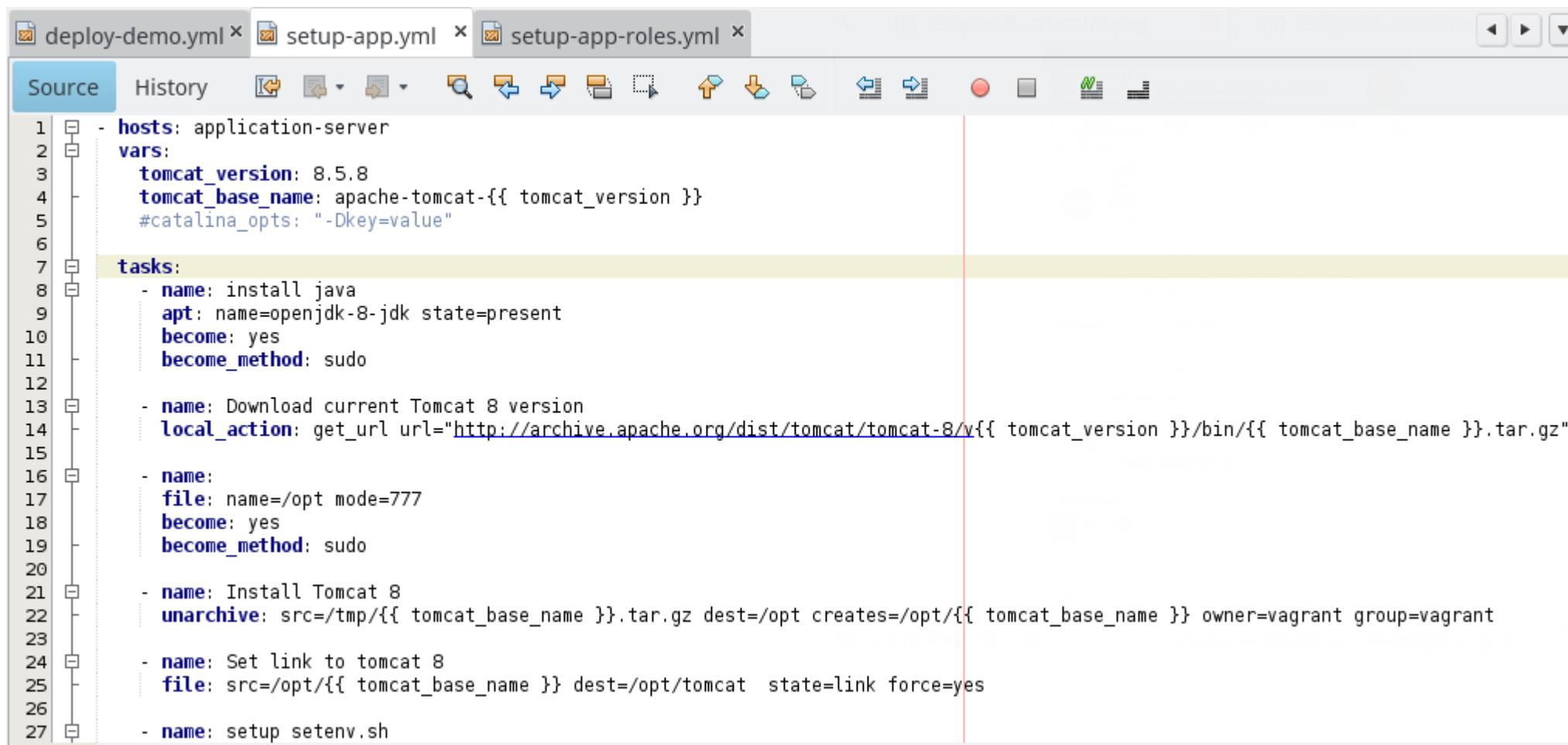
- YAML Support über Plugin *yEdit*



```
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 following details:

- Project Bar:** Shows the project name "ansible" and the file "setup-app-roles.yml".
- Toolbars:** Standard IntelliJ toolbars for Project, Packages, and Project Files.
- File Tabs:** Multiple tabs are open: "setup-app.yml", "setup-app-roles.yml" (the active tab), "setup-db.yml", and "main.yml".
- Structure View:** On the left, it shows the project structure with folders like ".idea", "group\_vars", "host\_vars", "inventories", "roles", "spec", and files like ".project", ".rspec", "ansible.iml", "demo-app-ansible-deploy-1.0-SNAPSHOT.war", "deploy-demo.yml", "Rakefile", "setup-app.yml", "setup-app-roles.yml", "setup-db.yml", and "setup-db-roles.yml".
- Code Editor:** The editor shows a snippet of Ansible YAML code:

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

A code completion dropdown is open at the end of the third role entry, listing suggestions: "deploy-on-tomcat", "jdk", "mysql", "roles", and "tomcat8". A tooltip at the bottom right of the dropdown says: "Dot, space and some other keys will also close this lookup and be inserted into editor".

# 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. The cursor is positioned over the word 'file' in a task definition, which has highlighted the entire line: `file: name=/opt mode=777`. A tooltip labeled 'Documentation for file' appears, providing details about the 'file' module. The tooltip content includes:

**file - Sets attributes of files**

- Synopsis
- Options
- Examples
- Notes

**file Synopsis**

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

# 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 buttons for "roles" and "role". A tooltip is displayed over the "role" button, showing the path "/ansible/roles/tomcat8/tasks/main.yml". The main editor area contains Ansible YAML code:

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

# 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



# 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

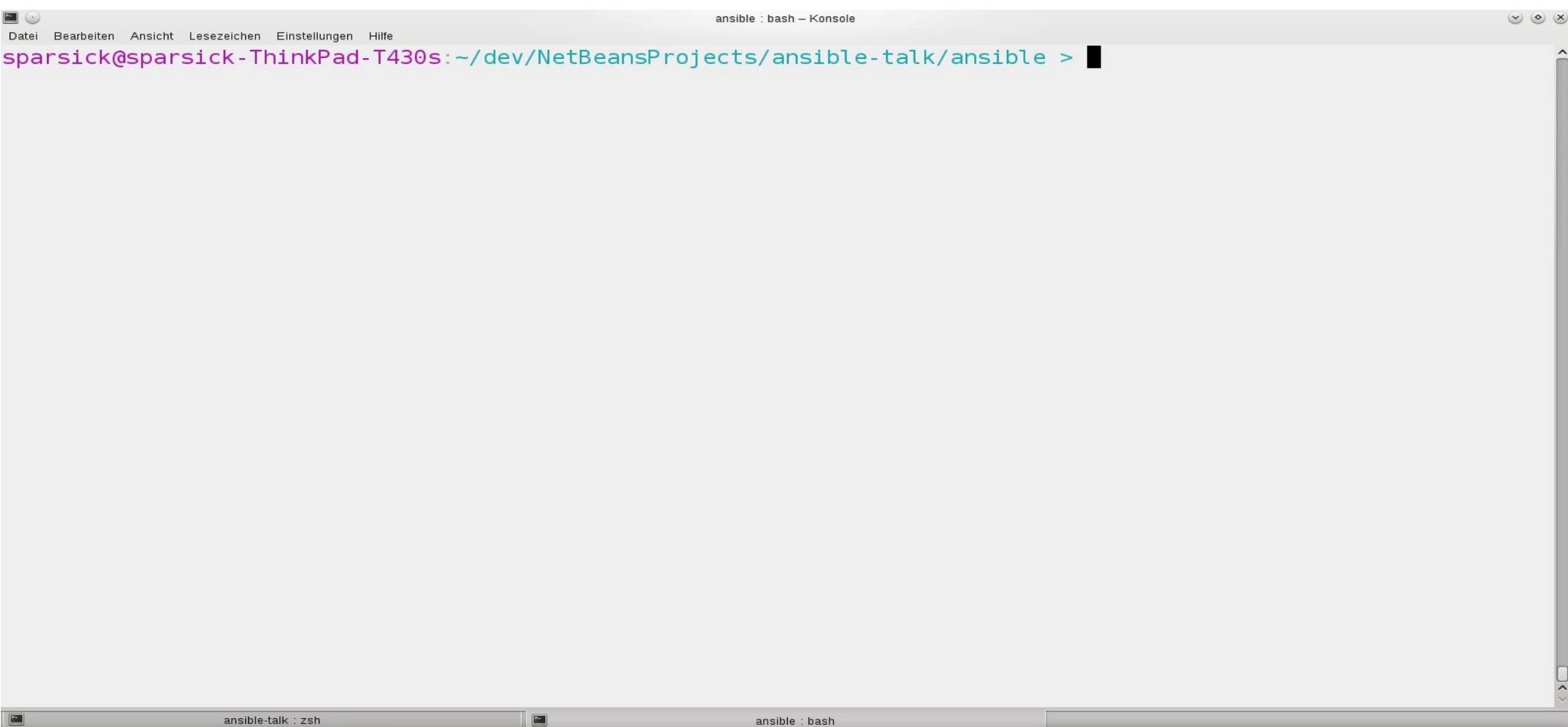
[ANSIBLE0009] Octal file permissions must contain leading zero
/home/sparsick/dev/NetBeansProjects/ansible-talk/ansible/roles/tomcat8/tasks/main.yml:22
Task/Handler: ensure tomcat scripts are executable
```

# ServerSpec Tests

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

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

# ServerSpec Tests



# testinfra

```
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")
```

```
1 def test_openjdk_is_installed(host):
2     openjdk = host.package("openjdk-8-jdk")
3     assert openjdk.is_installed
4
5 def test_tomcat_service_exists(host):
6     assert host.file("/etc/systemd/system/tomcat.service").exists
7
8 def test_tomcat_folder_exists(host):
9     assert host.file("/opt/tomcat").exists
```

# testinfra

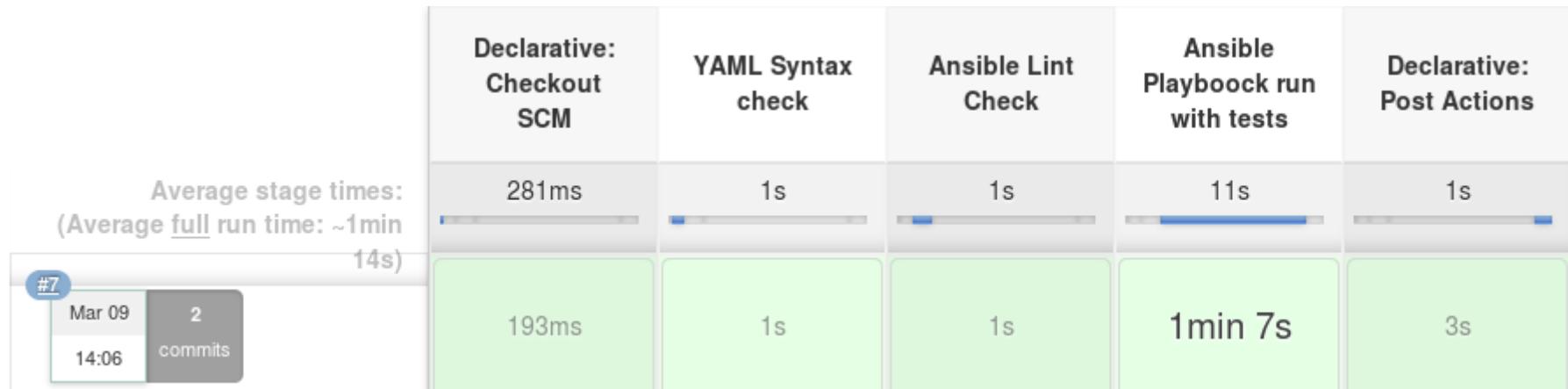
```
x sparsick@sparsick-ThinkPad-T460s ~/dev/NetBeansProjects/ansible-talk/ansible master py.test --connection=ansible  
--ansible-inventory inventories/test -v tests/*.py  
===== test session starts =====  
platform linux2 -- Python 2.7.12, pytest-3.4.2, py-1.5.2, pluggy-0.6.0 -- /usr/bin/python  
cachedir: .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

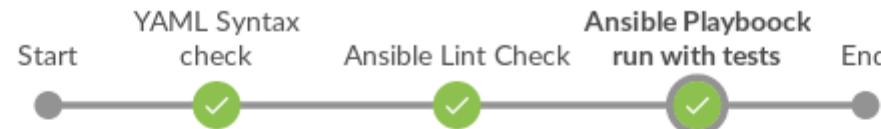
---

```
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

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



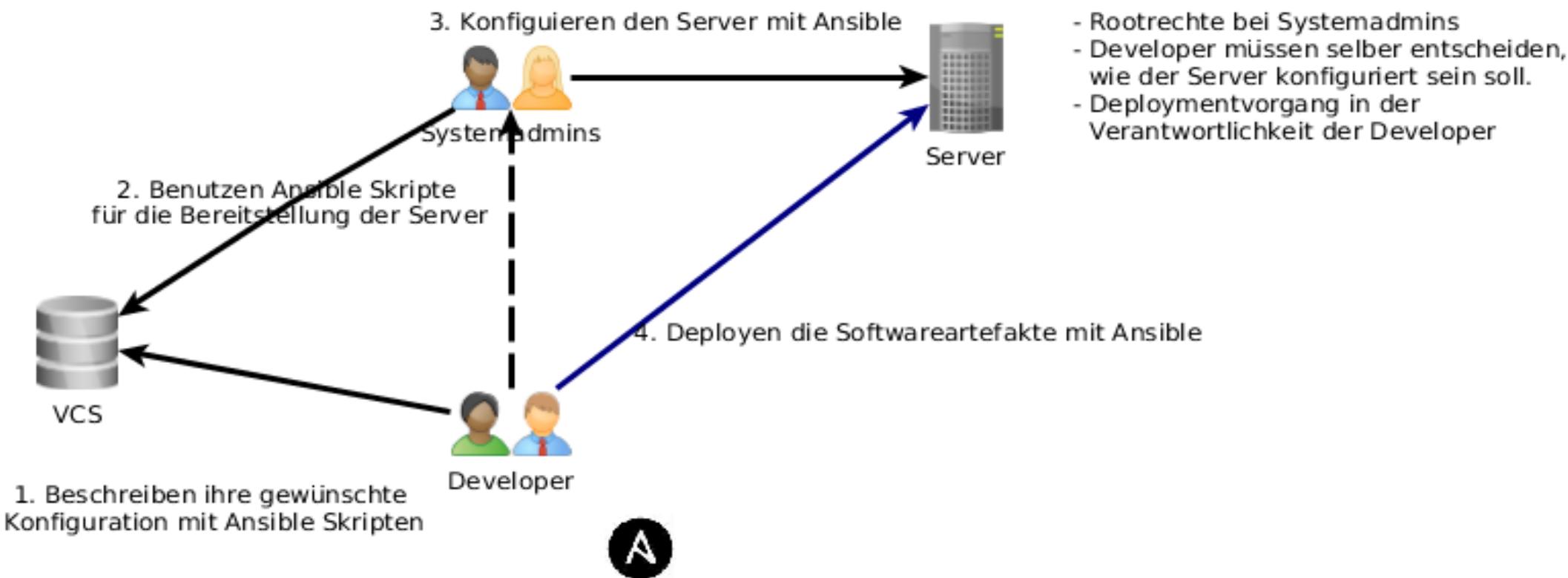
```
1 - hosts: 192.168.33.10
2   become: yes
3   become_method: sudo
4   tasks:
5     - name: Ensure Python modules
6       apt:
7         name: python-apt
8
9     - name: add nodejs ppa
10    apt_repository:
11      repo: ppa:chris-lea/node.js
12
13    - name: install nodejs package
14      apt:
15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

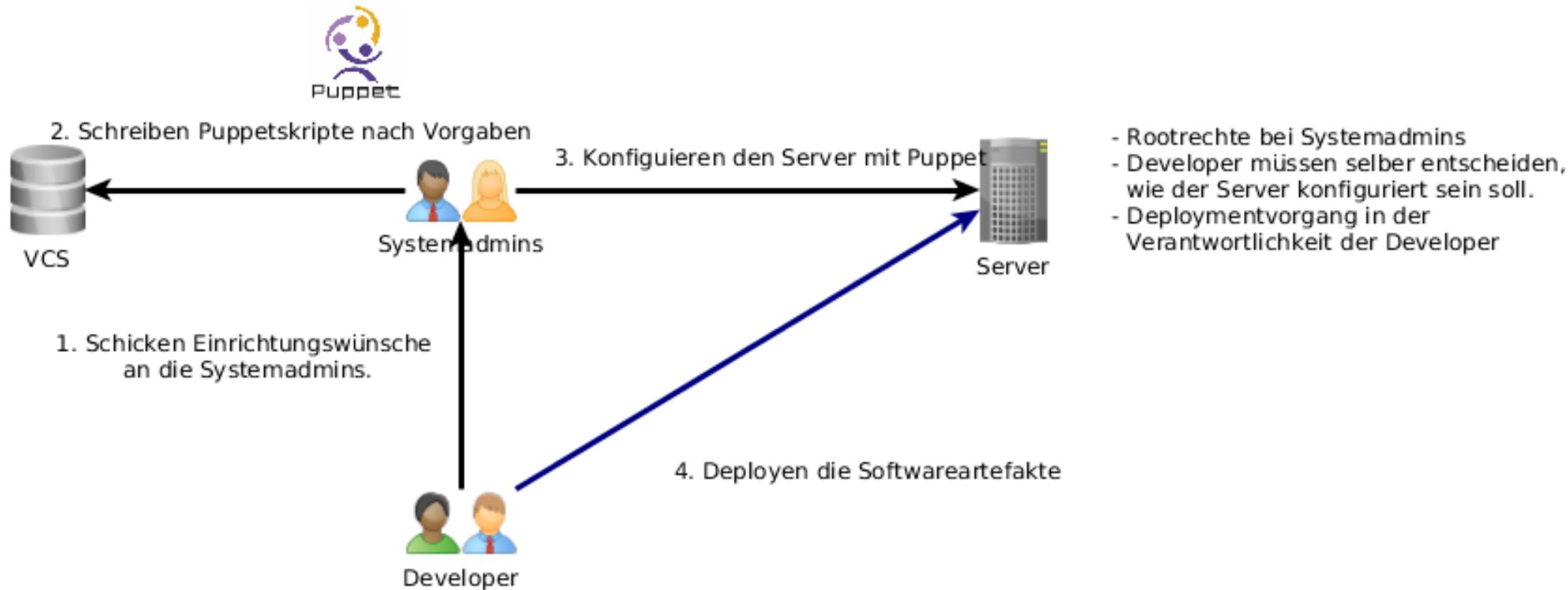
# Systemkonfiguration für Entwickler

## Lösungidee mit Ansible



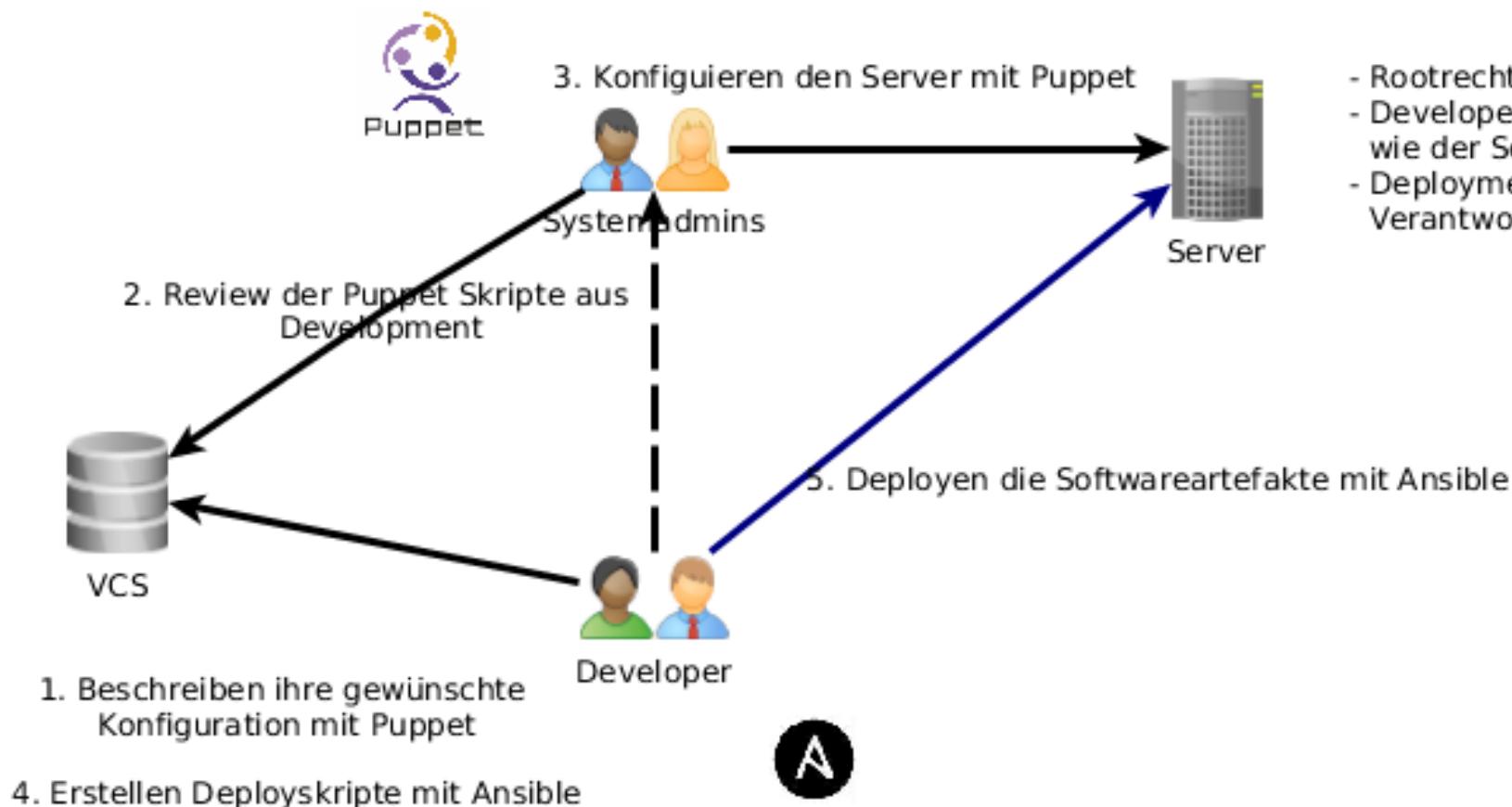
# Systemkonfiguration für Entwickler

## Variante - Prozess zwischen Development und Operation



# Systemkonfiguration für Entwickler

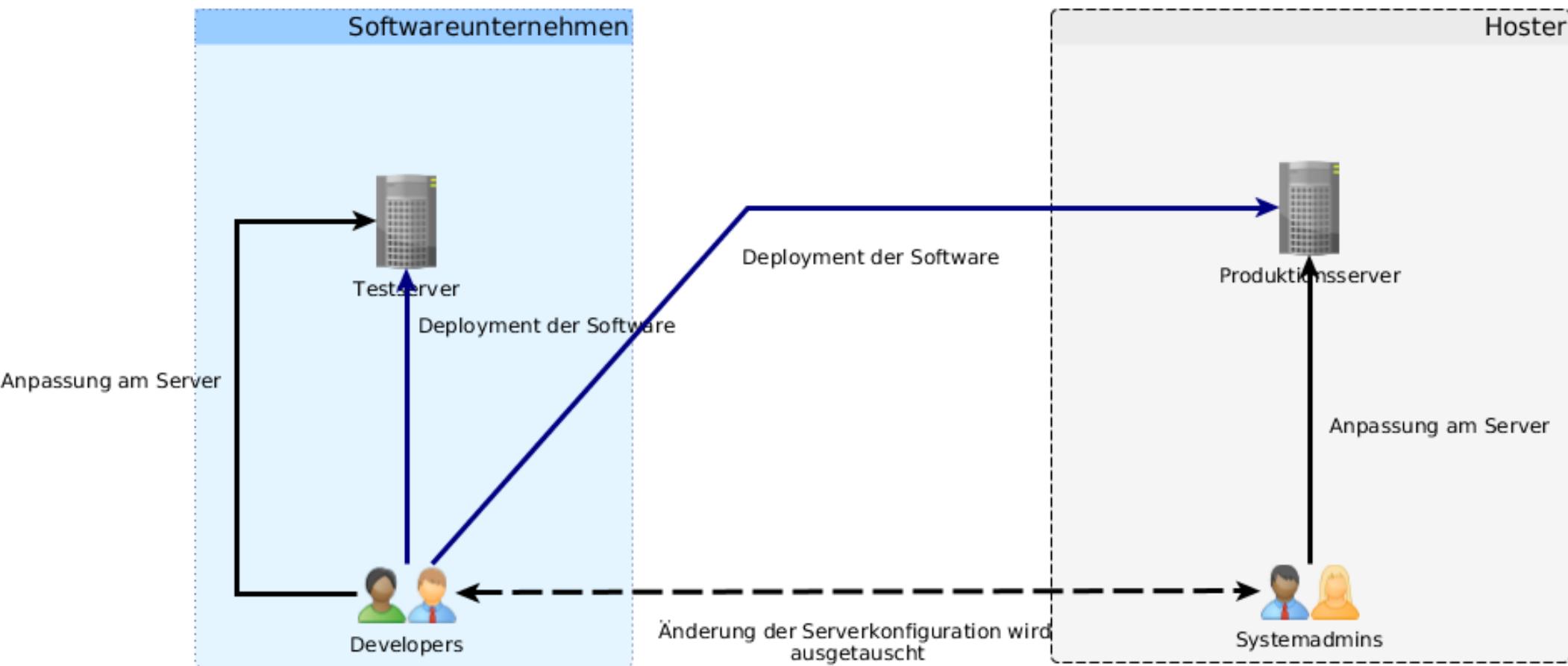
## Lösungsvariante



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

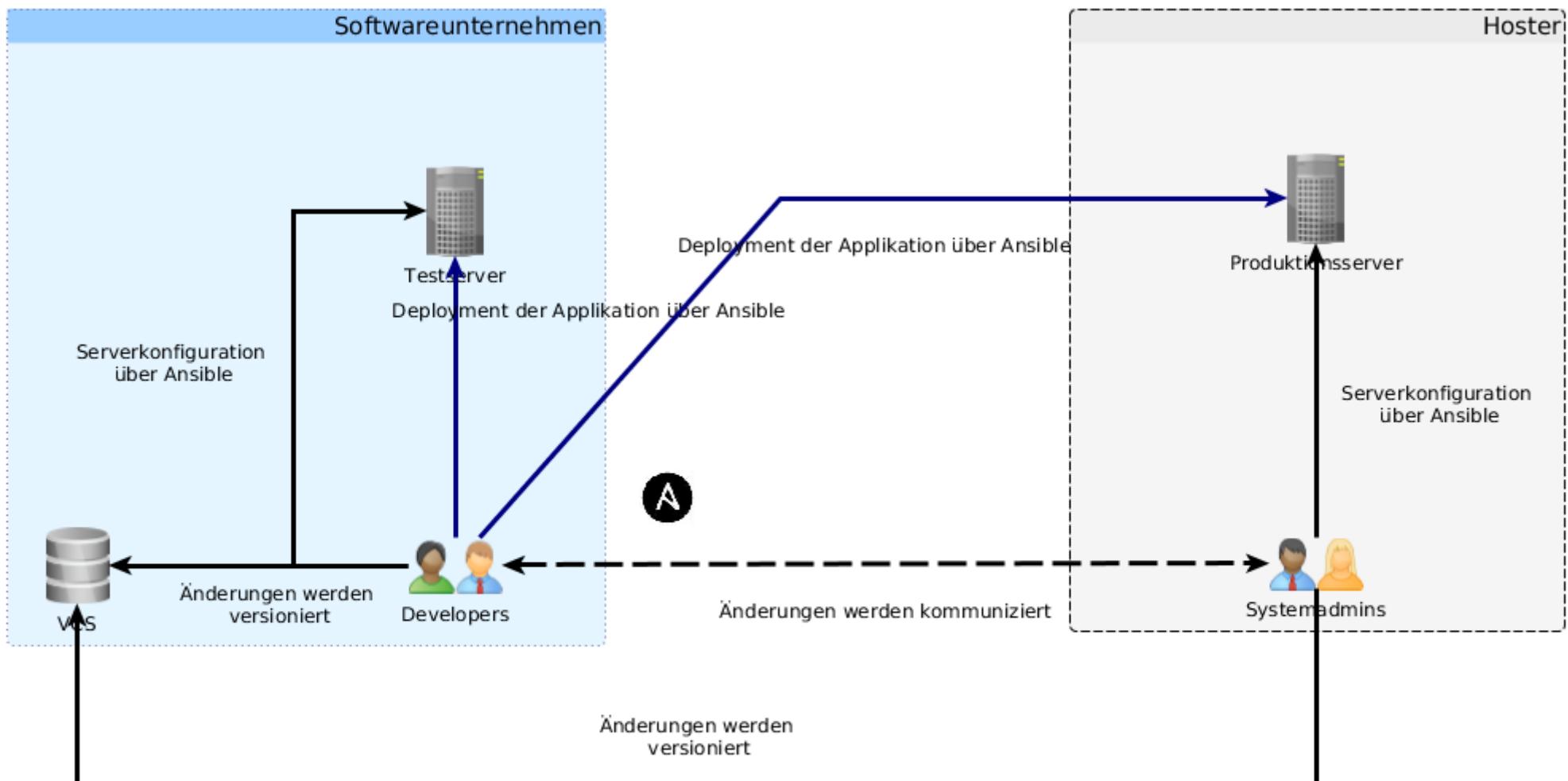
# Systemkonfiguration für Entwickler

Produktionsserver sind beim externen Hoster



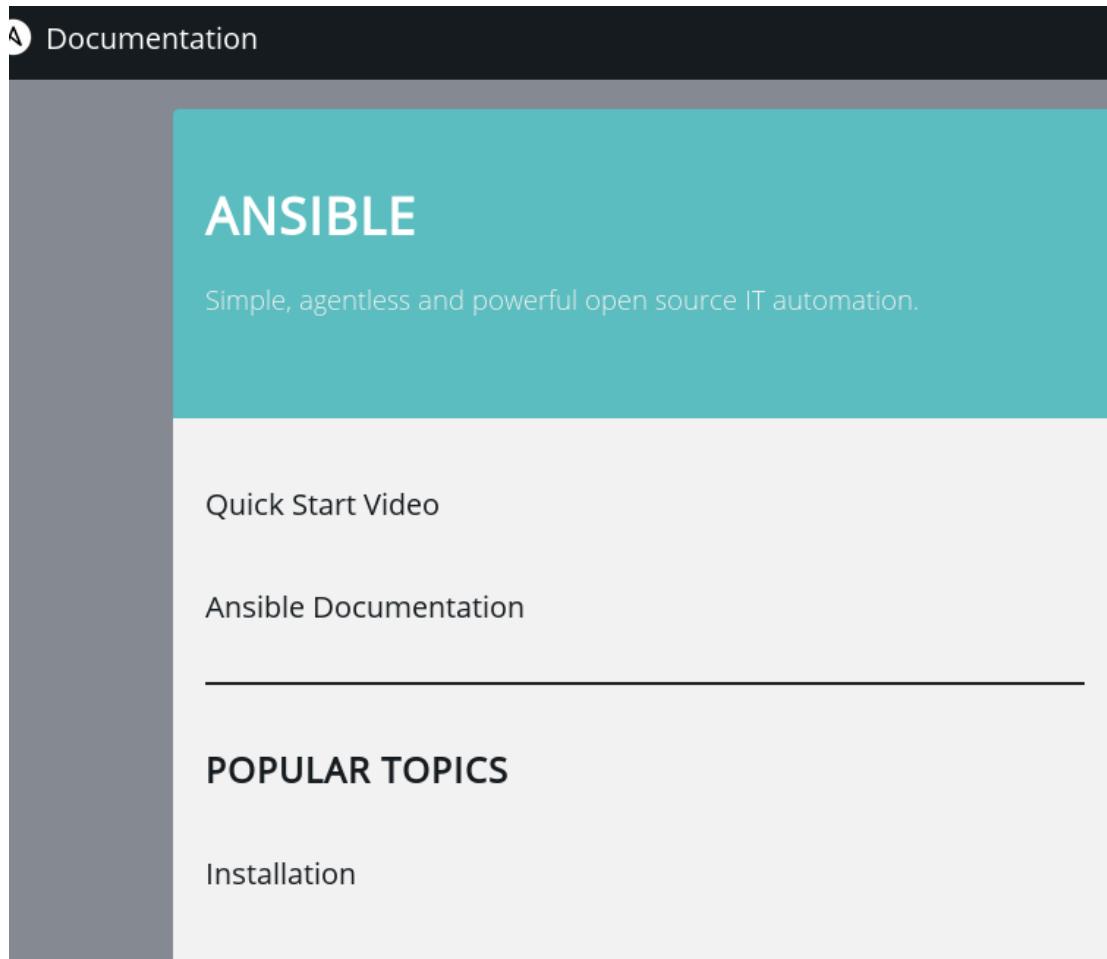
# Systemkonfiguration für Entwickler

## Lösungsidee



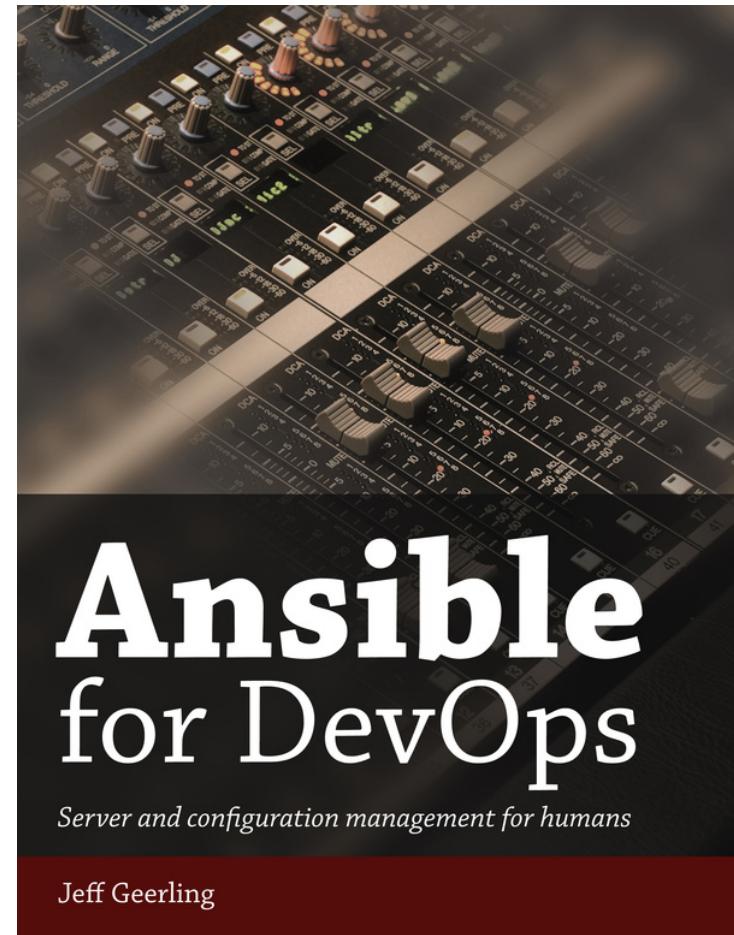
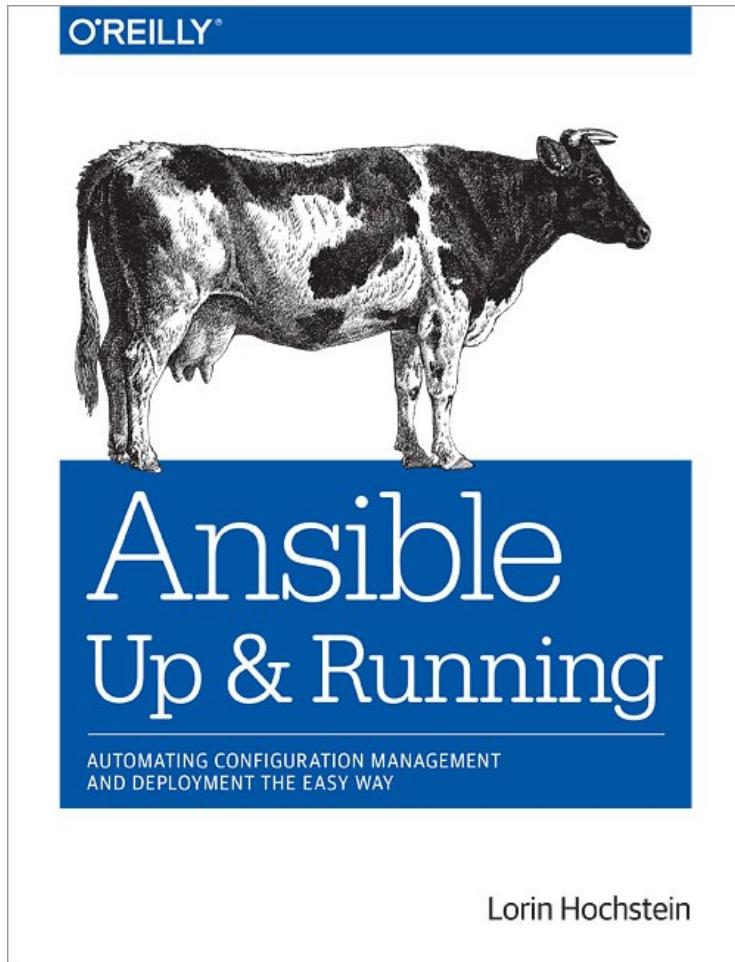
- Hoster verantwortlich für die Systemkonfiguration
- Softwareunternehmen verantwortlich für das Deployment
- Synchronisation zwischen Testserver und Produktionsserver wird vereinfacht

# Weitere Informationen



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

# Weitere Informationen



# Weitere Informationen

The image shows the front cover of the Java aktuell magazine, Winter 2016 issue. 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' and the tagline 'Aus der Community – für die Community' are visible. 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, and databases. The text 'Java ist vie' is partially visible above the arrows. At the bottom, there's an article summary: 'Ansible – warum Konfigurationsmanagement auch für Entwickler interessant sein kann' by Sandra Parsick. The right edge of the cover has some small text and a barcode.

04-2016 | Winter | www.ijug.eu

Java aktuell

Praxis. Wissen. Networking. Das Magazin für Entwickler  
Aus der Community – für die Community

Java ist vie

Software organisieren

Ansible – warum Konfigurationsmanagement auch für Entwickler interessant sein kann

Sandra Parsick

D: 4,90 EUR A: 5,60 EUR CH: 9,80 CHF Suisse: 5,30 EUR ISSN 2191-4977



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



JUnit 5  
Das nächste große Release steht vor der Tür

Ansible  
Konfigurationsmanagement auch für Entwickler

Spring Boot Starter  
Komfortable Modularisierung und Konfiguration



# Fragen?

@SandraParsick

mail@sandra-parsick.de

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