

JUG Essen, 30.05.2017

Ansible für Java-Entwickler

Sandra Parsick

mail@sandra-parsick.de

@SandraParsick

Zur meiner Person

- Sandra Parsick
- Freiberuflicher Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
 - Java Enterprise Anwendungen
 - Agile Methoden
 - Software Craftmanship
 - Automatisierung von Entwicklungsprozessen
- Trainings
- Workshops
- Softwerkskammer Ruhrgebiet
- Twitter: @SandraParsick
- Blog:
<http://blog.sandra-parsick.de>
- E-Mail: mail@sandra-parsick.de



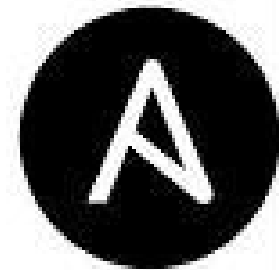
Agenda

1. Ansible – Was ist das?
2. Warum ist es für Entwickler interessant?
3. Einführung in Ansible
4. Wie unterscheidet sich Ansible zu seiner Konkurrenz?
5. Weitere Einsatzszenarien aus Entwicklersicht

Ansible
Was ist das?

Ansible

- Software für
 - Konfigurationsmanagement,
 - Softwareverteilung und
 - Ad-hoc-Kommando-Ausführung



Konfigurationsmanagement (KM)

„Das KM umfasst alle technischen, organisatorischen und beschlussfassenden Maßnahmen und Strukturen, die sich mit der Konfiguration (Spezifikation) eines Produkts befassen.“

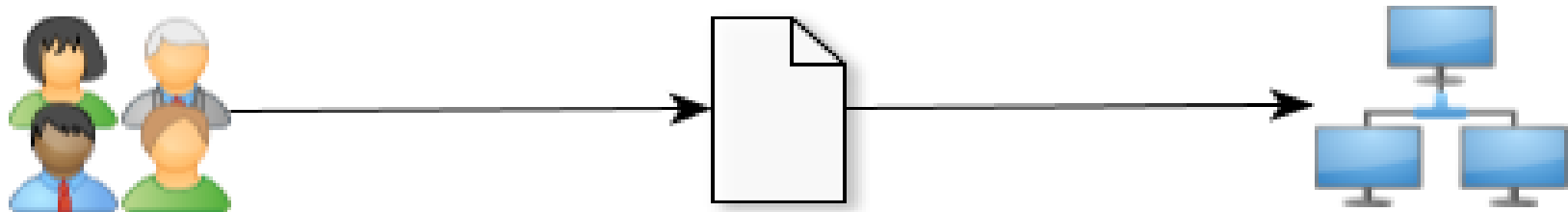
<https://www.projektmagazin.de/glossarterm/konfigurationsmanagement>

Konfigurationsmanagement (KM)

- Softwarekonfiguration
- Hardwarekonfiguration
- Dienstleistungskonfiguration
- Systemkonfiguration

Systemkonfiguration

- „Infrastructure As Code“



Systemkonfiguration

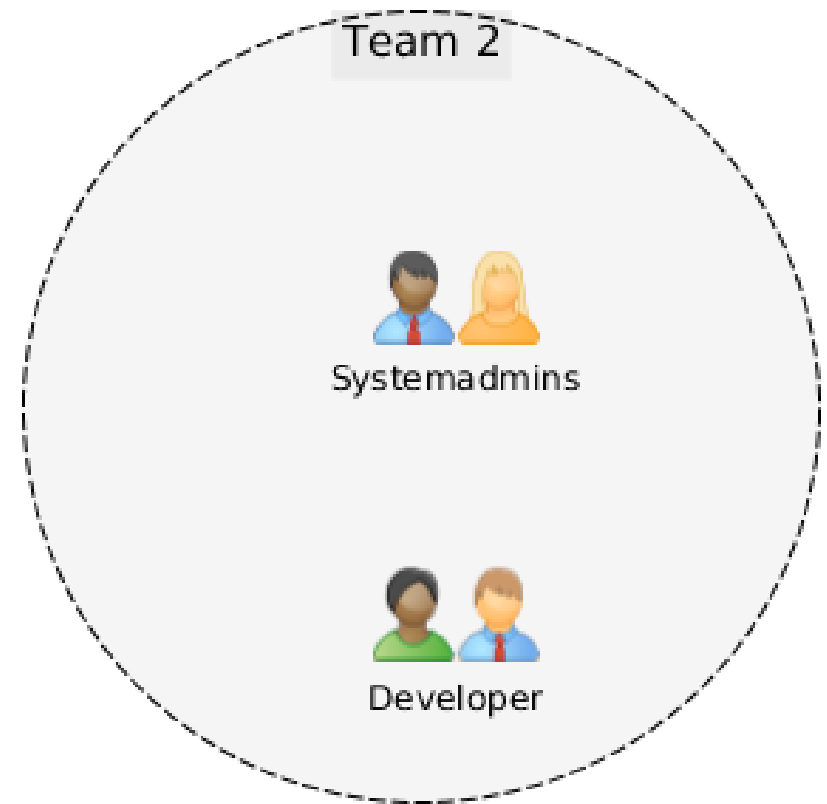
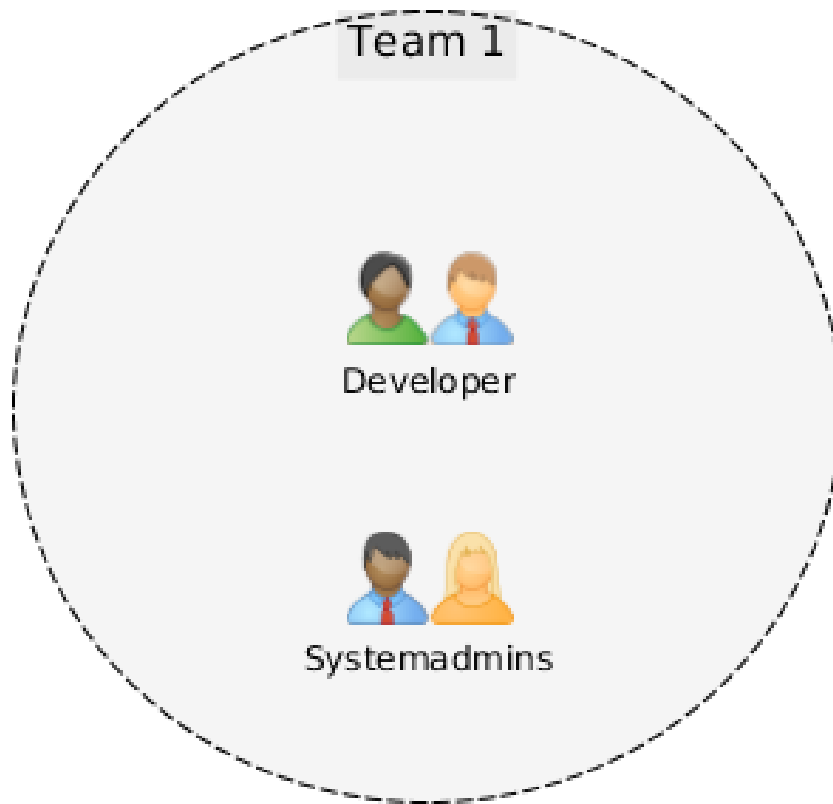
- „Infrastructure As Code“



Warum ist es für Entwickler
interessant?

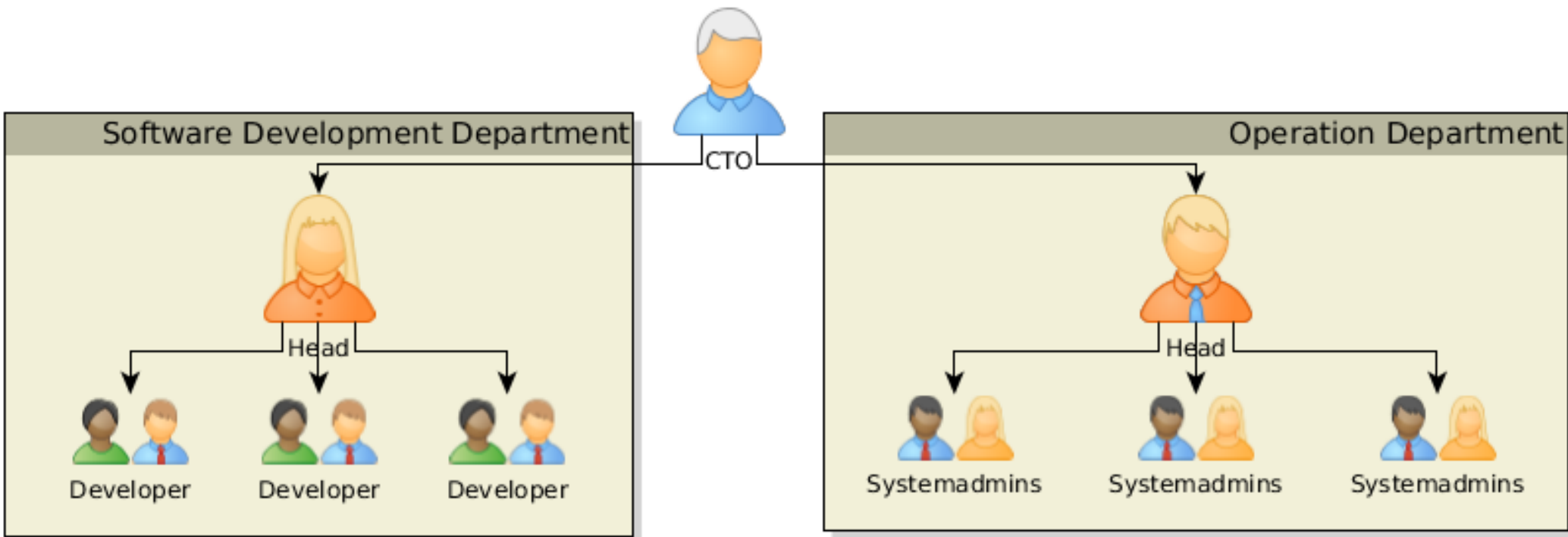
Systemkonfiguration für Entwickler

Organisatorische Ausgangslage
Wunsch



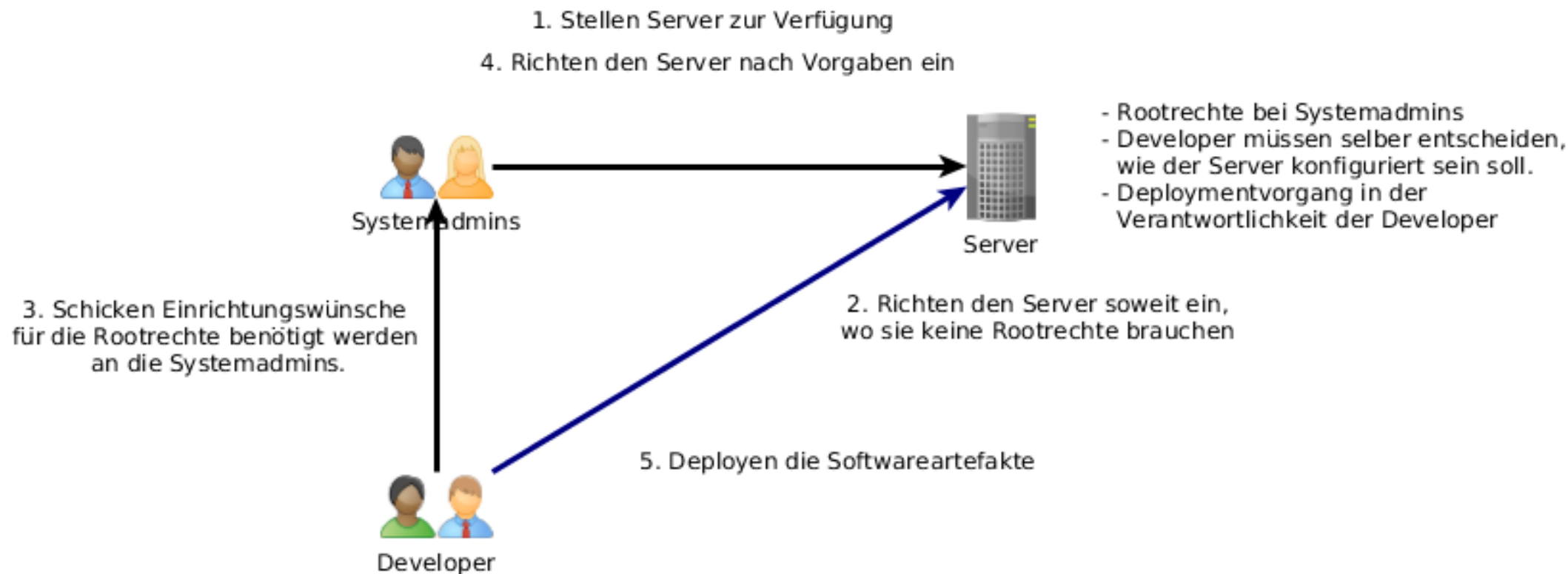
Systemkonfiguration für Entwickler

Organisatorische Ausgangslage
Realität



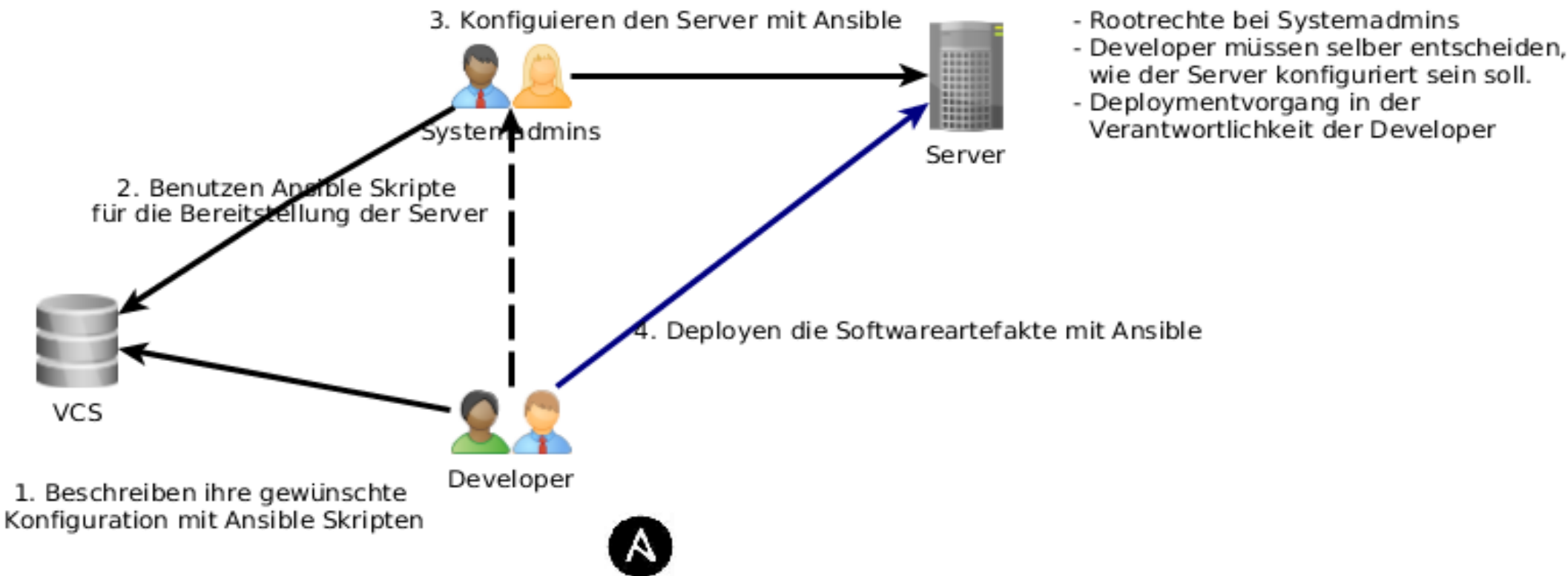
Systemkonfiguration für Entwickler

Prozess zwischen Development und Operation



Systemkonfiguration für Entwickler

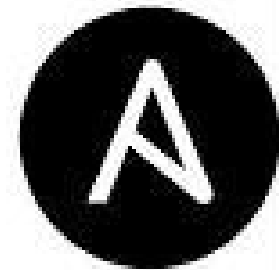
Lösungsidee mit Ansible



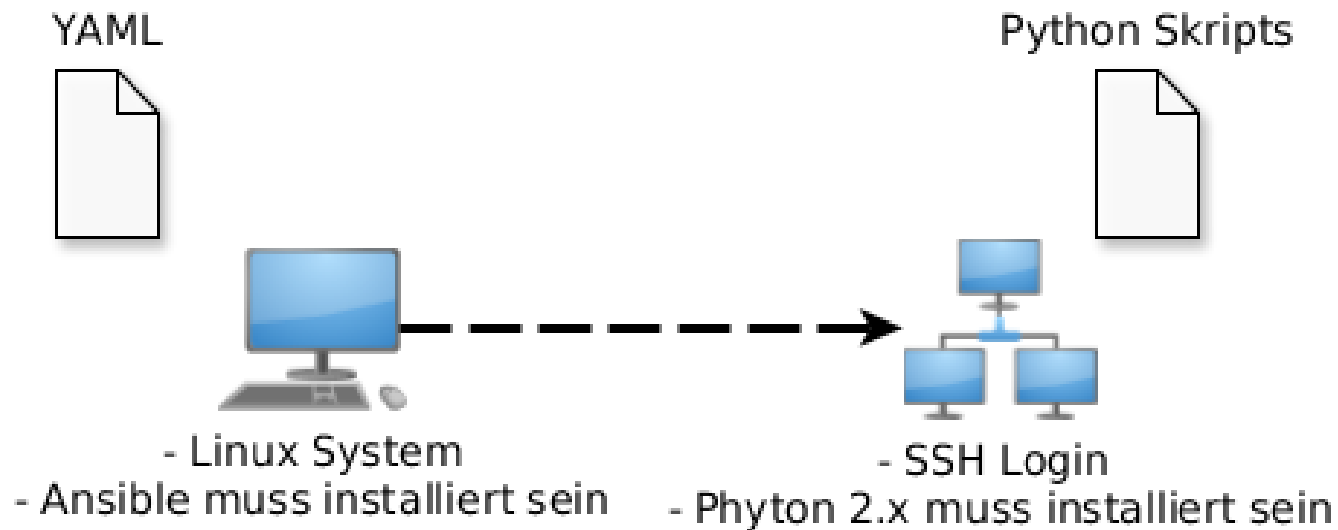
Einführung in Ansible

Ansible

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



Funktionsweise



Exkurs: YAML

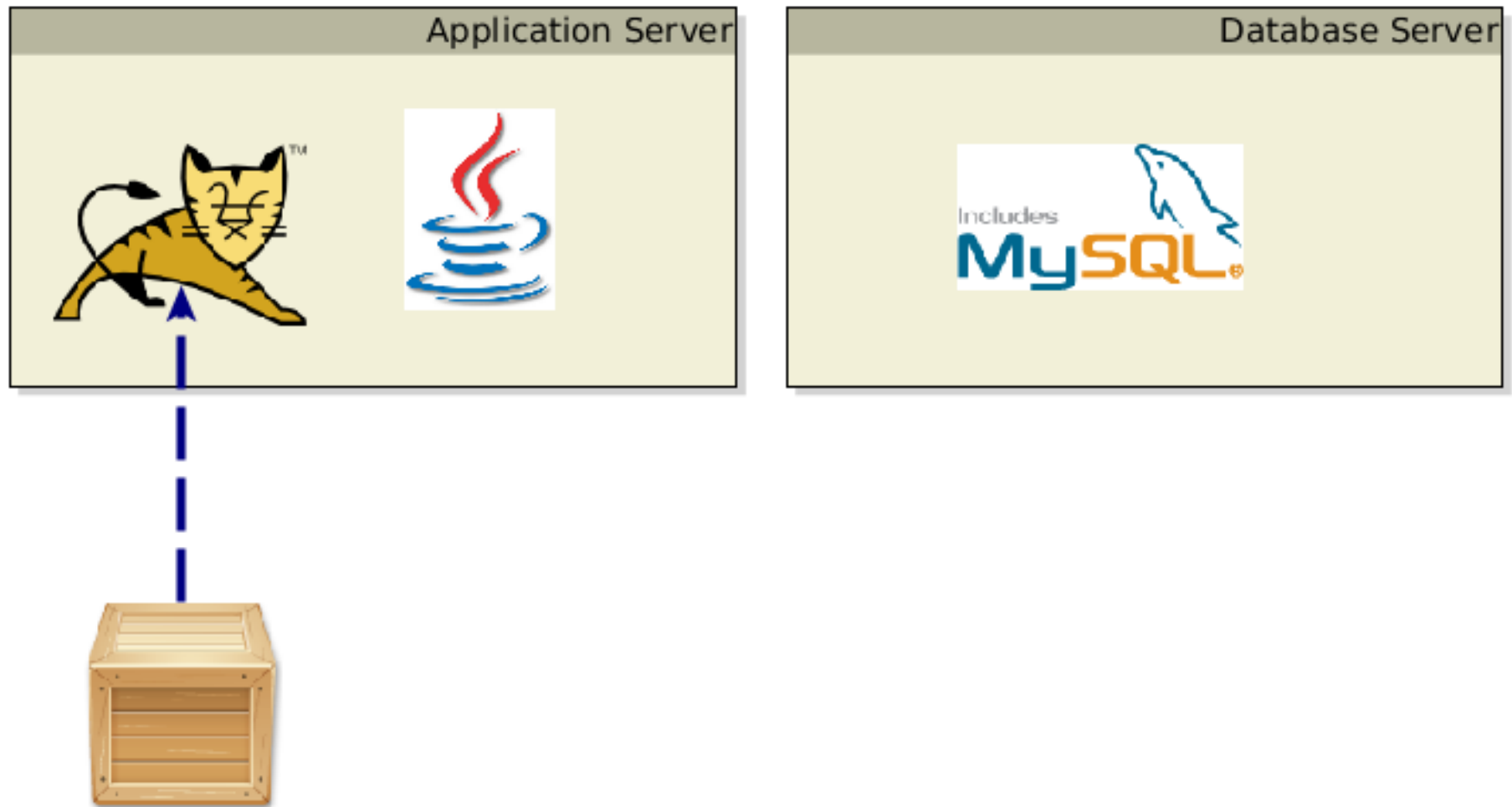
YAML

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

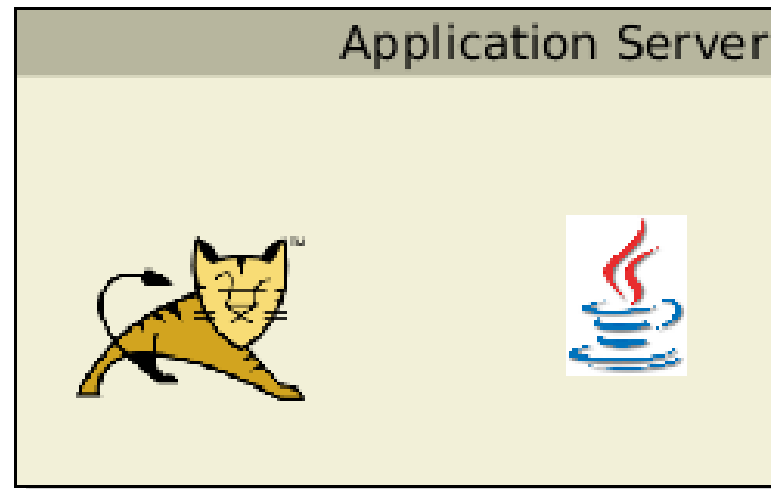
JSON

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

Inventories

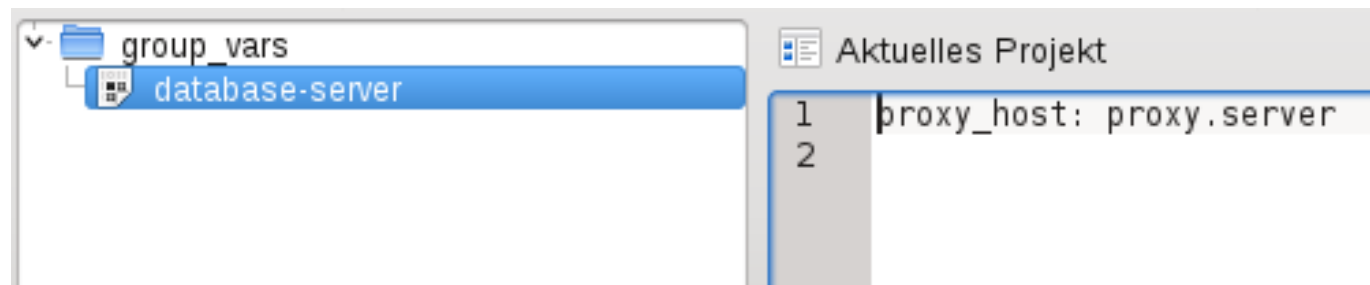
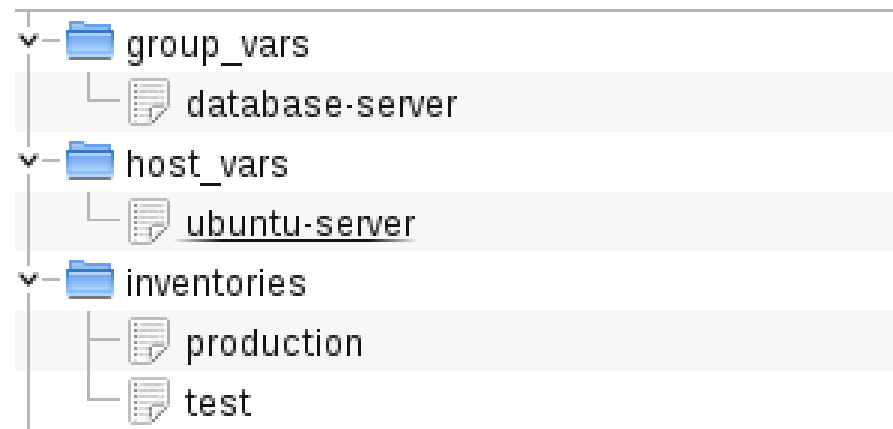
Production

```
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



```
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 }}
23     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/setenv.sh.j2" mode=755
30     when: catalina_opts is defined
31
32   - find: paths="/opt/{{ tomcat_base_name }}/bin" patterns="*.sh"
33     register: result
34
35   - name: ensure tomcat scripts are executable
36     file: name={{item.path}} mode=755
37     with_items: '{{ result.files }}'
38
39   - name: install tomcat as service
40     copy: src=roles/tomcat8/files/tomcat.service dest=/etc/systemd/system/
41     become: yes
42     become_method: sudo
```


Templates

- setenv.sh.j2

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

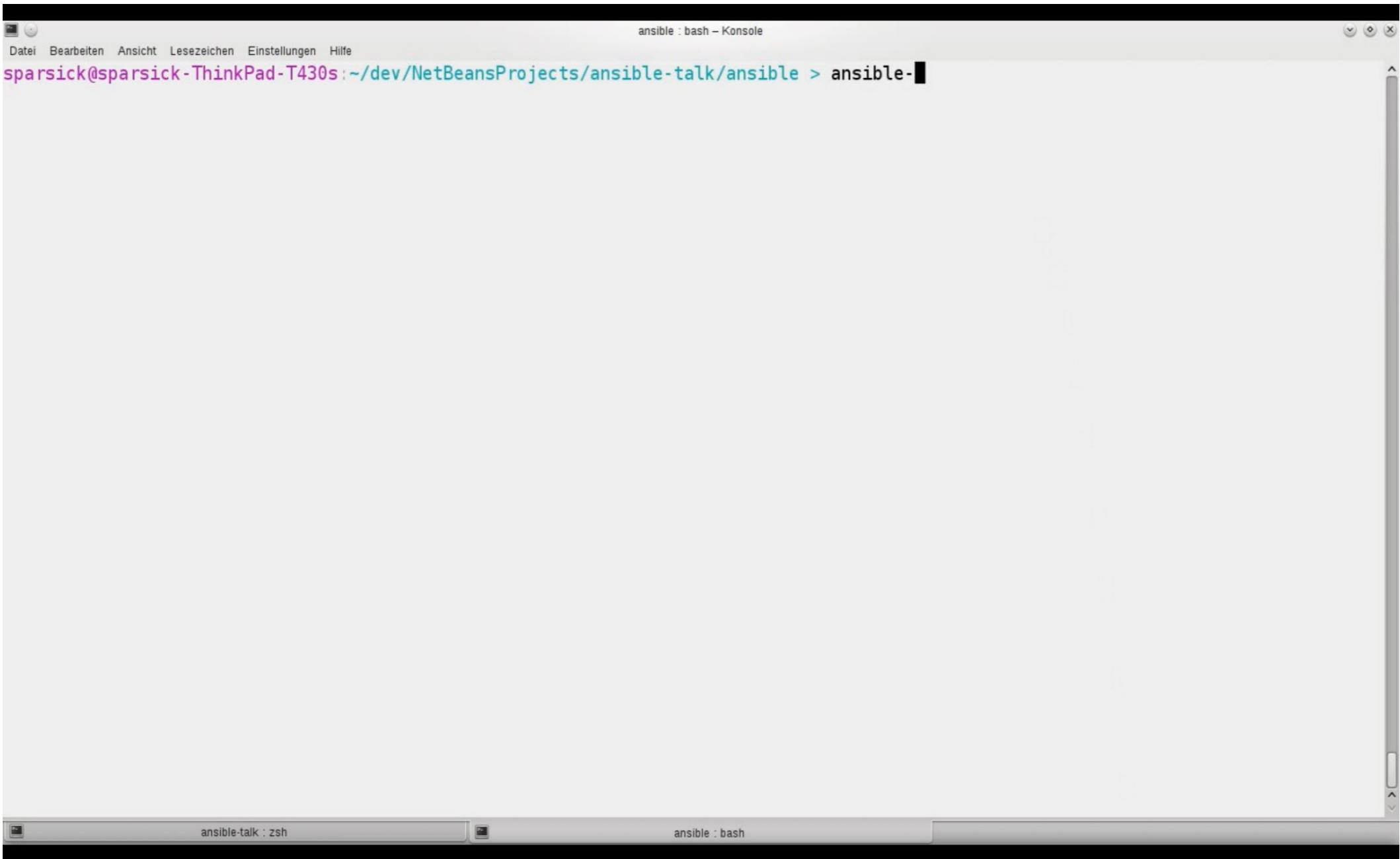
Templates - Jinja2

- Templating engine für Python

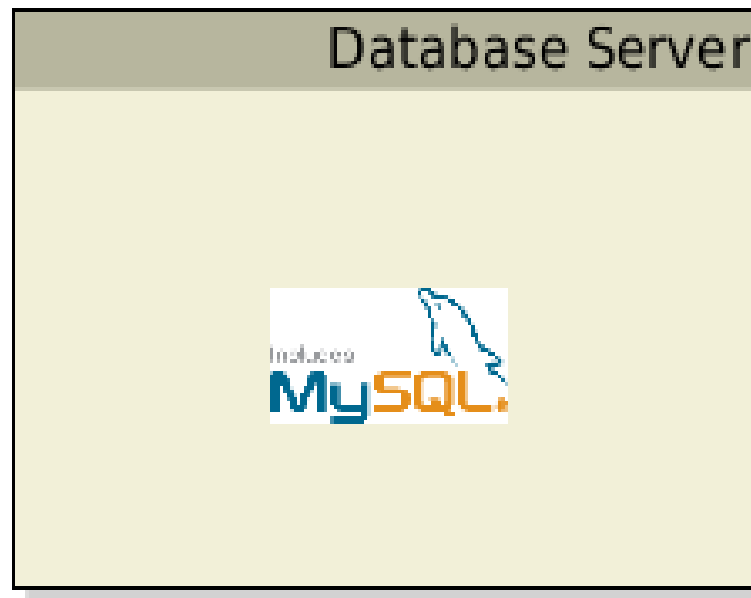
```
<title>{% block title %}{% endblock %}</title>
<ul>
  {% for user in users %}
    <li><a href="{{ user.url }}">{{ user.username }}</a></li>
  {% endfor %}
</ul>
```

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

Setup Application Server Playbook

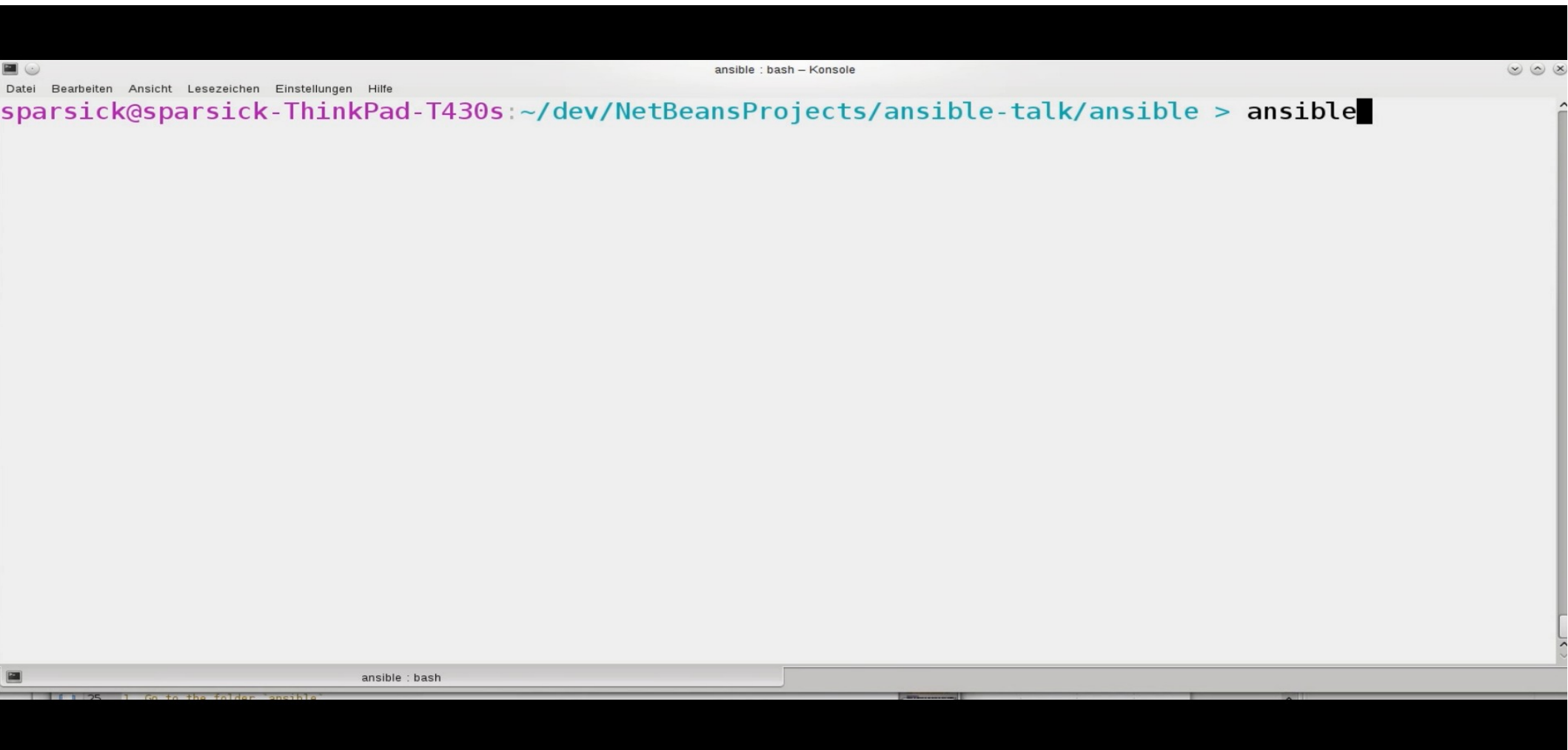


Setup Database Server Playbook

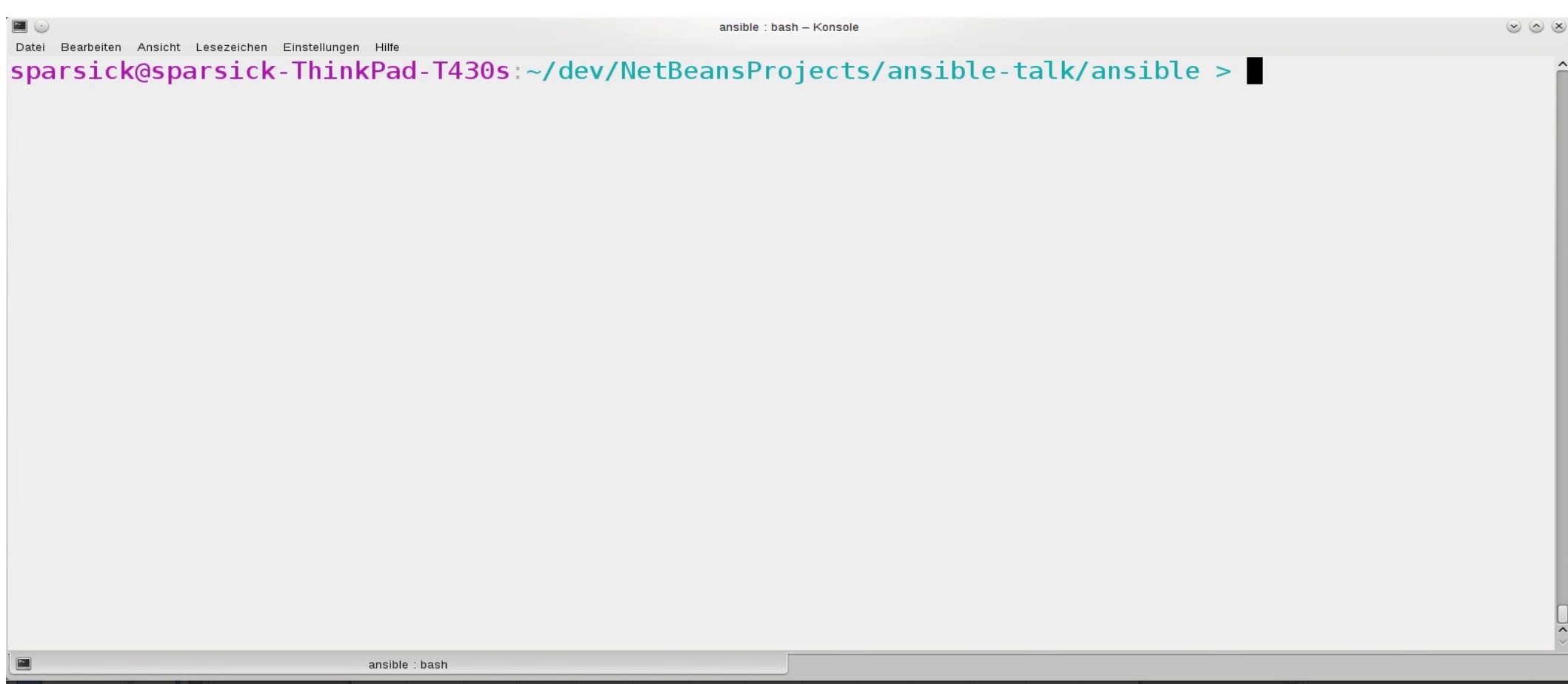


```
1 - hosts: database-server
2   become: yes
3   become_method: sudo
4
5   tasks:
6     - name: install mysql db
7       apt: name=mysql-server state=present
8
9     - name: installs python-mysqldb
10      apt: name=python-mysqldb state=present
11
12     - name: start mysql
13       service: name=mysql 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=g3h3lm priv=*.*:ALL,GRANT state=present host=%
24
25   handlers:
26     - name: restart mysql
27       service: name=mysql state=restarted
```

Setup Database Server Playbook



Setup Database Server Playbook



The image shows a terminal window with a light gray background. The title bar at the top reads "ansible : bash - Konsole". Below the title bar is a menu bar with the following items: "Datei", "Bearbeiten", "Ansicht", "Lesezeichen", "Einstellungen", and "Hilfe". The main area of the terminal is white and contains a single line of text: "sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible >". The text is in a monospaced font, with the username "sparsick" in purple, the host "sparsick-ThinkPad-T430s" in purple, the path "~/dev/NetBeansProjects/ansible-talk/ansible" in cyan, and the prompt ">" in black. A black cursor is positioned at the end of the line. At the bottom of the terminal window, there is a status bar with the text "ansible : bash".

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

```

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

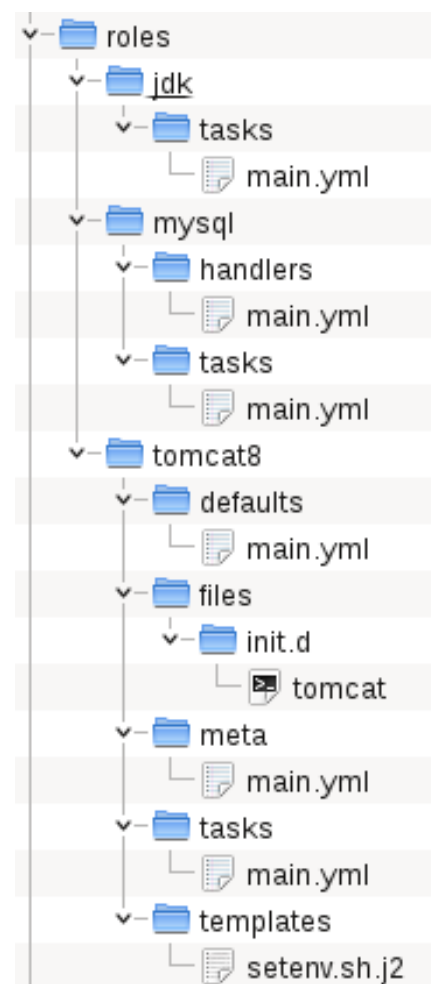
```



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

Roles

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



Setup Playbooks mit Roles

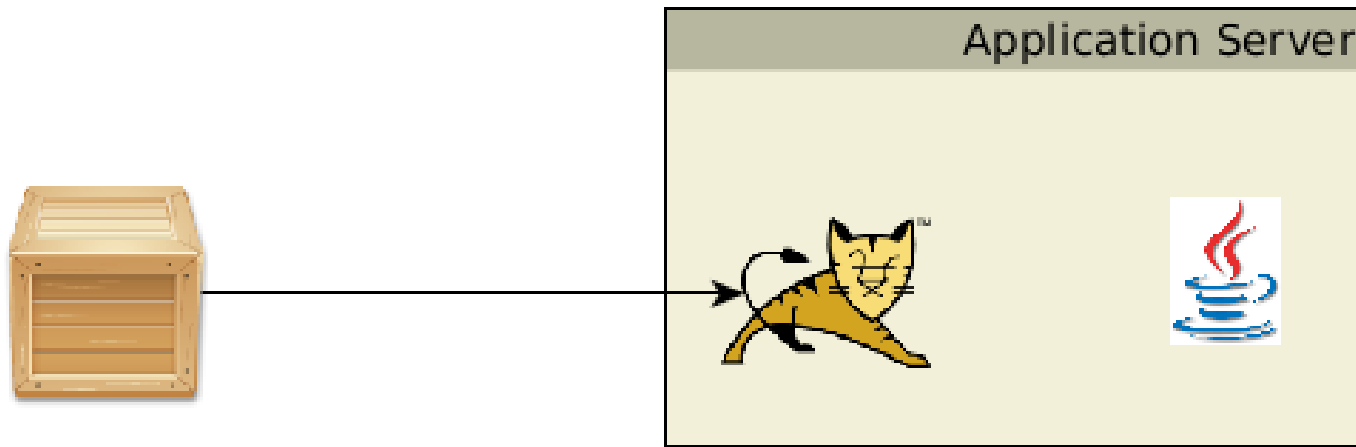
- Setup Application Server

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

- Setup Database Server

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

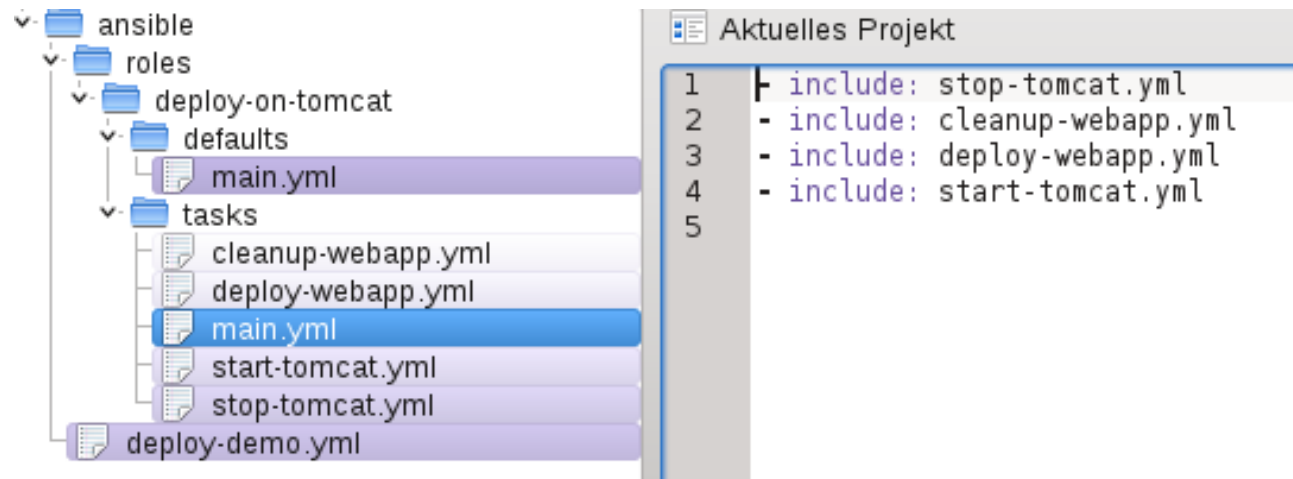
Java Webapplikation Deployment



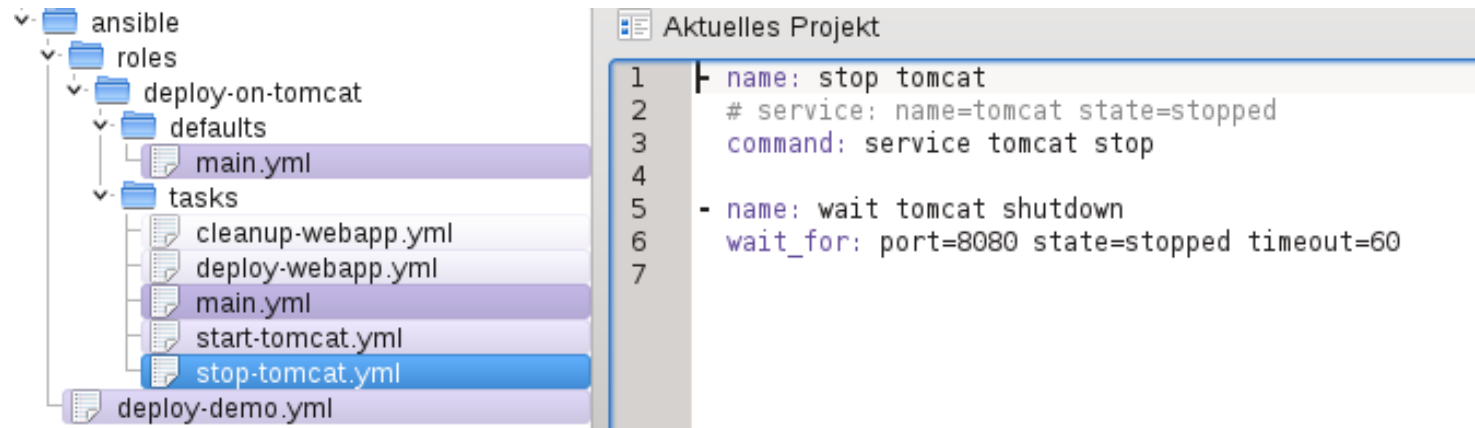
Deploy Application Playbook

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

deploy-on-tomcat Role



deploy-on-tomcat Role



The image shows a screenshot of an IDE with two panels. The left panel displays a file tree for an Ansible project. The right panel shows the content of the selected file, `stop-tomcat.yml`.

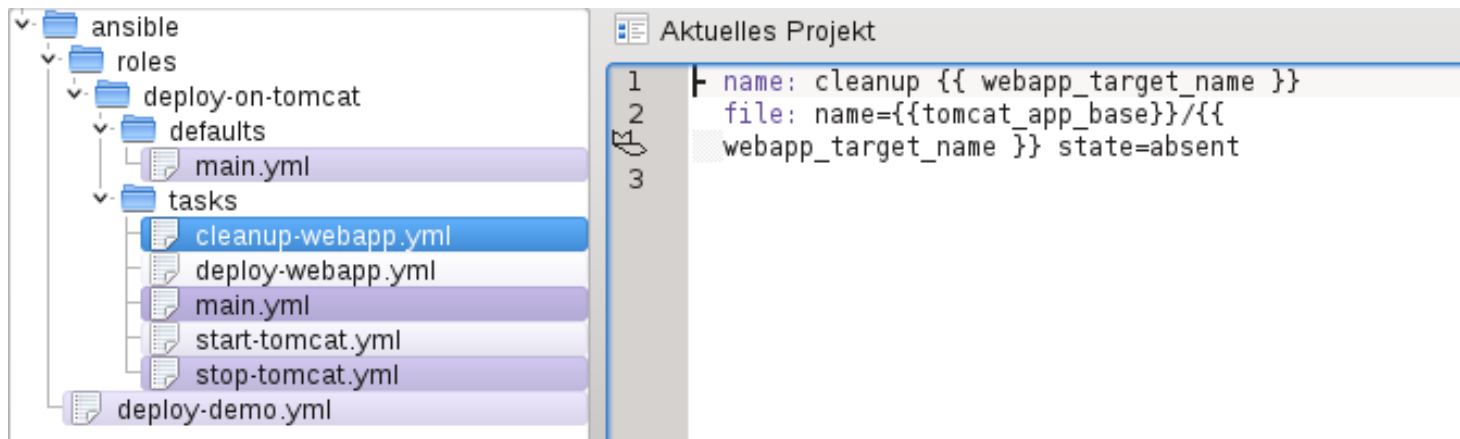
File Tree Structure:

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

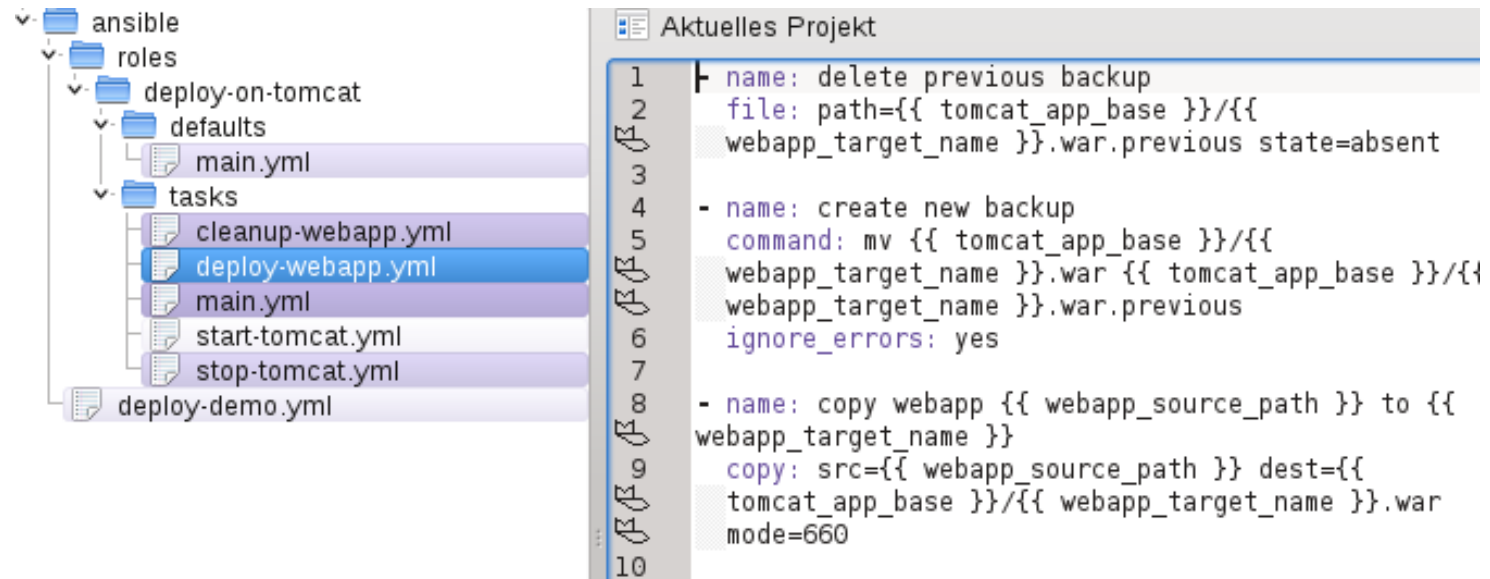
Task Content (stop-tomcat.yml):

```
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



deploy-on-tomcat Role



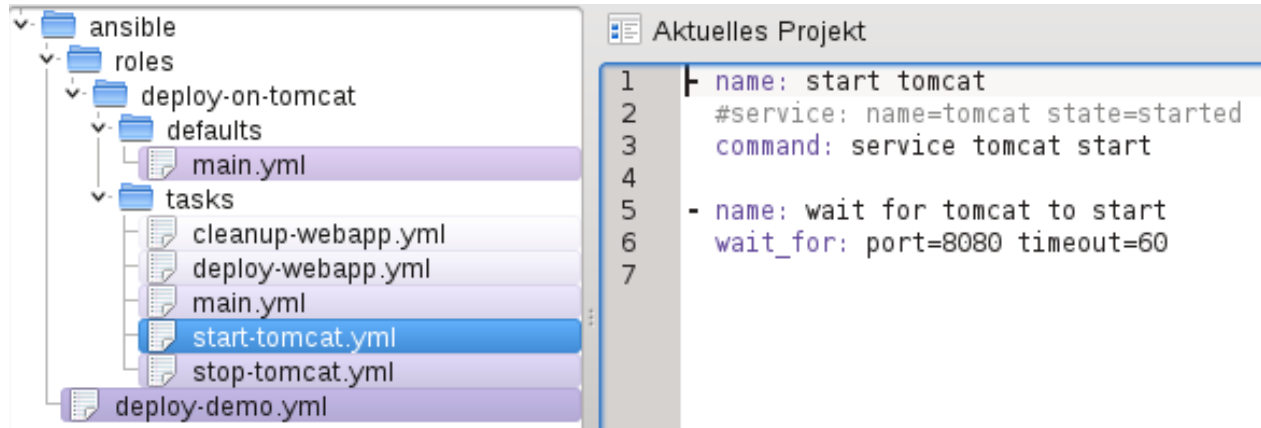
The image shows a file explorer on the left and a code editor on the right. The file explorer displays the following structure:

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

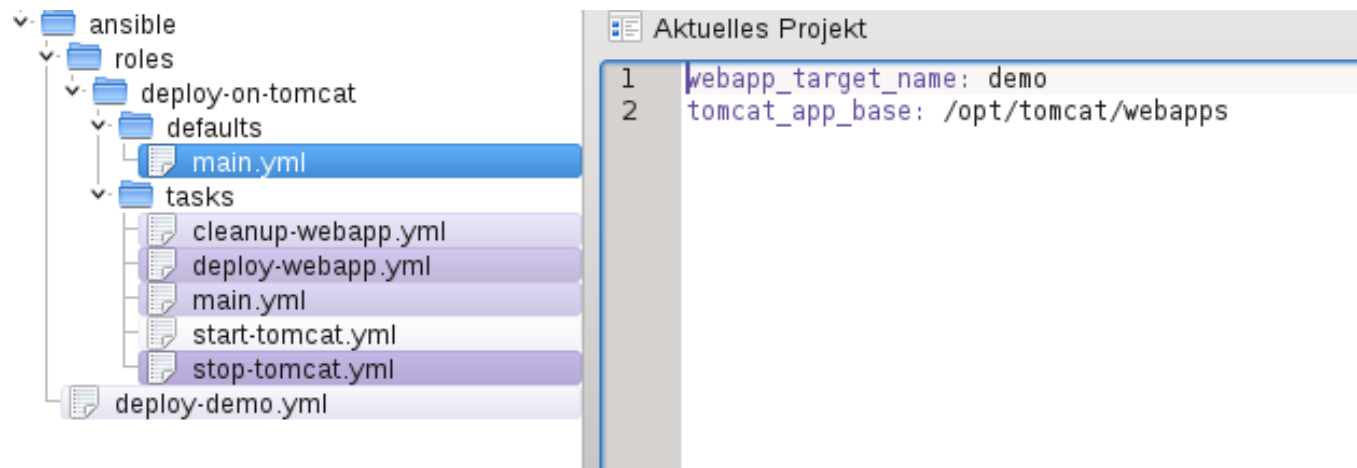
The code editor on the right, titled 'Aktuelles Projekt', shows the content of the `deploy-webapp.yml` file:

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

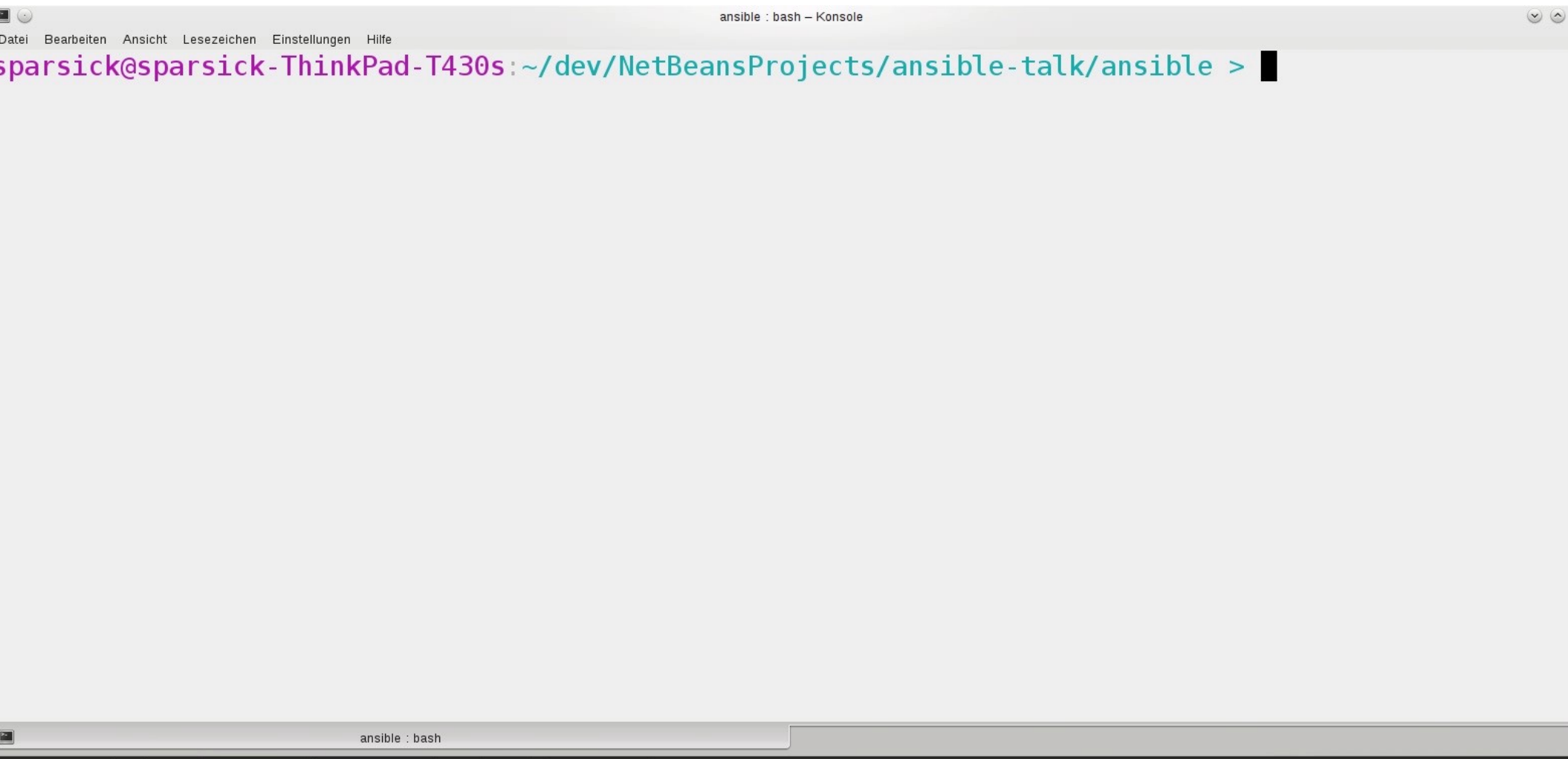
deploy-on-tomcat Role



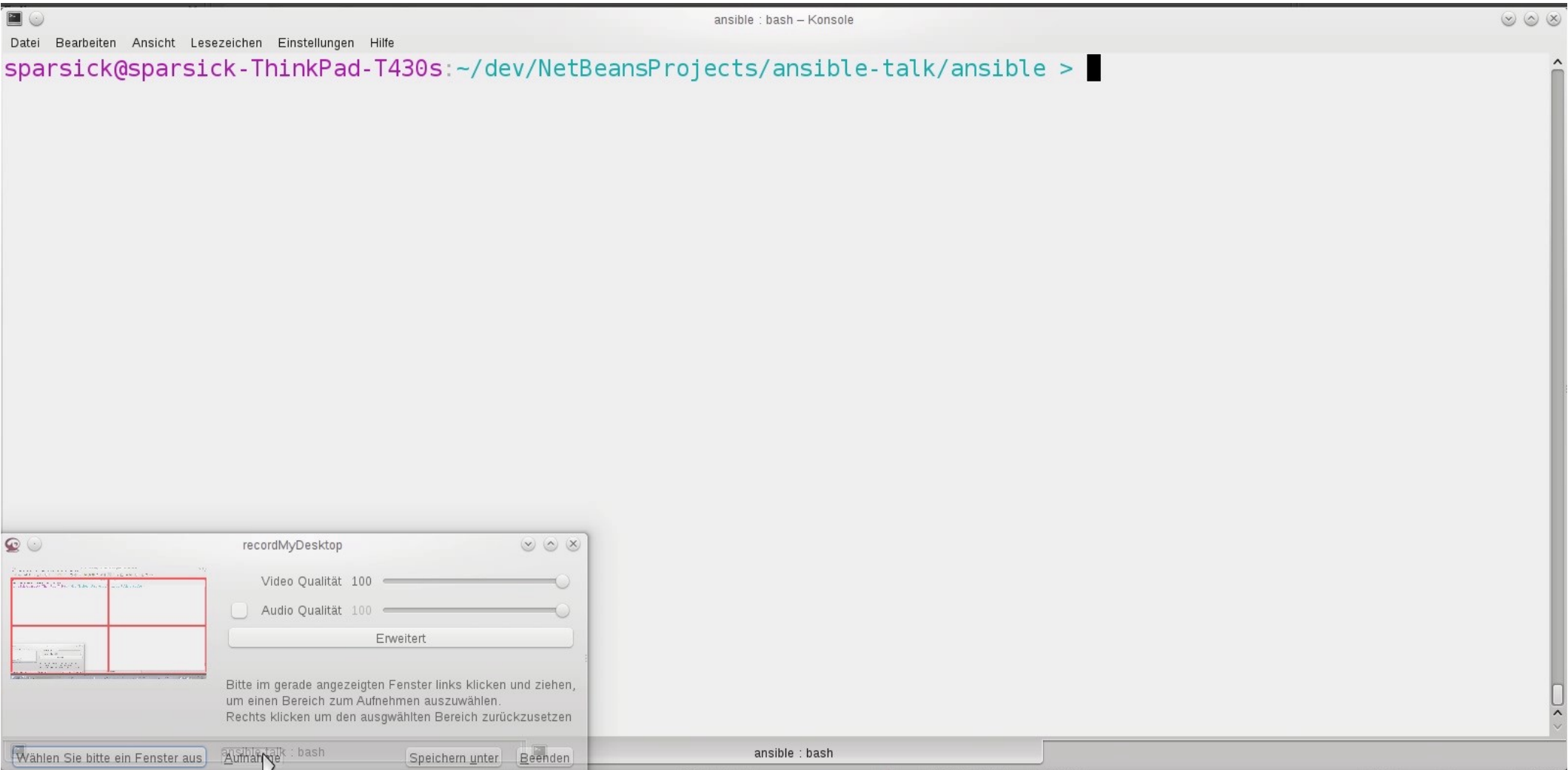
deploy-on-tomcat Role



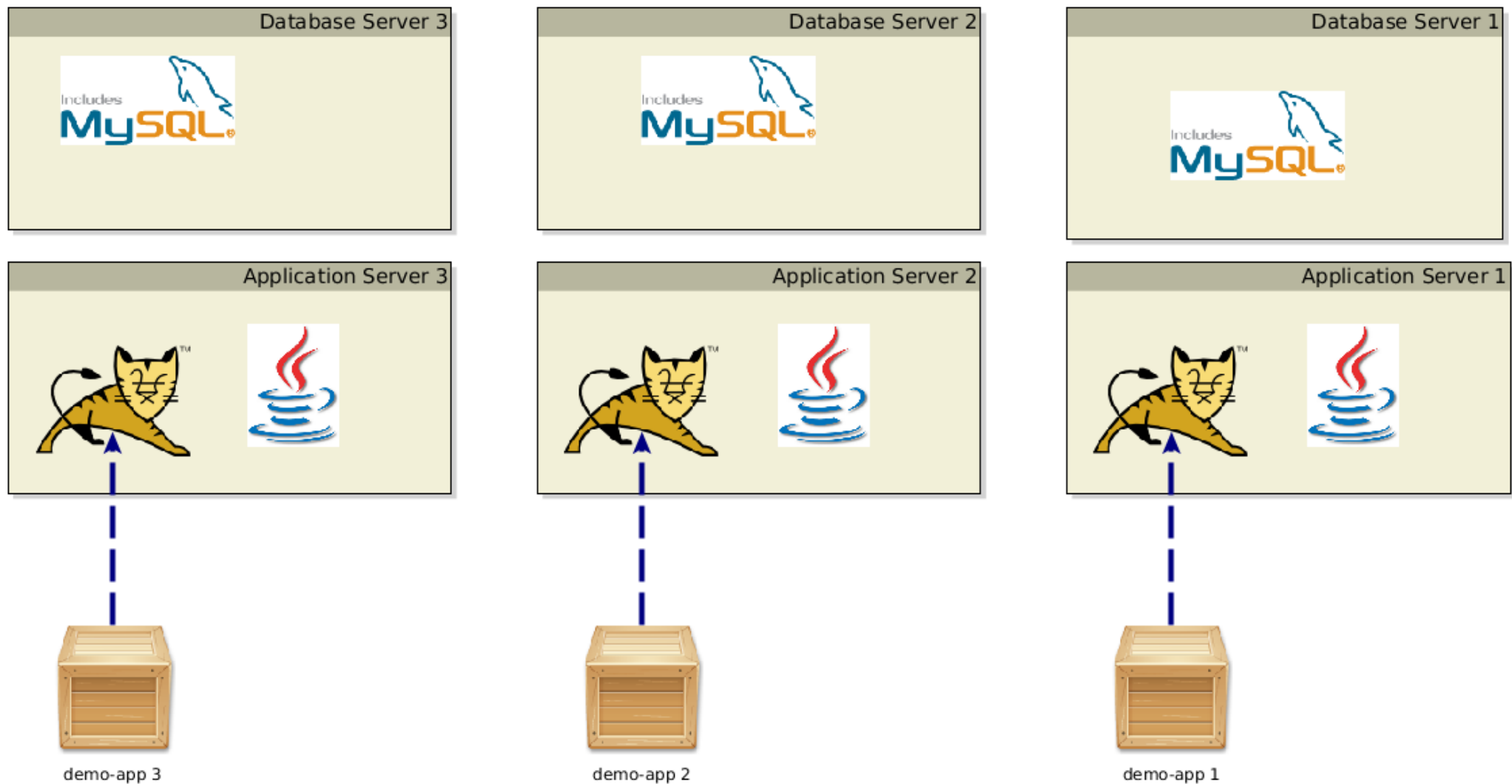
Deploy Application Playbook









Ad-hoc-Kommando



Warum Roles?



Warum Roles?

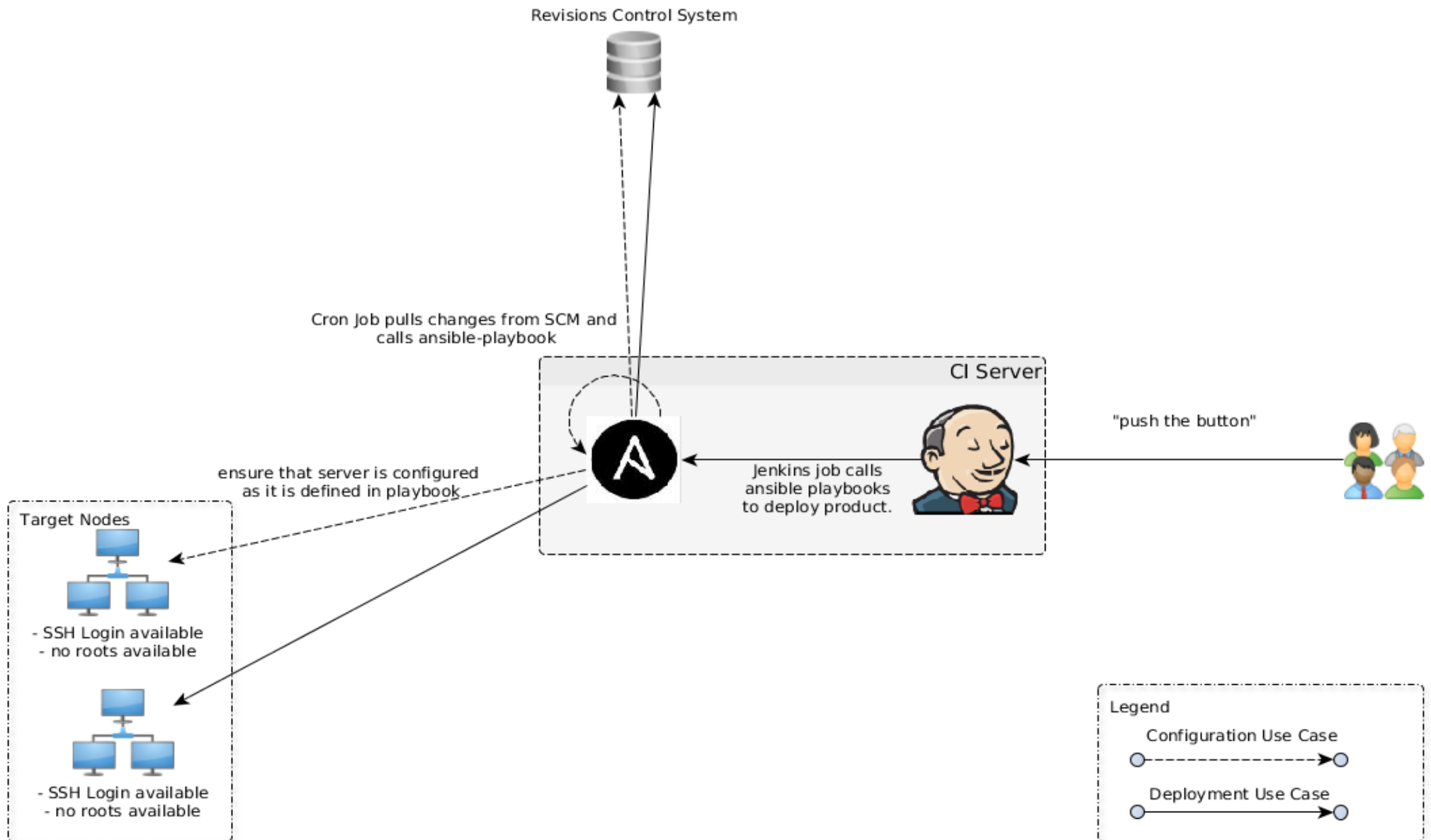
	deploy-demo1-app.yml
	deploy-demo2-app.yml
	setup-demo1-app-server.yml
	setup-demo1-database.yml
	setup-demo2-app-server.yml
	setup-demo2-database.yml

Warum Roles?

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

```
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 }
5
```


Ansible Infrastruktur



Ansible Tower



TOWER

Organizations

Users

Teams

Credentials

Projects

Inventories

Job Templates

Jobs

Hello, admin



0

Hosts

0

Failed Hosts

1

Inventories

1

Inventory Sync Failures

0

Projects

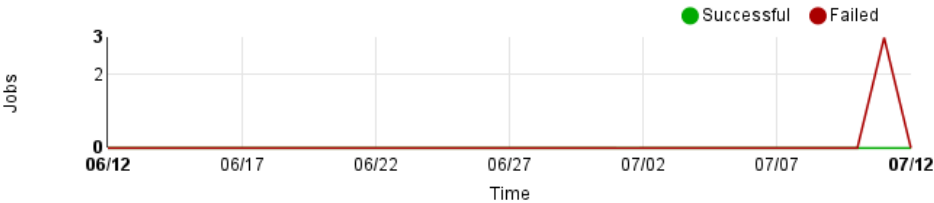
0

Project Sync Failures

Job Status

Job Type: All

Period: Past Month



Host Status

No Host data

Jobs

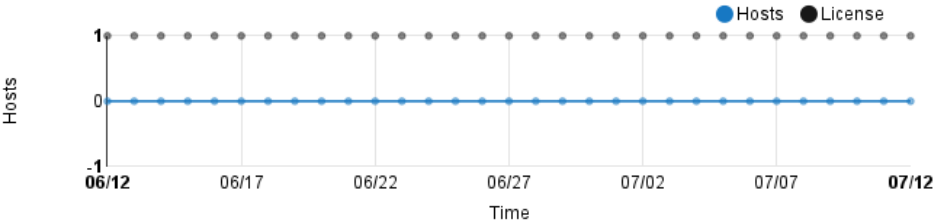
Schedule

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

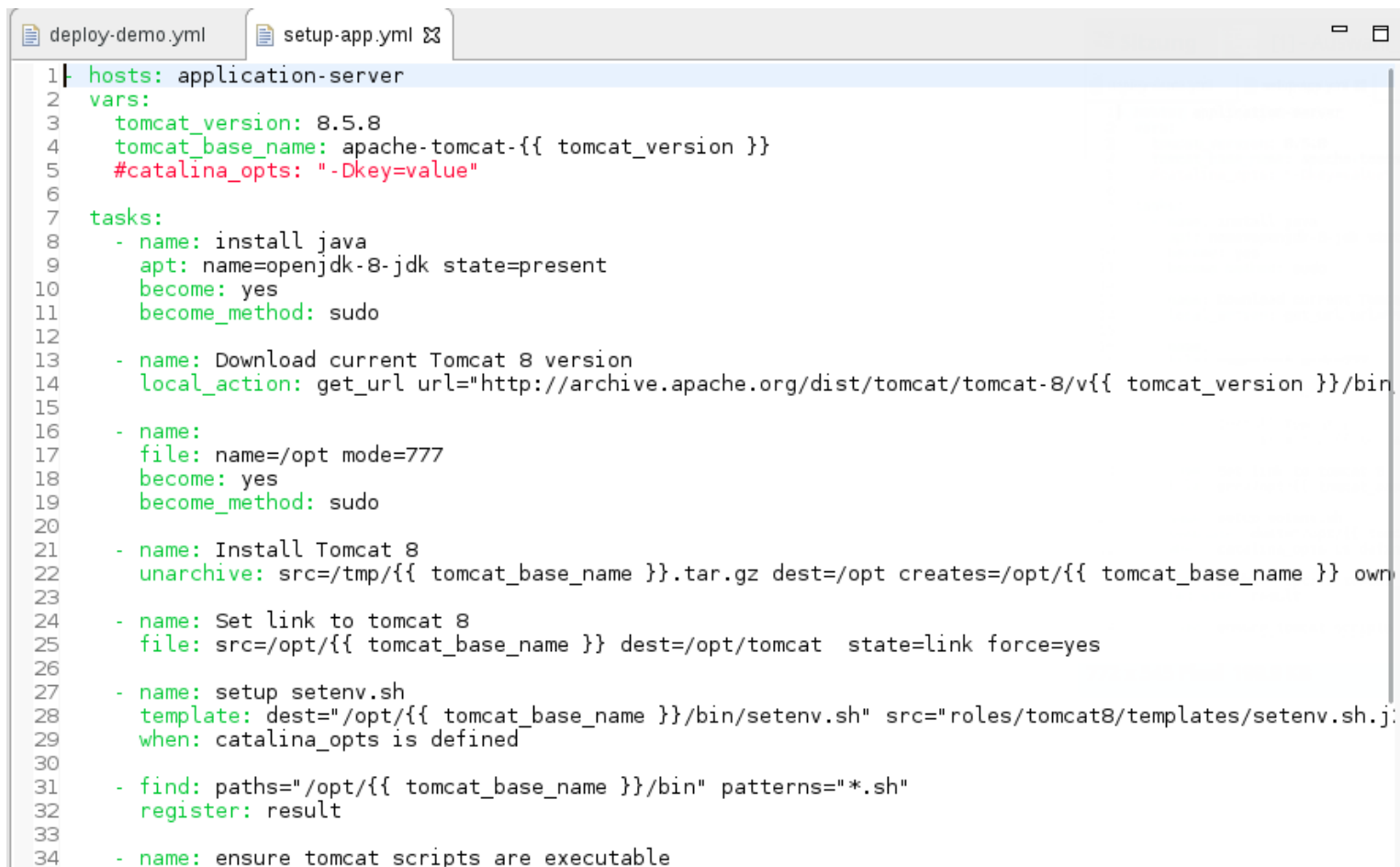


IDE - Support

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

IDE-Support - Eclipse

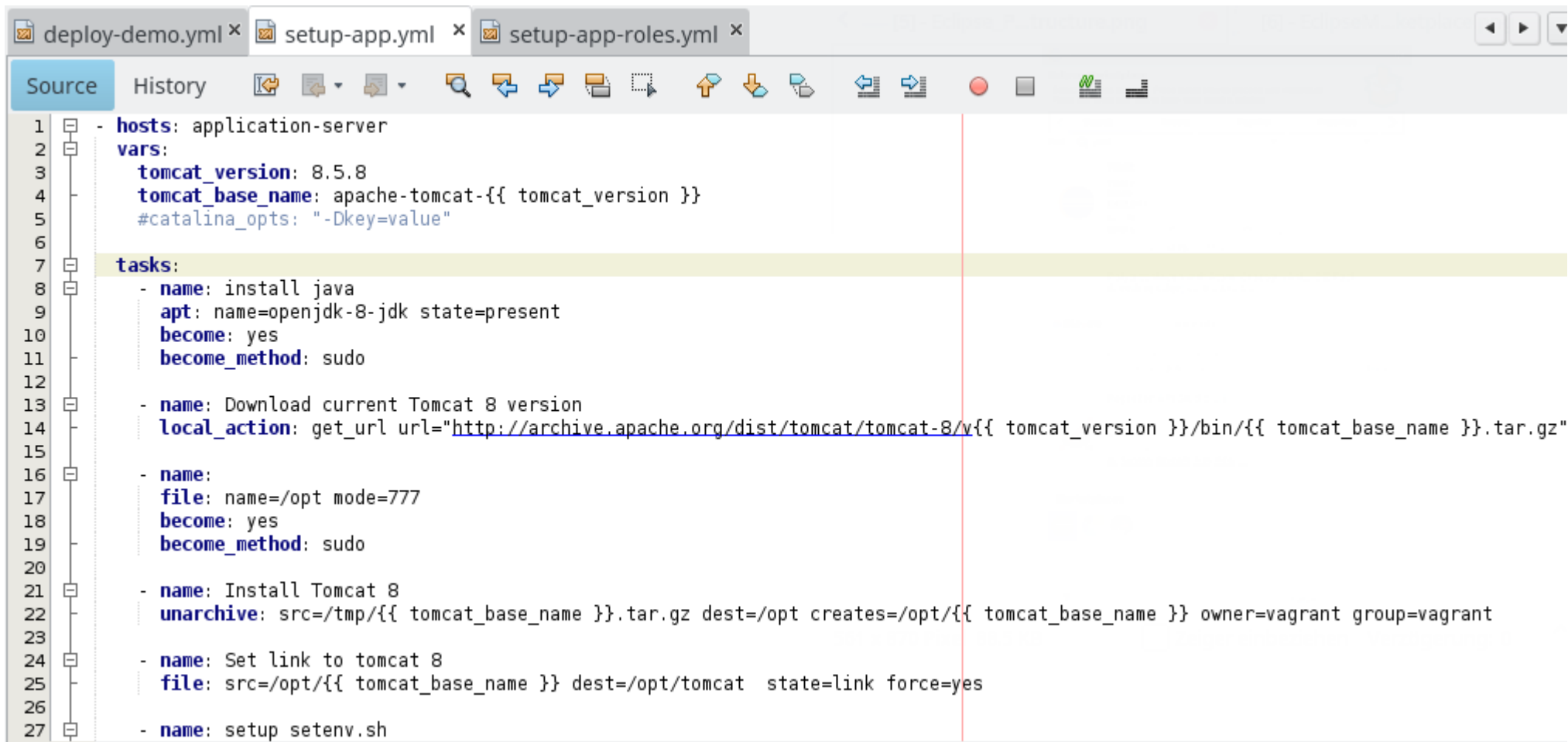
- YAML Support über Plugin *yEdit*



```
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 creates=/opt/{{ tomcat_base_name }} own
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.j
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: `deploy-demo.yml`, `setup-app.yml`, and `setup-app-roles.yml`. The `Source` tab is active, displaying the content of `setup-app.yml`. The editor shows a YAML playbook with a tree view on the left and a toolbar at the top. The playbook content is as follows:

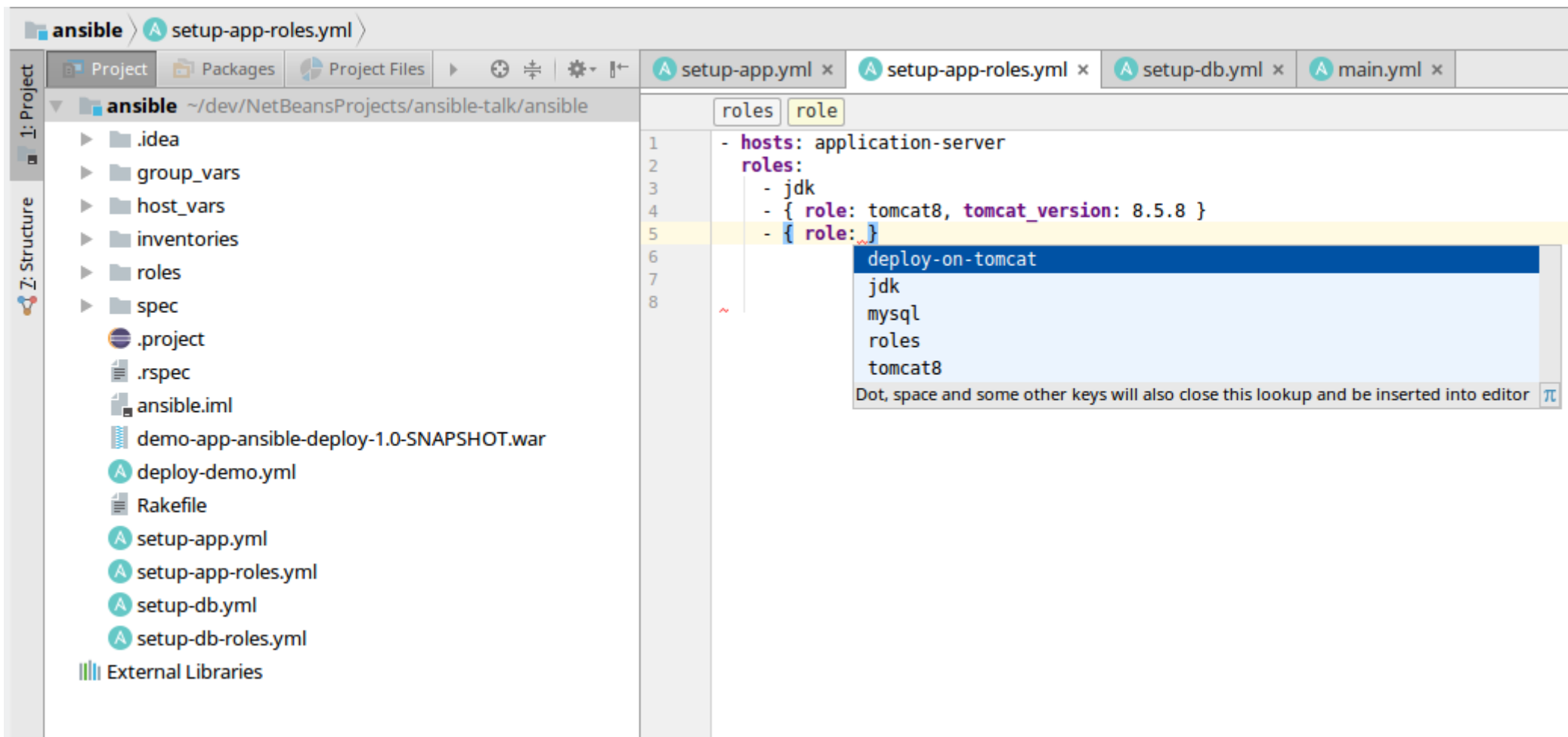
```
1 - hosts: application-server
2 vars:
3   tomcat_version: 8.5.8
4   tomcat_base_name: apache-tomcat-{{ tomcat_version }}
5   #catalina_opts: "-Dkey=value"
6
7 tasks:
8   - name: install java
9     apt: name=openjdk-8-jdk state=present
10    become: yes
11    become_method: sudo
12
13   - name: Download 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*)

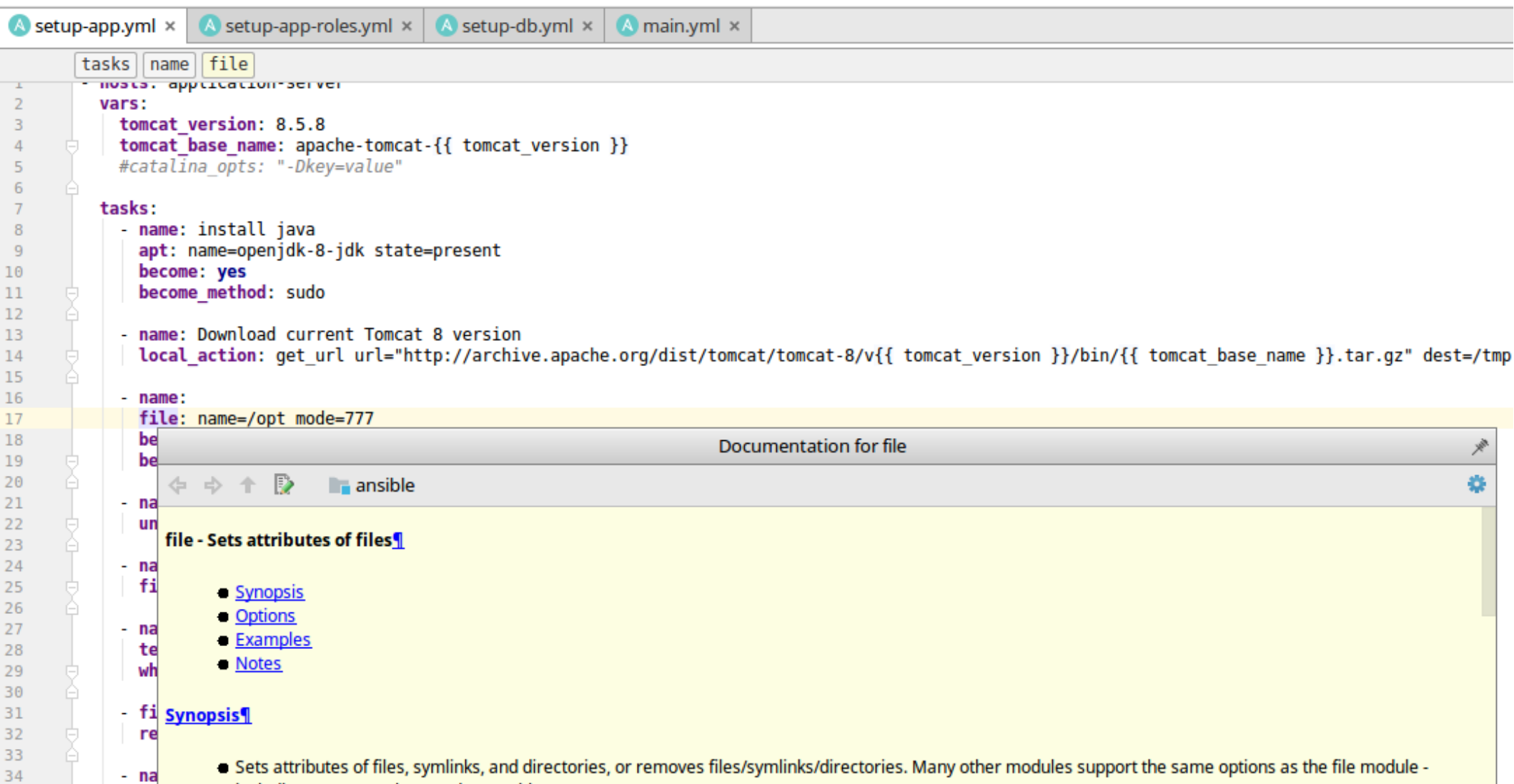
IDE-Support - IntelliJ IDEA

- Autovervollständigung für Roles



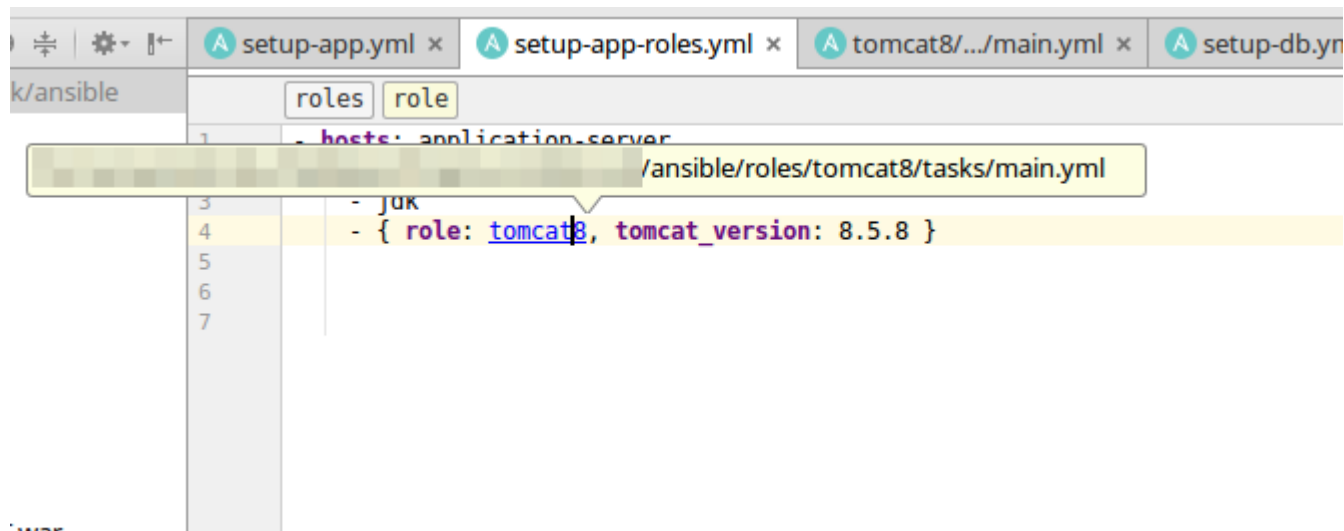
IDE-Support - IntelliJ IDEA

- Anzeige der Dokumentation für Ansible Module



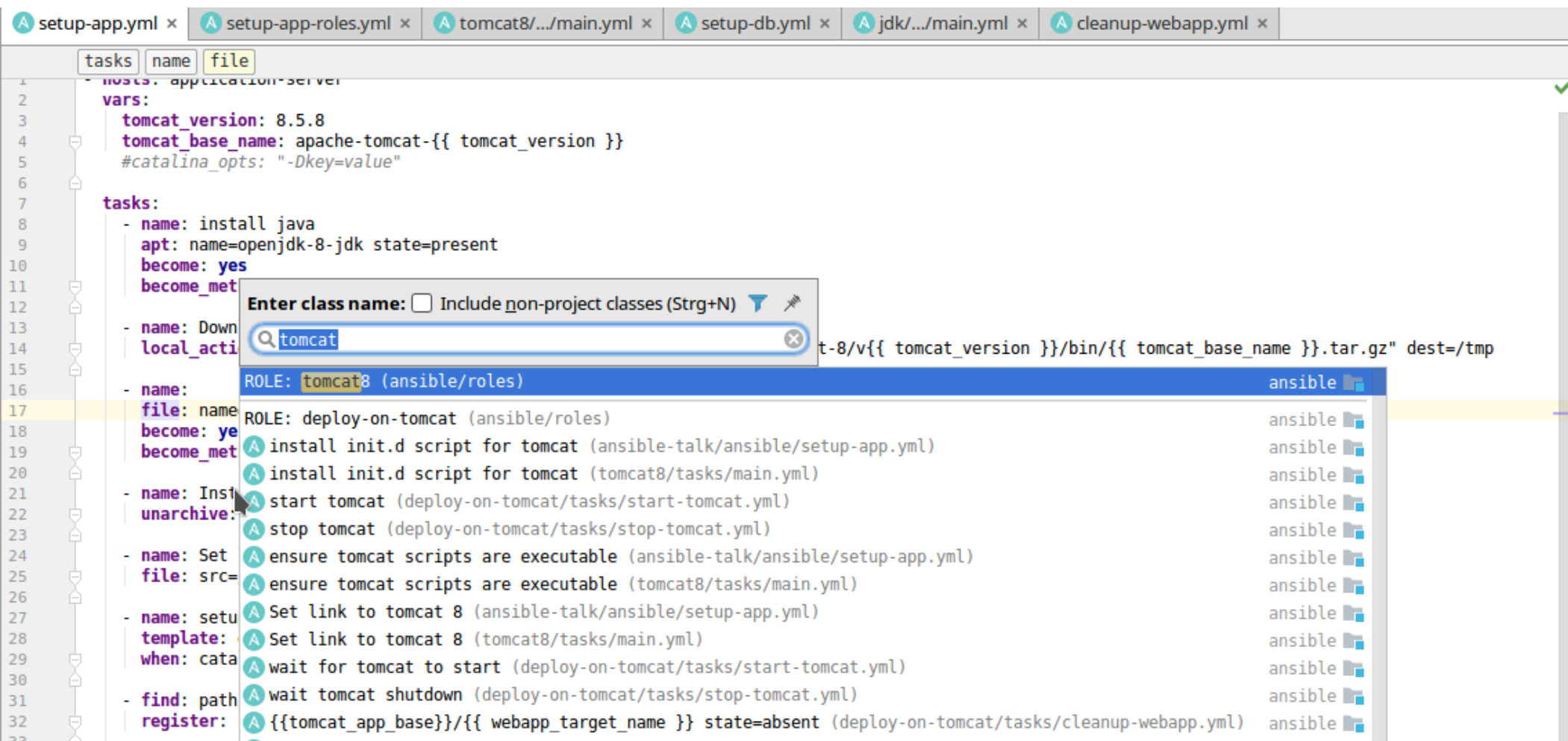
IDE-Support - IntelliJ IDEA

- Direkte Navigation zu der Rollendefinition



IDE-Support - IntelliJ IDEA

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



Weitere Features

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

Wie werden Ansible Skripte getestet?

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



ServerSpec Tests

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

```
1 require 'spec_helper'
2
3 describe package('openjdk-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
14
```

ServerSpec Tests



Wie unterscheidet sich Ansible zu seiner Konkurrenz?



Vergleich



- Orchestrierung über SSH
- Benötigt keine Rootrechte auf Zielsystem
- Konfigurationsmgmt + Applikationsdeployment
- Monitoringtool nur in der Enterprise Variante
- Skripte mehr imperativ
- Windows-Support rudimentär
- Skripte OS- bzw. Distributions-spezifisch

- Client-Server Architektur
- Für komfortables Arbeiten benötigt es Rootrechte
- Konfigurationsmgmt
- Monitoringtools Open Source
- Skripte mehr deklarativ
- Windows-Support
- Skripte können OS-unspezifisch sein



Vergleich



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

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

```
▼ class nodejs {
  ▼ class { 'apt':
    }
  ▼ exec { 'apt-get-update':
    command => '/usr/bin/apt-get update',
    }
  ▼ package {'software-properties-common' :
    ensure=> installed,
    require => Exec['apt-get-update'],
    }

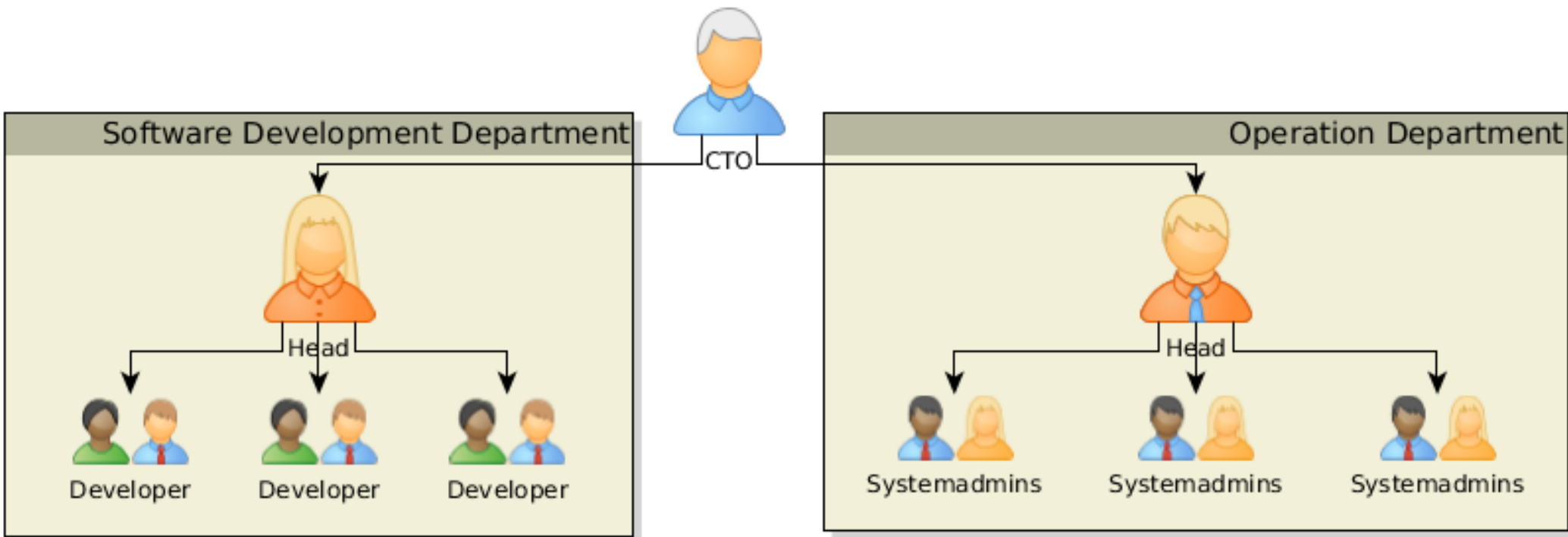
  apt::ppa {'ppa:chris-lea/node.js' :}

  ▼ package { 'nodejs' :
    ensure => installed,
    require => Apt::Ppa ['ppa:chris-lea/node.js'],
    }
}
```

Weitere Einsatzszenarien aus Entwicklersicht

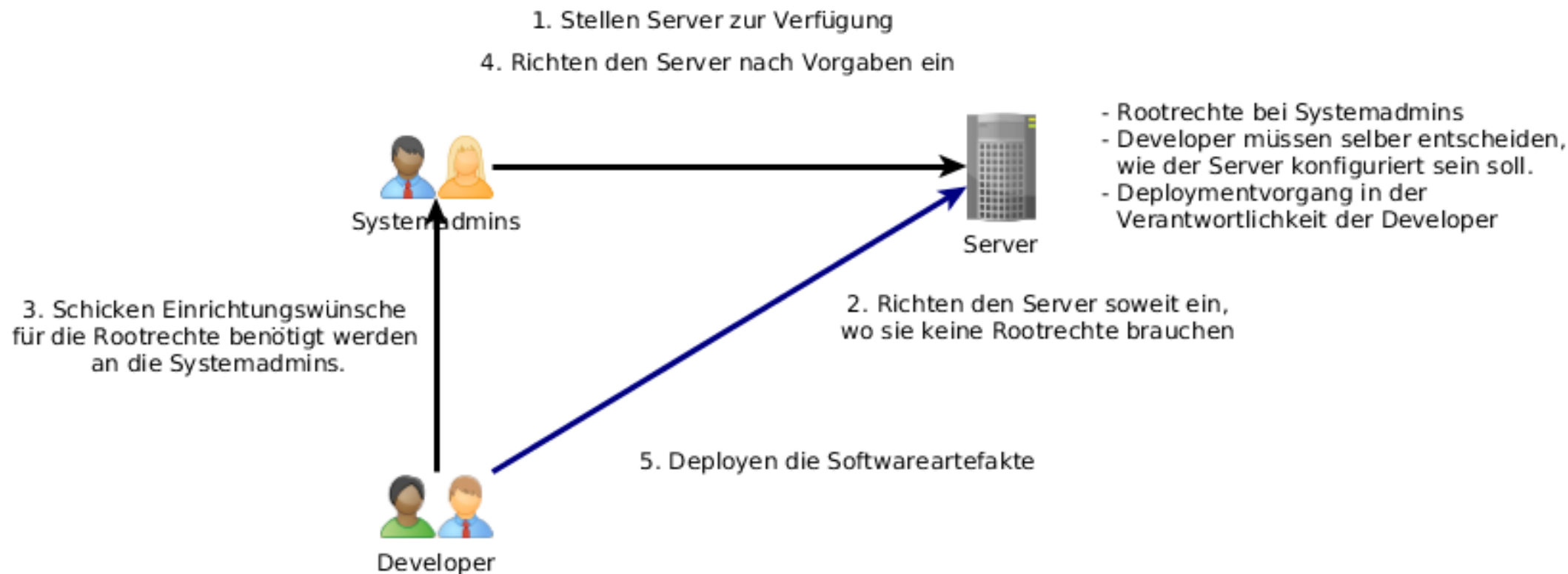
Systemkonfiguration für Entwickler

Organisatorische Ausgangslage
Realität



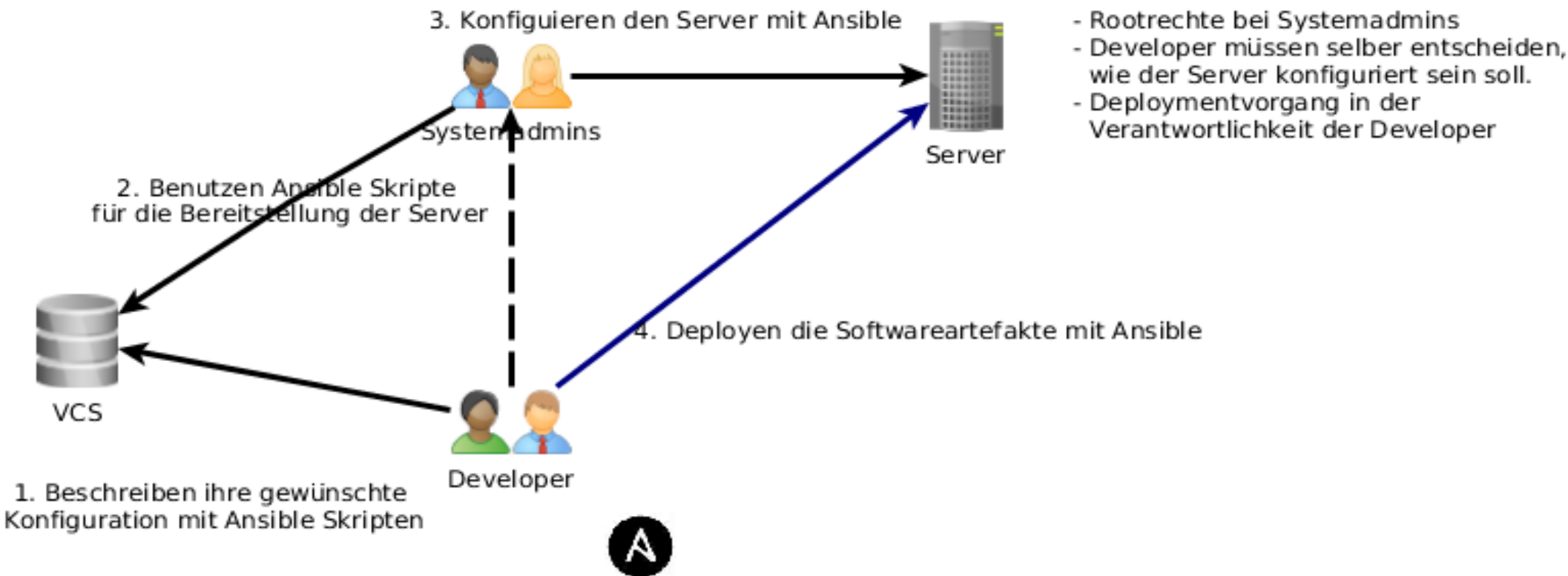
Systemkonfiguration für Entwickler

Prozess zwischen Development und Operation



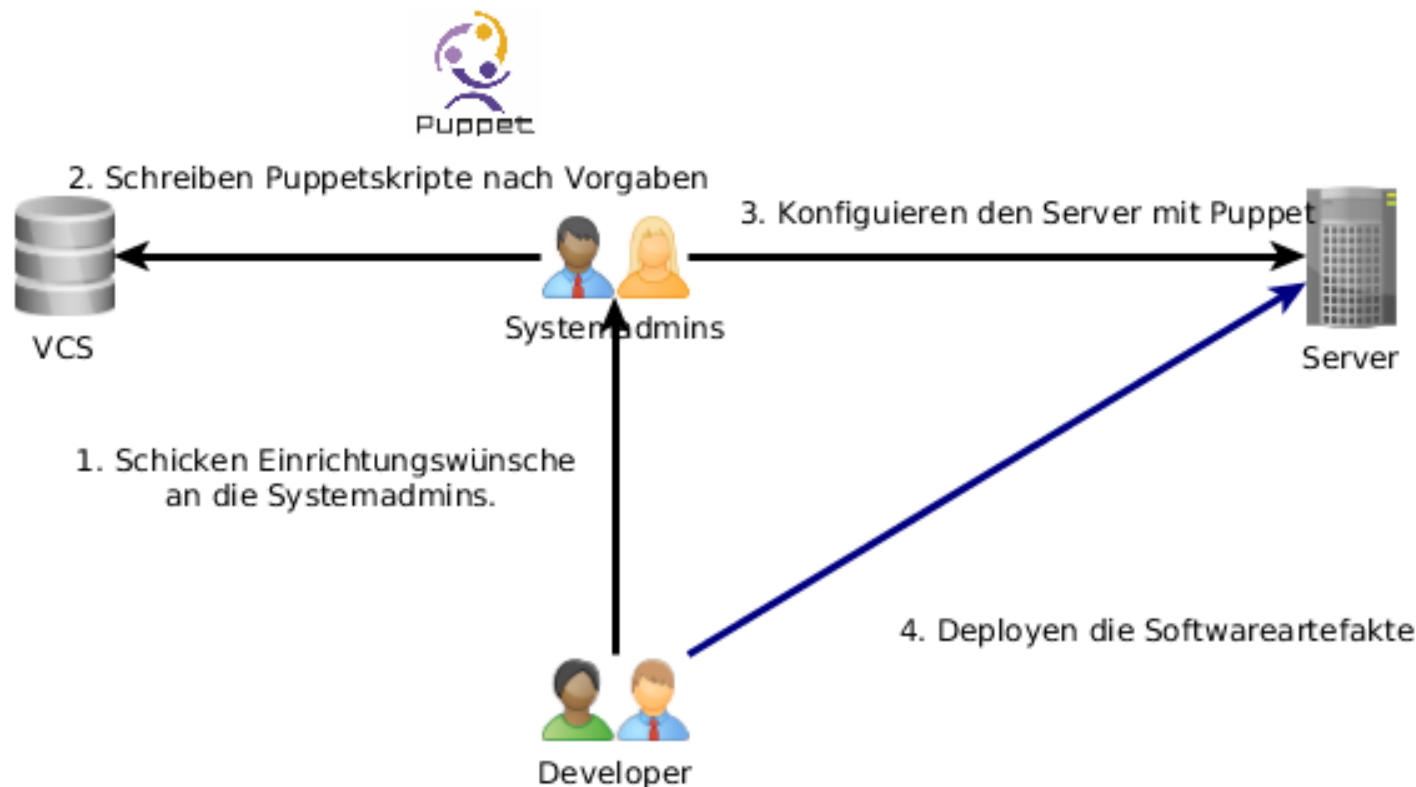
Systemkonfiguration für Entwickler

Lösungsidee mit Ansible



Systemkonfiguration für Entwickler

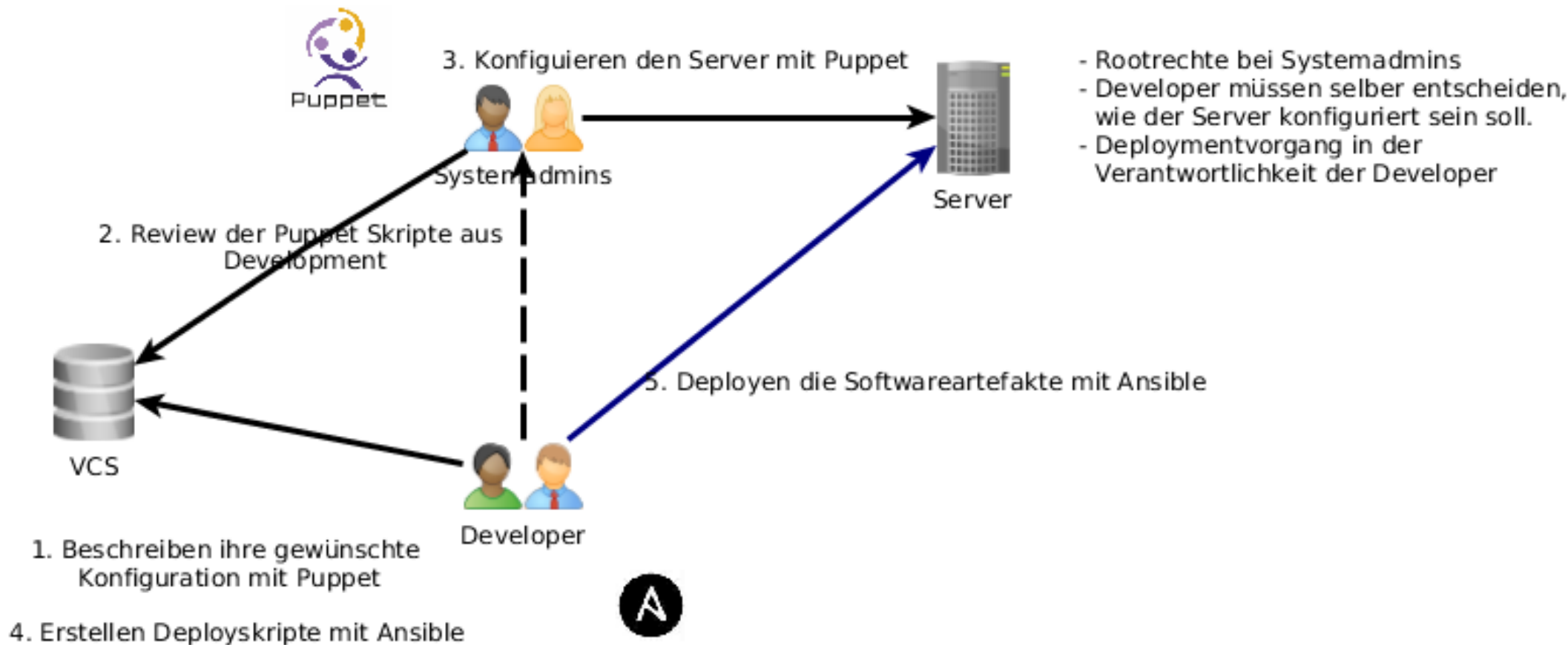
Variante - Prozess zwischen Development und Operation



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

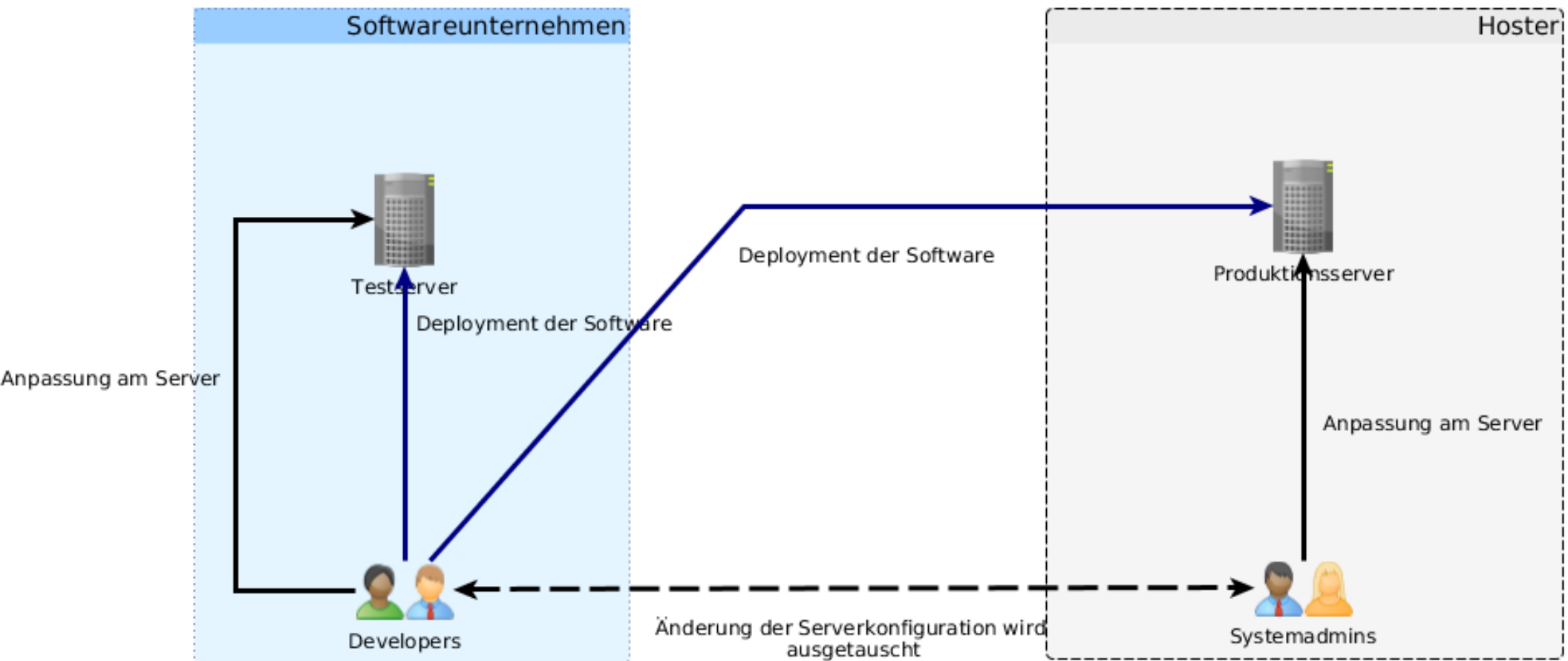
Systemkonfiguration für Entwickler

Lösungsvariante



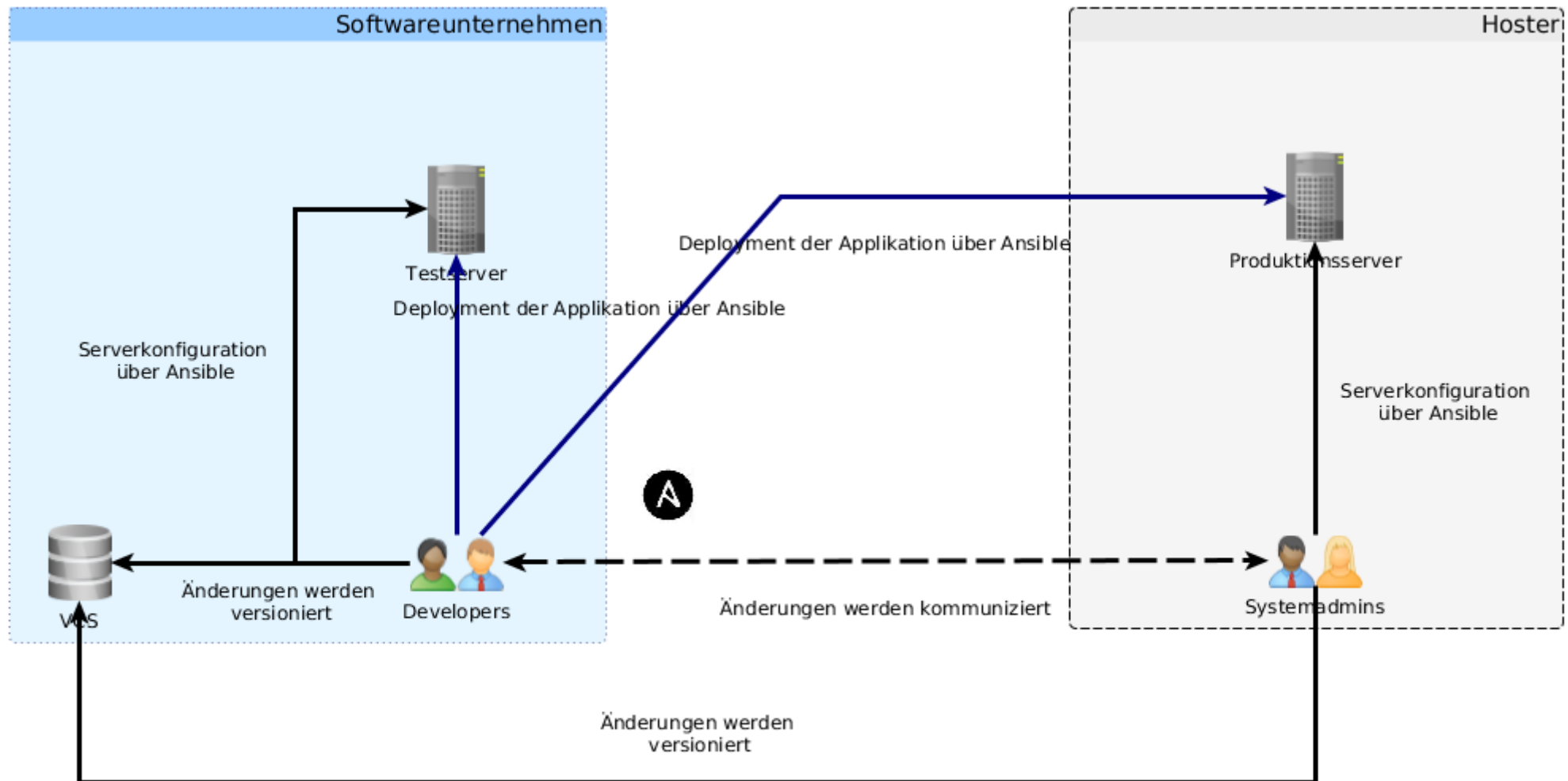
Systemkonfiguration für Entwickler

Produktionsserver sind beim externen Hoster



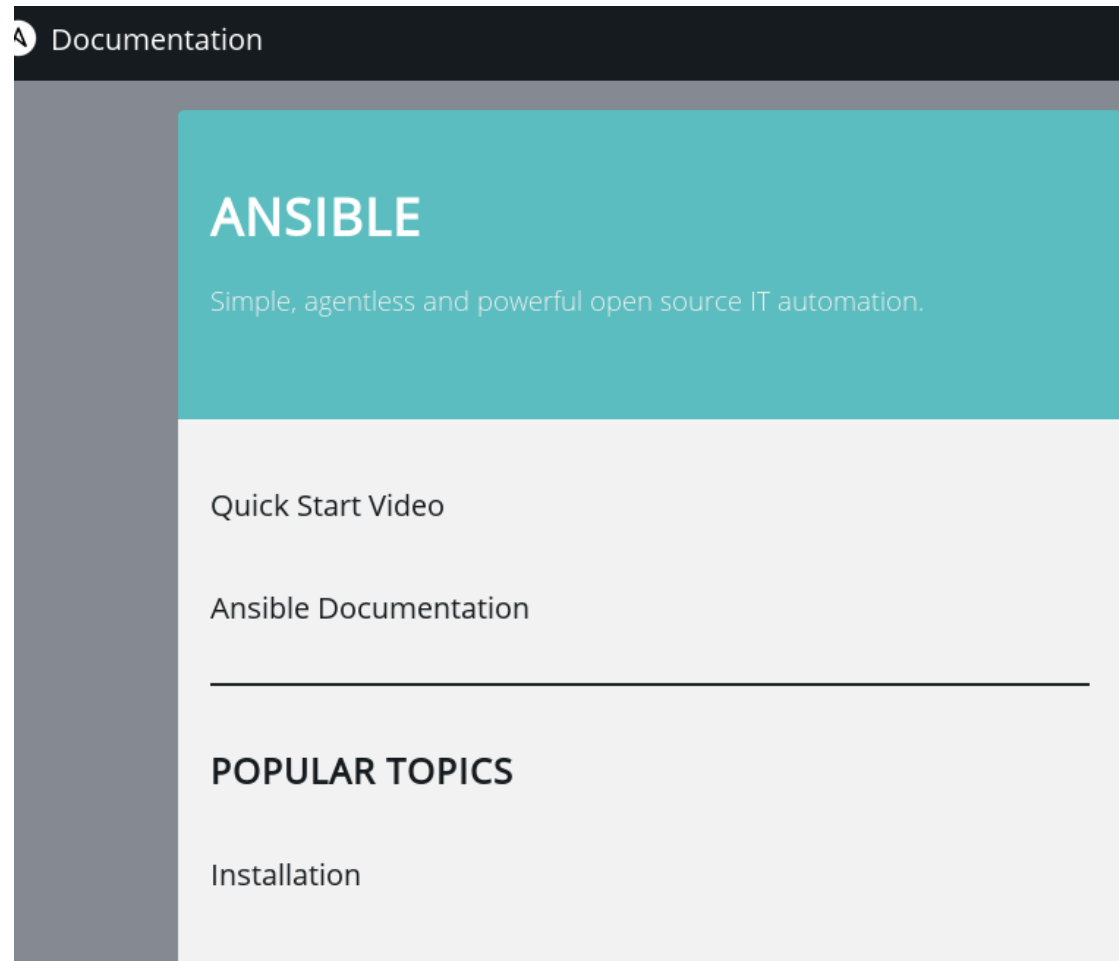
Systemkonfiguration für Entwickler

Lösungsidee



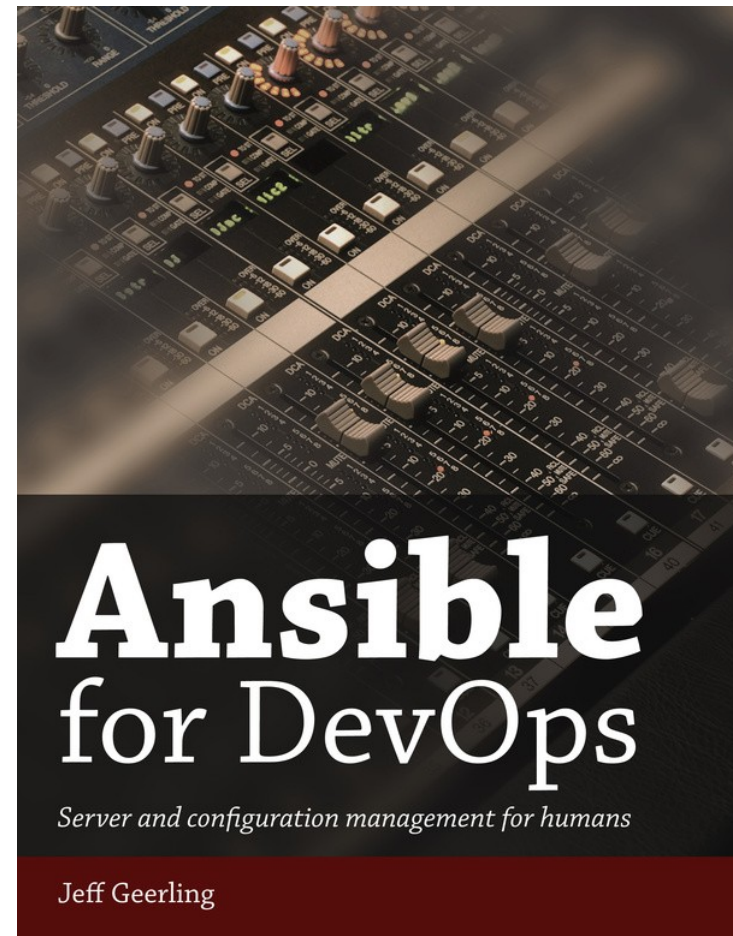
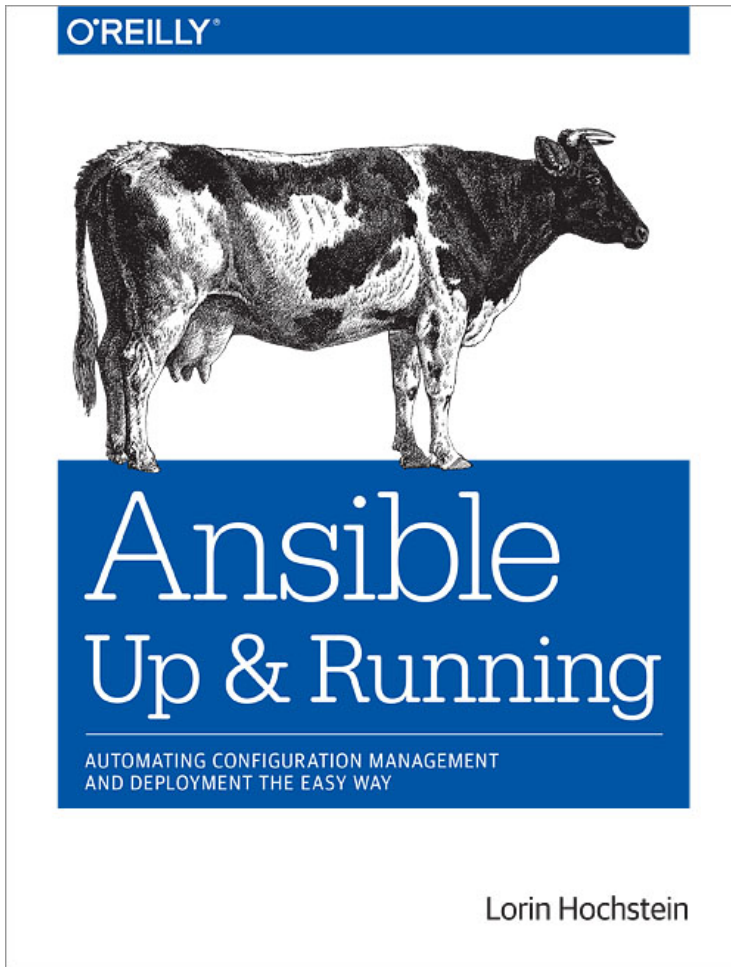
- Hoster verantwortlich für die Systemkonfiguration
- Softwareunternehmen verantwortlich für das Deployment
- Synchronisation zwischen Testserver und Produktionsserver wird vereinfacht

Weitere Informationen



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

Weitere Informationen



Weitere Informationen



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



JUnit 5
Das nächste große
Release steht vor
der Tür

Ansible
Konfigurationsmana-
gement 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>