

# Frankfurter Entwicklertage 2016

## Ansible für Entwickler

**Sandra Parsick**

info@sandra-parsick.de

@SandraParsick

# Zur meiner Person

- Freiberufliche Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
  - Java Enterprise Anwendungen
  - Agile Methoden
  - Software Craftmanship
  - Automatisierung von Entwicklungsprozessen
- Softwerkskammer Dortmund



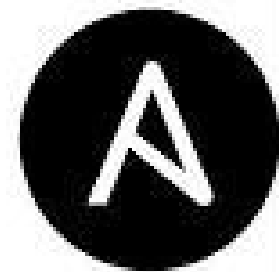
# Agenda

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.“*

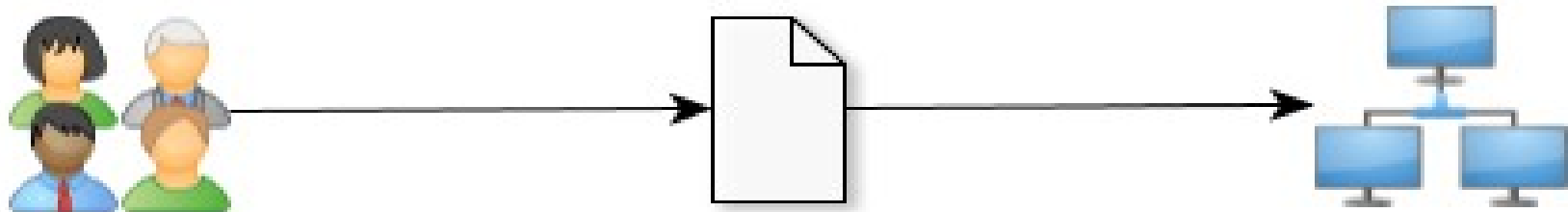
*<https://www.projektmagazin.de/glossarterm/konfigurationsmanagement>*

# Konfigurationsmanagement (KM)

- Softwarekonfiguration
- Hardwarekonfiguration
- Dienstleistungskonfiguration
- Systemkonfiguration

# Systemkonfiguration

- „Infrastructure As Code“





# Systemkonfiguration

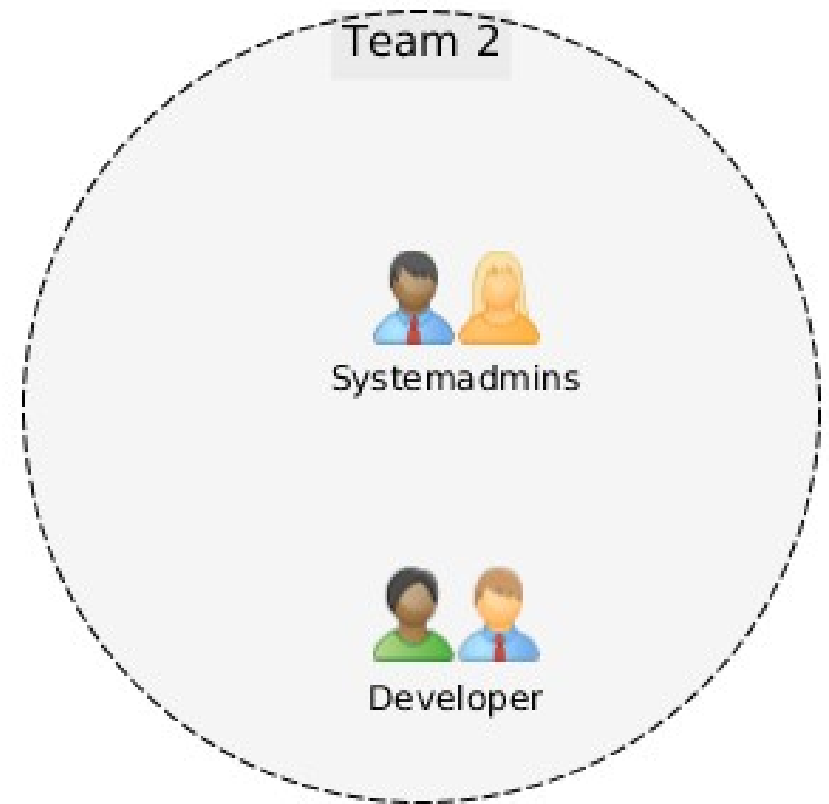
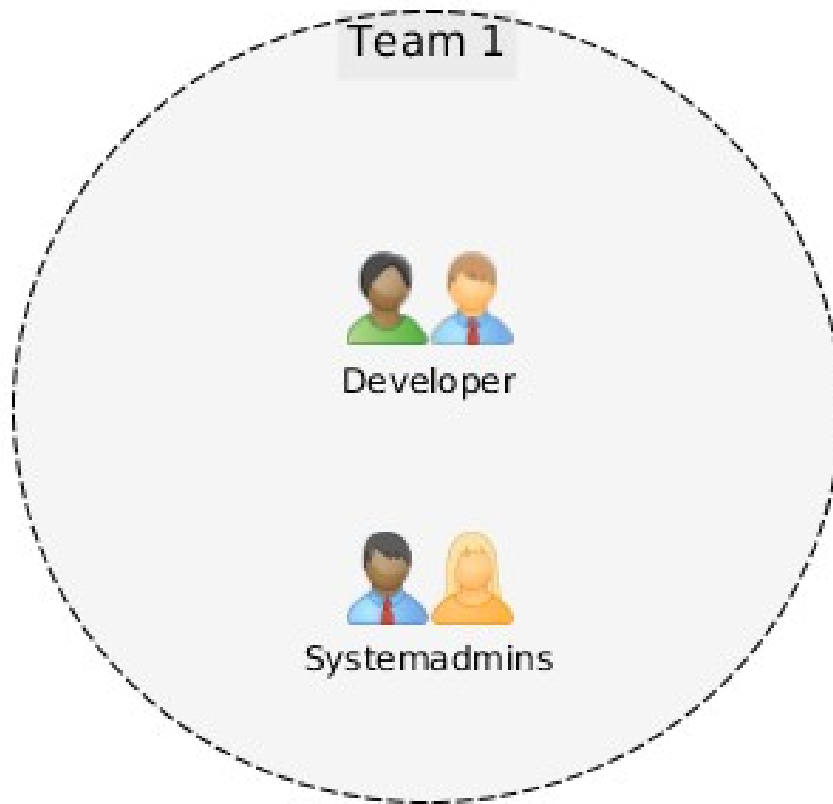
## - „Infrastructure As Code“



Warum ist es für Entwickler  
interessant?

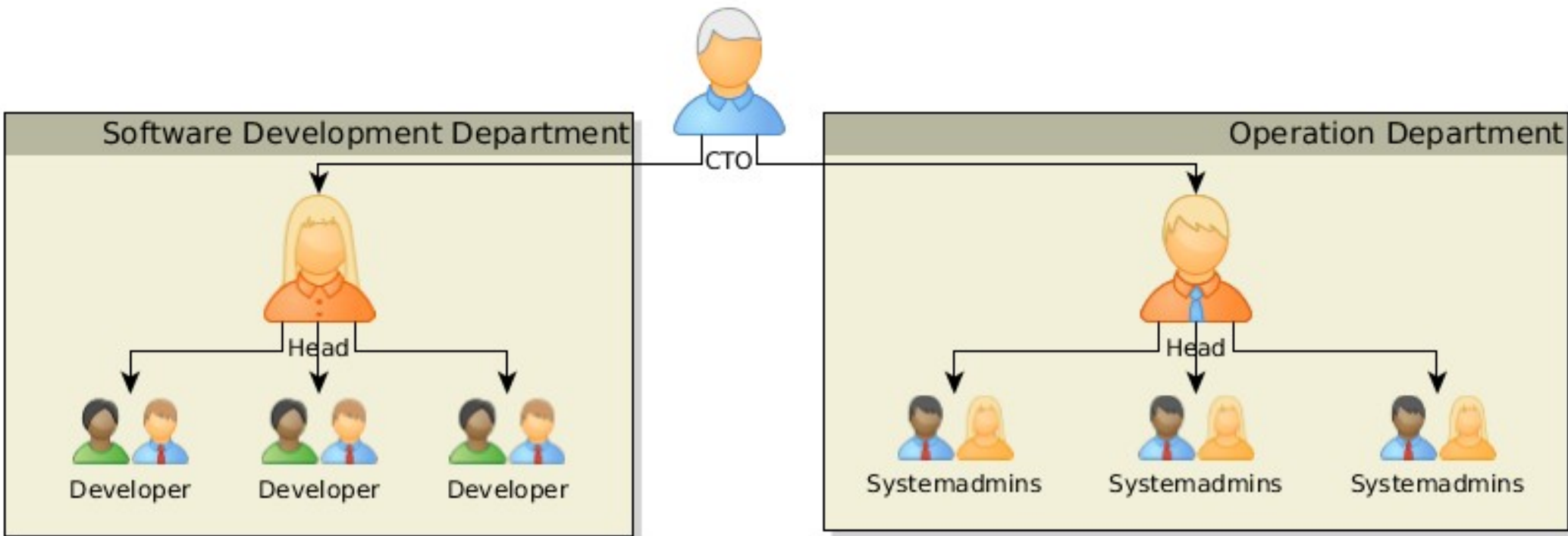
# Systemkonfiguration für Entwickler

Organisatorische Ausgangslage  
Wunsch



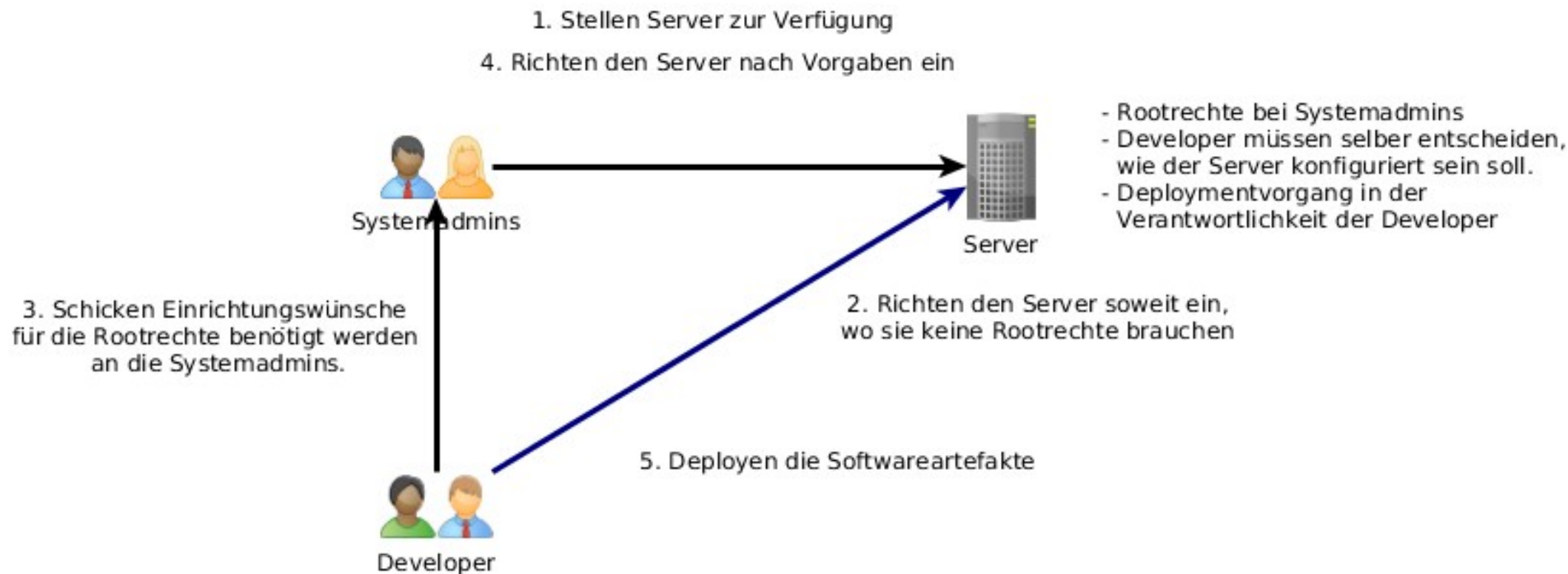
# Systemkonfiguration für Entwickler

Organisatorische Ausgangslage  
Realität



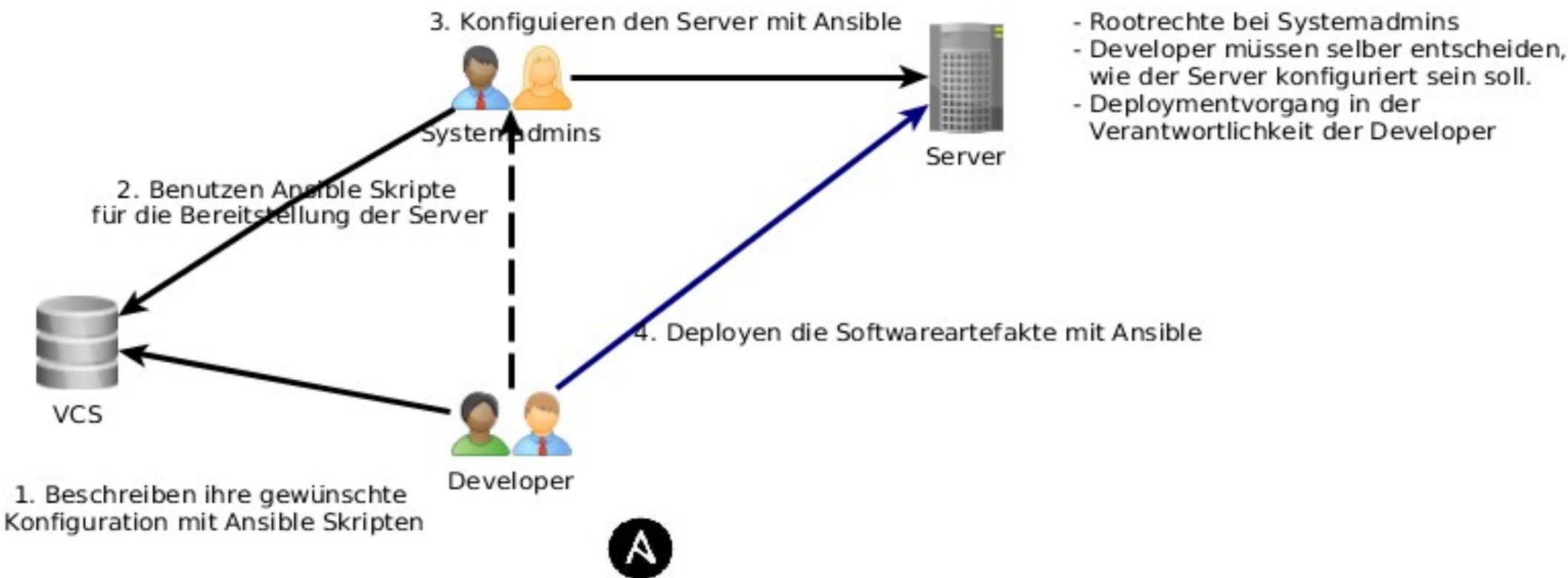
# Systemkonfiguration für Entwickler

## Prozess zwischen Development und Operation



# Systemkonfiguration für Entwickler

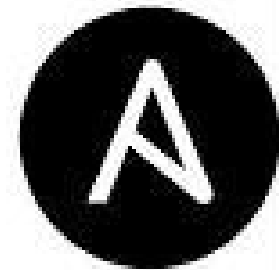
## Lösungsidee mit Ansible



# Einführung in Ansible

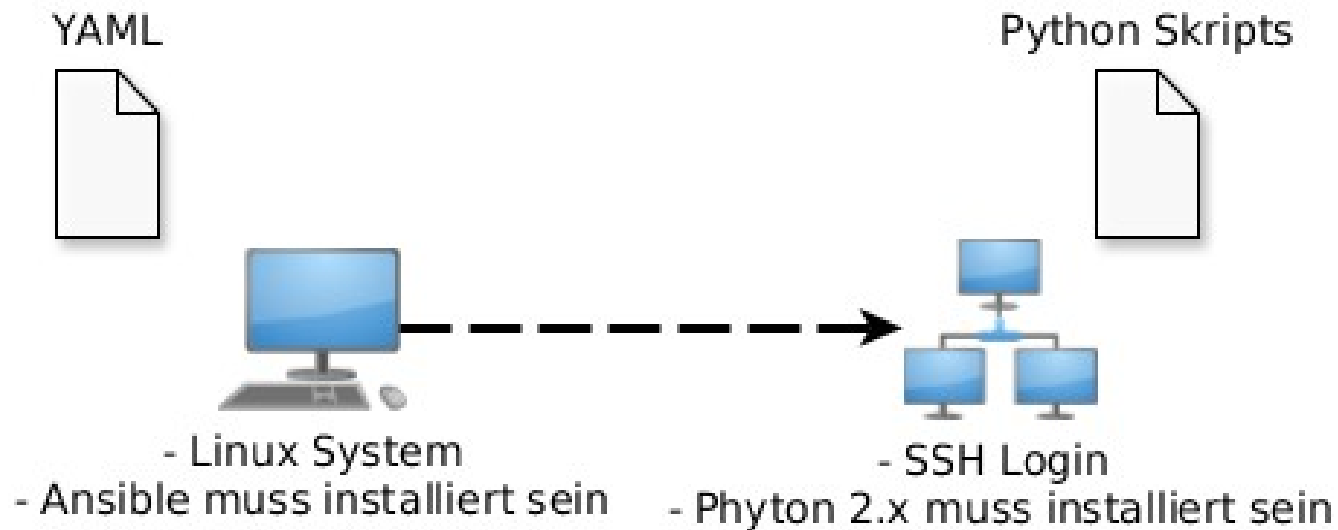
# Ansible

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





# Funktionsweise



# Exkurs: YAML

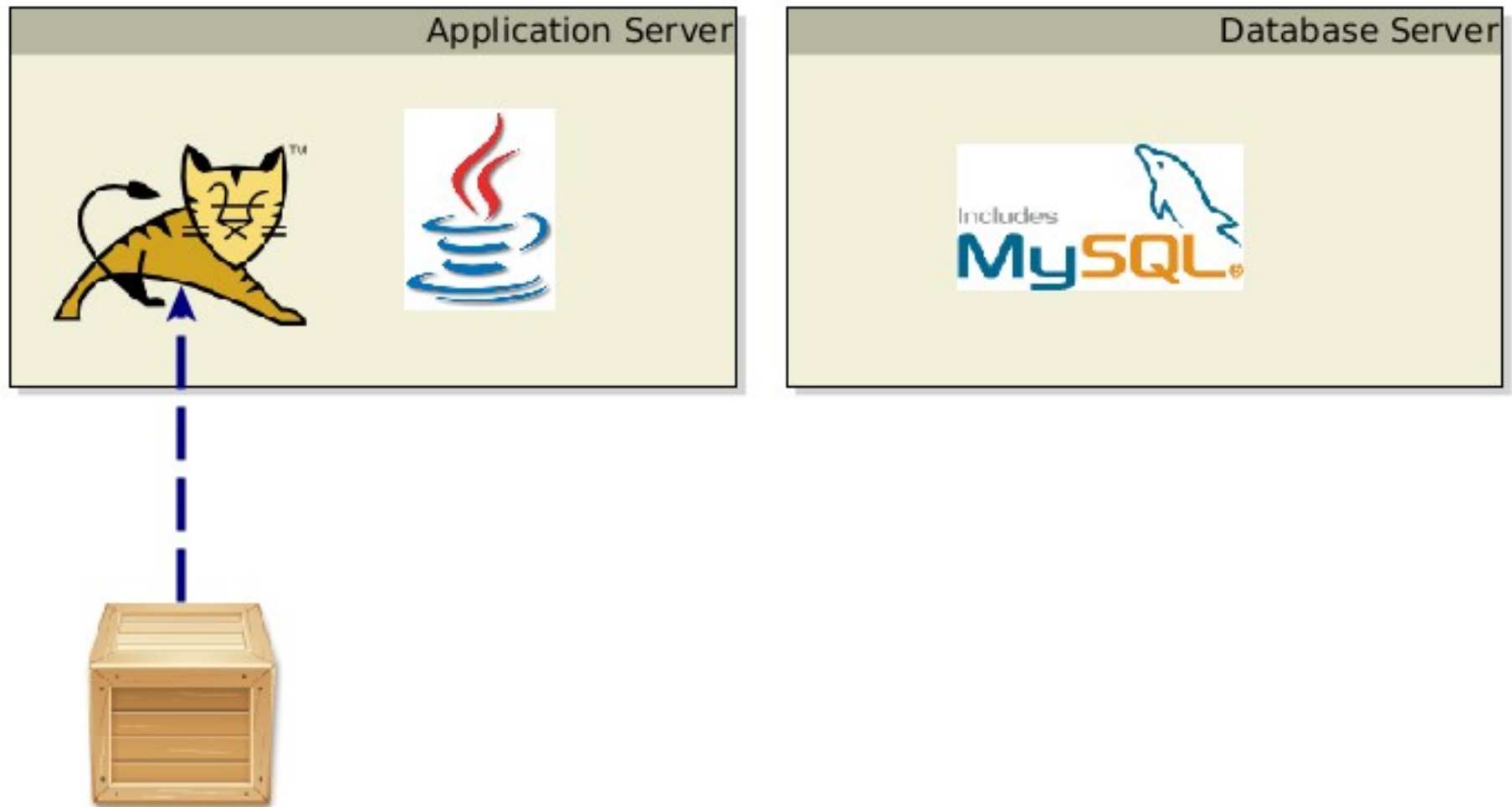
## YAML

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

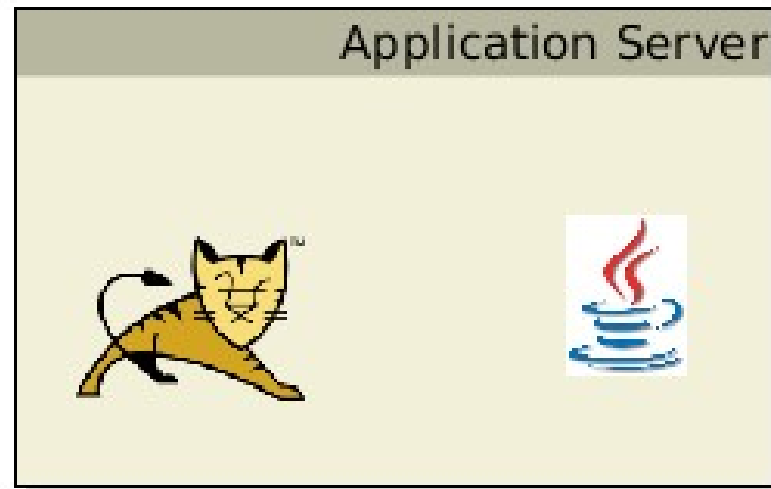
## JSON

```
{
  "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: 8.0.24
4      tomcat_base_name: apache-tomcat-{{ tomcat_version }}
5      #catalina_opts: "-Dkey=value"
6
7  tasks:
8      - name: install java
9        apt: name=openjdk-7-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
15        }}/bin/{{ 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"
31        src="roles/tomcat8/templates/setenv.sh.j2" mode=755
32        when: catalina_opts is defined
33
34      - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
35        register: tomcat_scripts
36        ignore_errors: yes
```

# Inventories

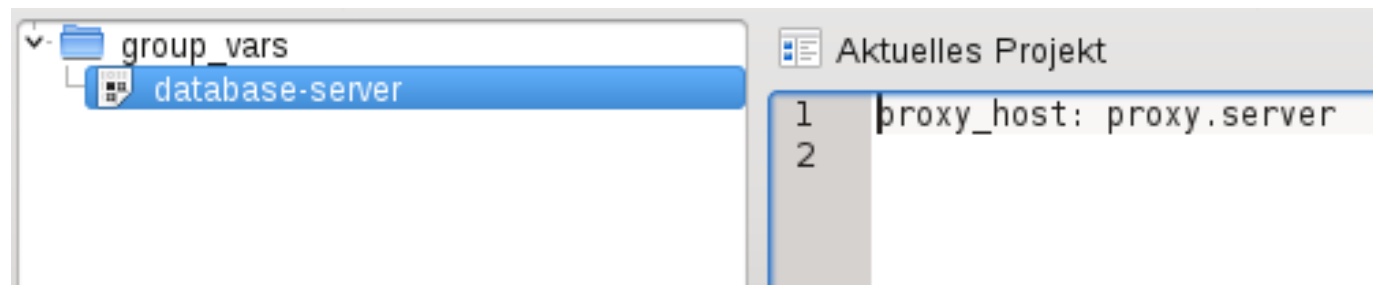
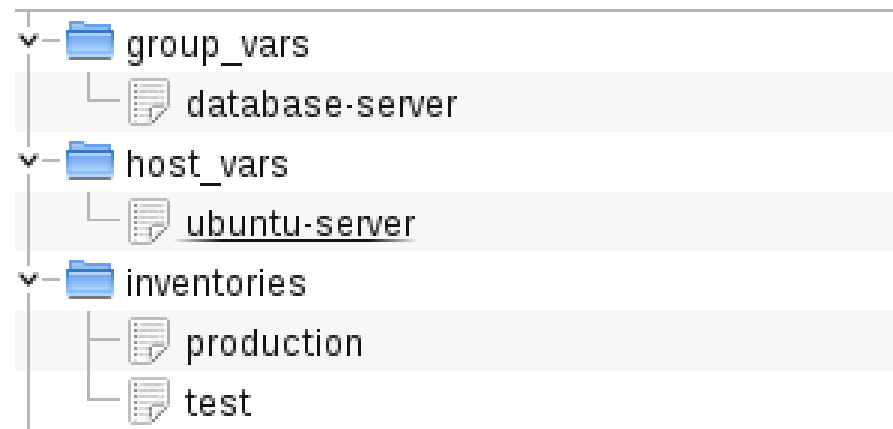
## Production

```
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



```
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
15   }}/bin/{{ 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"
31   src="roles/tomcat8/templates/setenv.sh.j2" mode=755
32   when: catalina_opts is defined
33
34 - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
35   register: tomcat_scripts
36   ignore_errors: yes
37
38 - name: ensure tomcat scripts are executable
39   file: name={{item}} mode=755
40   with_items: tomcat_scripts.stdout_lines
41
42 - name: install init.d script for tomcat
43   copy: src=roles/tomcat8/files/init.d/tomcat dest=/etc/init.d/tomcat owner=vagrant
44   group=vagrant mode=755
45   become: yes
46   become_method: sudo
```



# Templates

- setenv.sh.j2

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

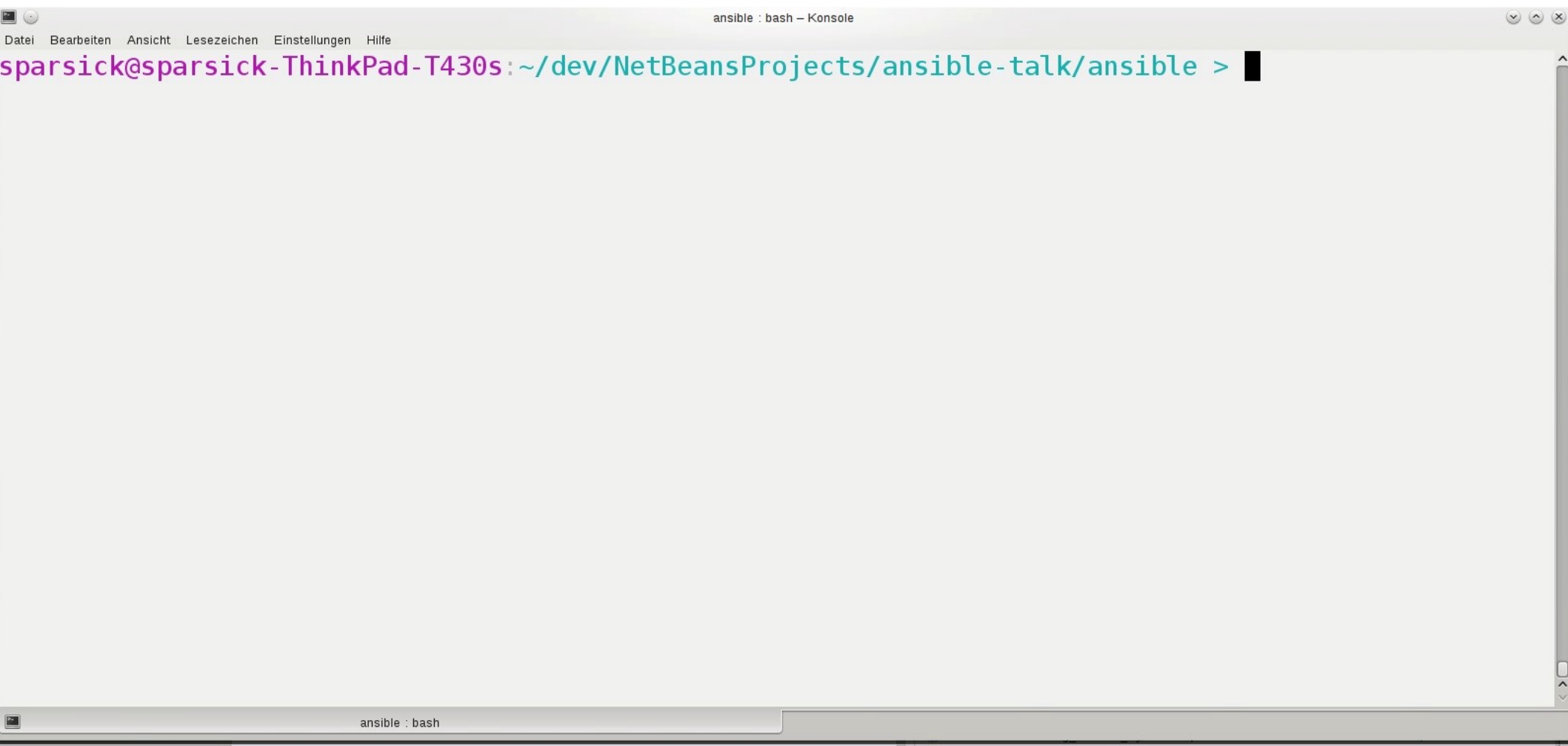
# Templates - Jinja2

- 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

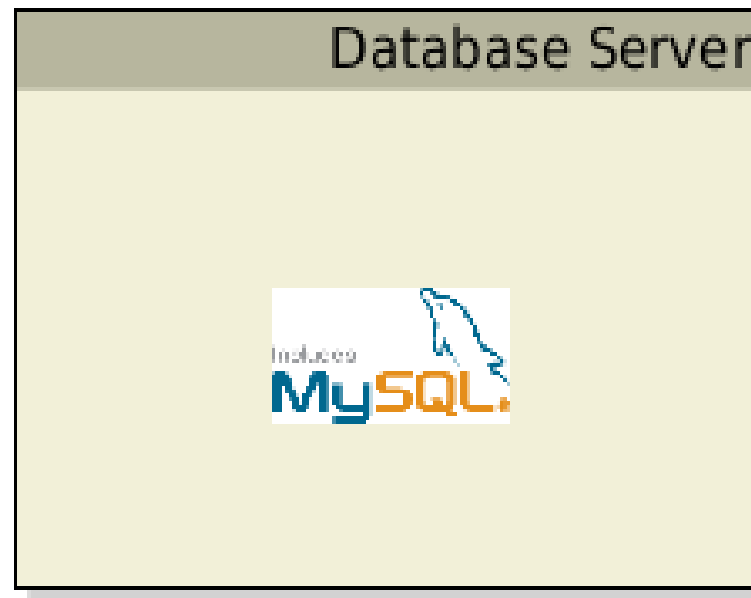


The image shows a screenshot of a terminal window within the NetBeans IDE. The window title is "ansible : bash – Konsole". The menu bar at the top includes "Datei", "Bearbeiten", "Ansicht", "Lesezeichen", "Einstellungen", and "Hilfe". The terminal content shows the user "sparsick" at the machine "sparsick-ThinkPad-T430s" with a prompt character "s". The current directory is "~/dev/NetBeansProjects/ansible-talk/ansible", and the prompt is ">". A black cursor is visible after the prompt. The status bar at the bottom of the terminal window displays "ansible : bash".

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

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/my.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
28
```

# Setup Database Server Playbook



# Setup Database Server Playbook



```
1  hosts: application-server
2  vars:
3      tomcat_version: 8.0.24
4      tomcat_base_name: apache-tomcat-{{ tomcat_version }}
5      #catalina_opts: "-Dkey=value"
6
7  tasks:
8      - name: install java
9        apt: name=openjdk-7-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
15        }}/bin/{{ 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"
31        src="roles/tomcat8/templates/setenv.sh.j2" mode=755
32        when: catalina_opts is defined
33
34      - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
35        register: tomcat_scripts
36        ignore_errors: yes
```

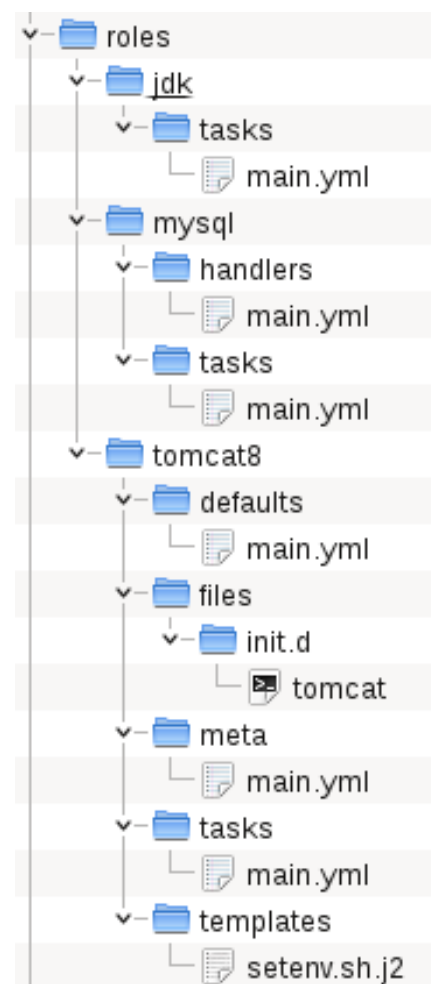


```
12
13 - name: Download current Tomcat 8 version
14   local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version
15   }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp
16
17 - name:
18   file: name=/opt mode=777
19   become: yes
20   become_method: sudo
21
22 - name: 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"
31   src="roles/tomcat8/templates/setenv.sh.j2" mode=755
32   when: catalina_opts is defined
33
34 - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
35   register: tomcat_scripts
36   ignore_errors: yes
37
38 - name: ensure tomcat scripts are executable
39   file: name={{item}} mode=755
40   with_items: tomcat_scripts.stdout_lines
41
42 - name: install init.d script for tomcat
43   copy: src=roles/tomcat8/files/init.d/tomcat dest=/etc/init.d/tomcat owner=vagrant
44   group=vagrant mode=755
45   become: yes
46   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/my.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
28
```

# Roles

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



# Setup Playbooks mit Roles

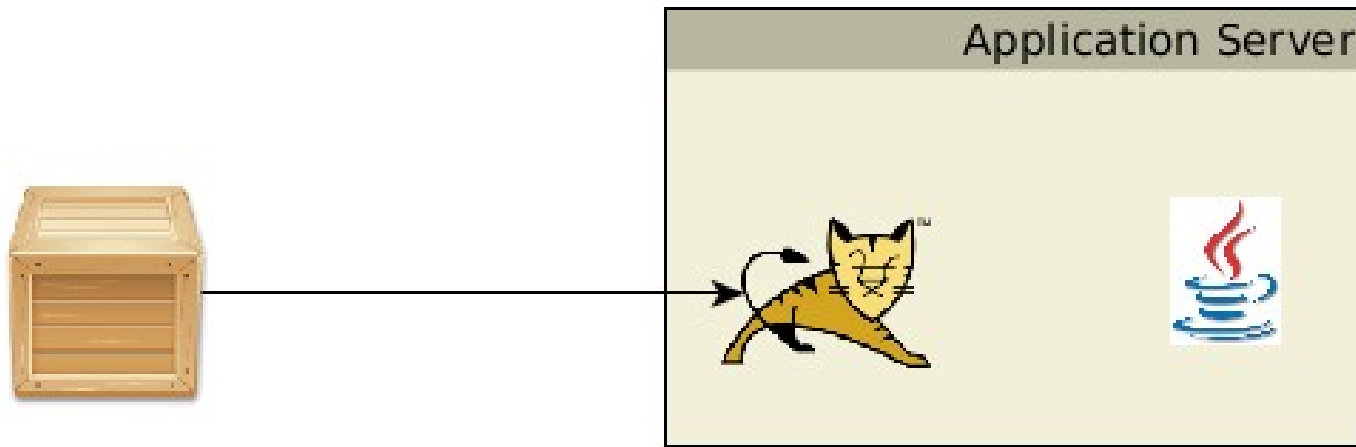
- Setup Application Server

```
1  - hosts: application-server
2    roles:
3      - jdk
4      - { role: tomcat8, tomcat_version: 8.0.30 }
5
6
7
```

- Setup Database Server

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

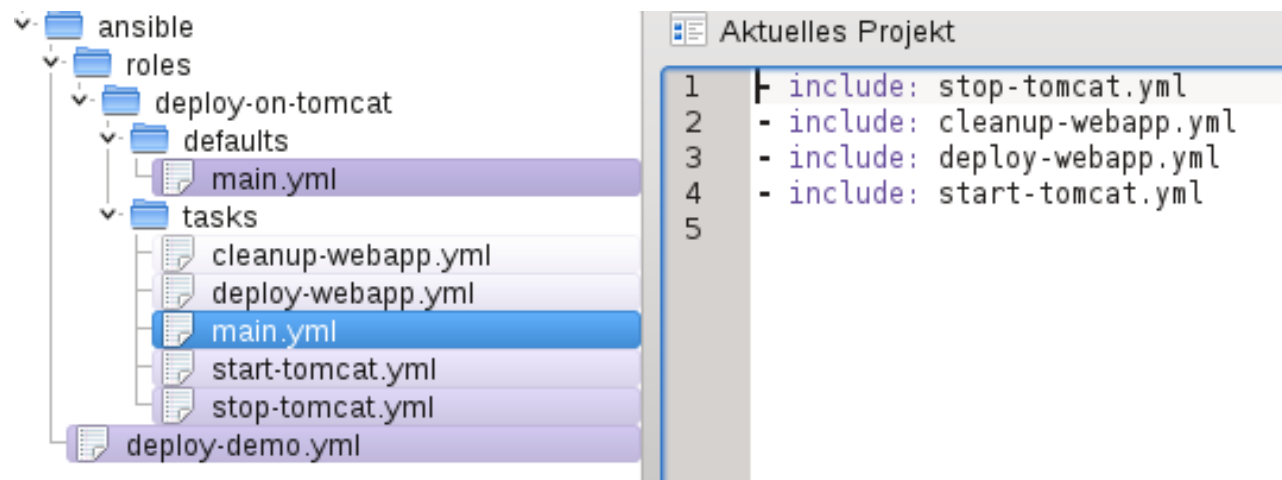
# Java Webapplikation Deployment



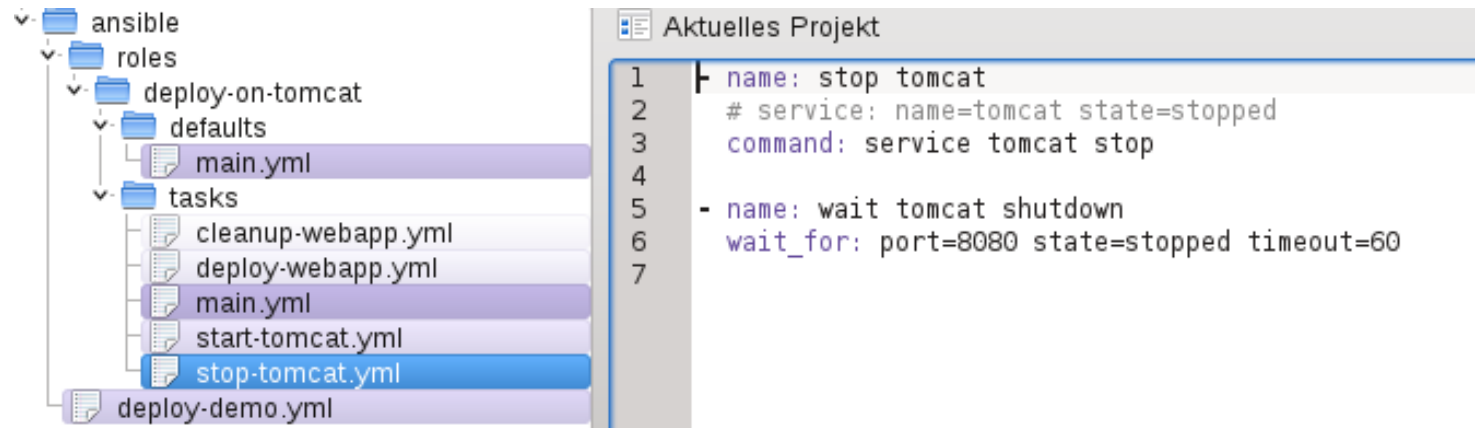
# Deploy Application Playbook

```
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:**

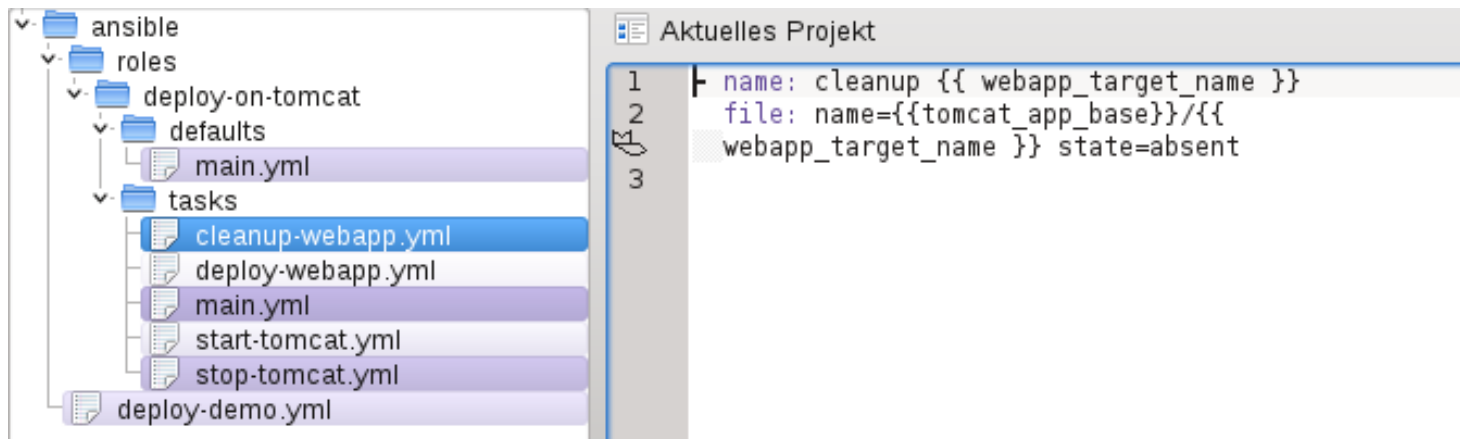
- ansible
  - roles
    - deploy-on-tomcat
      - defaults
        - main.yml
      - tasks
        - cleanup-webapp.yml
        - deploy-webapp.yml
        - main.yml
        - start-tomcat.yml
        - stop-tomcat.yml (Selected)
  - deploy-demo.yml

**Task Content (stop-tomcat.yml):**

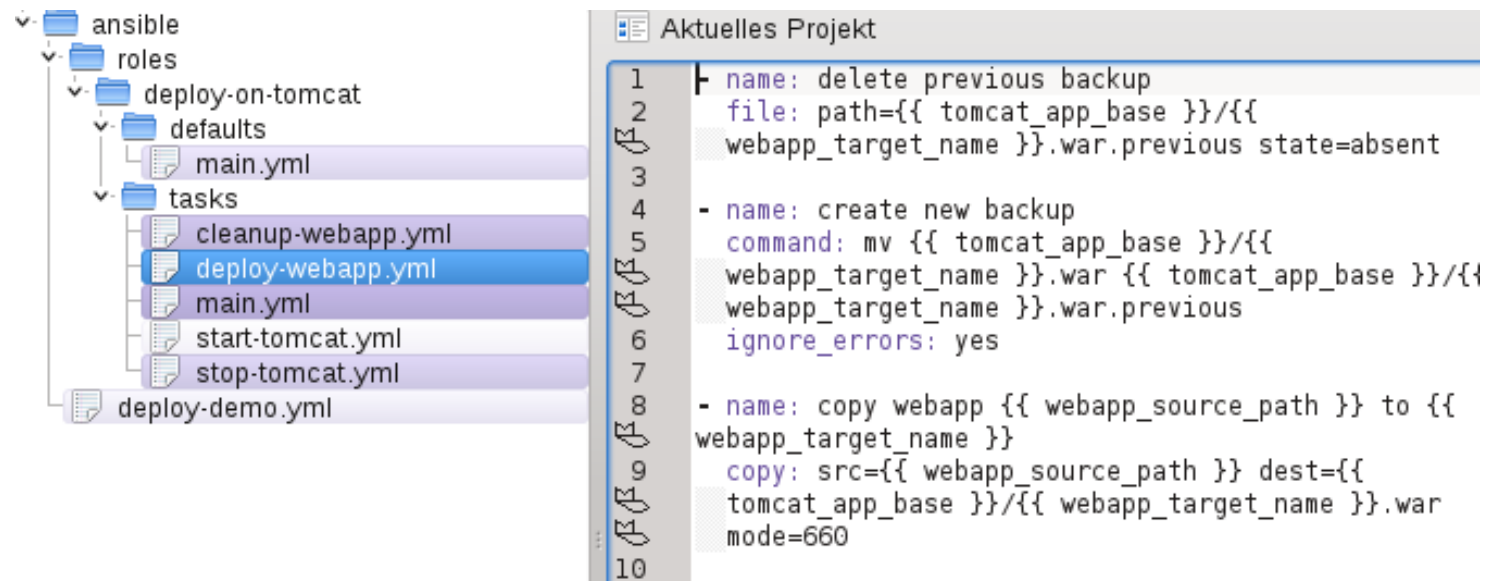
```
1  name: stop tomcat
2    # service: name=tomcat state=stopped
3    command: service tomcat stop
4
5  - name: wait tomcat shutdown
6    wait_for: port=8080 state=stopped timeout=60
7
```



# deploy-on-tomcat Role



# deploy-on-tomcat Role



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

**File Explorer (Left Panel):**

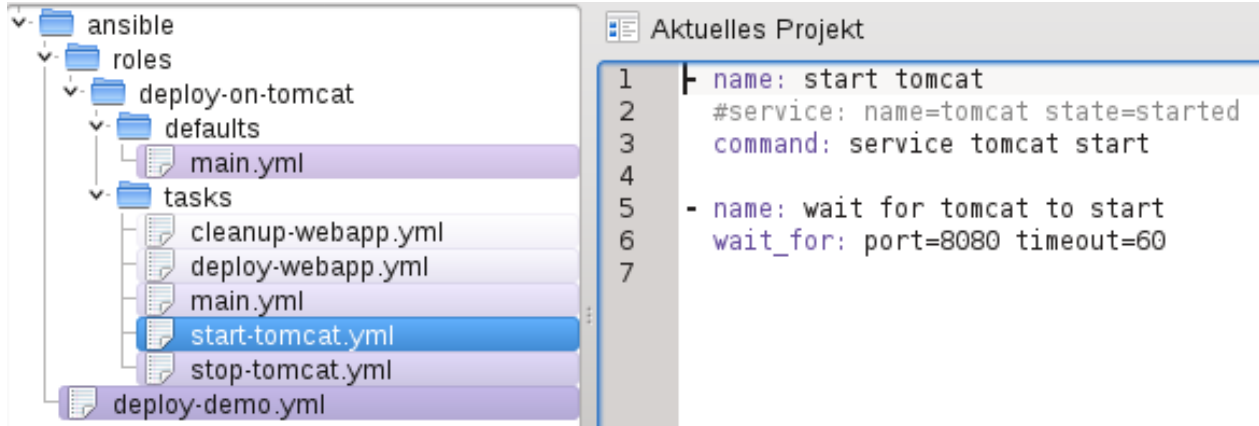
- ansible
  - roles
    - deploy-on-tomcat
      - defaults
        - main.yml
      - tasks
        - cleanup-webapp.yml
        - deploy-webapp.yml (selected)
        - main.yml
        - start-tomcat.yml
        - stop-tomcat.yml
  - deploy-demo.yml

**Task List (Right Panel):**

Aktuelles Projekt

```
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



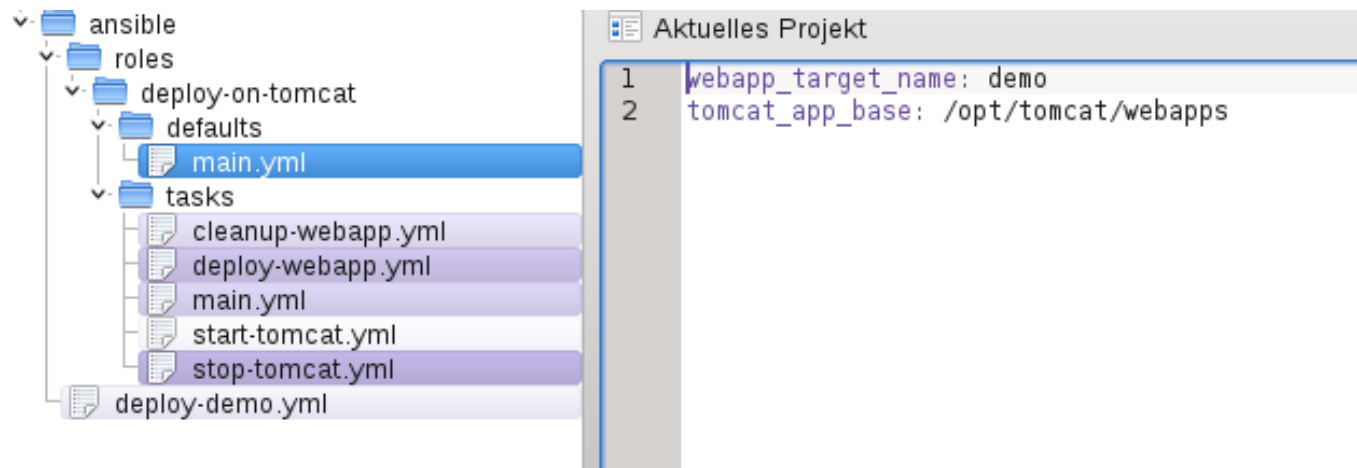
The screenshot displays an IDE interface with a file explorer on the left and a task editor on the right. The file explorer shows the following structure:

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

The task editor on the right, titled "Aktuelles Projekt", shows the following tasks:

```
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



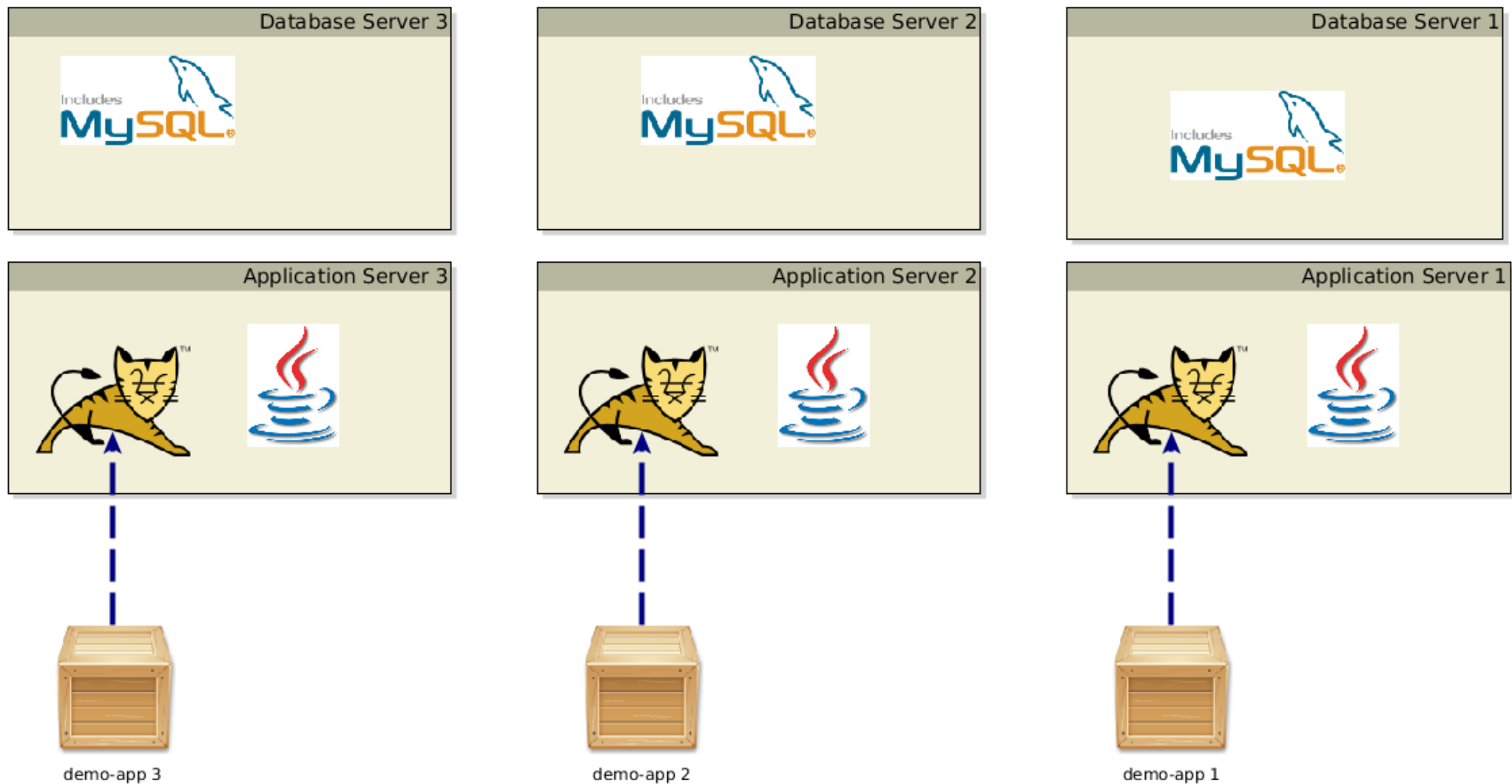
# Deploy Application Playbook









# 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

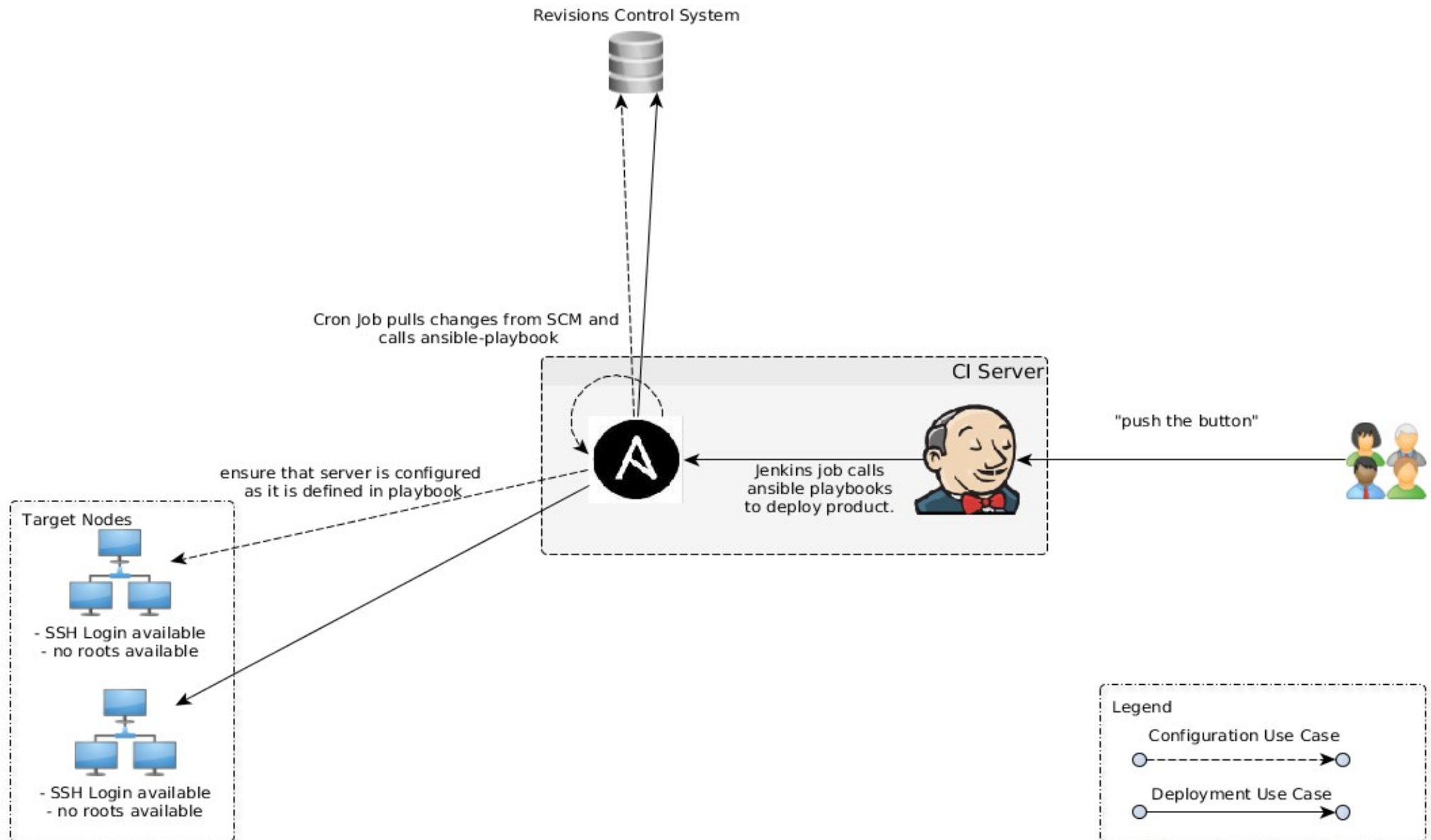


# Warum Roles?

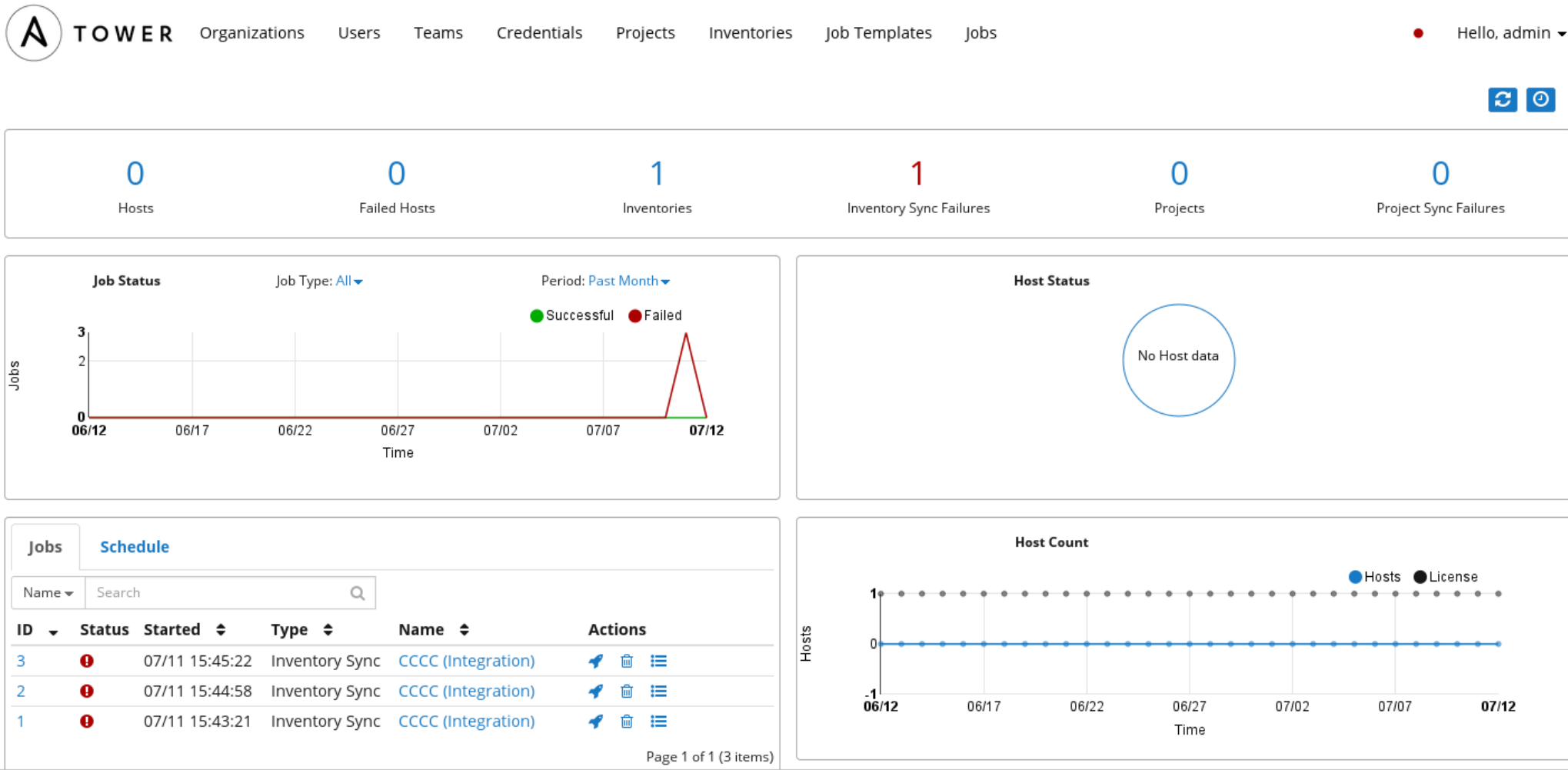
```
1 - hosts: demo1-application-server
2   roles:
3     - {role: deploy-on-tomcat, webapp_source_path: ./demo1-1.0-
4       SNAPSHOT.war, webapp_target_name: demo1 }
```

```
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



# 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-7-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

## Ansible

- 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

## Puppet

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



# Vergleich

## Puppet

```
▼ 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' ],  
    }  
}
```

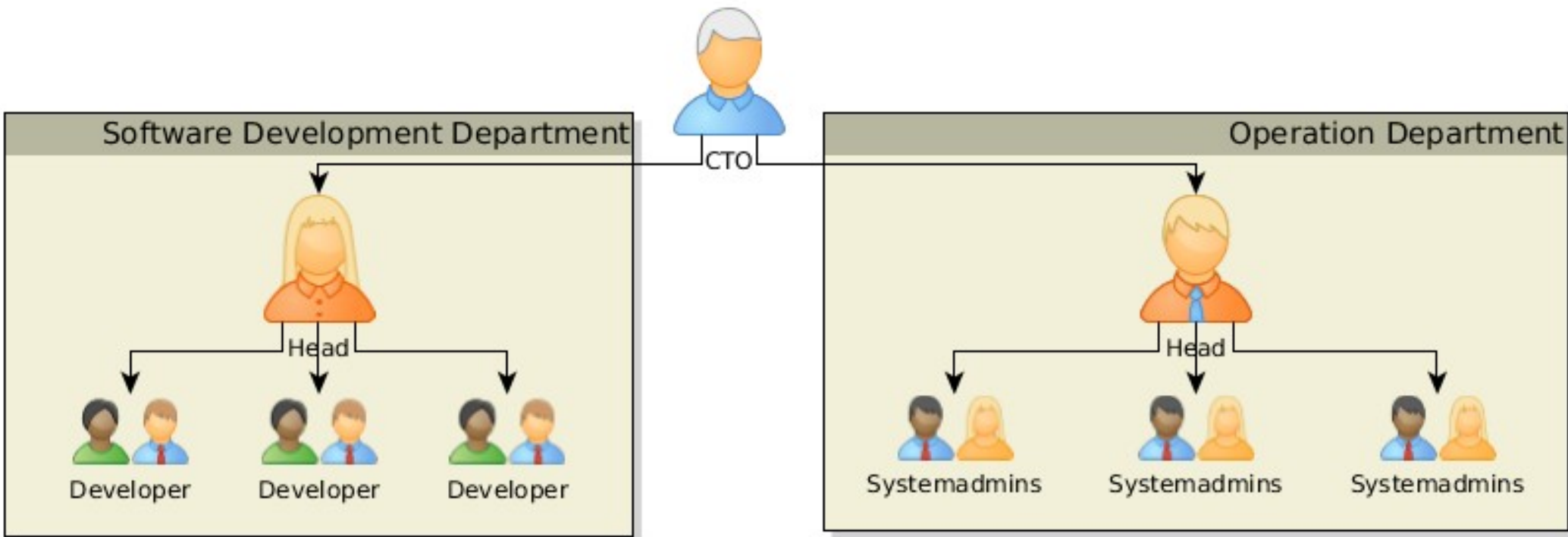
## Ansible

```
- name: add nodejs ppa  
  apt_repository: repo='ppa:chris-lea/node.js'  
  
- name: install nodejs package  
  apt: name=nodejs update-cache=yes
```

# Weitere Einsatzszenarien aus Entwicklersicht

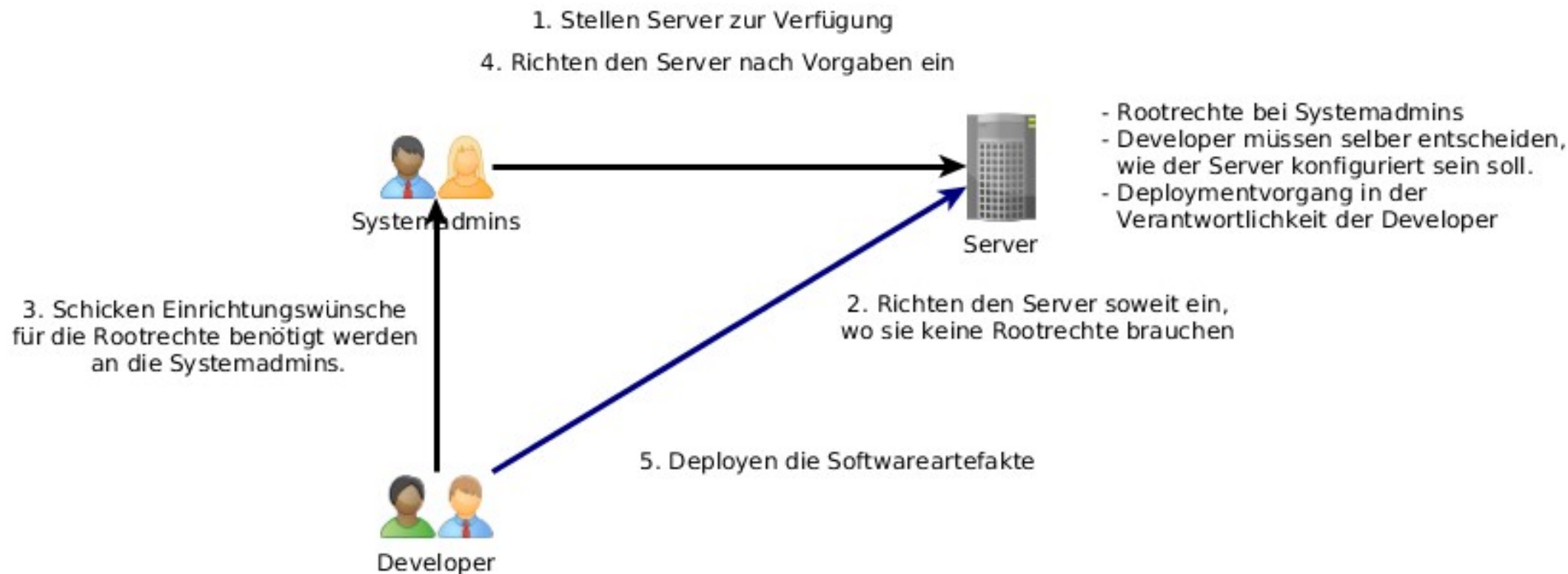
# Systemkonfiguration für Entwickler

Organisatorische Ausgangslage  
Realität



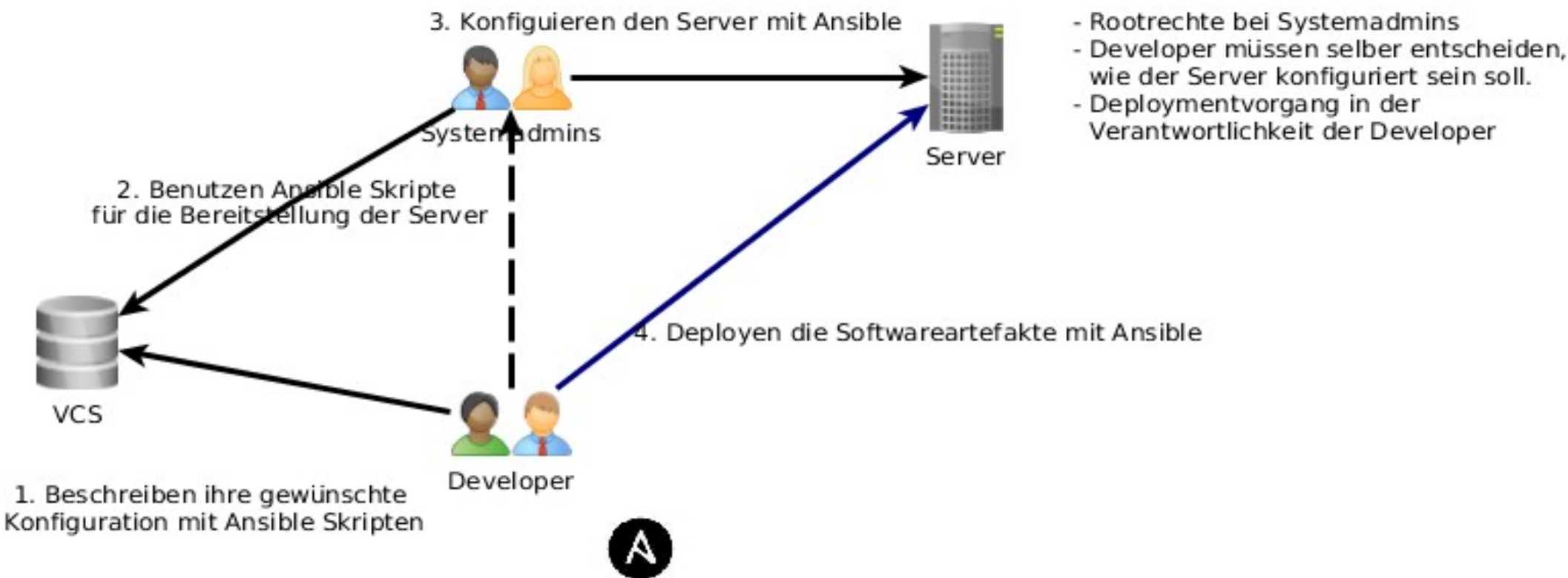
# Systemkonfiguration für Entwickler

## Prozess zwischen Development und Operation



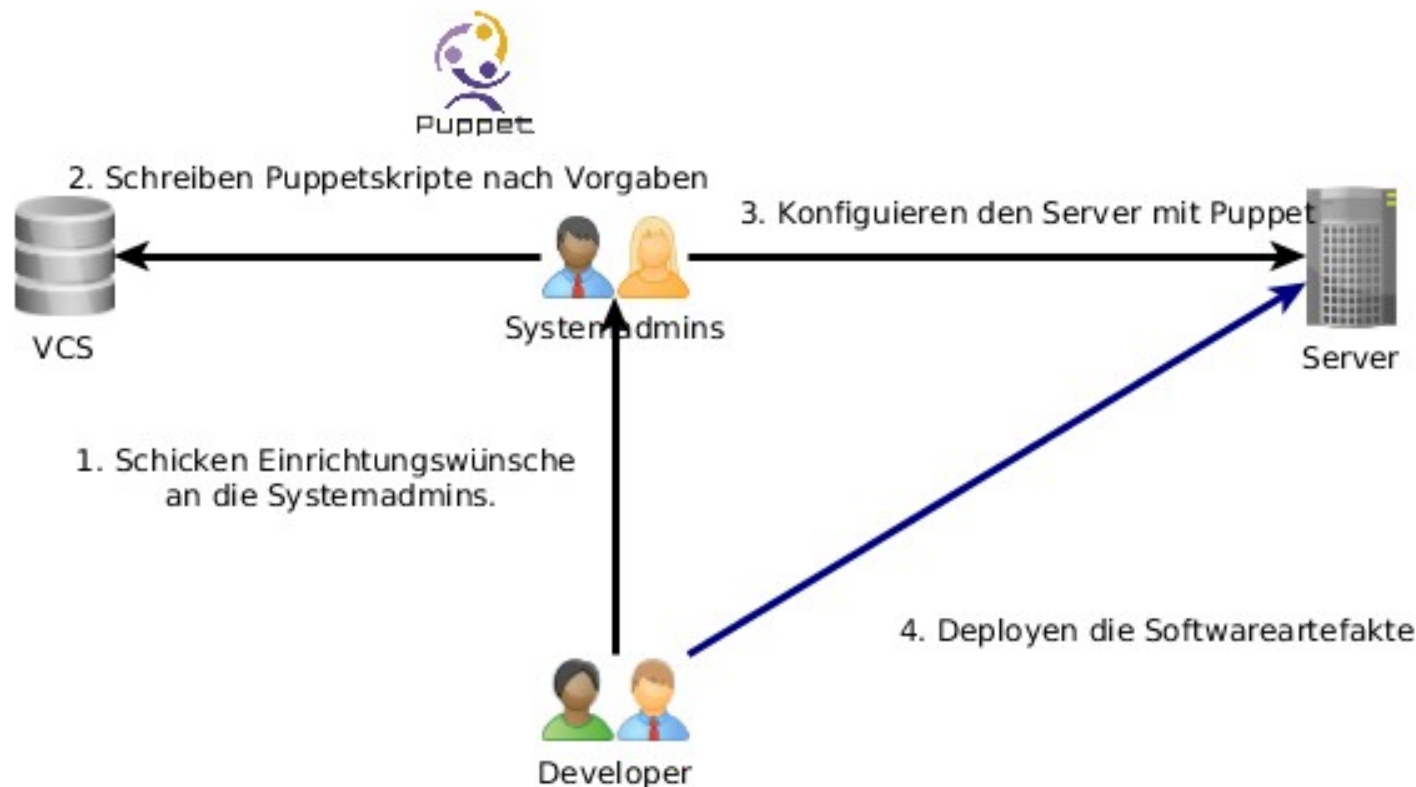
# Systemkonfiguration für Entwickler

## Lösungsidee mit Ansible



# Systemkonfiguration für Entwickler

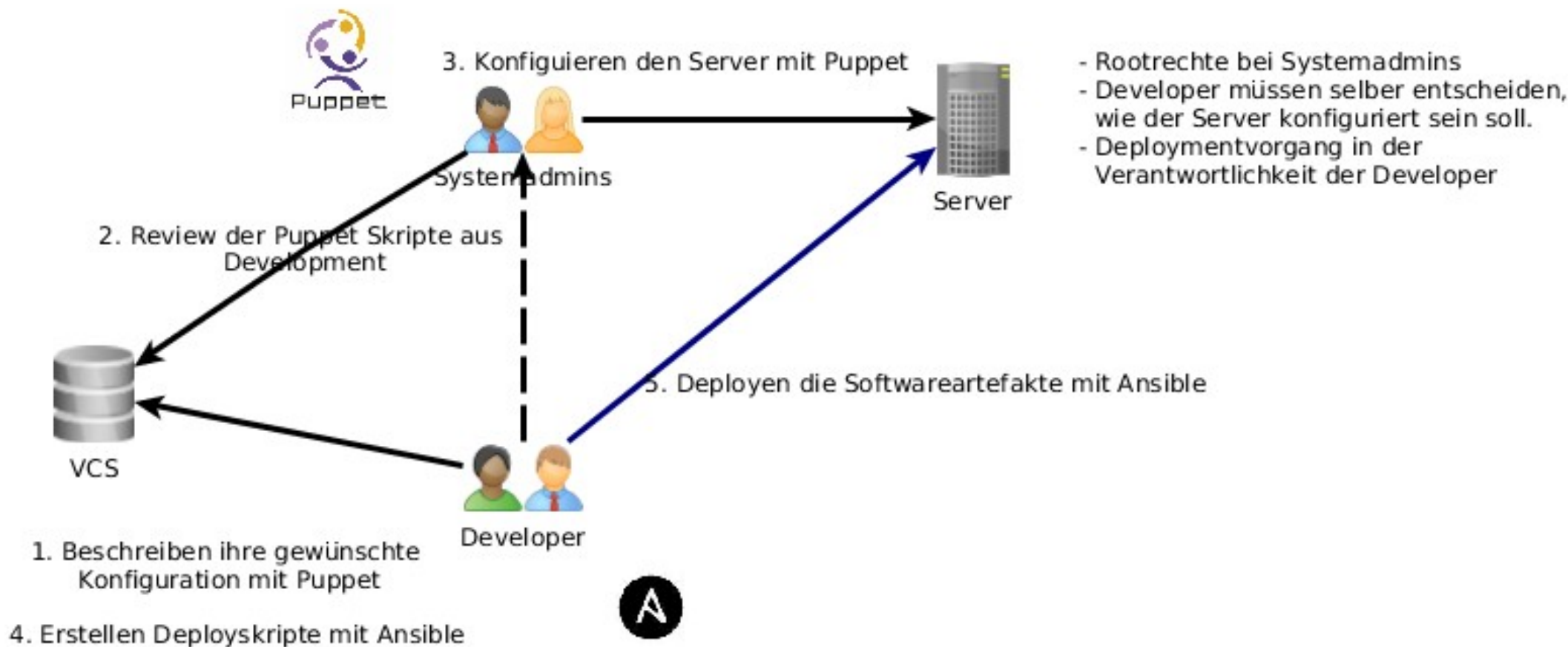
## Variante - Prozess zwischen Development und Operation



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

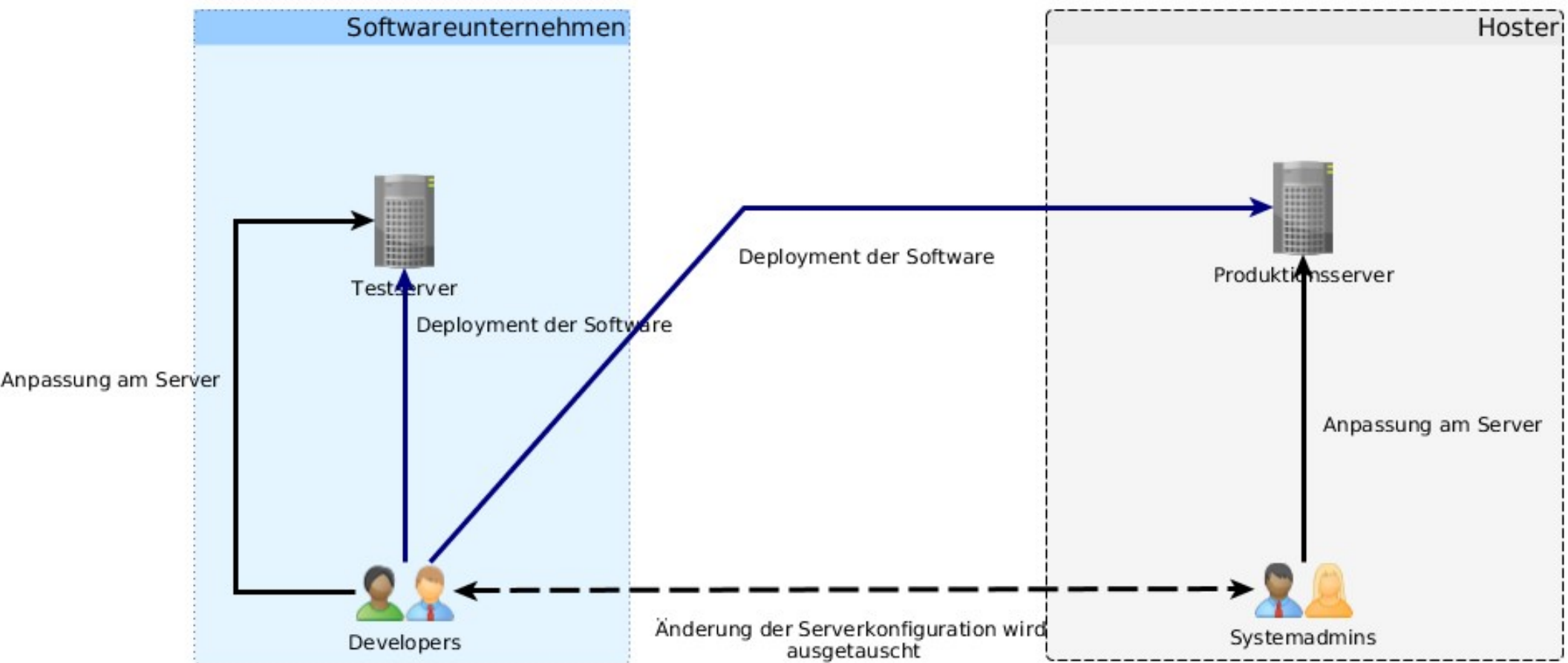
# Systemkonfiguration für Entwickler

## Lösungsvariante



# Systemkonfiguration für Entwickler

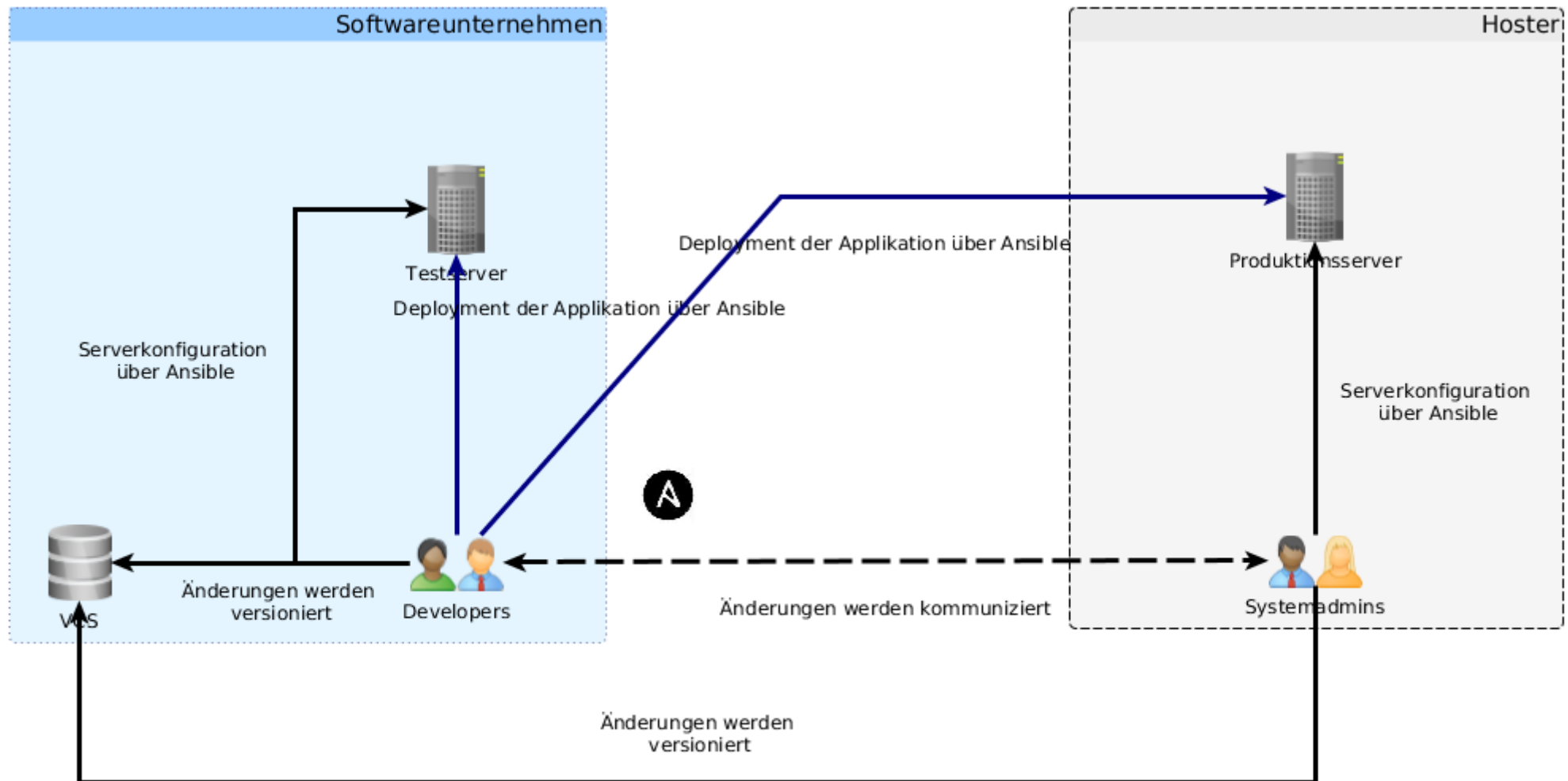
Produktionsserver sind beim externen Hoster





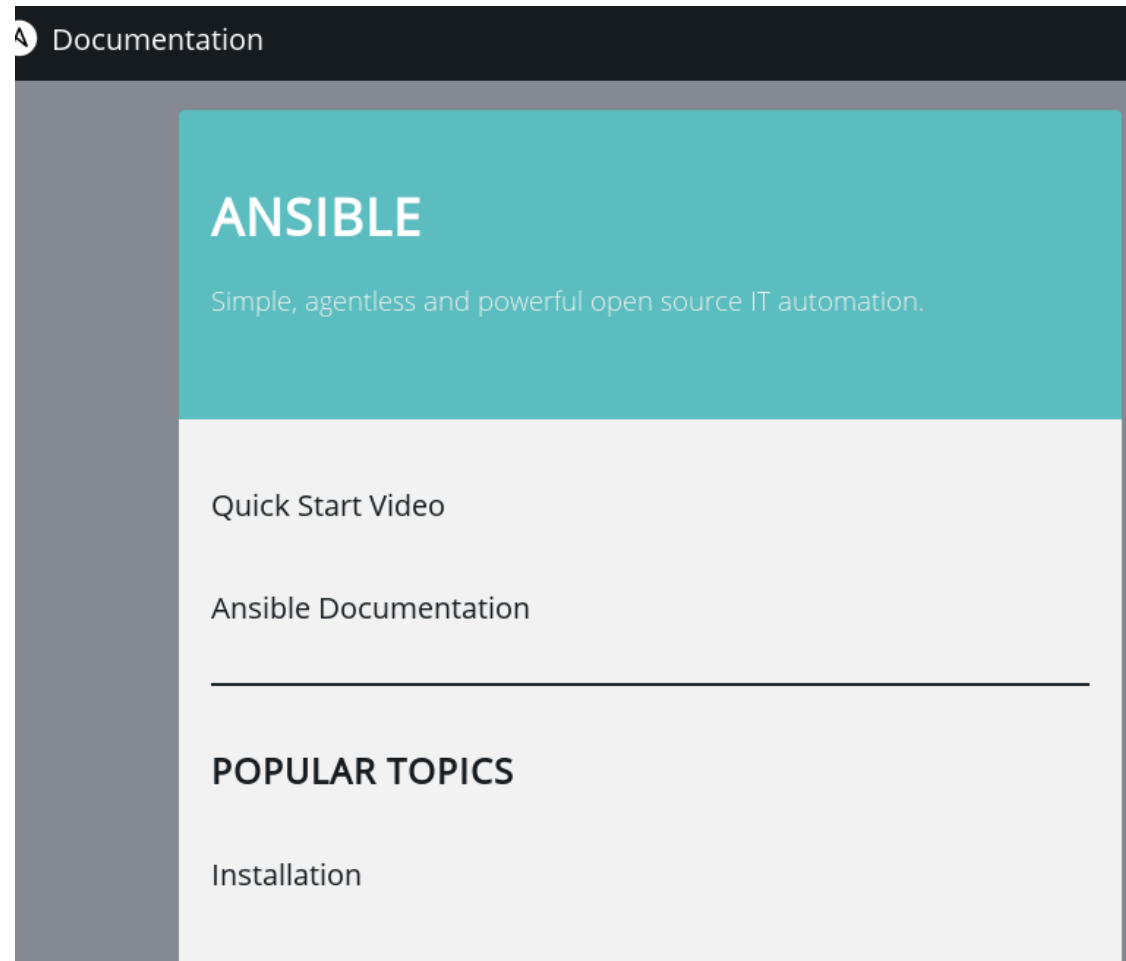
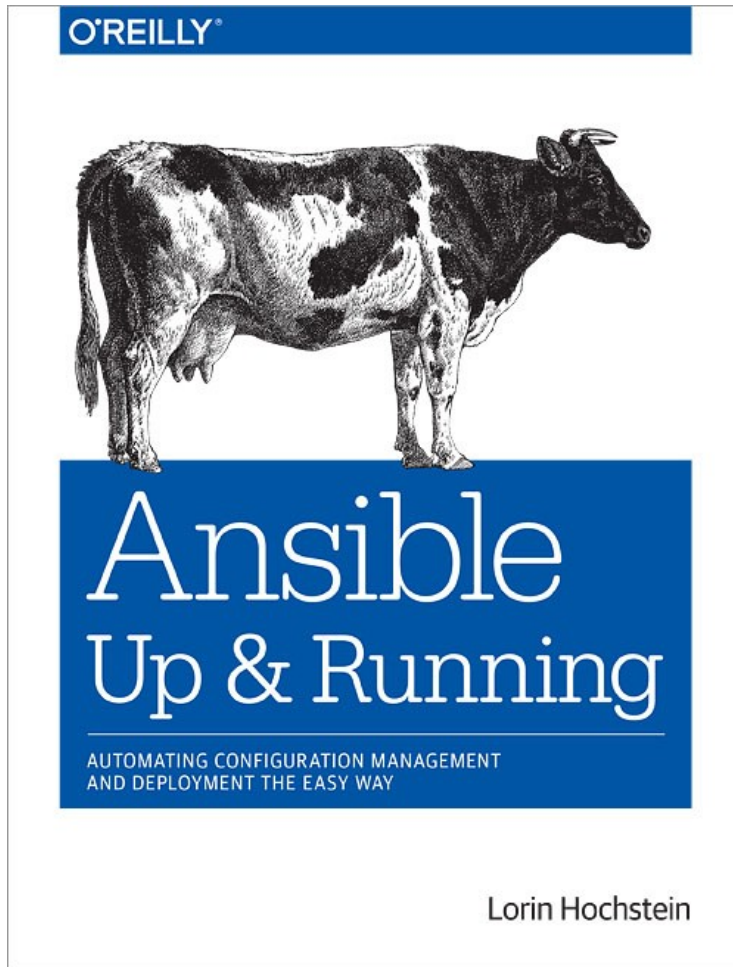
# Systemkonfiguration für Entwickler

## Lösungsidee



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

# Weitere Informationen



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

# Fragen?

@SandraParsick  
info@sandra-parsick.de