JUG Saxony Day 2016, Dresden

# Ansible für Entwickler

Sandra Parsick

mail@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 zur seiner Konkurrenz?
5. Weitere Einsatzszenarien aus Entwicklersicht

# Ansible
# Was ist das?

# Ansible

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

# Konfigurationsmanagement (KM)

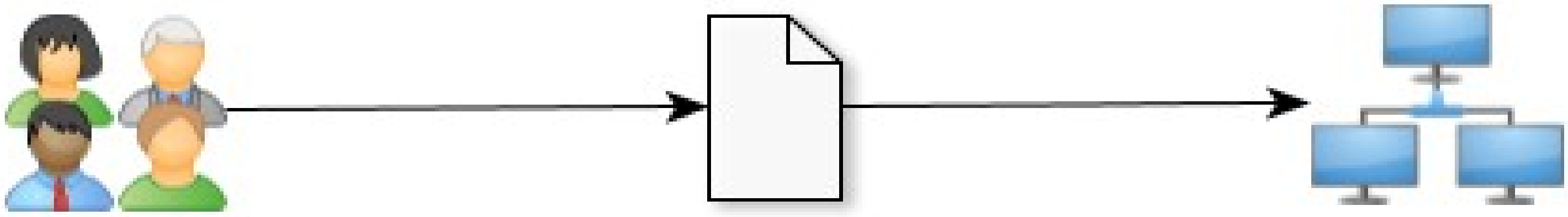
*„Das KM umfasst alle technischen, organisatorischen und beschlussfassenden Maßnahmen und Strukturen, die sich mit der Konfiguration (Spezifikation) eines Produkts befassen.“*

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

# Konfigurationsmanagement (KM)

- Softwarekonfiguration
- Hardwarekonfiguration
- Dienstleistungskonfiguration
- Systemkonfiguration

# Systemkonfiguration
# - „Infrastructure As Code“



# Systemkonfiguration
# - „Infrastructure As Code“

SALTSTACK

CFEngine

CHEF™

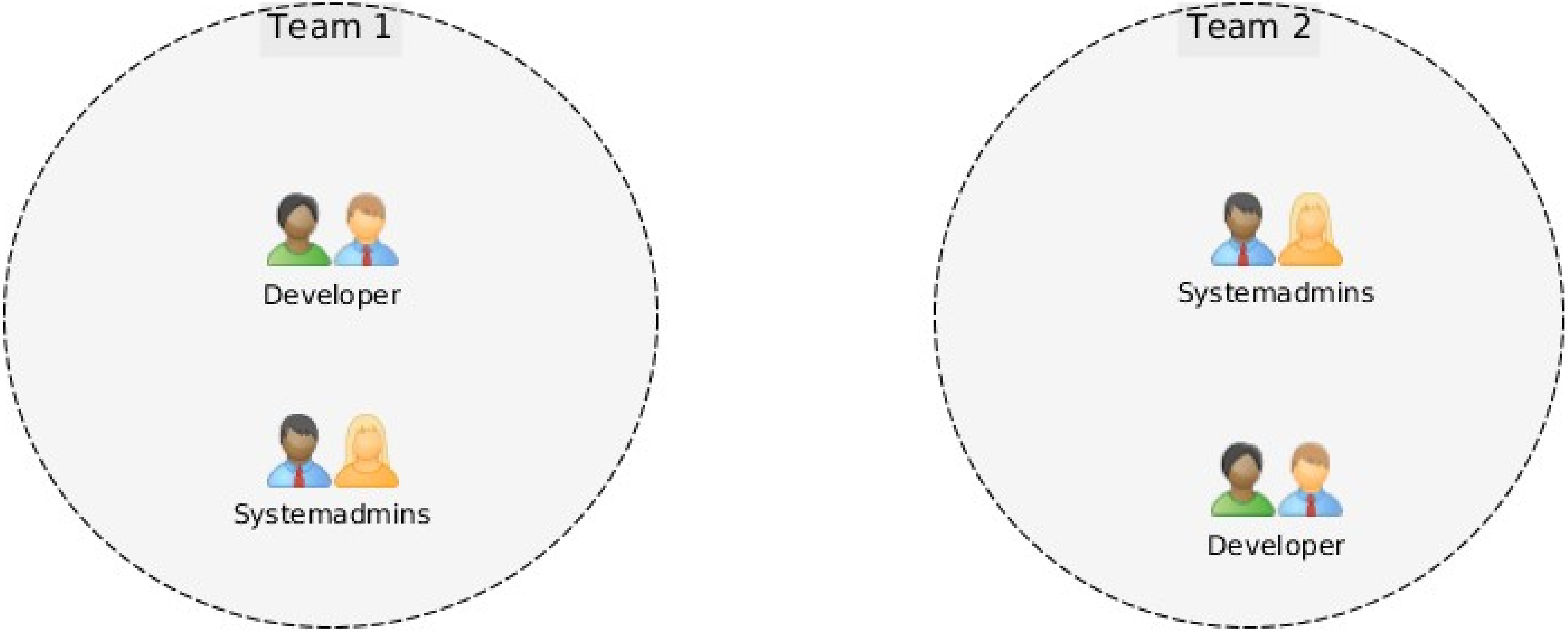
Puppet

Ansible

# Warum ist es für Entwickler interessant?

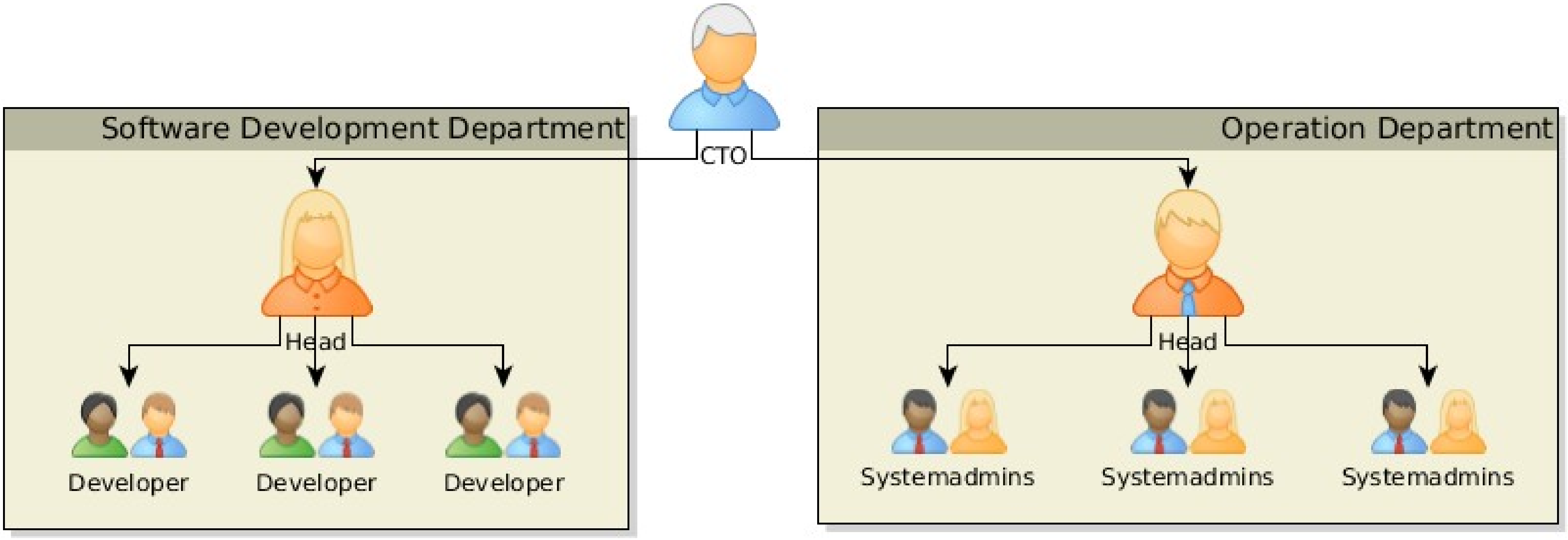
# Systemkonfiguration für Entwickler

## Organisatorische Ausgangslage
## Wunsch



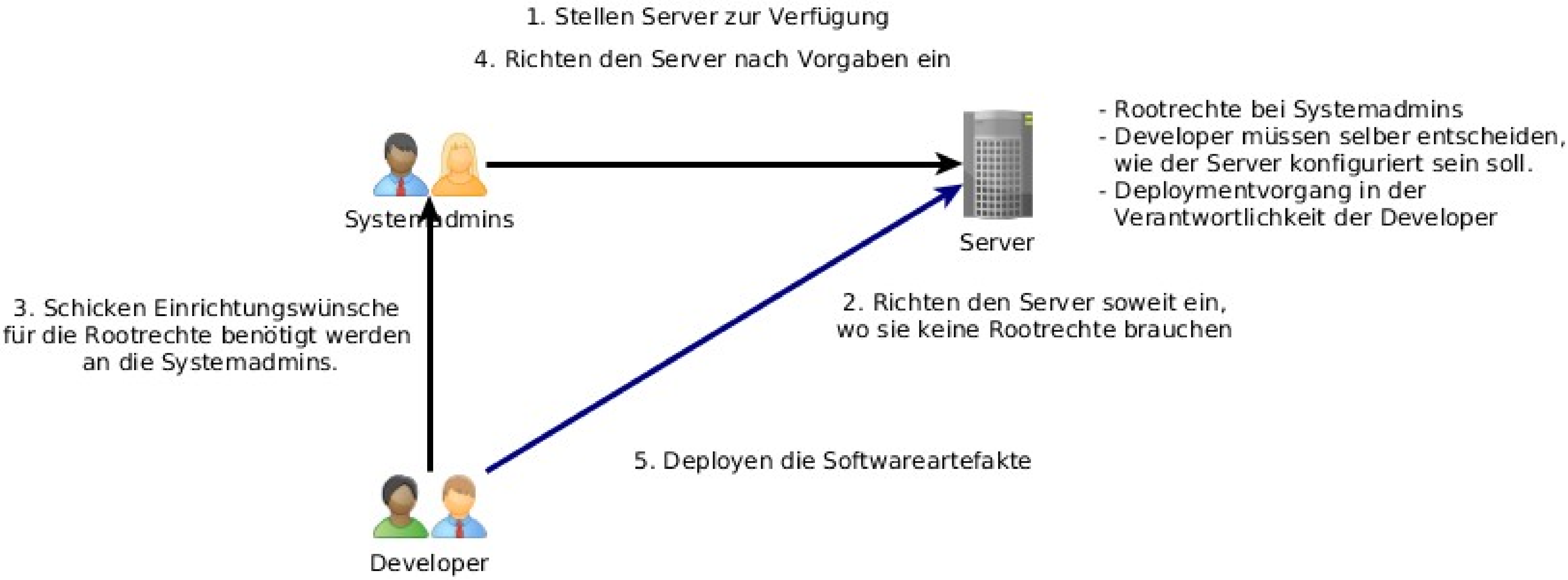
# Systemkonfiguration für Entwickler

## Organisatorische Ausgangslage
## Realität



# Systemkonfiguration für Entwickler

## Prozess zwischen Development und Operation



1. Stellen Server zur Verfügung

4. Richten den Server nach Vorgaben ein

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

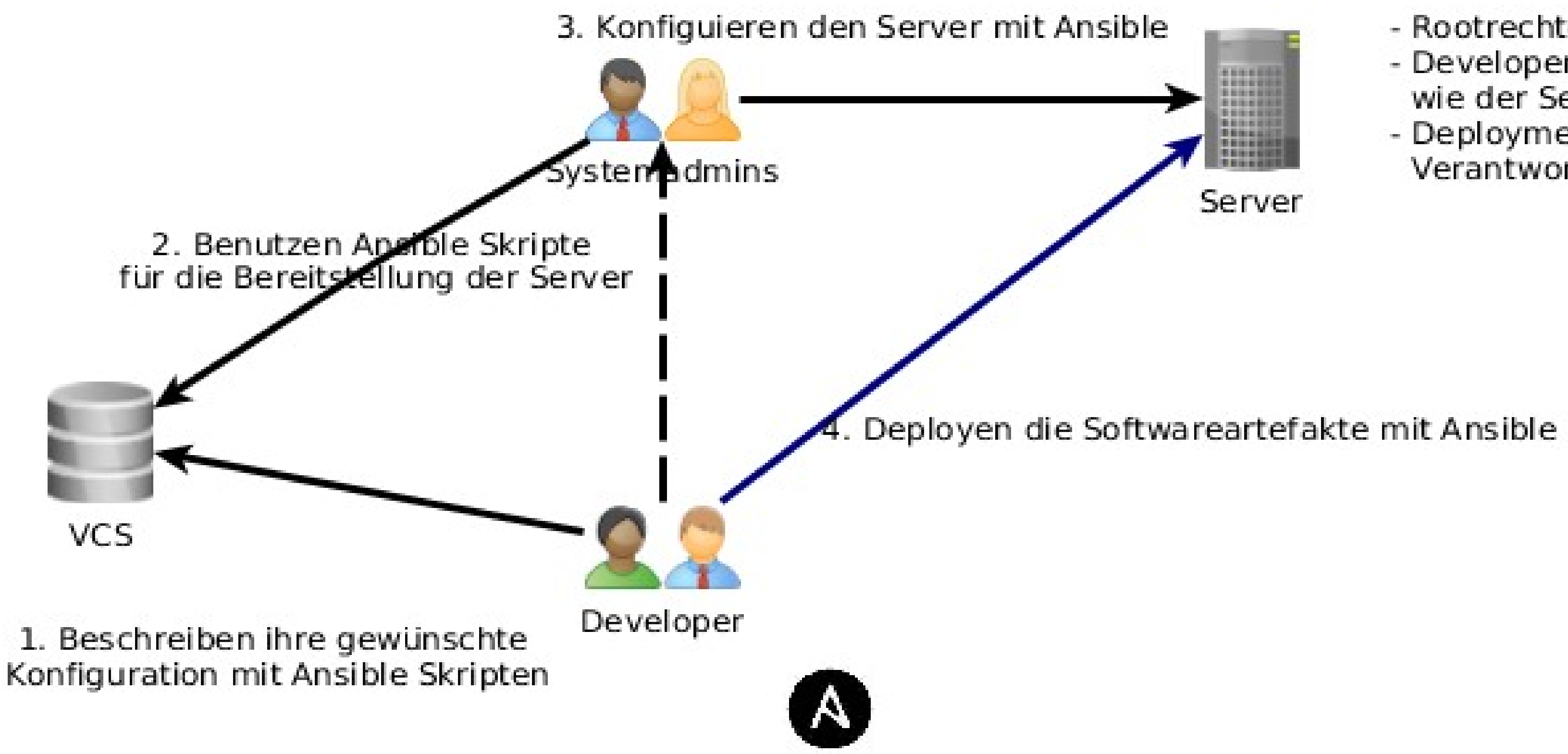
3. Schicken Einrichtungswünsche für die Rootrechte benötigt werden an die Systemadmins.

2. Richten den Server soweit ein, wo sie keine Rootrechte brauchen

5. Deployen die Softwareartefakte

# Systemkonfiguration für Entwickler

## Lösungidee mit Ansible



3. Konfiguieren den Server mit Ansible

Systemadmins

Server

2. Benutzen Ansible Skripte für die Bereitstellung der Server

VCS

4. Deployen die Softwareartefakte mit Ansible

Developer

1. Beschreiben ihre gewünschte Konfiguration mit Ansible Skripten

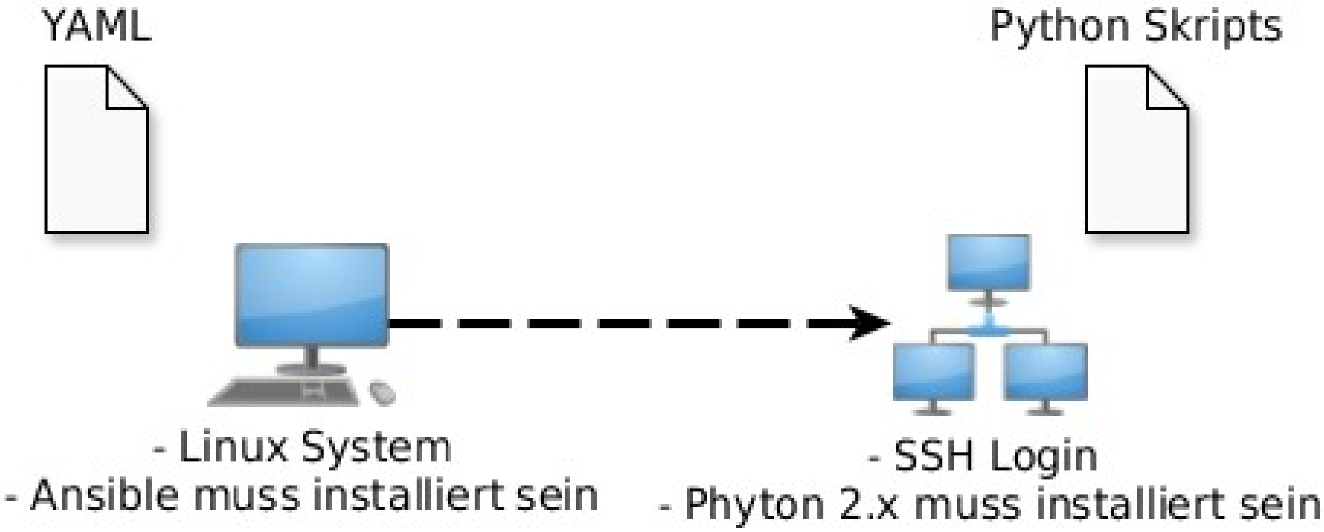
- Rootrechte bei Systemadmins
- Developer müssen selber entscheiden, wie der Server konfiguriert sein soll.
- Deploymentvorgang in der Verantwortlichkeit der Developer

# Einführung in Ansible

# Ansible

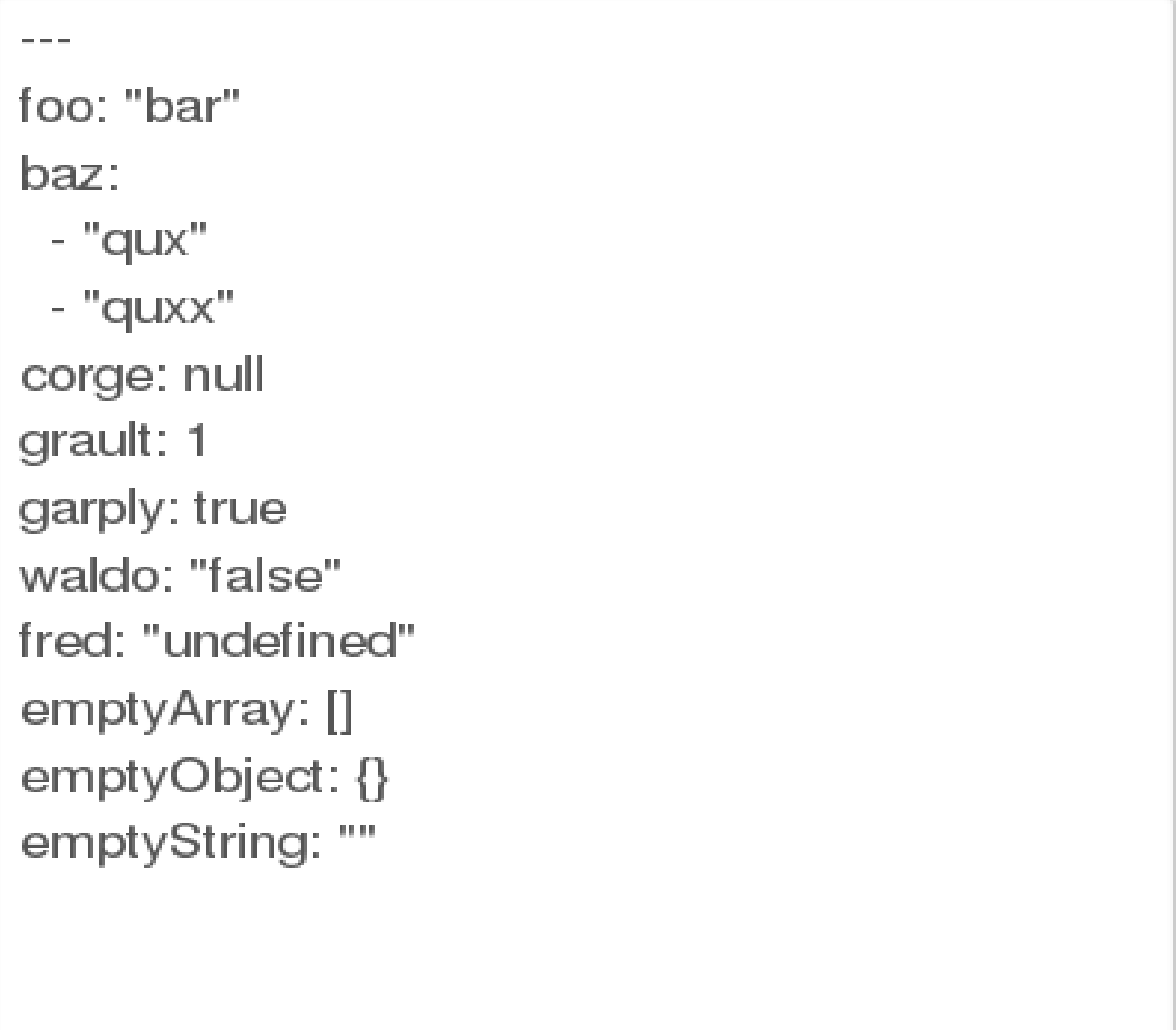
- Software für
  - Konfigurationsmanagement,
  - Softwareverteilung und
  - Ad-hoc-Kommando-Ausführung
- Sprache: Python
- Ansible Skripte: YAML

# Funktionsweise


YAML
Python Skripts
- Linux System
- Ansible muss installiert sein
- SSH Login
- Phyton 2.x muss installiert sein

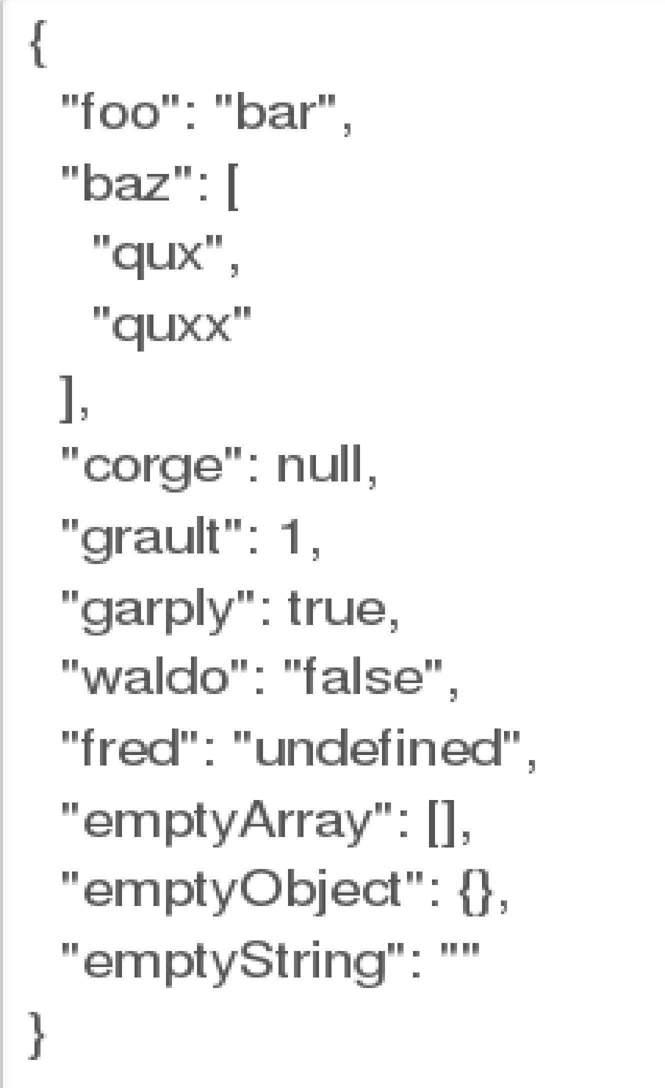
# Exkurs: YAML

**YAML**



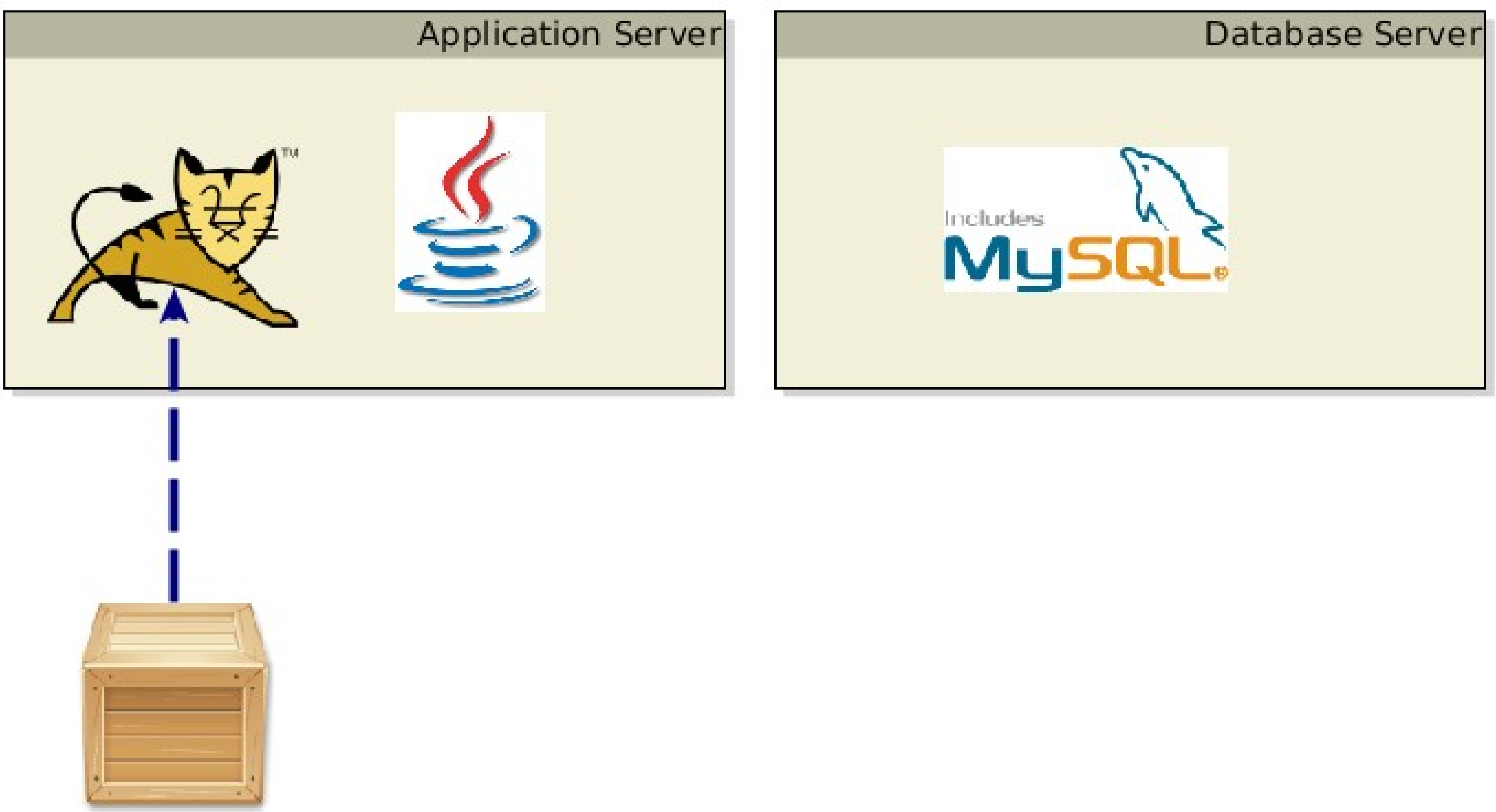
```yaml
---
foo: "bar"
baz:
  - "qux"
  - "quxx"
corge: null
grault: 1
garply: true
waldo: "false"
fred: "undefined"
emptyArray: []
emptyObject: {}
emptyString: ""
```

**JSON**

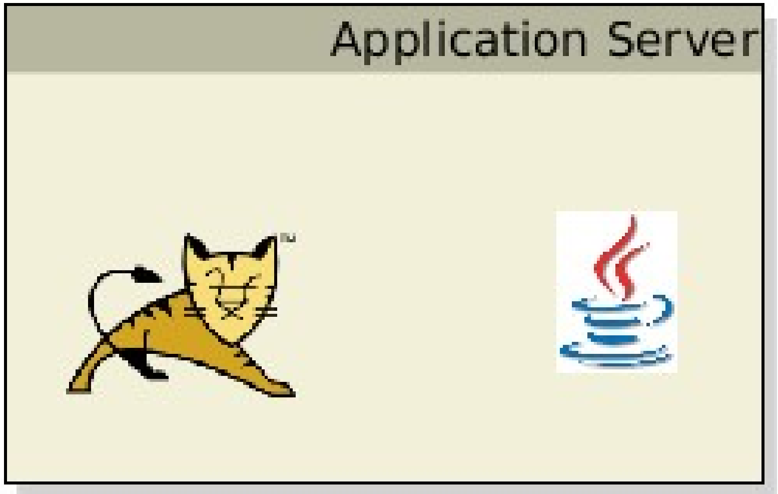


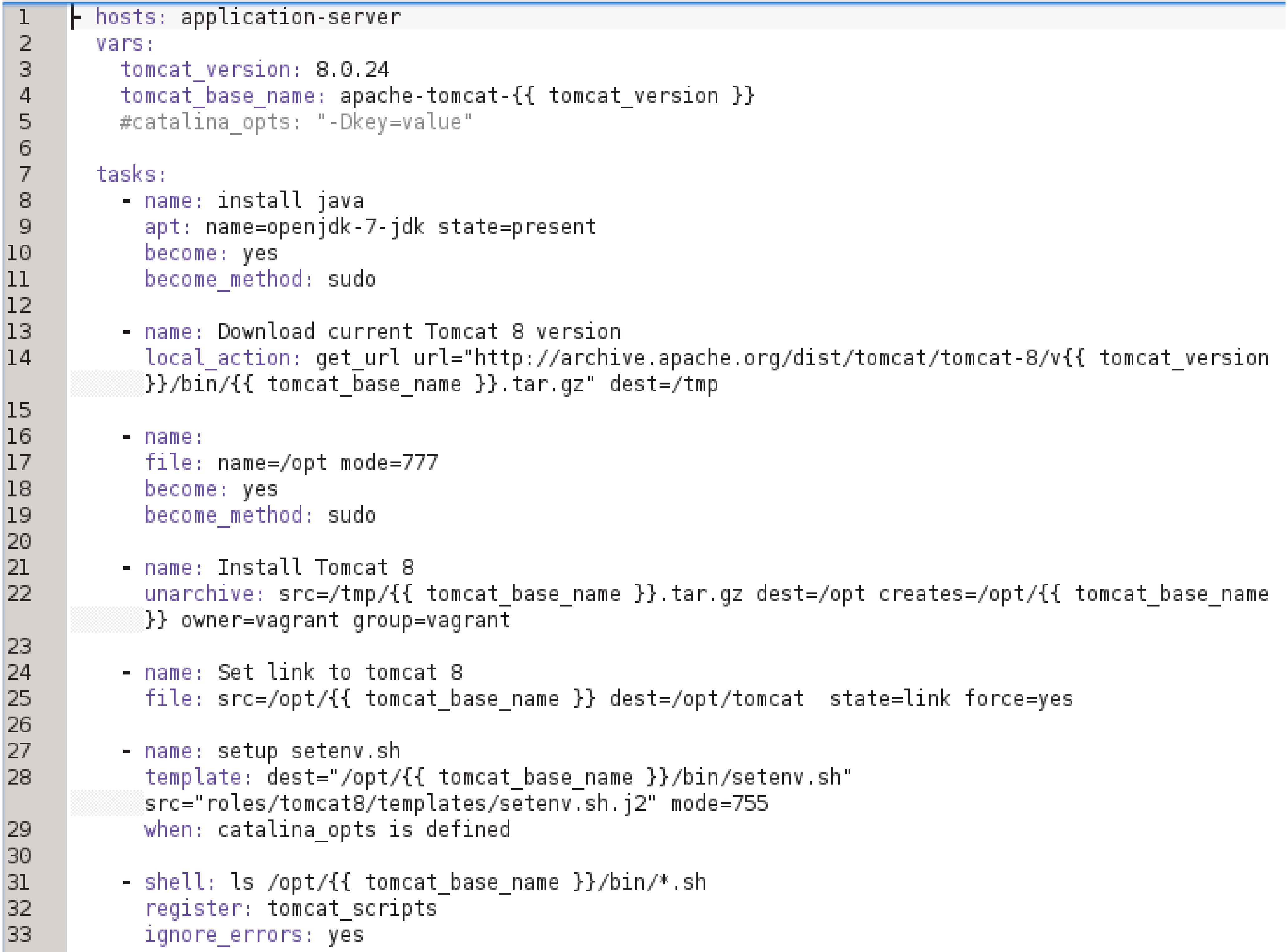
```json
{
  "foo": "bar",
  "baz": [
    "qux",
    "quxx"
  ],
  "corge": null,
  "grault": 1,
  "garply": true,
  "waldo": "false",
  "fred": "undefined",
  "emptyArray": [],
  "emptyObject": {},
  "emptyString": ""
}
```

# Ansible Beispiel



# Setup Application Server Playbook

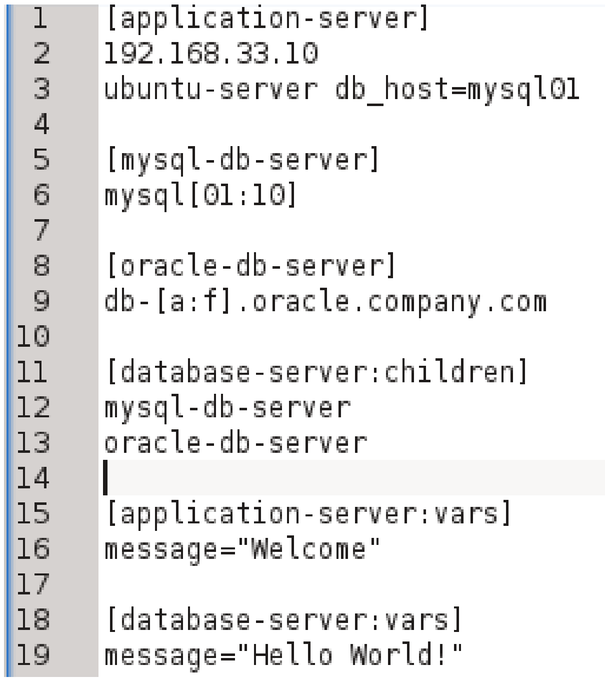




```yaml
- hosts: application-server
  vars:
    tomcat_version: 8.0.24
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      apt: name=openjdk-7-jdk state=present
      become: yes
      become_method: sudo

    - name: Download current Tomcat 8 version
      local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp

    - name:
      file: name=/opt mode=777
      become: yes
      become_method: sudo

    - name: Install Tomcat 8
      unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name }} owner=vagrant group=vagrant

    - name: Set link to tomcat 8
      file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat  state=link force=yes

    - name: setup setenv.sh
      template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh" src="roles/tomcat8/templates/setenv.sh.j2" mode=755
      when: catalina_opts is defined

    - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
      register: tomcat_scripts
      ignore_errors: yes
```

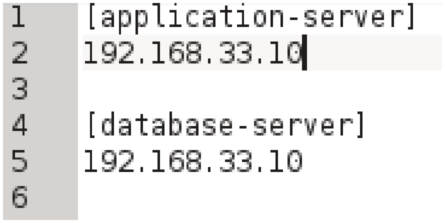
# Inventories

## Production



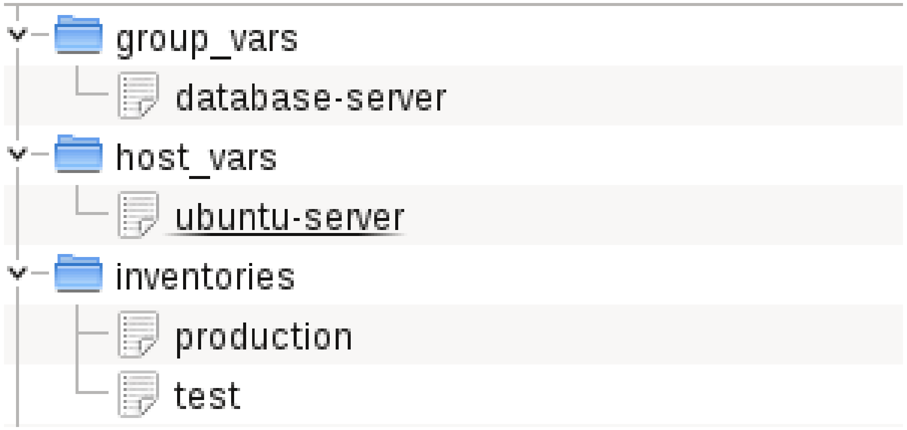
```
[application-server]
192.168.33.10
ubuntu-server db_host=mysql01

[mysql-db-server]
mysql[01:10]

[oracle-db-server]
db-[a:f].oracle.company.com

[database-server:children]
mysql-db-server
oracle-db-server

[application-server:vars]
message="Welcome"

[database-server:vars]
message="Hello World!"
```

## Test

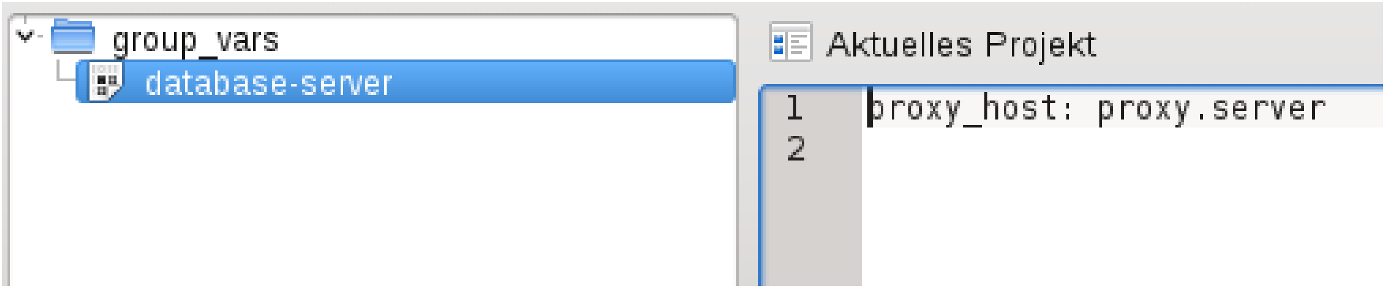


```
[application-server]
192.168.33.10

[database-server]
192.168.33.10

```

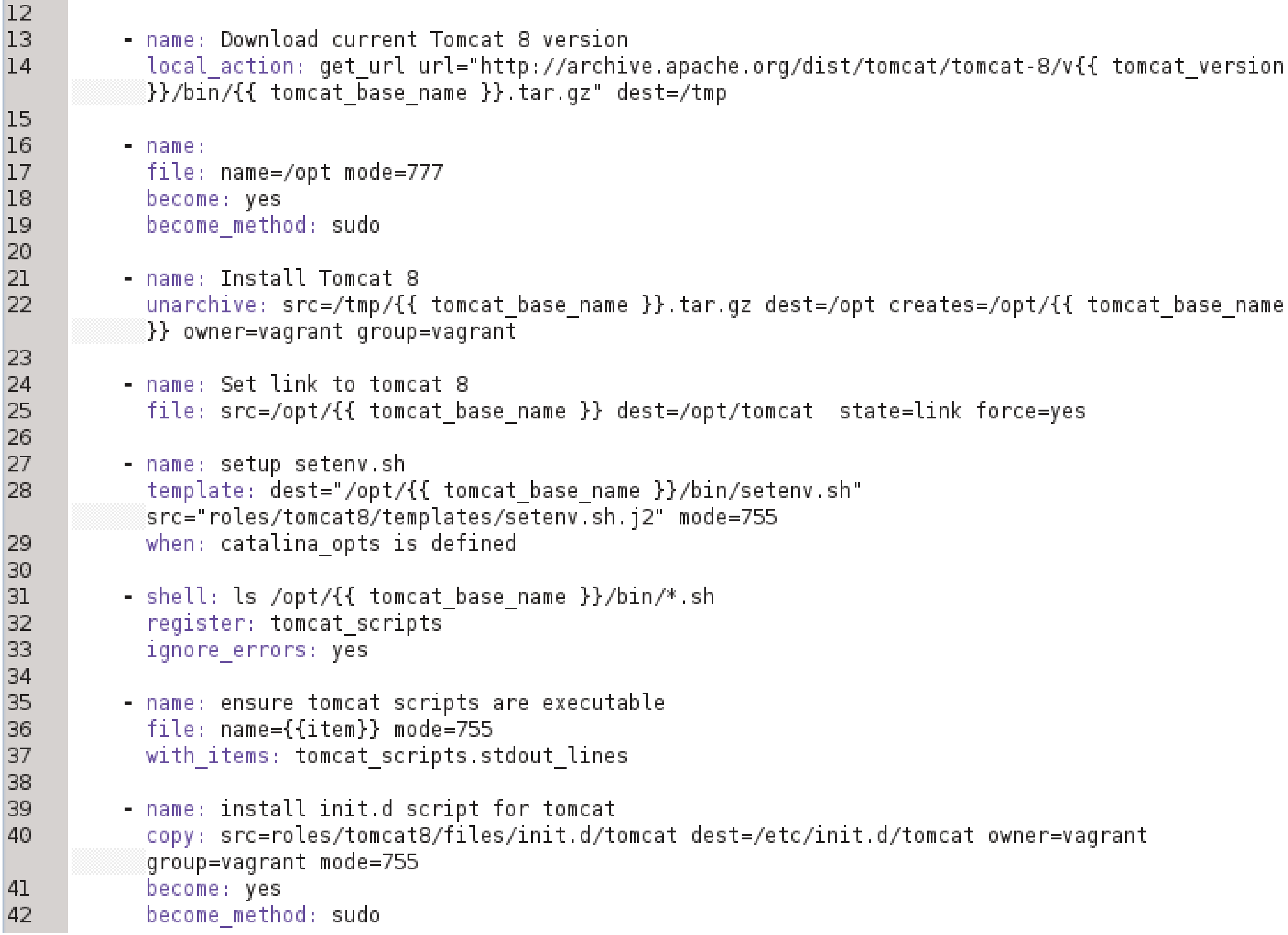
# Inventories



```
group_vars
    database-server
host_vars
    ubuntu-server
inventories
    production
    test
```



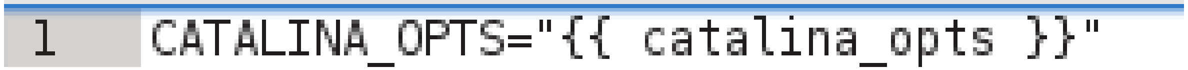
```
proxy_host: proxy.server
```



```
    - name: Download current Tomcat 8 version
      local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp

    - name:
      file: name=/opt mode=777
      become: yes
      become_method: sudo

    - name: Install Tomcat 8
      unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name }} owner=vagrant group=vagrant

    - name: Set link to tomcat 8
      file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat  state=link force=yes

    - name: setup setenv.sh
      template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh" src="roles/tomcat8/templates/setenv.sh.j2" mode=755
      when: catalina_opts is defined

    - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
      register: tomcat_scripts
      ignore_errors: yes

    - name: ensure tomcat scripts are executable
      file: name={{item}} mode=755
      with_items: tomcat_scripts.stdout_lines

    - name: install init.d script for tomcat
      copy: src=roles/tomcat8/files/init.d/tomcat dest=/etc/init.d/tomcat owner=vagrant group=vagrant mode=755
      become: yes
      become_method: sudo
```

# Templates

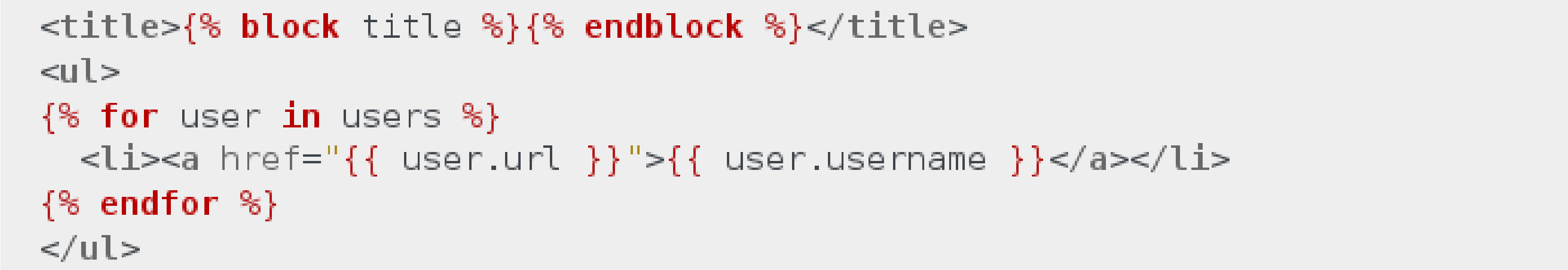
- setenv.sh.j2



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

# Templates - Jinja2

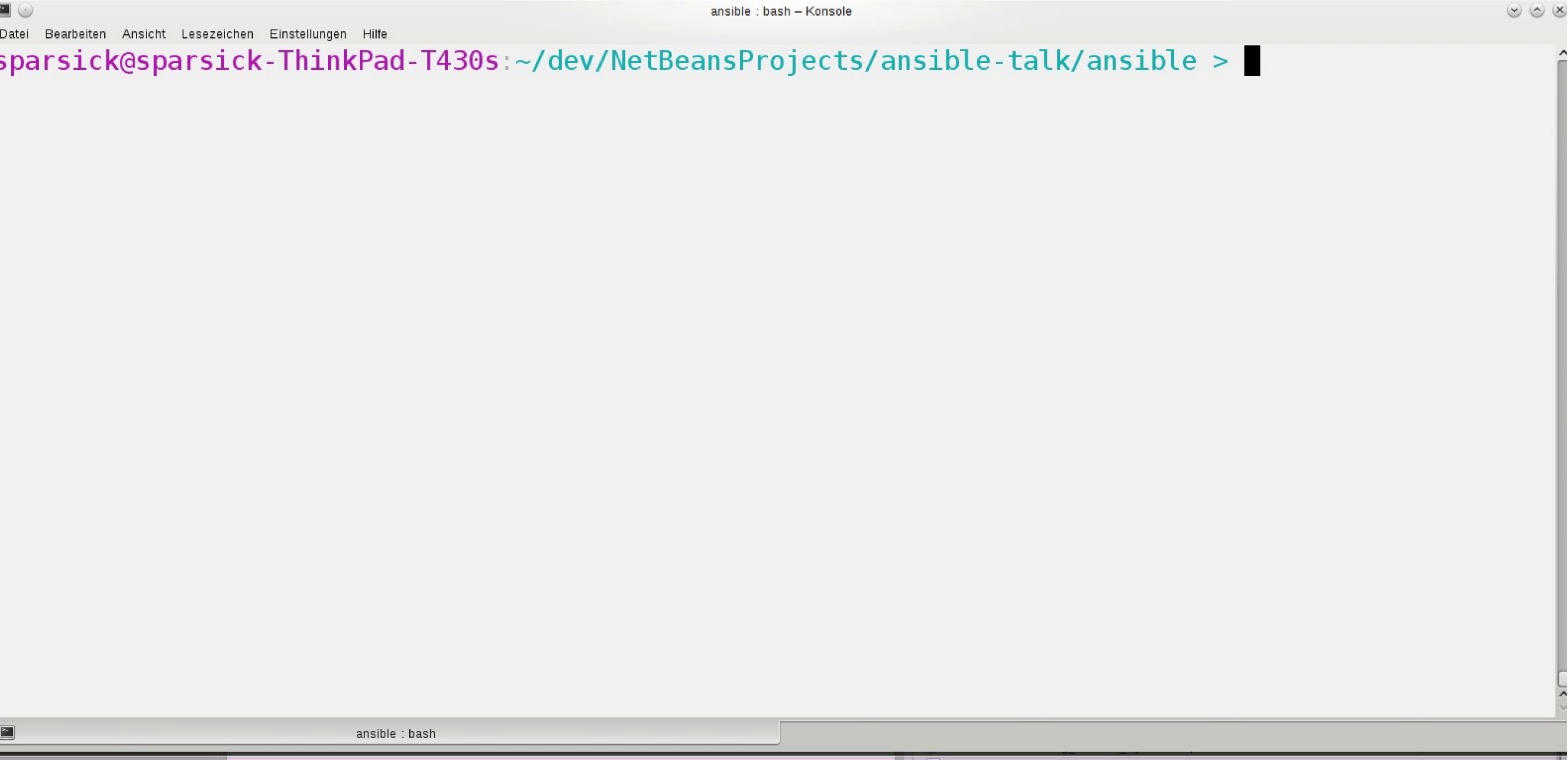
- Templating engine für Python



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

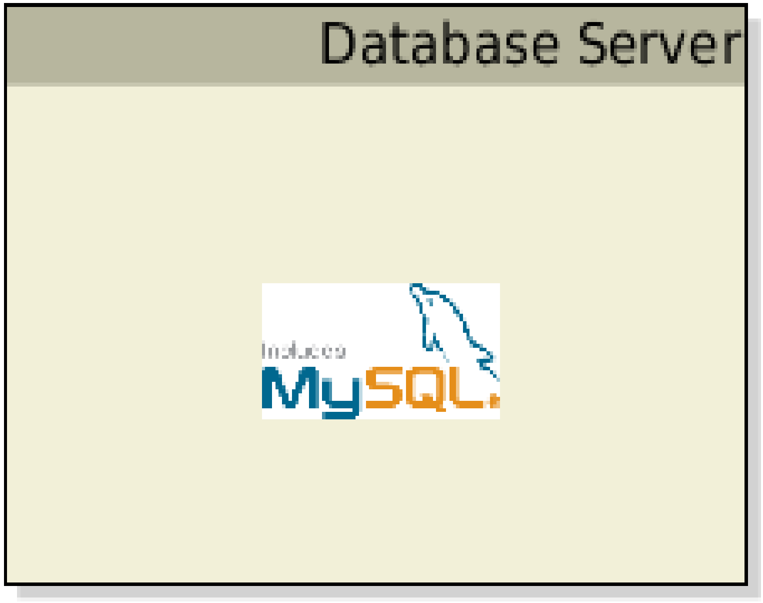
- Mehr Information unter http://jinja.pocoo.org/docs/dev/

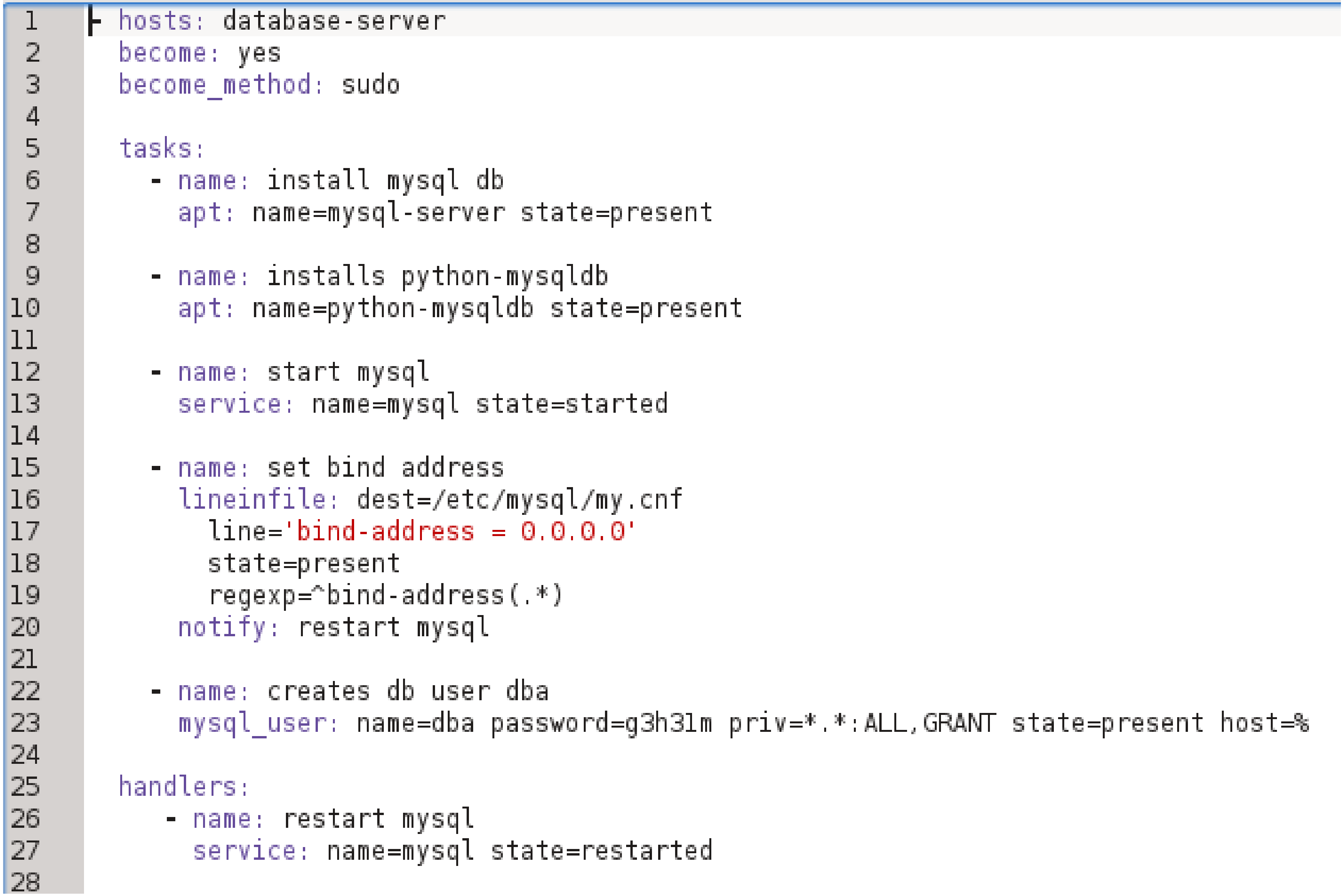
# Setup Application Server Playbook



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

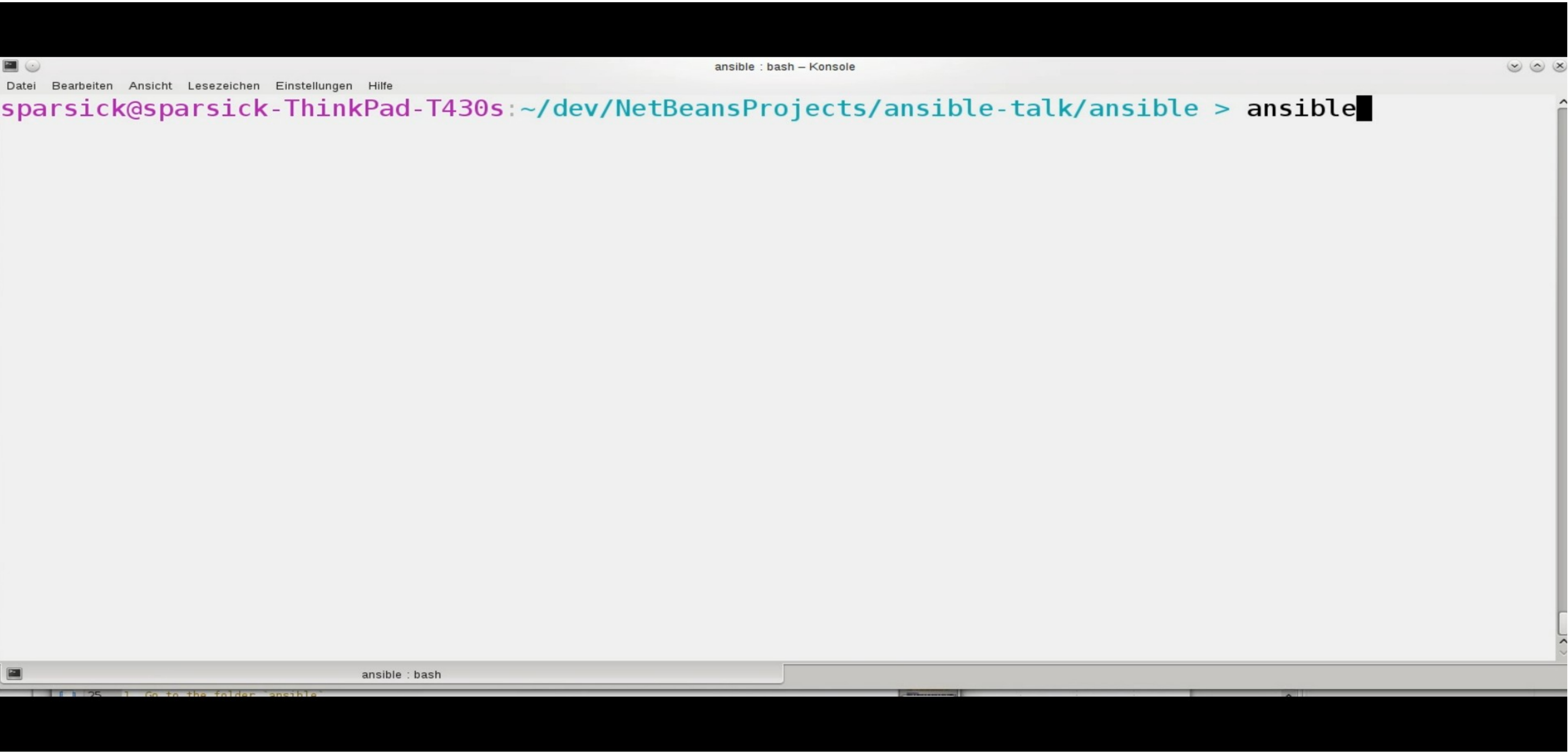
# Setup Database Server Playbook





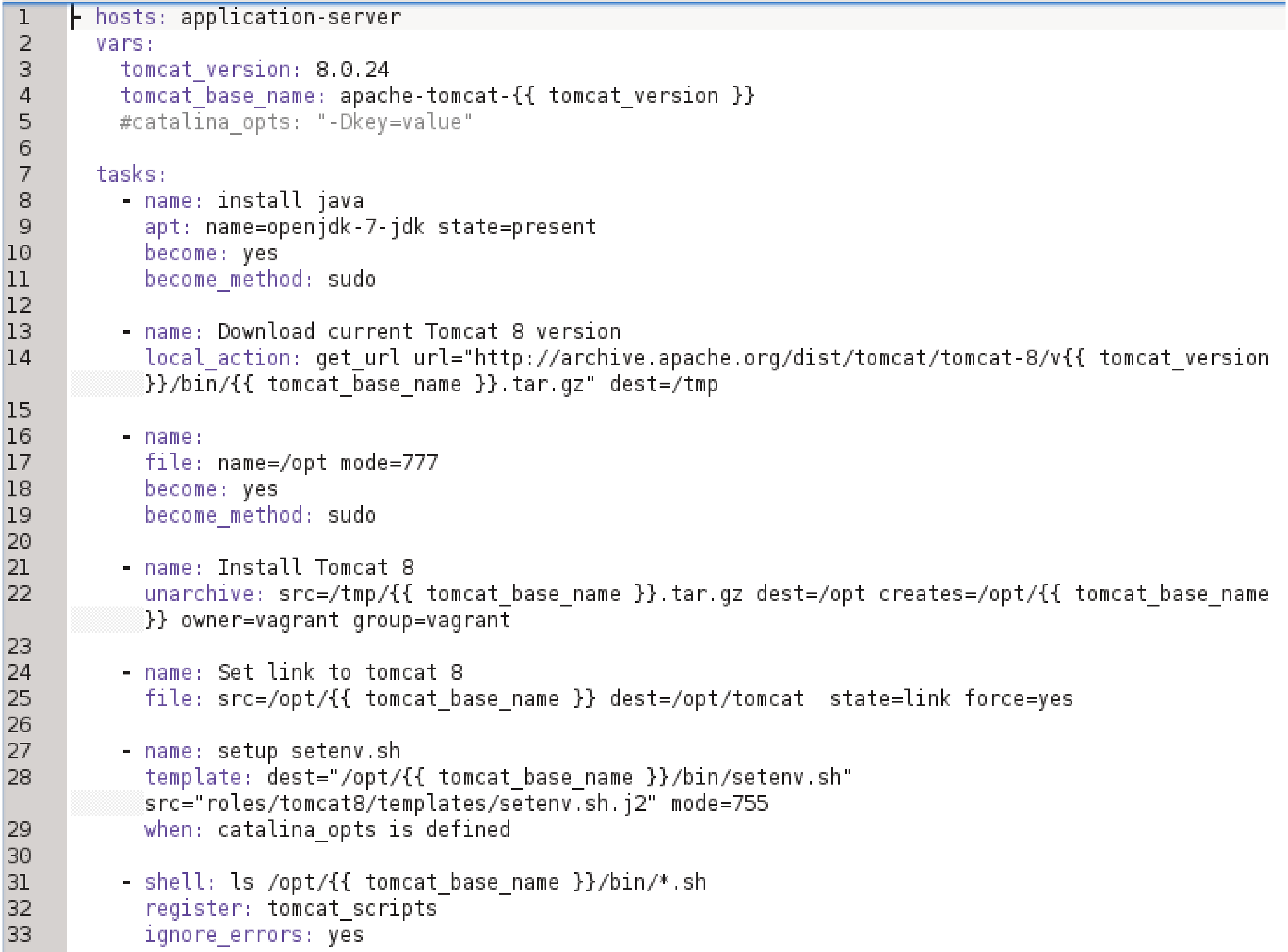
```
- hosts: database-server
  become: yes
  become_method: sudo

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

    - name: installs python-mysqldb
      apt: name=python-mysqldb state=present

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

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

    - name: creates db user dba
      mysql_user: name=dba password=g3h31m priv=*.*:ALL,GRANT state=present host=%

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

```

# Setup Database Server Playbook

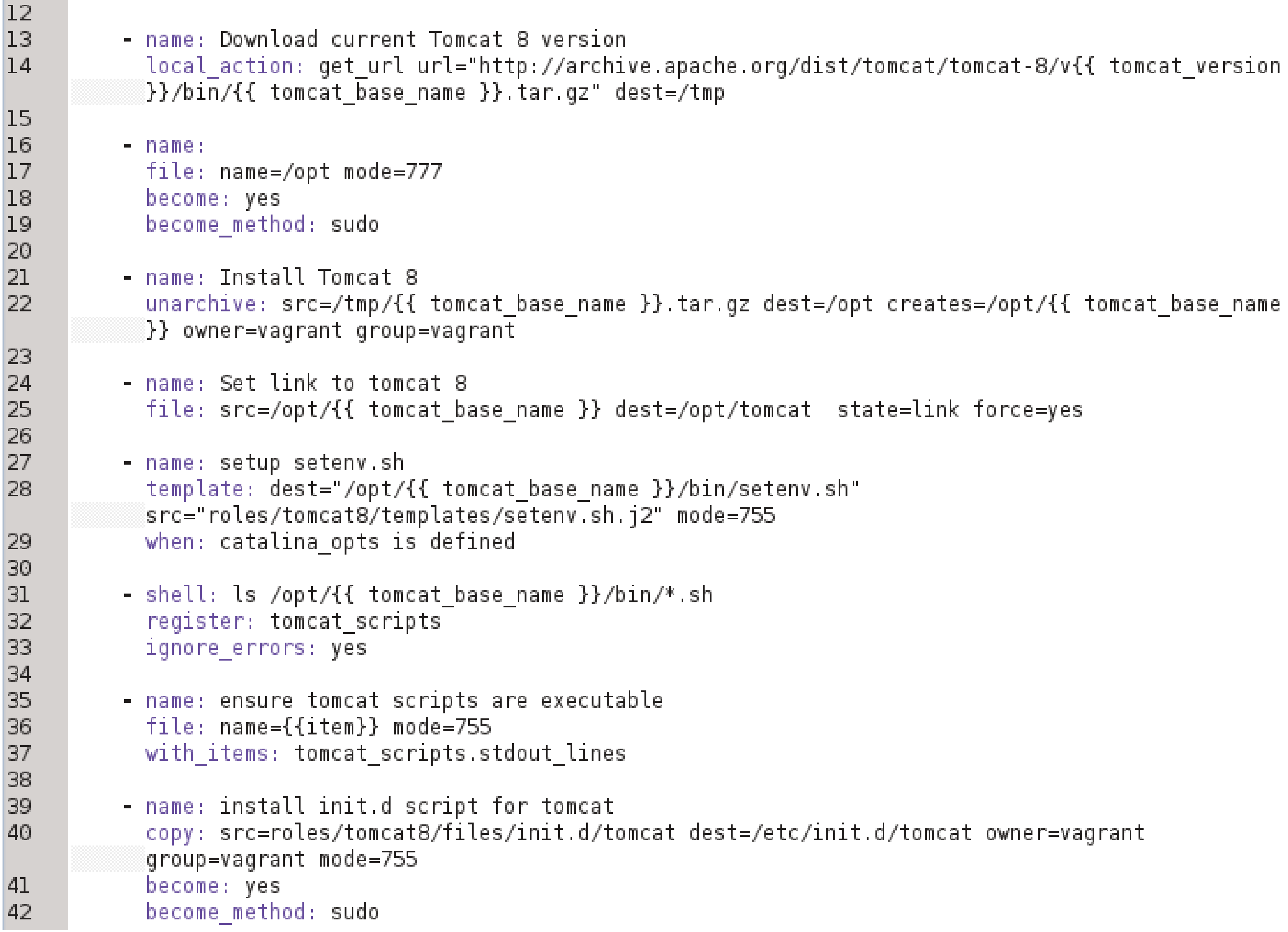


```
sparsick@sparsick-ThinkPad-T430s:~/dev/NetBeansProjects/ansible-talk/ansible > ansible
```

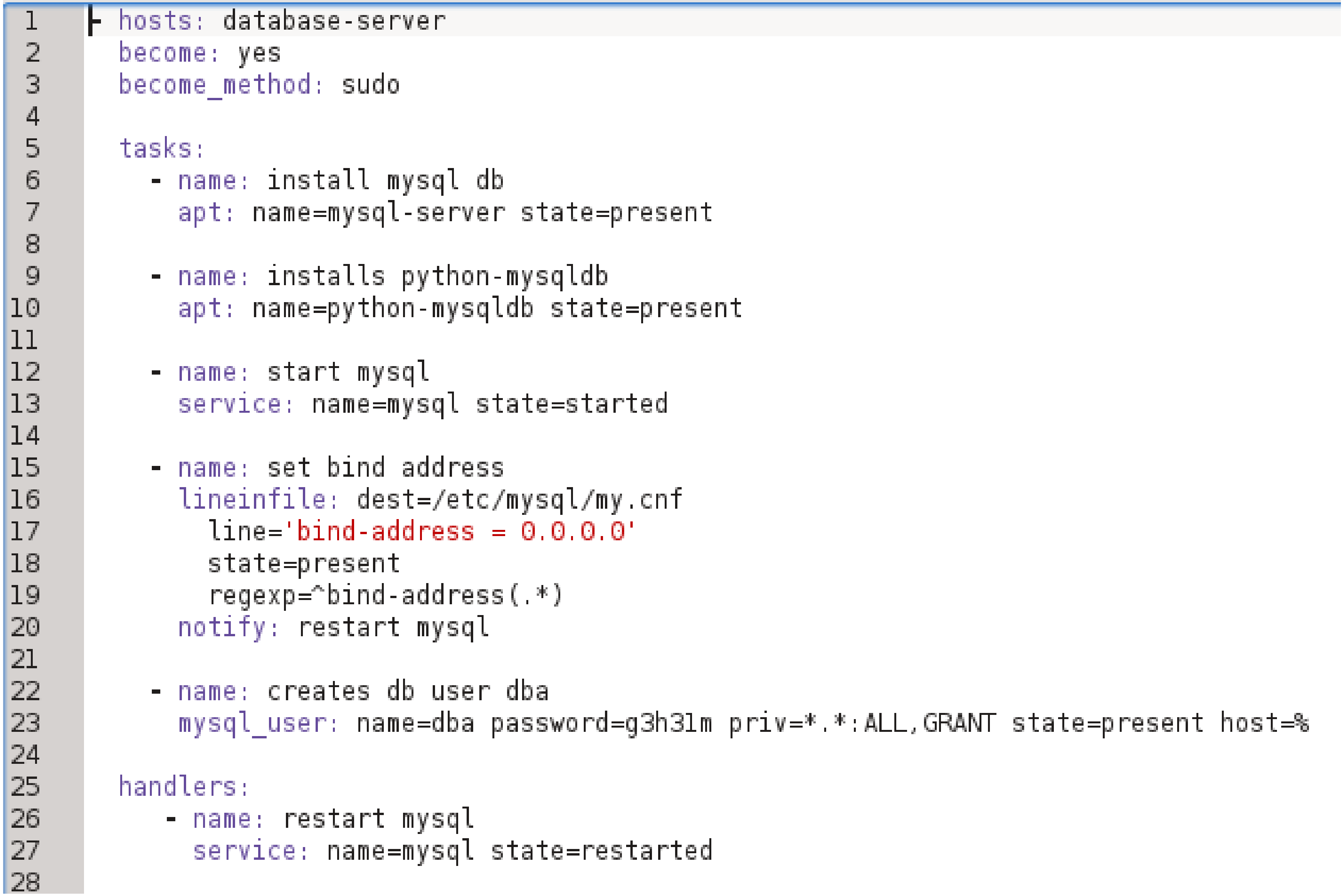
# Setup Database Server Playbook



```yaml
- hosts: application-server
  vars:
    tomcat_version: 8.0.24
    tomcat_base_name: apache-tomcat-{{ tomcat_version }}
    #catalina_opts: "-Dkey=value"

  tasks:
    - name: install java
      apt: name=openjdk-7-jdk state=present
      become: yes
      become_method: sudo

    - name: Download current Tomcat 8 version
      local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp

    - name:
      file: name=/opt mode=777
      become: yes
      become_method: sudo

    - name: Install Tomcat 8
      unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name }} owner=vagrant group=vagrant

    - name: Set link to tomcat 8
      file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat  state=link force=yes

    - name: setup setenv.sh
      template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh" src="roles/tomcat8/templates/setenv.sh.j2" mode=755
      when: catalina_opts is defined

    - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
      register: tomcat_scripts
      ignore_errors: yes
```

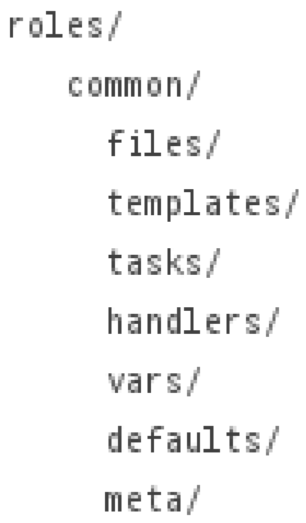


```
    - name: Download current Tomcat 8 version
      local_action: get_url url="http://archive.apache.org/dist/tomcat/tomcat-8/v{{ tomcat_version }}/bin/{{ tomcat_base_name }}.tar.gz" dest=/tmp

    - name:
      file: name=/opt mode=777
      become: yes
      become_method: sudo

    - name: Install Tomcat 8
      unarchive: src=/tmp/{{ tomcat_base_name }}.tar.gz dest=/opt creates=/opt/{{ tomcat_base_name }} owner=vagrant group=vagrant

    - name: Set link to tomcat 8
      file: src=/opt/{{ tomcat_base_name }} dest=/opt/tomcat  state=link force=yes

    - name: setup setenv.sh
      template: dest="/opt/{{ tomcat_base_name }}/bin/setenv.sh" src="roles/tomcat8/templates/setenv.sh.j2" mode=755
      when: catalina_opts is defined

    - shell: ls /opt/{{ tomcat_base_name }}/bin/*.sh
      register: tomcat_scripts
      ignore_errors: yes

    - name: ensure tomcat scripts are executable
      file: name={{item}} mode=755
      with_items: tomcat_scripts.stdout_lines

    - name: install init.d script for tomcat
      copy: src=roles/tomcat8/files/init.d/tomcat dest=/etc/init.d/tomcat owner=vagrant group=vagrant mode=755
      become: yes
      become_method: sudo
```

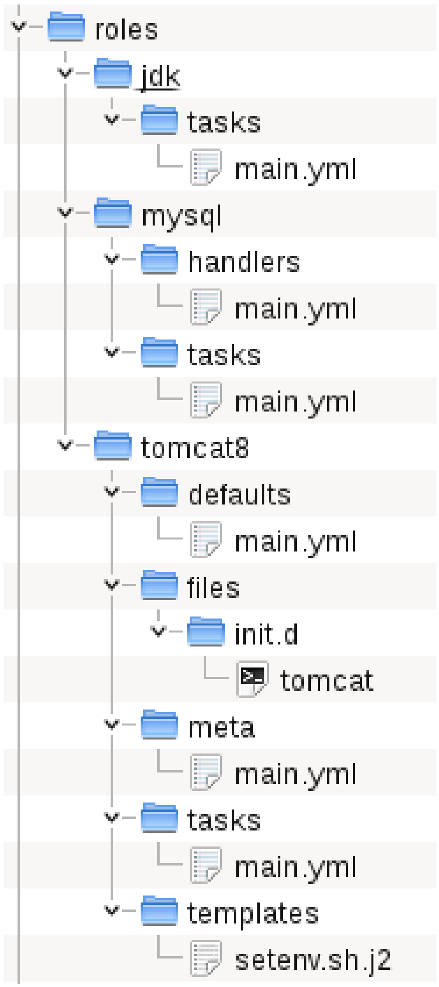


```
- hosts: database-server
  become: yes
  become_method: sudo

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

    - name: installs python-mysqldb
      apt: name=python-mysqldb state=present

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

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

    - name: creates db user dba
      mysql_user: name=dba password=g3h31m priv=*.*:ALL,GRANT state=present host=%

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

```

# Roles



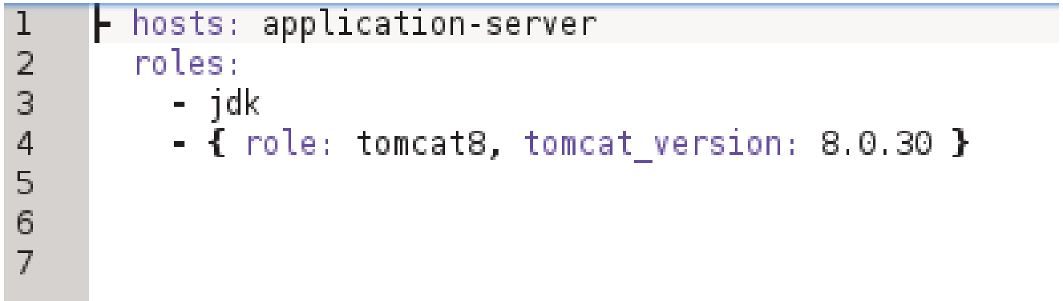
```
roles/
   common/
     files/
     templates/
     tasks/
     handlers/
     vars/
     defaults/
     meta/
```



```
roles
  jdk
    tasks
      main.yml
  mysql
    handlers
      main.yml
    tasks
      main.yml
  tomcat8
    defaults
      main.yml
    files
      init.d
        tomcat
    meta
      main.yml
    tasks
      main.yml
    templates
      setenv.sh.j2
```

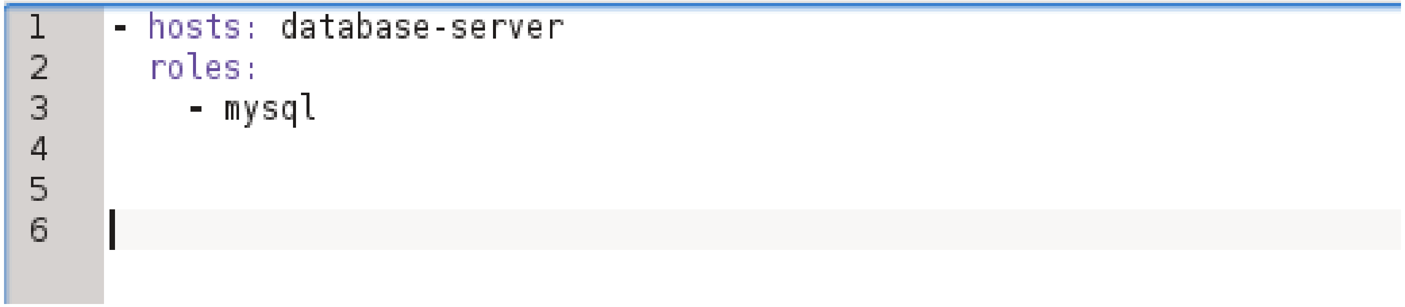
# Setup Playbooks mit Roles

- Setup Application Server



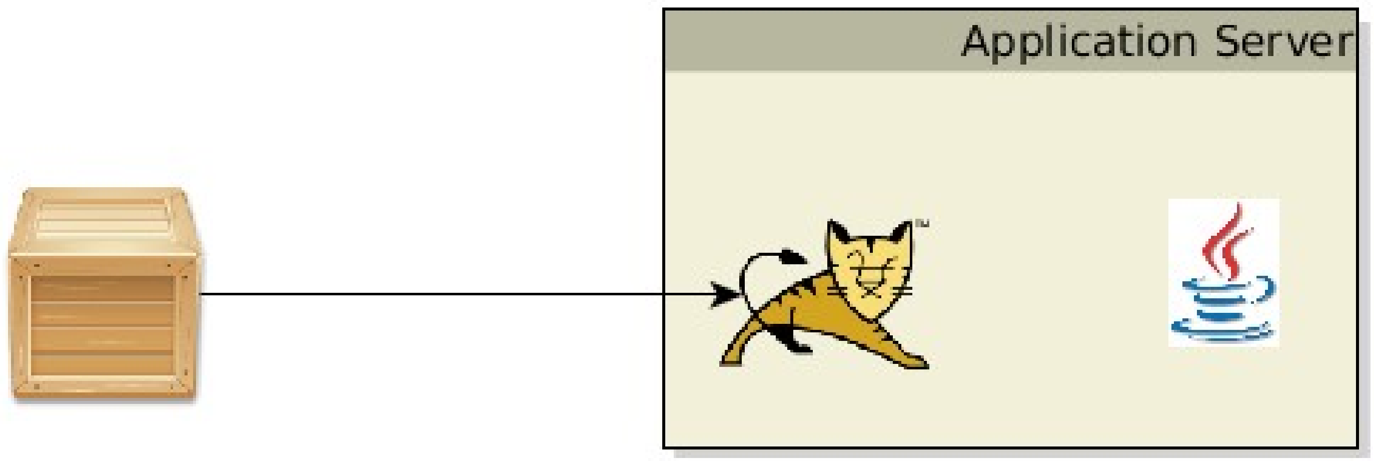
```yaml
- hosts: application-server
  roles:
    - jdk
    - { role: tomcat8, tomcat_version: 8.0.30 }
```

- Setup Database Server

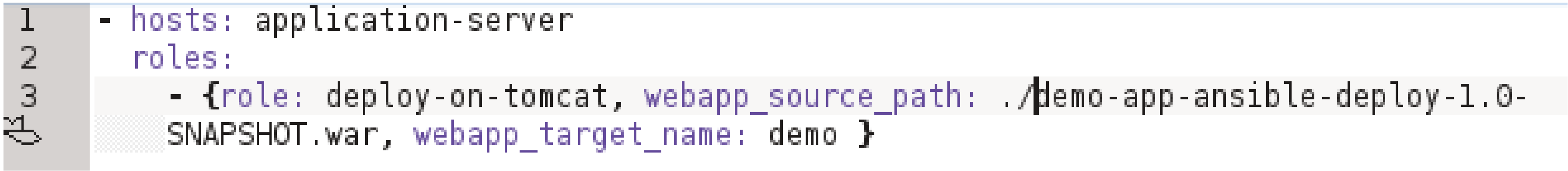


```yaml
- hosts: database-server
  roles:
    - mysql
```

# Java Webapplikation Deployment

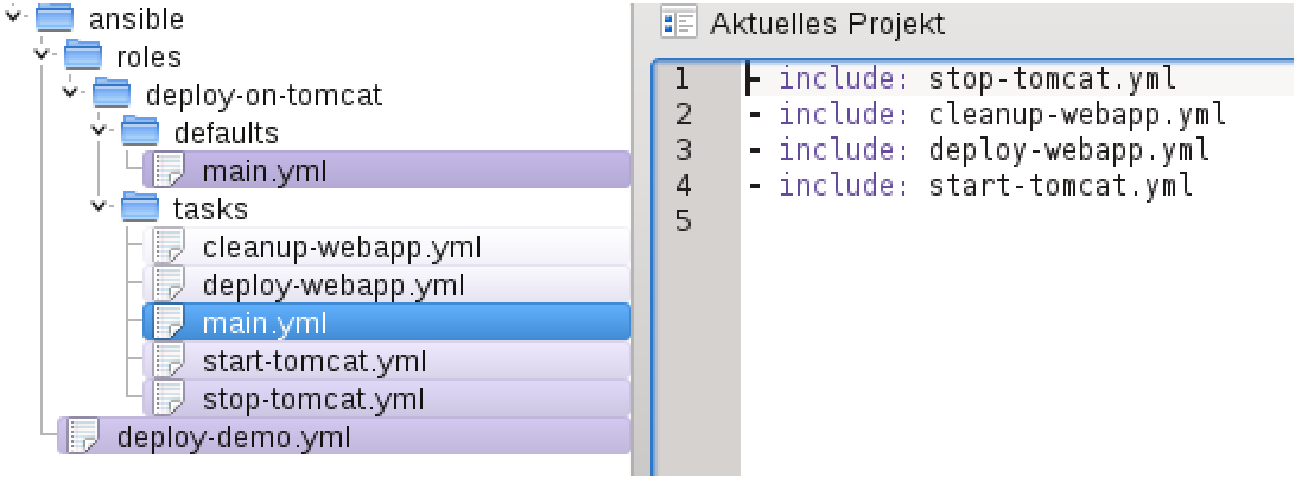


# Deploy Application Playbook



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

# deploy-on-tomcat Role

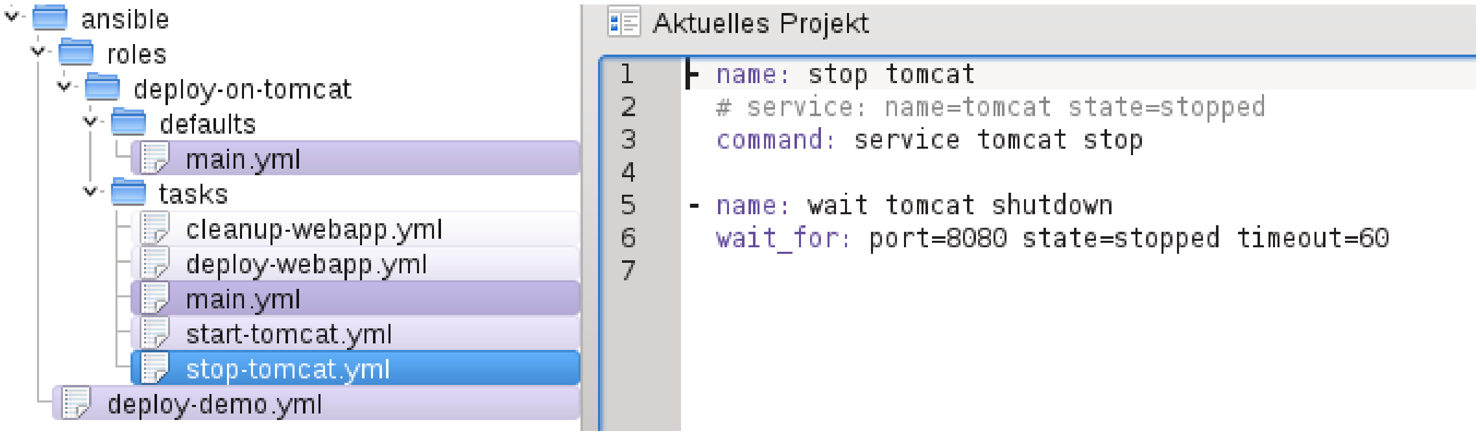


```
ansible
  roles
    deploy-on-tomcat
      defaults
        main.yml
      tasks
        cleanup-webapp.yml
        deploy-webapp.yml
        main.yml
        start-tomcat.yml
        stop-tomcat.yml
  deploy-demo.yml
```

Aktuelles Projekt

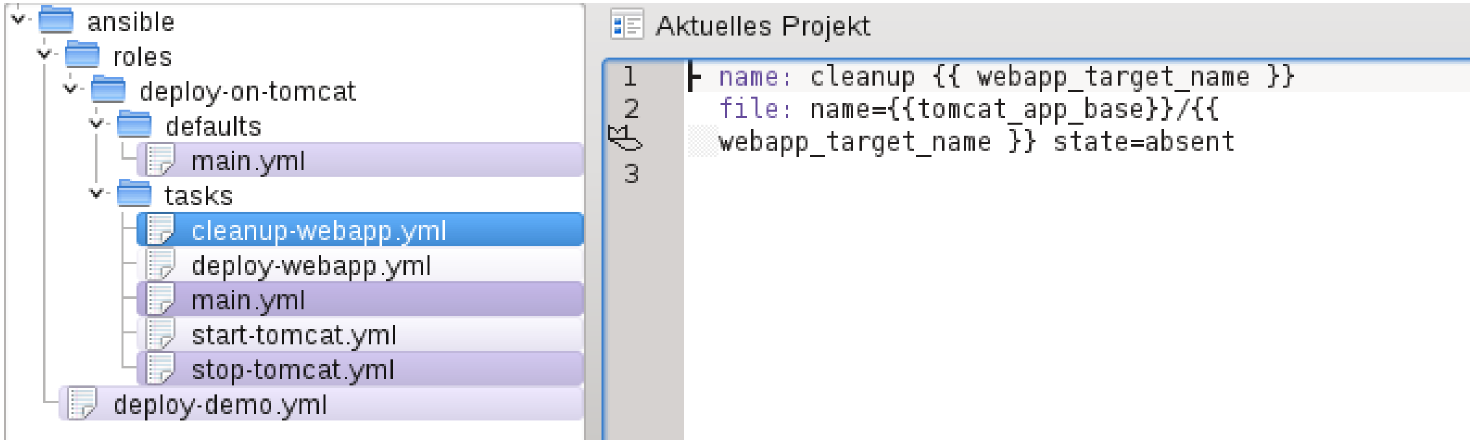
```
- include: stop-tomcat.yml
- include: cleanup-webapp.yml
- include: deploy-webapp.yml
- include: start-tomcat.yml

```

# deploy-on-tomcat Role



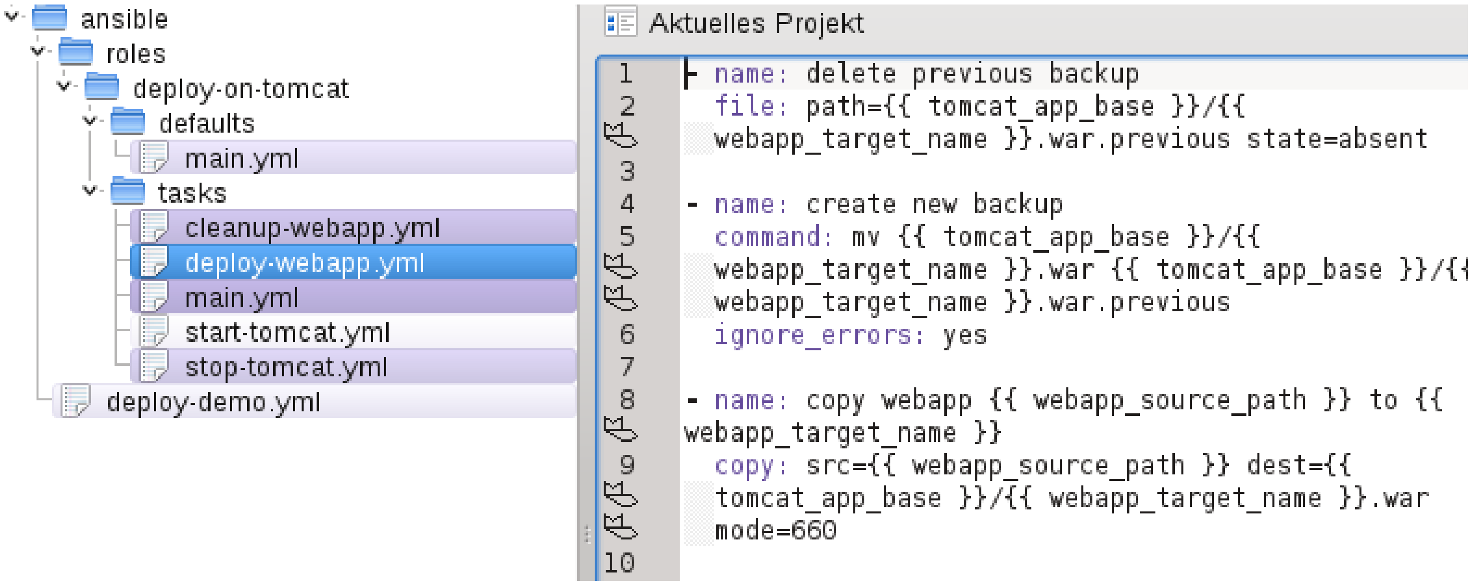
```yaml
- name: stop tomcat
  # service: name=tomcat state=stopped
  command: service tomcat stop

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

# deploy-on-tomcat Role



```yaml
- name: cleanup {{ webapp_target_name }}
  file: name={{tomcat_app_base}}/{{ webapp_target_name }} state=absent
```

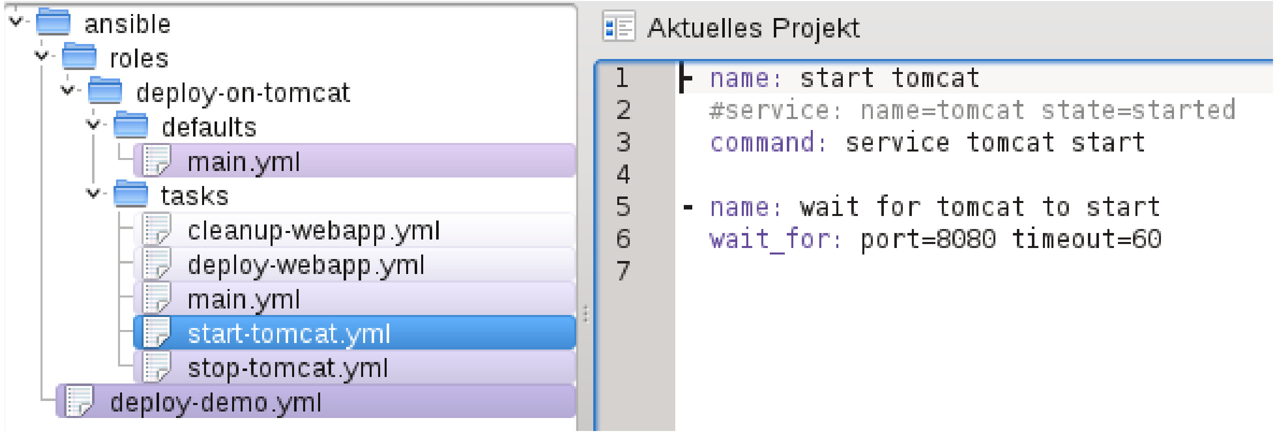
# deploy-on-tomcat Role



```
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
```

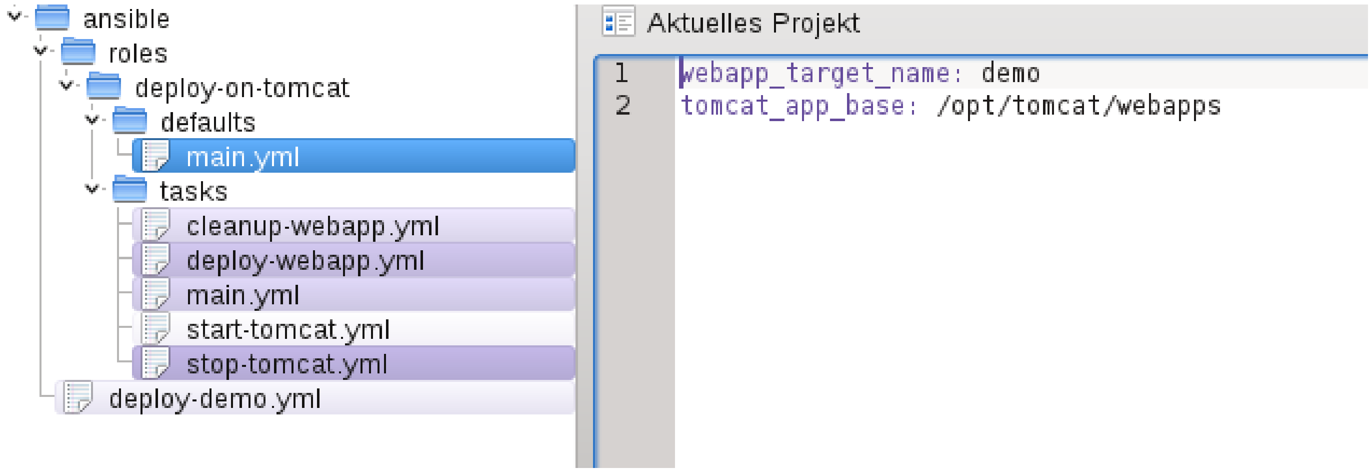
```
- name: delete previous backup
  file: path={{ tomcat_app_base }}/{{ webapp_target_name }}.war.previous state=absent

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

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

# deploy-on-tomcat Role



```yaml
- name: start tomcat
  #service: name=tomcat state=started
  command: service tomcat start

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

# deploy-on-tomcat Role

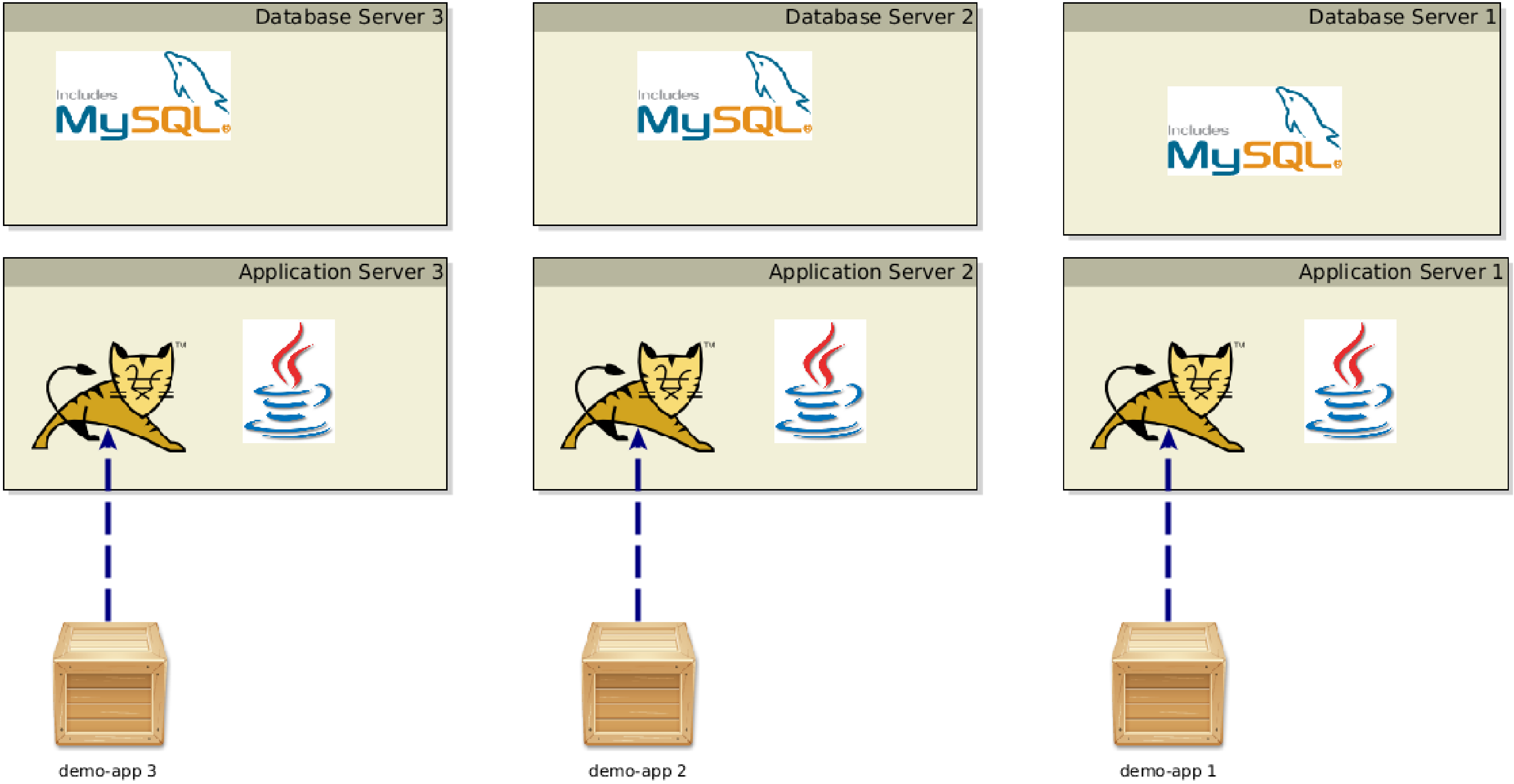


```
webapp_target_name: demo
tomcat_app_base: /opt/tomcat/webapps
```

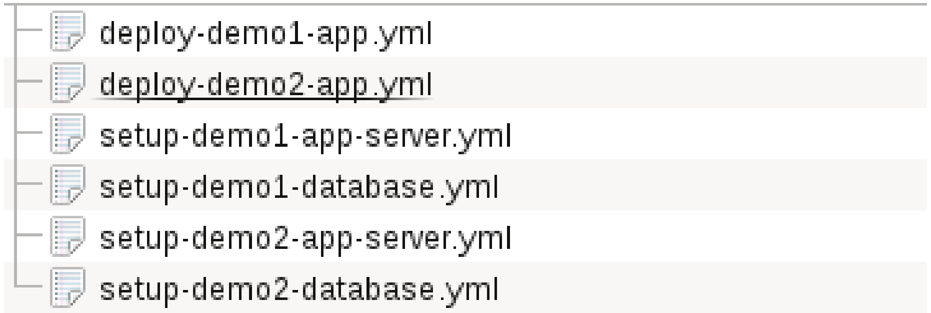
# Deploy Application Playbook

# Ad-hoc-Kommando

# Warum Roles?

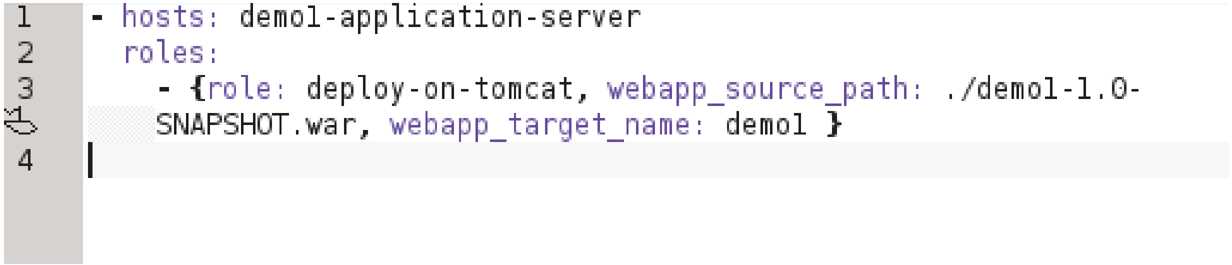

Database Server 3
Database Server 2
Database Server 1
Includes MySQL
Includes MySQL
Includes MySQL
Application Server 3
Application Server 2
Application Server 1
demo-app 3
demo-app 2
demo-app 1

# Warum Roles?

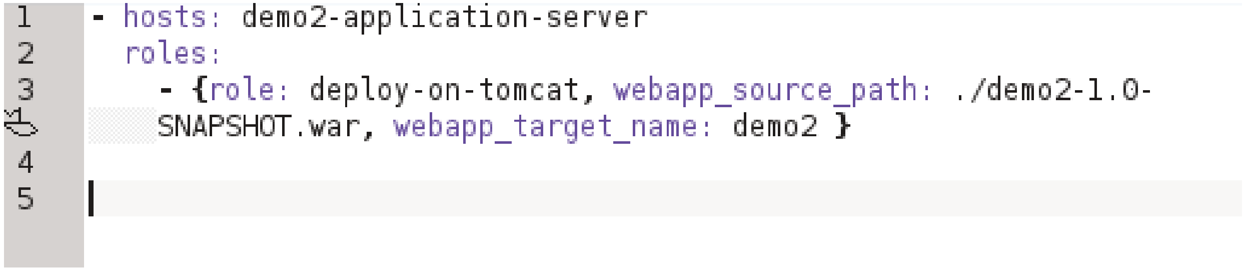


```
deploy-demo1-app.yml
deploy-demo2-app.yml
setup-demo1-app-server.yml
setup-demo1-database.yml
setup-demo2-app-server.yml
setup-demo2-database.yml
```

# Warum Roles?

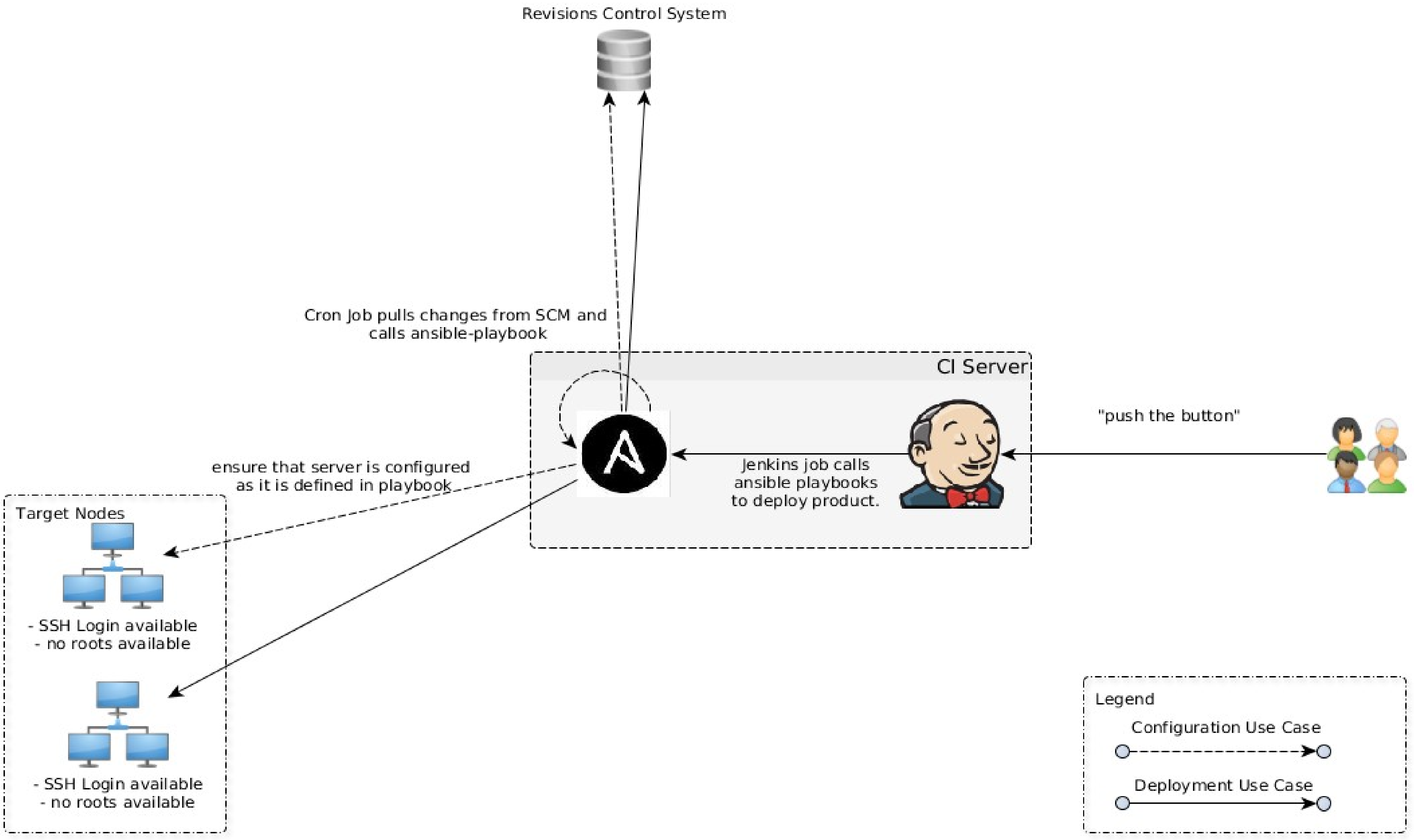


```
- hosts: demo1-application-server
  roles:
    - {role: deploy-on-tomcat, webapp_source_path: ./demo1-1.0-SNAPSHOT.war, webapp_target_name: demo1 }
```

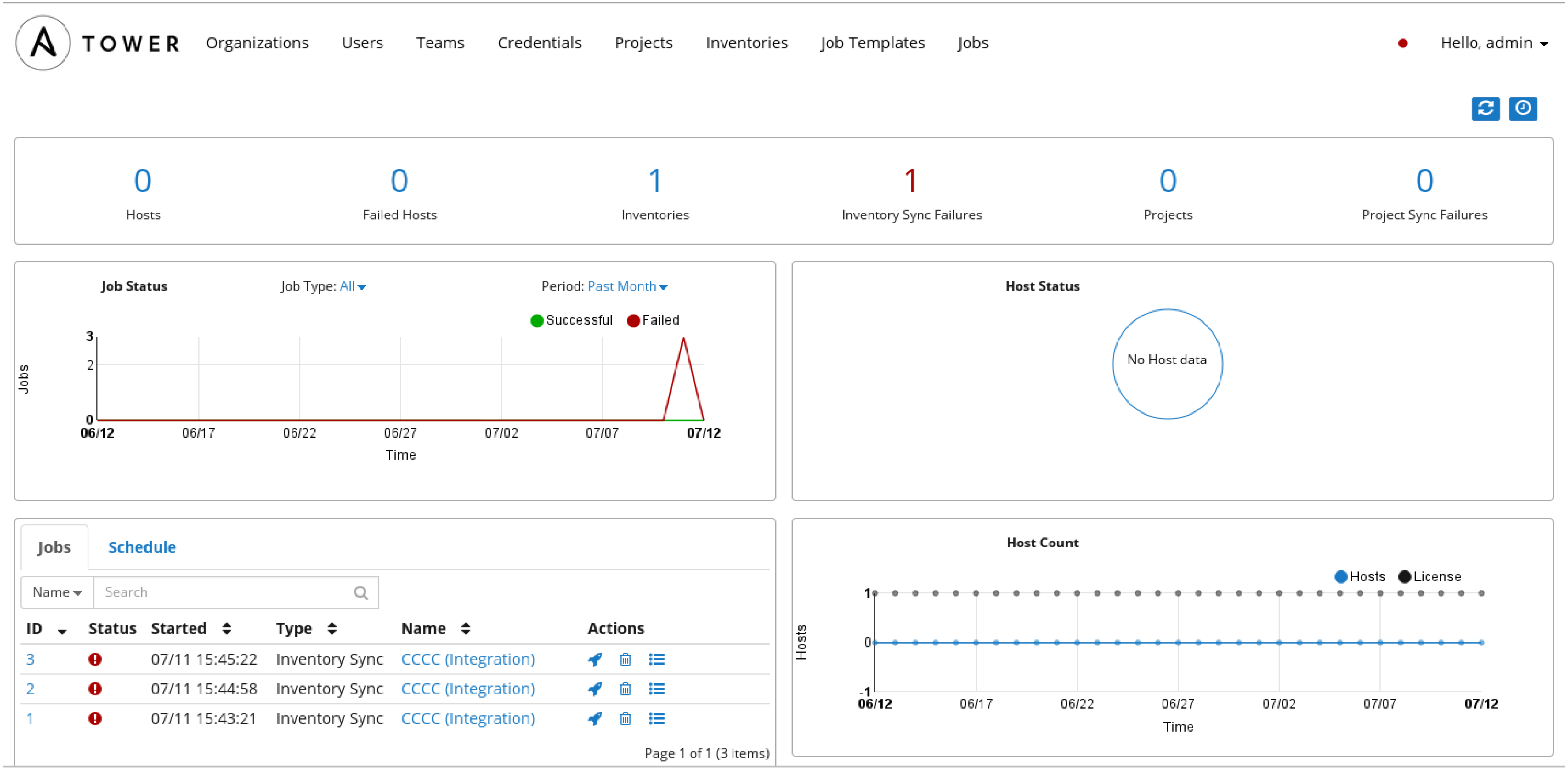


```
- hosts: demo2-application-server
  roles:
    - {role: deploy-on-tomcat, webapp_source_path: ./demo2-1.0-SNAPSHOT.war, webapp_target_name: demo2 }

```

# Ansible Infrastruktur



# Ansible Tower



| 0 | 0 | 1 | 1 | 0 | 0 |
|---|---|---|---|---|---|
| Hosts | Failed Hosts | Inventories | Inventory Sync Failures | Projects | Project Sync Failures |

| 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) | |

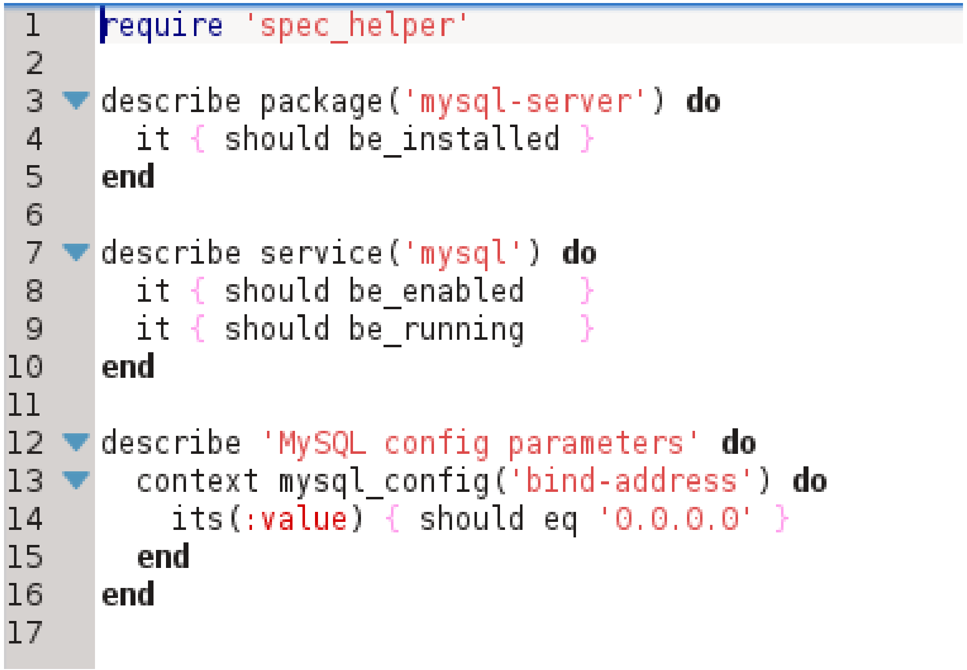
Page 1 of 1 (3 items)

# Wie werden Ansible Skripte getestet?

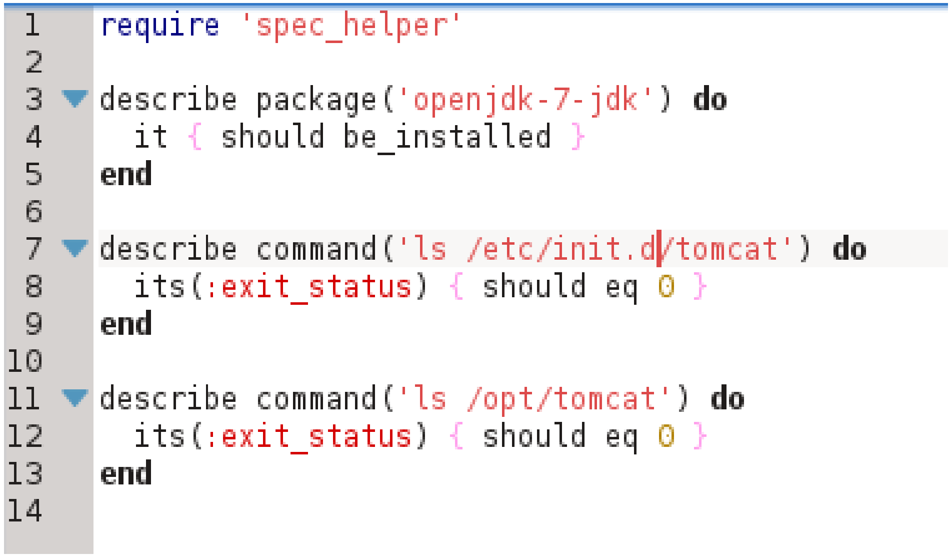
- `ansible-playbook --check`
- `ansible-playbook --syntax-check`
- Jenkins + Vagrant
- Rspec tests

SERVERSPEC

# ServerSpec Tests



```ruby
require 'spec_helper'

describe package('mysql-server') do
  it { should be_installed }
end

describe service('mysql') do
  it { should be_enabled   }
  it { should be_running   }
end

describe 'MySQL config parameters' do
  context mysql_config('bind-address') do
    its(:value) { should eq '0.0.0.0' }
  end
end

```



```ruby
require 'spec_helper'

describe package('openjdk-7-jdk') do
  it { should be_installed }
end

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

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

```

# ServerSpec Tests

# Wie unterscheidet sich Ansible zu seiner Konkurrenz?

# Vergleich

## Ansible

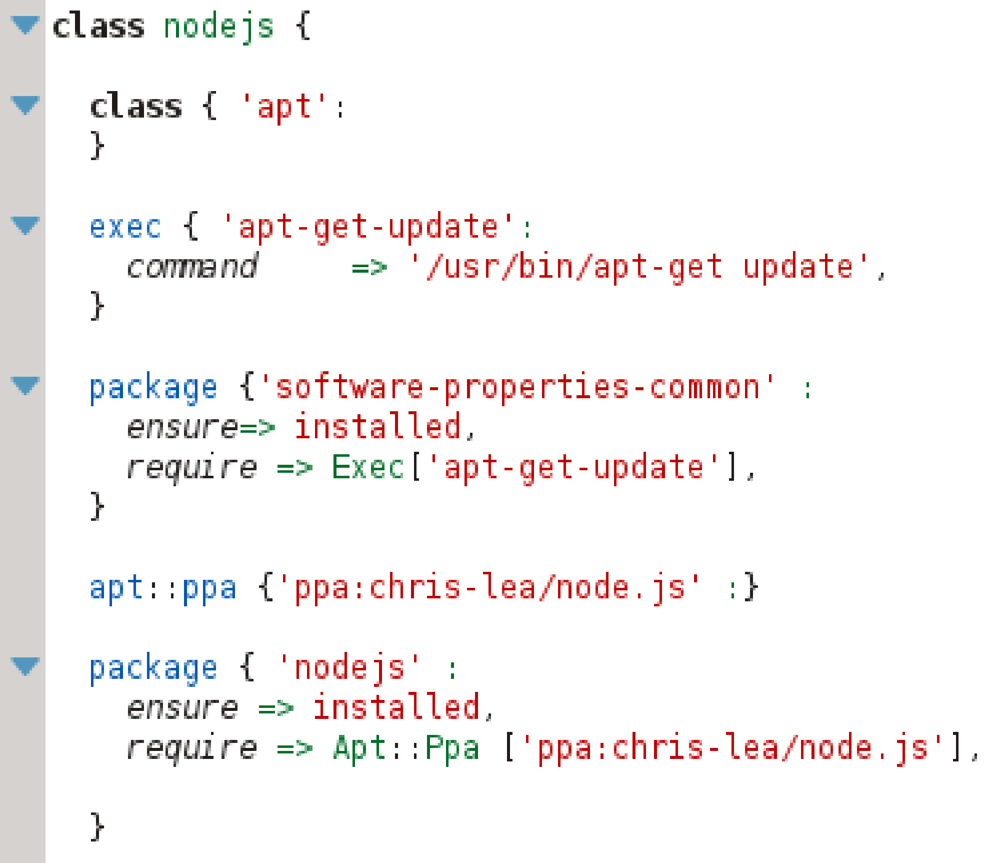
- Orchestrierung über SSH
- Benötigt keine Rootrechte auf Zielsystem
- Konfigurationsmgmt + Applikationsdeployment
- Monitoringtool nur in der Enterprise Variante
- Skripte mehr imperativ
- Windows-Support rudimentär
- Skripte OS- bzw. Distributions-spezifisch

## Puppet

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

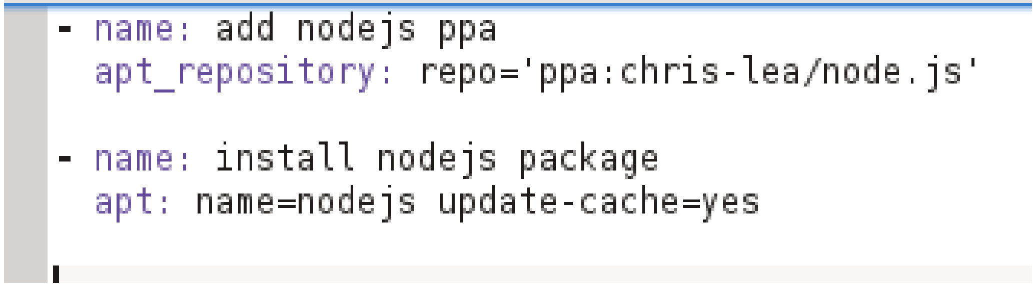
# Vergleich

## 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

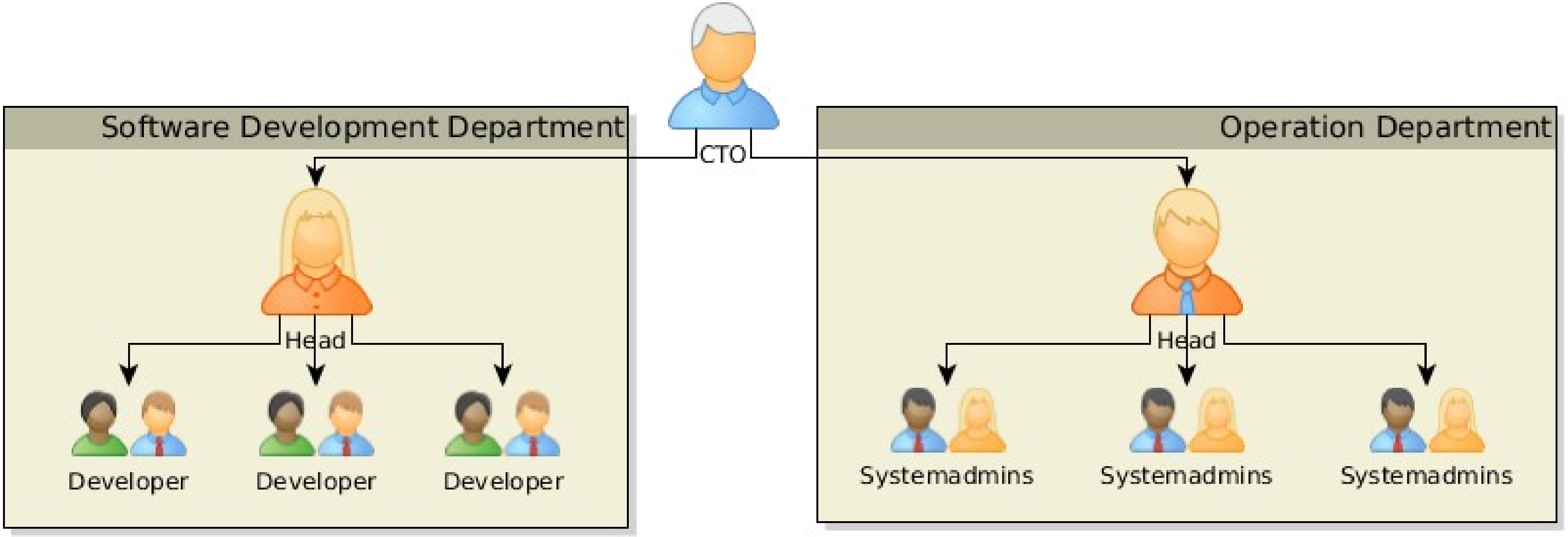


```
- name: add nodejs ppa
  apt_repository: repo='ppa:chris-lea/node.js'

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

# Weitere Einsatzszenarien aus Entwicklersicht

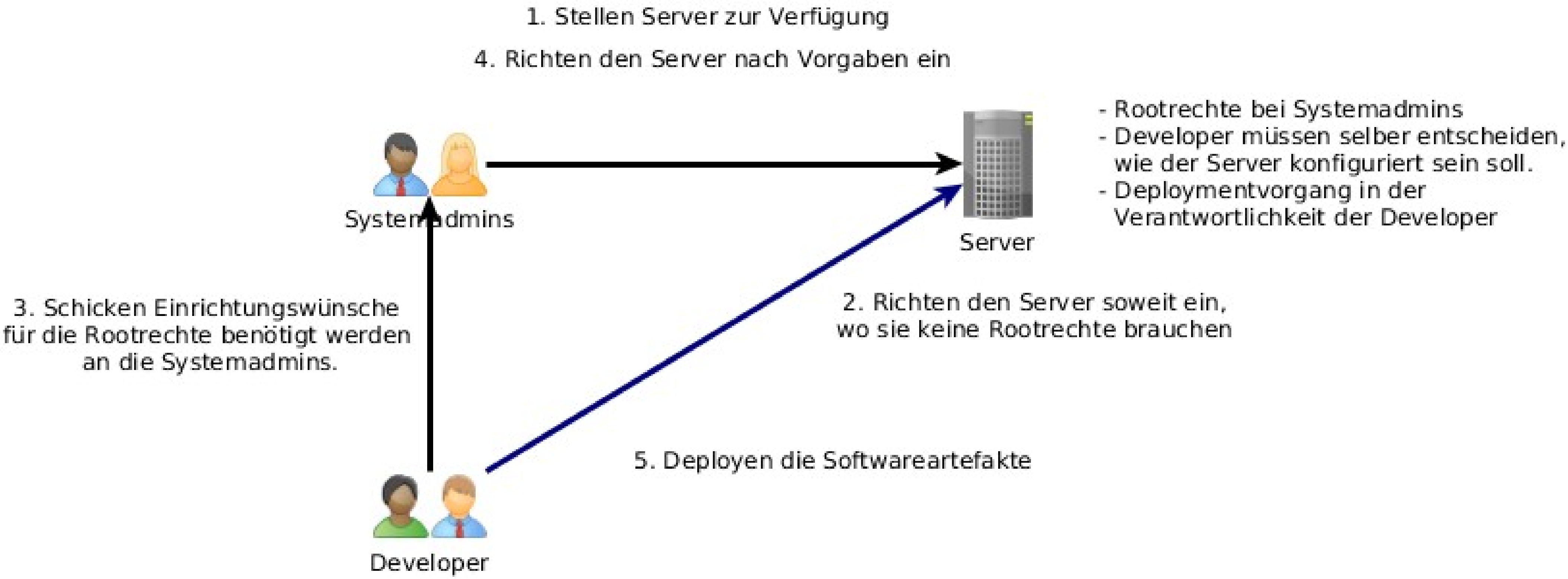
# Systemkonfiguration für Entwickler

## Organisatorische Ausgangslage
## Realität



# Systemkonfiguration für Entwickler

## Prozess zwischen Development und Operation



1. Stellen Server zur Verfügung

4. Richten den Server nach Vorgaben ein

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

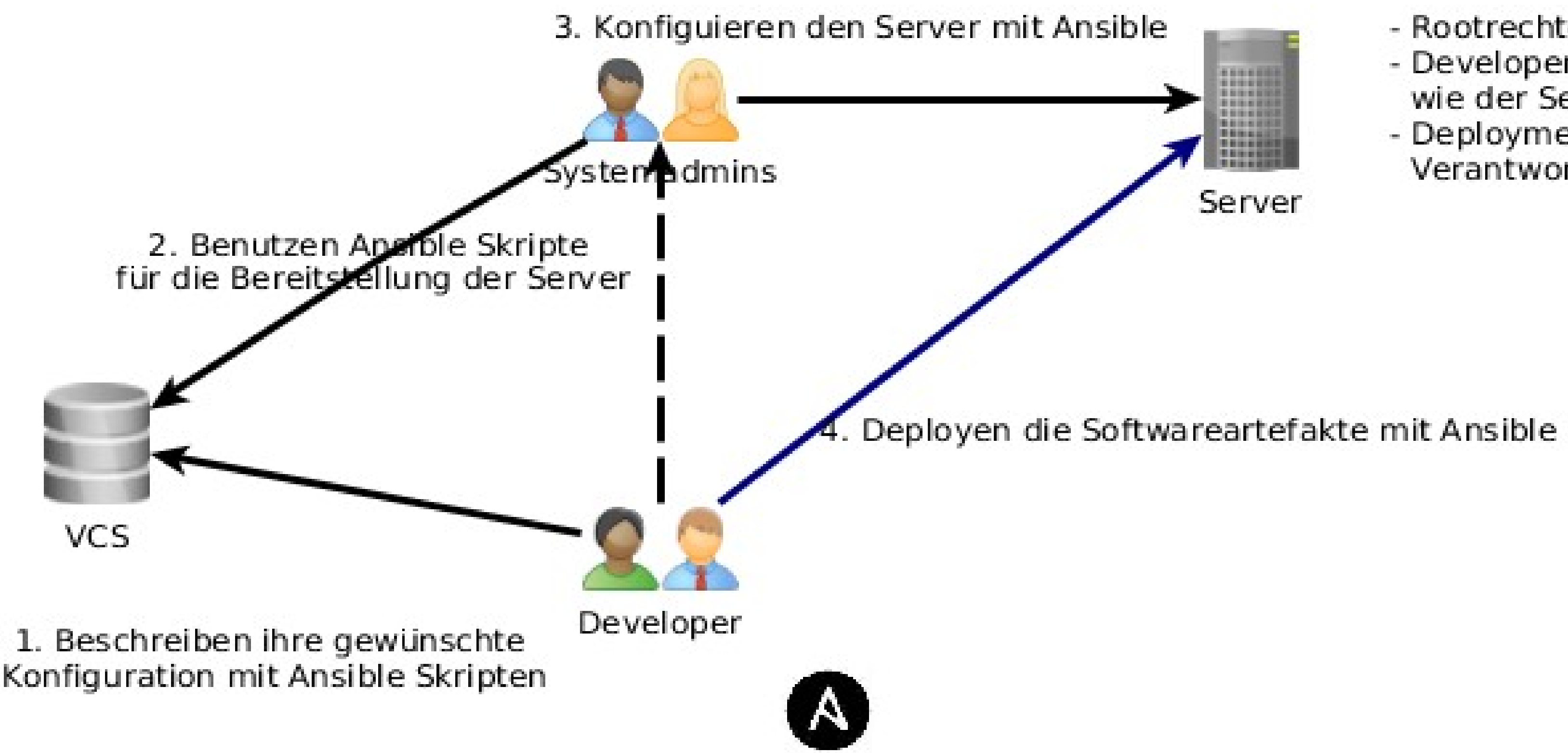
2. Richten den Server soweit ein,
wo sie keine Rootrechte brauchen

3. Schicken Einrichtungswünsche
für die Rootrechte benötigt werden
an die Systemadmins.

5. Deployen die Softwareartefakte

# Systemkonfiguration für Entwickler

## Lösungidee mit Ansible



1. Beschreiben ihre gewünschte Konfiguration mit Ansible Skripten

2. Benutzen Ansible Skripte für die Bereitstellung der Server

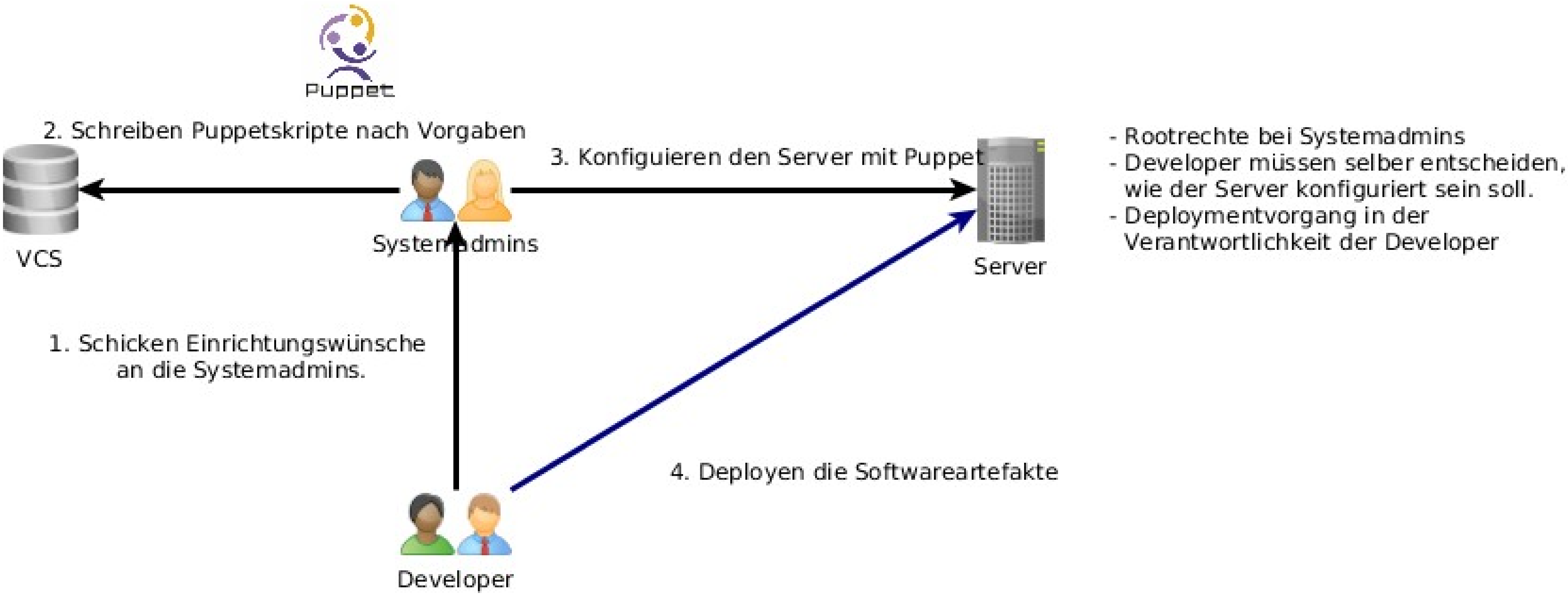
3. Konfiguieren den Server mit Ansible

4. Deployen die Softwareartefakte mit Ansible

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

# Systemkonfiguration für Entwickler

## Variante - Prozess zwischen Development und Operation

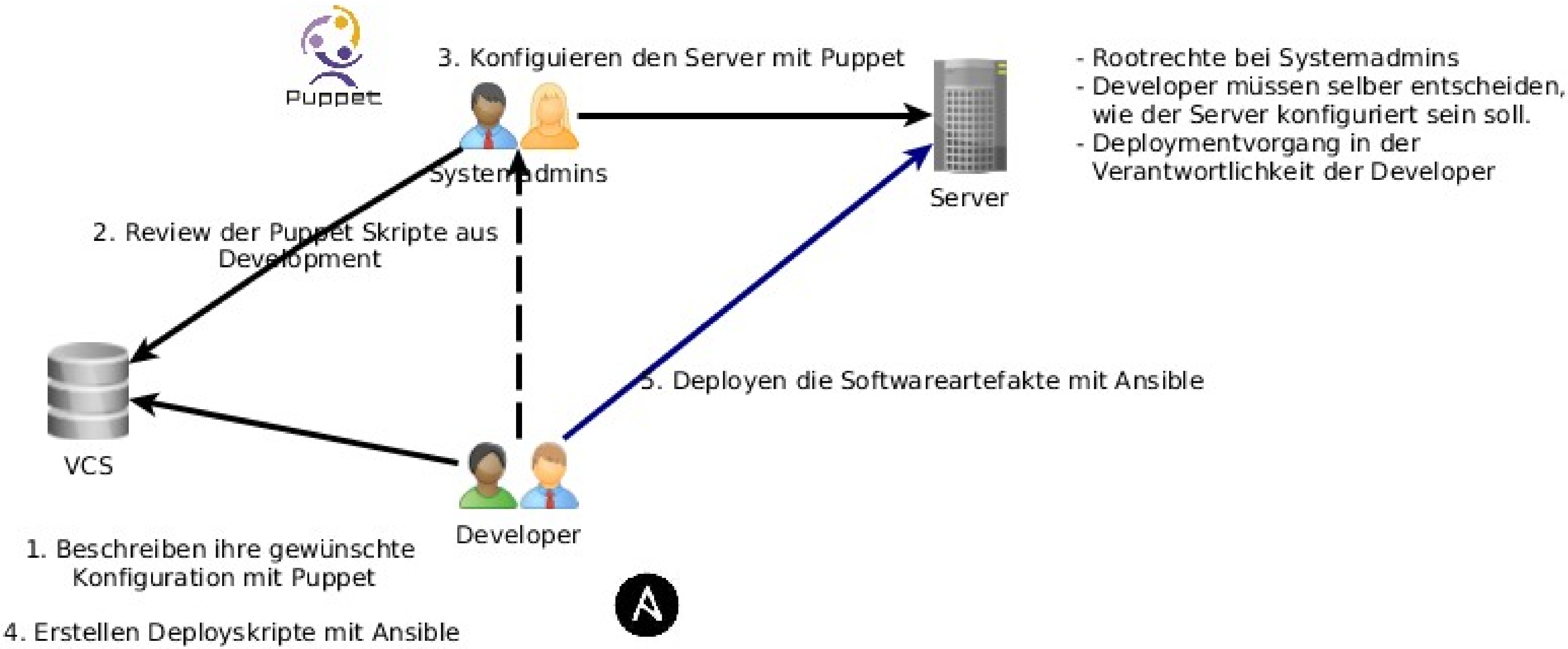


Puppet
2. Schreiben Puppetskripte nach Vorgaben
VCS
Systemadmins
3. Konfiguieren den Server mit Puppet
Server
1. Schicken Einrichtungswünsche an die Systemadmins.
4. Deployen die Softwareartefakte
Developer

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

# Systemkonfiguration für Entwickler

## Lösungsvariante



1. Beschreiben ihre gewünschte Konfiguration mit Puppet

2. Review der Puppet Skripte aus Development

3. Konfiguieren den Server mit Puppet

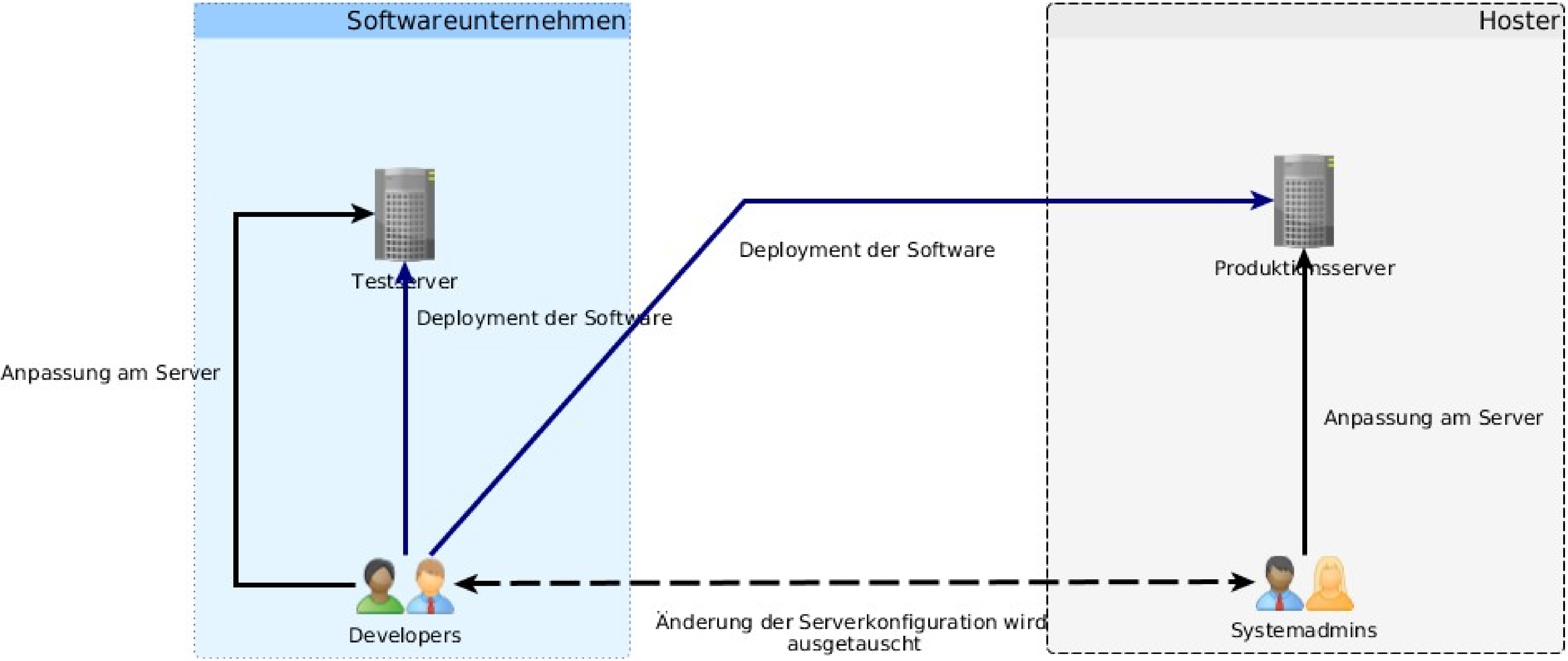
4. Erstellen Deployskripte mit Ansible

5. Deployen die Softwareartefakte mit Ansible

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

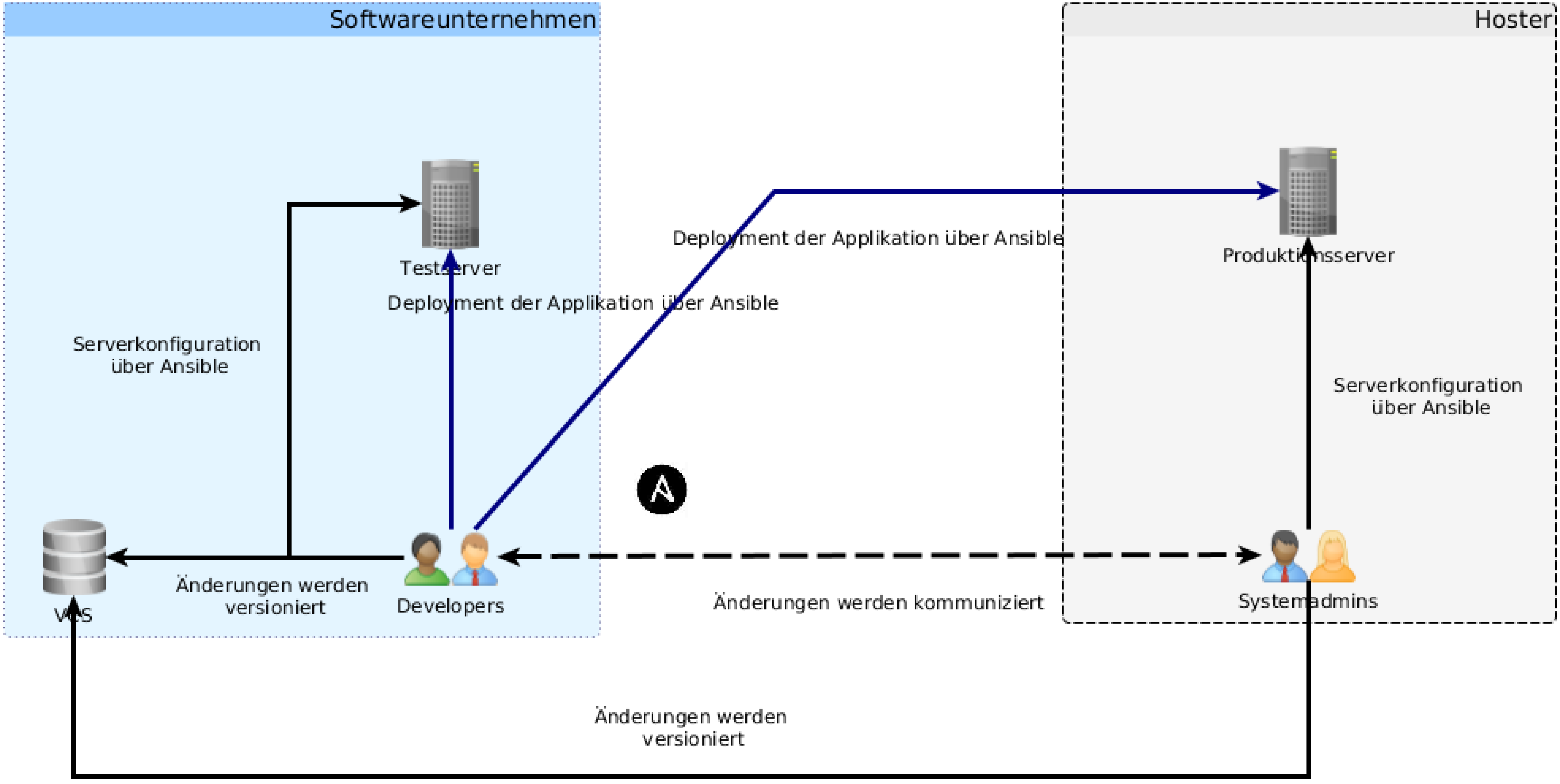
# Systemkonfiguration für Entwickler

## Produktionsserver sind beim externen Hoster



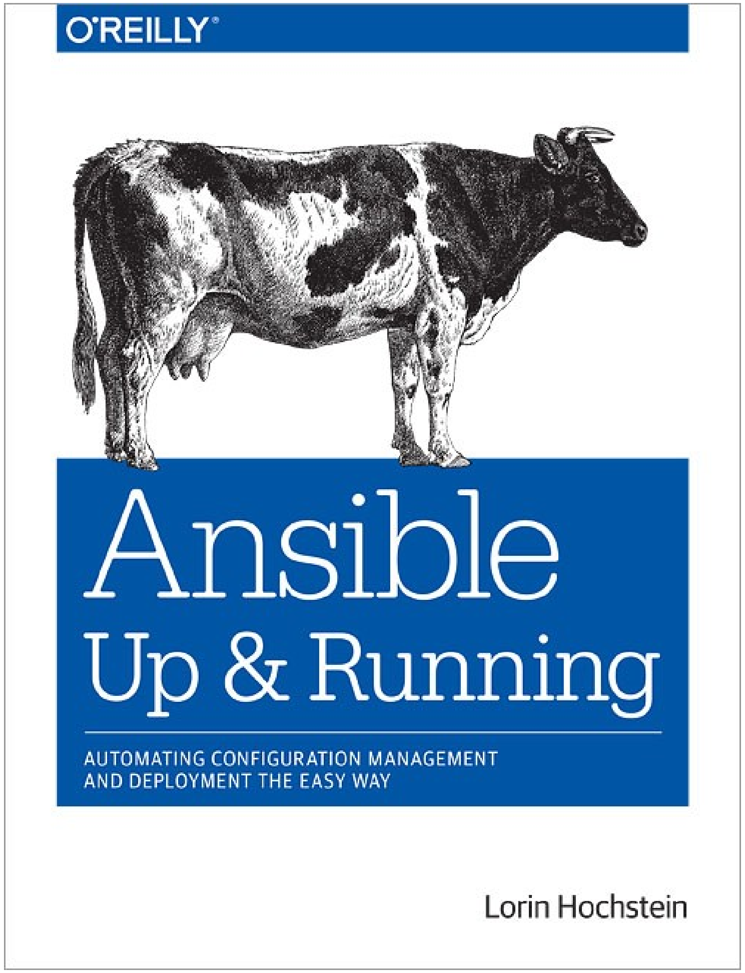
# Systemkonfiguration für Entwickler

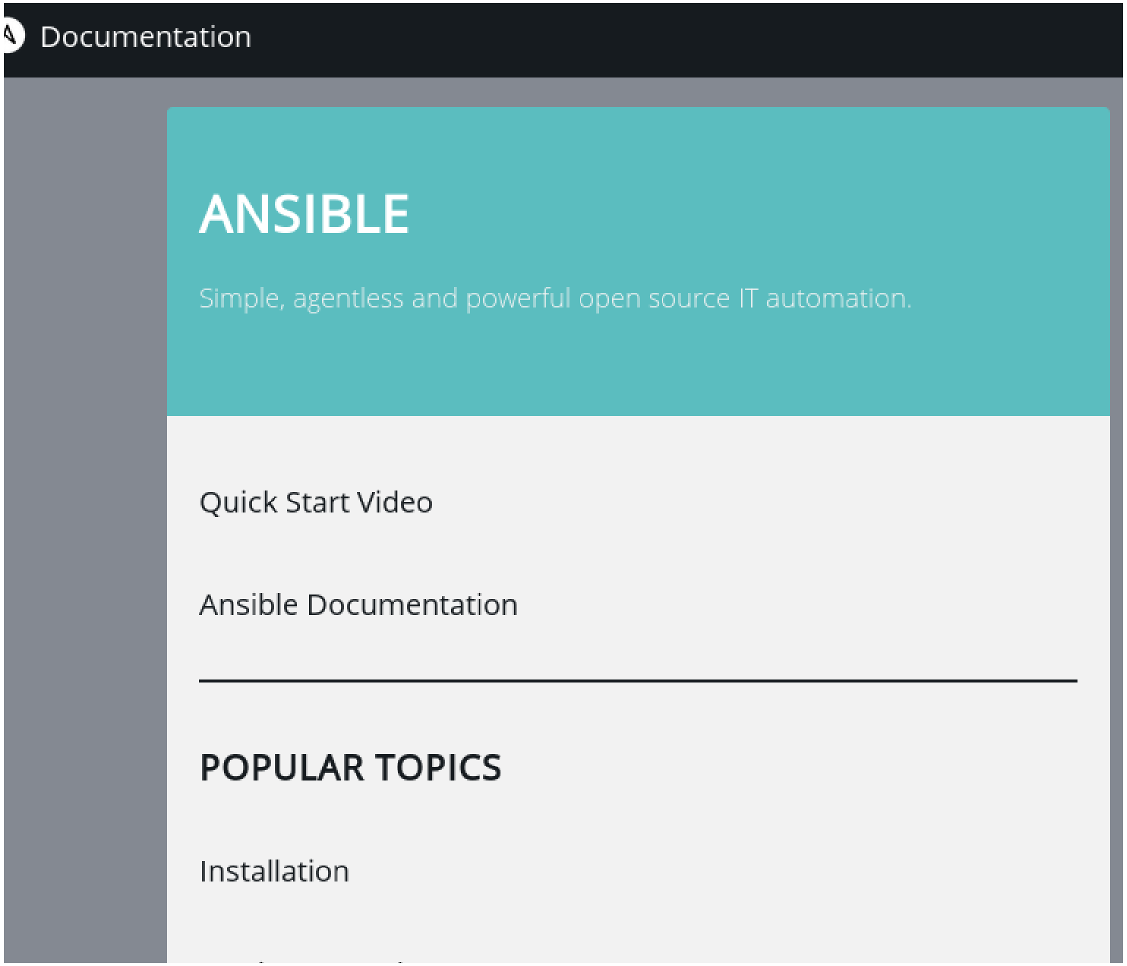
## Lösungsidee


Softwareunternehmen
Hoster
Testserver
Produktionsserver
Deployment der Applikation über Ansible
Deployment der Applikation über Ansible
Serverkonfiguration über Ansible
Serverkonfiguration über Ansible
VCS
Änderungen werden versioniert
Developers
Änderungen werden kommuniziert
Systemadmins
Änderungen werden versioniert

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

# Weitere Informationen


O'REILLY
Ansible
Up & Running
AUTOMATING CONFIGURATION MANAGEMENT
AND DEPLOYMENT THE EASY WAY
Lorin Hochstein


Documentation

ANSIBLE

Simple, agentless and powerful open source IT automation.

Quick Start Video

Ansible Documentation

POPULAR TOPICS

Installation

http://docs.ansible.com/

# Weitere Informationen

04-2016 | Winter | www.ijug.eu

**Java aktuell**

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

Java aktuell

D: 4,90 EUR A: 5,60 EUR CH: 9,80 CHF Benelux: 5,80 EUR ISSN 2191-6977

**Java ist vielseitig**

**JUnit 5**
Das nächste große Release steht vor der Tür

**Ansible**
Konfigurationsmanagement auch für Entwickler

**Spring Boot Starter**
Komfortable Modularisierung und Konfiguration

iJUG Verbund

*Software organisieren*

## Ansible – warum Konfigurationsmanagement auch für Entwickler interessant sein kann

Sandra Parsick

# Fragen?

@SandraParsick
mail@sandra-parsick.de
https://github.com/sparsick/ansible-talk.git