

# Erste Anwendungsbeispiele mit Ansible – Von der Theorie zur Praxis

## Der Start in die Automatisierung

Nachdem Sie Ansible installiert und die Grundlagen der Konfiguration verstanden haben, ist es Zeit, erste praktische Anwendungen zu erstellen. In diesem Artikel zeigen wir Ihnen, wie Sie mit Ansible einfache Aufgaben automatisieren können. Wir beginnen mit der Installation von Softwarepaketen und der Benutzerverwaltung und führen diese Anwendungsfälle Schritt für Schritt aus. Diese Beispiele bieten eine solide Basis für komplexere Automatisierungen.

## 1. Automatisierte Installation von Softwarepaketen

### 1.1 Szenario: Apache-Webserver installieren

Viele Systemadministratoren installieren Webserver wie Apache regelmäßig. Mit Ansible kann diese Aufgabe automatisiert werden.

#### Playbook-Beispiel

Erstellen Sie eine Datei `install_apache.yml` mit folgendem Inhalt:

```
---
- name: Apache-Webserver Installation
  hosts: webserver
  become: true
  tasks:
    - name: Update der Paketliste
      apt:
        update_cache: yes

    - name: Installiere Apache
      apt:
        name: apache2
        state: present

    - name: Apache starten und aktivieren
      service:
        name: apache2
        state: started
```

```
enabled: yes
```

### Erklärung der Tasks:

- **update\_cache**: Aktualisiert die Paketliste.
- **apt**: Installiert das Paket apache2.
- **service**: Stellt sicher, dass der Apache-Webserver gestartet und beim Boot aktiviert ist.

## Ausführung des Playbooks

Führen Sie das Playbook mit folgendem Befehl aus:

```
ansible-playbook -i inventory install_apache.yml
```

Nach der Ausführung können Sie den Webserver überprüfen, indem Sie die IP-Adresse des Servers in einem Browser aufrufen.

## 2. Automatisierte Benutzerverwaltung

### 2.1 Szenario: Benutzer erstellen und zu Gruppen hinzufügen

Die manuelle Benutzerverwaltung auf mehreren Servern ist zeitaufwändig und fehleranfällig. Mit Ansible können Sie Benutzer automatisch anlegen und deren Gruppenzugehörigkeit verwalten.

### Playbook-Beispiel

Erstellen Sie eine Datei `manage_users.yml`:

```
---
- name: Benutzerverwaltung
  hosts: all
  become: true
  tasks:
    - name: Erstelle Benutzer
      user:
        name: "{{ item.name }}"
        groups: "{{ item.groups }}"
        state: present
        shell: /bin/bash
      loop:
        - { name: 'developer1', groups: 'developers' }
        - { name: 'developer2', groups: 'developers' }
        - { name: 'admin1', groups: 'admins' }
```

### Erklärung der Tasks:

- **user**: Modul zur Benutzerverwaltung.
- **loop**: Erlaubt die Erstellung mehrerer Benutzer mit unterschiedlichen Attributen.

## Ausführung des Playbooks

Führen Sie das Playbook aus:

```
ansible-playbook -i inventory manage_users.yml
```

Nach der Ausführung sind die Benutzer erstellt und den jeweiligen Gruppen zugewiesen.

## 3. Sicherheit: Benutzerpasswörter setzen

### 3.1 Szenario: Passwörter für Benutzer setzen

Passwörter können mit Ansible sicher gesetzt werden, indem Sie den Hash des Passworts verwenden.

#### Playbook-Beispiel

Erstellen Sie eine Datei `set_passwords.yml`:

```
---
- name: Setze Benutzerpasswörter
  hosts: all
  become: true
  vars:
    admin_password: "$6$randomsalt$abc123hashedpassword"
  tasks:
    - name: Setze Passwort für Benutzer
      user:
        name: admin1
        password: "{{ admin_password }}"
```

**Hinweis:** Verwenden Sie Tools wie `mkpasswd`, um sichere, verschlüsselte Passwörter zu generieren.

```
bash
Code kopieren
mkpasswd --method=SHA-512
```

---

## 4. Automatisierte Konfigurationsänderungen

### 4.1 Szenario: Konfigurationsdateien verwalten

Mit Ansible können Sie Konfigurationsdateien einfach auf mehreren Systemen bereitstellen.

#### Playbook-Beispiel

Erstellen Sie eine Datei `deploy_config.yml`:

```
---
- name: Konfigurationsdateien bereitstellen
  hosts: webserver
  become: true
  tasks:
    - name: Lade Apache-Konfigurationsdatei hoch
      copy:
        src: files/apache2.conf
        dest: /etc/apache2/apache2.conf
        owner: root
        group: root
        mode: '0644'
    - name: Apache neu starten
      service:
        name: apache2
        state: restarted
```

Speichern Sie die Konfigurationsdatei `apache2.conf` im Verzeichnis `files/`.

#### Ausführung des Playbooks

```
ansible-playbook -i inventory deploy_config.yml
```

## 5. Praxis-Tipps und Best Practices

### 5.1 Wiederverwendbarkeit fördern

- Nutzen Sie Variablen für dynamische Inhalte, z. B. für Paketnamen oder Benutzergruppen.
- Definieren Sie gemeinsame Aufgaben in Rollen, um sie mehrfach verwenden zu können.

## 5.2 Sicherheit gewährleisten

- Sensible Daten wie Passwörter sollten mit **Ansible Vault** verschlüsselt werden:

```
ansible-vault encrypt vars.yml
```

- Verwenden Sie den `--ask-vault-pass` Parameter beim Ausführen von Playbooks.

## 5.3 Testen im Check-Modus

Führen Sie Playbooks im Testmodus aus, bevor Sie Änderungen vornehmen:

```
ansible-playbook -i inventory playbook.yml --check
```

## 6. Fazit und Ausblick

Mit diesen ersten Anwendungsbeispielen haben Sie die Grundlagen der Automatisierung mit Ansible erlernt. Von der Installation von Softwarepaketen bis zur Verwaltung von Benutzern und Konfigurationsdateien ermöglicht Ansible eine effiziente und konsistente Verwaltung Ihrer IT-Infrastruktur.

Im nächsten Artikel widmen wir uns einem komplexeren Szenario: der Automatisierung von Paketupdates abhängig von der installierten Linux-Distribution. Dieser Use-Case zeigt, wie Sie Bedingungslogik in Ihren Playbooks einsetzen und Ihre Automatisierung weiter verfeinern können.

## Über Achim Schmidt

Achim Schmidt ist ein erfahrener IT-Spezialist mit über 30 Jahren Expertise in den Bereichen IT-Sicherheit, Netzwerkarchitektur und Infrastrukturmanagement. Seit den frühen 90er Jahren begleitete er den Aufbau des Internets in Deutschland, besonders in Bayern. Als technischer Leiter und später Produktmanager prägte er bedeutende Internetprojekte und Netzwerklösungen für namhafte Unternehmen. Neben seiner Tätigkeit als Autor zahlreicher Fachartikel und Bücher, u. a. zu Linux, IT-Sicherheit und Netzwerktechnologien, engagiert er sich als zertifizierter Datenmanager und Teilnehmer des BSI IT-Sicherheitskongresses 2024 für aktuelle Themen der Datensicherheit und KI.

Im Internet findet man weitere Informationen zu ihm unter <http://www.achim-schmidt.de/> und <http://www.it-phoenix.de/>