Konfiguration von Ansible und Einführung in Playbooks

Einleitung: Vom Werkzeug zur Lösung

Nach der Installation von Ansible ist der nächste Schritt, es für Ihre Umgebung zu konfigurieren und mit sogenannten Playbooks zu arbeiten. Playbooks sind das Herzstück von Ansible: Sie definieren, welche Aufgaben auf welchen Systemen ausgeführt werden sollen. In diesem Artikel erfahren Sie, wie Sie Ansible optimal konfigurieren, Inventories erstellen und Ihre ersten Playbooks entwickeln, um Automatisierung effektiv einzusetzen.

1. Grundlagen der Ansible-Konfiguration

1.1 Verzeichnisstruktur

Nach der Installation sollten Sie Ansible in einer strukturierten Umgebung nutzen. Eine empfohlene Verzeichnisstruktur sieht wie folgt aus:

```
project/
├── ansible.cfg          # Zentrale Konfigurationsdatei
├── inventory/           # Verzeichnis für Inventardateien
│   ├── hosts.ini        # Statische Inventardatei
│   └── dynamic/         # Optionale dynamische Inventare
├── playbooks/           # Verzeichnis für Playbooks
│   └── example.yml      # Beispiel-Playbook
└── roles/               # Verzeichnis für Rollen
    ├── common/
    └── webserver/
```

1.2 Die Konfigurationsdatei ansible.cfg

Die Datei ansible.cfg ist das Rückgrat Ihrer Ansible-Umgebung. Sie erlaubt die Anpassung globaler Parameter. Ein Beispiel für eine typische Konfiguration:

```
[defaults]
inventory = inventory/hosts.ini    # Pfad zur Inventardatei
remote_user = ansible              # Standardbenutzer für SSH
host_key_checking = False          # Deaktiviert SSH-Hostschlüsselprüfung
forks = 10                         # Maximale parallele Verbindungen
timeout = 30                       # Timeout für SSH-Verbindungen
log_path = logs/ansible.log        # Log-Datei

[privilege_escalation]
become = True                      # Aktiviert sudo
become_method = sudo               # Methode: sudo
become_user = root                 # Zielbenutzer: root
```

Speichern Sie diese Datei im Root-Verzeichnis Ihres Projekts.

2. Inventories – Die Zielsysteme definieren

2.1 Statische Inventardateien

Ein statisches Inventory ist eine einfache Datei, die die verwalteten Systeme definiert. Eine typische hosts.ini-Datei könnte wie folgt aussehen:

```
[webserver]
192.168.1.10
192.168.1.11

[database]
192.168.1.20 ansible_user=dbadmin ansible_port=2222
```

Erklärung der Parameter:

- **[Gruppe]**: Gruppiert Server, die gemeinsam verwaltet werden sollen.
- **ansible_user**: Benutzername für SSH-Verbindung.
- **ansible_port**: Alternativer SSH-Port (standardmäßig 22).

2.2 Dynamische Inventardateien

Für größere Umgebungen oder Cloud-Ressourcen können Sie dynamische Inventories nutzen. Ansible unterstützt Plugins für AWS, Azure, Google Cloud und mehr. Beispiel: AWS-Inventory über das Plugin aws_ec2.

```
[defaults]
inventory = inventory/dynamic/aws_ec2.yml
```

Konfigurieren Sie aws_ec2.yml, um Ihre EC2-Instanzen zu laden. Dies erfordert zusätzliche Bibliotheken wie boto3.

3. Einführung in Ansible-Playbooks

3.1 Was sind Playbooks?

Playbooks sind YAML-Dateien, die beschreiben, welche Aufgaben (Tasks) Ansible auf den Zielsystemen ausführen soll. Sie bestehen aus:

- **Hosts**: Zielsysteme oder -gruppen.
- **Tasks**: Einzelne Anweisungen, die ausgeführt werden.
- **Handler**: Aktionen, die nur bei bestimmten Änderungen ausgeführt werden.

3.2 Ein einfaches Playbook

Erstellen Sie ein Playbook playbooks/update_system.yml, um Pakete zu aktualisieren:

```
---
- name: Systemaktualisierung durchführen
  hosts: all
  become: true
  tasks:
    - name: Aktualisiere Paketliste
      apt:
        update_cache: yes

    - name: Führe Upgrade durch
      apt:
        upgrade: dist
```

Ausführung:

[bash]

```
ansible-playbook playbooks/update_system.yml
```

3.3 Best Practices für Playbooks

1. **Wiederverwendbarkeit fördern**:
   Nutzen Sie Variablen, um Playbooks flexibler zu gestalten.
2. **Module nutzen**:
   Verwenden Sie die umfangreiche Modulbibliothek von Ansible, z. B. file, user, service.
3. **Aufgaben strukturieren**:
   Gruppieren Sie Aufgaben in Rollen, um sie modular und übersichtlich zu halten.

4. Rollen in Ansible

4.1 Einführung in Rollen

Rollen sind vorstrukturierte Verzeichnisse, die Aufgaben, Variablen, Templates und Dateien bündeln.

Verzeichnisstruktur einer Rolle:

```
roles/
└── webserver/
    ├── tasks/
    │   └── main.yml    # Hauptaufgaben
    ├── handlers/
    │   └── main.yml    # Aktionen bei Änderungen
    ├── templates/
    │   └── nginx.conf.j2
    └── files/
        └── index.html
```

4.2 Beispiel einer Rolle: Webserver

tasks/main.yml:

```
---
- name: Apache installieren
  apt:
    name: apache2
    state: present

- name: Starte Apache
  service:
    name: apache2
    state: started
    enabled: true
```

Einbindung der Rolle in ein Playbook:

```
---
- name: Webserver installieren
  hosts: webserver
  roles:
    - webserver
```

5. Fehlerbehebung und Debugging

5.1 Ansible-Check-Modus

Führen Sie Playbooks im Testmodus aus, um Änderungen zu simulieren:

[bash]

ansible-playbook playbooks/update_system.yml --check

5.2 Fehlerlogging

Aktivieren Sie ausführliches Logging für detaillierte Fehleranalysen:

[bash]

ansible-playbook playbooks/update_system.yml -vvv

6. Nützliche Module und Funktionen

6.1 Wichtige Module

- **apt**: Paketverwaltung für Debian-basierte Systeme.
- **user**: Benutzerverwaltung.
- **file**: Bearbeitung von Dateiberechtigungen.
- **copy**: Dateien auf Zielsysteme kopieren.

6.2 Bedingte Aufgaben

Führen Sie Aufgaben nur unter bestimmten Bedingungen aus:

```
---
- name: Nur auf Ubuntu 20.04 ausführen
  apt:
    name: apache2
  when: ansible_distribution_version == "20.04"
```

Fazit und Ausblick

In diesem Artikel haben Sie gelernt, wie Sie Ansible konfigurieren, Inventories erstellen und erste Playbooks schreiben. Mit diesen Grundlagen können Sie Ihre Umgebung effizient verwalten und optimieren. Im nächsten Artikel setzen wir das Wissen in die Praxis um und erstellen erste Automatisierungslösungen wie Software-Installationen und Benutzerverwaltung.

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/