

Thema: Netzwerkautomatisierung	Ansible	Aufgaben
--------------------------------	---------	----------

Situation

Sie sind bei einem IT-Dienstleister mit den Schwerpunkten ‚Managed Hosting‘ und ‚Cloud Services‘ beschäftigt. Die internen Prozesse sollen mit Hilfe von Automatisierung zeiteffizient gestaltet werden. Sie erhalten den Auftrag Ansible-Playbooks für Netzwerkkomponenten zu entwickeln.

Nachfolgend mit dem Betriebssystem IOS (z.B. *IOS XE* oder *IOS 15*).

Aufträge

Vorbereitungen:

- Überprüfen Sie, ob Ansible installiert ist (z.B. `ansible --version`), ansonsten installieren Sie Ansible.
- Legen Sie zunächst folgendes Verzeichnis an:

```
ansible/
|-- roles
```
- Zum Ausführen der Aufgaben muss Ansible auf eine Inventory-Datei zugreifen, die `hosts` genannt wird. Erstellen Sie die `hosts`-Datei, speichern Sie diese Datei im Ordner `ansible` ab und ergänzen Sie die Angaben Ihres Routers. *Beispiel:*

```
[routers]
CSR1 ansible_host=192.168.178.98

[all:vars]
# treats the node as a network device
ansible_connection=ansible.netcommon.network_cli

# which network platform - ios, aruba, etc.
ansible_network_os=ios

ansible_user=admin
ansible_password=geheim

# privilege mode
ansible_become=true
ansible_become_method=enable
ansible_become_password=geheim

---
```

Erklärung – Auszug aus: https://cn-ansibledoc.readthedocs.io/zh_CN/latest/network/user_guide/network_best_practices_2.5.html

ansible_connection: Ansible uses the `ansible_connection` setting to determine how to connect to a remote device. When working with Ansible Networking, set this to `network_cli` so Ansible treats the remote node as a network device with a limited execution environment. Without this setting, Ansible would attempt to use `ssh` to connect to the remote and execute the Python script on the network device, which would fail because Python generally isn't available on network devices.

Thema: Netzwerkautomatisierung	Ansible	Aufgaben
--------------------------------	---------	----------

ansible_network_os: Informs Ansible which Network platform this hosts corresponds to. This is required when using `network_cli` or `netconf`.

- In der `ansible.cfg`-Datei werden Angaben zur Konfiguration hinterlegt. Passen Sie die Konfiguration an um u.a. den Pfad zur `host`-Datei und zum Ordner `roles` zu bestimmen. Berücksichtigen Sie dabei folgende Information:

Changes can be made and used in a configuration file which will be searched for in the following order:

- ANSIBLE_CONFIG (environment variable if set)*
- ansible.cfg (in the current directory)*
- ~/.ansible.cfg (in the home directory)*
- /etc/ansible/ansible.cfg*

Ansible will process the above list and use the first file found, all others are ignored.

- Auszug:

https://docs.ansible.com/ansible/latest/reference_appendices/config.html#ansible-configuration-settings

```
[defaults]
inventory = hosts
```

Mithilfe von Ansible-Modulen kann die Konfiguration von Netzwerkgeräten abgerufen oder geändert werden. Eine Übersicht der Module für das IOS finden Sie hier:

<https://docs.ansible.com/ansible/latest/collections/cisco/ios/index.html>

<https://github.com/ansible-collections/cisco.ios>

- 1) Mittels des ersten Playbooks soll der Befehl `show version` vom Router angezeigt werden. Kopieren Sie hierzu die Datei `1_version_playbook.yaml` und speichern Sie die Datei im Ordner `playbooks`.

Öffnen Sie das Terminal, navigieren Sie zum Ordner `playbooks` und führen Sie das Playbook aus:

```
ansible-playbook 1_version_playbook.yaml
```

- 2) Erweitern Sie das Playbook und ergänzen Sie die Ausgabe um den Befehl:

```
show ip interface brief
```

- 3) Entwickeln Sie ein Ansible-Playbook für die folgenden Befehle im Rahmen einer Grundkonfiguration:

- MOTD-Anmeldebanner einrichten
- Passwörter verschlüsseln [`service password-encryption`]
- NTP-Server (selbst gewählt)
- Speichern der Konfigurationsdatei

Version A) Stellen Sie die Module zusammen und führen Sie sie aus.

Version B) Setzen Sie im Playbook für den NTP-Server eine Variable ein.

Beispiel:

```
...
vars:
    ntp: "192.168.178.2"

tasks:
...
- name: NTP Server
  cisco.ios.ios_ntp_global:
    config:
      servers:
        - server: "{{ntp}}"
        state: replaced
...
```

Version C) Fügen Sie für das Speichern der Konfiguration eine Rolle ein.

In den Ordner `roles` einen Ordner `copy_run_start` anlegen und darin eine Datei `main.yaml` mit dem Zielzustand:

```
- name: Save running to start-config when modified
  ios_config:
    save_when: modified
```

Im Playbook kann die Rolle am Ende wie folgt aufgerufen werden:

```
- include_roles:
    name: Copy_run_start
```

- 4) [Router] Fügen Sie ein Loopback-Interface mit Beschreibung hinzu und konfigurieren Sie eine IPv4 und IPv6-Adresse. Bezeichnung des Interface sowie die IP-Adressen sollen als Variable ausgewiesen werden und die Überprüfung erfolgt anhand einer Rolle.
- 5) [Router] Löschen Sie das zuvor erzeugte Loopback-Interface.
- 6) [Router] Konfigurieren Sie eine statische Route und lassen Sie die Routingtabelle zur Kontrolle anzeigen. Optional: Löschen Sie die statische Route.
- 7) Erzeugen Sie Backups und lassen Sie diese als Datei in einem selbst gewählten Ordner speichern. Nutzen Sie hierfür das Modul `ios_config → backup`.
- 8) Für den Switch sollen VLAN 20 mit dem Namen Verwaltung angelegt werden. Der Uplink soll in den Trunk-Modus sowie ein selbst gewähltes Interface in den Access-Modus mit VLAN-ID 20.