



The Linux & Open Source Company

Eine praxisorientierte Einführung in AWX

Dr. Ottavia Balducci, ATIX AG

- ▶ CLI-Ansible-Wiederholung
- ▶ AWX-Überblick
- ▶ Projekte in AWX
- ▶ Job Templates und Inventories
- ▶ Workflows
- ▶ Execution Environments

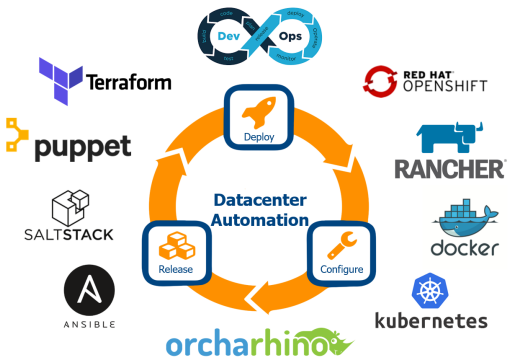
- ▶ Ottavia Balducci
- ▶ IT-Beraterin bei ATIX AG

Mein Fokus:

- ▶ Ansible
- ▶ AWX/AAP
- ▶ orcharhino/Foreman



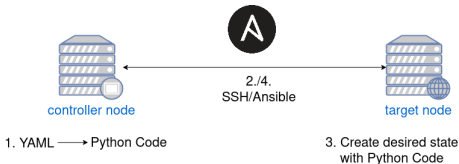
▶ Email: balducci@atix.de



Open-Source Linux & Automatisierung:

- ▶ Consulting
- ▶ Engineering
- ▶ Support
- ▶ Training (atix.de/trainings)

- ▶ Werkzeug für die Automatisierung von Konfigurationsmanagement
- ▶ Ohne Clients und Daemons
- ▶ Idempotent (beschreibe nur den Endzustand)
- ▶ Parallel auf Zielhosts ausgeführt
- ▶ Zielhosts können in Gruppen unterteilt werden



Einfachstes Szenario: Nur 2 Dateien nötig!

- ▶ Inventory: Beschreibt die Zielhosts
- ▶ Playbook: Beschreibt den gewünschten Endzustand der Zielhosts

```
ansible-playbook -i inventory.yaml playbook.yaml
```

Beispiel: Inventory

```
---
all:
  hosts:
    my-host.example.com:
      ansible_host: 192.168.165.5  # you can specify the IP if needed
      ansible_user: atix  # used by Ansible for SSH connection
      ansible_password: secret  # avoid this: use ansible-vault!
    my-other-host.example.com:
      ansible_host: 192.168.165.6
      ansible_user: atix
      ansible_password: secret
  vars:  # these variables are assigned to all hosts above
    my_var: 'Chemnitz'
    another_var: '2025'
...
```

Beispiel: Playbook

```
---  
- name: Set correct timezone  
  hosts: all  
  tasks:  
    - name: Install chrony client on hosts  
      become: true  
      ansible.builtin.package:  
        name: "chrony"  
        state: present  
  
    - name: Set timezone  
      become: true  
      community.general.timezone:  
        name: Europe/Berlin  
...  

```



- ▶ Auf Ansible aufbauende Weboberfläche
- ▶ Role Based Access Control
- ▶ Credentials-Verwaltung
- ▶ Execution Environments
- ▶ Hochverfügbarkeit
- ▶ Automation Mesh
- ▶ ...



- ▶ In k8s-Cluster mit dem AWX-Operator: <https://github.com/ansible/awx-operator>
- ▶ Datenbank kann extern sein
- ▶ ⚠ Projekt wird gerade neu strukturiert ⇒ Release-Pause!
- ▶ Upgrades und Backups auch mit Operator möglich
- ▶ Meine Empfehlung zum Ausprobieren: <https://github.com/kurokobo/awx-on-k3s>

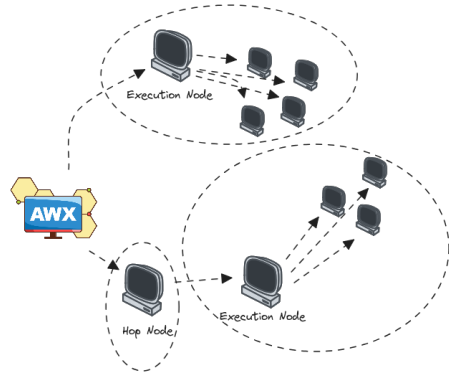
Organizations:

- ▶ Users
- ▶ Teams
- ▶ Projekte
- ▶ Inventories
- ▶ ...

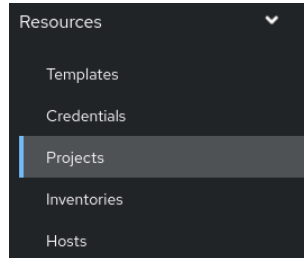
Rollenzuweisung (read, write, execute, ...):

- ▶ Users
- ▶ Teams

- ▶ Execution Nodes
- ▶ Hop Nodes
- ▶ Instance groups



- ▶ enthält Ansible-Content (Playbooks, Rollen, usw.)
- ▶ "Repo"
- ▶ verschiedene Quellen (Git, Subversion, ...)
- ▶ davon werden Job Templates abgeleitet
- ▶ gehört einer Organisation



- ▶ mitgelieferte Credential-Types:

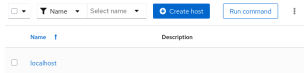
- ▶ Ansible Galaxy
- ▶ Container Registry
- ▶ GitLab PAT
- ▶ Machine
- ▶ Source Control
- ▶ ...

- ▶ Eigene Credential-Types erstellbar
- ▶ Ersatz für `ansible-vault`
- ▶ Verschlüsselt in DB

- ▶ Sammlung von Hosts
- ▶ ähnlich zu `inventory.yaml`
- ▶ Gruppen und Untergruppen möglich
- ▶ dynamische Inventories möglich
 - ▶ VMWare vCenter
 - ▶ AWS EC2
 - ▶ Hetzner
 - ▶ ...

Standard:

► Hosts von Hand



Standard inventory creation interface. It features a dropdown menu for selection, a text input field labeled 'Name', a 'Select name' button, a 'Create host' button, and a 'Run command' button. Below this is a table with columns 'Name' and 'Description'. The table contains one entry: 'localhost'.

Smart:

► Filter auf vorhandenen Hosts

► (deprecated)

Smart host filter * ⓘ

Enter smart host filter

Constructed:

► aus anderen Inventories gebaut

Input inventories * ⓘ

Demo Inventory X Standard Inventory X X

- ▶ einem Inventory zugewiesen
- ▶ von Hand oder aus einer externen Quelle
- ▶ Variablen-Zuweisung
- ▶ Credentials vom Typ "Machine"

```
---  
ansible_user: <user>  
ansible_password: <password>  
ansible_private_key_file: <path>  
...
```

- ▶ Aus einem Projekt erstellt
- ▶ Wähle ein Playbook aus dem Projekt
- ▶ Zusatzvariablen
- ▶ Inventory
- ▶ und mehr...

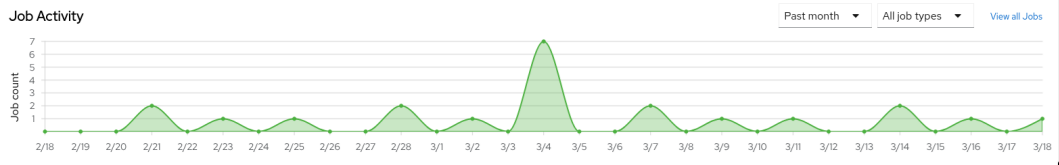
Templates

Create New Job Template



Name *	Description	Job Type * ⓘ Run	☐ Prompt on launch
Inventory * ⓘ 	☐ Prompt on launch	Project * ⓘ Demo Project	Execution Environment ⓘ ☐ Prompt on launch
Playbook * ⓘ hello_world.yml			
Credentials ⓘ 			☐ Prompt on launch
Labels ⓘ 			☐ Prompt on launch
Variables ⓘ YAML JSON 1 --- 2		☐ Prompt on launch	

Job Activity



Demo Job Template

Success

Plays 1

Tasks 2

Hosts 1

Elapsed 00:00:10

Success 100%

Search output

Filter by keyword



^

^

^

^

^

```
0 [DEPRECATION WARNING]: ANSIBLE_COLLECTIONS_PATHS option, does not fit var
1 naming standard, use the singular form ANSIBLE_COLLECTIONS_PATH instead. This
2 feature will be removed from ansible-core in version 2.19. Deprecation warnings
3 can be disabled by setting deprecation_warnings=False in ansible.cfg.
```

^

```
5 PLAY [Hello World Sample] ..... 2:48:25 PM
```

^

```
7 TASK [Gathering Facts] ..... 2:48:25 PM
```

```
8 ok: [localhost]
```

^

```
10 TASK [Hello Message] ..... 2:48:27 PM
```

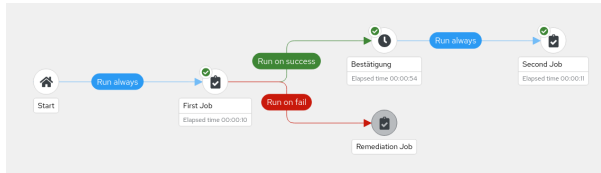
```
11 ok: [localhost] => {
12   "msg": "Hello World!"
13 }
```

```
15 PLAY RECAP .....
```

```
16 localhost                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- ▶ Zentral gesammelt
- ▶ Filter
- ▶ JSON
- ▶ Anpassbare Ausführlichkeit

- ▶ Ketten von Jobs
- ▶ Mögliche Schritten:
 - ▶ Job Template
 - ▶ Workflow Job Template
 - ▶ Project Sync
 - ▶ Approval
 - ▶ Inventory Source Sync
 - ▶ Management Job



- ▶ Container-Images für die eigentliche Ansible-Ausführung
- ▶ erstellt mit `ansible-builder`
- ▶ konfigurierbar:
 - ▶ eigene Ansible-Collections
 - ▶ eigene Python-Pakete
 - ▶ eigene Zertifikate
 - ▶ ...
- ▶ nutzbar mit AWX, AAP, `ansible-navigator`



execution-environment.yml

```
---
version: 3

dependencies:
  galaxy: requirements.yml

images:
  base_image:
    name: quay.io/ansible/awx-ee:24.6.1

additional_build_steps:

  prepend_final: |
    RUN whoami
    RUN cat /etc/os-release
  append_final:
    - RUN echo This is a post-install command!
    - RUN ls -la /etc
```

Dokumentation:

<https://ansible.readthedocs.io/projects/builder/en/stable/definition/>

requirements.yml

collections:

- name: https://gitlab-trainer.awx.atix-training.de/root/atix.workshop.git
type: git
- community.general

- ▶ Sammlung von Ansible-Code
- ▶ Kann enthalten
 - ▶ Playbooks
 - ▶ Roles
 - ▶ Plugins (Filter, Module, Lookup, ...)
 - ▶ ...
- ▶ FQCN: Fully Qualified Collection Name

```
---  
- name: Check pod status  
  kubernetes.core.k8s_info:  
    kind: Pod  
    namespace: awx  
    kubeconfig: "{{ kubeconfig_path }}"  
    register: pod_status  
  
- name: Return the difference of list1 and list2.  
  ansible.builtin.debug:  
    msg: "{{ list1 | community.general.lists_difference(list2) }}"  
  vars:  
    list1: [1, 2, 5, 3, 4, 10]  
    list2: [1, 2, 3, 4, 5, 11, 99]  
# => [10]
```

Syntax für ansible-builder:

```
ansible-builder build --container-runtime docker -t <registry>/<image-name>:<tag>
```

Pushe nach Registry:

```
docker push <registry>/<image-name>:<tag>
```

Vielen Dank für die Aufmerksamkeit!

 balducci@atix.de

 atix.de