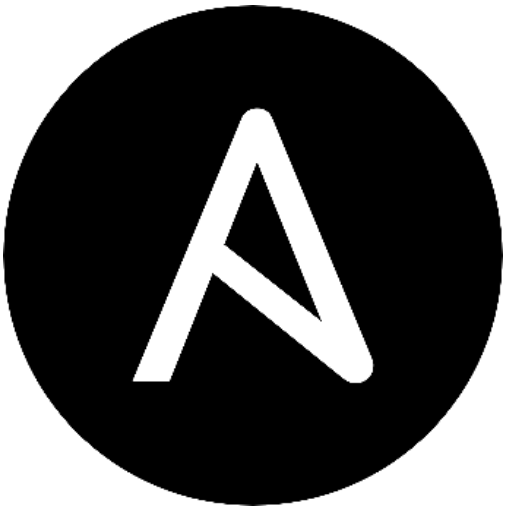
Ansible Dokumentation



Alle Ansible Seiten

Inhalt

Suche

---

1 Was ist Ansible?

Ansible ist eine Plattform zum Administrieren und zum „Befeuern“ von Servern und Befehlen.
Ansible ist ein Open-Source-Automatisierungswerkzeug zur Orchestrierung und allgemeinen Konfiguration und Administration von Computern. Es kombiniert Softwareverteilung, Ad-hoc-Kommando-Ausführung und Software-Configuration-Management.

2 Ansible in der KfW - Erste Schritte

2.1 Benötigte Berechtigungen

2.1.1 RSA-Token

Bitte im IT-Webshop folgende Berechtigung bestellen:

B_P_01_RSAAM_Extranet_Token_000000037908

2.1.2 Subidentität

Diese bekommt man unter dem IT-Webshop unter: Auftrag > Neuer Auftrag > Identität > Subidentität; anlegen und bestellen. Im RSA-KfW-Portal Passwortänderung und PIN-Änderung Mit RSA-Token

2.1.3 Berechtigungen

Standarduser

B_P_01_RSAAM_Extranet_Token_000000037908

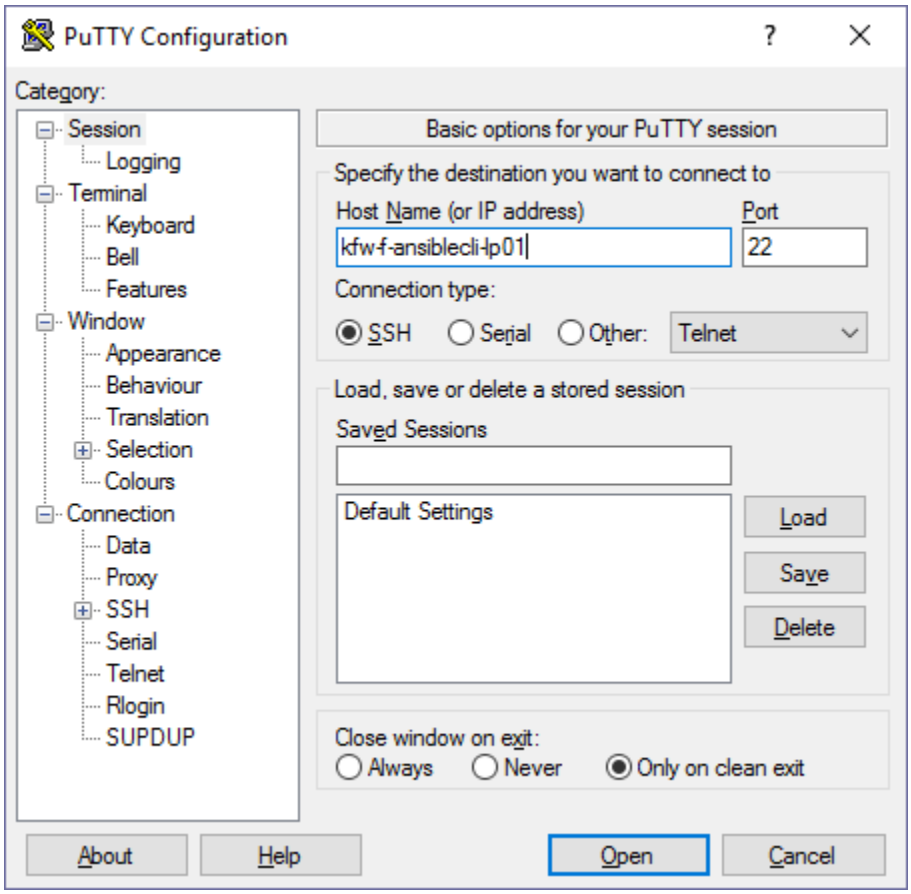
SU User

A_P_MobaXtermCitrix_100
B_P_01_Ansible_000000031028
B_P_00_Windows_000000012066
B_P_01_Ansible_000000027925
B_P_01_Ansible_000000027883

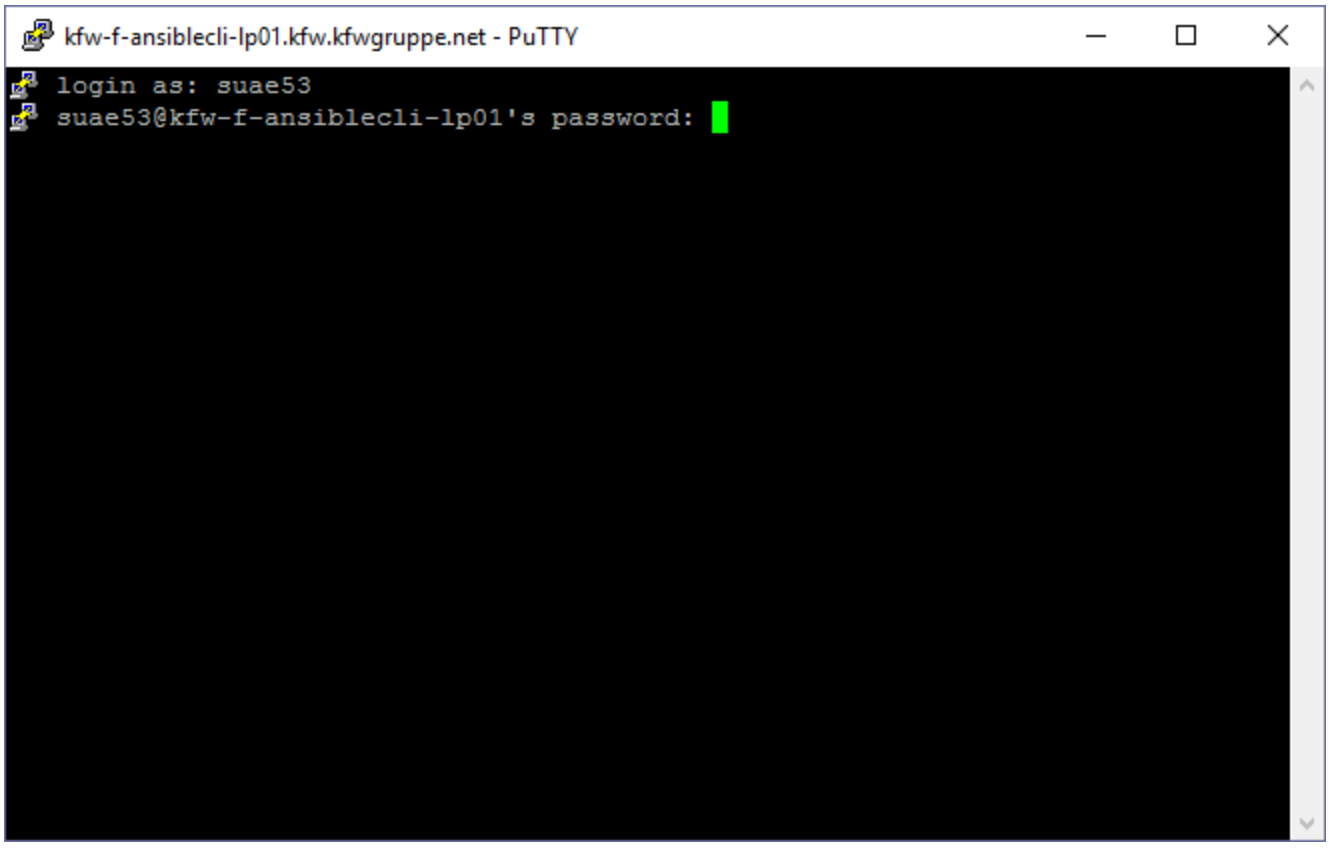
2.2 SSH-Key generieren auf dem Ansible-CLI-Server

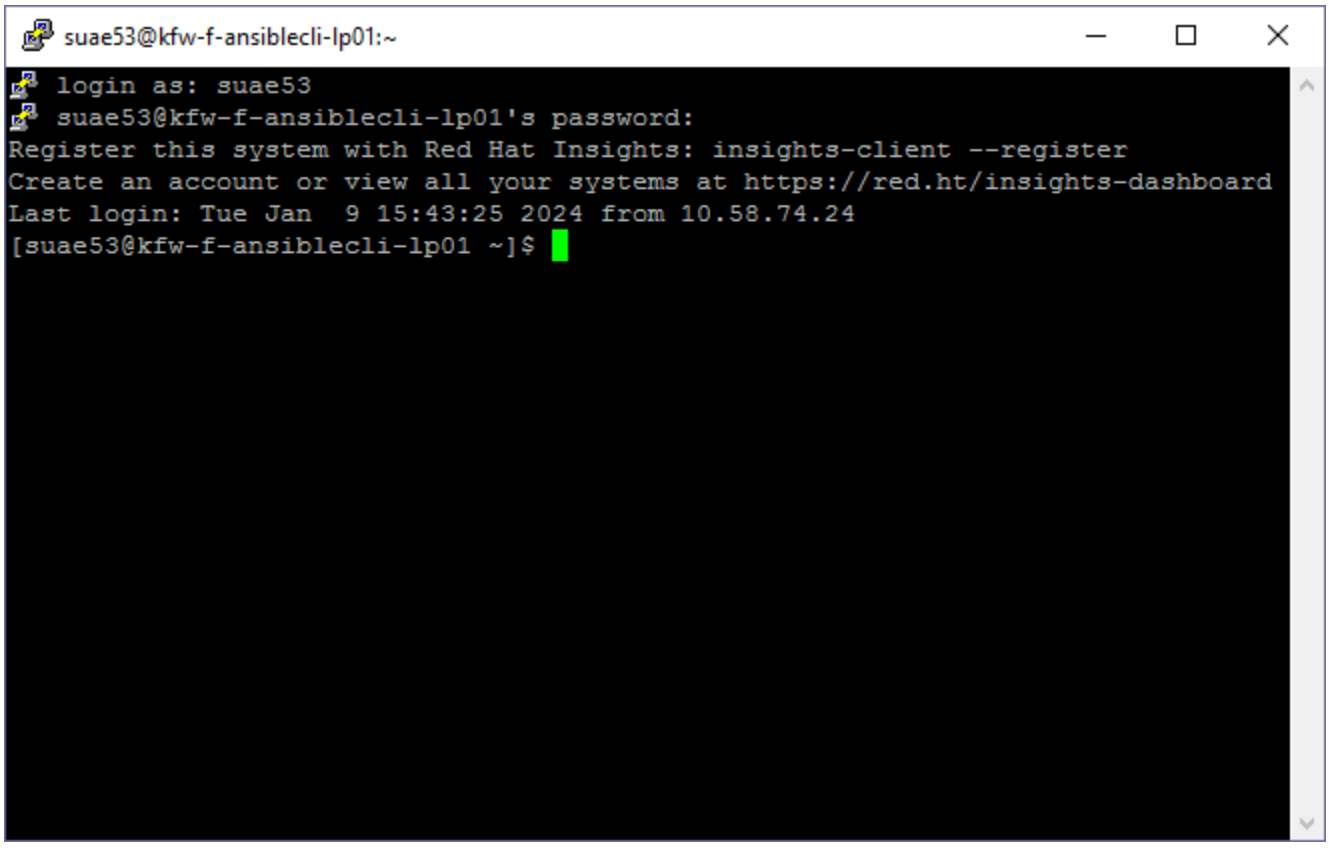
2.2.1 Über Putty Admin

Öffne über Putty Admin per SSH `kfw-f-ansiblecli-lp01`.



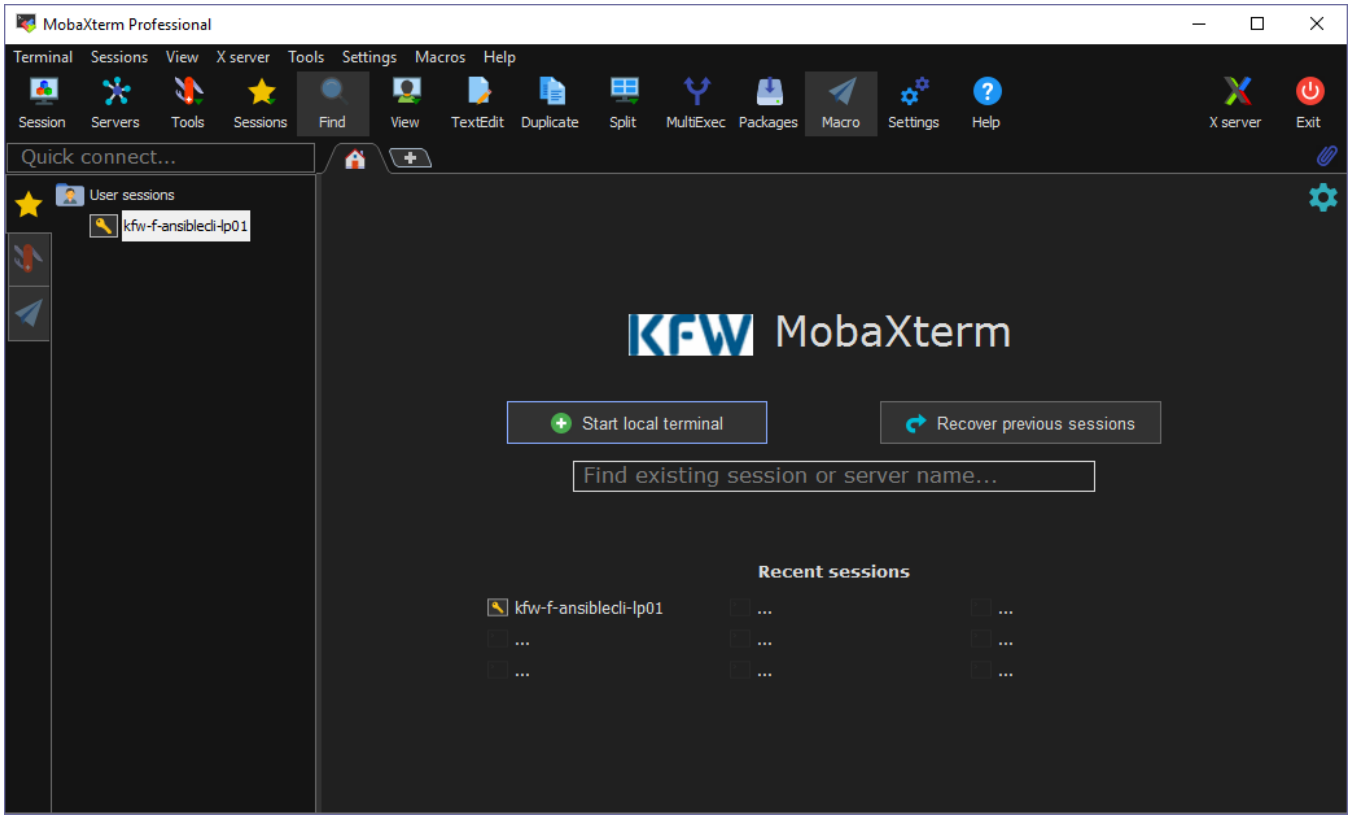
Dort kommt eine Benutzer Abfrage, gebe den Benutzername der Subidentität ein und drücke [Enter] und dann gebe das Passwort für die Subidentität ein.



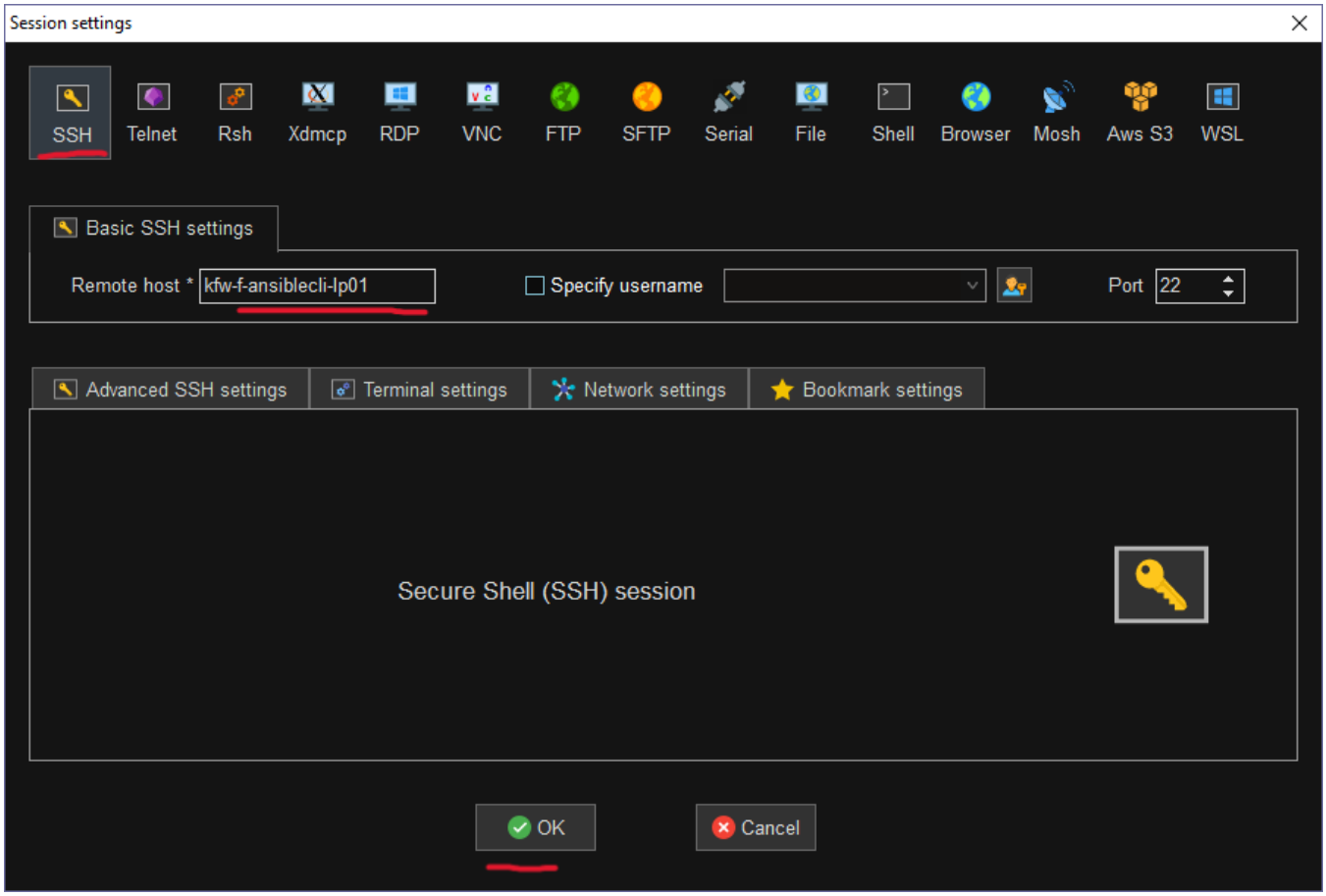


2.2.1 Über MobaXtermAdmin

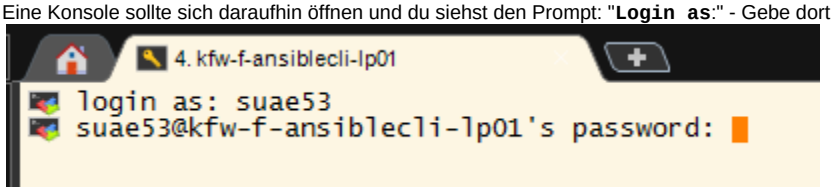
MobaXterm aufrufen und auf **Session** klicken.



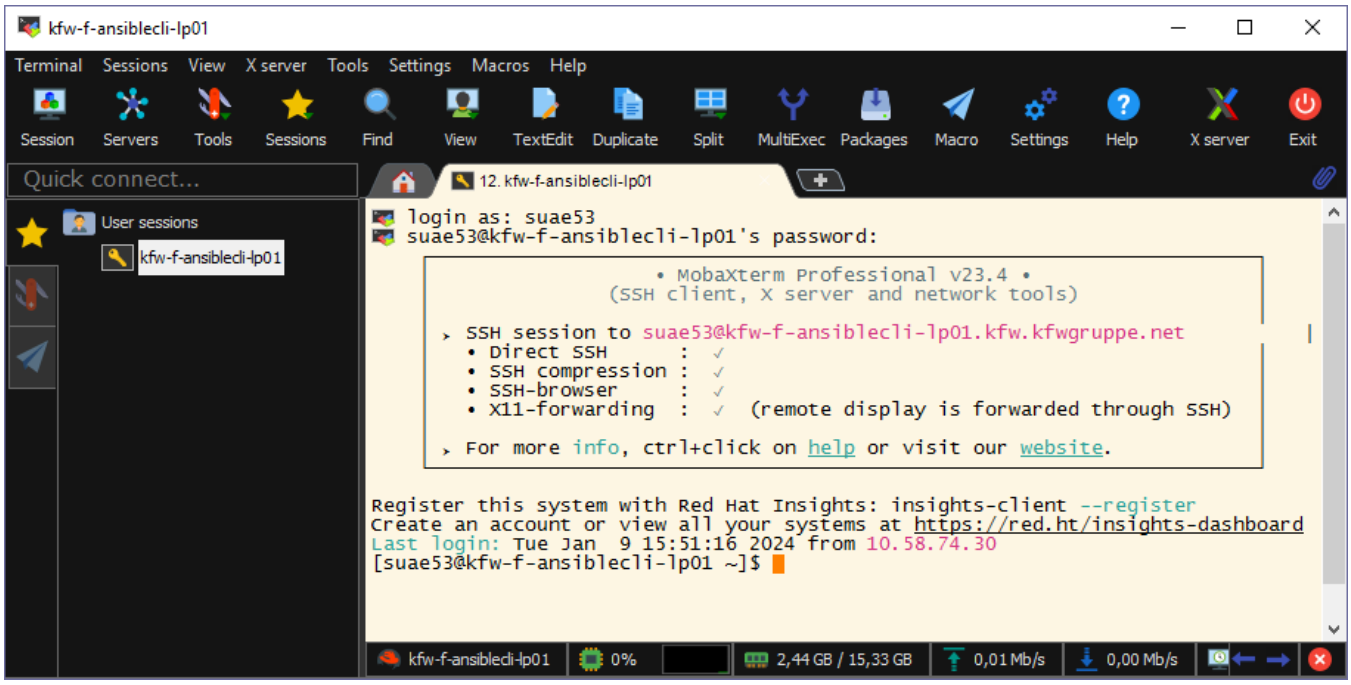
Es öffnet sich ein neues Fenster. **SSH** auswählen und unter **Remote host*** den Server, zu dem du dich verbinden willst, eingeben. Hier der Ansible-CLI-Server **`kfw-f-ansiblecli-lp01`.** Dann auf **OK.**



Eine Konsole sollte sich daraufhin öffnen und du siehst den Prompt: "**`Login as`**:" - Gebe dort deinen SU-User ein (***Bsp.: suae53***)



Daraufhin wirst du aufgefordert dein Passwort einzugeben. Wenn du schreibst, siehst du nicht, dass du tatsächlich das Passwort eintippst. Tippe dort dein SU-User Passwort. Tippe daraufhin **Enter** um dich einzuloggen.



2.2.2 SSH für Git konfigurieren

Weißt du nicht, was GIT ist? Lies dich hier schlau!

Du landest in einer Shell, befolge bitte jeden Schritt einzeln:

```
$ git config –global --edit
```

[i] drücken zum Bearbeiten

Den Inhalt hinter name= zum Standarduser-Namen ändern und hinter email= deine KfW-E-Mail-Adresse

[Esc][:][w][q][Enter] drücken zum Schreiben und Beenden

```
$ ssh-keygen -t rsa -b 4096
```

[Enter][Enter][Enter] Drücken

```
$ cat ~/.ssh/id_rsa.pub
```

Text darunter kopieren bis zur nächsten Shell-Input und in **GitLab** > **Profil** > **Edit Profil** > **SSH Keys** einfügen bei dem Feld für den Schlüssel

2.3 Ansible Automation Platform (AAP) und Gitlab (Git)

2.3.1 Erstellung von Playbooks in Gitlab for AAP

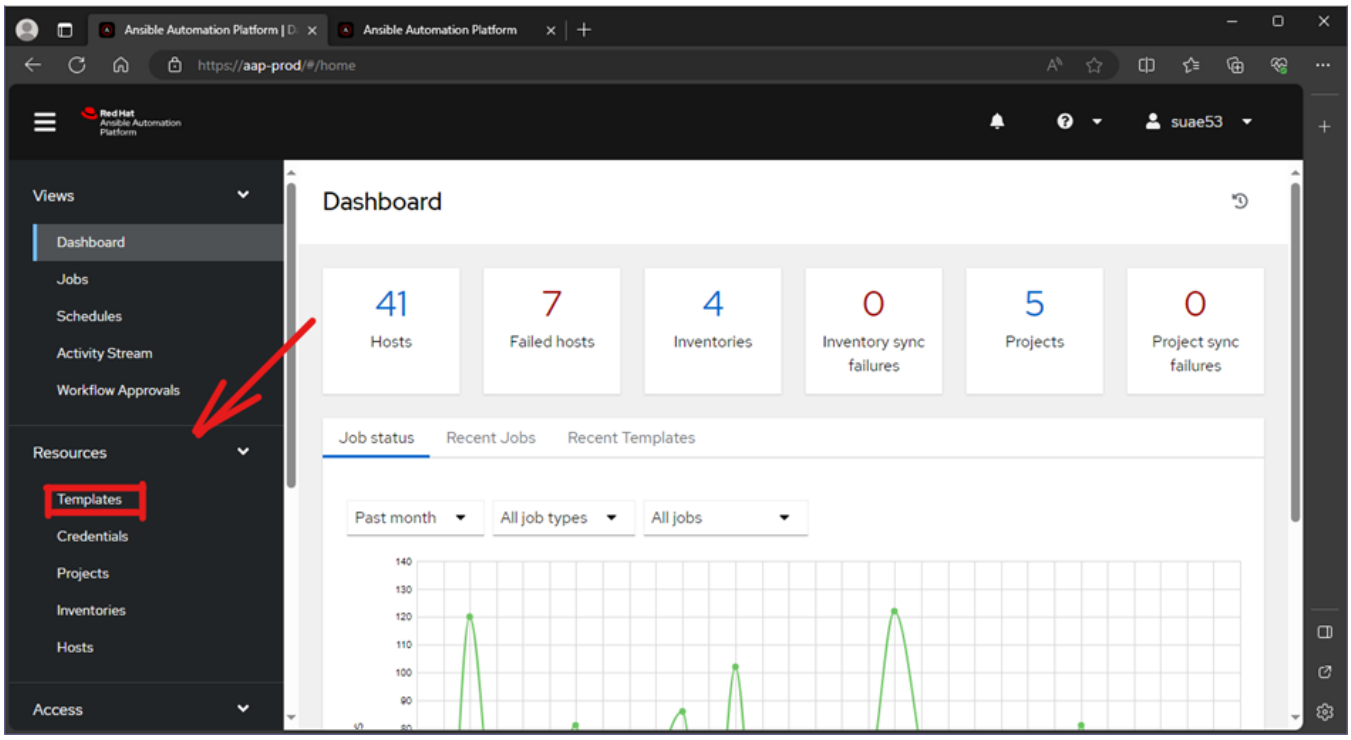
Die Erstellung von Playbooks erfolgt ausschließlich im **DEV-Branch** der jeweiligen Repositorys entsprechend der Projekte in der AAP.

In Informatica werden Playbooks, die nur für einen User angelegt werden, unter **Diverse-Informatica_Administration** > **Playbooks** > **user_defined** gespeichert. Namenskonvention: **sux123_name_des_playbooks** beachten

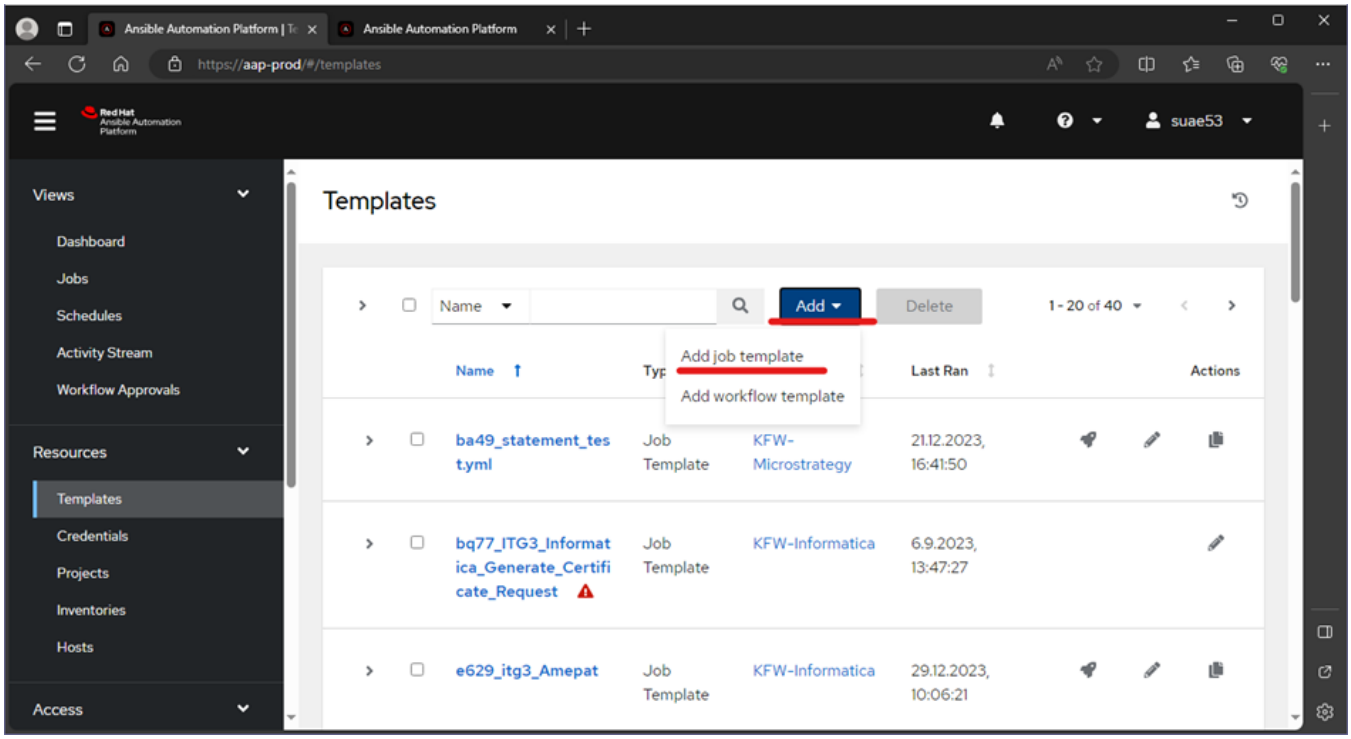
In MicroStrategy werden alle Playbooks unter **Diverse-MicroStrategy_Administration** > **Playbooks** gespeichert. Namenskonvention: **ITg3_MSTR_eigenerName_name_des_playbooks** beachten.

2.3.2 Erstellung von Templates der Jobs

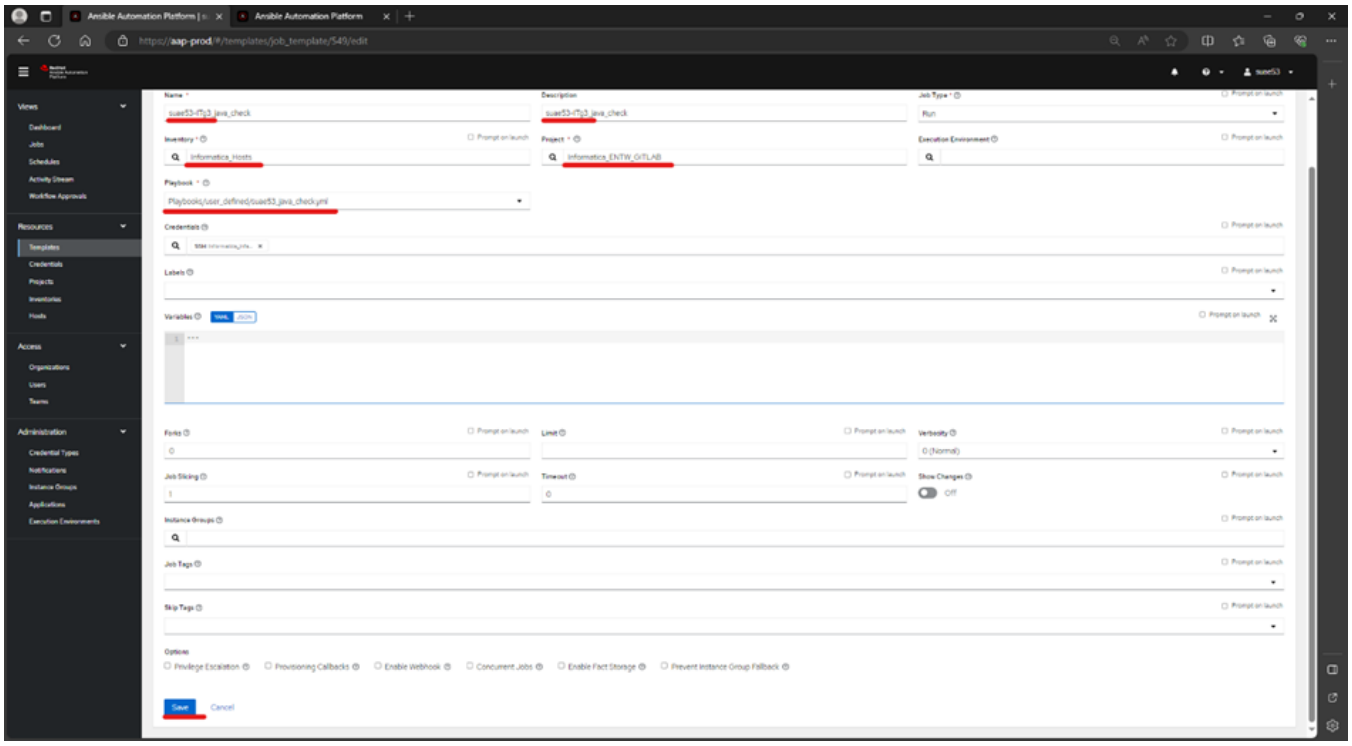
- Gehe zu **https://aap-prod/**, im **Edge Admin Browser** und logge dich mit dem SU User ein
- Öffne das Menü **Templates**



- Klicke auf **Add** und dann auf **Add job template**.

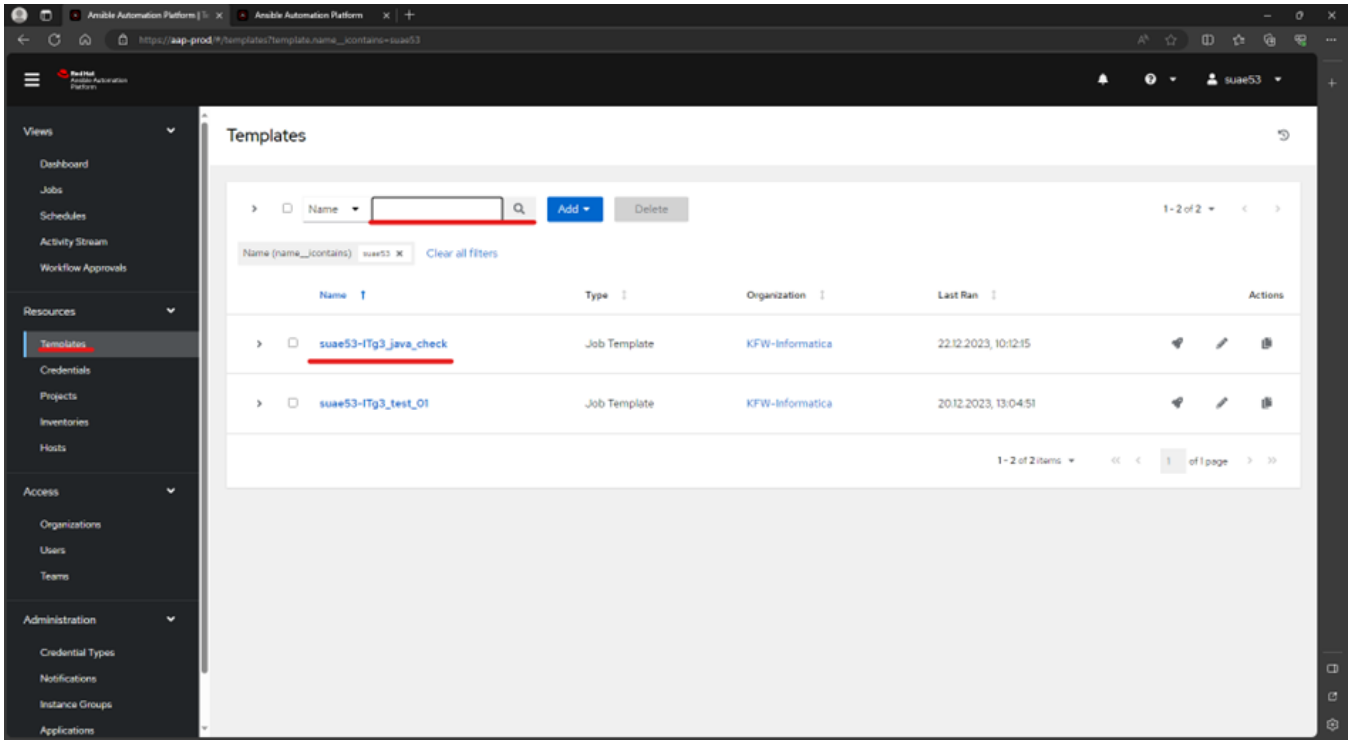


- Gib deinen **Usernamen** von der Subidentität und **Abteilung** unter **Name** und **Description** im Template an. Gib das entsprechende **Inventory** an, indem sich die Server befinden, die du ansprechen möchtest. Wähle das richtige **GIT Projekt,** in dem sich dein Playbook befindet und wähle dann dein **Playbook** aus dem Drop-Down Menü aus. Speicher dein Template.

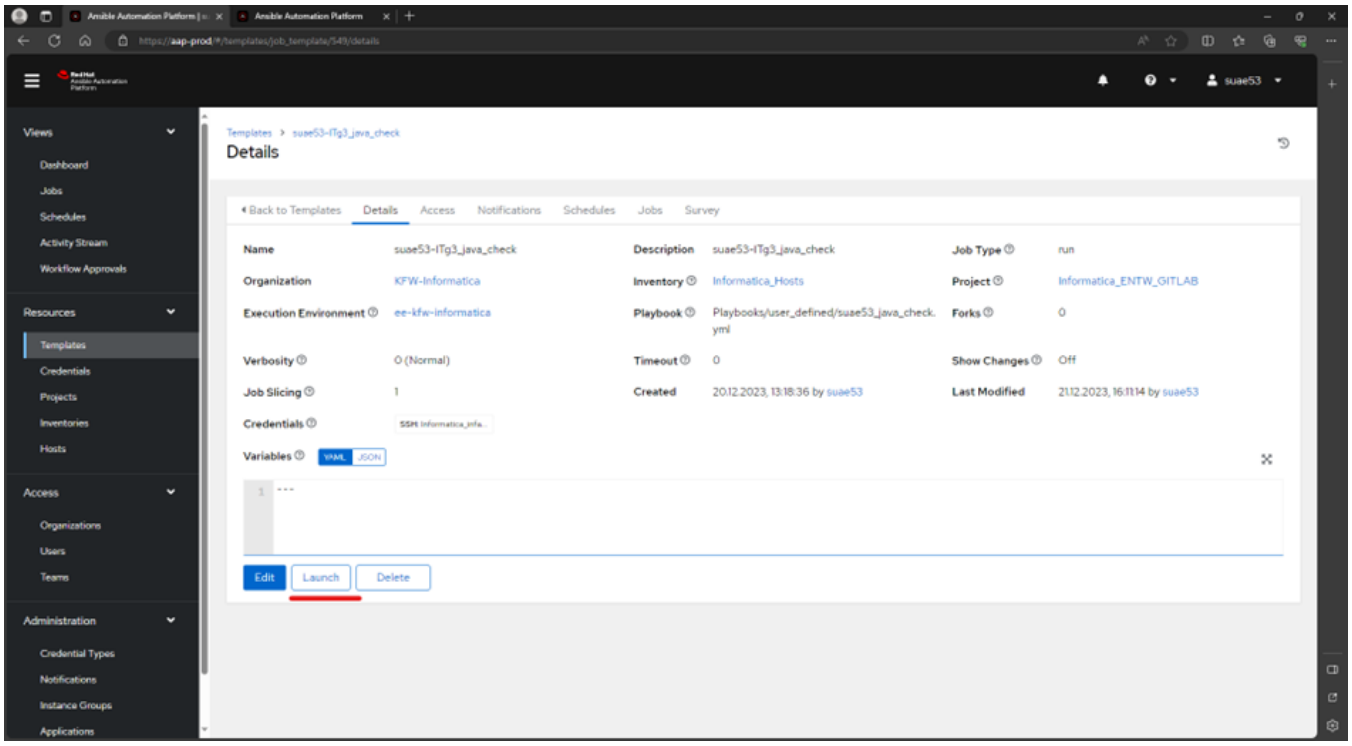


2.3.3 Starten eines Playbooks durch ein Job Template

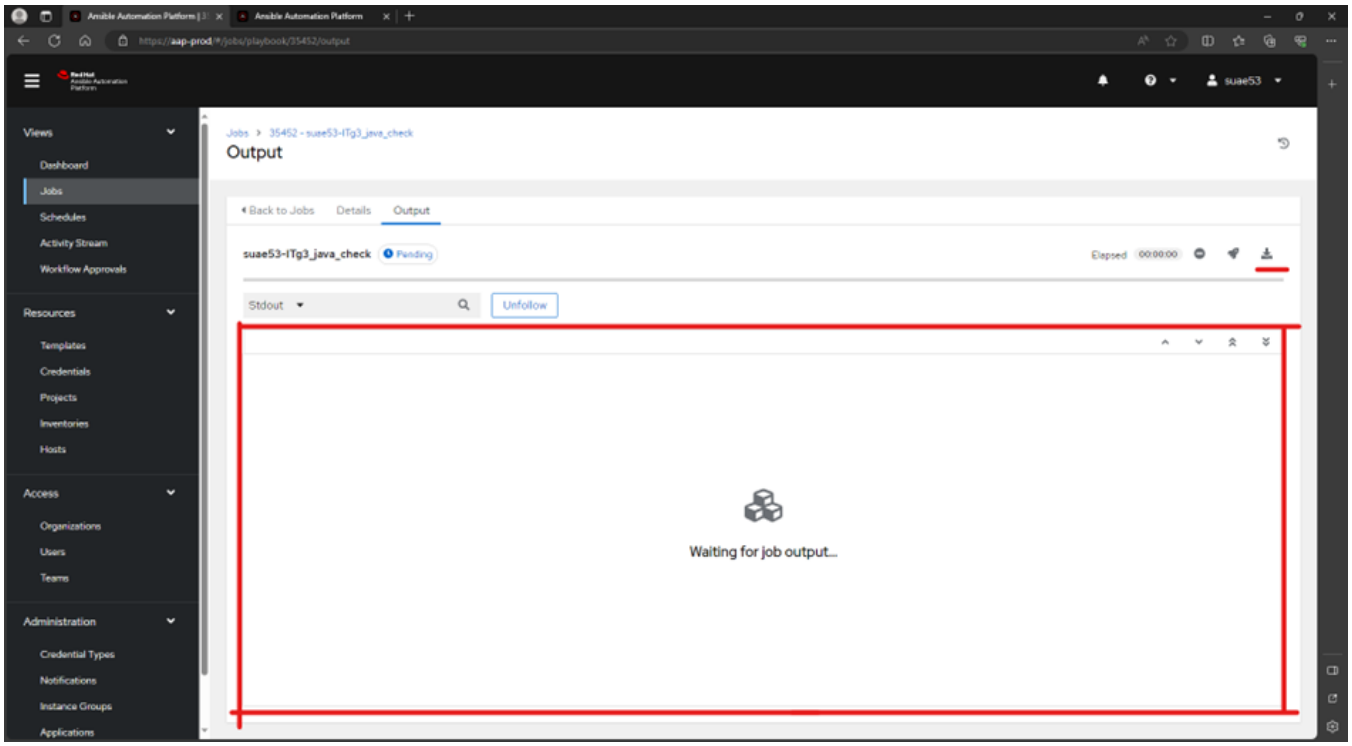
- Öffne das Job Template.



- Klicke auf **Launch**, um das Playbook zu starten.



- Wenn du das Log sehen kannst, startet es und du kannst die Infos dort herausholen durch direktes Kopieren oder durch Herunterladen. Sollte sich nach einer Weile nichts getan haben, einfach die Seite neu laden.



3 Inventories

3.1 Was sind Inventories?

Ein Inventory ist eine Liste von verwalteten Knoten oder Hosts, die Ansible bereitstellt und konfiguriert. Sprich ein Inventory enthält unsere Server, die wir zum Betreiben der Anwendungen brauchen.

3.2.1 Beispiel eines Inventories

Die Inventories beinhalten alle IP Adressen von Hosts / Server, auf die du zugreifen musst.

Unsere Inventories in Informatica findest du im **GIT** Projekt **Diverse-Informatica_Administration > hosts, hosts_prod**

Unsere Inventories in MicroStrategy findest du im **GIT** Projekt **Diverse-Microstrategy_Administration > conf > hosts_prod, hosts_nonprod**

```
[Platform1:children]
Sandbox_Platform1
Schulung_Platform1
Entwicklung_Platform1
Test_Platform1
PreProd_Platform1

[Sandbox_Platform1]
kfw-etl-as608 ansible_host=kfw-etl-as608.kfw.kfwgruppe.net

[Schulung_Platform1]
kfw-dis-as607 ansible_host=kfw-dis-as607.kfw.kfwgruppe.net

[Entwicklung_Platform1]
kfw-etl-ae623 ansible_host=kfw-etl-ae623.kfw.kfwgruppe.net
kfw-etl-ae515 ansible_host=kfw-etl-ae515.kfw.kfwgruppe.net

[Test_Platform1]
kfw-etl-at623 ansible_host=kfw-etl-at623.kfw.kfwgruppe.net
kfw-etl-at517 ansible_host=kfw-etl-at517.kfw.kfwgruppe.net
kfw-etl-at531 ansible_host=kfw-etl-at531.kfw.kfwgruppe.net

[PreProd_Platform1]
kfw-etl-ax515 ansible_host=kfw-etl-ax515.kfw.kfwgruppe.net

[Platform1:vars]

ansible_python_interpreter=/opt/freeware/bin/python3
```

Unsere Inventories sind im INI Format geschrieben, daher sind die [Gruppennamen] in Klammern geschrieben. Grundsätzlich kann man Inventories auch im YAML Format schreiben. Platform1 ist eine Gruppe, in der sich weitere Gruppen befinden. Die child Gruppen enthalten dann die einzelnen Server.

---

4 Playbook

4.1 Was ist sind Playbooks, Plays und Tasks?

Ein **Playbook** ist eine Liste von Abläufen, die die Reihenfolge festlegen, in der Ansible Operationen von oben nach unten durchführt, um ein Gesamtziel zu erreichen. Ansible Playbooks werden zur Orchestrierung von IT-Prozessen verwendet. Bei einem Playbook handelt es sich um eine YAML-Datei, die ein oder mehrere Plays enthält und dazu verwendet wird, den gewünschten Zustand eines Systems zu definieren.

Plays bestehen aus einer bestimmten Abfolge von Aufgaben, die für die ausgewählten Hosts aus Ihrer Ansible-Inventory-Datei ausgeführt werden.

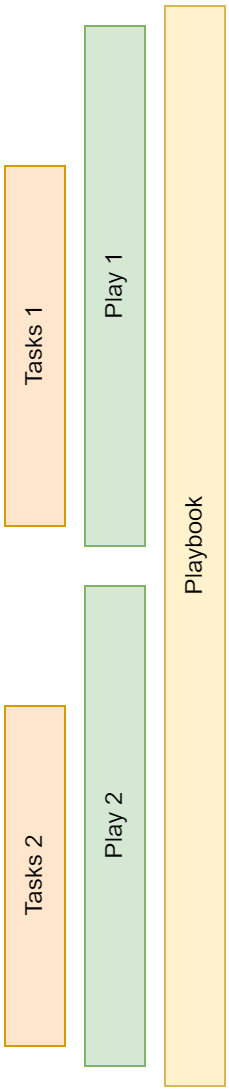
Aufgaben (**Tasks**) sind dabei die Bestandteile, die sich zu einem Play zusammensetzen und Ansible-Module aufrufen. Diese Aufgaben werden in einem Play in der Reihenfolge ausgeführt, in der sie geschrieben sind.

```yaml
---
- hosts: webservers
  remote_user: root

  tasks:
  - name: ensure apache is at the latest version
    yum:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf

- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that postgresql is started
    service:
      name: postgresql
      state: started
```



4.2

4.3

4.4 Beispiel eines Playbooks

Hier sieht man ein einfaches Playbook

```
---
- name: java_check
  hosts: all
  gather_facts: true

  tasks:
  - name: get java version
    ansible.builtin.shell:
      lslpp -L | grep 'Java Runtime' | awk '{print $2}'
    register: java_version
    no_log: yes

  - name: Print hostname and java version
    ansible.builtin.debug:
      msg: "Hostname: {{  ansible_hostname  }}, Java Version: {{  java_version.stdout  }}"
```

Bei **- name:** vergibt man den Namen des Plays oder einer Task im Playbook. Dies ist optional, aber extrem nützlich. Ansible zeigt im Output diesen Namen an. Benutze Namen, die beschreibend sind für die Tätigkeit.

Unter **hosts:** gibt man die Hostnamen der Server an, auf denen man das Skript ausführen möchte (s. Abschnitt Patterns)

```
- name: suae53_java_check
  hosts: all,!kfw-etl-at637
```

gather_facts ist ein boolescher Wert, der angibt, ob das Playbook automatisch die setup task laufen lässt, um Fakten zu sammeln, die später verwendet werden können. Der Default ist true, könnte in diesem Fall also auch weggelassen werden.

```
gather_facts: true
```

Das Stichwort **tasks** leitet den Teil im Skript ein, in dem sich alle Tasks befinden.

```
  tasks:
```

Dies ist die erste Task in unserem Skript. Bei **name** wird wieder eine beschreibender Name für unsere Task gewählt. Gefolgt von einem Modul, mit dem wir ausführen können was wir vorhaben. Hier ist es das shell module **ansible.builtin.shell**, welches einen shell Befehl als Input erwartet. Mit unserem shell Befehl fragen wir die Java Version auf dem Server ab und speichern den Output in der Variable **java_version** durch **register.**

```
 - name: get java version
   ansible.builtin.shell:
      lslpp -L | grep 'Java Runtime' | awk '{print $2}'
   register: java_version
   no_log: yes
```

In dieser Task lassen wir uns den Hostname und die Java Version, welche wir zuvor in der Variable gespeichert haben, über das Debug Modul ausgeben.

```
- name: Print hostname and java version
  ansible.builtin.debug:
      msg: "Hostname: {{  ansible_hostname  }}, Java Version: {{  java_version.stdout  }}"
```

4.3.1 Output des Beispiel Playbooks

Das Beispiel Output wurde beim Ausführen begrenzt auf den Sandbox Server, um das Beispiel übersichtlicher zu gestalten

Ansible Automation Platform Output

```
Enter passphrase for /runner/artifacts/37043/ssh_key_data:
Identity added: /runner/artifacts/37043/ssh_key_data (infaadm@kfw-etl-as608)
PLAY [suae53_java_check] *******************************************************

TASK [Gathering Facts] *********************************************************
ok: [kfw-etl-as608]

TASK [get java version] ********************************************************
changed: [kfw-etl-as608]

TASK [Print hostname and java version] *****************************************
ok: [kfw-etl-as608] => {
    "msg": "Hostname: kfw-etl-as608, Java Version: 8.0.0.710"
}

PLAY RECAP *********************************************************************
kfw-etl-as608              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

4.4 CLI (console line interface)

```
$ ansible-playbook -i /path/my_invetory_file -u my_connection_user -k -f 3 -T 30 -t my_tag -M /path/to
/my_modules -b -K my_playbook.yml
```

Lädt `my_playbook.yml` aus dem aktuellen Arbeitsverzeichnis und:

- `-i my_inventory_file` im angegebenen Pfad für das Inventar, um dem Muster zu entsprechen.
- `-u` stellt eine SSH-Verbindung als `my_connection_user` her.
- `-k` fordert nach einem Passwort, das dann für die SSH-Authentifizierung bereitgestellt wird.
- `-f` weist 3 Forks zu.
- `-T` setzt ein Zeitlimit von `30` Sekunden.
- `-t` führt nur Aufgaben aus, die mit dem Tag `my_tag` markiert sind.
- `-M` lädt lokale Module aus `/path/to/my/modules`.
- `-b` führt mit erhöhten Berechtigungen aus (verwendet become).
- `-K` fordert den Benutzer nach dem Become-Passwort.

4.5 Wie spreche ich die Hosts in einem Skript an? - Patterns

Um mehr Freiheit in Selektion in Hosts zu haben kann man über Patterns diese genauer festgelegen. Z.B können in der gezeigten Tabelle alle Gruppen von Hosts ausgewählt werden mit dem Schlüsselwort all oder mit einem *, zudem können mehrere Gruppen ausgewählt oder ausgeschlossen werden.

Description	Pattern(s)	Targets
Alle hosts	all (oder *)	
Ein host	kfw-etl-ae128	
Multiple hosts	kfw-etl-ae128:kfw-etl-ae618 (oder kfw-etl-ae128,kfw-etl-ae128	
Eine Gruppe	Entwicklung_Platform2	
Multiple Gruppen	Entwicklung_Platform2:Test_Platform2	alle Hosts in Entwicklung_Platform2 plus alle Hosts in Test_Plattform2
Ausschließende (Excluding) Gruppen	Platform2:!PreProd_Platform2	alle Hosts in Platform2, außer die Hosts in PreProd_Platform2
Logisches Und	Entwicklung_Platform2:&Test_Platform2	alle Hosts die in Entwicklung_Platform2 UND Test_Platform2 sind

4.6 Wie nutze ich Variablen?

4.6.1 Valide Variablen verwenden

Valid Variable names	not valid
kfw	*kfw, Python keywords wie `async` und `lambda`
kfw_env	playbook keywords wie `environment`
kfw5, _kfw	5kfw, 12

4.6.2 Variablen definieren

Variablen können im Standard YAML Format im Variablen Bereich des Skripts definiert werden. Zum Beispiel:

```
---
- name: Beispiel
  hosts: all
  gather_facts: true

  vars:
    beispiel: 'Hello World'
```

4.6.3 Referenzieren auf Variablen

Um auf die vorher definierte Variable zu referenzieren wird "{{ }}", einschließlich der ", benutzt.

```
---
- name: Beispiel
  hosts: all
  gather_facts: true

  vars:
    beispiel: 'Hello World'

  tasks:
  - name: Print var
    debug:
      msg: "In meiner Variable Beispiel steht: {{beispiel}}!!!"
```

---

5 Module

5.1 Was Sind Module?

Module sind die Hauptbestandteile von Playbooks und sind ein Type von Plugins.

5.2 Häufig benutzte Module bei uns

Hier gehts zur Extra Seite: Ansible Modules - ITg3-Teambereich - on Prem (kfwgruppe.net)

5.3 Aktivierung von eigenen Modulen

5.3.1 env-Variable

Dafür muss man den Ablage-Ordner-Path in $ANSIBLE_LIBRARY angeben.

5.3.2 Im Ansible-Module-Verzeichnis im Home-Verzeichnis

Einfach im ~/.ansible/plugins/modules/ Verzeichnis ablegen.

5.3.3 Im globalen Ansible-Module-Verzeichnis

Einfach im /usr/share/ansible/plugins/modules/ Verzeichnis ablegen, kann Superuser-Rechte erfordern.

---

6 Roles

Infos die wir noch irgendwo unterbringen wollen

- [ ] Warum nutzen wir keine Roles ?
- [ ] For many modules, the `state` parameter is optional. Different modules have different default settings for `state`, and some modules support several `state` settings. Explicitly setting `state: present` or `state: absent` makes playbooks and roles clearer. - ????????? ist das wichtig für uns ohne roles
- [ ] Even with task names and explicit state, sometimes a part of a playbook or role (or inventory/variable file) needs more explanation. Adding a comment (any line starting with #) helps others (and possibly yourself in future) understand what a play or task (or variable setting) does, how it does it, and why
- [ ] Output mit aufnehmen
- [ ]

Lieber die alte Dokumentation verwenden: Alte Ansible Dokumentation Öffnen

---

7 Quellen

- Ansible Documentation
- Introduction to Ansible — Ansible Documentation
- Patterns: targeting hosts and groups — Ansible Documentation
- Ansible playbooks — Ansible Documentation
- Creating a playbook — Ansible Documentation
- Using Variables — Ansible Documentation
- Module Index — Ansible Documentation
- How to build your inventory — Ansible Documentation
- Ansible - ITg3-Teambereich - on Prem (kfwgruppe.net)