Abgabe 3.md

Aufgabe 3a

Konfigurieren Sie die virtuellen Maschinen aus Aufgabe 1 mit Ansible. Verwenden Sie hierfür folgende Vorgaben:

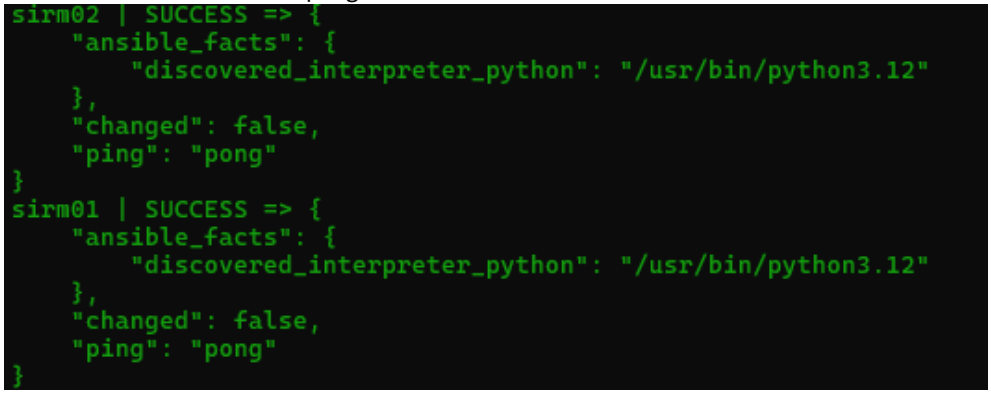
- Erstellen Sie eine ansible.cfg. Erläutern Sie, welche Standardkonfiguration sie hier sinnvollerweise setzen sollten.
  - ansible.cfg

```
[defaults]
inventory = ./hosts.yml           // Legt die Inventory Datei fest
remote_user = osboxes             // Benutzer, mit dem sich per SSH verbunden wird
host_key_checking = False         // SSH fragt, ob man sich verbinden möchte. Das wird so übersprungen
interpreter_python = auto_silent  // Unterdrückt die Meldungen, dass Python automatisch gefunden wurd

[privilege_escalation]
become = True                     // Jeder Task läuft über sudo, wenn nicht anders definiert
become_method = sudo              // Wie Ansible root-Rechte bekommen soll
```

- Erstellen Sie eine Inventory Datei und gruppieren Sie ihre Maschinen so, wie sie es für sinnvoll erachten.
  - hosts.yml

```
all:
hosts:
  sirm01:
    ansible_host: 10.11.12.2
  sirm02:
    ansible_host: 10.11.12.3

children:
  webservers:
    hosts:
      sirm01:
      sirm02:
    vars:
      http_port: 8081

  dbservers:
    hosts:
      sirm01:
    vars:
      db_port: 3306
```

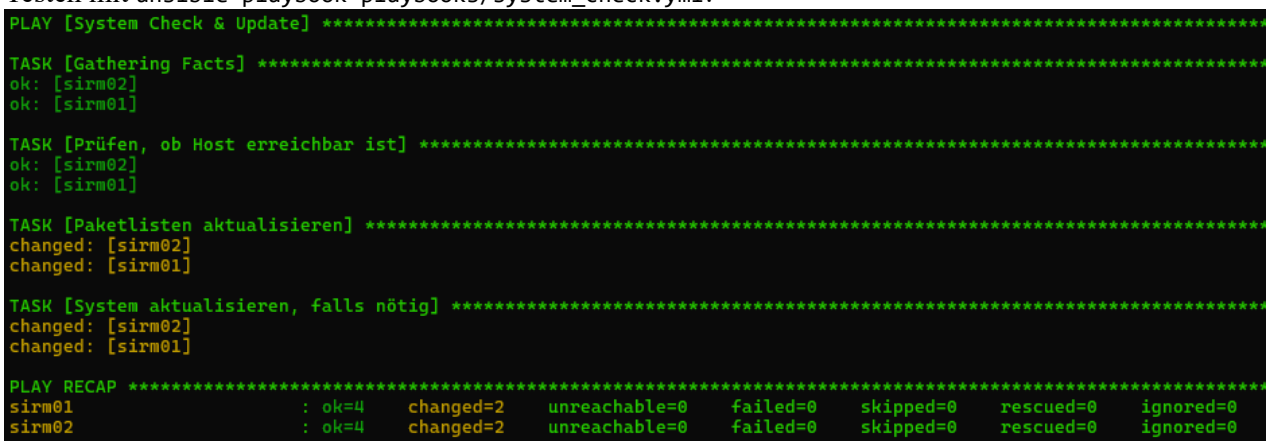
  - Test mit `ansible all -m ping`:



- Schreiben Sie ein Playbook, das zeigt, dass alle Maschinen erreichbar sind und das System aktuell ist bzw. aktualisiert, wenn es nicht aktuell ist.
  - system_check.yml

```
---
- name: System Check & Update
  hosts: all
  become: yes
  gather_facts: yes

  tasks:

  - name: Prüfen, ob Host erreichbar ist
    ping:

  - name: Paketlisten aktualisieren
    apt:
    update_cache: yes
    register: update_result

  - name: System aktualisieren, falls nötig
    apt:
    upgrade: dist
    autoremove: yes
    when: update_result is changed
```

- Testen mit `ansible-playbook playbooks/system_check.yml`:



- Schreiben Sie ein Playbook, das jeweils Nginx auf den VMs web01 und web02 installiert und über Port 8081 erreichbar ist. Lassen sie auf der Startseite den Namen ihrer VM ausgeben.
  - nginx_install.yml

```
---
- name: Install and configure Nginx on web servers
  hosts: webservers
  become: yes
  gather_facts: yes

  tasks:

  - name: Install Nginx
    apt:
    name: nginx
    state: present
    update_cache: yes

  - name: Ensure Nginx is running and enabled
    service:
    name: nginx
    state: started
    enabled: yes

  - name: Change default Nginx port to 8081
    lineinfile:
    path: /etc/nginx/sites-available/default
    regexp: '^\s*listen\s+80'
    line: "        listen {{ http_port }};"
    backrefs: yes
    notify: Restart Nginx

  - name: Set start page with hostname
    copy:
    dest: /var/www/html/index.html
    content: "Hostname: {{ inventory_hostname }}!"
```

```
handlers:
- name: Restart Nginx
  service:
  name: nginx
  state: restarted
```

- Testen mit `ansible-playbook playbooks/nginx_install.yml`:

```
PLAY [Install and configure Nginx on web servers] ******************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [sirm02]
ok: [sirm01]

TASK [Install Nginx] ***********************************************************************************
changed: [sirm01]
changed: [sirm02]

TASK [Ensure Nginx is running and enabled] **************************************************************
ok: [sirm01]
ok: [sirm02]

TASK [Change default Nginx port to 8081] ***************************************************************
changed: [sirm02]
changed: [sirm01]

TASK [Set start page with hostname] ********************************************************************
changed: [sirm02]
changed: [sirm01]

RUNNING HANDLER [Restart Nginx] ************************************************************************
changed: [sirm01]
changed: [sirm02]

PLAY RECAP *********************************************************************************************
sirm01                     : ok=6    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
sirm02                     : ok=6    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- Webserver erreichbar und Hostname wird angezeigt:

```
osboxes@sirm01:~$ curl localhost:8081
Hostname: sirm01osboxes@sirm01:~$
```

- Es soll mittels eines Playbooks MariaDB auf der VM db01 installiert werden. Der Port 3306 soll freigegeben werden
  - mariadb_install.yml

```
---
- name: Install and configure MariaDB on DB server
  hosts: dbservers
  become: yes
  gather_facts: yes

  tasks:

  - name: Update apt cache
    apt:
    update_cache: yes

  - name: Install MariaDB server
    apt:
    name: mariadb-server
    state: present

  - name: Ensure MariaDB service is running
    service:
    name: mariadb
    state: started
    enabled: yes

  - name: Allow incoming traffic on MariaDB port
    ufw:
    rule: allow
    port: "{{ db_port }}"
    proto: tcp

  - name: Ensure UFW is enabled
    ufw:
    state: enabled
```

- Testen mit `ansible-playbook playbooks/mariadb_install.yml`:

```
PLAY [Install and configure MariaDB on DB server] ******************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [sirm01]

TASK [Update apt cache] ********************************************************************************************************
changed: [sirm01]

TASK [Install MariaDB server] **************************************************************************************************
changed: [sirm01]

TASK [Ensure MariaDB service is running] ***************************************************************************************
ok: [sirm01]

TASK [Allow incoming traffic on MariaDB port] **********************************************************************************
changed: [sirm01]

TASK [Ensure UFW is enabled] ***************************************************************************************************
changed: [sirm01]

PLAY RECAP *********************************************************************************************************************
sirm01                     : ok=6    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- MariaDB erreichbar und Port freigegeben: `ansible dbservers -m shell -a "systemctl status mariadb"`

```
sirm01 | CHANGED | rc=0 >>
● mariadb.service - MariaDB 10.11.13 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
     Active: active (running) since Sat 2026-01-10 07:27:00 EST; 1min 1s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 27222 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 14945)
     Memory: 78.9M (peak: 82.1M)
        CPU: 754ms
     CGroup: /system.slice/mariadb.service
             └─27222 /usr/sbin/mariadbd

Jan 10 07:27:00 sirm01 mariadbd[27222]: 2026-01-10  7:27:00 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
Jan 10 07:27:00 sirm01 mariadbd[27222]: 2026-01-10  7:27:00 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
Jan 10 07:27:00 sirm01 mariadbd[27222]: 2026-01-10  7:27:00 0 [Note] Server socket created on IP: '127.0.0.1'.
Jan 10 07:27:00 sirm01 mariadbd[27222]: 2026-01-10  7:27:00 0 [Note] InnoDB: Buffer pool(s) load completed at 260110  7:27:00
Jan 10 07:27:00 sirm01 mariadbd[27222]: 2026-01-10  7:27:00 0 [Note] /usr/sbin/mariadbd: ready for connections.
Jan 10 07:27:00 sirm01 mariadbd[27222]: Version: '10.11.13-MariaDB-0ubuntu0.24.04.1'  socket: '/run/mysqld/mysqld.sock'  port: 3306  Ubuntu 24.04
Jan 10 07:27:00 sirm01 systemd[1]: Started mariadb.service - MariaDB 10.11.13 database server.
Jan 10 07:27:00 sirm01 /etc/mysql/debian-start[27240]: Upgrading MariaDB tables if necessary.
Jan 10 07:27:00 sirm01 /etc/mysql/debian-start[27251]: Checking for insecure root accounts.
Jan 10 07:27:00 sirm01 /etc/mysql/debian-start[27256]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
```

Aufgabe 3b

Ansible ermöglicht die Verwendung von Templates auf Basis von Jinja2.

- Erläutern Sie, was solche Templates sind, und wozu und in welcher Weise Templates in Ansible genutzt werden können.
  - Templates sind eine Art Textvorlagen mit Platzhaltern und Logik, die beim Ausführen dynamisch mit Daten gefüllt werden.
  - Inhalt ist eine Konfigurationsdatei, die um Logik-Elemente erweitert wurde.
  - Sie können für Skripte, Configs für Services, Environment-Files und weiteres benutzt werden.
  - Vorteil gegenüber Variablen ist, dass ganze Dateien generiert werden können statt einzelne Zeilen austuschen zu müssen.

Verwendung von Jinja2-Templates.

- Sorgen Sie dafür, dass mittels Templates der Port der Konfiguration über eine Variable im Playbook frei wählbar ist.
  - port_nginx.j2

```
server {
  listen {{ http_port }};
  root /var/www/html;
  index index.html;
}
```

- Überprüfen Sie mittels Ansible assert, ob die Maschinen über den entsprechenden Port 8081 erreichbar sind
  - Abändern des Playbooks `nginx_install.yml` in neuer Datei: nginx_install_with_template.yml

```
---
- name: Install and configure Nginx on web servers
  hosts: webservers
  become: yes
  gather_facts: yes

  tasks:
```

```
- name: Install Nginx
  apt:
  name: nginx
  state: present
  update_cache: yes

- name: Ensure Nginx is running and enabled
  service:
  name: nginx
  state: started
  enabled: yes

- name: Deploy Nginx configuration from template
  template:
  src: ../templates/port_nginx.j2
  dest: /etc/nginx/sites-available/default
  notify: Restart Nginx

- name: Set start page with hostname
  copy:
  dest: /var/www/html/index.html
  content: "Willkommen auf {{ inventory_hostname }}!"

- name: Check that port is reachable locally
  wait_for:
  host: 127.0.0.1
  port: "{{ http_port }}"
  timeout: 5
  register: port_check

- name: Assert port is open
  assert:
  that:
  - port_check is succeeded
  fail_msg: "Port {{ http_port }} auf {{ inventory_hostname }} ist nicht erreichbar!"
  success_msg: "Port {{ http_port }} auf {{ inventory_hostname }} ist erreichbar."

handlers:
- name: Restart Nginx
  service:
  name: nginx
  state: restarted
```

- Testen mit `ansible-playbook playbooks/nginx_install_with_template.yml`:
- Anmerkung: SSH funktioniert scheinbar auf der ersten VM nicht mehr, ohne Änderungen vorgenommen zu haben.

```
PLAY [Install and configure Nginx on web servers] ******************************************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************************************************************
ok: [sirm02]
[ERROR]: Task failed: Failed to connect to the host via ssh: ssh: connect to host 10.11.12.2 port 22: Connection timed out
fatal: [sirm01]: UNREACHABLE! => {"changed": false, "msg": "Task failed: Failed to connect to the host via ssh: ssh: connect to host 10.11.12.2 port 22: Connection timed out", "unreachable": true}

TASK [Install Nginx] ***********************************************************************************************************************************************************************************************
ok: [sirm02]

TASK [Ensure Nginx is running and enabled] *************************************************************************************************************************************************************************
changed: [sirm02]

TASK [Deploy Nginx configuration from template] ********************************************************************************************************************************************************************
changed: [sirm02]

TASK [Set start page with hostname] ********************************************************************************************************************************************************************************
changed: [sirm02]

TASK [Check that port is reachable locally] ************************************************************************************************************************************************************************
ok: [sirm02]

TASK [Assert port is open] *****************************************************************************************************************************************************************************************
ok: [sirm02] => {
    "changed": false,
    "msg": "Port 8081 auf sirm02 ist erreichbar."
}

RUNNING HANDLER [Restart Nginx] ************************************************************************************************************************************************************************************
changed: [sirm02]

PLAY RECAP *********************************************************************************************************************************************************************************************************
sirm01                     : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
sirm02                     : ok=8    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- Über Assert ist zu sehen, dass der Port erreichbar ist.