



KN02: Automatisierung / Cloud-init

- A) Cloud-init Datei Verstehen
- B) Cloud-init Datei verwenden.
- C) Cloud-init Datei ändern und erstellen

KN02: Automatisierung / Cloud-init

Beachten Sie die [allgemeinen Informationen zu den Abgaben](#).

In dieser Kompetenz werden Sie die Schritte aus der Kompetenz KN01 automatisieren und dabei ein Beschreibungsformat erlernen.

Die wichtigsten Informationen zu Cloud-init und YAML [finden Sie in der Theorie](#).

A) Cloud-init Datei Verstehen

Laden Sie die [cloud-init Datei für den Web Sever](#) herunter. Erklären Sie alle Zeilen der Cloud-init Datei in folgendem Schema. **Achtung:** Viele Werte sind nicht definiert von cloud-init, sondern von Linux/Ubuntu. z.B. `sudo: <sudo-Regeln>`. Die Sudo-Regeln sind definiert in Linux (sudoers-file).

```
users: # Collection von Benutzern
- name: ubuntu # Der Benutzername des ersten Benutzers
  sudo: ALL=(ALL) NOPASSWD:ALL # sudo-Regeln für diesen Benutzer
  #...
  #...
```

B) Cloud-init Datei verwenden.

Verwenden Sie nun die Cloud-init Datei aus A), um eine Multipass Instanz zu starten. Sie benötigen die bisherigen Instanzen nicht mehr und können diese Löschen, wenn Sie möchten.

```
multipass launch 22.04 --name web --cloud-init <pfad>\cloud-init-web.yaml
```

Sie werden bemerken, dass die Installation ihrer Instanz länger dauert als vorher. Dafür benötigen Sie keine manuellen Schritte und die Installation läuft im Hintergrund.

Zeigen Sie, dass ihr Webserver läuft, indem Sie (mit Screenshot) zeigen dass

1. der Apache-Service läuft (`sudo systemctl status apache2.service`)
2. Sie die Webseite aufrufen können auf dem Host-System (index.html)

C) Cloud-init Datei ändern und erstellen

Wenn Sie die Cloud-init Konfigurationsdatei aus A) mit der Anleitung aus KN01 vergleicht, merken Sie, dass einige Schritte fehlen. Hier werden Sie die fehlenden Elemente hinzufügen. Wir möchten aber nun auch den Datenbank- und Webserver trennen in zwei unterschiedlichen Instanzen. Führen Sie die folgenden Aufgaben aus:

a) Erweitern Sie die Cloud-init Konfiguration aus A) um folgende Elemente:

1. Erstellen Sie die beiden PHP-Seiten (aus KN01). Es gibt **eine Anweisung in Cloud-init** mit der Sie Dateien erstellen können. **Hinweis:** Sie verwenden **nicht** mehr Git um die Inhalte zu erhalten, sondern erzeugen die Dateien in der Cloud-init Konfiguration.
2. Fügen Sie das Paket *adminer* hinzu. Adminer bietet eine GUI, um Datenbanken zu administrieren. Sie müssen anschließend die beiden folgenden Befehle ausführen. Es gibt **eine Anweisung in Cloud-init** mit der Sie Befehle ausführen können:
 1. `sudo a2enconf adminer`. Dies fügt die Konfiguration für das Paket *adminer* der *apache* Config hinzu
 2. `sudo systemctl restart apache2`. Dies startet den Service für *apache* neu.

b) Erstellen Sie eine zusätzliche cloud-init Konfiguration mit dem Namen *cloud-init-db.yaml* und starten Sie eine neue Instanz mit dem Namen *db*. Rufen Sie die beiden PHP-Seiten auf. Sie kriegen wahrscheinlich einen Fehler, wenn sie *db.php* aufrufen. **Erklären** Sie wieso. Sie können dieses Problem momentan nicht mit Cloud-init lösen. Welche manuellen Schritte **wären** notwendig? Sie dürfen (müssen aber nicht) die Schritte ausführen und die Seite zum Laufen bringen.

Abgabe:

- Fügen Sie beide cloud-init Dateien Ihrem Git-Repo hinzu.
- Zeigen Sie mit Screenshots, dass Sie die verschiedenen Seiten aufrufen können. Fügen Sie die Links auf die Cloud-init Referenz zu den neuen Befehlen hinzu.
- Zeigen Sie mit Screenshots, dass Sie *adminer* aufrufen können (<http://ihre-ip/adminer/>)

Nur Windows 10/11 Pro

- Zeigen Sie, dass Sie sich mit der Datenbank verbinden können.