



KN02: Automatisierung / Cloud-init

- A) Cloud-init Datei Verstehen
- B) Cloud-init Datei verwenden.
- C) Cloud-init Datei ändern und erstellen
 - a) Datenbankserver
 - b) Webserver

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 2 unterschiedliche Instanzen.**



Führen Sie die folgenden Aufgaben aus:

a) Datenbankserver

Erstellen Sie eine cloud-init Konfiguration mit dem Namen *cloud-init-db.yaml* für den **Datenbank Server** und starten Sie eine neue Instanz mit einem passenden Namen (z. B. *db*). Verwenden Sie die Befehle aus KN01 und pflegen Sie diese in die korrekten Bereiche in der Cloud-init Datei ein.

1. Die notwendigen Pakete sollen mit der Cloud-init-Anweisung für Pakete installiert werden.
2. Die notwendigen Befehle (aus KN01) sollen mit der entsprechenden Cloud-init-Anweisung ausgeführt werden. Finden Sie den entsprechenden Bereich in der Cloud-Init Dokumentation.
3. Fügen Sie diesen Befehl hinzu (bevor Sie den Datenbank-Service neustarten): `sudo sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/mariadb.conf.d/50-server.cnf`. Dieser Befehl ändert die Konfigurationsdatei der Datenbank und lässt externe Verbindungen zu.

Beweisführung

1. Verbinden Sie sich mit dem Server (Shell) und zeigen Sie, dass, die Datenbankverbindung mit dem Benutzer *admin* funktioniert. Der Befehl dazu lautet `mysql -u admin -p`. Sie müssen dann anschließend das Passwort eingeben. Erstellen Sie einen Screenshot, der den Befehl und die CLI von mysql zeigt.
2. **Nur bei Verwendung mit Hyper-V:** Von ihrem lokalen System zeigen Sie, dass Sie von extern auf den Datenbank Server zugreifen können. Verwenden Sie dazu telnet mit dem Befehl `telnet <IP> 3306`. Erstellen Sie einen Screenshot des Befehls und des Resultats.
3. Fügen Sie die Cloud-Init-Datei im Git-Repository hinzu

b) Webserver

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. **Achtung:** Sie müssen mindestens einen Datei-Inhalt ändern.
2. Fügen Sie das Paket *adminer* hinzu. Adminer bietet eine GUI, um Datenbanken zu administrieren. Sie müssen anschliessend die beiden folgenden Befehle ausführen (Die entsprechende Cloud-init-Anweisung kennen Sie bereits):
 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.

Beweisführung

1. Rufen Sie die Seite index.html auf und erstellen Sie einen Screenshot der URL und des Inhalts.
2. Rufen Sie die Seite info.php auf und erstellen Sie einen Screenshot der URL und des Inhalts.
3. Rufen Sie die Seite db.php auf und erstellen Sie einen Screenshot der URL und des Inhalts. Wenn Sie Hyper-V verwenden (Windows Pro), müssen sie das entsprechende Resultat kriegen.
4. Rufen Sie Adminer auf (<http://ihre-ip/adminer/>) und erstellen Sie einen entsprechenden Screenshot. Falls Sie Windows Pro verwenden: Verbinden Sie sich mit dem DB-Server und zeigen Sie mit Screenshots, dass die Verbindung funktioniert.
5. Fügen Sie die Cloud-Init-Datei im Git-Repository hinzu.