



## KN08: Automation

- A) Automatisierung mit Command Line Interface (CLI)
- B) Terraform
- C) Beliebige Erweiterungen

# KN08: Automation

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

## A) Automatisierung mit Command Line Interface (CLI)

Lesen sie das [entsprechende Kapitel in der hier verlinkten Theorie](#).

Installieren Sie die CLI auf ihrem Rechner <https://aws.amazon.com/cli/> (rechts). Sie müssen die Umgebung nun so konfigurieren, dass Sie mit dem CLI in Ihrer Umgebung arbeiten können. Dafür gibt es verschiedene Methoden. Da wir aber mit einer Schulungsumgebung arbeiten, sind unsere Mittel begrenzt.

**Achtung:** Die folgenden Schritte müssen Sie **jedes Mal** wiederholen, wenn Sie die AWS-Umgebung starten. Ihre Zugangsdaten ändern sich immer.

The screenshot shows the AWS CLI configuration interface. The 'AWS CLI' section is highlighted with a red box, and a red arrow points to the 'Show' button. The 'Cloud Access' section is also highlighted with a red box, and a red arrow points to the 'AWS CLI' section. The 'Cloud Access' section shows the 'AWS CLI' configuration details, including the 'aws\_access\_key\_id', 'aws\_secret\_access\_key', and 'aws\_session\_token'.

Weitere Infos finden Sie unter [diesem Link](#). Testen Sie nun die Konfiguration und rufen Sie den Befehl `aws ec2 describe-instances` in der Konsole auf (z. B. Powershell). Hilfeseiten können Sie mit den Befehlen `aws help` (Bereichsübersicht), `aws ec2 help` (Befehlsübersicht für den Bereich `ec2`) aufrufen oder `aws ec2 describe-instance help` (Konkrete Hilfe für einen Befehl). Erstellen Sie nun Befehle für folgende Aufgaben und testen Sie diese gleich:

- Stoppen einer Instanz (nicht terminieren)
- Starten der gleichen Instanz
- Erstellen einer neuen Instanz. Sie erstellen wieder den Datenbank-Server aus den früheren Kompetenzen. Verwenden Sie die entsprechende Cloud-Init-Datei.

### Abgaben:

1. Screenshot der Details der Instanz, die Sie stoppen und starten
2. Befehle als Skript-Datei (oder die Befehle in Markdown).
3. Screenshot der Details der neu-erstellten Instanz.
4. Screenshot des Befehls `telnet ihre-IP 3306`, um zu zeigen, dass das Cloud-Init auch tatsächlich ausgeführt wurde.

Recherchieren Sie nun Befehle, die Sie verwenden müssten, um KN04 via CLI nachzubilden (IP erstellen, Security-Group erstellen, etc). Sie müssen die Befehle nicht ausführen, aber konzeptionell hinschreiben, z.B. wie folgt. [Referenz der Befehle finden Sie unter diesem Link](#).

```
aws ec2 create-subnet --vpc-id vpc-081ec835f3EXAMPLE --cidr-block 10.0.0.0/24 --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=subnet-kn08}]
aws ec2 .....
```

Beachten Sie:

- Dabei müssen die Referenzen nicht mit ihren echten Referenzen (--vpc-id) übereinstimmen. Die Befehle sollten stimmen, werden aber **nicht** ausgeführt, wegen den falschen Referenzen.
- Geben Sie **jedem** Objekt einen Namen wie im Beispiel oben!
- Halten Sie die Reihenfolge und Logik von KN04 bei

### Abgaben:

1. Die Liste der Befehle in einer Skript-Datei oder in Markdown.
2. Ein paar Sätze zur Automatisierung. Was ist nun notwendig für die Automatisierung? Sie kennen die Befehle, können aber nicht einfach die Befehle in der Reihenfolge ausführen. Was ist notwendig? Wie gehen Sie vor?

## B) Terraform

[Lesen sie das entsprechende Kapitel in der hier verlinkten Theorie](#).

Erstellen Sie eine AWS-Instanz mit **Terraform**. Sie finden den [Einstieg unter diesem Link](#).

Ihre Instanz hat die folgenden Eigenschaften:

- Sie erstellen wiederum nur den Datenbank-Server.
- Ihre Terraform-Konfiguration hat Anweisungen für
  - Die Security-Group
  - Das Cloud-Init Skript
  - Die Instanz
- Sie müssen also **nicht** alle Objekte wie Subnetz, Netzwerk-Interface, IPs erstellen.

### Abgaben:

1. Terraform Konfiguration als zusätzliche Datei in Git (nicht im Markdown)
2. Screenshot des Telnet-Befehls der zeigt, dass Ihr Datenbank-Zugriff auch tatsächlich funktioniert
3. Konsolen-Befehle, die Sie ausführen mussten (Terraform CLI).
4. Bei Aufgabe A haben Sie argumentiert was von Ihrer Seite her noch notwendig ist für die Automatisierung. Wieso ist das bei Terraform nicht notwendig?

## C) Beliebige Erweiterungen

---

Dieser Teil gehört nicht mehr zu den offiziellen Aufgaben. Sie können daran arbeiten, falls Sie noch Zeit haben. Folgend finden Sie ein paar Ideen, die man umsetzen kann. Sie können aber gerne auch eigenes Testen

- Gesamte Infrastruktur definieren in Terraform (inkl. IP, Netzwerk-Interface, etc)
- Adaption auf Azure (CLI, Terraform, etc). Mit Ihrer TBZ-E-Mail Adresse haben Sie ein Guthaben von 100 USD bei Azure. Seien Sie vorsichtig, dass Sie nicht alles verwenden, weil Azure evtl. noch in anderen Modulen verwendet wird.
- Automatisieren Sie die Distribution einer Applikation bei Heroku. Bei Heroku kriegen Sie einen free-tier.