UNIKASSEL
VERSITÄT

# Entwicklung eines Quellcodegenerators für „Infrastructure-as-Code“ mit Terraform zur Bereitstellung von Ressourcen bei Hetzner Cloud

## EXPERTENINTERVIEW

als Teil der Masterarbeit
im Fachbereich Elektrotechnik/Informatik
der Universität Kassel


| | |
|---|---|
| Durchgeführt von: | Keanu Stückrad, B. Sc. |
| Matrikelnummer: | 35198923 |
| E-Mail: | uk054459@student.uni-kassel.de |
| Vorgelegt im: | Fachbereich Elektrotechnik/Informatik<br>Fachgebiet Software Engineering |
| Gutachter: | Prof. Dr. Oliver Hohlfeld<br>Prof. Dr. Gerd Stumme |
| Betreuer: | Sebastian Copei, M. Sc. |
| Durchgeführt am: | 17. Juli 2023 |
| Durchgeführt mit: | _____ |
| Berufs-Tätigkeit: | _____ |

| | **Hintergrund** | **-2** | **-1** | $\pm 0$ | **+1** | **+2** |
|---|---|---|---|---|---|---|
| **1.1** | Was sind Ihre Erfahrungen mit Cloud Computing? Kennen Sie die vier verschiedenen Service-Ebenen? Hinweis: XaaS | ☐ | ☐ | ☐ | ☐ | ☐ |
| **1.2** | Haben Sie sich bereits mit verschiedenen Cloud-Anbieter beschäftigt? Können Sie Cloud-Anbieter nennen? Können Sie die vier größten Cloud-Anbieter in der richtigen Reihenfolge auflisten? | ☐ | ☐ | ☐ | ☐ | ☐ |
| **1.3** | Wie gut kennen Sie sich mit der Hetzner Cloud der Hetzner Online GmbH aus? Haben Sie bereits Cloud-Services der Hetzner Cloud verwendet? | ☐ | ☐ | ☐ | ☐ | ☐ |
| **1.4** | Wie viel haben Sie sich bereits mit dem Bereitstellen von Infrastruktur auseinander gesetzt? Welche Ansätze haben Sie angewandt? Hinweis: Web-UI, CLI, SDKs, IaC | ☐ | ☐ | ☐ | ☐ | ☐ |
| **1.5** | Kennen Sie sich mit dem Infrastructure-as-Code-Tool Terraform aus? Wofür haben Sie es bereits verwendet? | ☐ | ☐ | ☐ | ☐ | ☐ |

**1.6** Haben Sie bisher viele Terraform-Provider der Terraform-Registry verwendet? Welche haben Sie verwendet? ☐ ☐ ☐ ☐ ☐

**1.7** Wie gut ist Ihnen die Terraform-CLI vertraut? Welche Befehle verwenden Sie am häufigsten? ☐ ☐ ☐ ☐ ☐

**1.8** Ist Ihnen die Programmiersprache Kotlin bekannt? Haben Sie Kotlin zur Backend-, App-Entwicklung oder in einem anderen Kontext verwendet? ☐ ☐ ☐ ☐ ☐

**1.9** Haben Sie Erfahrungen mit der Generation von Quellcode durch eine Programmiersprache? Welche Sprachen zum Generieren wurde verwendet? In welcher Sprache wurde Quellcode erzeugt? ☐ ☐ ☐ ☐ ☐

Nun folgt die Einführung in das Thema.
Informationen dazu auf den nächsten
Seiten und der PowerPoint Präsentation!

## Thema der Arbeit

Alles in der Farbe Lila stellt den normalen Terraform Ablauf dar.
Mit der Farbe Orange werden die Änderungen und Eingriffe in diesen Ablauf abgebildet.

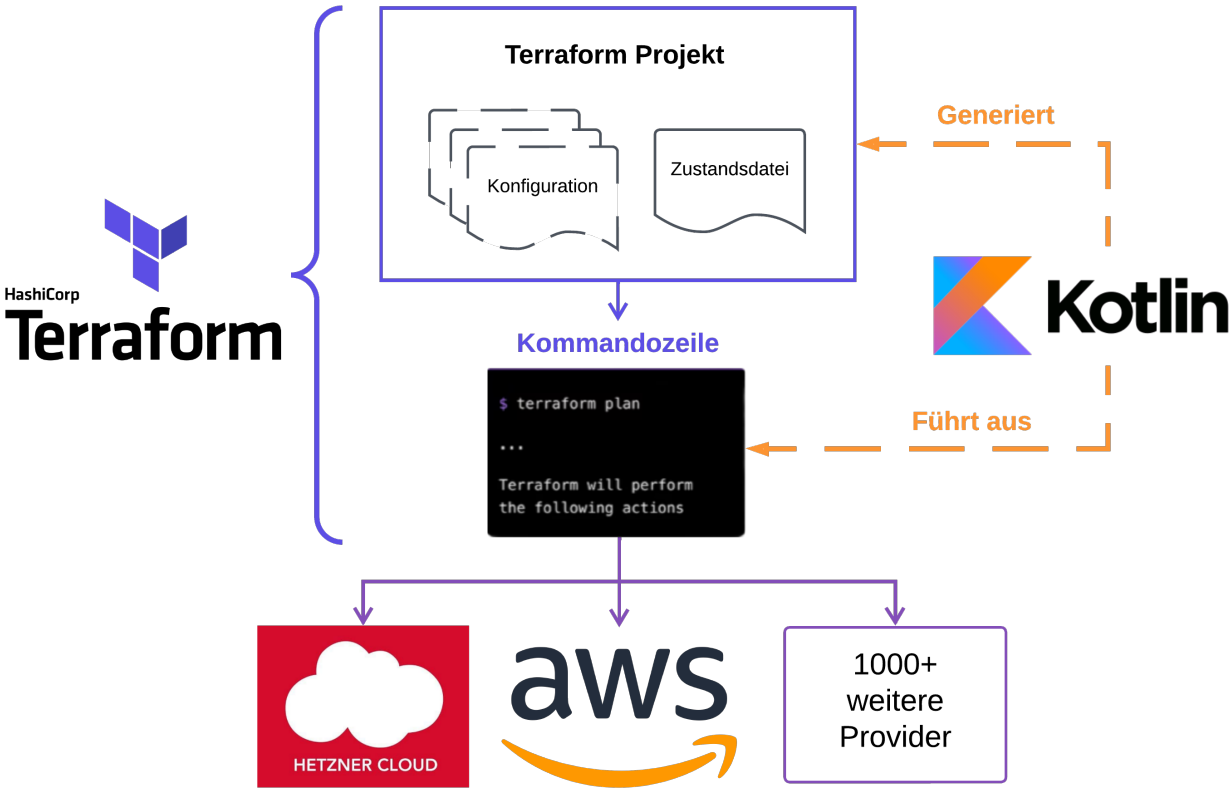


Abbildung 1: Konzept der Masterarbeit

## Beispielablauf: Von Kotlin in die Cloud

1. Definiere Infrastruktur durch Kotlin Quellcode:

```
val hCloudServer: HCloudServer.Resource = HCloudServer.Resource.Builder()
        .resourceName("resNameTest")
        .name("test")
        .image("debian-11")
        .serverType("cx11")
        .location(HCloudServer.Location.NBG1)
        .build()
```

Abbildung 2: Ausgang: Kotlin Quellcode

2. Generiere Terraform-Dateien daraus:

```
HCloud
    .addTokenToTfVarsFromEnv()
    .addProviderToRootModule()
```

Abbildung 3: Hinzufügen des Hetzner Cloud Providers und Tokens

```
RootModule
    .setConfiguration(terraform)
    .addChild(HCloud.token)
    .addChild(hCloudServer)
    .generateFiles(true)
```

Abbildung 4: Setzen des terraform-Blocks, Hinzufügen des Servers aus Schritt 1 und Generation der Dateien

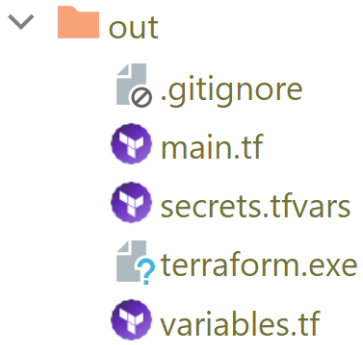


Abbildung 5: Ergebnis: Verzeichnisstruktur

Der Server (vgl. Abbildung 6) befindet sich (mit Hetzner Cloud-Provider-Block und Terraform-Block) in der „main.tf“-Datei. Die „.gitignore“-Datei verbirgt die Terraform-Zustandsdatei, Secrets-Datei „secrets.tfvars“ und Ausführungsdatei („terraform.exe“, mehr dazu im 4. Schritt). Die Secrets-Datei enthält das Hetzner Cloud Token. „variables.tf“ enthält die Terraform-Variable dazu.

```
resource "hcloud_server" "resNameTest" {
  location    = "nbg1"
  name        = "test"
  server_type = "cx11"
  image       = "debian-11"
}
```

Abbildung 6: Ergebnis: Terraform Quellcode

3. Ausführen den Terrafrom-Quellcodes mit Kotlin:

```
Executor.downloadTerraformBinary(
    "https://releases.hashicorp.com/terraform/" +
    "1.3.3/terraform_1.3.3_windows_386.zip",
    true
)
// Executor.copyBinaryFromPath() // Alternative zum Herunterladen
Executor.init()
Executor.plan()
Executor.apply()
// Executor.destroy()
```

Abbildung 7: CLI durch Kotlin: Initialisierung, Planung und Bereitstellen von Infrastruktur

Zuerst wird die Ausführungsdatei heruntergeladen (Alternativ aus dem PATH kopiert).
Danach wird die Terraform Umgebung mit dem init-Befehl eingerichtet.
Durch den plan-Befehl werden Änderungen zur Zustandsdatei angezeigt.
Mit dem apply-Befehl werden die Änderungen vorgenommen.
(Durch den destroy-Befehl rückgängig gemacht.)

4. Infrastruktur ist beim Cloud-Anbieter hochgefahren:

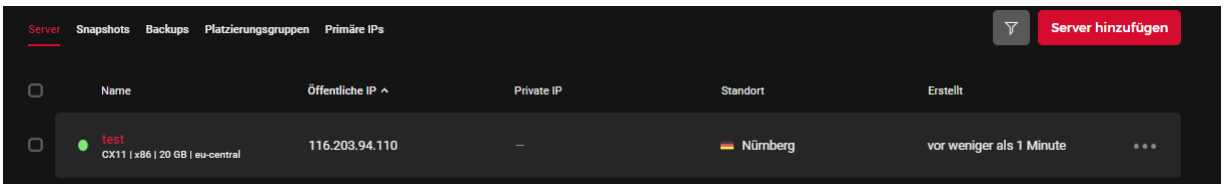


Abbildung 8: Hetzner Cloud Web-UI

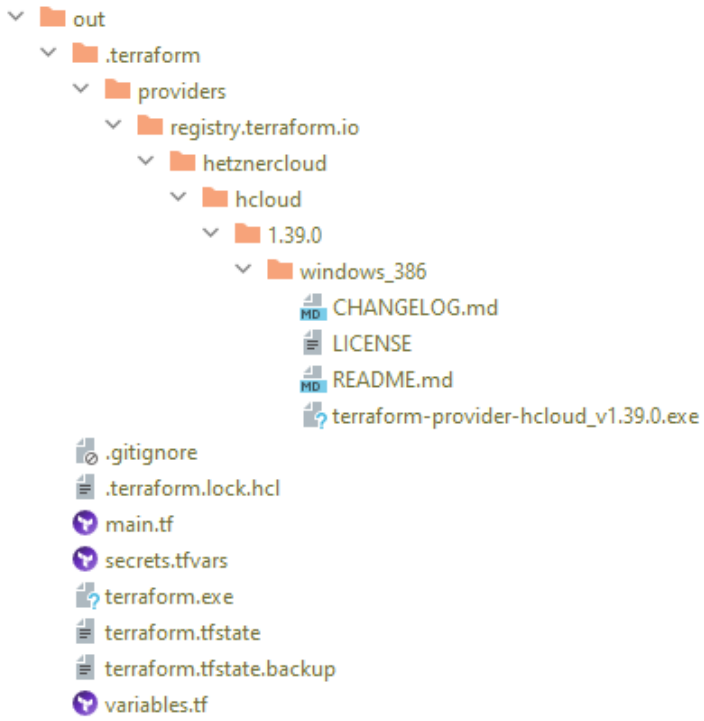


Abbildung 9: Verzeichnisstruktur nach Bereitstellen

Hetzner Cloud-Provider wurde in das lange verschachtelte Unterverzeichnis heruntergeladen.
Terraform-Lock-Datei mit Providerversionen wurde angelegt.
Terraform-Zustandsdatei und Backup sind angelegt worden.

# Weiter zu den nächsten Fragen!

| **2** | **Prinzipien und Konzepte** | **-2** | **-1** | **±0** | **+1** | **+2** |
|---|---|---|---|---|---|---|
| **2.1** | Was halten Sie vom Konzept, Terraform-Quellcode zu generieren? Sehen Sie mögliche Vorteile oder Herausforderungen?<br>Hinweis: Wartung, Lesbarkeit, Fehler-Prävention, Validierung, Best-Practises | ☐ | ☐ | ☐ | ☐ | ☐ |
| **2.2** | Wirkt sich Ihrer Meinung nach der Quellcodegenerator positiv auf die Arbeitsabläufe im Management von Infrastruktur aus? Auf was wirkt er sich aus?<br>Hinweis: Wartung, Organisatzion, Modularität, Wiederverwendbarkeit, Effizienz | ☐ | ☐ | ☐ | ☐ | ☐ |
| **2.3** | Wie stehen Sie zur Verwendung von Kotlin zum Bereitstellen von Infrastruktur? Was sind Ihrer Meinung nach die Hauptvorteile im Vergleich zu herkömmlichen Methoden?<br>Hinweis: Compiler, Objekt-Orientierung, Java-Kompatibelität, Anwendungs-Integration | ☐ | ☐ | ☐ | ☐ | ☐ |
| **2.4** | Ist die Integration einzelner Terraform-Provider in die Bibliothek von Vorteil? Die Hetzner Cloud ist bereits im Quellcodegenerator implementiert und weitere Provider können hinzugefügt werden. | ☐ | ☐ | ☐ | ☐ | ☐ |

Als nächstes: Quellcode-Beispiele
in der Entwicklungsumgebung!

| | **3 Funktionalität und Nutzbarkeit** | **-2** | **-1** | **±0** | **+1** | **+2** |
|---|---|---|---|---|---|---|
| **3.1** | Sind Herausforderungen hoch oder Einschränkungen groß, denen Entwickler bei der Verwendung von Kotlin zur Erstellung von Terraform-Konfigurationen begegnen können? Können Sie Beispiele nennen? | ☐ | ☐ | ☐ | ☐ | ☐ |
| **3.2** | Gibt es viele Bereiche, in denen der Quellcodegenerator (erhebliche) Vorteile bieten könnte? | ☐ | ☐ | ☐ | ☐ | ☐ |
| **3.3** | Erleichtert das Verwenden der Terraform-CLI-Befehle in Kotlin den Prozess zum Bereitstellen von Infrastruktur? Inwiefern vereinfacht es den Prozess? | ☐ | ☐ | ☐ | ☐ | ☐ |
| **3.4** | Sind Open-Source, Erweiterbarkeit der Bibliothek, Maven-Abhängigkeit und das Beispiel-GitHub-Repository für Anwender hilfreich? | ☐ | ☐ | ☐ | ☐ | ☐ |
| **3.5** | Ist der generierte Verzeichnisaufbau nach Konvention für Sie als Endnutzer der Bibliothek ein Hindernis? Würden Sie einen anderen Aufbau bevorzugen? | ☐ | ☐ | ☐ | ☐ | ☐ |

**3.6** Vereinfacht für Sie das Programmieren mit Objekt-Orientierung und dem Builder-Pattern das Aufbauen von Terraform-Konstrukten? ☐ ☐ ☐ ☐ ☐

**3.7** Die Bibliothek wird um Terraform-Konstrukte erweitert, indem der Block als Parameter in den Konstrukten übergeben wird. Terraform-Provider werden über Vererbung von Ressourcen- und Datenquellen-Klasse implementiert. Ist dieser Implementierungsansatz Ihrer Meinung nach sinnvoll? (Können Sie bessere Alternativen vorschlagen?) ☐ ☐ ☐ ☐ ☐

**3.8** Ist die Fehleranzeige bei doppelt und unerwünscht angegebenen Werten ausreichend? ☐ ☐ ☐ ☐ ☐

**3.9** Werden sensitive Werte vernünftig vor ihrer Veröffentlichung in Onlinequellen (z.B. GitHub) gesichert, indem System-Umgebungsvariablen und die „secrets.tfvars“-Datei verwendet werden? ☐ ☐ ☐ ☐ ☐

**3.10** Haben Sie zusätzliche Vorschläge oder Erkenntnisse, die die Effektivität des vorgeschlagenen Quellcodegenerators verbessern könnten? ☐ ☐ ☐ ☐ ☐

Vielen Dank für die Teilnahme!
Weitere Fragen?