Technische Berufsschule Zürich TBZ

KN03: Cloud-init und AWS

Beachten Sie die allgemeinen Informationen zu den Abgaben.

In dieser Kompetenz werden wir mit der Automatisierung der Installation arbeiten. Sie werden Cloud-init verwenden um Server zu installieren.

A) Cloud-init Datei Verstehen (10%)

Sie werden nun zuerst Cloud-init Dateien und das entsprechende Format YAML verstehen lernen. **Lesen Sie sich in der Theorie ein zu den Themen *Cloud-init* und *YAML* .** Laden Sie nun diese cloud-init Datei herunter. Erklären Sie alle Zeilen der Cloud-init Datei in folgendem Schema.

```
#cloud-config
users: # ....
 - name: ubuntu # Benutzername
   sudo: ALL=(ALL) NOPASSWD:ALL # sudo-Regeln für diesen Benutzer
   #...
   #...
```

Abgaben:

- Die dokumentierte YAML-**Datei** in Git (nicht als Screenshot)

B) SSH-Key und Cloud-init (15%)

Wichtig: Lesen Sie sich in die funktionsweise von SSH und Private/Public Key ein.

Anstatt, dass Sie den SSH-Key im GUI auswählen, können Sie ihn auch im Cloud-Init mitgeben. Dies werden wir folgend tun. Extrahieren Sie die beiden public-Keys aus Ihren privaten Schlüssel, welche Sie in KN02 erstellt haben.

Erstellen Sie eine neue Instanz in AWS mit den folgenden Einstellungen:

- Ubuntu 22.04
- Bei "Key pair" verwenden Sie ihren **zweiten** Schlüssel.
- Keine Änderungen bei den restlichen Einstellungen, ausser..

- ... verwenden Sie die cloud-init Datei und ersetzen Sie den Schlüssel mit ihrem **ersten öffentlichen** Schlüssel. Beachten Sie das Format `ssh-rsa <ihr-Schlüssel-ohne-zeilenumbrüche> aws-key`
- Sie können ihre Cloud-init Konfiguration einfach in dem Feld "*user data*" in der Sektion "*Advanced details*" reinkopieren - ähnlich wie Sie es in KN02 gemacht haben.

  Hinweis: In KN02 hatten Sie in diesem Feld ("user data") ein Bash-Skript ausgeführt. Dies weil die erste Zeile auf ein Bash-Script hinweist. Wenn Sie nun aber ihre YAML-Datei in das Feld kopieren, wird AWS automatisch das richtige installieren.

  Hinweis: Stellen Sie sicher, dass die Zeile `#cloud-config` auch tatsächlich in der YAML-Datei steht, sonst wird der Inhalt nicht korrekt ausgeführt.

Sie verwenden nun also beide Schlüssel (einmal via cloud-init, einmal via Eigenschaften im GUI).

Zeigen Sie nun, dass Sie **nur** mit dem **ersten privaten Key** aus der Cloud-Init Datei einloggen können.

Rufen Sie nun das cloud-init-log (`/var/log/cloud-init-output.log`) auf und merken Sie sich diesen Pfad für zukünftige Aufträge für die Fehlerfindung.

Abgaben:

- Ihre angepasste Cloud-init Konfiguration als **Datei** im Git-Repository.
- Ein Screenshot der Details der Instanz. Scrollen Sie so weit runter, dass das Feld "**Key pair assigned at launch**", sichtbar ist.
- Screenshot mit dem ssh-Befehl und des Resultats unter Verwendung des **ersten** Schlüssels.
- Screenshot mit dem ssh-Befehl und des Resultats unter Verwendung des **zweiten** Schlüssels.
- Screenshot mit dem **Auszug** aus dem Cloud-Init-Log. Der **Befehl** den Sie aufgerufen haben und der obere Teil des Logs sollten sichtbar sein. Sie müssen nicht das gesamte Log abgeben.

C) Template (5%)

Erstellen Sie sich ein Cloud-Init Template aufgrund der Datei aus B), welches Sie bei den folgenden Kompetenzen einsetzen.

Fügen Sie auch direkt den öffentlichen Schlüssel für Lehrpersonen hinzu, so dass die Lehrpersonen Zugriff auf Ihre Instanzen kriegen. Sie haben nun also **zwei** öffentliche Schlüssel drin - Ihren und den der Lehrperson.

D) Auftrennung von Web- und Datenbankserver (70%)

In KN02 hatten Sie eine virtuelle Instanz installiert mit Web- und Datenbankserver. Sie haben gesehen, dass viele Befehle ausgeführt werden müssen. Wenn Sie nun eine weitere Instanz installieren, müssen Sie die selben Schritte immer wieder durchführen. Cloud-init hilft Ihnen die Installation zu vereinfachen.

In dieser Teilaufgabe werden wir die Datenbank und den Webserver **voneinander trennen** in zwei Unterschiedliche Instanzen. Für beide Instanzen werden Sie eine Cloud-init Datei (***cloud-init-web.yaml, und cloud-init-db.yaml***) erstellen.

Das folgende Schema zeigt Ihnen unsere Zielarchitektur mit den entsprechenden Ports. **Beachten Sie den zusätzlichen Port, den Sie nun ebenfalls öffnen müssen (auf der richtigen Instanz).**

Aufgaben:

- Lesen Sie das folgende kurz durch bevor Sie beginnen. Erstellen Sie dann **zuerst** den **Datenbank**-Server. Wenn dieser läuft und die Screenshots gemacht wurden, erstellen Sie den Web-Server.
- Installieren Sie die gleichen **Pakete** wie in KN02 via Cloud-Init. Verteilen Sie die Pakete korrekt auf die beiden Cloud-init Dateien.
- In KN02 hatten Sie Dateien via GIT geklont und dann an die Zielorte kopiert. Sie gehen nun ein wenig anders vor. Sie verwenden die **write_files**-Anweisung von Cloud-init, um Dateien zu erstellen. Sie finden in der Doku zu Cloud-init weitere Informationen. Sie können die Datei-Inhalte von Gitlab kopieren.
- In KN02 hatten Sie verschiedene Befehle ausführen müssen. Verwenden Sie die **runcmd**-Anweisung von Cloud-init, um Befehle auszuführen. Sie müssen natürlich nur noch die Befehle ausführen, die in den vorherigen Punkten nicht abgedeckt sind.
- Führen Sie den folgenden Befehl noch zusätzlich aus: `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.
- 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. Starten Sie den Apache Service neu.

Beweisführung DB Server

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. 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. Sie können den Telnet-Client über die Windows-Features aktivieren. Wenn Sie kryptische Zeichen als Antwort bekommen, funktioniert der Zugriff.
3. Fügen Sie die Cloud-Init-Datei im Git-Repository hinzu.

Beweisführung Webserver

1. Rufen Sie die Seiten index.html, info.php und db.php auf und erstellen Sie einen Screenshot der URL und des Inhalts.

2. Rufen Sie Adminer auf (http://ihre-ip/adminer/), verbinden Sie sich mit dem DB-Server und zeigen Sie mit Screenshots, dass die Verbindung funktioniert.
3. Fügen Sie die Cloud-Init-Datei im Git-Repository hinzu.

Häufige Fehlerquellen

- Sie vergessen die erste Zeile in der Cloud-init Datei `#cloud-config`. Diese ist notwendig!
- Einrückungen wurden nicht korrekt erstellt. Das Format können Sie hier überprüfen: https://www.yamllint.com/.
- Sie können auf dem Server die Log-Datei für Fehler durchsuchen: `/var/log/cloud-init-output.log`
- Das Format des SSH-Schlüssels ist falsch. Korrekt ist: `ssh-key XXXXXXXXXX aws-key`.

Quellen

puttygen: Puttygen download

Thorntech: Passwords vs. SSH keys - what's better for authentication?