Technische Berufsschule Zürich TBZ

KN03: Cloud-init und AWS / Storage

Beachten Sie die allgemeinen Informationen zu den Abgaben.

In dieser Kompetenz werden Sie Cloud-init verwenden um Server zu installieren.

A) Auftrennung von Web- und Datenbankserver

In KN01 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 ein Cloud-init Skript (***cloud-init-web.yaml, und cloud-init-db.yaml***) erstellen. Als Vorlage verwenden Sie Ihre Lösung aus KN02 D).

Das folgende Schema zeigt Ihnen unsere Zielarchitektur mit den entsprechenden Ports.

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.
- Die in KN01 installierten Pakete installieren Sie nun mit Cloud-init. Verteilen Sie die **Pakete** korrekt auf die beiden Cloud-init Dateien. Aus KN02 kennen Sie die entsprechende Anweisung
- In KN01 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 KN01 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, **bevor** Sie den DB-Service neu starten: `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. `sudo systemctl restart apache2`. Dies startet den Service für *apache* 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.
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.

B) Speicher

Sie haben in der Aufgabe A) von KN02 bereits mit dem S3 Speicher gearbeitet. In der Theorie finden Sie weitere Informationen zu Speichermodellen.

a) Welchem Speichermodell wird S3 zugeordnet? Begründen Sie ihre Antwort.

b) Wenn Sie eine EC2-Instanz erstellen, fügen Sie auch Speicher hinzu. Es handelt sich hierbei um einen **Hot Storage**, welcher in AWS Elastic Block Storage (EBS) genannt wird. Folgend zeigen Sie, dass dieser Speicher sowohl persistent als auch flüchtig sein kann mit den Schritten:

1. Erstellen Sie eine neue EC2-Instanz (ohne spezielle Einstellungen oder Cloud-init Datei)
2. Erstellen Sie ein neues *Volume* mit **100GB** in der AWS Konsole. Suchen Sie **selbstständig** nach Informationen dazu.
3. Fügen Sie das neue Volumen der bestehenden Instanz hinzu. **ACHTUNG**: der Mount-Pfad ist wichtig. Lesen Sie sich ein was */dev/sda1* und */dev/sdf* bedeutet.
4. **Erstellen Sie ein Screenshot der Details ihrer Instanz, der beide EBS-Disks zeigt.**
5. Löschen Sie die **Instanz**. Lesen Sie zuerst die Bestätigungsmeldung durch und geben Sie in **eigenen** Worten wieder was in der Meldung erklärt wird.

6. Erklären Sie wieso noch ein Volumen existiert. Machen Sie ein Gedankenexperiment und erklären Sie ein Szenario in dem dieses Verhalten Sinn ergibt in ein paar schlüssigen Sätzen.

Abgaben:

- Screenshots der Liste der EBS (2 Volumen) vor der Löschung.
- Screenshots der Liste der EBS nach der Löschung.