

# Aufgabe 1

## Szenario

- Für die Migration einer bestehenden Website, die aktuell on-premises (im eigenen Rechenzentrum) gehostet wird, soll die Verwendung von EC2 Auto Scaling Groups und ELB getestet werden
- Du als AWS Experte, wurdest beauftragt die Infrastruktur zu erstellen
- Es geht hier nur um einen ersten Test, an dem die Technologie evaluiert werden kann

## Vorbereitungen

- Logge dich in der AWS Console (Techstarter Sandbox)
- Stelle sicher, dass du dich in der `eu-central-1` Frankfurt Region befindest
- Stelle sicher, dass das default VPC erstellt wurde:
  - Navigiere hierzu zu `Services` -> `VPC` -> `Your VPCs`
  - In der Liste sollte ein VPC vorhanden sein, welches `Yes` in der Spalte `Default` hat
  - Falls nicht, klicke auf `Actions` -> `Create Default VPC`

## 1a Security Groups

- Um die Infrastruktur zu erstellen, benötigst du zwei Security Groups:
  - Für die EC2 Instanzen
  - Für den ELB
- Navigiere hierfür zu `Services` -> `EC2` -> `Security Groups`

### EC2 Security Group

- Erstelle eine Security Group mit dem Namen `EC2SecurityGroup` und folgenden Ingress Rules:
  - SSH (Port 22) von dem gesamten Internet (0.0.0.0/0)
  - HTTP (Port 80) von dem gesamten Internet (wird später noch geändert)
- Stelle sicher, dass die Egress Rule richtig erstellt wurden, falls nicht, füge eine hinzu:
  - Wähle bei **Type** `All traffic` aus und dann bei **Destination** `Anywhere-IPv4`
- Wähle das Default VPC aus und schreibe eine Beschreibung wie zB. `Security group for the EC2 Launch Template`

### ELB Security Group

- Erstelle eine Security Group mit dem Namen `ELBSecurityGroup` und folgenden Ingress Rules:
  - HTTP (Port 80) von dem gesamten Internet
- Stelle sicher, dass die Egress Rule richtig erstellt wurden, falls nicht, füge eine hinzu:
  - Wähle bei **Type** `All traffic` aus und dann bei **Destination** `Anywhere-IPv4`
- Wähle das Default VPC aus und schreibe eine Beschreibung wie zB. `Security group for the ELB`

## 1b EC2 Launch Template

- Um die EC2 Auto Scaling Group zu erstellen, benötigst du ein Launch Template
- Klicke hierfür auf `Services` -> `EC2` -> `Launch Templates`
- Klicke auf `Create launch template`

- Gib dem Template den Namen `EC2LaunchTemplate`
- Klicke die Checkbox für `Auto Scaling guidance` an
- Benutze folgende Werte:
  - AMI: `Amazon Linux 2 AMI (HVM), SSD Volume Type`
  - Instance Type: `t2.micro`
  - Key Pair: (ohne Keypair)
  - Security Groups: `EC2SecurityGroup`
  - User Data (Advanced Details):

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<h1>Hello from $(hostname)</h1>" > /var/www/html/index.html
```

- Klicke auf `Create launch template`

## 1c EC2 Auto Scaling Group

Klicke immer zwischen den einzelnen Schritten auf Next um zum nächsten Schritt zu gelangen.

- Erstelle nun eine Auto Scaling Group die auf dem gerade erzeugten Launch Template basiert
- Navigiere hierzu zum Reiter Auto Scaling Reiter des EC2 Service
- Klicke auf `Create Auto Scaling group`
- Gib der Auto Scaling Group den Namen `EC2AutoScalingGroup`
- Wähle das gerade erstellte Launch Template aus ( `EC2LaunchTemplate` )
- Wähle das Default VPC und alle 3 verfügbaren Subnets aus
- Wähle `No load balancer` aus, da wir diesen von Hand erstellen wollen
- Setze folgende Werte:
  - Minimum size: `1`
  - Maximum size: `3`
  - Desired capacity: `2`
- Unter `Scaling Policy` wähle `Target tracking scaling` aus
- Setze folgende Werte:
  - Metric type: `Average CPU Utilization`
  - Target value: `50`
- Überspringe die SNS notifications
- Klicke auf `Create Auto Scaling group`

## 1d ELB Target Group

- Navigiere nun zum ELB Service `Services` -> `EC2` -> `Target Groups`
- Klicke auf `Create target group`
- Setze folgende Werte:
  - Target type: `Instance`
  - Name: `EC2AutoScalingGroupTargetGroup`
  - Protocol: `HTTP`
  - Port: `80`
  - VPC: `Default VPC`

- Health check protocol: `HTTP`
- Health check port: `traffic-port`
- Health check path: `/`
- Health check interval: `30`
- Health check timeout: `5`
- Healthy threshold: `5`
- Unhealthy threshold: `2`
- Wähle alle verfügbaren Instances aus und klicke auf `Include as pending below`

## 1e ELB

---

- Navigiere nun zum ELB Service `Services` -> `EC2` -> `Load Balancers`
- Klicke auf `Create Load Balancer`
- Wähle `Application Load Balancer` aus
- Gib dem Load Balancer den Namen `EC2AutoScalingGroupELB`
- Setze folgende Werte:
  - Scheme: `Internet-facing`
  - IP address type: `IPv4`
  - VPC: `Default VPC`
  - Subnets: Alle verfügbaren Subnets
  - Security groups: `ELBSecurityGroup` (Lösche die default Security Group)
  - Listeners: `HTTP` auf Port `80` Forward to `EC2AutoScalingGroupTargetGroup`
- Klicke auf `Create`

## 1f EC2 Auto Scaling Group anpassen

---

- Navigiere nun zum EC2 Auto Scaling Group Service `Services` -> `EC2` -> `Auto Scaling Groups`
- Wähle die `EC2AutoScalingGroup` aus
- Scrolle nach unten zu `Load balancer` und klicke auf `Edit`
- Setze die Checkbox bei `Application, Network or Gateway Load Balancer target groups` an
- Wähle die `EC2AutoScalingGroupTargetGroup` aus

## 2 Tests

---

- **Stelle bei allen Urls die du in den Browser kopierst sicher, dass du `http://` verwendest!**

### 2a Tests des aktuellen Standes

---

- Navigiere nun zum ELB Service `Services` -> `EC2` -> `Load Balancers`
- Kopiere dir die DNS Adresse des ELB und öffne sie in einem Browser
- Du solltest nun eine Seite sehen, welche die IP Adresse der EC2 Instanz anzeigt
- Aktualisiere die Seite mehrmals und du solltest sehen, dass die IP Adresse sich ändert
- Navigiere nun zum EC2 Auto Scaling Group Service `Services` -> `EC2` -> `Auto Scaling Groups`
- Klicke auf `EC2AutoScalingGroup`
- Klicke auf `Activity History`
- Du solltest nun sehen, dass die Auto Scaling Group die Instanzen erstellt hat

### 2b Auto Scaling Group Test

---

- Navigiere zum EC2 Service und terminiere eine der EC2 Instanzen (Rechtsklick -> `Instance State` -> `Terminate`)
- Du solltest nun sehen, dass die Auto Scaling Group eine neue Instanz erstellt hat (kann etwas dauern)
- Sollte keine neue Instanz erstellt werden, dann klicke auf `EC2AutoScalingGroup` -> `Activity History` und schau ob dort eine neue Instanz erstellt wurde (falls nicht, stelle sicher, dass die Desired Capacity auf 2 steht)

## 2c Auto Scaling Group Stress Test

---

- Verbinde dich mit einer der EC2 Instanzen (Rechtsklick -> Connect)
- Kopiere folgenden Befehl in die Konsole und führe ihn aus:

```
sudo yum install stress -y && sudo stress --cpu 4 --timeout 600
```

- **Wichtig den Browser tab offen lassen und das command nicht abbrechen!**
- Öffne nun in einem anderen Browser Tab die andere EC2 Instanz und verbinde dich auch und führe den Befehl aus
- Du solltest nun sehen, dass die Auto Scaling Group eine neue Instanz erstellt hat (kann etwas dauern)

## 3a Sicherheitsrelevante Verbesserungen (Optional / Freiwillig)

---

- Aktuell kann auch noch über die öffentliche IP Adresse auf die Instanzen zugegriffen werden
- Dies wollen wir ändern, indem wir die `EC2SecurityGroup` anpassen
- Versuche dies selbst zu lösen, und lese dir die offizielle Dokumentation durch, falls du nicht weiter kommst
- **Wichtig: Die Security Group muss so konfiguriert werden, dass du dich immer noch über SSH auf die Instanzen verbinden kannst!**

## 4 Cleanup

---

- Lösche alle Ressourcen die du erstellt hast in folgender Reihenfolge:
  - Load Balancer
  - Target Group
  - Auto Scaling Group
  - Launch Template
  - Security Group