

AWS Infrastruktur aufbauen

Themenübersicht

1. EC2-Instanzen und User Data Script
2. EBS Volumes
3. Snapshots
4. Auto Scaling Group (ASG)
5. Elastic Load Balancer (ELB)

EC2-Instanzen

Was ist EC2?

- Amazon EC2 (Elastic Compute Cloud) ist ein skalierbarer Cloud-Computing-Dienst.
- Bietet virtuelle Maschinen („Instanzen“) zum Ausführen von Anwendungen.
- Vorteile: Skalierbarkeit, Flexibilität, verschiedene Instanztypen für unterschiedliche Workloads.

Technische Hintergründe

- Instanzen können für kurze Zeit (Spot Instances), lange Zeit (Reserved Instances) oder on-demand gestartet werden.
- Konfigurationen wie CPU, RAM und Speicher hängen vom gewählten Instanztyp ab.
- Sicherheit und Netzwerkzugriff werden über **Security Groups** gesteuert.

User Data Script

Was ist User Data?

- Skripte oder Befehle, die beim Start einer EC2-Instanz ausgeführt werden.
- Nützlich zur Automatisierung von Software-Installationen und Konfigurationen direkt beim Start.

Technische Details

- Wird als Bash-Skript (Linux) oder PowerShell (Windows) beim Start eingelesen.
- Das Skript läuft nur einmal bei der ersten Initialisierung der Instanz.
- Beispiel: Einrichten einer Python-Umgebung oder Starten einer Webanwendung.

```
#!/bin/bash
yum update -y
yum install -y python3
echo "print(\"Hello, World!\")" > /home/ec2-user/script.py
python3 /home/ec2-user/script.py
```

EBS Volumes

Was ist EBS?

- Amazon Elastic Block Store (EBS) bietet persistenten Speicher für EC2-Instanzen.
- Wird wie eine Festplatte an die Instanz angehängt.
- Verschiedene Typen (gp2, io1, etc.) für unterschiedliche Leistung und Kosten.

Technische Hintergründe

- EBS ist persistent, d.h., Daten bleiben auch nach dem Herunterfahren der Instanz erhalten.
- EBS Volumes können von einer Instanz getrennt und an eine andere Instanz angehängt werden.
- Volumes werden im selben Verfügbarkeitsbereich wie die EC2-Instanz erstellt.

```
# EBS Volume erstellen  
aws ec2 create-volume --availability-zone <deine-AZ> --size 10 --volume-type gp2
```

Snapshots

Was ist ein Snapshot?

- Ein Snapshot ist ein Speicherabbild von einem EBS Volume.
- Nützlich als Backup oder zur Wiederherstellung eines früheren Zustands.

Technische Details

- Snapshots werden inkrementell erstellt; nur Änderungen seit dem letzten Snapshot werden gespeichert.
- Snapshots können verwendet werden, um ein neues EBS Volume in derselben oder einer anderen Region zu erstellen.

```
# Snapshot erstellen  
aws ec2 create-snapshot --volume-id <VolumeId> --description "Backup of my EBS volume"
```

Auto Scaling Group (ASG)

Was ist eine Auto Scaling Group?

- ASG passt die Anzahl der EC2-Instanzen basierend auf dem Bedarf an.
- Bietet Hochverfügbarkeit und Flexibilität durch automatische Skalierung.

Technische Hintergründe

- Besteht aus einer Gruppe von EC2-Instanzen, die nach definierten Regeln skaliert werden.
- ASG arbeitet mit **Launch Configurations** oder **Launch Templates**, die die Startparameter der Instanzen festlegen.
- Skalierungsoptionen:
 - **Horizontal Scaling**: Instanzen hinzufügen oder entfernen.
 - **Vertical Scaling**: Leistungsstärkere Instanzen verwenden.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --launch-configuration-name my-launch-config --min-size 1 --max-size 3 --desired-capacity 1 --vpc-zone-identifier <SubnetIds>
```

Elastic Load Balancer (ELB)

Was ist ein Elastic Load Balancer?

- ELB verteilt eingehenden Datenverkehr auf mehrere EC2-Instanzen.
- Bietet Skalierbarkeit, Redundanz und hohe Verfügbarkeit.

Technische Hintergründe

- Es gibt verschiedene ELB-Typen:
 - **Application Load Balancer (ALB)**: Lastausgleich auf Anwendungsebene (HTTP, HTTPS).
 - **Network Load Balancer (NLB)**: Hohe Durchsatzanforderungen auf Netzwerkebene (TCP).
 - **Gateway Load Balancer (GLB)**: Unterstützt Routing von Traffic durch virtuelle Netzwerksicherheitsebenen.
- Ein ELB kann in mehreren Availability Zones arbeiten, um Failover-Schutz zu gewährleisten.

```
# ELB erstellen und konfigurieren  
aws elbv2 create-load-balancer --name my-load-balancer --subnets <SubnetIds> --security-groups <SecurityGroupId>
```

Zusammenfassung

- **EC2 und User Data:** Instanzen flexibel verwalten und automatisch konfigurieren.
- **EBS und Snapshots:** Persistenter Speicher und Backups.
- **ASG:** Sicherstellen, dass genügend Instanzen verfügbar sind.
- **ELB:** Lastverteilung für bessere Verfügbarkeit und Skalierbarkeit.

Diese Infrastruktur ermöglicht eine robuste und skalierbare Anwendungsarchitektur auf AWS.

AWS Infrastruktur aufbauen und erweitern

Themenübersicht und Beispielintegration

1. EC2-Instanzen und User Data Script
2. EBS Volumes
3. Snapshots
4. Auto Scaling Group (ASG)
5. Elastic Load Balancer (ELB)

Ausgangsbeispiel: Zwei EC2-Instanzen

1. Eine **Python-Instanz** im Public Subnet, die ein Python-Skript ausführt.
2. Eine **MySQL-Instanz** im Private Subnet, mit Zugriff nur von der Python-Instanz über Port 3306.
3. Netzwerk-Setup:
 - **VPC mit Subnets:** Public und Private Subnet
 - **Internet Gateway und NAT Gateway** für Netzwerkzugriff
4. **Security Groups:** Zugang auf Ports 22 und 3306 konfiguriert

EC2 und User Data Script im Beispiel

Wie nutzen wir EC2 und User Data im Setup?

- **Ziel:** Automatisierung der Einrichtung und Konfiguration beider Instanzen
- **Python-Instanz User Data:** Installiert Python und führt ein Python-Skript aus
- **MySQL-Instanz User Data:** Installiert MariaDB und konfiguriert die Datenbank

Vorteile

- Mit User Data kann die Initialisierung und Bereitstellung automatisiert werden.
- EC2-Instanzen können für Änderungen (Updates, neue Software) mit neuen User Data Scripts neu gestartet werden.

EBS Volumes im Beispiel

Warum EBS für die Datenbank-Instanz?

- **Ziel:** Persistenter Speicher für MySQL-Daten, auch nach Neustart oder Ausfall
- Für unsere MySQL-Instanz erstellen wir ein zusätzliches EBS Volume für die Datenbankdaten.
- Dieses Volume bleibt auch nach dem Stoppen oder Neustarten der Instanz erhalten.

Umsetzung

```
aws ec2 create-volume --availability-zone <deine-AZ> --size 10 --volume-type gp2  
aws ec2 attach-volume --volume-id <VolumeId> --instance-id <InstanceId> --device /dev/sdf
```

Snapshots im Beispiel

Wie sichern wir die Daten der Datenbank-Instanz?

- **Ziel:** Backup für das MySQL-EBS-Volume erstellen
- Snapshots bieten eine einfache Methode, Daten regelmäßig zu sichern und bei Bedarf wiederherzustellen.

Anwendung

- Regelmäßig Snapshots des EBS-Volumes erstellen, um im Falle eines Datenverlusts den letzten Stand wiederherzustellen.

```
aws ec2 create-snapshot --volume-id <VolumeId> --description "Backup of MySQL volume"
```

Auto Scaling Group (ASG) im Beispiel

Warum ASG für die Python-Instanz?

- **Ziel:** Hochverfügbarkeit sicherstellen und flexibel skalieren
- Falls die Python-Instanz mehr Anfragen verarbeiten muss, kann ASG zusätzliche Instanzen bereitstellen.

Umsetzung

- **Launch Configuration** für die Python-Instanz erstellen, die User Data Script und EBS-Volumes enthält.
- ASG erstellen, die basierend auf Nachfrage skaliert.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name python-asg --launch-configuration-name python-config --min-size 1 --max-size 3 --desired-capacity 1 --vpc-zone-identifier <SubnetIds>
```

Elastic Load Balancer (ELB) im Beispiel

Warum einen ELB für die Python-Instanz?

- **Ziel:** Lastverteilung und erhöhte Verfügbarkeit
- Der ELB leitet Anfragen an die verfügbaren Python-Instanzen weiter und stellt sicher, dass die Last auf mehrere Instanzen verteilt wird.

Anwendung

- ELB zwischen den öffentlichen Python-Instanzen einrichten und über **Target Groups** alle Instanzen in der ASG ansprechen.

```
aws elbv2 create-load-balancer --name python-elb --subnets <PublicSubnetId> --security-groups <PythonSGId>
```

Zusammenfassung

Infrastrukturübersicht

1. **EC2 mit User Data:** Automatisierung und flexible Bereitstellung.
2. **EBS und Snapshots:** Persistenz und Backup für Daten.
3. **ASG:** Hochverfügbarkeit und automatische Skalierung für die Python-Instanz.
4. **ELB:** Lastverteilung für erhöhte Verfügbarkeit und Leistung.

Diese Infrastruktur bietet eine robuste, flexible und skalierbare AWS-Architektur.

Fragen?

Vielen Dank für die Aufmerksamkeit! 