

Middleware – Cloud Computing – Übung

Hybride Cloud

Wintersemester 2025/26

Paul Bergmann, Christian Berger

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Systemsoftware)

<https://sys.cs.fau.de>



Lehrstuhl für Informatik 4
Systemsoftware



Friedrich-Alexander-Universität
Technische Fakultät

Hybride Cloud

Hybride Cloud & Virtualisierung

Aufgabe 2

Hybride Cloud

Hybride Cloud & Virtualisierung

- Öffentliche Cloud: Cloud-Dienste frei für jeden verfügbar
 - *aaS: „X as a Service“-Gedanke
 - Scheinbar unbegrenzte Ressourcen
- Private Cloud: IT- bzw. Cloud-Dienste werden z. B. von einem Unternehmen oder einer Einrichtung selbst betrieben
 - Interne Nutzung: Datenschutz und IT-Sicherheit
 - Aber auch: Bereitstellung von eigenen Ressourcen für öffentliche Nutzung
- **Hybride Cloud:** Mischform aus privater und öffentlicher Cloud
 - Sicherheitskritische Teile einer Anwendung laufen nur in der privaten Cloud
 - Skalierbarkeit, Ausdehnung auf öffentliche Cloud (z. B. beim Auftreten von Lastspitzen)

■ Notwendige Betriebsmittel

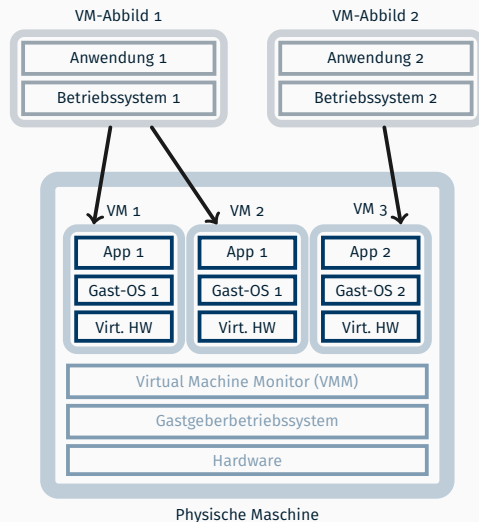
- Physische Maschine und Gastgeberbetriebssystem („Host“)
- Virtualisierungssoftware, die den Virtual Machine Monitor bereitstellt
- **Abbild der virtuellen Maschine**

■ Analogie zur Objektorientierung

- Das statische Abbild einer virtuellen Maschine entspricht einer **Klasse**
- Eine im Betrieb befindliche virtuelle Maschine ist die **Instanz** eines solchen Abbilds

■ Aufbau des Abbilds einer virtuellen Maschine

- **Dateisystem**, beinhaltet für gewöhnlich:
 - Kern des Gastbetriebssystems („Guest“)
 - User-Space-Komponenten des Gastbetriebssystems
 - Anwendung
- Meta-Informationen (VMM-spezifisch)

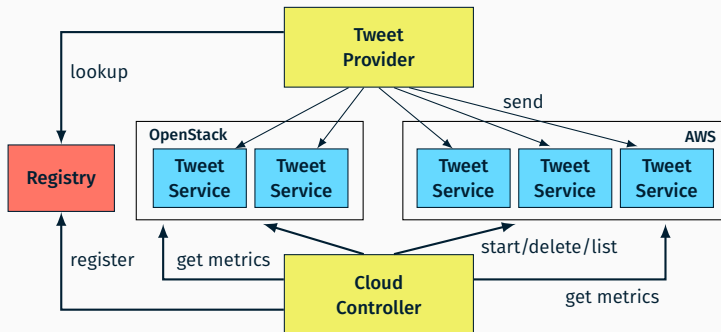


Hybride Cloud

Aufgabe 2

Aufgabe 2: Hybride Cloud

- Bereitgestellten Tweet-Service in hybrider Cloud ausführen
 - Ein bis maximal zwei VMs in privater Cloud
 - Öffentliche Cloud für Lastspitzen
- Teilaufgaben
 - Cloud-Controller für manuelle **Cloud-Ansteuerung** (VMs starten, beenden, auflisten)
 - **Lastverteilung** für Tweet-Anfragen im Provider, VMs per Registry abfragen
 - Erweiterter Cloud-Controller zur **dynamischen Skalierung** der VMs (nur 7.5 ECTS)



Aufgabe 2: Hybride Cloud

- Public Cloud: Amazon Web Services
 - Limitiertes Guthaben: Rund 10 US-Dollar Guthaben pro Gruppe
 - Guthaben kann lediglich für Amazon Web Services verwendet werden
 - Aktuelle AWS-Kosten: <http://aws.amazon.com/pricing/>
- Globaler Systemstatus der Amazon Web Services
 - Bei Störungen können (Teile der) Amazon Web Services ausfallen
 - Aktueller Status: <http://status.aws.amazon.com/>
- Private Cloud: OpenStack-Umgebung des Lehrstuhls
 - Ressourcen der drei Node-Controller sind **beschränkt**
 - Jederzeit auf faire Verwendung achten
- OpenStack-Infrastruktur
 - Bitte sendet bei Problemen oder Ungereimtheiten schnellstmöglichst eine E-Mail an i4mw-owner@lists.cs.fau.de

Achtung!

Bitte stets sicherstellen, dass **alle unbenutzten** Instanzen beendet (gelöscht) werden!

Aufgabe 2: Hybride Cloud

- Gemeinsame Schnittstelle: MWCloudPlatform
 - Instanzen starten / beenden / auflisten
 - Metriken der Instanzen abrufen
- MWCloudPlatformAWS: Betrieb des Dienstes in AWS EC2
 - Java 21 & Java-Bibliotheken bereits in vorkonfiguriertem Image (ami-ob44ee2dcf07ee291) enthalten
 - Passende Konfigurationsparameter userdata übergeben
 - Metriken aus AWS CloudWatch abfragen
- MWCloudPlatformOpenStack: Betrieb des Dienstes in OpenStack Nova
 - Erzeugung und Konfiguration eines eigenen VM-Abbilds
 - Installation des Grundsystems
 - Hinzufügen von Java, Java-Bibliotheken für Dienst
 - ↔ Schritt-für-Schritt Anleitung im Übungsteil „Erstellen eines VM-Abbilds in OpenStack“
 - Metriken aus Gnocchi abfragen
- Hinterlegen des JAR-Archivs des Tweet-Service auf AWS S3

Testen der Dienstlauffähigkeit

- Direkter Zugriff über HTTP-Anfrage (hier: GET-Anfrage)

```
> curl http://<ip-address>:<port>/tweetservice
```

→ Innerhalb der VM unter localhost erreichbar

- Direkter Zugriff über den Web-Browser möglich

- Instanz nicht erreichbar?

→ Konfiguration der cloudseitigen Firewall
durch die Security Groups kontrollieren

- Logs per SSH einsehen

```
> ssh -i <private_key (e.g., gruppeX.pem)> <user>@<ip_address>
```

- Benutzername für AWS: ec2-user, für OpenStack: cloud
- Bei Anmeldeproblemen Benutzernamen und SSH-Key kontrollieren
- Überprüfen, ob Java-Prozess läuft: > ps aux | grep java
oder > sudo systemctl status i4mw-service
- Fehlersuche: Protokolle durchsuchen mit > sudo less /var/log/syslog
oder > sudo journalctl -u i4mw-service

