**Wie ist der Begriff „Cloud“ entstanden? Warum heißt es Cloud?**

Der Begriff „Cloud“ kommt von der Darstellung von Netzwerken als Wolken in Diagrammen. Es symbolisiert, dass Ressourcen über das Internet bereitgestellt werden, ohne dass Nutzer die zugrunde liegende Infrastruktur sehen.

**Wie wird der Begriff Cloud definiert? Z.B. gemäss NIST**

Laut NIST ist Cloud Computing die Bereitstellung von IT-Ressourcen über das Internet, die nach Bedarf genutzt und bezahlt werden, ohne eigene Infrastruktur zu benötigen.

**Welche sind die 5 Merkmale einer Cloud?**

1. On-Demand Self-Service
2. Breit angelegter Netzwerkzugriff
3. Ressourcen-Pooling
4. Schnelle Elastizität
5. Messbarer Service

**Welche Cloud-Dienstleistungen kennen Sie?**

* **IaaS**: Infrastruktur wie Server und Speicher
* **PaaS**: Plattform für Entwickler
* **SaaS**: Software über das Internet (z.B. Google Docs)

**Welche Cloud-Anbieter kennen Sie?**

* AWS
* Microsoft Azure
* Google Cloud
* IBM Cloud
* Oracle Cloud

**Welche Cloud-Deployment-Modelle kennen Sie?**

* **Private Cloud**: Nur für eine Organisation
* **Public Cloud**: Für alle zugänglich
* **Hybrid Cloud**: Mischung aus beiden
* **Community Cloud**: Für mehrere Organisationen mit gemeinsamen Interessen

**Was sind Cloud-Service-Modelle?**

* IaaS: Infrastruktur
* PaaS: Plattform
* SaaS: Software

**Weshalb soll ich Dienste aus der Cloud beziehen? Was sind die Vorteile?**

* Kosteneffizienz
* Skalierbarkeit
* Flexibilität und Mobilität
* Wartungsaufwand entfällt
* Sicherheit

**Was sind die Nachteile?**

* Abhängigkeit vom Internet
* Sicherheitsbedenken
* Unvorhersehbare Kosten

**Welche Dienstleistungen werden in Ihrem Betrieb On-Premise betrieben?**

* Datenbanken
* Interne Anwendungen
* E-Mail-Server
* Backup-Lösungen

**Wie werden technologische Beiträge in der Cloud geteilt?**

* APIs
* Dokumentationen und Foren
* Open-Source-Projekte
* Cloud-Marktplätze

**Git-Grundbegriffe – kurz erklärt**

* **Repository**: Die Datenbank, in der Git alle Änderungen (Commits) speichert.
* **Working Tree**: Dein aktuelles Arbeitsverzeichnis, wo du am Code arbeitest.
* **Commit**: Ein gespeicherter Stand deines Projekts mit Änderungen und Infos wie Autor, Datum und Nachricht.
* **Clone**: Eine Kopie eines Git-Repositories, inkl. kompletter Versionsgeschichte.
* **Branch**: Eine Entwicklungsabzweigung für neue Features, Tests oder Bugfixes.
* **Tag**: Eine Markierung für bestimmte Commits (z. B. Versionen wie „v1.0“).
* **Fork**: Eine eigene Kopie eines Open-Source-Projekts, um unabhängig daran weiterzuarbeiten.

**✅ Container-Technologie / Virtualisierung**

* App + Abhängigkeiten laufen isoliert in Containern
* Container teilen sich den Kernel des Host-Systems
* Leicht, schnell, portabel

**✅ Vor- & Nachteile vs. Virtuelle Maschinen (VMs)**

**Vorteile von Containern:**

* Schneller Start
* Weniger Speicherplatz
* Einfaches Deployment
* Bessere Portabilität

**Nachteile:**

* Weniger isoliert als VMs
* Gleicher Kernel wie Host (kein anderes OS möglich)
* Sicherheitsrisiken bei falscher Konfiguration

**✅ Bekannte Produkte**

**VMs:**

* VMware
* VirtualBox
* Hyper-V
* Proxmox

**Container:**

* Docker
* Podman
* Kubernetes (für Orchestrierung)

**✅ Unterschiede: VM vs. Container**

| **Kriterium** | **VM** | **Container** |
| --- | --- | --- |
| Bereitstellung | Minuten | Sekunden |
| Speicherplatz | Groß (inkl. OS) | Klein (nur App + Libs) |
| Portabilität | Eingeschränkt (OS-abhängig) | Hoch (läuft überall mit Docker) |
| Effizienz | Weniger effizient | Sehr effizient |
| Betriebssystem | Eigenes OS pro VM | Nutzt Host-Kernel |

**✅ Können VMs immer ersetzt werden?**

* ❌ Nein, z. B. wenn man ein anderes OS braucht oder starke Isolation nötig ist

**✅ Self-Managed vs. Fully Managed**

| **Typ** | **Beschreibung** |
| --- | --- |
| **Self-Managed** | Alles selbst verwalten (Updates, Sicherheit) |
| **Fully Managed** | Anbieter kümmert sich um alles (wartungsfrei) |