Modul 346 - Cloud Lösungen konzipieren

KN00 Prüfungsvorbereitung:

Virtualisierung

- Virtualisierung => Das Aufteilen von physischen Ressourcen (Server, etc.) in mehrere virtuellen Maschinen.

- Hypervisor => Ein Prozess (Software) der die Virtualisierung macht, indem er dem physischen Host-Rechner ermöglich mehrere VMs zu betreiben.

(führt mehrere VMs auf einer einzigen physischen Maschine aus.)

Hypervisor Typ1 = (native) Läuft direkt auf der Hardware des Hostsystems

Hypervisor Typ 2 = (Hosted)Läuft als Applikation auf dem Betriebssystem und verwendet entsprechende OS-Prozesse. -> kann mehr Ressourcen zuordnen als vorhanden auf Hardware.

(parallelisiert CPU und lagert RAM auf Harddisk)

-Hyperscaler => sind grosse Cloud-Service-Anbieter die Dienste wie Computing und Storage im Unternehmenmassstab anbieten. (z. B. AWS, Azure, Google Cloud)

Hyperscale computing -> eine Methode der Datenverarbeitung, bei der die Softwarearchitektur skaliert und wächst wenn die Nachfrage steigt.

-Schichten von Cloud-Systemen => Infrastruktur, Plattform und Software (z. B. IaaS, PaaS, SaaS).

Betriebsmodelle

On premise -> Alle Hard- und Software werden in eigenen oder gemieteten Datenzentren selbst betrieben.

Hyprid Cloud -> Server und Applikationen sind teilweise "On Premise" und teilweise in der Cloud.

Cloud-Native -> In diesem Modell ist die Komplette infrastruktur und alle Applikationen in der Cloud.

Unterteilung **Public** und **Privat** Cloud
(Public = offen zugänglicher Server, Private = Firmeninterner Server)

- ➔ Unabhängige Apps / Services (Kein bezug privat und public)
- ➔ Verteilte Systeme (DB -> OnPremise / APP,Service -> Public Cloud)
- ➔ Synchronisierte Systeme (DB -> OnPremise / DB -> Public Cloud)

Deployment-Modelle: Public

› Bereitstellung im **geteilten** Rechenzentrum

› Keine Kontrolle über die Infrastruktur

Deployment-Modelle: Private

› Bereitstellung im **eigenen** Rechenzentrum

› Volle Kontrolle über die Infrastruktur

Deployment-Modelle: Hybrid

› Mischung aus Public- und Private-Cloud Anbietern

Servicemodelle

-IaaS (Infrastructure as a Service) = Physische Server werden durch den Anbieter verwaltet und gewartet. Betriebssystem und Software werden selbst verwaltet.
-> Microsoft Azure, AWS, Cisco Metapod etc.

(Beispiel: Küche mieten mit Küchengeräten, noch bessere Küche mieten für noch bessere Geräte)

-PaaS (Plattform as a Service) = Laufzeitumgebung für Applikationen wird durch Anbieter zur Verfügung gestellt. Applikation wird selbst verwaltet
-> AWS Elastic Beanstalk, Windows Azure, Heroku, Salesforce, Dropbox, Microsoft 365 etc.

(Beispiel: Man geht im Restaurant essen, muss das Essen nicht selbst kochen aber kann den Kellner wünsche oder Forderungen machen.)

-SaaS (Software as a Service) Die gesamte Applikation wird durch den Anbieter zur Verfügung gestellt.
-> Google Workspace, Dropbox, Salesforce, Microsoft 365 etc.

(Beispiel: All you can eat buffet, man kann auf jedes Gericht zugreifen und so viel essen wie man will. Muss nicht daran denken wie es gemacht ist)

-FaaS (Function as a Service) / Serverless = Laufzeitumgebung inkl. Middleware (z.B. HTTP Server) wird zur Verfügung gestellt. Nur die eigentliche Funktion selbst wird selbst verwaltet

Der Vorteil sind reduzierte Kosten. Nur wenn die Funktion ausgeführt wird die kosten anfallen.
-> AWS Lambda etc.

(Skaliert sich automatisch)

=> Abhängigkeit: SaaS baut auf PaaS auf / PaaS baut auf IaaS auf

(Cloud-) Migrationsmodelle

Definition **Migration** => Daten oder Software von einem System in ein anderes verschoben. / Prozess bei dem ein Teil eines Systems in ein anderes gewechselt wird. (Umstellungsprozess)

- **Rehosting** (**Lift & Shift**) = Dabei wird eine bestehende On-Premise-App mit minimalen anpassungen in die Cloud übertragen. Die App wird auf virtuellen Servern installiert (IaaS aufsetzen).

- **Replattforming** (**Lift & Reshape**) = Eine bestehende On-Premise-App wird so angepasst, dass sie auf die bestehende Infrastruktur optimal gehostet werden kann. . (mit PaaS hosten)

- **Refactoring** / **Replace** = Die App wird so umgeschrieben, dass sie mit den Cloud Services optimal läuft. Meistens wird dabei auf Microservices mit Kubernetes oder auf das FaaS-Modell gewechselt. Man kann aber auch Mischformen nutzen.

- **Repurchasing** = Die bestehende App wird mit einer bestehende SaaS lösung abgelöst. (Keine Eigene App sondern Externe App verwendung)

Speichermodelle

- **Hot Storage** = Speicher, welcher oft aufgerufen werden kann und nur sehr kleine Verzögerungen hat.

  (Verwendung: z.B. beim Zugriff auf Datenbanken, Echtzeit-Analysen oder laufende Prozesse in virtuellen Maschinen)

- **Warm Storage** = Speicehr welcher regelmässig aufgerufen wird, aber nicht die Performance eines Hot storage bieten muss. (auch Cool Storage genannt, z.B: S3)

  (Verwendung: Für Daten, die weniger oft benötigt werden, aber dennoch schnell verfügbar sein sollen. Backups, Content Distribution (z.B. Videostreams))

- **Cold Storage** = Speicher welcher sehr selten aufgerufen wird und dadurch auch nicht schnell sein muss. (auch Archive Storage genannt)

  (Verwendung: Für Datenarchivierung und langfristige Aufbewahrung, bei denen Geschwindigkeit unwichtig ist. Backup-Archive oder historische Daten)

Definition **Persistenz** => ein prozess in dem ein programm daten für den späteren gebrauch speichert.

- Persistenter Speicher: Daten bleiben auch nach dem Ausschalten des Geräts erhalten (z.B. SSD, HDD).

- Flüchtiger Speicher: Daten gehen verloren, wenn der Strom abgeschaltet wird (z.B. RAM).

- Virtuelle Instanzen: In Cloud-Umgebungen wird flüchtiger Speicher oft für laufende Systeme genutzt, persistenter Speicher für Backups oder Datenbanken.

Lese- und Schreibgeschwindigkeiten -> In fast allen Fällen ist das Lesen schneller als das Schreiben, weil der Schreibprozess technisch aufwändiger ist und zusätzliche Schritte wie Löschen, Speichern und Validieren umfasst.

Skalierung (anpassen/veränderte massstäbe->grössenveränderung)

- **Vertikale Skalierung (Scale up)**

Hier erhöht man die Ressourcenzuteilung einer Virtuellen Instanz (Mehr RAM, höherer anzahl CPUs oder mehr Speicher).

-> Applikation/Umgebung muss heruntergefahren werden

- **Horizontaler Skalierung (Scale out)**

Hier wird nicht die Instanz verändert sondern weitere Instanzen werden hinzugefügt. Die last wird so auf mehrere Instanzen verteilt. (Prinzip wie Hyperscaler)

-> Applikation/Umgebung muss nicht heruntergefahren werden

➔ **Load Balancer**

Ein Load Balancer wird benötigt, um die Last auf mehrere Server zu verteilen, bei einer horizontalen Skalierung.

Ein Load Balancer verteilt Anfragen an verschiedene Instanzen. Er kann die Anfragen an die am wenigsten ausgelasteten Instanzen verteilen.

➔ Auto Scaling = Vorgang bei der automatisch neue Instanzen hinzugefügt und wieder entfernt werden. Das abhängig von der Auslastung der bestehenden servern.

-> Applikation kann auf diese Weise jederzeit optimal antworten.

Netzwerk und Sicherheit

- **VPC (Virtual Private Cloud)**

Bezeichnet das eigene Internet/Netzwerk innerhalb eines Anbieters.

- **Subnetz**

Subnetz innerhalb der VPC. Alle ihre virtuellen Server werden in den Subnetzen plaziert. (Netzwerk innerhalb eines Netzwerks)

- **Gateway**

Steht hier stellvertretend für das Routing zwischen dem Internet und den Privaten Subnetzen.

- **Netzwerkinterface**

Eine physische oder virtuelle Schnittstelle, über die Geräte mit einem Netzwerk verbunden werden.

- **Security Group**

Im Prinzip eine NAT-Firewall, die Anfragen auf spezifischem Port blockiert oder weiterleitet.

- **Firewall**

Kontrolliert Datenfluss zwischen internes Netzwerk und externen Netzwerk und blockiert bösartiger Datenverkehr.

- **Virtueller Netzwerk-Adapter**

Eine virtuelle Instanz hat zwei virtuelle Netzwerk-Adapter: einen für das private Subnetz und eine öffentliche IP.

(ermöglicht es VMs mit einem Netzwerk zu verbinden)

- **IP-adresse**

Eine eindeutige Adresse, die ein gerät im Netzwerk identifiziert

- **IP-Range**

Ein Bereich von IP-Adressen der durch eine Startadresse und ein Netz Maske definiert wird. (unterschied zu IP-Adresse, umfasst IP-Range mehrere Adressen)

- **Port**

Ein Port ist eine virtuelle Türe, über die Applikationen daten senden und empfangen kann. Ports dienen dazu, den Datenverkehr für bestimmte Protokolle und Dienste zu trennen.

3306 - MySQL

80 - Http

22 - SSH

443 - HTTPS

Cloud-init

- **Infrastructure as code**

Infrastruktur as Code automatisiert die Konfigurationen eines Systems. (automatisiert die Einrichtung von Infrastruktur.) Dabei ermöglicht es die Entwicklung, Bereitstellung und Skalierung von Cloud-Anwendungen mit höherer Geschwindigkeit, geringerem Risiko, niedrigeren Kosten und Versionierung.

(Tools : Terraform, Ansible, CloudFormation, Cloud-init)

- **YAML**

Ist ein Datenorientiertes Format, das häufig in Konf Dateien verwendet wird.

> Cloudinit nutzen YAML um initialiserungsdaten für VMs zu definieren. Die Cloud-init-Daten werden beim Bootvorgang angewendet.

Elemente von Cloud-Init:

> #cloud-config = Gibt an, dass dies eine Cloud-Init Configfile ist.

> users = Hier wird ein Benutzer definiert.

> name: ubuntu = Erstellt einen Benutzer mit dem Namen "ubuntu".

> sudo: ALL=(ALL) NOPASSWD:ALL = Dieser Benutzer kann alle Befehle mit Administratorrechten (sudo) ausführen, ohne ein Passwort eingeben zu müssen.

> groups: users, admin = Fügt den Benutzer zu den Gruppen "users" und "admin" hinzu.

> home: /home/ubuntu = Setzt das Home-Verzeichnis des Benutzers auf /home/ubuntu.

> shell: /bin/bash = Legt die Standardshell des Benutzers auf "bash" fest.

> ssh_authorized_keys: = Hier wird ein öffentlicher SSH-Schlüssel angegeben, mit dem sich der Benutzer per SSH anmelden kann.

> disable_root: false = Aktiviert den Root-Benutzer

> package_update: true packages: - curl - wget = Führt ein Update der Paketquellen aus, um sicherzustellen, dass die neuesten Softwarelisten verwendet werden.

> Curl ist ein Tool, um Daten von oder zu einer URL zu übertragen. Wget ist ein weiteres Tool, um Dateien von einer URL herunterzuladen.

> runcmd = Führt Kommandos nach der Initialisierung aus (z. B. - apt update).

> write_files = Erstellt oder schreibt Dateien auf der Instanz.

Weitere Begriffe:

Instanzen = Virtuelle Maschine