**On-Premises** **Beschreibung**: Software und Anwendungen lokal im Unternehmen betrieben und verwaltet **Vorteile**: Hohe Kontrolle, Datenschutz, einmalige Lizenzkosten, Unabhängigkeit, Offline-Zugriff. **Nachteile**: Hohe Kosten, Zeitaufwand für Wartung, Laufende Kosten, Mangelnde Skalierbarkeit. **Beispiel**: Ein Finanzunternehmen zur Buchhaltung, volle Kontrolle sensible Finanzdaten zu behalten.

**Private Cloud Beschreibung**: Ressourcen über Cloud-Dienste bezogen, nur für Unternehmen. Bereitstellung im eigenen Rechenzentrum. **Vorteile**: Anpassbarkeit, flexible Skalierung, erhöhte Sicherheit, volle Kontrolle über die Infrastruktur. **Nachteile**: Verwaltungsaufwand, höhere Kosten Public Cloud. **Beispiel**: Unternehmen hostet Entwicklungsplattform in einer privaten Cloud, um Anpassbarkeit und Sicherheit zu gewährleisten.

**Public Cloud Beschreibung**: IT-Services über Internet von Cloud-Anbietern gemietet, Verwaltung durch Anbieter. Bereitstellung geteilten Rechenzentrum. **Vorteile**: Skalierbare Abonnements, reduzierte Hardware-Kosten, Entlastung der Ressourcen, **Nachteile**: Eingeschränkte Anbieterwahl, Sicherheitsrisiken, keine Kontrolle über die Infrastruktur. **Beispiel**: Ein E-Commerce nutzt, um Webanwendung zu hosten & flexibel steigende Benutzerzahlen zu reagieren. > AWS, Microsoft Azure

**Wichtiger Unterschied** Hauptunterschied Public & Private Cloud, dass Public Cloud-Ressourcen vielen Kunden gemeinsam genutzt, Private Cloud-Ressourcen exklusiv für einzelnes Unternehmen

**Betriebsmodell** Ein Softwareunternehmen hat ein Betriebsmodell, bei dem es Entwicklungs-, Test- und Supportabteilungen gibt, die an verschiedenen Standorten arbeiten.

**A diagram of a software system

Description automatically generated with medium confidenceA diagram of a software system

Description automatically generated with medium confidenceServicemodell** Das gleiche Unternehmen hat ein Servicemodell, bei dem es seine Software über Abonnementpläne online verkauft und Kunden über Chat und E-Mail unterstützt.

|  |  |
| --- | --- |
| **IaaS (Infrastructure as a Service)** | **Selbstgemachte Pizza aus Zutaten des Supermarkts. Du kontrollierst die Zubereitung, aber die Vorbereitung erfordert deine Mühe. -> aws, azure** |
| **PaaS (Platform as a Service)** | **Pizza im Restaurant mit vorgefertigtem Teig und Auswahl an Belägen. Du gestaltest Pizza, aber Basis und Infrastruktur sind bereitgestellt. -> heroku(appl. Bereitstellen, skalieren)** |
| **SaaS (Software as a Service)** | **Bestellung einer fertigen Pizza beim Lieferdienst. Du nutzt die Anwendung, ohne Installation oder Wartung. -> gmail, google docs** |
| **FaaS (Function as a Service)** | **Vorgefertigte Pizzateile, die du in deinen eigenen Pizzateig einfügst. Du fügst Geschäftslogik hinzu, die Plattform kümmert sich um den Rest.** |

**A diagram of a software

Description automatically generatedA diagram of a system

Description automatically generated**

**Begriffe**

* EC2 → Elastic Compute Cloud
* S3 → Simple Storage Service
* AWS → von Amazon
* Security Group → Firewall → SSH Root Port 22 um reinzukommen
* RDP → Port 3989

**Instanztypen**

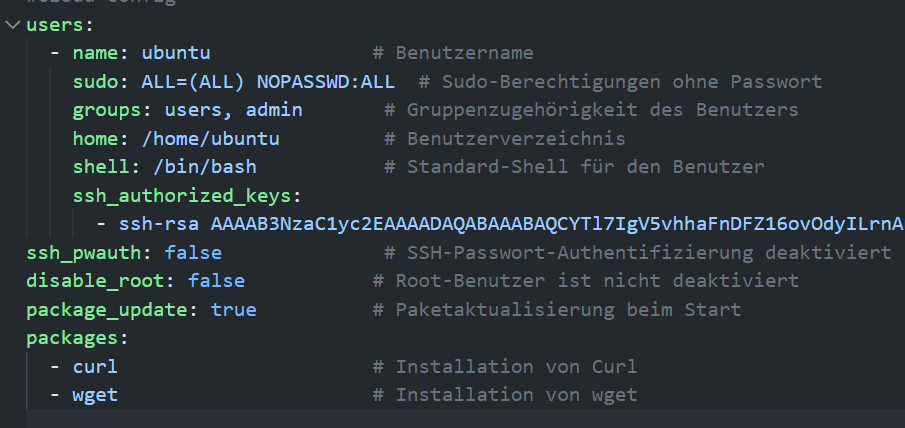
**T2.micro -** Einfache Webanwendungenmit geringem Traffic und Rechenanforderungen. Entwicklungsumgebungen: für Entwicklungs- und Testumgebungen, bei denen kostengünstige Ressourcen benötigt werden.- kostengünstig,

**m4.xlarge** gut geeignet für datenintensive Anwendungen und Datenbankserver. Anwendungen mit hoher Leistungsanforderung: Ideal für Anwendungen, die eine hohe Rechenleistung und Speicherbandbreite benötigen.

S3-**Bucket**: Virtueller Speicherplatz in AWS zum Speichern von Objekten.

**Konfiguration**: Anpassen von Einstellungen wie Berechtigungen und Region.

**Objekte** ablegen: Hochladen von Dateien und Daten in den Bucket.

**Vorteile**: Hohe Haltbarkeit, Skalierbarkeit und Integration mit AWS.

**EC2**-Instanz: Virtuelle Server in AWS.

**Cloud**-Init-Skript: Automatisiert die Konfiguration der EC2-Instanz.

**Ziel**: Webserver, der ein Bild aus dem S3-Bucket anzeigt.

**EBS**-Volume: Zusätzlicher Datenspeicher für EC2-Instanzen.

**Hinzufügen** im Betrieb: Dynamisches Hinzufügen ohne Ausfallzeiten.

**Zweck**: Zusätzlicher Datenspeicher für die Windows-Server-Instanz.

**Persistenz**: Wie dauerhaft ist der Speicher?

**Geschwindigkeit**: Wie schnell kann auf den Speicher zugegriffen werden?

**Sicherheit**: Welche Sicherheitsmaßnahmen gibt es?

**Standort**: Wo werden die Daten physisch gespeichert? Weitere Charakteristiken: Welche weiteren Eigenschaften sind relevant?

**Verbinden zur Instanz**

ssh ubuntu@34.228.190.14 -i jann-1.pem -o ServerAliveInterval=30

#### Welcher Schlüssel wird von welchem überschrieben?

Der erste Schlüssel überschreibt den zweiten Schlüssel - **jann-1.pem** überschreibt **jann-2.pem**

->Grund dafür ist, dass zu Beginn bei allen Einstellungen ein yaml File generiert wird - falls ich dann in den additional Features ein neues yaml File einfüge, mit einem anderen SSH Key, wird der neue SSH Key den alten überschreiben

Unterschied Passwort und SSH

- \*\*Passwort\*\*:

- Textbasierte Authentifizierung. Kann leicht erraten oder geknackt werden, wenn es schwach ist.

- \*\*SSH-Schlüssel\*\*:

- Kryptografisches Schlüsselpaar: öffentlicher und privater Schlüssel. Sicherer als ein Passwort, da es auf komplexer Kryptographie basiert.

1. \*\*Einzelne Cloud-Instanz\*\*: In dieser Struktur wird alles in einer einzelnen Cloud-Umgebung bereitgestellt und verwaltet. Dies kann eine einfache Möglichkeit sein, mit der Cloud zu beginnen, eignet sich jedoch möglicherweise nicht für komplexe Anforderungen.

2. \*\*Multi-Cloud\*\*: Bei dieser Struktur werden Ressourcen und Dienste über mehrere Cloud-Anbieter verteilt. Dies kann die Verme dung von Vendor-Lock-In und die Nutzung der Stärken verschiedener Anbieter ermöglichen.

3. \*\*Hybrid-Cloud\*\*: Hierbei handelt es sich um eine Kombination aus Cloud- und On-Premises-Ressourcen. Diese Struktur bietet Flexibilität und kann für Datenschutz- oder Compliance-Anforderungen relevant sein.