

# Daten

# Optionen

**Virtuelle  
Maschinen**

**Kubernetes**

**gemanagte  
Datenbanken**

# Eigenbetrieb in virtueller Maschine

- Große Auswahl an Datenbanken
- Feingranulare Konfigurationsmöglichkeiten
- Etablierte Installations- und Betriebsverfahren
- Möglichkeit dedizierter Ressourcen (IO, CPU, Arbeitsspeicher)
- Relativ geringe Ressourcenkosten

## Aber

- u.U. Kompliziertes Setup
- Erhöhter Aufwand für Backups
- Relativ aufwendig zu skalieren

# Eigenbetrieb in Kubernetes

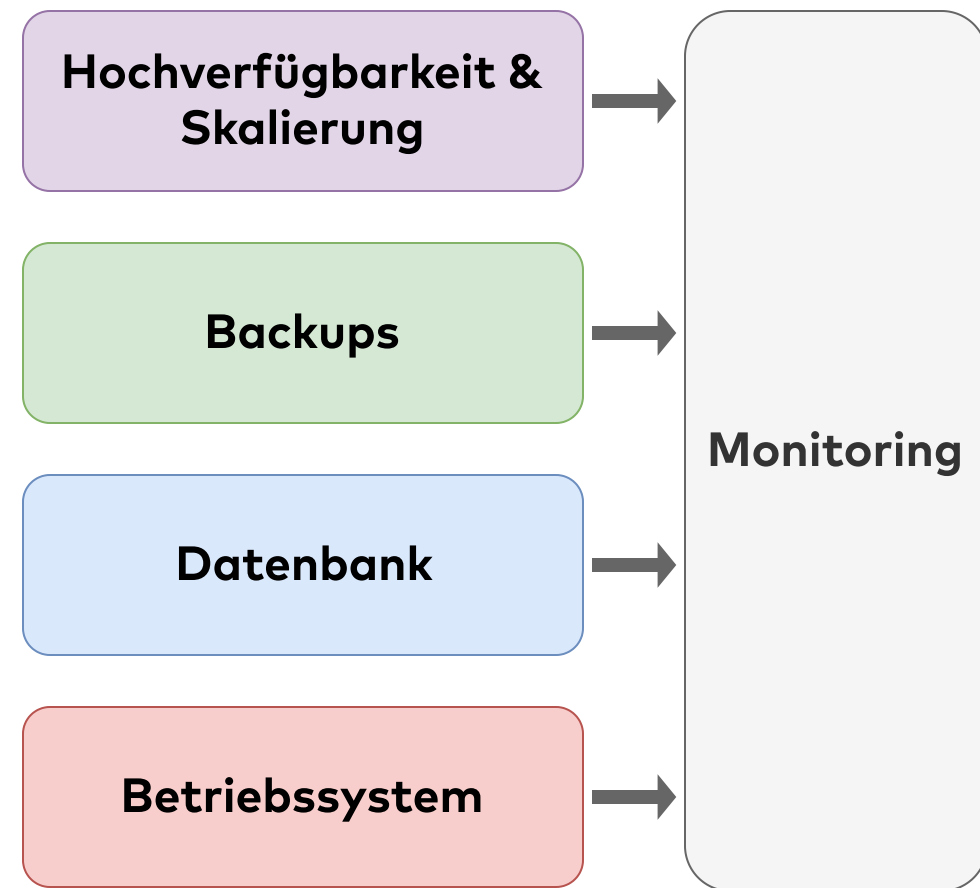
- Jede Datenbank die in einem Container läuft kann verwendet werden
- Zahlreiche Konfigurationsmöglichkeiten

Aber

- Aufwendiges Setup
- Persistent Volumes notwendig
- IO (Latenz, Durchsatz) und Arbeitsspeicher Anforderungen der Datenbank müssen erfüllt werden
- Funktionsumfang und Qualität von Operatoren noch schwankend
- Aufwand für Backups

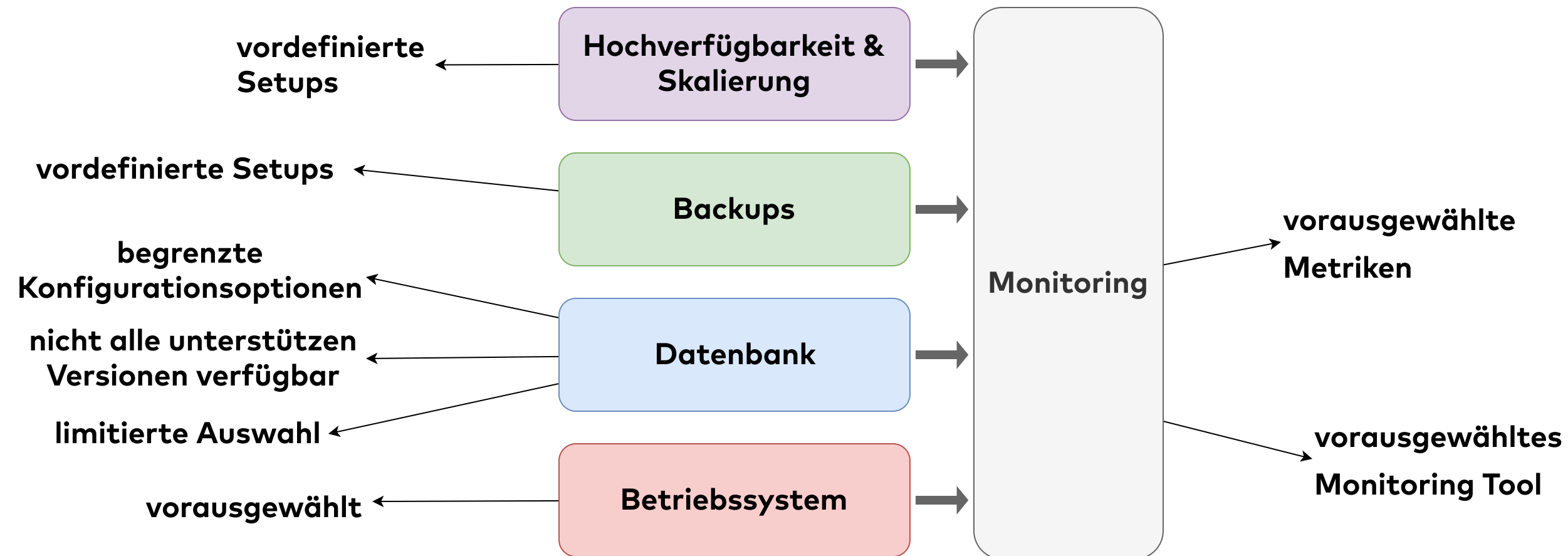
# Vorteile von Managed Datastores

## reduzierte Betriebsaufwände



# Nachteile von Managed Datastores

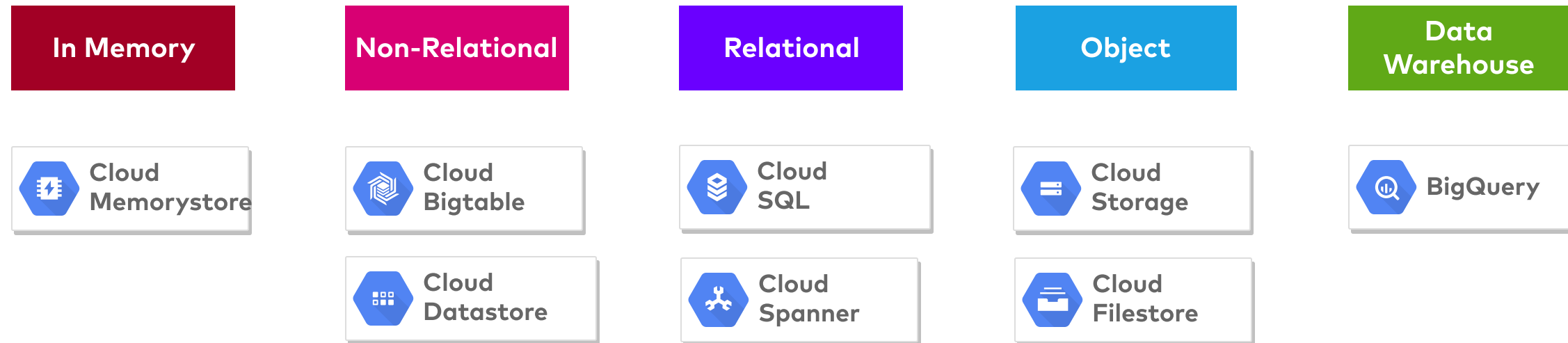
## Limitierte Optionen



**Empfehlung:**

**Wenn keine dedizierte  
Datenbankadminerfahrung vorhanden ist,  
mit managed Datenbank starten**

# Übersicht Google Datenspeicher





# **In-Memory: Memorystore**

- **Redis**
  - **Key-Value-Store**
  - **Aktuelle Version: 3.2.**
- **Einsatzbereich: Caching, Session Management, Gaming Leaderboards, etc.**
- **2 Service Tiers:**
  - **Basic (Standalone Instanz)**
  - **Standard (High Availability)**
- **5 Capacity Tiers**
  - **Unterscheiden sich in Größe und Netzwerkdurchsatz**



# Nicht-Relational: Cloud Bigtable

- **Verteilte, spaltenorientierte Datenbank**
  - "Map von Map von Maps" - Row, Column, Timestamp
- **Entstehung: Speichersystem für Googles Suchindex**
  - Optimiert für Geschwindigkeit und Verfügbarkeit
- **Für Anwendungen mit**
  - riesigen Datenmengen
  - extremen Schreib- und Lesezugriffen
  - einfachen, key-basierten Abfragen
  - keinen sequentiellen Keys
- **Closed Source**
  - Reimplementiert als Apache HBase
  - Unterstützt HBase API

# Cloud Bigtable: Datenmodell

	Column Family 1		Column Family 2	
Row Key	Column A	Column B	Column C	Column D
r1	<div>2018-12-01</div> <div>2018-12-02</div>	<div>2018-11-30</div>	<div>2018-11-18</div> <div>2018-11-28</div>	<div>2018-11-18</div> <div>2018-11-20</div> <div>2018-12-02</div>
r2	<div>2018-11-21</div> <div>2018-11-22</div> <div>2018-11-23</div> <div>2018-12-01</div>		<div>2018-11-18</div> <div>2018-11-19</div> <div>2018-12-01</div>	<div>2018-11-29</div>



# Nicht-Relational: Cloud Datastore

- **Dokumentenorientierte Datenbank**
  - **Dokument: Typ, Key, Sammlung von Key-Value Attributen**
  - **Schemafrei**
- **Für Anwendungen mit**
  - **hohen Anforderungen an Skalierbarkeit**
  - **semi-strukturierten, insbesondere hierarchischen Daten z.B. Benutzerprofilen**
- **Wird seit 8 Jahren für App Engine verwendet**
- **Keine referentielle Integrität, eventual consistency**
- **Multi-Region Replikation möglich, keine globale Replikation**
- **Unterstützung von Multi-Dokument Transaktionen**
- **Attribut Queries erfordern Index Definition für die zu suchenden Attribute**



# Nachfolger: Firestore

- **Betaversion, Noch nicht in allen Regionen verfügbar**
- **2 Modi**
  - **Native Mode**
  - **Datastore Mode (Abwärtskompatibel mit DataStore)**
- **Automatisches Upgrade von Datastore zu Firestore**
- **Native Mode**
  - **Für Web, Mobile und IoT Anwendungen**
  - **Multi-device Synchronisation, Offline support**
  - **Mobile Bibliotheken**
- **Beide**
  - **Strong consistency**

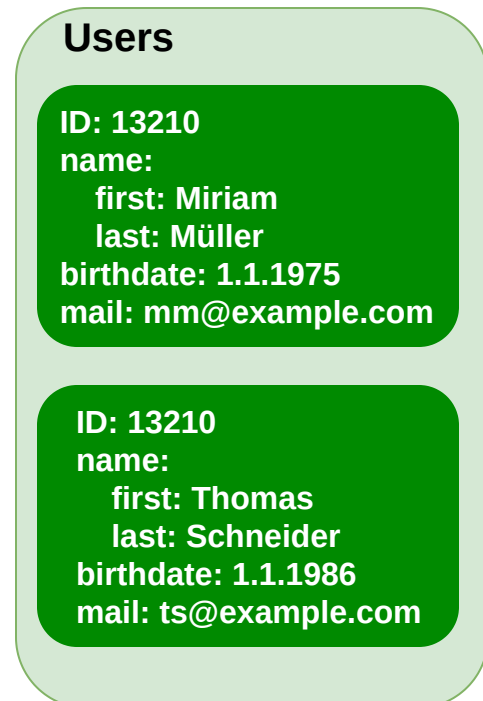


# Cloud Firestore: Datenmodell

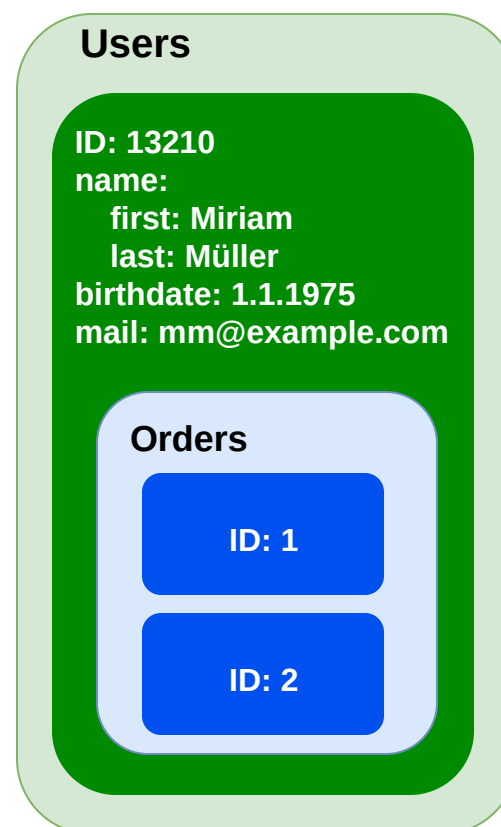
## Dokument

ID: 13210  
name:  
  first: Miriam  
  last: Müller  
birthdate: 1.1.1975  
mail: mm@example.com

## Dokumente werden in Collections gespeichert



## Hierarchische Daten via Subcollections





# Relational: Cloud SQL

- **Gemanagte relationale Datenbank**
  - PostgreSQL (9.6, 11)
  - MySQL (5.6 oder 5.7)
- **Läuft in virtueller Maschine, gehostet bei Google Cloud Engine**
- **Ähnlich wie Amazon RDS**
- **Unterstützung von Read Replicas und Failover Replicas**
  - Keine Multi-Master Unterstützung
- **Automatische Backups möglich**
- **Veränderung der initialen Konfiguration von CPUs und Memory**
  - Downtime von mehreren Minuten
  - Speicherplatz wächst automatisch mit



# Relational: Cloud Spanner

- "NewSQL"
  - SQL, Strong consistency, referentielle Integrität, Transaktionen
  - Horizontale Skalierung
- Für Anwendungen mit extremen Schreib- und Lesezugriffen die eine relationale Datenbank benötigen
- Synchrone Replikation
- Automatisches Sharding der Daten
- Besonderheit
  - Jede Tabelle muss einen Primary Key haben
  - Foreign Keys nur mittels sogenannter interleaved tables
- Closed Source
  - Keine Überprüfung der Konsistenzversprechen möglich
- Sehr teuer





# Relational: Cloud Spanner

	Cloud Spanner	Traditional Relational	Traditional Non-Relational
Schema	✓ Yes	✓ Yes	✗ No
SQL	✓ Yes	✓ Yes	✗ No
Consistency	✓ Strong	✓ Strong	✗ Eventual
Availability	✓ High	✗ Failover	✓ High
Scalability	✓ Horizontal	✗ Vertical	✓ Horizontal
Replication	✓ Automatic	⚙️ Configurable	⚙️ Configurable

Quelle: <https://cloud.google.com/spanner/>



# Objektspeicher: Cloud Storage

- Geeignet für Bilder, Mediendateien, Backups
- Ähnlich zu Amazon S3
- Objekte werden in Buckets mit global eindeutigem Namen gespeichert
  - Verzeichnisse möglich
- Automatische Versionierung möglich
- Automatische Löschpolicies möglich
- Verschicken von Benachrichtigungen
- Statisches Webhosting



# Cloud Storage: Storage Classes

	Multi-regional	Regional	Nearline	Coldline
Häufigkeit des Datenzugriffs	sehr häufig	häufig innerhalb einer Region	weniger als einmal im Monat	weniger als einmal im Jahr
SLA Verfügbarkeit	99,95%	99,90%	99,00%	99,00%
Zugriffszeit		Millisekunden		
Preis Speicherung				
Preis Abruf				



# Dateispeicher: Cloud Filestore

- **Gemeinsam genutzte Netzlaufwerke**
- **Network Attached Storage**
  - **NFSv3**
- **Kein automatisches Backup oder Failover**
- **Geschwindigkeit**
  - **Lesen: bis 700 MB/s**
  - **Schreiben: bis 350 MB/s**



# Data Warehouse: BigQuery

- Data Warehouse für die Speicherung und schnelle Auswertung riesiger Datenmengen
- Einsatzbereich: Erstellung von Analysen und Dashboards
- Unterstützt SQL
- Unterstützt NICHT: Foreign Keys, unique Constraints, Transaktionen
- Zusätzliche Datentypen: Geospatial, JSON, Arrays
- Mit anderen Google Services integriert
  - Außerdem: Connector für Excel
- Audit Logging möglich
- Preisberechnung für Abfragen abhängig von verarbeiteter Datenmenge

# Vergleich Data Stores

	Memorystore	Datastore	SQL	Spanner
Transaktionen	Nein	Ja	Ja	Ja
Komplexe Abfragen	Nein	Nein	Ja	Ja
Kapazität	300 GB	Terabytes+	500 GB	Petabytes
max. Größe Speichereinheit	512 MB/Wert	1 MB/Dokument	Abhängig von DB	4 GB/Zeile

	Storage	Filestore	Bigtable	BigQuery
Transaktionen	Nein	Nein	Single-row	Nein
Komplexe Abfragen	Nein	Nein	Nein	Ja
Kapazität	Petabytes+	Terabytes	Petabytes+	Petabytes+
max. Größe Speichereinheit	5 TB/Objekt	16 GB	~10 MB/Zelle ~100MB/Zeile	10 MB/Zeile

# Weitere Möglichkeiten

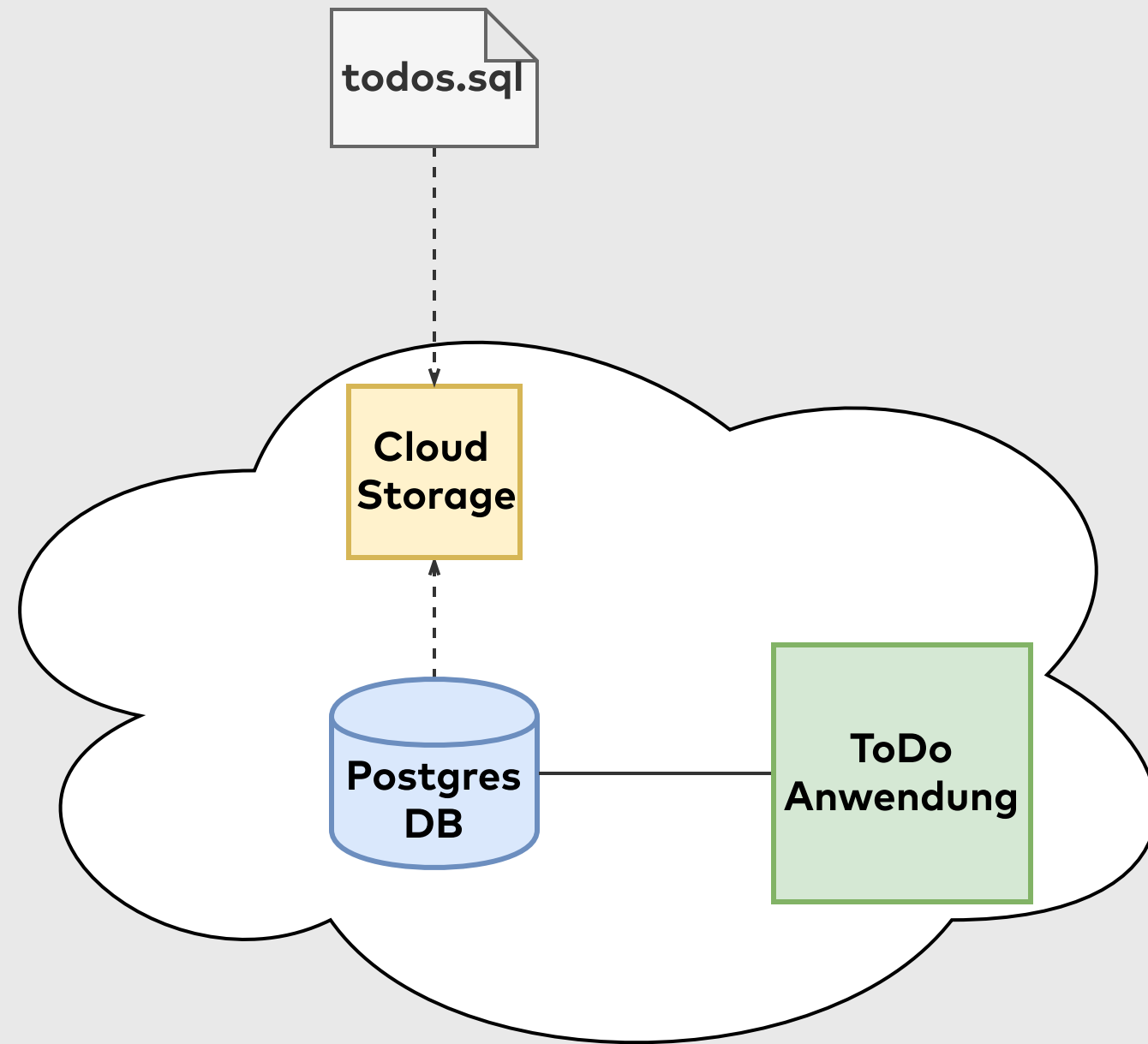
## Drive Enterprise

- Kollaboratives Arbeiten: Präsentationen, Textdokumente, etc.
- Dateiaustausch & Speicherung
- API Zugriff möglich

## Persistent Disk

- Block-Storage für virtuelle Maschinen oder Kubernetes
- SSD oder HDD
- Bis zu 64 GB Speicherplatz

# Demo







# Pub/Sub

- **Messaging Service**
  - **Produzenten senden Messages an Topics**
  - **Konsumenten können Topics via Subscription abonnieren**
  - **Pro Subscription wird eine Message "at least once" zugestellt**
  - **Mehrere Konsumenten können eine Subscription gemeinsam bearbeiten**
- **Flexible Kommunikation: one-to-many, many-to-one, and many-to-many**
- **Message Delivery: Pull oder Push (HTTP Webhook)**
  - **Pull: Empfang muss vom Empfänger bestätigt werden**
  - **HTTP Statuscode 204**
- **Replay/Seek in Beta**
- **Abrechnung erfolgt auf Basis des Datenvolumens**



# DataLab

- Jupyter Notebooks in der Cloud
  - Daten Analysieren und Visualisieren
- Unterstützt Python, SQL, JavaScript
  - Vorinstallierte Python Bibliotheken: google-cloud-dataflow, matplotlib, numpy, pandas, PyYAML, requests, scikit-learn, seaborn, tensorflow, etc.
  - Weitere Bibliotheken: Installation mit pip
- Kann um GPUs erweitert werden
- Integration mit Cloud Storage, Big Query und CloudML

# Demo

