

# DATENBANK LÖSUNGEN

mit Azure

Peter Schneider  
Trainer und Consultant

- Cloud Services, Data Platform, Azure Portal
- Datenbanken in Virtuelle Maschinen
- Azure SQL Datenbanken und Elastic Database Pools
- NoSQL, DocumentDB
- Ausblick

# AZURE ÜBERBLICK

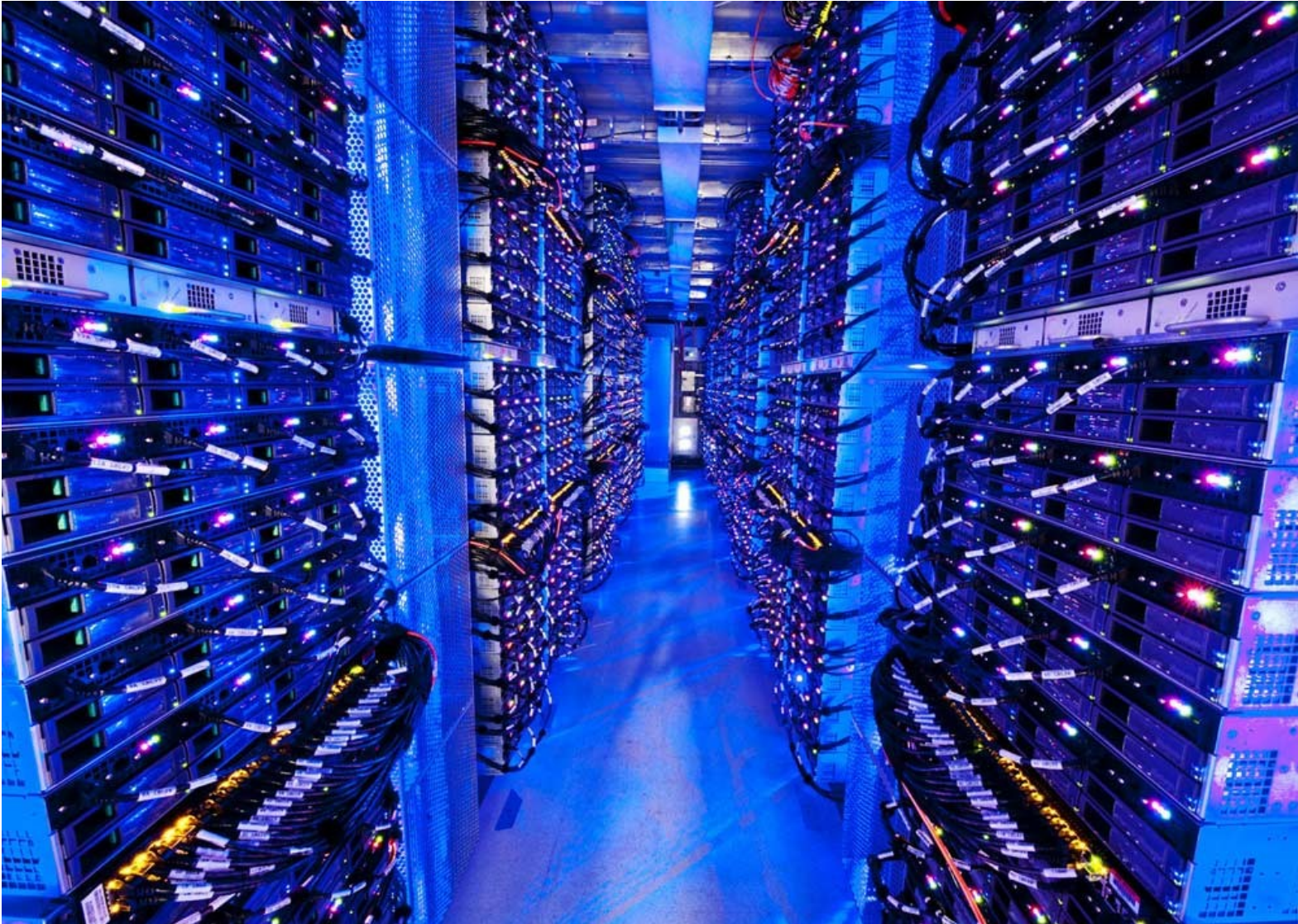
# Data Center in Chicago



# Azure Container



# Azure Container Innenansicht

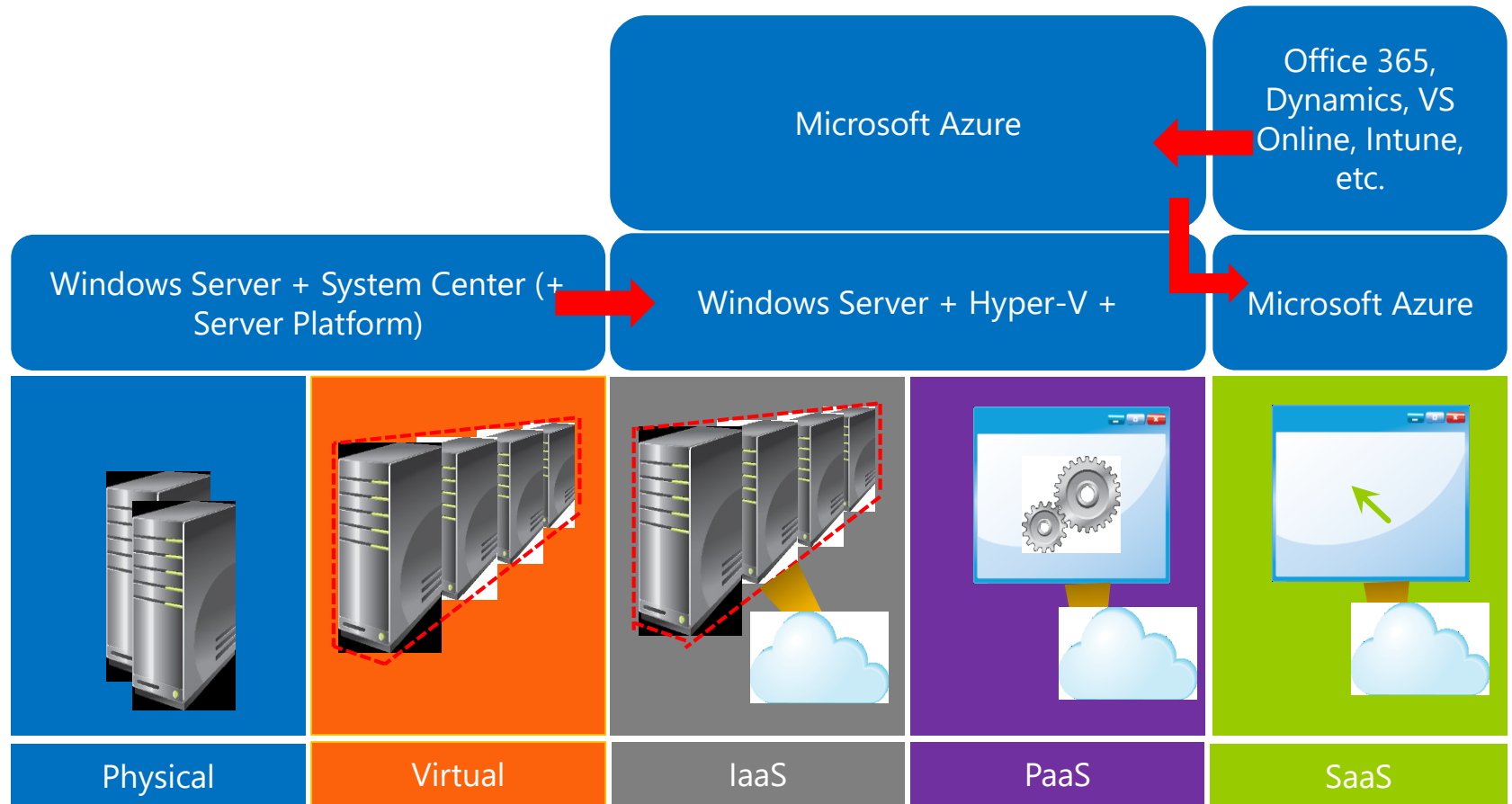




# Azure Service Map

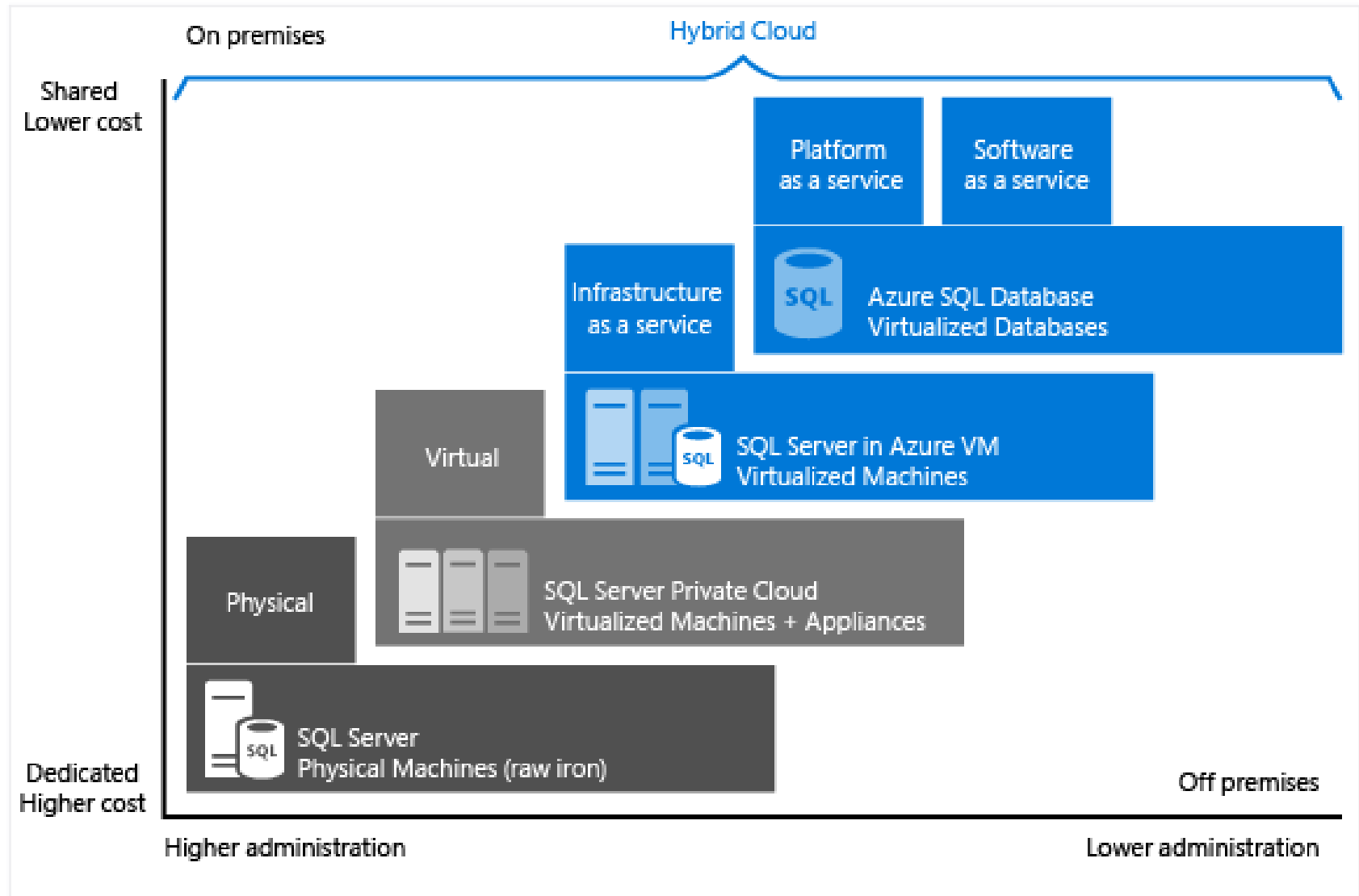


# Cloud Services

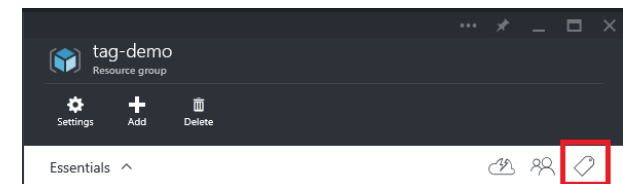




# Azure Daten Plattform

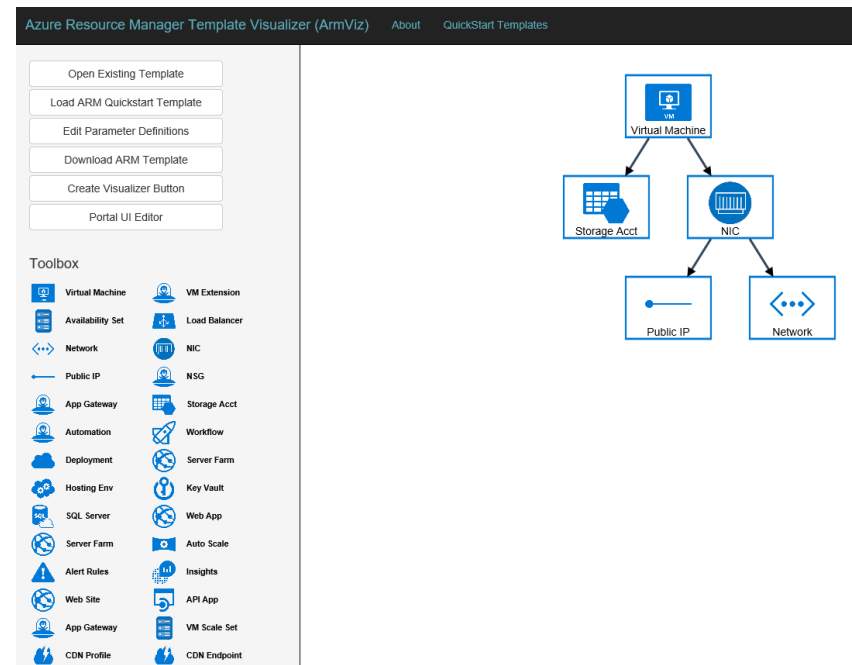
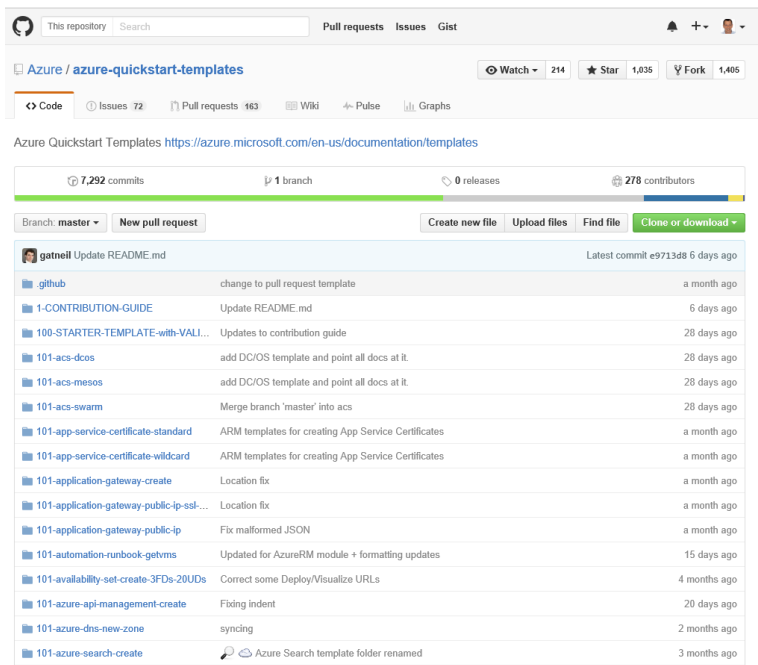


- Klassisch
  - ⇒ Ursprüngliches Modell
  - ⇒ Brauchen kein virtuelles Netzwerk
- Ressourcen-Manager
  - ⇒ Bereitstellung, Verwaltung und Überwachung als Gruppe
  - ⇒ Wiederholte Bereitstellung mit deklarativen Vorlagen
  - ⇒ Abhängigkeit zwischen Ressourcen
  - ⇒ Zugriffssteuerung (RBAC)
  - ⇒ Logische Organisation mit Hilfe von Tags
- Bereitstellungsmodelle nicht vollständig kompatibel!



# Demonstration: Resource Manager Vorlagen

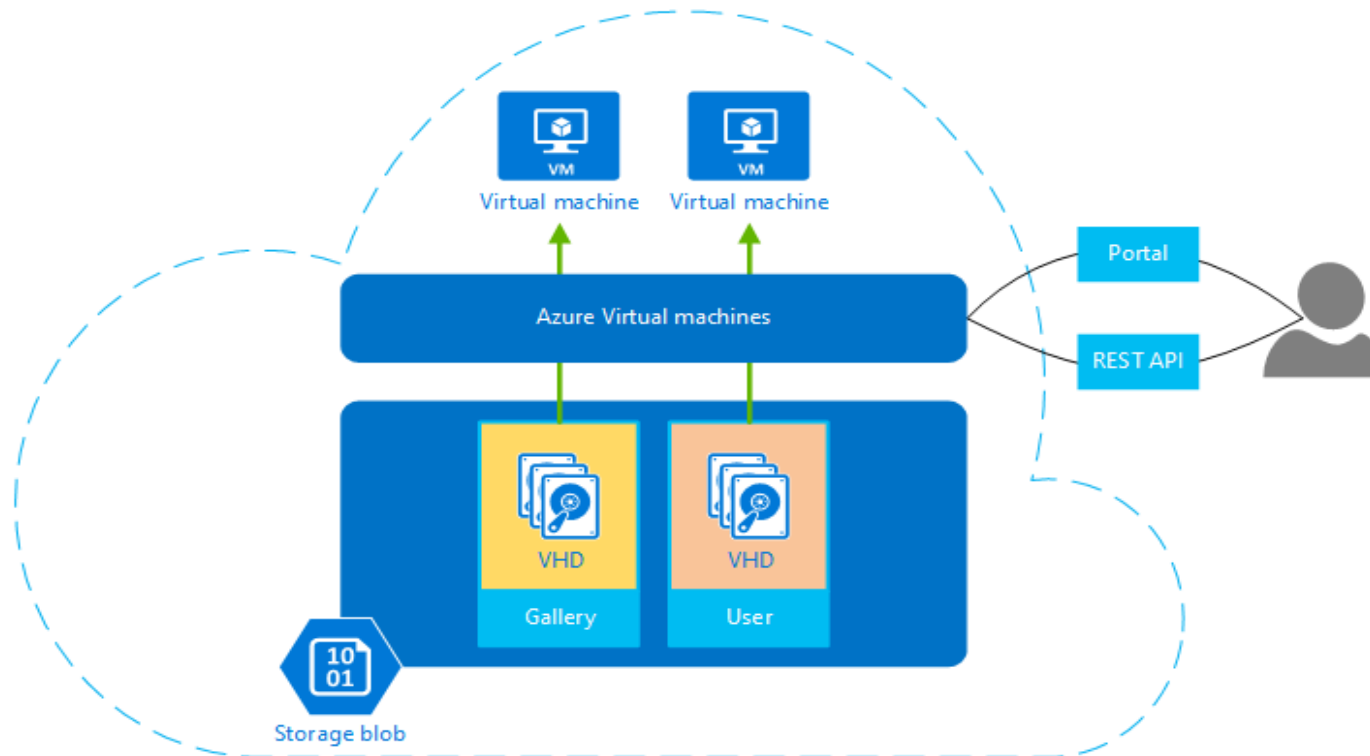
- <http://github.com/azure/azure-quickstart-templates>
- Visualisierung: <http://armviz.io>



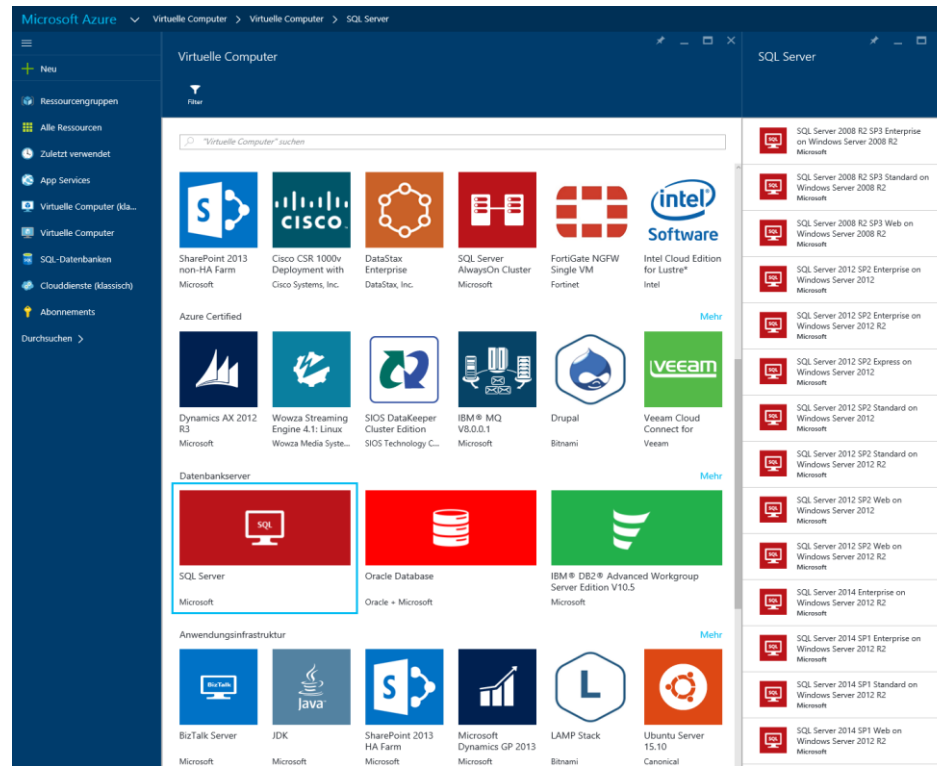
IaaS

# SQL SERVER IN VIRTUELLEN MASCHINEN

- Azure Virtuelle Maschinen bieten IaaS (Infrastructure as a Service)



- Eigene VM
  - ⇒ Lizenzmobilität unter Software Assurance bzw.
  - ⇒ Subscriber Access License (SAL) über SPLA
  
- Vordefinierte Vorlage mit bereitgestelltem SQL Server
  - ⇒ Lizenzierung inkludiert, jedoch ohne Windows Server Lizenz



# DEMO SQL TEMPLATES



# Checkliste für optimale Leistung 1

- Größe des virtuellen Computers
  - ⇒ DS3 oder höher für SQL Enterprise
  - ⇒ DS2 oder höher für SQL Standard und Web Edition
  
- Speicher
  - ⇒ Premium Storage Konto (SSD) – nur verwendbar mit DS, DSv2 oder GS Serie
  - ⇒ Speicherkonto und Virtueller Computer in gleicher Region
  - ⇒ Deaktivierung der Georeplikation auf Speicherkonto
  
- Datenträger
  - ⇒ Mindestens 2 Datenträger (Protokoll und Daten)
  - ⇒ Daten sollten nicht auf Betriebssystem Datenträger gespeichert werden
  - ⇒ Aktivierung vom Caching auf dem Protokolldatenträger
  - ⇒ Disk Striping für höheren E/A Durchsatz

- E/A

- ⇒ Datenbankseiten: Komprimierung aktivieren
- ⇒ Sofortige Dateiinitialisierung für Datendateien
- ⇒ Begrenzung / Deaktivierung der automatischen Vergrößerung für die Datenbank
- ⇒ Systemdatenbanken auf eigenen Datenträger verschieben

# SQL Server in virtuellen Maschinen

## Vorteile

- Vorhandene Anwendungen können schnell migriert werden
- Zugriff auf lokale Daten (über sicheren Tunnel – Hybrid Cloud)
- Unterstützung von Datenbankgrößen > 1 TB
- IT Umgebung mit vollen Administratorrechten
- Schnelle Entwicklungs- und Testszenarien ohne lokale SQL Server Hardware

### Gesamtkosten der Anwendung

minimierte Kosten für Softwareentwicklung/-änderungen  
+ Verwaltungskosten  
+ SQL Server- und Windows Server-Lizenzkosten  
+ Kosten für Azure Storage

# SQL Server in virtuellen Maschinen

## Nachteile

- IT Ressourcen für die Wartung des Betriebssystems
- Erhöhter administrativer Aufwand für Hochverfügbarkeits Szenarien
- Generell höherer administrativer Aufwand
- 99,95% SLA

PaaS

# AZURE SQL DATENBANKEN

- Unterschiedliche Versionen verfügbar (V11, V12)



Logischer V12-Server



Logischer Server einer früheren Version

- Upgrade empfehlenswert, da neue Funktionen nur bei V12 hinzugefügt werden

- PaaS Datenbank – auch DBaaS (Database as a Service)
- Optimiert für SaaS Anwendungsentwicklung
- Basic, Standard und Premium

|                         | Basic                                    | Standard                                    |     |       |       | Premium                                                           |       |       |        |        |
|-------------------------|------------------------------------------|---------------------------------------------|-----|-------|-------|-------------------------------------------------------------------|-------|-------|--------|--------|
|                         |                                          | S0                                          | S1  | S2    | S3    | P1                                                                | P2    | P4    | P6/P3  | P11    |
| Maximum DB Size         | 2 GB                                     | 250 GB                                      |     |       |       | 500 GB                                                            |       |       |        | 1 TB   |
| DTUs                    | 5                                        | 10                                          | 20  | 50    | 100   | 125                                                               | 250   | 500   | 1,000  | 1,750  |
| Point-in-Time Restore   | Any point last 7 days                    | Any point last 14 days                      |     |       |       | Any point last 35 days                                            |       |       |        |        |
| Disaster Recovery       | Geo-restore, restore to any Azure region | Standard Geo-Replication, offline secondary |     |       |       | Active Geo-Replication, up to 4 only (readable) secondary backups |       |       |        |        |
| Transactions per Second | 5.6                                      | 8.7                                         | 17  | 46    | 102   | 127                                                               | 263   | 494   | 1,108  | 1,761  |
| Max Concurrent Requests | 30                                       | 60                                          | 90  | 120   | 200   | 200                                                               | 400   | 800   | 1,600  | 2,400  |
| Max Concurrent Logins   | 30                                       | 60                                          | 90  | 120   | 200   | 200                                                               | 400   | 800   | 1,600  | 2,400  |
| Max Sessions            | 300                                      | 600                                         | 900 | 1,200 | 2,400 | 2,400                                                             | 4,800 | 9,600 | 19,200 | 32,000 |

- Die Datenbank-Transaktionseinheit (Database Transaction Unit, DTU) ist die Maßeinheit, die die relative Leistung von Datenbanken basierend auf einer echten Maßeinheit darstellt: der Datenbanktransaktion.

Wie viele Transaktionen pro Sekunde können unter maximaler Belastung vollständig abgeschlossen werden.

- Genaue Berechnung:  
<https://azure.microsoft.com/de-de/documentation/articles/sql-database-benchmark-overview/>



- Schnelle Anwendungsentwicklung / SaaS Anwendungen
- Integrierte hohe Verfügbarkeit und Notfallwiederherstellung
- Keine Verwaltung des zugrundeliegenden Betriebssystems und Konfigurationseinstellungen
- Horizontale Skalierung
- Einfache Migration

# Azure SQL Database V12

Höhere Anwendungskompatibilität mit SQL Server 2014 und 2016

25% mehr DTUs bei Premium Stufen ohne Zusatz Kosten

Volle Leistungsfähigkeit bedarf zusätzlicher Ports

Verbesserungen bei der Sicherheit

Bessere Recovery Point Objectives (RPO) und geschätzte Wiederherstellungszeiten (ERTs)

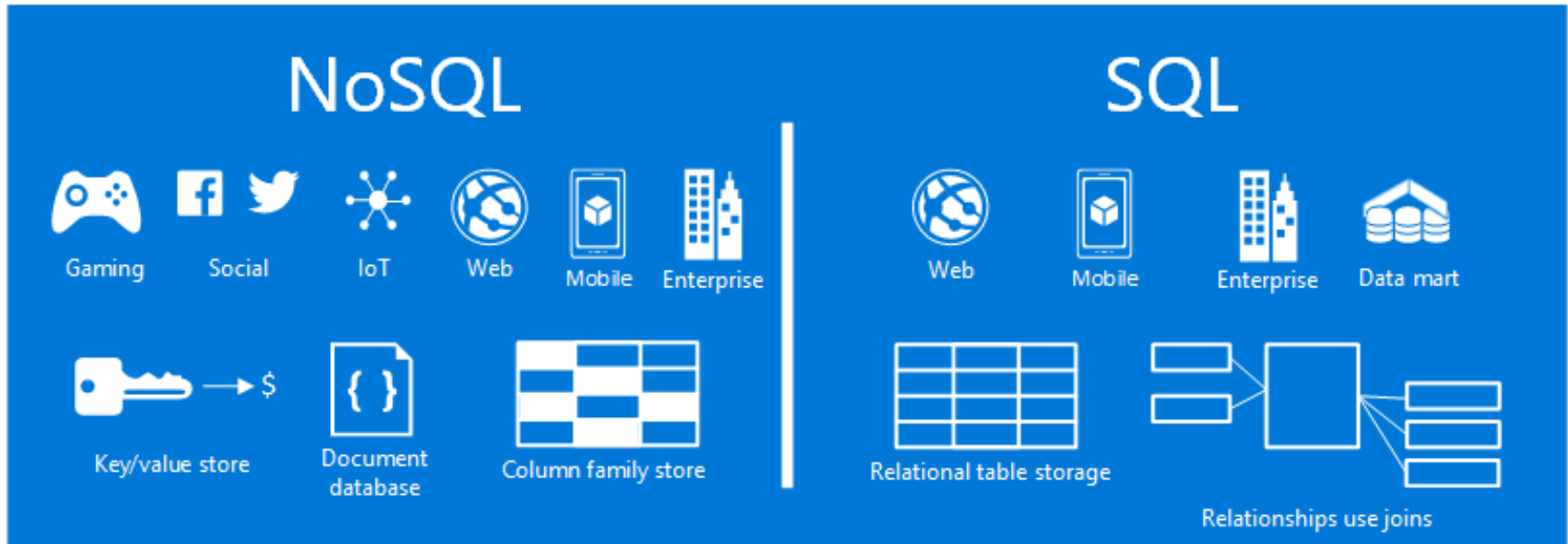
Voraussetzung für Elastic Database Pools

- Erleichtert die Erstellung, Wartung und Leistungsverwaltung vieler einzelner Datenbanken
- Optimal für SaaS Anwendungen
- Konfigurierbare Eigenschaften
  - ⇒ Dienstebene: Basic, Standard, Premium
  - ⇒ eDTU pro Pool
  - ⇒ Speicher pro Pool
  - ⇒ Minimale und maximale Anzahl von eDTUs pro Datenbank
- Elastische Aufträge, Transaktionen, Abfragen
- Vereinfachte Berechnung der Ausgaben

PaaS

# NOSQL, DOCUMENTDB

# NoSQL vs. SQL



# NoSQL und SQL im Vergleich

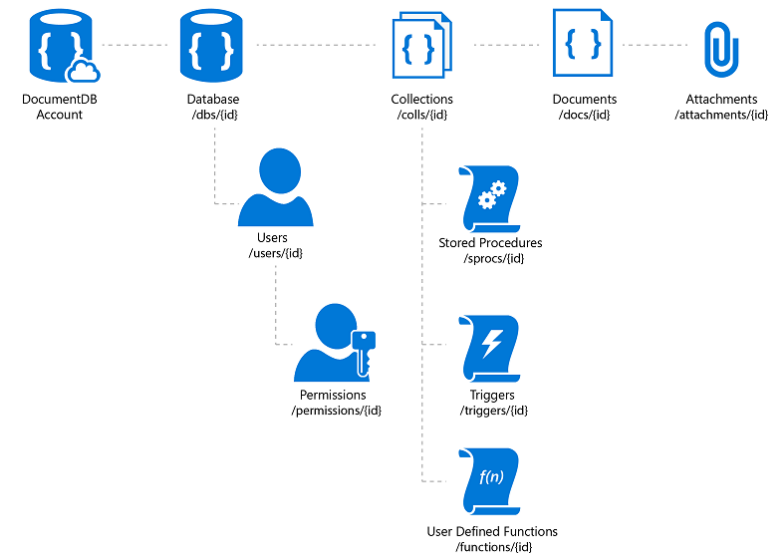
|        | NoSQL                                                                                 | SQL                                                                                     |
|--------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| Modell | Nicht relational                                                                      | Relational                                                                              |
|        | Speichert Daten in JSON Dokumenten, Key/Value Paaren, Wide Column Stores oder Graphen | Speichert Daten in Tabellen                                                             |
| Daten  | Flexibel – Nicht alle Datensätze speichern dieselben Eigenschaften                    | Jeder Datensatz hat diesselben Eigenschaften                                            |
|        | Neue Eigenschaften können jederzeit hinzugefügt werden                                | Neue Eigenschaften erfordern Schema Änderungen und Aktualisierung der vorhandenen Daten |
|        | Beziehungen werden häufig denormalisiert                                              | Beziehungen werden über Joins aufgelöst                                                 |
|        | Gut für Semi-Strukturierte Daten                                                      | Gut für Strukturierte Daten                                                             |

# NoSQL und SQL im Vergleich

|               | NoSQL                                                | SQL                                         |
|---------------|------------------------------------------------------|---------------------------------------------|
| Schema        | Dynamisch                                            | Strikt                                      |
|               | Höhere Agilität und bessere iterative Entwicklung    | Schema und Applikation müssen konform gehen |
| Transaktionen | ACID Transaktionen von der gewählten Lösung abhängig | ACID Transaktionen werden unterstützt       |
| Konsistenz    | Konsistenz ist von der gewählten Lösung abhängig     | Hohe Konsistenz gewährleistet               |
| Skalierung    | Skaliert gut horizontal                              | Skaliert gut vertikal                       |

# DocumentDB

- Flexibel skalierbarer Durchsatz und Speicher
- Ad-Hoc Abfragen mit vertrauter SQL Syntax
- Javascript Ausführung innerhalb der Datenbank
- Optimierbare Konsistenzebenen
- Vollständig verwaltet
- Offen konzipiert





# NoSQL Angebote in Azure

## Azure DocumentDB

- JSON Datenbank

## Azure Table Storage

- Schemaloser Key / Value Store mit OData Unterstützung
- kostengünstig

## Azure HBase als Teil von HDInsight

- Key / Value Store
- Sensordaten
- Echtzeitabfragen
- HBase als Plattform

## Azure Redis Cache

- Speicherung von Zeichenfolgen, Hashes, Listen, Sätzen mit Transaktionen und hierarchischer Replikation

# Zusammenfassung

## Was und das wofür?

### SQL IaaS

- Voller Admin
- Management selbst
- Skalierung komplex
- Legacy Apps
- Hybrid Szenarien
- Volle MS SQL Funktionalität (DB+AS+RS)
- Kosten: \$\$\$

### SQL PaaS

- Management durch MS
- Nur DB Engine
- Geeignet für SaaS Anwendungen
- Skalierung einfach
- Kosten: \$\$

### NoSQL

- Nicht Relational sondern dokumentorientiert
- Key/Value Store
- Flexiblere Anwendungsentwicklung
- Kosten: \$

# AUSBLICK

# Die Zukunft bringt

Data Lake  
Speicher

SQL Server  
Stretch  
Datenbank

SQL Data  
Warehouse

## SQL Data Warehouse VORSCHAU

Ein elastisches Data Warehouse-as-a-Service-Angebot mit Funktionalität auf Unternehmensniveau.

- ✓ Skalierung im Petabytebereich dank MPP (Massively Parallel Processing)
- ✓ Unabhängige Skalierung von Compute- und Speicherressourcen innerhalb von Sekunden
- ✓ Transact-SQL-Abfragen über relationale und nicht relationale Daten hinweg
- ✓ Vollständige SQL Server-Funktionalität auf Unternehmensniveau
- ✓ Nahtlose Integration mit Power BI, Machine Learning, HDInsight und Data Factory

# Q&A

