



# SAVE MONEY WITH AZURE SQL SERVERLESS

Haiko Hertes  
sqlDays 2020





- Seit 2019 bei SoftwareONE / COMPAREX
- Erst als Solutions Architect Future Datacenter im Solution Sales Team
- Jetzt als Principal Consultant & Architect im Azure Consulting Team
- Vorher IT-Leiter im Mittelstand
- Microsoft MVP und Speaker in diversen Communities



[www.hertes.net](http://www.hertes.net)



[about.me/haiko.hertes](mailto:about.me/haiko.hertes)



[twitter.com/HHertes](https://twitter.com/HHertes)



Auch auf YouTube...

softwareONE

**HAIKO HERTES**  
CLOUD ARCHITECT / PRINCIPAL CONSULTANT



# AGENDA

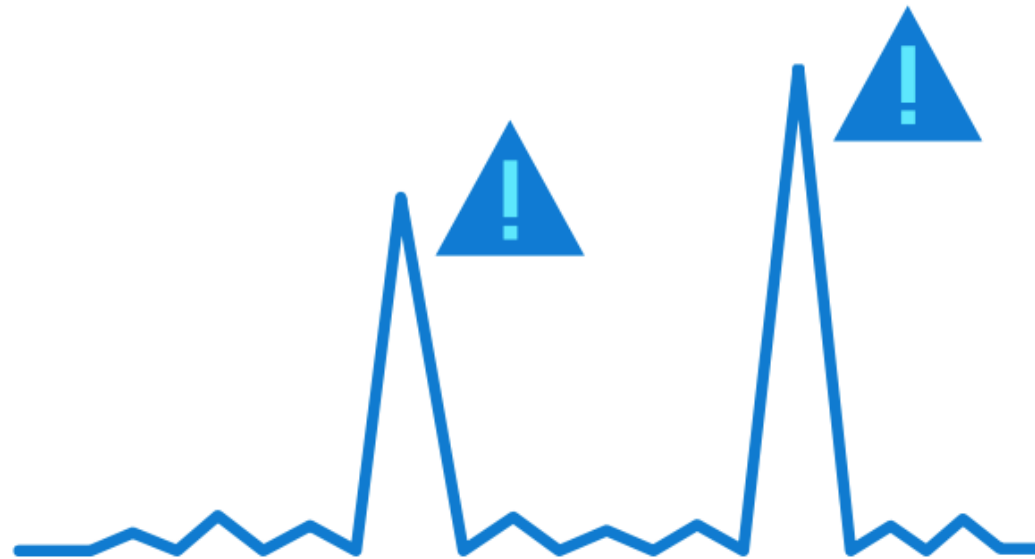
1. Das Problem
2. Die Lösung
3. Demo



# DAS PROBLEM

# DAS PROBLEM

- Die Auslastung einer (herkömmlichen) Azure SQL Datenbank sieht so aus:

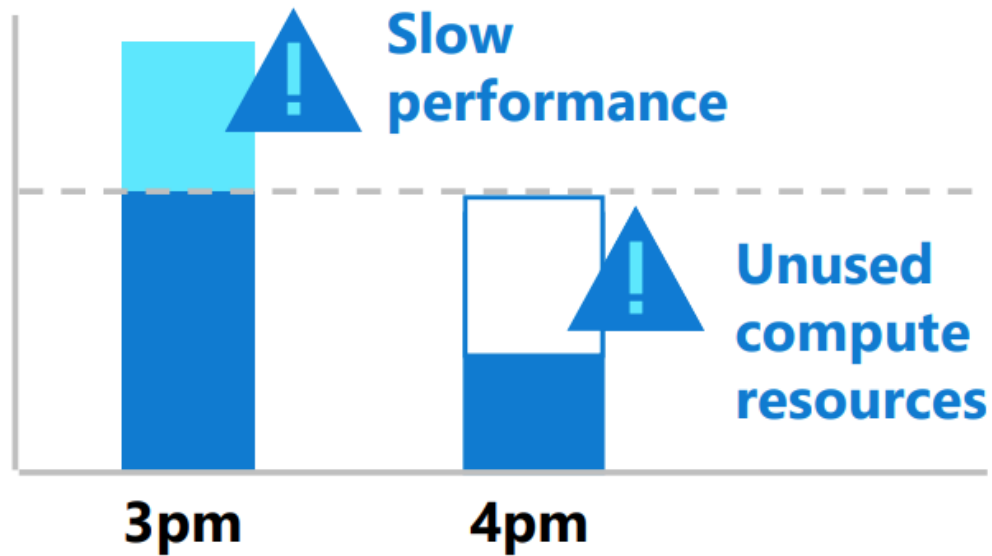


Schwankende, unvorhersehbare Auslastung

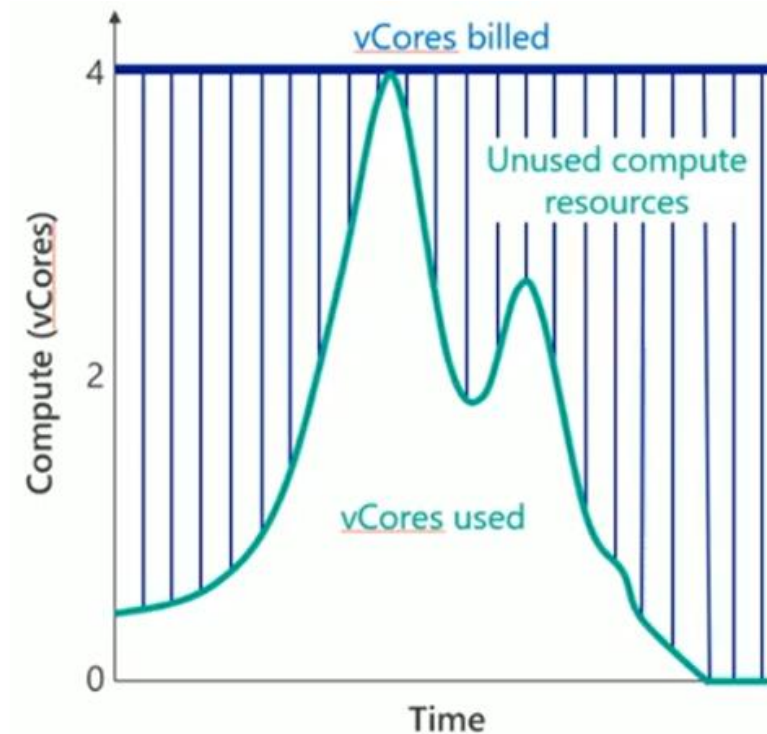
- Ebenso gibt es Zeiten, in denen die DB gar keine Last hat

# DAS PROBLEM

- Für Azure SQL Datenbanken müssen DTUs oder vCores konfiguriert werden (konkrete Festlegung)



Wenn nicht für das absolute Maximum ausgelegt, wird es immer Zeiten mit "zu wenig Leistung" geben

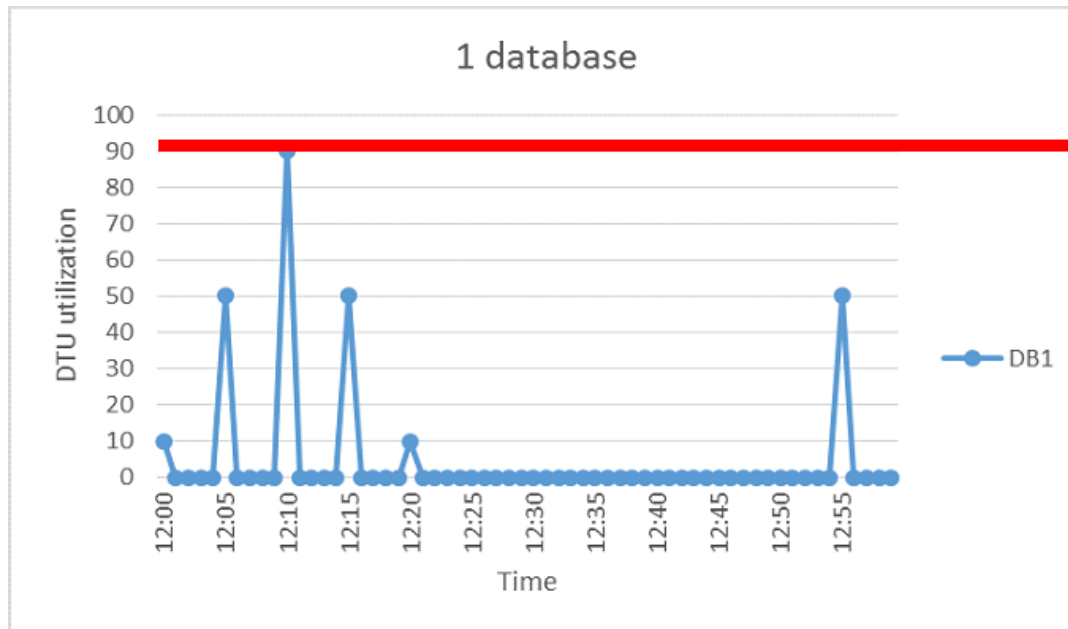


Gleichzeitig wird üblicherweise in einem großen Teil der Zeit die zur Verfügung stehende Leistung nicht zu 100% abgerufen

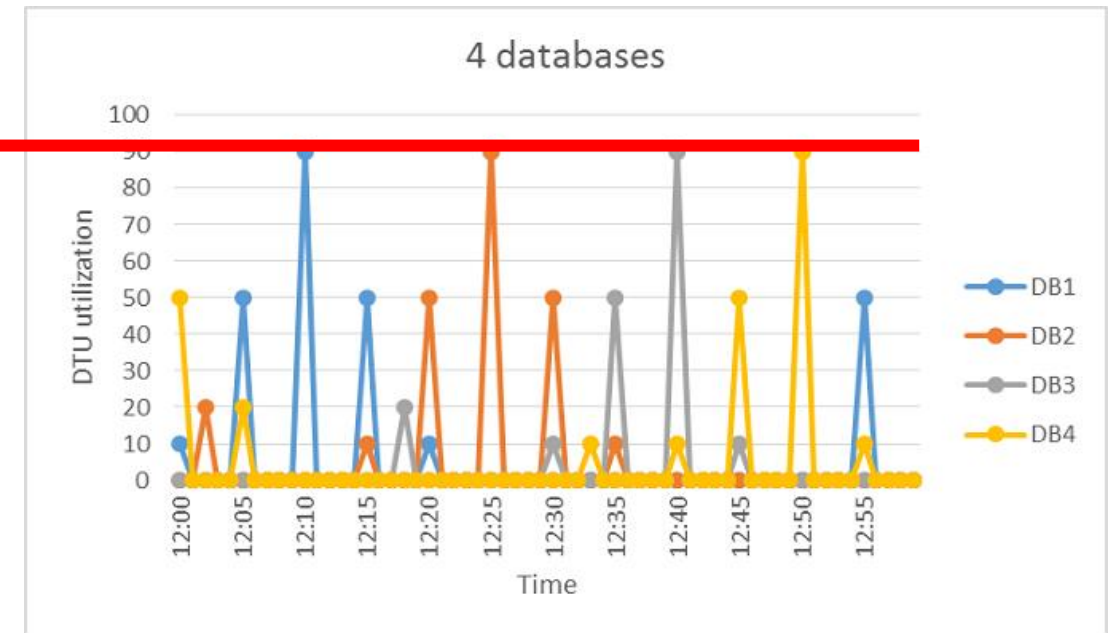


# DAS PROBLEM

- Warum nicht einfach Elastic Pool nutzen?



= S3 (100 DTUs) notwendig  
0,2016\$/hour  
für eine DB!



= 100 eDTUs ausreichend  
0,30\$/hour für  
4 Datenbanken

# DAS PROBLEM

- Mit Elastic Pools lassen sich schwankende Auslastungen „nivellieren“
- Dies funktioniert aber nur, wenn
  - Mehrere Datenbanken betrieben werden
  - Diese Datenbanken ihre hohen Leistungsanforderungen nicht gleichzeitig haben
- Allerdings ist es selbst hier eher unwahrscheinlich, dass die zur Verfügung stehende Leistung immer auch voll ausgenutzt wird
- Selbst wenn keine Leistung abgerufen wird, entstehen Kosten (PaaS Services lassen sich idR auch nicht kostenseitig stoppen)





# DIE LÖSUNG

# DIE LÖSUNG

- Azure SQL Serverless Tier
  - Skaliert automatisch in festgelegten, einstellbaren Grenzen (Min 0.5, Max 40 vCores)
  - Pausiert optional bei Nichtnutzung
  - Verrechnet nur nach tatsächlich benutzter Compute-Leistung (in vCore-Sekunden) (\*) und Storage

The screenshot displays the configuration options for the Azure SQL Serverless tier. It features two horizontal sliders for 'Max vCores' and 'Min vCores'. The 'Max vCores' slider is set to 4, with a range from 1 to 40. The 'Min vCores' slider is set to 0.5, with a range from 0.5 to 4. Below these sliders, the memory settings are shown as '2.1 GB MIN MEMORY' and '12 GB MAX MEMORY'. The 'Auto-pause delay' section includes a checkbox for 'Enable auto-pause' which is checked, and three dropdown menus for 'Days' (0), 'Hours' (1), and 'Minutes' (0).

Max vCores

1 2 4 6 8 10 12 14 16 18 20 24 32 40 4 vCores

Min vCores

0.5 0.75 1 1.25 1.5 1.75 2 2.25 2.5 3 4 0.5 vCores

2.1 GB MIN MEMORY 12 GB MAX MEMORY

Auto-pause delay

The database automatically pauses if it is inactive for the time period specified here, and automatically resumes when database activity recurs. Alternatively, auto-pausing can be disabled.

☒ Enable auto-pause

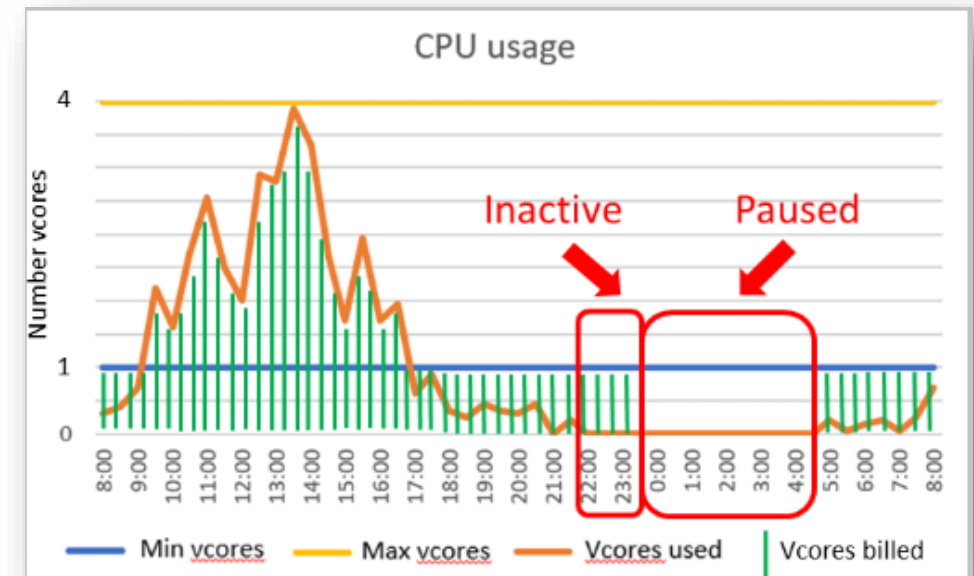
Days Hours Minutes

0 1 0

(\*) In inaktiver, nicht pausierter Zeit wird "Min vCores" berechnet

# DIE LÖSUNG

- Auto-Pause schaltet die Compute-Leistung nach einem eingestellten Zeitintervall ab
- Damit stoppt auch die In-Rechnung-Stellung
- Minimum: 1h Wartezeit nach letzter Inaktivität
- Ausführung wird mit nächstem Login oder andere Operation gegen die DB fortgesetzt



Resource group (change) : [RG-sqlDays](#)

Status : Paused

Location : West Europe

Subscription (change) : [Microsoft Azure Sponsorship](#)

Subscription ID :

Tags (change) : [Click here to add tags](#)

Server name : [\[redacted\]](#)

Connection strings : [Show database connection strings](#)

Pricing tier : [General Purpose: Serverless, Gen5, 4 vCores](#)

Auto-pause delay : 1 hour

Earliest restore point : 2020-10-19 11:39 UTC

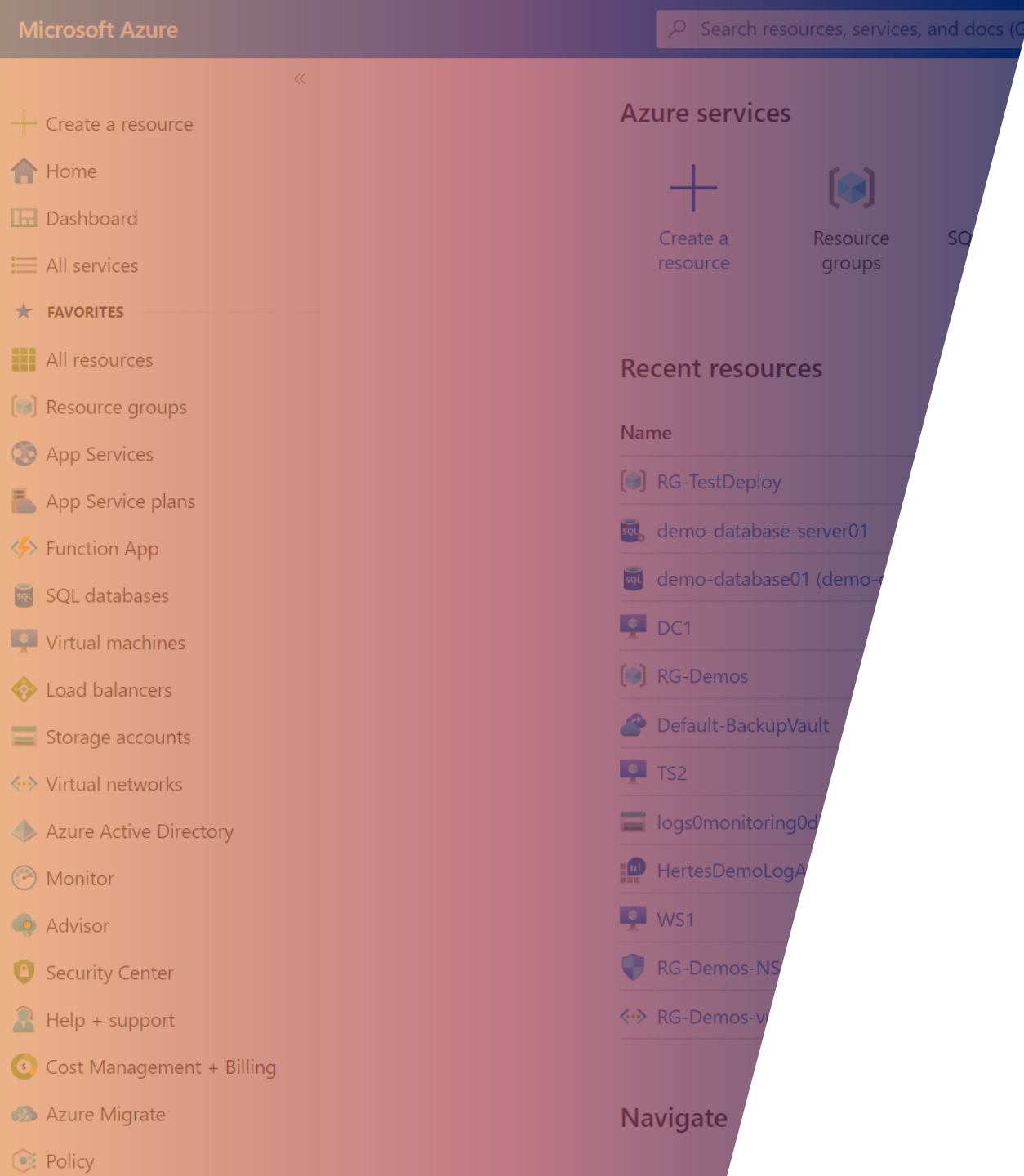


# UNTERSCHIED ZUM PROVISIONIERTEN AZURE SQL

- Klassischer Azure Server steht dauerhaft ohne Anlaufzeit zur Verfügung, SQL Serverless benötigt nach Auto-Pause eine Anlaufzeit (dann auch mit leerem Cache) von ca. 1 Minute
- Das hat zwangsweise zur Folge, dass die erste Abfrage an die Datenbank (die ja den Resume auslöst) zwangsweise mit Error 40613 fehlschlägt
- SQL Serverless Tier stellt „nur“ 3GB RAM pro Core bereit, Provisioned Tier stellt zwischen 5.5 und 7GB pro Core bereit

# ZU BEACHTEN

- Durch belegten Speicher lassen sich die Kosten nie ganz auf Null herunterfahren
- In der Zeit zwischen letztem DB-Zugriff und der automatischen Pausierung entstehen weiter die Compute-Kosten gemäß eingestelltem vCore-Minimum
- Das Serverless Compute Tier ist derzeit nur im „vCore“ Purchase Model und dem „General Purpose“ Service Tier verfügbar
- Es ist kein Hybrid Benefit anwendbar
- Wenn beim automatischen Hochskalieren nicht mehr ausreichend Cores vorhanden sind, kann es zu einem Loadbalancing kommen, bei dem kurz alle Verbindungen gedroppt werden
- Arbeitsspeicher wird häufiger / schneller wieder freigegeben
- Bei keiner oder sehr niedriger Auslastung der DB wird auch der Cache-Speicher freigegeben, dabei werden natürlich Cacheeinträge entfernt



# DEMO



# WOFÜR EIGNET ES SICH UND WOFÜR NICHT?

- Gut geeignet für
  - Einzelne Datenbanken
  - Unvorhersehbare, aber zeitlich nicht überwiegende Spitzenlasten
  - Unvorhersehbare Auslastung
  - Anwendungen, die einen kurzen Delay nach der Auto-Pausierung vertragen
  - Neue Datenbanken ohne Erfahrungen zur Auslastung, wenn Sizing schwierig ist
  - Bisher als Provisioned Tier betriebene DBs, die häufig neu skaliert werden mussten

# WOFÜR EIGNET ES SICH UND WOFÜR NICHT?

- Nicht so gut geeignet für
  - Anwendungen / DBs, die keinen Delay nach Auto-Pause vertragen
  - DBs, die mehr RAM benötigen
  - DBs mit konstant hoher Auslastung
- Szenarien, für die es bessere Lösungen geben könnte
  - Mehrere gleichzeitig betriebene DBs mit unterschiedlicher, schwankender, unvorhersehbarer Auslastung

# WIEVIEL GELD KANN MAN DENN DAMIT NUN SPAREN?

SQL Server Provisioned Tier  
General Purpose  
4 vCores

Ca. 671€ / Monat

SQL Server Serverless Tier  
General Purpose  
1 – 4 vCores

40% Pausiert  
40% 2 vCores  
20% 4 vCores

Ca. 576€ / Monat

**Ca. 15% Einsparung!**



# WARUM IST DAS TROTZDEM NOCH SO TEUER?

- „Serverless“ vCore-Sekunden sind heruntergebrochen genau doppelt so teuer wie „Provisioned“ vCore-Sekunden!

SQL Server Serverless Tier  
General Purpose  
1 – 4 vCores  
  
100% 4 vCores

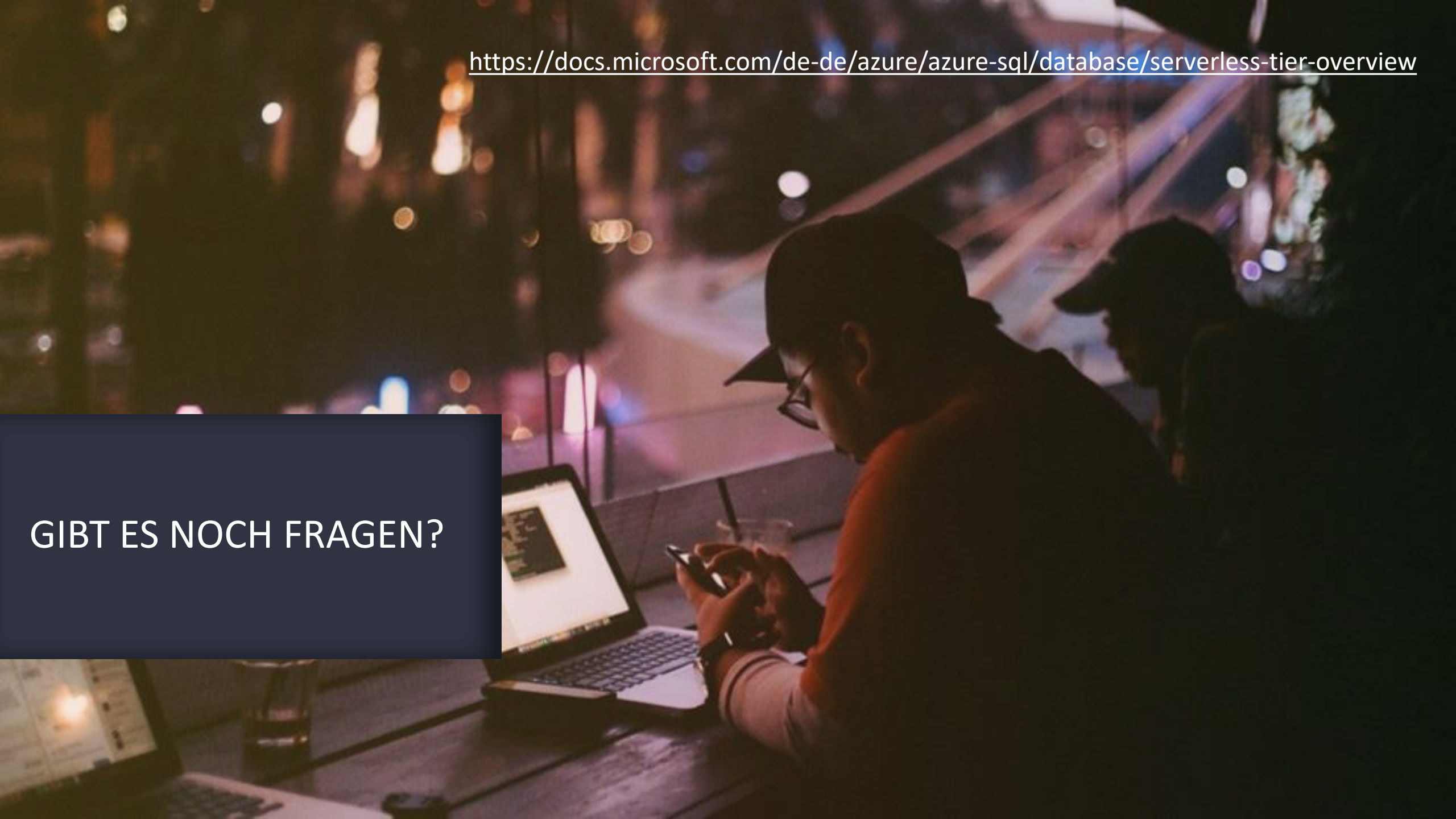
Ca. 1440€ / Monat

SQL Server Provisioned Tier  
General Purpose  
4 vCores

Ca. 576€ / Monat

<https://docs.microsoft.com/de-de/azure/azure-sql/database/serverless-tier-overview>

GIBT ES NOCH FRAGEN?





# VIELEN DANK

[www.hertes.net](http://www.hertes.net)



Haiko Hertes



[about.me/haiko.hertes](https://about.me/haiko.hertes)