



AZURE SMART NUTZEN

Kostenfallen vermeiden und Einsparpotenziale nutzen

Florian Bader

MIT WEM HABT IHR ES ZU TUN?



Florian Bader

Solution Architect | CTO

Florian.Bader@lunaris.digital

<https://lunaris.digital>

<https://github.com/florianbader>

KEY TAKE AWAYS



Wie können Kosten transparent gemacht werden?



Welche Sofortmaßnahmen bringen den größten Effekt?



Warum ist Kostenoptimierung mehr als Sparen?



Wie wird Kostenbewusstsein Teil des Entwicklungsprozesses?

WARUM REDEN WIR ÜBER KOSTEN?

Kosten als Architektur-Feature und Business-Faktor verstehen



Budget &
Verantwortung



Architektur
entscheidet



Kontinuierliche
Optimierung



Kultur &
Prozesse

DIE GRÖßTEN HEBEL FÜR SOFORTIGE EINSPARUNGEN

9 Maßnahmen, die ihr morgen umsetzen könnt



Ungenutzte
Ressourcen
identifizieren



Größe der
Ressourcen
anpassen



Logging-Kosten
optimieren



Reserved
Instances, Saving
Plans & Co.



Neue VM-
Generation
nutzen



Automatische
Skalierung
einsetzen



Kosteneffiziente
Speicheroption
wählen



Ressourcen
temporär
herunterfahren



Cluster über
Environments
nutzen

FINOPS: KULTUR STATT EINMALAKTION

Kostenoptimierung als kontinuierlicher Prozess

Was ist denn jetzt schon wieder FinOps?

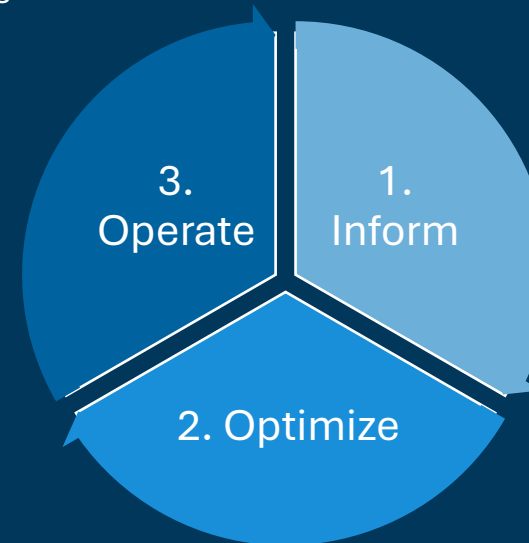
- Framework als Brücke zwischen Finance, IT und Business
- Ziel: Transparenz, Verantwortlichkeit und kontinuierliche Optimierung von Kosten und Business Value

FinOps, DevOps oder doch DevFinSecOps?

- FinOps ergänzt die bestehenden Prinzipien von DevOps oder SecOps um den finanziellen Aspekt

Wie fange ich damit am besten an?

- FinOps Assessment
 - Reifegrad prüfen: Transparenz, Prozesse, Tools
- Budgeting & Forecasting
 - Budgets pro Team/Projekt, Alerts bei Überschreitung
- Kostenschätzung im Entwicklungsprozess
 - Infracost in PRs, IaC-Kostenanalyse



Prinzipien

Zusammenarbeiten

Business, IT und Finance ziehen an einem Strang

Mehrwert entscheidet

Technologieeinsatz immer am Nutzen fürs Business messen

Verantwortung übernehmen

Jede Person achtet selbst auf Kosten und Nutzung

Daten nutzen

FinOps-Daten sind schnell, klar und für alle zugänglich

Zentrale Unterstützung

Ein Team stellt Standards und Tools bereit

Cloud-Vorteil nutzen

Kosten flexibel anpassen statt fix bezahlen



EFFIZIENZ KANN QUALITÄT STEIGERN

Kostenoptimierung ist kein Sparzwang

Freiheit ohne Leitplanken

Cloud kann flexibel skalieren. Heißt aber nicht, dass sie es auch muss.

- Brauche ich wirklich für jeden App Service einen eigenen Plan?

Hohe Betriebskosten ohne echten Mehrwert

- Brauche ich einen Dedicated SQL Server oder reicht mit das DTU-Modell?

Keine Abwägungen bei Detailentscheidung

- Muss es wirklich ein Info-Log sein oder ein Error-Log?

Kostenbewusstsein in der Architektur ist ein Muss

Ziel: Bewusste Entscheidungen, nicht extreme Optimierungen

Evolutionäre Architektur statt Big Design Up Front

- Was brauche ich jetzt, ohne mir den Weg in die Zukunft zu verbauen?

Forecasting von Kosten

- Wann muss ich skalieren? Mit wie vielen Kosten rechne ich wann?

Azure Well Architected Framework gibt Hilfe mit eigenem Cost Pillar

```
[NoChange] log-osp-dev-weu
  \--- Type: Microsoft.Operational
  \--- Location: westeurope
  \--- Total cost: 0,56 EUR
  \--- Delta: 0,00 EUR
```

Aggregated metrics:

```
→ Basic Logs Data Ingestion [0.560 EUR]
```

Used metrics:

```
→ Basic Logs | Azure Monitor | Basic Logs
```

```
[NoChange] db-osp-dev-weu-app
  \--- Type: Microsoft.Sql/servers
  \--- Location: westeurope
  \--- Total cost: 4,17 EUR
  \--- Delta: 0,00 EUR
```

Aggregated metrics:

```
→ B DTU [4.167 EUR]
```

Used metrics:

```
→ B | SQL Database Single Basic |
```

```
[Ignore] master
  \--- Type: Microsoft.Sql/servers
  \--- Location: westeurope
  \--- Total cost: 4,17 EUR
  \--- Delta: 0,00 EUR
```

KOSTENKONTROLLE BEGINNT IM CODE

Kosten sichtbar machen, bevor sie entstehen

Design & Entwicklung

- Kosten früh berücksichtigen (Pricing Calculator, Architecture Center)
- Kostenbewusstsein in der Definition of Done

Quality Gates & Reviews

- Kosten-Check über Infrastructure as Code als PR-Gate (Azure Cost Estimator)
- Tagging & Ressourcen-Ownership prüfen
- Prognose von Kosten für neue Features dokumentieren

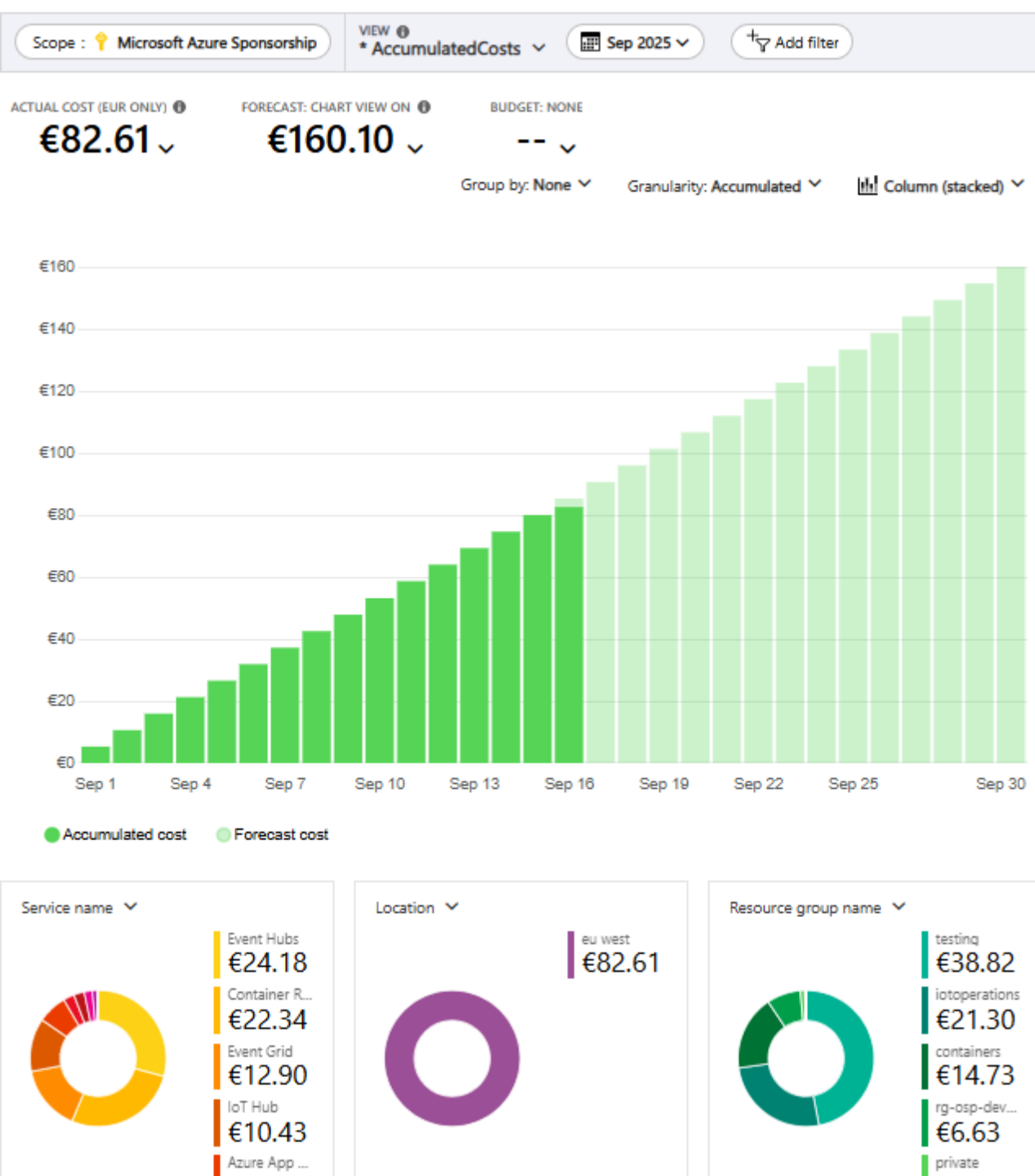
Operations & Optimierung

- Regelmäßige Kosten-Reviews (azure-cost-cli)
- Cost & Advisor Alerts (Advisor Workbook)
- Azure Policies definieren
 - Audit Warnings oder Deny
 - Policies as Code (Azure Enterprise Policy as Code)

DEMO

Kosten im Entwicklungsprozess





TRANSPARENZ IST DER ERSTE HEBEL

Ohne Sichtbarkeit keine Steuerung

Tools für Transparenz

- **Azure Cost Management**
 - Analyse von Kosten nach Service, Resource Group, Tag
 - Forecasts und Budgets setzen
- **Azure Advisor**
 - Empfehlungen für Right-Sizing und Reservierungen
 - Hinweise auf ungenutzte Ressourcen

Wichtige Metriken

- Cost per Service
- Cost per Request
- Cost per Environment (Prod vs. Dev/Test)

Tagging & Ownership

- Tags: Owner, Cost Center, Project
- Vererben von Tags über Resource Group und Subscription möglich
- Ohne Tags keine Zuordnung: Keine Verantwortlichkeit

DEMO

Azure Cost Management
& Azure Advisor



RECAP



Ohne Sichtbarkeit, keine
Steuerung:
Kosten früh analysieren



Mit kleinen Maßnahmen lässt
sich schon viel sparen



Kostenoptimierung ist ein
Prozess,
keine Einmal-Aktion



Kosten müssen im gesamten
Entwicklungsprozess
berücksichtigt werden

