

FRIENDS AND POSITIONS

<<API GROUP>>

Florian Tünte
Alexander Dall
Marek Brüning
Benedikt Schwering

AGENDA

1. Vorgehensweise
2. Technologien und Verwendung
 - i. Terraform
 - ii. Azure
 - a. CosmosDB
 - b. Functions
 - iii. Python
3. Herausforderungen
4. Blick in den Code und Demo

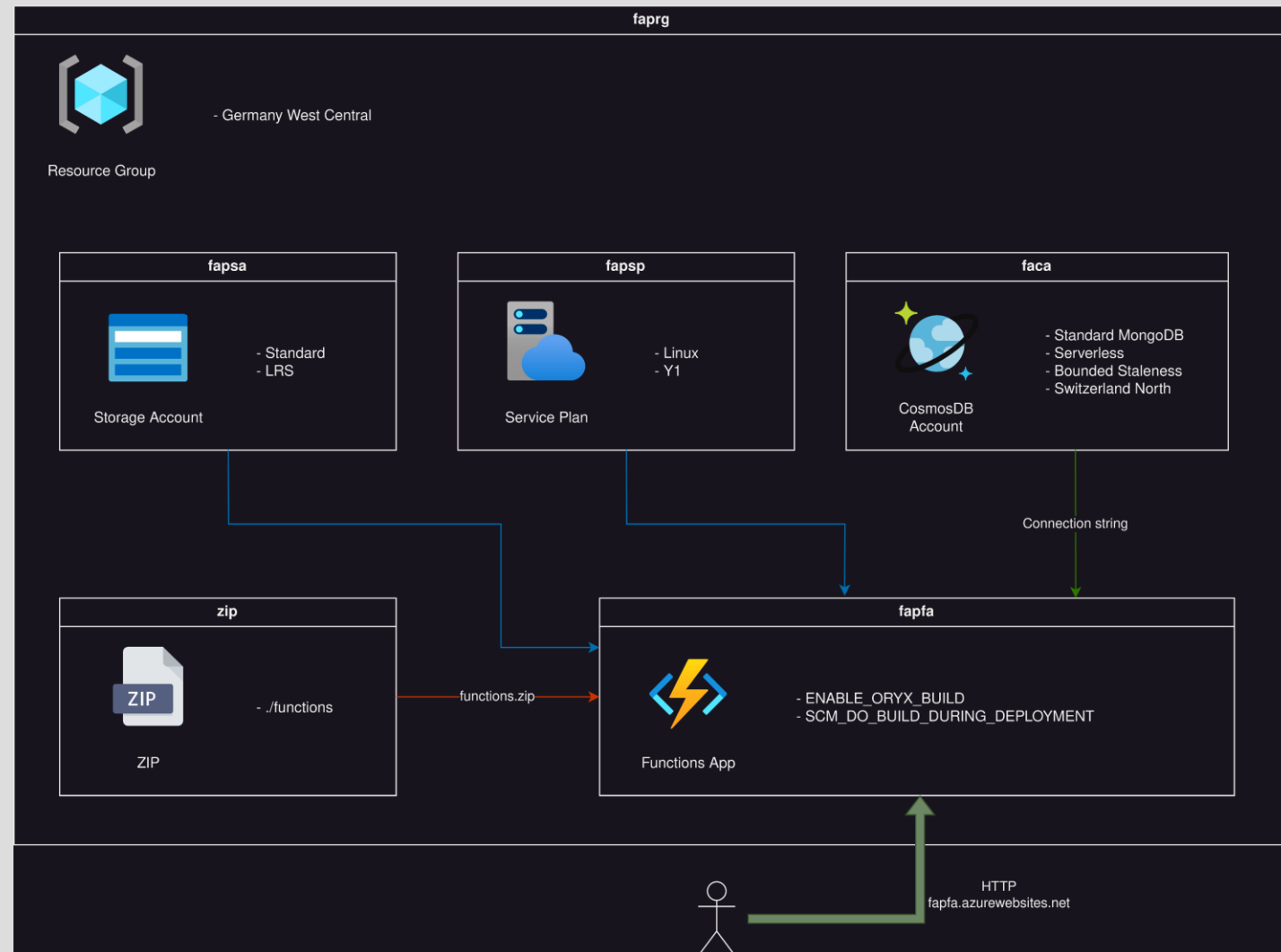
VORGEHENSWEISE





TERRAFORM

- Infrastructure as Code (IaC)
- Definition der Infrastruktur in Markup Language
- Provider für alle Cloud Hyperscaler
- Planung, Bereitstellung und Entfernen der Cloud Ressourcen per CLI
- Übersichtlich, strukturiert und reproduzierbar → Kein „zusammenklicken“ der einzelnen Komponenten





AZURE

- Cloud Computing Plattform von Microsoft
- Cloud Services
 - Komplette Anwendung kann über Azure Dienste abgebildet werden
 - Flexibel und Skalierbar
 - Weltweit
- Terraform Provider azurearm

Übrigens: jedes Jahr 100 USD
Guthaben für Studenten.
Kostenlos und steuerfrei!





AZURE COSMOSDB

- Im MongoDB Modus
 - Schemalos
- Optional Serverless
- Selbst gewählte Konsistenz
- Hohe Verfügbarkeit
- Skalierbarkeit ohne Aufwand
- Persistenz der User, Standorte sowie Sessions



AZURE FUNCTIONS

- Serverless Computing
 - Skalierbarkeit ohne Aufwand
 - Ereignis basierte Ausführung
 - Pay-as-you-go
- Azure Gegenstück zu AWS Lambda
- Eingebunden per HTTP-Trigger um die Services nach außen bereitzustellen



PYTHON

- Simple Sprache um die einzelnen Services abzubilden
- SDK für Azure Functions
- Einzelne Functions
 - Mit mongodb verbinden
 - Aktion ausführen
 - Ergebnis zurückgeben
- Wenig Overhead

HERAUSFORDERUNGEN

- Ausführbarkeit der Azure Function
 - Abhängigkeiten wurden nicht geladen
 - Terraform nicht der "Azure Weg"
 - Dokumentation nicht offensichtlich
- „Was brauchen wir“ zu „Wie geht das in Azure“

BLICK IN DEN CODE UND DEMO



https://github.com/DlieBG/friends_and_positions

<https://fapfa.azurewebsites.net/FAPServer/service/fapservice>