

Microservices Architecture

Doelgroep Cursus Microservices Architecture

De cursus Microservices Architecture is bedoeld voor developers en architecten en anderen die inzicht willen krijgen het waarom en hoe van een Microservices Architecture.

Voorkennis training Microservices Architecture

Om aan de cursus Microservices Architecture deel te nemen is algemene kennis van software ontwikkeling en software design wenselijk. Voorafgaande kennis van moderne programmeertalen is bevorderlijk voor de begripsvorming.

Uitvoering cursus Microservices Architecture

De theorie wordt behandeld aan de hand van presentaties. Illustratieve demo's worden gebruikt om de behandelde concepten te verduidelijken. Er is voldoende gelegenheid om te oefenen en afwisseling van theorie en praktijk. De cursustijden zijn van 9.30 tot 16.30.

Certificering cursus Microservices Architecture

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat Microservices Architecture.

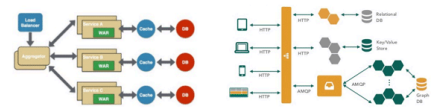
Duur: 3 dagen

Prijs: € 1999

Open Rooster



Micro Service Architecture



Inhoud Cursus Microservices Architecture

In de cursus Microservices Architecture leren de deelnemers de voordelen van de ontwikkeling van applicaties in een Microservices Architectuur ten opzichte van een Monolitische Architectuur. In de Microservices Architecture zijn kleine teams verantwoordelijk voor ontwikkeling en deployment van de microservices die elk een eigen database en user interface kunnen hebben.

Intro Microservices

De cursus Microservices Architecture gaat van start een bespreking van de problemen van monolitische applicaties. Daarna wordt ingegaan op de kenmerken van een Microservices Architecture zoals independent deployability, independent scaling en gescheiden databases.

Design Considerations

Vervolgens wordt uitgelegd hoe volgens de wet van Conway de architectuur van applicaties gereflecteerd wordt in de organisatie structuur. De overstap op een Microservices Architectuur vereist dus aanpassingen in de organisatie. Ook wordt dan ingegaan op de principes van Domain Driven Design.

Interprocess Communication

Microservices zijn aparte processen en communiceren via interprocess communicatie. Aandacht wordt besteed aan synchrone communicatie via REST, asynchrone communicatie via messaging en communicatie via een binair protocol als Thrift.

Micro Frontends

Dan is het tijd voor een bespreking van de user interfaces van Microservices. Diverse benaderingen voor de integratie een user interface worden besproken zoals custom elements, server side templates en built time integratie van JavaScript libraries.

Data Management

Eveneens komt data management in een Microservices Architectuur aan bod. De verschillende patterns voor data opslag zoals database per service, shared database, het saga pattern en event sourcing worden dan besproken.

Discovery en Deployment

Tenslotte komen de opties voor service discovery aan de orde zoals client en server side discovery en het gebruik van service registrars. Ook passeren dan deployment mogelijkheden van microservices zoals in virtual machines en in containers de revue.

Modules Cursus Microservices Architecture

Module 1 : Intro Microservices	Module 2 : Design Considerations	Module 3 : Interprocess Communication
Microservices Architecture Monolithic Applications Software Monolith Problems of Monoliths Layered Architecture Growing Beyond Limits Microservices Characteristics Underlying Principles Independent Deployability Independent Scaling Separated Databases Size of Microservices Frontend Monolith Micro Frontend Architecture Blurry Service Boundaries	Conways's Law Law as Enabler Domain Driven Design Building Blocks Bounded Context Reactive Manifesto Reactive Microservices Microservices with UI Microservices Benefits Effective Modularization Replaceability Microservices Continuous Delivery Pipeline Free Technology Choice Team Independence Microservices and Languages	Communicating Processes Interaction Styles Service Communications Styles Defining API's Netflix Hystrix Asynchronous Messaging Messaging Models Publish and Subscribe Point to Point Advantages of Messaging Synchronous IPC REST Services Resource URI Access Thrift Content Negotiation
Module 4 : Micro Frontends	Module 5 : Reactive Microservices	Module 6 : Data Management
What are Micro Frontends? Monolith versus Micro Frontends Benefits Micro Frontends Delivery Pipeline per Service Avoid Shared Artifacts FrontEnd Integration ESI Composition Links and JavaScript Custom Elements Integration Approaches Server Side Templates Built Time Integration Shared Component Libraries Cross Application Communication Backend Communication	What is Reactive? Reactive Programming Reactive Extensions Observables Reactive Systems Elasticity and Resilience Reactive Microservices Asynchronous Development Event Loop Reactor Pattern Multireactor Pattern Verticles Callbacks vs Observables RxJava API Monitoring	Distributed Data Complex Data Access Polyglot Architecture Private Access Design Patterns Database per Service Shared Database Pattern Saga Pattern Event Publishing Consuming Events Responding to Events Base Model Transactions Local Transactions Database Transaction Log Event Sourcing
Module 7 : Service Discovery	Module 8 : Deployment Strategies	Module 9 : Security
Why Service Discovery Finding Services Client Side Discovery Service Registry Server Side Discovery Load Balancers Service Registries Self Registration Pattern Third Party Registration Pattern Service Registrars Netflix Eureka HashiCorp Consul	Deployment Patterns Virtual Machines Creating Virtual Machines Drawbacks of VM's Containers Containers versus VM's Container Orchestration Kubernetes Multiple Instances Per Host Service Instance per Host Service Instance per Container Serverless Deployment	Microservices Security Security Challenges Key Security Fundamentals Confidentiality Edge Security Securing with OAuth2 Authorization Server Interactions Actors OAuth2.0 Flow OAuth Roles Application Registration Securing with API Gateway Zuul Proxy and OAuth2