

# Hakim Jonas Ghoula

*Data Engineer with a focus on functional programming and scalable solutions*

🏠 Hothers Plads 11, 3, 2200 Cph. N

☎ +45 26335577

✉ hakim@walkthisway.dk



## 📄 Summary

Data Engineer with 3+ years of experience designing, optimizing, and managing large-scale data pipelines using Scala, Spark, and Python. Skilled in building robust ETL processes, implementing test-driven development, and leveraging AWS cloud services. Key projects at LiveIntent (acquired by Zeta Global) include Dormant Email Reactivation and PersonID/offline graph pipelines. Currently exploring compiler and language design through a personal functional programming language project in Rust, focusing on type systems and parsing techniques.

## 🧰 Experience

### Professional Experience

#### Data Engineer

**LiveIntent (acquired by Zeta Global in November 2024) | 2022 – Present**

Designed and maintained large-scale ETL pipelines using Scala and Spark, improving data processing efficiency.

- ▶ Collaborated on data ingestion and delivery processes for external partners
- ▶ Contributed to Dormant Email Reactivation and PersonID/offline graph projects, supporting business objectives
- ▶ Emphasized functional programming and TDD to create maintainable and reliable code

**Technologies:** Scala, Spark, AWS, Functional Programming, TDD

#### Associate Data Scientist

**LiveIntent (acquired by Zeta Global in November 2024) | 2022 – 2022**

Supported the development of analytical pipelines using Scala-based systems.

**Technologies:** Scala, Data Analysis

#### Python Back End Developer

**Ocean.io | 2018 – 2019**

Developed email prediction and language detection modules for the Growth Intelligence platform.

- ▶ Maintained and optimized internal APIs to ensure seamless data integration

**Technologies:** Python, APIs, NLP

#### Front End Developer / Data Engineer

**Købstædernes Forsikring | 2017 – 2018**

Contributed to the development of a new data model for the insurance customer database.

- ▶ Implemented white-label solutions and managed web projects for external partners

**Technologies:** Front End, Data Modeling

#### Research Assistant

**LETSMT! / University of Copenhagen | 2012 – 2012**

Conducted data acquisition and analysis for an EU-funded machine translation project.

**Technologies:** Data Analysis, Machine Translation

#### Independent Web Developer

**Self-employed | 2004 – 2010**

Designed and built websites and portals for a range of clients.

## ⚙️ Skills

### Key Skills

#### Languages

- ▶ Scala: Functional & OOP, strongly typed systems, TDD
- ▶ Python: Airflow (DAGs), data processing (Pandas, NumPy), machine learning, NLP
- ▶ SQL: AWS Athena, performance tuning, complex queries
- ▶ Rust: Compiler/language design project & more

#### Data Engineering & Big Data

- ▶ Apache Spark: ETL, transformations, performance optimizations
- ▶ Apache Airflow: DAG creation, scheduling, orchestration
- ▶ AWS: S3, Athena, EC2, data warehousing, analytics
- ▶ Docker & Kubernetes: Containerization, orchestration, deployment

## 📖 Languages

**English:** Fluent

**Danish:** Native

## 🎓 Education

### Education

#### MSc in IT and Cognition

University of Copenhagen  
2014-2016

#### Bachelor in Philosophy

University of Copenhagen  
2008-2011

#### Diploma in Music Management

The Royal Danish Conservatory  
2009-2011

### Certificates

- ▶ Python Certified Developer Professional – Superusers (2020)

**Technologies:** Web Development, HTML/CSS

## Projects

### **valar - Scala Repository**

Valar: Type-safe validation for Scala 3 with boilerplate-free derivation via macros. Features error accumulation and fail-fast modes.

**Technologies:** Scala, data-validation, data-validation-, error-handling, maven-central, scala, scala3, validation

### **fin - Rust Repository**

Finë is a simple, GTK4-based session transition tool for Linux desktops—built in Rust with a functional, modular design.

**Technologies:** Rust

### **spore-print (Fungal) - Rust Repository**

A Rust library providing a SporePrint trait and derive macro for immutable string representations.

**Technologies:** Rust