



# Matt Rust

## Backend Software Engineer

### About Me

- Backend-focused software developer with hands-on experience building and maintaining scalable, serverless development and production systems with AWS
- Skilled in Python (7+ years), SQL and cloud architecture, with a strong foundation in CI/CD, TDD and security best practices
- Cloud certifications: **AWS Certified Developer—Associate** and **AWS Certified AI Practitioner**
- Taught Python for several years as a secondary school teacher (11-18)
- Experienced in communication with colleagues and other stakeholders at all levels, problem-solving and effective collaboration
- Currently learning **Next.js** and exploring **AI** tools and services

### Contact Information

**Location:** Remote, UK

**Email:** mmrust515@gmail.com

### Skills & Expertise

#### Languages

- Python
- Shell Script
- SQL

Frameworks

### Professional Experience

#### Genomics England

*Python Engineer January 2026—Current*

#### Responsibilities

- Delivering business requirements primarily using Python and Terraform
- Working collaboratively as part of the team to solve petabyte-scale problems
- Taking a test-first approach to delivery from acceptance tests all the way down to unit tests
- Monitoring, troubleshooting and supporting services running at scale in production
- Contributing to code reviews and architecture discussions
- Using Git, Confluence and JIRA daily

#### Skills and experience

- Python programming, delivering production-ready code
- Working on distributed systems using managed cloud services (AWS)
- Provisioning modern cloud infrastructure with Terraform
- CI/CD pipelines (GitLab CI)
- Evaluation of the pros and cons of different technologies (e.g. SQL vs NoSQL databases)
- Test automation and a quality-first engineering mindset
- Working in agile squads with a DevOps culture
- Familiarity with how Non-Functional Requirements impact the design of a system
- Excellent communication skills, able to work across cross-functional teams

#### Grid Smarter Cities

*Backend Engineer January 2023—January 2026*

#### Backend Development and Automation

- Responsible for maintenance and development of the business' products using Python and SQL for backend APIs that integrate with Front End and Mobile applications
- Planning and creating new features that integrate with existing services, with a focus on future scalability and long-term performance stability
- Designing, building and maintaining REST APIs and backend logic using Python and SQL
- Planning, creation and application of database migrations
- Migrated production database from MySQL to PostgreSQL and configured associated infrastructure (e.g. custom VPC, RDS Proxy, EC2 autoscaling)
- Developed and deployed custom ORM layer with SQLAlchemy to interface with MySQL and PostgreSQL
- Writing and scheduling shell scripts to automate backup, migration and cleanup tasks

#### Cloud and Serverless Infrastructure

- Designing and maintaining event-driven microservices using AWS Lambda, SQS, EventBridge and DynamoDB
- Configuring and maintaining EC2 autoscaling groups of Bastion instances for secure database access
- Actively participating in AWS Well-Architected Reviews to align legacy infrastructure with best practices and implement follow-up remediation
- Configuring AWS GuardDuty, Security Hub and Backup for enhanced security and compliance
-

- Django
- REST APIs

## Learning

- TypeScript
- Next.js

## AWS

- EC2
- S3
- IAM
- CloudFormation
- Lambda
- API Gateway
- ECR
- SAM
- EventBridge
- SSM Session Manager
- SSM Parameter Store
- SNS
- SES
- CodeBuild
- CloudWatch
- VPC
- KMS
- Secrets Manager
- Route 53
- CloudFront

## Database

- DynamoDB
- RDS (MySQL, PostgreSQL)
- DMS

## Security, Observability and Ops

- CloudTrail
- AWS X-Ray
- GuardDuty
- Config
- Security Hub

## Learning

- SageMaker
- Bedrock
- Amazon Q/Q Business

## Methods

- Agile Development
- Test-Driven Development (TDD)
- DRY, KISS and SOLID principles
- Well-documented code

## Tools

Creating CloudFormation templates to provision cloud infrastructure (IaC) and automate deployments

## Testing and Quality Assurance

- Practice test-driven development (TDD) to maintain robust, production-grade systems
- Maintain and create unit, integration and contract tests using Pytest, Unittest and Schemathesis

## Monitoring and Observability

- Configured and maintained SonarCloud integration for static security analysis as part of the CI/CD process
- Integrated services with Datadog for real-time metrics, logs and dashboards as part of Security Information and Event Management (SIEM) improvements
- Set up alerting for anomalies and service health to ensure rapid incident response

## Career Transition May 2022–December 2022

- After several enjoyable years teaching the theory of code and Python to 11–18 year-olds, including students taking national qualifications, I was inspired to pursue a new challenge in software engineering. I like guiding learners through problem solving with code — encouraging structured thinking, debugging collaboratively and asking the right questions rather than simply giving answers. These experiences mirror many core practices in development, like pair programming and code review. Eager to deepen my skills, I successfully completed courses with HarvardX and Udemy.

## Heathside School

### Vice Principal & Teacher of Computer Science September 2021–April 2022

- Taught 11–18 year-old students Computer Science, including teaching Python to students studying for national GCSE and A Level qualifications.
- Senior school leadership role with transferable leadership, communication and problem-solving skills.
- Reason for leaving: Career change to become a Software Engineer.

## Howard of Effingham School

### THPT Standards & Performance Lead September 2019–August 2021

### Deputy Headteacher September 2016–August 2021

### Assistant Headteacher June 2014–August 2016

- Senior Trust Leader and Teacher of Computer Science (including Python), ages 11–18.

## Fullbrook School

### Assistant Headteacher September 2013–May 2014

### Associate Senior Leader September 2011–August 2013

### Head of Science April 2008–August 2013

### Head of Biology September 2004–March 2008

### Teacher of Science September 2002–May 2014

## Achievements

### AWS Certified AI Practitioner

- Confluence
- Datadog
- Docker
- GitHub
- GitLab
- JetBrains IDEs:
  - PyCharm
  - WebStorm
- Jira
- SharePoint
- Terraform
- VS Code IDE
- WordPress

### Amazon Web Services 2025

Earners of this certification understand AI, ML, and generative AI concepts, methods, and strategies in general and on AWS. They can determine the correct types of AI/ML technologies to apply to specific use cases and know how to use AI, ML, and generative AI technologies responsibly. They are familiar with the AWS Global Infrastructure, core AWS services and use cases, AWS service pricing models, and the AWS shared responsibility model for security and compliance in the AWS Cloud.

### **AWS Certified Developer – Associate**

#### Amazon Web Services 2025

Earners of this certification have a comprehensive understanding of application life-cycle management. They demonstrated proficiency in writing applications with AWS service APIs, AWS CLI, and SDKs; using containers; and deploying with a CI/CD pipeline. Badge owners are able to develop, deploy, and debug cloud-based applications that follow AWS best practices.

### **CS50 Introduction to Artificial Intelligence with Python**

#### Harvard 2024

### **Build a Backend REST API with Python & Django – Advanced**

#### Udemy 2022

### **CS50 Web Programming with Python and JavaScript**

#### Harvard 2022

### **CS50 Computer Science for Business Professionals**

#### Harvard 2022

### **Microsoft Innovative Educator Expert**

#### Microsoft 2021

### **Teach Computer Science Professional Development**

#### BCS 2020

### **Certified Entry-Level Python Programmer**

#### Python Institute 2019

### **MA in Education: Leading Innovation and Change**

#### St Mary's University 2012

### **PGCE Secondary Science (QTS)**

#### Kingston University 2002

### **BSc (Hons) Human Physiology**

#### Hertfordshire University 2001

### **Hobbies & Interests**

I enjoy snowsports, swimming and enjoy being outdoors in good weather. I often visit the cinema and enjoy reading both fiction and non-fiction.

### **Projects & Learning**

I'm currently learning **Next.js** with **NextAuth** and **Mantine** and exploring AI tools and services including: **Amazon Bedrock**, **Amazon Q** and **Amazon SageMaker** to build modern, integrated full-stack apps.

## GitHub Projects

### rst515.github.io

My GitHub website.

</> *html*

### aws-serverless-boilerplate

AWS serverless cloud resources template.

</> *aws, aws-lambda, aws-lambda-powertools, aws-sam, python, rest-api, tdd*

### django-graphene-energy-api

A mini GraphQL API demo built with Django + Graphene, modelling energy tariffs and usage data.

</> *django, graphene, python*

### FastAPI-CityApi

A simple FastAPI example project for storing cities.

</> *fastapi, python*

### videos-hooks

A YouTube video search app with React custom hooks.

</> *hooks, html, javascript, react*

### django-lottery-app

App to manage a lottery bonus ball game. Manage players and their allocated numbers and draws.

</> *aws-ec2, cron, css, django, html, javascript, python*

### serverless-booking-api

Serverless booking api demo with Python & AWS.

</> *aws, aws-lambda, aws-sam, python, shellscript*

### pwa-nextjs-demo

Demo of Progressive Web App (PWA) using Next.js.

</> *javascript, nextjs, push-notifications, pwa, react, typescript*

### rst515

My GitHub README.md for my public profile.

### github-readme-stats

Dynamically generated stats for my GitHub readme.

</> *javascript, stats*

### DjangoTDD

Example of TDD for a simple Django app.

</> *django, html, python, tdd*

### celeryExample

Basic example of celery task queuing with Django and Redis.

</> *celery, django, python, redis*

## **simple-password-generator**

API endpoint to generate a simple password. Returns a simple password string. This API endpoint is written in Python and built with the AWS Serverless Application Model (SAM).

`</> aws-lambda, aws-sam, html, python, rest-api`

## **network**

Social network app with Python and Django.

`</> css, django, html, javascript, python`

## **recipe-app-api**

Recipe API with the Django Rest Framework.

`</> django, django-rest-framework, python`

## **django-social-auth**

Django authentication demo with Google

`</> authentication, django, python`

## **django-dynamic-forms-with-htmx**

Example of replacing Django Formsets with HTMX elements, without using JavaScript.

`</> django, htmx, python`

## **pokemon**

A Pokemon crawler built with Python and Django.

`</> django, python`

## **react-pics**

An image search page with React, Unsplash API and custom CSS.

`</> css, html, javascript, react`

## **seasons**

Webapp that uses the current user's current location and date to display a message about the current season. Uses React, HTML, JSX, CSS and Semantic-UI.

`</> css, html, javascript, react`

## **widgets**

Simple React widgets (learning React)

`</> html, javascript, react`

## **videos**

A YouTube search app with React.

`</> html, javascript, react`

## **todo-htmx**

A single page todo app (SPA) with Python and HTMX.

`</> html, htmx, python`

## **clinic**

A patient booking app for a clinic with a Django backend and HTML, Bootstrap, JavaScript with some HTMX for the frontend.

`</> django, harvard-cs50, html, htmx, javascript, python`