Mahdi Haghverdi

Phone: +98 913 602 8987

Email: mahdihaghverdiliewpl@gmail.com

LinkedIn: mahdi-haghverdi

Website: mahdihaghverdi.github.io

Profile

Software Engineer with 2 years of professional experience building production backend systems and microservices, backed by over 6 years of Python programming, self-directed learning, and personal projects. Experienced in observability, distributed tracing, containerized deployments, CI/CD, and event-driven architectures. Currently expanding expertise in Kubernetes and cloud-native infrastructure while focusing on reliability, automation, and developer productivity.

Skills

- **Backend**: Python, FastAPI, FastStream, Redis, Redis Streams, Pydantic, PydanticAI, SQLAlchemy
- **Observability**: OpenTelemetry, Grafana, Grafana Alloy, Grafana Tempo, FluentBit, VictoriaMetrics, VictoriaLogs, LogFire, StructLog, Jeager
- **DevOps & Cloud**: Docker, Docker Compose, Kubernetes (learning), OpenStack (learning)
- **Architecture**: Microservices, REST APIs, Event-driven Architecture, System Design
- **CI/CD:** Design, Test, Deploy, and Production with GitHub
- **Soft Skills:** Communication, Curiousity, Teamwork, Collaboration, Problem-solving, and Public speaking

Professional Experience

Site Reliability Engineer — *Tehran, Iran*

Jast — *Jun 2024 – Dec 2025*

Description: JAST is a dynamic company focused on improving lifestyles through exercise and nutrition. Derived from the ancient Persian language, JAST represents the concepts of jump, mutation, and leap, symbolizing our commitment to dynamism and improvement. We offer personalized workout programs and nutrition plans that guide people toward their health and wellness goals. Join our vibrant community and experience the transformative power of JAST as we help you reach new heights in fitness and vitality.

- Designed and implemented a production-grade observability stack across microservices, combining structured logging, distributed tracing, metrics, and operational dashboards with Structlog, Fluent Bit, VictoriaLogs, Logfire, Grafana Alloy, and Grafana Tempo.
- Built real-time observability and incident diagnostics for production services, enabling developers to identify failures and performance bottlenecks and reducing incident investigation time by approximately 80%.
- Architected event-driven communication services supporting Web Push, real-time in-app notifications, email, and SMS delivery, enabling asynchronous and decoupled user-facing workflows.
- Developed a high-performance media processing service for image compression, resizing, cropping, and video thumbnail generation, reducing image payload sizes by 60%–80%. Whole processing, storing and returning the status between services took just 1 to 3 seconds for relatively large images.
- Built a real-time chat service for trainer–trainee communication and introduced a layered service architecture to decouple business logic from API and persistence layers across backend services.
- Modernized over the 70% of the backend API and data layer code base with typed SQLAlchemy 2.0 models, redesigned Pydantic schemas, standardized domain-specific error handling, and comprehensive unit and integration testing with Pytest, improving API consistency, validation, and OpenAPI documentation.

Projects

ConfigBundleOperator *Kubernetes, Operators*
k8s-operators-docs/ConfigBundleOperator

Description: A Python/Kopf-based Kubernetes Operator for managing application configuration through a Kubernetes-native `ConfigBundle` API. Implemented a reconciliation-based control loop that creates and updates managed `ConfigMaps`, synchronizes labels and annotations, detects configuration drift, and automatically self-heals missing or modified `ConfigMaps` using Kubernetes owner references and continuous daemon-based reconciliation.

- Designed and implemented a Kubernetes operator, introducing a declarative `ConfigBundle` CRD to manage and continuously reconcile application configuration as Kubernetes-native resources.
- Engineered a self-healing reconciliation mechanism following the Observe–Evaluate–Act pattern, continuously detecting configuration drift and automatically restoring the desired state when managed `ConfigMap` resources are modified or deleted.
- Implemented Kubernetes-native lifecycle and security management using Owner References and garbage collection for dependent resources, alongside RBAC policies supporting both namespace-scoped and cluster-wide deployments.
- Built an automated cross-repository documentation CI/CD pipeline with GitHub Actions and repository-dispatch triggers, dynamically integrating source repositories and generating versioned project documentation with Zensical.

pys3fuse *FUSE, Filesystem, S3*
pys3fuse

Description: A Python-based FUSE filesystem that exposes S3 buckets as POSIX-compatible filesystems, with bidirectional synchronization, metadata preservation, and concurrent S3 transfers.

- Designed and implemented a POSIX-compatible filesystem over S3 using FUSE and Python, exposing remote S3 objects through standard filesystem operations.
- Engineered bidirectional synchronization between local filesystems and S3, with concurrent change detection, conflict handling, deletion propagation, and POSIX metadata preservation.
- Implemented concurrent and configurable S3 transfers using `ThreadPoolExecutor` and `boto3` multipart uploads, supporting efficient synchronization of large files and multi-user environments.

Articles

Timsort Algorithm *Algorithm, CPython*
timsort-algorithm

Description: A technical deep dive into Timsort, covering its adaptive sorting strategy, run detection, merging mechanism, and CPython implementation details.

- Deep-dived into Timsort's adaptive stable merge-sort architecture, analyzing its hybrid use of Binary Insertion Sort for small runs and Merge Sort for large-scale merging.
- Analyzed natural run detection and `minrun` computation, demonstrating how Timsort exploits partially ordered data to reduce merge operations and improve practical sorting performance.
- Studied the CPython implementation of Timsort and its performance characteristics, including comparison costs, memory usage, cache locality, and merge-tree optimization.

Education

Bachlors of Computer Engineering *Isfahan, Iran*
University Of Isfahan *2021 – 2025*

- Grade: 17.44/20
- USA GPA: 3.48/4
- Thesis: Implemented transparent FUSE filesystem layer to access S3-based storages with pure POSIX commands (ls, cd, mkdir, ...)