

# Pete Doherty

Principal Owner, 21st Century Alchemy

pdoherly@protonmail.com · peterdohertys.website · linkedin.com/in/pdoherty926 · New York City Metro

## SUMMARY

---

AI/ML ops practitioner and software engineer with 15+ years building production systems across startups, agencies, and enterprise clients. Specializes in the infrastructure layer of modern ML: model retraining pipelines (Prefect), vector and embedding workloads (pgvector, TimescaleDB, DuckDB) and the platforms that run them (AWS, GCP, local Mac). Designed and operated retraining pipelines for proprietary computer-vision models at Iterative Health (e.g. procedure event detection) and built large-scale semantic video search and recommendations at OZU on pgvector + TimescaleDB — including diagnosing and reporting a vector-index bug in TimescaleDB that was subsequently patched upstream. Equally fluent in the underlying engineering: Python (FastAPI, Flask, uv, pytest), Rust (Tokio, duckdb-rs), Clojure, AWS infrastructure-as-code (Terraform, Terragrunt), and zero-trust security design. Notable clients include Iterative Health, OZU, Conde Nast (Vanity Fair), The New York Times (Fake Love), Bleacher Report, MycoWorks, Stink Studios, and Deeplocal.

## CORE COMPETENCIES

---

<b>ML Ops</b>	Prefect 2.x (flows, deployments, concurrency limits, work pools), model retraining pipelines, AWS Batch for GPU job scheduling, Weights & Biases (wandb) experiment tracking, data versioning, training/eval orchestration, experiment reproducibility, drift monitoring
<b>Vector / Embeddings</b>	pgvector (HNSW indexing), TimescaleDB vector extension (incl. upstream bug triage and fix collaboration), embedding pipelines, semantic search at scale, hybrid lexical/vector retrieval, multi-tenant vector architectures with django-tenants (schema-per-tenant Postgres)
<b>DevOps</b>	Terraform, Terragrunt module composition, CI/CD (GitHub, GitLab, TeamCity, Jenkins, etc.) Linux (Ubuntu, systemd, bash), Docker, containerization, monitoring, Heroku, Digital Ocean, GCP
<b>Python</b>	FastAPI, Flask, Django, uv, pytest, PyPI publishing, asyncio, boto3, Pydantic, data pipelines (Prefect), CLI tooling, scientific stack (NumPy, pandas, PyTorch interop)
<b>AWS</b>	Batch, Deadline Cloud, EC2 (GPU & CPU fleets), Lambda, S3, FSx (ONTAP & OpenZFS), EFS, API Gateway, VPC, IAM, Secrets Manager, Route 53, SES, CloudWatch, VPC Lattice, CloudFront, Elasticsearch
<b>Security</b>	Zero-trust architecture, principle of least privilege, IAM policy design, security audits, secrets rotation (Secrets Manager), VPC network segmentation, CVE remediation, Cloudflare Zero Trust (WARP, Tunnel) for identity-aware remote access
<b>Databases</b>	PostgreSQL (schema design, query optimization, high-concurrency tuning, connection pooling, pgvector), TimescaleDB, DuckDB (embedded OLAP, Parquet/Arrow, VSS), Datomic
<b>Other Languages</b>	Rust (Tokio, duckdb-rs, async services, CLI utilities, GStreamer integrations), Clojure (Ring/Pedestal, Datomic, EDN, REPL-driven dev), Elm/Gren (production SPAs, 0.19 → Gren migrations)

## EXPERIENCE

---

**Principal Owner · 21st Century Alchemy**

*Aug 2020 – Present · New York*

*Creative technology, ML-ops and DevOps consultancy — clients include Iterative Health, OZU, Bleacher Report, Deeplocal, ESI/NBBJ, Hi5, Hush, Leo Villareal Studio, MycoWorks, Verse, and others*

- **Iterative Health — ML retraining pipelines (Prefect 2):** Designed and operated retraining pipelines for proprietary computer-vision models detecting clinical events in endoscopy video. Built Prefect 2 flows

that dispatched GPU-bound training stages to AWS Batch, with concurrency-limited work pools controlling parallelism across training, data prep, and evaluation stages and deployment-level scheduling for periodic and on-demand retraining triggers. Integrated Weights & Biases (wandb) for experiment tracking and run comparison; implemented run-level idempotency, artifact persistence to S3, and structured failure handling so partial pipeline failures could be resumed without redoing expensive training stages.

- **Iterative Health — Cloudflare Zero Trust remote access:** Designed and dynamically integrated Cloudflare WARP and Cloudflare Tunnel into the lab environment to provide secure, identity-aware remote access for researchers and engineers without exposing internal services to the public internet. Stood up per-user, per-application access policies aligned with zero-trust principles in place of a traditional VPN.
- **OZU — semantic video search & recommendations on pgvector + TimescaleDB:** Architected and built a production vector-search platform for video discovery and recommendations. Generated and indexed embeddings in PostgreSQL/pgvector with TimescaleDB for time-partitioned content, tuning HNSW parameters and recall/latency trade-offs against real query workloads. Used django-tenants to partition vector data into per-tenant Postgres schemas, keeping individual indexes small and bounding insert and query cost per tenant rather than letting a single shared table dominate performance. Diagnosed a non-obvious correctness bug in TimescaleDB's vector index behavior, isolated a reproducer, and engaged with the Timescale team — the issue has since been fixed upstream.
- **AWS render farm with Deadline Cloud:** Architected and operate a cloud-native render farm: EC2 Linux/Windows fleet, FSx for ONTAP, LucidLink asset synchronization and Secrets Manager credential management. Selected ONTAP after evaluating Windows SMB compatibility. Authored all infrastructure in Terraform/Terragrunt; used an 80/20 hot/cold storage model, cutting projected monthly storage spend by ~80% vs. a non-tiered baseline.
- **Zero-trust IAM and network design:** Designed all IAM policies under strict principle of least privilege: scoped execution roles for Deadline Cloud workers and Prefect agents, per-service Secrets Manager grants, and VPC segmentation with explicit security-group egress rules.
- **DuckDB analytical layer:** Deploy DuckDB for in-process analytical queries and ingress of various data streams; built Rust services against duckdb-rs for persistent, file-backed analytical workloads and published a technical write-up (“A Dab of DuckDB”) featured in MotherDuck’s newsletter. Ongoing personal exploration of DuckDB VSS as a lightweight embedded vector store for edge and single-tenant scenarios.
- **Polyglot service work:** Build and ship production Clojure services (RESTful APIs, background workers, EDN config, REPL operational tooling); develop Rust CLI tools and media-pipeline components including GStreamer-backed video processing and a Raspberry Pi WebRTC video-wall driver; maintain Elm and Gren frontends, including an Elm 0.19 → Gren migration .

#### **Freelance Software Developer · Bleacher Report**

*Jun 2015 – Jun 2020 · Remote*

- Built a custom CMS powering 300+ long-form stories on thelab.bleacherreport.com and mag.bleacherreport.com, with full editorial workflow and asset management.
- Developed a Clojure number-crunching web service for the ‘Emoji is Life’ editorial project, processing large structured datasets to derive emoji-usage statistics at scale.
- **Stack:** AWS, Clojure, Cucumber, Docker, Digital Ocean, Elm, Jenkins, Python, React, Ruby/Rails.

#### **Selected Engagements**

*Feb 2019 – Mar 2020*

- **MycoWorks:** Built and scaled a Python/Flask mobile web application for laboratory mushroom grow-operation tracking; led security audits (including CVE triage and remediation), Heroku migration, performance improvements, structured database migrations, and containerization. Stack: Python, Flask, PostgreSQL, Heroku, GCP.
- **Stink Studios / Chobani:** Migrated a high-traffic legacy microservice application between AWS accounts under a zero-trust security model; performed comprehensive security audit, enforced least-privilege IAM roles, and remediated all findings. Stack: AWS (EC2, S3, API Gateway, Lambda, Elasticsearch, Route 53), Clojure.
- **Stink Studios / YouTube:** Built a Python-based video-generation service dynamically pairing creator content with Stink’s RiTA service to produce personalized YouTube Creator Award videos. Stack:

Python, Django/Djangae, Node.js, GCP (App Engine, Cloud Functions, Cloud Storage), FFmpeg.

**Freelance Software Developer · Fake Love (The New York Times)** Jun 2013 – Dec 2018

*Software development, systems architecture, Linux sysadmin, DevOps, and monitoring across interactive experiences, digital signage, and red-carpet installations*

- **Selected projects:** AKQA/Levi's Station-to-Station; Droga5/belVita #morningWin Trophybot; Equinox Equifit; HP/Star Wars Red Carpet Sound-Off at The Force Awakens world premiere; Mother/Samsung Funhouse @ Virgin Freefest; Nokia Bell Labs data visualization; SK-II Wonderland photo installation (Tokyo); Samsung/Google Liquid Canvas (worldwide).
- Applied security-first practices throughout: hardened Ubuntu servers, principle of least privilege for all AWS service roles, encrypted secrets handling, and regular dependency audits across polyglot stacks.
- **Stack:** AWS (EC2, Lambda, S3, CloudFront), Digital Ocean, Docker, Heroku, Linux/Ubuntu/systemd, Clojure, Kotlin, Python, JavaScript (Backbone.js, Vue.js), Node.js, Ruby/Rails/Sinatra, PostgreSQL, MongoDB, Redis, RabbitMQ.

**Selected Short-Form Engagements** 2018

- **Vanity Fair (Condé Nast):** Web development for '2019 Hollywood Cover: Moving Pictures' and '25 Most Influential Movie Scenes of the Past 25 Years.' Stack: Gatsby, React, Airtable, S3, CloudFront, GitLab CI/CD.
- **Havas Sports & Entertainment:** Built an interactive mobile-first application for The Atlantic's Intelligent Future conference. Stack: Elm, AWS (S3, CloudFront, Lambda, SES).

## EARLIER EXPERIENCE (SELECTED)

---

**Wondersauce · Signpost · Lamplighter Labs · Huge · GramGoods · Poppin · Penguin Digital (acq. Shutterfly) · Massify · SignalFive**

- **Recurring stack:** Ruby on Rails, PostgreSQL, Python/Flask, Backbone.js, Heroku, Redis, Stripe, Sass, Node.js.
- **Highlights:** Penguin Digital photo-merchandising apps (MoPho, PicsCliq); Lamplighter Labs hypermedia API for mobile image-sharing; GramGoods e-commerce from wireframes to market (Stripe, Instagram API); Afghan Youth Initiative pro-bono platform.
- Massify and Vidburn: both mentioned in *The New York Times*.

## EDUCATION

---

**Bachelor of Arts · Roger Williams University**

2002 – 2006

## TECHNOLOGY KEYWORDS

---

Prefect · Model Retraining · MLOps · Computer Vision · Weights & Biases · wandb · pgvector · TimescaleDB · DuckDB · DuckDB VSS · Vector Search · Embeddings · Semantic Search · Multi-tenant Architecture · django-tenants · Cloudflare Zero Trust · Cloudflare WARP · Cloudflare Tunnel · Python · FastAPI · Flask · Django · uv · pyproject.toml · pytest · boto3 · Pydantic · asyncio · AWS Batch · Deadline Cloud · CloudFront · EC2 · Lambda · S3 · FSx ONTAP · EFS · API Gateway · VPC · IAM · Secrets Manager · Route 53 · SES · CloudWatch · VPC Lattice · Zero Trust · Principle of Least Privilege · Security Audits · Terraform · Terragrunt · Rust · Tokio · duckdb-rs · GStreamer · Clojure · Datomic · Elm · GREN · PostgreSQL · Parquet · Apache Arrow · Ruby on Rails · Node.js · React · Docker · Linux · systemd · bash · Jenkins · GitLab CI/CD · Redis · MongoDB · RabbitMQ · Digital Ocean · GCP · Heroku · FFmpeg · WebRTC