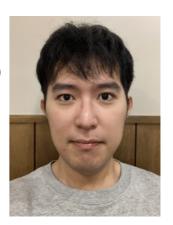# Kun-Chieh (KC) Hsu

Experience

- 9/14 National Tsing Hua University, Taiwan: materials science major, CS minor
- 7/17 ~ 8/19 (**2.2Y**) HIGH5.ai: **ML and Full Stack Developer** (founding team) Designed a system enabling customers to retrain their own sentence classification models upon request. Later, took on the responsibility of maintaining both the frontend and backend codebases.
Stack: **NumPy**, **scikit-learn**, **PyTorch**, **Optuna**, **NLP**, **NetworkX**, **aiohttp**, **Angular**, **Node.js**, **Elasticsearch**, **RabbitMQ**, **Docker**, **BitBucket Pipelines**, **Kubernetes**, **GCP**
- 6/20 ~ 6/23 (**3Y**) aetherAI: **Sr. DevOps** (the only DevOps)
- 7/23 ~ 12/23: doing projects, traveling, playing GarageBand, working out

Missions in aetherAI

The product was an AI pathology system, so most of the production deployments were done **offline** in **hospitals**.

- **Kubernetes**: Migrated the product from **Docker** to Kubernetes by integrating a range of open-source solutions, including **Helm**, **MicroK8s**, **MetalLB** (bare metal load-balancer), **Longhorn** (distributed block storage), **Patroni** (**high-availability PostgreSQL**), **SMB** driver, **Elastic stack**.

- **AWS**: Managing **EC2**, **RDS**, **ELB**, **EFS**, **S3** and **Storage Gateway** with console and **Terraform**. Found a solution to mount an **S3 bucket** as a **file system** with Storage Gateway.

- **Backend**: Designed a **network topology** not only compatible with 3 different architectures: single-node, dual-node, and Kubernetes, but also secure for cross-node communication. **Nginx**, **Caddy** and **mTLS** were heavily used. Developed a load testing tool with **Golang**.

- **Monitoring**: Designed a remote monitoring solution for hospitals that only allowed **SMTP** outflow traffic. This included a **Python** container, built upon **Docker SDK** and **Kubernetes API**, for periodically emailing metrics. An internal CI would fetch, process and store them in **Elasticsearch**. **Grafana** for visualization. **ElastAlert** for Slack alerting. **Sentry** for on-site monitoring.

- **CI/CD**: Responsible for optimizing all **Dockerfile**s, **GitLab CI** pipelines, managing on-premises **GitLab Runner**s and **Harbor** (**container registry**). Integrated **Earthly**, **Trivy** and **ArgoCD** into the CI. Identified the problem and designed a pipeline to create unique offline deployables for each customer, using ClickUp as the single source of truth. **Bash** and **Python** were heavily used.

- **Windows**: Ported a Python server to Windows and designed a solution to make it **highly available** and secure. **PowerShell**, **Nginx**, **Caddy** and **mTLS** were heavily used. Developed a toolset based on **Ansible Playbooks** for FAEs to install, update, and control it from Linux.

- **Environment Parity**: Identified the problem and designed a unified toolset for developers to bring up their own development environments independently and for field application engineers (FAEs) to deploy various production environments effortlessly. It had to be compatible with both **Docker Compose** and **Helm**.

- **Mentoring**: Helped mentor the FAE team on Linux, Docker, Kubernetes and Windows. Additionally, provided on-site and remote operation support.

Personal Projects

ipttrace

A CLI let you trace iptables rules at ease
Stack: **Python**, **typer**, **iptables**, **dmesg**, **PyPI**

kwarg-sort

A VS Code extension that sorts the keyword arguments in a Python function alphabetically.
Stack: **JavaScript**, **VS Code API**

stock-sense.info

A website features animated income statements and P/E ratio bands for public companies in the US and Taiwan.
Stack: **Python**, **Plotly**, **Plotly Dash**, **Pandas**, **SEC API**, **Docker**, **GitHub Actions**, **Azure Container Apps**, **Namecheap**, **CloudFlare**

portman

My own portfolio management server which is capable of fetching historical prices for US and Taiwan stocks, as well as cryptocurrencies. Based on that, it can calculate downside risk for each position and conduct MACD parameter optimization. The trading signals are presented through a Streamlit app.
Stack: **Numba**, **Python**, **FastAPI**, **NumPy**, **Streamlit**

UltraTracer

Fast and Concise Ray Tracer with SIMD Acceleration
Stack: **C++**