SHOMIL JAIN

shomil.jain@gmail.com • (408) 960-3403 • linkedin.com/in/shomilj

SUMMARY

Experience in building distributed systems and foundational infrastructure to orchestrate systems for machine learning at large scale.

WORK EXPERIENCE

Staff Software Engineer / Tech Lead, Infrastructure - Anyscale **August 2022 - Present**

Anyscale on Kubernetes / Machine Pools Tech Lead

- Led the design and implementation of the Anyscale data plane for Kubernetes and on-premise machines. Built a Kubernetes operator, Anyscale daemon for machine management, and control plane adapters. For Kubernetes, lead seven engineers over a three month cross-team effort to to unblock online inference and batch workloads on K8S for Anyscale's largest customers.
- Built a scheduler to maximize workload GPU utilization with fair-share/FIFO scheduling policies, queuing, and preemption.

Core Infrastructure Systems

- Built deployment, service discovery, and configuration management systems from scratch to deploy thousands of single-tenant services on Kubernetes; then, refactored services to support secure multi-tenant deployments (GRPC, Kubernetes).
- Redesigned and shipped new customer & internal-facing observability stacks for perf. & scalability (Loki, Cortex, Grafana).
- Identified a need and introduced Kafka to Anyscale as a means of decoupling nearline systems (e.g. observability, state for UI) from online ones (e.g. controllers, operators). Built and maintained client libraries for event consuming / producing.

Scaling / Stability

- Worked on full-stack performance improvements for Ray/Anyscale infrastructure scalability, reaching millions of CPU's/GPU's under management supporting diverse Ray workloads, e.g. foundation model training, RL, protein folding, quant. trading.
- Made it possible to run Ray clusters spanning multiple regions & clouds using overlay networks (using Tailscale/WireGuard), allowing Anyscale to leverage cheaper / more available cross-region GPU's for Anyscale Endpoints + Hosted Anyscale products.
- Contributed to Anyscale's serverless offering through fast startup (building VM instance warm pooling) and decorator-driven autoscaling, e.g. @ray.remote(accelerator_type="A100-80G") to automatically find and launch an A100 machine (EC2, GCE).
- Oversaw infrastructure team on-call, dashboards, alerting, runbooks, etc. to streamline incident detection and response. Debugged many full-stack issues on both internal and customer workloads (pprof, strace, py-spy, nvidia-smi).

Software Engineering Intern, Infrastructure - Affirm **May 2021 - August 2021**

- Designed a system to configure and manage Elasticsearch/Kibana alerts and dashboards through a template language. Built systems to provide out-of-the-box monitoring for Affirm's 100+ microservices to reduce operational overhead.

Software Engineering Intern, Palo Alto Networks **May 2021 - August 2021**

- Wrote a framework to statistically analyze terabytes of logs collected from Palo Alto Network's VPN products to flag anomalies.

SIDE PROJECTS

Reverse Engineering / Privacy Engineering Articles **November 2021 - May 2022**

A variety of reverse engineering explorations to identify privacy concerns with popular apps (BeReal, Snackpass, COVID-19 Apps).

Orbit - React Native/JavaScript (v2), iOS/Swift and Android/Java (v1) **May 2016 - May 2022**

A microservice-driven platform powering Bear Central (+ other campus apps). Used by thousands of students (portfolio, github).

Paz/Sona - iOS, Swift, Google Cloud **April 2021 - May 2021**

A music-streaming platform focused on restorative music. Designed/launched the startup's iOS app (portfolio, app store).

Diversity in EECS @ UC Berkeley - Python, Pandas, Plotly, Flask **August 2020 - October 2020**

An interactive article and API endpoint exploring campus-wide admissions data from 2000 to the present (article).

EDUCATION

University of California, Berkeley – B.S. Electrical Engineering & Computer Science **August 2018- May 2022**

- Served as Head TA for CS 161 - Computer Security. Taught OS-level, network-level, and web attacks + defenses.
- Built infrastructure for running classes at scale, including an automated extensions system (cs161-staff/extensions).