Arvind Iyengar

Staff Software Engineer

linkedin.com/in/iyengararvind
iyengararvind@gmail.com | (954) 249-5297
San Francisco, CA 94109

Education & Teaching

University of California, Berkeley

B.A. Computer Science (GPA: 3.5)

Student Instructor | UC Berkeley

Developed course content & taught laboratory / discussion sections for:

- Data Science (3 semesters)
- Probability Theory (2 semesters)
- Computer Security (2 semesters)

Textbook Co-Author | UC Berkeley

Co-wrote, revised, and finalized the textbook used for STAT 88

Relevant Coursework

Computer Science: Algorithms, Data Structures, Machine Structures, Computer Security, Cryptography
AI / ML: Probability Theory, Stochastic Processes, Natural Language Processing, Artificial Intelligence

Technical Skills

Cloud Providers
AWS | Azure | Google Cloud

Programming Languages
Go | Python | Bash | Java | Javascript

Technologies
Kubernetes (RKE | k3s | RKE2) | Helm
Terraform | Containerization (e.g. Docker)
Prometheus | Grafana | Alertmanager

About Me

- Passionate about scalable AI tooling, such as vector search, agentic workflows, and AI infrastructure
- Excited by products that leverage modern technologies like cloud computing and Kubernetes
- Believe in writing simple, readable, and maintainable code in a world of complex applications
- Highly self-motivated with a proven track record through open-source contributions
- Strong communicator with an extensive background in teaching & writing technical documentation

Experience Summary github.com/aiyengar2

Rancher Labs | Staff Software Engineer

Feb 2020 - Jun 2024 (4 yrs 4 mos)

- Worked as a team lead and subject matter expert for several components:

Monitoring / Alerting

Github: rancher/charts | rancher/prometheus-federator

- Designed & maintained Helm charts to deploy Prometheus, Alertmanager, Grafana, and Prometheus Operator with custom exporters and dashboards
- Developed Rancher Pushprox, a Helm chart that allows Prometheus to securely scrape host network processes without firewall exposure. Configured for Kubernetes internal components in RKE, RKE2, k3s, kubeAdm, GKE, AKS, & EKS
- Implemented a reverse TCP tunneling proxy in wins, enabling Docker containers to proxy host ports on Windows hosts added to a Kubernetes cluster
- Developed Prometheus Federator, a Kubernetes operator that allowed administrators to group namespaces, automatically deploy scoped Prometheus & Grafana instances, and enforce hard multi-tenant security by locking resources and blocking unauthorized changes on the deployed Kubernetes resources

Helm Charts

Github: rancher/charts-build-scripts | rancher/hull

- Managed the release of 50+ Helm charts over 9+ release cycles
- Developed Charts Build Scripts, a tool for managing and releasing Helm charts, enabling Rancher to streamline chart updates across all repositories
- Created Hull, a Go-based testing framework for writing comprehensive unit tests on Helm charts, improving reliability and validation

Windows

Github: rancher/windows | rancher/Rancher-Plugin-gMSA

- Developed 18 Terraform modules (6 public, 12 private) for consistent Windows cluster provisioning across cloud providers, simplifying issue reproduction
- Designed a Container Credential Guard plugin to enable internal Windows applications deployed on Kubernetes to authenticate users against Active Directory via group managed service accounts (gMSA)
- Authored 20+ Markdown files documenting Windows, Kubernetes, and Active Directory, preserving tribal knowledge and streamlining onboarding

Other Responsibilities

- Ran knowledge transfers on developing Kubernetes operators & other topics
- Established deep expertise in Fleet, Rancher's GitOps solution
- Implemented continous integration via scripts written in Bash & Github workflows

AWS Sagemaker | Software Engineer

Jul 2019 - Jan 2020 (7 mos)

- Developed and executed Bash scripts and CloudFormation templates to provision AWS infrastructure for the AWS Sagemaker's API Service

AWS Lambda | Software Engineer Intern

Jun 2018 - Aug 2018 (3 mos)

- Designed a serverless ETL (Extract, Transform, Load) pipeline to reduce query time for complex analytics queries on Redshift databases in slow, limited-access data centers using CloudWatch Events, Lambda, and CloudFormation