Arvind Iyengar

Santa Clara, CA 95054 | (954) 249-5297

iyengararvind@gmail.com | github.com/aiyengar2 | linkedin.com/in/iyengararvind

Senior Software Engineer

University of California, Berkeley - B.A. Computer Science (GPA 3.5)

- Strong foundations in software engineering & development principles with a passion for **learning about new technologies**, **delivering customer impact**, and **designing simple, readable, and maintainable code** in a world of complex applications
- Excited about **driving product design and development** that leverages modern technologies like **cloud** and **Kubernetes**
- Highly self-motivated with **strong communication skills** stemming from **a diverse background in teaching technical concepts**

Professional Experience

SUSE Software Solutions Germany GmbH (Rancher Labs)

Feb 2020 - Present (Software Engineer)

- Architected & maintain **rancher/charts-build-scripts**, a set of Go scripts that enable Rancher to **build, test, & release charts** in a Helm repository (**rancher/charts**, **rancher/rke2charts**); these scripts support **developing patched versions of upstream charts**, which ensures that feature charts (Monitoring, Istio) are closely aligned / up-to-date with upstream community solutions
- Designed & manage **Rancher Monitoring and Alerting V2**, a Helm chart that enables users to quickly deploy **Prometheus, Alertmanager**, and **Grafana** on a Kubernetes cluster with **default Prometheus exporters** and **Grafana dashboards**
- Engineered **Rancher PushProx**, a Helm chart that configures **Prometheus** to securely **scrape internal Kubernetes components** (that run as **host network processes without exposed ports**) in RKE, RKE2, k3s, kubeadm, and GKE / AKS / EKS clusters
- Coordinated the v0.1.0 release of **wins**, a binary that **enables Docker containers to manage host processes in Windows** (which is currently used by the SIG-Windows Kubernetes community to enable Windows hosts to join a cluster), by **adding support for proxying host ports** via a wins client and **designing a Helm chart capable of upgrading wins** across Windows hosts in a cluster
- Produced **Rancher Windows Exporter**, a Helm chart that **deploys windows_exporter.exe as a host process** on Windows hosts, **exposes Windows host metrics to Prometheus via a wins proxy**, and **enables Grafana dashboards for Windows**
- Devised a **unit testing framework for Helm charts** that defines tests on resources produced by the template on multiple values.yaml
- Upgraded **Azure** client and added support for **SLES Docker installs** to the set of drivers used for **Kubernetes node provisioning**

Amazon (AWS Lambda, SageMaker)

Jun 2018 - Aug 2018 (Software Engineer Intern) | Jul 2019 - Jan 2020 (Software Engineer)

- Generated a serverless application on **SAM / CloudFormation** that **executes ETL (Extract, Transform, Load) workflows** via **CloudWatch Events** and **Lambda** on any specified **Redshift** database to **reduce the query time** necessary for **running complex customer data analytics queries** frequently on existing raw log data sources housed in slow data centers with limited access. This application supported **monitoring & logging via CloudWatch**, customizable **networking and access management via VPC / IAM**, and easy deployment / validation via **operational scripts written in Bash**
- Collaborated with a team of engineers across **SageMaker** to build out the **API Service** for the product in a new AWS region

Publicis Groupe (Razorfish)

Jun 2016 - Aug 2016 (Machine Learning Intern) | Aug 2016 - Dec 2016 (Presentation Layer Engineer)

- Constructed an **Express.js (Node.js) application** built on Microsoft Bot Framework, Microsoft Cognitive Services, and a natural language processing framework (Wit.ai) that enabled clients to **quickly train / deploy chat bots** on FB Messenger, Skype, and Slack

Relevant Experience

Work Experience, Personal Projects, Technical Skills, and Coursework

Undergraduate Student Instructor (uGSI) – Developed course content & taught laboratory / discussion sections for a total of six semesters at UC Berkeley for CS 161 (Computer Security), STAT 140 (Probability Theory), and DATA 8 (Data Science)

Microservice Architecture – Created a simple serverless application that stands up two Flask programs running on different Docker containers that communicate with each other through HTTP requests to demonstrate modern software architecture
Probabilistic Decoder – Built a probabilistic decoder to break encrypted ciphers using Markov Chain Monte Carlo methods
Speech Analysis – Researched Obama's speeches using text analysis (Latent Dirichlet Allocation) and visualization techniques
Probability & Mathematical Statistics Textbook – Co-wrote, revised, and finalized the textbook used for STAT 88 at UC Berkeley

Programming Languages – Go, Bash, PowerShell, Python, Java, Javascript (Node.js, Express.js), Ruby, C, HTML / CSS, SQL
Open-Source Technologies – Kubernetes (RKE, RKE2, k3s), Docker, Helm, Prometheus, Grafana, Alertmanager, nginx, wins
Cloud Technologies (AWS, Azure, Google Cloud) – Familiar with the following technologies and their equivalents across cloud providers: CloudFormation, Lambda, EC2, ECS, S3, CloudWatch, CloudTrail, IAM, SQS, SNS, Redshift, DynamoDB

Coursework – Algorithms, Data Structures, Databases, Computer Security, Cryptography, Natural Language Processing, Stochastic Processes, Probability Theory, Principles & Techniques of Data Science, Artificial Intelligence, Machine Structures