Kevin Mok

647-685-2500
me@kevin-mok.com
linkedin.com/in/Kev-Mok
github.com/Kevin-Mok

Work Experience

Red Hat

May 2020 — Aug 2021

Cloud/Software Engineer Intern <Kubernetes, GoLang, Jenkins>

- Reduced deployment time by **66%** by implementing ability to deploy locally-compiled binaries onto **Kubernetes**/**OpenShift** using only command-line (**Kubernetes**/**GoLang** used for this and three below).
- Implemented ability for Kubernetes operator to fetch data from a deployed service and update config with data to deprecate reliance on startup script.
- Added startup probes to handle starting legacy application containers that require additional startup time.
- Refactored probes to have default values assigned based on deployed YAML while also fixing reconciliation issues.
- Rewrote the **Jenkins (Groovy)** nightly pipeline to run in a GitHub PR using a trigger keyword to test all the team's submitted PR's before merging to the main branch.
- Took on tasks and contributed ideas in **Agile sprint** planning meetings in a team of 12 people.
- Took initiative to write file to define the GitHub parameters for the above pipeline so that it can be recreated easily.
- Took initiative to write documentation on how to get started with the project to onboard new developers and mentored the incoming intern.

Projects

Rarity Surf <Python, Django, React, GraphQL>

Oct 2021

- Web app to give rarity rankings to NFT's and check which are listed on the OpenSea marketplace using their API.
- Reverse engineered the ranking algorithm to match the leading rarity ranking site's rankings (scraped using **Selenium**) with a **discrepancy of <0.25%**.
- Used app to frontrun purchases of **top 0.5%** rarity NFT's against competing buyers.
- Wrote **Django (Python)** backend to fetch metadata from IPFS, store rarity rankings in PostgreSQL and serve rarity data using GraphQL.
- Wrote **React** frontend with hooks to dynamically load rarity data. Styled with Tailwind.

Spotify Graphs <Python, Django, PostgreSQL>

June 2020

- Web app to visualize user's library and listening history from Spotify.
- Fetched and organized user data from Spotify API into PostgreSQL database with **Django (Python)** backend.
- Created and tested various relational database schemas to maximize efficiency for use cases.
- Generated various charts to visualize the artists and genres of tracks in library.

AWS Server <AWS, Kubernetes, Terraform, Docker>

May 2024

- Deployed various web apps using **Docker** (Compose) on an **AWS EC2** Debian/**Linux** server.
- Created **Kubernetes** manifest files to quickly recreate my server setup with persistent storage/restarts and open ports.
- Created **Terraform** files to deploy an AWS EC2 instance and Docker containers.
- Used Amazon Route 53's DNS and **NGINX** to route subdomains to each web application.
- Wrote a **JavaScript** server script and systemd service/timer to display the uptime of my pages every hour.

Skills

Python, **Django**, **Linux**, **Git**, AWS, Kubernetes, Docker (Compose), Ansible, Jenkins, Selenium, Terraform, Go(Lang), PostgreSQL, **Command Line**, Groovy, Bash, **JavaScript**, **React**, Node.js, MongoDB, Solidity, C

Education

University of Toronto

2019 — 2024

Computer Science Specialist — 3.84 GPA (CS). Graduated with High Distinction.

References

See my LinkedIn for references from my Red Hat managers/mentee, a startup client and a graduate student mentor.