Neil N. Toledo

(619) 990 - 3034 · ntoledo@berkeley.edu · ntoledo.me

EDUCATION

University of California, Berkeley | Berkeley, CA | Aug 2015 – May 2019
B.S. in Electrical Engineering and Computer Science (EECS)

PROFESSIONAL EXPERIENCE

Zume, Inc. | Site Reliability Engineer [2019 - 2020]

- Worked extensively with Google Kubernetes/GCP, Datadog and Helm Charts
- Developed a suite of python and bash scripts for automating set up of infrastructural and application monitoring and started initiatives for consolidating our third party monitoring services for better resource organization and reducing unnecessary costs.
- Assisted with reorganization of SRE On Call practices as well as pushed initiatives for reorganization of team documentation

Infinite Uptime | Data Science Intern [Summer 2018]

- Developed a responsive data visualization app for monitoring movement and conditions of manufacturing presses and machines using Python, Flask, and the Bokeh library. Featured a tab system to display data in different interactive formats and make future development and features easier to include and implement. Deployed using Docker and Google Compute Engine and Kubernetes.
- Added a feature to an existing admin Flask API that invokes a Google Compute Engine VM instance to run Google Big Table data migration jobs and automatically shutdown to minimize VM uptime and costs.

PERSONAL PROJECTS

Keel Helm3 Provider - Open Source Contribution
Summer 2020

Added support for Helm v3 to Keel.sh, an open source Kubernetes Operator that automates Helm and Application version updates. Upgraded Golang dependencies to work with both Helm v3 and Helm v2 and wrote and added a new "Helm3 Provider" to the application source code.

Portable CI/CD System
Spring 2020

Developed a custom polling based CI/CD system with portability in mind. Intended to be deployed on Kubernetes using Helmsman and ConfigMaps for configuration. Includes separate pipelines for both Docker Images and Helm Charts while allocating code ownerships appropriately between developers..

MMORPG System - Software Design Document
Summer 2020

Wrote comprehensive design documents for the software architecture of an web MMORPG project featuring a distributed system deployed using Kubernetes.

- Features custom implementation details for a horizontally scaling fleet of game server instances to divide authority over in-game geographical areas.
- Includes a microservice architecture for implementing additional services such as in-game chatting, character data management and quest management.
- Component design documents include step by step flowcharts for describing exactly how features are implemented and interactions between different components.

CMS API - Software Design Document
Summer 2019

Wrote a comprehensive Software Design Document that includes a general overview and implementation details of a CMS API project that I'm planning to develop as an open source project. The Design Document covers the project's Microservice Architecture, Data and API Design, and Interface Design with some implementation details.

Other Projects

Personal Site/Blog: Personal Jekyll website for showcasing projects and maintaining a consolidated reference of notes. Website also showcases and maintains documentation for larger projects.
mini5-engine: Barebones game engine for JavaScript and HTML5 Canvas. Handles keyboard and mouse input processing, robust update and draw loops, scene and object rendering and basic AABB collision detection. Published on github and npm.
Worc_Lock: Virtual work clock featuring a CRUD API using Express.js, React.js, and SQLite.

SKILLS

Technical Languages: Python, Javascript, Golang, Java, C, Bash Scripting
Additional Technologies: ReactJS, HTML, CSS, SQL, Django, Flask, BokehJS, Jekyll, NumPy and Pandas, Express.js
Software/Tools: Google Compute Engine and Kubernetes, Docker, Helm, Datadog, Git/Github, Unity, Adobe Illustrator, Autodesk Maya, Fusion360
Languages: Fluent in English and Tagalog