Adamo Devigili

Austin, TX

(609)658-6387

adam.devigili@gmail.com

github.com/adamdevigili

linkedin.com/in/adamdevigili

devigi.li

Education

Western New England University - *School of Engineering* — *Aug 2011 - May 2015*

B.S. Computer Science; Concetration in A.I. and distrubted computing

Experience

IBM - *Advisory Software Engineer* *(Band 8)* — *Jan 2020 - Present*

- Senior engineer on rapid boostrap effort for new VSI metadata service. Contributed to architecting and prototyping service, including building API specification to match customer requirements, as well as being hyper-performant, scalable, and extremely secure and stable (FedRAMP). Required interfacing with many development teams, management, and customers directly, ensuring their and our requirements and deadlines were being met. Mentored junior engineers to department best-practices and standards
- Technical lead for regional-scale cloud health and monitoring solution for IBM Public Cloud. Architected pipeline to configure service externally without code changes or downtime, allowing dynamic health input metrics, as well as dynamic targets for publushing results. Responsible for design, development, and deployment from concept to production

- *Staff Software Engineer* *(Band 7)* — *Aug 2017- Jan 2020*

- Core engineer of IBM Public Cloud API, designed to be a highly scalable, performance-driven, RESTful API, consumed by a UI, CLI, and customers directly to manage cloud compute resources. Go microservices deployed to Kubernetes, multiple custom Kubernetes controllers, and integration with numerous existing IBM services
- Core engineer of IBM NextGen Control Plane API, a datacenter level API that provided a RESTful interface on compute, networking, and storage resources. Created a custom Kubernetes API "extension" server, allowing these resources to be managed by custom Kubernetes controllers
- Technical lead for a new Kubernetes service to enforce global quota limits for customers. Developed generic middleware to allow all existing APIs to leverage this new service with minimal work required for integration
- Led refactor of effort of a large, monolithic, legacy API into a lightweight, container-based application to be deployed with Kubernetes. Required ground-up redesign, as well as adhering to existing API specification

- *Software Engineer* *(Band 6)* — *June 2015 - Aug 2017*

- Designed and developed automation using Python andAnsible for region-scale cloud deployments
- Deployed and managed Elastic components (ELK stack) for distributedobject storage offering
- Created system-level Python services for cloud operations such as health checks, log rotations, and PXE-booting VSIs
- Contributed to OpenStack Swift Object Storage development for integration into IBM Cloud

Skills

Languages

Go, Python, Java

Technologies/Frameworks/Tools

Kubernetes, Docker, Helm, gRPC/protobuf, redis, Ansible
ConcourceCI, ELK

Ideologies

Agile, TDD/BDD, microservices,
data-driven design

General

Linux/UNIX, git, CI/CD, UX, metrics,
documentation, mentoring

Projects (all code available on Github)

- **tarkov-charts.com (WIP)** - Web application displaying dynamic, interactive 3D charts of critical parameters for the online game, Escape From Tarkov. This allows players to determine the best in-game loadouts to choose for their situation (React/Go/MongoDB)
- **skillbased.io (WIP)** - Web application and developer API that allows players of pickup type sports to quickly create balanced teams for their sport, create custome sports, as well as create seasons and input statistics (React/Go/Postgres/Docker)
- **WoTBoT** - Bot that plays World of Tanks, a large scale MMO PVP tank game. Utilized A* pathfinding, computer vision, real-time event handling, and human interaction simulation to play without suspicion (Java)
- **Alan** - A.I. to play and compete in a “Poker Squares” competition run by Gettysburg University. Used a combination of MinMax and Monte Carlo algorithms to predict best moves based on an initial variable scoring system (Java)
- **Ambiled** - Dynamic RGB LED lighting for a desktop computer. Used computer vision and set LEDs based on the screen’s image in real-time. LEDs Driven by an Arduino, dynamically controlled with desktop application over serial port (Java)
- **3D-Tic-Tac-Toe** - Interactive game with computer player. Used MinMax algorithm with variable difficulty levels (Java)