Thien Tran

✉ ttran384@gatech.edu | 📞 (346)-204-9381 | 🌐 /thienlongtran.com | in /thienlongtran | ⛘ /thienlongtran

EDUCATION

University of New Orleans   August 2019 - December 2021
Bachelor of Science in Computer Science - GPA: 3.99/4.00   *New Orleans, LA*

SKILLS

Languages	Python, Java, Go, HTML, CSS, JavaScript, SQL
Technologies	Git, REST APIs, Unity, Jupyter Notebook
DevOps	Amazon Web Services (AWS), Terraform, GitHub Actions, Docker, Kubernetes
Certifications	AWS Solutions Architect - Associate, AWS Cloud Practitioner

EXPERIENCE

Venmo   May 2022 - Present
Software Engineer   *Austin, TX*

- Saved company $1,242,700 annually in compute and data transfer costs by identifying an unoptimized, continuously running metrics collection process and helping replace it with an interval aggregation algorithm using Python.
- Established visibility into cost areas and optimization possibilities for $14,400,000 worth of enterprise CI/CD jobs annually by integrating GitHub Actions Observability API with DataDog cost metrics using Python.
- Led a project to improve GitHub Actions observability by designing and developing a proprietary, scalable, high performance microservice using Python, Flask, Docker, AWS Elastic Kubernetes Service, AWS DynamoDB, and DataDog which delivers 14 key real-time metrics about active workflow jobs across Venmo's 1,200+ repositories.
- Enabled 24x7x365 reliability of GitHub Actions synthetic tests, recovering up to 25 failures daily, by developing a failure recovery script using Python and deploying it to AWS Lambda using Terraform.

USAA   May 2021 - July 2021
Software Engineer Intern   *Plano, TX*

- Reduced cluttering of a qTest archive by 84% and allowed for easier feature-based auditing by designing a new directory structure for publishing automated infrastructure test results that affected 70 projects.
- Enabled automatic AWS resource tagging on one parameter if not provided by a developer or optional manual tagging otherwise by modifying a custom Terraform provider utilized by 55 projects using GoLang.
- Decreased the cost of conducting network connectivity testing on AWS EC2 instances by 92.38% by developing a selection of 5 AWS Systems Manager (SSM) testing automations using Bash, Terraform, and GitLab CI/CD, saving the company $56,700 annually in lost wages and productivity.

PROJECTS

NBA Sportsbook Investment w/ Machine Learning | *Python, XGBoost*

- Generated 35.66% return on portfolio per month with a 1.42 Sharpe Ratio by developing an NBA Sports Betting Strategy on Over/Under (OU) Lines when using a configuration of 2.5% of fixed monthly bankroll per bet.
- Trained an XGBoost model to predict OU results with 56.37% accuracy, gaining an expected 7.67% ROI per bet.
- Automated betting by utilizing CloudBet APIs to get latest data and place bets right before the start of games.
- Enabled real-time trade monitoring by integrating DataDog with live scores and FanDuel's OU Line movements.

Stocks Simple Moving Average | *Python, Amazon Web Services*

- Built an AWS pipeline that computes the Simple Moving Average (SMA) of historical OHLC-type stocks.
- Created the cloud infrastructure using the AWS Python SDK (Boto3) to automatically initialize and connect two S3 buckets, two Lambda functions, one SNS topic, and one DynamoDB NoSQL database table.
- Decreased the time it takes to acquire the SMA of an input file by 99.87% compared to manual calculation.

Warframe Inventory Market Info | *Python, OpenCV, PyTesseract*

- Developed a program that automatically gathers 4 different economic attributes about users' in-game Warframe inventory items, saving users about 52 seconds of work per item page compared to manual calculation.
- Generated a list of users' inventory items using OpenCV to isolate item names from the inventory-screen image by thresholding the text colors, and using PyTesseract to read and save the remaining text.
- Enabled better investment decisions and comparisons by collecting the average currency price of the 10 current cheapest live web market value sell-orders using the warframe.market API for each item in users' inventory.