

Jason Traub

<https://www.linkedin.com/in/jtraub/>

U.S. Citizen

Summary:

I am a software developer and consultant, specializing in web development, DevOps, and automation. I am extremely proficient in Python, and an expert consultant and open-source contributor to the Salt automation framework. As a DevOps practitioner, I have experience with many other technologies and tools of the trade. My experience developing and deploying web applications includes the full stack from frontend HTML/CSS/JavaScript to backend with Python or Node.js, databases such as PostgreSQL and MySQL with raw SQL and/or ORMs, tools and frameworks such as React, Tailwind, Bootstrap, and HTMX, and other technologies such as Apache2, NGINX, and more.

Programming Languages:

• Python	10 years
• JavaScript	5 years
• HTML5	5 years
• CSS3	5 years
• Node.js	4 years
• SQL	4 years
• C	3 years
• Bash	2 years
• Ruby	1 year

Education:

- ❖ Bachelor of Science in Electrical Engineering @ University of Florida
 - *Magna Cum Laude (high honors)*
- ❖ Bachelor of Science in Chemical Engineering @ University of Florida
 - *Cum Laude (honors)*
 - Undergraduate Research Thesis
 - [*Vacuum Tube Amplifier Modeling With Dynamic Convolution*](#)
 - Vice President of UF Audio Engineering Society
 - Founder and President of UF Engineering Entrepreneurs
- ❖ Olympic Heights Community High School - Boca Raton, FL
 - National Honor Society

Projects:

- ❖ PTC automation platform revitalization (March '24 – May '24)
 - Identified and opened Salt pillar cache [bug report](#), reviewed PR
 - Wrote Python program to record salt master event frequency over time, led to discovery of auth event storms which hampered performance
 - Advised on salt automation best practices, demonstrated performance benefits, and refactored customer production code
 - Streamlined python web server integration for more effective salt monitoring and reporting with slack notifications.
 - Improved client's CI/CD by establishing and advising on frameworks for code linting and unit testing
- ❖ renderfarms.io fullstack and integration (August '23 – February '24)
 - Implemented design from Figma with HTML, CSS, and Tailwind
 - Frontend interactivity implemented with JavaScript and HTMX as requested
 - Backend developed with Django and a PostgreSQL database
 - API developed with Django-ninja
 - Deployed with Daphne ASGI server
- ❖ Fleet Monitor @ Terminal Labs (August '23)
 - Developed frontend and backend to monitor fleet of KeyShot managers and workers
 - HTML/CSS+Bootstrap frontend, Flask (Python) backend
- ❖ ShelfGenie website support (May '23 - ongoing)
 - Edit frontend Svelte (JavaScript) codebase in azure repository
 - Configure Strapi (Node.js) backend and test deployments on Heroku
- ❖ Authored [bits](#), a Python library for Bitcoin (July '22 – ongoing)
 - Developed Python program to generate Bitcoin keys, addresses, sign transactions, and more
 - Built from scratch - uses no 3rd party dependencies
 - Intuitive command line interface
- ❖ Chef to Salt conversion for HPE (March '23 - April '23)
 - Analyze chef recipes and ruby code and convert them salt state equivalents, reduced complexity where applicable
- ❖ Deployed, developed, and consulted on Salt on behalf of VMware @ US Bank (August '21 - March '24)
 - Developed salt state / orchestration code to deploy a full enterprise installation of Salt, meeting unique internal requirements, in all 3 environments (Dev / UAT / Prod)

- Consulted with various teams instructing and helping to write salt state code to perform their various business and engineering tasks
- Advise, test, and coordinate with US Bank and VMware on performance tuning configurations
- ❖ Salt and backend code repository assessment, ongoing support @ Bacon Unlimited (December '21 - Jan '22)
 - Assess code repository, make comments, suggestions, etc.
 - Developed salt states and code and reviewed PRs
 - Assisted in testing salt states across the code repository, finding and fixing bugs, etc.
- ❖ Python program to automatically generate Salt states that configure Windows LGPO @ Bacon Unlimited (February '22 - June '22)
 - Developed Python program to examine Policy Definitions XML file and generate an appropriate salt state YAML file to configure each, respectively
 - Developed test suite with PyTest to test the states produced by the aforementioned Python program
 - Generated states are publicly available [here](#)
- ❖ Developed software and advised on Salt @ IBM (August '19 - January '21)
 - Assist in deployment and development of high availability solution for SaltStack Enterprise
 - Developed software and provided code review
 - Custom salt module development
 - Configured and deployed proxy minions interfacing with thousands of network devices
 - Tested native minion and deltaproxy for network devices
 - Advised and developed scripts to assist in discovering performance deficiencies
 - Advised on Python programming best practices
 - Created wiki documentation for advanced topics in Salt
 - Instructed and developed [Salt NetOps lab](#)
- ❖ Assistant instructor for salt lab @ SaltConf '19 (Nov '19)
 - Developed lab procedure and content involving grains, beacons, etc.
 - Assisted attendees in completing lab
- ❖ Developed content for and instructed Salt and SaltStack Enterprise Courses (February '21)
 - Developed lab content and documentation on Salt
 - Particular focus on network devices

- Deployed lab setup for students
- Co-instructed course
- ❖ TACC 1 million salt minions experiment (June '21 - July '21)
 - Goal of deploying 1 million salt minion on TACC supercomputer
 - Developed salt code to configure many salt minion on a single VM
 - Research and experimentation on scaling Dask workers
- ❖ Developed Python program to perform MySQL to PostgreSQL data migration for SportsJaw @ Terminal Labs (January '21 - March '21)
 - Included command line interface with various options, to facilitate long-running data migration
 - Leveraged SQLAlchemy, to store data from MySQL as pickle objects and use to then save data in PostgreSQL
- ❖ Developed and tested code for SportsJaw @ Terminal Labs (January '20 - August '21)
 - Found and fixed bugs in API code
 - Developed unit tests with Pytest
 - Modified Flask database models, perform migrations
- ❖ Frontend interactive map development and integration for MuchADU website (May '21 - June '21)
 - Developed HTML / CSS / JavaScript to embed and integrate leaflet map into MuchADU's Wordpress website
- ❖ Developed DensifyLA website (May '19 – July '20)
 - Developed for a client architect
 - User would enter an address in Los Angeles county and download a feasibility report, for assessing a development at that address
 - Interfaced with a backend report generator program developed by the architect
 - Frontend map interface developed with leaflet.js
 - Backend developed with Flask
- ❖ Developed nauticalminds.com (December '18 – ongoing)
 - Website to showcase my band's self-titled music EP, Nautical Minds
 - Single page app included audio player built with React
 - Entirely static site, currently deployed via Netlify
- ❖ Crew Dragon Audio Board Acceptance Test Procedure @ SpaceX (March '18 - Dec '18)
 - Developed Test Rack, Printed Circuit Board, Software, and Procedure for Acceptance Testing of Crew Dragon Audio Board

- ❖ Designed and developed printed circuit board to interface with memristive neural network prototype @ Rain Neuromorphics (August '17 - Feb '18)
 - Designed printed circuit board with Eagle
 - Programmed Arduino board in C and interfaced with printed circuit board to send input signal, and measure output
 - Instrumental in achieving YCombinator investment
 - ❖ Prototyped spiking neuron circuit @ Rain Neuromorphics (June '17)
 - Ran LTspice simulations
 - Used breadboard and interfaced with memristive neuromorphic hardware
-

Hardware:

- ❖ Audio Board Test Rack and PCB for Crew Dragon @ SpaceX
 - ❖ Designed PCB for prototyping “neuromorphic” hardware @ Rain Neuromorphics
 - ❖ Led Ibanez Tube Screamer circuit build with University of Florida Audio Engineering Society
 - ❖ Led Subwoofer build with University of Florida Audio Engineering Society
 - ❖ Analog / Digital FM synthesizer for Electrical Engineering Senior Design Project
-

Tools and Applications

Web Development:

- HTML5
- CSS3
- JavaScript
- HTMX
- React
- Tailwind
- Bootstrap
- SemanticUI
- Python
- Node.js
- Django

Python libraries:

- Django
- Flask
- FastAPI
- Salt
- Numpy
- Napalm
- Netmiko
- Pytest
- Boto3

Databases:

- PostgreSQL
- MySQL
- MongoDB
- Redis
- SQLite
- SQLAlchemy

Servers:

- Apache2
- NGINX
- Uvicorn

- Flask
- FastAPI
- NPM

Text Editors:

- Vim
- Nano
- Visual Studio Code

Misc Tools:

- Lastpass
- LetsEncrypt (SSL/TLS)
- Certbot
- 7-zip

Command line:

- tar
- Htop
- Curl
- Wget
- Nmap
- awk

Amazon Web Services:

- S3
- EKS
- EC2
- AWS CLI
- Route 53
- Lambda
- IAM
- CloudFormation
- Control Tower
- SSO
- Organizations
- Active Directory
- Elastic Beanstalk

DevOps:

- Salt
- SaltStack
- SaltStack Enterprise
- SaltStack Config
- Aria Automation Config
- Docker
- Redis
- Terraform
- Kubernetes
- Netbox
- Hashicorp Vault

Operating Systems:

- Linux
- Windows
- MacOS
- Alpine
- Ubuntu
- Debian
- RedHat
- CentOS
- RockyLinux
- AlmaLinux

Image Editors:

- GIMP
- Inkscape

Templates/Preprocessors

- Sphinx
- Jinja2
- Sass
- Less
- Webpack
- PostCSS

- Daphne
- waitress

Cloud Providers:

- AWS
- Azure
- GCP
- DigitalOcean
- Heroku

Conference Software:

- Google Meet
- Zoom
- Discord
- Webex
- Skype

Revision Control:

- GitHub
- GitLab
- Git

CI/CD

- GitHub actions
- GitLab CI/CD
- Travis CI
- Circle CI
- Jenkins
- DataDog
- Splunk

VMware

- ESXI
- NSX-T
- VMware Horizon

Package Managers

- yum
- apt
- snap
- dpkg
- pip
- poetry
- conda