Alexander Jansing

https://github.com/apjansing

Email : alexander.jansing@gmail.com
Mobile : +1-315-601-8991

Highlights

- 4+ years in data science and software engineering industry
- Active DoD TS/SCI
- M.S. in Computer Science with a focus in Software Engineering and Mathematics
- B.S. in Applied Mathematics
- Operating Systems most comfortable with Linux, MacOS
- Proficient with Java, Python, Bash
- Comfortable with Matlab, R, LaTeX, Dart
- Familiar with HTML, CSS, Javascript, Scala, Groovy, OCaml
- Technologies comfortable with Git, Nifi, Hadoop, Spark, AWS, Maven, Docker, Docker-compose, ArgoCD, Kubernetes, MongoDB, Accumulo, Postgres, Jira, Confluence

Experience

- **Booz Allen Hamilton** Rome, NY
  Software Engineer, Implementation Specialist *Oct 2018 - Present*
  - **VI2E - Pipeline Delivery**: Using Concourse, Docker, ArgoCD, Kubernetes, Python, and Bash scripts to create CI/CD pipelines for the Air Force's VI2E program.
  - **Swift**: Used Concourse, Sonarqube, Docker, Python, and Bash scripts to create CI/CD pipelines for the United States Air Force Research Laboritory.
- **Lockheed Martin** Liverpool, NY
  Software Engineer, Asc *Mar 2018 - Sept 2018*
  - **SEWIP and Q-53 BEMA**: Developed analytics for noise reduction and identification of Modulation techniques using technologies like Matlab and Tensorflow.
- **Booz Allen Hamilton** Rome, NY
  Data Scientist, Junior – Computer Science *Jan 2016 - Mar 2018*
  - **Active Insights**: Designed a data lake based using an Accumulo backend and OrientdDB for provenance tracking. ETL processes were performed with Apache Nifi.
- **SUNY Polytechnic** Utica, NY
  Graduate Assistant *Aug 2015 - Jan 2016*
  - **Finite Mathematics**: Graded homework, held office hours, and designed grading schemes for Finite Mathematics.

Projects

- **Graduate School Final Project — Open House Route Planner**: A project that allows the user to provide a series of calendar events and returns several routes one could take to visit as many open houses as possible. Project was written in Python, uses the *Esri API* for geocoding and route finding, *MongoDB* for caching of locations, and *Docker-compose* for infrastructure. Idea originally worked on during hack Upstate (see Projects section).
- **Hack Upstate XI — Open House Route Planner***: Javascript, Docker, MongoDB, Esri API *Grand Prize* and *Esri API Prize*
  A minimum viable product for a hackathon that took lat-long locations and found the optimal route to visit all points using the Esri API. Project was primarily written in Javascript, uses the *Esri API* for geocoding and route finding, and *Docker-compose* for infrastructure. Time-boxing of open houses and travel times were touched upon, but not fully implemented.
  * No code survived between hackathon and graduate school.
- **Hack Mohawk Valley — Move Helper**: HTML, CSS, Javascript, MongoDB *Best Use of Open Data*
  An app to help you get the info you need to move into a new area. Data returned includes sources from Syracuse Open Data crime, lead, and code violations.
- **Hack Upstate X — Buffalo Crime Data**: Python, MongoDB *Best Use of Open Data*
  Hackathon project to discover crimes that occurred a specified distance away from police cameras and plotten them to show clusters of crimes in the city of Buffalo.

Education

- **SUNY Polytechnic Institute** Utica, NY
  Master of Science in Computer Science; GPA: 3.64 *Aug. 2015 – May. 2019*
  - **Final Project: Open House Route Planner**: See Projects Section.
  - **Relevant Classwork**: Quantum Computing, AI Topic: Data Science, Machine Learning, Formal Methods, Big Data Platforms, Numerical Diff Equations
- **SUNY Oswego** Oswego, NY
  Bachelor of Science in Applied Mathematics *Aug. 2012 – May. 2015*

Publications

- **Medium**

  April 2020 - Present

  - **Writing a Custom Concourse Resource — (Four Parts)**:

    1. — Overview
       An overview of all the components involved in writing your own Concourse resource. This and the following articles were inspired by the process it took to write my first resource as the tutorials available were not sufficient.
    2. — the check
       The tutorial goes over how the check component detects new versions at an endpoint and how to use that in the in component of the resource.
    3. — the in
       The tutorial goes over how the in uses the output of the check component to fetch data from an endpoint, save that data, what needs to be output, and how it interacts with the out.
    4. — the out
       The tutorial goes over how the out component delivers data to an endpoint and how it runs the in component to fetch the data the out just wrote.

Programming Skills

- **Languages**: Java, Python, Matlab, R, Scala, Groovy, OCaml, Bash, LaTeX, HTML, CSS, Javascript

- **Technologies**: Git, Nifi, Hadoop, Spark, Concourse, Maven, Docker, Docker-compose, MongoDB, Accumulo, Postgres, OrientDB, TitanDB, AWS, Jira, Confluence

- **Operating Systems**: Linux, MacOS