Kaleb Smart

Software Engineer

☎ (256) 506-8537 | ✉ kalebsma@gmail.com | yeyande | in kaleb-smart

Education

University of Alabama in Huntsville *Huntsville, AL*

Bachelor of Science in Computer Science, GPA: 3.7/4.0 *May 2016*

Skills

Languages	Python, Groovy, JavaScript, C++
Frameworks	Pytest, Behave, Jenkins, Angular, Flask, Docker, Packer, Laravel, Ansible, OpenStack, Artifactory, Jinja
SAFe Certifications	Practitioner, DevOps Practitioner, Scrum Master, Advanced Scrum Master

Experience

ADTRAN *Huntsville, AL*

DVT Engineer *June 2016 - May 2020*

- Architect and develop software for developers to work on customer-oriented deployments
- Serve on DevOps governance team to architect a company-wide vision of Continuous Integration/Continuous Deployment pipeline
- Operate as Scrum Master for a system team to drive the development of architectural runway and Developer Experience tooling
- Lead continuous improvement efforts within team via Retrospectives and Inspect and Adapt sessions
- Coordinate with other Scrum Masters to ensure cross-team dependencies and milestones were met and on track
- Design and manage lab networks for our products
- Interface with developers in Canada, Germany, and India to coordinate new product development efforts

ADTRAN *Huntsville, AL*

Co-op Engineer *June 2014 - December 2015*

- Develop test aggregation and hardware resource management software for Continuous Integration pipeline in the enterprise network element product line
- Provide Linux security best practices and bug-fixes for cloud-based WiFi solution

Projects

Hackathons

- Reduced build time of SDX-6210 software by 70% by consuming infrequently changing code as binary packages
- Containerized services running on SDX-6210 hardware to lay groundwork towards network feature virtualization for the product
- Created proof of concept for using Alpine packages to maintain dependencies and build infrastructure for internal packages
- Automated the population of CVEs and known software issues published in release notes and product security document
- Deployed a Github code review tool to keep track of requested changes
- Worked on proof of concept Github bot to automatically create pull requests to integrate package dependencies to provide pre-commit integration on dependency chains

SDX Aggregation Switches

April 2020 - May 2020

- Architected pipeline modifications to test common feature sets on virtualized aggregation switch from on a common software image and the fan out procedure for product-specific acceptance tests on feature sets

Total Access 5000

December 2019 - May 2020

- Architected layout and methodologies of testing framework for TA-5000 GPON OLT line cards
- Developed automated provisioning of test assets for a given testbed, which included SIP and TFTP servers with configuration options provided from the testbed definition
- Redesigned lab network to allow for strict layer 3 isolation between testbeds in order to prevent network outages from misbehaving software. This also resulted in a separation of responsibilities for individual routers
- Designed testbed configuration restoration process to ensure testbeds were in a known default state prior to beginning tests

Release Notes and Security Documentation

November 2019 - December 2019

- Designed generic templating software for automatically generating release notes and product security documentation for any product
- Augmented release note generation to populate new, longstanding, and fixed software issues from ticketing system
- Processed internal security scans and tests to populate product security document with known security vulnerabilities

SDX-6210

March 2017 - December 2019

- Integrated the SDX-6210 product into existing Continuous Integration (CI) pipeline to include tests already written for existing product features
- Containerized cloud-based network element controller software to decrease software upgrade time by a factor of 6
- Developed system level verification pipeline and test framework to validate the SDX-6210 and the cloud-based controller software a against customer requirements prior to software release
- Developed a CI pipeline information radiator and value stream map to visualize how software was flowing through our pipelines, including current state of each stage, delay of code flowing through the system, and the likelihood that the code would fail in each stage, to help identify and address code flow issues within our solution
- Organized community/task force for collective ownership and troubleshooting of CI pipeline failures
- Designed automated ticket creation and triage process for CI pipeline failures to collect granular metrics and direct the goal of the CI pipeline task force each iteration
- Developed deterministic automated build procedure and integration for acquired and newly developed software
- Designed high level API library to interface with traffic generator hardware for use in CI testing
- Developed automated provisioning of test assets for a given testbed, which included DHCP, TFTP, RADIUS, and TACACS servers with configuration options provided from the testbed definition
- Created test asset diagnostics tooling which would dump application state and logs to quickly identify the cause of test failures
- Designed service to provision ONU management software on the SDX-6210 hardware upon contact from the network element controller
- Created self-checking startup procedure for basic FPGA and Broadcom SOC traffic forwarding
- Designed replicatable hardware configurations based around customer deployments of our products, allowing our entire acceptance test suite to be ran on any of our testbeds, and reducing the testbed bringup time 8x
- Created automatically deployed and populated DNS docker containers triggered off any changes of the testbed definitions
- Deployed a network of ONOS controlled whitebox switches to validate solution interoperability with an ONOS managed network
- Deployed Nagios monitoring software to collect health metrics on testbeds and their infrastructure

MOSAIC OS

June 2016 - February 2017

- Developed CI pipeline to allow for feature tests to be consumed via a product’s capabilities along with generic high level libraries to enable developers to write product-agnostic feature level tests.

Skynet and Hydra

June 2016 - March 2017

- Developed testbed inventory and reservation service which was used in CI pipelines, developers, and product stakeholders to visualize and control the priority, allocation, and utilization of testbed resources
- Designed test results aggregation software and trend visualization for developers and product stakeholders to understand reliability of a software build or test suite
- Created high level groovy and python APIs to allow developers to interact with these services in the same way as the CI pipelines