Junior Eluhu

CURRENT WORK

GITHUB /MICROSOFT

Junior is currently a Senior Software Engineer in the GitHub Enterprise Deployment Team part of GHES and under Microsoft CoreAI - Platform and Tool Group. Junior builds tools and services that manage the deployment of GitHub Enterprise for premium external customers. Those tools and systems are hosted on Azure, AWS and GCP. All of our code base is in GoLang, and Ruby. We also have some codes in bash soon-to-be deprecated. Unfortunately, because GHES mostly runs on-premises, we have a lot of codes in bash that manage bootstrapping, clustering, backup, and other basic computing utilities.
Junior works in rotation to push ¼ GHES Feature Version Releases per year as well as several patches every 2-3 weeks to address security and minor updates.
Because Microsoft's primary priority is Security, Junior closely works with the Security Team to address Bounty and other vulnerabilities weekly.
Junior is also responsible for creating the complete Terraform templates to run GHES in the cloud (AWS & Azure). These templates support Single Instance, HA, Geo-replica HA and Cluster. Those templates should be released to the public soon.

PREVIOUS WORK

MEDIAVINE

Junior worked as a Senior Site Reliability Engineer. He was responsible for migrating key applications from Heroku to AWS. AdTech Applications have services that require to be highly available so they can continue to process millions of advertisement bid transactions uninterrupted. Being able to scale and recover containers quickly while using micro-services is crucial. So, cloud services such as ECS, Fargate, Lambda and other serverless tools were heavily used to have high reliability and keep the billing cost down.
One other important application that Junior migrated was **Dashboard**. A reporting tool where Publishers who subscribed their websites to Mediavine, could see their data in real-time to help them make informed decisions on turning more profit.
Junior also spent a lot of time training and documenting a lot of the above processes as well as assisting the Support Team with high-severity issues.

AMAZON WEB SERVICES

Junior joined Amazon originally to work with the EC2 Nitro–Instance Quality Team however due to his prior release deployment experiences, he was then moved to the EC2 Deployment Services Team (under the EC2 Release Safety Team) where he built internal deployment tools for EC2.

He later moved to Paris, France to help build AWS Datacenters across Europe and implement EC2 Automatic Region Build. This project included other smaller projects such as Gunpowder host deployment and automatic hosts remediation. Gunpowder hosts were based on EC2 Nitro arm hardware architecture that enables compute instances such as C5, M5, C6g and others to exist. This technology did solidify AWS's leadership position in terms of hardware appliance virtualization.

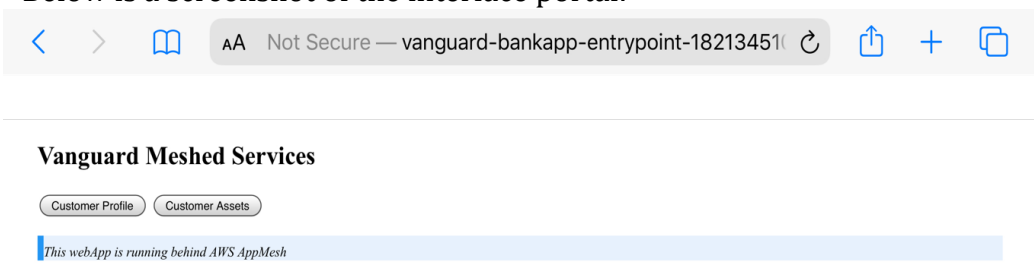
Below are some of his work at Amazon Web Services:

Professional Services - AMAZON HR – COVID-19 Tools

- Effort: AWS Account Migration | Attestation Notification API
- Personal Contribution: Sole Owner of the Migration Effort and AWS Kendra Effort
- Type of Work: CIA, DevOps and Cloud Support

Professional Services - Vanguard

- Effort: AppMesh Proof of Concept
- Personal Contribution: Design | Build | Demonstration of AppMesh at Client Request
- Type of Work: AppMesh for their App Gateway Services
- Details: The customer requested a Service Mesh Solution for some of their financial internal services to be regulated. They also wanted their Application Administrators to have control over (who/when/where/what/how) API calls were made. Their other need was to have their Apigee integrated into the process. Junior built a proof of concept to demonstrate how services were interacting before AppMesh and After AppMesh. Amazon Xray was also used to show the interaction between services.
  However, because of financial regulations, Junior had to write services from scratch using Node.js (The PoC was supposed to borrow a few of Vanguard Services to showcase AppMesh). The customer approved AppMesh's addition to their AWS Portfolio as a result of the PoC.
  Technology used: ECS/Fargate, AppMesh, Route53, CloudFormation, Certificate Manager, ALB, API Gateway, Xray, EC2, ServiceMap, Node.js, Python, HTML
  Below is a screenshot of the interface portal:



Professional Services – Joint Strike Force (Public Sector - DoD)

- Effort: Built DevSecOps tools for the F35 Effort
- Personal Contribution: Automation | Orchestration | Training
- Type of Work: DevSecOps
- Details: Junior implemented a lot of solutions for this account. This is a Public Sector Customer who wanted to modernize all of the DevSecOps Tools for all of its Defense Contractors as well as its Allied Countries' Defense Contractors. The first task was to create a pipeline for each of their most used programming language. Junior wrote an orchestration tool in bash that created the Jenkins pipelines and launched them for each of the 4 HelloWorldApp pipelines (.NET, C++, Node.js, Java). For this task, Junior also presented a PoC of the DevOps Tools in EKS for the Defense Contractors. The tools are (Jenkins + replica Nodes, Nexus and, Atlassian Tools: BitBucket, Jira, Confluence). Those tools were also deployed in the PKS (Pivotal Container Service) Cluster of the Customer. Many external tools were used in this project because AWS's offering in the Air-Gap region is limited.
  Junior also created a Transit Gateway Monitoring system that would detect who is tampering with the Client Transit Gateway. The tool would alert the Network Administrators and temporarily remove admin-elevation of the offender using IAM then put him in Permission-Jail. The system heavily relied on CloudWatch Event Rule and Lambda/Boto3 API calls (for write operations that Event Filter Patterns couldn't provide), SNS, etc.
  Ansible Tower in EKS was also deployed for the customer by Junior and he wrote several Ansible playbooks to deploy the Shared Delivery Network Resources for the Client using the CloudFormation Plugin.
  Moreover, Junior leveraged S3 as a Private Yum Repo for the Customer. Last but not least, Junior also documented all his work with how-to(s) and runbook(s) using the Customer Confluence tool.The technology used: Atlassian Tools, Jenkins_slave nodes, Nexus, Yum, RPM - gcc, Python, Java, C++, Vault, PKS, Ansible and, AWS Tools in IL5.

Professional Services - Nike AWS CoInnovation (ML Benchmarking)

- Effort: Automation & Orchestration
- Personal Contribution: Step Function Orchestration for Nike Benchmarking Machine Learning Optimization / Several Documentations
- Type of Work: ML Optimization and Benchmarking Orchestration
- Details: Nike's co-innovation performance benchmarking project focuses on building a solution that can accept any Nike ML model, run performance benchmarks against it, and generate a report on how ML models perform in different hardware and configurations. It could also be used for optimizing ML models using Amazon SageMaker NEO for various device targets in parallel.

This project specifically focuses on performance benchmarking ML models on EC2 instances. Junior built the workflow (Amazon State Machine) that orchestrates the whole process to help Data Scientists make conscious and cost-effective decisions on which target device to use. Data Scientists will use an external facing API (API Gateway, SAM) to query ML model metrics. Technology used: Jira, Confluence, GitHub, Step Functions, SageMaker, API Gateway, Terraform, ECR, DynamoDB, EC2, IAM, Python, Lambda.

Europe Data Centers - CDG | FRA | MXP |ARN

- Timeline: 06/2017 - 01/2019
- Effort: Region Build - Data Center Production Work
- Personal Contribution: Physical Build, Automation, Orchestration, SnowBall Cluster Owner, Operation Support & Recruiting Effort
- Type of Work: Region Build, Automation and DataCenter Operations
- Details: AZ Team - CDG | FRA | ARN | MXP
  Because of the confidentiality nature of the work at the Data Centers. Much cannot be said about Junior's work specifics. However, Junior moved to Paris at the early stage of the Region build and did in general the following work:
  Technology used: AWS Infrastructure Team Confidential (NDA)
  Physically installing racks from empty rooms to going live-production
  Euclid Brick Build | NTP Build | Built Corporate fabric (Wi-Fi survey-association-configuration, corporate tools, IP phone, printing, etc.) - No one was allowed in the Data Center except AWS Blue/Tiger Badge employees (Not even Amazon IT). Built EC2 Fabric and worked on EC2 Bootstrap
  Built AWS Production Fabric (a build involves installing servers, doing the networking, running automation scripts, writing automation scripts, troubleshooting, Assisting various Networking teams in their remote tasks, etc.)
  Decommissioned hosts and Network Appliances,
  SME and Primary owner of CDG Snowball - Fully built the Snowball Room from scratch,
  Installing and configuring DWDM and Dark Fiber | A lot of Fiber cabling. A lot,
  Procurements alongside with AWS InfraOps - Logistics,
  Participated in Game Days - Simulate Disaster Events Then Recover - SEV1 & SEV2,
  Rack down Drill Event - Mostly for TOR Replacement,
  Large Scale Upgrade & Maintenance Events - Power and Thermo,
  Did support operation work via on-call shift - worked daily Tickets,
  A lot of interviews for recruiting,
  Over 100hours of training,
  EC2/PROD Fabric Lab Setup,
  Worked in other Europe Data Centers as needed - FRA, ARN, Remote (MXP. LHR and DUB).

Technology used: BuilderTools, Apollo, DJS, DevDesktop, Carnaval, Metrics/ Graphs, Beachcomber, Boiler, Koji, EDS, Pipeline, Odin, Brazil, Octane, code.amazon, Infrastructure Provisioning, Mjollnir, Admiral, etc.

EC2-EDS-REX - Release Deployment Team

- Timeline: 12/2015 - 06/2017
- Effort: EC2 Package Deployments
- Personal Contribution: Python Automation and Orchestration
- Type of Work: Automation and Support
- Details: EC2 Deployment Services - Release Excellence/Experience.
  This team was renamed as EC2 Release Safety under EC2 Nitro. As a Backend EC2 Systems Engineer, Junior essentially worked on deploying software RPM packages/artefacts from all AWS/EC2 internal teams to their droplets (physical EC2 servers) targets worldwide; meaning in all datacentres across the globe. The whole deployment of packages took 3 weeks to deploy in millions of servers in all AWS regions. The Packages were deployed using many Backend Tools that Junior helped develop. Mainly the Rex Deployment Tool. The overall process was as such: the client submits their RPM packages then Junior's Tools will run a lot of tests on the packages (functional tests, unit tests, integration tests, regression tests, etc.) then the package would be approved for deployment. The package would go through the deployment train and land at its intended targets within 3 weeks or less. The package version would be checked for package diversity. The package would then be promoted to the EC2 hosts to be active after the next server reboot.
  This whole process happened in a pre-production environment first then would go to the production environment.
  Junior also participated in operation work (25% of the time) to support the tools built via on-call for one week every two months. 75% of Junior's work in Rex Team was software development-based.

Certifications

- Cybersecurity: Technology, Application and Policy - MIT Professional Education
- Extreme Networks - ENS - Extreme Networks Specialist
- Extreme Networks - ENA - Extreme Networks Associate
- CompTIA - Network+
- GitHub Actions
- AWS InfraOps - DevOps Engineering on AWS

Past Organizations

- Eta Kappa Nu (Electrical and Computer Engineering Honorary Society)
- IEEE (Professional)
- Underwater Robotics Club (NC State Group - Competition Level)

Areas of Specialization

DevOps, Site Reliability Engineering (SRE), Software Deployment and Release Engineering, Infrastructure Architecture

Skills

- Programming Languages: Python, GoLang, Bash, Ruby
- Infrastructure as Code: Terraform, CloudFormation, CDK
- CI/CD & Automation: Ansible, Jenkins, CircleCI, Azure DevOps, Github Actions, etc.
- Messaging & Streaming: Kafka, SQS, SNS
- Monitoring & Observability: Grafana, Prometheus, CloudWatch, Splunk, CloudTrail, Datadog, Sentry, PagerDuty, ServiceNow
- Cloud Platforms: AWS, Azure, GCP
- Other Tools: ECS, Fargate, Lambda, AppMesh

EXTRA STUff

Infrastructure diagrams, technical documents and Repositories could be shared if requested => https://github.com/sweluhu/eluhujStuff