Tyler Boyd
Silver Spring, MD, United States | Mobile: (202)578-7828 | www.tylerboyd.net | LinkedIn

SUMMARY

Motivated Cloud Engineer with experience in designing & deploying scalable cloud solutions using Infrastructure as Code. My expertise in containerization includes efficiently Dockerizing applications on Linux servers, significantly reducing deployment time, & enhancing application reliability. Skilled in leveraging tools such as Terraform, Jenkins, Docker, GitHub, & AWS services such as ECS, ECR, Route 53, VPC, RDS, DynamoDB, CloudFormation, & Lambda. Solid experience in implementing & managing CI/CD pipelines & an in-depth understanding of all aspects of the Software Development Lifecycle. Seeking to leverage my skills & experience to contribute to an AWS Cloud Engineer role.

PROFESSIONAL EXPERIENCE

Data Service Group|*Cloud Operations Engineer*|Bear, DE|Sept 2023 - Current

- **Container Orchestration:** Led the successful migration of an on-premise application to Amazon Elastic Container Service (ECS). Delivered a presentation on ECS, showcasing its capabilities through a sample demonstration.
- **Enhanced Website Performance:** Regularly configured on-premises load balancers using HAProxy & NGINX, resulting in a 25% improvement in website traffic management & a smoother interaction between frontend & backend services.
- **Infrastructure Optimization with Terraform:** Achieved a 30% reduction in AWS infrastructure costs by utilizing Infracost to identify & eliminate redundancies, while enhancing automation.
- **CI/CD:** Collaborated with developers to optimize application code & reduce dependencies. Managed Docker images for microservices, supporting live website updates, order history functionality, & program registration. Maintained repositories on GitHub, ECR, & DockerHub.
- **AWS Environment Provisioning:** Leveraged Terraform-based infrastructure-as-code solutions to provision new environments on AWS (VPCs, subnets, NAT Gateways, ELB, etc.) in less than 10 minutes, resulting in a ~30% increase in deployment frequency.
- **Bash Scripting:** Reduced lead time to change by 45% through Bash scripting automation for testing & deployment processes. Manually configured EC2 servers to install Python applications & dependencies, including load balancers.
- **Project Visualization & Team Collaboration:** Worked closely with cloud architects to design project frameworks & visuals using tools like draw.io. Provided constructive feedback, enhancing project clarity & effectiveness.

DR Financial Services|*Junior Cloud Tech*|Houston, TX| January 2022 - June 2023

- **Optimized Application Deployment:** Successfully deployed Python & Node.js applications on EC2 servers, containerized them with Docker, & created Dockerfiles for streamlined installation & dependency management, reducing deployment time by 20%.
- **Containerization:** Effectively utilized Docker Compose to deploy & manage containers, including WordPress & MariaDB, supporting the company’s website & database maintenance.
- **AWS Migration with Terraform:** Collaborated with infrastructure & configuration teams to migrate a multi-tier app to AWS. Secured the environment with Terraform, utilizing VPCs, Security Groups, & more, ensuring state management with an S3 backend.
- **Enhanced Team Collaboration:** Coordinated with development teams to manage Terraform code & Docker updates through GitHub & DockerHub, ensuring seamless branch management & version integration, which improved overall team efficiency & reduced integration conflicts.

LetBob Technologies|*Linux Administrator Intern*|Washington, DC|May 2021 - June 2022

- **Permissions & User Management:** Administered Linux services by managing file & directory permissions, ensuring proper user & group access through command-line tools.
- **Software Installation:** Automated software package installations & updates via Bash scripting, maintaining system security & efficiency.
- **System Monitoring & Inventory:** Conducted daily system monitoring, documenting server performance issues such as high CPU usage & storage capacity, & reporting findings to senior administrators.
- **Security Management:** Secured servers against unauthorized access by configuring SSH for remote access & regularly patching vulnerabilities.

TECHNICAL SKILLS

Technical Skills:

- **Infrastructure as Code (IaC):** Terraform
- **Linux:** Server SSH, Commands, Server Patching, Upgrading, & Updating
- **Cloud Technologies:** Amazon AWS (including S3 buckets), Terraform, Repository, Lambda
- **Containerization:** Docker, Docker Compose, Docker Swarm
- **Continuous Integration/Continuous Deployment (CI/CD):** Git, AWS, Jenkins, Agile
- **Scripting & Automation:** Bash Shell Script, LAMP Stack
- **Version Control:** Git, GitHub
- **Collaboration & Communication:** Problem Solver, Adaptable, Clear Communicator, Jira
- **Operating Systems:** Linux, Windows, MacOS, Unix, Virtual Machines, Database
- **Database:** MySQL

PROJECTS

Project 1: Built application Infrastructure on AWS. iCare Medical wanted to run its application with the entire infrastructure on AWS. My team's task was to build & manage the following infrastructure components: VPC, Subnets (public & private), ELB, Auto Scaling Group (CPU over usage, instance terminate), RDS, CloudWatch for app monitoring, SNS to alert us when our system is running (such as when new instances are created or terminated), Route 53 (Domain Name), EBS & S3 buckets for Instance generated log messages that needed storage, EFS service to work as a shared filesystem between the instances since they were using the same content. Needed to manage who could access the infrastructure through the console & to do that, we needed to configure access using the IAM (Identity & Access Management) service. Instances needed access to the S3 bucket for backups & logs, IAM roles were created. To ensure we can recreate all of this without manual effort, used CloudFormation service & Terraform to write the entire infrastructure as code.

Project 2: Hosted a secure static company information website. It had to be accessible with a domain name, needed to be available globally, needed the website to load fast, wanted a cost-effective but efficient solution, & also needed the website to be secured with HTTPS. Create an Amazon S3 bucket to hold static website files & an Amazon CloudFront distribution to serve the website globally. Utilized AWS services such as, Amazon Route 53 managed the domain name, & AWS Certificate Manager provided a valid SSL/TLS certificate. Utilized, Amazon S3, Amazon CloudFront, Amazon Route 53, & AWS Certificate Manager

Project 3: Built a User registration API with API Gateway, Lambda & DynamoDB. Was head in charge of a project to register employees in my company. Given that this project involves registering users and searching for various users, launching an entire server for this will be of great loss to the company. So my task here was to look for a possible solution that didn't involve launching a whole server, thereby reducing cost. After doing some research and discussing with the developers, I decided to go serverless. I utilized Serverless Solution Components (Frontend, Backend, & Data Transfer and Storage). Utilized AWS Lambda, AWS API Gateway, Amazon DynamoDB, A NoSQL database, Amazon S3, Amazon CloudFront, Amazon Route 53, AWS Certificate Manager.

EDUCATION

University of Maryland, College Park
Bachelor of Science in Computer Science

University of Maryland Baltimore County
Information Technology