# Marlon Mejia

**Download in** [PDF](resume.pdf) || [DOCX](resume.docx)

**Phone:** [1-631-480-7675](tel:1-631-480-7675) | **E-Mail:** marlon.junior.mejia@gmail.com | **LinkedIn:** [marlon-mejia](https://www.linkedin.com/in/marlon-mejia/) | **Github:** [marlon-mejia](https://github.com/MarlonJMejia)

## Skills

* **Tools**: Terraform, Ansible, Docker, Active Directory
* **CI/CD**: Jenkins, Github Actions, AWS CodePipeline
* **Operating Systems**: Linux (RedHat, Debian), Windows, Unix
* **Programming**: Bash, Python, Powershell
* **Monitoring**: Grafana, Splunk, Humio, Prometheus, Influxdb

## Certifications

* [RHCSA](https://rhtapps.redhat.com/verify?certId=220-057-368) - Apr 08, 2022
* [EX200 Red Hat Certified System Administrator](https://rhtapps.redhat.com/verify?certId=220-057-368) - Apr 08, 2022
* [AWS SAA-C03](https://www.credly.com/badges/838a30cd-0701-4069-b4be-68fe22d6962a) - March 31, 2023
* [Comptia Security+](https://www.credly.com/badges/136d58c4-24d3-4487-aad5-c51e120a3e7f) - November 02, 2020
* API Security Architect - Jan 20 2024
* [Comptia A+](https://www.credly.com/badges/89fca521-f3de-4c36-90f5-7552f9c4c26e) - May 22, 2020

# Bloomberg LP

**Datacenter Operations Engineer**  
*Nov 2020 - Present* *Fulltime*

* **Data Center Operations**:
  + Rack and Stack: Installed and configured servers and network equipment.
  + Decommissioning: Managed server and cable removal, data sanitization, and disposal.
* **Issue Diagnosis and Resolution**:
  + Address Layer 1 & 2 connectivity issues across 1000+ servers, switches, routers, and firewalls.
  + Resolve issues across operating systems, including Windows and Linux (Red Hat, Debian) to ensure consistent and reliable functionality.
* **Automation**:
  + Led a project to automate case opening and log gathering across multiple systems by utilizing **REST APIs** and **Python**.
  + Reduced operation time from over 20 minutes to just 30 seconds per task, saving approximately 1690 hours annually.
* **Legacy Modernization**: Contribute to the overhaul of outdated programs and documentation with **Python**, **Bash**, **Git**.
* **Containerization**: Develop over 50 **Dockerfiles** to containerize and facilitate consistent deployment and testing of **Python** and **Bash** .
* **Incident Management**: Utilize **Jira** to plan, track, support tickets, and manage incidents, ensuring efficient resolution.
* **Monitoring and Analysis**: Servers and Network Devices across datacenters, tracking disruptions, resource utilization, and power consumption using **Grafana**, **Splunk**, and **Humio**.
* **Cross-Team Collaboration**: Collaborate across multiple technical teams to deliver Agile-based projects, ensuring seamless communication and coordination across multiple Datacenter sites.
* **Data Center Optimization**: Execute regular audits of data center operations, identifying improvement opportunities and implementing strategies to enhance performance and reduce costs.
* **System Maintenance and Upgrades**: Perform routine maintenance, hardware upgrades, firmware updates, and patch management, to ensure peak system performance.

# NYI - New York Internet

**Datacenter Technician**  
*Jul 2020 - Nov 2020* *Fulltime*

* **Customer Support**: Provided remote technical support, including device configuration, troubleshooting, and optimization.
* **Network Monitoring**: Monitored over 1000 devices using **LogicMonitor, ConnectWise, and Meraki**. resolved outages and network issues.
* **Automation**: Automated Google Drive tasks with **Python** scripts using Drive **API**.
* **Hardware Management**: Installed and organized hardware, performed cabling and tested with Fluke equipment.
* **Customer Interaction**: Communicated with clients about services and provided performance tips.
* **Documentation**: Documented server setups and task methodologies for efficient handovers.

# Projects

### CI/CD Project for AWS and GitHub Pages

* **Objective**: Developed a robust CI/CD pipeline to automate the deployment of a static website hosted on AWS.
* **Technologies Used**:
  + **AWS Services**: Utilized S3 for object storage, CloudFront for content distribution, and Route 53 for domain and DNS management. Implemented SSL certificates using AWS Certificate Manager for enhanced security.
  + **Development**: Created content in Markdown for ease of editing and used pandoc to convert Markdown files into multiple formats such as PDF and DOCX.
  + **Automation**: Implemented CI/CD pipelines using CodePipeline and GitHub Actions to automate the deployment and testing processes, ensuring seamless updates and multi-format document generation.
* **Outcome**: Achieved a streamlined and automated workflow for static website deployment and maintenance, resulting in increased efficiency and reduced manual intervention.

### Cloud Proxy Server ([Diagram](https://github.com/MarlonJMejia/MarlonJMejia.github.io/blob/main/Reverse_Proxy_mermaid.md))

* **Objective**: Designed and automated the deployment of secure, scalable cloud infrastructure on Oracle Cloud to expose local resources.
* **Technologies Used**:
  + **Infrastructure as Code**: Automated the provisioning and management of cloud resources on Oracle Cloud with Terraform.
  + **Configuration Management**: Utilized Ansible to automate the setup and configuration of Wireguard VPN and NGINX on the provisioned infrastructure.
  + **Reverse Proxy**: Implemented a reverse proxy to securely route traffic to a Grafana local endpoint and a local website through a Wireguard connection between an OPNsense firewall and the OCI instance.
  + **Security**: Deployed CrowdSec on OPNsense to protect the reverse proxy, enhancing security and mitigating potential threats.
* **Outcome**: Established a robust, automated infrastructure that securely exposed local resources while enhancing performance and security. The solution reduced manual configuration efforts and improved the overall reliability and protection of the hosted services.

### Automated Provisioning with Proxmox, Terraform, and Ansible

* **Objective**: Streamlined the provisioning and configuration of LXC containers and VM instances on Proxmox to enhance infrastructure management and automation.
* **Technologies Used**:
  + **Provisioning**: Utilized Terraform to automate the creation of LXC containers and VM instances on Proxmox, enabling scalable and efficient infrastructure deployment.
  + **Configuration Management**: Applied Ansible for post-provisioning configuration and management, ensuring uniform setup and operational consistency.
  + **Backups**: Set up automated backups using **Kopia**, with infrastructure code securely stored in GitHub for version control and disaster recovery.
* **Outcome**: Achieved a highly automated and efficient infrastructure management process, significantly reducing manual intervention, enhancing configuration consistency, and ensuring reliable backup and recovery.