
Project Proposal: Job Scraper

Author: Daniel Bonney

Timeline: 14 Days (Proof of Concept)

Objective: To modernize 7+ years of AWS ADC systems engineering into a high-level, GitOps-driven Platform Engineering framework.

+1

---

1. Executive Summary

The Job Scraper is a technical artifact designed to demonstrate mastery of the modern cloud-native stack: **Terraform, Docker, Kubernetes (EKS), and Python**. This project transitions "Air-Gapped" security principles—honed during my tenure in Amazon Dedicated Cloud—into public cloud Infrastructure as Code (IaC) and automated data ingestion.

2. Core Pillars of Execution

- **Infrastructure (Terraform):** Modularized HCL to provision a secure VPC, EKS Cluster, and S3 backend for remote state management.
- **Containerization (Docker):** Security-hardened, multi-stage images housing application logic, leveraging my Master’s in Cybersecurity.
- **Orchestration (Kubernetes):** EKS-based CronJobs triggering ephemeral pods for high-throughput data scraping.
- **Logic (Python/Boto3):** Custom resilient scripts for data ingestion and S3 integration, fulfilling a career-long goal to master Python automation.

3. Strategic & Financial Value

- **FinOps Driven:** Implements a "Scale-to-Zero" architecture using AWS Spot Instances and ephemeral compute to minimize idle costs.
  Security First: Eliminates static credentials by using IAM OIDC (IRSA) to grant least-privilege access directly to Kubernetes Service Accounts, aligning with my experience as a Security Guardian.
- **Operational Rigor:** Automates complex workflows to prove that high-demand skillsets should be engineered, not manually managed.

4. Implementation Roadmap (14 Days)

Week 1: Infrastructure & Logic Foundations

- **Terraform:** Provisioning core VPC and EKS networking.
- **Python:** Developing the core scraping engine with error handling and S3 upload logic.
- **Local Testing:** Validating scraper performance in local Docker environments to bypass anti-bot detections.

Week 2: Orchestration & Deployment

- **Dockerization:** Hardening images and pushing to ECR.
- **EKS Deployment:** Configuring CronJobs and IAM OIDC roles for secure pod communication.
- **Optimization:** Tuning auto-scaling parameters to ensure $0.00 idle compute costs.

5. Alignment with Leadership Principles

This project serves as a definitive proof of my Git, Python, Docker, Terraform, and Kubernetes skills, demonstrating the ability to rapidly acquire new technical skills and deliver a functional, high-standard product ahead of expectations