Dimitrije Davidovic Vedda

Cloud / DevOps Engineer

New York, NY (open to relocation) | dimitrije.vedda@outlook.com | linkedin.com/in/dimitrije-vedda | github.com/Dimi-DV

SUMMARY

Junior cloud / DevOps engineer with hands-on AWS infrastructure experience and a production-style AIOps capstone built on AWS Bedrock AgentCore. Strong Terraform, Python, and Linux fundamentals from a focused, self-directed study sprint. Open to early-career cloud/DevOps and data roles; native English and Serbian.

EXPERIENCE

Il Mulino

June 2023 - May 2025

Operations Analyst | New York, NY

- Built a Python pipeline (BeautifulSoup, Selenium, sentence-transformers, scikit-learn, TF-IDF) standardizing dish-name nomenclature across franchise locations into a unified master sales sheet - enabling franchise-level financial analysis that was previously impossible on a per-location basis.
- Gathered each location's dish-naming conventions from regional and local managers to build the terminology map the standardization pipeline depended on; when managers were resistant, surfaced the scale of the inconsistency to the CEO, who drove the locations toward a common standard.
- Worked around the legacy Compeat POS's lack of an API: first prototyped GUI screen-mapping automation, then pivoted to a separate data-ingestion + standardization pipeline. Debugging discipline and pragmatic scope adjustment.
- Reconciled monthly sales data from Excel across multiple locations against Compeat records, flagging discrepancies and inputting corrections.
- Cleansed and validated raw multi-location Excel exports before loading into accounting systems, keeping data consistent across locations.
- Self-initiated the project during downtime, then presented it to the CEO and Director of Finance not as a technical "standardization tool" but as the franchise-wide per-dish sales they could now investigate - translating the work into business terms and securing their agreement to make the reports recurring.

SELECTED PROJECTS

Triage - Autonomous Incident Response Agent on AWS Bedrock AgentCore 2026

- Built an autonomous incident-response agent that watches CloudWatch alarms, reasons about failures, calls AWS observability tools through a **four-namespace custom MCP server (6 tools)**, and posts structured diagnoses to Slack. Writes are **Cedar-gated** at the AgentCore Gateway (6 active policies); immutable **S3 Object Lock** audit trail.
- Evaluated against a **9-scenario outage corpus** (4 AWS Fault Injection Service experiments + 5 Terraform misconfiguration overlays). **Scored Match on all 9 scenarios** (most-recent runs) via AgentCore Evaluations - 5 built-in evaluators plus 3 custom LLM-as-judge evaluators, including a MAST failure-taxonomy classifier.
- Mirrors the AWS DevOps Agent reference architecture (Molumuri et al., AWS DevOps Blog, March 31, 2026).

AWS (Bedrock AgentCore, ECS Fargate, Multi-AZ VPC, RDS, ALB, WAF, Route 53, CloudWatch, S3 Object Lock, IAM, ACM, FIS, Lambda, SNS, Secrets Manager), Terraform 1.14, Python 3.12, custom MCP server, Cedar policy engine, OpenTelemetry | Repo: github.com/Dimi-DV/triage

Production VPC in Terraform 2026

- Production-style VPC deployed via Terraform: 4 subnets across 2 AZs, internet gateway, multi-AZ NAT, custom route tables, layered security groups (18 resources). Remote state in S3 with DynamoDB locking.
- Validated with EC2 test instances exercising public to internet, private to NAT, and bastion (public to private) traffic patterns.

AWS VPC, Terraform, Multi-AZ NAT, public/private subnets, route tables, security groups, S3 + DynamoDB remote state | Repo: github.com/Dimi-DV/devops-learning

CI/CD Pipeline: GitHub Actions to ECR to ECS Fargate 2026

- End-to-end CI/CD: lint to test to build to push to ECR to deploy to ECS Fargate. Authenticated via OIDC (no long-lived AWS keys in GitHub Secrets).
- Multi-stage Dockerfile with a non-root user and container health checks; healthcheck-gated startup.

Docker, GitHub Actions, AWS ECR, ECS Fargate, ALB, OIDC | Repo: github.com/Dimi-DV/devops-learning

Customer Review NLP Pipeline 2023

- Scraped Amazon product reviews with a Selenium + BeautifulSoup pipeline (rotating user agents), storing structured output as JSON.
- Encoded reviews with Sentence-Transformers, clustered with KMeans, and extracted per-cluster keywords with TF-IDF to surface common customer grievances.
- Visualized high-dimensional review vectors in 2D with t-SNE/PCA and Matplotlib to map sentiment patterns.

Python, NLTK, Sentence-Transformers (Hugging Face), scikit-learn (KMeans), TF-IDF, Selenium, BeautifulSoup, t-SNE/PCA, Matplotlib

EDUCATION

Baruch College, CUNY (Zicklin School of Business) **- BBA Economics, Minor in Political Science** August 2019 - June 2023

Relevant coursework: Predictive Analytics & Decision Modeling, Business Statistics II, Statistical Analysis for Economics & Finance

TECHNICAL SKILLS

Cloud (AWS): VPC, EC2, EBS, S3, IAM, Security Groups, ECS Fargate, ECR, Lambda, RDS, ALB, Auto Scaling Groups, CloudWatch, Route 53, ACM, WAF, FIS, Secrets Manager, SNS, S3 Object Lock, Bedrock AgentCore

IaC: Terraform 1.14 (modules, remote state) **CI/CD:** GitHub Actions (matrix, OIDC, branch protection), Docker, ECR

Containers: Docker, Docker Compose, ECS Fargate, Kubernetes Basics

Languages: Python (boto3, pytest, requests, subprocess), Bash, HCL, SQL

Data & ML (Python): scikit-learn, NLTK, Transformers (Hugging Face), Sentence-Transformers, Pandas, NumPy, Matplotlib; KMeans, TF-IDF, t-SNE/PCA; A/B & hypothesis testing

Observability: CloudWatch (metrics, alarms, dashboards, Logs Insights), OpenTelemetry

OS/Networking: Linux (Ubuntu), TCP/IP, DNS, DHCP, subnetting, firewalls (iptables/ufw), HTTP/TLS, SSH, systemd, cron, process monitoring

Support & Ops: incident documentation, SLA-driven execution, troubleshooting & root-cause analysis, work-order tracking (informal), report writing

Client-facing & communication: stakeholder management, executive reporting, translating technical concepts for non-technical audiences, community engagement

Tools: Git (branch workflows, GitHub CLI), AWS CLI, VS Code, MS Office (Excel, Word, Outlook)

Languages (spoken): English (native), Serbian (native)