NISHANK KOUL

+91 9873446506 koulnishank5@gmail.com nishankkoul nishank-koul nishank

Summary

AWS Certified Cloud Practitioner with expertise in DevOps, specializing in automating infrastructure and delivering scalable cloud solutions.

Education

PES University - Bengaluru, India Dec 2021 - May 2025
Bachelor of Technology: Computer Science *Current CGPA: 7.41/10*

Sachdeva Public School - Delhi, India Sept 2021
Sr. Secondary School *XII (CBSE): 95.6*

Technical Skills

Languages: Python, Javascript, Bash
Cloud Platforms: Amazon Web Services (AWS), Google Cloud Platform (GCP)
CI/CD Tools: Jenkins, GitHub Actions
Containerization: Docker, Kubernetes
Monitoring: Prometheus, Grafana
Infrastructure as Code: Terraform, Ansible

Experience

Stringify AI | *DevOps Engineer* Feb 2025 - Present

- Containerized four cloud-native applications using **multi-stage Docker builds** and deployed it on Google Cloud Run; integrated **Cloud Load Balancer with CDN, reducing latency by 35%** and optimizing global response times.
- Provisioned a **production-grade PostgreSQL database on GCP Compute Engine** with SSH access tightly controlled through a bastion host; implemented **SSH tunneling on PgAdmin4** for secure local connectivity and visualization.
- Streamlined CI/CD processes using GitHub Actions, **accelerating deployment cycles by 40%** while ensuring consistency, reliability, and faster time-to-market.
- Created **custom events on Google Tag Manager** and integrated them with Google Analytics, enabling **precise tracking of user interactions and providing actionable insights** to optimize marketing strategies and improve conversion rates.

Bimaplan | *DevOps Engineer Intern* Sep 2024 - Feb 2025

- Developed Python scripts for AWS Lambda functions to automatically shut down EC2 instances in the Dev and UAT environments during non-business hours, leading to a **25%** reduction in overall cloud costs by optimizing resource utilization and minimizing idle time.
- Orchestrated **zero-touch deployment** by engineering Terraform scripts to replicate AWS infrastructure, **automating 90% of provisioning**. Additionally, established Disaster Recovery by replicating the infrastructure to a different region using the same Terraform scripts, ensuring business continuity.
- Executed the setup of a **read replica for the RDS Database** to enhance availability and scalability, **improving read query performance by 40%** and reducing downtime risks.
- **Refined Jenkins CI/CD pipelines** across Dev, UAT, and Prod by integrating Terraform, ensuring **100% consistency** in provisioning. **Established backup strategies** for pipeline code and statefiles, reducing rollback time by **60%**.
- Delivered an **efficient rate-limiting strategy for API Gateway** by analyzing historical traffic trends to improve performance and prevent abuse. Configured AWS CloudWatch alarms to monitor **HTTP 429 (Too Many Requests) errors** and integrated alerts with Slack for real-time monitoring and rapid incident resolution.

Projects

Celestia Validator Node Deployment on Mocha-4 Testnet | *Blockchain, Ansible, AWS EC2, Prometheus, Grafana* |

- Built an end-to-end Ansible playbook to **automate Celestia validator node** provisioning, **reducing manual setup time by 80%** and ensuring consistent deployments with zero configuration drift.
- Configured a **Grafana-based monitoring system with custom dashboards** to track node performance metrics, including block height, sync status, and resource utilization in real-time, enhancing operational visibility and reducing incident resolution time by **50%**.
- Developed industry-standard security protocols by applying encryption and access **restrictions for sensitive credentials using Ansible Vault** and designed rollback mechanisms, **reducing downtime risk by 30%** and improving validator resilience.

Scalable LLM Inference Service with Ollama | *LLMs, Flask, Docker, AWS EKS, K6.io, GitHub Actions* |

- Engineered a **scalable LLM inference service** using Ollama, integrating the moondream model. This involved containerization and API development, where a Dockerfile was built with Ollama as the base image, and a Flask API wrapper was created to interact with the model. The application was orchestrated on AWS Elastic Kubernetes Service (EKS) to ensure **high availability and scalability.**
- Accelerated application performance by identifying and resolving memory allocation bottlenecks during **Load Testing with K6.io, improving container accessibility and response times.**
- Executed auto-scaling strategies, increasing the successful request response rate from **53.66% to 85.49%.**

Certification

AWS Certified Cloud Practitioner Achieved in Oct 2024