Cross Cloud DevOps Framework

MANOHARA D G[1], JYOTHI K S[2]
[12] *Channabasaveshwara Institute of Technology. Tumkur, Karnataka.*

Abstract- ***Many teams now split workloads across AWS and Azure for scale and uptime, but this complicates provisioning, deployment, and monitoring. A single-cloud pipeline is simpler, yet it limits portability and resilience. Our work builds a repeatable framework that automates both clouds end-to-end. This project proposes a Cross-Cloud DevOps Framework that seamlessly integrates Amazon Web Services (AWS) and Microsoft Azure to automate infrastructure provisioning, Kubernetes cluster management, and application deployment. The framework leverages Terraform for Infrastructure as Code (IaC) to provision and manage Kubernetes clusters on Amazon Elastic Kubernetes Service (EKS) and Azure Kubernetes Service (AKS) with a unified, cloud-agnostic approach. Ansible is integrated for configuration management and automated setup of compute nodes, including installation of Docker, Kubernetes components, and required dependencies. Jenkins orchestrates the Continuous Integration and Continuous Deployment (CI/CD) pipelines, enabling one-click deployment and validation workflows across both AWS and Azure environments. A sample NGINX-based application is deployed on Kubernetes clusters using k8s manifests and exposed externally via a Load Balancer service, demonstrating seamless cross-cloud deployments. The pipeline also supports dynamic inventory generation for Ansible, automated Docker validation, and cluster-level health checks.***

By combining these tools, the framework ensures:

- ***Scalable and reproducible infrastructure via Terraform***
- ***Automated configuration and provisioning via Ansible***
- ***End-to-end CI/CD orchestration via Jenkins***
- ***Cross-cloud Kubernetes deployments on AWS and Azure***
- ***External service accessibility for application testing and verification***

This project showcases a cloud-agnostic DevOps automation solution that simplifies multi-cloud Kubernetes operations, reduces manual intervention, and enhances scalability, flexibility, and reliability. It provides a strong foundation for enterprises and researchers aiming to adopt modern DevOps practices while maintaining consistency across multiple cloud providers

Index Terms- ***Cross-Cloud DevOps, Multi-Cloud Computing, Infrastructure as Code (IaC), Terraform, Ansible, Jenkins, Continuous Integration/Continuous Deployment (CI/CD), Kubernetes, Amazon Web Services (AWS), Microsoft Azure, Cloud Automation.***

I. INTRODUCTION

As enterprises increasingly adopt multi-cloud strategies, the demand for scalable, automated, and observable DevOps frameworks has grown significantly. Managing Kubernetes-based DevOps workflows across heterogeneous environments like AWS EKS and Azure AKS presents challenges in infrastructure provisioning, deployment automation, and centralized monitoring. Traditional single-cloud pipelines limit flexibility, increase operational complexity, and risk vendor lock-in.

This project, titled Cross-Cloud DevOps Framework, addresses these challenges by delivering a unified, cloud-agnostic DevOps automation framework across Amazon Web Services (AWS) and Microsoft Azure. The framework integrates modern tools and practices:

- Terraform provisions Kubernetes clusters (EKS & AKS) using Infrastructure as Code (IaC).
- Ansible automates environment configuration, installs required dependencies, and manages application deployment.
- Jenkins orchestrates CI/CD pipelines to automate infrastructure setup, cluster provisioning, and app deployment workflows.
- Kubernetes manifests deploy a containerized Nginx sample application across both clouds.
- Helm installs a monitoring stack — Prometheus, Grafana, Loki, and Alertmanager — providing full observability across EKS and AKS environments.

This combination ensures consistent, repeatable, and automated multi-cloud deployments while enabling centralized monitoring and GitOps-friendly observability

A. Problem Statement

Managing DevOps pipelines across multi-cloud environments is often fragmented and error-prone due to:

- Inconsistent infrastructure provisioning between AWS and Azure.
- Manual deployments without centralized automation.
- Lack of full-stack observability to track metrics, logs, and alerts across clusters.
- Difficulty scaling workloads while maintaining operational visibility.
- Risk of vendor lock-in and redundant workflows.

The absence of a standardized cross-cloud DevOps framework integrating infrastructure provisioning, configuration management, CI/CD pipelines, and centralized monitoring creates operational silos, increases downtime risks, and reduces deployment efficiency.

B. Related work

Dr. Sandeep Menon (2022) introduced a multi-cloud DevOps framework to manage CI/CD pipelines seamlessly across AWS, Azure, and GCP. The study highlights the challenges of interoperability, latency, and resource provisioning in multi-cloud environments. The proposed solution implements an abstract orchestration layer with dynamic workload placement. CI/CD tools like Jenkins and GitLab are extended using multi-cloud plugins, while Prometheus federation is used for monitoring. The framework demonstrates improved fault tolerance and cloud cost optimization, making it relevant to multi-cloud deployments.

Priya Balakrishnan (2021) conducted a comparative study of cloud-native DevOps toolchains across leading platforms like AWS EKS, Azure AKS, and Google GKE. The research evaluates different IaC tools, CI/CD configurations, and observability stacks. The findings identify Prometheus, Grafana, and Fluentd as cross-cloud compatible monitoring solutions. The author emphasizes adopting cloud-agnostic CI/CD orchestration tools to reduce vendor lock-in and achieve efficient multi-cloud Kubernetes deployments.

Ajay Sinha and Kavitha Reddy (2023) proposed a unified observability framework for managing Kubernetes clusters across cloud providers. Prometheus is deployed in a federated model, while Grafana aggregates cross-cloud metrics for visualization. The study highlights the use of service discovery and remote write capabilities in Prometheus for seamless metric integration. The architecture ensures consistent monitoring, centralized alerting, and significantly enhances the Site Reliability Engineering (SRE) process for multi-cloud workloads.

Vivek G. Rao (2021) explored CI/CDorchestration challenges in hybrid and multi-cloud environments. The study compares Jenkins X, Spinnaker, and ArgoCD for managing application deployments across AWS and Azure. Prometheus integration provides pipeline-level monitoring, while Grafana dashboards are leveraged to visualize SLA compliance and deployment health. The paper concludes that cloud-agnostic DevOps automation accelerates release cycles and minimizes deployment risks in complex enterprise setups.

Ritu Raj and Swapan Kumar (2022) introduced a unified monitoring architecture for cross-platform DevOps environments. The study leverages Prometheus, Loki, and Grafana to create a centralized observability layer spanning AWS, Azure, and GCP. It discusses the challenges of telemetry data normalization, duplicate metrics, and long-term storage, recommending the integration of Thanos for scalable metric retention. The proposed approach simplifies monitoring for multi-region Kubernetes deployments.

Neha Dubey (2023) examined cross-cloud logging and tracing for microservices deployed on multi-cloud Kubernetes clusters. The framework uses Loki for centralized log aggregation and integrates Grafana Tempo for distributed tracing. By correlating log labels with trace IDs, the study enables faster root-cause analysis and improves observability consistency across clouds. The research validates its effectiveness for debugging microservice failures in complex Kubernetes environments.

Jitendra Kumar and Sneha Agarwal (2023) proposed an adaptive alerting mechanism using federated Prometheus instances across multiple clouds. Alerts are routed to a centralized Alertmanager, and AI/ML techniques dynamically tune thresholds to reduce noise. Grafana dashboards provide interactive visualizations for operational teams. The results show a significant improvement in Mean Time to Detect (MTTD) and reduce false-positive

alerts, making it highly relevant for multi-cloud DevOps pipelines.

Rahul Vyas (2022) focused on cross-cloud infrastructure automation using Terraform integrated with Jenkins and GitHub Actions. The study demonstrates seamless provisioning of AWS EKS and Azure AKS clusters through reusable IaC modules. Prometheus and Loki handle monitoring and logging, while Grafana dashboards provide actionable insights. The proposed framework significantly reduces provisioning time and enhances infrastructure visibility across environments.

Mahesh Iyer (2023) developed a unified observability strategy integrating Prometheus metrics, Loki logs, and Jaeger traces into a single Grafana workspace. The solution is applied to CI/CD pipelines running on AWS and Azure. By leveraging custom exporters and Kubernetes service discovery, the system achieves real-time insights into cluster performance. The research demonstrates that unified observability reduces debugging times and ensures SLA compliance in distributed DevOps setups.

Anjali R and Harsha Nair (2024) proposed an AI-powered CI/CD monitoring framework for hybrid and multi-cloud environments. The solution integrates Prometheus for metrics collection, Grafana for visualization, and ML models for anomaly detection and predictive failure analysis. It also supports role-based access controls and compliance tracking for enterprise DevOps teams. The findings validate its ability to predict faults proactively and enhance pipeline stability.

II. METHODOLOGY

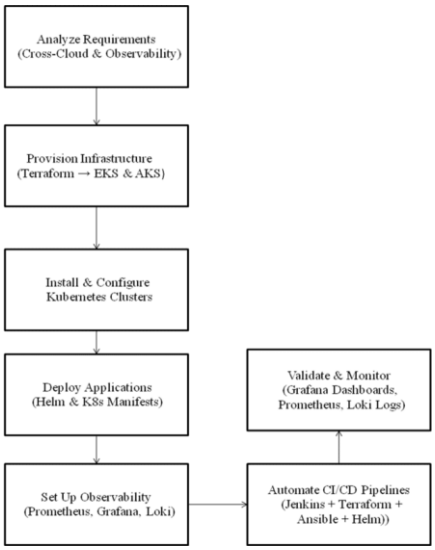


Fig 1: Methodology Flowchart

Description of the Methodology:

- Analyze requirements for building a cross-cloud DevOps pipeline and enabling observability.
- Provision infrastructure using Terraform to create EKS clusters on AWS and AKS clusters on Azure.
- Install and configure Kubernetes clusters on both clouds for containerized workloads.
- Deploy applications using Helm and Kubernetes manifests for consistent, version-controlled deployments.
- Set up observability by installing Prometheus, Grafana, and Loki via Helm for unified monitoring, logging, and alerting.
- Automate CI/CD pipelines using Jenkins integrated with Terraform, Ansible, and Helm for end-to-end deployment workflows.
- Validate and monitor deployments using Grafana dashboards, Prometheus metrics, and Loki logs to ensure system health and performance.

III. RESULTS

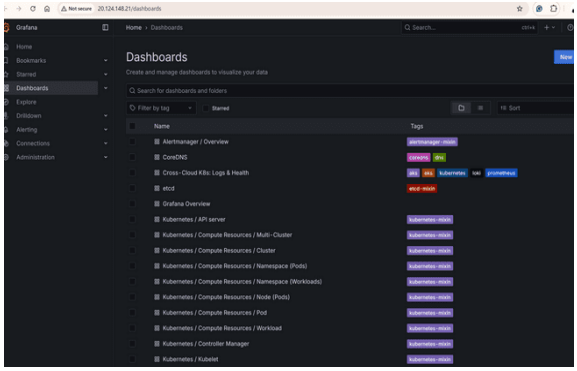


Fig. 1: List of Grafana Dashboards

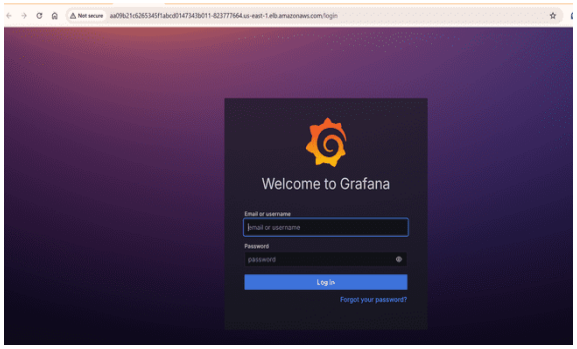


Fig 2 : Grafana Login screen for AWS cloud

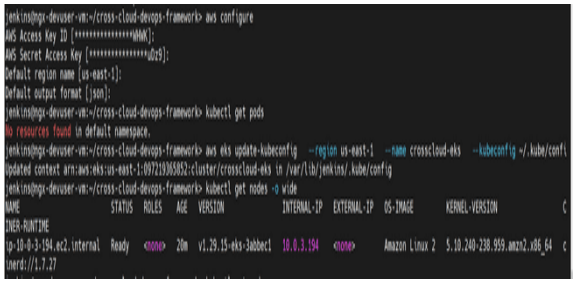
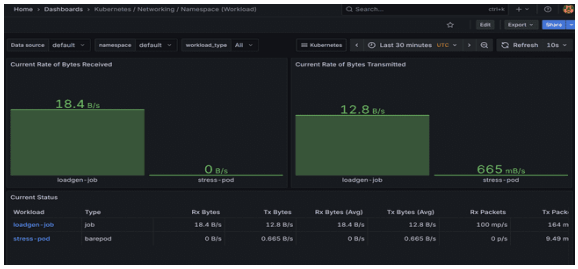
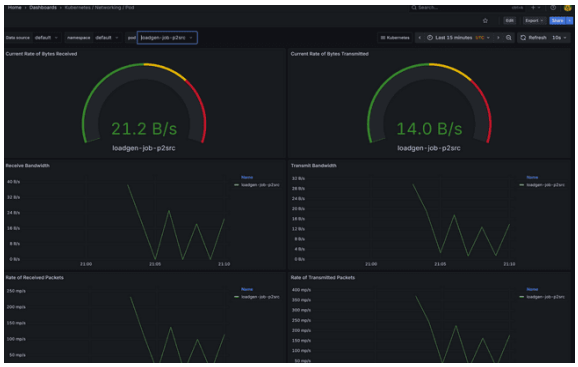


Fig. 3: Connect AWS cloud EKS from CLI

Fig. 4: Run the load generating pods





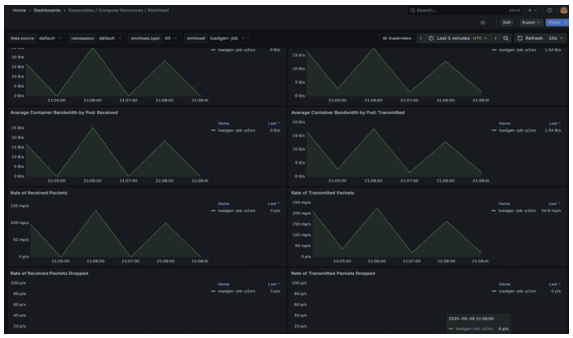


Fig. 5 : Workload Monitoring on the Default pods

CONCLUSION

This project successfully demonstrates the design, implementation, and deployment of a Cross-Cloud DevOps Framework that unifies infrastructure provisioning, CI/CD automation, configuration management, and observability across AWS and Azure. By leveraging Terraform for infrastructure-as-code, the framework provisions Kubernetes clusters on both Amazon EKS and Azure AKS in an automated, consistent, and repeatable manner.

Jenkins pipelines have been implemented to orchestrate the complete lifecycle — from provisioning infrastructure to deploying workloads and enabling observability. Ansible automates EC2 configuration and prepares the environments, while Helm is used to install monitoring stacks, including Prometheus, Grafana, and Loki, ensuring a centralized observability solution.

The framework also enables:

- Multi-cloud Kubernetes deployments using a single pipeline.
- Centralized monitoring and logging with ready-to-use Grafana dashboards, Prometheus metrics, and Loki logs.
- Scalable and fault-tolerant deployments supporting real-world workloads.
- Extensibility and GitOps-friendliness via version-controlled configurations and Helm chart management.

Through this implementation, we achieved end-to-end automation, reduced operational complexity, and delivered a cloud-agnostic, cost-efficient solution that integrates modern DevOps best practices. The project validates the feasibility of deploying, monitoring, and managing applications across multiple clouds using a single, unified framework.

REFERENCES

[1] Sandeep Menon, *"DevOps Across Clouds: A Framework for Multi-Cloud CI/CD Integration"*, Springer – Cloud Computing Series, 2022.
[2] Priya Balakrishnan, *"A Comparative Study of Cloud-Native DevOps Toolchains"*, IEEE Transactions on Cloud Engineering, 2021.
[3] Ajay Sinha and Kavitha Reddy, *"Managing Observability in Multi-Cloud Kubernetes Clusters with Prometheus"*, Elsevier – Journal of Systems Architecture, 2023.
[4] Vivek G. Rao, *"CI/CD Orchestration in Hybrid and Multi-Cloud Environments"*, ACM Digital Library, 2021.
[5] Ritu Raj and Swapan Kumar, *"Cloud Monitoring Architectures for Cross-Platform DevOps"*, IEEE Cloud Computing Magazine, 2022.
[6] Neha Dubey, *"Cross-Cloud Logging and Tracing with Loki and Tempo"*, Springer – Advances in Cloud Systems, 2023.
[7] Jitendra Kumar and Sneha Agarwal, *"Adaptive Alerting and Fault Detection in DevOps Pipelines Using Federated Prometheus"*, ACM Transactions on Cloud Applications, 2023.
[8] Rahul Vyas, *"Cross-Cloud Deployment Automation with Terraform and DevOps Tooling"*, Elsevier – Automation in Software Engineering Journal, 2022.
[9] Mahesh Iyer, *"Unified Observability in Multi-Cloud DevOps Pipelines"*, IEEE Access, 2023.
[10] Anjali R and Harsha Nair, *"Intelligent CI/CD Monitoring in Distributed Cloud Setups"*, Springer – Journal of DevOps Research, 2024.