National College of Ireland

Design and Implementation of a single CI/CD Pipeline to Deploy MultiCloud Kubernetes Clusters via Terraform and Jenkins

MSc Research Project
Cloud Computing

Jeena George
Student ID: x23379944

School of Computing
National College of Ireland

Supervisor: Abubakr Siddig

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Jeena George
Student ID:	x23379944
Programme:	Cloud Computing
Year:	2025-2026
Module:	MSc Research Project
Supervisor:	Abubakr Siddig
Submission Due Date:	28/01/2026
Project Title:	Design and Implementation of a single CI/CD Pipeline to Deploy MultiCloud Kubernetes Clusters via Terraform and Jenkins
Word Count:	1155
Page Count:	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jeena George
Date:	28/01/2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Design and Implementation of a single CI/CD Pipeline to Deploy MultiCloud Kubernetes Clusters via Terraform and Jenkins

Jeena George
x23379944

1 Requirements

The usage of the Multi-Cloud CI/CD Pipeline based on the Kubernetes (GKE and EKS) demanded a number of tools, cloud services, and development environments. This configuration explains how to go about configuring the entire multi-cloud environment and provisioning Kubernetes clusters, installing the application, and configuring to the rest, failover, and monitoring. The installation is separated into two large categories:

(1) Terraform Multi-Cloud Infrastructure Deployment.

(2) Grafana CI/CD Deployment and monitoring or Jenkins, Cloudflare and Prometheus.

1.1 Multi-Cloud Infrastructure (GCP And AWS)

In this research project, Google Cloud Platform (GCP) and Amazon Web Services (AWS) were set up and prepared to run Kubernetes clusters (GKE and EKS). Access credentials were developed on the individual cloud providers, and the Terraform was selected as the common Infrastructure-as-Code (IaC) tool to deploy resources in both environments in a similar fashion.

GCP:GCP:To segregate the workloads of Kubernetes in Google Cloud Platform (GCP) envir- onment,Challita et al. (2018) a new project was developed, and the APIs required, including Kubernetes Engine API and Compute Engine API were enabled to allow provisioning. capabilities within the clusters. Kubernetes Engine describes its roles as the one in Terraform automation. Admin, Compute Admin and Service Account User was then created and the JSON key was stored securely to be authenticated to delegate the pipeline to Jenkins.

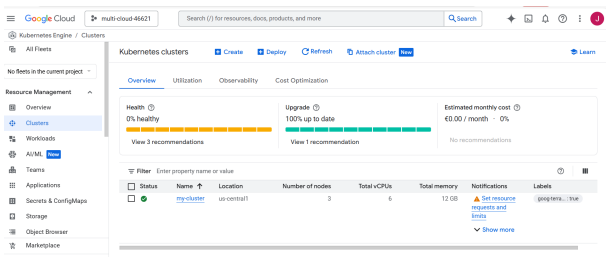


Figure 1: GCP Dashboard

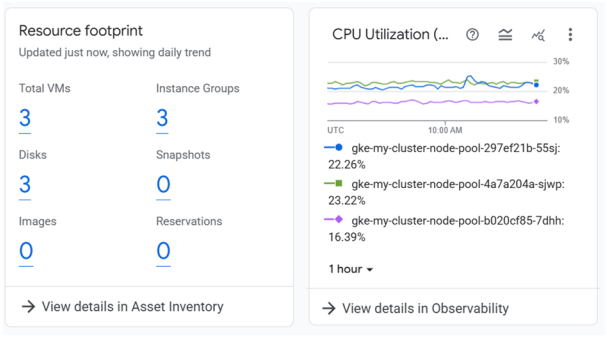


Figure 2: GKE Compute Engine

AWS: The environment used in terms of Amazon Web Services (AWS) consisted of establishing the required identity and infrastructure set-ups critical in deploying an EKS cluster via TerraformBorra (2024). AdministratorAccess IAM user was created so that the programmatic provisioning is possible and the Access Key ID and Secret Key were obtained so that the integration with Jenkins is possible and secure. Both control plane and the worker node group IAM roles were configured to guarantee that the clusters would operate correctly. The AWS CLI was then configured by installing the AWS configure command to create authentication to do the Terraform and deployment processes.

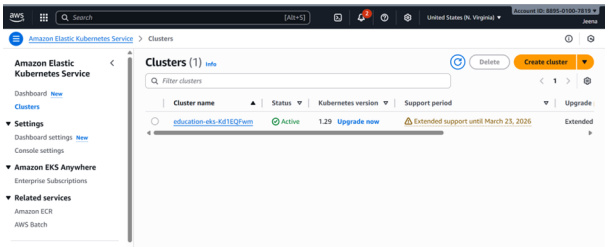


Figure 3: AWS Dashboard

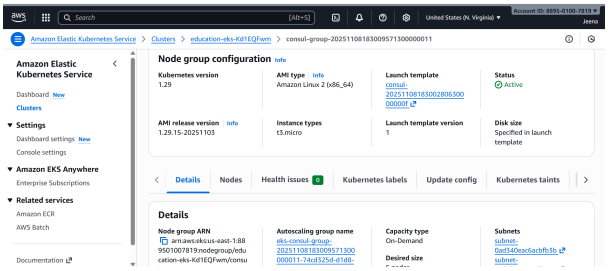


Figure 4: EKS Nodegroup Setting

1.2 Terraform Environment Setup

Vendor-neutral and repeatable multi-cloud architecture was constructed with terraform. This project has been discussed in the work with by the Morris (2016). The local system was set up using Terraform. Jenkins Credentials Manager accepts the credentials of AWS and GCP. A reusable body of It was developed into Terraform which is composed of VPC networking providers, GKE provisioning. utilities, EKS Infrastructure provisioning module and application deployment. Once these modules are configured, and Terraform

was applied using the standard commands such as terraform init, terraform plan and terraform apply. These, which were cluster, were produced. Then names and API endpoints and paths to kubeconfig were read using Jenkins is envisioned to have kubectl being automatically configured.

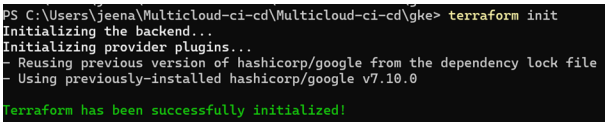


Figure 5: Terraform init

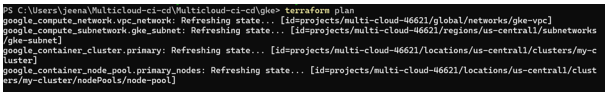


Figure 6: Terraform Plan

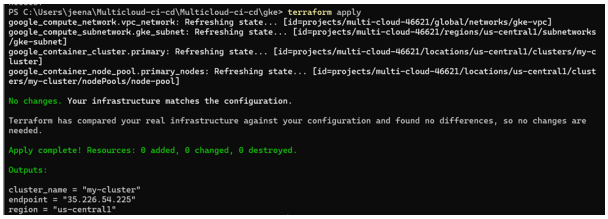


Figure 7: Terraform Apply

1.3 CI/CD Setup Using Jenkins

Jenkins tool is used for the CI/CD pipeline, and it automated the processes of infrastructure provisioning via Terraform, configuration of Kubernetes contexts, both GKE and EKS,and the application deployment via kubectl.In order to facilitate this automation, Jenkins was set up using necessary plug-ins like Pipeline, Terraform, and Credentials Binding and a declarative Jenkinsfile which forms all pipeline stepsArmenise (2015). The use build with parameters was also introduced to enable users to choose the selected deployment environment to be used: GCP, AWS or both, and the desired action to be used in Terraform such as plan, validate, apply or destroy. The deployment of the GCP service account JSON and AWS access keys on the Jenkins Credentials Manager was done to guarantee secure cloud authentication. Under this configuration, Jenkins offers a fully automated, repeatable and consistent deploy multiple cloud Kubernetes infrastructure.

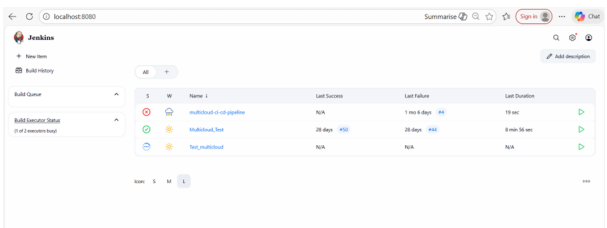


Figure 8: Jenkin Dashboard

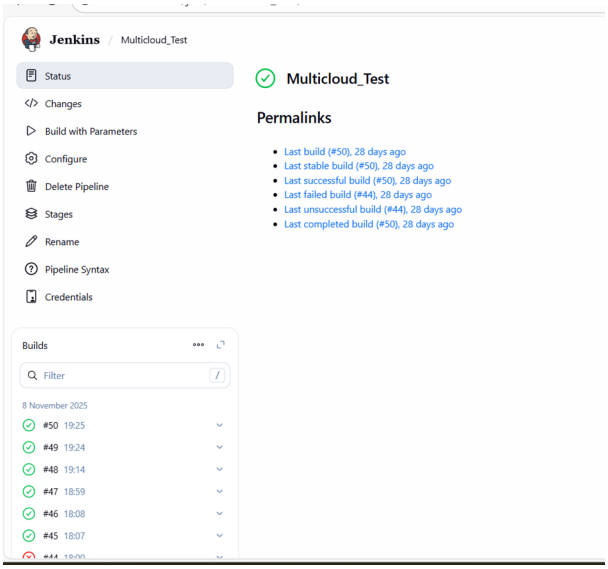
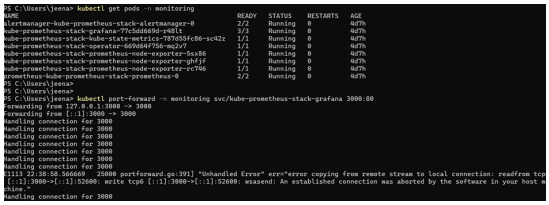


Figure 9: The ability to build with parameters where users can choose GCP, AWS, or Both

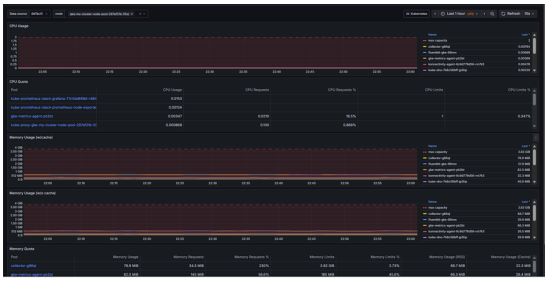
Figure 10: Output of Jenkins console confirming the successful running of CI /CD pipeline.

1.4 Monitoring And Failover Setup

In order to attain visibility of operations and ensure high availability of the multi-cloud architecture, PrometheusTurnbull (2018), Grafana, and Cloudflare Workers were set up as part of the central monitoring and failover. Helm chart kube-prometheus-stack was installed on the GKE cluster to allow monitoring metrics which had to be constantly collected on the cluster nodes, pods, kubectl, and Kubernetes API server. All these metrics were presented to visualize them in Grafana dashboardsSiddiqui et al. (2023) that contained information on the CPU and memory resources, the state of pods, and network performance. In order to achieve high availability, a custom Cloudflare Worker script was used to check the health on a regular basis, whereby GKE was the primary and EKS the secondary endpoint. In case the primary cluster fails, the worker will automatically reroute to EKS and reroutes back to GKE as soon as it comes back online. This integration guarantees a clear-cut failover, less service impediment, and tough multi-cloud routing.



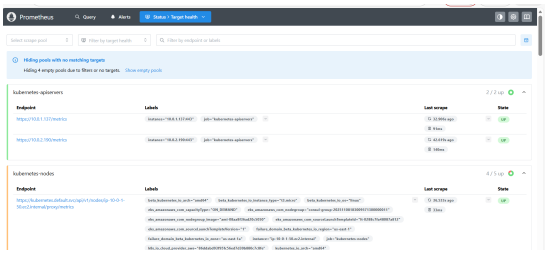
(a) Monitoring pods in GKE



(b) Grafana visualization for GKE

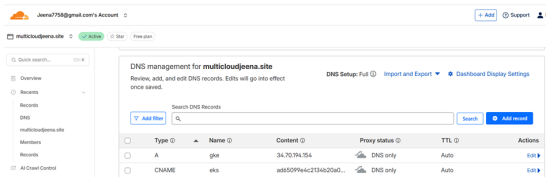
Figure 11: GKE-Monitoring Using Grafana

(a) EKS Port forwad to access small prometheus

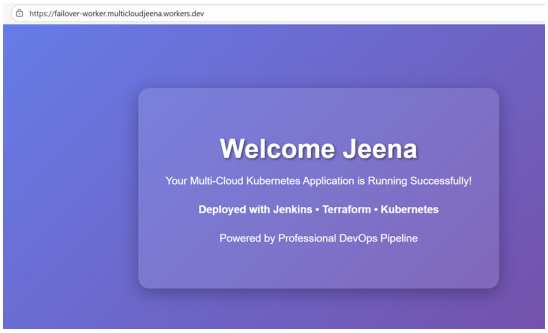


(b) EKS monitoring.

Figure 12: Prometheus dashboard for eks monitoring



(a) Cloudflare Dashboard.



(b) mplement The cloudflare worker script. between automated multi-cloud failover between GKE and EKS

Figure 13: Cloudflare dashboard settings of routing rules and worker on. traffic failover integration.

1.5 Application Deployment Testing.

Status of the pods and the functionality of the cluster were checked by using kubectl.

Figure 14: Using the pods in the GKE cluster Running The application pods in the cluster were checked with kubectl get pods.

Figure 15: Service specifications that include the LoadBalancer external IP that contains the external application access.

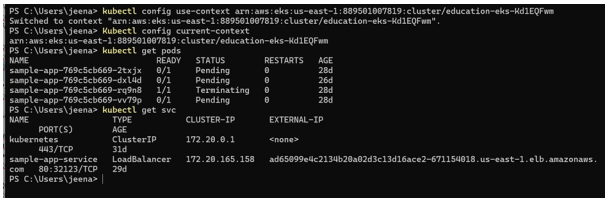


Figure 16: Using the pods in the EKS cluster Running The application pods in the cluster were checked with kubectl get svc.

2 GitHub Repository

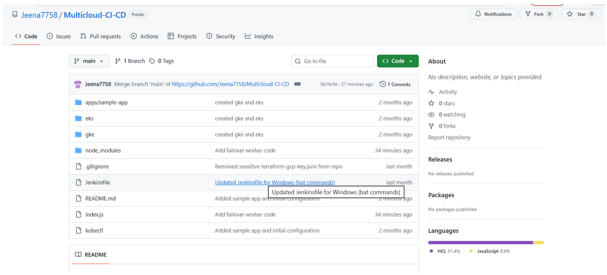


Figure 17: GitHub Repository

GitHub Repository URL: https://github.com/Jeena7758/Multicloud-CI-CD.git

References

Armenise, V. (2015). Continuous delivery with jenkins: Jenkins solutions to implement continuous delivery, *2015 IEEE/ACM 3rd International Workshop on Release Engineering*, IEEE, pp. 24–27.

Borra, P. (2024). Comprehensive survey of amazon web services (aws): techniques, tools, and best practices for cloud solutions, *International Research Journal of Advanced Engineering and Science* **9**(3): 24–29.

Challita, S., Zalila, F., Gourdin, C. and Merle, P. (2018). A precise model for google cloud platform, *2018 IEEE international conference on cloud engineering (IC2E)*, IEEE, pp. 177–183.

Morris, K. (2016). *Infrastructure as code: managing servers in the cloud*, " O'Reilly Media, Inc.".

Siddiqui, I., Pandey, A., Jain, S., Kothadia, H., Agrawal, R. and Chankhore, N. (2023). Comprehensive monitoring and observability with jenkins and grafana: a review of integration strategies, best practices, and emerging trends, *2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, pp. 1–5.

Turnbull, J. (2018). *Monitoring with Prometheus*, Turnbull Press.