BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ)
I SEMESTER 2025-2026
ASSIGNMENT

Course No.:	**CS/SS G527**	**Course Title:**	**Cloud Computing**
Deadline:	**As on Nalanda**	**Maximum Marks:**	**60 (15%)**

Note:
- **Group of maximum four students**
- **Submit deliverables on Nalanda (only one per group)**
- **Demonstration <u>may</u> be conducted after submissions.**

In this problem, you will implement a cloud application with at least 6 microservices using (multiple) cloud provider environment, out of which at least one being a web service accessible over public URL.

Following requirements must be met.

a. All cloud infrastructure (networks, clusters, databases, storage, IAM policies, etc.) across all cloud providers must be provisioned exclusively using an Infrastructure as Code (IaC) platform (e.g., Terraform).

b. Chose any domain such as education, healthcare etc. Your application must consist of at least six microservices in that domain, each serving a distinct functional purpose. Data analytical service (1 out of 6) must run on a different cloud provider (B). Include at least one serverless function (e.g., AWS Lambda, Google Cloud Functions, or Azure Functions) to perform an asynchronous, event-driven task. Microservices must communicate using cloud-native mechanisms such as REST, gRPC, or message queues (e.g., Kafka, AWS SQS, Google Pub/Sub).

c. Use a managed K8s service (e.g., EKS / GKE / AKS) to host stateless microservices. Implement Horizontal Pod Autoscalers (HPAs) for at least two critical services, configured to scale based on CPU or memory utilization.

d. All Kubernetes deployments and application updates must be managed via a GitOps controller (e.g., ArgoCD, Flux). The controller must track a Git repository. Direct kubectl apply is forbidden for service deployment.

e. One of your services must be a real-time stream processing service (Flink) running on the managed cluster (e.g., Google Dataproc, AWS EMR, or Azure Synapse/HDInsight) in Provider B. This service must consume event from a Kafka topic. It must perform a stateful, time-windowed aggregation (e.g., "count of unique users per 1-minute window"). The

aggregated results must be published back to a separate 'results' Kafka topic. The Kafka cluster itself must be a managed cloud service (e.g., MSK, Confluent Cloud, Aiven).

f. Use <u>at least one</u> distinct cloud storage products. For example: an object store (e.g., S3, GCS) for raw data (e.g., file uploads that trigger the serverless function), a managed SQL database (e.g., RDS, Cloud SQL) for relational data (e.g., user accounts, structured metadata), a managed NoSQL database (e.g., DynamoDB, Firestore, MongoDB Atlas) for high-throughput, semi-structured data (e.g., session state, real-time analytics results).

g. Implement a comprehensive observability stack. Metrics: Deploy Prometheus and Grafana. Create a dashboard showing key service metrics (RPS, error rate, latency) and Kubernetes cluster health. Logging: Implement a centralized logging solution (e.g., EFK stack, Loki, or Cloud Logging) to aggregate logs from all microservices, including the analytics job.

h. Use a load testing tool (e.g., k6, JMeter, Gatling) to validate your system's resilience and scalability. Load Test: Generate sustained traffic to demonstrate the HPA scaling out your services.

Deliverables:

*   Design document containing System overview, Cloud deployment architecture, Microservices architecture digrams, microservice responsibilities, interconnection mechanisms, and the rationale behind design choices in the cloud.
*   Microservices code (in Github repository), all IaC script code, all Kubernetes manifests and GitOps configuration
*   Each student, for their part, should record a video with his/her own voice explaining the different sections in source/script code or configuration they have written <u>with <idno> visible in the terminal or code</u>. Video link must be present in <idno>_video.txt. Based on video, marks will be allotted to individuals. 12M.
*   A video demonstrating end to end working of your application including the testing phase must be recorded with explanation. Video link must be present in demo_video.txt

[60M]

--&--