COMSATS University Islamabad, Lahore Campus

☑ Lab Final Examination   FALL 2025

Course Title:	DevOps for Cloud Computing		Course Code:	CSC418	Credit Hours:	3(2,1)	Programme: BCS
Course Instructor:	Dr. Hasan Jamal				Date:	16/12/2025	
Semester:	7th	Batch:	FA22	Section:	G1	Time Allowed:	3 Hours
Student's Name:		Reg. No.			Maximum Marks:	50	

Important Instructions / Guidelines:

- Direct reuse of prior submissions will lead to disqualification.
- All CI/CD logs and Docker images must be time-stamped within the exam period.
- Hardcoded AWS keys or passwords = automatic deduction (10 marks)
- Terraform plan copied or not functional = automatic deduction (10 marks)
- No Kubernetes manifests = automatic deduction (10 marks)
- Only screenshots, no repo or working code = automatic deduction (10 marks)
- All group members must have visible contributions (commits/PRs).
- Submit your solution on Google Classroom before the given deadline.

Question: [50 Marks]

CLO: <5> Develop cloud native applications using current DevOps tools: Bloom Taxonomy Level: <Creating>

Objectives:

1. **Design, containerize,** and **automate** an open-source app with DB + cache/message queue.
2. **Provision infrastructure** using Terraform on AWS.
3. **Deploy** and **manage** the app using Kubernetes (EKS or Minikube).
4. **Automate configuration** using Ansible.
5. **Implement CI/CD** using Jenkins and/or GitHub Actions.
6. **Monitor** the system with Grafana & Prometheus.
7. **Deliver a full production-ready DevOps stack.**

Step 1 – Project Selection and Containerization

Extend the same project that was assigned to you for your midterm exam. The project must include database and cache/message queue. It should be extensible to microservices/Kubernetes. It must have the following:

1. Dockerfile (optimized, multistage)
2. Docker-compose.yml (for local testing).
3. Container networking verified.
4. Persistent storage for DB.
5. No hardcoded secrets.

Step 2 – Infrastructure Provisioning with Terraform **[10 Marks]**

Goal: Automate AWS setup. Terraform must provision at least:

- **VPC + Subnets + Security Groups**
- **EKS Cluster** (Kubernetes) or EC2 instances (fallback option)
- **RDS/PostgreSQL** or **S3 bucket** for persistence

Deliverables:

- `infra`/ folder with `.tf` files.
- Outputs (`terraform output`) showing provisioned resources.
- Screenshot of AWS Console with created resources.
- `terraform destroy` proof (cleanup).

Step 4 – Configuration Management (Ansible) **[05 Marks]**

Goal: Configure servers or containers automatically. You can use Ansible to:

- Install dependencies on EC2s or Kubernetes nodes.
- Deploy app configs or secrets.
- Automate Docker or K8s deployment.

Deliverables:

- `ansible/playbook.yaml`
- Inventory file (`host.ini` or dynamic AWS inventory).
- Screenshot of successful playbook run.

Step 5 – Kubernetes Deployment **[10 Marks]**

Goal: Deploy the containerized app to Kubernetes (EKS or local Minikube). Required Manifest include:

- `deployment.yaml`
- `service.yaml`
- `configmap.yaml` or `secret.yaml`
- (Optional) `ingress.yaml` or `helm/` chart.

Expected:

- App pod communicates with DB (service)
- Redis/Queue as sidecar or separate deployment
- Namespace organization (`dev, prod`)
- Screenshot: `kubectl get pods, kubectl get svc, kubectl describe pod <name>`

Step 6 – CI/CD Pipeline (Jenkins or GitHub Actions) **[10 Marks]**

Goal: Fully automated multi-stage pipeline.

1. Build & test
2. Security/ linting
3. Docker build and push
4. Terraform apply (infrastructure provisioning)
5. Ansible deploy or `kubectl apply`
6. Post-deploy smoke tests

Deliverables:

- `Jenkinsfile` or `.github/workflows/main.yml`
- Pipeline screenshot with all stages passed.

Step 7 – Monitoring & Observability (Grafana + Prometheus) **[10 Marks]**

Goal: Integrate monitoring for app and DB performance.

Requirements:

- Prometheus collects metrics (can be from app or node-exporter)
- Grafana dashboard visualizes key metrics
- Include screenshots of dashboards (CPU, memory, request count, etc.)

Step 8 – Documentation and Report **[5 Marks]**

Each group must include two documents:

1. `README.md`:
    - How to run locally, via Compose, and via K8s
    - Infra setup and teardown instructions
2. `devops_report.md`:
    - Technologies used
    - Pipeline + infrastructure diagram
    - Secret management strategy
    - Monitoring strategy
    - Lessons learned