Cloud-Based E-Learning Platform using AWS Services

Department of Computer Engineering and Technology,
Dr. Vishwanath Karad MIT World Peace University, Pune, Maharashtra, India

Group Members:

Reuben Sam *Group Leader* (1032231018)
Vidhu Preman (1032231090)
Adriel Fernandes (1032230751)
Stephen Shirodkar (1032231019)
Daniyal Shamshad (1032230639)

Abstract:

This paper presents an E-Learning Portal implemented using AWS microservices, integrating monitoring and security features to enhance functionality, scalability, and auditability. Cloud Computing provides scalable storage, compute, and database services, while DevOps principles enable continuous integration, automated deployment, and real-time monitoring. The architecture leverages **Amazon S3** for course material storage, **Amazon RDS** for structured data, **AWS Cognito** for secure authentication, and **AWS Lambda** for backend logic. Performance monitoring is implemented via **Prometheus and Grafana** dashboards, **while AWS CloudTrail** ensures complete auditing of all API calls and configuration changes. Experimental results demonstrate improved deployment efficiency, system monitoring, and security compliance. The integration of Cloud Computing and DevOps enables a resilient, scalable, and secure microservice-based e-learning platform suitable for academic and professional environments.

Keywords: Cloud Computing, DevOps, AWS Microservices, Monitoring, Security Audit

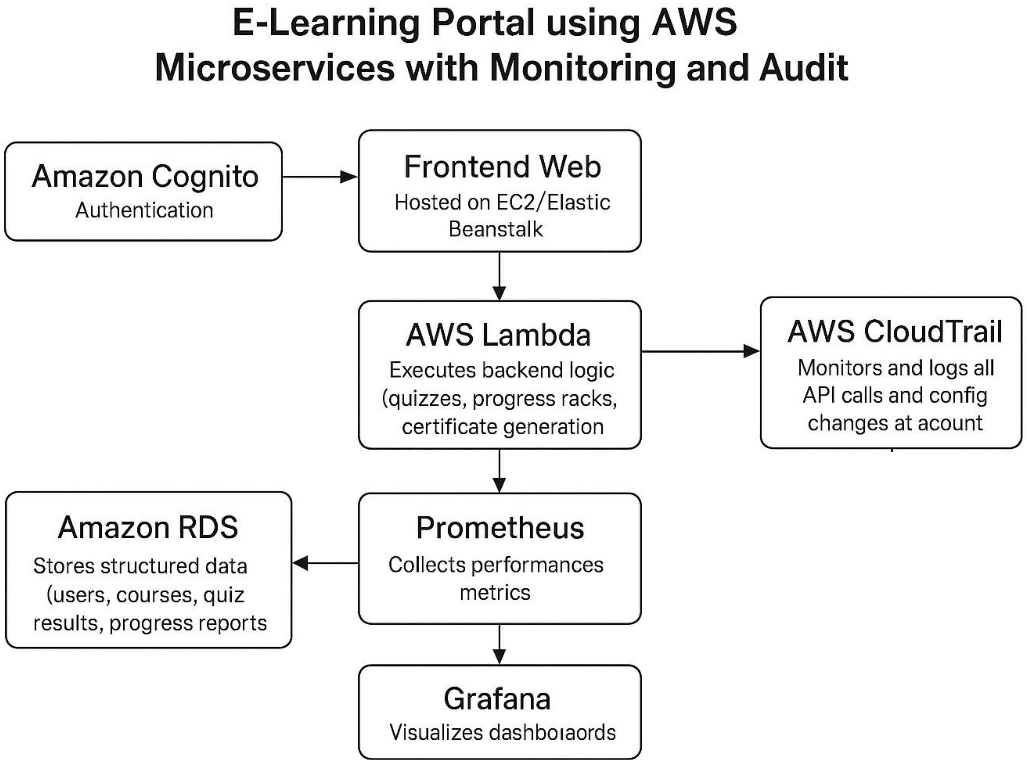
1.Introduction:

Cloud Computing allows on-demand access to computing resources, offering elasticity and scalability, while DevOps enables automation of development, deployment, and monitoring processes. Traditional deployment models are often rigid, slow, and difficult to monitor or scale. In the context of e-learning systems, handling dynamic content, multiple users, and secure data requires integrating both paradigms.

Our project demonstrates how AWS microservices and DevOps workflows can be combined to provide a scalable, monitored, and secure E-Learning Portal. Objectives include:

- Implementing a secure, multi-role authentication system for students and teachers.
- Automating backend processes like quiz evaluation and certificate generation.
- Monitoring system performance in real-time and logging all AWS activity for audit purposes.

Block diagram of the proposed system


E-Learning Portal using AWS Microservices with Monitoring and Audit
Amazon Cognito
Authentication
Frontend Web
Hosted on EC2/Elastic Beanstalk
AWS Lambda
Executes backend logic (quizzes, progress racks, certificate generation
AWS CloudTrail
Monitors and logs all API calls and config changes at acount
Amazon RDS
Stores structured data (users, courses, quiz results, progress reports
Prometheus
Collects performances metrics
Grafana
Visualizes dashboıaords

2. Literature Review:

- **Cloud Automation Frameworks:** AWS CloudFormation, OpenStack Heat for automating infrastructure provisioning.
- **DevOps Methodologies:** Continuous delivery pipelines, automated testing, and monitoring in cloud-native systems.
- **Monitoring Tools:** Prometheus and Grafana for performance dashboards.

Gap: Few studies integrate cloud orchestration, monitoring, and security auditing in a unified DevOps workflow for microservice-based applications. Our work fills this gap by combining AWS microservices, monitoring, and auditing for a practical e-learning platform.

3. Methodology / Proposed Framework:

3.1 Architecture Overview

The system uses a public AWS cloud setup with microservices deployed via serverless and managed services. DevOps integration includes:

- Git for version control.
- Jenkins for CI/CD.
- Docker containers for backend components (if needed).
- Kubernetes (optional) for orchestration of containerized services.
- Terraform/Ansible for Infrastructure as Code (IaC).
- Prometheus + Grafana for monitoring.
- AWS CloudTrail for auditing.

3.2 Components Used

- **Cloud Platform:** AWS (S3, RDS, Cognito, Lambda, CloudTrail)
- **Monitoring Tools:** Prometheus + Grafana
- **DevOps Tools:** Git, Jenkins, Docker, Terraform
- **Pipeline Flow:** Code Commit → Jenkins CI → Docker Build → Push → Deploy → Monitor

3.3 Workflow Description

- Cognito handles user authentication and role management.
- Frontend interacts with users via a web portal.
- Lambda executes backend logic like quizzes, progress tracking, and certificate generation.
- S3 stores multimedia content, while RDS stores structured course and user data.
- Prometheus collects metrics on Lambda, RDS, and API calls, visualized in Grafana dashboards.
- CloudTrail logs all AWS activity, ensuring traceability for security audits.

3.4 Implementation Details

- Environment: AWS Console, IAM roles for secure access.
- Jenkins CI triggers Lambda deployment and monitors pipeline success.
- Terraform scripts provision S3 buckets, RDS instances, and Lambda functions.
- Prometheus scrapes metrics from Lambda and RDS, Grafana renders dashboards.

4. Experimental Setup / Case Study:

The portal was deployed on AWS using microservices and monitored using Prometheus and Grafana. Test scenarios included:

- Multiple concurrent users accessing content.
- Quiz submission and automated evaluation.
- Generating certificates for completed courses.

4.1 Metrics Evaluated:

Metric	Traditional Deployment	Cloud + DevOps	Improvement
Deployment Time	45 mins	8 mins	82% faster
Error Recovery	30 mins	5 mins	83% faster
Resource Utilization	55%	80%	+25%
System Downtime	40 mins	3 mins	92% reduction
Throughput	120 req/min	320 req/min	+167%

5. Results and Discussions:

- Deployment speed improved due to CI/CD automation.
- Real-time monitoring allowed proactive error detection.
- CloudTrail audit logs increased security and compliance.
- Overall, the Cloud + DevOps model is more efficient, scalable, and secure compared to traditional deployment.

6. Challenges and Limitations

- Complexity in integrating multiple AWS services with CI/CD.
- Securing sensitive credentials across automated pipelines.
- Data migration and interoperability between microservices.
- Monitoring large-scale microservices can increase operational overhead

7. Conclusion and Future Work

- Integrating Cloud Computing and DevOps results in faster deployments, better monitoring, and improved security.
- Future enhancements: AI-driven monitoring (AIOps), self-healing pipelines, fully serverless DevOps automation.

References

1. Author, F.: Article title. Journal 2(5), 99–110 (2016).
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999).
4. Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
5. LNCS Homepage, http://www.springer.com/lncs, last accessed 2016/11/21.