

# AI Developer Assessment - Nithish Kumar B

## Cloud-Native E-commerce Recommendation System

### Assessment Overview

**Candidate:** Nithish Kumar B

**Duration:** 7 days

**Difficulty Level:** Intermediate

**Focus Areas:** DevOps, Cloud Technologies, MLOps

**Specialization:** Cloud-based AI/ML Systems

### Candidate Profile

- **CGPA:** 8.2/10
- **Key Strengths:** DevOps, Azure DevOps, AWS, Kubernetes, Cloud Architecture
- **Technical Stack:** Python, JavaScript, Node.js, Docker, Terraform, Ansible
- **Recent Work:** Cloud-based platforms, EEG Analysis with LSTM

### Core Requirements

#### 1. Cloud-Native Architecture Design (30 points)

##### Leveraging your cloud expertise:

- Design microservices architecture for recommendation system
- Implement containerization with Docker and Kubernetes
- Set up CI/CD pipelines using Azure DevOps or GitHub Actions
- Configure auto-scaling based on demand
- Implement load balancing and fault tolerance

#### 2. MLOps Pipeline Implementation (25 points)

##### Building on your DevOps experience:

- Model training pipeline with automated retraining
- Model versioning and experiment tracking
- A/B testing framework for model deployment
- Monitoring and alerting for model performance
- Automated rollback mechanisms

### 3. Multi-Cloud Deployment Strategy (20 points)

#### Utilizing your AWS and Azure knowledge:

- Deploy on both AWS and Azure platforms
- Implement data synchronization across clouds
- Cost optimization strategies
- Disaster recovery and backup solutions
- Performance comparison analysis

### 4. Real-time Analytics and Monitoring (15 points)

#### Infrastructure monitoring focus:

- Real-time recommendation serving with <100ms latency
- System metrics and performance monitoring
- User interaction tracking and analytics
- Resource utilization optimization
- Automated scaling based on traffic patterns

### 5. Security and Compliance (10 points)

#### Enterprise-grade security:

- Identity and access management
- Data encryption in transit and at rest
- API security and rate limiting
- Compliance with data protection regulations
- Security vulnerability scanning

### Technical Specifications

#### Required Technologies:

- **Cloud Platforms:** AWS (EC2, S3, Lambda, ECS), Azure (AKS, Blob Storage)
- **Containerization:** Docker, Kubernetes, Helm
- **CI/CD:** Azure DevOps, GitHub Actions, Jenkins
- **Infrastructure as Code:** Terraform, Ansible
- **Monitoring:** Prometheus, Grafana, ELK Stack
- **ML Serving:** TensorFlow Serving, MLflow, Kubeflow

Infrastructure Components:

- **API Gateway:** Kong or AWS API Gateway
- **Message Queues:** Redis, Apache Kafka
- **Databases:** MongoDB Atlas, PostgreSQL on RDS
- **Caching:** Redis Cluster, CloudFront CDN
- **Logging:** Centralized logging with ELK/EFK stack

Deliverables

1. **Infrastructure as Code** (Terraform scripts)
2. **Kubernetes Manifests** for deployment
3. **CI/CD Pipeline** configuration files
4. **Monitoring Dashboard** (Grafana)
5. **Performance Benchmarking** report
6. **Cost Analysis** and optimization recommendations

Evaluation Criteria

Component	Weight	Key Metrics
Cloud Architecture	30%	Scalability, fault tolerance, cost efficiency
MLOps Pipeline	25%	Automation level, deployment speed, reliability
Multi-Cloud Setup	20%	Deployment success, performance consistency
Monitoring & Analytics	15%	Real-time capabilities, comprehensive metrics
Security Implementation	10%	Security best practices, compliance adherence

Performance Requirements

- **Availability:** 99.9% uptime
- **Latency:** <100ms for recommendations
- **Throughput:** Handle 10k+ concurrent users
- **Scalability:** Auto-scale from 2 to 50 instances
- **Cost:** Optimize for <\$500/month operational cost

Bonus Challenges (+25 points)

- **Serverless Architecture:** Implement with AWS Lambda/Azure Functions
- **Edge Computing:** Deploy recommendation models to edge locations
- **Data Lake Integration:** Build data pipeline with AWS S3/Azure Data Lake
- **Machine Learning at Scale:** Implement distributed training with Kubernetes

- **Advanced Monitoring:** Implement predictive alerting and anomaly detection

## Cloud-Specific Tasks

### AWS Implementation:

- ECS/EKS for container orchestration
- Lambda for serverless components
- CloudFormation for infrastructure
- CloudWatch for monitoring

### Azure Implementation:

- AKS for Kubernetes deployment
- Azure Functions for event-driven processing
- ARM templates for resource management
- Azure Monitor for observability

## Timeline

- **Day 1:** Infrastructure design and Terraform scripts
- **Day 2:** Kubernetes setup and containerization
- **Day 3:** CI/CD pipeline configuration
- **Day 4:** Model serving and MLOps implementation
- **Day 5:** Multi-cloud deployment and testing
- **Day 6:** Monitoring, security, and optimization
- **Day 7:** Performance testing and documentation

## Submission Requirements

- **GitHub Repository:** Complete infrastructure code
- **Live Deployments:** Working systems on both AWS and Azure
- **Monitoring Dashboards:** Real-time system metrics
- **Technical Documentation:** Architecture diagrams and runbooks
- **Demo Video:** End-to-end system demonstration
- **Cost Analysis Report:** Detailed cost breakdown and optimization

**Assessment Aligned With:** Your DevOps expertise and cloud platform experience