Senior Backend Developer Assignment — Unite

Objective:

Evaluate the candidate's ability to build a production-ready backend system with AWS, Node.js/TypeScript, CI/CD, security, scalability, and real-world development practices.

Mandatory Technologies:

EC2 deployment, S3 image storage, SNS system, Twilio SMS, MySQL + MongoDB, Node.js + Express (TypeScript), AWS CodePipeline + CodeBuild, DigitalOcean backend server, CI/CD deployment, CRUD REST APIs, Docker + Redis.

1. Core Requirements (Must Deliver)

1.1 Authentication & Roles

- Implement JWT or AWS Cognito authentication.
- Roles: admin, manager, agent.
- Token lifecycle strategy (refresh or documented revocation).
- Secrets via environment variables or AWS Secrets Manager.
- Must use Node.js + Express + TypeScript.

1.2 Lead Management

- CRUD for leads: name, phone, email, status, source, assigned_to.
- Validation + duplicate prevention.
- Filtering: status, source, assigned_to.
- Image upload to S3 via pre-signed URL.
- Store data in MySQL and logs/metadata optionally in MongoDB.

1.3 Call Task Management

- Create and complete call tasks.
- Agents add notes and outcomes.
- Must include idempotency checks.
- Immutable call logs.
- On task assignment/completion: trigger SNS notification + Twilio SMS.

1.4 CSV Upload + Async Processing

- CSV upload to S3.
- Async processing (Lambda/SQS/worker).

- Validate rows, create leads, generate error report.
- Use MySQL for main data and MongoDB for logs.

1.5 Reporting

- Daily summary: total calls, completed, missed, per-agent.
- Insights: completion %, busiest agent, etc.
- One endpoint must use MongoDB aggregation.

2. Infrastructure Requirements

2.1 IaC – Terraform or CloudFormation

- API hosting: ECS, Lambda, or EC2 (required).
- S3 buckets for CSV + images.
- RDS MySQL + MongoDB config notes.
- SQS/EventBridge for async tasks.
- IAM least-privilege policies.

2.2 Deployment & DevOps

- Production Dockerfile.
- docker-compose: app, MySQL, MongoDB, Redis, worker.
- GitHub Actions pipeline: lint → test → build → deploy.
- AWS CodePipeline + CodeBuild CI/CD.
- Deployment docs for EC2 + DigitalOcean.
- Blue/green or canary strategy notes.

3. Observability & Resilience

- Structured logging with correlation IDs.
- Error tracking (Sentry or similar).
- CloudWatch metrics + alarms.
- Retry/backoff for external services.
- Circuit breaker strategy.
- Log SNS + Twilio responses.

4. Testing Requirements

- Unit tests (≥75% coverage).

- Integration tests via Docker.
- Load testing (k6 or artillery).
- Failure simulations (DB down, Redis fail).
- Tests for S3 uploads, SNS, and Twilio (mocked).

5. Deliverables Checklist

- Git repo with commit history.
- README with architecture, deployment steps, CI/CD, env vars.
- Architecture diagram (PNG/SVG).
- Terraform/CloudFormation templates.
- Dockerfile + docker-compose.
- GitHub Actions workflow.
- OpenAPI/Postman collection.
- Tests + coverage report.
- Load test scripts + results.
- sample_leads.csv (100 rows).
- Logs showing S3 upload + SNS + Twilio.

6. Evaluation Criteria (100 Points)

- Functionality – 30
- Architecture & AWS Design – 25
- DevOps & Deployment – 15
- Reliability & Observability – 15
- Code Quality & Testing – 10
- Documentation – 5
- Bonus +10 for live deploy + advanced AWS patterns.

Candidate Prompt:

“Develop a production-ready Node.js (TypeScript) backend for Unite with RBAC auth, lead management, call tasks, CSV import, reporting, SNS notifications, Twilio SMS, Redis caching, Docker, EC2 deployment, DigitalOcean deployment, CI/CD via AWS CodePipeline + CodeBuild. Deliver infra code, documentation, tests, and load-tested results.”