

Senior DevOps Engineer — In home test

Candidate Instructions

You have 3 days to complete the assignment.

Deliverables:

- A written technical document (PDF, Markdown or Google Doc) explaining your solutions and trade-offs.
- A short presentation deck (5–8 slides) summarizing the design choices.
- All code (Terraform, YAML, scripts) in a compressed folder or Git repo link.

Section A — Kubernetes (Core Skill)

Task A1 — Deploy a microservice to Kubernetes:

You are given frontend, backend and postgres components.

Deliver production-ready Kubernetes manifests (or Helm chart) including Deployments, Services, Ingress, ConfigMap, Secrets, resource requests/limits, readiness/liveness probes, HPA, PodDisruptionBudget and NetworkPolicies. Explain your choices for security, scalability and rollout strategy.

Task A2 — Debug a broken cluster:

Scenario: Pods stuck in CrashLoopBackOff, Service not reachable, Ingress returns 502, one node in NotReady (DiskPressure).

Deliver: Document exact troubleshooting steps you would run (kubectl commands, logs, node inspection), root cause analysis, and permanent fixes.

Section B — AWS (Cloud Engineering & Reliability)

Task B1 — Design a Highly Available Architecture in AWS:

Deliver an AWS architecture diagram (draw.io, mermaid or similar) showing VPC with public/private subnets, ALB, ASG or EKS nodes, RDS/Aurora, ElastiCache, NAT Gateways, CloudWatch alerts, S3 + CloudFront and IAM least-privilege roles. Include HA strategy, DR strategy, logging & monitoring, and cost optimization notes.

Task B2 — Fix AWS Infra Issues:

Five scenarios to solve: Internet access from private EC2, S3 AccessDenied on uploads, Lambda cannot reach RDS, app loses DB during ASG scale events, CloudWatch not collecting logs. For each explain root cause and exact fix (console/API/terraform commands if relevant).

Task B3 — Build a CI/CD pipeline for AWS:

Provide a pipeline (GitHub Actions / GitLab CI / Jenkins) that builds Docker images, runs tests, pushes to ECR, deploys to EKS/ECS, uses IaC and environment promotion (dev→stage→prod) with protected environments.

Section C — Infrastructure as Code (Terraform)

Task C1 — Create Terraform for AWS:

Deliver Terraform modules: vpc/, eks/, nodegroups/, iam/, rds/. Include variables with validation, modules, outputs and remote state configuration. Provide README with usage instructions and example workspaces.

Task C2 — Troubleshoot a Broken Terraform Deployment:

Given errors like 'cycle detected', 'IAM role missing permissions', and 'resource address has changed', explain causes and step-by-step fixes, including state inspection and addressing drift.

Section D — Observability & Monitoring

Task D1 — Build a Logging & Monitoring Strategy:

Include CloudWatch logs & metrics, Prometheus + Grafana, OpenTelemetry, alerting strategy (incidents/SEV definitions), log retention and indexing plan, and example dashboards.

Task D2 — Fix Latency Issues:

Given API latency increased (40ms→800ms), CPU/memory normal, DB load high, cache hit ratio <10%, 10% 5xx errors — provide root cause analysis and remediation plan (caching strategy, DB indexing, throttling, circuit breaker, autoscaling recommendations).

Section E — System Design (DevOps Architecture)

Task E1 — Design a Zero-Downtime Deployment Strategy:

Document options (Blue/Green, Canary, Rolling, A/B, traffic splitting via ALB/Route53) and pick one with justification for a microservice architecture.

Task E2 — Secure the entire system:

Provide IAM least-privilege examples, multi-account strategy, secrets management approach (AWS Secrets Manager/KMS), Kubernetes RBAC, network restrictions (SGs/NACLs), PodSecurity and CI/CD security controls (image scanning, signing).

Section F — Documentation & Presentation

Task F1 — Deliver a Professional Technical Document:

Document should have structure: Problem → Approach → Solution → Result, diagrams, code snippets, troubleshooting steps, risk analysis, and future improvements.

Task F2 — Deliver a Presentation Deck (5–8 slides):

Include system summary, key decisions & trade-offs, AWS design summary, Kubernetes design summary, reliability enhancements and final recommendations.

Submission Checklist

Include a README at the repo root explaining how to run Terraform, deploy manifests, and validate the solution. Provide any credentials or test accounts if necessary in a secure manner (one-time tokens or instructions to create test resources).