DevOps Interview Prep Sheet

Top 50 questions with production answers

Linux & Networking (10)

Q: What happens when you type a URL?

A: DNS > TCP > TLS > HTTP request > server > response > render. Mention CDN, LB.

Q: Troubleshoot a slow server?

A: top (CPU), free -m (memory), iostat (disk), ss (connections), logs. Narrow: CPU/mem/IO/network bound.

Q: TCP vs UDP?

A: TCP: reliable, ordered (HTTP, SSH). UDP: fast, best-effort (DNS, video, gaming).

Q: What is a reverse proxy?

A: Sits before backends. SSL termination, load balancing, caching. Nginx, HAProxy.

Q: How does DNS work?

A: Browser cache > OS > resolver > root > TLD > authoritative > IP. TTL controls cache.

Docker & Containers (10)

Q: Image vs container?

A: Image = read-only blueprint. Container = running instance. Class vs object.

Q: Multi-stage build?

A: Multiple FROM stages. Build in one, copy artifacts to slim final. Reduces size 5-10x.

Q: Reduce image size?

A: Multi-stage, alpine base, .dockerignore, combine RUN, remove caches.

Q: Compose vs Kubernetes?

A: Compose: single host, dev. K8s: multi-node, production, self-healing, scaling.

Q: Containers vs VMs?

A: Containers share kernel (fast, light). VMs full OS (heavier, more isolated).

CI/CD (10)

Q: Describe your pipeline.

A: Push > lint/test > build image > push ECR > deploy ECS > health check. Use OIDC, no keys.

Q: How do you rollback?

A: Previous task def/image tag. Blue-green = instant switch. Keep last 5 versions.

Q: Blue-green vs canary?

A: Blue-green: two envs, instant switch. Canary: 5% new, monitor, gradually increase.

Q: Secure CI/CD secrets?

A: OIDC (keyless), GitHub Secrets, AWS Secrets Manager, SSM Parameter Store.

Q: What is GitOps?

A: Git = source of truth. Push > ArgoCD syncs to cluster. Auditable, reversible.

AWS (10)

Q: VPC architecture?

A: Public (ALB, NAT), private (app), database (RDS) subnets. Multi-AZ. SG + NACLs.

Q: How did you cut costs?

A: Right-sizing, gp2>gp3, scheduling dev, RIs, CDN offloading. Quote specific %.

Q: S3 vs EBS vs EFS?

A: S3: objects, unlimited. EBS: block, single EC2. EFS: shared, multiple EC2s.

Q: IAM least privilege?

A: Min permissions needed. Specific ARNs, conditions. IAM Access Analyzer. Never root.

Q: Auto Scaling?

A: CloudWatch alarm > policy > launch/terminate. Target tracking (60% CPU) simplest.

Terraform (5)

Q: State file?

A: Maps code to real infra. S3 + DynamoDB lock. Never in git.

Q: Modules?

A: Reusable components (VPC, RDS, ECS). Use for anything repeated across envs.

Q: Secrets in Terraform?

A: Never in .tf. Use Secrets Manager, SSM, env vars. Mark sensitive=true.

Q: Plan shows destroy?

A: STOP. Check renames. Use moved blocks or import. Never blind apply.

Q: Workspaces vs state files?

A: Separate state per env is cleaner. Workspaces for simple setups.

Kubernetes (5)

Q: Pod vs Deployment vs Service?

A: Pod: container group. Deployment: replicas + updates. Service: stable endpoint.

Q: Self-healing?

A: Liveness probes restart bad containers. ReplicaSet keeps pod count. Scheduler replaces.

Q: CrashLoopBackOff?

A: kubectl logs, describe pod. Check: image, ports, env vars, health checks.