## ✅ Mock Interview Q&A – DevOps Cloud Engineer (2 YOE)

---

### ☁️ Cloud (AWS)

**Q1. Explain the difference between Security Groups and NACLs in AWS.**
✅ **Answer:**

- **Security Groups**: Instance-level firewall, **stateful** (inbound allows response).
- **NACLs**: Subnet-level, **stateless** (need explicit inbound + outbound rules).
  → Use SGs for fine-grained access, NACLs for subnet-wide rules.

---

**Q2. How would you design a highly available web application in AWS?**
✅ **Answer:**

- Use **ALB** with EC2s in multiple AZs.
- **Auto Scaling Group** for resilience.
- **RDS Multi-AZ** or Aurora.
- **S3 + CloudFront** for static content.
- **Route 53** for DNS + health checks.

---

**Q3. Differences between S3 Standard, S3 IA, and Glacier.**
✅ **Answer:**

- **S3 Standard** → Frequent access, low latency.
- **S3 IA (Infrequent Access)** → Cheaper, for data not accessed often.
- **Glacier** → Archival storage, cheapest, retrieval takes minutes-hours.

---

**Q4. How do you implement hybrid connectivity between on-prem and AWS?**
✅ **Answer:**

- **Site-to-Site VPN** for quick setup.
- **Direct Connect** for dedicated, low-latency connection.
- Use **Transit Gateway** for central hub.
- Ensure encryption + routing policies.

---

## 🔧 Infrastructure as Code

### Q5. How do you manage Terraform state in a team environment?

✅ **Answer:**

- Store state in **S3 bucket** (remote backend).
- Enable **DynamoDB table lock** to prevent conflicts.
- Use versioning and encryption.

---

### Q6. Difference between Terraform and Ansible.

✅ **Answer:**

- **Terraform**: Provision infra (VPCs, EC2, EKS). Declarative, idempotent.
- **Ansible**: Configure software (install packages, patch servers). Procedural.
- Often used together → Terraform builds infra, Ansible configures it.

---

### Q7. How do you organize Terraform modules for reusability?

✅ **Answer:**

- Create **separate modules** (VPC, EC2, EKS).
- Use **variables.tf** and **outputs.tf**.
- Store in Git repos and version them.
- Keep modules generic, not hardcoded.

---

## 🐳 Containers & Kubernetes

### Q8. Difference between Deployment and StatefulSet in Kubernetes.

✅ **Answer:**

- **Deployment**: For stateless apps, all pods identical.
- **StatefulSet**: For stateful apps (databases), pods have stable IDs, persistent storage.

---

### Q9. How do you secure pod-to-pod communication in EKS?

✅ **Answer:**

- Use **Network Policies**.
- Enable **mTLS with Istio**.
- Restrict access with **RBAC**.
- Integrate with AWS IAM via IRSA.

---

### Q10. What is IRSA in EKS and why is it used?

✅ **Answer:**

- **IAM Roles for Service Accounts**.
- Allows pods to assume **fine-grained IAM roles**.
- Removes need for storing AWS creds inside pods.

---

### Q11. How do you troubleshoot a pod stuck in CrashLoopBackOff?

✅ **Answer:**

- Run kubectl logs <pod>.
- Check kubectl describe pod.
- Common causes: misconfigured env vars, bad image, insufficient resources.
- Fix config/image → redeploy.

---

## 🔄 CI/CD & Automation

### Q12. Walk me through a CI/CD pipeline you built.

✅ **Answer:**

- Code pushed → GitHub Actions triggers build.
- Run tests → build Docker image → push to **ECR**.
- Deploy to **EKS** using Helm.
- Add approval gates for prod.

---

### Q13. How do you secure secrets in a pipeline?

✅ **Answer:**

- Store in **AWS Secrets Manager** or **SSM Parameter Store**.
- Use Jenkins credentials store or GitHub encrypted secrets.
- Never commit secrets to repo.

---

### Q14. Difference between Blue-Green and Canary deployments.

✅ **Answer:**

- **Blue-Green**: Two environments, switch traffic fully at once.
- **Canary**: Gradual rollout to a small % of users, then expand.

---

### Q15. How would you integrate Terraform with a pipeline?

✅ **Answer:**

- Add stages for terraform init, plan, apply.
- Store state remotely (S3 + DynamoDB).
- Use approvals for production apply.

---

## 🛠️ Monitoring & Logging

### Q16. What tools have you used for monitoring and alerting?

✅ **Answer:**

- **CloudWatch** (metrics, alarms).
- **Prometheus + Grafana** for K8s metrics.
- Alerting via **SNS / Slack / Email**.

---

### Q17. How would you set up centralized logging for multiple microservices in AWS?

✅ **Answer:**

- Use **Fluentd/FluentBit** to send logs to **CloudWatch Logs**.
- Or use **EFK (Elasticsearch, Fluentd, Kibana)**.
- Use log aggregation + filters for quick search.

---

## ⚡ General & Situational

**Q18. Tell me about a time you handled a failed deployment in production.**

✅ **Answer (STAR):**

- **Situation**: Deployment to EKS failed, app crashed.
- **Task**: Restore service quickly.
- **Action**: Rolled back via Helm to last stable version, checked logs, fixed misconfigured env var.
- **Result**: Restored service in <30 mins, updated pipeline validation to catch similar issues.

---

**Q19. What is the most challenging DevOps project you worked on?**

✅ **Answer:**

- Automated provisioning of EKS using Terraform + Ansible.
- Faced networking/RBAC issues, solved by modularizing Terraform code and implementing IRSA.
- Learned importance of automation + security-first design.

---

**Q20. If your CI/CD pipeline suddenly becomes very slow, how would you troubleshoot?**

✅ **Answer:**

- Identify bottleneck (build, test, deploy).
- Check build agents → increase resources/parallelism.
- Cache dependencies (e.g., Docker layers, Maven cache).
- Optimize long-running test suites.

---

## ⚡ How to Use

- Read question → Answer in your own words.
- Then compare with my **model answers**.
- For situational Qs (#18–20), always use **real examples** from your 2 years of work.