CloudOps Engineer – Detailed Concepts & Lab Demos

A CloudOps Engineer manages cloud infrastructure automation, monitoring, CI/CD pipelines, security, and operations. This document provides detailed concepts with hands-on lab demos ideal for real-world cloud operations using AWS and Kubernetes.

1. Cloud Fundamentals for CloudOps

Core responsibilities include: - Provisioning infrastructure using IaC (Terraform/CloudFormation). - Managing container platforms like Docker and Kubernetes. - Observability: Logging, monitoring, tracing. - Automation: CI/CD, GitOps, Event-driven automation. - Security: IAM, network security, secrets, compliance. - Cost Optimization: Analyzing cloud spend and resource utilization.

2. LAB 1: Create AWS VPC + Public/Private Subnets

Objective: Deploy a production-grade VPC. **Steps:** 1. Create VPC: CIDR 10.0.0.0/16 2. Create subnets: - Public Subnets: 10.0.1.0/24, 10.0.2.0/24 - Private Subnets: 10.0.3.0/24, 10.0.4.0/24 3. Attach an Internet Gateway (IGW). 4. Create NAT Gateway for private subnet. 5. Configure route tables for each subnet. **Validation:** - Launch EC2 in public subnet → SSH allowed. - Launch EC2 in private subnet → reachable only through public EC2.

3. LAB 2: Deploy an EC2 Linux Server & Configure CloudWatch Logs

Commands: \`sudo yum install awslogs -y\` Edit \`/etc/awslogs/awslogs.conf\` to include \`/var/log/messages\`. Start and enable service: \`sudo systemctl enable awslogsd\` \`sudo systemctl start awslogsd\` **Expected Outcome:** Logs appear in CloudWatch Log Group \`/aws/ec2/server-logs\`.

4. LAB 3: Create an EKS Cluster Using eksctl

Command: `eksctl create cluster --name cloudops-demo --region ap-south-1 --nodes 2` **Verify:** `kubectl get nodes` `kubectl get pods -A` Cluster should show two worker nodes.

5. LAB 4: Deploy NGINX App on EKS

Commands: `kubectl create deployment nginx --image=nginx` `kubectl expose deployment nginx --port=80 --type=LoadBalancer` **Outcome:** ELB created and accessible publicly.

6. LAB 5: Autoscaling Setup (HPA + Cluster Autoscaler)

Install metrics-server: `kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml` Autoscale: `kubectl autoscale deployment nginx --cpu-percent=50 --min=1 --max=10`

7. LAB 6: GitHub Actions CI/CD Pipeline for EKS

Example workflow `.github/workflows/deploy.yml`: ```yaml name: Deploy to EKS on: [push] jobs: deploy: runs-on: ubuntu-latest steps: - uses: actions/checkout@v2 - name: Configure AWS uses: aws-actions/configure-aws-credentials@v2 with: role-to-assume: arn:aws:iam::123456789:role/GitHubOIDCRole aws-region: ap-south-1 - name: Deploy to cluster run: | kubectl apply -f manifests/ ``` Outcome: Automatic deployment to EKS on every push.

8. LAB 7: Observability with Prometheus & Grafana

Install using Helm: `helm repo add prometheus-community https://prometheus-community.github.io/helm-charts` `helm install monitoring prometheus-community/kube-prometheus-stack` Access Grafana dashboard: - Port-forward: `kubectl port-forward svc/monitoring-grafana 3000:80` - Login: admin/prom-operator

9. LAB 8: Security Lab – IAM Roles + Secrets Encryption

Enable IRSA: `eksctl utils associate-iam-oidc-provider --cluster cloudops-demo --approve` Create IAM role for pod: `eksctl create iamserviceaccount --name s3-reader ...` Encrypt Secrets with KMS: Update EKS cluster config: `encryptionConfig: providers: - kms: keyArn: arn:aws:kms:...`